

Michael Kofler

*Aktuell zu Fedora, CentOS,  
RHEL, openSUSE, Debian,  
Ubuntu und Mint*

# Linux

Das umfassende Handbuch



- ▶ Das Standardwerk für Einsteiger und Fortgeschrittene
- ▶ Für Desktop und Server: Installation, Konfiguration, Administration
- ▶ Mit zahlreichen Praxistipps, Raspberry-Pi-Kapitel, »Linux on Windows«

15., aktualisierte Auflage

 Rheinwerk  
Computing

## Liebe Leserin, lieber Leser,

dieses Handbuch – vielen einfach als »der Kofler« bekannt – gilt bei seinen Lesern, in der Fachpresse und im Buchhandel als das Standardwerk zu allen Fragen rund um Linux. Seit mehr als zwanzig Jahren steht es für herausragende Qualität, umfassende Informationen und präzise Anleitungen.

Unabhängig davon, ob Sie das Buch schon aus früheren Auflagen kennen oder ob Sie es neu entdecken: Sie werden von den aktuellen und wertvollen Informationen profitieren. Als Einsteiger werden Sie schnell sehen, warum diese Buch Kultstatus genießt. Sowohl die inhaltliche Tiefe, mit der die Fülle der Themen behandelt wird, als auch die anschauliche Art der Erläuterung machen es einzigartig. Und als Kenner früherer Auflagen werden Sie zu schätzen wissen, wie sehr es am Puls der Zeit ist: Nicht nur die aktuellen Ausgaben aller wichtigen Linux-Distributionen werden berücksichtigt, auch weitere wichtige Themen kommen zur Sprache. In dieser Auflage werden die modernen Editoren *Atom* und *Visual Code* genauso behandelt wie neue Virtualisierungsthemen und *Let's Encrypt*. Sie sehen schon: Egal, ob Sie Linux als Entwicklungsumgebung einsetzen möchten, es als Basissystem für Server und Apps dienen soll oder Sie Ihre private Webseite absichern wollen: Michael Kofler liefert Ihnen die passenden Anleitungen und Hintergrundinfos! Als besonderen Abschluss erfahren Sie schließlich alles über das neue *Windows Subsystem für Linux*. Denn inzwischen hat auch Microsoft erkannt, wie nützlich viele Linux-Tools und die *bash* bei der täglichen Arbeit sein können – sie sind neuerdings in Windows 10 integriert und warten nur darauf, ausprobiert zu werden.

Abschließend noch ein Hinweis in eigener Sache: Dieses Buch wurde mit großer Sorgfalt geschrieben, geprüft und produziert. Sollte dennoch einmal etwas nicht so funktionieren, wie Sie es erwarten, freue ich mich, wenn Sie sich direkt mit mir in Verbindung setzen. Ihre Anregungen und Fragen sind jederzeit herzlich willkommen!

Nun bleibt mir nur noch, Ihnen viel Freude und Erfolg mit diesem Handbuch zu wünschen, mit dem bereits Generationen von »Linuxern« ihre ersten (und alle weiteren) Schritte in Linux unternommen haben.

**Ihr Christoph Meister**

Lektorat Rheinwerk Computing

[christoph.meister@rheinwerk-verlag.de](mailto:christoph.meister@rheinwerk-verlag.de)

[www.rheinwerk-verlag.de](http://www.rheinwerk-verlag.de)

Rheinwerk Verlag • Rheinwerkallee 4 • 53227 Bonn



# Hinweise zur Benutzung

Dieses E-Book ist **urheberrechtlich geschützt**. Mit dem Erwerb des E-Books haben Sie sich verpflichtet, die Urheberrechte anzuerkennen und einzuhalten. Sie sind berechtigt, dieses E-Book für persönliche Zwecke zu nutzen. Sie dürfen es auch ausdrucken und kopieren, aber auch dies nur für den persönlichen Gebrauch. Die Weitergabe einer elektronischen oder gedruckten Kopie an Dritte ist dagegen nicht erlaubt, weder ganz noch in Teilen. Und auch nicht eine Veröffentlichung im Internet oder in einem Firmennetzwerk.

Die ausführlichen und rechtlich verbindlichen Nutzungsbedingungen lesen Sie im Abschnitt *Rechtliche Hinweise*.

Dieses E-Book-Exemplar ist mit einem **digitalen Wasserzeichen** versehen, einem Vermerk, der kenntlich macht, welche Person dieses Exemplar nutzen darf:

Exemplar Nr. x72b-z8iv-sfnd-4k9u  
zum persönlichen Gebrauch für  
Uwe Hametner,  
uh@pidirect.com

# Impressum

Dieses E-Book ist ein Verlagsprodukt, an dem viele mitgewirkt haben, insbesondere:

**Lektorat** Christoph Meister

**Korrektorat** Friederike Daenecke, Zülpich

**Herstellung E-Book** Norbert Englert

**Covergestaltung** Mai Loan Nguyen Duy

**Satz E-Book** Michael Kofler

Wir hoffen sehr, dass Ihnen dieses Buch gefallen hat. Bitte teilen Sie uns doch Ihre Meinung mit und lesen Sie weiter auf den *Serviceseiten*.

Bibliografische Information der Deutschen Nationalbibliothek:  
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

**ISBN 978-3-8362-5855-5 (E-Book)**

**ISBN 978-3-8362-5858-6 (E-Book zum Buch)**

**ISBN 978-3-8362-5857-9 (Bundle)**

15., aktualisierte Auflage 2017 (3. Auflage im Rheinwerk Verlag)

© Rheinwerk Verlag GmbH, Bonn 2017

[www.rheinwerk-verlag.de](http://www.rheinwerk-verlag.de)

*Dieses Buch ist meiner Frau Heidi und  
meinen Kindern Sebastian und Matthias gewidmet.*





# Inhalt

|                      |    |
|----------------------|----|
| <b>Vorwort</b> ..... | 19 |
|----------------------|----|

## TEIL I Installation

---

|  |    |
|--|----|
| <b>1 Was ist Linux?</b> .....                          | 25 |
| 1.1 Einführung .....                                   | 25 |
| 1.2 Hardware-Unterstützung .....                       | 26 |
| 1.3 Distributionen .....                               | 28 |
| 1.4 Open-Source-Lizenzen (GPL & Co.) .....             | 32 |
| 1.5 Die Geschichte von Linux .....                     | 35 |
| 1.6 Software-Patente und andere Ärgernisse .....       | 36 |
| <b>2 Installationsgrundlagen</b> .....                 | 39 |
| 2.1 Voraussetzungen .....                              | 39 |
| 2.2 BIOS und EFI .....                                 | 41 |
| 2.3 Installationsvarianten .....                       | 44 |
| 2.4 Überblick über den Installationsprozess .....      | 47 |
| 2.5 Start der Linux-Installation .....                 | 49 |
| 2.6 Grundlagen der Festplattenpartitionierung .....    | 50 |
| 2.7 RAID, LVM und Verschlüsselung .....                | 57 |
| 2.8 Partitionierung der Festplatte .....               | 64 |
| 2.9 Installationsumfang festlegen (Paketauswahl) ..... | 70 |
| 2.10 Grundkonfiguration .....                          | 72 |
| 2.11 Probleme beheben .....                            | 75 |
| 2.12 Systemveränderungen, Erweiterungen, Updates ..... | 78 |
| 2.13 Linux wieder entfernen .....                      | 81 |
| <b>3 Installationsanleitungen</b> .....                | 83 |
| 3.1 CentOS .....                                       | 84 |
| 3.2 Debian .....                                       | 91 |

|     |                     |     |
|-----|---------------------|-----|
| 3.3 | Fedora .....        | 99  |
| 3.4 | Linux Mint .....    | 106 |
| 3.5 | openSUSE .....      | 110 |
| 3.6 | Ubuntu .....        | 120 |
| 3.7 | Ubuntu Server ..... | 131 |

## TEIL II Linux anwenden

---

|          |   |     |
|----------|---|-----|
| <b>4</b> | <b>Linux-Schnelleinstieg</b> .....                        | 139 |
| 4.1      | Linux starten und beenden .....                           | 139 |
| 4.2      | Tastatur, Maus und Zwischenablage .....                   | 141 |
| 4.3      | Umgang mit Dateien, Zugriff auf externe Datenträger ..... | 144 |
| 4.4      | Dokumentation zu Linux .....                              | 145 |
| <b>5</b> | <b>Gnome</b> .....  | 147 |
| 5.1      | Erste Schritte .....                                      | 148 |
| 5.2      | Dateimanager .....  | 153 |
| 5.3      | Systemkonfiguration .....                                 | 162 |
| 5.4      | Schriften (Fonts) .....                                   | 173 |
| 5.5      | Gnome Tweak Tool .....                                    | 174 |
| 5.6      | Gnome-Shell-Erweiterungen .....                           | 176 |
| 5.7      | Gnome Shell Themes .....                                  | 179 |
| 5.8      | Gnome-Interna .....                                       | 181 |
| 5.9      | Der Gnome-Klassikmodus .....                              | 184 |
| 5.10     | MATE .....  | 185 |
| 5.11     | Cinnamon .....  | 186 |
| <b>6</b> | <b>KDE und Unity</b> .....                                | 189 |
| 6.1      | KDE .....   | 190 |
| 6.2      | KDE-Dateimanager .....                                    | 196 |
| 6.3      | KDE-Konfiguration .....                                   | 199 |
| 6.4      | Unity .....   | 204 |
| <b>7</b> | <b>Desktop-Apps</b> .....                                 | 213 |
| 7.1      | Firefox .....   | 214 |
| 7.2      | Google Chrome .....                                       | 220 |
| 7.3      | Thunderbird .....   | 222 |
| 7.4      | Evolution, KMail und Geary .....                          | 229 |



|          |  |            |
|----------|--|------------|
| 7.5      | Dropbox .....  | 235        |
| 7.6      | FileZilla und BitTorrent .....                                   | 237        |
| 7.7      | Shotwell .....   | 238        |
| 7.8      | digiKam .....  | 240        |
| 7.9      | GIMP .....   | 242        |
| 7.10     | RawTherapee, Darktable und Luminance (RAW- und HDR-Bilder) ..... | 246        |
| 7.11     | Multimedia-Grundlagen .....                                      | 248        |
| 7.12     | Rhythmbox, AmaroK & Co .....                                     | 251        |
| 7.13     | Spotify .....  | 254        |
| 7.14     | VLC .....  | 255        |
| 7.15     | Audio- und Video-Tools .....                                     | 256        |
| 7.16     | Screenshots und Screencasts .....                                | 264        |
| <b>8</b> | <b>Raspberry Pi</b> .....  | <b>267</b> |
| 8.1      | Grundlagen .....   | 268        |
| 8.2      | Raspbian installieren und konfigurieren .....                    | 272        |
| 8.3      | Kodi und LibreELEC .....   | 283        |
| 8.4      | Hardware-Basteleien .....  | 298        |
| 8.5      | Interneta und Backups .....                                      | 317        |
| 8.6      | Wenn es Probleme gibt .....                                      | 324        |

## TEIL III Linux-Grundlagen

---

|           |   |            |
|-----------|---|------------|
| <b>9</b>  | <b>Terminalfenster und Konsolen</b> ..... | <b>329</b> |
| 9.1       | Textkonsolen und Terminalfenster .....    | 330        |
| 9.2       | Textdateien anzeigen und editieren .....  | 334        |
| 9.3       | man und info .....                        | 338        |
| <b>10</b> | <b>bash (Shell)</b> .....                 | <b>341</b> |
| 10.1      | Was ist eine Shell? .....                 | 341        |
| 10.2      | Basiskonfiguration .....                  | 343        |
| 10.3      | Kommandoeingabe .....                     | 344        |
| 10.4      | Ein- und Ausgabeumleitung .....           | 349        |
| 10.5      | Kommandos ausführen .....                 | 352        |
| 10.6      | Substitutionsmechanismen .....            | 354        |
| 10.7      | Shell-Variablen .....                     | 359        |
| 10.8      | bash-Script-Beispiele .....               | 363        |
| 10.9      | bash-Script-Grundregeln .....             | 370        |

|           |  |            |
|-----------|--|------------|
| 10.10     | Variablen in bash-Scripts .....                                    | 371        |
| 10.11     | Codestrukturierung in bash-Scripts .....                           | 378        |
| 10.12     | Referenz wichtiger bash-Sonderzeichen .....                        | 386        |
| <b>11</b> | <b>Dateien und Verzeichnisse .....</b>                             | <b>389</b> |
| 11.1      | Umgang mit Dateien und Verzeichnissen .....                        | 389        |
| 11.2      | Links .....  | 400        |
| 11.3      | Dateitypen (MIME) .....  | 402        |
| 11.4      | Dateien suchen (find, grep, locate) .....                          | 404        |
| 11.5      | Zugriffsrechte, Benutzer und Gruppenzugehörigkeit .....            | 409        |
| 11.6      | Spezialbits und die umask-Einstellung .....                        | 415        |
| 11.7      | Access Control Lists und Extended Attributes .....                 | 420        |
| 11.8      | Die Linux-Verzeichnisstruktur .....                                | 425        |
| 11.9      | Device-Dateien .....   | 429        |
| <b>12</b> | <b>Prozessverwaltung .....</b>                                     | <b>433</b> |
| 12.1      | Prozesse starten, verwalten und stoppen .....                      | 433        |
| 12.2      | Prozesse unter einer anderen Identität ausführen (su) .....        | 441        |
| 12.3      | Prozesse unter einer anderen Identität ausführen (sudo) .....      | 443        |
| 12.4      | Prozesse unter einer anderen Identität ausführen (PolicyKit) ..... | 447        |
| 12.5      | Systemprozesse (Dämonen) .....                                     | 450        |
| 12.6      | Prozesse automatisch starten (Cron) .....                          | 454        |
| 12.7      | Prozesse automatisch starten (systemd-Timer) .....                 | 459        |
| <b>13</b> | <b>Konverter für Grafik, Text und Multimedia .....</b>             | <b>463</b> |
| 13.1      | Grafik-Konverter .....   | 463        |
| 13.2      | Audio- und Video-Konverter .....                                   | 465        |
| 13.3      | Textkonverter (Zeichensatz und Zeilentrennung) .....               | 468        |
| 13.4      | Dateinamenkonverter (Zeichensatz) .....                            | 469        |
| 13.5      | Dokumentkonverter (PostScript, PDF, HTML, LaTeX) .....             | 469        |
| 13.6      | Markdown und Pandoc .....  | 477        |
| <b>14</b> | <b>Netzwerk-Tools .....</b>  | <b>481</b> |
| 14.1      | Netzwerkstatus ermitteln .....                                     | 481        |
| 14.2      | Auf anderen Rechnern arbeiten (SSH) .....                          | 485        |
| 14.3      | Dateien übertragen (FTP) .....                                     | 491        |
| 14.4      | Lynx .....   | 496        |
| 14.5      | Mutt .....   | 497        |

## TEIL IV Text- und Code-Editoren

---

|           |   |     |
|-----------|---|-----|
| <b>15</b> | <b>Vim</b> .....                              | 501 |
| 15.1      | Schnelleinstieg .....                         | 503 |
| 15.2      | Cursorbewegung .....                          | 505 |
| 15.3      | Text bearbeiten .....                         | 506 |
| 15.4      | Suchen und Ersetzen .....                     | 510 |
| 15.5      | Mehrere Dateien gleichzeitig bearbeiten ..... | 511 |
| 15.6      | Interna .....                                 | 513 |
| 15.7      | Tipps und Tricks .....                        | 516 |
| <b>16</b> | <b>Emacs</b> .....                            | 519 |
| 16.1      | Schnelleinstieg .....                         | 519 |
| 16.2      | Grundlagen .....                              | 523 |
| 16.3      | Cursorbewegung .....                          | 525 |
| 16.4      | Text markieren, löschen und einfügen .....    | 527 |
| 16.5      | Text bearbeiten .....                         | 528 |
| 16.6      | Fließtext .....                               | 531 |
| 16.7      | Suchen und Ersetzen .....                     | 534 |
| 16.8      | Puffer und Fenster .....                      | 537 |
| 16.9      | Besondere Bearbeitungsmodi .....              | 539 |
| 16.10     | Konfiguration .....                           | 541 |
| 16.11     | MELPA .....                                   | 544 |
| 16.12     | Unicode .....                                 | 545 |
| <b>17</b> | <b>Atom und VSCode</b> .....                  | 547 |
| 17.1      | Atom .....                                    | 548 |
| 17.2      | VSCode .....                                  | 555 |

## TEIL V Systemkonfiguration und Administration

---

|           |   |     |
|-----------|---|-----|
| <b>18</b> | <b>Basiskonfiguration</b> .....                 | 563 |
| 18.1      | Einführung .....                                | 563 |
| 18.2      | Konfiguration der Textkonsolen .....            | 567 |
| 18.3      | Datum und Uhrzeit .....                         | 570 |
| 18.4      | Datum und Uhrzeit via NTP synchronisieren ..... | 572 |
| 18.5      | Benutzer und Gruppen, Passwörter .....          | 576 |



|           |  |            |
|-----------|--|------------|
| 18.6      | PAM, NSS und nscd .....                                    | 587        |
| 18.7      | Spracheinstellung, Internationalisierung, Unicode .....    | 592        |
| 18.8      | Hardware-Referenz .....                                    | 598        |
| 18.9      | Logging (Syslog) .....                                     | 611        |
| 18.10     | Logging (Journal) .....                                    | 619        |
| <b>19</b> | <b>Software- und Paketverwaltung</b> .....                 | <b>623</b> |
| 19.1      | Einführung .....   | 623        |
| 19.2      | RPM-Paketverwaltung .....                                  | 627        |
| 19.3      | Yum .....  | 631        |
| 19.4      | DNF .....  | 636        |
| 19.5      | ZYpp .....   | 638        |
| 19.6      | Debian-Paketverwaltung (dpkg) .....                        | 640        |
| 19.7      | APT .....  | 643        |
| 19.8      | PackageKit .....   | 655        |
| 19.9      | tar .....  | 656        |
| 19.10     | Umwandlung zwischen Paketformaten (alien) .....            | 656        |
| 19.11     | Verwaltung von Parallelinstallationen (alternatives) ..... | 657        |
| 19.12     | Flatpak und Snap .....                                     | 659        |
| 19.13     | Distributionsspezifische Eigenheiten .....                 | 665        |
| <b>20</b> | <b>Bibliotheken und Java</b> .....                         | <b>677</b> |
| 20.1      | Bibliotheken .....   | 677        |
| 20.2      | Programme selbst kompilieren .....                         | 682        |
| 20.3      | Java .....   | 687        |
| <b>21</b> | <b>Grafiksystem</b> .....                                  | <b>689</b> |
| 21.1      | Grundlagen .....   | 690        |
| 21.2      | Grafiktreiber .....  | 694        |
| 21.3      | NVIDIA-Treiberinstallation .....                           | 700        |
| 21.4      | Status des Grafiksystems feststellen .....                 | 702        |
| 21.5      | Start des Grafiksystems .....                              | 705        |
| 21.6      | Konfiguration von X (xorg.conf) .....                      | 710        |
| 21.7      | Dynamische Konfigurationsänderungen mit RandR .....        | 716        |
| <b>22</b> | <b>Administration des Dateisystems</b> .....               | <b>721</b> |
| 22.1      | Wie alles zusammenhängt .....                              | 723        |
| 22.2      | USB-Datenträger formatieren und nutzen .....               | 725        |

|           |   |            |
|-----------|---|------------|
| 22.3      | Device-Namen für Festplatten und andere Datenträger .....         | 728        |
| 22.4      | Partitionierung der Festplatte oder SSD .....                     | 733        |
| 22.5      | parted-Kommando .....   | 737        |
| 22.6      | Partitionierungswerkzeuge mit grafischer Benutzeroberfläche ..... | 742        |
| 22.7      | Dateisystemtypen .....  | 744        |
| 22.8      | Verwaltung des Dateisystems (mount und /etc/fstab) .....          | 749        |
| 22.9      | Dateisystemgrundlagen .....                                       | 755        |
| 22.10     | Das ext-Dateisystem (ext2, ext3, ext4) .....                      | 758        |
| 22.11     | Das btrfs-Dateisystem .....                                       | 764        |
| 22.12     | Das xfs-Dateisystem .....   | 778        |
| 22.13     | Windows-Dateisysteme (vfat, ntfs) .....                           | 780        |
| 22.14     | CDs und DVDs .....  | 784        |
| 22.15     | Externe Datenträger .....   | 786        |
| 22.16     | Swap-Partitionen und -Dateien .....                               | 787        |
| 22.17     | RAID .....  | 790        |
| 22.18     | Logical Volume Manager (LVM) .....                                | 798        |
| 22.19     | SMART .....   | 803        |
| 22.20     | SSD-TRIM .....  | 807        |
| 22.21     | Verschlüsselung .....   | 808        |
| <b>23</b> | <b>GRUB</b> .....   | <b>817</b> |
| 23.1      | GRUB-Grundlagen .....   | 817        |
| 23.2      | GRUB-Bedienung (Anwendersicht) .....                              | 826        |
| 23.3      | GRUB-Konfiguration .....  | 827        |
| 23.4      | Manuelle GRUB-Installation und Erste Hilfe .....                  | 841        |
| <b>24</b> | <b>Das Init-System</b> .....                                      | <b>847</b> |
| 24.1      | systemd .....   | 848        |
| 24.2      | Das Init-V-System .....   | 857        |
| 24.3      | Eigene Init-Scripts bzw. Init-Konfigurationsdateien .....         | 861        |
| 24.4      | Systemstart bei CentOS, Fedora und RHEL .....                     | 865        |
| 24.5      | Systemstart bei Debian, Raspbian und Ubuntu .....                 | 867        |
| 24.6      | Systemstart bei SUSE/openSUSE .....                               | 869        |
| 24.7      | Internet Service Daemon .....                                     | 870        |
| <b>25</b> | <b>Kernel und Module</b> .....                                    | <b>875</b> |
| 25.1      | Kernelmodule .....  | 876        |
| 25.2      | Device Trees .....  | 882        |
| 25.3      | Kernelmodule selbst kompilieren .....                             | 885        |

|      |   |     |
|------|---|-----|
| 25.4 | Kernel selbst konfigurieren und kompilieren ..... | 889 |
| 25.5 | Kernel-Neustart mit kexec .....                   | 899 |
| 25.6 | Kernel-Live-Patches .....                         | 900 |
| 25.7 | Die Verzeichnisse /proc und /sys .....            | 902 |
| 25.8 | Kernel-Boot-Optionen .....                        | 904 |
| 25.9 | Kernelparameter verändern .....                   | 908 |

## TEIL VI LAN-Server

---

|           |   |      |
|-----------|---|------|
| <b>26</b> | <b>Netzwerkkonfiguration</b> .....                  | 911  |
| 26.1      | Der NetworkManager .....                            | 911  |
| 26.2      | Proxy-Konfiguration .....                           | 919  |
| 26.3      | Netzwerkgrundlagen und Glossar .....                | 920  |
| 26.4      | Manuelle LAN- und WLAN-Konfiguration .....          | 933  |
| 26.5      | LAN-Konfigurationsdateien .....                     | 942  |
| 26.6      | Distributionspezifische Konfigurationsdateien ..... | 948  |
| 26.7      | Zeroconf und Avahi .....                            | 958  |
| <b>27</b> | <b>Internet-Gateway</b> .....                       | 961  |
| 27.1      | Einführung .....                                    | 961  |
| 27.2      | Netzwerkkonfiguration .....                         | 967  |
| 27.3      | Masquerading (NAT) .....                            | 970  |
| 27.4      | Der WLAN-Authenticator hostapd .....                | 973  |
| 27.5      | DHCP- und Nameserver-Grundlagen .....               | 976  |
| 27.6      | Dnsmasq (DHCP- und Nameserver) .....                | 978  |
| <b>28</b> | <b>Samba</b> .....                                  | 987  |
| 28.1      | Grundlagen und Glossar .....                        | 988  |
| 28.2      | Basiskonfiguration und Inbetriebnahme .....         | 992  |
| 28.3      | Passwortverwaltung .....                            | 999  |
| 28.4      | Netzwerkverzeichnisse .....                         | 1006 |
| 28.5      | Beispiel – Home- und Medien-Server .....            | 1013 |
| 28.6      | Beispiel – Firmen-Server .....                      | 1016 |
| 28.7      | Client-Zugriff .....                                | 1019 |
| <b>29</b> | <b>NFS und AFP</b> .....                            | 1025 |
| 29.1      | NFS .....   | 1025 |
| 29.2      | Apple Filing Protocol .....                         | 1032 |



|           |                                     |      |
|-----------|-------------------------------------|------|
| <b>30</b> | <b>CUPS</b> .....                   | 1037 |
| 30.1      | Grundlagen .....                    | 1037 |
| 30.2      | CUPS-Interns .....                  | 1040 |
| 30.3      | Druckerkonfiguration .....          | 1046 |
| 30.4      | Drucken in lokalen Netzwerken ..... | 1049 |
| 30.5      | AirPrint .....                      | 1052 |

## TEIL VII Root-Server

---

|           |   |      |
|-----------|---|------|
| <b>31</b> | <b>Secure Shell (SSH)</b> .....                 | 1057 |
| 31.1      | Installation .....                              | 1058 |
| 31.2      | Konfiguration und Absicherung .....             | 1058 |
| 31.3      | DenyHosts und Fail2Ban .....                    | 1061 |
| 31.4      | Authentifizierung mit Schlüsseln .....          | 1064 |
| 31.5      | Zusatzwerkzeuge .....                           | 1067 |
| <b>32</b> | <b>Apache</b> .....                             | 1073 |
| 32.1      | Apache .....                                    | 1073 |
| 32.2      | Webverzeichnisse einrichten und absichern ..... | 1081 |
| 32.3      | Virtuelle Hosts .....                           | 1089 |
| 32.4      | Verschlüsselte Verbindungen (HTTPS) .....       | 1095 |
| 32.5      | Let's Encrypt .....                             | 1105 |
| 32.6      | Webzugriffsstatistiken .....                    | 1112 |
| 32.7      | PHP .....                                       | 1116 |
| 32.8      | FTP-Server (vsftpd) .....                       | 1118 |
| <b>33</b> | <b>MySQL und MariaDB</b> .....                  | 1123 |
| 33.1      | Installation und Inbetriebnahme .....           | 1124 |
| 33.2      | Administrationswerkzeuge .....                  | 1133 |
| 33.3      | Backups .....                                   | 1138 |
| <b>34</b> | <b>Postfix und Dovecot</b> .....                | 1143 |
| 34.1      | Einführung und Grundlagen .....                 | 1143 |
| 34.2      | Postfix (MTA) .....                             | 1154 |
| 34.3      | Postfix-Verschlüsselung (TLS/STARTTLS) .....    | 1162 |
| 34.4      | Postfix-Konten .....                            | 1169 |
| 34.5      | Dovecot (POP- und IMAP-Server) .....            | 1179 |

|           |  |      |
|-----------|--|------|
| 34.6      | Client-Konfiguration .....               | 1186 |
| 34.7      | Spam-Abwehr .....                        | 1187 |
| 34.8      | ClamAV (Virenabwehr) .....               | 1193 |
| 34.9      | SPF, DKIM und DMARC .....                | 1195 |
| 34.10     | Konfigurationstest und Fehlersuche ..... | 1205 |
| <b>35</b> | <b>Nextcloud</b> .....                   | 1207 |
| 35.1      | Installation .....                       | 1208 |
| 35.2      | Wartung .....                            | 1215 |
| 35.3      | Betrieb .....                            | 1217 |
| 35.4      | Kontakte und Termine .....               | 1219 |

## TEIL VIII Sicherheit

---

|           |   |      |
|-----------|---|------|
| <b>36</b> | <b>Backups</b> .....                        | 1225 |
| 36.1      | Backup-Benutzeroberflächen .....            | 1225 |
| 36.2      | Backups auf NAS-Geräten .....               | 1231 |
| 36.3      | Dateien komprimieren und archivieren .....  | 1232 |
| 36.4      | Verzeichnisse synchronisieren (rsync) ..... | 1235 |
| 36.5      | Inkrementelle Backups (rdiff-backup) .....  | 1238 |
| 36.6      | Inkrementelle Backups (rsnapshot) .....     | 1240 |
| 36.7      | Backup-Scripts .....                        | 1243 |
| 36.8      | Backups auf S3-Speicher .....               | 1246 |
| <b>37</b> | <b>Firewalls</b> .....                      | 1251 |
| 37.1      | Netzwerkgrundlagen und -analyse .....       | 1251 |
| 37.2      | Basisabsicherung von Netzwerkdiensten ..... | 1257 |
| 37.3      | Firewall-Grundlagen .....                   | 1261 |
| 37.4      | Firewall-Konfigurationshilfen .....         | 1267 |
| 37.5      | Firewall mit iptables selbst gebaut .....   | 1274 |
| <b>38</b> | <b>SELinux und AppArmor</b> .....           | 1283 |
| 38.1      | SELinux .....                               | 1283 |
| 38.2      | AppArmor .....                              | 1291 |

**TEIL IX Virtualisierung & Co.**

---

|              |  |      |
|--------------|--|------|
| <b>39</b>    | <b>VirtualBox und Vagrant</b> .....                                    | 1299 |
| 39.1         | VirtualBox installieren .....  | 1300 |
| 39.2         | VirtualBox-Maschinen einrichten .....                                  | 1304 |
| 39.3         | Arbeitstechniken und Konfigurationstipps .....                         | 1309 |
| 39.4         | Vagrant .....  | 1316 |
| <b>40</b>    | <b>KVM</b> .....   | 1329 |
| 40.1         | Grundlagen .....   | 1330 |
| 40.2         | KVM ohne libvirt .....   | 1337 |
| 40.3         | Der Virtual Machine Manager .....                                      | 1339 |
| 40.4         | libvirt-Kommandos .....  | 1348 |
| 40.5         | Integration der virtuellen Maschinen in das LAN (Netzwerkbrücke) ..... | 1354 |
| 40.6         | Direkter Zugriff auf den Inhalt einer Image-Datei .....                | 1357 |
| <b>41</b>    | <b>Docker</b> .....  | 1363 |
| 41.1         | Grundlagen, Nomenklatur und Installation .....                         | 1364 |
| 41.2         | Docker kennenlernen .....  | 1367 |
| 41.3         | Docker administrieren .....  | 1380 |
| 41.4         | Docker Images erzeugen und weitergeben .....                           | 1390 |
| 41.5         | Interna .....  | 1398 |
| <b>42</b>    | <b>Linux on Windows</b> .....  | 1405 |
| 42.1         | WSL ausprobieren .....   | 1406 |
| 42.2         | Serverbetrieb .....  | 1411 |
| 42.3         | Interna .....  | 1417 |
| <b>Index</b> | .....  | 1419 |



# Vorwort

Linux hat in den vergangenen 25 Jahren einen Siegeszug hingelegt, der seinesgleichen sucht. Linux begann als Hobby-Projekt des finnischen Programmierers Linus Torvalds und dominiert heute viele Segmente des IT-Markts. Erstaunlich ist nur, dass viele die Existenz von Linux gar nicht bemerken. Das liegt daran, dass Linux seine Erfolge quasi überall feiert – nur nicht im Desktop-Segment, wo die Dominanz von Windows anscheinend nicht zu brechen ist. Aber wussten Sie,

Linux ist unsichtbar, aber allgegenwärtig

- ▶ dass auf jedem Android-Gerät (Smartphone, Tablet, TV-Gerät) Linux läuft?
- ▶ dass ein Großteil der Server- und Cloud-Infrastruktur (Amazon, Google etc.) aus Linux-Rechnern gebildet wird?
- ▶ dass auf unzähligen *Embedded Devices* (also z. B. in WLAN-Routern, NAS-Geräten und Bastelrechnern wie dem Raspberry Pi) Linux läuft?
- ▶ dass zuletzt 498 der Top-500-Superrechner Linux verwendeten?

Was macht Linux so erfolgreich? Die freie und kostenlose Verfügbarkeit des Quellcodes von Linux und der meisten unter Linux laufenden Programme macht es möglich, Linux viel schneller und unkomplizierter als andere Betriebssysteme an neue Herausforderungen anzupassen – ganz egal, ob es um das *Internet of Things* geht, das momentan in aller Munde ist, um Hausautomation, um künstliche Intelligenz, um Software für selbststeuernde Autos oder um Simulationsmodelle für die Klimaforschung.

Der Schlüssel zum Erfolg: Open Source

## Was dieses Buch kann – und was nicht

In diesem Buch stelle ich Ihnen Linux von Grund auf vor. Die Themenpalette reicht über die Installation von Linux auf einem Notebook oder PC über die Desktop-Anwendung bis hin zum Server-Einsatz und zur Virtualisierung. Ein umfassendes Kapitel stellt den Minicomputer Raspberry Pi vor. Er eignet sich nicht nur für Elektronikbasteleien, sondern ermöglicht auch einen besonders kostengünstigen Einstieg in die Linux-Embedded-Welt.

Besonders wichtig ist mir, dass Sie Linux nicht nur anwenden, sondern auch verstehen lernen: Ausführliche Grundlagenkapitel erklären, wie Sie Linux im Terminal bedienen, wie Sie Linux optimal konfigurieren und warum Linux so funktioniert.

Nach der Lektüre dieses Kapitel kennen Sie nicht nur Linux an sich, sondern auch die Philosophie von Unix/Linux – also gewissermaßen *the Linux way to do it*.

Trotz mehr als 1400 Seiten kann das Buch nicht jedes Problem beschreiben, das beim Betrieb mit Linux auftreten kann. Insbesondere bei Hardware-Inkompatibilitäten kann ich in der Regel nicht weiterhelfen – ganz einfach, weil ich nicht die Möglichkeit habe, diverse Linux-Distributionen auf jeder erdenklichen Hardware auszuprobieren.

**Neu in der  
15. Auflage**

Für die vorliegende 15. Auflage habe ich dieses Buch einmal mehr umfassend überarbeitet und aktualisiert und gleichzeitig Platz für neue Inhalte geschaffen. Zu den neuen Themen in dieser Auflage zählen unter anderem:

- ▶ die Editoren *Atom* und *VSCoDe*
- ▶ das neue Grafikprotokoll *Wayland*
- ▶ die neuen Paketsysteme *Snap* und *Flatpak*
- ▶ das Kernel-Live-Patching
- ▶ *Let's Encrypt* zum kostenlosen Erstellen von Zertifikaten für Apache
- ▶ die DKIM-Konfiguration für den Mail-Server Postfix
- ▶ das Greylisting zur Minimierung von Spam
- ▶ *Nextcloud* zum Einrichten eines eigenen Datei-Austausch-Servers
- ▶ das Werkzeug *Vagrant* zum automatisierten Einrichten virtueller Maschinen
- ▶ das Container-System *Docker*
- ▶ das *Windows Subsystem for Linux*, mit dem Sie Linux-Tools neuerdings direkt unter Windows nutzen können

**Viel Spaß!**

Im Vergleich zu kommerziellen Produkten bietet Linux Ihnen die Möglichkeit, das Betriebssystem beinahe grenzenlos an Ihre eigenen Bedürfnisse und Vorlieben anzupassen – sei es zur Programmierung, für den Netzeinsatz oder als Server. Für nahezu jede Aufgabe stehen mehrere Werkzeuge zur Wahl. Und je mehr Sie sich in die Linux-Welt einarbeiten, desto mehr wird Linux *Ihr* Betriebssystem. Ich wünsche Ihnen viel Freude beim Experimentieren und Arbeiten mit Linux!

*Michael Kofler*  
<https://kofler.info>  
<https://twitter.com/michaelkofler>

# Konzeption

Das Buch ist in neun Teile gegliedert:

- ▶ **Teil I** erklärt, was Linux eigentlich ist, und vermittelt das Grundlagenwissen, das Sie für eine optimale und sichere **Installation** brauchen. Hier finden Sie konkrete Installationsanleitungen für etliche Distributionen: CentOS, Debian, Fedora, Linux Mint, openSUSE, Ubuntu und Ubuntu Server.
- ▶ **Teil II** behandelt Linux auf dem **Desktop**. Hier lernen Sie verschiedene Desktop-Systeme kennen. Den Schwerpunkt lege ich klar auf das einsteigerfreundliche Gnome, aber auch KDE, Unity, Cinnamon & Co. kommen nicht zu kurz. Außerdem stelle ich Ihnen die wichtigsten Programme vor, um im Web zu surfen, E-Mails und Fotos zu verwalten und um Audio-Dateien und Filme abzuspielen. Ein umfassendes Kapitel zum Minicomputer Raspberry Pi zeigt Ihnen, wie Sie Linux auf einem Minicomputer als Medien-Center oder als Plattform für Bastelprojekte einsetzen können.
- ▶ In **Teil III** lernen Sie das **Terminal** kennen. In mehreren Kapiteln lernen Sie, mit welchen Kommandos Sie das Dateisystem durchsuchen, wie Sie Dokumente und Bilder in andere Formate konvertieren und wie Sie den Kommandointerpreter bash nutzen.
- ▶ In **Teil IV** stehen verschiedene **Texteditoren** im Mittelpunkt. Neben den Urgesteinen Vi und Emacs stelle ich Ihnen auch zwei Vertreter einer neuen Generation von Editoren vor – Atom und VSCode.
- ▶ **Teil V** widmet sich der **Systemkonfiguration**. Egal, ob es gerade bei Ihrer Hardware Probleme gibt oder ob Sie ganz besondere Anforderungen stellen – hier erfahren Sie, wie Sie das Dateisystem administrieren, das Grafiksystem konfigurieren, Software-Pakete installieren und aktualisieren, den Systemstart konfigurieren sowie den Kernel und seine Module einrichten bzw. neu kompilieren.
- ▶ **Teil VI** zeigt, wie Sie **Linux im lokalen Netzwerk** nutzen – vom Client (LAN, WLAN) bis zum Server (Internet-Gateway, DNS, NFS, Samba, AFP), wahlweise mit IPv4 oder mit IPv6.
- ▶ **Teil VII** behandelt **Root-Server**, also Server, die extern in einem Rechenzentrum laufen und direkten Internetzugang haben. Wenn Sie einen derartigen Server mieten, werden Sie dort wahrscheinlich Web-, Mail- und Datenbank-Server einrichten – und vielleicht sogar mit Nextcloud Ihre eigene Mini-Cloud bilden.

- ▶ **Teil VIII** hat verschiedene Aspekte der **Sicherheit** zum Thema. Dort erfahren Sie, wie Sie Backups durchführen und Ihre Server durch Firewalls, SELinux oder AppArmor schützen.
- ▶ In **Teil IX** geht es um verschiedene Arten der Virtualisierung: Hier lernen Sie das **Desktop-Virtualisierungssystem** VirtualBox (samt Vagrant) sowie das Server-Virtualisierungssystem KVM kennen. Ein weiteres Kapitel stellt das Container-System Docker vor. Zuletzt erfahren Sie, dass Sie Linux mittlerweile sogar direkt in Windows ausführen können – mit dem Windows Subsystem for Linux (WSL).

## Formales

**Kommandos** In diesem Buch sind die Teile eines Kommandos, die tatsächlich einzugeben sind, fett hervorgehoben. Im folgenden Beispiel müssen Sie also nur `ls *.tex` eingeben, um sich die Liste aller `*.tex`-Dateien im aktuellen Verzeichnis anzeigen zu lassen:

```
user$ ls *.tex
article.tex
...
```

**Mehrzeilige Kommandos** Falls einzelne Kommandos nicht in einer Zeile Platz finden, werden sie mit dem Zeichen `\` auf zwei oder mehr Zeilen verteilt. `\` ist ein unter Linux zulässiges Zeichen, um mehrzeilige Kommandoangaben durchzuführen. Sie können das Kommando aber natürlich auch einzellig ohne `\` eintippen.

**root** Manche Kommandos können nur vom Systemadministrator `root` ausgeführt werden. In diesem Fall wird der Kommandoprompt als `root#` dargestellt:

```
root# systemctl restart apache2
```

**sudo** Kommandos mit `root`-Rechten führen Sie auf vielen Distributionen am einfachsten mit `sudo` aus. Unter Ubuntu ist das sogar der einzig mögliche Weg:

```
user$ sudo systemctl restart apache2
Password: *****
```

**MiB versus MByte** In der EDV ist es üblich, mit Zweierpotenzen zu rechnen. Ein Megabyte sind demnach nicht eine Million Byte, sondern  $2^{20}$  Byte, also exakt 1.048.576 Byte. Um diesen Umstand zu betonen, empfiehlt die IEC (International Electrotechnical Commission) die Verwendung der Einheiten KiB, MiB, GiB und TiB. Mittlerweile halten sich viele Linux-Programme an diese Empfehlung, und in der vorliegenden 15. Auflage habe ich auch dieses Buch entsprechend angepasst. Hintergrundinformationen finden Sie in der Wikipedia:

<https://de.wikipedia.org/wiki/Byte>



TEIL I  
**Installation**



# Kapitel 1

## Was ist Linux?

Um die einleitende Frage zu beantworten, erkläre ich in diesem Kapitel zuerst einige wichtige Begriffe, die im gesamten Buch immer wieder verwendet werden: Betriebssystem, Unix, Distribution, Kernel etc. Ein knapper Überblick über die Merkmale von Linux und die verfügbaren Programme macht deutlich, wie weit die Anwendungsmöglichkeiten von Linux reichen. Es folgt ein kurzer Ausflug in die Geschichte von Linux: Sie erfahren, wie Linux entstanden ist und auf welchen Komponenten es basiert.

Von zentraler Bedeutung ist dabei natürlich die *General Public License* (kurz GPL), die angibt, unter welchen Bedingungen Linux weitergegeben werden darf. Erst die GPL macht Linux zu einem freien System, wobei »frei« mehr heißt als einfach »kostenlos«.

### 1.1 Einführung

Linux ist ein Unix-ähnliches Betriebssystem. Der wichtigste Unterschied gegenüber historischen Unix-Systemen besteht darin, dass Linux zusammen mit dem vollständigen Quellcode frei kopiert werden darf.

Ein Betriebssystem ist ein Bündel von Programmen, mit denen die grundlegendsten Funktionen eines Rechners realisiert werden: die Schnittstelle zwischen Mensch und Maschine (also konkret: die Verwaltung von Tastatur, Bildschirm etc.) und die Verwaltung der Systemressourcen (CPU-Zeit, Speicher etc.). Sie benötigen ein Betriebssystem, damit Sie ein Anwendungsprogramm überhaupt starten und eigene Daten in einer Datei speichern können. Populäre Betriebssysteme sind Windows, Linux, BSD, macOS und iOS.

Betriebssystem

Schon lange vor Windows, Linux oder macOS gab es Unix. Dieses Betriebssystem war technisch gesehen seiner Zeit voraus: echtes Multitasking, eine Trennung der Prozesse voneinander, klar definierte Zugriffsrechte für Dateien, ausgereifte Netzwerkfunktionen etc. Allerdings bot Unix anfänglich nur eine spartanische Benutzeroberfläche und stellte hohe Hardware-Anforderungen. Das erklärt, warum Unix fast ausschließlich im wissenschaftlichen und industriellen Bereich eingesetzt wurde.

Unix versus Linux

In seiner Verbreitung hat Linux Unix fast vollständig verdrängt: Große Teile des Internets (z. B. Google) werden heute von Linux getragen. Linux läuft auf herkömmlichen Rechnern, in Form von Android auf Smartphones und Tablets, auf Embedded Devices (z. B. ADSL-Routern, NAS-Festplatten) und in Supercomputern: Mehr als 99 Prozent der 500 schnellsten Rechner der Welt laufen heute unter Linux (<https://top500.org/statistics/list>).

**Kernel** Genau genommen bezeichnet der Begriff Linux nur den Kernel: Er ist der innerste Teil (Kern) eines Betriebssystems mit ganz elementaren Funktionen, wie Speicherverwaltung, Prozessverwaltung und Steuerung der Hardware. Die Informationen in diesem Buch beziehen sich auf den Kernel 4.*n*.

## 1.2 Hardware-Unterstützung

Linux unterstützt beinahe die gesamte gängige PC-Hardware und läuft darüber hinaus auch auf anderen Hardware-Plattformen, z. B. auf Smartphones mit ARM-CPU. Dennoch müssen Sie beim Kauf eines neuen Rechners aufpassen. Es gibt einige Hardware-Komponenten, die im Zusammenspiel mit Linux oft Probleme machen:

- ▶ **Grafikkarten:** Fast alle auf dem Markt vertretenen Grafikkarten bzw. in die CPU integrierten Grafik-Cores funktionieren unter Linux. Für viele Linux-Anwender ohne besondere Anforderungen an das Grafiksysteem sind Intel-CPU mit eingebautem Grafik-Core die optimale Lösung. Neue Grafikkarten von NVIDIA und ATI/AMD erfordern hingegen oft Zusatztreiber, damit die Karte perfekt genutzt werden kann. Die Installation dieser Treiber bereitet oft Probleme.
- ▶ **Hybrid-Grafiksysteme:** Besonders problematisch sind Grafiksysteme, bei denen ein energiesparender interner Grafik-Core mit einer schnelleren externen Grafikkarte kombiniert wird. Mit geeigneten Windows- oder macOS-Treibern wechselt das Betriebssystem im laufenden Betrieb zwischen dem Grafik-Core und der Grafikkarte. Unter Linux funktioniert das – wenn überhaupt – nur nach einer komplizierten Konfiguration.
- ▶ **Hochauflösende Displays:** Hardware-technisch kommt Linux mit nahezu allen auf dem Markt erhältlichen Bildschirmen zurecht. Relativ schlecht ist leider die Software-Unterstützung von hochauflösenden Displays, die je nach Marketing-Sprech Retina-, XHD- oder HiDPI-Displays, 4k- oder 5k-Monitore genannt werden. Wenn Sie das Display in der vollen Auflösung betreiben, bleiben Teile der Benutzeroberfläche unleserlich klein.
- ▶ **WLAN- und Netzwerkadapter:** WLAN- und LAN-Controller machen selten Probleme. Nur ganz neue Modelle werden von Linux mitunter noch nicht unterstützt.

- ▶ **SSD-Cache:** Manche Notebooks kombinieren eine herkömmliche Festplatte mit einer kleinen SSD. In der Theorie erhalten Sie damit das Beste aus beiden Welten, also viel Speicherplatz und hohe Geschwindigkeit für wenig Geld. Die Praxis sieht schon unter Windows weit weniger rosig aus. Wenn dann auch noch Linux ins Spiel kommt, ist es mit den Vorteilen des SSD-Caches ganz vorbei. Im besten Fall ignoriert Linux den SSD-Cache ganz einfach und läuft so, als gäbe es nur eine herkömmliche Festplatte; im ungünstigsten Fall verursachen Sie ein defektes Dateisystem, wenn Sie unter Linux in eine Windows-Partition schreiben, deren Daten sich teilweise im SSD-Cache befinden. Investieren Sie ein paar Euro mehr in ein Notebook, das nur eine SSD enthält – es lohnt sich!
- ▶ **Energiesparfunktionen:** Gerade neue Notebooks haben unter Linux oft deutlich kürzere Akku-Laufzeiten als unter Windows. Dieses Ärgernis resultiert daraus, dass das Zusammenspiel diverser Energiesparfunktionen optimale Treiber voraussetzt, die für Linux oft gar nicht oder erst ein, zwei Jahre nach der Markteinführung verfügbar sind.

Stellen Sie also *vor* dem Kauf eines neuen Rechners bzw. einer Hardware-Erweiterung sicher, dass alle Komponenten von Linux unterstützt werden. Auch eine Internet-suche nach *linux <hardwarename>* kann nicht schaden. Lesenswert sind außerdem Testberichte der Zeitschrift c't: Deren Redakteure machen sich bei den meisten Geräten die Mühe, auch die Linux-Kompatibilität zu testen – zuletzt z. B. in diesem Artikel über Business-Notebooks:

<http://heise.de/-3678376> (kostenpflichtig)

### Checkliste für das ideale Linux-Notebook bzw. den idealen Linux-PC

Wenn ich mir einen neuen Rechner kaufe, achte ich zumeist auf die folgenden Punkte:

- ▶ **CPU und Grafik:** Für mich kommt nur eine in die 64-Bit-CPU integrierte Grafiklösung infrage, die mit Open-Source-Treibern gut funktioniert. Diese Voraussetzungen erfüllen die meisten Intel-CPU's.
- ▶ **Display:** Bei Notebooks erspare ich mir den Aufpreis für ein Display mit mehr als 1920 × 1080 Pixel – eine optimale Nutzung ist unter Linux leider schwer möglich. Immerhin ist gerade beim Desktop-System Gnome Besserung in Sicht (im wahren Sinne des Wortes); man kann also argumentieren, dass ein XHD-Display eine Investition für die Zukunft ist.
- ▶ **Speicher:** Es muss eine SSD sein. Größere Datenmengen speichere ich extern auf einem NAS-Gerät, in einem Cloud-Speicher etc.
- ▶ **Kein Windows:** Nach Möglichkeit kaufe ich Geräte ohne vorinstalliertes Windows, auch wenn die Preisersparnis gering ist.
- ▶ **Lieber etwas älter:** Um ganz neue Geräte mache ich nach Möglichkeit einen großen Bogen, auch wenn die Spezifikationen noch so verlockend sind.

## 1.3 Distributionen

Noch immer ist die einleitende Frage – Was ist Linux? – nicht ganz beantwortet. Viele Anwender interessiert der Kernel nämlich herzlich wenig, sofern er nur läuft und die vorhandene Hardware unterstützt. Für sie umfasst der Begriff Linux, wie er umgangssprachlich verwendet wird, neben dem Kernel auch das riesige Bündel von Programmen, das mit Linux mitgeliefert wird: Dazu zählen neben unzähligen Kommandos die Desktop-Systeme KDE und Gnome, das Office-Paket LibreOffice, der Webbrowser Firefox, das Zeichenprogramm GIMP sowie zahllose Programmiersprachen und Server-Programme (Webserver, Mail-Server, File-Server etc.).

Als Linux-Distribution wird die Einheit bezeichnet, die aus dem eigentlichen Betriebssystem (Kernel) und seinen Zusatzprogrammen besteht. Eine Distribution ermöglicht eine rasche und bequeme Installation von Linux. Die meisten Distributionen können kostenlos aus dem Internet heruntergeladen werden.

Distributionen unterscheiden sich vor allem durch folgende Punkte voneinander:

- ▶ **Umfang, Aktualität:** Die Anzahl, Auswahl und Aktualität der mitgelieferten Programme und Bibliotheken variiert stark. Manche Distributionen setzen bewusst auf etwas ältere, stabile Versionen – z. B. Debian.
- ▶ **Installations- und Konfigurationswerkzeuge:** Die mitgelieferten Programme zur Installation, Konfiguration und Wartung des Systems helfen dabei, die Konfigurationsdateien einzustellen. Das kann viel Zeit sparen.
- ▶ **Konfiguration des Desktops (KDE, Gnome):** Manche Distributionen lassen dem Anwender die Wahl zwischen KDE, Gnome und anderen Desktop-Systemen. Auch die Detailkonfiguration und optische Gestaltung variiert je nach Distribution.
- ▶ **Hardware-Unterstützung:** Linux kommt mit den meisten PC-Hardware-Komponenten zurecht. Dennoch gibt es im Detail Unterschiede zwischen den Distributionen, insbesondere wenn es darum geht, Nicht-Open-Source-Treiber (z. B. für NVIDIA-Grafikkarten) in das System zu integrieren.
- ▶ **Updates:** Sie können eine Linux-Distribution nur so lange sicher betreiben, wie Sie Updates bekommen. Danach ist aus Sicherheitsgründen ein Wechsel auf eine neue Version der Distribution erforderlich. Deswegen ist es bedeutsam, wie lange es für eine Distribution Updates gibt. Hier gilt meist die Grundregel: je teurer der Support, desto länger der Zeitraum. Einige Beispiele (Stand: Frühjahr 2017):

|                              |   |
|------------------------------|---|
| Fedora:                      | 13 Monate                                     |
| openSUSE Leap:               | ca. 18 bis 24 Monate                          |
| Red Hat Enterprise Linux:    | 10 Jahre (mit Einschränkungen sogar 13 Jahre) |
| SUSE Enterprise Server:      | 10 Jahre (mit Einschränkungen sogar 13 Jahre) |
| Ubuntu LTS:                  | 3 bis 5 Jahre                                 |
| Ubuntu (sonstige Versionen): | 9 Monate                                      |

- ▶ **Live-System:** Viele Distributionen ermöglichen den Linux-Betrieb direkt von einer CD/DVD oder von einem USB-Stick. Das ermöglicht ein einfaches Ausprobieren von Linux. Außerdem bieten Live-Systeme eine ideale Möglichkeit, um ein defektes Linux-System zu reparieren.
- ▶ **Zielplattform (CPU-Architektur):** Viele Distributionen sind nur für Intel- und AMD-kompatible Prozessoren erhältlich, in der Regel in einer 32- und in einer 64-Bit-Variante. Es gibt aber auch Distributionen für andere Prozessorplattformen, z. B. für ARM- oder für PowerPC-CPU's.
- ▶ **Support:** Bei kommerziellen Distributionen bekommen Sie Hilfe bei der Installation (via E-Mail und/oder per Telefon).
- ▶ **Lizenz:** Die meisten Distributionen sind kostenlos erhältlich. Bei einigen Distributionen gibt es hier aber Einschränkungen: Beispielsweise ist bei den Enterprise-Distributionen von Red Hat und SUSE ein Zugriff auf das Update-System nur für registrierte Kunden möglich. Sie zahlen hier nicht für die Software an sich, sondern für das Service-Angebot rund herum.

So belebend die Konkurrenz vieler Distributionen für deren Weiterentwicklung ist, so lästig ist sie bei der Installation von Programmen, die nicht mit der Distribution mitgeliefert werden: Eine fehlende oder veraltete Programm-Bibliothek kann die Ursache dafür sein, dass ein Programm nicht läuft. Abhilfe versucht das Linux-Standard-Base-Projekt (LSB) zu schaffen: Die LSB-Spezifikation definiert Regeln, die einen gemeinsamen Nenner aller am LSB-Projekt beteiligten Distributionen sicherstellen:

Linux Standard  
Base (LSB)

<https://wiki.linuxfoundation.org/lb/lsb/start>

## Gängige Linux-Distributionen

Der folgende Überblick über die wichtigsten verfügbaren Distributionen soll Ihnen eine erste Orientierungshilfe geben. Die Liste ist alphabetisch geordnet und erhebt keinen Anspruch auf Vollständigkeit.

**Android** ist eine von Google entwickelte Plattform für Mobilfunkgeräte und Tablets. Android hat damit Linux zu der Weltdominanz verholfen, über die Linux-Entwickler in der Vergangenheit gescherzt haben. Aber Android ist natürlich keine typische, PC-taugliche Distribution.

Android

**Arch Linux** ist eine für technische Anwender optimierte Linux-Distribution. Die manuell im Textmodus durchzuführende Installation stellt sicher, dass Einsteiger einen großen Bogen um Arch Linux machen. Dafür zählen <https://wiki.archlinux.org> und <https://wiki.archlinux.de> zu den besten Quellen für Linux-Konfigurationsdetails im Netz. Arch-Linux-Derivate wie **Manjaro** und **Antergos** mit grafischen Installations-

Arch Linux

und Konfigurationsprogrammen haben Arch-Linux zuletzt sogar in die Top-10-Liste von *distrowatch.com* gebracht.

### CentOS und Scientific Linux

**CentOS** und **Scientific Linux** sind zwei kostenlose Varianten zu Red Hat Enterprise Linux (RHEL). Beide Distributionen sind binärkompatibel zu RHEL, es fehlen aber alle Red-Hat-Markenzeichen, -Logos etc. Die Distributionen sind vor allem für Server-Betreiber interessant, die kompatibel zu RHEL sein möchten, sich die hohen RHEL-Kosten aber nicht leisten können.

### Chrome OS

Das **Chrome OS** wird wie Android von Google entwickelt. Es ist für Notebooks optimiert und setzt zur Nutzung eine aktive Internetverbindung voraus. Die Benutzeroberfläche basiert auf dem Google Chrome Webbrowser. Chrome OS spielt aktuell in Europa keine große Rolle, wohl aber auf dem Bildungsmarkt in den USA: Dort werden billige Chrome-Books (also Notebooks mit Chrome OS) häufig in Schulen eingesetzt.

### Debian

**Debian** ist die älteste vollkommen freie Distribution. Sie wird von engagierten Linux-Entwicklern zusammengestellt, wobei die Einhaltung der Spielregeln »freier« Software eine hohe Priorität genießt. Die strikte Auslegung dieser Philosophie hat in der Vergangenheit mehrfach zu Verzögerungen geführt.

Debian richtet sich an fortgeschrittene Linux-Anwender und hat einen großen Marktanteil bei Server-Installationen. Im Vergleich zu anderen Distributionen ist Debian stark auf maximale Stabilität hin optimiert und enthält deswegen oft relativ alte Programmversionen. Dafür steht Debian für viele Hardware-Plattformen zur Verfügung, unter anderem für AMD64, ARM64, ARMEL, ARMHF, i386, Mips, Mipsel, PowerPC, PPC64EL und S390X. Es gibt zahlreiche Distributionen, die sich von Debian ableiten, z. B. Raspbian und Ubuntu.

### Fedora

**Fedora** ist der kostenlose Entwicklungszweig von Red Hat Linux. Die Entwicklung wird von Red Hat unterstützt und gelenkt. Für Red Hat ist Fedora eine Art Spielwiese, auf der neue Funktionen ausprobiert werden können, ohne die Stabilität der Enterprise-Versionen zu gefährden. Programme, die sich unter Fedora bewähren, werden später in die Enterprise-Versionen integriert. Bei technisch interessierten Linux-Fans ist Fedora beliebt, weil diese Distribution oft eine Vorreiterrolle spielt: Neue Linux-Funktionen finden sich oft zuerst in Fedora und erst später in anderen Distributionen. Neue Fedora-Versionen erscheinen alle sechs Monate. Updates werden einen Monat nach dem Erscheinen der übernächsten Version eingestellt, d. h., die Lebensdauer ist mit 13 Monaten sehr kurz.

### Kali Linux

Das auf Debian basierende Kali Linux enthält eine riesige Sammlung von Hacking- und Pen-Testing-Werkzeugen. Die Distribution gilt als *der* Werkzeugkasten für Hacker und Sicherheits-Experten.



**openSUSE** ist eine kostenlose Linux-Distribution. Beginnend mit der seit November 2015 verfügbaren Version »Leap 42.n« basiert openSUSE auf den Enterprise-Versionen von SUSE, ersetzt aber viele Programme durch aktuellere Versionen. Voraussichtlich ab 2018 sollen die Versionsnummern von SUSE Enterprise und openSUSE Leap zusammengeführt werden. openSUSE wird dann einen Sprung zurück zu Version 16 machen.

openSUSE

Oracle bietet unter dem Namen **Oracle Linux** eine Variante zu Red Hat Enterprise Linux (RHEL) an. Das ist aufgrund der Open-Source-Lizenzen eine zulässige Vorgehensweise. Technisch gibt es nur wenige Unterschiede zu RHEL, die Oracle-Variante ist aber billiger und ohne Support sogar kostenlos verfügbar. Dennoch ist die Verbreitung von Oracles Linux-Variante verhältnismäßig gering.

Oracle

**Raspbian** ist die Standard-Distribution für den beliebten Minicomputer Raspberry Pi. Raspbian basiert auf Debian, wurde für den Raspberry Pi aber speziell adaptiert und erweitert.

Raspbian

**Red Hat** ist die international bekannteste und erfolgreichste Linux-Firma. Red-Hat-Distributionen dominieren insbesondere den amerikanischen Markt. Die Paketverwaltung auf der Basis des RPM-Formats (einer Eigenentwicklung von Red Hat) wurde von vielen anderen Distributionen übernommen.

Red Hat

Red Hat ist überwiegend auf Unternehmenskunden ausgerichtet. Die Enterprise-Versionen (RHEL = **Red Hat Enterprise Linux**) sind vergleichsweise teuer. Sie zeichnen sich durch hohe Stabilität und einen zehnjährigen Update-Zeitraum aus. Für Linux-Enthusiasten und -Entwickler, die ein Red-Hat-ähnliches System zum Nulltarif suchen, bieten sich **CentOS** und **Fedora** an.

SUSE gilt weltweit als die Nummer zwei auf dem kommerziellen Linux-Markt. **SUSE Enterprise** ist vor allem im europäischen Markt verankert.

SUSE

**Ubuntu** ist die zurzeit populärste Distribution für Privatanwender. Ubuntu verwendet als Basis Debian, ist aber besser für Desktop-Anwender optimiert (Motto: *Linux for human beings*). Die kostenlose Distribution erscheint im Halbjahresrhythmus. Für gewöhnliche Versionen werden Updates über neun Monate zur Verfügung gestellt. Für die alle zwei Jahre erscheinenden LTS-Versionen gibt es sogar 3 bzw. 5 Jahre lang Updates (für Desktop- bzw. Server-Pakete). Finanziell wird Ubuntu Linux durch die Firma Canonical unterstützt.

Ubuntu

Zu Ubuntu gibt es eine Menge offizieller und inoffizieller Varianten. Etabliert und weit verbreitet sind **Ubuntu Server**, **Kubuntu**, **Xubuntu**, **Ubuntu MATE** und **Linux Mint**. Relativ neue Ubuntu-Derivate mit modernen Desktop-Systemen sind **Budgie** (Solus Desktop), **elementary OS** (macOS-ähnlicher Desktop), **Neon** und **Zorin OS** (Windows-ähnlicher Desktop). Besonders interessant ist Neon: Diese Distribution

Ubuntu-Derivate

kombiniert Ubuntu LTS mit stets aktuellen KDE-Paketen und etabliert sich damit aktuell als *die* Distribution für KDE-Fans.

### Andere Distributionen

Neben den oben aufgezählten »großen« Distributionen gibt es im Internet zahlreiche Zusammenstellungen von Miniatursystemen. Sie sind vor allem für Spezialaufgaben konzipiert, etwa für Wartungsarbeiten (Emergency-Systeme) oder um ein Linux-System ohne eigentliche Installation verwenden zu können (Live-Systeme). Populäre Vertreter dieser Linux-Gattung sind **Devil Linux**, **Parted Magic** und **TinyCore**.

Einen ziemlich guten Überblick über alle momentan verfügbaren Linux-Distributionen, egal ob kommerziellen oder anderen Ursprungs, finden Sie im Internet auf der folgenden Seite:

<https://distrowatch.com>

### Die Qual der Wahl

Eine Empfehlung für eine bestimmte Distribution ist schwierig. Für Linux-Einsteiger ist es zumeist von Vorteil, sich vorerst für eine weitverbreitete Distribution wie Debian, Fedora, openSUSE oder Ubuntu zu entscheiden. Eine gute Wahl ist auch Linux Mint. Zu diesen Distributionen sind sowohl im Internet als auch im Buch- und Zeitschriftenhandel viele Informationen verfügbar. Bei Problemen ist es vergleichsweise leicht, Hilfe zu finden.

Kommerzielle Linux-Anwender bzw. Server-Administratoren müssen sich entscheiden, ob sie bereit sind, für professionellen Support Geld auszugeben. In diesem Fall spricht wenig gegen die Marktführer Red Hat und SUSE. Andernfalls sind CentOS, Debian und Ubuntu attraktive kostenlose Alternativen.

## 1.4 Open-Source-Lizenzen (GPL & Co.)

Die Grundidee von »Open Source« besteht darin, dass der Quellcode von Programmen frei verfügbar ist und von jedem erweitert bzw. geändert werden darf. Allerdings ist damit auch eine Verpflichtung verbunden: Wer Open-Source-Code zur Entwicklung eigener Produkte verwendet, muss den gesamten Code ebenfalls wieder frei weitergeben.

Die Open-Source-Idee verbietet übrigens keinesfalls den Verkauf von Open-Source-Produkten. Auf den ersten Blick scheint das ein Widerspruch zu sein. Tatsächlich bezieht sich die Freiheit in »Open Source« mehr auf den Code als auf das fertige Produkt. Zudem regelt die freie Verfügbarkeit des Codes auch die Preisgestaltung von Open-Source-Produkten: Nur wer neben dem Kompilat eines Open-Source-Programms weitere Zusatzleistungen anbietet (Handbücher, Support etc.), wird überleben. Sobald der Preis in keinem vernünftigen Verhältnis zu den Leistungen steht, werden sich andere Firmen finden, die es günstiger machen.

Das Ziel der Open-Source-Entwickler ist es, Software zu schaffen, deren Quellen frei verfügbar sind und es auch bleiben. Um einen Missbrauch auszuschließen, sind viele Open-Source-Programme durch die *GNU General Public License* (kurz GPL) geschützt. Hinter der GPL steht die *Free Software Foundation* (FSF). Diese Organisation wurde von Richard Stallman gegründet, um hochwertige Software frei verfügbar zu machen. Richard Stallman ist übrigens auch der Autor des Editors Emacs, der in [Kapitel 16](#) beschrieben wird.

General Public License (GPL)

Die Kernaussage der GPL besteht darin, dass zwar jeder den Code verändern und sogar die resultierenden Programme verkaufen darf, dass aber gleichzeitig der Anwender/Käufer das Recht auf den vollständigen Code hat und diesen ebenfalls verändern und wieder kostenlos weitergeben darf. Jedes GNU-Programm muss zusammen mit dem vollständigen GPL-Text weitergegeben werden. Die GPL schließt damit aus, dass jemand ein GPL-Programm weiterentwickeln und verkaufen kann, *ohne* die Veränderungen öffentlich verfügbar zu machen. Jede Weiterentwicklung ist somit ein Gewinn für *alle* Anwender. Den vollständigen Text der GPL finden Sie hier:

<https://gnu.org/licenses/gpl.html>

Das Konzept der GPL ist recht einfach zu verstehen, im Detail treten aber immer wieder Fragen auf. Viele davon werden hier beantwortet:

<https://gnu.org/licenses/gpl-faq.html>

Wenn Sie glauben, dass Sie alles verstanden haben, sollten Sie das GPL-Quiz ausprobieren:

<https://gnu.org/cgi-bin/license-quiz.cgi>

Neben der GPL existiert noch die Variante LGPL (Lesser GPL). Der wesentliche Unterschied zur GPL besteht darin, dass eine derart geschützte Bibliothek auch von kommerziellen Produkten genutzt werden darf, deren Code *nicht* frei verfügbar ist. Ohne die LGPL könnten GPL-Bibliotheken nur wieder für GPL-Programme genutzt werden, was in vielen Fällen eine unerwünschte Einschränkung für kommerzielle Programmierer wäre.

Lesser General Public License (LGPL)

Durchaus nicht alle Teile einer Linux-Distribution unterliegen den gleichen Copyright-Bedingungen! Obwohl der Kernel und viele Tools der GPL unterliegen, gelten für manche Komponenten und Programme andere rechtliche Bedingungen:

Andere Lizenzen

- ▶ **MIT- und BSD-Lizenz:** Die MIT- und BSD-Lizenzen erlauben die kommerzielle Nutzung des Codes *ohne* die Verpflichtung, Änderungen öffentlich weiterzugeben. Die Lizenzen sind damit wesentlich liberaler als die GPL und eher mit der LGPL vergleichbar.

- ▶ **Doppellizenzen:** Für einige Programme gelten Doppellizenzen. Beispielsweise können Sie den Datenbank-Server MySQL für Open-Source-Projekte, auf einem eigenen Webserver bzw. für die innerbetriebliche Anwendung gemäß der GPL kostenlos einsetzen. Wenn Sie hingegen ein kommerzielles Produkt auf der Basis von MySQL entwickeln und samt MySQL verkaufen möchten, ohne Ihren Quellcode zur Verfügung zu stellen, dann kommt die kommerzielle Lizenz zum Einsatz. Die Weitergabe von MySQL wird in diesem Fall kostenpflichtig.
- ▶ **Kommerzielle Lizenzen:** Einige Programme unterstehen zwar einer kommerziellen Lizenz, dürfen aber dennoch kostenlos genutzt werden. Ein bekanntes Beispiel ist das Flash-Plugin von Adobe: Zwar ist das Programm unter Linux kostenlos erhältlich (und darf auch in Firmen kostenlos eingesetzt werden), aber der Quellcode zu diesem Programm ist nicht verfügbar.

Manche Distributionen kennzeichnen die Produkte, bei denen die Nutzung oder Weitergabe eventuell lizenzrechtliche Probleme verursachen könnte. Bei Debian befinden sich solche Programme in der Paketquelle *non-free*.

Das Dickicht der zahllosen, mehr oder weniger »freien« Lizenzen ist schwer zu durchschauen. Die Bandbreite zwischen der manchmal fundamentalistischen Auslegung von »frei« im Sinne der GPL und den verklausulierten Bestimmungen mancher Firmen, die ihr Software-Produkt zwar frei nennen möchten (weil dies gerade modern ist), in Wirklichkeit aber uneingeschränkte Kontrolle über den Code behalten möchten, ist groß.

Eine gute Einführung in das Thema geben die beiden folgenden Websites. Das Ziel von *opensource.org* ist es, unabhängig von Einzel- oder Firmeninteressen die Idee (oder das Ideal) von Software mit frei verfügbarem Quellcode zu fördern. Dort finden Sie auch eine Liste von Lizenzen, die der Open-Source-Idee entsprechen.

<https://heise.de/-221957>

<https://opensource.org>

### Lizenzkonflikte zwischen Open- und Closed-Source-Software

Open-Source-  
Lizenzen für  
Entwickler

Wenn Sie Programme entwickeln und diese zusammen mit Linux bzw. in Kombination mit Open-Source-Programmen oder -Bibliotheken verkaufen möchten, müssen Sie sich in die bisweilen verwirrende Problematik der unterschiedlichen Software-Lizenzen tiefer einarbeiten. Viele Open-Source-Lizenzen erlauben die Weitergabe nur, wenn auch Sie Ihren Quellcode im Rahmen einer Open-Source-Lizenz frei verfügbar machen. Auf je mehr Open-Source-Komponenten mit unterschiedlichen Lizenzen Ihr Programm basiert, desto komplizierter wird die Weitergabe.

Es gibt aber auch Ausnahmen, die die kommerzielle Nutzung von Open-Source-Komponenten erleichtern: Beispielsweise gilt für Apache und PHP sinngemäß, dass Sie diese Programme auch in Kombination mit einem Closed-Source-Programm frei weitergeben dürfen.

Manche proprietäre Treiber für Hardware-Komponenten (z.B. für NVIDIA-Grafikkarten) bestehen aus einem kleinen Kernelmodul (Open Source) und diversen externen Programmen oder Bibliotheken, deren Quellcode nicht verfügbar ist (Closed Source). Das Kernelmodul hat nur den Zweck, eine Verbindung zwischen dem Kernel und dem Closed-Source-Treiber herzustellen.

GPL-Probleme mit Hardware-Treibern

Diese Treiber sind aus Sicht vieler Linux-Anwender eine gute Sache: Sie sind kostenlos verfügbar und ermöglichen es, diverse Hardware-Komponenten zu nutzen, zu denen es entweder gar keine oder zumindest keine vollständigen Open-Source-Treiber für Linux gibt. Die Frage ist aber, ob bzw. in welchem Ausmaß die Closed-Source-Treiber wegen der engen Verzahnung mit dem Kernel, der ja der GPL untersteht, diese Lizenz verletzen. Viele Open-Source-Entwickler dulden die Treiber nur widerwillig. Eine direkte Weitergabe mit GPL-Produkten ist nicht zulässig, weswegen der Benutzer die Treiber in der Regel selbst herunterladen und installieren muss.

## 1.5 Die Geschichte von Linux

Da Linux ein Unix-ähnliches Betriebssystem ist, müsste ich an dieser Stelle eigentlich mit der Geschichte von Unix beginnen – aber dazu fehlt hier der Platz. Stattdessen beginnt diese Geschichtsstunde mit der Gründung des GNU-Projekts durch Richard Stallman. GNU steht für *GNU is not Unix*. In diesem Projekt wurden seit 1982 Open-Source-Werkzeuge entwickelt. Dazu zählen der GNU-C-Compiler, der Texteditor Emacs sowie diverse GNU-Utilities wie `find` und `grep` etc.

1982: GNU

Erst sieben Jahre nach dem Start des GNU-Projekts war die Zeit reif für die erste Version der *General Public License*. Diese Lizenz stellt sicher, dass freier Code frei bleibt.

1989: GPL

Die allerersten Teile des Linux-Kernels (Version 0.01) entwickelte Linus Torvalds. Er gab seinen Code im September 1991 über das Internet frei. Schnell fanden sich weltweit Programmierer, die an der Idee Interesse hatten und Erweiterungen dazu programmierten. Als der Kernel von Linux die Ausführung des GNU-C-Compiler erlaubte, stand auch die gesamte Palette der GNU-Tools zur Verfügung. Weitere Komponenten waren das Dateisystem Minix, Netzwerk-Software von BSD-Unix, das X Window System des MIT und dessen Portierung XFree86 etc.

1991: Linux-Kernel 0.01

Linux ist also nicht nur Linus Torvalds zu verdanken. Hinter Linux stehen vielmehr eine Menge engagierter Menschen, die in ihrer Freizeit, im Rahmen ihres Studiums oder bezahlt von Firmen wie Google, IBM oder HP freie Software produzieren.

- 1994: Erste Distributionen** Informatik-Freaks an Universitäten konnten sich Linux und seine Komponenten selbst herunterladen, kompilieren und installieren. Eine breite Anwendung fand Linux aber erst mit Linux-Distributionen, die Linux und die darum entstandene Software auf Disketten bzw. CD-ROMs verpackten und mit einem Installationsprogramm versahen. Vier der zu dieser Zeit entstandenen Distributionen existieren heute noch: Debian, Red Hat, Slackware und SUSE.
- 1996: Pinguin** 1996 wurde der Pinguin zum Linux-Logo.
- 1998: Microsoft nimmt Linux wahr** Mit dem rasanten Siegeszug des Internets stieg auch die Verbreitung von Linux, vor allem auf Servern. Gewissermaßen zum Ritterschlag für Linux wurde der legendäre Ausspruch von Steve Ballmer: *Microsoft is worried about free software ...* Ein Jahr später ging Red Hat spektakulär an die Börse.
- 2009: Android** Mit der Android-Plattform brachte Google Linux zuerst auf das Handy (2009), danach auch auf Tablets und in TV-Geräte.
- 2012: Raspberry Pi** 2012 eroberte der Minicomputer Raspberry Pi die Herzen von Elektronikbastlern. Für nur rund 40 EUR können Sie mit dem Raspberry Pi selbst Hardware-Experimente durchführen, in die Welt der Heimautomation einsteigen, ein Medien-Center oder einen Home-Server betreiben. Der Raspberry Pi macht Embedded Linux zu einem Massenphänomen.

## 1.6 Software-Patente und andere Ärgernisse

Patente schützen in den USA und anderen Ländern Software-Ideen, -Konzepte und Algorithmen. Alles Mögliche und Unmögliches ist patentiert, triviale Dinge wie die Darstellung eines Fortschrittsbalkens oder die 1-Click-Bestellung (Amazon). Der Missbrauch derartiger Trivialpatente und die für die schnelllebige Software-Branche sehr langen Laufzeiten von 20 Jahren tragen zum Widerwillen gegen Software-Patente bei.

Beispielsweise verzichteten viele Distributionen jahrelang aus Angst vor Klagen darauf, Bibliotheken zum Abspielen von MP3-Dateien mitzuliefern; die darin eingesetzten Algorithmen sind durch Patente geschützt. Die Anwender mussten sich die zum Abspielen von MP3-Dateien erforderlichen Bibliotheken selbst installieren. Glücklicherweise laufen die MP3-Patente 2017 aus, sodass zumindest dieses Problem jetzt aus der Welt geschafft ist.

Während Patente selten ein Risiko für einzelne Software-Entwickler sind, spielen sie im Kampf um Marktanteile eine immer größere Rolle, besonders im heiß umkämpften Smartphone- und Tablet-Markt. Jeder große Hersteller verklagt jeden anderen – mit ungewissem Ausgang, aber auf jeden Fall zur Freude der beteiligten Rechtsanwälte und Kanzleien.

Ganz aussichtslos ist die Lage zum Glück nicht. Das liegt vor allem daran, dass einige Linux nahestehende Firmen wie IBM selbst über riesige Patent-Pools verfügen. Diverse Linux-Firmen haben zudem begonnen, selbst Patente zu sammeln, die teilweise von anderen Firmen gleichsam für Open-Source-Zwecke »gespendet« wurden. Das Absurde an der Situation besteht darin, dass ein verfehltes Patentrecht die Open-Source-Gemeinde dazu zwingt, selbst Patente einzusetzen, um sich gegen eventuelle Klagen zu schützen. Details über Patent-Pools der Open-Source-Gemeinde finden Sie hier:

<https://openinventionnetwork.com>

Auch abseits der MP3-Dateien ist der Multimedia-Markt ein Problemfeld. Beispielsweise können Sie unter Linux DVDs nicht ohne Weiteres abspielen. Diverse Gesetze verbieten in vielen Ländern sowohl die Weitergabe der erforderlichen Bibliotheken als auch die bloße Beschreibung, wie diese zu installieren sind – z. B. das Urheberrechtsgesetz in Deutschland.

Nicht besser sieht es mit online erworbenen Daten (Videos, E-Books etc.) aus, die durch DRM geschützt sind. DRM steht für *Digital Rights Management* und bezeichnet diverse Verfahren, um die Nutzung der Daten so einzuschränken, dass sie nur auf einem ganz bestimmten Rechner möglich ist. Sozusagen nebenbei werden Sie dadurch auf eine bestimmte Hardware (z. B. iPod oder iPhone) bzw. auf ein bestimmtes Betriebssystem (z. B. Windows, macOS) beschränkt. DRM-Gegner bezeichnen das System nicht umsonst als *Digital Restriction Management*.

Patent-Pools der  
Open-Source-  
Gemeinde

Multimedia

Digital Rights  
Management





# Kapitel 2

## Installationsgrundlagen

Dieses Kapitel gibt Ihnen einen Überblick über die Installation eines Linux-Systems auf einem Notebook oder einem PC mit einem Intel-kompatiblen Prozessor. Das Kapitel bezieht sich nicht auf eine spezielle Distribution, sondern beschreibt wesentliche Installationsschritte (z. B. die Partitionierung der Festplatte) in allgemeiner Form und vermittelt das erforderliche Grundlagenwissen. Spezifische Details zur Installation einiger ausgewählter Distributionen folgen dann im nächsten Kapitel.

Die Installation ist in den vergangenen Jahren immer einfacher geworden. Im Idealfall – d. h., wenn Sie Standard-Hardware verwenden und ausreichend Platz für Linux vorhanden ist – sollten 30 Minuten ausreichen, um zu einem funktionierenden Linux-System zu gelangen. Schwierig wird die Installation zumeist nur deswegen, weil im Regelfall ein wechselweiser Betrieb eines schon vorhandenen Windows-Betriebssystems und von Linux gewährleistet werden soll. Probleme kann es aber auch bei der Unterstützung ungewöhnlicher oder ganz neuer Hardware geben.

### Linux-Installation auf dem Raspberry Pi

Für die Installation von Linux auf Minicomputern bzw. Embedded Devices gelten vollkommen andere Regeln als bei einer PC-Installation. [Kapitel 8](#), »Raspberry Pi«, zeigt dies am Beispiel des Raspberry Pi.

## 2.1 Voraussetzungen

Damit Sie Linux installieren können, müssen mehrere Voraussetzungen erfüllt sein:

- ▶ Sie benötigen einen PC bzw. ein Notebook mit einem Intel-kompatiblen Prozessor. Dazu zählen alle gängigen 32- und 64-Bit-Prozessoren von Intel oder AMD. Es gibt auch Linux-Distributionen für Systeme mit anderen Prozessor-Architekturen (z. B. ARM) – sie sind aber nicht Thema dieses Kapitels.
- ▶ Sie benötigen eine freie Partition mit ausreichend Platz auf Ihrer Festplatte. Wie viel »ausreichend« ist, hängt von der Distribution und davon ab, wie viele Pro-

gramme Sie installieren und welche persönlichen Daten Sie speichern möchten (Fotos, Videos etc.). Meine Empfehlung lautet mindestens 25 GiB, um Linux einfach nur auszuprobieren.

- Sie benötigen Hardware-Komponenten, die von Linux erkannt und unterstützt werden. Gegenwärtig ist das bei einem Großteil der Standard-Hardware der Fall. Probleme bereiten momentan vor allem ganz neue WLAN-Adapter, Hybrid-Grafiksysteme und SSD-Caches (siehe auch [Abschnitt 1.2](#), »Hardware-Unterstützung«).

### Distributionen für Uralt-PCs sowie Installationen in virtuellen Maschinen

Wie ich im vorigen Kapitel erwähnt habe, gibt es auch Minimaldistributionen, die wesentlich geringere Hardware-Anforderungen stellen. In diesem Kapitel gehe ich aber davon aus, dass Sie eine gewöhnliche Distribution installieren – z. B. CentOS, Debian, Fedora, Kubuntu, RHEL, SUSE oder Ubuntu.

Wenn Sie Virtualisierungsprogramme wie VirtualBox oder VMware einsetzen, können Sie Linux auch innerhalb von Windows oder macOS in einer virtuellen Umgebung installieren und ausführen. Das vereinfacht die Installation, mindert aber auch die Funktionalität (limitierter Hardware-Zugriff, langsame 3D-Grafik etc.).

### 32 oder 64 Bit?

In fast allen gängigen PCs und Notebooks befinden sich 64-Bit-Prozessoren. Der wesentliche Unterschied zwischen 32- und 64-Bit-Prozessoren besteht darin, dass mit 64-Bit-Prozessoren Speicherbereiche über 4 GiB direkt adressiert werden können.

**64-Bit-Distributionen** Aus technischen Gründen werden die Vorteile der 64-Bit-Architektur nur wirksam, wenn die gesamte Distribution aus 64-Bit-Programmen und -Bibliotheken besteht. Deswegen gibt es von den meisten Distributionen zwei Ausführungen: eine 32-Bit-Version (übliche Kürzel sind i386, i586 oder i686, die sich auf Intel-Prozessorfamilien beziehen) und eine 64-Bit-Version (Kürzel x86\_64 oder AMD64).

**Empfehlung** In der Vergangenheit mussten sich Anwender von 64-Bit-Distributionen damit herummärgern, dass es diverse Nicht-Open-Source-Programme und -Treiber nur in 32-Bit-Versionen gab. Das hat sich mittlerweile zum Glück geändert. Heute gibt es kaum noch Gründe, die *gegen* eine 64-Bit-Installation sprechen. Aktuelle 64-Bit-Distributionen haben zudem den Vorteil, dass sie EFI-kompatibel sind. Während ich dieses Buch schrieb, habe ich ausschließlich mit 64-Bit-Distributionen gearbeitet.

## 2.2 BIOS und EFI

Jahrzehntelang war für die Initialisierung von PCs und Notebooks das sogenannte BIOS (*Basic Input/Output System*) verantwortlich. Dabei handelt es sich um ein Programm, das unmittelbar nach dem Einschalten des Rechners ausgeführt wird. Das BIOS ist für die Erkennung der Hardware-Komponenten, für die Konfiguration der Hardware sowie für den Start des Betriebssystems verantwortlich. Der Begriff BIOS ist für diese Funktionen seit 40 Jahren gebräuchlich. BIOS

Das traditionelle BIOS schleppt eine Menge Altlasten mit sich herum. Deswegen begann Intel bereits 1998 mit der Entwicklung des BIOS-Nachfolgers EFI (*Extensible Firmware Interface*). Später beteiligten sich viele namhafte Firmen (AMD, Apple, Microsoft etc.) an der Weiterentwicklung, wobei die Software auch eine neue Abkürzung bekam: UEFI (*Unified Extensible Firmware Interface*). Die Kürzel EFI und UEFI werden seither oft synonym verwendet, auch in diesem Buch: Ist bei modernen Mainboards oder PCs von EFI die Rede, ist fast immer UEFI gemeint. EFI und UEFI

Während Apple schon früh auf den EFI-Zug aufsprang und seit vielen Jahren alle Macs mit einer EFI-Variante ausstattet, dauerte es in der PC-Welt wesentlich länger. Der Siegeszug von EFI hat erst 2012 mit der Markteinführung von Windows 8 begonnen. Seither kommt EFI auf nahezu allen neuen Notebooks und PCs zum Einsatz. Viele EFI-Implementierungen sind zudem BIOS-kompatibel.

Aus technischer Sicht bietet EFI viele grundlegende Vorteile im Vergleich zum BIOS (höhere Initialisierungsgeschwindigkeit, Unterstützung der Parallelinstallation mehrerer Betriebssysteme etc.). Aus Anwendersicht reduzieren sich die Argumente für EFI aber zumeist auf drei Punkte: Wozu EFI?

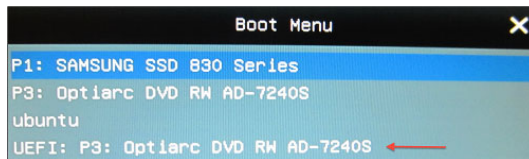
- ▶ EFI kommt mit Festplatten über 2 TiB (Terabyte) zurecht. Für das herkömmliche BIOS gilt das nur mit Einschränkungen.
- ▶ EFI ist kompatibel zu den GUID Partition Tables (GPT). Das ist eine modernere Form der Festplattenpartitionierung. Hintergrundinformationen zur Partitionierung und zu den GPT folgen in [Abschnitt 2.6](#).
- ▶ EFI vereinfacht die Parallelinstallation von Windows und Linux auf einem Rechner.

Die meisten aktuellen Linux-Distributionen sind EFI-kompatibel (siehe [Tabelle 2.1](#), Stand: Sommer 2017). Das Installationsmedium startet bei ihnen direkt im EFI-Modus (nicht im BIOS-Modus) und richtet Linux so ein, dass es direkt durch EFI hochgefahren werden kann. Linux und EFI

| Distribution | EFI-kompatibel ab | EFI Secure Boot ab       |
|--------------|-------------------|--------------------------|
| CentOS       | Version 6.4       | Version 7                |
| Debian       | Version 7         | eventuell ab Version 9.1 |
| Fedora       | Version 16        | Version 18               |
| Linux Mint   | Version 15        | nicht bekannt            |
| openSUSE     | Version 12.3      | Version 13.1             |
| RHEL         | Version 6.2       | Version 7                |
| Ubuntu       | Version 11.10     | Version 12.10            |

**Tabelle 2.1** EFI-Kompatibilität der wichtigsten Linux-Distributionen

Manche Mainboards unterstützen sowohl den herkömmlichen BIOS-Start als auch EFI: Im Boot-Menü erscheint das Installationsmedium dann möglicherweise doppelt, einmal mit der gewöhnlichen Bezeichnung und einmal mit dem vorangestellten Wort EFI oder UEFI. Sie müssen das Boot-Medium unbedingt in der EFI-Variante starten, wenn Sie eine EFI-Installation durchführen möchten (siehe [Abbildung 2.1](#)).



**Abbildung 2.1** Für eine EFI-Installation müssen Sie den EFI-Eintrag auswählen.

Es mag paradox wirken, aber aus Linux-Sicht bringt EFI kaum nennenswerte Vorteile mit sich: Linux unterstützte schon bisher im Zusammenspiel mit den meisten BIOS-Versionen GPT-partitionierte Festplatten in beliebiger Größe, und Linux konnte mit dem Bootloader GRUB schon bisher beliebig viele parallel installierte Betriebssysteme starten.

### EFI-Unterstützung nur für 64-Bit-Distributionen

Bei den meisten Linux-Distributionen kommt nur die 64-Bit-Variante mit EFI zurecht. Diese Einschränkung gilt z. B. für Debian und Ubuntu. Zwar sind Notebooks mit 32-Bit-CPU und EFI selten, aber es gibt sie. Vorsicht also beim Rechnerkauf: Auf einem derartigen Notebook ist es unmöglich, Linux zu installieren!

**Apple und EFI** Apple setzt schon viel länger auf EFI als herkömmliche PCs/Notebooks. Das bringt es leider mit sich, dass die EFI-Version auf Macs inkompatibel zu den (U)EFI-Versionen auf PCs ist. Trotz EFI ist die Installation von Linux auf einem Mac daher häufig mit Problemen verbunden.

Aus meiner persönlichen Erfahrung rate ich Einsteigern von der Linux-Installation auf Macs ab: Dabei treten nahezu garantiert Probleme auf! Zwar gibt es für die meisten Mac-Modelle im Internet Installationsanleitungen; um diese zu verstehen, brauchen Sie aber ein solides Linux-Grundwissen. In der Regel ist es zweckmäßiger, Linux auf dem Mac in einer virtuellen Maschine auszuführen.

Im Gegensatz zum BIOS sieht die EFI-Spezifikation die Parallelinstallation mehrerer Betriebssysteme sowie deren Auswahl während des Boot-Prozesses vor. Damit das funktioniert, muss es auf der Festplatte eine spezielle EFI-Partition geben, in der jedes Betriebssystem sein eigenes Startprogramm installiert (in der Fachsprache: seinen eigenen Bootloader).

EFI-Partition

Die EFI-Partition muss ein VFAT-Dateisystem enthalten, also ein Windows-95-kompatibles Dateisystem. Außerdem muss die Partition durch eine spezielle UID markiert sein. Microsoft empfiehlt, diese Partition als erste Partition auf der Festplatte einzurichten, obwohl der EFI-Standard dies nicht verlangt. Die Partition muss nicht besonders groß sein, ca. 100 bis 200 MiB reichen vollkommen aus.

Bei der Installation von Linux müssen Sie darauf achten, dass eine bereits vorhandene EFI-Partition in das Verzeichnis `/boot/efi` eingebunden werden muss. Existiert noch keine EFI-Partition, muss sie angelegt werden. Die Installationsprogramme der meisten aktuellen Linux-Distributionen kümmern sich automatisch um diesen Schritt, außer Sie entscheiden sich für eine manuelle Partitionierung: In diesem Fall sind Handarbeit und etwas Vorsicht angesagt. Auf keinen Fall darf eine vorhandene EFI-Partition formatiert werden, sonst kann keines der bereits installierten Betriebssysteme mehr gestartet werden!

UEFI Secure Boot ist eine von Microsoft betriebene Erweiterung der EFI-Funktionen: Wenn Secure Boot aktiv ist, kann nur ein Betriebssystem gestartet werden, das mit dem auf dem Mainboard hinterlegten Schlüssel signiert ist. Auf diese Weise ist ausgeschlossen, dass Viren oder andere Schadsoftware bereits in den Boot-Vorgang eingreifen – was in der Praxis in den letzten Jahren aber ohnedies nur äußerst selten der Fall war.

UEFI Secure Boot

Dennoch wird Secure Boot natürlich als Sicherheitsgewinn für Windows-Anwender verkauft. Aus Linux-Sicht verursacht diese Funktion hingegen Probleme: Bei aktivem Secure Boot kann Linux nur dann installiert und gestartet werden, wenn sein Startprogramm (genauer gesagt: sein Bootloader) mit einem auf dem Mainboard existierenden Schlüssel signiert ist. Auf den meisten Mainboards gibt es nur einen Schlüssel – den von Microsoft. Zwar stellt Microsoft den Schlüssel auch Linux-Distributoren gegen eine geringe Gebühr zur Verfügung, dennoch hat sich die Unterstützung von Secure Boot als relativ schwierig erwiesen. Für Linux-Anwender gibt es gegenwärtig zwei Wege, um Linux auf Rechnern mit UEFI Secure Boot einsetzen zu können:

- ▶ Sie verwenden eine Linux-Distribution, die kompatibel zu (U)EFI Secure Boot ist (siehe [Tabelle 2.1](#)). Bei diesen Distributionen kommt ein mit dem Microsoft-Schlüssel signierter Bootloader zum Einsatz, zumeist das Programm Shim. Dieses startet in einem zweiten Schritt den gewöhnlichen Linux-Bootloader GRUB.
- ▶ Sie deaktivieren UEFI Secure Boot vor der Installation. Die EFI-Spezifikation sieht diese Option erfreulicherweise vor. Wie diese Deaktivierung konkret aussieht, ist allerdings auf jedem Rechner bzw. bei jedem Mainboard anders und erfordert mitunter langes Suchen.

### Weiterführende Informationen

Weitere Details zu EFI können Sie auf den folgenden Webseiten nachlesen:

[https://de.wikipedia.org/wiki/Extensible\\_Firmware\\_Interface](https://de.wikipedia.org/wiki/Extensible_Firmware_Interface)

[https://wiki.archlinux.org/index.php/Unified\\_Extensible\\_Firmware\\_Interface](https://wiki.archlinux.org/index.php/Unified_Extensible_Firmware_Interface)

<https://help.ubuntu.com/community/UEFI>

<http://rodsbooks.com/efi-bootloaders/index.html>

## 2.3 Installationsvarianten

Bis vor wenigen Jahren verwendeten die meisten Distributionen dasselbe Installationsverfahren: Der Rechner wird neu gestartet, das auf der CD oder DVD befindliche Installationsprogramm wird ausgeführt und Linux wird auf die Festplatte installiert. Dieses Verfahren ist nach wie vor populär, es gibt aber mittlerweile eine Menge Varianten, die ich Ihnen hier vorstelle.

### CD/DVD als Installationsmedium

Falls Sie die CD/DVD nicht einer Zeitschrift oder einem Buch entnehmen, laden Sie die entsprechende ISO-Datei aus dem Internet herunter und brennen die CD oder DVD einfach selbst. Anschließend starten Sie Ihren Rechner neu und führen das auf der CD oder DVD befindliche Installationsprogramm aus.

### USB-Stick als Installationsmedium

Anstelle einer CD/DVD kann auch ein USB-Stick als Installationsmedium verwendet werden. Die ISO-Dateien fast aller Distributionen sind so konzipiert, dass sie direkt von USB-Datenträgern boot-fähig sind. Wenn Sie schon mit Linux vertraut sind, können Sie derartige Image-Dateien im Terminal einfach mit dem Kommando `dd` auf den USB-Stick oder eine Speicherkarte kopieren:

```
user$ dd if=ubuntu.img of=/dev/sdc bs=4M
```

Passen Sie aber auf, dass Sie mit `of=...` das korrekte Gerät angeben! Wer sich das nicht zutraut, kann diesen Schritt komfortabler mit einer Benutzeroberfläche durchführen, unter Linux beispielsweise mit dem auch für andere Distributionen geeigneten *Fedora Media Writer*, unter Windows oder macOS am komfortabelsten mit *Etcher* (<https://etcher.io>). Anschließend starten Sie Ihren Rechner neu und booten das Linux-Installationsprogramm vom USB-Stick.

Ein USB-Stick kann auch das Ziel einer Linux-Installation sein: Einige Distributionen bieten die Möglichkeit, Linux auf einen USB-Stick zu installieren. Das ergibt dann ein »Linux zum Mitnehmen«, das unterwegs nahezu auf jedem beliebigen Rechner ausgeführt werden kann – sofern Ihnen nicht BIOS/EFI-Inkompatibilitäten einen Strich durch die Rechnung machen.

Installation auf einen USB-Stick

Oft handelt es sich bei dieser Installationsvariante nicht um eine vollwertige Installation; vielmehr wird einfach ein Live-System auf den USB-Stick übertragen. Daraus ergeben sich diverse Einschränkungen, etwa was die Installation weiterer Programme oder die Durchführung von Updates betrifft. Ein Live-System – egal, ob auf einer CD oder auf einem USB-Stick – kann eine »richtige« Installation selten ersetzen.

Sie können Linux auch auf eine externe Festplatte installieren. Diese Variante sieht auf den ersten Blick verlockend aus, insbesondere bei Notebooks, deren eingebaute Festplatte oder SSD schon voll ist. Leider gibt es bei dieser Installationsvariante oft Probleme, das Linux-System anschließend zu starten. Deswegen ist diese Installationsform nur fortgeschrittenen Linux-Anwendern zu empfehlen.

Installation auf eine externe Festplatte

In der Vergangenheit war es üblich, vom Installationsmedium ein minimales Linux-System zu starten und damit dann ein Installationsprogramm auszuführen. Zunehmend wird aber ein anderes Konzept populärer: Es besteht darin, vom Installationsmedium ein vollständiges Linux-System zu starten, ein sogenanntes Live-System. Das Installationsprogramm wird dann innerhalb dieses Live-Systems ausgeführt. Diese z. B. bei Fedora und Ubuntu übliche Vorgehensweise hat den Vorteil, dass das Live-System auch für andere Zwecke verwendet werden kann – etwa um Linux auszuprobieren, um Reparaturarbeiten durchzuführen etc.

Installationsprogramm auf einem Live-System

Bei nahezu allen Distributionen erfolgt die Installation innerhalb einer grafischen Benutzeroberfläche. Optional kann die Installation zumeist auch im Textmodus durchgeführt werden, etwa wenn es Probleme bei der korrekten Erkennung der Grafikkarte gibt. Es gibt auch noch immer Distributionen, die *nur* im Textmodus installiert werden können, beispielsweise Arch Linux, Gentoo und Slackware sowie die Server-Variante von Ubuntu.

Installation im Textmodus

Bei einer Netzwerkinstallation werden die Installationsdateien nicht von einer CD/DVD oder einem USB-Stick gelesen, sondern aus dem Netzwerk. Dabei gibt es zwei Varianten, die sich darin unterscheiden, wie die Installation beginnt:

Netzwerkinstallation

- **Installationsstart mit einem herkömmlichen Medium:** Hier startet die Installation von einer CD oder einem USB-Stick. Das Installationsprogramm hilft bei der Herstellung der Netzwerkverbindung und lädt dann alle weiteren Daten aus dem Netz. Besonders populär ist diese Installationsform bei Debian mit dem sogenannten *netinst*-Image.

- **Installationsstart via Netzwerk:** Diese »echte« Netzwerkinstallation setzt voraus, dass Ihr Rechner die Boot-Daten aus dem lokalen Netzwerk laden kann. Die meisten gängigen Mainboards sind dazu in der Lage, wenn das BIOS oder EFI korrekt eingestellt wird.

Außerdem muss es im lokalen Netzwerk einen Server geben, der das Linux-Installationsprogramm in Form von Boot-Daten anbietet. Diese Vorgehensweise ist optimal, um viele Linux-Installationen auf einmal durchzuführen. Allerdings ist das Einrichten des Installations-Servers nicht ganz einfach. Nur ausgewählte Distributionen unterstützen dieses Installationsverfahren, unter anderem Red Hat und SUSE. Wenn Sie Debian auf mehreren Rechnern automatisch installieren möchten, werfen Sie einen Blick auf die folgende Seite:

*<https://fai-project.org>*

**Mehrere  
Distributionen  
auf einem PC**

Um verschiedene Distributionen auszuprobieren oder um eine neue Version Ihrer Distribution parallel zur vorhandenen Version zu testen, können Sie mehrere Distributionen nebeneinander auf Ihrer Festplatte oder SSD installieren. Dazu benötigt jede Distribution zumindest ihre eigene Systempartition. Die wichtigste Voraussetzung besteht also darin, dass auf Ihrer Festplatte Platz für weitere Partitionen ist.

Die Swap-Partition kann von unterschiedlichen Distributionen gemeinsam genutzt werden. Prinzipiell gilt dies auch für Datenpartitionen. Eine Home-Partition, die sich mehrere Distributionen »teilen«, ist aber problematisch: Je nach Distribution kommen unterschiedliche Versionen von Gnome, KDE oder Firefox zum Einsatz. Das kann zu Kollisionen und Fehlern aufgrund gemeinsamer Konfigurationsverzeichnisse führen.

**Server- und  
Cloud-  
Installationen**

Wenn Sie Linux auf einem Rechner installieren wollen, der weit entfernt in einem Rechenzentrum steht, gelten andere Regeln. Der Normalfall besteht darin, dass das Hosting- oder Cloud-Unternehmen Ihnen die Installation abnimmt. Sie müssen lediglich die gewünschte Distribution auswählen und ein paar Parameter einstellen (z. B. für die LVM- und RAID-Konfiguration). Wenige Minuten später können Sie sich mit einem zufällig generierten Passwort via SSH anmelden und können dann mit der Administration des Servers beginnen. (SSH steht für Secure Shell und ist, vereinfacht ausgedrückt, ein Hilfsmittel zur Fernadministration im Textmodus.)

Alternativ können Sie bei manchen Anbietern die Installation auch manuell durchführen. In diesem Fall wird das Installationsprogramm vom Hosting- oder Cloud-Provider so gestartet, dass Sie es in einem Webbrowser oder VNC-Client bedienen können.



## 2.4 Überblick über den Installationsprozess

Dieser Abschnitt fasst die Schritte einer gewöhnlichen Linux-Installation zusammen. »Gewöhnlich« bedeutet hier, dass auf dem Rechner bereits Microsoft Windows installiert ist. Wesentlich einfacher verläuft die Installation, wenn auf dem Rechner noch kein Betriebssystem installiert ist oder wenn dieses gelöscht werden darf.

Nun aber zu den Installationsschritten, die ich in den weiteren Abschnitten im Detail beschreiben werde:

- ▶ **Linux-Installation starten:** Legen Sie die Installations-CD/DVD in das Laufwerk oder stecken Sie den USB-Stick ein, und starten Sie den Rechner neu. Das Linux-Installationsprogramm sollte automatisch gestartet werden. Das Installationsprogramm sieht bei jeder Distribution ein wenig anders aus. Für die wichtigsten Distributionen folgen im nächsten Kapitel konkrete Tipps zur Bedienung des Installationsprogramms. Die ersten Fragen betreffen zumeist die Sprache der Benutzeroberfläche sowie die Konfiguration von Tastatur und Maus.

Falls anstelle des Linux-Installationsmediums weiterhin Windows gestartet wird, müssen Sie während des Rechnerstarts explizit das Boot-Medium angeben. Die erforderlichen Tastenkombinationen hängen vom BIOS bzw. EFI Ihres Rechners ab.

- ▶ **Windows-Partition verkleinern:** Normalerweise füllt Windows den Großteil der Festplatte oder SSD in einer einzigen, sehr großen Partition aus. Um Platz für Linux zu machen, muss diese Partition verkleinert werden. Empfehlenswert ist es, die Partition vor dem Start der Linux-Installation noch unter Windows zu verkleinern.

Alternativ sind dazu auch einige Linux-Installationsprogramme in der Lage – vorausgesetzt, Sie fahren Windows vorher vollständig herunter. Dazu müssen Sie die Eingabeaufforderung `cmd.exe` mit Administratorrechten starten und dort `shutdown /p` ausführen.

- ▶ **Linux-Partitionen anlegen:** Ein wesentlicher Schritt jeder Installation ist das Anlegen von Linux-Partitionen auf der Festplatte. Wie das Partitionierprogramm aussieht, hängt stark von der jeweiligen Distribution ab. Tipps zur optimalen Dimensionierung der Linux-Partitionen finden Sie in [Abschnitt 2.8](#), »Partitionierung der Festplatte«.
- ▶ **Installationsumfang auswählen:** Bei einigen Distributionen können Sie auswählen, welche Teile der Linux-Distribution Sie installieren möchten. Bei anderen Distributionen entfällt dieser Schritt (z. B. bei Ubuntu). Stattdessen wird hier ein relativ kleines Grundsystem installiert. Weitere Programme fügen Sie dann später bei Bedarf im laufenden Betrieb hinzu.

Für alle anderen Distributionen gilt in dieser Phase: Weniger ist oft mehr. Beschränken Sie sich anfangs auf die Programme, die Sie unbedingt brauchen, und fügen Sie weitere Software erst hinzu, wenn Sie diese benötigen. So halten Sie Ihr System schlank und minimieren den Aufwand der regelmäßigen Updates.

- ▶ **Konfiguration:** Je nach Installationsprogramm folgen nun diverse Rückfragen zur Konfiguration – z. B. zum gewünschten Passwort für den Administrator root, zu den Netzwerkeinstellungen, zur Druckerkonfiguration etc.
- ▶ **Bootloader:** Ungeklärt ist jetzt nur noch eine Frage: Wie soll Linux in Zukunft gestartet werden? Dazu wird bei den meisten Distributionen das Programm GRUB eingesetzt. Gänzlich unkompliziert ist die GRUB-Installation auf Rechnern mit EFI. GRUB wird dann einfach in ein Verzeichnis der EFI-Partition installiert. Auf diese Weise können beliebig viele Betriebssysteme parallel installiert werden.

Bei BIOS-Rechnern wird GRUB in den Boot-Sektor der Festplatte installiert. In Zukunft erscheint der Bootloader bei jedem Start, und Sie können auswählen, welches Betriebssystem Sie starten möchten. Allerdings kann die GRUB-Installation in seltenen Fällen Konflikte mit dem bisher installierten Bootloader verursachen.

Insgesamt wird die Erstinstallation von Linux vermutlich etwa eine Stunde in Anspruch nehmen. Mit etwas Übung und einem schnellen Rechner gelingt sie aber auch in 15 Minuten. Anschließend können Sie mit Linux zu arbeiten beginnen bzw. manuell weitere Konfigurationsschritte durchführen und Linux optimal an Ihre besonderen Ansprüche anpassen.

### Vorsicht bei der Partitionierung und bei der Konfiguration des Bootloaders

Es gibt während einer Linux-Installation nur zwei kritische Phasen, in denen Sie unbeabsichtigt Daten anderer Betriebssysteme zerstören oder Ihren Rechner nicht mehr startbar machen können: bei der Partitionierung der Festplatte und bei der Installation des Bootloaders auf die Festplatte. Führen Sie diese Schritte also mit besonderer Vorsicht aus.

### Festplatte oder SSD?

Aus Linux-Sicht ist es egal, ob sich in Ihrem Rechner eine herkömmliche Festplatte oder eine Solid State Disk (SSD) befindet. Wenn ich in diesem Buch also öfter einfach von der »Festplatte« schreibe, gilt dies gleichermaßen auch für SSDs. Ganz exakt wäre es, wenn ich jedes Mal »Datenträger« schreiben würde – aber dieser Begriff erschien mir zu sperrig.

## 2.5 Start der Linux-Installation

Sie beginnen die Installation damit, dass Sie die Installations-CD oder -DVD in Ihr DVD-Laufwerk legen bzw. den USB-Stick anstecken und den Rechner neu starten. Statt des üblichen Starts Ihres bereits installierten Betriebssystems sollte nun ein Linux-System bzw. das Linux-Installationsprogramm direkt von der CD starten.

Von einer  
CD/DVD starten

Sollte dies nicht gelingen, ist Ihr BIOS bzw. EFI vermutlich so konfiguriert, dass ein Booten von einer CD/DVD nicht möglich ist. Um die BIOS/EFI-Einstellungen zu ändern, müssen Sie unmittelbar nach dem Einschalten des Rechners eine Taste drücken, häufig **Entf** oder **F1**. Falls diese Tasten unwirksam bleiben, müssen Sie die richtige Tastenkombination im Internet recherchieren. Beachten Sie, dass während der BIOS-Einstellung meist das amerikanische Tastaturlayout gilt. Unter anderem sind **Y** und **Z** vertauscht!

Bei einigen Distributionen können Sie noch vor dem eigentlichen Start von Linux durch Funktionstasten die Sprache, das Tastaturlayout und eventuell einige weitere Parameter einstellen (siehe [Abbildung 2.2](#)). Das funktioniert allerdings aktuell nur auf BIOS-Rechnern bzw. in virtuellen Maschinen. Bei anderen Distributionen bzw. bei Installationen auf EFI-Rechnern erfolgen diese Einstellungen wenige Sekunden nach dem Start.

Erste  
Einstellungen

### Was sind tote Tasten?

Bei manchen Distributionen können Sie ein Tastaturlayout mit oder ohne sogenannte *Dead Keys* auswählen. Damit sind beispielsweise **~** und **^** gemeint. (De)aktiviert werden nicht die Tasten an sich, sondern das Zusammensetzen von Buchstaben mit diesen Tasten.

Bei einem Tastaturlayout mit Dead Keys können Sie beispielsweise den Buchstaben Ç in der Form **~**, **Ç** eingeben. Wenn Sie die Zeichen, die den Dead Keys zugeordnet sind, selbst eingeben möchten, müssen Sie die betreffende Taste und danach die Leerzeichentaste drücken – also **~** und dann die Leertaste für **~**. Beim Arbeiten im Terminal wird dieses Zeichen unter Linux häufig benötigt.

Bei einem Tastaturlayout ohne Dead Keys können Sie keine ausländischen Sonderzeichen zusammensetzen. Dafür ist jetzt die Eingabe der den Dead Keys zugeordneten Zeichen bequemer, weil das betreffende Zeichen sofort erscheint.



Abbildung 2.2 Sprachauswahl am Beginn einer SUSE-Installation

**Intern** Das Installationsprogramm läuft selbst bereits unter Linux. Dazu wird vom Installationsmedium zuerst der Linux-Kernel geladen. Der Kernel muss alle für die Installation relevanten Hardware-Komponenten richtig erkennen. Sollte das nicht gelingen, können Linux-Profis beim Start der Installation zusätzliche Kernelparameter angeben, um dem Kernel bei der Hardware-Erkennung auf die Sprünge zu helfen. Sobald der Kernel läuft, wird das eigentliche Installationsprogramm gestartet.

## 2.6 Grundlagen der Festplattenpartitionierung

Nach dem Start des Installationsprogramms und diversen elementaren Einstellungen ist die Partitionierung der Festplatte oder SSD der erste entscheidende Schritt der Installation. Zwar bieten viele Installationsprogramme an, diesen Schritt automatisch zu erledigen, dabei ist aber Vorsicht angebracht: Nicht immer entspricht das Resultat wirklich Ihren Bedürfnissen. Bevor ich in [Abschnitt 2.8](#), »Partitionierung der Festplatte«, konkrete Tipps zur Partitionierung gebe, erkläre ich Ihnen in diesem Abschnitt, was Partitionen überhaupt sind und welche Regeln beim Anlegen von Partitionen zu beachten sind.

Partitionen sind Abschnitte auf der Festplatte. Partitionen mit Windows-Dateisystemen bekommen eigene Buchstaben (C:, D: etc.) und verhalten sich scheinbar wie selbstständige Festplatten.

Was sind Partitionen?

Üblicherweise befinden sich auf dem Datenträger bereits drei Partitionen: eine winzige EFI-Partition, eine ebenso kleine Partition mit Dateien des Windows-Bootloaders sowie eine große Partition. Die große Partition füllt fast die gesamte Festplatte oder SSD und enthält neben Windows auch alle Ihre persönlichen Daten.

Auf manchen Rechnern gibt es darüber hinaus weitere Partitionen, die Dateien zur Wiederherstellung des Systems, Hardware-Treiber, Zusatz-Software etc. enthalten – alles Dinge, die früher oft mit einer DVD mitgeliefert wurden. Die Zeitschrift c't ist bei ihren Tests auf vorkonfigurierte Windows-Notebooks mit bis zu sechs Partitionen gestoßen.

Weitere Partitionen benötigen Sie, sobald Sie mehrere Betriebssysteme gleichzeitig auf Ihrem Rechner installieren möchten. Dafür gibt es zwei Gründe: Zum einen verwenden unterschiedliche Betriebssysteme oft auch unterschiedliche Dateisysteme, also unterschiedliche Verfahren, wie Dateien innerhalb der Partition abgelegt werden. Zum anderen vermeiden eigene Partitionen Doppelgleisigkeiten und Konflikte bei Verzeichnis- und Dateinamen.

Wozu noch mehr Partitionen?

Unter Linux kommt noch hinzu, dass es zumeist sinnvoll ist, für Linux selbst mehrere Partitionen vorzusehen – z. B. eine Partition für das Betriebssystem, eine weitere für Ihre eigenen Daten und eine dritte als sogenannte Swap-Partition. Dabei handelt es sich um das Gegenstück zur Auslagerungsdatei von Windows.

Für eine Linux-Installation kommt es also nicht darauf an, wie viel Platz auf Ihrer Festplatte unter Windows noch frei ist. Diesen Platz – innerhalb einer Windows-Partition – können Sie nämlich für Linux nicht nutzen. Sie benötigen für die Linux-Installation Platz *außerhalb* der bestehenden Windows-Partition(en), um dort neue Linux-Partitionen anzulegen.

#### Vorsicht, wenn Windows nicht erkannt wird ...

Die Installationsprogramme der meisten Distributionen bieten eine halb automatische Partitionierung an. Aufpassen müssen Sie dabei, was das Installationsprogramm mit den Windows-Partitionen machen möchte.

Vorsicht ist angebracht, wenn im Partitionierungsvorschlag keine Windows-Partitionen erscheinen. Dann hat das Installationsprogramm diese vermutlich nicht erkannt. Das kann z. B. bei Rechnern mit SSD-Cache passieren oder auf PCs mit mehreren Festplatten in einer RAID-Konfiguration. Führen Sie in solchen Fällen unbedingt eine manuelle Partitionierung durch!

**Partitionierungshilfen** Um die Aufteilung der Festplatte zu verändern, sieht jedes Betriebssystem eigene Werkzeuge vor:

- ▶ Unter Windows gibt es dafür ein komfortableres Werkzeug mit grafischer Benutzeroberfläche. Der Aufruf ist allerdings bei jeder Windows-Version ein wenig anders. Unter Windows 10 suchen Sie im Startmenü nach **FESTPLATTENPARTITIONEN ERSTELLEN UND FORMATIEREN** und starten so das Modul **DATENTRÄGERVERWALTUNG** aus den Systemeinstellungen.
- ▶ Unter Linux stehen je nach Installationsprogramm diverse Partitionierungshilfen zur Verfügung. Sollte es damit Probleme geben, können Linux-Profis auch auf das Kommando `parted` zurückgreifen.

### Mehr Flexibilität mit LVM

Die Partitionierung der Festplatte lässt sich nachträglich nur mit großem Aufwand ändern. In vielen Fällen geht der Inhalt einer Partition verloren, wenn deren Größe verändert wird. Auch ein Verschieben von Partitionen ist nicht vorgesehen. Daher ist es empfehlenswert, die Partitionierung von Anfang an gut zu bedenken.

Linux-Profis können viele Einschränkungen umgehen, indem sie das System LVM einsetzen (siehe [Abschnitt 2.7](#), »RAID, LVM und Verschlüsselung«). Dabei handelt es sich um eine Zwischenschicht zwischen Partitionen und Dateisystemen.

Unter Umständen auf LVM verzichten können Sie wiederum, wenn Sie Linux in virtuellen Maschinen oder in der Cloud ausgeführt. Dort ist es verhältnismäßig einfach, virtuelle Datenträger zu vergrößern oder hinzuzufügen.

**MBR versus GPT** Es gibt aktuell zwei Verfahren zur Verwaltung der Partitionierungsinformationen auf der Festplatte:

- ▶ **MBR:** Die Partitionierungskonzepte auf Basis der MBR-Partitionstabellen reichen bis in die DOS-Zeit zurück, und entsprechend angestaubt wirken manche Regeln und Einschränkungen. Dennoch gelten sie für die meisten Festplatten, die bis 2012 in Linux- oder Windows-PCs eingesetzt wurden. Die Partitionierungstabelle wird in diesem Fall im *Master Boot Record* (MBR) gespeichert, also im ersten Sektor der Festplatte. USB-Sticks und SD-Karten verwenden bis heute normalerweise MBR-Partitionstabellen.
- ▶ **GPT:** Um die vielen MBR-Einschränkungen zu umgehen, wurde schon vor mehr als einem Jahrzehnt ein neuer Standard geschaffen: *GUID Partition Tables*. Apple ist schon 2005 auf GPT umgestiegen. Der PC-Markt hat diesen Schritt im Herbst 2012 mit der Markteinführung von Windows 8 vollzogen.

## MBR-Grundlagen

Bei Festplatten mit MBR-Partitionierung gibt es drei Typen von Festplattenpartitionen: primäre, erweiterte und logische Partitionen. Auf der Festplatte können maximal vier primäre Partitionen existieren. Außerdem besteht die Möglichkeit, statt einer dieser vier primären Partitionen eine erweiterte Partition zu definieren. Innerhalb der erweiterten Partition können dann mehrere logische Partitionen angelegt werden.

Partitionstypen

Der Sinn von erweiterten und logischen Partitionen besteht darin, das historisch vorgegebene Limit von nur vier primären Partitionen zu umgehen. Beachten Sie, dass manche Partitionierwerkzeuge an der Oberfläche nicht zwischen verschiedenen Partitionstypen unterscheiden und sich selbstständig darum kümmern, wie die Partitionen intern angelegt werden.

Eine erweiterte Partition dient nur als Container für logische Partitionen. Zur eigentlichen Speicherung von Daten sind nur primäre und logische Partitionen geeignet.

Der Begriff »Partitionstyp« wird auch in einem anderen Kontext verwendet: Zusammen mit jeder Partition wird eine Zusatzinformation (eine Kennzahl) gespeichert, die angibt, für welches Betriebssystem die Partition gedacht ist (z. B. Windows, Linux, Novell Netware, BSD) bzw. welche Aufgabe der Partition zugeteilt ist.

Aktuelle Linux-Distributionen erlauben bei MBR-Partitionierung maximal drei primäre, eine erweiterte und darin 60 logische Partitionen. Es ist aber unwahrscheinlich, dass Sie wirklich so viele Partitionen benötigen – und wenn doch, sollten Sie sich besser mit LVM auseinandersetzen (siehe [Abschnitt 2.7](#), »RAID, LVM und Verschlüsselung«). Der *Linux Volume Manager* bietet viel mehr Flexibilität als die herkömmliche Partitionsverwaltung.

Maximalanzahl der Partitionen

Die maximale Partitionsgröße beträgt 2 TiB. Zur Not können Sie selbst Festplatten bis zu 4 TiB mit MBR-Partitionierung nutzen: Dazu lassen Sie die letzte primäre Partition gerade noch innerhalb der ersten 2 TiB beginnen und machen sie beinahe 2 TiB groß. Damit kann sie noch vollständig angesprochen werden – z. B. als Physical Volume für ein LVM-System. Empfehlenswert ist dieser Weg aber nicht. Steigen Sie besser auf GPT um!

Maximale Partitionsgröße

## GPT-Grundlagen

GPT steht für *GUID Partition Table*. Jede Partition wird durch einen *Global Unique Identifier* (GUID) gekennzeichnet. In der GPT-Partitionstabelle ist Platz für 128 Partitionen, die Sie alle unter Linux ansprechen können. Alle Partitionen sind gleichwertig, d. h., es gibt keine Unterscheidung zwischen primären, erweiterten und logischen Partitionen. Jede Partition kann bis zu 8 Zettabyte (ZiB) groß sein – also  $2^{73}$  Byte, das sind ca.  $9,4 \times 10^{21}$  Byte oder rund eine Milliarde TiB! Das sollte für die nächste Zeit reichen.

Die Partitionstabelle befindet sich in den ersten  $34 \times 512 = 17.408$  Byte der Festplatte. Eine Kopie dieser Informationen nimmt weitere 17 KiB am Ende der Festplatte in Anspruch. Aus Sicherheitsgründen beginnt die GPT-Partitionstabelle mit MBR-Partitionsinformationen, um MBR-kompatiblen Programmen den Eindruck zu vermitteln, die gesamte Festplatte würde bereits von einer Partition genutzt, die die gesamte Festplatte füllt.

Beachten Sie, dass die Partitionsnummern nicht mit der tatsächlichen Reihenfolge der Partitionen übereinstimmen müssen. Nehmen Sie an, Sie erzeugen drei Partitionen mit jeweils 20 GiB. Nun verkleinern Sie die zweite Partition auf 10 GiB. Damit entsteht zwischen den Partitionen 2 und 3 eine Lücke, in der Sie eine neue Partition einrichten können. Diese bekommt die Nummer 4 und entsprechend unter Linux den Device-Namen `/dev/sda4`!

Umfassende Informationen zum Aufbau der GPT-Partitionstabelle sowie zur Kompatibilität mit diversen Betriebssystemversionen gibt die englische Wikipedia-Seite:

[https://en.wikipedia.org/wiki/GUID\\_Partition\\_Table](https://en.wikipedia.org/wiki/GUID_Partition_Table)

**Umstieg auf GPT** Die meisten Linux-Installationsprogramme kommen zwar mühelos mit GPT-partitionierten Festplatten zurecht. Erstaunlicherweise gibt es aber kaum Distributionen, die Ihnen bei der Partitionierung neuer Festplatten die Wahl zwischen MBR und GPT lassen.

Die Umstellung einer Festplatte von MBR auf GPT bzw. die Initialisierung einer noch vollkommen leeren Festplatte mit einer GPT ist momentan also nur von Hand möglich. Dazu verwenden Sie am besten ein Linux-Live-System. Anschließend führen Sie das Kommando `parted` aus und darin wiederum den Befehl `mklabel gpt`. Damit wird die Partitionstabelle im GPT-Format neu eingerichtet. Beachten Sie aber, dass die folgenden Kommandos mit dem Verlust aller Daten auf der Festplatte verbunden sind!

```
root# parted /dev/sda
(parted) mklabel gpt
(parted) quit
```

### MBR oder GPT?

Bei Festplatten bis zu 2 TByte gibt es keinen zwingenden Grund für GPT. Persönlich richte ich allerdings schon seit geraumer Zeit auf allen neuen Festplatten und SSDs eine GPT ein. Damit erspare ich mir das Theater mit primären, erweiterten und logischen Partitionen. Nachteile sind mir keine aufgefallen.



## Festplatten mit 4-KiB-Sektoren

Neue Festplatten sowie Solid State Disks (SSDs) verwenden statt der jahrzehntelang üblichen 512-Byte-Sektoren längere Sektoren von 4096 Byte (4 KiB). Das hat viele Vorteile, unter anderem eine höhere Geschwindigkeit und eine höhere Festplattenkapazität. Aus Kompatibilitätsgründen melden aber auch Festplatten mit 4-KiB-Sektoren eine 512-Byte-Sektorgröße an das Betriebssystem.

Um Festplatten mit 4-KiB-Sektoren effizient zu nutzen, müssen Partitionen so eingerichtet werden, dass die Startposition jeder Partition ein Vielfaches von 4 KiB beträgt. Ist das nicht der Fall und will das Dateisystem einen 4-KiB-Bereich verändern, muss die Festplatte zwei 4-KiB-Sektoren lesen, modifizieren und schreiben. Das würde Schreibvorgänge massiv bremsen.

Die Installationsprogramme aller aktuellen Distributionen nehmen auf diesen Umstand mittlerweile Rücksicht. Aufpassen müssen nur Linux-Profis, die Festplatten mit Low-Level-Werkzeugen wie `parted` partitionieren. Dieses Kommando wird in [Abschnitt 22.5](#), »`parted`-Kommando«, näher vorgestellt.

## Dateisysteme

Durch das Partitionieren wird auf der Festplatte lediglich Platz reserviert. Bevor Sie in einer Partition Dateien speichern können, müssen Sie ein sogenanntes Dateisystem anlegen. Es enthält neben den eigentlichen Daten diverse Verwaltungsinformationen. Sowohl Windows als auch Linux kennen unterschiedliche Dateisystemtypen:

- ▶ Unter Windows sind VFAT (Windows 9x/ME) und NTFS (alle Versionen ab Windows NT) gebräuchlich. VFAT kommt darüber hinaus auch auf den meisten USB-Sticks sowie auf SD-Karten für Kameras, Smartphones etc. zum Einsatz.
- ▶ Unter Linux ist `ext4` der beliebteste Dateisystemtyp. Alternativen sind `xfs` (ideal für sehr große Dateisysteme) und `btrfs`.

Das Anlegen eines Dateisystems in einer Partition wird auch Formatieren genannt. Unter Windows können Sie diese Operation über ein Kontextmenü im Explorer oder mit dem Programm `FORMAT` durchführen. Bei einer Linux-Installation kümmert sich das Installationsprogramm um die Formatierung, wobei hinter den Kulissen ein Kommando wie `mkfs.ext4` zum Einsatz kommt.

### Achtung

Im Regelfall gehen sowohl durch die Partitionierung als auch durch das Formatieren alle in der betroffenen Partition gespeicherten Daten verloren! Es gibt jedoch auch Werkzeuge zur verlustfreien Größenänderung von Partitionen.

## Partitionsnamen

**Windows** Unter Windows werden Partitionen, die das Betriebssystem nutzen kann, mit Laufwerksbuchstaben bezeichnet. A: und B: sind aus historischen Gründen für Disketten reserviert. Die weiteren Buchstaben bezeichnen die primären und logischen Partitionen der Festplatte. Erweiterte Partitionen erhalten keinen Laufwerksbuchstaben und sind somit unsichtbar. Gleiches gilt für Partitionen mit fremden Dateisystemen (also z. B. Linux-Partitionen). Sie bekommen keinen Laufwerksbuchstaben und sind ebenfalls unsichtbar.

**Linux** Unter Linux erfolgt der interne Zugriff auf Festplatten bzw. deren Partitionen über sogenannte Device-Dateien (siehe [Tabelle 2.2](#)). Die Festplatten erhalten der Reihe nach die Bezeichnung `/dev/sda`, `/dev/sdb`, `/dev/sdc` etc.

| Device-Name            | Bedeutung  |
|------------------------|--|
| <code>/dev/sda</code>  | erste Festplatte/SSD   |
| <code>/dev/sdb</code>  | zweite Festplatte/SSD  |
| ...                    |  |
| <code>/dev/sda1</code> | die erste primäre Partition der Festplatte/SSD <code>/dev/sda</code> |
| <code>/dev/sda2</code> | die zweite primäre Partition   |
| <code>/dev/sda3</code> | die erweiterte Partition (nur MBR)                                   |
| <code>/dev/sda5</code> | die erste logische Partition (nur MBR)                               |
| <code>/dev/sda8</code> | die vierte logische Partition (nur MBR)                              |
| ...                    |  |

**Tabelle 2.2** Device-Namen von Festplatten- bzw. SSD-Partitionen

Um eine einzelne Partition und nicht die ganze Festplatte anzusprechen, wird der Name um die Partitionsnummer ergänzt. Bei der MBR-Partitionierung sind die Zahlen 1 bis 4 für primäre und erweiterte Partitionen reserviert.

Logische Partitionen beginnen mit der Nummer 5, auch dann, wenn es weniger als vier primäre oder erweiterte Partitionen gibt (so wie in [Tabelle 2.2](#), wo die dritte Partition der ersten Festplatte eine erweiterte Partition ist und es keine vierte erweiterte oder primäre Partition gibt). Bei der GPT-Partitionierung werden einfach alle Partitionen der Reihe nach durchnummeriert.

## 2.7 RAID, LVM und Verschlüsselung

Dieser Abschnitt führt in die Grundlagen von RAID und LVM ein und geht kurz auf das Thema Verschlüsselung ein. Wenn Sie vorhaben, eine ganz gewöhnliche Desktop-Installation auf einem Notebook bzw. Rechner mit nur einer Festplatte durchzuführen, können Sie diesen Abschnitt getrost überspringen. Wenn Sie aber eine Server-Installation planen, sollten Sie diesen Abschnitt in Ruhe lesen: Viele Linux-Installationsprogramme unterstützen alle drei Verfahren. Nur wenn Sie die zugrunde liegenden Techniken kennen, können Sie die Tragweite einer Entscheidung für oder wider den Einsatz von RAID, LVM und Verschlüsselungstechniken abschätzen.

### Redundant Array of Independent Disks (RAID)

Die Grundidee von RAID besteht darin, Partitionen mehrerer Festplatten logisch miteinander zu verknüpfen. Das Ziel ist dabei, ein zuverlässigeres und/oder schnelleres Gesamtsystem zu schaffen:

- ▶ Durch RAID kann die Datenübertragung gesteigert werden, indem der Datenzugriff von mehreren Festplatten parallel erfolgt.
- ▶ Durch RAID kann aber auch die Sicherheit gesteigert werden, indem Daten redundant (mehrfach) gespeichert werden. Das ist allerdings mit Geschwindigkeitseinbußen verbunden und beansprucht zusätzlichen Speicherplatz.

Es gibt zwei grundsätzliche Möglichkeiten, RAID zu realisieren: durch Hardware (also durch einen Festplattencontroller, der die RAID-Logik selbst ausführt) oder durch Software, die von der CPU des Rechners ausgeführt wird. Hardware-RAID kommt vor allem in teuren Server-Systemen zum Einsatz. Seine größten Vorzüge bestehen darin, dass die CPU nicht durch RAID-Aufgaben belastet wird und dass der RAID-Controller unabhängig vom Betriebssystem agiert.

Hardware- versus  
Software-RAID

Bei Software-RAID wird zwischen verschiedenen Formen unterschieden, je nachdem, woher die Software kommt:

- ▶ **Fake-RAID:** Beim Fake-RAID realisiert das BIOS bzw. EFI in Kombination mit einem Betriebssystemtreiber verschiedene RAID-Level. Der abfällige Begriff *Fake-RAID* erklärt sich daraus, dass viele RAID-Controller so angepriesen werden, als wären sie echte Hardware-RAID-Controller – und das ist unrichtig. In der Vergangenheit wurde Fake-RAID oft auch als BIOS-RAID bezeichnet. Der analoge Begriff EFI-RAID hat sich aber nicht eingebürgert.
- ▶ **Linux-Software-RAID:** Linux kann durch den *Multi Devices Driver Support* mehrere Festplatten(partitionen) zu einem RAID verbinden. Das ist genauso schnell wie Fake-RAID, lässt sich aber wesentlich besser administrieren. Aus Linux-Sicht

ist diese RAID-Variante vorzuziehen. Wenn in diesem Buch ohne weitere Erläuterungen von RAID die Rede ist, dann ist Linux-Software-RAID gemeint!

- ▶ **Windows-Software-RAID:** Auch Windows unterstützt verschiedene RAID-Varianten in Form von Software-RAID. Derart eingerichtete Windows-Dateisysteme sind für Linux nicht lesbar.

### Vermeiden Sie Fake-RAID!

Fake-RAID wird von vielen Distributionen nicht oder nur halbherzig unterstützt. Im schlimmsten Fall erkennt das Installationsprogramm ein vorhandenes Fake-RAID nicht und führt eine Neupartitionierung durch, was mit dem Verlust aller Daten einhergeht.

**RAID-Level** Es gibt verschiedene Verfahren, um Festplattenpartitionen zu verbinden. Diese Varianten werden als »RAID-Level« bezeichnet:

- ▶ **RAID-0 (Striping):** Bei RAID-0 werden mehrere physikalische Partitionen zu einer größeren Partition vereint. Dabei werden die Daten parallel in kleinen Blöcken (z. B. 4 KiB) auf die einzelnen Partitionen verteilt, sodass die Daten beim Zugriff alternierend von allen Festplatten gelesen werden. Daraus ergibt sich im Idealfall eine Vervielfachung der Datenrate (d. h. bei drei Festplatten eine Verdreifachung). In der Praxis ist der Effekt meist kleiner als erhofft und kommt nur bei großen Dateien wirklich zum Tragen. Die Anzahl der Random-Access-Zugriffe pro Sekunde wird durch das Striping nicht verbessert. RAID-0 hat einen gravierenden Nachteil: Das Ausfallrisiko ist hoch, weil *eine* defekte Festplatte zum Verlust *aller* Daten führt.
- ▶ **RAID-1 (Mirroring):** Bei RAID-1 werden dieselben Daten in der Regel auf zwei Festplatten gespeichert (selten auch auf mehr Festplatten). Wenn eine Festplatte ausfällt, stehen alle Daten auf der anderen Festplatte zur Verfügung. Der Vorteil ist die höhere Sicherheit, der Nachteil die halbierte Kapazität. RAID-1 bietet keine Geschwindigkeitsvorteile, vielmehr werden insbesondere Schreibvorgänge sogar ein wenig langsamer ausgeführt als bei der einfachen Verwendung einer Festplatte.
- ▶ **RAID-10:** RAID-10 kombiniert RAID-1 und RAID-0 und setzt mindestens vier Festplatten bzw. Partitionen voraus: Die Festplatten 1 und 2 bilden einen RAID-1-Verbund, die Festplatten 3 und 4 einen weiteren RAID-1-Verbund. Auf der nächsten Ebene werden die beiden RAID-1-Verbunde zu einem RAID-0-Verbund kombiniert. Damit kombiniert RAID-10 die Vorteile von RAID-0 (Geschwindigkeit) und RAID-1 (Sicherheit).
- ▶ **RAID-5 (Parity Striping):** RAID-5 funktioniert im Prinzip wie RAID-0, allerdings werden zusätzlich in einer (für jeden Datenblock wechselnden) Partition Paritätsinformationen gespeichert. Wenn eine Festplatte ausfällt, können die gesamten

Daten rekonstruiert werden. Der Ausfall von zwei oder mehr Festplatten führt allerdings zu einem kompletten Datenverlust. RAID-5 setzt zumindest drei Festplatten voraus.

RAID-5 ist ebenso sicher wie RAID-1 und bei Lesezugriffen etwa so schnell wie RAID-0. Zudem hat RAID-5 den Vorteil, dass der für die Redundanz erforderliche Datenanteil mit der Anzahl der Festplatten kleiner wird: Bei RAID-1 beträgt der Kapazitätsverlust immer 50 Prozent; bei RAID-5 beträgt er nur 33 Prozent bei drei Festplatten, 25 Prozent bei vier, 20 Prozent bei fünf etc.

RAID-5 hat gegenüber RAID-1 allerdings auch Nachteile: Zum einen sind Schreiboperationen langsamer als bei RAID-1, insbesondere wenn sich häufig kleine Datenmengen ändern. Der Grund ist, dass selbst bei kleinen Veränderungen die Paritätsinformationen für einen ganzen Datenblock neu berechnet und gespeichert werden müssen. Nach dem Austausch einer defekten Platte dauert die Rekonstruktion des RAID-5-Verbunds sehr lange, viel länger als bei RAID-1.

- ▶ **RAID-6:** RAID-6 funktioniert wie RAID-5, ist aber doppelt redundant und erfordert zumindest vier Festplatten. Selbst beim Ausfall von zwei Festplatten kommt es zu keinem Datenverlust.

Weitere RAID-Level sowie viele interessante Details und Grundlagen zu RAID finden Sie im folgenden Wikipedia-Artikel:

<https://de.wikipedia.org/wiki/RAID>

### Logical Volume Manager (LVM)

Der Logical Volume Manager setzt eine logische Schicht zwischen das Dateisystem und die Partitionen der Festplatte. Was zuerst sehr abstrakt klingt, hat in der Praxis durchaus handfeste Vorteile:

- ▶ Im Rahmen des von LVM verwalteten Festplattenbereichs können Sie im laufenden Betrieb ohne Rechnerneustart Partitionen anlegen, vergrößern und verkleinern. Den vorhandenen LVM-Speicherpool können Sie jederzeit durch den Einbau einer weiteren Festplatte vergrößern.
- ▶ Sie können dank LVM Bereiche mehrerer Festplatten zu einer einzigen, riesigen virtuellen Partition zusammenfassen.
- ▶ Sie können sehr einfach einen sogenannten Snapshot eines Dateisystems erstellen. Das ist ideal für Backups im laufenden Betrieb.
- ▶ LVM ist sehr schnell. Sie bezahlen für die höhere Flexibilität also nicht mit einer spürbar verringerten Geschwindigkeit. Der Geschwindigkeitsunterschied gegenüber dem direkten Ansprechen einer Festplattenpartition ist kaum messbar. Die CPU-Belastung ist nur geringfügig höher.

LVM kann mit RAID kombiniert werden, indem ein RAID-Verbund als Grundlage für LVM verwendet wird. In diesem Fall muss zuerst RAID und dann darauf aufbauend LVM konfiguriert werden.

**Glossar** Die Fülle ähnlich lautender Begriffe und Abkürzungen erschwert den Einstieg in die LVM-Welt. Um die Konfusion nicht noch zu vergrößern, verzichte ich in diesem Abschnitt bewusst auf eine Übersetzung der Begriffe. Zwischen der Festplatte und dem Dateisystem stehen drei Ebenen: Physical Volumes, Volume Groups und Logical Volumes:

- ▶ **Physical Volume (PV):** Ein PV ist im Regelfall eine von LVM verwaltete Partition der Festplatte. Es kann sich auch um eine ganze Festplatte oder um ein RAID-Device handeln. Entscheidend ist, dass die Partition, die Festplatte oder der RAID-Verbund als PV gekennzeichnet ist, damit die unterschiedlichen LVM-Kommandos funktionieren.
- ▶ **Volume Group (VG):** Ein oder mehrere Physical Volumes können zu einer Gruppe zusammengefasst werden. Auf diese Weise ist es möglich, Partitionen unterschiedlicher Festplatten quasi zusammenzuhängen, also einheitlich zu nutzen. Die Volume Group stellt eine Art Speicherpool dar, der alle zur Verfügung stehenden physikalischen Speichermedien vereint. Dieser Pool kann jederzeit um weitere Physical Volumes erweitert werden.
- ▶ **Logical Volume (LV):** Ein Logical Volume ist ein Teil der Volume Group. Für den Anwender wirkt ein Logical Volume wie eine virtuelle Partition. Im Logical Volume wird das Dateisystem angelegt. Das heißt, anstatt ein Dateisystem in `/dev/sda7` anzulegen, geben Sie jetzt den Device-Namen des Logical Volume an. Falls in der Volume Group noch Speicher verfügbar ist, können Logical Volumes jederzeit vergrößert werden.

In der LVM-Dokumentation kommen noch zwei weitere Begriffe häufig vor:

- ▶ **Physical Device (PD):** Dabei handelt es sich einfach um eine Festplatte. LVM kann die gesamte Festplatte oder auch Partitionen dieser Festplatte in Form von Physical Volumes nutzen.
- ▶ **Physical Extent (PE):** Bei Volume Groups und Logical Volumes kann nicht jedes einzelne Byte einzeln verwaltet werden. Die kleinste Dateneinheit ist vielmehr ein Physical Extent (standardmäßig 4 MiB). Die Anzahl der PEs ist unbegrenzt. Zu viele PEs machen aber die Verwaltung ineffizient, weswegen Sie für sehr große Logical Volumes die Größe von PEs hinaufsetzen sollten.

**Beispiel** Das Beispiel aus [Abbildung 2.3](#) veranschaulicht die oben definierten Begriffe: Auf einem System dienen die beiden Partitionen `/dev/sda3` und `/dev/sdb1` als Physical Volumes für eine Volume Group eines LVM-Systems. `/dev/sda3` umfasst 400 GiB, `/dev/sdb1` umfasst 900 GiB.

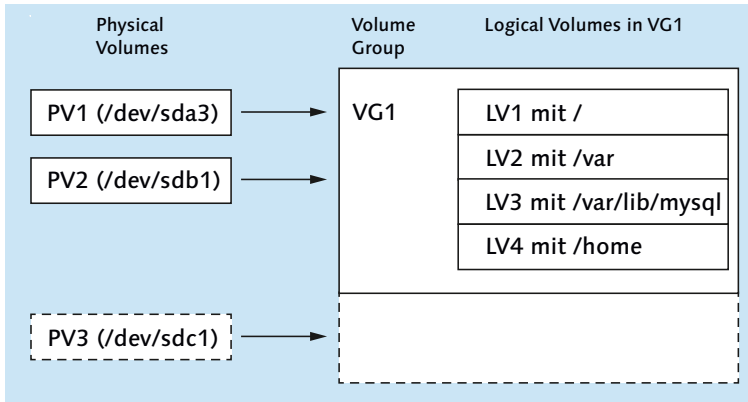


Abbildung 2.3 LVM-System

Der LVM-Speicherpool (also die Volume Group) ist somit 1,3 TiB groß. Darin befinden sich nun diverse Logical Volumes:

- LV1 mit der Systempartition (50 GiB)
- LV2 mit der Partition `/var` (200 GiB)
- LV3 mit der Partition `/var/lib/mysql` (200 GiB)
- LV4 mit der Partition `/home` (400 GiB)

Insgesamt sind somit 850 GiB mit Partitionen belegt, und 450 GiB sind noch frei. Damit können Sie zu einem späteren Zeitpunkt vorhandene Partitionen vergrößern oder neue Partitionen anlegen. Sollte der gesamte LVM-Pool erschöpft sein, können vorhandene LVs/Dateisysteme verkleinert werden (wenn sich herausgestellt hat, dass sie ursprünglich zu großzügig dimensioniert wurden), um so Platz zur Vergrößerung anderer LVs/Dateisysteme zu schaffen. Reicht das nicht aus, bauen Sie eine weitere Festplatte ein und fügen eine Partition dieser Festplatte als drittes Physical Volume zur Volume Group hinzu.

## Verschlüsselung

Viele Distributionen bieten die Möglichkeit, die Installation in verschlüsselten Partitionen durchzuführen bzw. zumindest die Partition für die persönlichen Daten verschlüsselt anzulegen. Beim Systemstart muss dann ein Passwort angegeben werden, bevor auf das Dateisystem zugegriffen werden kann. Sofern Sie ein ausreichend langes und nicht erratbares Passwort verwenden, schützt die Verschlüsselung Ihre Daten wirkungsvoll: Auch wenn Ihr Notebook in falsche Hände gelangt, kann niemand Ihre Dateien lesen.

Wahrscheinlich fragen Sie sich, was Verschlüsselung mit RAID und LVM zu tun hat: Die meisten Verschlüsselungssysteme beruhen darauf, dass das verschlüsselte Dateisystem nicht direkt angesprochen wird, sondern über eine Zwischenschicht, die für die Verschlüsselung verantwortlich ist. Technisch gesehen ist die Vorgehensweise ganz ähnlich wie bei LVM. Deswegen gelten dieselben Einschränkungen wie bei LVM.

### Einschränkungen

Der Einsatz von RAID, LVM und Verschlüsselung hat nicht nur Vorteile, sondern ist auch mit diversen Einschränkungen bzw. Nachteilen verbunden:

- ▶ Die Administration von LVM und RAID ist relativ kompliziert. Während der Installation unterstützt das Installationsprogramm Sie beim Einrichten von LVM, RAID bzw. der verschlüsselten Partition. Wenn Sie dann aber im laufenden Betrieb die Konfiguration verändern möchten, sind Sie bei den meisten Distributionen auf relativ sperrige Kommandos angewiesen. Ausführliche Informationen zum Umgang mit diesen Kommandos finden Sie in [Kapitel 22](#), »Administration des Dateisystems«.
- ▶ Wenn in einem RAID-Verbund ein Problem auftritt, wird die Fehlermeldung üblicherweise per E-Mail versandt. Das setzt voraus, dass auf dem Rechner ein E-Mail-Server läuft. Dessen sichere Konfiguration und Administration ist nicht ganz trivial. Gerade auf Privat-PCs gibt es normalerweise keinen Grund, einen eigenen E-Mail-Server zu betreiben.
- ▶ Swap-Partitionen sollten aus Performance-Gründen in gewöhnlichen Partitionen ohne die Verwendung von LVM oder RAID angelegt werden, idealerweise auf jeder Festplatte/SSD eine. Wenn Ihnen optimale Sicherheit wichtiger ist als maximale Geschwindigkeit, sollten Sie bei RAID-Systemen allerdings auch die Swap-Partition innerhalb des RAID-Verbunds einrichten. Das stellt sicher, dass auch bei einem Festplattenausfall keine Daten des Swap-Speichers verloren gehen.
- ▶ Wenn Sie ein verschlüsseltes Dateisystem einsetzen, um Ihre Daten zu schützen, sollte auch die Swap-Partition verschlüsselt werden. Noch besser ist es zumeist, auf die Swap-Partition gleich ganz zu verzichten. Was nützt es, wenn Ihr Dateisystem sicher ist, die Swap-Partition aber ausgelagerte Speicherblöcke mit unverschlüsselten kritischen Daten enthält?
- ▶ Während LVM und RAID die Geschwindigkeit Ihres Systems kaum beeinträchtigen und manche RAID-Level sogar zu einem besseren Datenumsatz führen können, kostet die Verschlüsselung viel CPU-Kapazität und verlangsamt Lese- und Schreiboperationen spürbar. Ein weiterer Nachteil besteht darin, dass das Verschlüsselungspasswort bei jedem Rechnerstart manuell eingegeben werden muss. Prinzipbedingt ist die Verschlüsselung ganzer Dateisysteme somit ungeeignet für Server, die automatisch (neu) starten sollen.



Kurz und gut: Bei allen Vorteilen, die mit RAID, LVM und diversen Verschlüsselungstechniken verbunden sind, nimmt die Komplexität des Gesamtsystems doch sehr stark zu.

### Empfehlung

Linux-Einsteigern rate ich, wegen der damit verbundenen Komplexität auf RAID, LVM und Verschlüsselung gleichermaßen zu verzichten. Das gilt insbesondere, wenn Sie RAID-1 oder RAID-5 einsetzen möchten, um eine höhere Datensicherheit zu erzielen!

Sicherheit vor Datenverlusten haben Sie nur, wenn Sie nach dem Ausfall einer Festplatte auch in der Lage sind, die richtigen Kommandos auszuführen, um die defekte Platte zu deaktivieren und um dem RAID-Verbund eine neue Festplatte hinzuzufügen. Linux-Einsteiger sind damit sicherlich überfordert, vor allem, wenn sie aufgrund des drohenden Datenverlusts gerade unter Stress stehen. Aus diesem Grund sind einfache, aber konsequent durchgeführte Backups besser als eine technisch noch so hervorragende RAID-Konfiguration!

Beim Einrichten eines Servers sind RAID und LVM dagegen empfehlenswert, aber selbst da lautet die goldene Regel KISS (*Keep it simple, stupid!*, sinngemäß also: *Mach's einfach, Dummkopf!*).

Persönlich bevorzuge ich in solchen Fällen den Einsatz zweier gleich großer Festplatten oder SSDs, auf denen ich jeweils drei Partitionen einrichte. Diese Partitionen verbinde ich zu drei RAID-1-Verbunden für die Boot-Partition, die Swap-Partition und für LVM (siehe [Abbildung 2.4](#)). Im LVM-Bereich richte ich dann nach Bedarf die Root- sowie diverse Datenpartitionen ein.

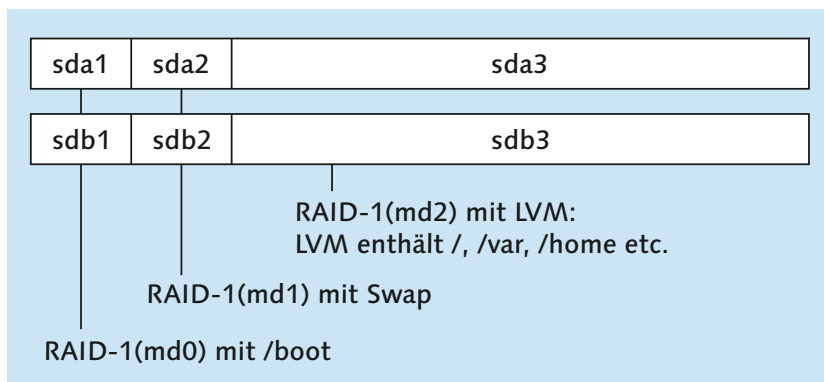


Abbildung 2.4 Server-Konfiguration mit RAID-1 und LVM auf zwei Festplatten

### Zwei Hersteller sind sicherer als einer

Aus Sicherheitsgründen ist es bei RAID-Konfigurationen empfehlenswert, Festplatten unterschiedlicher Hersteller einzusetzen! Wenn Sie nämlich zwei baugleiche Festplatten kaufen (ich weiß, die Versuchung ist groß) und der Hersteller gerade Fertigungsprobleme hatte, kann es Ihnen passieren, dass beide Festplatten innerhalb weniger Tage ausfallen.

## 2.8 Partitionierung der Festplatte

Einer der wichtigsten Schritte während der Linux-Installation ist das Anlegen neuer Linux-Partitionen. Alle gängigen Installationsprogramme enthalten zu diesem Zweck einfach zu bedienende Partitionierungshilfen. [Abbildung 2.5](#) zeigt den Partitionseditor von Ubuntu, ausgeführt auf einem Testrechner mit vielen Partitionen.

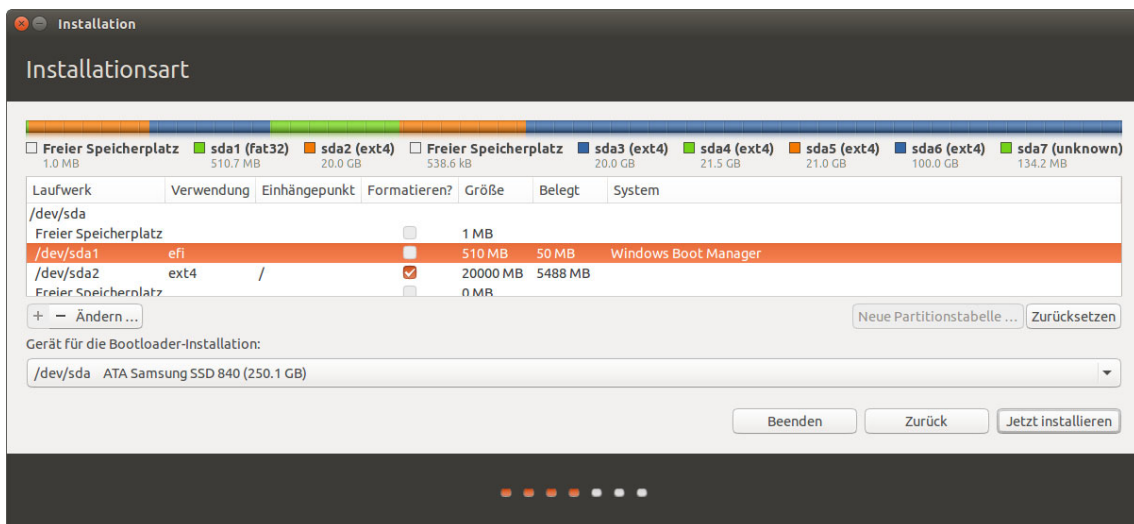


Abbildung 2.5 Ubuntu-Partitionseditor

An dieser Stelle geht es um grundsätzliche Fragen: Wie viele Partitionen sollten Sie für Linux einrichten? In welcher Größe? Welche Auswirkungen hat dies auf die Geschwindigkeit, auf die spätere Wartung und auf eine eventuelle Neuinstallation einer anderen oder aktualisierten Linux-Distribution?

### Partitionierung im laufenden Betrieb ändern

Wenn Sie Linux bereits installiert haben und im laufenden Betrieb eine neue Partition anlegen möchten, brauchen Sie ein Partitionierwerkzeug, das unabhängig vom Installationsprogramm Ihrer Distribution funktioniert. Dazu zählen die Programme parted und gparted, die ich Ihnen in [Kapitel 22](#), »Administration des Dateisystems«, näher vorstelle.

## Windows-Partition verkleinern

Oft befindet sich das bereits installierte Windows in einer einzigen, sehr großen Partition, die nahezu die gesamte Festplatte ausfüllt. Dass innerhalb dieser Partition womöglich Hunderte GiB frei sind, nützt nichts: Linux braucht für die Installation eine oder besser gleich mehrere eigene Partitionen. Und bevor Sie diese Partitionen anlegen können, müssen Sie die Windows-Partition verkleinern – und das möglichst ohne Datenverlust!

Die radikalere und einfachere Lösung bestünde darin, die Windows-Partition(en) einfach zu löschen. Aber die meisten Linux-Umsteiger wollen Windows als alternatives Betriebssystem vorerst erhalten – beispielsweise zum Spielen oder zur Ausführung von Programmen, die es unter Linux nicht gibt. Deswegen gehe ich in diesem Buch davon aus, dass Windows bereits installiert ist und auch weiterhin genutzt werden soll.

Bei den meisten Distributionen ist das Installationsprogramm selbst in der Lage, die Windows-Partition und das darin befindliche Dateisystem zu verkleinern. Je nach Distribution ändern Sie die Größe der Windows-Partition einfach im Partitionierungsprogramm oder rufen die entsprechende Verkleinerungsfunktion über ein Menü auf. Die Verkleinerung funktioniert sowohl für VFAT- als auch für NTFS-Dateisysteme.

Verkleinerung  
während der  
Installation

### Sie müssen Windows vollständig herunterfahren!

Eine Verkleinerung von Windows-Partitionen unter Linux ist nur möglich, wenn Sie Windows vorher vollständig herunterfahren. Bei einem gewöhnlichen Ausschalten ist das aber nicht der Fall! Vielmehr wird Windows für einen späteren Schnellstart in einen speziellen Modus heruntergefahren. Um ein »richtiges« Herunterfahren zu erzwingen, schließen Sie zuerst alle Programme und starten das Eingabeaufforderungsprogramm `cmd.exe` mit Administratorrechten. Dort führen Sie den Befehl `shutdown /p` aus.

Wenn eine Verkleinerung der Windows-Partition durch das Linux-Installationsprogramm scheitert, können Sie diesen Schritt auch *vor* der Installation durch andere Werkzeuge vornehmen. Hier eine kleine Auswahl:

Verkleinern vor  
der Installation

- ▶ **Direkt unter Windows:** Unter Windows ist eine verlustfreie Verkleinerung von Windows-Partitionen im laufenden Betrieb möglich. Unter Windows 10 suchen Sie im Startmenü nach `FESTPLATTENPARTITIONEN ERSTELLEN UND FORMATIEREN` und starten so das Modul `DATENTRÄGERVERWALTUNG` aus den Systemeinstellungen. Dort klicken Sie die Windows-Partition mit der rechten Maustaste an und führen `VOLUME VERKLEINERN AUS`.

- ▶ **Mit einem Live-System:** Live-Systeme wie Knoppix, GParted oder SystemRescueCD enthalten verschiedene Kommandos bzw. Programme, um Windows-Partitionen zu verkleinern. Die Bedienung dieser Werkzeuge ist allerdings teilweise kompliziert.
- ▶ **Kommerzielle Programme:** Den größten Komfort bieten kommerzielle Partitionierungsprogramme, die aber leider relativ teuer sind:

*<http://acronis.com/en-eu/personal/disk-manager>*

### Windows und Linux neu installieren

Falls auf Ihrem Rechner noch gar kein Betriebssystem installiert ist und Sie vorhaben, sowohl Windows als auch Linux zu installieren, sollten Sie mit Windows beginnen. Auch während der Windows-Installation müssen Sie die Festplatte partitionieren. Geben Sie hier an, dass die Windows-Partition nicht die ganze Festplatte füllen soll, sondern nur so viele GiB, wie Sie unter Windows eben nutzen möchten (z. B. 200 GiB).

### Zweier-Potenzen versus Zehner-Potenzen

In diesem Buch, im Großteil der sonstigen Linux-Dokumentation und für die meisten Linux-Werkzeuge werden Bytes in Zweier-Potenzen gerechnet:

1 KiB =  $2^{10}$  Byte = 1024 Byte

1 MiB =  $2^{20}$  Byte =  $1024^2$  Byte = 1.048.576 Byte

1 GiB =  $2^{30}$  Byte =  $1024^3$  Byte = 1.073.741.824 Byte

1 TiB =  $2^{40}$  Byte =  $1024^4$  Byte = 1.099.511.627.776 Byte

Viele Festplattenhersteller und auch manche Dateimanager rechnen dagegen dezimal, also mit 10er-Potenzen: 1 TB = 1 TByte =  $10^{12}$  Byte = 1.000.000.000.000 Byte. Damit hat eine Festplatte, die laut Hersteller 1 TByte umfasst, gemäß den Konventionen in diesem Buch nur ca. 931 GiB.

### Anzahl und Größe von Linux-Partitionen

Immer wieder wird mir die Frage gestellt, wie eine Festplatte mit  $n$  GiB am besten in Partitionen zerlegt werden soll. Leider gibt es darauf keine allgemeingültige Antwort. Dieser Abschnitt soll Ihnen aber zumindest ein paar Faustregeln für die richtige Anzahl und Größe von Partitionen vermitteln.

### Systempartition

Die Systempartition ist die einzige Partition, die Sie unbedingt benötigen. Sie nimmt das Linux-System mit all seinen Programmen auf. Diese Partition bekommt immer den Namen `/`. Dabei handelt es sich genau genommen um den Punkt, an dem die Partition in das Dateisystem eingebunden wird (den `mount`-Punkt). Wenn das System also einmal läuft, sprechen Sie diese Partition mit dem Pfad `/` an. `/` bezeichnet die Wurzel, also den Anfang des Dateisystems. Aus diesem Grund wird die Systempartition oft als Root-Partition bezeichnet.

Eine vernünftige Größe für die Installation und den Betrieb einer gängigen Distribution für den Desktop-Einsatz liegt bei 25 GiB. Dazu kommt natürlich noch der Platzbedarf für Ihre eigenen Daten – es sei denn, Sie speichern eigene Dateien in einer separaten Datenpartition.

Bis ca. 2010 war es oft erforderlich, eine eigene Boot-Partition mit dem Namen `/boot` anzulegen. Wenn es eine derartige Partition gibt, beherbergt sie lediglich die Dateien, die während der ersten Phase des Rechnerstarts benötigt werden. Dabei handelt es sich insbesondere um die Kerneldatei `vmlinuz*` und um die Initial-RAM-Disk-Datei `initrd*`.

**Boot-Partition**

Seit sich EFI und der Bootloader GRUB 2 durchgesetzt haben, ist die Boot-Partition selbst bei LVM- und Software-RAID-Setups selten erforderlich. Dennoch schlagen manche Distributionen (CentOS, RHEL) weiterhin vor, diese Partition einzurichten. Sie können diesem Vorschlag bedenkenlos folgen: Auch wenn die Partition nicht unbedingt erforderlich ist, richtet sie auf jeden Fall keinen Schaden an.

Wenn Sie sich für eine eigene Boot-Partition entscheiden, sollte diese rund 1 GiB groß sein. Gerade unter Ubuntu bereiten zu kleine Boot-Partitionen oft Probleme bei Kernel-Updates: Alte Kernelversionen werden standardmäßig nicht automatisch gelöscht, sodass die Boot-Partition früher oder später vollläuft.

Mit einer Datenpartition trennen Sie den Speicherort für die Systemdateien und für Ihre eigenen Dateien. Das hat einen wesentlichen Vorteil: Sie können später problemlos eine neue Distribution in die Systempartition installieren, ohne die davon getrennte Datenpartition mit Ihren eigenen Daten zu gefährden.

**Home-Partition**

Bei der Datenpartition wird üblicherweise `/home` als Name bzw. mount-Punkt verwendet, weswegen oft von der Home-Partition die Rede ist. Es ist nicht möglich, eine Empfehlung für die Größe der Datenpartition zu geben – das hängt zu sehr davon ab, welche Aufgaben Sie mit Ihrem Linux-System erledigen möchten. Wenn Sie sich sicher sind, dass Sie auf Ihrem Rechner keine weiteren Betriebssysteme mehr installieren möchten, können Sie die Home-Partition so groß machen, dass sie den gesamten Rest der Festplatte bzw. SSD füllt.

Die Aufteilung der Festplatte in Partitionen lässt sich noch weiter treiben. Wenn Sie den Linux-Rechner beispielsweise innerhalb eines größeren Netzwerks als speziellen Server für Netzwerk- oder Datenbank-Aufgaben einsetzen möchten, können Sie für die dabei anfallenden Daten eigene Partitionen vorsehen und ein für die Art des Datenzugriffs optimales Dateisystem auswählen. Diese Art der Optimierung ist allerdings nur für Linux-Experten zweckmäßig.

**Weitere Datenpartitionen**

Sofern auf Ihrer Festplatte noch unpartitionierter Platz frei ist, stellt es kein Problem dar, ein laufendes System um weitere Partitionen zu erweitern und gegebenenfalls

Daten von einer vorhandenen Partition in eine neue zu verschieben. Wenn Sie also unsicher sind, warten Sie mit der Partitionierung vorerst einfach noch ein wenig ab und lassen einen Teil der Festplatte ohne Partitionen.

**Swap-Partition** Die Swap-Partition ist das Gegenstück zur Auslagerungsdatei von Windows: Wenn Linux zu wenig RAM hat, lagert es Teile des gerade nicht benötigten RAM-Inhalts dort hin aus. Die Verwendung einer eigenen Partition statt einer gewöhnlichen Datei wie unter Windows hat vor allem Geschwindigkeitsvorteile. Linux kann zwar ebenfalls so konfiguriert werden, dass es statt einer Swap-Partition eine Swap-Datei verwendet. Das ist aber etwas langsamer.

Im Gegensatz zu den anderen Partitionen bekommt die Swap-Partition keinen Namen (keinen `mount`-Punkt). Der Grund: Aus Effizienzgründen wird die Swap-Partition direkt und nicht über ein Dateisystem angesprochen.

Wenn Sie viel RAM haben, können Sie ganz auf die Swap-Partition verzichten. Empfehlenswert ist dies aber nur, wenn Sie ohnedies schon unter argen Platzproblemen auf der Festplatte/SSD leiden.

Früher galt als Richtgröße für die Swap-Partition die Größe des RAMs. Bei einer zeitgemäßen RAM-Ausstattung (4 GiB und mehr) ist eine ebenso große Swap-Partition aber selten zweckmäßig. Derart viel Platz in der Swap-Partition benötigen Sie dann eigentlich nur, wenn Sie den Ruhestandsmodus eines Notebooks verwenden möchten (*Suspend to Disk*). Nach meinen Erfahrungen bereitet dieser Modus unter Linux aber so oft Probleme, dass ich von seiner Nutzung generell abrate. Fazit: 2 bis 4 GiB Platz für die Swap-Partition sind vollkommen ausreichend.

### Swap-Datei unter Ubuntu

Ubuntu richtet seit Version 17.04 bei Desktop-Installationen standardmäßig keine Swap-Partition mehr ein. Stattdessen sieht das Installationsprogramm ähnlich wie unter Windows oder macOS eine Swap-Datei vor. Das ist zwar minimal langsamer, bietet aber auch mehr Flexibilität, wenn die Swap-Datei später verkleinert oder vergrößert werden soll. Außerdem spart es eine Partition ein, die später nur noch schwer verändert werden kann.

**EFI-Partition** Auf EFI-Systemen muss es *eine* EFI-Partition geben. Diese Partition wird bei der Installation des ersten Betriebssystems standardmäßig eingerichtet, egal ob es sich um Windows oder Linux handelt. Wenn später weitere Betriebssysteme installiert werden, teilen sich alle Betriebssysteme die gemeinsame EFI-Partition und legen dort jeweils Dateien zum Start des Betriebssystems ab. Die EFI-Partition muss ein VFAT-Dateisystem enthalten und wird unter Linux über das Verzeichnis `/boot/efi` angesprochen.

Eine BIOS-GRUB-Partition ist nur in der äußerst seltenen Konstellation erforderlich, bei der Ihr Rechner ein BIOS (kein EFI) zum Hochfahren verwendet und die Festplatte oder SSD gleichzeitig eine GPT enthält (keine MBR-Partitionstabelle). Die üblicherweise nur 1 MiB große Partition dient dann als Ort zur Installation des Bootloaders. Diese Partition muss nicht formatiert werden. Dafür muss aber das Flag `bios_grub` gesetzt werden.

BIOS-GRUB-  
Partition

Bei jeder Linux-Installation benötigen Sie eine Systempartition. Darüber hinaus ist eine Swap-Partition sehr zu empfehlen. Das Einrichten weiterer Partitionen ist optional, sehr stark von der geplanten Anwendung von Linux abhängig und auch eine Geschmacksfrage. Meine persönliche Empfehlung für eine Linux-Erstinstallation ist in [Tabelle 2.3](#) zusammengefasst.

Fazit

| Verzeichnis | Verwendung   |
|-------------|--|
| /boot/efi   | EFI-Partition (bei EFI-Systemen, ca. 200 MiB)                      |
|             | BIOS-GRUB-Partition (1 MiB, nur für die Kombination BIOS plus GPT) |
|             | Swap-Partition (2 bis 4 GiB bzw. RAM-Größe für Suspend-to-Disk)    |
| /           | Systempartition (ca. 25 GiB)                                       |
| /home       | Datenpartition (Größe je nach geplanter Nutzung)                   |

**Tabelle 2.3** Empfohlene Partitionen für den Desktop-Einsatz

### Welches Dateisystem?

Linux unterstützt eine Menge unterschiedlicher Dateisysteme, unter anderem `ext4`, `btrfs` und `xfs`. Im Detail werden diese Dateisysteme in [Kapitel 22](#), »Administration des Dateisystems«, vorgestellt.

Der populärste Dateisystemtyp für Linux ist `ext4`. Dieses Dateisystem ist seit vielen Jahren ausgereift und gleichermaßen robust und effizient.

`ext4`

`xfs` galt schon lange als Geheimtipp für TiB-große Dateisysteme. Gleichsam der Ritter-schlag für dieses ursprünglich von SGI entwickelte Dateisystem war die Entscheidung von Red Hat, `xfs` als Defaultdateisystem für RHEL 7 zu verwenden. Seither nutzen immer mehr Linux-Anwender dieses robuste und schnelle Dateisystem als Alternative zu `ext4`.

`xfs`

Dessen ungeachtet gilt `btrfs` als *das* Linux-Dateisystem der Zukunft. Die etwas philosophische Frage ist freilich, ob die Zukunft schon da ist. Aktuell (im Sommer 2017) sind nur die SUSE-Entwickler dieser Ansicht und verwenden `btrfs` als Defaultdateisystem für die Systempartition.

`btrfs`

Desktop-Anwendern rate ich aber ausdrücklich von `btrfs` ab! `btrfs` erfordert insbesondere im SUSE-Default-Setup ein hohes Know-how bei der Administration und führt leicht zu unerwarteten Problemen. Die `btrfs`-Vorteile überwiegen bestenfalls bei Server-Installationen, die von erfahrenen Administratoren betreut werden. Persönlich nehme ich momentan auch bei der Server-Installation von `btrfs` Abstand. Die Einfachheit und Stabilität von `ext4` und `xfs` sind für mich wichtigere Argumente als alle Zusatzfeatures von `btrfs` zusammen.

**Swap-Partition** In der Swap-Partition wird *kein* richtiges Dateisystem eingerichtet! Die Partition muss aber vor der ersten Verwendung durch `mkswap` formatiert werden. Alle Linux-Distributionen kümmern sich automatisch darum.

**BIOS-GRUB-Partition** Auch in der BIOS-GRUB-Partition wird *kein* richtiges Dateisystem eingerichtet! Die Partition muss aber mit dem Flag `bios_grub` gekennzeichnet werden.

**Fazit** Tabelle 2.4 fasst zusammen, welche Dateisysteme Sie am besten für welche Partitionen einsetzen. Die Empfehlungen gelten für eine gewöhnliche Installation als Desktop- oder Entwicklungssystem.

| Partition           | Dateisystem                   |
|---------------------|-------------------------------|
| EFI-Partition       | VFAT (Windows)                |
| BIOS-GRUB-Partition | kein Dateisystem erforderlich |
| Swap-Partition      | kein Dateisystem erforderlich |
| /                   | ext4                          |
| /boot               | ext4                          |
| /home               | ext4 oder xfs                 |

**Tabelle 2.4** Empfohlene Dateisystemtypen für den Desktop-Einsatz

## 2.9 Installationsumfang festlegen (Paketauswahl)

Bei einigen Distributionen können Sie während der Installation auswählen, welche Komponenten, Programme bzw. Pakete installiert werden. Aus verschiedenen Gründen ist es nicht sinnvoll, einfach alles zu installieren:

- ▶ Die riesige Anzahl der verfügbaren Software-Pakete überfordert Einsteiger. Erheblich übersichtlicher ist es, vorerst nur eine Grundinstallation durchzuführen und die benötigten Zusatzprogramme später bei Bedarf nachzuinstallieren.



- ▶ Es gibt Programme, die sich gegenseitig im Weg sind. So können Sie beispielsweise auf einem Rechner nicht zwei verschiedene E-Mail-Server gleichzeitig betreiben. Sie müssen sich für eine Variante entscheiden.
- ▶ Wenn Sie vorhaben, den Rechner als Netzwerk-Server einzusetzen, vergrößert jeder aktive Netzwerkdienst die potenziellen Sicherheitsrisiken. Pakete für Netzwerkfunktionen, die Sie nicht benötigen, sollten Sie gar nicht erst installieren.

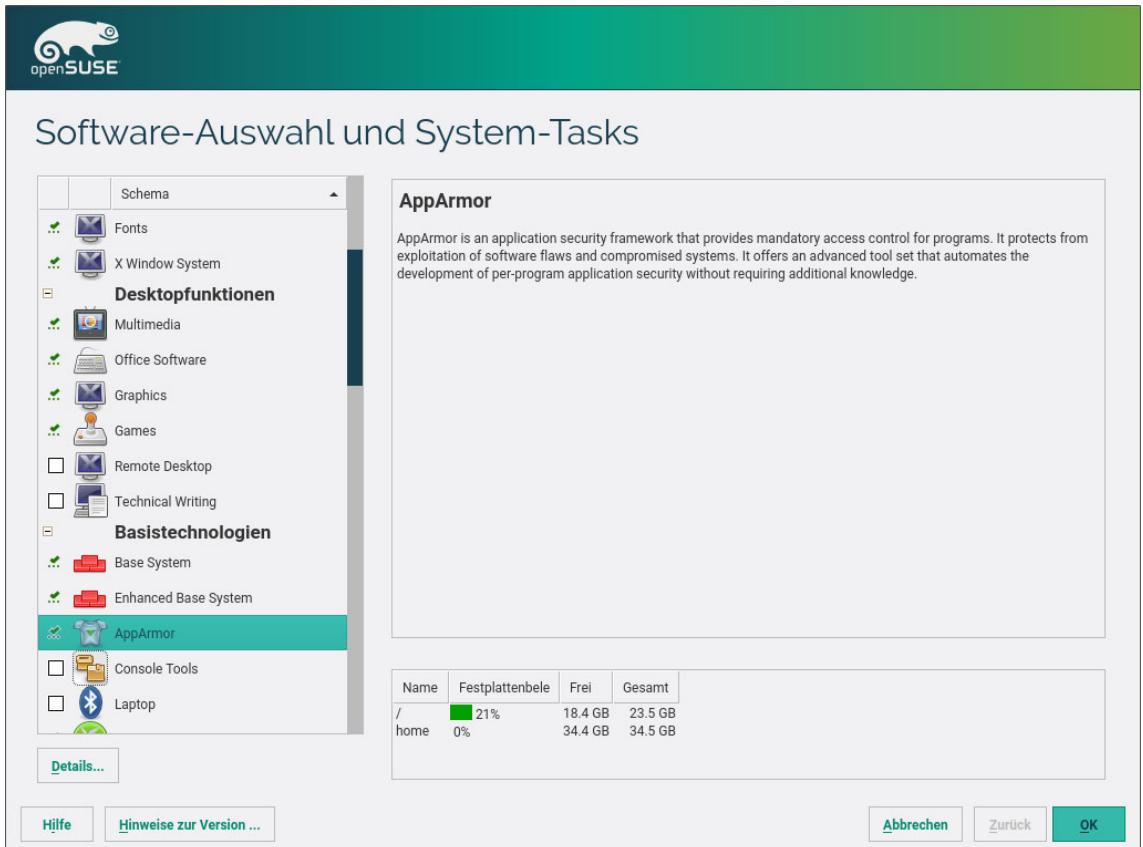


Abbildung 2.6 Paketauswahl während einer openSUSE-Installation

Die Auswahl der Software-Pakete erfolgt oft in Form von vorkonfigurierten Gruppen (siehe [Abbildung 2.6](#)). Es gibt auch Distributionen wie Ubuntu, bei denen Sie während der Installation gar keinen Einfluss auf die Paketauswahl haben: In diesem Fall wird einfach nur ein Grundsystem installiert. Auch bei den meisten Installationsprogrammen, die aus einem Live-System heraus gestartet werden, ist eine Paketauswahl unmöglich – es wird einfach das gesamte Live-System auf die Festplatte übertragen. In beiden Fällen installieren Sie alle weiteren Programme erst später bei Bedarf.

### Installationsempfehlungen

- Gnome oder KDE** Bei manchen Distributionen haben Sie die Wahl zwischen den Desktop-Systemen KDE und Gnome bzw. können sogar beide Systeme parallel installieren. Dabei handelt es sich um unterschiedliche Benutzeroberflächen zu Linux. Kurz gefasst: Gnome ist einfacher zu bedienen, dafür bietet KDE für technisch versierte Nutzer mehr Funktionen und Einstellmöglichkeiten. Bei einer Parallelinstallation haben Sie maximale Flexibilität und können bei jedem Login auswählen, ob Sie mit Gnome oder KDE arbeiten.
- Netzwerkdienste (Server)** Für die Desktop-Anwendung von Linux brauchen Sie üblicherweise keinen Web-, Mail- oder Datenbank-Server. Es gibt allerdings drei Ausnahmen:
- ▶ Um Ihren Drucker verwenden zu können, brauchen Sie einen Drucker-Server. Bei den meisten Distributionen wird dazu CUPS standardmäßig installiert.
  - ▶ Um Ihren Rechner über das Netzwerk steuern zu können, sollten Sie einen SSH-Server installieren.
  - ▶ Wenn Sie eigene Verzeichnisse mit Windows-Rechnern im lokalen Netz teilen möchten, müssen Sie das Programm Samba installieren.
- Entwicklungs-  
werkzeuge und  
Kernel-Header** Gerade Linux-Einsteiger haben vermutlich wenig Ambitionen, den Linux-Kernel neu zu kompilieren. Dennoch ist die Installation der elementaren Entwicklungswerkzeuge (C-Compiler, `make` etc.) und der sogenannten Kernel-Header-Dateien empfehlenswert. Damit sind Sie in der Lage, selbst neue Kernelmodule zu kompilieren. Das ist erforderlich, wenn Sie zusätzliche Hardware-Treiber installieren möchten, die nicht vollständig als Open-Source-Code verfügbar sind.

### 2.10 Grundkonfiguration

Dieser Abschnitt gibt einige Hintergrundinformationen zu den üblichen Schritten der Basiskonfiguration. Reihenfolge, Details und Umfang der Grundkonfiguration variieren stark je nach Distribution. Einige Distributionen beschränken die Konfiguration während der Installation auf ein Minimum. Die weitergehende Hardware-Konfiguration erfolgt dann erst im laufenden Grundsystem. Generell gilt: Nahezu alle Einstellungen können auch später durchgeführt werden. Verschieben Sie die Konfiguration von momentan nicht benötigten Komponenten einfach auf später!

- Root-Passwort** Unter Linux ist in der Regel der Benutzer `root` für die Systemadministration zuständig. Dieser Benutzer hat uneingeschränkte Rechte, aber natürlich ist damit auch das Schadenspotenzial uneingeschränkt. Es ist daher unbedingt erforderlich, dass der Zugang zu `root` mit einem Passwort abgesichert wird.

Bei Ubuntu und einigen anderen Distributionen ist der Benutzer-Account `root` vollständig deaktiviert. Eine Passwortabsicherung für `root` ist daher nicht nötig. Administrative Aufgaben werden bei Ubuntu von dafür vorgesehenen Benutzern durchgeführt und erfordern die nochmalige Angabe des Benutzerpassworts.

Bei openSUSE erhalten `root` und der Standardbenutzer dasselbe Passwort. Wenn Sie das nicht wünschen, müssen Sie die leicht zu übersehende Option im Installationsprogramm deaktivieren.

Es ist unter Linux unüblich, als `root` zu arbeiten – außer natürlich bei der Durchführung administrativer Aufgaben. Wenn Sie eine E-Mail schreiben, ein Programm kompilieren oder im Internet surfen, melden Sie sich als gewöhnlicher Benutzer an. Während der Installation haben Sie die Möglichkeit, einen oder mehrere derartige Benutzer samt Passwort einzurichten. Im laufenden Betrieb können Sie später weitere Benutzer hinzufügen, das Passwort vorhandener Benutzer verändern etc.

Benutzer-  
verwaltung

Den Benutzernamen (Account-Namen) setzen Sie am besten aus lauter Kleinbuchstaben zusammen. Leerzeichen sind nicht zulässig, und deutsche Sonderzeichen sollten Sie vermeiden.

Das Passwort sollte mindestens acht Zeichen lang sein. Idealerweise enthält es sowohl Groß- als auch Kleinbuchstaben sowie mindestens eine Ziffer. Auch diverse Sonderzeichen sind erlaubt, z. B. `+*/_.,;:()[]`. Deutsche Sonderzeichen (äöüß) und andere Buchstaben, die nicht im ASCII-Zeichensatz definiert sind, sollten Sie hingegen auch hier vermeiden. Solche Zeichen könnten dazu führen, dass Sie sich bei einer falschen oder fehlenden Konfiguration der Tastatur nicht einloggen können.

Die Netzwerkkonfiguration erfolgt vollautomatisch, wenn das Installationsprogramm im lokalen Netz einen sogenannten DHCP-Server erkennt. Das ist ein (ADSL-) Router, der allen Rechnern im Netzwerk automatisch die Netzwerkparameter sendet. In diesem Fall reduziert sich die Netzwerkkonfiguration auf die Angabe des gewünschten Rechnernamens und gegebenenfalls des WLAN-Passworts.

Netzwerk-  
konfiguration

Bei einer manuellen Netzwerkkonfiguration werden Sie nach den folgenden Parametern gefragt. Hintergrundinformationen und Erklärungen zu den hier verwendeten Fachausdrücken finden Sie in [Kapitel 26](#), »Netzwerkkonfiguration«.

- ▶ **Host- und Domainname:** Mit dem »Hostnamen« ist ganz einfach der Rechnername gemeint. Verwenden Sie als Hostnamen nicht `localhost`, dieser Name hat eine besondere Bedeutung!

Den Domainnamen können Sie zu Hause frei wählen. Bei Firmeninstallationen ist er vorgegeben und entspricht oft dem unter Windows geltenden Workgroup-Namen.

- ▶ **IP-Adresse des Rechners:** Diese Zahl in der Form a.b.c.d (z. B. 192.168.27.35) dient zur internen Identifizierung des Rechners im Netz. Oft sind die drei ersten Zahlengruppen bereits durch das lokale Netz vorgegeben (z. B. 192.168.27); die vierte Zahl muss innerhalb des Netzes eindeutig sein.
- ▶ **Netzwerkmaske, Netzwerkadresse und Broadcast-Adresse:** Die Ausdehnung eines lokalen Netzes wird durch zwei oder drei Masken ausgedrückt, die hier ganz kurz anhand eines Beispiels erläutert werden: Wenn das lokale Netz alle Nummern 192.168.27.n umfasst, lautet die dazugehörige Netzwerkmaske 255.255.255.0 (der Regelfall für kleine, lokale Netze). Als Netzwerkadresse ergibt sich 192.168.27.0, als Broadcast-Adresse 192.168.27.255.
- ▶ **Gateway-Adresse:** Wenn es im lokalen Netz einen Rechner oder ADSL-Router gibt, der für alle anderen Rechner den Internetzugang herstellt, dann geben Sie dessen IP-Adresse als Gateway-Adresse an.
- ▶ **Nameserver-Adresse:** Der sogenannte Nameserver (oft auch als DNS für *Domain Name Server* bezeichnet) ist für die Auflösung von Netzwerknamen in IP-Adressen zuständig. Der Nameserver ist also dafür verantwortlich, dass Sie in einem Webbrowser *www.google.de* eingeben können und der Rechner automatisch die dazugehörige IP-Adresse ermittelt. Beim Nameserver kann es sich wahlweise um einen Rechner im lokalen Netz handeln, der auch für die Auflösung lokaler Namen zuständig ist, oder um einen externen Rechner des Internet Service Providers.

**Firewall** Einige Distributionen schützen den Netzwerk- bzw. Internetzugang standardmäßig durch eine Firewall. Diese Firewall lässt von Ihnen initiierte Verbindungen zu, blockiert aber von außen kommende Anfragen und erhöht so die Sicherheit erheblich. Falls Sie vorhaben, auf Ihrem Rechner selbst Netzwerkdienste anzubieten (z. B. einen SSH- oder Webserver), können Sie für diese Dienste Ausnahmen definieren und externe Zugriffe zulassen.

**Zeitzone** Zur richtigen Einstellung der Uhrzeit fragt das Installationsprogramm, in welcher Zeitzone Sie sich befinden, und eventuell auch, ob die CMOS-Uhr Ihres Rechners die lokale Uhrzeit oder die *Universal Coordinated Time* (UTC) enthält. Da die meisten Rechner ohnedies ständigen Internetzugang haben, ermitteln viele Distributionen die aktuelle Zeit von einem Zeit-Server (NTP-Server).

**Sprache** Standardmäßig wird Linux in der zu Beginn der Installation eingestellten Sprache installiert – für die Leser dieses Buchs also in der Regel in Deutsch. Außerdem werden die englischen Sprachdateien installiert. Das stellt sicher, dass zumindest englische Menü-, Dialog- und Hilfetexte zur Verfügung stehen, falls es keine deutsche Übersetzung gibt.

Wenn einzelne Benutzer Ihres Rechners Linux auch in anderen Sprachen nutzen möchten, müssen Sie zusätzlich entsprechende Sprachdateien installieren (in Form von sogenannten Lokalisierungspaketen). Sie können dann bei jedem Login die gewünschte Sprache wählen.

Die letzte Frage ist nun noch, wie Linux in Zukunft gestartet werden soll. Dafür ist bei fast allen Distributionen das Programm GRUB verantwortlich. Es wird automatisch installiert; nur wenige Distributionen bieten bei diesem Punkt Konfigurationsmöglichkeiten. Debian fordert Sie bei BIOS-Rechnern explizit dazu auf, den Device-Namen des Installationsorts anzugeben. In aller Regel ist das `/dev/sda`, also die erste Festplatte/SSD. Hintergrundinformationen zur manuellen Installation, Konfiguration und Reparatur von GRUB folgen in [Kapitel 23](#).

Installation des  
Bootloaders

Nach der Installation wird der Rechner neu gestartet, wobei automatisch das frisch installierte Linux-System gebootet wird. Anschließend beginnen Sie Ihre erste Erkundungsreise durch die Linux-Welt.

Erster Start

Wenn Sie statt Linux Windows starten möchten, drücken Sie bei EFI-Rechnern eine spezielle Tastenkombination, um das EFI-Boot-Menü zu öffnen. Bei BIOS-Rechnern wählen Sie stattdessen Windows aus dem GRUB-Menü aus.

## 2.11 Probleme beheben

Dieser Abschnitt geht auf einige typische Probleme ein, die während oder nach der Installation auftreten können. Soweit möglich, finden Sie hier auch Lösungsansätze.

### Tastaturprobleme

In den ersten Phasen der Installation kann es vorkommen, dass noch kein deutscher Tastatortreiber installiert ist und daher das amerikanische Tastaturlayout gilt. Das trifft meistens auch während des Starts des Bootloaders zu.

Solange der Rechner glaubt, dass Sie mit einer US-Tastatur arbeiten, während tatsächlich aber ein deutsches Modell im Einsatz ist, sind `[Y]` und `[Z]` vertauscht. Außerdem bereitet die Eingabe von Sonderzeichen Probleme.

[Tabelle 2.5](#) zeigt, wie Sie diverse Sonderzeichen auf einer deutschen Tastatur trotz eines fehlenden Tastatortreibers eingeben können. Dabei zeigt die erste Spalte die auf einer deutschen Tastatur erforderliche Taste oder Tastenkombination, um das Zeichen in der zweiten Spalte zu erzeugen. Verwenden Sie auch den numerischen Tastaturblock – die dort befindlichen Sonderzeichen funktionieren mit Ausnahme des Kommas problemlos!

| Kürzel  | Ergebnis              | Kürzel  | Ergebnis | Kürzel  | Ergebnis |
|---------|-----------------------|---------|----------|---------|----------|
| [Z]     | Y                     | [Ö]     | ;        | [⇧]+[9] | (        |
| [Y]     | Z                     | [⇧]+[Ö] | :        | [⇧]+[0] | )        |
| [_]     | /                     | [⇧]+[-] | ?        | [Ü]     | [        |
| [#]     | \                     | [⇧]+[Ä] | "        | [+]     | ]        |
| [ß]     | - (Bindestrich/Minus) | [Ä]     | '        | [⇧]+[Ü] | {        |
| [⇧]+[ß] | _ (Unterstrich)       | [^]     | `        | [⇧]+[+] | }        |
| [']     | =                     | [⇧]+[^] | ~        | [⇧]+[=] | <        |
| [⇧]+['] | +                     | [⇧]+[2] | @        | [⇧]+[.] | >        |
| [⇧]+[8] | *                     | [⇧]+[3] | #        |         |          |
| [⇧]+[7] | &                     | [⇧]+[6] | ^        |         |          |

Tabelle 2.5 Tastenkürzel zur Eingabe von Sonderzeichen für das US-Tastaturlayout

Nach der Installation können Sie das gewünschte Tastaturlayout zumeist problemlos in den Systemeinstellungen ändern. Sollte das nicht klappen bzw. sollten Sie Linux im Textmodus nutzen, werfen Sie einen Blick in [Abschnitt 18.2](#), »Konfiguration der Textkonsolen«.

### Der Rechner kann nicht mehr gestartet werden

Der schlimmste Fall bei einer Linux-Installation besteht darin, dass der Rechner anschließend nicht mehr gestartet werden kann oder dass zumindest einzelne der installierten Betriebssysteme nicht mehr zugänglich sind. Dabei gibt es verschiedene Varianten, die im Folgenden erörtert werden.

- **Linux-Absturz (Hardware-Probleme):** Nach dem Neustart des Rechners erscheinen zuerst diverse Meldungen von Linux. Anschließend bleibt der Rechner stehen bzw. stürzt ab.

Mögliche Ursache: Die wahrscheinlichste Ursache sind Hardware-Probleme.

Abhilfe: Wenn Linux während oder nach der Installation wichtige Hardware-Komponenten nicht richtig erkennt oder hängen bleibt, helfen eventuell Kernel-Boot-Optionen weiter. Dahinter verbirgt sich ein Mechanismus, dem Kernel beim Start Informationen zur besseren Hardware-Erkennung zu geben. Weitere Informationen zu diesem Mechanismus und einen Überblick über einige wichtige Parameter finden Sie in [Abschnitt 25.8](#), »Kernel-Boot-Optionen«.

- ▶ **Linux-Absturz (unable to mount root fs):** Der Start des Linux-Kernels hat geklappt, Linux konnte aber anschließend die Linux-Systempartition nicht finden.  
Mögliche Ursache: Es liegt ein Problem in der GRUB-Konfiguration vor. Der Fehler kann auch dann auftreten, wenn die Verkabelung von Festplatten geändert wurde.  
Abhilfe: Wenn Sie den Device-Namen der Systempartition kennen, geben Sie diesen beim Linux-Start als Boot-Option in der Form `root=/dev/sda6` an. Wenn der Start so gelingt, können Sie unter Linux GRUB neu konfigurieren (siehe [Kapitel 23](#)).
- Unter Umständen müssen Sie auch die Datei `/etc/fstab` entsprechend anpassen. Wie das geht, beschreibt [Abschnitt 22.8](#), »Verwaltung des Dateisystems (mount und `/etc/fstab`)«.
- ▶ **Linux startet nicht (BIOS):** Nach dem Neustart des Rechners wird ohne Rückfrage einfach Windows gestartet. Von Linux ist keine Spur zu sehen.  
Mögliche Ursache: Die Installation von GRUB (oder eines anderen Bootloaders) auf die Festplatte hat aus irgendeinem Grund nicht funktioniert.  
Abhilfe: Starten Sie ein Rescue-System oder eine Live-CD, und installieren Sie GRUB neu (siehe [Abschnitt 23.4](#), »Manuelle GRUB-Installation und Erste Hilfe«).
- ▶ **Linux startet nicht (EFI):** Bei EFI-Rechnern kann es sein, dass die GRUB-Installation an sich zwar funktioniert hat und nur der Eintrag des Bootloaders in die Liste der EFI-Betriebssysteme gescheitert ist.  
Abhilfe: Werfen Sie in einem Live-System einen Blick in die EFI-Partition (Verzeichnis `/boot/efi/EFI`). Wenn Sie dort ein Unterverzeichnis mit dem Namen Ihrer Distribution entdecken, wurde GRUB dorthin installiert. Sie können nun versuchen, die GRUB- oder SHIM-Datei mit dem Kommando `efibootmgr` zur Liste der EFI-Boot-Einträge hinzuzufügen. Dieses Kommando wird ebenfalls in [Abschnitt 23.4](#), »Manuelle GRUB-Installation und Erste Hilfe«, beschrieben.
- ▶ **Linux startet nicht (EFI Secure Boot):** Beim Versuch, Linux zu starten, wird eine Fehlermeldung wie *secure boot violation* oder *invalid signature* angezeigt. Das deutet auf ein Problem mit EFI Secure Boot hin.  
Abhilfe: Die einfachste Lösung besteht darin, Secure Boot in den EFI-Einstellungen zu deaktivieren.
- ▶ **Windows startet nicht (BIOS):** Nach dem Neustart wird automatisch Linux gestartet. Windows scheint verschwunden zu sein.  
Mögliche Ursache: Wahrscheinlich hat die GRUB-Installation funktioniert. Sie können nun unmittelbar nach dem Rechnerstart auswählen, welches Betriebssystem gestartet werden soll. Tun Sie nichts, wird nach einer Weile automatisch Linux gestartet.

Abhilfe: Drücken Sie während des Boot-Vorgangs `[Esc]`, damit das GRUB-Menü erscheint. Wählen Sie dort mit den Cursorstasten `windows` aus, und drücken Sie `[↵]`.

- ▶ **Windows startet nicht (EFI):** Zu einem ähnlichen Problem kann es auch bei EFI-Installationen kommen. Durch die Linux-Installation gilt nun Linux als Default-Betriebssystem.

Abhilfe: Um Windows zu starten, drücken Sie unmittelbar nach dem Rechnerstart eine Tastenkombination, um das EFI-Boot-Menü anzuzeigen. Für die Tastenkombination gibt es leider keinen Standard, sie ist bei jedem Rechner bzw. Mainboard anders. Den EFI-Default-Boot-Eintrag können Sie entweder im EFI oder mit dem vorhin erwähnten Linux-Kommando `efibootmgr` einstellen.

- ▶ **Weder Linux noch Windows startet (BIOS):** Nach dem Rechnerstart wird GRUB ausgeführt, stürzt aber sofort ab bzw. zeigt eine endlose Liste von Fehlermeldungen an.

Mögliche Ursache: Die GRUB-Installation ist fehlgeschlagen.

Abhilfe: Starten Sie Ihren Rechner mit einem Rescue-System oder einer Live-CD neu, und installieren Sie GRUB neu. Eine Anleitung finden Sie in [Abschnitt 23.4](#), »Manuelle GRUB-Installation und Erste Hilfe«.

## 2.12 Systemveränderungen, Erweiterungen, Updates

Wenn Ihr Linux-System einmal stabil läuft, wollen Sie es zumeist nach Ihren eigenen Vorstellungen konfigurieren, erweitern, aktualisieren etc. Detaillierte Informationen zu diesen Themen sind über das gesamte Buch verteilt. Dieser Abschnitt dient daher primär als Referenz, um Ihnen die Sucharbeit so weit wie möglich zu ersparen.

### Software-Installation, Paketverwaltung

Je nach Distribution existieren verschiedene Kommandos und Programme, mit denen Sie im laufenden Betrieb weitere Software-Pakete installieren, aktualisieren oder entfernen. Bei den meisten auf Gnome basierenden Distributionen ist dafür das Programm *Software* zuständig. Die zugrunde liegenden Verfahren sowie Kommandos, mit denen Sie Software im Terminal installieren können, sind in [Kapitel 19](#), »Software- und Paketverwaltung«, beschrieben.

### Updates

**Normale Updates** Alle gängigen Distributionen weisen regelmäßig automatisch auf Updates hin. Mit wenigen Mausklicks können Sie alle betroffenen Programme aktualisieren. Anders



als unter Windows ist das Update-System für *alle* Komponenten zuständig. Es gibt also nicht verschiedene Update-Mechanismen für unterschiedliche Programme.

Durch das Update-System werden gravierende Fehler behoben und Sicherheits-Updates durchgeführt. Das erste Update nach der Neuinstallation einer Distribution dauert oft sehr lange, bisweilen länger als die eigentliche Installation! Das liegt daran, dass damit sämtliche Updates installiert werden, die seit der Fertigstellung der Distribution freigegeben wurden. Alle weiteren Updates, die regelmäßig durchgeführt werden, betreffen dann nur noch wenige Pakete und erfolgen entsprechend schneller. Derartige Updates sind ziemlich häufig: Sie erscheinen meist wöchentlich, mitunter sogar mehrmals pro Woche.

Durch das Update-System werden Fehler und Sicherheitsmängel behoben, aber in der Regel keine grundlegend neuen Programmversionen installiert. Die einzige Ausnahme sind Webbrowser. Auf ein Update von LibreOffice 5.n auf Version 6.n werden Sie also vergeblich warten. Stattdessen sind Sie zumeist gezwungen, die gesamte Distribution auf die nächste Version zu aktualisieren – daher die Bezeichnung »Distributions-Update«.

Distributions-  
Updates

Es gibt zwei unterschiedliche Verfahren für Distributions-Updates: Entweder beginnen Sie die Installation von einem Datenträger und geben dann an, dass Sie eine vorhandene Distribution aktualisieren möchten, oder Sie führen das Update im laufenden Betrieb durch und müssen anschließend nur einen Neustart durchführen. Das zweite Verfahren ist wesentlich eleganter, weil es ohne Installationsmedien durchgeführt werden kann. Die neuen Pakete werden einfach aus dem Internet heruntergeladen. Außerdem wird die Zeit minimiert, während der die Distribution nicht läuft bzw. während der ein Server offline ist.

Tabelle 2.6 fasst zusammen, welche Distributionen welche Verfahren unterstützen. Beachten Sie, dass Sie bei CentOS/RHEL über einen Zeitraum von vielen Jahren *automatisch* alle Updates innerhalb eines Major-Release erhalten. Ihre CentOS-Version steigt damit z. B. von 6.0 bis auf 6.9. Ein Update auf die nächste Major-Version, also z. B. von 6.9 auf 7.0, ist hingegen nicht vorgesehen.

|             | Update während der Installation | Update im laufenden Betrieb |
|-------------|---------------------------------|-----------------------------|
| CentOS/RHEL |                                 | •                           |
| Debian      |                                 | •                           |
| Fedora      | •                               | •                           |
| openSUSE    | •                               | •                           |
| Ubuntu      | •                               | •                           |

**Tabelle 2.6** Verfahren für Distributions-Updates

**Probleme** Was in der Theorie toll klingt, funktioniert in der Praxis leider oft schlecht. Nach dem Distributions-Update funktionieren bisweilen Programme nicht mehr wie vorher, und die Suche nach den Fehlern kann zeitraubend sein. Ich selbst habe nach zahllosen Problemen den Glauben an Distributions-Updates verloren.

Persönlich tendiere ich dazu, nicht jedes Distributions-Update mitzumachen, sofern mich nicht die Arbeit an diesem Buch dazu zwingt. Stattdessen führe ich bei Bedarf – oft erst nach drei, vier Jahren – eine komplette Neuinstallation durch, wobei ich die Datenpartition `/home` unverändert weiternutze.

**Rolling Releases** Rolling Releases sollen die Notwendigkeit von Distributions-Updates ganz eliminieren. Bei Distributionen, die dem Rolling-Release-Modell folgen, werden alle Pakete ständig auf die gerade aktuellste vorliegende Version aktualisiert – so wie dies bei vielen Webbrowsern gehandhabt wird.

Auch dieses Konzept klingt besser, als es tatsächlich funktioniert: Viele Neuerungen führen zwangsläufig zu Inkompatibilitäten oder Migrationsproblemen. Automatische Updates erfolgen unter Umständen zu einem ungünstigen Zeitpunkt, und der Benutzer wird plötzlich mit Programmen konfrontiert, die nicht mehr so funktionieren wie bisher – oder gar nicht mehr.

Aus diesen Gründen haben sich Rolling-Release-Distributionen bisher nicht durchsetzen können bzw. richten sich ausschließlich an technisch versierte Linux-Profis:

- ▶ Debian kommt dem Rolling-Release-Modell ziemlich nahe, wenn Sie die Paketquellen *testing* oder *unstable* aktivieren.
- ▶ SUSE bietet mit *Tumbleweed* eine Rolling-Release-Variante von openSUSE an, die seit 2014 recht problemfrei funktioniert.
- ▶ Die KDE-Distribution Neon folgt teilweise dem Rolling-Release-Modell: Zwar bleibt der Unterbau, eine Ubuntu-LTS-Version, bei Updates unverändert. Die KDE-Pakete werden aber laufend auf den aktuellsten Stand gebracht.

### Konfiguration

Zwar gab es in der Vergangenheit immer wieder Bemühungen, die Konfiguration von Linux zu vereinheitlichen, tatsächlich unterscheiden sich die einzelnen Distributionen leider nach wie vor erheblich. Aus diesem Grund sollten Sie zur weiteren Konfiguration nach Möglichkeit die jeweils mitgelieferten Werkzeuge einsetzen.

Die Lösung mancher Konfigurationsprobleme erfordert freilich mehr als ein paar Mausclicks. Deswegen gehe ich in diesem Buch losgelöst von speziellen Distributionen ausführlich auf Grundlagen und Hintergründe verschiedener Soft- und Hardware-Komponenten ein (siehe [Tabelle 2.7](#)).

| Thema                     | Kapitel                    | Thema                 | Kapitel                       |
|---------------------------|----------------------------|-----------------------|-------------------------------|
| Gnome                     | <a href="#">Kapitel 5</a>  | Systemstart           | <a href="#">Kapitel 23</a>    |
| KDE                       | <a href="#">Kapitel 6</a>  | Kernel, Module        | <a href="#">Kapitel 25</a>    |
| Basiskonfiguration        | <a href="#">Kapitel 18</a> | Netzwerkkonfiguration | <a href="#">Kapitel 26</a>    |
| Paketverwaltung           | <a href="#">Kapitel 19</a> | Server-Konfiguration  | ab <a href="#">Kapitel 27</a> |
| Grafiksystem (X, Wayland) | <a href="#">Kapitel 21</a> | Drucker (CUPS)        | <a href="#">Kapitel 30</a>    |

Tabelle 2.7 Linux-Konfiguration

## 2.13 Linux wieder entfernen

Persönlich kann ich mir das zwar kaum vorstellen, aber vielleicht sind Sie von Linux nicht so begeistert wie ich und möchten es wieder entfernen. Am einfachsten geht das, indem Sie Windows auf dem Rechner neu installieren und während der Installation die Festplatte neu partitionieren und die gesamte Festplatte für Windows nutzen.

Wenn Sie eine Windows-Neuinstallation vermeiden möchten und einfach nur das vorhandene Windows weaternutzen möchten, müssen Sie sich um zwei Dinge kümmern:

- ▶ Löschen Sie alle Linux-Partitionen, damit Sie den Platz später wieder unter Windows nutzen können.
- ▶ Stellen Sie sicher, dass Windows beim Einschalten des Rechners automatisch gestartet wird. Die genaue Vorgehensweise hängt davon ab, ob Ihr Rechner durch ein herkömmliches BIOS oder durch ein EFI gesteuert wird.

Es ist empfehlenswert, Partitionen eines bestimmten Betriebssystems möglichst nur mit den Werkzeugen dieses Betriebssystems zu ändern. Insofern sollten Sie zum Löschen der Linux-Partitionen idealerweise Linux-Werkzeuge einsetzen. Da es unmöglich ist, die Systempartition eines laufenden Linux-Systems direkt zu löschen, setzen Sie zum Löschen der Linux-Partitionen am besten ein Live-System ein.

Linux-Partitionen löschen

Das eigentliche Löschen der Linux-Distributionen gelingt dann mit dem Kommando `parted` bzw. mit dessen grafischer Variante `gparted`. Die Bedienung von `parted` ist in [Abschnitt 22.5](#) beschrieben.

Bei EFI-Rechnern gibt es für jedes installierte Betriebssystem einen Eintrag in der EFI-internen Liste der Betriebssysteme. Als Defaulteintrag gilt üblicherweise das zuletzt installierte Betriebssystem, also Linux. Um zu erreichen, dass wieder Windows zum EFI-Defaultsystem wird, müssen Sie beim Rechnerstart die EFI-Konfigurationsdialoge starten. Dazu drücken Sie eine Tastenkombination, die vom Rechner bzw. Mainboard

Windows zuerst booten (EFI-Rechner)

abhängig ist. Eine Internetsuche nach `<computermodell> start efi configuration` führt rasch zum Ziel.

In den EFI-Konfigurationsdialogen suchen Sie nach der Liste aller Betriebssysteme. Dort verschieben Sie den Windows-Eintrag an die erste Stelle. Sofern das EFI eine entsprechende Möglichkeit gibt, können Sie die Linux-Einträge ganz löschen.

### Original-MBR wiederherstellen (BIOS-Rechner)

Bei einem BIOS-Rechner enthält der Master-Boot-Record (MBR) normalerweise Daten des Bootloaders GRUB. Um GRUB zu deaktivieren, stellen Sie den ursprünglichen Zustand des MBRs wieder her. Bei neueren Windows-Versionen starten Sie den Rechner mit der Installations-DVD. Nach der Sprach- und Tastatureinstellung klicken Sie auf den Eintrag `COMPUTERREPARATUROPTIONEN` und wählen dann Ihre Windows-Version aus. Im Dialog `SYSTEMWIEDERHERSTELLUNGSOPTIONEN` wählen Sie den Punkt `EINGABEAUFFORDERUNG` und gelangen so in ein Konsolenfenster. Dort führen Sie das folgende Kommando aus:

```
> BOOTREC /fixmbr  
Der Vorgang wurde abgeschlossen.
```

Anschließend starten Sie den Rechner neu. Weitere Informationen zu `BOOTREC` finden Sie hier:

<https://support.microsoft.com/kb/927392/en-us>

# Kapitel 3

## Installationsanleitungen

Nachdem das vorige Kapitel ausführlich die Grundlagen einer Linux-Installation behandelt hat, folgen in diesem Kapitel konkrete Beispiele. Sie lernen hier einige ausgewählte Linux-Distributionen näher kennen und erfahren, welche Besonderheiten bei deren Installation zu beachten sind. Gleichzeitig gebe ich in diesem Kapitel Tipps für die ersten Schritte nach der Installation:

- ▶ CentOS / Red Hat Enterprise Linux
- ▶ Debian
- ▶ Fedora
- ▶ Linux Mint
- ▶ openSUSE
- ▶ Ubuntu und Ubuntu Server

Für Linux-Einsteiger am besten geeignet sind momentan Ubuntu und openSUSE. Beide Distributionen sind für den Desktop-Einsatz optimiert und weit verbreitet, d. h., es gibt eine Menge Foren und Wikis, die bei Problemen weiterhelfen. Ubuntu-LTS-Versionen können Sie mit gutem Gewissen auch auf den Rechnern Ihrer Freunde und Verwandten installieren – sie genießen einen drei- bis fünfjährigen Update-Service.

Fedora ist ebenfalls eine Desktop-Distribution, richtet sich aber an fortgeschrittene Linux-Anwender. Red Hat betrachtet Fedora als Experimentierplattform. Insofern ist Fedora ideal geeignet, um die neuesten Entwicklungen aus der Linux-Welt kennenzulernen. Die Stabilität bleibt dabei aber mitunter auf der Strecke.

Für den Server-Einsatz ist ein langer Update-Zeitraum noch wichtiger als für Desktop-Anwendungen. Fedora sowie gewöhnliche Ubuntu-Versionen (ohne LTS) scheiden deswegen von vornherein aus. Empfehlenswert sind hingegen CentOS, Debian sowie Ubuntu LTS.

Außer den hier präsentierten Distributionen gibt es natürlich noch Hunderte andere. Einige davon werden Sie in speziell dazu passenden Kapiteln kurz kennenlernen – z. B. Raspbian in [Kapitel 8](#), »Raspberry Pi«. Aktuelle Nachrichten zu den gerade populärsten Distributionen finden Sie hier:

<https://distrowatch.com>

**Welche  
Distribution für  
welche  
Anwendung?**

## 3.1 CentOS

**RHEL** Bevor ich die Besonderheiten von CentOS erklären kann, muss ich kurz Red Hat Enterprise Linux (RHEL) beschreiben. RHEL ist die kommerziell erfolgreichste Linux-Distribution. Sie kommt beispielsweise bei Banken, Versicherungen oder auf Großrechnern zum Einsatz – also immer dann, wenn Stabilität und professioneller Support höchste Priorität genießen. Neben dem Kernprodukt RHEL bietet Red Hat verschiedene RHEL-Erweiterungen und Spezialprodukte an, z. B. den Red Hat Storage Server, die JBoss Middleware (Java EE) und das Red Hat Virtualization System.

Neue Versionen von Red Hat Enterprise Linux (RHEL) basieren grundsätzlich auf der zuletzt erschienenen Fedora-Version. Bei RHEL 7 war das Fedora 19. Es gibt aber natürlich grundlegende Unterschiede zwischen Fedora und RHEL: In die Enterprise-Version werden keine Funktionen eingebaut, die noch nicht vollkommen stabil und ausgereift sind. Der Support-Zeitraum für RHEL ist wesentlich länger und beträgt zwischen 7 und 13 Jahre. Und schließlich hilft das *Red Hat Subscription Management* (RHSM) bei der zentralen Wartung mehrerer RHEL-Installationen.

Im Unterschied zu den anderen in diesem Buch vorgestellten Distributionen ist RHEL nicht frei erhältlich. Installationsmedien und Updates stehen nur zahlenden Kunden zur Verfügung. Aber selbstverständlich muss sich auch Red Hat an die Regeln der GPL halten und den Quellcode seiner Produkte zur Verfügung stellen.

**CentOS** CentOS (<https://www.centos.org>) ist der seit vielen Jahren populärste Klon von RHEL. CentOS ist binärkompatibel zu RHEL, im Gegensatz zu diesem aber inklusive aller Updates kostenlos verfügbar. Abstriche müssen Sie naturgemäß beim kommerziellen Support machen – den gibt es nicht. Dennoch ist CentOS für Administratoren mit Red-Hat- oder Fedora-Erfahrung eine tolle Möglichkeit, bei Projekten mit kleinem Budget Red-Hat-kompatible Server einzusetzen.

CentOS wurde im Januar 2014 etwas überraschend von Red Hat übernommen. Momentan sieht es so aus, als würden beide Seiten von dieser Kooperation profitieren: CentOS-Anwendern wird das Gefühl vermittelt, dass sie sich auch ohne kommerziellen Support im sicheren Hafen von Red Hat befinden und nicht länger allein vom winzigen CentOS-Team abhängig sind. Dank der besseren Zusammenarbeit erscheinen CentOS-Releases und -Updates nun auch viel schneller als in der Vergangenheit.

Für Red Hat besteht der größte Vorteil darin, dass es eine bessere Kontrolle über den RHEL-Klon-Markt hat – etwa nach dem Motto: Wer sich für das »offizielle« CentOS entscheidet, bringt Red Hat zwar kein Geld, aber verwendet zumindest nicht Oracle Linux.

CentOS basiert auf dem originalen Quellcode von Red Hat – RHEL ist ja ein Open-Source-Produkt! Trotz der engen Zusammenarbeit kann das CentOS-Projekt den RHEL-Quellcode aber nicht unverändert übernehmen: »Red Hat« ist ein geschütztes Markenzeichen. Alle Pakete, die Red-Hat-spezifische Zeichenketten, Logos oder Bilder enthalten, müssen modifiziert werden. Einige Red-Hat-spezifische Pakete, z. B. jene zum Zugriff auf das Red Hat Subscription Management, werden ganz entfernt. Die modifizierten Pakete müssen kompiliert, getestet und schließlich in Form von neuen Installationsmedien gebündelt werden.

All das kostet Zeit und Mühe und erklärt, warum es nach der Freigabe einer neuen RHEL-Version oft Wochen dauert, bis auch die entsprechende CentOS-Variante zur Verfügung steht.

Bei aller Begeisterung für CentOS muss Ihnen klar sein, dass diese Distribution natürlich kein vollwertiger Ersatz für RHEL ist:

- ▶ **Support:** Sie erhalten keinen kommerziellen Support. Bei Problemen sind Sie auf Foren, Mailinglisten und Selbsthilfe angewiesen.
- ▶ **CPU-Architekturen:** Während es einige RHEL-Versionen auch für ausgefallene CPU-Architekturen und Hardware-Plattformen gibt (IA-64, Power7 und Power8, IBM zSeries), konzentriert sich CentOS auf die PC-Architektur. Dafür bietet CentOS für kleine Server auch eine 32-Bit-Version an, während RHEL seit Version 7 nur noch 64-Bit-Architekturen unterstützt.
- ▶ **Updates:** Es kann ein paar Tage dauern, bis von Red Hat freigegebene Sicherheits-Updates auch für CentOS zur Verfügung stehen.

CentOS ist keineswegs der einzige RHEL-Klon. Ebenfalls weit verbreitet sind Scientific Linux und Oracle Linux. Während CentOS für den »gewöhnlichen« Server-Einsatz optimiert ist, hat Scientific Linux eine etwas andere Zielrichtung: Diese Distribution wird von Mitarbeitern der Forschungseinrichtungen Fermilab und CERN zusammengestellt und in unzähligen Universitäten und Forschungseinrichtungen verwendet. Nichtsdestotrotz ist auch Scientific Linux ein weitgehend unverändertes RHEL und ebenso gut für den Server-Alltag geeignet wie CentOS.

Scientific Linux

Auch Oracle ist 2006 in den Markt der RHEL-Klone eingestiegen – aber mit ganz anderen Konzepten: Zum einen ist Oracle Linux auch ein kommerzielles Angebot, zum anderen versucht Oracle Linux sich mit diversen Zusatzfunktionen von RHEL abzuheben: Dazu zählen ein neuerer Kernel mit diversen Zusatzfunktionen (im Marketing-Jargon: »Unbreakable Enterprise Kernel«), Ksplice-Kernel-Updates im laufenden Betrieb sowie eine bessere Unterstützung des von Oracle mitentwickelten Dateisystems btrfs.

Oracle Linux

Anfänglich war Oracle Linux wie RHEL ausschließlich für zahlende Kunden zugänglich, wenn auch zu deutlich günstigeren Preisen als bei RHEL. Seit 2012 kann Oracle Linux inklusive aller Updates kostenlos bezogen werden und steht damit auf einer Ebene mit CentOS und Scientific Linux. Einzig der kommerzielle Support ist weiterhin kostenpflichtig.

**Zielrichtung** Egal ob Original oder Klon – die Zielrichtung von RHEL ist ganz klar der Server-Einsatz bei maximaler Stabilität. Das erklärt auch, warum RHEL & Co. oft vergleichsweise alte Software-Versionen ausliefern (siehe [Abbildung 3.1](#)).

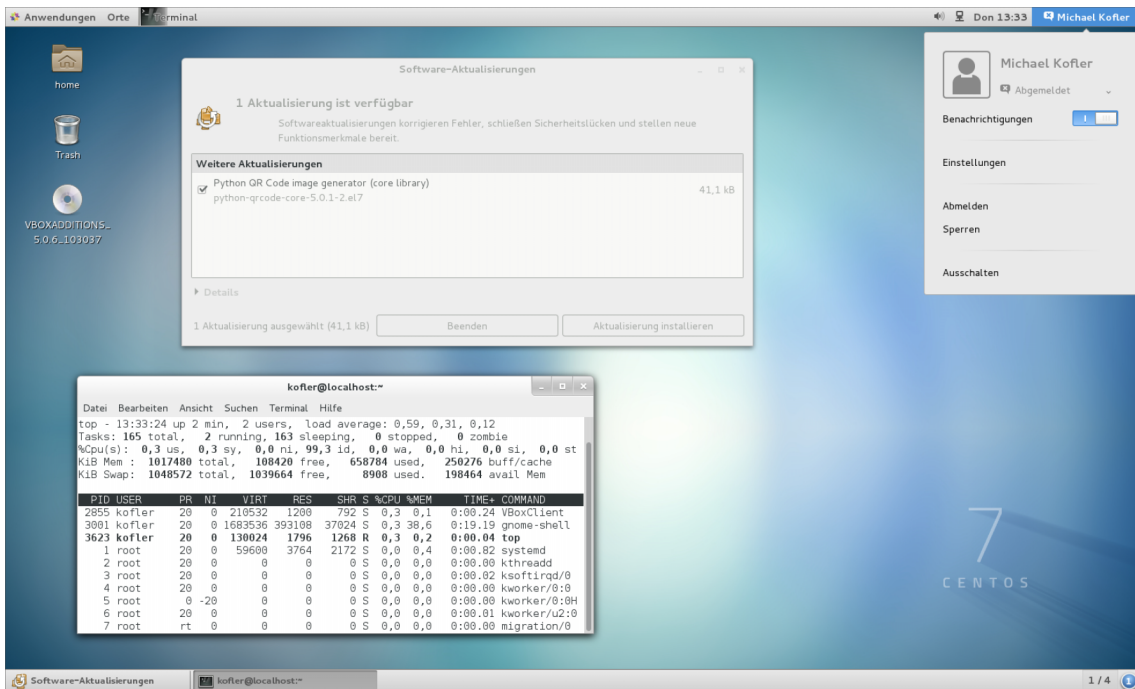


Abbildung 3.1 CentOS 7 mit Gnome im Klassikmodus

**Versionsnummern** RHEL, CentOS und alle anderen Klone verwenden ein Versionschema, das von den Schemata anderer Distributionen abweicht. Circa alle drei bis vier Jahre gibt es eine »große« neue RHEL-Version (Major Release). Momentan ist RHEL 7 aktuell. Diese Version wurde im Juni 2014 vorgestellt. Nach wie vor gewartet werden RHEL 5 (vorgestellt im März 2007) und RHEL 6 (vorgestellt im November 2010).

Innerhalb jeder großen RHEL-Version gibt es ein- bis zweimal im Jahr ein neues Minor Release (Version 7.1, 7.2 etc.). Im Zuge derartiger Updates gibt es auch neue Installationsmedien, die kompatibel zu aktueller Hardware sind. Fundamental neue Funktionen werden dabei nur in Ausnahmefällen eingebaut, und wenn doch, dann



meist unter der Bezeichnung »Technical Preview« (also im Klartext: ohne offiziellen Support).

Die Besonderheit der Minor Releases besteht darin, dass diese *keine* Neuinstallation erfordern. Wenn Sie also beispielsweise CentOS 7.1 installieren und dann regelmäßig alle Updates durchführen, erhalten Sie nach und nach CentOS 7.2, CentOS 7.3 etc. Erst für den Umstieg auf das nächste Major Release ist eine Neuinstallation erforderlich.

Allerdings dürfen Sie von diesem Update-Service keine Wunder erwarten: Bei den meisten Komponenten werden lediglich Bugfixes, aber keine Versions-Updates durchgeführt. Zu den wenigen Ausnahmen zählt zum einen Firefox, das alle neun Monate auf die jeweils aktuelle ESR-Version aktualisiert. Zum anderen machen seit der Hauptversion 7 hin und wieder auch andere Komponenten wie OpenJDK, Gnome und LibreOffice Versions sprünge.

Red Hat garantiert einen Wartungszeitraum von zehn Jahren, gerechnet vom Zeitpunkt der Freigabe des Major Release. Bei manchen Versionen gibt es danach für drei weitere Jahre Updates für kritische Sicherheitsprobleme, aber nur noch beschränkten Support. Auf der folgenden Webseite sind die verschiedenen Phasen im Lebenszyklus einer RHEL-Version detailliert aufgeschlüsselt:

Update-Zeitraum

<https://access.redhat.com/support/policy/updates/errata>

Die entsprechenden CentOS-Zeiträume sind etwas kürzer. Über circa sieben Jahre gibt es uneingeschränkte Updates, danach für etwa 3 weitere Jahre sogenannte *Maintenance Updates*. Die Details können Sie hier nachlesen:

<http://wiki.centos.org/About/Product>

## CentOS installieren

Zur Installation von CentOS müssen Sie die passenden ISO-Images herunterladen:

Installations-  
medien

<https://www.centos.org/download>

Es stehen etliche ISO-Images zur Auswahl:

- ▶ **Standard-DVD:** Dieses rund 4 GiB große Image reicht für normale Installationen aus. Alle weiteren Pakete können Sie später im laufenden Betrieb aus den Paketquellen im Internet installieren.
- ▶ **Everything-DVD:** Dieses Image mit rund 7 GiB enthält alle CentOS-Pakete.
- ▶ **Netinstall:** Wenn Sie über eine schnelle Internetverbindung verfügen, laden Sie das nur 400 MiB große Netinstall-Image herunter und brennen es auf eine CD bzw. übertragen es auf einen USB-Stick. Während der Installation werden alle weiteren Pakete aus dem Internet geladen.

- ▶ **Minimal:** Eine weitere Option ist das Minimal-Image. Damit wird wirklich eine Minimalinstallation durchgeführt: CentOS kann danach nur im Textmodus genutzt werden, Sie müssen sich selbst um die Netzwerkkonfiguration kümmern, es sind keinerlei Server-Dienste installiert etc. Das klingt unbequem, eignet sich aber z. B. gut als Startpunkt für eine möglichst ressourcensparende virtuelle Maschine.
- ▶ **Live-Systeme:** Zu guter Letzt stehen noch drei Live-Systeme zur Auswahl, je eines für Gnome, KDE und für den Textmodus. Die Live-Systeme können ebenfalls zur Installation verwendet werden, sind darüber hinaus aber für Wartungsarbeiten prädestiniert.

Abweichend von den ISO-Dateien für RHEL verwendet CentOS eine eigene Nomenklatur für die Dateinamen: Anstelle der Minor-Release-Nummer ist das Datum der Freigabe in der Form *YYMM* enthalten, also z. B. 1611 für die seit Ende 2016 verfügbare CentOS-Version 7.3.

#### Installationsprogramm

CentOS verwendet dasselbe Installationsprogramm wie Fedora. Seine Bedienung ist in [Abschnitt 3.3](#) näher beschrieben. Im Vergleich zu Fedora müssen Sie die folgenden Besonderheiten berücksichtigen:

- ▶ Das Partitionierwerkzeug schlägt das Dateisystem *xfs* zur Formatierung aller Partitionen bzw. Logical Volumes vor. Wenn Sie ein anderes Dateisystem vorziehen, müssen Sie die entsprechenden Optionen explizit setzen. Die wenig durchdachte Benutzerführung des Partitionierprogramms macht das leider nicht gerade einfach.
- ▶ Standardmäßig wird das Ethernet-Netzwerk nicht aktiviert. Sie müssen entweder die entsprechende, leicht zu übersehende Option aktivieren (siehe [Abbildung 3.2](#)), oder später im NetworkManager die Schnittstelle dauerhaft aktivieren.
- ▶ Standardmäßig wird CentOS für den Betrieb im Textmodus installiert (MINIMALE INSTALLATION). Für den Server-Betrieb ist das oft zweckmäßig. Wenn Sie CentOS hingegen über eine grafische Benutzeroberfläche bedienen möchten, vergessen Sie nicht, die Option GNOME-DESKTOP oder KDE PLASMA WORKSPACES auszuwählen (siehe [Abbildung 3.3](#))!

#### Gnome nachträglich installieren

Um nach einer minimalen Installation für den Textmodus nachträglich das X Window System sowie einen minimalen Gnome-Desktop zu installieren, führen Sie die folgenden Kommandos aus:

```
root# yum groupinstall "X Window System"
root# yum install gnome-classic-session gnome-terminal control-center \
      liberation-mono-fonts
root# systemctl set-default graphical.target
root# systemctl isolate graphical.target
```

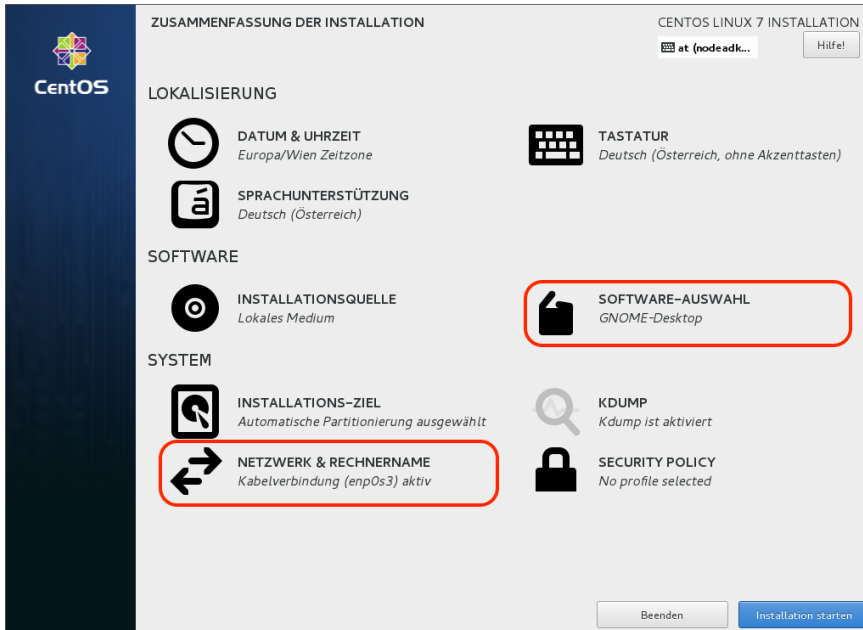


Abbildung 3.2 Grundeinstellungen im Setup-Programm

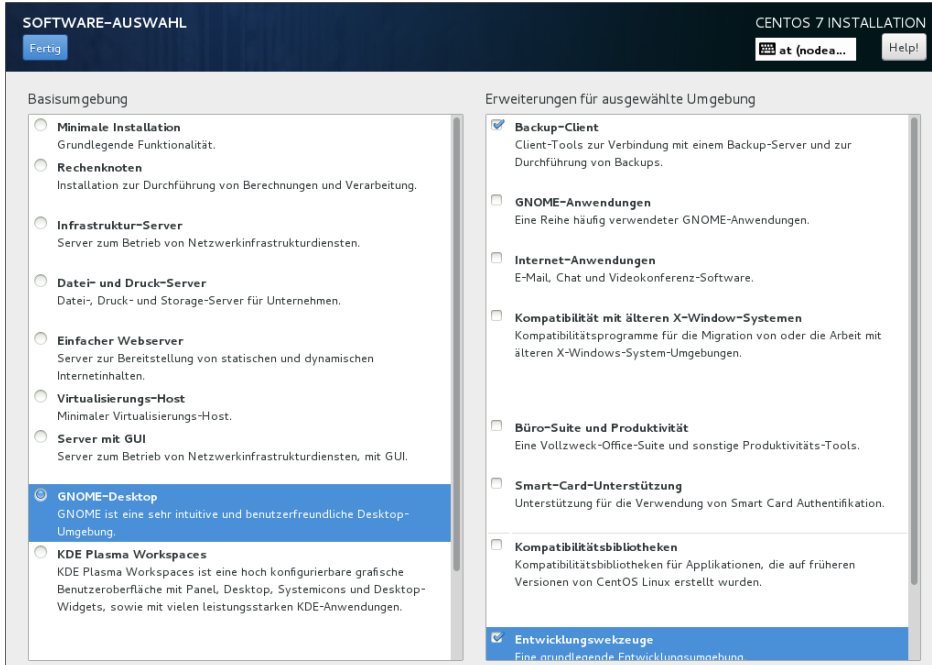


Abbildung 3.3 Auswahl der zu installierenden Komponenten

Wenn Sie den kompletten Gnome-Desktop mit allen erdenklichen Zusatzprogrammen wollen, ersetzen Sie die beiden obigen yum-Kommandos durch das folgende Kommando:

```
root# yum groupinstall "Gnome desktop"
```

Das alternative Grafiksystem Wayland steht unter CentOS 7 übrigens nicht zur Auswahl.

**Kickstart-Installation** Anstatt die Installation manuell durchzuführen, können Sie diesen Vorgang auch automatisieren. Das ist dann zweckmäßig, wenn Sie Dutzende gleichartige Installationen durchführen müssen. Eine ausführliche Beschreibung dieses Verfahrens finden Sie hier:

<https://www.tecmint.com/multiple-centos-installations-using-kickstart>  
[https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/7/html/Installation\\_Guide/sect-kickstart-syntax.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Installation_Guide/sect-kickstart-syntax.html)

#### Erste Schritte

**Netzwerk automatisch aktivieren** Bei Desktop-Installationen kümmert sich wie üblich der NetworkManager um den Netzwerkzugang. Unverständlicherweise wird bei einer Netzwerkanbindung per Ethernet-Kabel zu einem DHCP-Server die Ethernet-Schnittstelle nur dann aktiviert, wenn Sie die entsprechende Option bei der Installation nicht übersehen, sondern entsprechend geändert haben.

Abhilfe: Klicken Sie auf das NetworkManager-Icon im Panel, führen Sie NETZWERKEINSTELLUNGEN aus, wählen Sie die Ethernet-Schnittstelle aus und aktivieren Sie im Dialogblatt IDENTITÄT die Option AUTOMATISCH VERBINDEN.

**Updates durchführen** Mit SYSTEM • SOFTWARE-AKTUALISIERUNGEN bzw. mit yum update installieren Sie alle verfügbaren Updates.

**EPEL-Paketquelle** Die CentOS-Paketquellen enthalten nur Pakete, die offiziell von Red Hat unterstützt werden. Deshalb ist die Paketauswahl wesentlich kleiner als bei Desktop-Distributionen wie Fedora oder Ubuntu. Abhilfe schafft das Einrichten der Paketquelle EPEL (*Extra Packages for Enterprise Linux*). Das ist eine Sammlung von oft benötigten Zusatzpaketen für RHEL und seine Klone.

<https://fedoraproject.org/wiki/EPEL>

Für die Paketquelle sieht CentOS erfreulicherweise ein eigenes Paket vor. Dieses installieren Sie – fertig!

```
root# yum install epel-release
```

In VirtualBox ist die Bildschirmauflösung anfänglich auf 1024 × 768 Pixel limitiert. Abhilfe schafft die Installation der Guest Additions. Dazu führen Sie zuerst im VirtualBox-Fenster das Menükommando `DEVICES • INSERT GUEST ADDITIONS CD IMAGE` aus, dann innerhalb der virtuellen Maschine die folgenden Kommandos:

CentOS in  
VirtualBox

```
root# yum install gcc make kernel-headers kernel-devel
root# /run/media/<loginname>/VBOXADD*/VBoxLinuxAdditions.run
root# reboot
```

## 3.2 Debian

Keine Distribution steht so sehr für das »reine« Linux wie Debian – und das aus mehreren Gründen:

Debian – das  
»reine« Linux

- ▶ Die Entwicklung von Debian erfolgt ausschließlich durch eine freie Entwicklergemeinschaft. Hinter Debian stehen weder eine Firma noch kommerzielle Interessen, sondern laut Wikipedia über 1000 Entwickler, von denen die meisten ehrenamtlich für Debian arbeiten. In logischer Konsequenz ist sowohl Debian an sich als auch der Zugang zu Updates vollkommen frei.
- ▶ Zu den zentralen Zielen Debians zählt es, dass die Distribution wirklich »frei« im Sinne der Open-Source-Idee bleibt. Die Integration von Binärtreibern oder kommerzieller Software ohne frei verfügbaren Quellcode ist selbstverständlich tabu. Die Debian-Entwickler diskutieren aber auch darüber, ob es vertretbar ist, Firmware-Dateien für Hardware-Geräte mitzuliefern, wenn es dafür keinen Open-Source-Code gibt.
- ▶ Bei Debian sind Stabilität und Sicherheit wichtiger als ganz aktuelle Versionen. Deswegen hinkt eine gewöhnliche Debian-Installation dem aktuellen Entwicklungsstand bei nahezu allen wichtigen Komponenten (Kernel, Xorg, Gnome, KDE, Server-Komponenten etc.) immer ein bis zwei Versionsnummern hinterher. Wer aktuellere Versionen benötigt, kann diese aus den *testing-* oder *unstable-*Paketquellen installieren.
- ▶ Debian unterstützt wesentlich mehr Hardware-Plattformen als jede andere Distribution. Auch das ist ein Grund dafür, dass die Entwicklung einer neuen Debian-Version oft länger dauert als geplant.
- ▶ Die Leitung des Debian-Projekts erfolgt durch eine demokratische Organisation, deren Führungsmitglieder regelmäßig gewählt werden. Die Spielregeln sind in einem »Gesellschaftsvertrag« formuliert:

[https://www.debian.org/social\\_contract.de.html](https://www.debian.org/social_contract.de.html)

Dieser Gesellschaftsvertrag enthält »Richtlinien für Freie Software« (DFSG = *Debian Free Software Guidelines*). Dort sind Richtlinien formuliert, die ein Software-Projekt erfüllen muss, damit es Teil der offiziellen Debian-Pakete werden kann.

**Desktop- und Server-Einsatz** Debian hat im weltweiten Linux-Server-Segment einen relativ hohen Marktanteil. Laut <https://w3techs.com> läuft rund ein Sechstel aller Webserver im Unix/Linux-Segment unter Debian. Dafür gibt es mehrere Gründe: Debian hat sich einen Ruf als sicheres System erarbeitet, es ist kostenlos verfügbar und es lässt sich dank des Debian-Paketsystems über viele Jahre hinweg ohne Neuinstallation immer wieder aktualisieren.

Im Desktop-Segment war Debian nie so stark präsent. Debian hat noch immer den Ruf, dass es schwieriger zu bedienen sei als andere Distributionen. Dieses Vorurteil stimmt schon lange nicht mehr. Fakt ist aber, dass die Installation umständlicher als bei anderen Distributionen ist und dass Sie mit Debian durchwegs ältere Software-Versionen erhalten als beispielsweise mit Ubuntu oder Fedora. Wenn Sie Wert auf neue Gnome-, KDE-, GIMP- oder LibreOffice-Versionen sowie auf aktuelle Hardware-Treiber legen, treffen Sie mit Debian die falsche Wahl.

Die Bedeutung von Debian reicht weit über das hinaus, was sich in Marktanteilen messen lässt: Debian ist ein wichtiges und unverzichtbares Fundament für zahlreiche andere Distributionen, allen voran für Ubuntu. Viele Debian-Werkzeuge, angefangen bei der Paketverwaltung, haben Eingang in andere Distributionen gefunden.

**Kritik** Allen Errungenschaften zum Trotz gibt es natürlich auch Kritik an Debian. Heiß umstritten sind die oft jahrelangen Release-Zyklen, die durch interne Querelen um bisweilen fast schon philosophische Details regelmäßig größer werden als ursprünglich geplant. Ubuntu hat bewiesen, dass es auf der Basis der Debian-Pakete möglich ist, halbjährlich aktuelle Versionen pünktlich zu veröffentlichen. Und gerade der große Erfolg von Ubuntu irritiert manche Debian-Entwickler, weil es den Anschein hat, als würde Ubuntu dank einer besseren Vermarktung gewissermaßen die Ernte Debians einfahren.

**Versionen/Varianten** Im Vergleich zu anderen Distributionen verzichtet Debian dankenswerterweise auf unzählige Distributionsvarianten. Es gibt nur ein Debian, das aus einem Pool von rund 51.000 Paketen besteht. Die genaue Anzahl variiert je nach CPU-Architektur. Je nachdem, welches Installationsmedium Sie einsetzen, müssen Sie bei Bedarf mehr oder weniger Pakete aus dem Internet herunterladen.

Die Installation des Grundsystems kann wahlweise von einer oder mehreren DVDs oder von einem Netzwerkinstallations-Image (*netinst*-Image, rund 250 MiB) erfolgen. Dieses Image enthält nur das Installationsprogramm. Alle Pakete werden während der Installation aus dem Internet oder von einem lokalen Server heruntergeladen. Alle Images können natürlich auch auf einen USB-Stick übertragen werden.

Beeindruckend ist die Hardware-Unterstützung: Während andere Distributionen zumeist nur zwei oder drei CPU-Plattformen unterstützen, sind es bei Debian 9 gleich

zehn: neben Standard-PCs (amd64 und i386) sind dies ARM (arm, armhf und arm64), MIPS (mips, mipsel und mips64el), PowerPC (ppc64el) und S390. Für die x86-Linux-Varianten gibt es nicht nur Installations-, sondern auch Live-Images.

Debian verwendet für jede Version einen Codenamen, der mit Figuren aus dem Film Toy Story übereinstimmt (siehe [Tabelle 3.1](#)). In diesem Buch beziehe ich mich durchwegs auf Debian 9, wenn ich nicht explizit auf eine andere Versionsnummer hinweise.

Codennamen

| Codename       | Version   | Fertigstellung                       |
|----------------|-----------|--------------------------------------|
| <i>Squeeze</i> | Debian 6  | Februar 2011                         |
| <i>Wheezy</i>  | Debian 7  | Mai 2013                             |
| <i>Jessie</i>  | Debian 8  | April 2015                           |
| <i>Stretch</i> | Debian 9  | Juni 2017                            |
| <i>Buster</i>  | Debian 10 | in Entwicklung, voraussichtlich 2019 |

Tabelle 3.1 Debian-Versionen

Umfassende Informationen zu Debian finden Sie auf der offiziellen Website:

Weitere Infos

<https://www.debian.org>

<https://www.debian.org/releases/stretch/amd64>

<https://www.debian.org/releases/stretch/amd64/release-notes>

Werfen Sie auch einen Blick in das *Debian GNU/Linux Anwenderhandbuch* von Frank Ronneburg, dessen Inhalt zwar nicht mehr ganz aktuell ist (die letzten Updates wurden 2014 durchgeführt), das dafür aber vollständig online verfügbar ist:

<http://www.debiananwenderhandbuch.de>

## Debian installieren

ISO-Dateien zum Brennen einer CD/DVD bzw. zur Übertragung auf einen USB-Stick finden Sie hier zum kostenlosen Download:

Installationsmedien

<https://www.debian.org/CD/http-ftp>

Für PC- und Notebook-Installationen benötigen Sie in der Regel das DVD-Image für die Plattform amd64. Es ist selbstverständlich auch für alle 64-Bit-Prozessoren von Intel geeignet!

Anstelle der offiziellen ISO-Medien ist je nach Hardware die Verwendung der *Non-free*-Images empfehlenswert. Diese Medien enthalten Firmware-Dateien für Ethernet- und WLAN-Adapter sowie andere Hardware-Komponenten, die kostenlos weitergege-

ben werden dürfen, aber nicht als Open-Source-Code zur Verfügung stehen und somit der Debian-Philosophie widersprechen.

<https://cdimage.debian.org/cdimage/unofficial/non-free/cd-including-firmware>

#### Installation starten

Wie üblich beginnen Sie die Installation, indem Sie Ihren Rechner neu starten und die Debian-CD oder -DVD einlegen bzw. einen USB-Stick anstecken. Auf der Willkommensseite geben Sie an, welche Installationsvariante Sie nutzen möchten:

|                   |  |
|-------------------|--|
| GRAPHICAL INSTALL | Standardinstallation im Grafikmodus                        |
| INSTALL           | Standardinstallation im Textmodus                          |
| ADVANCED OPTIONS  | Installation für Experten, KDE-Installation, Rescue-System |
| HELP              | Hilfetexte (Drücken Sie <b>F2</b> bis <b>F10</b> .)        |

Standardmäßig startet das Installationsprogramm im Grafikmodus. In den ersten Schritten stellen Sie dann die Sprache und das Tastaturlayout ein.

Wenn Hardware-Probleme auftreten, führen Sie **ADVANCED OPTIONS • EXPERT INSTALL** aus. Sie können nun ganz genau Einfluss auf die einzelnen Installationsschritte und insbesondere auf das Laden von Kernelmodulen nehmen. Das setzt natürlich einiges Linux-Know-how voraus.

#### Fehlende Firmware- Dateien

Im Gegensatz zu den meisten anderen Linux-Distributionen enthält das offizielle Installations-Image keine Firmware-Dateien, deren Code nicht quelloffen zur Verfügung steht. Wenn das Installationsprogramm erkennt, dass eine für die Installation erforderliche Hardware-Komponente aufgrund fehlender Firmware nicht funktioniert, zeigt es eine entsprechende Warnung an.

In solchen Fällen müssen Sie sich auf die Suche nach den Firmware-Dateien machen. Die gängigsten Firmware-Dateien befinden sich in der Datei `firmware.zip` auf der folgenden Seite:

<https://cdimage.debian.org/cdimage/unofficial/non-free/firmware>

Die zu Ihrer Debian-Version passende Firmware-Datei finden Sie im Unterverzeichnis `codename/current`, für Debian 9 also in `stretch/current`. Den Inhalt der Datei packen Sie aus und schreiben ihn auf eine CD oder auf einen USB-Stick. Die Treiberdateien müssen sich auf diesem Datenträger im Unterverzeichnis `firmware` befinden. Der USB-Stick darf ein VFAT-Dateisystem enthalten. Der Platzbedarf für die Dateien beträgt nur einige Megabyte. Es ist nicht notwendig, den USB-Stick vorher zu formatieren. Nachdem Sie den USB-Stick angesteckt haben, können Sie fortfahren. Das Installationsprogramm sucht sich selbst die Firmware-Datei vom Datenträger.



### Verwenden Sie die Non-Free-ISO-Dateien!

Das nachträgliche Einbinden der Firmware-Dateien ist umständlich. Wesentlich einfacher ist es, das Installationsmedium aus den vorhin schon erwähnten Non-Free-ISO-Dateien zu erzeugen. Darin sind die kostenlos verfügbaren Treiber gleich enthalten:

<https://cdimage.debian.org/cdimage/unofficial/non-free/cd-including-firmware>

Falls Ihr Rechner über mehrere Netzwerkschnittstellen verfügt, müssen Sie die richtige auswählen. Sofern Ihr Rechner an einen lokalen Server oder einen ADSL-Router mit DHCP-Server angeschlossen ist, erfolgt die weitere Netzwerkkonfiguration automatisch; Sie müssen nur den gewünschten Rechnernamen (Hostnamen) angeben.

Netzwerk-  
konfiguration

In den nächsten Dialogen geben Sie das Passwort für root ein und legen einen neuen Benutzer an.


root-Passwort,  
Benutzer

Das Installationsprogramm stellt Ihnen unter anderem die folgenden Möglichkeiten zur Partitionierung der Festplatte zur Wahl:

Partitionierung  
der Festplatte

- ▶ **GEFÜHRT – DEN GRÖSSTEN FREIEN SPEICHERBEREICH VERWENDEN:** Diese Option wird nur angezeigt, wenn es auf der Festplatte oder SSD freien Platz gibt, der nicht von Partitionen belegt ist. Das Installationsprogramm richtet dann in diesem Bereich die für Debian erforderlichen Partitionen ein.
- ▶ **GEFÜHRT – VOLLSTÄNDIGE FESTPLATTE VERWENDEN:** Das Installationsprogramm löscht alle Partitionen und verwendet dann die gesamte Festplatte/SSD für die Debian-Installation. In einem weiteren Dialog erscheint wenig später die Frage, ob Sie alle Daten in einer Partition speichern möchten, ob Sie eine getrennte Home-Partition wünschen (das ist empfehlenswert) oder ob auch für die Verzeichnisse `/usr`, `/var` und `/tmp` eigene Partitionen eingerichtet werden sollen. Letzteres ist selten zweckmäßig.
- ▶ **GEFÜHRT – GESAMTE PLATTE VERWENDEN UND LVM EINRICHTEN:** Wie oben, allerdings mit einem LVM-System, das bei späteren Änderungen mehr Flexibilität gibt.
- ▶ **GEFÜHRT – GESAMTE PLATTE MIT VERSCHLÜSSELTEM LVM:** Wie oben, allerdings wird das LVM-System verschlüsselt. Der Schlüssel muss bei jedem Boot-Vorgang angegeben werden, d. h., diese Variante ist für Server-Installationen nur bedingt geeignet.
- ▶ **MANUELL:** Dieser Punkt gibt Ihnen die Möglichkeit, die Partitionierung selbst durchzuführen. Sie können aber auch eine der obigen Varianten wählen und den Vorschlag des Installationsprogramms nach Ihren eigenen Vorstellungen ändern.

Unabhängig davon, für welche Variante Sie sich entscheiden, müssen Sie den Partitionierungsplan nochmals explizit bestätigen. Es besteht also keine Gefahr, dass das Installationsprogramm die Partitionierung vorschnell und unwiderruflich vornimmt.

Bei der manuellen Partitionierung zeigt das Installationsprogramm eine Liste aller verfügbaren Partitionen an (siehe [Abbildung 3.4](#)). Vorhandene Partitionen wählen Sie per Doppelklick oder  aus. Neue Partitionen erstellen Sie, indem Sie den Punkt FREIER SPEICHER am Ende der Liste anklicken. Sie können auch vorhandene Windows- und Linux-Partitionen verkleinern, um so mehr Platz für neue Linux-Partitionen zu schaffen.

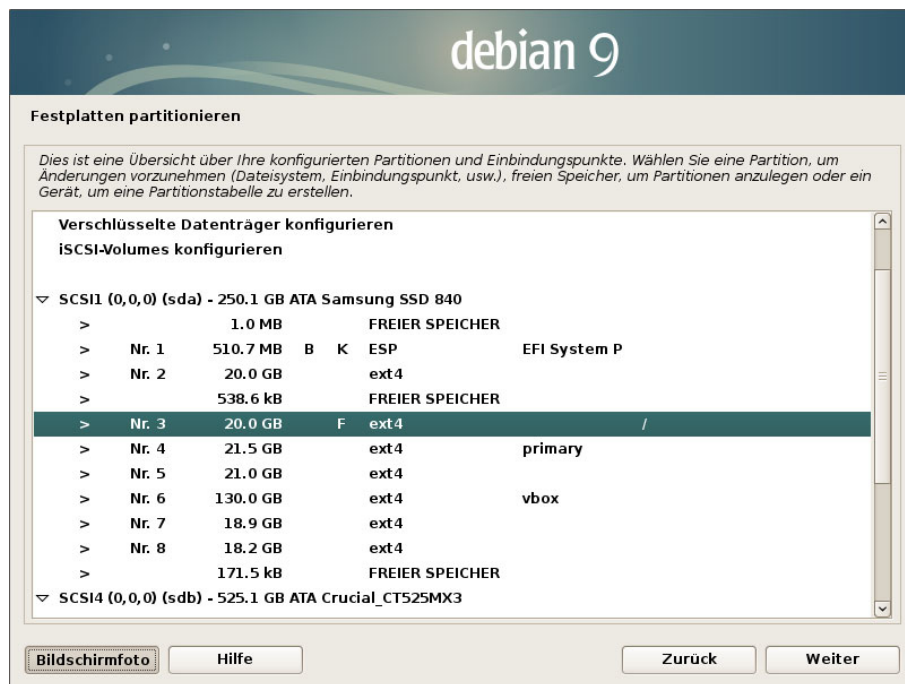


Abbildung 3.4 Partitionierung der Festplatte

Die verschachtelten Dialoge zur Bearbeitung der Partitionen sind leider unübersichtlich und machen von den Möglichkeiten einer grafischen Benutzeroberfläche wenig Gebrauch. Viele Texte in den Dialogen sind als Menükommandos zu interpretieren; sie führen beim Anklicken in weitere Dialoge. Beispielsweise öffnet ein Mausklick auf die Zeile BENUTZEN ALS: NICHT BENUTZEN eine Auswahlliste, in der Sie den gewünschten Dateisystemtyp angeben.

Mit ANLEGEN DER PARTITION BEENDEN speichern Sie die Einstellungen der zuletzt bearbeiteten Partition. Anschließend können Sie eine weitere Partition bearbeiten

oder die Partitionierung abschließen, in dem Sie unterhalb der Partitionsliste den Eintrag **PARTITIONIERUNG BEENDEN UND ALLE DURCHGEFÜHRTEN ÄNDERUNGEN ÜBERNEHMEN** anklicken. Das Installationsprogramm zeigt eine Zusammenfassung der geplanten Änderungen an der Festplattenpartitionierung an und führt diese nach einer weiteren Bestätigung schließlich aus.

Falls die Festplatte bisher unbenutzt war, muss vor der Partitionierung eine Partitionstabelle eingerichtet werden. Auf EFI-Rechnern entscheidet sich Debian für das GPT-Format, zeigt diesbezüglich aber keine Informationen an und bietet auch keine Wahlmöglichkeiten.

In den folgenden Dialogen fragt das Installationsprogramm, von welchem Server es Pakete herunterladen soll, die sich nicht auf dem Installations-Image befinden. Insbesondere bei einer Netzwerkinstallation (»netinst«-CD) ist dieser Punkt essenziell. Antworten Sie mit **JA**, und wählen Sie im nächsten Schritt einen geografisch nahe gelegenen Server aus.

Paketmanager  
konfigurieren

Nach der Installation einiger Pakete werden Sie gefragt, ob Ihre Paketauswahl an einen zentralen Server gemeldet werden soll, um so die populärsten Debian-Pakete zu ermitteln.

Im nächsten Dialog führen Sie eine erste Software-Auswahl durch: Dabei stehen die Paketgruppen **DEBIAN DESKTOP ENVIRONMENT** für verschiedene Desktop-Systeme sowie **WEB SERVER**, **DRUCKSERVER**, **SSH SERVER** und **STANDARD-SYSTEMWERKZEUGE** zur Wahl (siehe [Abbildung 3.5](#)).

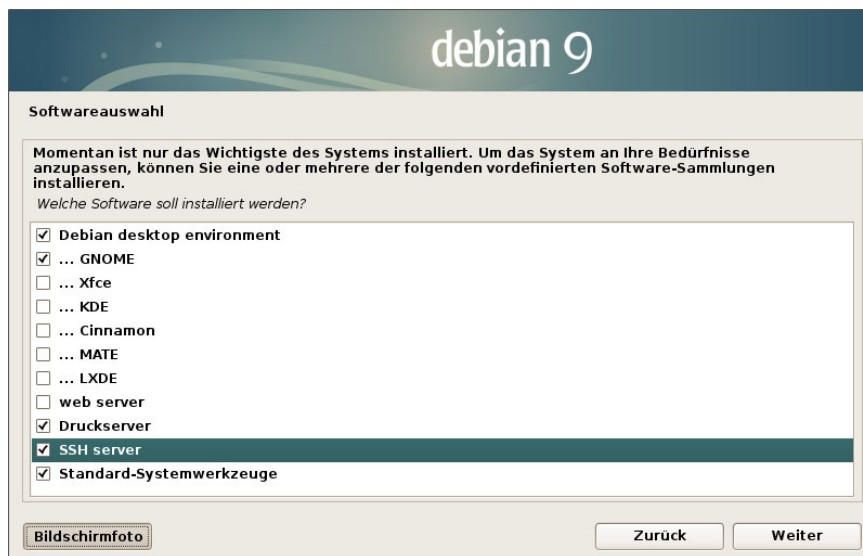


Abbildung 3.5 Installationsumfang einstellen

Sie können in diesem Dialog sogar mehrere Desktop-Systeme gleichzeitig auswählen: Dann haben Sie bei jedem Login die Wahl, welches Desktop-System Sie nutzen möchten. Gleichzeitig führt die Auswahl mehrerer Desktop-Systeme aber zu einem unnötig aufgeblähten System mit viel redundanter Software, also z. B. mehreren Audio-Playern, Foto-Programmen etc.

**GRUB und EFI** Debian verwendet GRUB 2 als Bootloader. Das Installationsprogramm trägt in das GRUB-Menü automatisch alle anderen Betriebssysteme ein, die bereits installiert sind. Nach der GRUB-Installation wird der Rechner neu gestartet.

Die amd64-Version von Debian erlaubt die Installation im EFI-Modus. Dabei wird der Bootloader GRUB in die EFI-Partition installiert. Die automatische Partitionierung denkt auch an die EFI-Partition, falls diese noch nicht existiert. UEFI Secure Boot wird allerdings nicht unterstützt.

**Screenshots** Während der Installation können Sie das aktuelle Aussehen des Installationsprogramms dank eines dafür vorgesehenen Buttons in Screenshots dokumentieren. Die Screenshots werden im Verzeichnis `/var/log` gespeichert, gehen aber beim Neustart des Rechners am Ende der Installation verloren. Um die Bilder zu archivieren, wechseln Sie vor dem Neustart in eine Textkonsole und kopieren die Dateien nach `/target/root` oder in ein anderes Verzeichnis Ihrer Wahl.

## Erste Schritte

**MP3 und Multimedia** Debian kann bereits nach einer Grundinstallation MP3-Dateien und die gängigsten Audio- und Video-Formate abspielen. Auch der MP3-Encoder `lame` kann problemlos installiert werden. Inoffizielle Pakete mit weiteren Codecs, Multimedia-Bibliotheken und -Programmen finden Sie in von Debian unabhängigen Paketquellen, beispielsweise hier:

<https://deb-multimedia.org>

**non-free-Paketquellen aktivieren** Zur Installation von Firmware- und Treiber-Paketen, deren Quellcode nicht frei verfügbar ist, müssen Sie in der Datei `/etc/apt/sources.list` am Ende jeder Zeile `non-free` hinzufügen. Die Datei sollte danach wie das folgende Beispiel aussehen, wobei anstelle von `debian.inode.at` ein möglichst nahe gelegener Mirror-Server anzugeben ist. Wenn Sie die Installation von einem inoffiziellen Non-Free-Medium durchgeführt haben, enthält die Datei die `non-free`-Spalte standardmäßig. Im folgenden Listing wurde eine `deb`-Zeile aus Platzgründen mit `\` über zwei Zeilen verteilt.

```
# in /etc/apt/sources.list
deb http://debian.inode.at/debian/ stretch main contrib non-free
deb http://security.debian.org/debian-security \
    stretch/updates main contrib non-free
...
```

Auf einem meiner Testrechner lief das Grafiksystem nach der Installation nur in der mageren Auflösung von 800 × 600 Pixel. Die Systemeinstellungen von Gnome beklagten sich darüber, dass der Monitor nicht erkannt werden könne, und boten auf meinem 4k-Monitor aber keine höhere Auflösung an.

Firmware-  
Dateien  
installieren

Schuld an dem Problem waren fehlende Firmware-Dateien, auf die der radeon-Treiber angewiesen ist. Sofern die *non-free*-Paketquellen eingerichtet sind (siehe oben), lässt sich das Problem leicht beheben:

```
root# apt install firmware-linux-nonfree
root# reboot
```

Weitere Tipps zum Umgang mit AMD-Grafikkarten finden Sie hier:

<https://wiki.debian.org/AtiHowTo>

Auch die binären Treiber für NVIDIA-Grafikkarten stehen als *non-free*-Pakete zur Verfügung. Vor der Installation müssen Sie in `/etc/apt/sources.list` die Paketquelle *non-free* hinzufügen.

NVIDIA-  
Grafiktreiber

```
root# apt install nvidia-driver
```

Weitere Tipps zur Installation von NVIDIA-Treibern finden Sie in [Abschnitt 21.3](#) sowie auf der folgenden Webseite:

<https://wiki.debian.org/NvidiaGraphicsDrivers>

### 3.3 Fedora

Fedora ist eine Variante von Red Hat Enterprise Linux (RHEL). Die Fedora-Entwicklung wird von Red Hat personell und finanziell unterstützt. Im Gegensatz zu RHEL sind sowohl Fedora an sich als auch alle Updates kostenlos verfügbar. Für Red Hat ist Fedora eine Art Testplattform, um neue Funktionen zu entwickeln und zu testen. Für viele Linux-Freaks ist Fedora hingegen die modernste verfügbare Linux-Distribution. Neue Linux-Konzepte und -Ideen finden sich oft zuerst in Fedora, bevor andere Distributionen nachziehen. Fedora ist üblicherweise auch die Linux-Distribution, mit der Sie die gerade aktuellste Gnome-Version zuerst ausprobieren können.

Trotz der Experimentierfreudigkeit der Entwickler hat sich Fedora in den letzten Jahren als relativ stabile Distribution erwiesen. Hier kommt ganz offensichtlich das Know-how der Red-Hat-Entwickler zum Tragen. Bei der Benutzerfreundlichkeit hat Fedora in den letzten Jahren große Fortschritte gemacht: Hatte Fedora früher den

Nimbus »von Freaks für Freaks«, so ist die Distribution mittlerweile ebenso einfach zu installieren und zu nutzen wie Ubuntu.

Der größte Nachteil von Fedora ist die kurze Lebensdauer: Fedora-Updates werden für den Zyklus von zwei Versionen plus einem Monat gepflegt. Mit anderen Worten: Der Update-Zeitraum für Fedora 26 endet einen Monat, nachdem Fedora 28 fertiggestellt ist. Da der Release-Zyklus normalerweise sechs Monate beträgt, entspricht dies einer Update-Spanne von nur 13 Monaten.

**Varianten und Spins** Fedora steht in den drei Varianten *Workstation*, *Server* und *Atomic* (für Linux-Docker-Kubernetes) zur Auswahl, wobei es nur die Workstation-Variante auch in einer 32-Bit-Edition gibt. Außerdem gibt es noch sogenannte *Spins*. Das sind Fedora-Varianten mit einer vordefinierten Paketauswahl für einen bestimmten Verwendungszweck, z. B. mit Werkzeugen zur Sicherheitsanalyse oder mit anderen Desktop-Systemen als Gnome. So gibt es Spins für KDE, Xfce, LXDE, MATE und Cinnamon:

<https://spins.fedoraproject.org>

Zu guter Letzt gibt es Fedora-Varianten für andere CPU-Architekturen (ARM, PowerPC, S390) sowie für diverse Virtualisierungs- und Cloud-Systeme vorbereitete Images:

<https://alt.fedoraproject.org>

**Dokumentation** Weitere Informationen zu Fedora finden Sie auf den folgenden Webseiten:

<https://getfedora.org>

<https://www.fedoraforum.de>

<https://docs.fedoraproject.org>

<https://access.redhat.com/documentation/en/red-hat-enterprise-linux>

### Fedora installieren

**Installationsmedien** Üblicherweise laden Sie Fedora als Live-ISO-Image von <https://getfedora.org> herunter. Das Image ist auch für USB-Sticks geeignet. Alternativ können Sie auch das Netzwerkinstallationsmedium verwenden. Das darauf enthaltene Installationsprogramm lädt die zu installierenden Pakete aus dem Internet herunter.

Im Live-System können Sie Fedora ausprobieren oder das Installationsprogramm starten. Dieses fasst nach der Einstellung der Sprache alle Konfigurationseinstellungen in einem einzigen Dialog zusammen (siehe [Abbildung 3.6](#)). Normalerweise müssen Sie dann nur einen einzigen Punkt ändern, nämlich das INSTALLATIONS-ZIEL.



Abbildung 3.6 Überblick über die Installationseinstellungen

Ein Klick auf das Icon INSTALLATIONS-ZIEL führt in einen Partitionseditor, der leider in Hinblick auf intuitive Bedienung keine Meisterleistung darstellt. (Anders formuliert: Ich habe in den letzten 20 Jahren mit vielen Partitionseditoren gearbeitet. Der von CentOS/Fedora/RHEL ist derjenige, dessen Bedienung am unlogischsten ist.)

Partitionierung

Im ersten Dialog werden die gefundenen lokalen Festplatten und SSDs aufgelistet (siehe [Abbildung 3.7](#)). Sie müssen darauf achten, dass diejenigen Festplatten mit einem Auswahlhäkchen versehen sind, auf denen Partitionen erstellt oder genutzt werden sollen. (Wenn es nur eine Festplatte gibt, ist diese schon ausgewählt und Sie müssen nur FERTIG anklicken. Entscheidend ist das Auswahlhäkchen. Ob der Datenträger gerade mit der Maus ausgewählt wurde und mit blauem Hintergrund dargestellt wird, ist hingegen irrelevant.)

Durch einige Optionen steuern Sie, wie Sie weiter vorgehen möchten:

- ▶ Mit AUTOMATISCH kümmert sich das Installationsprogramm um eine geeignete Partitionierung. Das funktioniert dann gut, wenn Sie einen noch leeren Datenträger vollständig nutzen möchten – z. B. bei der Installation in einer virtuellen Maschine. Wenn Sie diese Variante wählen, wird automatisch ein LVM-System eingerichtet, was spätere Änderungen am Setup erleichtert.
- ▶ ANGEPASST führt im nächsten Schritt in einen Editor, in dem Sie die Partitionen manuell einrichten und vorhandene Partitionen ändern (oder auch verkleinern) können.

- ▶ ADVANCED CUSTOM (BLIVET GUI) ist erst seit Fedora 26 verfügbar. Diese Option führt ebenfalls in einen Partitionseditor, der für fortgeschrittene Benutzer konzipiert ist. Persönlich finde ich diesen Editor einfacher zu verwenden als den Standardeditor, aber das liegt vielleicht daran, dass ich tatsächlich ein fortgeschrittener Linux-Anwender bin :-)
- ▶ Die Option ICH MÖCHTE WEITEREN SPEICHERPLATZ VERFÜGBAR MACHEN steht nur in Kombination mit der Option AUTOMATISCH zur Auswahl. Sie haben damit die Möglichkeit, vorweg einige Partitionen zu löschen oder zu verkleinern.
- ▶ Die Option MEINE DATEN VERSCHLÜSSELN führt dazu, dass das LVM-System verschlüsselt wird.

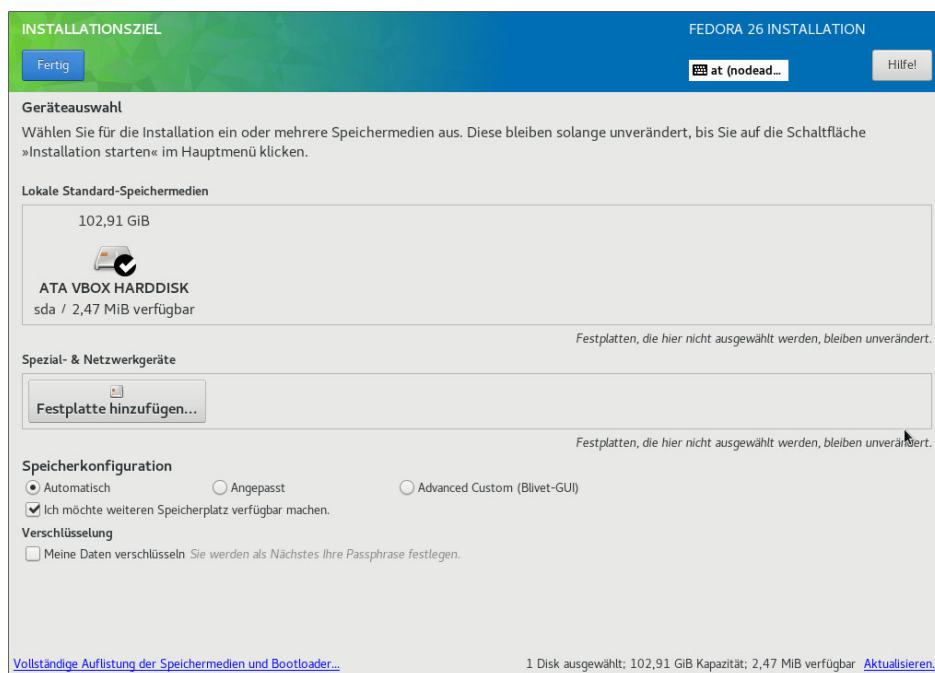


Abbildung 3.7 Auswahl des Partitionierungsverfahrens

Sofern Sie sich für die automatische Konfiguration entschieden haben und die Festplatte bisher leer war, sind Sie mit dem Anklicken von FERTIG tatsächlich fertig und gelangen zurück in den Hauptdialog. Fedora führt die Partitionierung selbstständig aus, findet es aber nicht der Mühe wert, Ihnen das Ergebnis seiner Überlegungen mitzuteilen. Das ist demnach meine Aufgabe.

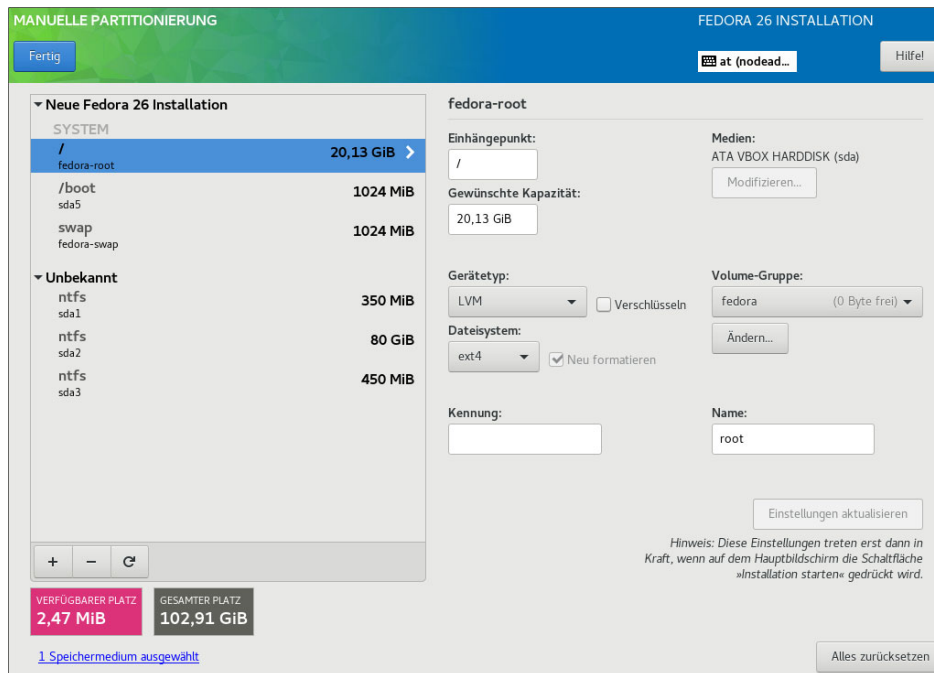
Das Installationsprogramm richtet eine 500 MiB große Boot-Partition sowie eine zweite Partition ein, die den restlichen Datenträger ausfüllt und als Physical Volume für ein LVM-System dient. Zwei oder drei Logical Volumes nehmen das Swap-System,



die bis zu 50 GiB große Systempartition und bei großen Datenträgern eine getrennte Home-Partition auf, wobei jeweils das Dateisystem ext4 zum Einsatz kommt. (Die Server-Variante von Fedora verwendet xfs.)

Es gibt keine Möglichkeit, die automatische Konfiguration zu beeinflussen oder nachträglich zu modifizieren. Wenn Sie die Partitionierung selbst durchführen möchten, müssen Sie die Option ANGEPASST wählen. FERTIG führt dann in einen weiteren Dialog (siehe [Abbildung 3.8](#)).

**Manuelle Partitionierung (Standardprogramm)**



**Abbildung 3.8** Manuelle Partitionierung im Standardeditor

In der Leiste links werden alle auf den Festplatten vorhandenen und neu einzurichtenden Partitionen bzw. Dateisysteme aufgelistet. Die Partitionen sind gruppiert: Die erste Gruppe, die anfänglich leer ist, beschreibt das neue Fedora-System. Die weiteren Gruppen ordnen die vorhandenen Partitionen der Festplatte oder SSD den bereits installierten Betriebssystemen zu. Dabei kann es durchaus vorkommen, dass ein und dieselbe Partition in mehreren Gruppen angezeigt wird – z. B. eine Swap-Partition, die parallel von mehreren Linux-Distributionen genutzt wird.

Um Platz zu schaffen, können Sie vorhandene Partitionen löschen oder verkleinern. Um neue Partitionen einzurichten, klicken Sie auf den Plus-Button und geben vorerst nur zwei Parameter an: unter EINHÄNGEPUNKT das Mount-Verzeichnis bzw. die Bezeichnung SWAP sowie die gewünschte Größe in MiB oder GiB. Erst im zweiten

Schritt wählen Sie den Dateisystemtyp (in der Regel ext4) und geben unter GERÄTETYP an, ob das Dateisystem in einer gewöhnlichen Partition oder in einem Logical Volume eingerichtet werden soll.

Wenn Sie alle Partitionen wunschgemäß eingerichtet haben, schließen Sie den Vorgang mit dem Button FERTIG oben links ab. Das Installationsprogramm zeigt nun eine Zusammenfassung der anstehenden Aktionen an, also z. B. welche Partitionen gelöscht oder neu erstellt werden. Mit der Bestätigung dieser Daten gelangen Sie zurück in den Hauptdialog.

### Die Festplatte/SSD wird erst nach Ihrem OK verändert

Auch wenn die Bedienung des Installationsprogramms gewöhnungsbedürftig ist, gibt es zumindest einen Pluspunkt: Änderungen, die Sie im Editor durchführen, werden nicht sofort ausgeführt. Vielmehr wartet das Installationsprogramm, bis Sie mit Ihrer Konfiguration fertig sind. Erst nach einer Rückfrage mit der Zusammenfassung aller ausstehenden Aktionen werden diese ausgeführt.

#### Manuelle Partitionierung (Blivet GUI)

Die neue Blivet GUI zur Partitionierung (verfügbar seit Fedora 26) zeigt in der linken Seitenleiste die auf dem Rechner vorgefundenen Festplatten, SSDs und LVM-Systeme an. Zum gerade ausgewählten Element zeigt der Hauptbereich des Editors dann die Liste der Partitionen bzw. Logical Volumes (LVs) an (siehe [Abbildung 3.9](#)). Über Buttons und Kontextmenükommandos können Sie nun Partitionen und LVs ändern, neu einrichten oder löschen.

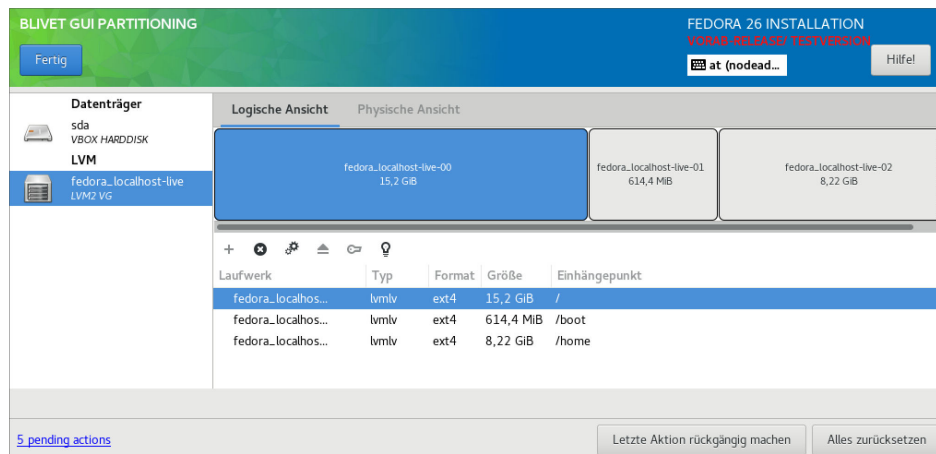


Abbildung 3.9 Manuelle Partitionierung in der Blivet GUI

#### Zeitzone, Hostname

Das Installationsprogramm wählt eine zur Spracheinstellung passende Zeitzone. Wenn Sie damit nicht einverstanden sind, können Sie mit ZEIT & DATUM eine andere Zeitzone auswählen und die Parameter des NTP-Servers einstellen.

Der Punkt NETZWERK & RECHNERNAME verspricht mehr, als er hält. Tatsächlich können Sie hiermit lediglich den Hostnamen einstellen. Die Defaulteinstellung lautet `localhost.localdomain`.

Sobald Sie den Hauptdialog mit **INSTALLATION STARTEN** abschließen, beginnt die eigentliche Installation. Die Wartezeit bis zum Abschluss der Installation können Sie nutzen, um das Passwort für `root` einzustellen sowie um einen gewöhnlichen Benutzer einzurichten.

**Root-Passwort,  
Benutzer  
einrichten**

Während Sie den Benutzer einrichten, können Sie ihn mit einer Option zum Administrator machen. Das bedeutet, dass der Benutzer zur Gruppe `wheel` hinzugefügt wird. Alle Benutzer dieser Gruppe dürfen `sudo` zur Durchführung von Administrationsarbeiten nutzen (siehe [Abschnitt 12.3](#), »Prozesse unter einer anderen Identität ausführen (sudo)«).

## Erste Schritte

Nach dem ersten Login sollten Sie ein Update durchführen – wahlweise mit dem Programm **SOFTWARE-AKTUALISIERUNG** oder im Terminal mit `dnf update`. Dieses Update dauert oft ähnlich lange wie die Installation. Sie können aber ohne Weiteres parallel dazu andere Arbeiten erledigen.

**Erstes Update**

Standardmäßig wird zwar ein SSH-Server installiert, aber nicht automatisch gestartet. Abhilfe schaffen die folgenden zwei Kommandos:

**SSH-Server  
aktivieren**

```
root# systemctl start sshd
root# systemctl enable sshd
```

Wenn Sie als `root` arbeiten, erscheinen bei der Ausführung von `mv` und `rm` ständig Sicherheitsabfragen, ob Sie die Operation wirklich durchführen möchten. Diese Sicherheitsabfragen hören auf, wenn Sie die `alias`-Anweisungen aus `/root/.bashrc` entfernen.

**alias-Einstel-  
lungen für root**

In den offiziellen Fedora-Paketen fehlen aus Lizenz- und Patentgründen eine Menge oft benötigter Pakete: Treiber für AMD- und NVIDIA-Grafikkarten, MP3-Unterstützung etc. Die größten alternativen Paketquellen, nämlich *Livna*, *Freshrpms* und *Dribble*, haben sich 2008 unter dem Namen *RPM Fusion* zusammengeschlossen. RPM Fusion ist somit die wichtigste inoffizielle Quelle für Fedora-Erweiterungen. Aus rechtlichen Gründen wurde ein traditionell von Livna angebotenes Paket nicht in RPM Fusion integriert. Die Livna-Paketquelle existiert deswegen weiterhin und bietet dieses eine Paket an. (Den Namen des Pakets mag ich hier – ebenfalls aus rechtlichen Gründen – nicht nennen.)

**Zusätzliche  
Paketquellen  
einrichten**

<https://rpmfusion.org>  
<http://rpm.livna.org>

Um eine zusätzliche Paketquelle in Yum zu integrieren, müssen Sie eine neue Datei in `/etc/yum.repos.d` anlegen sowie einen Schlüssel für die Paketquelle einrichten. (Yum ist das Paketverwaltungssystem von RHEL/CentOS, siehe [Abschnitt 19.3](#), »Yum«.) Die meisten Paketquellen erleichtern Ihnen diese Arbeit durch ein kleines RPM-Paket, das alle erforderlichen Dateien enthält. Die Kommandos zur Installation dieses RPM-Pakets kopieren Sie am einfachsten von den oben genannten Webseiten in ein Terminalfenster und führen sie dort aus.

RPM Fusion stellt eigentlich zwei Paketquellen zur Verfügung: `free` und `nonfree`. Der Unterschied besteht darin, dass `free`-Pakete Open-Source-Software enthalten, `nonfree`-Pakete dagegen kostenlose kommerzielle Software, etwa Grafiktreiber.

Bei der ersten Installation von Paketen aus der RPM-Fusion-Paketquelle müssen Sie die Schlüssel dieser Paketquelle importieren. Nach meinen Erfahrungen funktioniert das am besten, wenn Sie die Installation mit `yum` durchführen, nicht mit einem grafischen Paketverwaltungswerkzeug.

**Multimedia-Pakete** Mit Fedora 26 werden zwar standardmäßig mehr Decoder und Encoder mitgeliefert als je zuvor (unter anderem für das Format MP3, da die entsprechenden Patente abgelaufen sind), dennoch fehlen viele Multimedia-Codex in den offiziellen Paketquellen. Diesen Mangel beheben Sie, indem Sie zuerst die RPM-Fusion-Paketquellen `free` und `nonfree` einrichten und dann das folgende Kommando ausführen:

```
root# dnf install gstreamer1-plugins-bad* gstreamer1-plugins-ugly*
```

**NVIDIA-Treiber** Wenn Sie auf die binären Grafiktreiber von NVIDIA angewiesen sind, treffen Sie mit Fedora keine gute Wahl. Fedora enthält grundsätzlich die allerneueste Version des X-Servers bzw. von Wayland, oft lange bevor deren Entwicklung ausgereift oder gar abgeschlossen ist. Das hat zur Folge, dass die Treiber von NVIDIA häufig noch inkompatibel zum Grafiksystem sind.

Der NVIDIA-Treiber ist im Paket `akmod-nvidia` der RPM-Fusion-Paketquelle versteckt. Während der Installation wird automatisch `/etc/X11/xorg.conf` modifiziert, sodass der neue Treiber nach einem Neustart des Rechners automatisch zum Einsatz kommt. Sie installieren den Treiber mit:

```
root# dnf install akmod-nvidia
```

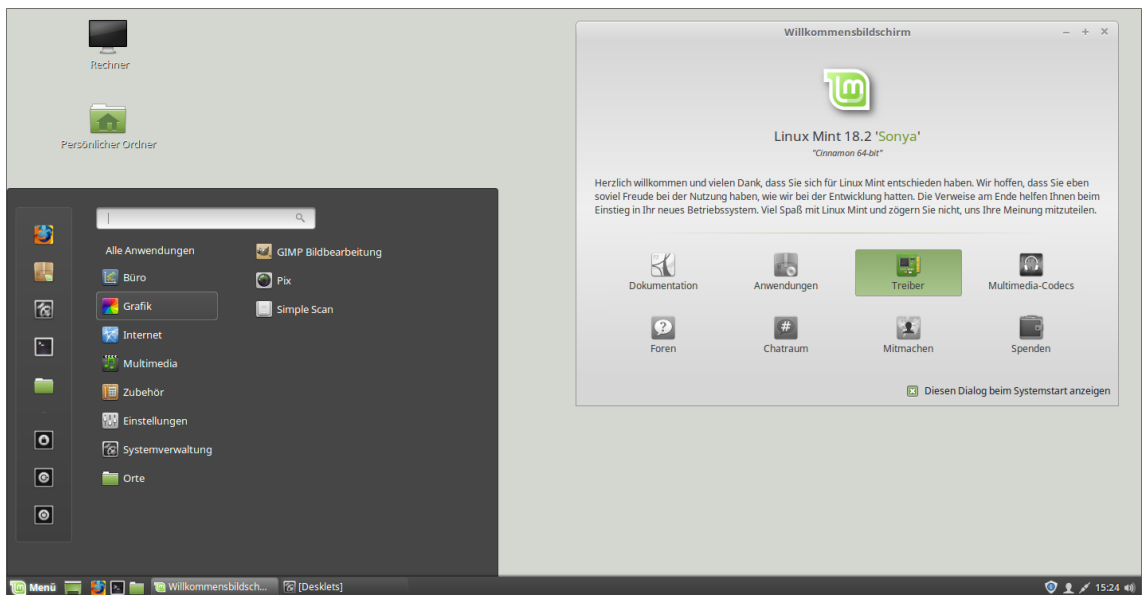
## 3.4 Linux Mint

Linux Mint (<http://linuxmint.com>) ist eine sehr populäre Variante zu Ubuntu Linux. Linux Mint basiert auf denselben Paketquellen wie Ubuntu, wobei generell nur LTS-Versionen von Ubuntu verwendet werden. Anstelle des Unity-Desktops verwendet Mint aber je nach Ausprägung das Desktop-System MATE oder Cinnamon (sie-

he [Abschnitt 5.10](#) und [Abschnitt 5.11](#)). Linux Mint weicht in weiteren Punkten von Ubuntu ab:

- ▶ Mint verwendet einen eigenen Dateimanager, eigene Systemeinstellungsmodul sowie diverse kleinere Zusatzprogramme und differenziert sich so auch funktionell von Ubuntu. Das Ziel aller Eigenentwicklungen ist eine einfachere Bedienung.
- ▶ Zusätzlich zu den Ubuntu-Paketquellen verwendet Mint eigene Paketquellen. Diese dienen nicht nur dazu, in Ubuntu nicht vorgesehene Pakete anzubieten, sondern sie ermöglichen es Mint auch, einzelne Ubuntu-Pakete durch neuere Versionen zu ersetzen.
- ▶ Die Benutzeroberfläche von Mint lässt sich viel weitergehender gestalten als die von Ubuntu. Auf dem Desktop können außer Icons auch Miniprogramme («Desklets») platziert werden.

Linux Mint zählt im Gegensatz zu Kubuntu, Lubuntu etc. nicht zu den »offiziellen« Ubuntu-Derivaten.



**Abbildung 3.10** Der Cinnamon Desktop von Linux Mint mit dem charakteristischen Startmenü und dem Willkommensassistenten, der unter anderem bei der Installation von Programmen, proprietären Treibern und Multimedia-Codices hilft.

Linux Mint steht in verschiedenen Ausprägungen zur Verfügung, die sich einerseits durch das verwendete Desktop-System unterscheiden (die Eigenentwicklung Cinnamon bzw. MATE, KDE oder Xfce) und andererseits durch die Code-Basis (Ubuntu oder

**Varianten**

Debian). Ich gehe in diesem Abschnitt nur auf die populärste Variante ein: auf Linux Mint mit Cinnamon Desktop und Ubuntu-Basis.

**Installation** ISO-Medien für Linux Mint stehen auf <http://www.linuxmint.com/download.php> zum Download zur Verfügung. Die Installation erfolgt genau wie bei Ubuntu (siehe [Abschnitt 3.6](#)), auch wenn die Dialoge des Installationsprogramms optisch ein wenig anders gestaltet sind.

**EFI** Auf EFI-Rechnern verwendet Linux Mint wie Ubuntu das Verzeichnis `/boot/efi/EFI/ubuntu` und legt dort seine Boot-Dateien ab. Das führt bei einer Parallelinstallation von Ubuntu und Linux Mint dazu, dass die eine Distribution die Boot-Dateien der anderen Distribution überschreibt. Das ist insofern kein großes Problem, als das GRUB-Menü beide Distributionen enthält. Sie können also weiterhin beide Distributionen starten.

Im Gegensatz zu Ubuntu ist Mint allerdings nicht kompatibel zu UEFI Secure Boot. Deswegen müssen Sie Secure Boot im EFI-Menü deaktivieren, bevor Sie Mint installieren.

**Cinnamon** Der Cinnamon Desktop basiert grundsätzlich auf Gnome 3 und dessen Bibliotheken. Anstelle der `gnome-shell` läuft unter Mint der selbst entwickelte Window Manager *Muffin* und anstelle des Dateimanagers Nautilus dessen Fork *Nemo*. Rein optisch hat Cinnamon stärkere Ähnlichkeiten zu Gnome 2 als zu Gnome 3 – und das ist durchaus gewollt. Eine kurze Beschreibung des Desktops folgt in [Abschnitt 5.11](#), »Cinnamon«.

**Paketverwaltung** Die Paketverwaltung funktioniert wie bei Ubuntu (siehe auch [Abschnitt 19.7](#), »APT«). Die Datei `/etc/apt/sources.list.d/official-package-repositories.list` enthält aber außer der Ubuntu-LTS-Paketquelle eine weitere Mint-spezifische Paketquelle. Auf Kommandoebene können Sie mit `apt` sowohl das ganze System aktualisieren als auch einzelne Pakete installieren.

Sehr eigenwillig sind aber die Einstellungen der grafischen Benutzeroberfläche `mint-update` zur Durchführung von Updates. Mint teilt die Pakete in fünf unterschiedliche Gruppen ein. Standardmäßig versucht Mint einen Mittelweg aus Stabilität und Sicherheit zu beschreiben. `mintupdate` berücksichtigt nur Updates, die die Mint-Entwickler empfehlen bzw. die Pakete betreffen, bei denen Linux Mint selbst nach einem fehlgeschlagenen Update funktionsfähig bleibt (siehe [Abbildung 3.11](#)).

Da stellt sich natürlich sofort die Frage, welche von Ubuntu vorgesehenen Updates Mint als unsicher betrachtet. Die Antwort erhalten Sie, wenn Sie in `mintupdate` das Kommando `BEARBEITEN • AKTUALISIERUNGSRICHTLINIE` ausführen und die Option `IMMER ALLES AKTUALISIEREN` aktivieren. Plötzlich zeigt `mintupdate` auch Updates für den Linux-Kernel, für `systemd` sowie für das Grafiksystem an.

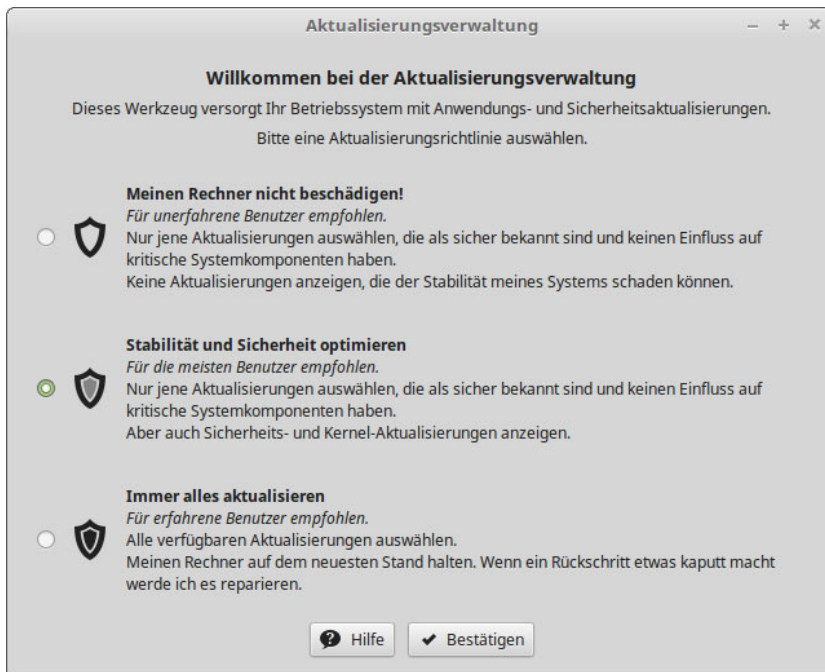


Abbildung 3.11 Die Defaulteinstellungen der Aktualisierungsverwaltung von Linux Mint

### Aktivieren Sie alle Updates!

Sofern Sie Updates nicht ohnedies mit `apt` oder `apt-get` durchführen, sollten Sie die Variante **IMMER ALLES AKTUALISIEREN** in den Update-Einstellungen aktivieren! `mint-update` verschweigt Ihnen sonst Updates, die durchaus wichtig sein können.

Neben `mintupdate` gibt es eine Reihe weiterer Mint-spezifischer Programme. Deren Namen spüren Sie am einfachsten auf, wenn Sie in einem Terminalfenster `mint` eingeben und dann `[↵]` drücken. Die folgende Liste nennt die wichtigsten Vertreter:

Mint-spezifische Programme

- ▶ `mintbackup` ist ein simples Backup-Werkzeug.
- ▶ `mintdrivers` unterstützt Sie ähnlich wie unter Ubuntu bei der Installation von Hardware-Treibern.
- ▶ `mintinstall` hilft bei der Installation von Programmen.
- ▶ `mintnanny` ist ein äußerst minimalistisches Programm zum Sperren ausgewählter Webseiten. Für jeden im Programm angegebenen Hostnamen (z. B. `youtube.com`) fügt das Programm die Zeile `127.0.0.1 hostname` in die Datei `/etc/hosts` ein. Jeder Versuch, eine entsprechende Seite aufzurufen, wird nun auf `localhost` umgeleitet.
- ▶ `mintstick` hilft dabei, eine ISO-Datei auf einen USB-Stick zu schreiben.

## 3.5 openSUSE

openSUSE zählt zu den im deutschen Sprachraum am weitesten verbreiteten Linux-Distributionen. Ein wesentliches Unterscheidungsmerkmal der diversen SUSE-Distributionen gegenüber der Konkurrenz ist das allumfassende Konfigurations- und Administrationswerkzeug YaST (*Yet another Setup Tool*). openSUSE gilt zudem als eine der besten KDE-Distributionen. openSUSE steht damit im Gegensatz zu den meisten anderen Distributoren, die sich primär auf Gnome konzentrieren.

**Der Name SUSE** Die Abkürzung SUSE stand ursprünglich für »Gesellschaft für Software und Systementwicklung«. SUSE (damals noch in der Schreibweise SuSE) war also ursprünglich eine deutsche Firma. 2003 hat Novell SUSE gekauft, 2010 übernahm Attachmate Novell. 2014 wechselte SUSE nochmals den Besitzer und gehört seither zum Software-Unternehmen Micro Focus.

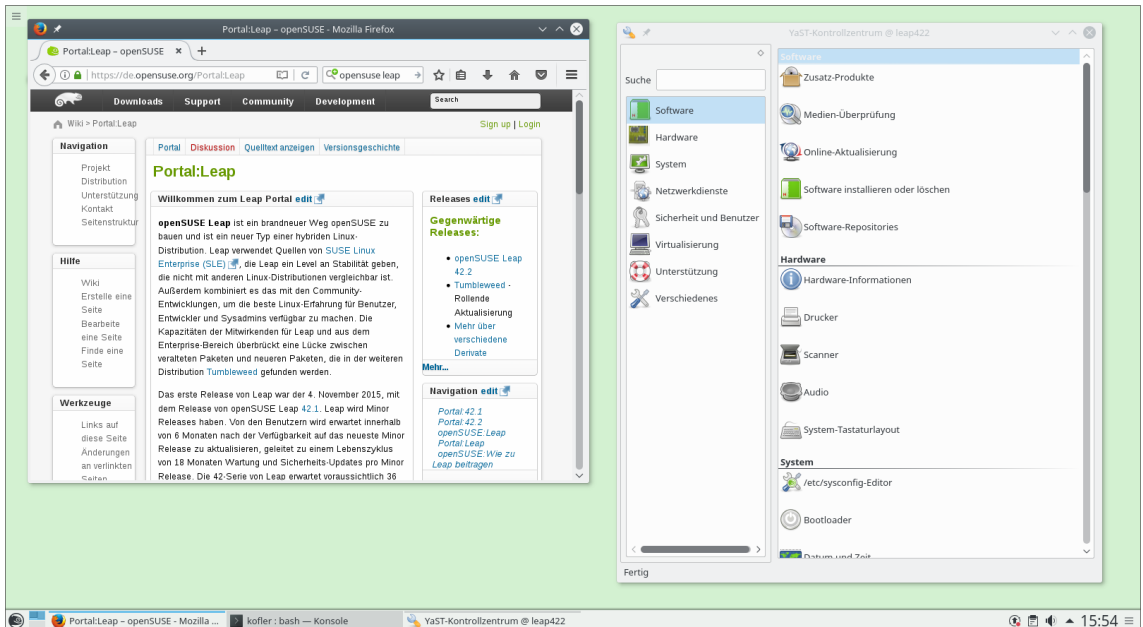
**openSUSE** openSUSE ist eine kostenlose Variante zu den kommerziellen SUSE-Distributionen. Die Entwicklung wird zwar stark von SUSE-Mitarbeitern getragen, es gibt aber öffentliche Beta-Versionen, Mailinglisten, eine Bug-Datenbank und eine aktive Community, die an der Entwicklung teilnimmt und diese unterstützt. Damit spielt openSUSE für SUSE eine ähnliche Rolle wie Fedora für Red Hat.

**openSUSE Leap** Trotz mehrerer optisch ansprechender openSUSE-Versionen in den vergangenen Jahren, die in der Fachpresse durchaus Lob erhielten, schlitterte das openSUSE-Projekt in eine Sinnfindungskrise. Wegen des Support-Zeitraums von nur 18 Monaten war die Distribution für den Langzeiteinsatz ungeeignet. Gleichzeitig konnte openSUSE bei der Entwicklung neuer Features bzw. bei der Aktualität nicht mit Fedora mithalten. Es gab also für Linux-Enthusiasten und -Programmierer wenig plausible Gründe, gerade openSUSE einzusetzen.

Diese Probleme haben auch die openSUSE-Entwickler erkannt und im Herbst 2015 das neu konzipierte »openSUSE Leap« mit der willkürlichen Versionsnummer 42.1 vorgestellt. Der Plan geht dahin, als Unterbau für openSUSE künftig die kommerzielle SUSE-Distribution zu verwenden (SUSE Linux Enterprise, kurz SLE). Das verspricht Stabilität und spart dem vergleichsweise kleinen openSUSE-Entwicklerteam Zeit und Mühe. Es kann sich nun darauf konzentrieren, openSUSE mit einem aktuellen Kernel sowie aktuellen Desktop-Komponenten zu auszustatten.

**Update-Zyklen** Major Releases von openSUSE Leap werden ungefähr alle drei Jahre freigegeben, Minor Release circa jährlich. Innerhalb eines Minor Release gibt es für rund 18 Monate Updates. Danach muss entweder eine Neuinstallation des nächsten Minor Release oder ein Distributions-Update durchgeführt werden. Letzten Endes bedeutet das aber, dass der Wartungszeitraum einer openSUSE-Leap-Version nicht länger ist als bisher.





**Abbildung 3.12** Der KDE-Desktop von openSUSE Leap mit je einem Firefox- und einem YaST-Fenster

Schon die Versionsnummer 42.1, 42.2 etc. waren schwer zu begreifen, aber es kommt noch besser: Voraussichtlich 2018 wird die nächste Major-Version von SLE erscheinen. Da man die Versionsnummer 13 vermeiden wollte, wird SLE einen Sprung auf die Versionsnummer 15 machen. Ab diesen Zeitpunkt werden dann die openSUSE-Leap-Nummern denen von SLE angeglichen. Damit wird also auf openSUSE Leap 42.3 (erscheint voraussichtlich im Juli 2017) die Version 15 folgen – quasi ein Sprung zurück.

**Versionsnummern-Chaos**

Wenn in diesem Buch einfach von *SUSE* die Rede ist, dann beziehe ich mich auf openSUSE Leap 42.2, das ich in meinen Tests verwendet habe. Der Großteil der zusammengestellten Informationen gilt aber auch für die SUSE-Enterprise-Distributionen der Version 12.

Eine sehr interessante Alternative zu openSUSE Leap ist Tumbleweed: Dabei handelt es sich um eine openSUSE-Variante, die als »Rolling Release« konzipiert ist: Einmal installiert, erhält diese Distribution im Rahmen des Update-Systems ständig aktuellere Software-Versionen, sodass (zumindest in der Theorie) nie wieder eine Neuinstallation bzw. ein Distributions-Update erforderlich ist.

**Tumbleweed**

Das Tumbleweed-Projekt basiert auf *Factory*, also dem Entwicklungsbranch von openSUSE. Die Tumbleweed-Entwickler bemühen sich zwar, neue Software-Versionen erst

dann freizugeben, wenn die Programme einigermaßen stabil laufen. Dennoch sind beim Einsatz von Tumbleweed natürlich gelegentlich Probleme zu erwarten, wenn eine neue Software-Version doch noch Fehler enthält oder Inkompatibilitäten mit anderen Komponenten verursacht.

Meine Erfahrungen mit Tumbleweed waren durchaus positiv. Tumbleweed vermittelt einen fließenden Übergang von einer openSUSE-Version zur nächsten. Es gibt aber auch Einschränkungen: Dateisystemwechsel wie jener von ext4 auf btrfs sind im Rahmen von Paket-Updates unmöglich. Hin und wieder erfordert also auch Tumbleweed eine Neuinstallation, wenn Sie alle aktuellen Features der Distribution verwenden möchten.

Die Tumbleweed-Projektseite warnt eindringlich vor dem Einsatz von Tumbleweed, wenn Sie proprietäre Treiber benötigen (NVIDIA) oder nichtoffizielle Paketquellen aktiviert haben:

<https://en.opensuse.org/Portal:Tumbleweed>

**Links** Ausführliche Informationen zu openSUSE und SUSE sowie Handbücher im HTML- und PDF-Format finden Sie auf den folgenden Webseiten:

<https://www.opensuse.org>

[https://en.opensuse.org/SDB:Official\\_documentation](https://en.opensuse.org/SDB:Official_documentation)

<https://www.suse.com>

#### openSUSE installieren

Auf der Website <https://software.opensuse.org> steht ein ca. 4,7 GiB großes ISO-Image zum Download zur Verfügung. Das Image können Sie auf eine DVD brennen oder auf einen USB-Stick übertragen. Es eignet sich zur Installation aller gängigen Desktop-Systeme: KDE, Gnome, Xfce oder LXDE.

Auf BIOS-Rechnern können Sie auf der Begrüßungsseite des Installationsprogramms mit **[F2]** die gewünschte Sprache einstellen und, falls notwendig, mit **[F3]** die Auflösung des Grafiksystems ändern. Falls das Grafiksystem Probleme bereitet, wählen Sie hier TEXTMODUS. Auf EFI-Rechnern fehlen diese Konfigurationsmöglichkeiten. Dort stellen Sie die Sprache direkt im Installationsprogramm ein.

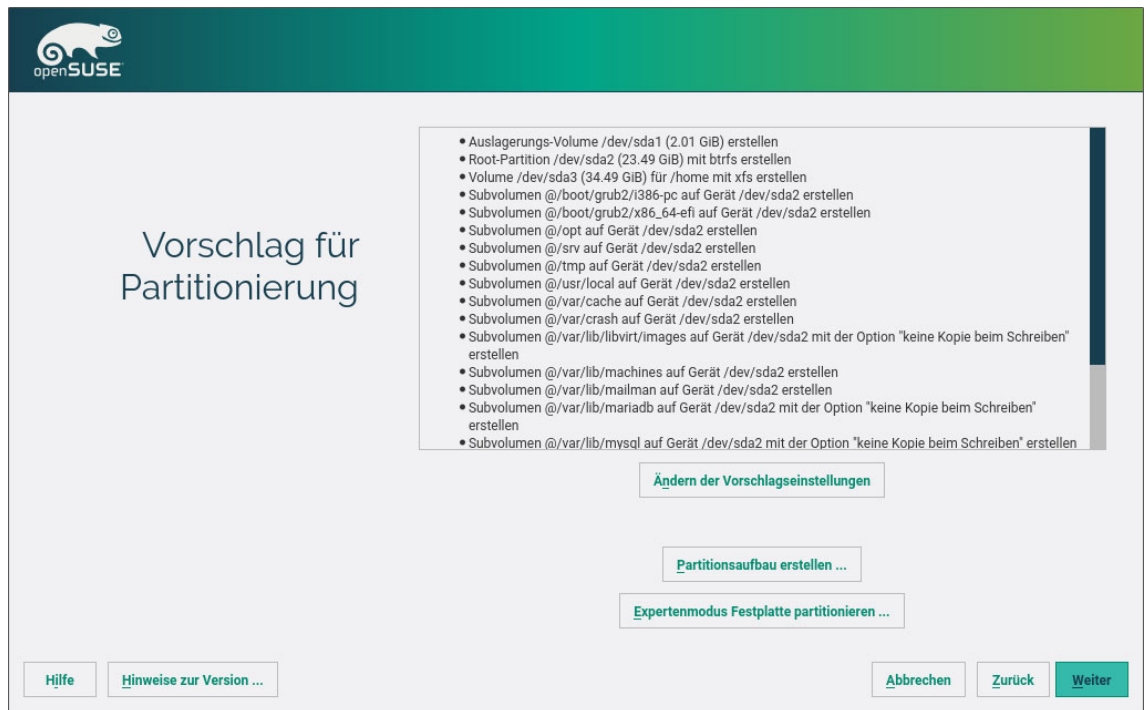
Im Startmenü des Installationsprogramms können Sie zudem zwischen einer Neuinstallation, dem Update einer vorhandenen SUSE-Version oder dem Start eines Rettungssystems wählen. Ich gehe hier nur auf die Neuinstallation ein.

**Installation  
starten**

Im ersten Dialog des Installationsprogramms stellen Sie die Sprache und Tastaturbelegung ein. Auf der nächsten Seite können Sie per Option zusätzliche Paketquellen hinzufügen. Dazu gibt es aber keinen Grund: Einfacher ist es, zuerst die Basisinstallation durchzuführen und weitere Paketquellen bei Bedarf später einzurichten.

Das Installationsprogramm macht nun einen Vorschlag für die Partitionierung der Festplatte (siehe [Abbildung 3.13](#)): Standardmäßig richtet das Programm eine Swap-Partition mit circa der eineinhalbfachen RAM-Größe ein, außerdem eine Root-Partition und bei ausreichend großen Datenträgern eine eigene /home-Partition. Das Installationsprogramm schlägt als Dateisystem für die Systempartition `btrfs` und für die Datenpartition `xfs` vor.

Partitionierung



**Abbildung 3.13** Partitionierungsvorschlag des Installationsprogramms

Wenn Sie mit dem Vorschlag einverstanden sind, klicken Sie einfach auf **WEITER**. Sehr praktisch ist der Button **ÄNDERN DER VORSCHLAGEEINSTELLUNGEN**: Wenn Sie ihn anklicken, gelangen Sie in einen Dialog, in dem Sie Eckparameter der automatischen Partitionierung verändern können. Hier gibt es die folgenden Optionen:

- ▶ **LVM-BASIERTEN VORSCHLAG ERSTELLEN**: Das Installationsprogramm soll automatisch LVM einrichten, wahlweise mit oder ohne Verschlüsselung.
- ▶ **DATEISYSTEM FÜR DIE ROOT-PARTITION**: Hier können Sie auswählen, ob SUSE `btrfs`, `ext4` oder `xfs` für die Root-Partition verwenden soll.
- ▶ **SEPARATE HOME-PARTITION**: Das Installationsprogramm soll eine von der Systempartition getrennte Home-Partition einrichten. Auch für diese Partition können Sie das Dateisystem frei wählen.

- ▶ **SWAP FÜR SUSPEND ERWEITERN:** Das Installationsprogramm soll die Swap-Partition großzügig dimensionieren, damit Sie die Suspend-Funktion nutzen können. In diesem Ruhezustandsmodus wird der gesamte Inhalt des RAMs in der Swap-Partition gespeichert. Anschließend wird das Notebook vollständig ausgeschaltet und verbraucht dann keinen Strom mehr.

#### Verwenden Sie ext4 für die Root-Partition!

Wenn Sie (noch) kein Linux-Experte sind, rate ich Ihnen dringend, für die Systempartition *nicht* das Dateisystem `btrfs` zu verwenden! Wählen Sie im Dialog VORSCHLAGS-EINSTELLUNGEN als Dateisystem `ext4`.

SUSE nutzt standardmäßig `btrfs` in Kombination mit der Snapper-Bibliothek so, dass administrative Arbeiten später rückgängig gemacht werden können. Das ist an sich eine feine Sache, es bringt aber eine große Komplexität mit sich, die selbst Linux-Profis zum Schwitzen bringt. Die Details können Sie in [Abschnitt 22.11](#), »Das `btrfs`-Dateisystem«, nachlesen. Einsteiger sind mit dem soliden `ext4`-Dateisystem definitiv besser beraten.

#### Automatische Partitionierung in Sonderfällen

Auch auf einem Rechner mit mehreren Festplatten/SSDs oder mit mehreren schon installierten Betriebssystemen müssen Sie auf die automatische Partitionierung nicht verzichten. In diesem Fall führt der Button PARTITIONSAUFBAU ERSTELLEN ZU besseren Ergebnissen. Das gibt Ihnen die Möglichkeit, vorweg auszuwählen, auf welchen Datenträgern neue Partitionen eingerichtet werden sollen und welche Partitionen vorweg gelöscht werden sollen.

#### Manuelle Partitionierung

Der Button EXPERTENMODUS FESTPLATTE PARTITIONIEREN ermöglicht es Ihnen, die Partitionierung selbst durchzuführen. Sie gelangen damit in den Partitionseditor (siehe [Abbildung 3.14](#)). Der Vorschlag der automatischen Partitionierung wird dabei übernommen. Sie haben nun die Möglichkeit, die vorgegebenen Einstellungen zu verändern oder ein vollkommen neues Setup einzurichten. Dazu legen Sie im freien Bereich der Festplatten oder SSDs neue Partitionen an. Bei vorhandenen Partitionen gelangen Sie mit einem Doppelklick in den Bearbeitungsmodus. Dort können Sie deren Größe ändern. Das ist vor allem dann wichtig, wenn Sie eine vorhandene Windows-Partition verkleinern wollen.

Sie können auch Partitionen löschen oder bereits vorhandene Partitionen nutzen. Dazu führen Sie das Kontextmenükommando BEARBEITEN aus und geben den gewünschten Mount-Punkt an, z. B. `/`. Wahlweise können Sie die Partition auch formatieren. Dabei gehen alle darin enthaltenen Daten verloren. YaST unterstützt die Linux-Dateisysteme `ext2`, `ext3`, `ext4`, `btrfs`, `reiserfs` und `xf`.

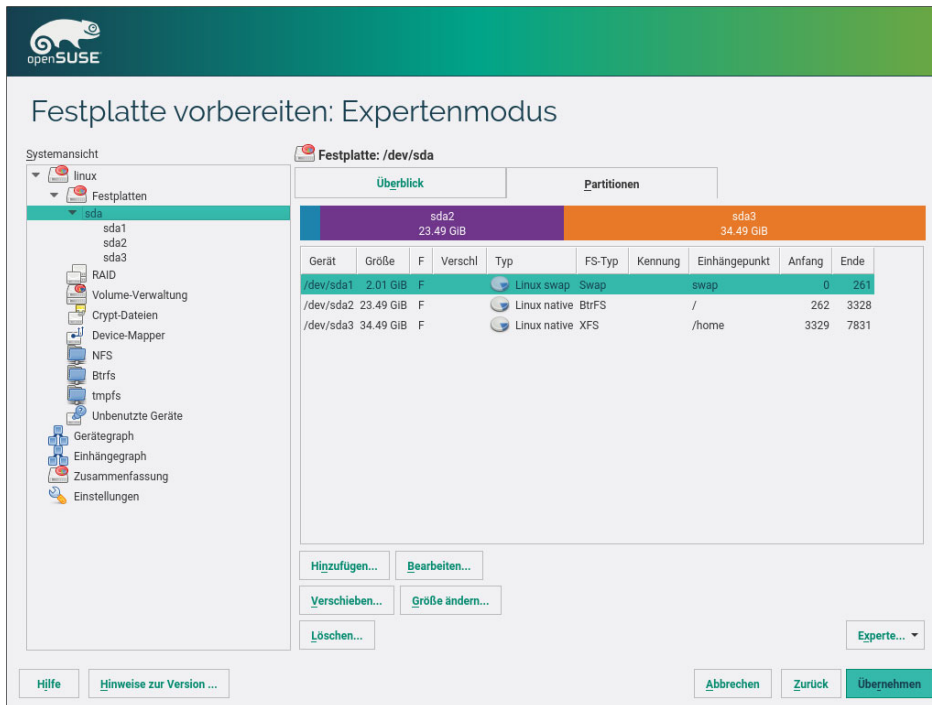



Abbildung 3.14 Partitionseditor

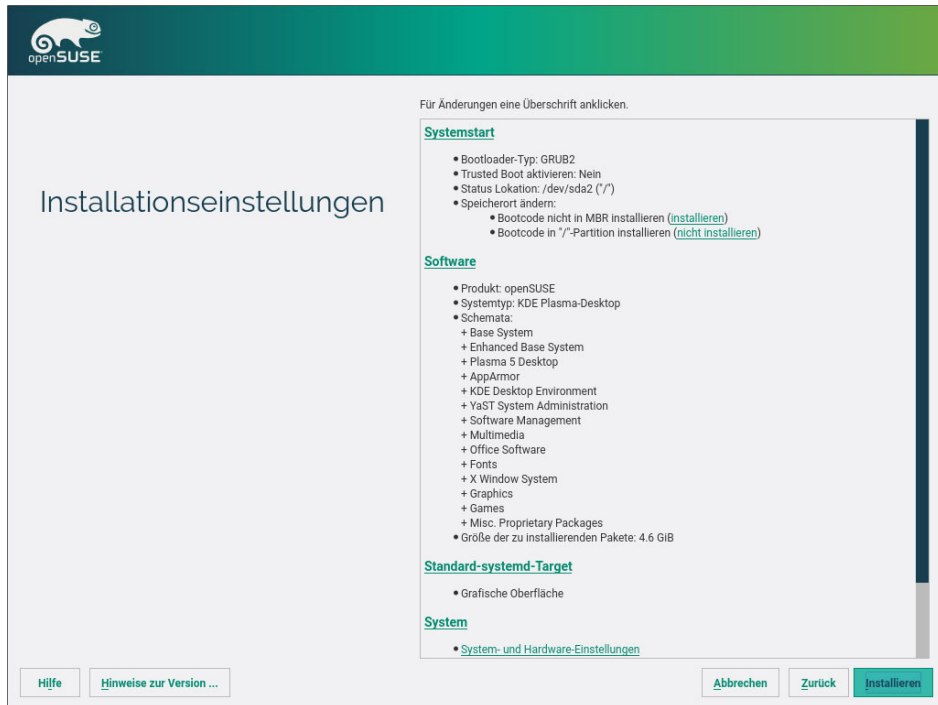
Für Linux-Profis bietet der Partitionseditor einige Besonderheiten: So ist es möglich, eine bereits vorhandene `/etc/fstab`-Datei zu nutzen oder ein LVM- bzw. ein RAID-System einzurichten. Diese Spezialfunktionen erfordern zwar LVM- bzw. RAID-Basiswissen, die Bedienung des Partitionseditors ist aber ungleich eleganter als bei CentOS/Fedora/RHEL.

Nach der Partitionierung stellen Sie die Zeitzone ein. Im nächsten Dialogblatt haben Sie die Wahl zwischen vier verschiedenen Desktop-Systemen: KDE (voreingestellt), Gnome, Xfce und gar kein Desktop (Server-Installation).

Grund-  
einstellungen

Nun folgen die Account-Einstellungen: Hier geben Sie Ihren vollständigen Namen, den gewünschten Login-Namen sowie zweimal das Passwort an. Etwas bedenklich ist der Umstand, dass das Benutzerpasswort standardmäßig auch für `root` gilt. Nur wenn Sie die diesbezügliche Option deaktivieren, haben Sie die Möglichkeit, im nächsten Dialog ein eigenes `root`-Passwort anzugeben. Die SUSE-Entwickler begründen ihre Vorgehensweise damit, dass ohnedies mehr als 75 Prozent aller Benutzer für `root` dasselbe Passwort verwenden wie für den ersten Benutzer-Account. Das mag sein, aber vom Sicherheitsstandpunkt aus betrachtet ist das natürlich nicht ganz optimal ...

**Zusammenfassung** Das Installationsprogramm zeigt nun eine Zusammenfassung aller Einstellungen an (siehe [Abbildung 3.15](#)). Wenn Sie damit einverstanden sind, klicken Sie einfach auf **INSTALLIEREN**, und los geht's. Sie sollten sich aber die Mühe machen, den Installationsvorschlag vorher in Ruhe durchzulesen! Oft ist es sinnvoll bzw. notwendig, Details zu verändern. Zur Änderung wählen Sie einfach den entsprechenden Punkt in der Zusammenfassung mit der Maus bzw. mit  aus.



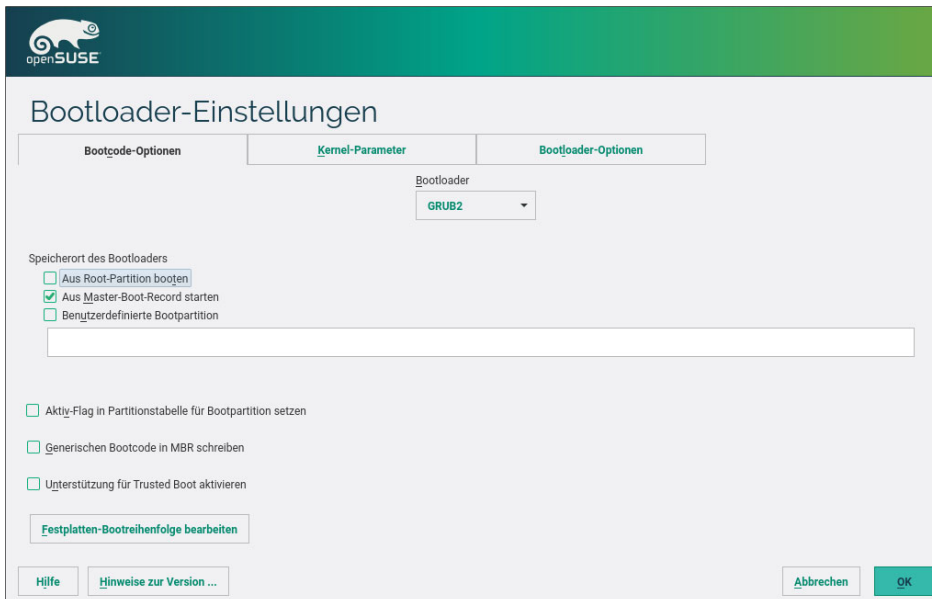
**Abbildung 3.15** Zusammenfassung der Installationseinstellungen

**Systemstart bei BIOS-Rechnern** Die Details des Systemstarts hängen davon ab, ob openSUSE im BIOS- oder im EFI-Modus installiert wurde. Im Gegensatz zu den meisten anderen Distributionen installiert SUSE bei BIOS-Rechnern GRUB normalerweise *nicht* in den Master Boot Record der ersten Festplatte. Wenn es eine erweiterte Partition gibt, schlägt YaST vor, GRUB in deren Boot-Sektor zu installieren, andernfalls in den Boot-Sektor der System- oder Boot-Partition. Die GRUB-Entwickler raten allerdings explizit von der Installation in einzelne Partitionen ab und empfehlen die Installation in den Boot-Sektor der Festplatte.

YaST markiert außerdem die Partition, in deren Boot-Sektor GRUB installiert wird, als *aktive* Partition – also als die Partition, von der gebootet werden soll. Schließlich überschreibt YaST den MBR mit einem Code, um den Bootloader der aktiven Partition

zu laden. Diese relativ komplizierte Vorgehensweise soll ein möglichst konfliktfreies Zusammenspiel mit Windows garantieren.

Beachten Sie, dass openSUSE in der Standardkonfiguration eine bereits vorhandene GRUB-Installation einer anderen Linux-Distribution überschreibt. Wenn Sie das nicht möchten, klicken Sie im Zusammenfassungsdialog auf den Punkt SYSTEMSTART. Damit gelangen Sie in das Dialogblatt BOOTLOADER-EINSTELLUNGEN. Dort deaktivieren Sie die Option GENERISCHEN BOOTCODE IN MBR SCHREIBEN (siehe [Abbildung 3.16](#)).



**Abbildung 3.16** Bei der Installation auf BIOS-Rechnern gibt es mehrere Bootloader-Optionen.

Wenn Sie GRUB – wie von den GRUB-Entwicklern vorgesehen – in den Startsektor der Festplatte bzw. SSD installieren möchten, aktivieren Sie im Dialog BOOTLOADER-EINSTELLUNGEN die Option AUS MASTER-BOOT-RECORD STARTEN und deaktivieren alle anderen Optionen (siehe nochmals [Abbildung 3.16](#)).

### Vorsicht bei der Bootloader-Konfiguration!

Ich kenne keine andere Distribution, bei der die Bootloader-Konfiguration so kompliziert und unübersichtlich ist wie bei openSUSE! Nur wenn auf Ihrem Rechner keine anderen Betriebssysteme installiert sind, können Sie die zahllosen Optionen bedenkenlos so lassen, wie sie sind. Ansonsten sollten Sie unbedingt einen genauen Blick auf sämtliche Bootloader-Optionen werfen.

**Systemstart im EFI-Modus** Bei Rechnern mit EFI wird GRUB in das Verzeichnis `/boot/efi/EFI/opensuse` installiert. In diesem Fall sind keine Änderungen an der Konfiguration sinnvoll.

**Software-Auswahl** Mit einem Klick auf den Eintrag SOFTWARE in der Zusammenfassung der Installationseinstellungen können Sie ganze Software-Gruppen oder auch nur einzelne Pakete hinzufügen: Server-Komponenten, Entwicklerwerkzeuge etc. YaST zeigt dabei an, wie viel Platz die Installation auf Ihrer Festplatte ungefähr beanspruchen wird. Wenn Sie ein bestimmtes Programm nicht finden, klicken Sie auf DETAILS und verwenden dann die Suchfunktion (FILTER SUCHE).

**Firewall und SSH** Standardmäßig aktiviert das Installationsprogramm eine Firewall. Ein SSH-Server wird zwar installiert, aber er wird zum einen durch die Firewall blockiert und zum anderen nicht gestartet. Wenn Sie den SSH-Server nutzen möchten, müssen Sie den SSH-Port öffnen und den SSH-Dienst aktivieren. Dazu klicken Sie die entsprechenden Links am unteren Ende der Zusammenfassung an (siehe [Abbildung 3.17](#)).



**Abbildung 3.17** Standardmäßig wird der SSH-Dienst durch die Firewall blockiert und nicht gestartet.

**Installation durchführen** Sobald Sie mit allen Einstellungen einverstanden sind, klicken Sie den Button INSTALLIEREN an. Die Installation dauert einige Minuten. Sie können sich während dieser Wartezeit eine Diashow ansehen oder die Release-Notes lesen. Anschließend wird der Rechner neu gestartet.

### Erste Schritte

**Netzwerk-konfiguration und Hostname** openSUSE fährt bei der Netzwerkkonfiguration zweigleisig: Auf Notebooks kommt der NetworkManager zum Einsatz, während auf Desktop-PCs und Servern standardmäßig die SUSE-Eigenentwicklung wicked läuft. Dieser Dämon ist speziell für Server und virtuelle Maschinen optimiert. Um zwischen diesen beiden Konfigurationsvarianten umzuschalten, öffnen Sie im Konfigurationsprogramm YaST das Modul SYSTEM • NETZWERKEINSTELLUNGEN. Das Konfigurationsverfahren wird im Dialogblatt GLOBALE OPTIONEN eingestellt.



Während der Installation besteht keine Möglichkeit, den Hostnamen einzustellen. openSUSE verwendet stattdessen eine Zufallszeichenkette wie `linux-q2uf`. Abhilfe: Wechseln Sie im gerade erwähnten YaST-Modul zur Netzwerkeinstellung in das Dialogblatt `HOSTNAME/DNS`, und geben Sie dort den gewünschten Hostnamen an. Damit alle KDE- und Gnome-Programme diese Änderung nachvollziehen, müssen Sie sich ab- und neu anmelden.

Unter openSUSE ist standardmäßig eine Firewall installiert und auch aktiv. Damit wird nahezu der gesamte nach außen gehende Netzwerkverkehr blockiert. Unter anderem ist es deswegen unmöglich, auf Windows- oder Samba-Netzwerkfreigaben zuzugreifen. Die Konfiguration erfolgt durch das YaST-Modul `SICHERHEIT • FIREWALL` (siehe auch [Abschnitt 37.4](#), »Firewall-Konfigurationshilfen«). Wenn sich Ihr Rechner in einem sicheren lokalen Netz befindet, sollten Sie die Netzwerkschnittstelle der `INTERNEN ZONE` zuordnen. Andernfalls schalten Sie im Dialogblatt `ERLAUBTE DIENSTE` die Ports frei, die Sie nutzen möchten – z. B. um den Datenaustausch mit anderen Windows-Rechnern zu ermöglichen.

**Firewall  
einstellen**

Das erste Update führen Sie wahlweise mit dem von KDE oder Gnome vorgeschlagenen Update-Tool oder in einem Terminal mit dem Kommando `zypper update` durch.

**Erstes Update**

Nach der Installation von einer DVD dient diese als Paketquelle für die Installation weiterer Pakete. Deswegen fragt YaST in Zukunft bei jeder Paketinstallation nach dieser DVD. Das ist nicht sinnvoll, weil im Internet normalerweise aktuellere Pakete bereitstehen. Deswegen sollten Sie die DVD als Paketquelle deaktivieren. Dazu starten Sie das YaST-Modul `SOFTWARE-REPOSITORIES` und löschen die Paketquelle `URL cd:///<name>`.

**CD/DVD als  
Paketquelle  
deaktivieren**

openSUSE liefert zwar von Haus aus diverse Audio- und Video-Player mit, dennoch ist die Multimedia-Unterstützung relativ schlecht: Viele Audio- und Video-Formate können nicht abgespielt werden. Ebenso wenig können kopiergeschützte DVDs angesehen werden. Das hat nicht technische, sondern rechtliche Gründe: Diverse Codecs sind in manchen Ländern durch Patente geschützt. Deswegen ist die Weitergabe von Open-Source-Implementierungen nicht überall zulässig. Und da openSUSE international verbreitet wird, gilt eben der kleinste gemeinsame Nenner.

**Multimedia**

Die Lösung dieses Problems ist die Packman-Paketquelle. Um diese zu aktivieren, starten Sie nochmals das YaST-Modul `SOFTWARE-REPOSITORIES`, führen dort `HINZUFÜGEN • COMMUNITY/GEMEINSCHAFTS-REPOSITORIES` aus und aktivieren die Packman-Paketquelle. Dabei müssen Sie den Schlüssel der Packman-Paketquelle als vertrauenswürdig bestätigen.

**Packman-  
Paketquelle  
einrichten**

Danach starten Sie das YaST-Modul SOFTWARE INSTALLIEREN UND LÖSCHEN, wählen mit ANZEIGEN • INSTALLATIONSQUELLEN die Packman-Paketquelle aus und klicken dann auf den Link WECHSEL VON SYSTEMPAKETEN. Alternativ können Sie auch in einem Terminalfenster das Kommando `zypper dup` ausführen. In jedem Fall werden nun eine Menge openSUSE-Pakete durch äquivalente Packman-Pakete ersetzt. Außerdem stehen Ihnen nun diverse Zusatzpakete zur Verfügung, unter anderem die Codec-Pakete `gststreamer-*-plugins-bad`, `-ugly` und `-ffmpeg`.

#### NVIDIA-Grafiktreiber

Wenn Ihr Rechner eine NVIDIA-Grafikkarte enthält, kommt standardmäßig der `nouveau`-Treiber zum Einsatz. Der Treiber funktioniert mit nahezu allen Modellen gut, kann aber die Energiesparfunktionen der Grafikkarte nicht optimal nutzen.

Zum Glück ist die Installation des proprietären NVIDIA-Treibers ganz einfach: Dazu aktivieren Sie mit YaST die NVIDIA-Paketquelle. Anschließend starten Sie das YaST-Modul ONLINE-AKTUALISIERUNG und führen dann das Menükommando EXTRAS • ALLE PASSENDE EMPFOHLENE PAKETE INSTALLIEREN aus. YaST stellt nun fest, welche GPU Ihre NVIDIA-Grafikkarte enthält, und installiert das passende Treiberpaket. (Aktuell gibt es drei verschiedene derartige Pakete, und es ist nicht ganz einfach zu erraten, welches Paket zu welcher Grafikkarte passt.)

Falls Probleme beim Einrichten des NVIDIA-Treibers auftreten sollten oder wenn Sie die aktuellste Version des Treibers manuell installieren möchten, finden Sie hier weitere Informationen bzw. ein Installationsprogramm für den NVIDIA-Treiber:

[https://en.opensuse.org/SDB:NVIDIA\\_drivers](https://en.opensuse.org/SDB:NVIDIA_drivers)

## 3.6 Ubuntu

Ubuntu ist momentan die populärste und im Privatbereich am weitesten verbreitete Distribution. Das Motto von Ubuntu lautet *Linux for human beings* – also gewissermaßen »das menschliche Linux«. Das Zulu-Wort *ubuntu* steht denn auch für *Menschlichkeit gegenüber anderen* oder *achtsames Miteinander* oder auch für *I am what I am because of who we all are*. Ubuntu soll also nicht nur eine Menge Software-Technik sein, sondern eine ganze Philosophie.

#### Canonical

Hinter Ubuntu steht die Firma Canonical Ltd. des südafrikanischen Millionärs Mark Shuttleworth – ehemals Eigentümer von Thawte Consulting. Im Vergleich zu Red Hat hat Canonical aber wesentlich weniger Mitarbeiter. Zuletzt tanzte Canonical sprichwörtlich auf allen Hochzeiten und versuchte neben dem klassischen Ubuntu für PCs auch Ubuntu-Versionen für Smartphones, Tablets, TV-Geräte und die Cloud fertigzustellen. Damit ist seit Frühjahr 2017 Schluss: Canonical hat beschlossen, sich auf

Cloud- und Server-Kunden zu konzentrieren; dort war Canonical zuletzt sehr erfolgreich, und dort lässt sich auch Geld verdienen.

Diese strategische Neuausrichtung große Auswirkungen auch auf Desktop-Anwender: Canonical hat beschlossen, die Desktop-Eigenentwicklung Unity nicht mehr weiterzuführen und den Desktop beginnend mit Ubuntu 17.10 stattdessen auf Gnome umzustellen. Für die Linux-Gemeinde als Ganzes ist dies sicherlich ein Fortschritt: Statt zwei konkurrierende Desktop-Systeme voranzubringen, konzentrieren sich die vorhandenen Entwicklerressourcen im Linux-Enterprise-Segment (Red Hat, SUSE, Canonical) nun auf *einen* Desktop. Ob der Gnome-Umstieg auch für die Anwender ein Vorteil ist, wird sich zeigen. Unmittelbar werden sicherlich einige Unity-spezifische Besonderheiten verloren gehen. Möglicherweise wird es aber in Zukunft eine von der Community getragene Unity-Variante von Ubuntu geben.

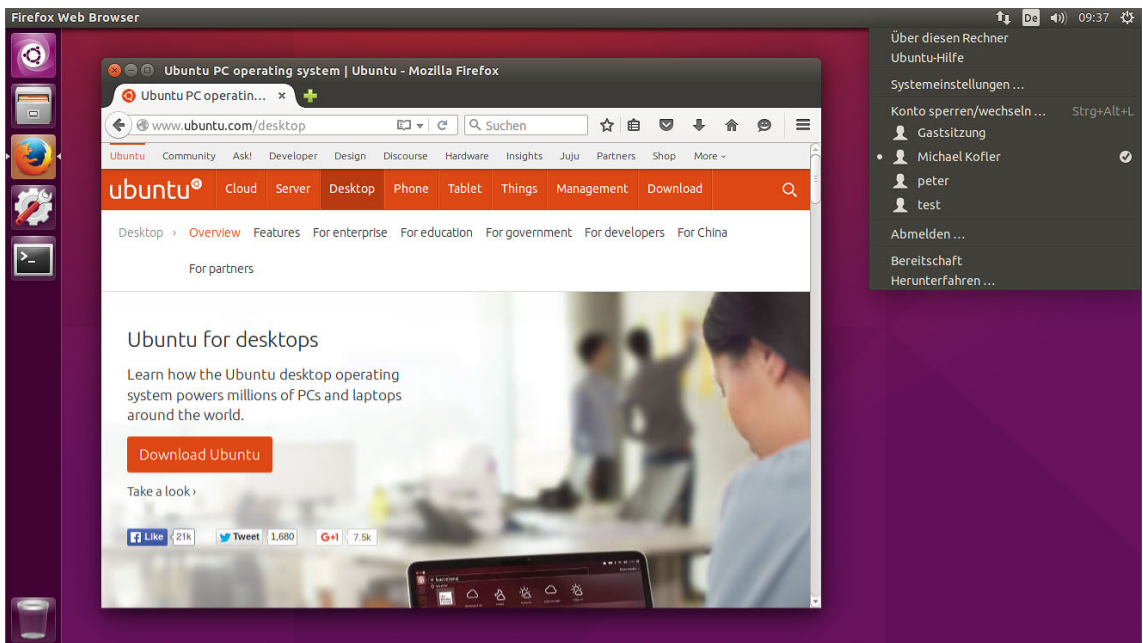


Abbildung 3.18 Der Ubuntu-Desktop, hier noch in der Unity-Variante von Ubuntu 17.04

Es gibt halbjährlich neue Ubuntu-Versionen, deren Versionsnummer das Datum der Fertigstellung widerspiegelt. Ubuntu 17.04 meint also die im April 2017 fertiggestellte Ubuntu-Version. Außerdem hat jede Ubuntu-Version einen merkwürdigen Codenamen, für Version 17.04 lautet er z. B. *Zesty Zapus*. Diese Codenamen sind perfekt für Suchanfragen geeignet! Eine Suche nach *zesty nvidia* wird wesentlich spezifischere Ergebnisse liefern als eine Suche nach *ubuntu nvidia* oder gar nach *linux nvidia*.

**Standard-  
versionen**

Für die halbjährlichen Ubuntu-Versionen gibt es nur einen neunmonatigen Update-Service. Damit sind diese Versionen eigentlich nur noch Linux-Profis zu empfehlen, die sich nicht an regelmäßigen Distributions-Updates stören.

**LTS-Versionen** Eine Sonderstellung innerhalb der vielen Ubuntu-Versionen nehmen die sogenannten LTS-Versionen ein (*Long Term Support*), deren Desktop-Pakete und Server-Pakete fünf Jahre lang gewartet werden. Als ich dieses Buch fertiggestellt habe, war 16.04 die letzte verfügbare LTS-Version. Erst im April 2018 wird es mit Ubuntu 18.04 die nächste LTS-Version geben. Ob für die Desktop-Variante von Ubuntu auch dann noch die fünfjährige Update-Garantie gelten wird, bleibt abzuwarten.

**Varianten** Es gibt zahlreiche von Ubuntu abgeleitete Distributionen. [Tabelle 3.2](#) fasst die wichtigsten offiziellen, also von Canonical unterstützten, sowie einige inoffizielle Varianten zusammen.

Alle offiziellen Varianten greifen auf dieselben Paketquellen zurück und lassen sich daher beliebig erweitern. Sie können auch zuerst Xubuntu installieren und später die Gnome-Pakete von Ubuntu hinzufügen. Der Hauptunterschied zwischen den verschiedenen Ubuntu-Varianten besteht darin, welche Paketauswahl auf dem Datenträger mitgeliefert und erstmalig installiert wird.

| Variante      | Beschreibung  |
|---------------|---|
| Kubuntu       | Ubuntu mit KDE  |
| Ubuntu Gnome  | Ubuntu mit Gnome Shell statt Unity  |
| Ubuntu MATE   | Ubuntu mit dem MATE-Desktop   |
| Xubuntu       | Ubuntu mit Xfce   |
| Lubuntu       | Ubuntu mit LXDE   |
| Ubuntu Server | Ubuntu für den Server-Einsatz (ohne Grafiksystem)   |
| Ubuntu Studio | Ubuntu für Multimedia-Anwender  |
| Mythbuntu     | Ubuntu als Media-Center mit Myth TV   |
| Linux Mint    | populäre Ubuntu-Variante ohne Unity (inoffiziell)   |
| Elementary OS | Ubuntu-Variante mit macOS-ähnlichem Desktop (inoffiziell)                                   |
| Ubuntu Budgie | Ubuntu mit Budgie-Desktop   |
| Zentyal       | kommerzielle Ubuntu-Server-Variante mit webbasierten Konfigurationswerkzeugen (inoffiziell) |

**Tabelle 3.2** Ubuntu-Varianten

Ubuntu ist zwar sehr populär, hat aber wie jede andere Distribution Schwächen:

Nachteile/Kritik

- ▶ Canonical geht mit Ubuntu viele eigene Wege und unterscheidet sich mittlerweile stark vom Linux-Mainstream. Das hat in der Vergangenheit oft zu Irritationen geführt. Gleichzeitig war das relativ kleine Canonical-Entwicklerteam mit der Wartung der eigenen Software überfordert. Aus diesem Grund sind etliche Ubuntu-Projekte gescheitert (Upstart, Unity, Mir etc.).
- ▶ Einige Open-Source-Entwickler kritisieren, dass sich Ubuntu zwar großzügig im Open-Source-Supermarkt bedient, aber vergleichsweise wenig eigenen Code zurückgibt.
- ▶ Das pünktliche, halbjährliche Erscheinen neuer Ubuntu-Versionen geht mitunter auf Kosten der Stabilität.
- ▶ Standardmäßig richtet Ubuntu keine Firewall ein.

## Ubuntu installieren

Auf der Website <https://www.ubuntu.com/download> können Sie kostenlos Ubuntu-Installationsmedien herunterladen. Dabei haben Sie die Wahl zwischen einer 32- und einer 64-Bit-Variante. Die ISO-Images müssen nun auf eine DVD gebrannt oder auf einen USB-Stick übertragen werden.

Installationsmedien

### Alternative Ubuntu-Downloads

Auf der Website <http://cdimage.ubuntu.com> finden Sie diverse alternative Installationsmedien. Dazu zählen Ubuntu-Varianten wie Kubuntu, Xubuntu oder Ubuntu Server sowie experimentelle Versionen für andere CPU-Plattformen. Auf dieser Website finden Sie auch die sogenannten Daily-Images der gerade in Entwicklung befindlichen nächsten Ubuntu-Version sowie ein kleines Netinstall-Image, das nur das Installationsprogramm enthält und die zu installierenden Pakete erst bei Bedarf aus dem Internet herunterlädt.

Das Netinstall-Image und Ubuntu Server unterscheiden sich insofern von den anderen Varianten, als ein textbasiertes Installationsprogramm zum Einsatz kommt. Eine ausführliche Beschreibung des Installationsverlaufs folgt in [Abschnitt 3.7](#), »Ubuntu Server«.

Um eine Ubuntu-Standardinstallation durchzuführen, starten Sie den Rechner mit der Ubuntu-DVD oder einem entsprechenden USB-Stick neu. Damit gelangen Sie in ein Live-System, in dem Sie Ubuntu sowohl ausprobieren als auch installieren können.

Standardinstallation starten

#### Grundeinstellungen

Nach dem Start des Installationsprogramms wählen Sie im ersten Schritt die gewünschte Sprache aus. Anschließend stellen Sie durch die Option **WÄHREND UBUNTU INSTALLIERT WIRD AKTUALISIERUNGEN HERUNTERLADEN** (siehe [Abbildung 3.19](#)) ein, ob während der Installation Updates heruntergeladen werden sollen. Das soll sicherstellen, dass Sie vom ersten Start an ein aktuelles System haben. Bei meinen Tests hat diese Option aber nur unzuverlässig funktioniert.



Abbildung 3.19 Grundeinstellungen im Ubuntu-Installationsprogramm

Eine weitere Option steuert, ob auch Pakete von Drittanbietern installiert werden sollen. Das betrifft z. B. das Adobe-Flash-Plugin sowie grundlegende Audio- und Video-Codecs. Diese Programme sind zwar kostenlos verfügbar, unterliegen aber nicht alle einer Open-Source-Lizenz. Ich verzichte normalerweise auf beide Optionen, da sie die Installationsdauer verlängern. Sowohl das Update als auch die Installation von Zusatzsoftware können Sie auch später vornehmen.



Abbildung 3.20 Installationsart einstellen

Je nachdem, wie die Festplatte momentan partitioniert ist, können Sie zwischen mehreren Optionen wählen (siehe [Abbildung 3.20](#)).

Welche der folgenden Optionen zur Auswahl stehen, hängt davon ab, welche anderen Betriebssysteme schon auf der Festplatte/SSD installiert sind:

- ▶ **UBUNTU NEBEN <BETRIEBSSYSTEM> INSTALLIEREN:** Bei dieser Variante können Sie im nächsten Dialog eine vorhandene Partition von Windows oder einer bereits installierten Linux-Distribution verkleinern und den freien Platz für die Ubuntu-Installation nutzen. Mit dem Schieberegler zwischen den beiden Bereichen bestimmen Sie, wie viel Platz für die bisherige Partition und wie viel für Linux reserviert werden soll.

Die Verkleinerung von Windows-Partitionen ist nur möglich, wenn Windows vorher mit `shutdown /p` vollständig heruntergefahren wurde. Die Verkleinerung kann einige Minuten dauern. Haben Sie Geduld! (Besser ist es generell, die Verkleinerung des Windows-Dateisystems unter Windows durchzuführen, nicht im Installationsprogramm egal welcher Linux-Distribution.)

- ▶ **<BETRIEBSSYSTEM> MIT UBUNTU ERSETZEN:** Das Installationsprogramm löscht Ihr vorhandenes Windows- oder Linux-System und nutzt anschließend die gesamte Festplatte zur Installation von Ubuntu.
- ▶ **UBUNTU AUF DIE VERSION NNN AKTUALISIEREN:** Das Installationsprogramm aktualisiert eine bereits vorhandene Ubuntu-Installation. Dabei bleiben das `/home`-Verzeichnis und einige Grundeinstellungen des Systems erhalten. Allerdings funktioniert dieses Update nicht immer problemlos. Führen Sie nach Möglichkeit eine Neuinstallation durch.
- ▶ **FESTPLATTE LÖSCHEN UND UBUNTU INSTALLIEREN:** Damit wird die gesamte Festplatte bzw. SSD gelöscht und anschließend neu partitioniert. Sie verlieren alle Daten auf der Festplatte. Diese Option wird unter anderem dann angezeigt, wenn die Festplatte noch leer ist und keine Partitionstabelle enthält.
- ▶ **UBUNTU LÖSCHEN UND NEU INSTALLIEREN:** Damit wird eine vorhandene Ubuntu-Installation gelöscht. Auf dem so frei gewordenen Platz der Festplatte wird Ubuntu neu installiert.
- ▶ **ETWAS ANDERES:** Hiermit führen Sie die Partitionierung selbst durch.

Sofern Sie sich nicht für ETWAS ANDERES entscheiden, richtet das Installationsprogramm seit Version 17.04 einfach nur eine Systempartition ein. Auf eine eigene Boot-Partition wird ebenso verzichtet wie auf eine Swap-Partition. Stattdessen legt das Installationsprogramm eine Auslagerungsdatei an.

#### Automatische Partitionierung mit LVM

Sofern Sie sich für die Variante **FESTPLATTE LÖSCHEN UND UBUNTU INSTALLIEREN** entscheiden, stehen Ihnen zwei weitere Optionen zur Auswahl:

- ▶ **DIE NEUE UBUNTU-INSTALLATION ZUR SICHERHEIT VERSCHLÜSSELN**
- ▶ **LVM BEI DER NEUEN UBUNTU-INSTALLATION VERWENDEN**

Beide Optionen führen dazu, dass ein LVM-System eingerichtet wird, einmal mit Verschlüsselung und einmal ohne. Das Installationsprogramm erstellt dann eine kleine Boot-Partition und richtet darin ein ext2-Dateisystem ein. Im verbleibenden Speicherbereich wird eine große Partition eingerichtet, die als Physical Volume für eine Volume Group mit dem Namen *ubuntu* dient. Innerhalb dieser Volume Group sieht das Installationsprogramm zwei Logical Volumes vor, die die Swap-Partition und die Systempartition aufnehmen.

Diese »LVM-Standardinstallation« funktioniert prinzipiell gut, bietet aber keinerlei Konfigurationsmöglichkeiten. Alle Versuche, mit der Option **ETWAS ANDERES** ein von einer früheren Installation bereits vorhandenes LVM-System zu modifizieren oder LVM manuell einzurichten, scheitern kläglich.

Das Ubuntu-Installationsprogramm kann in dieser Hinsicht nicht mit Fedora oder openSUSE mithalten. Wenn Sie eine individuelle LVM-Konfiguration wünschen, müssen Sie das textbasierte Installationsprogramm von Ubuntu Server oder des Net-install-Images verwenden.

#### Manuelle Partitionierung

Wenn Sie die Größe der Partitionen selbst einstellen möchten, eine eigene */home*-Partition wünschen etc., wählen Sie **ETWAS ANDERES**. Um eine neue Partition zu erzeugen, klicken Sie zuerst den Eintrag **FREIER SPEICHERPLATZ** und dann den Button **HINZUFÜGEN** an. Im nun erscheinenden Dialog geben Sie den Typ der Partition, die Größe in MiB und das Dateisystem an. Falls es auf Ihrer Festplatte bereits eine geeignete Partition gibt, in die Sie Ubuntu installieren möchten, wählen Sie diese Partition aus, ändern mit **PARTITION BEARBEITEN** den **EINHÄNGEPUNKT** und aktivieren das Auswahlhäkchen zur Neuformatierung der Partition (siehe [Abbildung 3.21](#)).

Wenn auf einem Rechner mit EFI bereits Betriebssysteme installiert sind, müssen Sie die EFI-Partition als solche markieren. Dazu klicken Sie zuerst die schon existierende EFI-Partition und dann den Button **ÄNDERN** an und stellen dann **BENUTZEN ALS = EFI-BOOT-PARTITION** ein. Sollte es bei einem fabrikneuen EFI-Rechner noch gar keine EFI-Partition geben, müssen Sie diese einrichten.

Sie beenden die Partitionierung mit dem Button **JETZT INSTALLIEREN**. Das Installationsprogramm beginnt jetzt sofort mit der Installation. Vorsicht: Sie können die Installation nun nicht mehr stoppen!



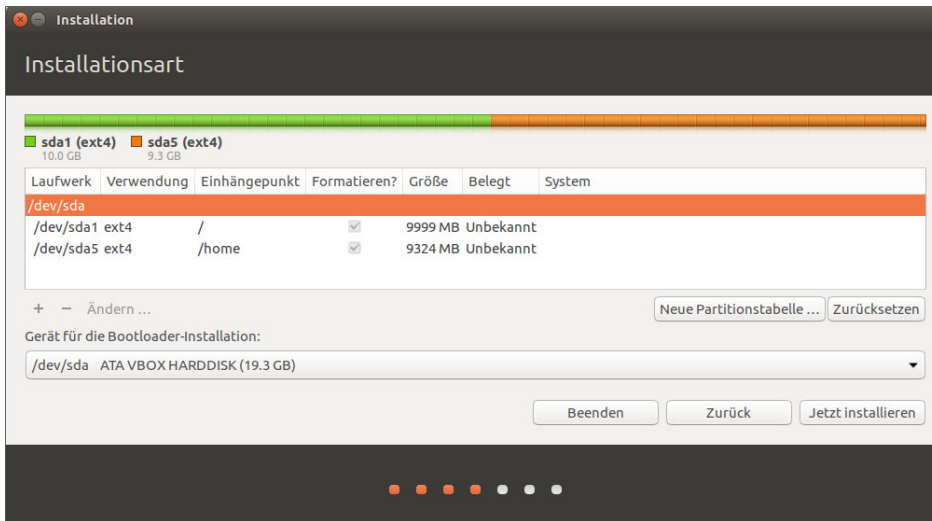




Abbildung 3.21 Manuelle Partitionierung

Der Bootloader GRUB wird im Verlauf der Installation ohne weitere Rückfrage in den Boot-Sektor (MBR) der ersten Festplatte installiert. Wenn Sie einen anderen Ort wünschen, müssen Sie die Installationsvariante ETWAS ANDERES wählen und können die gewünschte Partition dann durch ein Listenfeld am unteren Rand des Partitionierungsdialogs einstellen.

Diverse noch ausstehende Einstellungen können Sie parallel zur Installation vornehmen. Diese neue Abfolge der Installationsschritte spart Zeit. Als Erstes geben Sie die Zeitzone an, in der Sie sich befinden. Das Installationsprogramm nimmt an, dass die Uhr Ihres Rechners auf die lokale Uhrzeit eingestellt ist. Wenn das nicht der Fall ist, müssen Sie die Uhrzeit nach dem Ende der Installation korrigieren.

Zeitzone

Damit sich die Tastatur so verhält, wie Sie es von Windows gewohnt sind, müssen Sie das Layout DEUTSCH – NUR GRAVE- UND ACUTE-AKZENTTASTE wählen. Das bedeutet, dass das Zeichen ~ direkt eingegeben werden kann, die Zeichen ` und ´ aber zur Komposition von Buchstaben aus Fremdsprachen dienen. ,  ergibt daher é.

Tastatur

Im nächsten Schritt geben Sie den Benutzernamen und das Passwort für den ersten Ubuntu-Nutzer an (siehe [Abbildung 3.22](#)). Weitere Nutzer können Sie bei Bedarf später im laufenden Betrieb einrichten. Im Gegensatz zu anderen Linux-Installationsprogrammen müssen Sie kein root-Passwort angeben: Administrative Arbeiten werden unter Ubuntu von einem gewöhnlichen Benutzer mit `sudo` durchgeführt.

Benutzer einrichten

Abbildung 3.22 Benutzer einrichten

Sie haben an dieser Stelle die Wahl zwischen drei Sicherheitsoptionen: AUTOMATISCH ANMELDEN bewirkt, dass Sie beim Rechnerstart automatisch eingeloggt werden. Das ist bequem, aber natürlich ein Sicherheitsmangel. PASSWORT ZUM ANMELDEN ABFRAGEN ist die Standardeinstellung und erfordert nach dem Rechnerstart einen gewöhnlichen Login.

### Mehr Sicherheit durch Verschlüsselung?

Noch mehr Sicherheit verspricht die Option MEINE PERSÖNLICHEN DATEIEN VERSCHLÜSSELN. Damit wird Ihr gesamtes persönliches Verzeichnis verschlüsselt. Ein Zugriff auf die Daten ist nur nach einem Login möglich.

Dieses Ubuntu-spezifische Verschlüsselungsverfahren ist aber auch mit Nachteilen verbunden! Insbesondere ist es sehr schwierig, die eigenen Daten zu retten, wenn sich das System aus irgendeinem Grund nicht mehr starten lässt. Ich rate Ihnen deswegen entschieden von dieser Option ab!

Verwenden Sie gegebenenfalls lieber die Option DIE NEUE UBUNTU-INSTALLATION ZUR SICHERHEIT VERSCHLÜSSELN im Dialog INSTALLATIONSART (siehe [Abbildung 3.20](#)): Dann wird das gesamte LVM-System auf die gleiche Art und Weise wie bei den meisten anderen Distributionen verschlüsselt. Das vereinfacht die Wartung und verbessert die Chancen, dass sich Ihre Daten bei Problemen retten lassen.

## Installation von Kubuntu, Xubuntu, Lubuntu & Co.

Der Installationsprozess für die meisten Ubuntu-Derivate erfolgt auf die gleiche Weise wie vorhin beschrieben. Die Dialoge des Installationsprogramms sind in der Regel optisch ein wenig anders gestaltet, bieten aber dieselben Optionen. Das gilt auch für einige »inoffizielle« Ubuntu-Varianten, z. B. für Elementary OS und Linux Mint.

### USB-Stick-Installation

Das Programm STARTMEDIENERSTELLER (`usb-creator-gtk`), das Sie im Menü des Ubuntu-Live-Systems finden, überträgt die Installations-DVD oder eine entsprechende ISO-Datei auf einen USB-Stick. Dieser muss mindestens 2 GiB groß sein. Sein gesamter bisheriger Inhalt geht verloren.

Im Vergleich zu einer richtigen Installation ist das System auf dem USB-Stick nur bedingt modifizierbar und läuft langsamer. Es eignet sich als »Ubuntu zum Mitnehmen« oder zur Durchführung einer richtigen Ubuntu-Installation auf einem Computer ohne CD/DVD-Laufwerk, aber nicht für den Dauerbetrieb.

### Erste Schritte

Um alle installierten Ubuntu-Pakete auf den aktuellsten Stand zu bringen, führen Sie in einem Terminal-Fenster das folgende Kommando aus:

```
user$ sudo apt update
user$ sudo apt full-upgrade
```

Updates  
durchführen

Das Paket `ubuntu-restricted-extras` macht Ihr Ubuntu-System multimediatauglich. Installiert werden unter anderem das Adobe-Flash-Plugin, Codecs für alle möglichen Audio- und Video-Formate inklusive MP3, die kostenlosen Microsoft-Web-Fonts etc.

ubuntu-  
restricted-extras

```
user$ sudo apt install ubuntu-restricted-extras
```

Ubuntu stellt für NVIDIA-Grafikkarten sowie für einige WLAN-Adapter proprietäre Hardware-Treiber zur Verfügung. Um diese zu installieren, öffnen Sie im Startmenü das Programm SOFTWARE & AKTUALISIERUNG. Die für Ihr System passenden Treiber werden im Dialogblatt ZUSÄTZLICHE TREIBER aufgelistet (siehe [Abbildung 19.7](#) in [Kapitel 19](#), »Software- und Paketverwaltung«). Zur Aktivierung müssen Sie den Rechner neu starten.

Hardware-Treiber  
installieren

Alternativ kann die Installation derartiger Treiber auch im Textmodus erfolgen, was besonders dann sehr praktisch ist, wenn das Grafiksystem mangels Treiber nicht funktioniert. In diesem Fall ermitteln Sie zuerst mit `ubuntu-drivers devices` die Liste der zur Auswahl stehenden Treiber und installieren den empfohlenen Treiber dann mit `apt`:

```

root# ubuntu-drivers devices
model      : MCP89 [GeForce 320M]
vendor     : NVIDIA Corporation
modalias   : pci:v000010DEd000008A4sv0000106Bsd000000C0bc03sc00i00
driver     : xserver-xorg-video-nouveau - distro free      builtin
driver     : nvidia-304                  - distro non-free
driver     : nvidia-340                  - distro non-free recommended
...
root# apt install nvidia-340

```

#### Kernel-Updates (LTS Enablement Stack)

Im Zuge der gewöhnlichen Updates für Ubuntu LTS erhalten Sie für den Kernel und das X Window System ausschließlich Bugfixes. Die Grundversion des Kernels und des X Window Systems bleibt aber standardmäßig während der ganzen Lebensdauer von Ubuntu LTS unverändert. Ein wenig anders sieht es aus, wenn Sie eine LTS-Neuinstallation durchführen: Canonical aktualisiert circa zweimal im Jahr die Installationsmedien (Ubuntu 16.04.1, 16.04.2 etc.) und verwendet dabei mitunter aktuellere Kernel- und X-Window-Versionen. Das erhöht die Kompatibilität von Ubuntu LTS zu neuer Hardware.

Wie aber kommen Sie bei einer alten Ubuntu-LTS-Installation zu neueren Kernel- und X-Window-Versionen? Dazu müssen Sie explizit HWE-Pakete installieren (*Hardware Enablement*). Die beiden folgenden Kommandos gelten für Ubuntu 16.04 LTS. Bei Ubuntu 18.04 müssen Sie die Versionsnummer entsprechend anpassen. Ein Update des X-Server-Stacks inklusive Mesa ist naturgemäß nur bei Desktop-Installationen zweckmäßig.

```

root# sudo apt install --install-recommends linux-image-generic-hwe-16.04
root# sudo apt install --install-recommends xserver-xorg-hwe-16.04

```

Beachten Sie aber, dass Sie damit in einen Rolling-Release-Modus für den Kernel bzw. für den X-Server und für Mesa wechseln! Wenn in etwa einem halben Jahr die nächste Ubuntu-LTS-Update-Version freigegeben wird (16.04.3, 16.04.4 etc.), dann erhalten Sie im Rahmen der gewöhnlichen Updates wiederum den dann aktuellen Kernel und evt. auch neuere X-Window-Pakete! Im Detail ist dieses Update-Konzept hier beschrieben:

<https://wiki.ubuntu.com/Kernel/LTSEnablementStack>  
<https://wiki.ubuntu.com/Kernel/RollingLTSEnablementStack>

#### »If it ain't broken, don't fix it!«

Führen Sie ein Update der Kernel- bzw. der X- und Mesa-Versionen nur durch, wenn dazu wirklich eine Notwendigkeit besteht, d. h., wenn Sie sich davon z. B. eine bessere Unterstützung Ihrer Hardware erwarten. Ansonsten sollten Sie der Stabilität der ursprünglich ausgelieferten Versionen den Vorzug geben.

## 3.7 Ubuntu Server

Ubuntu Server ist eine für den Server-Betrieb optimierte Variante von Ubuntu. Vor allem die LTS-Versionen von Ubuntu Server zählen wegen des langen Support-Zeitraums von fünf Jahren mittlerweile neben CentOS und Debian zu den beliebtesten Linux-Server-Systemen im nichtkommerziellen Bereich. Canonical versucht Ubuntu Server aber auch im kommerziellen Segment zu etablieren und bietet für zahlungswillige Kunden mit *Landscape* ein optionales Werkzeug zur Überwachung mehrerer Server-Installationen:

<https://landscape.canonical.com>

Hinter den Kulissen unterscheidet sich Ubuntu Server eigentlich nicht vom gewöhnlichen Ubuntu, d. h., es verwendet dieselben Pakete und Paketquellen. Die Besonderheit von Ubuntu Server besteht vielmehr darin, dass es speziell für den Server-Einsatz ein eigenes, textbasiertes Installationsprogramm gibt. Im Unterschied zu dem im vorigen Abschnitt vorgestellten Desktop-Installationsprogramm kommt es gut mit LVM und Software-RAID zurecht.

Das Installationsprogramm richtet ein Grundsystem für den Server-Einsatz ohne grafische Benutzeroberfläche ein: Da viele Server-Systeme ohnedies via SSH administriert werden, ist eine Benutzeroberfläche selten notwendig. Grundsätzlich ist es natürlich auch möglich, eine gewöhnliche Ubuntu-Installation für den Server-Einsatz zu verwenden und die entsprechenden Server-Pakete einfach nachträglich zu installieren.

### Ubuntu Server installieren

Auf der Website <http://www.ubuntu.com/download/server> finden Sie ISO-Images zur Ubuntu-Server-Installation. In aller Regel sollten Sie sich für die 64-Bit-Variante der letzten LTS-Version entscheiden. Sie finden auf der Download-Seite auch Installationsmedien für Nicht-LTS-Versionen. Diese sind zwar aktueller, werden aber nur für neun Monate mit Updates versorgt. Für den Server-Einsatz ist das zu wenig!

Installations-  
medien

Nachdem Sie die ISO-Datei auf eine CD gebrannt oder auf einen USB-Stick übertragen haben, starten Sie Ihren Rechner damit neu. In den ersten Dialogen des Installationsprogramms wählen Sie die Sprache und Ihr Land oder Gebiet aus. Diese Information wird zur Auswahl des nächstgelegenen Mirror-Servers verwendet. Das Installationsprogramm kann das Tastaturlayout selbst erkennen. Wesentlich schneller ist es aber, das gewünschte Layout von Hand einzustellen.

Grund-  
einstellungen

Nach der Hardware-Erkennung versucht das Installationsprogramm, das Netzwerk automatisch zu konfigurieren. Das gelingt, wenn sich im lokalen Netzwerk ein Router

bzw. DHCP-Server befindet. Andernfalls haben Sie die Wahl, auf die Netzwerkkonfiguration vorerst zu verzichten oder die wichtigsten Parameter manuell einzugeben.

**Benutzerdaten** Im nächsten Schritt geben Sie den Namen und das Passwort eines Ubuntu-Benutzers an. Dieser Benutzer ist in Zukunft für die Administration des Servers zuständig. Weitere Benutzer können Sie später im laufenden Betrieb hinzufügen. Das Installationsprogramm fragt nun, ob es Ihr persönliches Verzeichnis verschlüsseln soll. Für den Server-Einsatz ist dies nicht zweckmäßig.

**Partitionierung der Festplatte** Die verschachtelten Dialoge zur Partitionierung der Festplatten sind leider sehr unübersichtlich. Im ersten Dialog stellt das Installationsprogramm verschiedene Kommandos zur Auswahl. Je nachdem, wie viele Festplatten Ihr Rechner hat und welche Partitionen sich darauf bereits befinden, kann das Auswahlménú zusätzliche Kommandos aufweisen:

- ▶ **GEFÜHRT – PARTITION N VERKLEINERN UND DEN FREIGEWORDENEN SPEICHER NUTZEN:** Diese Option erscheint nur, wenn (Windows-)Partitionen auf der Festplatte existieren. Mit dem Kommando können Sie deren Größe reduzieren, um so Platz für Linux-Partitionen zu machen.
- ▶ **GEFÜHRT – VOLLSTÄNDIGE FESTPLATTE VERWENDEN:** Das Installationsprogramm erstellt einen Vorschlag, wie die gesamte Festplatte für Linux-Partitionen genutzt werden kann. Diesen Vorschlag können Sie bestätigen (**PARTITIONIERUNG BEENDEN UND ÄNDERUNGEN SPEICHERN**) oder abbrechen (**ÄNDERUNGEN RÜCKGÄNGIG MACHEN**). Vorsicht: Mit dieser Option verlieren Sie alle bisher auf der Festplatte gespeicherten Daten!
- ▶ **GEFÜHRT – DEN GRÖSSTEN FREIEN SPEICHERBEREICH BENUTZEN:** Diese Option ist nur sinnvoll, wenn es auf der Festplatte partitionsfreien Platz gibt. Das Installationsprogramm erstellt einen Vorschlag, wie der freie Platz auf der Festplatte für Linux-Partitionen genutzt werden soll. Sie müssen diesen Vorschlag anschließend bestätigen. Wenn Sie das nicht tun (Antwort **NEIN**), gelangen Sie in einen Dialog mit der Partitionstabelle (siehe [Abbildung 3.23](#)). Dort können Sie die Partitionen manuell verändern oder alle **ÄNDERUNGEN AN DEN PARTITIONEN RÜCKGÄNGIG MACHEN**.
- ▶ **GEFÜHRT – GESAMTE PLATTE VERWENDEN UND LVM EINRICHTEN:** Auch mit dieser Option werden alle vorhandenen Daten der Festplatte gelöscht. Anschließend richtet das Installationsprogramm ein LVM-System ein.
- ▶ **GEFÜHRT – GESAMTE PLATTE MIT VERSCHLÜSSELTEM LVM:** Wie oben, allerdings wird das gesamte Dateisystem zusätzlich verschlüsselt. Für den Server-Einsatz ist diese Option ungeeignet, weil das Verschlüsselungspasswort bei jedem Start manuell eingegeben werden muss und die Verschlüsselung den Festplattenzugriff spürbar verlangsamt.

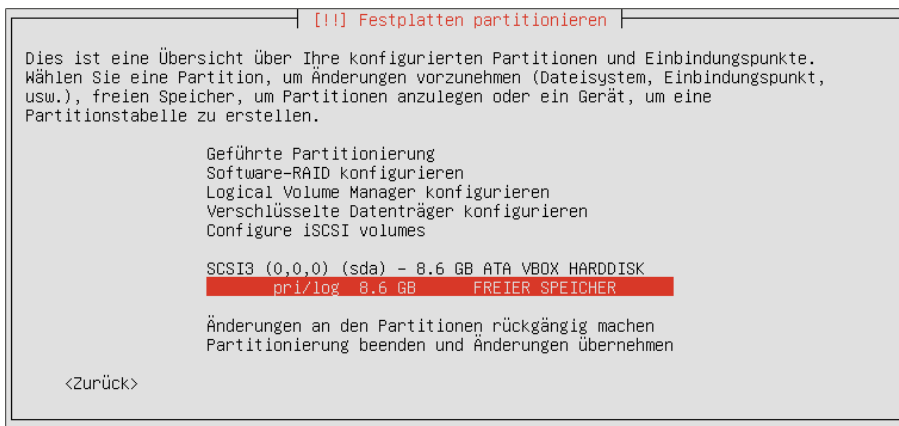
- ▶ **MANUELL**: Mit diesem Punkt können Sie neue Linux-Partitionen für die Ubuntu-Installation von Hand anlegen.

Bei allen Partitionierungsvarianten, deren Menüpunkt mit **GEFÜHRT** beginnt, überlassen Sie die Partitionierung dem Installationsprogramm. Dieses erzeugt eine kleine Swap-Partition und eine Systempartition, die den Rest der Festplatte füllt. Dieses Setup ist freilich selten optimal.

Automatische  
Partitionierung

Das Menükommando **MANUELL** führt in einen neuen Dialog, der einige Menükommandos sowie eine Liste aller vorhandenen Festplatten bzw. SSDs, Partitionen sowie Logical Volumes enthält (siehe [Abbildung 3.23](#)).

Manuelle  
Partitionierung



**Abbildung 3.23** Die Partitionstabelle

In der Partitionstabelle wählen Sie den Eintrag **FREIER SPEICHER** aus. Wenn es keinen freien Speicher gibt, müssen Sie zuerst eine vorhandene Partition löschen oder ändern. Wenn die Festplatte/SSD fabrikneu ist, müssen Sie sie zuerst initialisieren, also eine Partitionstabelle einrichten.

Im nächsten Dialog entscheiden Sie sich für die Option **EINE NEUE PARTITION ERSTELLEN**. Anschließend geben Sie die gewünschte Partitionsgröße an und wählen den Partitionstyp. (Die erste Partition der Festplatte muss eine primäre Partition sein. Oft befindet sich darin Windows. Für alle weiteren Partitionen wählen Sie **LOGISCH**.) Das Installationsprogramm zeigt nun eine Zusammenfassung der Einstellungen für diese Partition an (siehe [Abbildung 3.24](#)).

Für die Systempartition können Sie zumeist alle Einstellungen beibehalten und müssen diese nur noch durch **ANLEGEN DER PARTITION BEENDEN** bestätigen. Damit gelangen Sie zurück in die Partitionstabelle, die nun eine weitere Partition enthält.

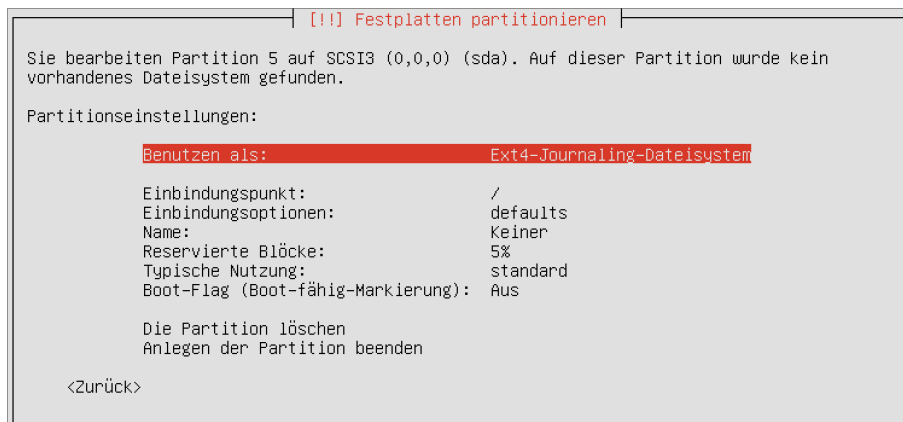


Abbildung 3.24 Die Einstellungen für die neue Partition

Beim Anlegen der Swap-Partition müssen Sie in den Partitionseinstellungen eine Änderung vornehmen: Den Punkt BENUTZEN ALS stellen Sie auf AUSLAGERUNGSDATEI (SWAP). Auch beim Anlegen zusätzlicher Partitionen (/home, /tmp etc.) müssen Sie die Partitionseinstellungen ändern: Diesmal wählen Sie den Punkt EINBINDUNGSPUNKT aus und stellen dann den gewünschten Verzeichnisnamen für die Partition ein.

Nach der Definition aller Partitionen führen Sie in der Partitionstabelle das Kommando PARTITIONIERUNG BEENDEN UND ÄNDERUNGEN ÜBERNEHMEN aus. Nach einer weiteren Rückfrage werden die Änderungen an der Festplatte tatsächlich durchgeführt. Anschließend kopiert das Installationsprogramm unzählige Dateien in die soeben angelegte Systempartition. Das dauert einige Minuten.

#### Automatische Updates

Wenn Sie möchten, installiert Ubuntu im laufenden Betrieb einmal täglich automatisch neue Updates, sobald diese verfügbar werden. Das ist praktisch, wenn Sie nicht regelmäßig kontrollieren möchten, ob es neue Sicherheits-Updates gibt.

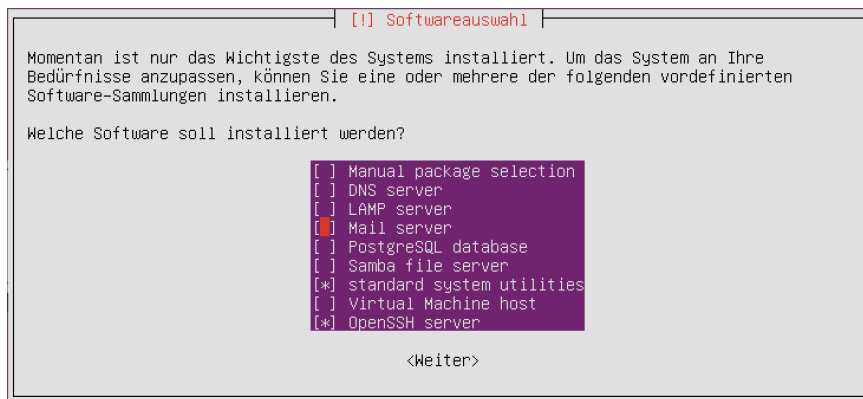


Abbildung 3.25 Paketauswahl



Außer den Grundpaketen kann das Installationsprogramm auch gleich diverse Server-Pakete installieren (siehe [Abbildung 3.25](#)). An dieser Stelle wähle ich normalerweise nur den OpenSSH-Server aus und installiere die restlichen Server-Dienste dann nach und nach selbst. Mit der Installation ist es ja normalerweise nicht getan – Server-Programme müssen auch konfiguriert werden.

**Paketauswahl**

Bei EFI-Rechnern wird GRUB in der EFI-Partition eingerichtet. Bei BIOS-Rechnern erscheint eine Rückfrage, ob Sie GRUB in den Startsektor (MBR) der ersten Festplatte installieren möchten. Diesen Vorschlag bestätigen Sie in der Regel einfach durch .

**Bootloader**



TEIL II

# Linux anwenden



# Kapitel 4

## Linux-Schnelleinstieg

Dieses Kapitel hilft Ihnen bei den ersten Schritten unter Linux: einloggen, Programme ausführen, auf Dateien und Datenträger zugreifen, ausloggen bzw. Rechner herunterfahren etc. Das Kapitel vermittelt ein minimales Grundlagenwissen über die Dateiverwaltung von Linux und verrät, wo Sie im installierten System bzw. im Internet nach Online-Dokumentation suchen können.

Ein Grundproblem bei einer allgemeinen Beschreibung von Linux besteht darin, dass nahezu jede Funktion frei konfigurierbar ist. Daher sieht das Startmenü jeder Distribution ein wenig anders aus. Es kann sein, dass eine bestimmte Tasten- oder Mauskombination unter Ubuntu eine andere Reaktion hervorruft als unter SUSE. Aus diesem Grund gibt es in diesem Kapitel viele Formulierungen mit *meistens*, *gewöhnlich* etc. Das ist leider nicht zu ändern.

### 4.1 Linux starten und beenden

Um Linux zu starten, müssen Sie Ihren Rechner neu starten. Beim Neustart geben Sie in einem Menü an, dass Sie Linux und nicht Windows ausführen möchten. Bei aktuellen Rechnern müssen Sie unter Umständen eine Tastenkombination drücken, damit das EFI-Bootmenü angezeigt wird. Es ist nicht möglich, Linux von Windows aus zu starten – es sei denn, Sie verwenden unter Windows ein Programm wie VirtualBox und führen Linux in einer virtuellen Umgebung aus.

Der Linux-Boot-Prozess dauert je nach Hardware ca. 15 Sekunden. Bei manchen Distributionen erscheint während dieser Zeit ein Fortschrittsbalken. Andere Distributionen zeigen hingegen im Textmodus unzählige Detailinformationen über den Systemstart an. Lassen Sie sich vom technischen Eindruck nicht irritieren! Sie können die Informationen normalerweise ignorieren. Interessant sind die ausgegebenen Daten nur, wenn irgendetwas nicht funktionieren sollte.

Im Normalfall endet der Boot-Prozess mit dem Erscheinen einer grafischen Login-Box (siehe [Abbildung 4.1](#)). Dort melden Sie sich mit Ihrem Benutzernamen und dem Passwort an. Bei einigen Distributionen haben Sie beim Login auch die Möglichkeit, **Login**

das Tastaturlayout und die Sprache einzustellen. Anschließend erscheint Ihre Arbeitsumgebung im Standard-Desktop-System Ihrer Distribution, üblicherweise Gnome, Unity oder KDE. Eine Einführung in diese Desktop-Systeme folgt in den beiden nächsten Kapiteln.




Abbildung 4.1 Login mit Passwordeingabe

Melden Sie sich nicht als `root` an, sondern verwenden Sie einen gewöhnlichen Login! Der Benutzer `root` hat uneingeschränkte Rechte. Es ist unter Linux nicht üblich, mit `root`-Rechten zu arbeiten. Stattdessen werden für administrative Aufgaben nur einzelne Programme mit `root`-Rechten ausgeführt, wobei Sie diesen Vorgang durch die Eingabe des `root`-Passworts bzw. Ihres eigenen Passworts (Ubuntu) bestätigen müssen. Die Veränderung des Passworts für `root` und das Einrichten neuer Benutzer wird in [Abschnitt 18.5](#), »Benutzer und Gruppen, Passwörter«, beschrieben.

#### Auto-Login – bequem oder sicher?

Linux kann so konfiguriert werden, dass nach dem Rechnerstart ein automatischer Login erfolgt. Das ist zwar bequem, aber sicherheitstechnisch nicht optimal. Tipps zur Konfiguration der Auto-Login-Funktion finden Sie in [Abschnitt 21.5](#), »Start des Grafiksystems«.

- Bildschirm Sperre** Nach einigen Minuten ohne Aktivität wird der Bildschirm automatisch gesperrt. Erst wenn Sie  drücken, erscheint eine Box, in der Sie mit Ihrem Passwort zurück in die weiterhin aktive Session gelangen.
- Logout** Die KDE- bzw. Gnome-Menüs bzw. das Panel sehen jeweils ein Kommando zum Logout vor. Die genaue Bezeichnung des Menükommandos variiert je nach Distribution und lautet z. B. `SYSTEM • BENUTZER ABMELDEN`. Damit werden sämtliche auf dem Desktop laufenden Programme beendet. Sichern Sie vorher alle noch offenen Dateien! Der Logout führt zurück zum Login-Bildschirm, in dem Sie sich nun neu einloggen oder den Rechner herunterfahren können.

Bei den meisten Desktop-Systemen ist für einen Benutzerwechsel kein Logout erforderlich. Stattdessen führen Sie im Systemmenü `BENUTZER WECHSELN` oder einen vergleichbaren Eintrag aus. Beachten Sie, dass damit zwei Benutzer parallel eingeloggt sind und somit auch beide Benutzer Speicherplatz und CPU-Zeit beanspruchen.

Benutzerwechsel

Viele Funktionen von Linux können auch im Textmodus genutzt werden. Gerade bei Server-Installationen wird manchmal ganz auf das Grafiksystem verzichtet. Zum Arbeiten führen Sie den Login in einer Textkonsole durch (siehe auch [Kapitel 9](#), »Terminalfenster und Konsolen«). Zum Logout drücken Sie einfach `[Strg]+[D]` oder führen das Kommando `exit` aus.

Login und Logout im Textmodus

Die grafischen Benutzeroberflächen sehen Menükommandos zum Herunterfahren des Rechners vor oder bieten eine entsprechende Option im Abmeldedialog. Im Textmodus erfolgt ein ordnungsgemäßes Herunterfahren des Systems mit dem Kommando `shutdown -h now`. Dieses Kommando darf allerdings nur von `root` ausgeführt werden.

Linux beenden (Shutdown)

Auf vielen Linux-Systemen gibt es eine bequemere Alternative zum `shutdown`-Kommando, die aber nur im Textmodus zur Verfügung steht: Drücken Sie einfach die Tasten `[Strg]+[Alt]+[Entf]`.

#### Sicherer Neustart ohne Root-Rechte

Wenn Linux auf `[Strg]+[Alt]+[Entf]` nicht reagiert und Sie kein `root`-Passwort besitzen, den Rechner aber dennoch neu starten müssen, sollten Sie vorher zumindest das Kommando `sync` ausführen. Damit werden alle gepufferten Schreibzugriffe auf die Festplatte ausgeführt. Unmittelbar danach schalten Sie Ihren Rechner aus. Das ist allerdings nur eine Notlösung zur Schadensminimierung.

## 4.2 Tastatur, Maus und Zwischenablage

Welche Tastenkürzel zur Verfügung stehen, hängt davon ab, ob Sie im Grafikmodus oder in einer Textkonsole arbeiten. Hier setze ich voraus, dass Sie den Grafikmodus nutzen. Die Tastenkürzel werden in drei Programmebenen definiert:

Wichtige Tastenkürzel

- ▶ Je nach Distribution ist das X Window System oder Wayland für die elementaren Funktionen des Grafiksystems verantwortlich. Es stellt nur relativ wenige Tastenkürzel direkt zur Verfügung (siehe [Tabelle 4.1](#)).
- ▶ Desktop-Systeme wie Gnome, KDE, Unity, Xfce oder LXDE bauen auf dem Grafiksystem auf. Auch durch sie werden Tastenkürzel definiert. Erfreulicherweise hat hier in den letzten Jahren eine Vereinheitlichung stattgefunden, sodass zumindest für die wichtigsten Operationen dieselben Tastenkürzel gelten (siehe [Tabelle 4.2](#)).

- Schließlich hängen die verfügbaren Tastenkürzel natürlich vom individuellen Programm ab, das gerade läuft. Je nachdem, ob Sie mit Firefox im Web surfen, mit LibreOffice einen Brief schreiben oder im Editor Emacs Programmcode ändern – in jedem Fall gelten andere Tastenkürzel, die in diesem Abschnitt natürlich nicht beschrieben werden können.

Bei Programmen mit grafischer Benutzeroberfläche gelten für wichtige Operationen dieselben Kürzel wie unter Windows. Das betrifft beispielsweise das Kopieren von Text in die Zwischenablage mit `[Strg]+[C]`, das Einfügen des kopierten Texts mit `[Strg]+[V]` oder das Speichern einer Datei mit `[Strg]+[S]`.

Für viele textorientierte Kommandos gelten andere Konventionen, die sich im Verlauf der Unix/Linux-Geschichte etabliert haben. Wichtige Tastenkürzel für derartige Programme sind in [Abschnitt 9.1](#), »Textkonsolen und Terminalfenster«, zusammengefasst.

| Kürzel                         | Bedeutung   |
|--------------------------------|---|
| <code>[Strg]+[Alt]+[←]</code>  | beendet bei wenigen Distributionen das Grafiksystem gewaltsam. Unter SUSE muss diese Tastenkombination zweimal gedrückt werden. Bei einigen Distributionen ist die Tastenkombination auch ganz deaktiviert oder führt in einen Dialog, in dem Sie den Rechner herunterfahren können.  |
| <code>[Strg]+[Alt]+[Fn]</code> | wechselt in die Text- oder Grafikkonsole <i>n</i> . Welche Konsole wie genutzt wird, hängt von der Distribution ab. Bei aktuellen Fedora-Desktop-Installationen läuft in der Konsole 1 immer eine grafische Login-Box. Konsole 2 wird für das Grafiksystem des gerade eingeloggten Benutzers verwendet. Die Konsolen 3 bis 6 laufen im Textmodus. Somit führt <code>[Strg]+[Alt]+[F3]</code> in eine Textkonsole und <code>[Strg]+[Alt]+[F2]</code> zurück in die Desktop-Umgebung. |

Tabelle 4.1 Tastenkürzel unter X

| Kürzel  | Bedeutung                                       |
|---|---|
| <code>[Alt]+[↩]</code>                        | wechselt das aktive Fenster.                    |
| <code>[Alt]+[F1]</code> oder <code>[≡]</code> | zeigt das Startmenü an.                         |
| <code>[Alt]+[F2]</code> name <code>[↵]</code> | startet das Programm name.                      |
| <code>[Alt]+[F4]</code>                       | schließt das Fenster bzw. beendet das Programm. |

Tabelle 4.2 Wichtige Tastenkürzel des Desktops (Gnome, KDE, Unity)



## Verwendung der Maus

Linux orientiert sich tendenziell immer mehr an den Konventionen der Windows- bzw. macOS-Welt. Dennoch existieren je nach Desktop bzw. je nachdem, welches Programm Sie gerade einsetzen, einige Besonderheiten, die in diesem Abschnitt zusammengefasst sind.

Bei den meisten Desktop-Systemen ist wie unter Windows für viele Operationen – etwa das Öffnen einer Datei – ein Doppelklick erforderlich. In KDE werden dagegen viele Mausoperationen standardmäßig durch einen einfachen Mausklick ausgeführt. Wie Sie auch in KDE den Doppelklickmodus aktivieren, ist in [Abschnitt 6.1](#), »KDE«, beschrieben.

Einfach- oder Doppelklick

Das Kopieren von Text mit `Strg+C` und `Strg+V` ist Ihnen von Windows natürlich vertraut. In fast allen Linux-Programmen können Sie aber auch mit der Maus Textauschnitte kopieren und an einer anderen Stelle oder in einem anderen Programm wieder einfügen. Zum Markieren von Textauschnitten bewegen Sie die Maus einfach mit gedrückter linker Maustaste über den Text. Der so markierte Text wird dabei automatisch in einen Puffer kopiert. Sobald Sie die mittlere Maustaste drücken, wird der Text dort eingefügt, wo der aktive Eingabecursor steht. Bei einzelnen Programmen können Sie auch die rechte Maustaste zum Einfügen verwenden, was besonders dann praktisch ist, wenn Sie eine Maus mit nur zwei Tasten verwenden.

Texte mit der Maus kopieren und einfügen

Das Markieren und Kopieren erfolgt also allein mit der Maus, ohne Tastatur. Wenn Sie sich einmal an diese Methode gewöhnt haben, werden Sie sich immer wieder fragen, warum das unter Windows oder macOS nicht ebenso einfach funktioniert.

Bei manchen alten Programmen kann bei Dialogen nur dann Text in Eingabefelder eingegeben werden, wenn sich die Maus über diesem Feld befindet. Der Eingabefokus hängt also nicht nur davon ab, welches Programm gerade aktiv ist, sondern auch davon, wo sich die Maus befindet.

Eingabefokus

Dieses Verhalten kann bei einigen Desktop-Systemen auch für Fenster aktiviert werden (*focus follows mouse*): Dann ist es nicht mehr erforderlich, ein Fenster anzuklicken, um darin Eingaben durchzuführen. Es reicht, die Maus richtig zu positionieren. Allerdings führt eine unbeabsichtigte Bewegung der Maus nun oft dazu, dass Texteingaben an das falsche Fenster oder Programm weitergeleitet werden. Aus diesem Grund ist der Modus *focus follows mouse* nicht gebräuchlich.

Wenn die Maus nicht funktioniert, können Sie den Mauszeiger bei einigen Distributionen zur Not auch mit der Tastatur steuern (siehe [Tabelle 4.3](#)). Dazu müssen Sie mit `⇧+Strg+Num` einen speziellen Tastaturmodus aktivieren, der leider nur bei Tastaturen mit eigenem Ziffernblock funktioniert.

Maussteuerung per Tastatur

| Kürzel   | Bedeutung  |
|----------|--|
| [4], [6] | Maus nach links bzw. rechts bewegen  |
| [2], [8] | Maus nach unten bzw. oben bewegen  |
| [5]      | linke Maustaste kurz drücken   |
| [+]      | Doppelklick  |
| [0]      | Maustaste bleibend drücken ([5] löst die Taste wieder.)  |
| [-]      | auf die rechte Maustaste umschalten<br>([5], [+] und [0] gelten jetzt für die rechte Maustaste.) |
| [*]      | wieder auf die linke Maustaste umschalten  |

**Tabelle 4.3** Tastenkürzel zur Maussteuerung durch den numerischen Ziffernblock

### 4.3 Umgang mit Dateien, Zugriff auf externe Datenträger

Das Dateisystem beginnt mit dem Wurzelverzeichnis /. Auch wenn es mehrere Festplatten bzw. Festplattenpartitionen und ein DVD-Laufwerk gibt, sind alle Daten in den Verzeichnisbaum eingebunden. Beispielsweise kann der Inhalt eines DVD-Laufwerks üblicherweise unter dem Verzeichnis /run/media/benutzernamedvname gelesen werden. Aus diesem Grund besteht unter Linux keine Notwendigkeit für die Laufwerksbuchstaben C:, D: etc., die unter Windows üblich sind.

Linux unterscheidet bei Datei- und Verzeichnisnamen zwischen Groß- und Kleinbuchstaben. `readme`, `Readme` und `README` bezeichnen drei verschiedene Dateien! Dateinamen dürfen bis zu 255 Zeichen lang sein.

#### Heimatverzeichnis

Nach dem Einloggen befinden Sie sich automatisch in einem Verzeichnis, das Ihnen gehört. Dieses Verzeichnis wird Heimat- oder Home-Verzeichnis genannt. Andere Benutzer dürfen diese Dateien weder verändern noch löschen, aber in der Regel lesen. Das Heimatverzeichnis wird oft mit der Tilde ~ abgekürzt. Bei gewöhnlichen Linux-Anwendern befindet sich das Heimatverzeichnis in /home/name. Bei root lautet der Speicherort dagegen /root.

#### DVDs, SD-Karten und USB-Sticks

Im Idealfall funktioniert der Zugriff auf externe Datenträger weitgehend automatisch: Nach dem Einlegen bzw. Anstecken eines Datenträgers erscheint auf dem Desktop automatisch ein entsprechendes Icon oder ein Fenster des Dateimanagers.

Wenn das automatische Einbinden externer Datenträger nicht funktioniert bzw. wenn Sie in einer Textkonsole arbeiten, müssen Sie manuell das Kommando `mount` ausführen – und später `umount` benutzen, um den Datenträger wieder zu lösen. Die

Vorgehensweise wird ausführlich in [Kapitel 22](#), »Administration des Dateisystems«, beschrieben.

#### Melden Sie USB-Sticks und externe Festplatten richtig ab!

Bevor Sie einen Datenträger mit Schreibzugriff entfernen bzw. das Kabel lösen, müssen Sie ihn explizit abmelden. Die Details hängen vom Desktop-System bzw. von der Distribution ab. In der Regel klicken Sie das Icon an und führen ein Kommando in der Art DATENTRÄGER LÖSEN oder DATENTRÄGER SICHER ENTFERNEN aus. Wenn Sie das vergessen, riskieren Sie ein inkonsistentes Dateisystem auf dem Datenträger und Datenverluste!

Mit dem Kommando `df` stellen Sie fest, welche Partitionen momentan in das Dateisystem eingebunden sind und wie viel Speicher dort noch frei ist. Die Option `-h` bewirkt, dass als Maßeinheit nicht starr KiB verwendet wird, sondern eine zur Größe der Partition passende Einheit (MiB, GiB etc.). Im folgenden Beispiel ist außer der Systempartition `/` noch die Datenpartition `/home` in das Dateisystem eingebunden:

```
user$ df -h -x tmpfs
Dateisystem  Größe Benutzt Verf. Verw% Eingehängt auf
/dev/sda1    16G   5,5G  9,5G  37% /
/dev/sda2    18G   3,6G  15G  20% /home
```

Sie können `df` auch ohne Optionen ausführen. Das führt aber zu einem unübersichtlichen Ergebnis. `-h` bewirkt, dass die Größenangaben vernünftig formatiert werden (*human readable*). `-x tmpfs` unterdrückt die Anzeige temporärer Dateisysteme, die in der Regel nicht von Belang sind.

## 4.4 Dokumentation zu Linux

Zu Linux gibt es nahezu unendlich viel Dokumentation, die teilweise gleich mitgeliefert wird und ansonsten im Internet zu finden ist. Experten können zudem einen Blick in den Quellcode aller Programme werfen. Dieser Abschnitt gibt einen Überblick über die wichtigsten Informationsquellen.

Der beste Startpunkt für Einsteiger sind die unzähligen Foren, Wikis und sonstige Websites von Linux-Firmen und -Enthusiasten. Eine Aufzählung ist hier überflüssig – eine kurze Suche nach *fedora forum* oder *ubuntu wiki* führt unweigerlich zu den richtigen Seiten. Bei distributionsspezifischen Fragen sind Sie im Vorteil, wenn Sie eine populäre Distribution einsetzen: Je größer die Verbreitung einer Distribution ist, desto reger ist der Informationsaustausch in den Foren. Ein leuchtendes Beispiel sind die Foren und Wikis zu Ubuntu.

Linux-Foren und  
-Wikis im Internet

**man und info** Bei vielen textorientierten Kommandos führen `man name` oder `info name` zu einer genauen Beschreibung und Syntaxreferenz. Weitere Informationen zu `man` und `info` finden Sie in [Abschnitt 9.3](#), »Online-Hilfe«. `man`- und `info`-Texte können Sie auch in den Hilfesystemen von Gnome und KDE lesen.

**Paketdokumentation** Unter Linux werden Programme in Form von Paketen installiert. Ein Paket enthält neben allen für ein Programm erforderlichen Dateien oft auch Dokumentationsdateien. Diese Dokumentation wird je nach Distribution in die folgenden Verzeichnisse installiert:

```
Debian, Fedora, Red Hat, Ubuntu:  /usr/share/doc/<paketname>
SUSE:                             /usr/share/doc/packages/<paketname>
```

Was tun Sie, wenn Sie Zusatzdokumentation zu einem bestimmten Kommando suchen, aber nicht wissen, zu welchem Paket das Kommando gehört? Der erste Schritt besteht darin, den genauen Dateinamen des Kommandos festzustellen. Dazu führen Sie `which -a kommando` aus:

```
user$ which -a cp
/bin/cp
```

Im nächsten Schritt ermitteln Sie, zu welchem Paket diese Datei gehört. Die Vorgehensweise hängt davon ab, welches Paketformat Ihre Distribution verwendet (siehe auch [Kapitel 19](#), »Software- und Paketverwaltung«). Das folgende Kommando verrät, dass `cp` ein Teil des Pakets `coreutils` ist:

```
user$ rpm -qf /bin/cp      (Fedora, Red Hat, SUSE)
coreutils-8.23-10
user$ dpkg -S /bin/cp     (Debian, Ubuntu)
coreutils: /bin/cp
```

**Kernel-dokumentation** Eine Menge hardware-spezifische Informationen finden Sie in der Kerneldokumentation. Sie ist Teil des Kernelcodes. Die Dokumentation des gerade aktuellen Linux-Kernels können Sie auch im Internet nachlesen:

<https://www.kernel.org/doc/Documentation>

**RFCs** RFC steht für *Request For Comments*. Dahinter verbergen sich Dokumente, die diverse Protokolle (darunter z. B. TCP, IP, FTP, PPP etc.) im Detail beschreiben. Das etwas merkwürdige Kürzel RFC deutet auf die Entstehungsgeschichte dieser Protokolle hin: Sie wurden im Regelfall nicht durch eine Person, Organisation oder Firma diktiert, sondern sind aus einem oft langwierigen Diskussionsprozess entstanden. Die hier dargestellten Informationen sind sehr technischer Natur und für Einsteiger ungeeignet. RFCs finden Sie z. B. hier:

<http://faqs.org/rfcs>

# Kapitel 5

## Gnome

Unter Windows oder macOS gibt es jeweils nur *eine* Desktop-Umgebung, deren Aussehen und Verhalten sich nur bei Versionswechseln merklich ändert. Unter Linux stehen dagegen eine ganz Menge Desktop-Systeme zur Auswahl:

- ▶ **Gnome** ist das populärste Desktop-System. Es kommt standardmäßig in allen Enterprise-Distributionen sowie unter Debian und Fedora zum Einsatz. Auch Ubuntu verwendet ab Version 17.10 wieder Gnome.

Gnome hat mit der 2011 vorgestellten und bis heute aktuellen Version 3 einen Neuanfang gewagt, der in seiner Radikalität manche langjährige Gnome-Fans überfordert bzw. verärgert hat. In der Folge sind mehrere Gnome-Ableger entstanden, unter anderem der **Cinnamon Desktop** und **MATE**. Selbst innerhalb des Gnome-Projekts gibt es mit dem sogenannten **Gnome-Klassikmodus** eine Variante, die in vielerlei Hinsicht Ähnlichkeiten zur Gnome-Version 2 zeigt.

- ▶ Mit **Unity** hat Canonical versucht, für Ubuntu einen eigenen Desktop zu schaffen. Als Basis diente auch hier Gnome, wenn auch mit vielen Änderungen. Obwohl Unity eine große Verbreitung gefunden hat, beschloss Canonical im Frühjahr 2017, die Weiterentwicklung zu stoppen.
- ▶ **KDE** ist ein Desktop-System für technisch versierte Anwender. Es bietet viel mehr Konfigurationsmöglichkeiten als Gnome und Unity, was leider mit vielen unübersichtlichen Dialogen einhergeht.
- ▶ **Pantheon** ist das Desktop-System der Distribution Elementary OS. Das Ziel der Pantheon-Entwickler ist es, die Einfachheit und Eleganz von macOS unter Linux nachzubilden. Optisch ist dies auch gut gelungen, Pantheon und Elementary OS sind aber noch unausgereift.
- ▶ **Xfce** und **LXDE** sind Desktop-Systeme, die speziell für Rechner mit nicht so leistungsstarker Hardware optimiert sind. LXDE hat zuletzt große Verbreitung auf Raspberry Pis gefunden.

In diesem Kapitel konzentriere ich mich auf Gnome, wobei ich auch auf die Varianten Cinnamon und MATE kurz eingehe. Das nächste Kapitel ist KDE und Unity gewidmet. Einige Informationen zu LXDE und dessen Variante »Pixel Desktop« finden Sie schließlich in [Kapitel 8](#), »Raspberry Pi«.

## 5.1 Erste Schritte

**X oder Wayland** Gnome läuft im Grafiksystem – klar! Aktuell gibt es aber zwei Implementierungsvarianten für das Grafiksystem: den traditionellen X-Server (Xorg) und das neuere Wayland-System. Im Detail beschreibe ich den Unterschied in [Kapitel 21](#), »Grafiksystem«. Gnome ist das erste Desktop-System, das weitestgehend Wayland-kompatibel ist. Fedora nutzt seit Version 26 standardmäßig Wayland; andere Distributionen werden voraussichtlich bald folgen.

Idealerweise sollten Sie gar nicht merken, welches Grafiksystem hinter den Kulissen von Gnome aktiv ist. Tatsächlich ist Wayland aktuell aber mit diversen Einschränkungen verbunden, die z. B. den Start von Programmen mit Administratorrechten betreffen. Diesen Problemen können Sie aus dem Weg gehen, wenn Sie sich beim Gnome-Login explizit für X entscheiden. Bei Fedora erreichen Sie über das Zahnradmenü der Login-Box den Eintrag GNOME UNTER XORG (siehe [Abbildung 21.3](#)).

**Willkommens-Assistent** Unmittelbar nach dem ersten Login erscheint ein Willkommens-Assistent. Dort können Sie die Sprache sowie das Tastaturlayout einstellen und Gnome mit einem Online-Konto verbinden, also z. B. mit Google oder Nextcloud. Diesen Schritt können Sie aber ebenso gut später in den Systemeinstellungen erledigen.

### Panel

Das einzige ständig sichtbare Bedienelement des Desktops ist das Panel, das unverrückbar am oberen Bildschirmrand angezeigt wird (siehe [Abbildung 5.1](#)). Es enthält den Button AKTIVITÄTEN, ein Icon für das gerade aktive Programm, die Uhrzeit sowie am rechten Rand diverse Status-Icons und -Menüs. Der eigentliche Arbeitsbereich ist – wenn man von eventuell offenen Fenstern einmal absieht – vollkommen leer. Die Darstellung von Icons auf dem Desktop ist nicht vorgesehen.

**Systemmenü** Die Icons ganz rechts im Panel führen in das Systemmenü. Dort können Sie die Netzwerkverbindungen konfigurieren, die Lautstärke einstellen, in einen anderen Benutzer-Account wechseln oder den Computer ausschalten.

Im Systemmenü können Sie sich auch abmelden (Logoff). Der entsprechende Eintrag ist aber gut versteckt: Sie müssen zuerst Ihren eigenen Namen anklicken – dann werden unterhalb die beiden Submenüeinträge ABMELDEN und KONTOEINSTELLUNGEN eingeblendet.

**Applikationsmenü** Einige Gnome-Programme machen ihre wichtigsten Menükommandos auch über das sogenannte »Applikationsmenü« zugänglich. In dieses Menü gelangen Sie, wenn Sie im Panel auf den Namen des gerade aktiven Programms klicken. Bei vielen Programmen, die nicht unmittelbar aus dem Gnome-Universum stammen, enthält das Menü nur den Eintrag BEENDEN.

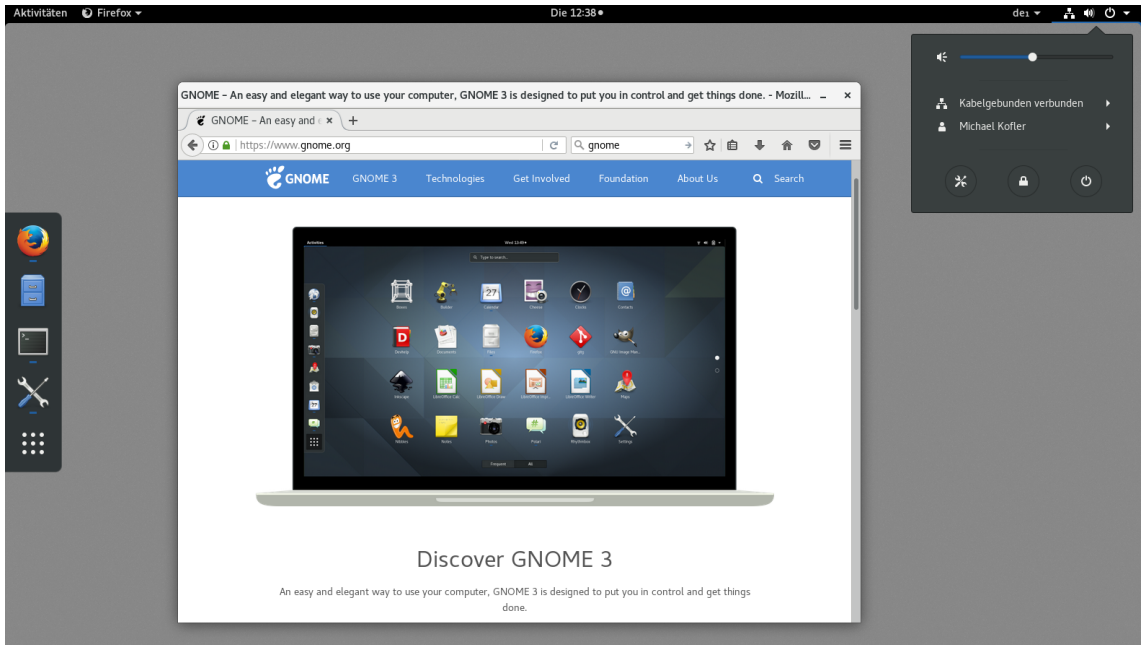



Abbildung 5.1 Der Gnome-Desktop

Die im rechten Teil des Panels angezeigten Miniprogramme (Applets) sind vorgegeben. Es gibt keinen Dialog, um weitere Applets hinzuzufügen. Sie können aber über die Seite <https://extensions.gnome.org> Erweiterungen herunterladen und aktivieren (siehe [Abschnitt 5.6](#), »Gnome-Shell-Erweiterungen«).



Applets






## Aktivitäten

Ein Mausklick auf den Button AKTIVITÄTEN, das Verschieben des Mauscurors in die linke obere Ecke des Bildschirms oder das Drücken der -Taste oder der Tasten **[Alt]+[F1]** öffnet die Aktivitäten-Ansicht. Standardmäßig zeigt diese Ansicht links ein Dock mit den Icons oft benötigter sowie aller laufenden Programme an, rechts eine Vorschau der aktiven Arbeitsflächen. Dazwischen werden in einer Art Exposé-Ansicht alle Fenster der Arbeitsfläche angezeigt. Nun können Sie beispielsweise Fenster in eine andere Arbeitsfläche verschieben, Icons von häufig benötigten Programmen in der Icon-Leiste am linken Bildschirmrand neu positionieren etc.

In der Aktivitäten-Ansicht ist ein Suchfeld aktiv. Sobald Sie per Tastatur einen Suchbegriff eingeben, ersetzt Gnome die Exposé-Ansicht aller Fenster durch die Suchergebnisse, wobei Programme, Systemeinstellungsmodule, Verzeichnisse, Kontakte sowie die zuletzt verwendeten Dateien berücksichtigt werden. Das gewünschte Objekt können Sie wahlweise mit der Maus oder mit den Cursortasten auswählen.

Suchfunktion

Die Suchfunktion ist eine ungemein praktische Sache. Wenn Sie beispielsweise rasch GIMP öffnen möchten, geben Sie einfach  gi  ein. Sobald Sie sich daran gewöhnt haben und die Anfangsbuchstaben der wichtigsten Programme auswendig kennen, gelingt der Programmstart so äußerst schnell und effizient.

Beachten Sie, dass  <name>  bereits laufende Programme aktiviert und nicht eine neue Instanz startet. Das ist meistens zweckmäßig, aber nicht immer: Wenn Sie beispielsweise nicht ein bereits laufendes Terminalfenster aktivieren möchten, sondern ein neues Terminalfenster öffnen möchten, müssen Sie + drücken bzw. das Terminal-Icon im Dock zusammen mit  anklicken.

## Dock (Dash)

In Gnome gibt es keine ständig sichtbare Task- oder Fensterleiste. Diese Rolle übernimmt die vertikale Icon-Leiste am linken Rand der Aktivitäten-Ansicht. Die Gnome-Entwickler bezeichnen sie als *Dash*, ich bleibe in diesem Buch aber bei dem gebräuchlicheren Begriff *Dock*.

Das Dock enthält im oberen Bereich standardmäßig einige Programme, von denen die Gnome-Entwickler denken, dass Sie sie häufig benötigen werden. Der untere Bereich des Docks enthält Icons aller gerade laufenden Programme, soweit sich diese nicht sowieso im Dock befinden. Laufende Programme werden hervorgehoben. Die Icon-Größe im Dock wird automatisch so angepasst, dass alle Icons angezeigt werden können. Wenn also viele Programme gleichzeitig laufen, schrumpfen die Icons entsprechend.

Um ein Icon aus dem Dock zu entfernen, führen Sie das Kontextmenükommando AUS FAVORITEN ENTFERNEN aus. Um dem Dock ein Programm hinzuzufügen, verschieben Sie das betreffende Programm per Drag & Drop aus der Ansicht ANWENDUNGEN in das Dock. Alternativ führen Sie bei einem bereits laufenden Programm das Kontextmenükommando ZU FAVORITEN HINZUFÜGEN aus.

### Dock ständig anzeigen

Unter Windows und macOS ist es üblich, dass die Taskleiste bzw. das Dock ständig sichtbar sind. In Gnome ist das nicht vorgesehen. Dieser Mangel lässt sich aber durch die Installation der Erweiterung *Dash to Dock* beheben (siehe [Abschnitt 5.6](#), »Gnome-Shell-Erweiterungen«).

## Statusbereich

Sofern im Hintergrund Programme laufen, die ihren Zustand oder andere Benachrichtigungen anzeigen, sieht Gnome im linken unteren Eck des Bildschirms ein kleines Statusfeld vor: Wenn Sie den Mauszeiger dorthin bewegen, werden die Icons



eingblendet und können dann bedient werden. Davon machen vorwiegend ältere Programme Gebrauch, die nicht speziell für Gnome optimiert sind. Dazu zählt z. B. der Dropbox-Client.

## Programme ausführen

Um ein Programm zu starten, dessen Namen Sie nicht kennen, aktivieren Sie die Aktivitäten-Ansicht und klicken auf das Icon ANWENDUNGEN ANZEIGEN unten im Dock. Damit gelangen Sie in eine Icon-Übersicht, die anfänglich die zuletzt benutzten Programme zeigt. Um zwischen allen Programmen wählen zu können, klicken Sie auf den Button ALLE. Nun werden alle installierten Programme angezeigt, wobei weniger häufig benötigte Programme in Gruppen wie HILFSPROGRAMME oder VERSCHIEDENES verborgen sind.

Programme  
starten

Alternativ können Sie in der Aktivitäten-Ansicht den Namen des gewünschten Programms auch per Tastatur eingeben. Das ist wesentlich schneller und erlaubt auch die Eingabe von Anwendungen, die noch gar nicht installiert sind. Gnome zeigt dann das Icon des Programms *Software* an, das bei der Installation der gewünschten Anwendung hilft. Dieser Mechanismus funktioniert nur für Desktop-Anwendungen, nicht für sonstige Pakete.

Gewöhnungsbedürftig ist die Bedienung von Gnome mit der Tastatur. **Alt** + **↩** wechselt nicht wie unter Windows zwischen Fenstern, sondern zwischen Programmen. Dieses Konzept verfolgt macOS schon lange, aber auch Apple hat mich nicht überzeugen können, dass das eine gute Idee ist.

Das aktive  
Programm  
wechseln

Besteht ein Programm aus mehreren Fenstern bzw. laufen mehrere Instanzen gleichzeitig (z. B. Terminal-Fenster), dann müssen Sie nun recht umständlich mit den Cursortasten das gewünschte Fenster auswählen. Dafür gibt es zwei neue Tastenkürzel: **Alt** + **Esc** wechselt zwischen allen Fenstern und **Alt** + **^** zwischen den Fenstern des gerade aktiven Programms. Und so haben wir nun *drei* Tastenkürzel, um das zu tun, was bisher mit einem Tastenkürzel wunderbar funktionierte.

In Gnome fehlen die Fensterbuttons MINIMIEREN und MAXIMIEREN. Um ein Fenster zu minimieren, klicken Sie die Fensterleiste mit der rechten Maustaste an und führen MINIMIEREN aus; um es zu maximieren, verschieben Sie es an den oberen Bildschirmrand oder doppelklicken auf die Fensterleiste. Wenn Sie sich nach »normalen« Fensterbuttons sehnen, führen Sie die entsprechende Konfiguration am besten mit dem *Gnome Tweak Tool* durch. Dieses Programm stelle ich Ihnen in [Abschnitt 5.5](#) vor.

Fenster

Wie unter Windows können Sie ein Fenster in der linken oder rechten Bildschirmhälfte platzieren, indem Sie es an den linken oder rechten Fensterrand verschieben.

**Arbeitsflächen** Arbeitsflächen ermöglichen es, die Fenster der laufenden Programme auf mehrere virtuelle Desktops zu verteilen und zwischen diesen Desktops zu wechseln. Das erleichtert die Arbeit und verbessert die Übersicht, wenn Sie sehr viele Fenster gleichzeitig öffnen. In der Aktivitätenansicht können Sie Fenster in eine zweite Arbeitsoberfläche verschieben. Sobald es zwei aktive Arbeitsflächen gibt, sieht Gnome eine dritte, vorerst leere Arbeitsfläche vor. Ganz egal, wie viele Arbeitsflächen Sie einsetzen – es gibt immer noch eine.

Für ständig benötigte Fenster besteht die Möglichkeit, diese so zu kennzeichnen, dass sie nicht auf einer, sondern auf allen Arbeitsflächen sichtbar sind. Dazu öffnen Sie mit der rechten Maustaste oder mit **Alt**+Leertaste das Fenstermenü und aktivieren die Option **IMMER AUF DER SICHTBAREN ARBEITSFLÄCHE**.

Um zwischen den Arbeitsflächen zu wechseln, können Sie die Aktivitätenansicht verwenden. Noch schneller klappt es mit den Tastenkürzeln **Strg**+**Alt**+**↑** bzw. **↓**.

**Tastenkürzel** Tabelle 5.1 fasst die wichtigsten Tastenkürzel zusammen. Weitere Tastenkürzel finden Sie in den Systemeinstellungen, Modul **TASTATUR**, Dialogblatt **TASTENKÜRZEL**.

| Tastenkürzel  | Bedeutung  |
|---|--|
| <b>Alt</b> oder <b>Alt</b> + <b>F1</b>                    | wechselt zwischen der Standardansicht und der Desktop-Übersicht (Exposé-Ansicht). In diese Ansicht gelangen Sie auch, wenn Sie die Maus in die linke obere Ecke des Fensters bewegen. Sie können nun die Tastatur zur Eingabe von Suchtexten verwenden.  |
| <b>Alt</b> + <b>F2</b>                                    | startet das Programm, dessen Namen Sie angeben.  |
| <b>Alt</b> + <b>Strg</b>                                  | wechselt zwischen Programmen (nicht Fenstern!).  |
| <b>Alt</b> + <b>Esc</b>                                   | wechselt zwischen allen Fenstern (so wie früher <b>Alt</b> + <b>Strg</b> ).  |
| <b>Alt</b> + <b>^</b>                                     | wechselt zwischen den Fenstern innerhalb des gerade aktiven Programms.   |
| <b>Strg</b> + <b>Alt</b> + <b>Strg</b>                    | bewegt in der Standardansicht den Eingabefokus in das Panel und ermöglicht so eine Bedienung der Panel-Elemente. In der Desktop-Übersicht wechselt <b>Strg</b> + <b>Alt</b> + <b>Strg</b> zwischen verschiedenen Desktop-Elementen, also dem Panel, der Seitenleiste (Dash), den Fenstern, den Arbeitsflächen etc. |
| <b>Strg</b> + <b>Alt</b> + <b>↑</b> / <b>↓</b>            | wechselt zwischen den Arbeitsflächen.  |
| <b>⇧</b> + <b>Strg</b> + <b>Alt</b> + <b>↑</b> / <b>↓</b> | verschiebt das aktuelle Fenster in die nächste Arbeitsfläche.  |

**Tabelle 5.1** Wichtige Gnome-Tastenkürzel

## 5.2 Dateimanager

Das Programm *Dateien* ist der Gnome-Dateimanager (siehe [Abbildung 5.2](#)). Es gewährt nicht nur den Zugriff auf Dateien und Verzeichnisse, sondern ermöglicht auch den Zugriff auf externe Datenträger und Netzwerkverzeichnisse. Dieser Abschnitt beschreibt die Bedienung des Dateimanagers, geht aber nicht im Detail auf die Besonderheiten der Dateiverwaltung unter Linux ein. Was Links sind, wie verborgene Dateien gekennzeichnet werden und wie Zugriffsrechte unter Linux funktionieren und viele weitere Details erfahren Sie in [Kapitel 11](#), »Dateien und Verzeichnisse«.

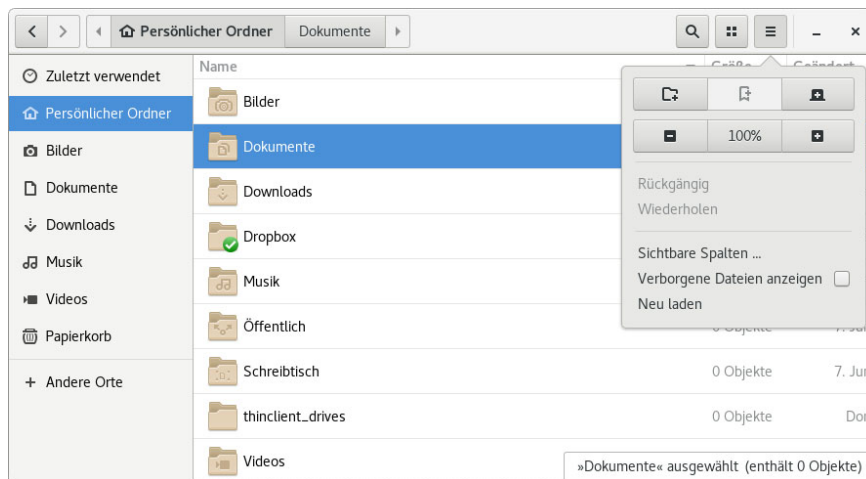


Abbildung 5.2 Der Gnome-Dateimanager

### »Nautilus« versus »Dateien«

In früheren Versionen hieß der Dateimanager *Nautilus*. Der neue Name lautet *Files* bzw. im Deutschen *Dateien*. Dieser Namenswechsel mag benutzerfreundlich gemeint sein, da sich mit der neuen Bezeichnung aber keine vernünftigen Sätze bilden lassen (»Verwenden Sie *Dateien*, um Ihre Dateien zu verwalten ...«), bleibe ich in diesem Buch beim alten Namen Nautilus oder schreibe »Dateimanager«. Auch intern ist es beim Namen Nautilus geblieben, z. B. für die Programmdatei und für den Paketnamen.

## Bedienung

Den Dateimanager starten Sie am einfachsten durch einen Klick auf dessen Icon im Dock der Aktivitätenansicht. Der Dateimanager zeigt den Inhalt des ausgewählten Verzeichnisses standardmäßig in der Symbolansicht an. Jede Datei wird durch ein

Dateien anzeigen

Icon dargestellt, das bei Bildern und einigen anderen Dateitypen gleichzeitig eine Vorschau auf den Inhalt gibt. Die Vorschau funktioniert standardmäßig nur bei lokalen Dateien bis zu 10 MiB. Damit die Vorschau auch in Netzwerkverzeichnissen sowie für größere Dateien funktioniert, verändern Sie die entsprechenden Optionen im Dialogblatt VORSCHAU der Einstellungen. In den Einstellungsdialog gelangen Sie über das Applikationsmenü im Panel.

Mit **Strg**+**1** und **Strg**+**2** können Sie zwischen der Symbolansicht und der Detailansicht wechseln. Innerhalb der Symbolansicht können Sie mit **Strg**+**+** und **Strg**+**-** die Icon-Größe einstellen.

Damit die Vorschau nicht immer wieder neu erzeugt werden muss, speichert der Dateimanager die Bilder im Verzeichnis `.cache/thumbnails`. Auch viele andere Gnome-Programme nutzen dieses Verzeichnis.

#### Verzeichnisse vor Dateien anzeigen

Standardmäßig sortiert Gnome alle Objekte alphabetisch und durchmischt dabei Dateien und Verzeichnisse. Wenn es Ihnen lieber ist, dass Gnome zuerst alle Verzeichnisse und dann alle Dateien anzeigt, aktivieren Sie in den Einstellungen die Option **ORDNER VOR DATEIEN ANZEIGEN**.

#### Ausklappbare Unterverzeichnisse

Nautilus kann in der Listenansicht die Ordner ausklappbar darstellen. Das ermöglicht eine einfachere Navigation durch den Verzeichnisbaum. Diese Funktion kann im Dialogblatt **ANZEIGE** der Programmeinstellungen aktiviert werden.

**Seitenleiste** Der linke Fensterrand enthält normalerweise eine Seitenleiste, die einen raschen Wechsel zu wichtigen Verzeichnissen ermöglicht. Per Drag & Drop können Sie oft benötigte Verzeichnisse der Seitenleiste hinzufügen und auf diese Weise Lesezeichen definieren. **F9** schaltet die Seitenleiste aus bzw. wieder ein.

**Verzeichniswechsel** In der Symbolleiste wird der Pfad zum gerade aktiven Verzeichnis durch Buttons dargestellt. Damit können Sie rasch in übergeordnete Verzeichnisse wechseln. Alternativ zeigt der Dateimanager an dieser Stelle mit **Strg**+**L** den kompletten Pfad an, was vor allem die rasche Eingabe eines anderen Verzeichnisses erleichtert.

**Reiter** Mit **Strg**+**T** öffnen Sie ein neues Dialogblatt. Besonders praktisch sind Dialogblätter, wenn Sie Dateien von einem Verzeichnis in ein anderes kopieren oder verschieben möchten: Während Drag & Drop-Operationen können Sie das aktive Dialogblatt wechseln.

**Dateien öffnen** Bei den meisten Dateitypen wird die Datei durch einen Doppelklick geöffnet. Der Dateimanager startet automatisch ein passendes Programm. Wenn der Dateityp dem Dateimanager nicht bekannt ist, klicken Sie die Datei mit der rechten Maustaste an

und führen MIT ANDERER ANWENDUNG ÖFFNEN aus. Damit gelangen Sie in einen Dialog, der die meisten auf dem Rechner installierten Programme zur Auswahl anbietet.

Bei manchen Dateien sind mehrere Programme zur Bearbeitung geeignet. Beispielsweise können Sie Bilddateien wahlweise mit einem Bildbetrachter, mit GIMP oder mit Firefox öffnen. Eines dieser Programme gilt als Standardprogramm und wird per Doppelklick gestartet. Wenn Sie das Standardprogramm ändern möchten, klicken Sie die Datei mit der rechten Maustaste an, führen EIGENSCHAFTEN • ÖFFNEN MIT aus und wählen das gewünschte Programm. Die Einstellung gilt in Zukunft für alle Dateien mit derselben Endung, also beispielsweise für alle \*.png-Dateien.

Zuvor markierte Dateien kopieren Sie mit **Strg**+**C** bzw. schneiden Sie mit **Strg**+**X** aus. Anschließend fügen Sie die betreffenden Dateien mit **Strg**+**V** am neuen Ort wieder ein. Die ausgeschnittenen Dateien werden erst jetzt am Ursprungsort entfernt.

**Dateien verschieben und kopieren**

Deutlich einfacher ist es, Dateien per Drag & Drop von einem Dateimanager-Fenster in ein zweites zu verschieben. Dabei werden die Dateien normalerweise verschoben, nicht kopiert! Eine Ausnahme von dieser Regel sind Drag&Drop-Operationen zwischen unterschiedlichen Datenträgern, also beispielsweise von der CD oder von einem Netzwerkverzeichnis in das lokale Dateisystem. Im Mauszeiger wird in solchen Fällen ein Plus-Symbol eingeblendet.

Wenn Sie eine Datei gezielt kopieren statt verschieben möchten, drücken Sie während der Drag&Drop-Operation die **Strg**-Taste. Wenn Sie den Verschiebemodus selbst angeben möchten, drücken Sie die **Alt**-Taste. Nach dem Loslassen der Maus haben Sie die Möglichkeit, die Datei zu kopieren, zu verschieben oder eine Verknüpfung (einen Link) einzurichten.

Mit dem SUCHEN-Button können Sie im Adressfeld einen Suchbegriff eingeben. Der Dateimanager liefert dann eine Liste aller Dateien, die den Suchbegriff im Dateinamen enthalten. Im Anschluss an die Suche können Sie die Suchergebnisse auf einen bestimmten Dokumenttyp oder ein Verzeichnis einschränken.

**Dateien suchen**

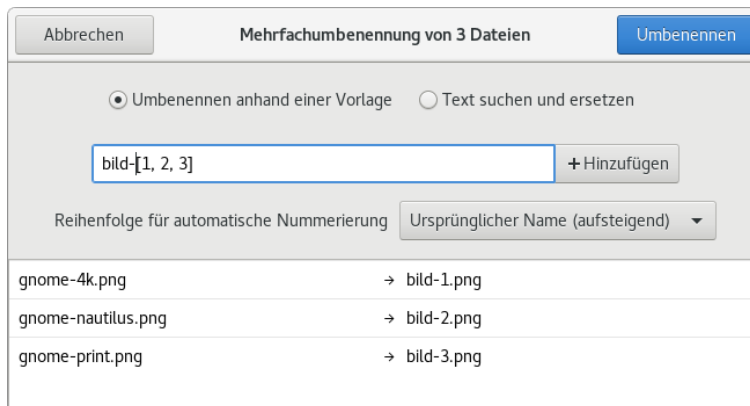
Unter Linux gelten alle Dateien und Verzeichnisse, deren Namen mit einem Punkt beginnen, als verborgen. Das bedeutet, dass sie im Dateimanager bzw. in Dateiauswahldialogen normalerweise nicht angezeigt werden. Verborgene Dateien enthalten oft Konfigurationseinstellungen oder andere Daten, die nicht direkt verändert werden sollen. Eine direkte Bearbeitung versteckter Dateien und Verzeichnisse ist nur in Ausnahmefällen zweckmäßig (z. B. wenn Sie ein Backup Ihrer E-Mail-Verzeichnisse in .thunderbird durchführen möchten). Damit solche Dateien und Verzeichnisse im Dateimanager sichtbar werden, führen Sie ANSICHT • VERBORGENE DATEIEN ANZEIGEN aus oder drücken **Strg**+**H**.

**Verborgene Dateien**

**Zugriffsrechte** Damit nicht jeder Benutzer alle Dateien und Verzeichnisse lesen bzw. verändern kann, speichert Linux zu jeder Datei und zu jedem Verzeichnis den Besitzer sowie die Zugriffsrechte. Das zugrunde liegende Konzept wird in [Abschnitt 11.5](#), »Zugriffsrechte, Benutzer und Gruppenzugehörigkeit«, ausführlich beschrieben. Um den Besitzer oder die Zugriffsrechte zu ändern, klicken Sie die Datei mit der rechten Maustaste an und führen **EIGENSCHAFTEN • ZUGRIFFSRECHTE** aus.

**Dateien löschen** Um Dateien zu löschen, müssen Sie **[Strg]+[Entf]** drücken. Die markierten Dateien und Verzeichnisse landen vorerst im Papierkorb. Den Inhalt des Papierkorbs sehen Sie durch einen Klick auf das Müll-Icon in der Seitenleiste des Dateimanagers. Erst wenn Sie den Papierkorb leeren, werden die Dateien endgültig gelöscht.

**Dateien umbenennen** Eine relativ neue und daher noch weitgehend unbekannt Funktion des Gnome-Dateimanagers ist die Möglichkeit, mehrere Dateien umzubenennen. Dazu markieren Sie die Dateien und führen dann das Kontextmenükommando **UMBENENNEN** aus (siehe [Abbildung 5.3](#)).



**Abbildung 5.3** Mehrere Dateien umbenennen

**Tastenkürzel** Nahezu alle Funktionen des Dateimanagers sind auch durch Tastenkürzel zugänglich. Zum Glück besteht kein Grund, die Kürzel auswendig zu lernen. Führen Sie stattdessen im Applikationsmenü (im Panel) das Kommando **TASTENKOMBINATIONEN** aus: Der Dateimanager zeigt dann eine gut gegliederte Übersicht aller Tastenkürzel an.

## Externe Datenträger

Beim Einlegen einer DVD bzw. beim Anstecken eines USB-Sticks oder -Laufwerks erscheint automatisch ein neues Dateimanager-Fenster mit dem Inhalt des Datenträgers. Die zugrunde liegenden Einstellungen finden Sie in den Systemeinstellungen im Modul **DETAILS • WECHSELMEDIEN**.

Denken Sie daran, dass Sie externe Festplatten oder USB-Sticks explizit abmelden müssen, bevor Sie das Kabel zum Computer lösen! Dazu klicken Sie auf den Auswerfen-Button in der Seitenleiste des Dateimanagers.

Um auf einem USB-Stick oder einer externen Festplatte ein Dateisystem einzurichten, starten Sie das Programm LAUFWERKE. Dieses Programm gibt Ihnen auch die Möglichkeit, neue Partitionen einzurichten oder den Zustand von Festplatten zu kontrollieren (siehe auch [Abbildung 22.2](#) in [Kapitel 22](#), »Administration des Dateisystems«).

Datenträger  
formatieren

Sofern das Programm *Brasero* installiert ist, können Sie von Gnome aus unkompliziert DVDs brennen: Das Fenster CD/DVD-ERSTELLER erscheint automatisch, sobald Sie einen CD- oder DVD-Rohling einlegen. Sollte das nicht funktionieren, starten Sie das Programm *Brasero* und klicken auf den Button DATEN-PROJEKT. Nun kopieren Sie von einem Dateimanager-Fenster aus die zu sichernden Dateien und Verzeichnisse per Drag & Drop in das Fenster des Brennprogramms.

CDs/DVDs  
brennen

### Zugriff auf Netzwerkverzeichnisse

Der Eintrag ANDERE ORTE in der Seitenleiste führt in eine Ansicht, die nach einigen Sekunden Icons für alle erkannten Netzwerke anzeigt. In der Praxis ist das oft nur ein WINDOWS-NETZWERK. Ein Doppelklick führt zur nächsten Ansicht mit allen erkannten Arbeitsgruppen. Ein weiterer Doppelklick zeigt alle dort sichtbaren Rechner an. Noch ein Doppelklick, und Sie wissen, welche Ressourcen dieser Rechner anbietet.

Windows-  
Freigaben

Wenn das Netzwerkverzeichnis durch ein Passwort geschützt ist, müssen Sie den Login-Namen und das Passwort angeben. Dabei bekommen Sie die Möglichkeit, diese Daten bleibend in einer Gnome-Passwortdatenbank zu speichern. Damit Sie den relativ umständlichen Weg in ein Netzwerkverzeichnis nicht immer wieder neu beschreiten müssen, richten Sie mit `[Strg]+[D]` ein Lesezeichen ein.

Wenn der Dateimanager ein Netzwerkverzeichnis ohne Passwort nutzen kann, entscheidet er sich automatisch für diese Variante. Diese Vorgehensweise ist allerdings nicht immer ideal: Je nachdem, wie der Windows- oder Samba-Server konfiguriert ist, zeigt der Dateimanager anschließend nur ein leeres Verzeichnis. Über die Benutzeroberfläche besteht nun keine Möglichkeit mehr, sich namentlich anzumelden. Abhilfe: Drücken Sie `[Strg]+[L]`, und fügen Sie Ihren Login-Namen in den Pfad ein. Die korrekte Schreibweise lautet `smb://loginname@servername/verzeichnisname`.

Sollte der Dateimanager keine Windows-Server finden, ist die wahrscheinlichste Fehlerursache eine zu restriktive Firewall zwischen Ihrem Rechner und dem Windows-Rechner. Unter CentOS, Fedora, RHEL und SUSE müssen Sie mit `firewall-config` bzw. mit YaST explizit den Dienst Samba erlauben (siehe [Abschnitt 37.4](#), »Firewall-Konfi-

gurationshilfen«). Oft funktioniert auch nur die Namensauflösung nicht. Abhilfe: Drücken Sie **[Strg]+[L]**, und geben Sie die Adresse `smb://servername` ein.

Analog können Sie auch Verbindungen zu anderen Server-Diensten herstellen. [Tabelle 5.2](#) fasst die wichtigsten Adressen bzw. Protokolle zusammen. In der Tabelle finden Sie auch die Spezialadressen `computer:` und `trash:`.

| Adresse                            | Ergebnis   |
|------------------------------------|--|
| <code>computer:</code>             | Liste aller Datenträger                                  |
| <code>afp://user@host/path</code>  | Zugriff auf AFP-Server (Apple)                           |
| <code>dav://user@host/path</code>  | Zugriff auf WebDAV-Server                                |
| <code>davs://user@host/path</code> | Zugriff auf WebDAV-Server (verschlüsselt, also HTTPS)    |
| <code>ftp://hostname</code>        | Zugriff auf FTP-Server                                   |
| <code>network:</code>              | Verwendung als allgemeiner Netzwerk-Browser              |
| <code>sftp://hostname</code>       | Zugriff auf SFTP-Server (SSH-Protokoll)                  |
| <code>smb:</code>                  | Verwendung als Windows-Netzwerk-Browser                  |
| <code>smb://hostname</code>        | Zugriff auf Netzwerkverzeichnisse eines Windows-Rechners |
| <code>trash:</code>                | Papierkorb (gelöschte Dateien)                           |

**Tabelle 5.2** Spezialadressen

**GVFS** Für den Zugriff auf Netzwerkverzeichnisse ist das *Gnome Virtual File System* (GVFS) verantwortlich. Es bindet externe Verzeichnisse als Unterverzeichnisse von `/run/user/<name>` in den Verzeichnisbaum ein. Der Dateimanager und die Dateiauswahl-dialoge zeigen externe Netzwerkverzeichnisse in der Seitenleiste an (drücken Sie gegebenenfalls **[F9]**).

## Netzwerkverzeichnisse freigeben

Gnome bietet erstaunlich viele Möglichkeiten, eigene Daten im lokalen Netzwerk zu teilen. Wirklich perfekt ist leider keine der Varianten.

**Beliebige  
Verzeichnisse per  
SMB freigeben**

SMB steht für *Server Message Block* und ist das Protokoll, das sowohl Windows als auch das Linux-Programm Samba verwenden, um Verzeichnisse im Netzwerk zu teilen. Um im Dateimanager ein eigenes Verzeichnis freizugeben, führen Sie in dessen Kontextmenü **FREIGABE IM LOKALEN NETZWERK** aus. Dieses Menükommando steht nur unter Debian, SUSE und Ubuntu zur Verfügung (Plugin `nautilus-share` aus dem gleichnamigen Paket).



Wenn Sie unter Debian oder Ubuntu die erste derartige Freigabe einrichten, wird vorher nach einer Rückfrage das Programm Samba installiert. Unter SUSE müssen Sie sich darum und um eine geeignete Firewall-Konfiguration selbst kümmern. Unter CentOS, Fedora und RHEL kann das Paket `nautilus-share` nicht installiert werden, vermutlich weil es zur SELinux-Konfiguration dieser Distributionen nicht kompatibel ist.

Freigaben ohne Passwort lassen sich mit `nautilus-share` unkompliziert einrichten. Wenn Sie das Verzeichnis dagegen mit einem Passwort absichern möchten, müssen Sie in einem Terminalfenster `smbpasswd -a ihr-login-name` ausführen und ein Passwort angeben (aber aus Sicherheitsgründen auf keinen Fall Ihr reguläres Login-Passwort!). Eine Menge Hintergrundinformationen zur manuellen Freigabe von Netzwerkverzeichnissen finden Sie in [Kapitel 28](#), »Samba«.

Im Modul FREIGABEN der Systemeinstellungen gibt es den Punkt DATEIFREIGABE (siehe [Abbildung 5.4](#)). Wenn Sie diese Art der Freigabe in einem Unterdialog aktivieren, wird das Verzeichnis öffentlich innerhalb Ihres Heimatverzeichnisses mit dem Verfahren WebDAV im lokalen Netzwerk freigegeben. Der Datenaustausch zwischen Linux-Rechnern funktioniert auf diese Art und Weise recht gut. Im Zusammenspiel mit anderen Betriebssystemen scheitert der Freigabemechanismus allerdings daran, dass andere Dateimanager die Freigabe gar nicht finden.

Öffentliches Verzeichnis per WebDAV freigeben

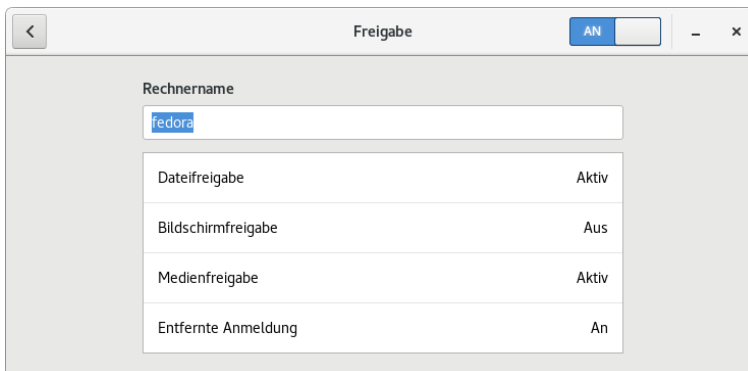


Abbildung 5.4 Freigabe-Varianten in den Gnome-Systemeinstellungen

Im Einstellungsmodul FREIGABEN finden Sie einen weiteren Eintrag MEDIENFREIGABE. Damit können Sie mit wenigen Klicks Ihre Verzeichnisse Bilder, Musik und Videos gemäß der DLNA-Richtlinien freigeben. Der *Digital Living Network Alliance* gehören diverse Hersteller von Multimedia-Geräten an – und auf derartigen Geräten können Sie die freigegebenen Dateien dann auch anzeigen oder abspielen. Aber auch manche Audio- und Video-Player sind in der Lage, die so freigegebenen Verzeichnisse zu entdecken. In einem herkömmlichen Dateimanager werden die Verzeichnisse hingegen nicht angezeigt.

Fotos, Videos und Audio-Dateien per DLNA freigeben

### Tipps aus der Praxis

Um rasch ein paar Dateien zwischen meinen Linux-Computern hin- und her zu kopieren, verwende ich in der Regel SSH. Auf jedem meiner Linux-Rechner läuft ein SSH-Server. In Nautilus können Sie unkompliziert mit `Strg+L` die Adresse `sftp://name@host` eintippen. Damit wird eine SSH-Verbindung zum angegebenen Rechner hergestellt und dessen Heimatverzeichnis in Nautilus dargestellt. Für meine Zwecke reicht das oft aus.

Für Linux-Einsteiger ist diese Variante zu kompliziert, außerdem erfordert sie zum Austausch von Daten zwischen unterschiedlichen Benutzern die Weitergabe des persönlichen Passworts. Eine unkomplizierte Alternative kann es sein, auf einem NAS-Gerät ein für alle les- und schreibbares Verzeichnis für den Datenaustausch einzurichten. Naturgemäß ist so ein Verzeichnis für sicherheitsrelevante Daten ungeeignet. Das Verfahren eignet sich aber gut, um rasch die letzten Geburtstagsfotos weiterzugeben, ohne mit einem USB-Stick zu hantieren oder den oft langsamen Umweg über die Cloud zu nehmen.

### Plugins

Der Dateimanager kann durch Plugins erweitert werden. Die meisten Distributionen stellen Pakete für einige Plugins zur Verfügung, installieren diese aber nicht standardmäßig. Suchen Sie im Paketverwaltungsprogramm Ihrer Distribution nach *nautilus*, installieren Sie die gewünschten Pakete, und loggen Sie sich dann neu in Gnome ein. Ich stelle Ihnen hier nur einige ausgewählte Pakete vor, wobei die Paketnamen für Debian und Ubuntu gelten:

- ▶ *nautilus-image-converter* und *nautilus-image-manipulator*: Die beiden Plugins ermöglichen es, Bilder per Kontextmenü zu drehen bzw. ihre Größe zu verändern.
- ▶ *nautilus-compare*: Mit dem Plugin können Sie zwei oder mehrere zuvor markierte Textdateien vergleichen. Die Unterschiede zwischen den Dateien werden grafisch im Programm *Meld* dargestellt (<http://meldmerge.org>).
- ▶ *nautilus-open-terminal*: Dieses Plugin ermöglicht es, ein Terminal-Fenster zu öffnen, indem man mit der rechten Maustaste auf den Desktop klickt.
- ▶ *nautilus-pastebin*: Mit diesem Plugin können Sie Textdateien zu einem *Pastebin*-Service hochladen.
- ▶ *seahorse-nautilus*: Das Plugin hilft dabei, die ausgewählten Dateien per Kontextmenü zu verschlüsseln.
- ▶ *nautilus-dropbox*: Das Dropbox-Plugin hilft bei der Synchronisation des Verzeichnisses Dropbox mit Ihrem Dropbox-Konto (siehe auch [Abschnitt 7.5](#), »Dropbox«).

## Zusatzprogramme

Zur Weitergabe von Dateien per E-Mail bzw. zum Anlegen von Sicherungskopien ist es oft zweckmäßig, mehrere Dateien oder den gesamten Inhalt eines Verzeichnisses zu komprimieren. Dabei hilft der sogenannte *Archivmanager*. Das Programm starten Sie üblicherweise durch einen Doppelklick auf die Archivdatei, wobei neben dem ZIP-natürlich auch das Linux-typische TAR-Format unterstützt wird.

ZIP-Dateien

Der Archivmanager zeigt das Archiv so an, als wäre es ein ganz gewöhnliches Verzeichnis. Wenn Sie rasch einen Überblick über alle Dateien bekommen möchten, führen Sie ALLE DATEIEN ANZEIGEN im Applikationsmenü aus. Damit werden gewissermaßen alle Unterverzeichnisse ausgeklappt.

Um das gesamte Archiv auszupacken, klicken Sie auf den Button ENTPACKEN. Um ein neues Archiv zu erstellen, führen Sie `[Alt]+[F2]` file-roller aus. Sie können nun einfach per Drag & Drop Dateien bzw. ganze Verzeichnisse einfügen.

Wenn Sie wissen möchten, in welchen Ihrer Verzeichnisse sich die größten Datenmengen befinden, ist das Programm *Festplattenbelegung analysieren* eine wertvolle Hilfe (Programmname baobab). Das Programm zeigt in einer anschaulichen Grafik an, welche Verzeichnisse und Unterverzeichnisse wie viele Daten enthalten (siehe [Abbildung 5.5](#)).

Festplatten-  
nutzung  
analysieren

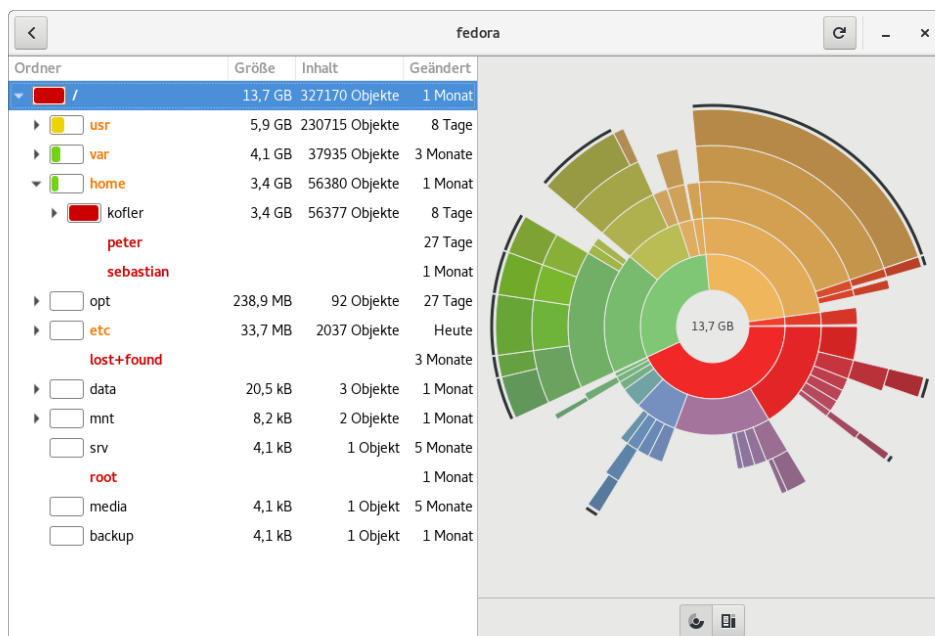
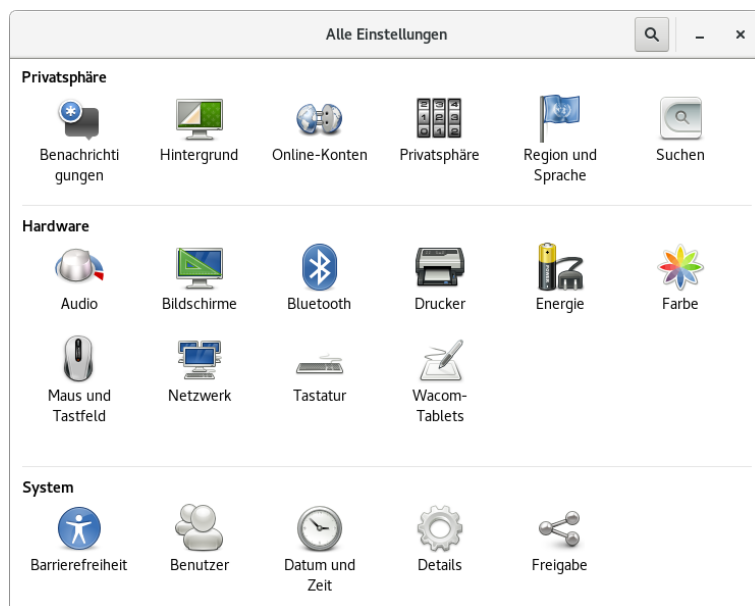


Abbildung 5.5 Platzbedarf von Verzeichnissen darstellen

## 5.3 Systemkonfiguration

Gnome ist zwar »nur« eine Desktop-Umgebung. Tatsächlich helfen Gnome-Programme und insbesondere die Systemeinstellungen (siehe [Abbildung 5.6](#)) bei vielen einfachen Administrationsaufgaben – z. B. ein Einrichten der WLAN-Verbindung oder bei der Konfiguration des Druckers. Dieser Abschnitt stellt derartige Einstellungsmodulare und Gnome-Programme vor und enthält Querverweise auf Kapitel mit weiterführenden Informationen und Details.



**Abbildung 5.6** Überblick über alle Module der Gnome-Systemeinstellungen

### Maus, Touchpad und Tastatur

**Tastatur** Die Tastatureinstellungen sind über zwei Module der Systemeinstellungen verteilt. Im Modul **TASTATUR** stellen Sie die Parameter der Tastenwiederholung sowie Tastaturkürzel ein. Einstellungen zum Tastaturlayout sind hingegen im Modul **REGION UND SPRACHE • EINGABEQUELLEN** gut versteckt. Sie können mehrere Tastaturlayouts einrichten – dann erscheint im Panel automatisch ein Icon, in dem Sie das gerade aktive Layout ändern können.

Wenn Sie die Funktion der CapsLock-Taste modifizieren, **[Alt]** und **[Strg]** vertauschen möchten oder andere Sonderwünsche haben, starten Sie das Programm `gnome-tweak-tool`. (Bei manchen Distributionen müssen Sie zuerst das gleichnamigen Paket installieren.) Im Modul **EINGABE** finden Sie unzählige Steuerungsoptionen (siehe [Abbildung 5.7](#)).

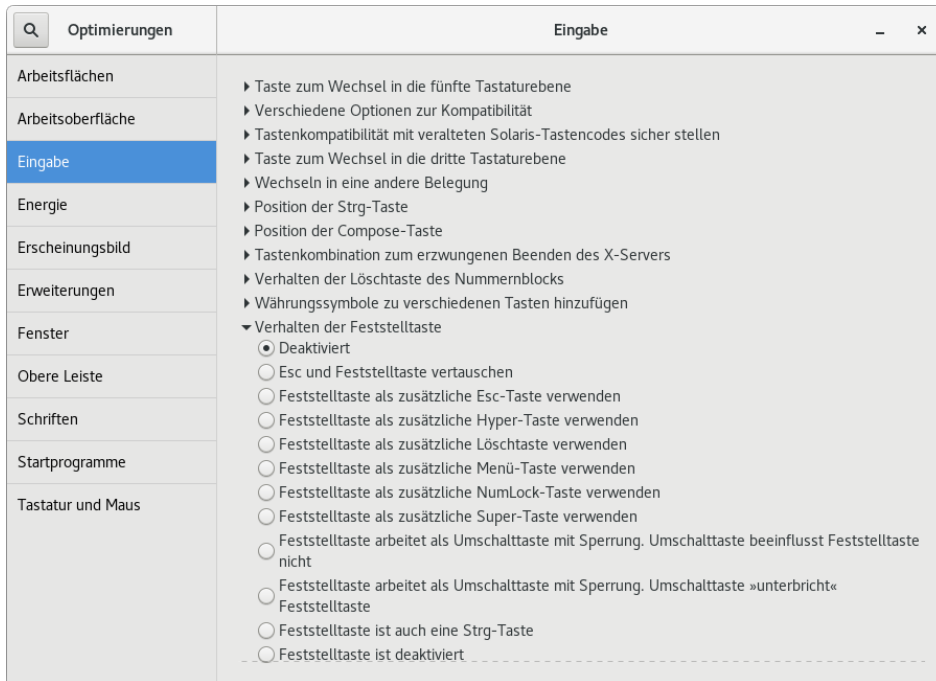


Abbildung 5.7 Funktion von Sondertasten modifizieren

Nur wenige Einstellmöglichkeiten bietet das Modul MAUS UND TASTFELD: Dort können Sie die Funktion der linken und rechten Maustaste vertauschen, die Geschwindigkeit der Maus bzw. des Trackpads einstellen und den sogenannten NATÜRLICHEN BILDLAUF aktivieren. Damit verschieben Sie auf dem Trackpad den Fensterinhalt und nicht wie bisher die Bildlaufleiste.

Maus und  
Touchpad

In der Regel ist es wünschenswert, das Touchpad eines Notebooks zu deaktivieren sobald eine Maus zur Verfügung steht. Gnome sieht dafür leider keine entsprechende Option vor, aber Sie erhalten diese Funktionalität durch die Installation des *Touchpad Indicators*:

Touchpad  
deaktivieren

<https://extensions.gnome.org/extension/131>

Der Touchpad Indicator ist eine Gnome-Shell-Erweiterung. Wie Sie derartige Erweiterungen installieren, ist in [Abschnitt 5.6](#) beschrieben. Nach der Installation müssen Sie je nach Distribution in den Einstellungen der Erweiterung die Umschaltmethode von GCONF SETTINGS auf XINPUT umstellen.

## Netzwerk

Wenn Ihr Rechner über ein Ethernet-Kabel mit einem (ADSL-)Router verbunden ist, stellt Linux die Netzwerkverbindung automatisch selbst her. Aber auch die WLAN-Konfiguration ist in den meisten Fällen denkbar einfach: Sie wählen im Systemmenü das gewünschte Funknetz aus und geben einmalig das Passwort ein. Linux merkt sich das Passwort und stellt in Zukunft die Netzwerkverbindung automatisch her, sobald sich Ihr Gerät in Funkreichweite des WLAN-Routers befindet. Wenn Sie ein WLAN-Passwort ändern müssen oder spezielle (VPN-)Konfigurationswünsche haben, finden Sie entsprechende Einstellmöglichkeiten im Modul NETZWERK der Systemeinstellungen.

**NetworkManager** Hinter den Kulissen ist für die Netzwerkkonfiguration der *NetworkManager* zuständig. Dieses Programm wird von nahezu allen Linux-Desktop-Distributionen verwendet. (Zu den Ausnahmen zählt Raspbian – siehe [Abschnitt 8.2](#), »Raspbian installieren und konfigurieren«.)

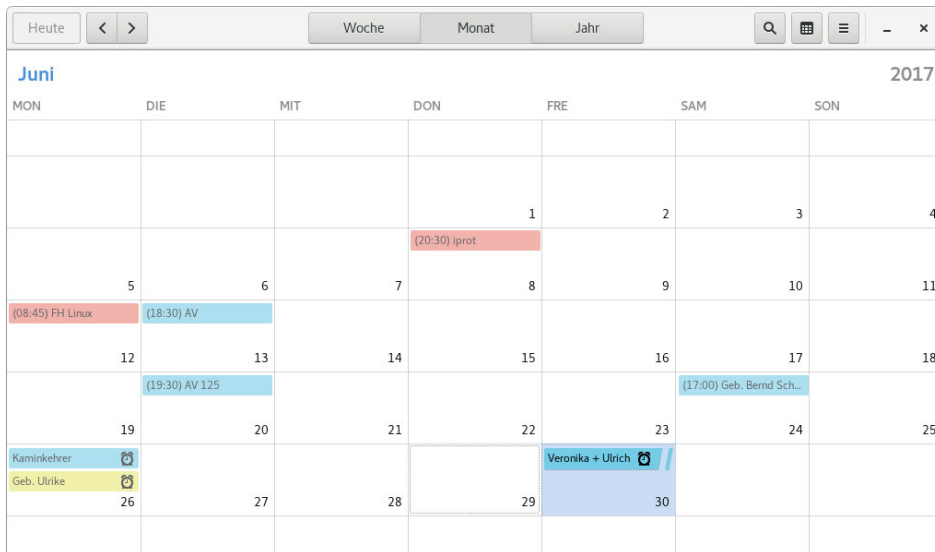
Den NetworkManager stelle ich Ihnen in [Kapitel 26](#), »Netzwerkkonfiguration«, näher vor. Dort finden Sie auch umfassende Erklärungen zu diversen Fachbegriffen und eine ausführliche Diskussion der Linux-Netzwerkinterna.

**Online-Konten** Im Modul ONLINE-KONTEN können Sie die Login-Parameter diverser Online-Dienste angeben, darunter Google, Facebook, Nextcloud und Microsoft. Die Konten können dann in anderen Gnome-Anwendungen verwendet werden, z. B. in den Programmen *Kontakte*, *Kalender* (siehe [Abbildung 5.8](#)) und *Evolution*. Bei meinen Tests hat sich das Programm *Kontakte* allerdings als recht absturzfremdig erwiesen.

## Drucker

Im Idealfall erfolgt die Druckerkonfiguration automatisch: Gnome versucht, USB-, Netzwerk- und WLAN-Drucker automatisch zu erkennen und führt die Konfiguration dann selbstständig durch. Wenige Sekunden, nachdem der Drucker angesteckt oder im Netzwerk gefunden wurde, ist das Gerät bereit zum Drucken. Bequemer geht es nicht mehr!

Leider funktioniert dieser Mechanismus nur bei relativ wenigen Druckermodellen. Im Regelfall ist Handarbeit erforderlich. Dazu öffnen Sie das Modul DRUCKER der Systemeinstellungen, klicken auf den Button ENTSPERREN und geben Ihr Passwort bzw. das root-Passwort an (Debian, SUSE). Wenn die Druckerkonfiguration länger als ein paar Minuten dauert, müssen Sie die Passwortheingabe wiederholen, was einigermaßen lästig ist.



**Abbildung 5.8** Das Programm »Kalender« mit Terminen aus einem Nextcloud-Konto

HINZUFÜGEN startet nun die Suche nach Druckern. Es kann sein, dass Ihr Drucker in der Liste der gefundenen Drucker mehrfach aufscheint. In diesem Fall gibt es verschiedene Möglichkeiten (Treiber), um den Drucker anzusprechen. Nach der Auswahl des Modells wird der Drucker eingerichtet. Unter Umständen versucht Gnome, einen genau zu Ihrem Druckermodell passenden Treiber zu installieren. Selbst wenn das scheitert, kann der Drucker oft unter Zuhilfenahme eines generischen Treibers oder eines Treibers eines anderen Modells genutzt werden. Das Kommando `DRUCKER-DETAILS` führt in einen weiteren Dialog, in dem Sie den gewünschten Treiber aus einer riesigen Liste auswählen können (siehe [Abbildung 5.9](#)). Bei vielen Laser-Druckern funktionieren die generischen Treiber für PCL- oder PostScript-Modelle.

Das `DRUCKER`-Modul der Gnome-Einstellungen hat sich in den vergangenen Jahren beinahe mit jeder Gnome-Version geändert, leider nicht immer zum Besseren. Wenn Ihnen damit die Konfiguration nicht gelingt, können Sie es alternativ mit dem alten Programm `system-config-printer` versuchen. Es wartet bei manchen Distributionen im gleichnamigen Paket auf seine Installation und Anwendung. Das Programm wird aber nicht mehr gewartet und wird daher längerfristig verschwinden.

`system-config-printer`

Eine Menge weitere Details und Interna zum Drucksystem von Linux, das auf dem auch unter macOS genutzten *Common Unix Printing System* (CUPS) basiert, finden Sie in [Kapitel 30](#). Dort zeige ich Ihnen einen weiteren Konfigurationsweg über die CUPS-Weboberfläche.

CUPS

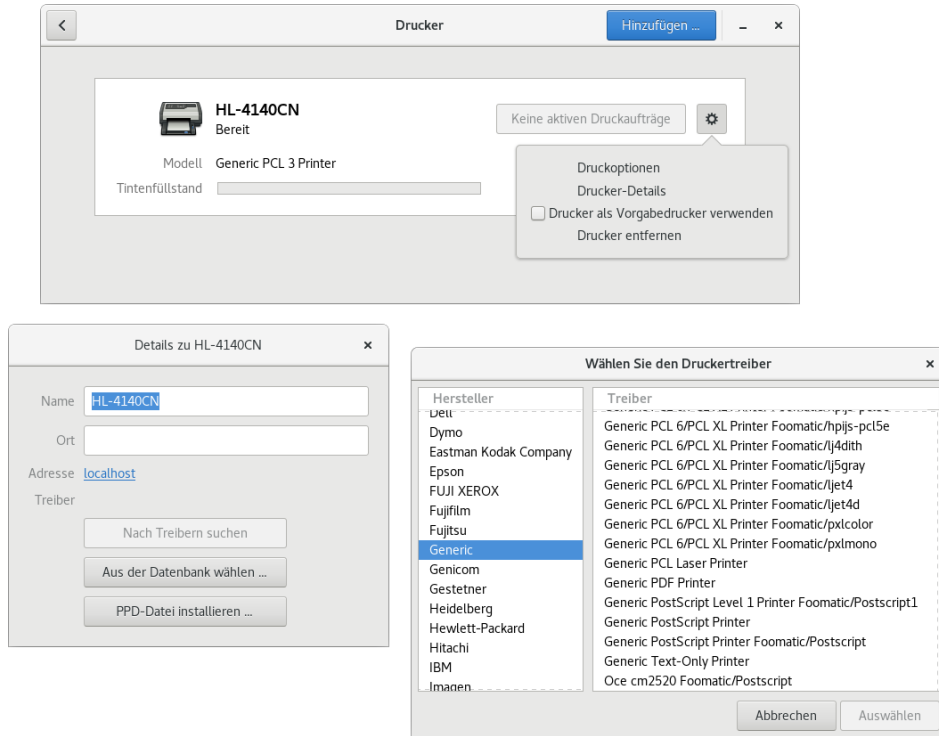


Abbildung 5.9 Druckerkonfiguration unter Gnome

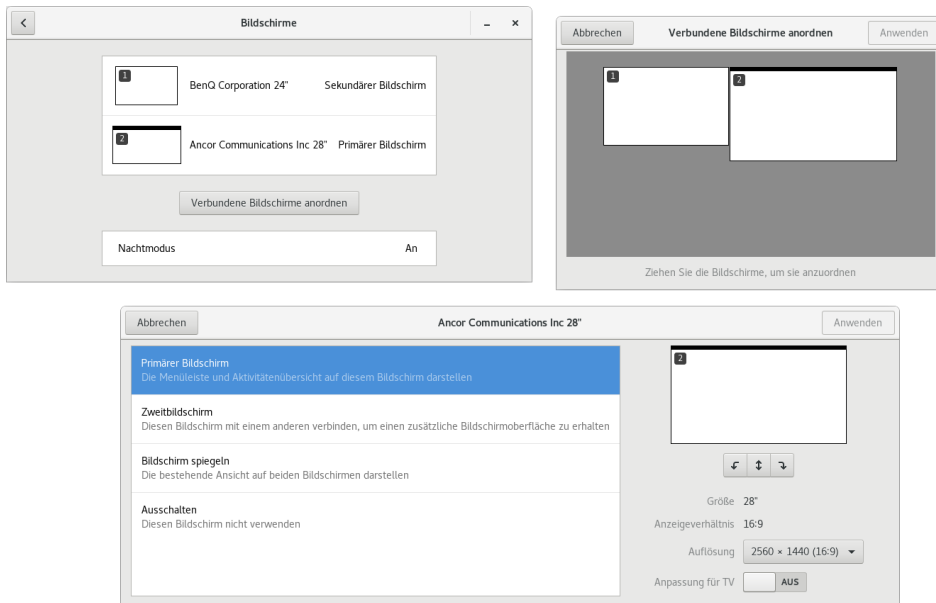
## Dual-Screen-, Beamer- und 4k-Konfiguration

Im Modul BILDSCHIRME können Sie die gewünschte Bildschirmauflösung verändern, wenn Sie mit der Defaultauflösung nicht einverstanden sind. Falls der Bildschirm dann schwarz wird, drücken Sie einfach **[Esc]** oder warten 15 Sekunden – dann wird die zuletzt gültige Konfiguration wiederhergestellt.

Außerdem ist hier der richtige Ort, um einen zweiten Bildschirm oder einen Beamer einzurichten. Der Konfiguration ist nicht ganz intuitiv über drei Dialoge verteilt (siehe [Abbildung 5.10](#)): Ein Klick auf das jeweilige Anzeigegerät führt in einen Monitor-spezifischen Dialog, in dem Sie die Auflösung, Rotation und den Modus auswählen können. BILDSCHIRM SPIEGELN ist vor allem bei Vorträgen wichtig, wenn der Beamer das gleiche Bild zeigen soll wie der Bildschirm Ihres Notebooks. Die Auflösung wird dabei automatisch so reduziert, dass sie für beide Geräte passt.

Wenn Sie dagegen beide Bildschirme unterschiedlich nutzen möchten, müssen Sie einen davon als PRIMÄREN BILDSCHIRM markieren. Das ist der Bildschirm, auf dem Gnome das Panel und das Dock anzeigt.





**Abbildung 5.10** Die Optionen zur Konfiguration zweier Bildschirme sind über drei Dialoge verteilt.

Im Gegensatz zu anderen Desktop-Systemen ist Gnome leider nicht in der Lage, das Panel auf allen Bildschirmen anzuzeigen. Welcher Bildschirm links, rechts, ober- oder unterhalb vom anderen platziert ist, stellen Sie im Dialog **VERBUNDENE BILDSCHIRME ANORDNEN** ein.

Gnome speichert die Einstellungen bildschirmspezifisch in der Datei `.config/monitors.xml` und aktiviert sie automatisch, wenn Sie Ihren Computer wieder an das gleiche Gerät anschließen. Eine Menge Hintergrundinformationen zur Funktionsweise des Grafiksystems folgen in [Kapitel 21](#).

### Kompatibilitätsprobleme

Wenn Sie mit beiden Grafiksystemen experimentieren, kann `.config/monitors.xml` zu Inkompatibilitäten führen. Wenn die Änderung der Grafikauflösung nach einem Wechsel zwischen X und Wayland plötzlich nicht mehr funktioniert, löschen Sie einfach die Datei `monitors.xml` und wiederholen die Konfiguration. Generell funktionierte die Änderung der Auflösung mit X als Grafiksystem wesentlich stabiler als mit Wayland.

**Nachtmodus** Erst seit Version 3.24 gibt es die Option NACHTMODUS. Wenn dieser Modus aktiv ist, wird nach Sonnenuntergang der Blauanteil reduziert. Das macht das Monitorbild wesentlich augenfreundlicher. (Merkwürdigerweise funktioniert die Farbanpassung nicht für den Mauszeiger, der dann blauweiß heraussticht.)

Bei älteren Gnome-Versionen können Sie einen ähnlichen Effekt durch die Installation der Programme *Redshift* oder *f.lux* erreichen. Diese Programme sind aktuell aber nicht Wayland-kompatibel. Sie funktionieren unter Wayland nur mit einer gepatchten Version des Programms `gnome-settings-daemon`.

<http://jonls.dk/redshift>

<https://extensions.gnome.org/extension/685/redshift>

<https://justgetflux.com/linux.html>

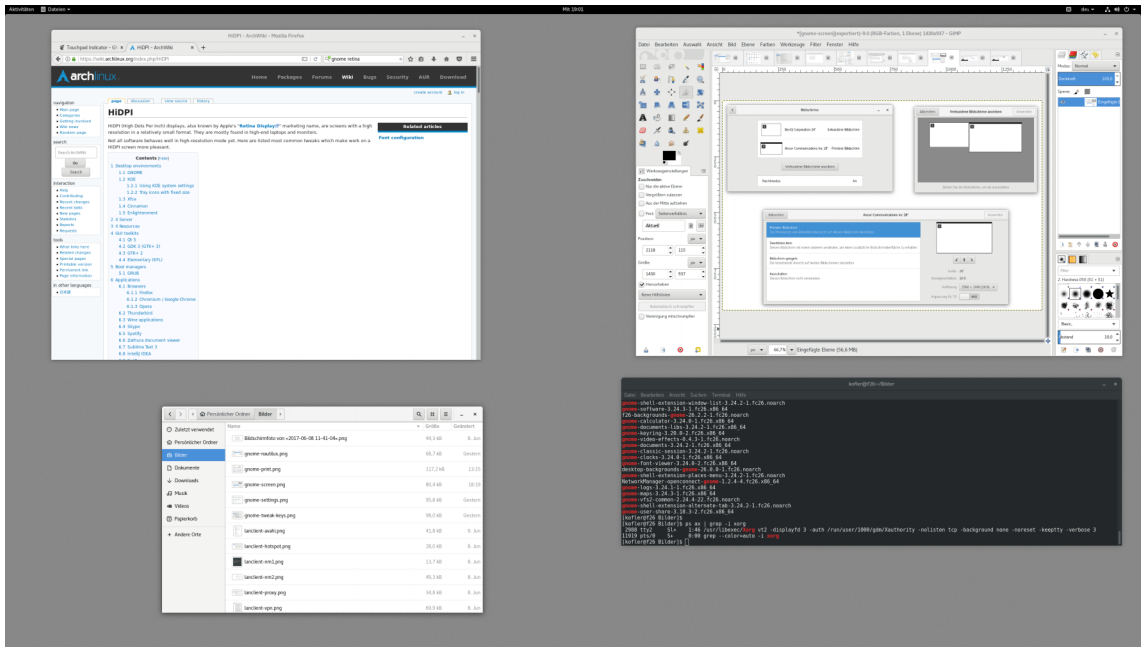
<http://www.thelinuxrain.com/articles/redshift-functionality-on-fedora-25-gnome-wayland-yes-its-possible>

**Farbprofile** Grafiker werden um den Nachtmodus und ähnliche Werkzeuge einen großen Bogen machen. Ihnen geht es ja gerade darum, dass Farben möglichst realitätsnah auf dem Monitor angezeigt werden. Farbprofilen legen Sie mit dem Einstellungsmodul FARBEN fest.

**High-DPI-Bildschirme** Bildschirme in besonders hoher Auflösung (also High-DPI-, 4k- oder im Apple-Sprech Retina-Monitore) bereiten Gnome aktuell, d. h. zumindest bis einschließlich Version 3.24 riesige Probleme: Im Einstellungsmodul BILDSCHIRME fehlt für solche Fälle jegliche Konfigurationsmöglichkeit. Daraus resultieren winzige Icons und Fenster sowie unleserlich kleine Schrift (siehe [Abbildung 5.11](#)).

Die Gnome-Entwickler arbeiten daran, die Konfiguration für High-DPI-Bildschirme zu verbessern. Bis es brauchbare Lösungen gibt, müssen Sie das *Optimierungswerkzeug* (Programm- und Paketname `gnome-tweak-tool`) zu Hilfe nehmen. Es bietet aktuell zwei Einstellungsmöglichkeiten:

- ▶ **Fensterskalierung:** Im Dialogblatt FENSTER können Sie eine ganze Zahl zur Skalierung der Fenster und aller anderen Elemente der Benutzeroberfläche angeben. Leider sind hier keine Zwischenwerte möglich.
- ▶ **Schriftskalierung:** Alternativ oder in Kombination mit dem obigen Parameter können Sie im Dialogblatt SCHRIFTEN den Skalierungsfaktor für die Anzeige von Schriften durch Gnome-Programme verändern. Dieser Faktor ist als Fließkommazahl realisiert, sodass z. B. eine Vergrößerung um den Faktor 1,66 möglich ist. Allerdings gilt der Parameter nicht so allumfassend wie der zur Fensterskalierung. Vielfach werden daher zwar die Schriften korrekt vergrößert, andere Bedienelemente bleiben aber klein.



**Abbildung 5.11** Ohne High-DPI-Konfiguration haben Sie zwar viel Platz auf dem Bildschirm, brauchen aber eine Lupe zum Arbeiten.

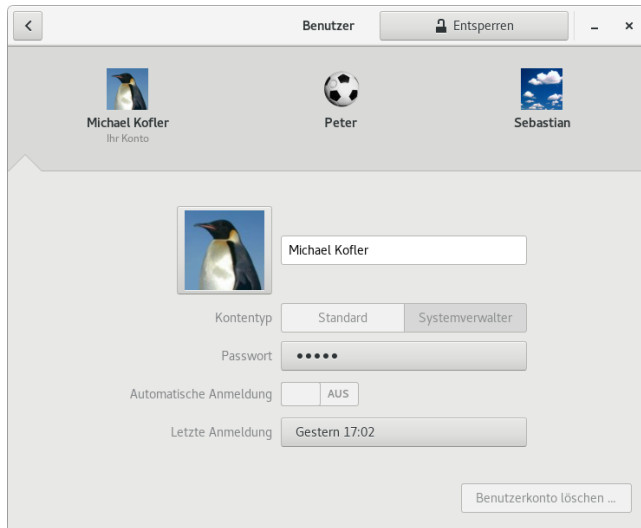
Bei meinen Tests funktionierte keine der beiden Varianten überzeugend. Das liegt vor allem daran, dass diverse Programme bzw. Programmkomponenten auf die Einstellungen unterschiedlich berücksichtigen. Daraus ergibt sich dann ein uneinheitlicher Gesamteindruck: Die Größe von Icons und Schrift passt nur selten zusammen. Auch der Mauszeiger änderte je nach dargestelltem Symbol seine Größe. Meine Notlösung bestand oft darin, Gnome auf meinem 4k-Monitor ohne Veränderung der beiden obigen Parameter, dafür aber in einer reduzierten Auflösung von  $2560 \times 1440$  Pixel zu nutzen. Das hat natürlich eine schlechtere Bildqualität zur Folge und führt die Idee eines hochauflösenden Bildschirms ad absurdum.

## Benutzerverwaltung

Im Systemmenü führt ein Klick auf Ihren Namen in ein Untermenü mit den Einträgen **ABMELDEN** und **KONTOEINSTELLUNGEN**. Der zweite Eintrag startet das Modul **BENUTZER** der Systemeinstellungen. Dort können Sie Ihr Passwort ändern und Ihrem Konto durch einen Klick auf Ihr Icon ein Bild zuordnen. Dabei können Sie entweder aus einigen Standardbildern wählen oder aber ein Foto aus Ihrem Verzeichnis **Bilder** nutzen.

**Account-Foto und -Passwort**

**Mehrere Benutzer** Das Modul **BENUTZER** bietet auch die Möglichkeit, weitere Accounts einzurichten, z. B. für Familienmitglieder. Dazu müssen Sie das Modul zuerst durch die Angabe Ihres Passworts bzw. des root-Passworts (Debian, SUSE) entsperren. Anschließend können Sie neue Benutzer hinzufügen oder das Passwort bzw. die Rechte anderer Benutzer ändern (siehe [Abbildung 5.12](#)).



**Abbildung 5.12** Benutzerverwaltung

Als **SYSTEMVERWALTER** gelten Benutzer, die durch Angabe Ihres eigenen Passworts mittels `sudo` Administratorrechte erlangen können. Details zu diesem Mechanismus sind in [Abschnitt 12.3](#), »Prozesse unter einer anderen Identität ausführen (sudo)«, beschrieben.

Beim Einrichten neuer Benutzer bietet Gnome die Option **BEI DER NÄCHSTEN ANMELDUNG PASSWORT WÄHLEN**. Damit hat der Benutzer vorerst kein Passwort; er oder sie muss dieses beim ersten Login festlegen. Das erspart die oft umständliche und unsichere Passwortweitergabe.

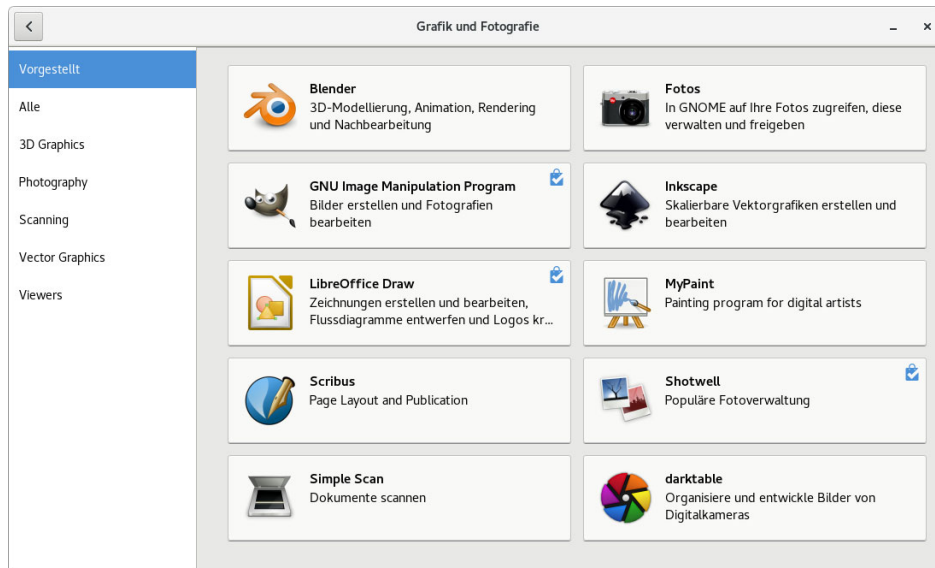
**Interna** Umfassende Informationen über die Benutzer- und Gruppenverwaltung unter Linux folgen in [Abschnitt 18.5](#), »Benutzer und Gruppen, Passwörter«. Das Gnome-Modul **BENUTZER** ist zwar einfach zu bedienen, bildet aber nur einen Bruchteil der Möglichkeiten von Linux ab.

## Software-Installation und -Updates

Gnome weist alle paar Tage darauf hin, dass zu den installierten Programmen Updates verfügbar sind. Ein Klick auf den Hinweis startet das Programm *Software* und gibt Ihnen die Möglichkeit, die Updates sofort durchzuführen. Manche Distribu-

tionen (z. B. Fedora) können Updates auf Wunsch auch im Rahmen eines Neustarts durchführen. Anders als unter Windows ist ein Neustart aber nur in Ausnahmefällen erforderlich (speziell bei Kernel-Updates).

Mit *Software* können Sie auch nach weiteren für Ihre Distribution vorgesehenen Programmen suchen und diese installieren (siehe [Abbildung 5.13](#)). *Software* wird schließlich auch dann gestartet, wenn Sie in einem Webbrowser ein für Ihre Distribution geeignetes Paket anklicken.



**Abbildung 5.13** Auf der Suche nach tollen Apps im Gnome-Programm »Software«

Fortgeschrittene Linux-Benutzer ziehen es zumeist vor, die Software-Verwaltung durch Kommandos in einem Terminalfenster durchzuführen. Eine ausführliche Vorstellung der erforderlichen Kommandos, die je nach Distribution variieren, gibt das [Kapitel 19](#), »Software- und Paketverwaltung«.

**Software-  
Verwaltung per  
Kommando**

## Fernwartung

Sofern Sie Gnome unter X ausführen, finden Sie als Hilfesuchender im Einstellungsmodul FREIGABE den Button BILDSCHIRMFREIGABE. Er führt in einen weiteren Dialog, in dem Sie Ihren Bildschirm zur Fernsteuerung freigeben können (siehe [Abbildung 5.14](#)).

Der Helfer muss nun einen VNC-Client starten. (VNC steht für *Virtual Network Computing*.) Unter Linux gebräuchliche VNC-Clients sind Vinagre und Remmina (beide Gnome), krdc (KDE), TightVNC sowie TigerVNC (Paket `tigervnc-viewer`, zugehöriges

Kommando `vncviewer`). Standardmäßig ist in der Regel keines dieser Programme installiert.

### Fernwartung nur im lokalen Netzwerk

VNC und alle darauf basierenden Programme sind nicht nur recht umständlich einzurichten, sie leiden auch unter einer wesentlichen Einschränkung: Die Fernwartung funktioniert nur in lokalen Netzwerken gut. Via Internet gibt es aber oft Probleme: Bei Privatanwendern erfolgt die Internetverbindung über einen Router bzw. mittels NAT (Network Address Translation). Das führt dazu, dass der Rechner keine öffentliche IP-Adresse hat, sondern eine IP-Adresse in einem privaten IP-Adressraum. Eine Fernwartung via VNC ist dann leider unmöglich.

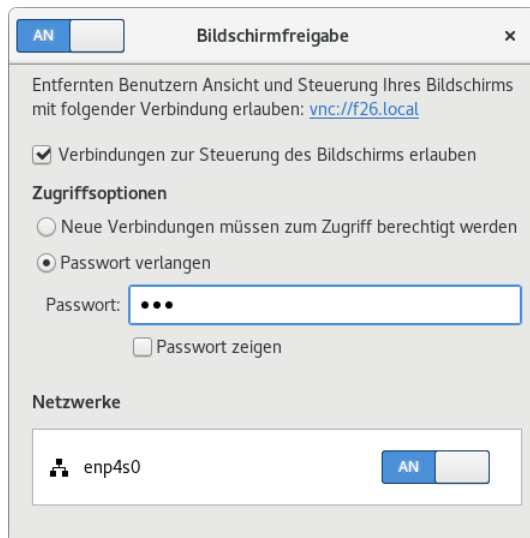


Abbildung 5.14 Bildschirmfreigabe zur Fernwartung einrichten

**TeamViewer** Eine attraktive Alternative zur Gnome-Bildschirmfreigabe ist das kommerzielle Programm TeamViewer. Es stellt die Verbindung im Zusammenspiel mit einem Server im Internet her und kann so die Schranken des lokalen Netzwerks überwinden. TeamViewer steht für Linux als RPM-, Debian- oder TAR-Paket zur Verfügung. Die private Nutzung ist kostenlos.

<https://www.teamviewer.com/de/download/linux>

**Wayland** Wenn als Grafiksystem Wayland läuft, ist keine Fernwartung möglich, weder mit Gnome-Tools noch mit externen Werkzeugen wie dem TeamViewer. Das liegt daran, dass Wayland aktuell kein Netzwerkprotokoll zur Übertragung des Bildschirminhalts und zur Fernbedienung vorsieht.

## 5.4 Schriften (Fonts)

Linux kommt grundsätzlich mit allen gängigen Font-Dateien zurecht, also mit TrueType-, Type-1- und OpenType-Schriften. Standardmäßig ist die Auswahl an Schriften zumeist überschaubar. Wenn Sie weitere Fonts installieren möchten, haben Sie diverse Möglichkeiten:

Fonts installieren

- ▶ Sie können mit den Paketverwaltungswerkzeugen Ihrer Distribution nach Font-Paketen suchen. Das ist aber nicht ganz einfach: Es gibt zwar unzählige Font-Pakete, viele davon sind aber nur für Spezialprogramme wie  $\text{\LaTeX}$  gedacht.
- ▶ Sie können eigene Font-Dateien bzw. im Internet frei verfügbare Fonts in das Verzeichnis `.fonts` kopieren. Die Fonts können dann ohne weitere Konfiguration in allen Programmen verwendet werden. (Eventuell müssen Sie sich vorher einmal aus- und neu einloggen.)

Standardmäßig existiert das `.fonts`-Verzeichnis oft nicht. Das lässt sich rasch ändern, indem Sie ein Terminalfenster öffnen und dort `mkdir .fonts` ausführen. Im Gnome-Dateimanager werden Sie das Verzeichnis auch jetzt nicht sehen, weil es als verborgenes Verzeichnis gilt. Drücken Sie `[Strg]+[H]`, um derartige Verzeichnisse einzublenden.

Microsoft bot einige Zeit lang mehrere TrueType-Fonts zum Download an (Andale Mono, Arial, Comic Sans etc.). Die Fonts sollten es allen Anwendern ermöglichen, Webseiten, in denen Microsoft-Fonts eingesetzt werden, in optimaler Qualität zu betrachten. Die ursprüngliche Download-Website gibt es zwar nicht mehr, die Fonts können nun aber von der unten angegebenen `corefonts`-Website heruntergeladen werden. Die Fonts dürfen kostenlos genutzt werden, die kommerzielle Weitergabe ist aber untersagt. Daher werden die Fonts bei kommerziellen Distributionen nicht mitgeliefert.

Microsoft-  
Internet-Fonts

Leider ist die Installation der Fonts unter Linux umständlich, weil die Fonts in `*.exe`-Dateien verpackt sind und nicht in einer anderen Form weitergegeben werden dürfen. Eine ausführliche Installationsanleitung für Distributionen mit RPM-Paketen finden Sie hier:

<http://corefonts.sourceforge.net>

Je nach Distribution gibt es Scripts, die beim Download und der Installation der Schriften helfen:

- ▶ Debian, Ubuntu: Das Paket `msttcorefonts` enthält das Script `update-ms-fonts`. Es installiert die Fonts in das Verzeichnis `/usr/share/fonts/truetype/msttcorefonts`.
- ▶ SUSE: Das Paket `fetchmsttfonts` enthält ein Script zum Download der Schriften. Sie finden die Font-Dateien danach im Verzeichnis `/usr/share/fonts/truetype`.

**Fonts anzeigen und auflisten**

Das Programm mit dem sperrigen Namen *Schriftartenbetrachtung* zeigt alle verfügbaren Fonts an (siehe [Abbildung 5.15](#)). Ein Klick auf eines der Icons zeigt diverse Mustertexte in dieser Schrift. Alternativ können Sie eine Liste aller Fonts auch in einem Terminalfenster mit dem Kommando `fc-list` ermitteln.



Abbildung 5.15 Überblick über die installierten Schriften

**Zeichentabelle**

Das Programm *Zeichentabelle* (interner Name `gucharmap`) zeigt alle Zeichen eines zuvor ausgewählten Fonts an und ermöglicht es, einzelne Sonderzeichen in die Zwischenablage zu kopieren (siehe [Abbildung 5.16](#)).



Abbildung 5.16 Suche nach Sonderzeichen einer Schrift

**Desktop-Fonts festlegen**

Welche Schrift in welcher Größe Gnome zur Darstellung der Desktop-Elemente, Menüs und Fenstertitel verwenden soll, können Sie im Gnome Tweak Tool einstellen (siehe den folgenden Abschnitt).

## 5.5 Gnome Tweak Tool

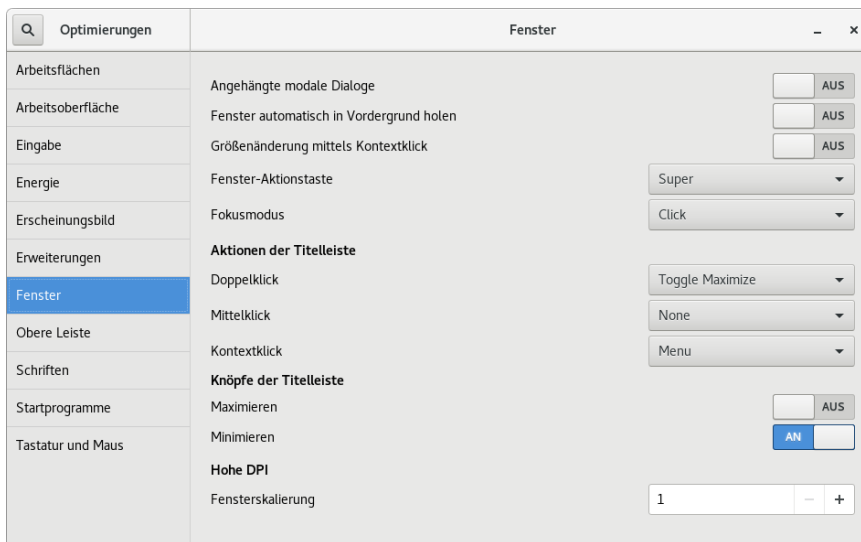
**Bildschirmhintergrund**

Die Gnome-Entwickler sind der Meinung, dass es nicht sinnvoll ist, Anwendern allzu viele Möglichkeiten zur Desktop-Konfiguration zu geben. Tatsächlich können Sie in den Systemeinstellungen gerade einmal den Bildschirmhintergrund ändern. (Noch einfacher ist es zumeist, im Dateimanager eine Bilddatei anzuklicken und das Kontextmenükommando `ALS HINTERGRUND FESTLEGEN` auszuführen.)



Viele Linux-Fans, die traditionell mit Linux auch die Freiheit verbinden, den Desktop nach eigenen Vorstellungen zu gestalten, waren deswegen von den ersten Versionen von Gnome 3 total frustriert. Mittlerweile hat sich das Bild aber gewandelt. Zwar hat sich an den Konfigurationsmöglichkeiten in den offiziellen Systemeinstellungen nichts geändert, dafür schafft aber das beliebte *Gnome Tweak Tool* Abhilfe (siehe [Abbildung 5.17](#)). Dieses Programm können Sie auch unter dem deutschen Namen *Optimierungswerkzeug* starten. Wenn Gnome das Programm nicht findet, müssen Sie es zuerst installieren. Der Paketname lautet `gnome-tweak-tool`.

Gnome Tweak  
Tool



**Abbildung 5.17** Das Gnome Tweak Tool bietet unzählige Konfigurationsmöglichkeiten.

Mit dem Programm können Sie unter anderem einstellen,

- ▶ welche Buttons in der Fensterleiste dargestellt werden sollen (auf Wunsch also auch die Buttons zum Minimieren und Maximieren des Fensters, die standardmäßig fehlen),
- ▶ ob die Fenster-Buttons rechts oder links in der Titelleiste angezeigt werden sollen (diese Option ist erst seit Sommer 2017 verfügbar),
- ▶ wie sich Gnome bei einem Doppelklick auf die Fensterleiste verhalten soll,
- ▶ welche Funktionen Sondertasten wie CapsLock- oder die Windows-Taste haben sollen,
- ▶ welche Schriften in welcher Größe auf dem Desktop verwendet werden sollen,
- ▶ mit welchen Kantenglättungsverfahren Schriften angezeigt werden sollen (Anti-Aliasing, Hinting),

- ▶ ob die Fenster bzw. die Schriften für hochauflösende Monitore skaliert werden sollen (siehe Abschnitt 5.3, »Systemkonfiguration«),
- ▶ ob Gnome auf Animationen verzichten soll,
- ▶ welche Themes und Icons zur Darstellung des Desktops verwendet werden sollen (siehe Abschnitt 5.7, »Gnome Shell Themes«),
- ▶ wie sich Notebooks beim Schließen des Deckels verhalten sollen,
- ▶ ob der Dateimanager auf dem Desktop Icons darstellen darf und
- ▶ ob zusammen mit der Uhrzeit auch das Datum angezeigt werden soll.

## 5.6 Gnome-Shell-Erweiterungen

Die Gnome Shell ist das Programm, das hinter den Kulissen für die Verwaltung der Fenster und für die Darstellung des Panels und des Docks zuständig ist. Dieses Programm greift stark auf JavaScript zurück. Deswegen ist es möglich, mit wenigen Zeilen JavaScript-Code umfassende Modifikationen am Desktop durchzuführen. Gnome sieht hierfür einen speziellen Extensions-Mechanismus vor. Die Erweiterungen können unkompliziert im Webbrowser aktiviert und bei Missfallen auch gleich wieder deaktiviert werden (siehe Abbildung 5.18). Die folgende Website lädt wirklich zum Ausprobieren ein!

<https://extensions.gnome.org>

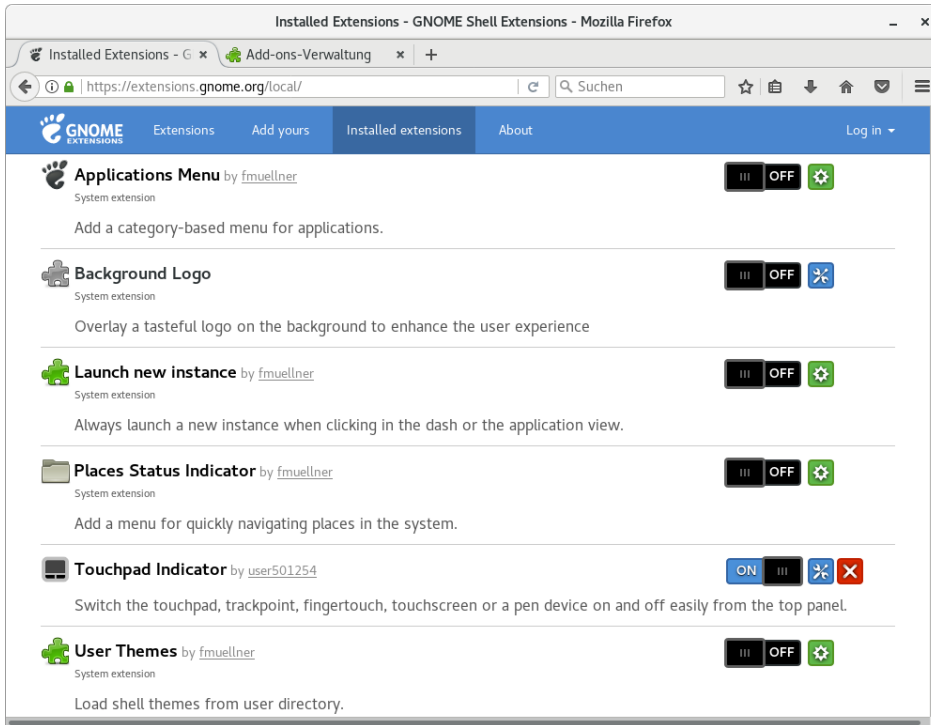
Vorbereitungs-  
arbeiten in  
Firefox

Im Idealfall funktioniert die Nutzung der Gnome-Extensions-Webseite auf Anhieb. Häufig bedarf es vorher aber je nach Distribution und Webbrowser gewisser Vorbereitungsarbeiten.

Wenn Sie die Gnome-Extensions-Seite zum ersten Mal besuchen, müssen Sie Firefox das Ausführen des Add-ons *Gnome Shell Integration* erlauben. Wenn das Add-on nicht ohnedies schon installiert ist, wird dies nach einer Rückfrage gleich erledigt. Danach müssen Sie die Seite neu laden.

Bei manchen Distributionen, z. B. unter Fedora, erscheint nun beim neuerlichen Besuch der Webseite die folgende Fehlermeldung: *Obwohl die Gnome-Shell-Integrations-Erweiterung läuft, wurde der Native-Host-Connector nicht erkannt*. In diesem Fall müssen Sie noch das Python-Script `chrome-gnome-shell` installieren. (Lassen Sie sich vom Namen nicht irritieren. Das Programm wurde zwar offensichtlich zuerst für Google Chrome entwickelt, unterstützt aber auch Firefox!) Installationsanleitungen für diverse Distributionen finden Sie hier:

<https://wiki.gnome.org/Projects/GnomeShellIntegrationForChrome/Installation>



**Abbildung 5.18** Gnome Shell Extensions werden direkt im Webbrowser gesucht, aktiviert und konfiguriert.

Unter Fedora führen Sie beispielsweise die folgenden zwei Kommandos in einem Terminalfenster aus:

```
user$ sudo dnf copr enable region51/chrome-gnome-shell
user$ sudo dnf install chrome-gnome-shell
```

Unter Ubuntu ist das Paket gleich in den offiziellen Paketquellen enthalten. Dort reicht das folgende Kommando aus:

```
root# sudo apt install chrome-gnome-shell
```

Auch Chromium bzw. Google Chrome fragt beim ersten Besuch der Seite, ob es das Add-on *Gnome Shell Integration* installieren darf. Außerdem muss das oben schon erwähnte Paket `chrome-gnome-shell` installiert werden. Anschließend starten Sie den Webbrowser neu.

Vorbereitungs-  
arbeiten in  
Chromium

Auf der Webseite können Sie nun nach Erweiterungen suchen und diese mit dem ON/OFF-Button unkompliziert installieren. Manche Erweiterungen sehen einen eigenen Konfigurationsdialog vor, den Sie durch einen Klick auf das blaue Werkzeug-Icon öffnen. Wenn es zu einer Erweiterung eine neue Version gibt, zeigt die Gnome-Extensions-Website neben der Erweiterung ein grünes Update-Icon an.

Erweiterungen  
verwalten

### Shell-Extensions als Pakete installieren

Manche Distributionen stellen wichtige Gnome-Erweiterungen in Form normaler Pakete zur Verfügung. Sie können die Paketverwaltungswerkzeuge Ihrer Distribution verwenden, um danach zu suchen und diese zu installieren. Beispielsweise liefert `dnf search shell-extension` eine Liste von rund 20 Erweiterungen. Diese Erweiterungen haben den offiziellen Segen der Fedora-Entwickler. Andererseits ist die Auswahl im Vergleich zur Gnome-Extensions-Website recht eingeschränkt.

### Shell Extensions im Gnome Tweak Tool verwalten

Anstatt die Seite <https://extensions.gnome.org> zur Steuerung der installierten Erweiterungen zu verwenden, können Sie auch das Dialogblatt ERWEITERUNGEN des Gnome Tweak Tools verwenden. Das Aktivieren, Deaktivieren und Konfigurieren von Erweiterungen gelingt dort gut; allerdings bietet das Programm weder Update-Optionen noch hilft es bei der Suche nach neuen Erweiterungen.

## Nützliche Erweiterungen

Hoffentlich habe ich Sie mit der langen Einleitung zu allen Eventualitäten nicht verschreckt – das wäre schade! Auf der Gnome-Erweiterungswebsite sind nämlich richtige »Perlen« zu finden, die das Arbeiten mit Gnome ganz wesentlich verbessern! Ich möchte Ihnen hier ganz kurz einige Highlights vorstellen:

- ▶ **Alternate Tab:** Wenn Sie diese Erweiterung aktivieren, funktioniert **Alt**+**Esc** wieder wie unter alten Gnome-Versionen bzw. unter Windows und wechselt zwischen Fenstern und nicht zwischen Programmen.
- ▶ **Application Men:** Die Erweiterung ersetzt den Button AKTIVITÄTEN durch ein vernünftiges Startmenü.
- ▶ **Dash to Dock:** Mit dieser Erweiterung können Sie das Dock dauerhaft sichtbar machen (Option AUTOMATISCH AUSBLENDEN = AUS). Das erspart das unsäglich-e Ein- und Ausschalten der Aktivitätenansicht für jeden Programmstart oder -wechsel mit der Maus. Besonders nützlich ist diese Erweiterung auf einem großen Bildschirm, wo Sie nicht um jedes Pixel geizen müssen. Ein weiterer Pluspunkt sind unzählige Optionen, mit denen Sie Aussehen und Größe der Icons im Dock sowie diverse Zusatzfunktionen steuern können. Auch eine Platzierung des Docks am unteren oder rechten Bildschirmrand ist möglich.
- ▶ **Impatience:** Die Erweiterung beschleunigt Gnome-Animationen.
- ▶ **No Top Left Hot Corner:** Der Name ist Programm. Wenn diese Erweiterung aktiv ist, muss die Aktivitätenansicht explizit durch einen Mausklick geöffnet werden. Das bloße Verschieben der Maus in das linke obere Bildschirm-eck reicht nicht aus. Das oft irritierende unbeabsichtigte Erscheinen der Aktivitätenansicht gehört der Vergangenheit an.
- ▶ **Places status indicator:** Diese Erweiterung fügt neben dem AKTIVITÄTEN-Button ein Menü ein, um wichtige Verzeichnisse zu öffnen.

- ▶ **Removable Drive Menu:** Die Erweiterung fügt rechts im Panel ein kleines Menü ein, in dem Sie USB-Sticks und andere externe Datenträger lösen («auswerfen») können.
- ▶ **Open Weather:** Diese Erweiterung zeigt neben der Uhrzeit die aktuelle Temperatur und auf Mausklick eine Wetterprognose an.
- ▶ **TopIcons Plus:** Die Erweiterung zeigt Statusinformationen von Hintergrundprogrammen wie Dropbox direkt im Panel an und versteckt diese nicht links unten auf dem Bildschirm.
- ▶ **Window List:** Diese Erweiterung zeigt am unteren Bildschirmrand eine Taskleiste für alle laufenden Programme an.
- ▶ **User Themes:** Diese Erweiterung ermöglicht es, Gnome-Themes zu aktivieren. Auf diese Form der Gnome-Modifizierung gehe ich im nächsten Abschnitt näher ein.

## 5.7 Gnome Shell Themes

*Gnome Shell Themes* sind Dateien, die das Erscheinungsbild des Desktops modifizieren. Veränderbare Elemente sind die Farbe des Panels, der Menüs und der Fensterumrandung, die Gestaltung von Buttons und anderen Bedienelementen etc.

Gnome Shell  
Themes

Bevor Sie eigene Themes verwenden können, müssen Sie zuerst die Gnome-Erweiterung *User Themes* aktivieren. Anschließend laden Sie das gewünschte Thema aus dem Internet herunter und packen es in einem neuen Unterverzeichnis in `.local/share/themes` oder einfach in `.themes` aus. Beide Orte sind zulässig. Wenn es noch keine `themes`-Verzeichnisse gibt, müssen Sie zumindest eines einrichten (z. B. mit `mkdir .themes` in einem Terminalfenster). Als zentrale Sammelstelle für unzählige Themes hat sich die folgende Seite etabliert:

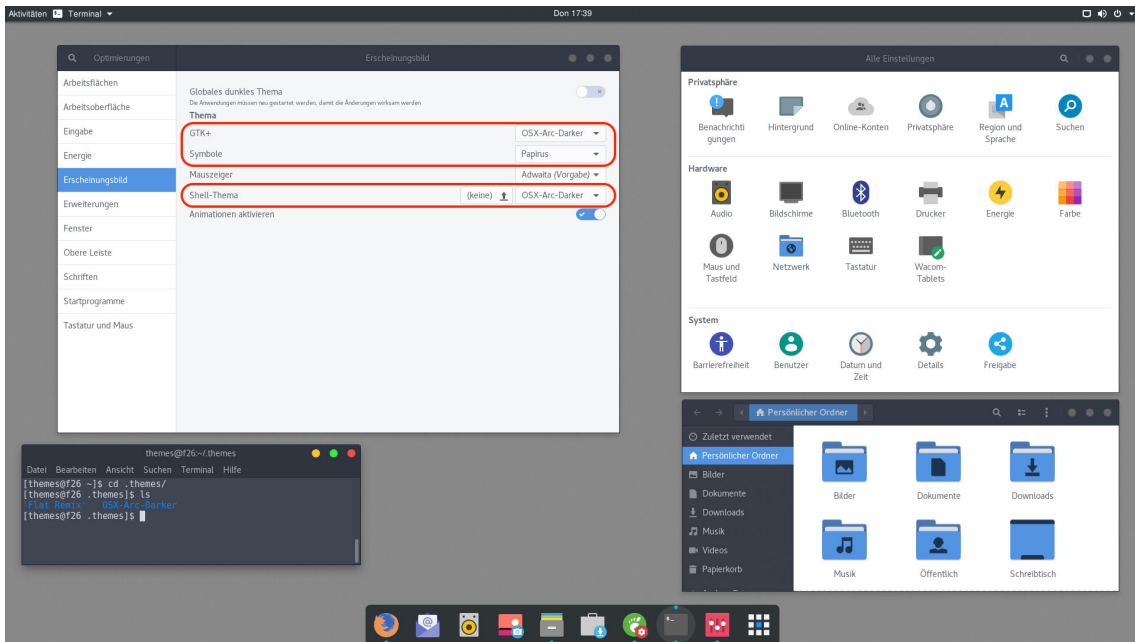
<https://www.gnome-look.org>

Allerdings stehen nicht alle Themes zum simplen Download zur Verfügung. Einige Themes befinden sich in Git-Repositories. Das Befolgen der Installationsanleitung erfordert dann ein wenig Linux-Routine. Beispielsweise müssen Sie zur Installation des *Flat Remix Themes* die folgenden Kommandos in einem Terminal ausführen:

```
user$ cd /tmp
user$ git clone https://github.com/daniruiz/Flat-Remix-GNOME-theme
user$ mkdir -p ~/.themes
user$ mv "Flat-Remix-GNOME-theme/Flat Remix" ~/.themes
```

Vorher müssen Sie in der Regel auch noch das Programm `git` installieren, z. B. mit `sudo apt install git`.

Sobald sich die Dateien des Themes in einem Unterverzeichnis von `.local/share/themes` oder einfach `.themes` befinden, können Sie das Theme im Gnome Tweak Tool aktivieren (siehe [Abbildung 5.19](#)). Dazu öffnen Sie das Dialogblatt ERSCHEINUNGSBILD und wählen sowohl bei GTK+ als auch bei SHELL-THEMA das gewünschte Thema aus. (Wenn das Gnome Tweak Tool schon läuft, müssen Sie es neu starten, damit es das Theme erkennt.)



**Abbildung 5.19** Individueller Gnome-Desktop mit dem Gnome Shell Theme »OSX-Arc-Darker«, dem Icon-Set »Papirus« und der Erweiterung »Dash to Dock«

**Icon Themes** Auch die in den Systemeinstellungen und im Dock verwendeten Icons können durch andere Bilder ausgetauscht werden. Auf der Website <https://www.gnome-look.org> finden Sie Icons-Sets zum Download. Das Verzeichnis mit den Icons müssen Sie nach dem Auspacken in den Ordner `/usr/share/icons` verschieben, z. B. so:

```
user$ sudo mv Downloads/some-icons /usr/share/icons
```

### OMG Ubuntu

Die Website <http://www.omgubuntu.co.uk> hat 2017 schon gut ein Dutzend Artikel zur Installation von Gnome Themes veröffentlicht. Wenn Sie auf der Suche nach Inspirationen sind, werden Sie dort sicher fündig. Eine andere Frage ist natürlich, wie weit es sinnvoll ist, den Desktop in mühevoller Arbeit stundenlang zu konfigurieren ...

## 5.8 Gnome-Interna

Es gibt gegenwärtig zwei Konfigurationssysteme für Gnome: das neue `dconf`-System und das aus Gnome-2-Zeiten stammende `gconf`-System. Die `dconf`-Daten befinden sich in der binären Datenbankdatei `.config/dconf/user`. Allerdings verwenden noch nicht alle Gnome-Programme das `dconf`-System.

**dconf-Datenbank**

Gnome-Programme greifen direkt über API-Funktionen (*Application Programming Interface*) auf die `dconf`-Datenbank zu. Wenn Sie `dconf`-Einstellungen von außen lesen oder ändern möchten, installieren Sie das Paket `dconf-editor` und starten das gleichnamige Programm (siehe [Abbildung 5.20](#)).

| Setting Name                               | Value       | Description  |
|--|-------------|--|
| <code>always-use-location-entry</code>     | falsch      | Immer das Adressfeld anstatt der Pfadleiste verwenden  |
| <code>automatic-decompression</code>       | wahr        | Legt fest, ob komprimierte Dateien entpackt werden sollen, anstatt sie mit einer anderen Anwendung zu öffnen |
| <code>bulk-rename-tool</code>              | []          | Werkzeug zum Umbenennen vieler Dateien   |
| <code>click-policy</code>                  | 'double'    | Klicktyp zum Ausführen/Öffnen von Dateien  |
| <code>confirm-trash</code>                 | wahr        | Legt fest, ob beim Löschen von Dateien oder beim Leeren des Papierkorbs nachgefragt werden soll              |
| <code>default-folder-viewer</code>         | 'list-view' | Voreingestellte Ordneransicht  |
| <code>default-sort-in-reverse-order</code> | falsch      | Sortierreihenfolge in neu geöffneten Fenstern umkehren?  |
| <code>default-sort-order</code>            | 'name'      | Voreingestellte Sortierreihenfolge   |
| <code>executable-text-activation</code>    | 'display'   | Wie soll mit ausführbaren Textdateien verfahren werden, wenn sie aktiviert werden?                           |
| <code>install-mime-activation</code>       | wahr        | Den Installationsdialog für unbekannte MIME-Typen anzeigen   |
| <code>mouse-back-button</code>             | 8           | Maustaste zum Auslösen des »Zurück«-Befehls in einem Browser-Fenster   |

**Abbildung 5.20** Einstellungen in der `dconf`-Datenbank verändern

Mit dem Kommando `gsettings` ist es möglich, die `dconf`-Einstellungen im Terminal oder durch ein Script zu verändern. Das folgende Kommando bewirkt, dass Nautilus standardmäßig die Listenansicht verwendet, nicht die Symbolansicht:

```
user$ gsettings set org.gnome.nautilus.preferences \
    default-folder-viewer 'list-view'
```

Ältere bzw. noch nicht auf das `dconf`-System umgestellte Gnome-Programme speichern ihre Einstellungen in der `gconf`-Datenbank. Intern besteht diese Datenbank aus unzähligen kleinen XML-Dateien, die im Verzeichnis `.gconf` sowie in dessen Unterverzeichnissen gespeichert werden. Auch zur Veränderung von `gconf`-Einstellungen gibt es mit dem `gconf-editor` eine einfache Benutzeroberfläche. Sie sieht ganz ähnlich aus wie die des `dconf-editor`. Alternativ dazu können Sie die Einstellungen auch ohne Benutzeroberfläche mit dem Kommando `gconftool-2` verändern.

**gconf-Datenbank**

Während des Starts von Gnome werden eine Menge Programme automatisch gestartet. Welche dies sind, steuern `*.desktop`-Dateien aus den folgenden Autostart-Verzeichnissen:

**Autostart**

|   |   |
|---|---|
| <code>~/.config/autostart/*.desktop</code>        | (persönliche Autostart-Programme)                                       |
| <code>/usr/share/gnome/autostart/*.desktop</code> | (globale Autostart-Programme für Gnome)                                 |
| <code>/etc/xdg/autostart/*.desktop</code>         | (globale Autostart-Programme für alle Desktops, also für Gnome und KDE) |

In den Systemeinstellungen gibt es kein Modul, um den automatischen Start von Programmen zu steuern. Abhilfe schaffen das Gnome Tweak Tool (Dialogblatt STARTPROGRAMME), die Gnome-Erweiterung *Startup Applications* oder die manuelle Konfiguration von `*.desktop`-Dateien. Der Aufbau solcher Dateien geht aus dem folgenden Beispiel hervor. Die Datei ist für den Start des Dropbox-Clients verantwortlich:

```
[Desktop Entry]
Name=Dropbox
GenericName=File Synchronizer
Comment=Sync your files across computers and to the web
Exec=dropbox start -i
Terminal=false
Type=Application
Icon=dropbox
Categories=Network;FileTransfer;
StartupNotify=false
```

**MIME** Wenn nach einem Doppelklick auf eine MP3-Datei in Nautilus automatisch Rhythmbox oder Banshee erscheint, dann sind hierfür die MIME-Einstellungen von Gnome verantwortlich. MIME steht für *Multipurpose Internet Mail Extensions* und ist eine Art Datenbank, die eine Zuordnung zwischen Dateitypen und Programmen herstellt.

Am einfachsten erfolgen Änderungen an der MIME-Konfiguration direkt im Dateimanager: Dort klicken Sie die betreffende Datei an, führen per Kontextmenü EIGENSCHAFTEN • ÖFFNEN MIT aus und wählen das gewünschte Programm. Die Einstellung gilt in Zukunft für alle Dateien mit derselben Endung.

Individuelle Änderungen an der MIME-Konfiguration werden hier gespeichert:

```
~/.local/share/mime/*
~/.local/share/applications/mimeapps.list
```

Weitere Informationen zur MIME-Datenbank unter Gnome finden Sie unter:

<https://standards.freedesktop.org/shared-mime-info-spec/shared-mime-info-spec-latest.html>

## XDG-Verzeichnisse und -Scripts

Vor einigen Jahren wurde im Rahmen des Portland-Projekts eine Reihe gemeinsamer Standards definiert. Sie helfen dabei, Programme unabhängig von Gnome oder KDE



richtig in den Desktop zu integrieren. Später führte die X Desktop Group (XDG) diese Bemühungen fort, und heute ist es das Projekt *freedesktop.org*.

Beim ersten Login werden im Heimatverzeichnis die Unterverzeichnisse Bilder, Dokumente, Downloads, Musik, Öffentlich, Videos und Vorlagen erzeugt. Wenn eine andere Sprache als Deutsch eingestellt ist, erhalten diese Verzeichnisse andere Namen. Hinter den Kulissen ist das Paket `xdg-user-dirs` für die Verzeichnisse verantwortlich.

**Standard-  
verzeichnisse**

Die Konfiguration erfolgt durch die Datei `.config/user-dirs.dirs`. Sie stellt sicher, dass XDG-kompatible Programme die Verzeichnisse unabhängig von der eingestellten Sprache finden. Unter Gnome werden die Verzeichnisse, wenn die Sprache verändert wurde, nach einer Rückfrage sogar entsprechend umbenannt (Paket `xdg-user-dirs-gtk`).

Wenn Sie die Standardverzeichnisse nicht wünschen, löschen Sie die Verzeichnisse und legen die folgende neue Datei an:

```
# ~/.config/user-dirs.conf
enabled=False
```

Sie können diese Einstellung systemweit in `/etc/xdg/user-dirs.conf` vornehmen.

Viele, wenn auch leider nicht alle Gnome- und KDE-Programme verwenden zum Speichern von Konfigurationseinstellungen und internen Daten speziell dafür vorgesehene Verzeichnisse. Die Verzeichnisnamen beginnen mit einem Punkt und gelten damit als »verborgen«. Die Verzeichnisse werden deswegen im Dateimanager standardmäßig nicht angezeigt.

**Konfigurations-  
verzeichnisse**

- ▶ Das `.cache`-Verzeichnis ist zur Speicherung von temporären Dateien gedacht, die bei Bedarf neuerlich erzeugt werden können – also z. B. verkleinerte Bilder (Thumbnails), Suchindizes etc. Die Zwischenspeicherung dient dazu, häufig vorkommende Arbeitsabläufe zu beschleunigen.
- ▶ Das `.config`-Verzeichnis ist zur Speicherung von Programmeinstellungen vorgesehen, wobei jedes Programm ein eigenes Unterverzeichnis verwendet.
- ▶ Im `.local`-Verzeichnis werden Benutzerdaten gespeichert. Üblicherweise legt jedes Programm hierfür das Unterverzeichnis `share/programmname` an.

Das Paket `xdg-utils` stellt die folgenden Scripts zur Verfügung. Eine genauere Beschreibung finden Sie in den `man`-Seiten der jeweiligen Kommandos.

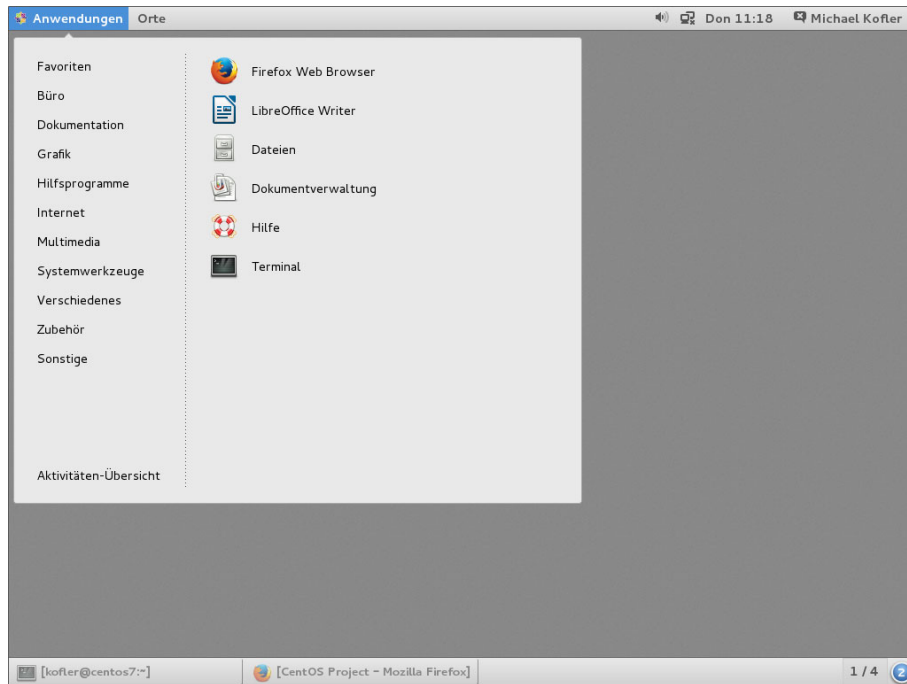
**xdg-Scripts**

- ▶ `xdg-desktop-menu` fügt dem Desktop-Menü einen neuen Eintrag hinzu.
- ▶ `xdg-desktop-icon` installiert ein neues Icon auf dem Desktop.
- ▶ `xdg-icon-resource` installiert Icon-Ressourcen.
- ▶ `xdg-mime` fragt die MIME-Datenbank ab bzw. richtet einen neuen MIME-Datentyp ein.

- ▶ `xdg-open` öffnet ein Dokument im Standardprogramm des Benutzers.
- ▶ `xdg-email` sendet eine E-Mail im Standard-E-Mail-Programm des Benutzers.
- ▶ `xdg-screensaver` steuert den Bildschirmschoner.

## 5.9 Der Gnome-Klassikmodus

Vielen Linux-Anwendern ist der Umstieg von Gnome 2 auf die aktuelle Version 3 schwergefallen. Speziell für diese Anwendergruppe haben die Gnome-Entwickler den Klassikmodus geschaffen. Dabei handelt es sich um eine vordefinierte Sammlung von Gnome-Erweiterungen (Extensions), mit denen Gnome 3 ähnlich aussieht wie Gnome 2: Es gibt ein traditionelles Startmenü, Icons auf dem Desktop und eine Taskleiste am unteren Bildschirmrand (siehe [Abbildung 5.21](#)).



**Abbildung 5.21** CentOS verwendet im Klassikmodus ein traditionelles Startmenü.

Bei manchen Distributionen müssen die Erweiterungspakete des Klassikmodus extra installiert werden. Der Paketname lautet oft `gnome-classic-session`. Beim nächsten Login können Sie dann zwischen `GNOME` und `GNOME CLASSIC` wählen.

Gerade umgekehrt ist es bei CentOS 7 bzw. RHEL 7: Um die konservativen Unternehmenskunden nicht mit zu viel Modernität zu überfordern, kommt der Klassikmodus

dort standardmäßig zum Einsatz. Gleichzeitig ist aber auch der »echte« Gnome-3-Desktop installiert. Wer will, kann also beim Login auf das neue Gnome umsteigen.

Der Klassikmodus bietet keine echte Kompatibilität zu Gnome 2. Insbesondere stehen die aus Gnome 2 vertrauten Applets nicht zur Verfügung. Auch bei der Konfiguration der beiden Panels gibt es wenig Spielraum. Persönlich hätte ich mir z. B. noch konfigurierbare Schnellstart-Icons im Panel gewünscht. Diese lassen sich aber bei Bedarf mit der Gnome-Extension *Frippery Panel Favorites* realisieren.

## 5.10 MATE

Seit der Fertigstellung von Gnome 3 wird der Code der Vorgängerversion Gnome 2 nicht mehr gewartet. Das hätte Distributionen, die eigentlich gerne bei Gnome 2 bleiben würden, auf kurz oder lang zu einem Umstieg auf Gnome 3 gezwungen – wäre da nicht MATE!

<https://mate-desktop.org>

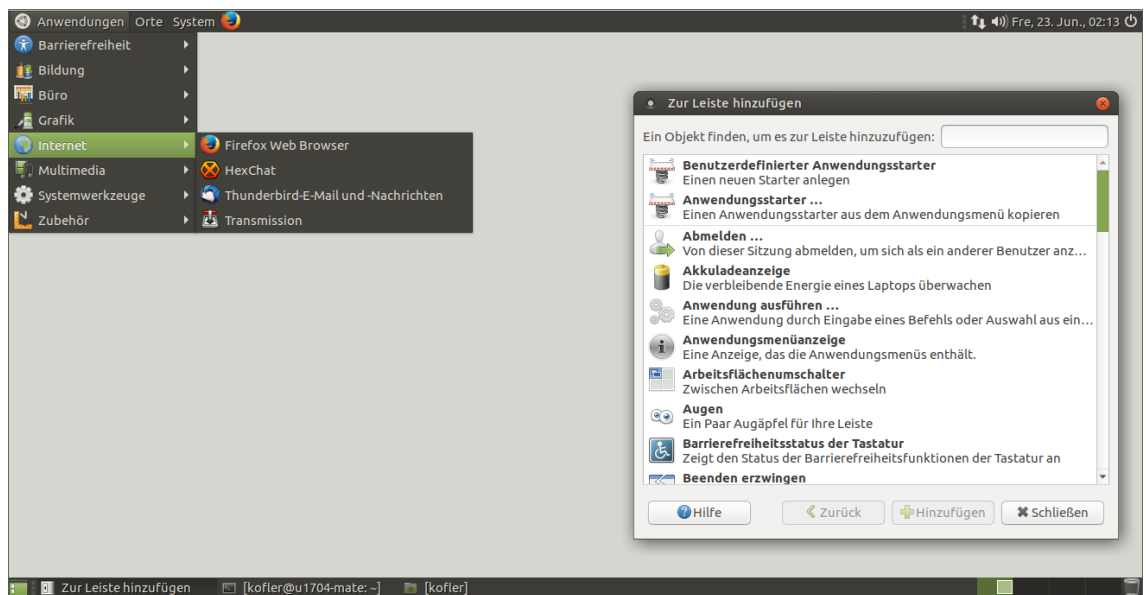


Abbildung 5.22 Der MATE-Desktop in der Konfiguration von Ubuntu MATE

MATE war ursprünglich ein Fork von Gnome 2. Das MATE-Projekt hat also den Code von Gnome 2 übernommen, den einzelnen Komponenten neue Namen gegeben (um Konflikte mit dem Gnome-Projekt zu vermeiden) und kümmert sich um Fehlerkorrekturen. Das MATE-Projekt wollte aber nicht in der Vergangenheit stehen bleiben. Deswegen verwendet Mate mittlerweile GTK-3-Bibliotheken und soll in Zukunft sogar Wayland-kompatibel werden.

**Distributionen mit MATE** Mit Ubuntu MATE und der MATE-Variante von Linux Mint gibt es zwei eigenständige MATE-Distributionen. Ubuntu MATE gibt es sogar in einer Raspberry-Pi-Variante. Auch viele andere Distributionen enthalten MATE-Pakete. Zum Teil können Sie während der Installation auswählen, welchen Desktop Sie verwenden möchten (Debian, openSUSE), zum Teil müssen Sie eigene Installationsmedien verwenden (z. B. den MATE-Compiz-Spin von Fedora).

**Bedienung** MATE sieht aus und verhält sich exakt so wie die vielen Linux-Distributionen, die im Zeitraum zwischen 2003 und 2012 Gnome 2 einsetzten. Unverändert geblieben sind leider auch die Ärgernisse von Gnome 2, etwa dass die Maus pixelgenau positioniert werden muss, um die Größe eines Fensters zu verändern.

Der Desktop ist standardmäßig durch zwei Panels geprägt. Das obere Panel enthält links ein Menü zum Start von Programmen, zum Öffnen wichtiger Verzeichnisse im Dateimanager sowie zur Durchführung von Konfigurationsarbeiten. Rechts zeigt das Panel den Netzwerkstatus und die Uhrzeit an. Das untere Panel enthält in der Art einer Taskleiste Symbole für alle laufenden Programme. Beide Panels können über Kontextmenüs konfiguriert werden und um zusätzliche Bedienungselemente erweitert werden.

## 5.11 Cinnamon

Der Cinnamon Desktop ist eine vom Linux-Mint-Team entwickelte Erweiterung zu Gnome 3. Cinnamon versucht, Gnome 3 so zu konfigurieren, dass es wie Gnome 2 zu bedienen ist – also mit herkömmlichen Panels, ohne Dock etc. Die Idee von Cinnamon ist somit dieselbe wie bei dem Gnome-Klassikmodus, auch wenn die Realisierung und die Grundkonfiguration vollkommen anders aussieht. Cinnamon-Anwender verzichten damit auf viele Neuerungen von Gnome 3, dennoch geht die Kompatibilität zu Gnome 2 verloren. Cinnamon verwendet eigene Applets und Erweiterungen, die inkompatibel zu Gnome 2, Gnome 3 und Unity sind.

*<https://linuxmint.com>*

Wie populär das Konzept trotz dieser Nachteile ist, beweist einerseits die erstaunlich große Anzahl von Applets, Desklets und anderen Erweiterungen auf der Cinnamon-Website, andererseits die Beliebtheit von Linux Mint: Die Distribution ist nun schon seit etlichen Jahren auf Platz 1 des Zugriffs-Rankings auf *<https://distrowatch.com>*.

**Panel** Auch bei Cinnamon beinhaltet das Panel alle zentralen Steuerungselemente des Desktops. Über ein Kontextmenü können Sie es wahlweise am oberen oder unteren Bildschirmrand platzieren, die darin enthaltenen Elemente manipulieren oder neue Applets hinzufügen. Wenn Sie möchten, können Sie dem Desktop auch ein zweites Panel hinzufügen.

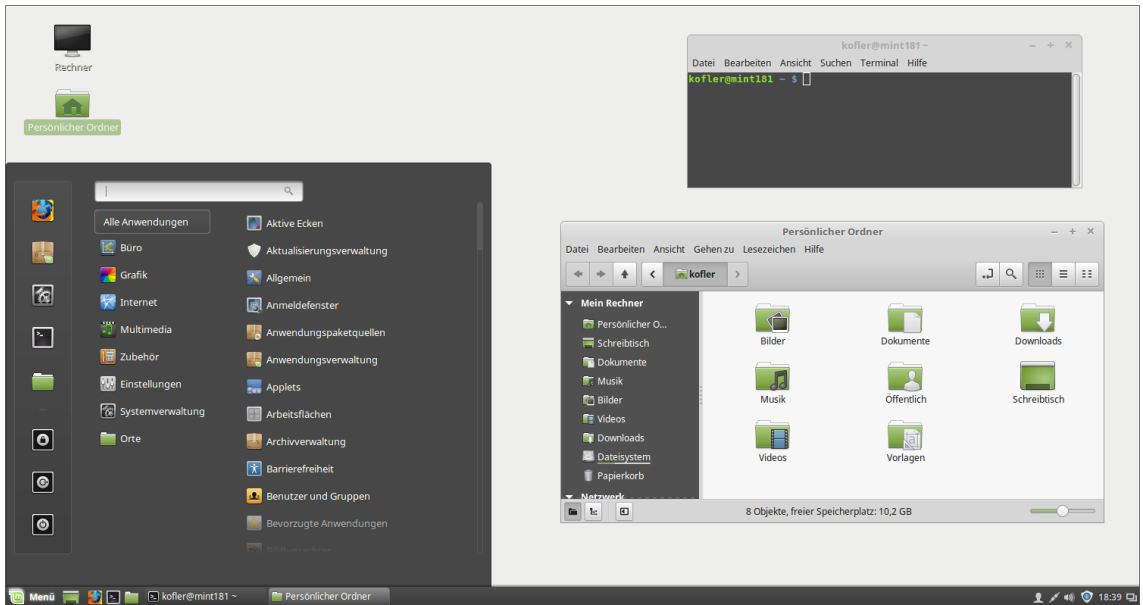


Abbildung 5.23 Der Cinnamon Desktop von Linux Mint

Cinnamon ist ein Eldorado für alle, die Ihren Desktop gerne individuell gestalten. Die Systemeinstellungen enthalten unzählige Module, in denen Sie das Aussehen und Verhalten der Fenster, des Desktop-Hintergrunds, der Arbeitsflächen etc. konfigurieren können.

**Konfiguration**

Über ein Kontextmenü des Desktops können Sie auf diesem sogenannte »Desklets« installieren. Das sind Miniprogramme, die direkt auf dem Desktop ausgeführt und von Fenstern überdeckt werden. Standardmäßig werden nur wenige Desklets mitgeliefert. Sie können aber mithilfe des Desklet-Dialogs eine Menge weiterer Miniprogramme aus dem Internet herunterladen.

**Desklets**



# Kapitel 6

## KDE und Unity

Nach der ausführlichen Vorstellung von Gnome im vorigen Kapitel stelle ich Ihnen hier zwei alternative Desktop-System für Linux vor. Beide haben ihre Vorzüge beide sind weite verbreitet.

- ▶ **KDE:** Die Abkürzung KDE stand ursprünglich für *Kool Desktop Environment*, später wurde daraus das *K Desktop Environment*. Während die Zielsetzung von Gnome *Keep it Simple* lautet, bietet KDE seinen Anwendern maximale Konfigurierbarkeit und nimmt dafür unübersichtliche Dialoge und verschachtelte Menüs in Kauf. Insofern richtet sich KDE an technisch orientierte Linux-Anwender, die Ihre Arbeitsumgebung uneingeschränkt von Gnome-Richtlinien gestalten möchten.
- ▶ **Unity:** Unity war ein Versuch Canonicals, eine eigene Desktop-Umgebung für Ubuntu zu schaffen. Im Verlauf von sieben Jahren hat Canonical eine Menge toller Ideen realisiert. Gleichzeitig litt Unity aber immer unter dem Status einer Inselösung. Letztlich reichte die personelle und finanzielle Kraft von Canonical nicht aus, um die vielen Desktop-Bestandteile weiterzuentwickeln. Deswegen wurde die Entwicklung Anfang 2017 eingestellt. Unity wird aber als Bestandteil von Ubuntu 16.04 noch bis 2021 gewartet und läuft aktuell auf unzähligen Linux-Rechnern. Zudem ist es möglich, dass es in Zukunft neben dem offiziellen Ubuntu mit Gnome auch eine Unity-Variante geben wird. Unity ist also (noch lange) nicht tot!

### Freie App-Wahl

Mit der Entscheidung für eine Desktop-Umgebung erhalten Sie bei jeder Distribution einen Mix von Anwendungen, der für das jeweilige Desktop-System optimiert ist. Dazu zählen ein Dateimanager, ein Terminalprogramm, Audio- und Video-Player etc.

Diese vorgegebene Auswahl schränkt Sie in keiner Weise ein! Sie können unter Gnome anstelle des PDF-Viewers Evince das leistungsfähigere KDE-Pendant Okular installieren und anwenden. Genauso gut können Sie unter KDE statt des Fotoverwaltungsprogramms digiKam, das mit unzähligen Funktionen für Profis optimiert wurde, das viel simplere Gegenstück Shotwell installieren.

Einen Überblick über die wichtigsten Desktop-Anwendungsprogramme für Linux gebe ich deswegen distributionsunabhängig im nächsten Kapitel.

## 6.1 KDE

KDE erfüllt im Prinzip dieselben Aufgaben wie Gnome, sieht aber anders aus und verwendet intern ganz andere Bibliotheken und Protokolle. KDE basiert auf Qt, einer Open-Source-Bibliothek, die ursprünglich von der Firma Troll Tech entwickelt wurde. Umfassende Informationen zu KDE gibt diese Website:

<https://kde.org>

Im Vergleich zu Gnome bietet KDE mehr Spezialfunktionen und Konfigurationsmöglichkeiten, die technisch versierten Linux-Benutzern entgegenkommen. Dem steht aber eine etwas schwierigere Bedienung gegenüber, weswegen viele Distributionen standardmäßig auf Gnome basieren.

Im Mittelpunkt dieses Kapitels stehen die Basisfunktionen von KDE. Freilich sieht KDE je nach Distribution ganz unterschiedlich aus: Der Aufbau des Login-Bildschirms, die Menüeinträge des Startmenüs, die optische Gestaltung des Desktops und die Auswahl der mitgelieferten Programme und Konfigurationshilfen variieren stark.

### Die komplizierte KDE-Nomenklatur

Wenn ich in diesem Buch einfach von KDE schreibe, so ist dies eine Verkürzung. Vielmehr ist die *KDE Community* eine Gruppe von Entwicklern, die diverse KDE-Software-Produkte entwickelt – in ihrer Gesamtheit die *KDE Software Compilation*. Diese besteht unter anderem aus den *KDE Applications* (diversen KDE-Anwendungsprogrammen), dem *KDE Framework* (dem Fundament, den Bibliotheken) sowie aus *Plasma* (dem eigentlichen Desktop). Selbstredend hat jede dieser Komponenten eigene Versionsnummern. Noch mehr Details können Sie hier nachlesen:

<https://www.kde.org/community/whatiskde/softwarecompilation.php>  
<https://ikhaya.ubuntuusers.de/2015/05/01/die-entwicklungen-bei-kde>

#### KDE-Distributionen (Neon)

Die meisten großen Distributionen unterstützen KDE, sei es als Derivat oder Spin (Kubuntu, Fedora KDE Spin), sei es als Option während der Installation (Debian, SUSE). Unter SUSE ist KDE sogar als Standard-Desktop vorgesehen. Insofern gilt SUSE als gute Wahl für KDE-Fans.

Persönlich noch attraktiver finde ich aktuell die Distribution Neon: Sie kombiniert Ubuntu LTS als Unterbau mit einem *Rolling Release* der KDE-Pakete als Ergänzung. Damit erhalten Sie einerseits ein absolut stabiles Fundament, bekommen aber andererseits stets die neueste Version aller KDE-Bibliotheken und -Programme. Innerhalb der Neon-Familie gibt es wiederum vier Varianten:



- ▶ Ich empfehle Ihnen die *User Edition*, in der Sie immer die gerade aktuellen KDE-Versionen erhalten, sobald diese offiziell freigegeben wurden. Dieses Kapitel habe ich auf der Basis der *Neon User Edition* verfasst.
- ▶ Bei der *User LTS Edition* werden die KDE-Pakete dagegen nicht aktualisiert. Diese Variante verhält sich insofern ähnlich wie Kubuntu LTS.
- ▶ In der *Developer Edition Stable* können Sie neue KDE-Komponenten schon in der Entwicklungsphase ausprobieren. Dabei müssen Sie aber mit den typischen Problemen von Beta-Software rechnen.
- ▶ Noch radikaler ist die *Developer Edition Unstable*, die einfach täglich mit den KDE-Entwicklerpaketen synchronisiert wird. Diese Edition kommt nur infrage, wenn Sie selbst KDE-Entwickler sind oder einer werden möchten.

Eine Eigenheit von Neon besteht darin, dass vorerst nur ein absolutes Minimum von Anwendungen installiert wird. Beispielsweise steht anfangs weder ein Audio-Player noch ein Bildverwaltungsprogramm zur Verfügung. Sie müssen sich selbst um die Installation geeigneter Programme kümmern (z. B. Amarok, digiKam).

### KDE-Besonderheiten

Ein ärgerliches Problem aller KDE-Distributionen besteht darin, dass die deutsche Lokalisierung nicht mit der Entwicklungsgeschwindigkeit mithalten kann. Deswegen tauchen immer wieder englische Texte in ansonsten deutschsprachigen Menüs und Dialogen auf.

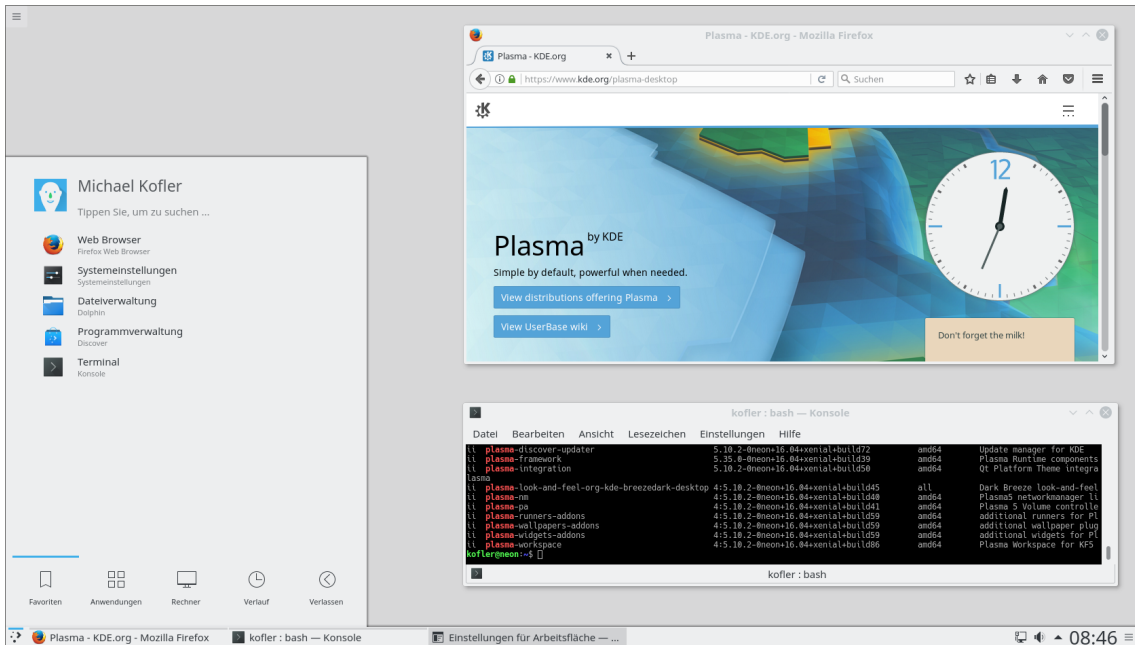
#### KDE und die Maus

Der vielleicht offensichtlichste Unterschied zwischen KDE und anderen Benutzeroberflächen ist der Umgang mit der Maus: Unter KDE reicht statt eines Doppelklicks ein einfacher Mausklick, um Dateien zu öffnen, Module zu starten oder vergleichbare Operationen durchzuführen. Das ist anfangs gewöhnungsbedürftig, ermöglicht aber ein effizientes und komfortables Arbeiten.

Wenn Sie sich nicht umstellen wollen, können Sie natürlich auch KDE doppelklickkonform einrichten: Dazu starten Sie im KDE-Menü die SYSTEMEINSTELLUNGEN, wechseln in das Modul EINGABEGERÄTE • MAUS und aktivieren die Option DOPPELKLICK ZUM ÖFFNEN VON DATEIEN UND ORDNERN. Allerdings ist jetzt auch in den Systemeinstellungen zur Auswahl eines Moduls ein Doppelklick erforderlich.

## Aufbau des Desktops

**Desktop** Abbildung 6.1 zeigt den KDE-Desktop. Er setzt sich standardmäßig aus einem Panel am unteren Bildschirmrand und dem eigentlichen Arbeitsbereich zusammen. Das Panel enthält das KDE-Menü, eventuell einige Icons zum raschen Start von Programmen, eine Taskleiste mit Icons aller offenen Fenster sowie diverse Hilfsprogramme.



**Abbildung 6.1** Der KDE-Desktop

**Plasmoids** Der eigentliche Arbeitsbereich (Desktop) ist anfänglich zumeist leer. Sie können direkt im Desktop oder im Panel Miniprogramme ausführen, die in der KDE-Nomenklatur *Plasmoids* heißen. Über das Kontextmenükommando **MINIPROGRAMME HINZUFÜGEN** bzw. über den Button, der in Abbildung 6.1 in der linken oberen Bildschirmcke platziert ist, fügen Sie Plasmoids in den Desktop ein (siehe Abbildung 6.2).

### Alle Plasmoids in die erste Reihe, bitte!

Persönlich bin ich kein Freund von Icons, Miniprogrammen und anderen Desktop-Objekten: Bei mir verdecken in der Regel mehrere große Fenster den gesamten Arbeitsbereich. Wenn Sie gern Icons und Plasmoids verwenden, sollten Sie sich die Tastenkombination **Strg+F12** merken: Sie rückt die Desktop-Elemente in den Vordergrund und stellt alle Fenster abgedunkelt in den Hintergrund. Nochmals **Strg+F12** oder **Esc** stellt den bisherigen Desktop-Zustand wieder her.

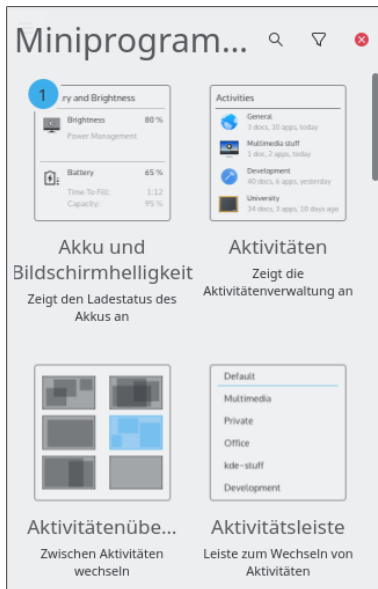


Abbildung 6.2 Miniprogramme (Plasmoids) einfügen

Das Panel bzw. in der KDE-Nomenklatur die Kontrollleiste befindet sich an einem Bildschirmrand (standardmäßig unten). Das Panel an sich hat keine Funktion, sondern dient nur als Container für Miniprogramme. Auch so grundlegende Elemente wie das Menü und die Taskleiste sind in KDE als Plasmoids implementiert. Deswegen ist es grundsätzlich möglich (wenngleich unüblich), auf ein Panel ganz zu verzichten und das Menü, die Taskleiste und andere typische Panel-Inhalte direkt auf dem Desktop abzulegen. Der größte Vorteil des Panels besteht darin, dass dieser Bereich nicht von Fenstern überdeckt werden kann. Außerdem spart die kompakte Anordnung mehrerer Plasmoids in einem Panel viel Platz.

Panels



Abbildung 6.3 Panel-Konfiguration

Über das Kontextmenükommando **EINSTELLUNGEN FÜR DIE KONTROLLEISTE** können Sie Größe, Position und andere Eigenschaften des Panels verändern sowie Miniprogramme hinzufügen, verschieben und entfernen (siehe [Abbildung 6.3](#)). Dazu wird oberhalb bzw. neben dem Panel eine Art Menü eingeblendet. Die Farbe bzw. Hintergrundgrafik des Panels ist übrigens durch das Desktop-Design vorgegeben und kann nur durch die Auswahl eines anderen Designs verändert werden (siehe [Abschnitt 6.3](#), »KDE-Konfiguration«).

### Wichtige Miniprogramme (Plasmoids)

**KDE-Menü** Das wahrscheinlich wichtigste Plasmoid ist der **ANWENDUNGSSTARTER** (links unten in [Abbildung 6.1](#)). Es ist für das KDE-Menü zuständig, das in fünf Kategorien gegliedert ist:

- ▶ **FAVORITEN** enthält die für den Benutzer wichtigsten Programme. Um ein Programm in diesen Bereich aufzunehmen, führen Sie in den anderen Menükategorien das Kontextmenükommando **ZU FAVORITEN HINZUFÜGEN** aus.
- ▶ **ANWENDUNGEN** führt in eine hierarchisch strukturierte Liste aller Programme.
- ▶ **RECHNER** gibt Ihnen die Möglichkeit, Administrationsprogramme zu starten sowie wichtige Verzeichnisse zu öffnen.
- ▶ **VERLAUF** enthält eine Liste der zuletzt gestarteten Programme bzw. der zuletzt genutzten Dateien oder Verzeichnisse.
- ▶ **VERLASSEN** enthält Kommandos zum Abmelden, zum Benutzerwechsel sowie zum Herunterfahren des Rechners.

Losgelöst von den Kategorien enthält das KDE-Menü eine Suchfunktion. Sie eignet sich insbesondere dazu, Programme rasch zu starten, ohne durch die Registerkarten des Menüs **PROGRAMME** zu navigieren. Sie können das Menü selbst modifizieren. Dazu klicken Sie den Menü-Startbutton mit der rechten Maustaste an und starten den **MENÜ-EDITOR**. Oft benötigte Programme können Sie per Drag & Drop in einen leeren Bereich des Panels oder Desktops verschieben. Sie erscheinen dort als Icons und ermöglichen so einen besonders schnellen Start.

**Taskleiste (Fensterleiste)** Das Miniprogramm **FENSTERLEISTE** zeigt für jedes Fenster ein Icon an und entspricht so der aus Windows bekannten Taskleiste. Über den Einstellungsdialog können Sie angeben, ob mehrere Fenster eines Programms zu einer Gruppe zusammengefasst werden sollen (z. B. alle GIMP-Fenster) und wie die Fenster sortiert werden sollen.

Die Fensterleiste kann ähnlich wie das Dock von macOS bzw. wie die Taskleiste unter Windows verwendet werden, um darin Start-Icons von gerade nicht laufenden Programmen abzulegen. Dazu führen Sie bei einem laufenden Programm das Kontextmenükommando **EINEN STARTER ANLEGEN** aus. Allerdings unterscheidet KDE

nun zwischen Starter-Icons und Icons laufender Programme, was nach dem Start zu Doppelgängern führt. Generell wirkt der Umgang mit Programm-Icons im Dock im Vergleich zu anderen Linux-Desktops, aber auch im Vergleich zu Windows oder macOS unausgegoren.

Arbeitsflächen ermöglichen es, die Fenster der laufenden Programme auf mehrere virtuelle Desktops zu verteilen und zwischen diesen Desktops zu wechseln. Das erleichtert die Arbeit und verbessert die Übersicht, wenn Sie sehr viele Fenster gleichzeitig öffnen. Für die Verwaltung der Arbeitsflächen ist das Plasmoid ARBEITSFLÄCHEN-UMSCHALTER verantwortlich. In dessen Einstellungsmenü stellen Sie die gewünschte Anzahl von Arbeitsflächen sowie diverse andere Optionen ein.

Arbeitsflächen

Für ständig benötigte Fenster besteht die Möglichkeit, diese so zu kennzeichnen, dass sie nicht auf einer, sondern auf allen Arbeitsflächen sichtbar sind. Dazu öffnen Sie mit der Maus oder mit **[Alt]+Leertaste** das Fenstermenü und aktivieren die Option **AUF ARBEITSFLÄCHE • ALLE ARBEITSFLÄCHEN**.

»Aktivitäten« verfolgen eine ähnliche Idee wie Arbeitsflächen. Über den durch drei farbige Punkte gekennzeichneten Aktivitäten-Button können Sie zwischen verschiedenen Desktops wechseln. Dabei startet der Pfeil-Button die Aktivität, der quadratische schwarze Button stoppt sie wieder.

Aktivitäten

Aktivitäten sind aber mehr als nur eine Neuimplementierung von Arbeitsflächen: Jede Aktivität kann einen eigenen Bildschirmhintergrund haben, eigene Programme und Plasmoide ausführen und eigene Energiespareinstellungen aufweisen (z. B. zur Deaktivierung des Bildschirmschoners und der Bildschirmsperre für die Aktivität *Vortrag*). Obwohl das Aktivitätenkonzept interessant ist, erschweren die fehlende Dokumentation und die unübersichtliche Konfiguration eine effiziente Nutzung.

Wenn das Panel den sogenannten Systemabschnitt enthält, können Hintergrundprogramme im Panel dort auf sich aufmerksam machen – z. B. wenn neue Updates verfügbar sind oder eine neue E-Mail eingetroffen ist. Der Systemabschnitt befindet sich normalerweise am rechten oder unteren Ende des Panels. Er erfüllt an sich keine Funktion, sondern ist lediglich ein Platzhalter, in dem andere Programme Icons darstellen können. Diese Funktion scheint selbstverständlich zu sein, und tatsächlich werden Sie auf den Systemabschnitt wohl nur aufmerksam, wenn er aus irgendeinem Grund im Panel fehlt und Benachrichtigungen über E-Mails, Updates etc. ausbleiben.

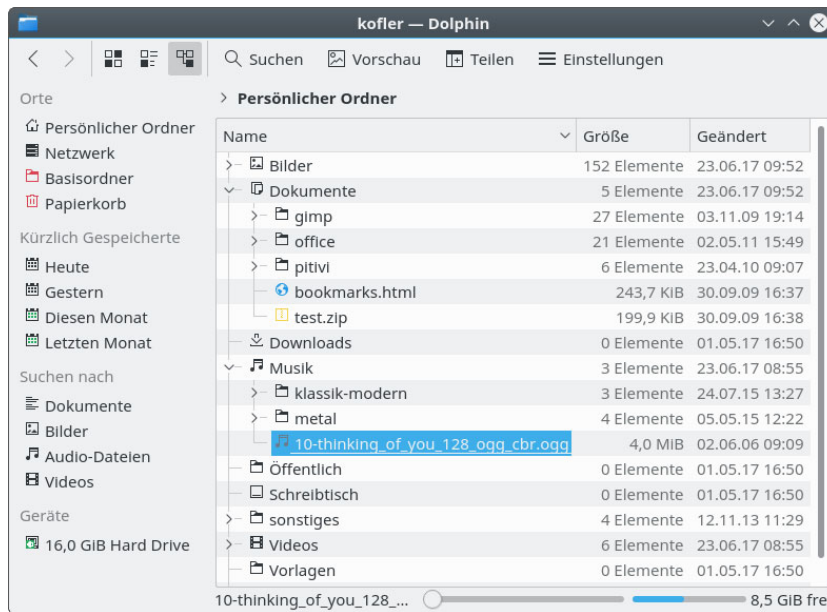
Systemabschnitt

Das Plasmoid GERÄTEÜBERWACHUNG informiert über neu angeschlossene externe Datenträger und hilft dabei, deren Dateisystem zu öffnen bzw. wieder sicher aus dem Verzeichnisbaum zu lösen (*umount*).

Geräte-  
überwachung

## 6.2 KDE-Dateimanager

**Ansichten** Das Programm *Dolphin* ist der Dateimanager von KDE (siehe [Abbildung 6.4](#)). Die Grundfunktionen des Programms sind rasch erklärt: Im Zentrum des Fensters werden die Dateien angezeigt, wobei es drei Darstellungsmodi gibt: SYMBOLE, DETAILS und KOMPAKT, die Sie mit `[Strg]+[1]` bis `[Strg]+[3]` aktivieren.



**Abbildung 6.4** In der Detailansicht von Dolphin können Verzeichnisse baumartig ausgeklappt werden.

Sehr praktisch ist die Gruppierungsfunktion: `EINSTELLUNGEN • ELEMENTE GRUPPIEREN` fasst Dateien mit demselben Typ bzw. nach Anfangsbuchstaben zu Gruppen zusammen.

Mit dem Button `VORSCHAU` aktivieren Sie unabhängig vom Darstellungsmodus bei Bildern und Dokumenten eine Vorschau. Die Größe der Vorschaubilder im Symbolmodus können Sie mit einem Schieberegler einstellen. Standardmäßig erstellt Dolphin keine Vorschau, wenn sich die Datei in einem Netzwerkverzeichnis befindet. Das Vorschauverhalten können Sie mit `EINSTELLUNGEN • DOLPHIN EINRICHTEN` im Dialogblatt `ALLGEMEIN • VORSCHAUEN` steuern.

Für Verschiebe- und Kopieroperationen kann der Innenbereich mit `TEILEN` vertikal geteilt werden, um zwei Verzeichnisse im selben Fenster darzustellen.

**Verzeichnispfad** Das aktuelle Verzeichnis wird in einer Navigationsleiste unterhalb des Menüs angezeigt. `[Strg]+[L]` schaltet zwischen zwei Ansichtsformen dieser Leiste um: Entweder

werden die einzelnen Verzeichnisse als Buttons dargestellt, was einen raschen Verzeichniswechsel erlaubt, oder das Verzeichnis wird in Textform angezeigt, was eine schnelle Eingabe eines anderen Verzeichnisses ermöglicht. Unabhängig von der gerade aktiven Ansichtsform können Sie mit **[F6]** ein neues Verzeichnis per Tastatur angeben.

Links, rechts und unterhalb des eigentlichen Fensterinhalts können Sie mit **EINSTELLUNGEN • SEITENLEISTEN** bzw. mit den Tasten **[F4]**, **[F7]**, **[F9]** und **[F11]** ein Terminal, die Verzeichnishierarchie, eine Liste häufig benötigter Orte sowie zusätzliche Informationen anzeigen. Zur Liste der Orte können Sie per Drag & Drop Lesezeichen für Verzeichnisse hinzufügen. **Seitenleisten**

Je nach Konfiguration wird Dolphin ohne Menü angezeigt. Die scheinbar fehlenden Kommandos sind über den Button **EINSTELLUNGEN** weiterhin verfügbar. Wenn Ihnen jedoch das herkömmliche Menü lieber ist, können Sie dieses mit **[Strg]+[M]** ein- und ausschalten. **Menü**

Eine Besonderheit betrifft die Markierung von Dateien: In der KDE-Grundeinstellung ist dazu ein einfacher Mausklick nicht geeignet, weil damit die Datei angezeigt oder ausgeführt wird. Sie müssen deswegen gleichzeitig **[Strg]** (für Mehrfachmarkierungen) oder **[⇧]** (für Bereichsmarkierungen) drücken. **Dateien markieren**

Noch eleganter ist ein weiterer Markierungsmodus: Wenn Sie den Mauszeiger einen Moment über einer Datei oder einem Verzeichnis ruhen lassen (*hover*), wird ein Plus-Zeichen eingeblendet. Ein Mausklick auf dieses Symbol markiert die Datei. Bei bereits markierten Zeichen erscheint ein rotes Verbotssymbol, mit dem Sie die Markierung wieder auflösen können.

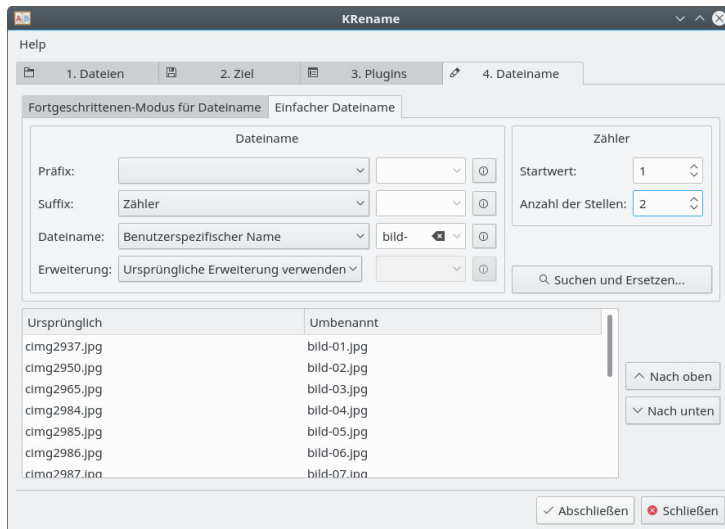
Wenn Sie Dateien und Verzeichnisse löschen, landen diese vorerst im Papierkorb. Um den Inhalt des Papierkorbs anzusehen, klicken Sie in der Seitenleiste **ORTE** (**[F9]**) den entsprechenden Eintrag an. Erst wenn Sie dort alle Objekte markieren und **[Entf]** drücken, werden die Dateien endgültig gelöscht. Um Dateien sofort unwiderruflich zu löschen, drücken Sie **[⇧]+[Entf]**. **Dateien löschen**

In den Dolphin-Einstellungen können Sie den maximalen Speicherbedarf für den Papierkorb limitieren oder veranlassen, dass dort befindliche Dateien nach einer bestimmten Zeit endgültig gelöscht werden.

Unter Linux gelten alle Dateien und Verzeichnisse, deren Namen mit einem Punkt beginnen, als verborgen. Dolphin zeigt diese Dateien normalerweise nicht an, es sei denn, Sie führen im Werkzeugmenü **VERSTECKTE DATEIEN ANZEIGEN** aus. Noch schneller können Sie die Anzeige verborgener Dateien mit **[Alt]+[.]** ein- und wieder ausschalten. **Verborgene Dateien**

**Zugriffsrechte** Damit nicht jeder Benutzer alle Dateien und Verzeichnisse lesen bzw. verändern kann, speichert Linux zu jeder Datei und zu jedem Verzeichnis den Besitzer sowie Zugriffsrechte. Das zugrunde liegende Konzept wird in [Abschnitt 11.5](#), »Zugriffsrechte, Benutzer und Gruppenzugehörigkeit«, ausführlich beschrieben. Um den Besitzer oder die Zugriffsrechte zu ändern, klicken Sie die Datei mit der rechten Maustaste an und führen **EIGENSCHAFTEN • BERECHTIGUNGEN** aus.

**Dateien umbenennen** Dolphin kann zwar mehrere Dateien umbenennen, bietet dabei aber wenig Flexibilität. Abhilfe bietet das relativ alte Programm *KRename* (siehe [Abbildung 6.5](#)): Damit können Sie Dateien effizient umbenennen, mit einer fortlaufenden Nummer austatten, das Datum in den Dateinamen einbauen etc. Zum Umbenennen sind sogar reguläre Ausdrücke erlaubt. Bevor die Dateien tatsächlich verändert werden, zeigt eine Vorschau die alten und neuen Dateinamen. KRename ist ein unentbehrliches Werkzeug, wenn Sie eine umfangreiche Sammlung von Audio- oder Bilddateien neu organisieren möchten! Bei vielen Distributionen müssen Sie das Programm vor der ersten Nutzung installieren (Paketname `krename`).



**Abbildung 6.5** Dateien umbenennen mit KRename

## Externe Datenträger und Netzwerkverzeichnisse

**Zugriff auf Datenträger** Die Seitenleiste **ORTE** (**F9**) enthält unter anderem eine Liste aller Festplattenpartitionen, die per Mausklick in das Dateisystem eingebunden werden können. Wenn Sie ein USB-Laufwerk anschließen, erscheint im Panel ein entsprechender Hinweis. Ein Mausklick öffnet dann den Dateimanager und zeigt den Inhalt des Datenträgers an. Bevor Sie das Kabel vom Laufwerk lösen, müssen Sie entweder im KDE-Menü das



Kontextmenükommando AUSWERFEN oder in Dolphin in der Seitenleiste ORTE das Kontextmenükommando EINBINDUNG LÖSEN ausführen.

Über die Seitenleiste ORTE bzw. durch die Adressangabe `smb://` können Sie auf das lokale Netzwerk zugreifen. Um direkt auf ein bestimmtes Verzeichnis auf einem Samba- oder Windows-Server zuzugreifen, verwenden Sie die Schreibweise `smb://servername/sharename`. Diese Schreibweise ist auch dann notwendig, wenn Dolphin im Netzwerk keine Windows-Server erkennt, was je nach Firewall- und Netzwerkkonfiguration häufig vorkommt.

Zugriff auf  
Netzwerk-  
verzeichnisse

#### Hinweis

Wenn Dolphin Windows- oder Samba-Server im lokalen Netzwerk nicht findet, ist möglicherweise die Firewall Ihrer Distribution schuld. Sowohl bei Fedora als auch bei SUSE verhindern die Standardeinstellungen der Firewall die Nutzung von Windows-Netzwerkverzeichnissen. Abhilfe schafft die richtige Konfiguration der Firewall.

Dolphin fragt jetzt nach dem Benutzernamen und dem Passwort für den Verbindungsaufbau zum Windows-Rechner oder Samba-Server. Diese Daten werden von *KDE Wallet* gespeichert, einem Programm zur Schlüsselverwaltung. Beim erstmaligen Start dieses Programms müssen Sie hierfür ein Zentralpasswort definieren. Das lohnt sich, weil KDE Wallet auch Web- und Mail-Passwörter verwaltet.

Passwort-  
verwaltung

Mit Dolphin können Sie auch über das sichere Protokoll SSH mit einem anderen Rechner kommunizieren und Dateien kopieren. Dazu geben Sie als Adresse `fish://username@rechnername/` ein. Nach dem Login zeigt Dolphin alle Dateien des externen Rechners an.

SSH

Zum Brennen von CDs/DVDs sieht KDE das Programm K3b vor. Es ist das vielseitigste Brennprogramm, das momentan unter Linux verfügbar ist. Der Funktionsreichtum des Programms begeistert selbst eingefleischte Gnome-Anhänger. Ein wenig abschreckend sind nur die bisweilen unübersichtlichen Menüs und Einstellungsdialoge. Aber keine Angst! Für Standardaufgaben, also beispielsweise für das Erstellen einer Backup-CD, sind diese Optionen nicht wichtig und können getrost ignoriert werden. K3b entscheidet sich praktisch immer für vernünftige Defaulteinstellungen.

DVDs brennen

## 6.3 KDE-Konfiguration

Zahllose Konfigurationsmodule sind in den SYSTEMEINSTELLUNGEN zusammengefasst (Kommando `systemsettings`, siehe [Abbildung 6.6](#)). Da es nicht immer ganz einfach ist, das richtige Modul zu finden, haben die KDE-Entwickler das Programm mit einer Suchfunktion ausgestattet, in der Sie nach Schlüsselwörtern (z. B. *Fenster*) suchen können.

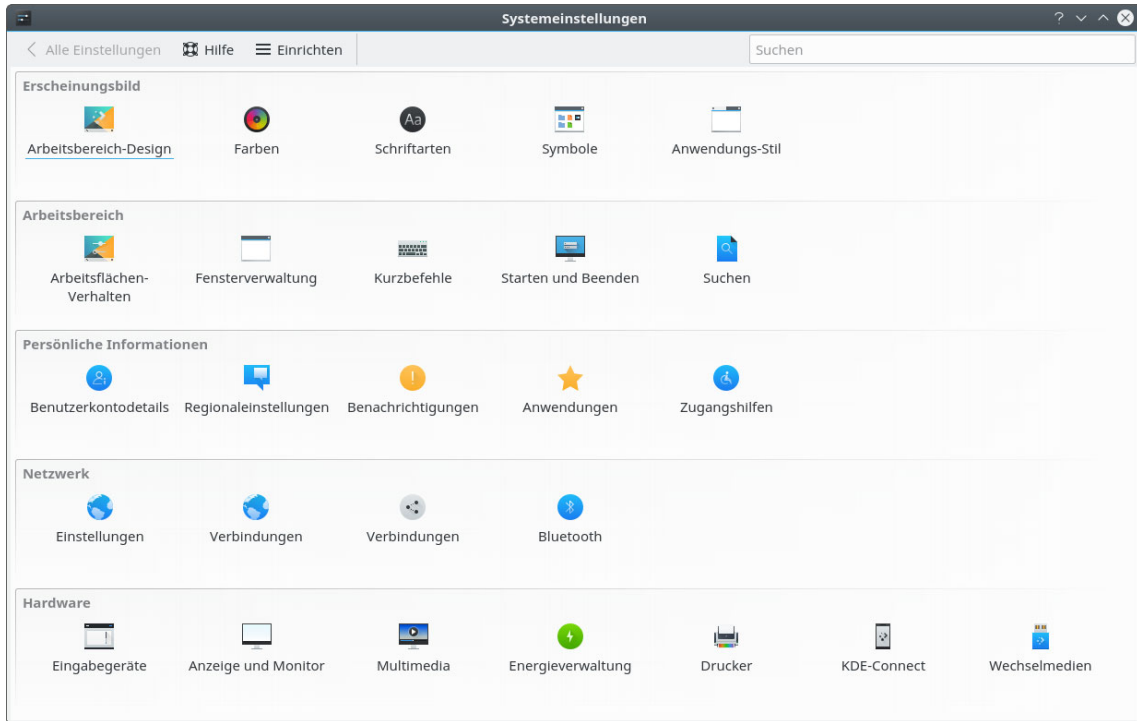


Abbildung 6.6 Das KDE-Kontrollzentrum

Beachten Sie bei der Bedienung der Module, dass geänderte Einstellungen erst wirksam werden, sobald sie durch den Button ANWENDEN bestätigt werden. In diesem Punkt unterscheidet sich KDE deutlich von Gnome, wo geänderte Einstellungen sofort aktiv sind.

KDE enthält auch Konfigurationsmodule, die nicht den Desktop betreffen, sondern Systemeinstellungen, also z. B. für die Netzwerkkonfiguration oder den Drucker. Diese Module sind bei solchen Distributionen sehr hilfreich, die keine eigenen Konfigurationswerkzeuge anbieten. Unter SUSE sollten Sie aber YaST zur Konfiguration vorziehen. Eventuell müssen Sie bei Konfigurationsmodulen, die Systemeinstellungen betreffen, zuerst unter Angabe des root-Passworts in einen Administratormodus wechseln.

**Konfigurationsverzeichnisse** Viele KDE-Programme sind konform zum XDG-Standard und verwenden die in [Abschnitt 5.8](#), »Gnome-Intern«, bereits aufgelisteten XDG-Verzeichnisse:

|                          |                               |
|--------------------------|-------------------------------|
| ~/.cache/                | (Cache)                       |
| ~/.config/progname/      | (Konfigurationseinstellungen) |
| ~/.local/share/progname/ | (Benutzerdaten)               |

Einige KDE-Programme speichern Ihre Einstellungen stattdessen in Dateien des Verzeichnisses `.kde`. Darin existieren unter anderem die folgenden Unterverzeichnisse:

|                                   |  |
|-----------------------------------|--|
| <code>~/.kde/Autostart/</code>    | (persönliche Autostart-Programme)      |
| <code>~/.kde/share/config/</code> | (Konfigurationseinstellungen)          |
| <code>~/.kde/share/apps/</code>   | (sonstige programmspezifische Dateien) |

Nach dem Start des Rechners müssen Sie sich einloggen, bevor Sie mit der Arbeit beginnen. Wenn Sie der einzige Benutzer des Rechners sind und keine Gefahr besteht, dass andere Personen Zugang zum Rechner haben, können Sie den ersten Login beim Rechnerstart automatisieren. Die Auto-Login-Funktion steuern Sie im Dialogblatt **STARTEN UND BEENDEN • ANMELDEBILDSCHIRM**. Hinter den Kulissen ist normalerweise der KDE-Display-Manager (`sddm`) für den Login verantwortlich.

Auto-Login

Davon abweichend verwendet Kubuntu den Display Manager LightDM (siehe [Abschnitt 21.5](#), »Start des Grafiksystems«). Bei SUSE-Distributionen erfolgt die Konfiguration der Auto-Login-Funktion desktop-unabhängig in der Datei `/etc/sysconfig/displaymanager`. Versuchen Sie nicht, den Auto-Login mit KDE-Werkzeugen zu verändern: Ihre Einstellungen werden bei nächster Gelegenheit von YaST überschrieben!

Bei jedem Logout werden alle laufenden Programme beendet. Beim nächsten Login bemüht sich KDE, die Programme, die zuletzt gelaufen sind, wieder zu starten, die letzte Sitzung also wiederherzustellen. Für KDE-Programme funktioniert das zumeist gut, für alle anderen Programme nur mit Einschränkungen oder gar nicht. Details zu diesem Verhalten stellen Sie im Modul **STARTEN UND BEENDEN • AUTOSTART** der Systemeinstellungen ein. Der zuletzt gültige Zustand der Sitzung wird in Dateien des Verzeichnisses `.kde/share/config/session` gespeichert.

Autostart

Unabhängig von der Sitzungsverwaltung können Sie im Verzeichnis `.kde/Autostart` Programme angeben, die nach jedem Login gestartet werden sollen. KDE erwartet in diesem Verzeichnis `*.desktop`-Dateien, die das zu startende Programm beschreiben. Am einfachsten erzeugen Sie derartige Dateien, indem Sie das Verzeichnis `.kde/Autostart` mit dem Dateimanager Dolphin öffnen und das gewünschte Programm aus dem KDE-Menü per Drag & Drop dorthin kopieren. Alternativ können Sie zur Konfiguration das Systemsteuerungsmodul **STARTEN UND BEENDEN** einsetzen.

Wenn Sie beide Mechanismen, also die Sitzungsverwaltung und Autostart-Verzeichnisse, parallel nutzen, kann es vorkommen, dass ein bereits laufendes Programm doppelt gestartet wird. Beachten Sie auch, dass KDE *mehrere* Autostart-Verzeichnisse berücksichtigt:

|                                    |   |
|------------------------------------|---|
| <code>~/.kde/Autostart/</code>     | (persönliche Autostart-Programme)               |
| <code>/usr/share/autostart/</code> | (globale Autostart-Programme für KDE)           |
| <code>/etc/xdg/autostart/</code>   | (globale Autostart-Programme für Gnome und KDE) |

**Monitor-konfiguration** Mit dem Modul ANZEIGE UND MONITOR stellen Sie ein, ob und wie mehrere Monitore bzw. Signalausgänge genutzt werden sollen und in welcher Auflösung Sie arbeiten möchten. Weitere Details zur Monitorkonfiguration finden Sie in Abschnitt 21.7, »Dynamische Konfigurationsänderungen mit RandR«.

**Desktop-Aussehen** Es gibt unzählige Möglichkeiten, auf das Aussehen (die Optik) des Desktops Einfluss zu nehmen. Wenn Sie Zeit und Lust haben, können Sie Stunden damit verbringen, den Desktop nach Ihren eigenen Vorstellungen zu gestalten.

- ▶ **Desktop-Hintergrund:** Um den Hintergrund einzustellen, klicken Sie mit der rechten Maustaste auf den Desktop und führen ARBEITSFLÄCHE EINRICHTEN aus. Anschließend stellen Sie ein Hintergrundbild oder eine Hintergrundfarbe ein.
- ▶ **Desktop-Design:** Im Modul ARBEITSBEREICH-DESIGN können Sie im Dialogblatt ARBEITSFLÄCHENDESIGN das Thema für die Arbeitsfläche einstellen. Es bestimmt die Grundeinstellungen für das Aussehen des Panels, des KDE-Menüs, der Fensterdekoration etc. sowie die hierfür eingesetzten Farben. Mit NEUES DESIGN HERUNTERLADEN können Sie weitere Designs von der Website <https://store.kde.org> herunterladen. Vergessen Sie nicht, das neue Design durch ANWENDEN auch zu aktivieren!
- ▶ **Gestaltung der Steuerelemente:** Im Systemeinstellungsmodul ANWENDUNGS-STIL können Sie zwischen mehreren Layoutvarianten für die optische Gestaltung von Buttons, Optionsfeldern, Bildlaufleisten etc. auswählen.
- ▶ **Gestaltung der Fenster:** Im gerade erwähnten Modul ANWENDUNGS-STIL können Sie im Dialogblatt FENSTERDEKORATION zwischen mehreren Fensterlayouts wählen. Damit ändern Sie die Farbe und die Gestaltung des Fensterrahmens. Im Dialogblatt KNÖPFE können Sie außerdem einstellen, wo in der Fensterleiste welche Buttons platziert werden sollen.
- ▶ **Farben:** Die Farben für die Fensterrahmen, das Menü, das Panel etc. sind durch das ARBEITSBEREICH-DESIGN und die Fensterdekorationsvariante vorgegeben. Das Modul FARBE ermöglicht die Auswahl anderer Farbschemas.

Abbildung 6.7 zeigt ein Beispiel dafür, wie stark Sie das Aussehen des KDE-Desktops mit wenigen Optionen verändern können:

- ▶ Das Panel wurde am linken Bildschirmrand angeordnet, und die Panel-Elemente wurden auf das Minimum reduziert.
- ▶ Die Anzahl der Fensterbuttons wurde auf zwei reduziert. Durch die fette Schrift ist der Fenstertitel zudem besser lesbar.
- ▶ Der Desktop wurde von allen Plasmoids befreit.

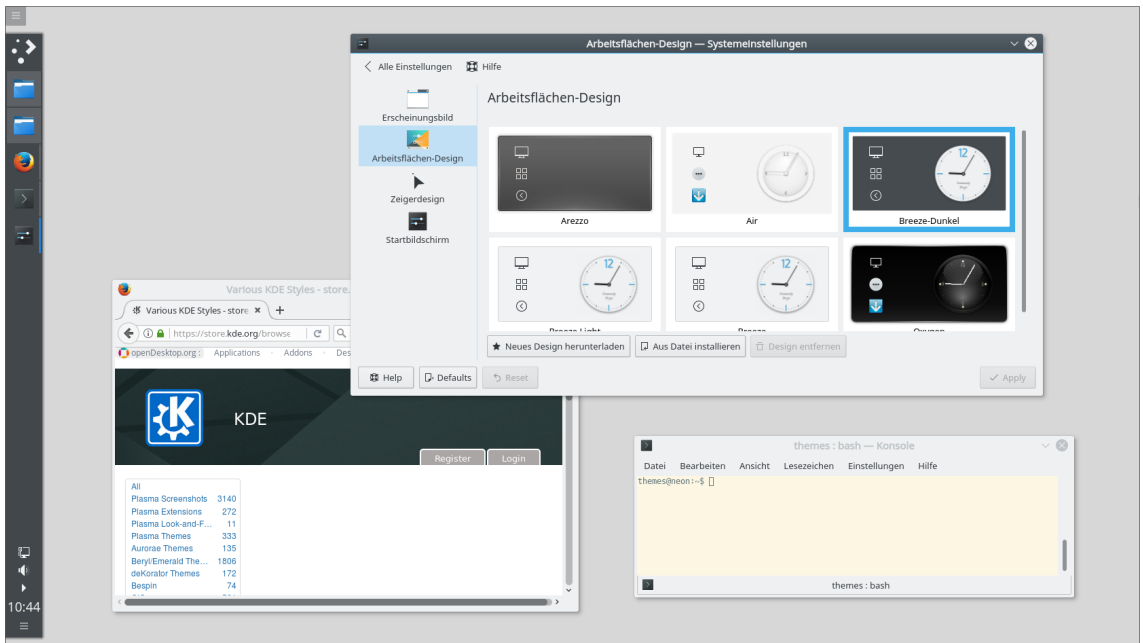


Abbildung 6.7 Reduzierte, klare KDE-Konfiguration nach dem Geschmack des Autors

Die Einstellung des Tastaturlayouts nehmen Sie im Modul EINGABEGERÄTE vor. Im Dialogblatt TASTATUR • BELEGUNGEN können Sie mehrere Tastaturlayouts zum raschen Wechseln einrichten. Sie können dort auch unzählige Optionen einstellen, die z. B. steuern, welche Funktionen die Tasten **Win**, **Alt** und **Strg** haben.

Tastatur und Maus

Auch die Optionen zur Konfiguration des Maus- bzw. Touchpad-Verhaltens finden Sie im Systemeinstellungsmodul EINGABEGERÄTE. Dort können Sie nicht nur das Doppelklickverhalten einstellen, sondern auch die gewünschte Scrollrichtung angeben: normal oder umgekehrt, d. h. wie auf Smartphones bzw. unter macOS.

Das KDE-Kontrollzentrum kann mit dem Modul DRUCKER auch zur Konfiguration von Druckern verwendet werden. Den Assistenten zur Druckerkonfiguration starten Sie mit NEUER DRUCKER. Im ersten Schritt geben Sie Ihren Druckertyp (z. B. NETZWERKDRUCKER) an, im zweiten Schritt den Hersteller und das Modell. Zum Abschluss der Konfiguration müssen Sie dem Drucker einen Namen geben.

Druckerkonfiguration

Wenn nach einem Klick auf eine MP3-Datei das Programm Amarok erscheint, dann sind hierfür die MIME-Einstellungen von KDE verantwortlich. MIME steht für *Multipurpose Internet Mail Extensions* und ist eine Datenbank, die eine Zuordnung zwischen Dateitypen und Programmen herstellt. Sie können die Liste der MIME-Dateitypen im Dialogblatt ANWENDUNGEN • DATEIZUORDNUNGEN der Systemeinstellungen verändern. Einzelnen Dateitypen können mehrere Programme zugeordnet

MIME

werden. Das in der Rangfolge am höchsten stehende Programm wird gestartet, wenn die Datei durch einen Mausklick geöffnet wird. Alle anderen Programme stehen zur Wahl, wenn Sie mit der rechten Maustaste ÖFFNEN MIT ausführen.

#### Standardprogramme einstellen

Standardmäßig verwendet KDE zumeist Konqueror als Webbrowser, KMail bzw. Kontact als E-Mail-Programm und konsole als Konsolenprogramm. Wenn Sie möchten, dass KDE beim Anklicken entsprechender Links andere Programme startet, finden Sie Einstellmöglichkeiten im Modul ANWENDUNGEN der Systemeinstellungen.

## 6.4 Unity

Unity ist eine von Canonical entwickelte Benutzeroberfläche, die standardmäßig in allen Ubuntu-Versionen von 10.10 bis 17.04 zum Einsatz kam. Anfang 2017 stoppte Canonical die Weiterentwicklung. Für Ubuntu 16.04 LTS wird Unity aber noch bis 2021 gewartet. Möglicherweise wird es in Zukunft parallel zum offiziellen Ubuntu auf Gnome-Basis auch eine von der Community getragene Unity-Variante geben. Kurz und gut: Die Zukunft von Unity sieht zwar nicht rosig aus, aber aktuell findet Unity noch eine große Verbreitung.

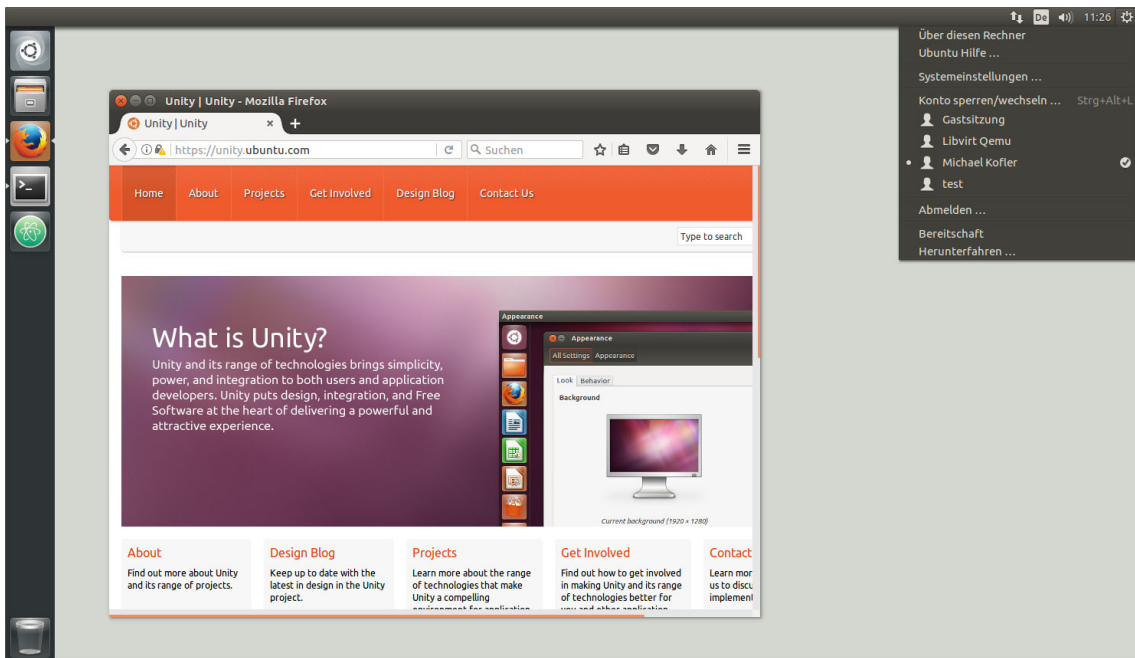


Abbildung 6.8 Der Unity-Desktop von Ubuntu

Genau genommen ist Unity kein komplett neuer Desktop; vielmehr basiert Unity auf Gnome. Canonical hat allerdings etliche Komponenten von Gnome ausgetauscht, unter anderem die Gnome Shell und die Systemeinstellungen. Stattdessen gibt es sogenannte Indikatorprogramme. Das sind Miniprogramme, die im Panel angezeigt werden.

## Komponenten

Unity ist unter Ubuntu für die Verwaltung der Fenster sowie für die Darstellung der Taskleiste (»Dock«) und des Startmenüs (»Dash«) zuständig. Technisch gesehen ist Unity eine Erweiterung zum 3D-Window-Manager Compiz. Unity ersetzt in dieser Funktion die Gnome Shell, aber keinesfalls das gesamte Gnome-Projekt! Nahezu alle Anwendungsprogramme unter Ubuntu basieren weiterhin auf Gnome – unter anderem der Dateimanager, der Audio-Player, das Fotoverwaltungsprogramm, das Terminal sowie viele Konfigurationswerkzeuge und sonstige Hilfsprogramme.

Außer durch Unity zeichnet sich der Ubuntu-Desktop durch weitere Ubuntu-spezifische Programme und Ergänzungen aus:

- ▶ **Indikatoren:** Bei den Indikatorprogrammen handelt es sich um Hintergrundprogramme, die durch ein Icon am rechten oberen Bildschirmrand dargestellt werden. Sie ersetzen Gnome-Applets bzw. KDE-Plasmoids.
- ▶ **Systemeinstellungen:** Um Ubuntu-spezifische Einstellungen besser in die Systemeinstellungen zu integrieren, hat Canonical die Gnome-Systemeinstellungen durch ein eigenes Programm ersetzt. Einige Einstellungsmodulare wurden unverändert von Gnome übernommen, andere neu entwickelt.
- ▶ **Web-Apps:** Web-Apps lassen einige im Webbrowser ausgeführte Seiten aussehen, als würde es sich um eigenständige Programme handeln. Diese 2012 eingeführte Technik funktionierte unter Ubuntu 16.04 allerdings nicht mehr richtig. Der einzige Grund, warum die entsprechenden Pakete nicht ganz entfernt wurden, war offensichtlich ein Kooperationsvertrag mit Amazon: Nach der Installation von Ubuntu enthält das Dock ein Icon, das in einer Web-App zur Webseite <https://amazon.com> führt. Die meisten Ubuntu-Anwender haben dieses Icon jedoch sofort aus dem Dock entfernt.

## Startmenü (Dash)


In das Ubuntu-Startmenü, das in der englischen Dokumentation auch *Dash* genannt wird, gelangen Sie durch einen Mausklick auf das Ubuntu-Icon oder mit . Das Startmenü besteht aus mehreren Dialogblättern zur Auswahl von Programmen, Dateien und Ordnern, Musiktiteln und Videos (siehe [Abbildung 6.9](#)).



Abbildung 6.9 Programme im Ubuntu-Startmenü

**Linsen (Lenses)** Die Dialogblätter im Startmenü ermöglichen gewissermaßen verschiedene Ansichten auf den Inhalt Ihres Computers und heißen deswegen »Linsen« (*Lenses*). Die Auswahl der Ansicht erfolgt wahlweise durch einen Mausklick auf die kleinen Icons am unteren Rand des Dash, mit `Strg` + `↩` bzw. mit den Cursortasten und `↩`.

- ▶ Die Standardansicht (*Home*) dient zum Start oft benötigter Programme. Hier werden die zuletzt am häufigsten eingesetzten Programme angezeigt, soweit sich diese nicht ohnehin im Dock befinden. Sobald Sie im Suchfeld einige Buchstaben eingeben, zeigt das Startmenü Programme mit den entsprechenden Anfangsbuchstaben an. Bei der Suche können Sie sowohl den Programm- bzw. Kommandonamen (z. B. `gnome-terminal`) als auch die deutsche Programmbezeichnung eingeben (z. B. *Terminal*). Zur Verfeinerung der Suchergebnisse gibt es diverse Filtermöglichkeiten.
- ▶ Die Ansicht ANWENDUNGEN hilft bei der Suche nach Programmen. Die Besonderheit dieser Ansicht besteht darin, dass in einer eigenen Gruppe noch gar nicht installierte Programme aufgelistet werden. Wenn Sie ein Icon aus dieser Gruppe auswählen, wird das Ubuntu Software-Center gestartet. Dort können Sie das gewünschte Programm dann herunterladen und installieren.
- ▶ Die Ansicht DATEIEN UND ORDNER hilft bei der gezielten Suche nach Dateien, die Sie in letzter Zeit bearbeitet haben. Durch Filter können Sie die Suche auf bestimmte Dateitypen, Dateigrößen oder Bearbeitungszeiträume eingrenzen, z. B. auf die letzten 30 Tage.



- ▶ In den Ansichten MUSIK, VIDEOS und FOTOS können Sie Ihre lokale Musik-, Video- und Fotosammlung durchsuchen. Außerdem bietet das Dash nun MP3-Dateien zum Kauf bzw. YouTube-Videos zum Ansehen an.

### Dock (Seitenleiste, Launcher)

Die Seitenleiste am linken Bildschirmrand hat zwei Funktionen: Sie ermöglicht einerseits einen raschen Start häufig benötigter Programme und hilft andererseits beim Wechsel zwischen den laufenden Programmen. In der offiziellen englischen Dokumentation wird die Seitenleiste *Launcher* genannt, gebräuchlicher ist aber der von Apple geprägte Begriff *Dock*.

Anfänglich enthält das Dock eine von den Ubuntu-Entwicklern vordefinierte Liste von Icons. Wenn Sie die Auswahl oder Reihenfolge der Icons verändern möchten, gehen Sie wie folgt vor:

- ▶ **Icon hinzufügen:** Um ein Icon hinzuzufügen, starten Sie zuerst das gewünschte Programm über das Startmenü. Das Icon erscheint nun im Dock, solange das Programm läuft. Damit das Icon im Dock bleibt, auch wenn das Programm nicht mehr läuft, klicken Sie es mit der rechten Maustaste an und wählen das Menükommando **IM STARTER BEHALTEN** aus. Es ist auch möglich, Icons aus dem Startmenü oder aus dem Dialog **ANWENDUNGEN** per Drag & Drop hinzuzufügen.
- ▶ **Icon entfernen:** Um ein selten benötigtes Icon zu entfernen, klicken Sie es mit der rechten Maustaste an und deaktivieren den Menüeintrag **IM STARTER BEHALTEN**. Eine andere Möglichkeit besteht darin, das Icon zuerst nach rechts aus dem Dock hinauszuziehen und es dann in den Mülleimer am unteren Ende des Docks zu bewegen.
- ▶ **Icon verschieben:** Um die Icon-Reihenfolge zu ändern, ziehen Sie das Icon nach rechts aus dem Dock hinaus und bewegen es dann an der gewünschten neuen Position wieder in das Dock hinein.

Bei laufenden Programmen geben weiße Dreiecke links neben dem Icon an, wie viele Fenster offen sind. (Beachten Sie aber, dass ein Programm auch laufen kann, obwohl kein Fenster offen ist – etwa beim Audio-Player. In diesem Fall ist im Dock nicht erkennbar, ob das Programm läuft oder nicht.) Das gerade aktive Programm ist im Dock durch ein weißes Dreieck rechts neben dem Icon gekennzeichnet. Wenn ein Programm Ihre Aufmerksamkeit erfordert, wird es im Dock durch einen blauen Indikator hervorgehoben.

## Menüleiste (Panel) mit Indikatoren

**Panel** Unter Ubuntu gibt es ein Panel am oberen Bildschirmrand. Dieses Panel, das vielfach auch einfach als »Menüleiste« bezeichnet wird, enthält mehrere Elemente:

- ▶ **Programmname und Menüs:** Im linken Rand des Panels wird der Name des gerade aktiven Programms angezeigt. Diese reichlich überflüssige Information wird durch das Menü des Programms ersetzt, wenn Sie die Maus in das Panel bewegen oder **[Alt]** drücken. Ubuntu folgt in diesem Punkt also Apple, dessen Betriebssystem macOS ebenfalls keine Menüs in der Fensterleiste vorsieht und stattdessen das Menü des gerade aktiven Programms zentral links oben auf dem Bildschirm anzeigt. Auf Notebooks mit kleinen Bildschirmen ist das praktisch, auf großen Bildschirmen weniger. Gegebenenfalls aktivieren Sie in den Systemeinstellungen im Dialogblatt DARSTELLUNG • VERHALTEN einfach die Option ZEIGE DIE MENÜS IN DER TITELLEISTE DES FENSTERS.
- ▶ **Indikatoren:** Der rechte Rand des Panels ist den sogenannten Indikatoren vorbehalten. Diese Miniprogramme zeigen diverse Statusinformationen an: die Uhrzeit, die Netzwerkverbindung, die eingestellte Lautstärke etc. Alle Indikatoranwendungen sind mit einem Menü ausgestattet, das zur Steuerung diverser Funktionen dient.



Es ist nicht möglich, die Reihenfolge der Indikatoren zu verändern oder sie aus der Menüleiste zu entfernen. Wenn Sie einzelne Indikatoren nicht verwenden möchten, müssen Sie das betreffende Paket deinstallieren. Wenn Sie umgekehrt einen zusätzlichen Indikator wünschen, müssen Sie dessen Paket suchen, installieren und ausführen; die Auswahl ist allerdings klein.









**Indikatoren** Das Panel zeigt rechts unter anderem den Status der Netzwerkverbindung, die Lautstärke des Audio-Systems, den Zustand des Notebook-Akkus, den Bluetooth-Status, den Eingang neuer Nachrichten, die Uhrzeit und das Icon des Systemmenüs an. Über die zugeordneten Menüs können Sie die Netzwerkverbindung konfigurieren, den Audio-Player steuern, Chat-, E-Mail- und Social-Messaging-Clients bedienen, Termine verwalten, Ihren Online-Status verändern, sich abmelden oder die Systemeinstellungen starten.

Die für die Anzeige dieser Statusinformationen und Menüs verantwortlichen Programme heißen Indikatorprogramme bzw. Indikatormenüs. Ihre Anordnung kann nicht verändert werden. Das Panel bietet auch keine Möglichkeit, zusätzliche Indikatorprogramme hinzuzufügen. Wenn Sie das möchten, müssen Sie zuerst ein entsprechendes Paket installieren und das darin enthaltene Programm dann starten. Dazu suchen Sie im Ubuntu Software-Center oder in Synaptic nach Paketen, die mit `indicator` beginnen. An der Technik interessierte Benutzer werden z. B. am Systemmonitor Gefallen finden (Paketname `indicator-multiloop`).

Der einzige Weg, um nicht benötigte Indikatoren zu entfernen, ist die Deinstallation des betreffenden Pakets. Am ehesten besteht dieser Wunsch beim Messages-Menü, das den Mail- und Twitter-Status anzeigt. Das betreffende Paket heißt `indicator-messages`.

## Tastatur, Maus, HUD-Menü

Zu den Stärken von Unity zählt die Möglichkeit, den Desktop vollständig per Tastatur zu steuern (siehe [Tabelle 6.1](#)). Das ermöglicht eine sehr effiziente Bedienung. Es ist nicht nötig, alle Tastenkürzel auswendig zu lernen! Drücken Sie einfach für zwei Sekunden , dann erscheint eine Zusammenfassung der wichtigsten Kürzel.  wird dort als `Super` bezeichnet, das sollte Sie aber nicht weiter irritieren.

| Tastenkürzel  | Bedeutung  |
|---|--|
|                                        | öffnet das Startmenü (Dash) zum Start von Programmen. Wenn Sie die Windows-Taste länger gedrückt halten, werden die Icons nummeriert.  |
|  + <code>1</code> , <code>2</code> ... | aktiviert das erste, zweite ... Programm im Dock. Wenn Sie zusätzlich  drücken, wird eine neue Instanz eines bereits laufenden Programms gestartet. |
|  + <code>A</code>                      | öffnet das Dash-Fenster ANWENDUNGEN zum Start vorhandener bzw. zur Installation neuer Programme.   |
|  + <code>F</code>                      | öffnet das Dash-Fenster DATEIEN UND VERZEICHNISSE zur Auswahl zuletzt benutzter Dateien.   |
|  + <code>S</code>                    | aktiviert den Arbeitsflächenumschalter.  |
|  + <code>T</code>                    | öffnet den Papierkorb (Trash).   |
|  + <code>W</code>                    | aktiviert die Exposé-Ansicht mit verkleinerten Darstellungen aller offenen Fenster.  |
| <code>Alt</code> + <code>t</code>   | zeigt im Panel das Menü des aktiven Programms an.  |
| <code>Alt</code> + <code>F1</code>  | bewegt den Eingabefokus in das Dock.   |
| <code>Alt</code> + <code>F2</code>  | öffnet einen Dialog zum raschen Start eines Programms, dessen Namen Sie per Tastatur eingeben.   |
| <code>Alt</code> + <code>F7</code>  | ermöglicht es, das aktuelle Fenster mit den Cursortasten nach links, rechts, oben oder unten zu verschieben.   |
| <code>Alt</code> + <code>F8</code>  | ermöglicht es, die Größe des aktuellen Fensters mit den Cursortasten zu verändern.   |
| <code>Alt</code> + <code>F10</code>   | zeigt das Menü an.   |

**Tabelle 6.1** Wichtige Unity-Tastenkürzel



| Tastenkürzel   | Bedeutung   |
|--|---|
| <b>Strg</b> +  + <b>D</b>                 | minimiert alle Fenster bzw. öffnet sie wieder.                |
| <b>Strg</b> + <b>Alt</b> + Cursortaste   | wechselt in eine andere Arbeitsfläche.                        |
|  + <b>Strg</b> + <b>Alt</b> + Cursortaste | verschiebt das aktuelle Fenster in eine andere Arbeitsfläche. |
| <b>Strg</b> + <b>Alt</b> + <b>T</b>  | öffnet ein Terminal-Fenster.                                  |

Tabelle 6.1 Wichtige Unity-Tastenkürzel (Forts.)

**Maus** Innerhalb des Docks erfüllt die Maus diverse Zusatzfunktionen:

- ▶ Ein Mausklick auf ein Icon startet erwartungsgemäß das betreffende Programm.
- ▶ Ein Mausklick auf das Icon eines bereits laufenden Programms aktiviert dieses und bringt seine Fenster in den Vordergrund. Wenn das Programm bereits aktiv ist, bewirkt der Mausklick einen Exposé-Effekt, d. h., alle Fenster des Programms werden in Miniaturansicht nebeneinander dargestellt. Per Mausklick kann dann das gewünschte Fenster aktiviert werden.
- ▶ Mit der rechten Maustaste gelangen Sie in ein Kontextmenü des Icons. Darin können Sie das Programm beenden, die Verankerung im Dock einstellen und bei einigen Programmen auch diverse andere Kommandos ausführen.
- ▶ Um eine neue Instanz eines bereits laufenden Programms zu starten (z. B. ein weiteres Webbrowser- oder Terminal-Fenster), klicken Sie das Icon mit der mittleren Maustaste an.
- ▶ Wenn der Platz im Dock nicht ausreicht, um alle Icons vollständig anzuzeigen, können Sie mit der Maus und besonders gut mit einem Mousrad durch die Icons scrollen. In solchen Fällen ist es zweckmäßig, die Icon-Größe im Modul DARSTELLUNG der Systemeinstellungen zu reduzieren.
- ▶ Bei manchen Programmen ist es möglich, Dateien aus dem Dateimanager per Drag & Drop in das jeweilige Icon zu verschieben, um diese Datei zu öffnen. Wenn Sie beispielsweise eine MP3-Datei über dem Icon des Audio-Players fallen lassen, wird die Datei abgespielt.

**HUD-Menüs** Gewöhnliche Menüs werden in Ubuntu nicht in der Fensterleiste angezeigt, sondern im Panel – und auch dort nur, wenn Sie die Maus dorthin bewegen oder circa eine Sekunde lang **Alt** drücken.

Daneben gibt es in Ubuntu die sogenannten HUD-Menüs: Wenn Sie kurz **Alt** drücken, erscheint ein Eingabefeld. Nach der Eingabe einiger Zeichen zeigt das Eingabefeld alle passenden Menükommandos an, aus denen Sie dann eines mit den Cursortasten auswählen können. Abbildung 6.10 zeigt alle Menükommandos von

Firefox, in denen der Begriff *Datei* vorkommt. Nach einer kurzen Gewöhnungszeit ist diese Art der Menübedienung effizient, zumal öfter genutzte Menükommandos in der Liste der Kommandos zuerst angezeigt werden.

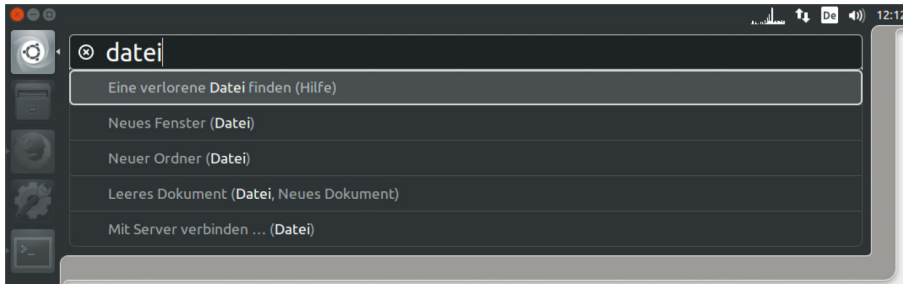


Abbildung 6.10 HUD-Menüs

Die Abkürzung HUD steht für *Head-up Display* und bezieht sich auf den Umstand, dass das Menü losgelöst vom Programmfenster bedient wird. Das HUD-Menü-Konzept ist durchaus originell, kann das gewöhnliche Menü aber nicht vollständig ersetzen: Es gibt keine Möglichkeit, durch alle Menüeinträge zu navigieren und ein Menükommando zu suchen, dessen Name bzw. Position unbekannt ist.

## Unity-Konfiguration

Ubuntu verwendet eine eigene Variante der Gnome-Systemeinstellungen (siehe [Abbildung 6.11](#)). Dort finden Sie neben einigen aus Gnome vertrauten Modulen noch weitere, die Ubuntu-spezifisch sind. Der schnellste Weg in die Systemeinstellungen führt über das gleichnamige Kommando im Systemmenü, das Sie durch einen Klick auf das Zahnrad-Icon ganz rechts im Panel öffnen.

System-  
einstellungen

Die wenigen Unity-Einstellungen finden Sie im Modul DARSTELLUNG. Dort können Sie den Desktop-Hintergrund, die Gestaltung der Fensterrahmen (THEMA) und die Breite des Docks einstellen (GRÖSSE DER STARTERSYMBOLS). Im Dialogblatt VERHALTEN gibt es außerdem noch einige Optionen, mit denen Sie das Startmenü automatisch ein- und ausblenden sowie Menüs wahlweise im Panel oder direkt im Fenster anzeigen lassen können.

In den Unity-Systemeinstellungen fehlt die Möglichkeit, Autostart-Programme einzurichten. Abhilfe: Führen Sie `[Alt]+[F2] gnome-session-properties` aus!

Autostart

In einem Punkt waren die Ubuntu-Systemeinstellungen denen von Gnome um Jahre voraus: Das Modul ANZEIGEGERÄTE bietet die Möglichkeit, einen Skalierungsfaktor für die Darstellung von Menüs und Fenstern einzustellen. Auch wenn die Lösung nicht perfekt war, gelang das Arbeiten auf hochauflösenden Bildschirmen in Ubuntu doch besser und unkomplizierter als mit den meisten anderen Distributionen.

High-DPI-  
Montore

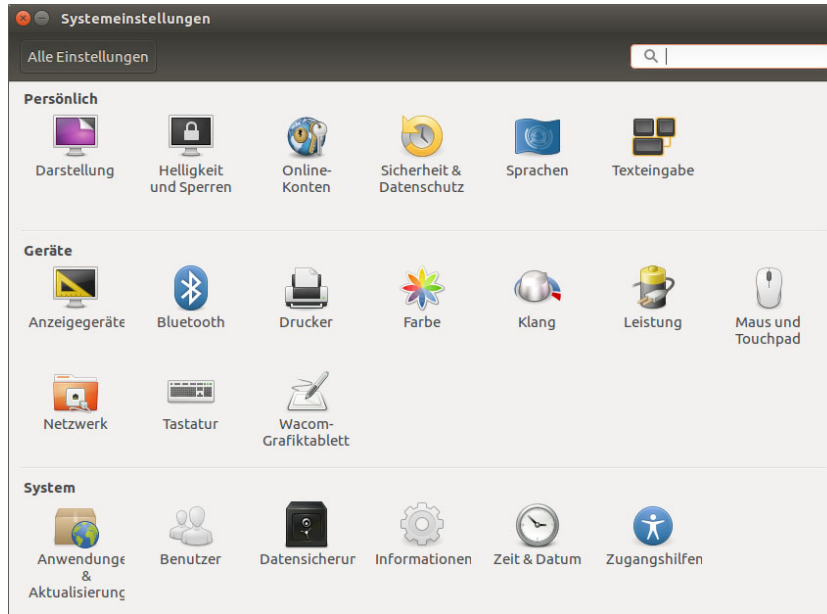


Abbildung 6.11 Ubuntu-Systemeinstellungen

**Unity Tweak Tool** Gewissermaßen ein Gegenstück zum Gnome Tweak Tool ist das offiziell nicht unterstützte, aber in den Paketquellen enthaltene Programm *Unity Tweak Tool* (Paketname `unity-tweak-tool`). Es bietet eine Menge zusätzlicher Einstelloptionen. Unter anderem können Sie damit im Modul `UNITY • STARTER` das Dock an den unteren Bildschirmrand platzieren – und sich so einen lang gehegten Wunsch vieler Ubuntu-Fans erfüllen.

# Kapitel 7

## Desktop-Apps

In diesem Kapitel stelle ich Ihnen im Schnelldurchlauf wichtige Desktop-Programme für Linux vor. Einige dieser Programme wie Firefox oder Google Chrome kennen Sie sicherlich schon von Windows oder macOS: Zum einen wurden manche ursprünglich für Linux entwickelte Open-Source-Programme später auch für Windows und macOS portiert. Zum anderen gibt es eine ganze Reihe von (oft kommerziellen) Programmen, die ursprünglich für Windows gedacht waren, mittlerweile aber plattformübergreifend angeboten werden.

Vorweg ein Überblick über die vorgestellten Programme:

- ▶ Webbrowser: Firefox, Google Chrome bzw. Chromium
- ▶ E-Mail-Clients: Thunderbird, Evolution, Kontact bzw. KMail, Geary
- ▶ Datenaustausch im Internet: Dropbox, FileZilla, Transmission
- ▶ Fotos verwalten: Shotwell, Digikam
- ▶ Bildbearbeitung: GIMP, RawTherapee, Darktable, Luminance
- ▶ Audio-Player: Amarok, Rhythmbox, Spotify
- ▶ Video-Player: VLC
- ▶ Audio- und Video-Tools: Audacity, Sound Converter, EasyTag, Sound Juicer, Handbrake
- ▶ Screenshots und Screencasts: Shutter, recordMyDesktop, Kazam

Es versteht sich von selbst, dass ich mit diesem Kapitel in keinster Weise einen Anspruch auf Vollständigkeit erhebe! Bewusst verzichtet habe ich auf die Beschreibung des Office-Pakets LibreOffice, das auch unter Windows und macOS weit verbreitet ist und dessen Bedienung unter Linux exakt so erfolgt wie unter anderen Betriebssystemen.

## 7.1 Firefox

**Geschichte und Namen** Firefox ist der populärste Webbrowser für Linux und wird von den meisten Distributionen standardmäßig installiert. Firefox ist aus dem Netscape Navigator hervorgegangen, der ursprünglich auch einen E-Mail-Client und einen HTML-Editor enthielt. Später wurde der Code in Komponenten zerlegt – so entstanden Firefox und Thunderbird.

**ESR-Versionen** Seit Mitte 2011 erscheint alle sechs Wochen eine neue Firefox-Version, die gleichermaßen Sicherheits-Updates und neue Features enthält. Ältere Firefox-Versionen werden nicht gewartet.

Neben den »gewöhnlichen« Firefox-Versionen gibt es auch spezielle ESR-Versionen (*Extended Support Release*) für den kommerziellen Einsatz. Der Vorteil von ESR-Versionen besteht darin, dass diese circa ein dreiviertel Jahr lang mit Sicherheits-Updates versorgt werden. Das erspart ESR-Anwendern ständig neue Firefox-Versionen. Die Firefox-ESR-Versionen kommen z. B. in Debian und in Enterprise-Distributionen wie CentOS und RHEL zum Einsatz.

**Updates** Anders als unter Windows, wo Firefox selbst für seine Updates zuständig ist, kümmert sich in Linux die Paketverwaltung um die regelmäßige Aktualisierung von Firefox. Alle größeren Distributoren stellen dazu während der Lebenszeit der jeweiligen Distribution circa alle sechs Wochen bzw. alle neun Monate neue Firefox-Pakete zur Verfügung. Das ist insofern bemerkenswert, weil es ansonsten in Linux-Distributionen unüblich ist, im Rahmen der Updates auch Versionswechsel durchzuführen.

**IPv6-Adressen** Sofern Ihr Computer über eine IPv6-Verbindung verfügt, können Sie mit Firefox natürlich auch IPv6-Webseiten besuchen. Dabei geben Sie wie üblich den Hostnamen der Seite an, also z. B. <https://heise.de>. Nur in Ausnahmefällen bzw. bei einer Fehlkonfiguration ist es erforderlich, die IPv6-Adresse direkt anzugeben. Damit Firefox nicht mit den Doppelpunkten durcheinanderkommt, geben Sie die Adresse in eckigen Klammern an. Die selten erforderliche Port-Nummer folgt außerhalb der eckigen Klammern. Sofern Sie einen IPv6-Internetzugang haben, führt der folgende Link zur Website <https://kofler.info>:

[https://\[2a01:4f8:171:2baf::4\]:80](https://[2a01:4f8:171:2baf::4]:80)

### Konfiguration und Interna

**Konfigurationsdateien** Firefox erzeugt beim ersten Start das Verzeichnis `.mozilla/firefox/<profil>.default`, wobei `<profil>` eine zufällige Zeichenkette ist. In diesem Verzeichnis speichert Firefox alle Einstellungen, Bookmarks, den Cache etc.



Um Lesezeichen, Passwörter, Add-ons, offene Tabs und andere Daten zwischen mehreren Firefox-Installationen zu synchronisieren, können Sie die in Firefox integrierte Sync-Funktion nutzen. Diese Funktion ist ein praktisches Hilfsmittel, um einen regelmäßigen Wechsel zwischen Windows (am Arbeitsplatz) und Linux (zu Hause) so komfortabel wie möglich zu machen.

Firefox Sync

Falls Ihr Rechner an das Internet bzw. an das lokale Netzwerk angeschlossen ist, aber dennoch kein Webzugang möglich ist, verwendet Ihr lokales Netzwerk wahrscheinlich einen Proxy-Server. Das ist ein Rechner, der zwischen Ihrem PC und dem Internet steht. Er dient als Zwischenspeicher und beschleunigt den Zugriff auf häufig benötigte Seiten. Der Proxy kann aber auch dazu dienen, bestimmte Webseiten zu blockieren oder alle Webzugriffe zu protokollieren.

Proxy

Damit Firefox den Proxy nutzt, öffnen Sie den Dialog EINSTELLUNGEN • ERWEITERT • NETZWERK • EINSTELLUNGEN und geben die erforderliche(n) Proxy-Adresse(n) an. Im Regelfall reicht es aus, die Felder für den HTTP- und FTP-Proxy auszufüllen. Die richtige Port-Nummer lautet zumeist 8080. Fragen Sie Ihren Systemadministrator, wenn Sie die Proxy-Adresse nicht kennen.

Firefox verwaltet einen lokalen Zwischenspeicher, in dem zuletzt besuchte Webseiten, Bilder etc. gespeichert werden. Wenn dieselbe Seite später ein zweites Mal betrachtet wird und sich seither nicht geändert hat, kann sie aus dem Cache geladen werden, was natürlich schneller ist. Standardmäßig werden bis zu 300 MiB für den Cache reserviert. Im Dialog EINSTELLUNGEN • ERWEITERT • NETZWERK stellen Sie die Cache-Größe ein bzw. löschen den Cache (siehe [Abbildung 7.1](#)). Die Adresse `about:cache` führt zu einer Liste aller Dateien, die momentan zwischengespeichert sind.

Lokaler Cache

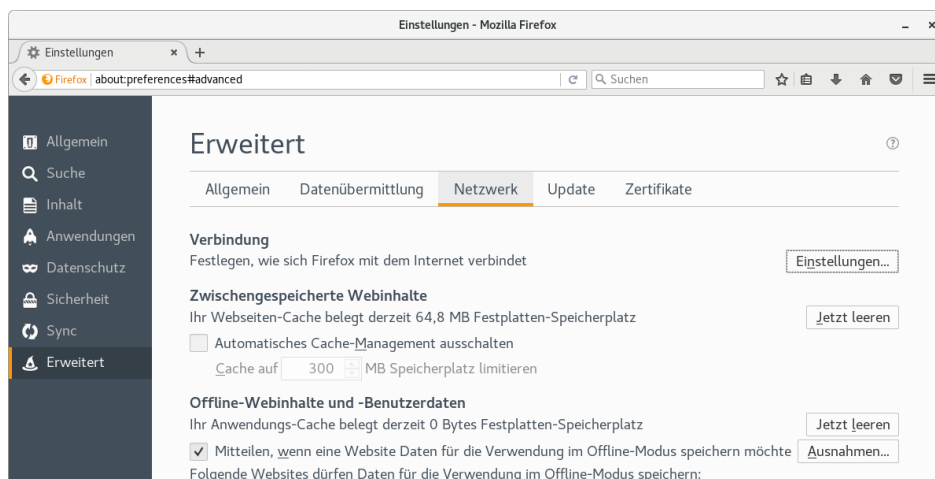

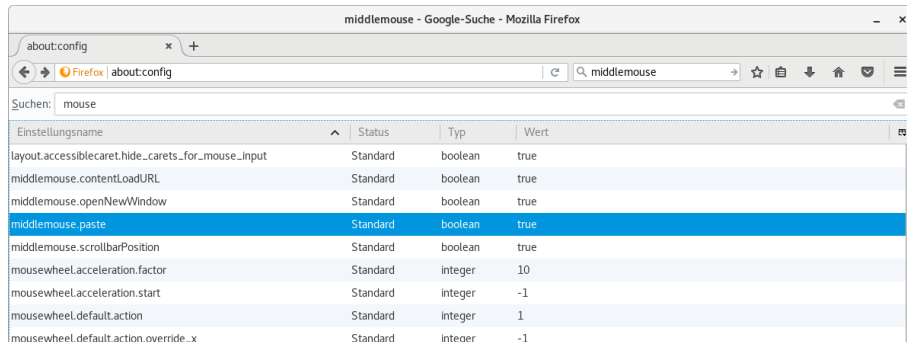


Abbildung 7.1 Firefox-Cache-Einstellungen

**about:config** Die wichtigsten Konfigurationsparameter ändern Sie direkt in den Einstellungsdialogen. Daneben gibt es unzählige weitere Optionen, die seltener benötigt werden. Eine alphabetische Liste dieser Optionen sowie deren aktuelle Einstellungen erhalten Sie, wenn Sie als Adresse *about:config* eingeben und dann  drücken (siehe [Abbildung 7.2](#)). Im Textfeld **SUCHEN** können Sie die Optionsliste auf alle Einträge reduzieren, die den angegebenen Suchtext enthalten. Um eine Option zu verändern, führen Sie einen Doppelklick aus.



| Einstellungsname                                   | Status   | Typ     | Wert |
|--|----------|---------|------|
| layout.accessiblecaret.hide_carets_for_mouse_input | Standard | boolean | true |
| middlemouse.contentLoadURL                         | Standard | boolean | true |
| middlemouse.openNewWindow                          | Standard | boolean | true |
| middlemouse.paste                                  | Standard | boolean | true |
| middlemouse.scrollBarPosition                      | Standard | boolean | true |
| mousewheel.acceleration.factor                     | Standard | integer | 10   |
| mousewheel.acceleration.start                      | Standard | integer | -1   |
| mousewheel.default.action                          | Standard | integer | 1    |
| mousewheel.default.action.override_x               | Standard | integer | -1   |

**Abbildung 7.2** Firefox-Konfiguration

Die folgende Aufzählung nennt einige ausgewählte Optionen:

- ▶ `layout.css.devPixelsPerPx`: Hier können Sie einen Skalierungsfaktor einstellen, damit Webseiten auf hochauflösenden Bildschirmen (High-DPI- bzw. Retina-Displays) größer dargestellt werden.
- ▶ `mousewheel.default.delta_multiplier_y`: Wenn Sie die Scroll-Richtung umdrehen möchten (wie auf Smartphones bzw. unter macOS), stellen Sie hier anstelle des Standardwerts 100 den Wert `-100` ein.
- ▶ `middlemouse.contentLoadURL`: Wenn dieser Parameter auf `true` gesetzt ist, können Sie Webadressen zuerst mit der Maus markieren und dann durch einen Klick auf die mittlere Maustaste besuchen. Bei einigen Distributionen, darunter älteren Ubuntu-Versionen, ist diese Funktion deaktiviert.

**MIME** Die Abkürzung MIME steht für *Multipurpose Internet Mail Extensions*. MIME ist dafür verantwortlich, dass der Webbrowser weiß, welches Programm er starten soll, wenn Sie einen Link auf eine MP3- oder PDF-Datei anklicken. Firefox berücksichtigt die allgemeinen Linux-MIME-Einstellungen (siehe [Abschnitt 11.3](#), »Dateitypen (MIME)«) sowie die MIME-Informationen aller installierten Browser-Plugins.

Einen Überblick über alle Firefox-spezifischen MIME-Einstellungen gibt **EINSTELLUNGEN • ANWENDUNGEN**. Dort können Sie vorhandene Einstellungen ändern und löschen, aber leider keine neuen Einträge hinzufügen. Ihre Einstellungen speichert Firefox in der XML-Datei `.mozilla/firefox/<profil>/mimeTypes.rdf`.

## Firefox-Erweiterungen (XPI-Dateien)

Firefox kann sehr universell durch XPI-Dateien erweitert werden. XPI steht für *Cross Platform Installation*. XPI-Dateien enthalten Firefox-Erweiterungen, wobei die Installationsdateien in einem Archiv samt JavaScript-Installationscode verpackt sind. Die Bandbreite der verfügbaren Erweiterungen reicht von Werbeblockern über Erweiterungen der Benutzeroberfläche, Download-Hilfen bis hin zu Werkzeugen für Webentwickler.

Im Dialog **EINSTELLUNGEN • ADD-ONS** können Sie Erweiterungen suchen, installieren und bei Bedarf auch wieder deaktivieren (siehe [Abbildung 7.3](#)). Viele Erweiterungen werden erst nach einem Neustart von Firefox wirksam.

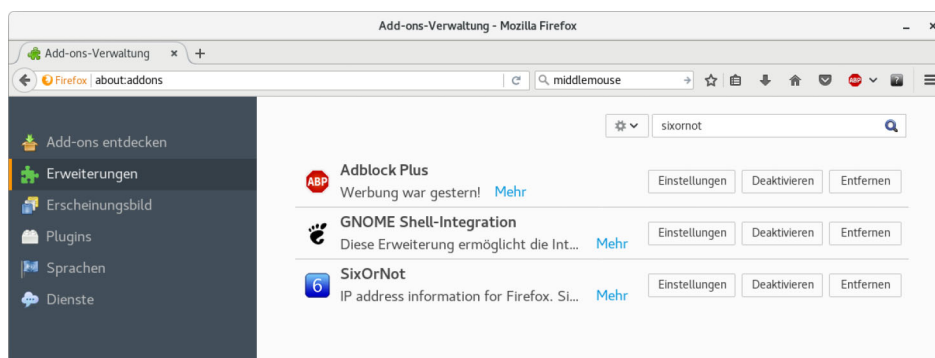


Abbildung 7.3 Firefox-Add-ons-Verwaltung

### Sicherheitsrisiko XPI

Auch ein Klick auf einen XPI-Link im Internet initiiert eine Add-on-Installation. In solchen Fällen warnt Firefox davor, dass XPI-Dateien auch böartigen Code enthalten können. Nehmen Sie diese Warnung ernst. Installieren Sie keine Erweiterungen, von deren Notwendigkeit und Sicherheit Sie nicht überzeugt sind!

## Browser-Plugins

Zwei Jahrzehnte lang ermöglichte die 1995 von Netscape entwickelte Plugin-Architektur (NPAPI) in vielen Webbrowsern das Abspielen von Audio-Dateien, Videos, Flash-Animationen bzw. das Ausführen von Java-Applets. Und beinahe ebenso lange ist diese Architektur für eine schier endlose Reihe von Abstürzen, Sicherheitsproblemen und 32/64-Bit-Inkompatibilitäten verantwortlich. Deswegen wurde diese Technologie schrittweise aus vielen Webbrowsern entfernt. Die so verlorene Funktionalität wird überwiegend durch modernere Webtechnologien inklusive HTML5,

**Installation**

JavaScript sowie durch in den Webbrowser eingebaute Zusatzfunktionen zur Verfügung gestellt.

Die Firefox-Entwickler waren in dieser Hinsicht etwas zögerlicher. Seit Firefox 52 werden aber sämtliche Plugins mit Ausnahme von Adobe Flash automatisch deaktiviert. Flash wird nur deswegen weiter unterstützt, weil es noch immer Millionen veralteter Seiten gibt, die Flash voraussetzen. (Zuletzt bin ich über Flash gestolpert, als mein Sohn seine Online-Englisch-Hausaufgabe machen sollte.)

<https://support.mozilla.org/de/kb/npapi-plugins>

Einen Überblick über alle momentan in Firefox verfügbaren Plugins samt der zugeordneten Dateiformate erhalten Sie, wenn Sie als Adresse `about:plugins` eingeben und `↵` drücken. Auch das Dialogblatt **EINSTELLUNGEN • ADD-ONS • PLUGINS** liefert eine Liste aller Plugins. Dort können Sie Plugins explizit aktivieren bzw. deaktivieren. Abbildung 7.4 zeigt die Plugins, die unter Ubuntu nach der Installation des Zusatzpakets `ubuntu-restricted-extras` zur Verfügung stehen.

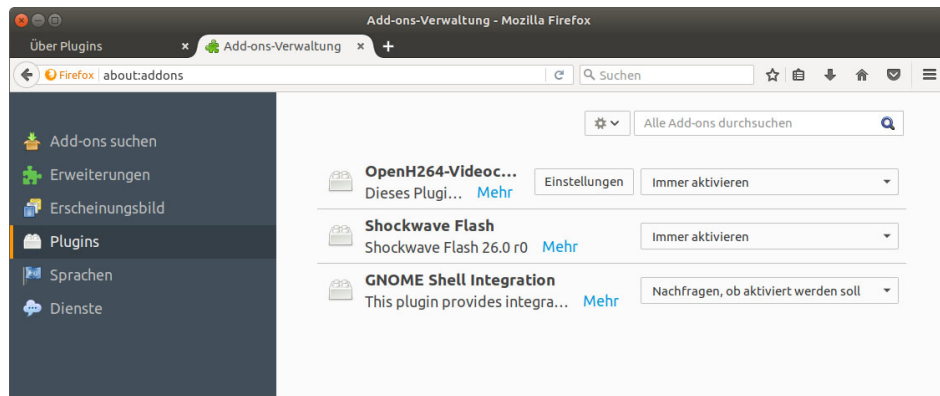


Abbildung 7.4 Plugins

## Flash-Plugin

Der Adobe Flash Player (ehemals Macromedia Flash) ermöglicht das Abspielen von Flash-Animationen im Webbrowser. Glücklicherweise verdrängt HTML5 Flash zunehmend. Dennoch gibt es noch immer viele Webseiten, die Sie nur mit Flash benutzen können.

Standardmäßig ist Flash unter Linux nicht installiert. Wenn Sie Flash brauchen, bestehen drei Möglichkeiten:

- ▶ Sie installieren das Flash-Plugin im alten, Firefox-kompatiblen NPAPI-Format. Adobe hat eigentlich 2012 angekündigt, das NPAPI-Plugin von Flash nicht mehr weiterzuentwickeln und mit Version 11.2 einzufrieren. 2016 wurde diese Entscheidung aber rückgängig gemacht. Seither stellt Adobe aktuelle Flash-Versionen wieder im NPAPI-Format zur Verfügung.
- ▶ Die einfachste Lösung besteht darin, Google Chrome zu installieren. In diesem Webbrowser ist die jeweils aktuellste Version von Flash enthalten. Sie verwendet das *Pepper Application Interface* (PPAPI), das von Adobe und Google entwickelt wurde.
- ▶ Sie installieren Chromium. Dort fehlt das integrierte Plugin zwar, aber Sie können es manuell selbst als PPAPI-Plugin installieren. Allerdings müssen Sie sich jetzt auch selbst darum kümmern, dieses Plugin stets auf dem neuesten Stand zu halten.

Die Lizenzbedingungen von Adobe machen es den meisten Distributionen unmöglich, Pakete für Flash selbst zur Verfügung zu stellen. In der Vergangenheit stellten einige Distributionen Installations-Skripts in Form von Paketen zur Verfügung. Bei der Installation wurde das Skript ausgeführt. Es lud das Flash-Plugin (die NPAPI-Variante) herunter und installierte es. Mit der schwindenden Bedeutung von Flash verschwanden auch diese Skripts. Einzig in Ubuntu gibt es noch das Paket `flashplugin-installer`. Es wird automatisch mit den `ubuntu-restricted-extras` installiert.

Installation des  
NPAPI-Plugins

Wenn Sie Flash wirklich brauchen, stellt auch die manuelle Installation von Flash keine große Hürde dar. Auf der folgenden Seite finden Sie das Flash-Plugin in Form von Debian- und RPM-Paketen, als Yum-Paketquelle sowie als TAR-Archiv. Falls Sie sich für die Yum-Paketquelle entscheiden, müssen Sie die eigentliche Plugin-Installation nach dem Einrichten der Paketquelle manuell starten (`yum install flash-plugin`).

<https://get.adobe.com/de/flashplayer>

Das TAR-Archiv enthält lediglich die Plugin-Datei `libflashplayer.so`. Diese Datei kopieren Sie in das Verzeichnis `.mozilla/plugins` oder `/usr/lib64/mozilla/plugins`. Anschließend starten Sie Ihren Webbrowser neu – fertig!

Nach einem Neustart von Firefox besuchen Sie die folgende Seite, um die Installation zu testen:

Flash testen

<https://www.adobe.com/software/flash/about>

Die Testseite zeigt eine kurze Animation an und gibt Auskunft darüber, welche Flash-Version gerade aktuell ist und welche auf Ihrem Rechner installiert ist.

## 7.2 Google Chrome

Unter Linux ist Firefox zwar noch der dominierende Webbrowser, weil er von fast allen Distributionen standardmäßig installiert wird. Unter Windows und macOS hat sich hingegen Google Chrome zum populärsten Programm entwickelt, und auch in der Linux-Welt haben Chrome und seine Open-Source-Variante Chromium große Verbreitung gefunden.

Für Chrome sprechen die ausgezeichnete Wartung, die schnellstmögliche Behebung von Sicherheitsproblemen und natürlich die im vorigen Abschnitt schon erwähnte Integration des Flash-Plugins. Gegen Google Chrome sprechen eigentlich nur Datenschutzbedenken.

### Google Chrome versus Chromium

Google stellt Chrome zwar kostenlos zur Verfügung, die Binärpakete von *google.com* stehen aber nicht unter einer Open-Source-Lizenz zur Verfügung. Deswegen ist die feste Integration von Chrome in eine Linux-Distribution schwer möglich.

Wenn Sie auf reinen Open-Source-Code Wert legen, müssen Sie statt Google Chrome dessen Open-Source-Basis Chromium installieren. Chromium steht bei vielen Distributionen als Paket zur Verfügung und kann mühelos installiert werden.

Es gibt nur wenige Unterschiede zwischen Google Chrome und Chromium: Bei Chromium fehlen das Google-Logo und das Google-Update-System. Stattdessen beziehen Sie Chromium-Updates über die Paketverwaltung Ihrer Distribution. Damit sind Sie darauf angewiesen, dass Ihre Distribution das Chromium-Paket gut wartet. In der Vergangenheit hat das leider nicht bei allen Distributionen gut geklappt! Auch auf die Integration des Flash-Plugins müssen Sie in Chromium verzichten.

### Installation

Sie finden RPM- und DEB-Installationspakete für Debian, Fedora, SUSE und Ubuntu in 32- und 64-Bit-Versionen auf der folgenden Seite zum Download:

<https://www.google.com/chrome>

Bei den meisten Distributionen wird nach dem Download automatisch ein geeignetes Paketinstallationsprogramm gestartet. Während der Installation wird automatisch eine eigene Paketquelle eingerichtet:

```
/etc/apt/sources.list.d/google-chrome.list    (Debian, Ubuntu)
/etc/yum.repos.d/google-chrome.repo          (CentOS, Fedora, RHEL)
/etc/zypp/repos.d/google-chrome.repo        (SUSE)
```

Die Paketquelle stellt sicher, dass Sie in Zukunft über das Update-System neue Google-Chrome-Versionen erhalten.

### Anmeldung

Beim ersten Start bietet Google Chrome Ihnen an, sich bei Ihrem Google-Konto anzumelden. Dieser Schritt ist freiwillig; wenn Sie aber ohnedies ein Google-Konto

bzw. eine Gmail-Adresse haben, bietet die Verbindung des Webbrowsers zum Google-Konto eine Menge Vorteile: Ihre Lesezeichen, Online-Passwörter, offenen Tabs, Google Apps etc. können nun über alle Ihre Geräte bzw. Webbrowser-Instanzen synchronisiert werden. (Die Verbindung zu einem Google-Konto ist auch in Chromium möglich.)

Im Detail steuern Sie mit **EINSTELLUNGEN • ERWEITERTE SYNCHRONISIERUNGSEINSTELLUNGEN**, welche Daten abgeglichen werden sollen und ob Ihre Daten mit einem eigenen Passwort verschlüsselt werden sollen. Um der NSA und anderen Geheimdiensten den Zugriff auf Ihre persönlichen Daten nicht allzu leicht zu machen, ist Letzteres zu empfehlen – etwa nach dem Motto: »Wer meine Daten lesen will, muss sich zumindest anstrengen!«

Google Chrome kann wie Firefox um zusätzliche Funktionen erweitert werden. Darüber hinaus kann Chrome dazu verwendet werden, sogenannte Apps auszuführen, also gewissermaßen eigenständige Programme, die im Browser laufen. **EINSTELLUNGEN • WEITERE TOOLS • ERWEITERUNGEN** öffnet die Seite `chrome://extensions` (siehe [Abbildung 7.5](#)). Sie listet alle installierten Erweiterungen auf und gibt Ihnen die Möglichkeit, Erweiterungen zu deaktivieren oder zu entfernen. Erweiterungen und Apps (die meisten sind kostenlos) finden Sie auch im Chrome Web Store:

Erweiterungen

<https://chrome.google.com/webstore>

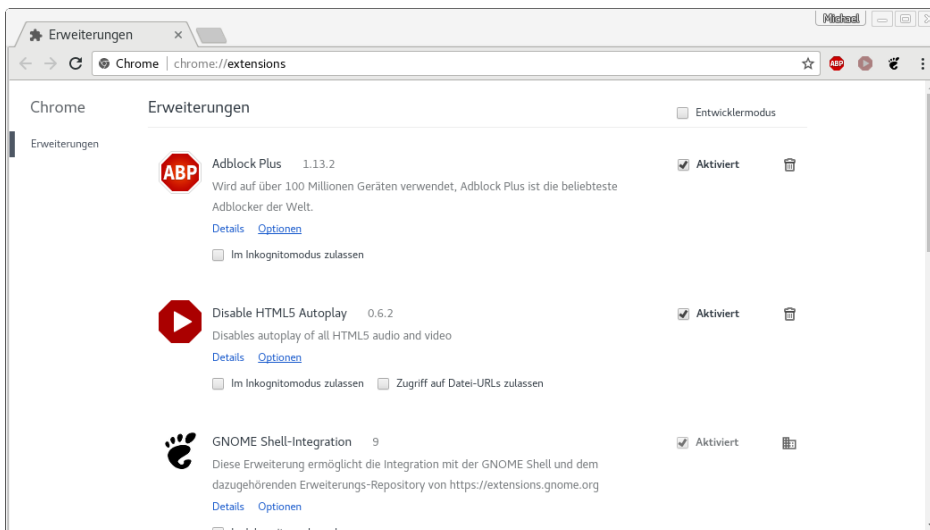


Abbildung 7.5 Erweiterungen in Google Chrome

Abbildung 7.5 zeigt außerdem, dass Chrome den Fensterrahmen selbst zeichnet. Deswegen fügt sich Chrome optisch nicht besonders gut in die meisten Desktop-Systeme ein.

**Flash** Google Chrome wirbt damit, dass ein sicherer PDF-Reader sowie das gerade aktuellste Flash-Plugin direkt in den Browser integriert sind. Die für Flash relevanten Einstellungen finden Sie auf der folgenden Seite:

*chrome://settings/content/flash*

Wenn Sie Chromium anstelle von Chrome verwenden, müssen Sie Flash als PPAPI-Plugin selbst installieren. Sie finden es auf der folgenden Webseite zum Download:

*https://www.adobe.com/software/flash/about*

Manche Ubuntu-Versionen stellen das PPAPI-Plugin auch im Paket `adobe-flashplugin` in der Partner-Paketquelle zur Verfügung. Diese Paketquelle müssen Sie gegebenenfalls vorher aktivieren.

### Alternative Webbrowser

Außer Firefox und Chrome gibt es diverse weitere Browser, die vor allem durch ihre perfekte Integration in das jeweilige Desktop-System brillieren. Ob das allein als Grund für ihren Einsatz ausreicht, müssen Sie selbst entscheiden. Tabelle 7.1 zählt kurz die wichtigsten Vertreter auf.

| Webbrowser         | Beschreibung                              |
|--------------------|---|
| Dillo              | minimalistischer Browser ohne JavaScript  |
| Konqueror, Rekonq  | KDE-Browser                               |
| Midori             | Xfce-Browser (auch in Elementary OS)      |
| Web alias Epiphany | Gnome-Standard-Browser (auch in Raspbian) |
| Lynx, ELinks, w3m  | Textmodus-Browser                         |

**Tabelle 7.1** Alternative Webbrowser

## 7.3 Thunderbird

Das E-Mail-Programm Thunderbird ist wie Firefox aus dem ehemaligen Mozilla-Projekt hervorgegangen. Obwohl es für die meisten Distributionen Thunderbird-Pakete gibt, ist das Programm oft nicht installiert. Der Grund: Gnome und KDE sehen Evolution bzw. KMail als Standard-E-Mail-Client vor.

Bei vielen Distributionen ist Thunderbird in mehrere Pakete aufgeteilt. Eines enthält die Grundfunktionen, und weitere Pakete enthalten die Menü- und Dialogtexte für verschiedene Sprachen. Vergessen Sie nicht, auch das deutsche Sprachpaket zu



installieren! In Gnome bzw. KDE sollten Sie anschließend Thunderbird in den System-einstellungen als Standard-E-Mail-Programm einrichten.

Die Mozilla Foundation hat die Weiterentwicklung von Thunderbird 2012 eingestellt. Sie stellt aber weiterhin die Infrastruktur für das Projekt zur Verfügung. Die Wartung und Weiterentwicklung erfolgt seither durch die Community.

Versionen

Was die Versionsnummern betrifft, ist Thunderbird mit Firefox gleichgeschaltet. Allerdings folgt Thunderbird dem ESR-Modell (*Extended Support Release*). Deswegen gibt es nur circa alle neun Monate eine grundlegend neue Version. Die Versionsnummer springt dabei jeweils um 7. So gibt es seit April 2017 die Version 52. Die nächste Major-Version wird voraussichtlich die Versionsnummer 59 haben und Anfang 2018 fertiggestellt werden.

### Account-Konfiguration

Beim ersten Start erscheint automatisch der Konten-Assistent, der Ihnen die Einrichtung eines neuen E-Mail-Kontos anbietet. Im Regelfall werden Sie diesen Schritt überspringen und sich stattdessen für die Option MEINE EXISTIERENDE E-MAIL-ADRESSE VERWENDEN entscheiden.

Im Folgenden müssen Sie zumindest drei Informationen angeben: Ihren Namen, Ihre E-Mail-Adresse und das Passwort für den E-Mail-Zugang. Thunderbird versucht die restlichen Parameter selbst zu erraten, was in vielen Fällen auch gelingt (siehe [Abbildung 7.6](#)). Falls Ihr E-Mail-Server sowohl POP als auch IMAP unterstützt, entscheidet sich Thunderbird für IMAP. Bei Bedarf können Sie mit MANUELL BEARBEITEN unzählige weitere Optionen einstellen (siehe auch [Abbildung 34.3](#) in [Abschnitt 34.6](#), »Client-Konfiguration«).

Abbildung 7.6 Account-Konfiguration in Thunderbird

Bei IMAP-Konten beginnt Thunderbird nach der Konfiguration, sämtliche E-Mails aus allen Verzeichnissen herunterzuladen. Die lokalen E-Mail-Kopien beschleunigen die Suchfunktionen, verursachen bei großen E-Mail-Konten aber eine Menge Download-Volumen und beanspruchen viel Platz auf der lokalen Festplatte oder SSD.

Das können Sie vermeiden, indem Sie in den Konteneinstellungen im Punkt SYNCHRONISATION & SPEICHERPLATZ die Synchronisation ganz abstellen, sie auf einzelne Postfächer limitieren oder pro Postfach nur ausgewählte Nachrichten synchronisieren, z. B. nur die aktuellsten E-Mails oder nur kleine E-Mails (siehe [Abbildung 7.7](#)).

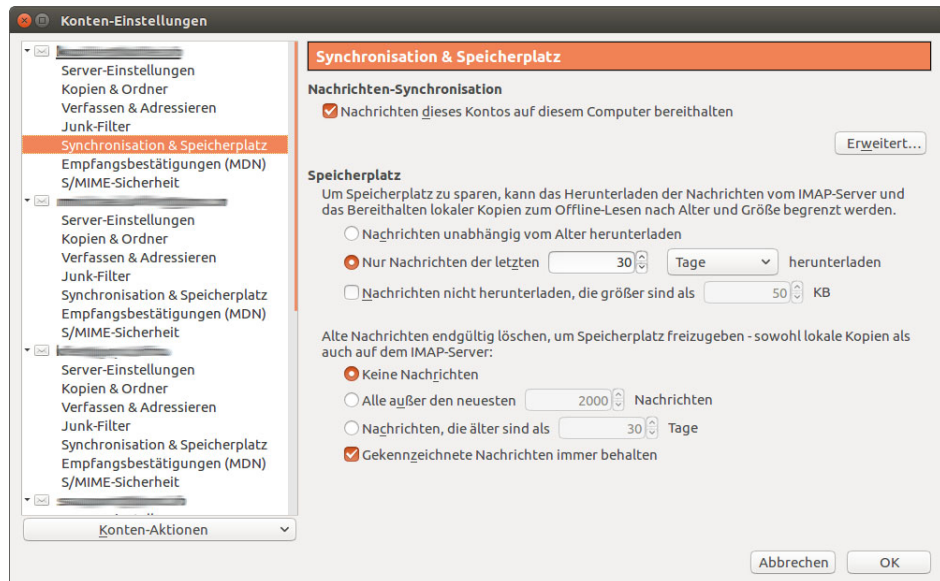


Abbildung 7.7 IMAP-Synchronisationseinstellungen

### Vorsicht

Im unteren Teil des Dialogs aus [Abbildung 7.7](#) befinden sich weitere Optionen, die auf den ersten Blick so ähnlich aussehen. Dabei geht es aber darum, alte Nachrichten nach einer gewissen Zeit automatisch zu löschen – und zwar unwiderruflich sowohl lokal als auch auf dem IMAP-Server! Passen Sie auf, dass Sie diese Option nicht versehentlich aktivieren. Mir ist das schon einmal passiert. Die Rekonstruktion meines Mail-Archivs aus Backups hat mich einen halben Arbeitstag gekostet.

## Grundfunktionen

Um Platz zu sparen, zeigen aktuelle Thunderbird-Versionen keine Menüleiste mehr an. Das Menü ist nun wie bei Firefox und Google Chrome hinter einem Button mit drei horizontalen Linien rechts oben im Fenster versteckt. Aus Platzgründen enthält dieses Menü aber nicht alle Einträge!

Menüleiste

Wenn Sie ein traditionelles Menü vorziehen, aktivieren Sie im Seitenmenü die Option **EINSTELLUNGEN • MENÜLEISTE**. Die folgenden Menükommandos beziehen sich auf die herkömmliche Menüleiste.

Neue E-Mails werden im Ordner **POSTEINGANG** gesammelt (siehe [Abbildung 7.8](#)). Unterhalb der Nachrichtenliste wird der Text der gerade ausgewählten E-Mail angezeigt. Mit einem Doppelklick innerhalb der Nachrichtenliste öffnen Sie ein eigenes E-Mail-Dialogblatt (Tab), das mehr Komfort und Platz zum Lesen umfangreicher E-Mails bietet. Wenn in HTML-Mails enthaltene Dateien und Bilder aus Sicherheitsgründen nicht geladen werden, schafft der Button **EXTERNE INHALTE ANZEIGEN** Abhilfe.

Posteingang

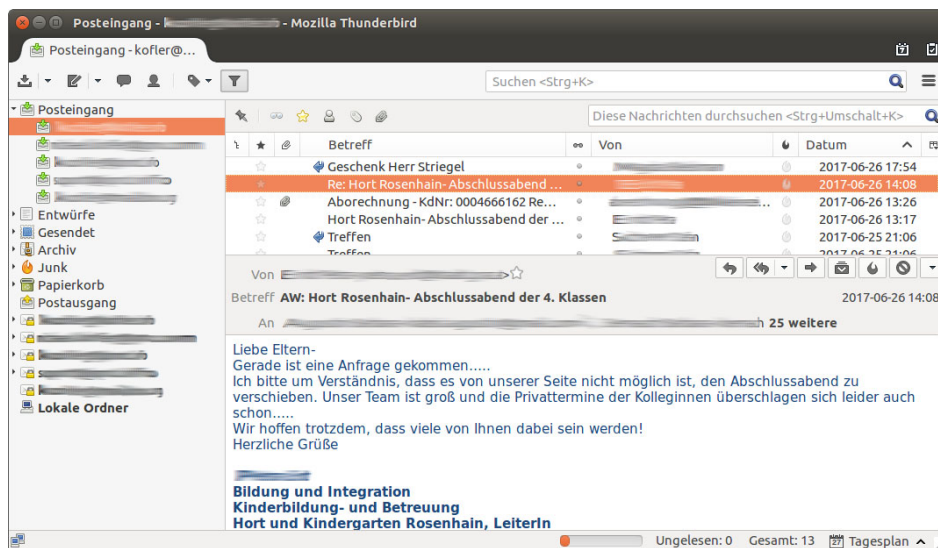



Abbildung 7.8 E-Mail-Verwaltung in Thunderbird

Im Menü **ANSICHT • ORDNER** können Sie zwischen verschiedenen Darstellungsformen wählen:


Ordneransichten

- ▶ Die Ansicht **ALLE ORDNER** ordnet alle Ordner dem jeweiligen Konto oder dem **LOKALEN ORDNER** zu. Der **LOKALE ORDNER** ist ein kontenunabhängiger Speicherort auf der Festplatte bzw. SSD. Der **LOKALE ORDNER** wird automatisch eingerichtet.

- ▶ Die Ansicht GRUPPIERTE ORDNER ist vor allem dann vorteilhaft, wenn Sie mehrere E-Mail-Konten eingerichtet haben. In diesem Fall werden Ordner aus verschiedenen Konten zusammengefasst. Damit sehen Sie alle neuen Nachrichten in einem zentralen Posteingangsortner, alle gelöschten Nachrichten in einem zentralen Papierkorb etc.
- ▶ UNGELESENE ORDNER zeigt alle Ordner, die ungelesene E-Mails enthalten.
- ▶ FAVORITEN-ORDNER zeigt alle Ordner, die zuvor in einer anderen Ordneransicht per Kontextmenü als FAVORITEN deklariert wurden.
- ▶ LETZTE ORDNER zeigt die zuletzt aktiven Ordner.

**HTML-Mails** Beim Verfassen neuer E-Mails verwendet Thunderbird automatisch das HTML-Format. Beachten Sie aber, dass sich nicht jeder Empfänger über diese Formatierung freut. Um eine einzelne E-Mail als reine Textnachricht zu verfassen, drücken Sie die -Taste, während Sie den Button VERFASSEN oder ANTWORTEN anklicken. Wenn Sie generell nur Text-Mails erstellen möchten, deaktivieren Sie im Konfigurationsdialog BEARBEITEN • KONTEN • VERFASSEN die Option NACHRICHTEN IM HTML-FORMAT VERFASSEN. Diese Einstellung ist für jedes Postfach erforderlich.

**E-Mails suchen und filtern** Thunderbird bietet drei Möglichkeiten, um nach E-Mails zu suchen:

- ▶ **Globale Suche:** Um eine Suche in *allen* E-Mails durchzuführen, geben Sie die Suchbegriffe im Textfeld rechts oben im Thunderbird-Fenster ein. Nach wenigen Sekunden zeigt Thunderbird in einem Dialogblatt alle Suchergebnisse an. Sie können nun die Suchergebnisse einschränken und nur die E-Mails aus einer bestimmten Zeit, von oder an bestimmte Personen, aus einem bestimmten Ordner etc. anzeigen.  
  
Die globale Suche setzt voraus, dass Thunderbird einen Index über alle E-Mails einrichtet. Bei großen E-Mail-Accounts ist das ein recht zeitaufwendiger Prozess, weswegen die Suchfunktion in EINSTELLUNGEN • ERWEITERT • ALLGEMEIN deaktiviert werden kann.
- ▶ **Suchfilter:** Hier geben Sie die Suchbegriffe im Eingabefeld LISTE FILTERN ein und drücken . Thunderbird reduziert nun die Liste der E-Mails im gerade aktuellen Verzeichnis auf alle E-Mails, die die Suchbegriffe im Absender-, Empfänger- oder Betreff-Feld enthalten. Optional können Sie die Suche auch auf den Nachrichteninhalte ausweiten.
- ▶ **Virtuelle Ordner:** Mit DATEI • NEU • VIRTUELLER ORDNER können Sie Suchkriterien formulieren. Diese Kriterien werden als virtueller Ordner gespeichert. Immer, wenn Sie diesen Ordner auswählen, werden darin alle E-Mails angezeigt, die den Suchkriterien entsprechen.

Mit **☞**+**Strg**+**B** öffnen Sie das Adressbuch. Dort können Sie mehrere Adresslisten verwalten. Standardmäßig sind zwei Listen vorgesehen: PERSÖNLICHES ADRESSBUCH und GESAMMELTE ADRESSEN. Wenn Sie möchten, speichert Thunderbird automatisch alle Adressen, an die Sie E-Mails senden, in einem Adressbuch. Die entsprechende Option finden Sie im Dialogblatt EINSTELLUNGEN • VERFASSEN • ADRESSIEREN.

Adressbuch

Um E-Mail-Adressen manuell zu speichern, reicht ein einfacher Mausklick auf den Stern, der neben jeder E-Mail-Adresse in der Nachrichtenansicht angezeigt wird. Bei bereits bekannten Adressen wird dieser Stern gefüllt angezeigt, bei unbekanntem Adressen als Kontur. Weitere Kontaktdaten können Sie anschließend im Adressbuchfenster hinzufügen. Mit EXTRAS • IMPORTIEREN können Sie zudem bereits vorhandene Adressbuchdateien in den verschiedensten Formaten einlesen.

Um das Thunderbird-Adressbuch mit dem Ihres Google-Kontos zu synchronisieren, müssen Sie auf ein Add-on zurückgreifen, z. B. auf *google contacts*. Bis zur ersten Synchronisation sind aber gleich zwei Thunderbird-Neustarts erforderlich: einmal nach der Add-on-Installation und ein zweites Mal nach dem Einrichten des Google-Mail-Kontos innerhalb des Google-Contacts-Add-ons (EXTRAS • ADD-ONS • ERWEITERUNGEN).

Leider ist das Thunderbird-Adressbuch für andere Linux-Programme unzugänglich und somit eine Insellösung. Auch die minimalistische, listenförmige Darstellung des Adressbuchs löst wenig Begeisterung aus.

In Thunderbird ist ein Spam-Filter integriert. Alle spam-verdächtigen E-Mails werden als Junk klassifiziert und in den gleichnamigen Ordner verschoben. Optionen zur Spam-Bekämpfung finden Sie im Konfigurationsdialog EINSTELLUNGEN • SICHERHEIT. Nach meinen Erfahrungen funktioniert der Spam-Filter aber nur mittelmäßig, auch dann, wenn Sie sich die Mühe machen, den Filter eine Weile zu trainieren.

Spam-Filter

Unabhängig vom Spam-Filter können Sie mit EXTRAS • FILTER weitere Filterregeln definieren. Auf diese Weise können Sie alle eintreffenden E-Mails, die ein bestimmtes Kriterium erfüllen, markieren oder automatisch in einen beliebigen Ordner verschieben. Das ist insbesondere zur automatischen Verarbeitung von E-Mails aus Mailing-Listen praktisch.

Filter

Thunderbird speichert lokal heruntergeladene E-Mails sowie alle Konfigurationseinstellungen im Verzeichnis `.thunderbird/xxxxxxx.default`, wobei xxxxxxx eine zufällig generierte Zeichenkette ist. Die E-Mail-Ordner liegen im mbox-Format vor und befinden sich im Unterverzeichnis Mail.

Interna

Wenn Sie von Windows auf Linux umsteigen, können Sie Ihr Mail-Verzeichnis der Thunderbird-Installation unter Windows einfach in das betreffende Linux-Verzeichnis kopieren. Wenn Sie unter Windows mit einem anderen E-Mail-Client gearbeitet haben, empfiehlt es sich, einen Zwischenschritt einzulegen: Die Windows-Version von Thunderbird bietet wesentlich bessere Import-Werkzeuge als die Linux-Version und hilft bei der Übertragung Ihrer E-Mail-Archive in ein Linux-kompatibles Format.

Beachten Sie, dass Thunderbird E-Mails normalerweise nicht physikalisch löscht. Die E-Mails werden nur als gelöscht markiert, verbleiben aber in der Datei. Deswegen beanspruchen Verzeichnisse für den Posteingang, für Spam-Mails sowie der Papierkorb oft unverhältnismäßig viel Platz. Abhilfe schafft das Kontextmenükommando **KOMPRIMIEREN**, das gelöschte E-Mails endgültig aus den mbox-Dateien entfernt.

### Erweiterungen und Zusatzfunktionen

**Add-ons** Ähnlich wie bei Firefox können Sie auch bei Thunderbird mit **EXTRAS • ADD-ONS** zusätzliche Funktionen in Form von Add-ons hinzufügen werden. Erweiterungen werden erst nach einem Neustart von Thunderbird wirksam. Nach jedem Thunderbird-Update müssen in der Regel auch die Erweiterungen aktualisiert werden, was mitunter Probleme verursacht (z. B. wenn die Erweiterung nicht ebenfalls in einer aktualisierten Version zur Verfügung steht).

Um eine manuell heruntergeladene XPI-Datei mit einem Thunderbird-Add-on zu installieren, führen Sie **EXTRAS • ADD-ONS** aus. Im Add-on-Dialog befindet sich am oberen Rand in der Mitte ein Werkzeug-Button, der in ein Menü führt. Dort haben die Thunderbird-Entwickler das Kommando **ADD-ON AUS DATEI INSTALLIEREN** versteckt.

**E-Mails signieren/  
verschlüsseln** In Thunderbird sind Kryptografiefunktionen für S/MIME bereits fix integriert. Sie finden alle erforderlichen Einstellungen im Dialogblatt **KONTEN-EINSTELLUNGEN • S/MIME-SICHERHEIT**. Der Button **ZERTIFIKATE** führt zu einem weiteren Dialog zur Verwaltung der X.509-Zertifikate, die bei S/MIME als Schlüssel dienen.

Damit Sie in Thunderbird PGP-signierte oder -verschlüsselte E-Mails lesen oder selbst verfassen können, müssen Sie das Add-on *Enigmail* installieren. Das Add-on setzt voraus, dass auf dem Rechner *gnupg* installiert ist. Das ist bei nahezu allen Distributionen der Fall. Alle Verschlüsselungsfunktionen sind über das **OPENPGP**-Menü im Hauptfenster und im **VERFASSEN**-Fenster zugänglich. Bei manchen Distributionen gibt es für Enigmail sogar ein eigenes Paket, das mit den Paketverwaltungswerkzeugen installiert werden kann.

**Termin-  
verwaltung** Seit Version 38 ist das Add-on *Lightning* in Thunderbird fix integriert. Bei älteren Versionen musste das Add-on extra installiert werden. Es hilft bei der Synchronisation von Terminen mit externen Servern in den Formaten CalDAV oder WCAP. Es kommt

grundsätzlich mit NextCloud-Kalendern zurecht, allerdings müssen Sie jeden Kalender einzeln einrichten (den für Ihre beruflichen Termine, den mit privaten Terminen etc.). Andere Programme agieren da intelligenter.

Neue Kalender richten Sie mit DATEI • NEU • KALENDER ein. Für den Google-Kalender müssen Sie dabei die folgende Adresse verwenden:

<https://www.google.com/calendar/dav/ID/events>

Dabei müssen Sie anstelle der *ID* für den Hauptkalender Ihre Google-E-Mail-Adresse angeben. Für die anderen Kalender ermitteln Sie die ID-Zeichenkette in der Google-Mail-Weboberfläche in den Einstellungen.

Das Add-on *Quick Folders* ermöglicht es Ihnen, besonders wichtige E-Mail-Ordner in einer eigenen Symbolleiste anzuordnen. Das Add-on ist ausgesprochen praktisch, wenn Sie mit Thunderbird viele Accounts mit vielen Verzeichnissen verwalten.

**Quick Folders**

## 7.4 Evolution, KMail und Geary

Die Entscheidung zwischen Thunderbird, Evolution oder KMail fällt schwer. Alle drei Programme bieten zahllose Funktionen und richten sich an fortgeschrittene Anwender. Persönlich bin ich ein Thunderbird-Fan: Thunderbird ist aus meiner Sicht der ausgereifteste E-Mail-Client für gehobene Ansprüche. Für die in diesem Abschnitt vorgestellten Programme Evolution und KMail sprechen hingegen die bessere Integration in Gnome und KDE, besonders bei der Kontakt- und Terminverwaltung.

Für Einsteiger sind weder Thunderbird noch Evolution noch KMail geeignet. Alle drei Programme verwirren Anwender mit viel zu vielen Optionen und Konfigurationsmöglichkeiten. Eine denkbare Alternative wäre das Programm Geary, das aber noch nicht wirklich stabil läuft. Ansonsten ist es vermutlich das Einfachste, auf einen Mail-Client ganz zu verzichten und stattdessen die Weboberfläche des jeweiligen Mail-Providers (z. B. GMX, Gmail oder Mailbox) zu verwenden.

### Evolution

Evolution ist das Standard-E-Mail-Programm des Gnome-Desktops. Evolution kann nicht nur zum Lesen und Schreiben von E-Mails verwendet werden, sondern enthält auch Funktionen zur Adress- und Terminverwaltung, zur Synchronisation dieser Daten mit dem Microsoft Exchange Server, zur Verschlüsselung von E-Mails mit PGP oder S/MIME etc.

**Account-Konfiguration**

Beim ersten Start von Evolution erscheint ein Assistent zur Einrichtung des E-Mail-Accounts. Der Assistent muss vollständig ausgeführt werden, bevor Evolution genutzt werden kann.

Die Konfiguration beginnt mit der Angabe Ihres Namens und Ihrer E-Mail-Adresse. Im nächsten Dialog folgen die Daten des Mail-Servers, von dem Sie Ihre E-Mail holen. Hier geben Sie den Server-Typ (z. B. POP oder IMAP), die Adresse des Servers sowie Ihren Login-Namen (Benutzernamen) an. Die unzähligen zur Auswahl stehenden Optionen belassen Sie auf den Vorgabeeinstellungen.

In einem weiteren Schritt konfigurieren Sie den Mail-Server (SMTP), an den Sie E-Mail senden. Sie müssen nicht nur den SMTP-Rechnernamen eingeben, sondern auch die Authentifizierungsoptionen einstellen. In den meisten Fällen lautet der richtige Legitimationstyp ANMELDEN. BENUTZERNAME bezeichnet nun den Login-Namen für SMTP. Nach dem Passwort werden Sie erst beim ersten Verbindungsaufbau gefragt, und zwar getrennt für das Empfangen und Senden von E-Mails.

Weitere Einstellungen können Sie später mit BEARBEITEN • EINSTELLUNGEN • E-MAIL-KONTEN vornehmen. Wenn Ihre E-Mails am Schluss immer denselben Text enthalten (z. B. *Mit freundlichen Grüßen ...*), können Sie hierfür im Dialogblatt IDENTITÄT eine Signatur angeben.

**Grundfunktionen**

Evolution lädt bei HTML-Mails aus Sicherheitsgründen keine Dateien (auch keine Bilder), auf die die HTML-Nachricht verweist. Sie können dieses Verhalten im Konfigurationsdialog BEARBEITEN • EINSTELLUNGEN • E-MAIL-EINSTELLUNGEN • HTML-NACHRICHTEN ändern.

Neue E-Mails verfassen Sie mit **[Strg]+[N]** und versenden sie mit **[Strg]+[↵]**. Standardmäßig erzeugt Evolution reine Text-Mails. Um eine HTML-Mail zu schreiben, wählen Sie im VERFASSEN-Fenster die Option HTML anstelle von EINFACHER TEXT. Anschließend bieten diverse Buttons und die Menüs EINFÜGEN und FORMAT eine Menge Formatierungsmöglichkeiten. Wenn Sie E-Mails grundsätzlich als HTML-Mails schreiben möchten, führen Sie BEARBEITEN • EINSTELLUNGEN aus und aktivieren im Dialogblatt EDITOREINSTELLUNGEN • ALLGEMEIN die Option NACHRICHTEN IN HTML FORMATIEREN.

Evolution enthält unterhalb der Symbolleiste ein Suchfeld, um rasch nach E-Mails zu suchen. Wenn Sie immer wieder dieselben Suchkriterien nutzen, lohnt es sich, einen sogenannten Suchordner (*Search Folder*) einzurichten. Darin werden alle E-Mails angezeigt, die bestimmten Suchkriterien entsprechen. Sie erstellen derartige Ordner mit BEARBEITEN • SUCHORDNER oder SUCHEN • SUCHORDNER AUS SUCHE ANLEGEN.



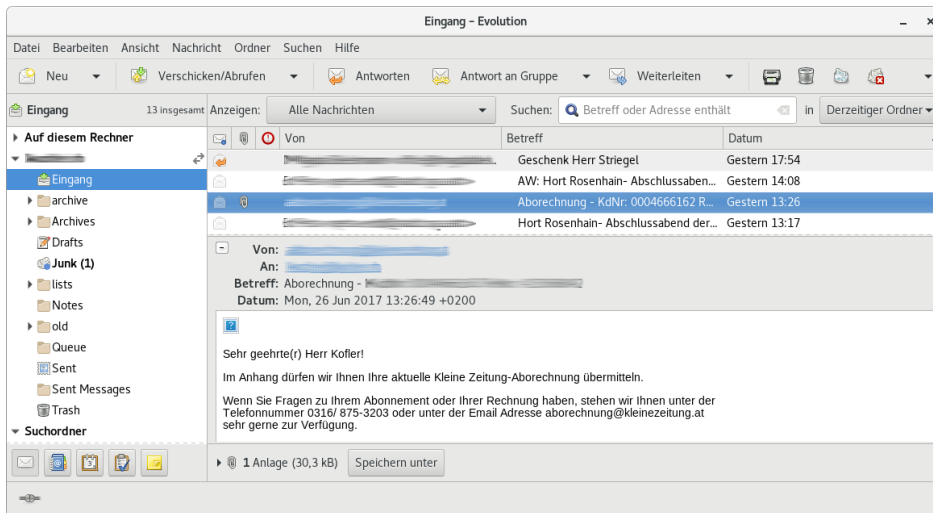


Abbildung 7.9 E-Mail-Verwaltung in Evolution

Evolution kann eintreffende E-Mails mit Filterregeln automatisch in bestimmte Verzeichnisse verschieben oder auch gleich löschen. Das ist praktisch, wenn Sie sehr viele E-Mails erhalten und diese anhand von Mustern eindeutig zuzuordnen sind, z. B. anhand bestimmter Wörter in der Betreffzeile. Das ist typischerweise dann der Fall, wenn Sie in mehreren Mailing-Listen eingetragen sind.

Filter

Der einfachste Weg zur Definition einer neuen Filterregel besteht darin, die Nachricht zu markieren und dann **NACHRICHT • ERSTELLEN • FILTERREGEL ÜBER MAILINGLISTE** auszuführen. Wenn Evolution die Filterregel nicht selbst richtig erkennt, können Sie sie ändern bzw. weitere Kriterien hinzufügen.

Das Evolution-Adressbuch ist eine vollständige Kontaktverwaltung, in der Sie neben Namen und E-Mail-Adressen viele weitere Daten speichern können. In das Adressbuch gelangen Sie mit **ANSICHT • FENSTER • KONTAKTE** oder einfach mit **[Strg]+[2]**. Mit **DATEI • IMPORTIEREN • EINZELNE DATEI IMPORTIEREN** können Sie Adressbuchdateien im Format LDIF (*Lightweight Directory Interchange Format*) importieren.

Adress- und Kontaktverwaltung

Mit **DATEI • NEU • ADRESSBUCH** können Sie neue Adressbücher einrichten, wobei als Datenquellen auch ein LDAP- oder WebDAV/CardDAV-Server sowie Google vorgesehen sind. Bei meinen Tests gelang auch der Adressabgleich mit ownCloud.

Das **KALENDER**-Modul hilft bei der Terminverwaltung. Vorhandene Termine können in unterschiedlichen Ansichten dargestellt werden: alle Termine eines Tags, einer Arbeitswoche, der gesamten Woche oder eines Monats. Viele Darstellungsdetails, z. B. die typische Arbeitszeit oder Schriftfarben, können Sie mit **BEARBEITEN • EINSTELLUNGEN • KALENDER** Ihren persönlichen Vorlieben anpassen.

Kalender und Terminverwaltung

Mit DATEI • NEU • KALENDER können Sie auch externe Kalender einrichten. Evolution unterstützt dabei die Protokolle WebCal und CalDAV sowie Google.

- Aufgabenliste** Evolution enthält auch ein Modul zur Verwaltung von Aufgaben (also eine Art To-do-Liste). Die Aufgaben können wahlweise in einer eigenen Ansicht oder als Teilbereich der Kalenderansicht dargestellt werden.
- Interna** Evolution speichert E-Mails in `.local/share/evolution`, Konfigurationseinstellungen in `.config/evolution` und diverse Cache-Dateien in `.cache/evolution`.
- Backups** Im DATEI-Menü können Sie ein vollständiges Backup aller Evolution-Daten anlegen. Eine derartige Sicherung ist auch dann praktisch, wenn Sie Ihr gesamtes Mail-Archiv inklusive aller Evolution-Einstellungen auf einen anderen Rechner übertragen möchten: Wenn Sie auf dem zweiten Rechner Evolution erstmalig starten, bietet das Programm Ihnen die Möglichkeit, die Daten aus einem Backup einzulesen.

### Kontakt bzw. KMail

Kontakt ist ein universelles Programm zur Verwaltung von E-Mails, Kontakten, Terminen, Aufgaben, Notizen sowie zur Anzeige von Nachrichten aus RSS-Feeds. Hinter den Kulissen ist Kontakt eigentlich nur eine Benutzeroberfläche, um verschiedene KDE-Programme einheitlich zu bedienen. Beachten Sie, dass sich das Menü von Kontakt verändert, je nachdem, welche Komponente gerade aktiv ist.

Für die E-Mail-Funktionen von Kontakt ist KMail verantwortlich. Wenn Sie die restlichen Funktionen von Kontakt nicht benötigen, können Sie KMail auch als eigenständiges Programm starten und ersparen sich so den durch Kontakt bedingten Overhead (siehe [Abbildung 7.10](#)). KMail ist stark technisch orientiert. Das Programm bietet zahllose Funktionen und lässt sich von Linux-Profis sehr effizient nutzen. Die Bedienung ist aber nicht immer intuitiv. Linux-Einsteigern ist das Programm daher nur eingeschränkt zu empfehlen.

- Account-Konfiguration** Beim ersten Start erscheint ein Kontenassistent, in dem Sie drei Informationen angeben müssen: Ihren Namen, Ihre E-Mail-Adresse und das dazugehörige Passwort. In vielen Fällen reichen diese Angaben zur Account-Konfiguration aus. Kontakt speichert die Passwörter in KWallet, einem KDE-Programm zur Verwaltung von Passwörtern und Schlüsseln. Wenn Sie KWallet bisher nicht verwendet haben, müssen Sie auch dieses Programm einrichten.

Nach der Erstkonfiguration laden Sie die E-Mails mit dem Button NACH E-MAILS SEHEN herunter. Im Dialogblatt EINSTELLUNGEN • KMAIL EINRICHTEN • ZUGÄNGE können Sie weitere Konten einrichten. Irritierend ist dabei, dass POP-, IMAP- und SMTP-Server jeweils getrennt konfiguriert werden müssen. Wenn Sie also ein weiteres

E-Mail-Konto hinzufügen möchten, müssen Sie *zwei* neue Zugänge einrichten: einen zum Empfang der Nachrichten (POP oder IMAP) und einen zweiten zum Versenden neuer E-Mails (SMTP). Dieser Doppelgleisigkeit gehen Sie aus dem Weg, wenn Sie zum Einrichten neuer Konten EXTRAS • KONTEN-ASSISTENT ausführen.

Bei meinen Tests mit zwei verschiedenen KMail-Versionen unter openSUSE Leap und unter Neon hat sich die Account-Konfiguration als extrem fehleranfällig erwiesen. Bei mehreren meiner E-Mail-Accounts, die sich mit anderen Mail-Clients problemlos nutzen ließen, scheiterte das Einrichten mit nichtssagenden Fehlermeldungen.

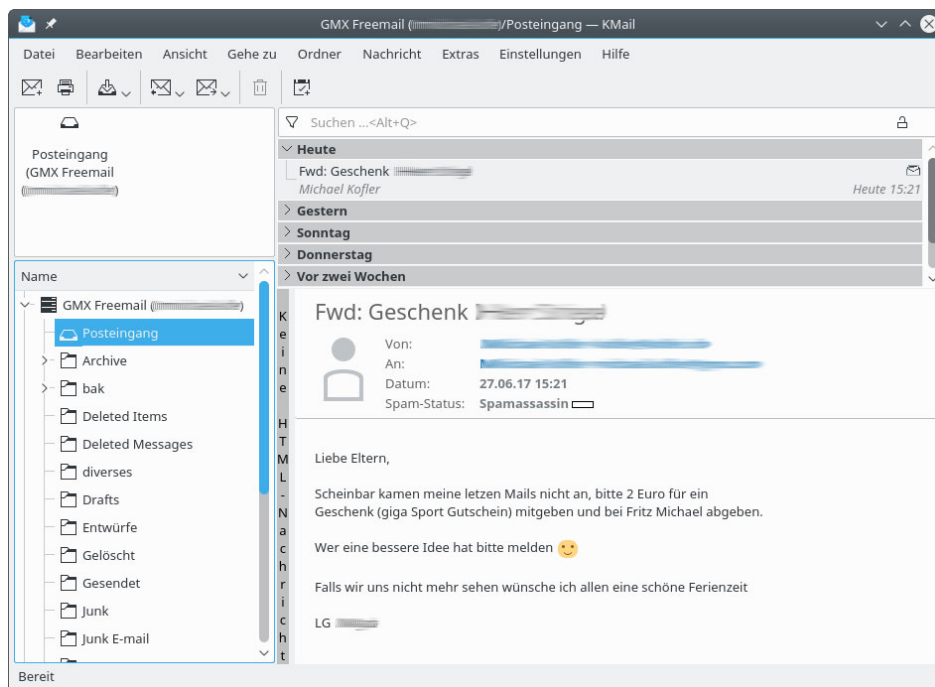


Abbildung 7.10 E-Mails verwalten mit KMail

Neue E-Mails sind standardmäßig reine Text-Mails. Wenn Sie eine HTML-Formatierung wünschen, führen Sie im Mail-Editor OPTIONEN • BEARBEITEN IM RICHT-TEXT aus. Die fertige E-Mail versenden Sie mit `[Strg]+[↵]`.

E-Mails verfassen

Im Menü EINSTELLUNGEN können Sie Filter definieren, um E-Mails anhand verschiedener Kriterien in verschiedenen Verzeichnissen abzulegen oder auf andere Weise zu bearbeiten.

Filter

## Geary

Die E-Mail-Programme, die ich bisher vorgestellt habe, sind durchweg schon ein bis zwei Jahrzehnte alt. Das merkt man an den ausgereiften Funktionen, aber leider auch an den verkrusteten Strukturen und dem oft trostlosen Erscheinungsbild. Das Entwicklerteam Yorba hat 2012 mit Geary einen Neuanfang gewagt. Das Ziel war bzw. ist es, einen einfach zu bedienenden und gleichzeitig optisch ansprechenden E-Mail-Client zu schaffen. In der getesteten Version 0.11 ist dies im Großen und Ganzen bereits gut gelungen (siehe [Abbildung 7.11](#)).

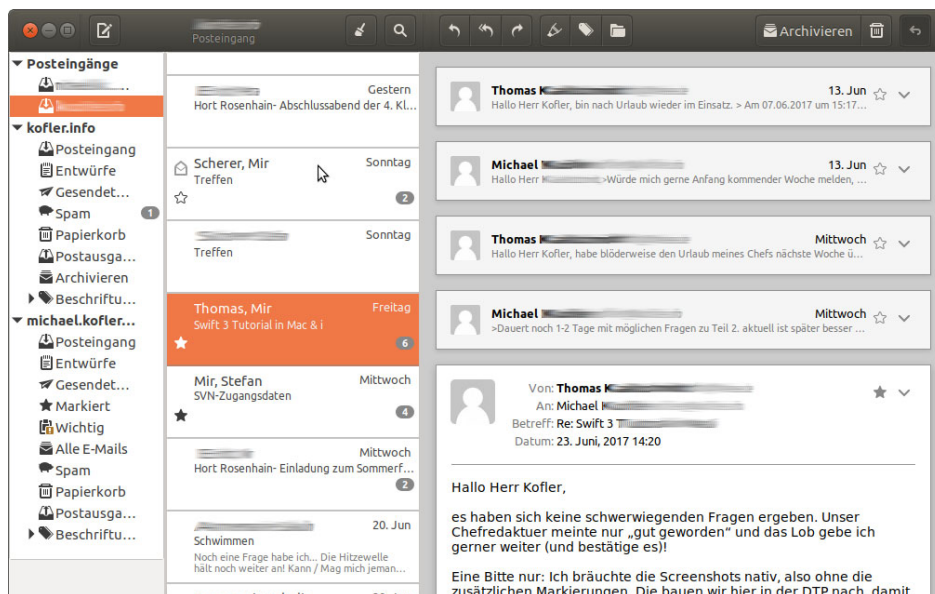


Abbildung 7.11 Geary

**Konfiguration** Beim ersten Start des Programms müssen Sie zumindest ein Konto einrichten. Unterstützt werden die Dienste GMail, Yahoo Mail und Outlook.com sowie standardkonforme IMAP-Server (z. B. Postfix). Beachten Sie, dass Sie bei vielen IMAP-Servern den SMTP-Port von 587 auf 25 umstellen müssen, wenn Sie das übliche Verschlüsselungsverfahren STARTTLS verwenden möchten. Die Account-Konfiguration und die Passwörter werden sofort überprüft und erst gespeichert, wenn eine Kommunikation mit dem Server möglich ist. Weitere E-Mail-Server können über das Gnome-App-Menü KONTEN eingerichtet werden.

Geary fasst automatisch alle zu einer Konversation gehörenden E-Mails zusammen. Das macht es leicht, das Frage-Antwort-Spiel auch später noch nachzuvollziehen.

### Gmail-Probleme

Das Einrichten eines Gmail-Kontos in Geary funktioniert aktuell nur, wenn Sie in der Gmail-Weboberfläche die Option WENIGER SICHERE APPS ZULASSEN aktivieren.

Die Konfigurationsmöglichkeiten des Programms beschränken sich – durchaus **Mängel** – auf ein absolutes Minimum. Eine Bedienung des Programms nur über die Tastatur ist nahezu unmöglich. Auf fortgeschrittene Funktionen müssen Sie ebenfalls verzichten. Aus meiner Sicht besonders schmerzhaft ist das Fehlen von Filterregeln zur automatischen Verarbeitung eintreffender E-Mails.

## 7.5 Dropbox

Dropbox ermöglicht es, das lokale Verzeichnis `Dropbox` mit einem Online-Speicher zu synchronisieren. Dropbox-ähnliche Dienste gibt es mittlerweile fast wie Sand am Meer: Apple, Google, Microsoft, alle buhlen um Ihre Daten und bieten zum Teil weit größere Gratiskontingente. Dropbox hat den großen Vorteil, dass die Integration unter Linux perfekt ist – vor allem, wenn Sie den Dateimanager Nautilus verwenden, der aus dem Gnome-Projekt stammt.

Einige Distributionen stellen die Dropbox-Erweiterung für den Dateimanager Nautilus in fertigen Paketen zur Verfügung, z. B. `nautilus-dropbox` in Ubuntu. Für alle anderen Distributionen finden Sie den Dropbox-Client auf der Dropbox-Website zum Download:

*<https://www.dropbox.com/install-linux>*

Nach der Installation führen Sie das Programm Dropbox oder das Kommando `dropbox start -i` aus und richten ein neues Dropbox-Konto ein bzw. melden sich bei Ihrem existierenden Konto an. Dabei wird automatisch das Verzeichnis `Dropbox` eingerichtet. Unter Fedora müssen Sie vor dem ersten Start von Dropbox das Paket `python2-pygments` installieren, indem Sie in einem Terminalfenster `sudo dnf install python2-pygments` ausführen.

Nach einem Neustart von Nautilus werden darin alle synchronisierten Dateien durch ein grünes OK-Häkchen gekennzeichnet (siehe [Abbildung 7.12](#)). Bei umfangreichen Änderungen im `Dropbox`-Verzeichnis dauert die Synchronisation eine Weile.

**Dropbox unter KDE und Gnome**

Bei KDE müssen Sie auf die Integration in den Dateimanager verzichten. Damit entfällt das optische Feedback, Dropbox ist aber natürlich dennoch verwendbar. Eine weitere Hürde ist der automatische Start des Dropbox-Clients, der leider oft nicht funktioniert.

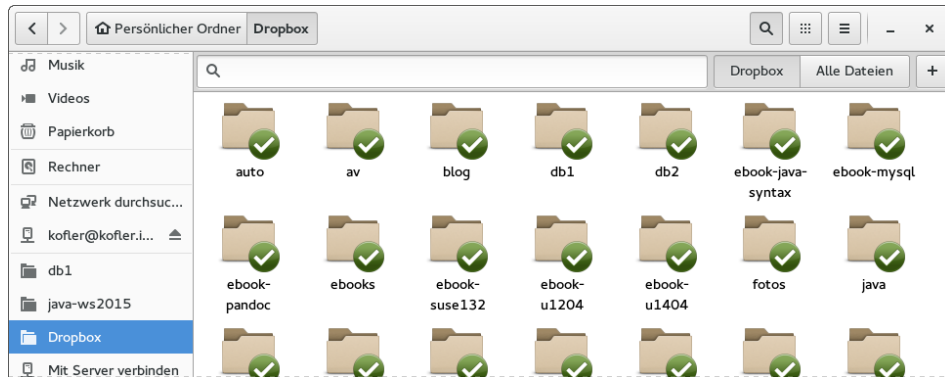


Abbildung 7.12 Dropbox-Integration in Nautilus

Um das Problem zu beheben, öffnen Sie in den Systemeinstellungen das Dialogblatt AUTOSTART des Moduls STARTEN UND BEENDEN und klicken Sie auf den Button PROGRAMM HINZUFÜGEN. Im Dialog PROGRAMM AUSWÄHLEN geben Sie den exakten Pfad des Programms `dropboxd` an, also beispielsweise `/home/kofler/.dropbox-dist/dropboxd`.

**Konfiguration** Unter Ubuntu (Unity) gibt ein Panel-Icon Auskunft über den Status der Synchronisation. Unter Gnome befindet sich ein entsprechendes Icon im Statusbereich, der sich bei aktuellen Versionen links unten auf dem Bildschirm befindet.

Über das Dropbox-Menü können Sie diverse Dropbox-Einstellungen verändern. Insbesondere können Sie im Dialogblatt KONTO mit dem Button SELEKTIVE SYNCHRONISATION einzelne Unterverzeichnisse innerhalb des Dropbox-Verzeichnisses von der Synchronisation ausschließen. Es gibt aber leider keine Möglichkeit, die Synchronisation für bestimmte Dateitypen zu deaktivieren.

### Wie sicher sind Ihre Daten bei Dropbox?

Ihre Dateien werden auf den Dropbox-Servern zwar verschlüsselt, der Schlüssel ist allerdings von Dropbox vorgegeben und kann nicht individuell gewählt werden. Dieses Verfahren ist nur mäßig sicher. Es ist zu befürchten, dass die NSA und andere Geheimdienste Zugang zu Ihren Daten haben. Persönliche bzw. unternehmenskritische Daten sollten daher nicht bzw. nur verschlüsselt im Dropbox-Verzeichnis gespeichert werden!

**Alternativen** Sofern Sie über einen eigenen Server verfügen, können Sie mit ownCloud oder Nextcloud (siehe [Kapitel 35](#)) unbegrenzt große Verzeichnisse ohne zusätzliche Kosten synchronisieren. Daneben gibt es unzählige weitere Anbieter zur Synchronisation von Dateien in der Cloud: Google Drive, Microsoft OneDrive etc. Freilich stehen nicht für alle Cloud-Dienste Linux-Clients zur Verfügung. Außerdem gibt es nur wenige Dienste, die derart einfach und komfortabel zu nutzen sind wie Dropbox.

## 7.6 FileZilla und BitTorrent

Anders als unter Windows gibt es für Linux nur wenige populäre FTP- und Download-Manager. Das hat drei Gründe: Erstens können Sie mit jedem Dateimanager FTP-Verzeichnisse genauso komfortabel wie lokale Verzeichnisse bearbeiten, zweitens gibt es für Firefox und Google Chrome tolle Download-Manager als Erweiterungen, und schließlich können Sie im Terminal unzählige Download-Kommandos einsetzen (z. B. `wget`, `curl` und `mirror`). Damit lassen sich Downloads perfekt automatisieren.

Der beliebteste Download-Client mit grafischer Benutzeroberfläche ist FileZilla. Dieses Programm unterstützt neben FTP auch die Protokolle SFTP und SSH, nicht aber HTTP. Die Bedienung ist etwas unübersichtlich und gewöhnungsbedürftig, davon abgesehen bietet das Programm aber alle Funktionen, die man sich von einem Download-Manager wünscht.

FileZilla

BitTorrent ist ein Protokoll zum effizienten Download großer Dateien, die oft von vielen Benutzern gleichzeitig gewünscht werden. Die Grundidee ist einfach: Der Download erfolgt nicht von einem zentralen Server, sondern von allen im Netz verfügbaren Rechnern, auf denen zumindest Teile der Datei zur Verfügung stehen (also sogenanntes *Peer-to-Peer Networking*). Umgekehrt bedeutet das: Wenn Sie via BitTorrent eine große Datei herunterladen, stellen Sie diese Datei während dieser Zeit (und idealerweise auch danach) auch allen anderen BitTorrent-Benutzern im Netz zur Verfügung.

BitTorrent

In der Linux-Praxis ist BitTorrent insofern interessant, als einige Distributionen DVD-Images als »Torrents« zur Verfügung stellen. Bei der Vorstellung einer neuen Version starten oft Tausende von Benutzern nahezu gleichzeitig den Download. Dank BitTorrent ist in solchen Fällen ein schnellerer Download möglich. Weitere Informationen zu den Grundlagen und Techniken des BitTorrent-Verfahrens sind im folgenden Wikipedia-Artikel gut zusammengefasst:

<https://de.wikipedia.org/wiki/BitTorrent>

BitTorrent-Downloads werden durch `.torrent`-Dateien bekannt gegeben. Dabei handelt es sich um relativ kleine Binärdateien, die unter anderem Prüfsummen für zahllose Teilstücke der Datei enthalten. Das ermöglicht es, den Download nicht sequenziell, sondern in zufälliger Reihenfolge und parallel von mehreren im Netz verfügbaren BitTorrent-Quellen durchzuführen.

BitTorrent-Clients sind Programme, die einerseits den Download durchführen und andererseits heruntergeladene Dateien anderen BitTorrent-Clients anbieten. Populäre Programme sind BitTorrent, KTorrent (KDE) sowie Transmission (Gnome), die alle eine ansprechende Oberfläche haben. Das KDE-Programm KTorrent zeigt an, welche Teile der Datei bereits heruntergeladen wurden. Wenn Sie BitTorrent-Downloads

KTorrent und Transmission

interaktiv in einer Konsole oder automatisiert per Script ausführen möchten, sollten Sie einen Blick auf die BitTorrent-Varianten `bittorrent-curses` und `bittorrent-console` werfen, die im `bittorrent`-Paket gleich mitgeliefert werden.

## 7.7 Shotwell

Shotwell ist bei den meisten Gnome-basierten Distributionen als Bildverwaltungsprogramm vorinstalliert. Shotwell bietet vergleichsweise wenige Funktionen. Dafür ist das Programm aber einfach zu bedienen und läuft schnell und stabil.

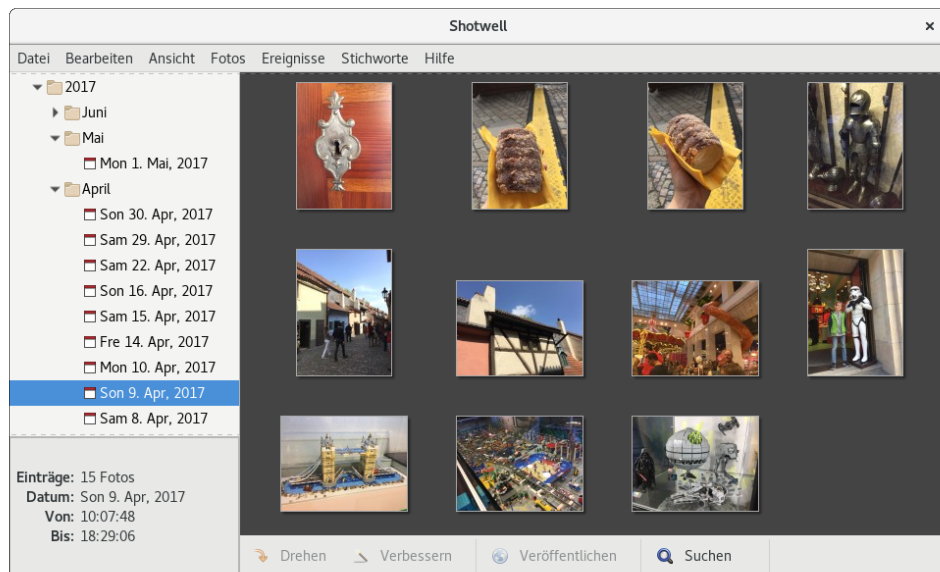


Abbildung 7.13 Bildverwaltung mit Shotwell

- Import** Die Importfunktion erscheint beim ersten Start und in der Zukunft immer dann, wenn Sie eine Kamera oder deren Speicherkarte mit dem Computer verbinden. Außerdem können Sie mit `DATEI • AUS ORDNER IMPORTIEREN` Bilder und Filme aus einem Verzeichnis einlesen. Dabei haben Sie die Wahl, ob die Bild- und Videodateien an ihrem bisherigen Ort bleiben sollen oder ob sie in ein von Shotwell verwaltetes Verzeichnis kopiert werden sollen. Fotos können auch per Drag & Drop aus dem Dateimanager in Shotwell importiert werden.
- Ereignisse** Die Bilder werden beim Import automatisch »Ereignissen« zugeordnet, wobei jeder Tag, an dem Fotos entstanden sind, als Ereignis gilt. Ereignisse können mit `[F2]` problemlos umbenannt werden. Um die Fotos mehrerer Tage zu einem Ereignis zu vereinen, markieren Sie die betreffenden Tage in der Monatsübersicht und führen dann das Kontextmenükommando `EREIGNISSE ZUSAMMENFÜHREN` aus. Leider ist dieser Vorgang bei vielen Bildern recht langsam. Um umgekehrt die Fotos eines Tages



mehreren Ereignissen zuzuordnen, markieren Sie mehrere Fotos und führen dann **[Strg]+[N]** aus.

Sie können ein einzelnes Foto per Kontextmenü **ZUM SCHLÜSSELFOTO FÜR DIESES EREIGNIS MACHEN**. Das Bild wird dann in der Ereignisübersicht angezeigt.

Mit **[Strg]+[T]** statten Sie Bilder mit sogenannten *Tags* aus, also mit Begriffen, nach denen Sie später suchen können. Für jedes Bild dürfen mehrere, nur durch Kommas getrennte Tags angegeben werden.

**Bilder organisieren**

Mit den Tasten **[1]** bis **[5]** bewerten Sie ein Bild mit ein bis fünf Sternen. **[0]** entfernt die Bewertung, **[9]** kennzeichnet das Bild als **ABGELEHNT**. Mit **ANZEIGEN • FOTOS FILTERN** können Sie anschließend nur solche Bilder anzeigen, die positiv bewertet wurden. Noch mehr Suchmöglichkeiten bietet die Suchleiste, die Sie mit **[F8]** ein- bzw. wieder ausblenden.

Shotwell bietet einige simple Bearbeitungsfunktionen an: Die Bilder können in 90-Grad-Schritten gedreht (**[Strg]+[R]**) und beschnitten werden. Außerdem kann der Rote-Augen-Effekt behoben und der Kontrast der Bilder verbessert werden.

**Bilder bearbeiten**

Sämtliche Bearbeitungsschritte werden nicht direkt an der Bilddatei durchgeführt, sondern in der Datenbank des Programms gespeichert und bei der Anzeige des Bilds angewendet. Mit dem Kontextmenükommando **ZURÜCK ZUM ORIGINAL** kann jedes veränderte Bild wiederhergestellt werden. Das sichert einerseits die Integrität der Originaldateien, erschwert aber andererseits einen späteren Wechsel auf ein anderes Programm.

Um ein Bild zu löschen, führen Sie **[Entf]** oder das Kontextmenükommando **IN DEN MÜLL VERSCHIEBEN** aus. Damit wird das Bild innerhalb der Bilddatenbank in einen OpenShot-eigenen **PAPIERKORB** gelegt. Erst wenn Sie den Papierkorb explizit leeren, werden die Bilddateien nach einer Rückfrage endgültig gelöscht.

Ausgewählte Bilder können mit Tags (Markierungen) versehen, in einer sehr einfachen Diaschau angezeigt, in ein Verzeichnis exportiert oder auf Facebook, Flickr oder Picasa veröffentlicht werden.

**Sonstige Funktionen**

Standardmäßig verteilt Shotwell die importierten Bilder über die Verzeichnisse **Bilder/jahr/monat/tag**. Wenn Sie ein anderes Basisverzeichnis oder eine andere Organisationsstruktur wünschen, finden Sie entsprechende Optionen im Dialog **BEARBEITEN • EINSTELLUNGEN**.

**Interna**

Außer den Bilddateien speichert Shotwell im verborgenen Verzeichnis **.local/share/shotwell/data** eine Bilddatenbank mit Zusatzinformationen zu allen Bildern. Darüber hinaus befinden im Verzeichnis **.cache/shotwell/thumbs** verkleinerte Vorschaubilder zu allen Fotos. Diese Vorschaubilder sind entscheidend für die hohe Darstellungsgeschwindigkeit von Shotwell.

## 7.8 digiKam

digiKam ist ein sehr vielseitiges KDE-Programm zum Fotoimport von Digitalkameras, zur Verwaltung der Bilder und zur Durchführung einfacher Bearbeitungsschritte. digiKam ist durch ein Plugin-System erweiterbar. Dank derartiger Plugins kann es direkt mit RAW-Dateien umgehen, Farbprofile verwalten, diverse Filter auf Bilder anwenden etc. Wie viele andere KDE-Programme glänzt digiKam durch eine unvergleichliche Funktionsvielfalt und richtet sich an Profis. Gleichzeitig ist aber die Benutzeroberfläche überladen und die Bedienung unübersichtlich.

Beim ersten Start des Programms durchlaufen Sie die Dialoge des Einrichtungssistenten. Dort müssen Sie ein Basisverzeichnis für Ihre Bilder konfigurieren. Bei allen weiteren Optionen können Sie jeweils die Vorgabeeinstellungen übernehmen. Bei Bedarf können Sie mit **EINSTELLUNGEN • DIGIKAM EINRICHTEN • ALBEN** sämtliche digiKam-Optionen in einem Dialog einstellen, der 19 Registerkarten mit mehreren Hundert Parametern umfasst.

**Fotoimport** Wenn Linux Ihre Digitalkamera als USB-Speichermedium betrachtet, starten Sie den Import der dort befindlichen Bilder mit **IMPORTIEREN • USB-SPEICHERGERÄTE**. Alle anderen Kameras müssen Sie vor dem ersten Import konfigurieren: In den meisten Fällen ist es ausreichend, im Dialog **IMPORTIEREN • KAMERAS • KAMERA MANUELL HINZUFÜGEN** den Button **AUTOMATISCHE ERKENNUNG** anzuklicken. Die Kamera wird von nun an im **KAMERA**-Menü aufgelistet.

Den Import starten Sie dann mit **IMPORTIEREN • KAMERA • KAMERANAME**. In jedem Fall erscheint nun ein Dialog mit einer Vorschau aller Bilder. Der Button **AUSGEWÄHLTE HERUNTERLADEN** führt in einen Dialog zur Auswahl des Zielverzeichnisses. Anschließend werden die markierten Bilder dorthin kopiert.

Um ein bereits vorhandenes Fotoverzeichnis zu importieren, führen Sie **IMPORTIEREN • ORDNER HINZUFÜGEN** aus. Dieses Kommando kann nur verwendet werden, wenn Sie vorher in der Albenansicht ein Album (z. B. `/home/name/Bilder`) auswählen. Die Bilddateien werden beim Import kopiert.

**Bildverwaltung** Jedes Verzeichnis innerhalb des Basisverzeichnisses bezeichnet digiKam als *Album*. Alternativ können Sie auch in der Datumsansicht oder in einer Zeitleiste nach den Bildern suchen (siehe [Abbildung 7.14](#)). Ein Mausklick auf das gerade aktuelle Bild vergrößert es, ein weiterer Klick führt zurück in die Albenansicht. Auf der rechten Fensterseite können Sie zusätzliche Bildeigenschaften, Kommentare und Stichwörter einblenden bzw. dort ändern. Diese zusätzlichen Daten, die bei der späteren Suche nach Bildern helfen, werden nicht direkt in den Bildern, sondern in der Datei `digi-kam4.db` im Basisverzeichnis der Bilder gespeichert.

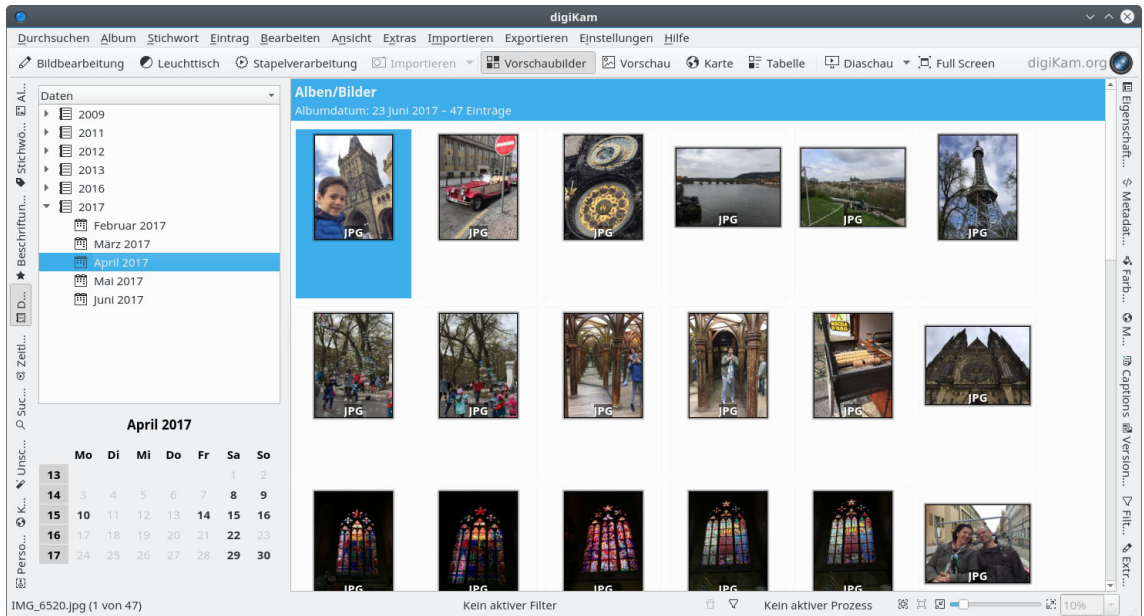


Abbildung 7.14 Bildverwaltung mit digiKam

Soweit die Kamera die Orientierung der Bilder in den EXIF-Daten vermerkt, dreht BILD • AUTOMATISCHES DREHEN/SPIEGELN alle Bilder im aktuellen Verzeichnis richtig. Zum manuellen Rotieren sind die Tastenkürzel **[Strg] + [↻] + [←]** bzw. **[→]** vorgesehen.

**Bilder richtig drehen**

In der Ordneransicht können Sie nur ganz elementare Bearbeitungsschritte durchführen, z. B. das Bild drehen. Weitergehende Operationen stehen zur Verfügung, wenn Sie **[F4]** drücken. digiKam zeigt das Bild dann in einem neuen Fenster an. Dort können Sie das Bild rahmen, beschriften, Farben und Helligkeit optimieren, weichzeichnen, schärfen, rote Augen korrigieren, die Größe ändern etc. Mit **ÄNDERUNGEN SPEICHERN** bzw. **ALS NEUE VERSION SPEICHERN** bestimmen Sie, ob die Änderungen in der Originaldatei oder in einer neuen Datei gespeichert werden.

**Bilder bearbeiten**

Das Menü **EXTRAS** führt zu diversen Zusatzfunktionen. Zahlreiche digiKam-Funktionen sind als Plugins realisiert (KIPI = *KDE Image Plugin Interface*). Wenn einzelne digiKam-Funktionen bei Ihnen fehlen, sollten Sie prüfen, ob die Plugins installiert und im Dialogblatt **EINSTELLUNGEN • DIGIKAM EINRICHTEN • KIPI-MODULE** auch aktiviert sind. Mit **EXTRAS • STAPELVERARBEITUNG** können Sie alle markierten Bilder gemeinsam konvertieren oder ändern.

**ANSICHT • DIASCHAU • ALLE** bzw. **AUSWAHL** präsentiert das aktuelle Album bzw. die gerade ausgewählten Bilder als Diaschau ohne besondere Effekte. Während die Präsentation aktiv ist, können Sie auch mit dem Mausrad vor- und zurückblättern. Das

**Bilder ansehen und exportieren**

Zeitintervall für den Bildwechsel sowie einige andere Optionen können Sie mit **EINSTELLUNGEN • DIGIKAM EINRICHTEN • DIASCHAU** angeben.

Wenn Sie die Diaschau mit Überblendeffekten und Musikuntermalung durchführen möchten, verwenden Sie **ANSICHT • DIASCHAU • ERWEITERTE DIASCHAU**. In einem Dialog können Sie diverse Einstellungen vornehmen, die allerdings immer für alle Bilder gelten: Sie können also nicht einem bestimmten Bild einen Effekt zuordnen oder je nach Bild unterschiedliche Zeitintervalle für den Bildwechsel einstellen. Beachten Sie, dass die Auswahl der Bildeffekte variiert, je nachdem, ob Sie die Option **OPENGL-ÜBERGÄNGE** aktivieren oder nicht.

Das Menü **EXPORTIEREN** enthält ein ganzes Dutzend Kommandos, um die zuvor ausgewählten Bilder auf Facebook, Flickr, Picasa etc. zu exportieren, als HTML-Galerie zu speichern, in eine Flash-Präsentation umzuwandeln oder in ein beliebiges Verzeichnis auf dem lokalen Rechner oder einem Rechner im Netzwerk zu speichern (**AUF FREMDRECHNER EXPORTIEREN**).

Wenn Sie mehrere zuvor markierte Bilder als E-Mail versenden möchten, führen Sie **EXPORTIEREN • BILDER PER E-MAIL VERSENDEN** aus. Dabei können Sie zwischen verschiedenen E-Mail-Clients auswählen und wahlweise die maximale E-Mail-Größe oder die gewünschte Bildgröße (z. B. maximal 800 Pixel) vorgeben. digiKam skaliert die Bilder dann entsprechend. Den Exportdialog dürfen Sie erst schließen, nachdem Sie die E-Mail versandt haben – andernfalls findet das E-Mail-Programm die temporären Bilddateien nicht mehr!

## 7.9 GIMP

GIMP ist das Open-Source-Gegenstück zu Adobe Photoshop. Auch wenn GIMP nicht alle Funktionen von Photoshop aufweisen kann, so ist es doch ein unglaublich vielseitiges Werkzeug zur Bildbearbeitung. Sie können damit Fotos retuschieren, Bilder für Ihre Website optimieren, Plakate gestalten etc.

Leider ist die Bedienung von GIMP alles andere als einfach. Das Programm ist deswegen in erster Linie als Werkzeug für Bildverarbeitungsprofis geeignet; Gelegenheitsanwender werden mit ihm nicht glücklich werden. In diesem Abschnitt stelle ich lediglich einige Grundfunktionen des Programms vor.

### Einzelfenster-Modus

GIMP verfügt seit Version 2.8 über einen Einzelfenster-Modus (siehe [Abbildung 7.15](#)), den Sie im **FENSTER**-Menü aktivieren müssen. Damit werden die vielen Teilfenster von GIMP zu einem einzigen Fenster zusammengefügt.

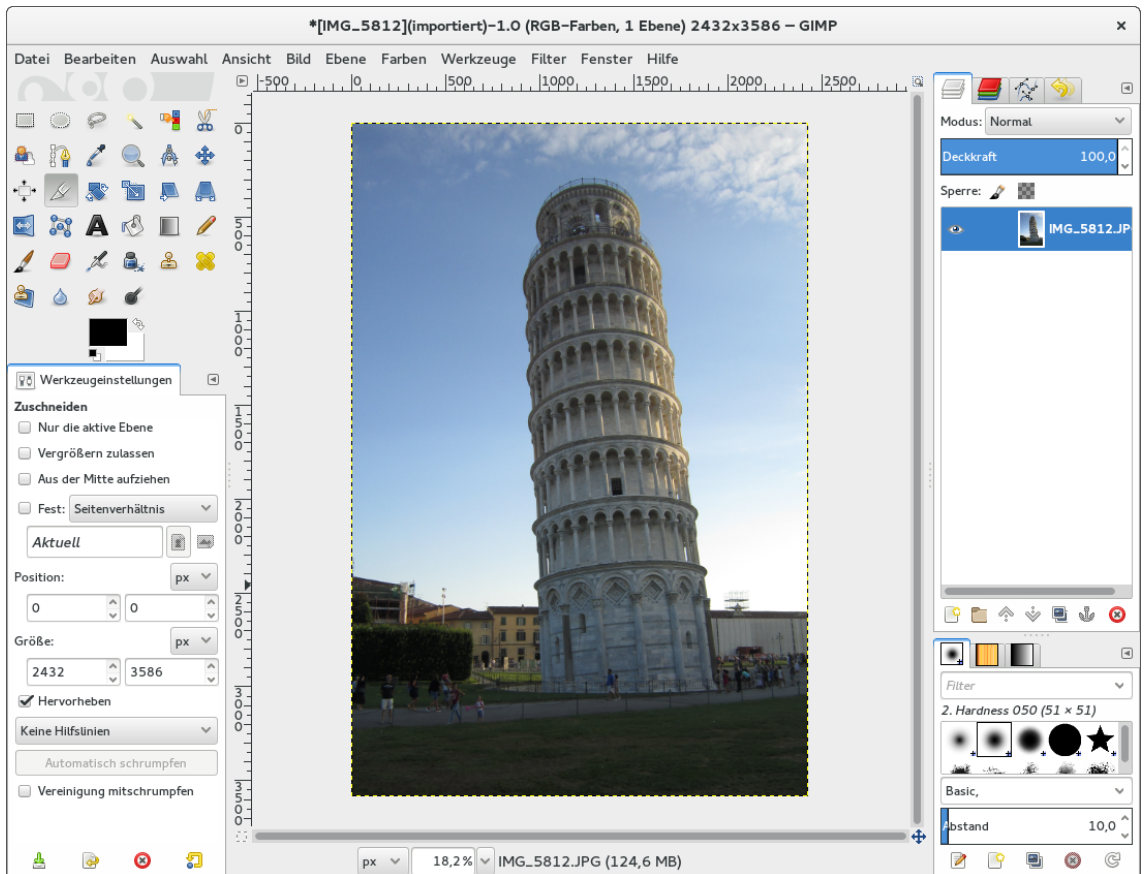


Abbildung 7.15 GIMP im Einzelfenster-Modus

DATEI • ÖFFNEN führt zu einem Dateiauswahldialog samt Bildvorschau. Nach dem Öffnen wird die Bilddatei in einem neuen Bildfenster angezeigt. Wenn das Bildfenster leer war, ersetzt das neue Bildfenster das bisherige.

**Bilder laden und bearbeiten**

Solange das Bildfenster aktiv ist, können Sie mit  $\oplus$  in das Bild hineinzoomen und mit  $\ominus$  hinaus.  $\boxed{1}$  setzt den Zoomfaktor auf 1:1, das heißt, jedes Pixel des Bilds wird auf einem Bildschirmpixel abgebildet.

DATEI • SPEICHERN speichert die Bilddatei im XCF-Format (Kennung \*.xcf). Der Vorteil dieses Formats besteht darin, dass nicht nur das Bild an sich gespeichert wird, sondern auch dessen Komposition sowie diverse Kontextinformationen und GIMP-Einstellungen. Wenn Sie dem Bild beispielsweise Text oder Ausschnitte anderer Bilder hinzugefügt haben, setzt sich das Bild aus mehreren Ebenen zusammen. Nur im XCF-Format werden alle Ebenen gespeichert. Das XCF-Format hat somit den Vorteil, dass es viel bessere Voraussetzungen für eine spätere Weiterverarbeitung des Bilds bietet.

Wenn Sie statt der Dateikennung \*.xcf die Kennungen \*.xcf.gz oder \*.xcf.bz2 verwenden, wird die Bilddatei zusätzlich komprimiert. Die Datei wird dadurch deutlich kleiner.

**Bilder speichern** Mit DATEI • SPEICHERN bzw. **[Strg]+[S]** können Sie nur im GIMP-eigenen XCF-Format speichern. Um ein Bild in einem anderen Format zu speichern, müssen Sie DATEI • EXPORTIEREN bzw. **[Strg]+[E]** ausführen. Wenn Sie das Export-Format ändern möchten, führen Sie **[⇧]+[Strg]+[E]** aus.

**Bilder drehen** Mit BILD • TRANSFORMATIONEN drehen Sie das Bild um 90 Grad nach rechts oder links, stellen es auf den Kopf oder spiegeln es horizontal oder vertikal.

**Bilder skalieren und ausschneiden** Mit BILD • BILD SKALIEREN gelangen Sie in den Skalierungsdialog. Dort geben Sie einfach die gewünschte Bildgröße in Pixel an. Alternativ kann die Größenangabe auch in Prozent erfolgen, z. B. um die Breite und Höhe des Bilds auf 25 Prozent seiner Größe zu verringern.

Um das Bild auf einen Ausschnitt zu verkleinern, aktivieren Sie in der Toolbox das Zuschneidewerkzeug (WERKZEUGE • TRANSFORMATIONEN • ZUSCHNEIDEN). Anschließend können Sie mit der Maus den gewünschten Bildausschnitt markieren. Ein Mausklick in den markierten Bereich schneidet das Bild aus.

**Helligkeit, Kontrast und Farben ändern** Mit WERKZEUGE • FARBEN • HELLGKEIT-KONTRAST gelangen Sie in einen einfachen Dialog, in dem Sie die Helligkeit und den Kontrast mit zwei Schieberegler verändern können.

Fotos nutzen selten den gesamten Farbraum. Der hellste Punkt im Bild, der oft weiß sein sollte, ist meist nur ein flauer Grauton. Mit FARBEN • WERTE können Sie diesen Mangel beheben. Der WERTE-Dialog bietet eine Menge Bearbeitungsmöglichkeiten, von denen hier nur die wichtigsten erwähnt werden:

- ▶ Mit dem Button AUTOMATISCH führen Sie einen automatischen Weißabgleich durch. Das Ergebnis ist zwar mathematisch optimal, liefert aber oft eine zu extreme Helligkeits- bzw. Farbverteilung.
- ▶ Mit den drei Pipetten-Buttons markieren Sie jeweils einen Punkt im Bild, der schwarz, in einem mittleren Grau erscheinen bzw. weiß sein sollte.
- ▶ Im Dialogbereich QUELLWERTE können Sie die drei Dreiecke verschieben, um so den Weiß-, Grau- und Schwarzpunkt zu markieren. Das darüber angezeigte Histogramm gibt an, wie viele Punkte des Bilds eine bestimmte Helligkeit haben. Üblicherweise wird der Schwarzpunkt an den Beginn und der Weißpunkt an das Ende des Histogramms verschoben. Der Graupunkt sollte in der Mitte zwischen Weiß- und Schwarzpunkt liegen. Wenn Sie den Graupunkt verschieben, wird das Bild blasser (links) bzw. farbintensiver (rechts).

Mit den folgenden Operationen können Sie die Erscheinungsqualität eines Bilds spürbar verbessern. Beachten Sie, dass alle hier beschriebenen Filter immer nur für den gerade markierten Bildbereich gelten. Führen Sie gegebenenfalls vorher **Strg**+**A** aus, um das gesamte Bild zu markieren!

Bild schärfen oder weichzeichnen

- ▶ **Schärfen:** FILTER • VERBESSERN • SCHÄRFEN versucht das Bild zu schärfen, indem es Helligkeitsveränderungen betont. Relativ gut funktioniert das bei Nachtaufnahmen. Eine mögliche Alternative ist das Kommando FILTER • VERBESSERN • NL FILTER mit der Option KANTENVERSTÄRKUNG. Auch der Filter VERBESSERN • UNSCHARF MASKIEREN ist einen Versuch wert.
- ▶ **Weichzeichnen:** Die gegenteilige Wirkung haben die diversen Kommandos unter FILTER • WEICHZEICHNEN. Diese Filter mindern Helligkeitsübergänge. Das Bild wirkt dadurch weicher, aber auch etwas unschärfer. Relativ starke Effekte erzielen Sie mit dem GAUSSSCHEN WEICHZEICHNER.
- ▶ **Rauschen eliminieren:** Geradezu spektakuläre Verbesserungen bei verrauschten Bildern (auch bei schlecht eingescannten Fotos) erzielen Sie mit FILTER • WEICHZEICHNEN • SELEKTIVER GAUSSSCHER WEICHZEICHNER. Probieren Sie es beispielsweise mit einem Radius von 4 Pixeln und einem maximalen Deltawert von 10. Das bedeutet, dass der Weichzeichner nur dann zum Einsatz kommt, wenn der Farbunterschied nahe beieinander liegender Pixel gering ist (kleiner gleich 10). Bei starken Farbunterschieden – z. B. entlang einer Hauskante – bleibt der Weichzeichner dagegen unwirksam, weswegen die Schärfe des Bilds weniger leidet als bei anderen Weichzeichnern.

Der Rote-Augen-Effekt entsteht vor allem bei Porträtaufnahmen, wenn der Blitz nahe am Objekt ist: Die Pupillen sind weit geöffnet. Deswegen wird das Blitzlicht von der durchbluteten Netzhaut rot reflektiert.

Rote Augen entfernen

GIMP enthält ein eigenes Werkzeug zur Eliminierung des Rote-Augen-Effekts. Bevor Sie es anwenden können, müssen Sie den roten Bereich der Augen markieren. Dazu verwenden Sie das Werkzeug ELLIPTISCHE AUSWAHL. Beim zweiten Auge drücken Sie zusätzlich **⇧**, um die bereits vorhandene Markierung zu ergänzen. Markieren Sie lieber ein bisschen zu viel als ein bisschen zu wenig!

FILTER • VERBESSERN • ROTE AUGEN ENTFERNEN ersetzt nun das Rot der Augen durch einen Grauton. Der Lichtreflex im Auge bleibt dabei erhalten. Den Schwellenwert für den Rot-Ton, ab dem die Farbe verändert wird, müssen Sie nur in Ausnahmefällen verändern.



## 7.10 RawTherapee, Darktable und Luminance (RAW- und HDR-Bilder)

Die meisten Digitalkameras speichern Bilder im JPEG-Format, das einen guten Kompromiss zwischen Bildgröße und Qualität bietet. Bessere Kameras bieten darüber hinaus die Möglichkeit, Bilder im sogenannten RAW-Format zu speichern. Dabei handelt es sich um herstellerspezifische Formate, die sicherstellen, dass keinerlei Bildinformationen verloren gehen. Allerdings sind RAW-Dateien zumeist sehr groß und können nur mit speziellen Programmen betrachtet werden. Unter Linux sind hierfür die Programme RawTherapee und Darktable geeignet. Diese Programme richten sich explizit an professionelle Fotografen, die das Optimum aus RAW-Fotodateien herauskitzeln möchten. Mit Luminance erstellen Sie hingegen HDR-Bilder.

**RawTherapee** Mit RawTherapee (siehe [Abbildung 7.16](#)) können Sie RAW-Bilddateien der meisten gängigen Kameras laden. Anschließend haben Sie unzählige Möglichkeiten, das Bild durch Filter und andere Funktionen zu optimieren und schließlich in einem anderen Bildformat zu speichern.

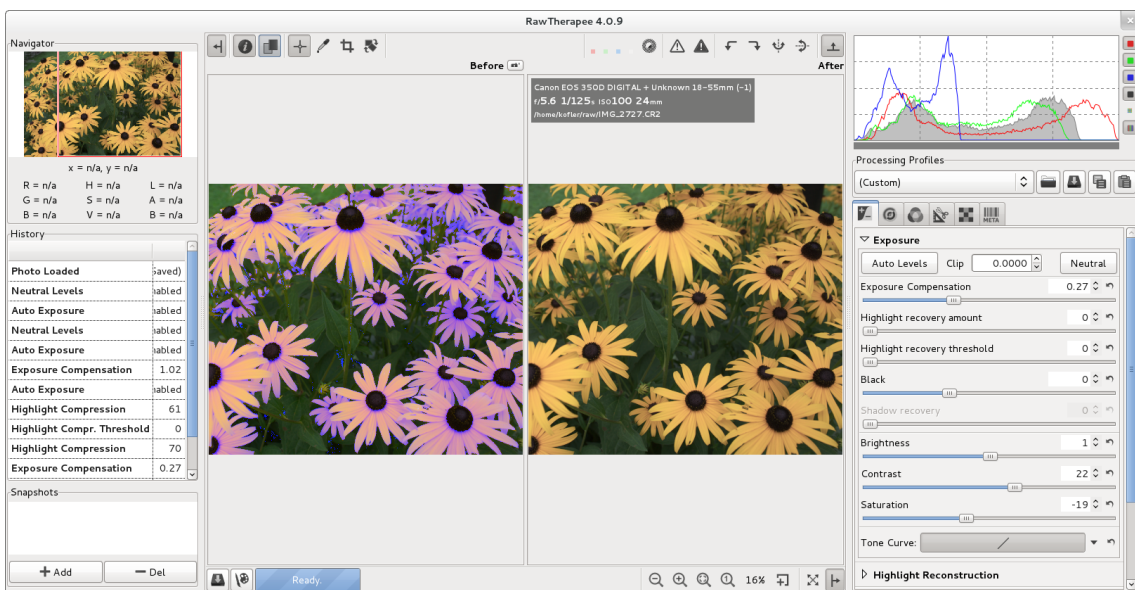


Abbildung 7.16 RawTherapee

**Darktable** Darktable (siehe [Abbildung 7.17](#)) versucht, den ganzen »Fotoworkflow« unter Linux abzubilden, vom Import der RAW-Daten bis hin zur Anwendung von Bildbearbeitungsfunktionen. Die Funktionen zur Verarbeitung der RAW-Daten sind ähnlich wie



bei RawTherapee. Darüber hinaus bietet Darktable auf dem sogenannten Leuchttisch aber auch Funktionen, um Bilder zu bewerten, mit Tags zu versehen etc. Die Bedienung von Darktable orientiert sich in Grundzügen an Adobe Photoshop Lightroom.

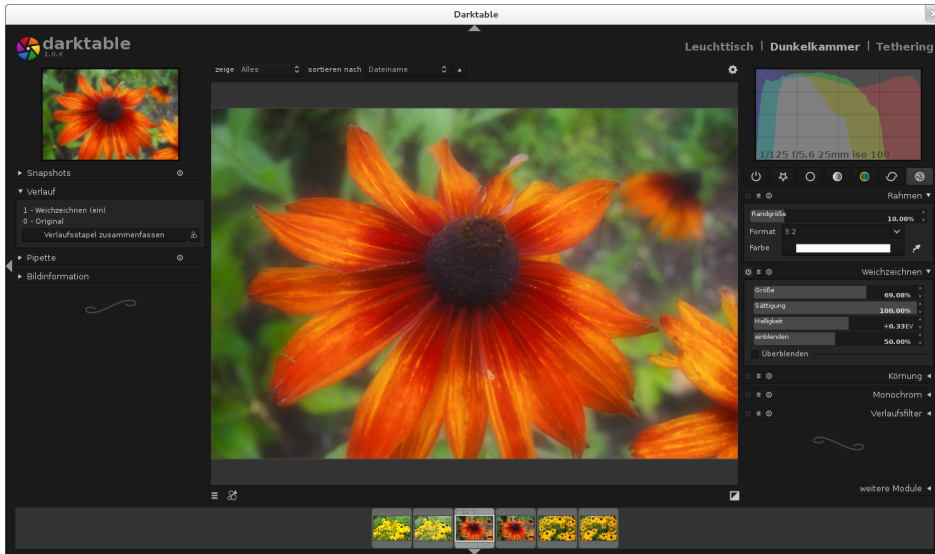


Abbildung 7.17 Darktable

Auf Kommandoebene können Sie RAW-Dateien mit `dcraw` in andere Bildformate umwandeln. Wenn Sie mit GIMP arbeiten (siehe [Abschnitt 7.9](#)), ermöglicht das Zusatzpaket `gimp-dcraw` den direkten Import von RAW-Dateien, ohne aber so viele Einstellmöglichkeiten wie RawTherapee oder Darktable zu bieten. Weitere Informationen zu `dcraw` finden Sie unter:

<https://www.cybercom.net/~dcoffin/dcraw>

HDR steht für *High Dynamic Range* und ist eine Technik, um auch bei extremen Helligkeitsunterschieden sowohl die dunklen als auch die hellen Teile des Bilds in guter Qualität darzustellen. Zur Erstellung von HDR-Bildern werden üblicherweise zwei Fotos mit unterschiedlicher Belichtung zusammengefügt.

Luminance HDR

Unter Linux hilft dabei das Programm Luminance HDR. Einige Distributionen (darunter Ubuntu) bieten das Programm in den Standardpaketquellen an. Sollte das bei Ihrer Distribution nicht der Fall sein, können Sie das Programm hier herunterladen:

<https://github.com/LuminanceHDR/LuminanceHDR>

## 7.11 Multimedia-Grundlagen

Bei aller Begeisterung für Linux – das ideale System für Multimedia-Anwendungen ist es leider nicht. Die Einschränkungen sind nicht technischer Art, sondern ergeben sich aus den rechtlichen Rahmenbedingungen: Patente und Verschlüsselungstechnologien verhindern, dass Linux von Haus aus multimediatauglich ist. Natürlich gibt es entsprechende Codecs und Bibliotheken, für deren Installation ist aber Handarbeit angesagt.

### Encoder und Decoder

**Encoder** Encoder wandeln unkomprimierte Audio- oder Video-Daten in ein komprimiertes Format um (z.B. MP3, Ogg oder MPEG-4). Die Aufgabe des Encoders ist es, die Daten einerseits möglichst stark zu komprimieren, andererseits aber für geringe Qualitätsverluste zu sorgen. Das ist ein rechenintensiver und daher verhältnismäßig langsamer Vorgang.

**Decoder** Decoder sind für die umgekehrte Richtung zuständig, also für die Umwandlung der komprimierten Daten in ein Format, das an Soundkarten bzw. die Grafikkarte weitergegeben werden kann. Jeder Audio- oder Video-Player muss daher auf Decoder für das jeweilige Format zurückgreifen. Decoder benötigen Sie aber auch, wenn Sie komprimierte Dateien in ein unkomprimiertes Format zurückverwandeln möchten (siehe auch [Abschnitt 13.2](#), »Audio- und Video-Konverter«). Das ist zweckmäßig, wenn Sie eine herkömmliche Audio-CD erzeugen möchten – denn dazu benötigen Sie unkomprimierte WAV-Dateien.

Manchmal besteht auch der Wunsch, Audio- oder Video-Dateien von einem Format in ein anderes umzuwandeln oder stärker zu komprimieren. Dieser Vorgang wird üblicherweise als *Recodieren* (Recoding) bezeichnet.

**Codecs** Das dem Encoder/Decoder zugrunde liegende Verfahren wird als Codec bezeichnet. Umgangssprachlich meint Codec aber zumeist die Bibliothek bzw. das Modul/Plugin zur (De-)Codierung eines bestimmten Audio/Video-Formats.

Es existieren zahllose Codecs, wobei für Windows und macOS der Decoder-Teil zumeist kostenlos verfügbar ist. Codec-Entwickler versuchen damit einen möglichst hohen Marktanteil ihres Formats zu erreichen. Etwas schwieriger ist die Situation unter Linux: Im Rahmen der Projekte FFmpeg und avconv gibt es zu vielen populären Codecs Open-Source-Implementierungen für den Decoder und zumeist auch für den Encoder. Allerdings ist der rechtliche Status dieser Programme bzw. Bibliotheken teilweise zweifelhaft, weil viele Codecs durch Patente und Lizenzen geschützt sind. Mangels besserer Alternativen greifen dennoch die meisten Audio- und Video-Player

auf die Bibliotheken dieser Projekte zurück. Dies gilt z. B. für MPlayer, VLC, xine sowie für alle Programme, die auf GStreamer basieren.

Aufgrund der unklaren rechtlichen Situation, die auch von der nationalen Gesetzgebung abhängt, sind im Standardlieferungsumfang der meisten Linux-Distributionen nur wenige Codecs enthalten. Immerhin sind die meisten aktuellen Linux-Distributionen jetzt endlich ad hoc MP3-tauglich: Die wichtigsten MP3-Patente sind nämlich 2017 ausgelaufen.

### Verschlüsselung, CSS, DRM

Nahezu alle Video-DVDs sind durch das CSS (*Content Scrambling System*) verschlüsselt. Der dadurch erreichte Schutz hat sich freilich als gering erwiesen. Die Verschlüsselung ist ziemlich primitiv und wurde rasch geknackt. CSS

Weit mehr Mühe als mit dem Verschlüsselungsalgorithmus hat sich die Medienindustrie gegeben, um jegliche Open-Source-Techniken zur Entschlüsselung zu kriminalisieren. Aus diesem Grund ist der Einsatz einer Entschlüsselungsbibliothek in vielen Ländern verboten.

In Deutschland ist es aufgrund des Urheberrechtsgesetzes sogar verboten, die Installation einer Bibliothek zur CSS-Entschlüsselung zu beschreiben. Auch wenn es im Internet unzählige Websites mit entsprechenden Anleitungen gibt, darf ich diese Informationen hier weder wiedergeben noch einen entsprechenden Link nennen. Die Grenzen der Pressefreiheit sind enger, als man denkt.

Damit Sie mich richtig verstehen: Es geht hier nicht um illegales Kopieren! Ich darf Ihnen nicht einmal erklären, wie Sie Ihre gerade in einem Geschäft erworbene DVD unter Linux ansehen können. Ein gewisser Trost ist hier nur der Umstand, dass dies auch unter Windows und macOS immer schwieriger wird. Zudem sind Notebooks und PCs mit einem DVD-Laufwerk im Aussterben begriffen.

CSS war nur der Anfang: Blu-ray Discs sehen wesentlich bessere Schutzmechanismen vor, die zum Teil direkt in der Hardware implementiert werden müssen, also beispielsweise in der Grafikkarte. Zwar wurden auch diese Schutzmaßnahmen bereits geknackt, die Vorgehensweise ist aber komplizierter geworden. Blu-ray

DRM steht für *Digital Rights Management*. Mit dieser Technik wird eine Audio- oder Video-Datei an eine bestimmte Hardware gebunden. Die Datei kann zwar mühelos kopiert, auf einem anderen Rechner aber nicht abgespielt werden. DRM

Digitale Musik wird mittlerweile zunehmend DRM-frei verkauft – nicht zuletzt aufgrund des Drucks, den Apple auf die Musikindustrie ausgeübt hat. Tot ist DRM leider noch lange nicht: Obwohl es bei Musik nicht funktioniert hat, versuchen Medien-

anbieter nun Videos, E-Books etc. DRM-geschützt zu verkaufen. Linux-Anwender sind von der legalen Nutzung DRM-geschützter Medien weitgehend ausgeschlossen, und eine Besserung dieser Situation ist nicht in Sicht.

### Multimedia-Zusatzpakete installieren

Weil die Gesetzgebung sowie die Reichweite bzw. Gültigkeit von Patenten je nach Land unterschiedlich sind, ist eine standardmäßige Auslieferung diverser Codecs und Entschlüsselungs-Software unmöglich. Da Linux-Distributionen international heruntergeladen werden, müssen sie dem kleinsten gemeinsamen Nenner entsprechen.

Zu vielen Distributionen gibt es aber inoffizielle Paketquellen, in denen solche Pakete gesammelt sind (siehe [Tabelle 7.2](#)). Sie müssen diese Paketquellen zumeist selbst einrichten und können die gewünschten Pakete dann herunterladen. Auf den Websites der Paketquellen werden Sie oft einen Hinweis finden, dass Sie sich vor dem Download vergewissern müssen, dass die Verwendung der so zur Verfügung gestellten Software in Ihrem Land zulässig ist.

| Distribution | Multimedia-Website oder -Paketquelle  |
|--------------|---|
| Debian       | <a href="https://deb-multimedia.org">https://deb-multimedia.org</a>   |
| Fedora       | <a href="https://fedoraproject.org/wiki/Multimedia">https://fedoraproject.org/wiki/Multimedia</a><br><a href="https://rpmfusion.org">https://rpmfusion.org</a><br><a href="http://rpm.livna.org">http://rpm.livna.org</a> |
| openSUSE     | <a href="https://en.opensuse.org/Restricted_formats">https://en.opensuse.org/Restricted_formats</a><br><a href="http://packman.links2linux.de">http://packman.links2linux.de</a>  |
| Ubuntu       | <a href="https://help.ubuntu.com/community/RestrictedFormats">https://help.ubuntu.com/community/RestrictedFormats</a>   |

**Tabelle 7.2** Populäre Multimedia-Websites und -Paketquellen

#### gstreamer-Bibliotheken

Distributionsspezifische Tipps zur Nutzung dieser Paketquellen finden Sie in [Kapitel 3](#). Ein guter Startpunkt ist zumeist die Aktivierung der für Ihre Distribution geeigneten Paketquellen und danach die Installation der folgenden Pakete:

```
root# apt/dnf/zypper install gstreamer*-plugins-bad* gstreamer*-plugins-ugly*
```

gstreamer ist ein Multimedia-Framework, das von vielen Programmen genutzt wird – besonders im Gnome-Umfeld, aber auch darüber hinaus. Die Codec-Bibliotheken sind in mehrere Klassen eingeteilt: base, good, ugly und bad. Die base- und good-Bibliotheken liegen als Open-Source-Code vor und sind – soweit die Entwickler dies beurteilen können – frei von Patentproblemen. Bei den ugly-Bibliotheken gibt es möglicherweise Lizenzprobleme. Bei den bad-Bibliotheken ist zudem die Code-Qualität zweifelhaft.

<https://gstreamer.freedesktop.org/documentation/splitup.html>

### Ubuntu – Multimedia-Genuss für Linux-Einsteiger

Wenn Sie sich nicht mit der Suche nach Paketquellen und den richtigen Codecs plagen wollen, empfehle ich Ihnen Ubuntu oder eine von Ubuntu abgeleitete Distribution. Sie installieren einfach das Paket `ubuntu-restricted-extras` bzw. je nach Variante `kubuntu-restricted-extras`, `xubuntu-restricted-extras` etc., und nahezu alle Codec-Probleme sind gelöst.

## 7.12 Rhythmbox, Amarok & Co

Dieser Abschnitt setzt voraus, dass Sie eine Sammlung von MP3-Dateien auf Ihrer Festplatte/SSD oder auf einem NAS-Gerät haben und gezielt einzelne Alben oder zufällige Titel eines Genres abspielen möchten. Standardmäßig sehen die meisten Gnome-basierten Distributionen zu diesem Zweck das Programm Rhythmbox vor, KDE-Distributionen Amarok. Diese beiden Programme stelle ich Ihnen im Folgenden kurz vor. Außerdem gebe ich Ihnen eine Übersicht über alternative Audio-Player (die Auswahl ist groß!).

Beachten Sie, dass die Player nur für die Benutzeroberfläche zuständig sind. Welche Audio-Formate die Programme abspielen können, hängt davon ab, welche zum Player passenden Codec-Bibliotheken installiert sind (siehe den vorigen Abschnitt).

### Rhythmbox

Rhythmbox bzw. einfach *Musik*, wie sich das Programm in aktuellen Versionen nennt, ist der Standard-Audio-Player des Gnome-Desktops (siehe [Abbildung 7.18](#)). Soweit sich Ihre Audio-Dateien nicht im Verzeichnis `Musik` befinden, müssen Sie das entsprechende Verzeichnis zuerst mit `MUSIK HINZUFÜGEN` erfassen. Rhythmbox überwacht diesen Ordner nun auf Veränderungen; die entsprechende Option finden Sie in `EINSTELLUNGEN • MUSIK`.

Wenn Sie Ihre Audio-Verzeichnisse grundlegend ändern, ist es das Beste, in Rhythmbox alle Titel zu markieren, per Kontextmenü zu entfernen und anschließend neu zu importieren. Verwenden Sie zum Löschen von Titeln aus der Datenbank aber auf keinen Fall das Kommando `IN DEN PAPIERKORB VERSCHIEBEN`! Dieses Kommando betrifft nicht die Rhythmbox-Musikdatenbank, sondern es löscht Ihre MP3-Dateien! Rhythmbox speichert die Musikdatenbank in der Datei `.local/share/rhythmbox/rhythmdb.xml`.

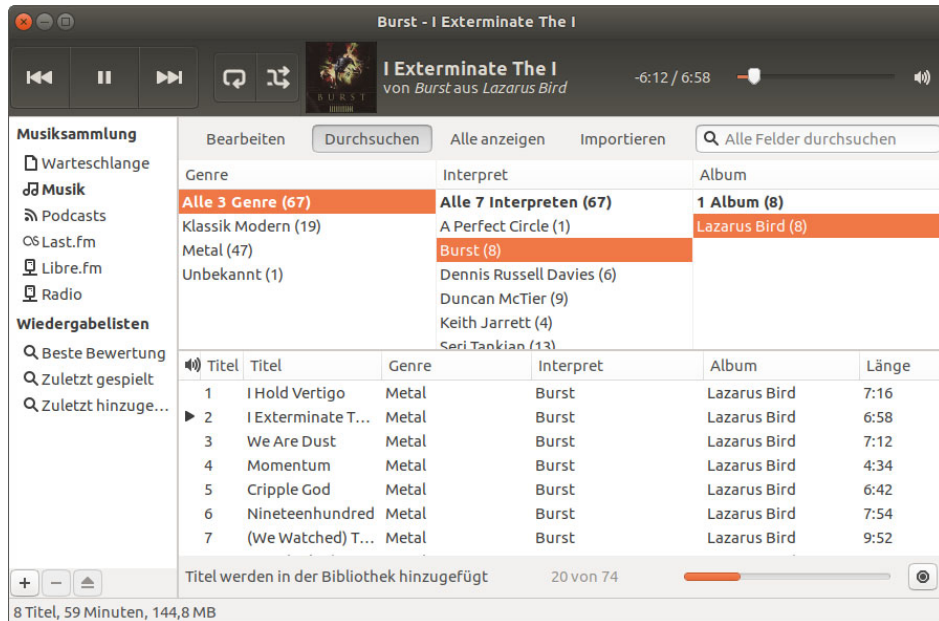


Abbildung 7.18 Audio-Dateien mit dem Gnome-Programm Rhythmbox anhören

Die Bedienung von Rhythmbox ist einfach: Sie wählen ein Genre, einen Interpreten und/oder ein Album aus und klicken auf den Button WIEDERGABE. Rhythmbox spielt nun alle in der Liste angezeigten Titel. Die Genre-Auswahlliste wird standardmäßig nicht angezeigt. Um die Liste einzublenden, führen Sie EINSTELLUNGEN • ALLGEMEIN aus und wählen die Browser-Ansicht GENRES, KÜNSTLER UND ALBEN.

Eigene Wiedergabelisten erzeugen Sie in der Seitenleiste mit dem Plus-Button. Anschließend fügen Sie die gewünschten Titel per Drag & Drop in die neue Liste ein. Es ist auch möglich, ganze Genres, Interpreten oder Alben einzufügen.

Rhythmbox kann auch zum Anhören von Internet-Radiostationen verwendet werden und durch Erweiterungen um Zusatzfunktionen ergänzt werden. Standardmäßig sind zumeist schon eine Menge Erweiterungen installiert, aber nur ein Teil davon ist aktiv. Werfen Sie einen Blick in den Dialog ERWEITERUNGEN, den Sie über das gleichnamige Menükommando öffnen.

## Amarok

Amarok (siehe [Abbildung 7.19](#)) ist das populärste und ausgereifteste KDE-Programm zum Abspielen von Audio-Dateien und zur Verwaltung großer Audio-Bibliotheken. Amarok greift zur Audio-Wiedergabe auf das KDE-Sound-System Phonon zurück.

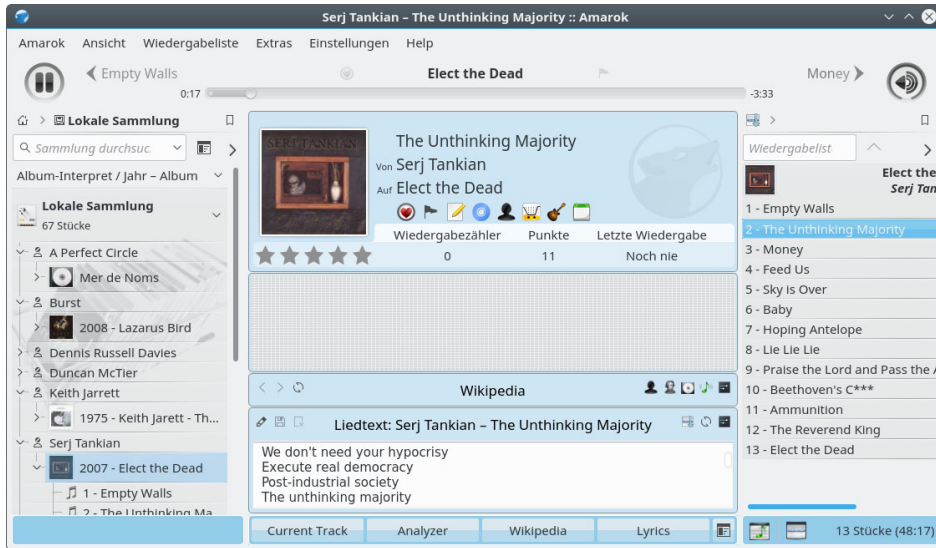


Abbildung 7.19 Audio-Dateien mit Amarok anhören

Amarok erstellt beim ersten Start eine Bibliothek aller Audio-Dateien im Verzeichnis Musik. Amarok speichert diese Informationen in einer Datenbank. Über das Kontextmenükommando METADATEN BEARBEITEN können Sie die ID3-Tags eines einzelnen Titels oder eines ganzen Albums ändern.

Ihre Audio-Sammlung können Sie in der linken Seitenleiste LOKALE SAMMLUNG anzeigen und nach verschiedenen Kriterien ordnen. Ein Doppelklick auf ein Genre oder Album fügt alle entsprechenden Tracks der Wiedergabeliste (rechts) hinzu. Im mittleren Teil des Fensters können Sie Informationen zum gerade gespielten Titel einblenden, z. B. den Liedtext oder die Wikipedia-Seite der Band.

## Alternativen

Das Angebot an Audio-Playern ist unüberschaubar groß. Die folgende Liste zählt einige Programme auf. Beachten Sie, dass viele Projekte voller Enthusiasmus und Engagement gestartet worden sind, mittlerweile aber nur noch halbherzig verfolgt werden oder ganz eingeschlafen sind:

- ▶ **Audacious** (<http://audacious-media-player.org>) ist ein schlanker Audio-Player. Er eignet sich besonders gut dazu, einfach die Audio-Dateien eines Verzeichnisses abzuspielen, ohne gleich eine ganze Musikdatenbank zu erfassen.
- ▶ **Babe** (<https://milohr.github.io/Babelt>) ist ein eleganter Audio-Player auf der Basis der Qt-Bibliotheken. Dieses Programm ist besonders gut geeignet, wenn Sie KDE nutzen.



- ▶ **cmus** (<https://cmus.github.io>) ist ein attraktiver Audio-Player für den Textmodus.
- ▶ **Musique** (<http://flavio.tordini.org/musique>) ist ein minimalistischer Audio-Player mit einer schlanken und eleganten Benutzeroberfläche. Das Programm zeigt alle Künstler, Alben oder Ordner in Form von Icons an. Die Ordneransicht ist besonders attraktiv, wenn Sie Ihre Musik in Form von Verzeichnissen organisiert haben.

Einige weitere Programme hat die Website OMG Ubuntu zusammengestellt:

<http://www.omgubuntu.co.uk/2017/01/best-music-player-apps-ubuntu-linux>

## 7.13 Spotify

Das Verwalten einer eigenen MP3-Bibliothek ist mühsam, das Einlesen (»Rippen«) der CD-Sammlung im Keller zeitaufwendig und der Kauf neuer MP3-Dateien teuer. Außerdem ist die eigene MP3-Sammlung unterwegs schwer zu nutzen. All diese Gründe machen Streaming-Angebote wie Spotify, Apple Music oder Amazon attraktiv. Das Problem ist aber die Nutzung dieser Dienste unter Linux.

Spotify ist aktuell die einzige Firma, die überhaupt einen Client für Linux anbietet (siehe [Abbildung 7.20](#)). Allerdings beschränkt sich das Angebot auch in diesem Fall auf die Ubuntu-Familie. Zudem wird der Linux-Client nicht offiziell unterstützt. Eine Installationsanleitung finden Sie hier:

<https://www.spotify.com/at/download/linux>

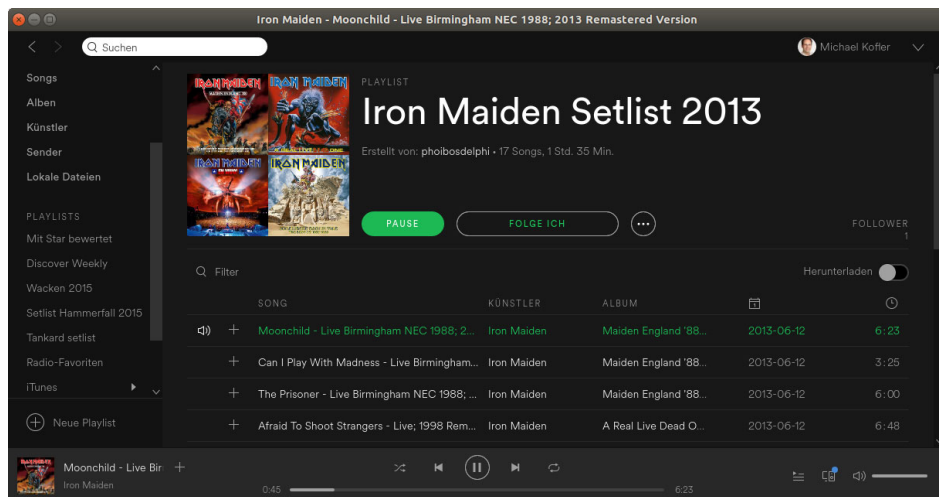


Abbildung 7.20 Spotify



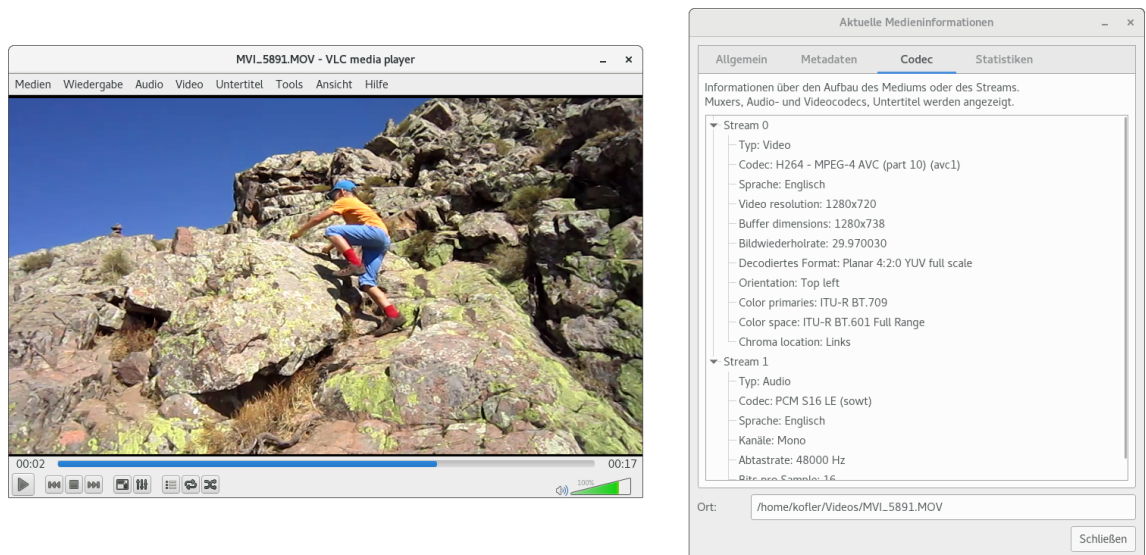
Nach dem ersten Login sollten Sie die Sprache von Spotify auf DEUTSCH umstellen. Die entsprechende Auswahlliste finden Sie im Einstellungsdialog. Bei meinen Tests hat die Spotify-Vorversion problemlos und stabil funktioniert.

Wenn Sie nicht Spotify, sondern einen anderen Streaming-Dienst verwenden, können Sie den Nuvola Player ausprobieren. Dieses Programm unterstützte zuletzt immerhin 19 Streaming-Dienste (nicht aber Apple Music). Die aktuelle Version ist als Flatpak-Paket erhältlich.

<https://tiliado.eu/nuvolaplayer>

## 7.14 VLC

Die Auswahl unter Video-Playern für Linux ist fast so groß wie bei den Audio-Playern. Ich empfehle Ihnen das Programm VLC (ehemals *VideoLan Client*, siehe [Abbildung 7.21](#)), der aus meiner Sicht das beste Programm ist. VLC greift auf externe Codec-Bibliotheken zurück (z. B. FFmpeg, libmpeg2 und x264). Seine Benutzeroberfläche basiert auf der Qt-Bibliothek.



**Abbildung 7.21** Der VLC-Player mit detaillierten Codec-Informationen

Eine Besonderheit des Programms besteht darin, dass Filtereffekte in Echtzeit angewendet werden können. Das ermöglicht es z. B., ein mit einer Digitalkamera hochkant aufgenommenes Video beim Abspielen richtig zu drehen. **TOOLS • MEDIENINFORMATIONEN** zeigt an, welche Codecs die Video-Datei verwendet.

**Installation** Unter Fedora müssen Sie vor der Installation von VLC die `rpmfusion`-Paketquelle einrichten. openSUSE stellt zwar standardmäßig VLC-Pakete zur Verfügung, dennoch empfiehlt sich hier die Aktivierung der PackMan-Paketquelle. Sie enthält mehr Codecs, die sich teilweise in eigenen Paketen befinden (`vlc-codecs`).

Wenn VLC eine Codec-Bibliothek nicht findet, liefert es die Fehlermeldung: *VLC unterstützt das Audio- oder Videoformat <name> nicht. Leider können Sie daran nichts ändern.* Lassen Sie sich davon nicht entmutigen, sondern stellen Sie sicher, dass auch die VLC-Codec-Pakete installiert sind.

### Alternativen

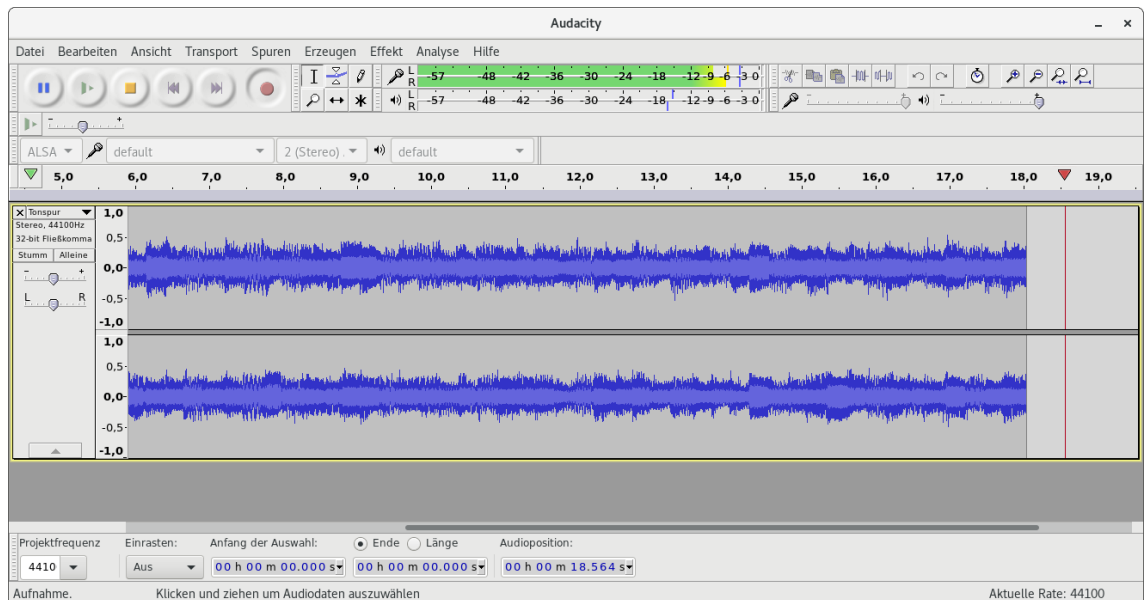
- ▶ **Dragon und Kaffeine** sind zwei populäre Video-Player des KDE-Desktops.
- ▶ **Kodi** ist ein komplettes Multimedia-Center. Wie Sie Kodi auf einem Raspberry Pi einsetzen, zeigt [Abschnitt 8.3](#), »Kodi und LibreELEC«.
- ▶ **MPlayer** war im vorigen Jahrzehnt der beste und populärste Multimedia-Player für Linux. Diesen Status hat das Programm aber mittlerweile verloren: Im Vergleich zu den anderen hier vorgestellten Programmen ist MPlayer sperrig in der Bedienung und schwierig bei der Konfiguration.
- ▶ **Totem** ist der Video-Player des Gnome-Desktops. Das Programm ist bei den meisten Distributionen standardmäßig installiert. Soweit Totem mit den Codecs zurechtkommt, kann es den Film abspielen. Zusatzfunktionen gibt es keine.
- ▶ **xine** ist ein Video-Player, dessen Benutzeroberfläche der eines alten DVD-Players nachempfunden ist. Die komplexen Konfigurationsdialoge wirken ebenfalls antiquiert. Die Stärken des Programms liegen in der Unterstützung zahlloser Audio- und Video-Formate durch eigene Bibliotheken. Die xine-Oberfläche ist vollständig von den zugrunde liegenden Bibliotheken getrennt. Das ermöglicht es auch anderen Programmen, auf die xine-Bibliotheken zurückzugreifen.

## 7.15 Audio- und Video-Tools

Dieser Abschnitt stellt einige Programme vor, die beim Erzeugen und Verwalten von Audio- und Video-Dateien helfen: Audacity hilft beim Aufnehmen und Schneiden von Audio-Dateien. Sound Converter wandelt Audio-Dateien in andere Formate um. EasyTAG erlaubt das Einstellen bzw. Ändern der ID3-Tags von MP3-Dateien, und Sound Juicer liest Audio-CDs aus und erzeugt daraus MP3-Dateien. Analog liest HandBrake DVDs ein und macht daraus Video-Dateien.

## Audacity

Audacity ist ein sehr vielseitiges, aber deswegen auch komplexes Programm: Sie können damit mehrere Audio-Spuren aufnehmen, bearbeiten, schneiden, übereinanderlegen, mit Effekten verändern etc. (siehe [Abbildung 7.22](#)). Ich stelle hier aber nur wenige, ganz elementare Funktionen vor, um Audio-Aufnahmen durchzuführen und Teile aus einer Audio-Datei herauszuschneiden.



**Abbildung 7.22** Audio-Tracks mit Audacity aufnehmen und schneiden

Es mag übertrieben erscheinen, für solche Aufgaben Audacity einzusetzen, aber das Programm erledigt nach einer kurzen Einarbeitung auch derart triviale Tätigkeiten effizienter und zuverlässiger als vorgeblich einfachere Audio-Tools. Wenn es Ihnen nur darum geht, einen Audio-Kanal aufzunehmen und das Ergebnis gleich als MP3- oder Ogg-Datei zu speichern, bietet sich unter Gnome der Einsatz des Programms `gnome-sound-recorder` an.

Um eine Aufnahme zu starten, stellen Sie in dem neben dem Mikrofonsymbol dargestellten Listenfeld das gewünschte Input-Device ein und klicken auf den roten Aufnahme-Button. Audacity erzeugt eine Stereo-Audio-Spur und beginnt unverzüglich mit der Aufnahme. Die aufgenommenen Daten werden unkomprimiert im Verzeichnis `.audacityN-name/projectN` gespeichert. Stellen Sie sicher, dass in Ihrem Heimatverzeichnis ausreichend Platz ist!

**Aufnehmen**

Standardmäßig können Sie die laufende Aufnahme nicht mithören. Wenn Sie aber mithören möchten, müssen Sie vor Beginn der Aufnahme `TRANSPORT • SOFTWARE`

PLAYTHROUGH bzw. TRANSPORT • SOFTWARE PLAYBACK aktivieren. Diese Funktion wird in den Audacity-FAQs aber zu Recht als *clunky* beschrieben und führte bei meinen Tests zu einer hohen CPU-Last und einem Abbruch der Aufnahme. Besser ist es, im Audio-Mixer, zur Not einfach mit `alsamixer` in einer Konsole, den Line- oder Mikrofon-Eingang zu aktivieren. Sie können die Wiedergabelautstärke dieses Kanals verändern, ohne die Aufnahme zu beeinflussen.

**Schneiden** Nach Abschluss der Aufnahme können Sie diese anhören und bei Bedarf verändern, also Teile ausschneiden oder bei leisen Aufnahmen die Lautstärke durch **EFFEKTE • NORMALISIEREN** anheben. Dazu müssen Sie den gewünschten Bereich zuerst markieren. Am einfachsten geht das mit der Maus, Audacity bietet aber unzählige weitere Möglichkeiten, um Beginn und Ende der Markierung exakt festzulegen.

**Speichern/Exportieren** **DATEI • EXPORTIEREN** speichert den markierten Bereich in einer Audio-Datei beliebigen Formats, **BEARBEITEN • TRIMMEN** löscht alles außer der Markierung, **[Entf]** löscht den markierten Bereich.

Wenn Sie ein Audacity-Projekt sichern, werden neben einer relativ kleinen Projektdatei alle Kanäle in einem eigenen, verlustfreien Format gespeichert, das in einem eigenen Verzeichnis `name_data` sehr viel Platz beansprucht. Um die Audio-Dateien problemlos mit einem anderen Programm anzuhören, exportieren Sie das Projekt im WAV-, Ogg- oder MP3-Format. Letzteres erfordert die Installation von `lame`. Wenn Sie bereits vorhandene MP3-Dateien oder andere Audio-Dateien bearbeiten möchten, laden Sie diese einfach in ein leeres Audacity-Projekt. Sobald der Import erledigt ist, haben Sie dieselben Bearbeitungsmöglichkeiten wie bei einer Aufnahme.

### Die aktuelle Tonausgabe mitschneiden

Je nach Audio-Konfiguration bzw. -Hardware scheint es für Audacity unmöglich zu sein, die aktuelle Tonausgabe aufzunehmen, also das, was Sie über den Lautsprecher des Computers gerade hören. Das wäre mitunter praktisch, beispielsweise, um den Ton eines Flash-Videos oder eines Internetradiosenders mitszuschneiden.

Falls Sie PulseAudio als Audio-System einsetzen, können Sie diese Einschränkung umgehen: Dazu starten Sie Audacity und das Programm, das als Audio-Quelle dienen soll, z. B. einen Webbrowser. In Audacity müssen Sie die Aufnahme starten!

Nun starten Sie das Programm `pavucontrol`. Dieses Programm dient zur Steuerung von PulseAudio und muss oft extra installiert werden. In `pavucontrol` wechseln Sie in das Dialogblatt **AUFNAHME** und stellen dort beim Punkt **ALSA PLUG-IN (AUDACITY)** als Aufnahmequelle **MONITOR OF <GERÄT>** ein (siehe [Abbildung 7.23](#)).

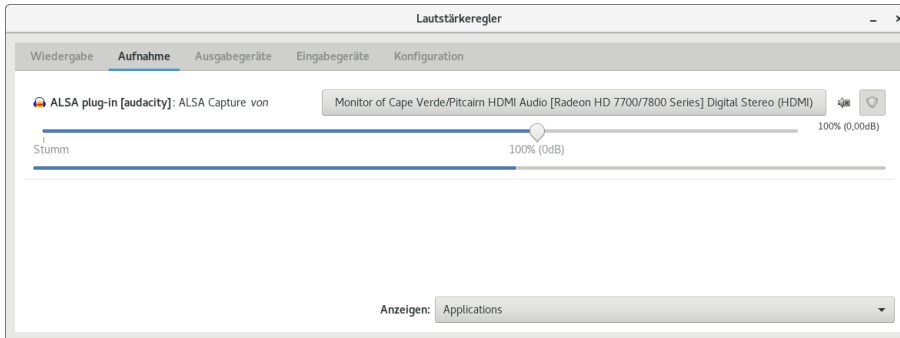


Abbildung 7.23 PulseAudio-Einstellungen, um die aktuelle Tonausgabe mitzuschneiden

## Sound Converter

Um Audio-Dateien von einem Format in ein anderes umzuwandeln, bedienen sich Linux-Profis diverser Kommandos (siehe [Abschnitt 13.2](#), »Audio- und Video-Konverter«). Sie können es sich natürlich auch leichter machen und stattdessen eine grafische Benutzeroberfläche verwenden. Es stehen diverse derartige Programme zur Auswahl. Einfach zu bedienen ist der *Sound Converter* (siehe [Abbildung 7.24](#)).

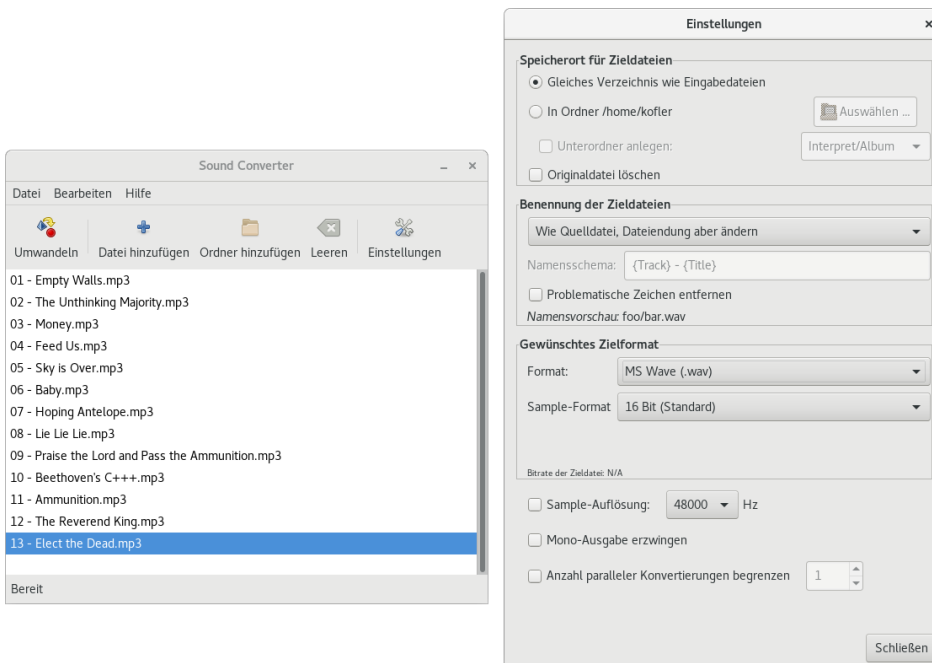


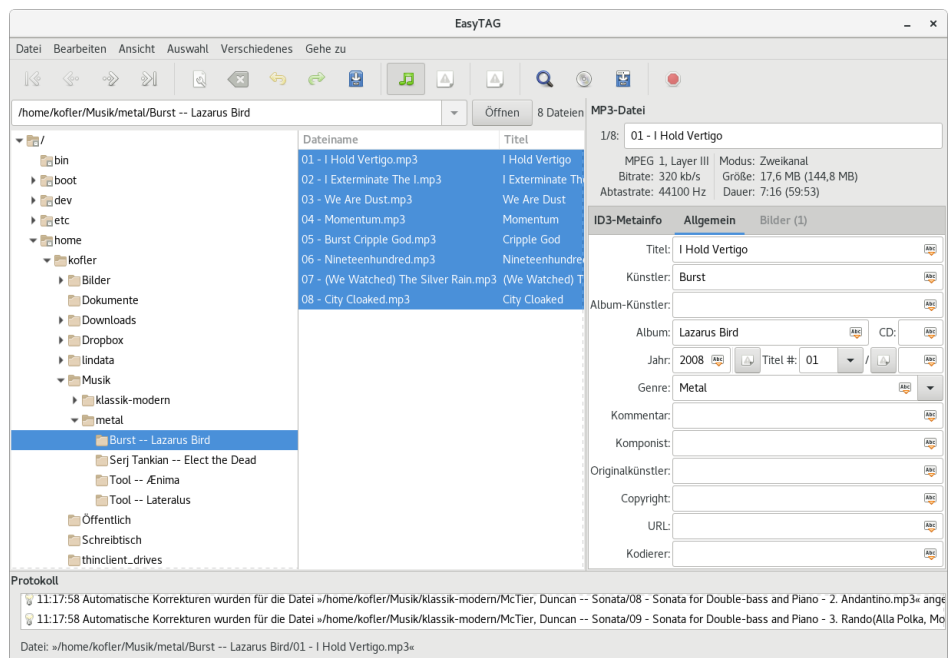
Abbildung 7.24 Das Programm »Sound Converter« mit seinem Einstellungsdialog

Standardmäßig wandelt das Programm die Ausgangsdateien in das außerhalb der Linux-Welt unübliche Ogg-Vorbis-Format um. Wenn Sie MP3-Dateien erzeugen möchten, müssen Sie vorher die Bibliothek `gststreamer-plugins-ugly` installieren, die den erforderlichen MP3-Encoder enthält.

**Alternativen** Anstelle von Sound Converter können Sie je nach Distribution auch eines der folgenden Programme ausprobieren: *curllew*, *FF Multi Converter*, *Format Junkie*, oder *soundKonverter* (KDE).

## EasyTAG

Wer selbst eine größere MP3-Sammlung pflegt, der weiß, dass die richtige Einstellung der MP3-Tags viel Zeit und Mühe macht: Die ID3-Daten von gekauften oder selbst gerippten MP3-Dateien entsprechen selten den eigenen Vorstellungen, Cover-Informationen fehlen etc. Es gibt unzählige Programme, die dabei helfen, ID3-Parameter effizient einzustellen bzw. zu verändern (suchen Sie in Ihrem Paketmanager nach *id3*). Persönlich ist mir das Programm EasyTAG am liebsten. Es erlaubt es, schnell alle MP3-Dateien eines Verzeichnisses gemeinsam zu bearbeiten.



**Abbildung 7.25** ID3-Tags neu einstellen

Die Bedienung des Programms ist allerdings gewöhnungsbedürftig. Nach dem Start wählen Sie das Verzeichnis aus, in dem sich die MP3-Dateien befinden. EasyTAG liest

nun alle MP3-Dateien in diesem Verzeichnis *und* in allen Unterverzeichnissen ein. Sie können dann eine einzelne MP3-Datei auswählen und deren ID3-Tags verändern (siehe [Abbildung 7.25](#)).

Zur Einstellung gemeinsamer Eigenschaften ist es effizienter, mehrere bzw. mit **[Strg]+[A]** alle MP3-Dateien des aktuellen Verzeichnisses zu markieren und dann das Album, den Komponisten etc. neu einzustellen. Aus Sicherheitsgründen müssen Sie nun jede Änderung durch einen Klick auf den winzigen Button rechts vom Einstellungsfeld bestätigen – andernfalls gelten die Änderungen nur für die gerade aktive Datei, nicht für alle markierten Dateien. Anfangs werden Sie diesen zusätzlichen Mausclick sicher hin und wieder vergessen.

Um in MP3-Dateien das Bild des CD-Covers zu speichern, markieren Sie alle betreffenden Dateien, wechseln in EasyTAG in das Dialogblatt BILDER, laden mit dem Plus-Button eine neue Bilddatei (JPEG oder PNG) und wählen das geladene Bild aus. EasyTAG sucht die Bilddatei standardmäßig im selben Verzeichnis, in dem sich auch die gerade bearbeiteten MP3-Dateien befinden. Vergessen Sie nun nicht, auf den winzigen Bestätigungs-Button zu klicken, damit das Bild in *allen* ausgewählten MP3-Dateien gespeichert wird! Anders als die meisten Audio-Player bietet EasyTAG leider keine Funktion, um nach Covern im Internet zu suchen – das müssen Sie selbst erledigen.

Cover-Bild  
einbetten

Beachten Sie, dass alle Änderungen an den MP3-Tags erst dann tatsächlich gespeichert werden, wenn Sie auf den SPEICHERN-Button in der Symbolleiste klicken. Im Einstellungsdialog können Sie angeben, in welcher ID3-Version die Tags geschrieben werden sollen, welcher Zeichensatz zur Anwendung kommen soll etc.

## Sound Juicer

Das Gnome-Programm Sound Juicer spielt Audio-CDs ab bzw. liest die Tracks der CDs aus und speichert sie als Dateien im MP3-, Ogg-Vorbis- oder in einem anderen Format. Voraussetzung für die Verwendung des Programms ist natürlich ein DVD- oder CD-Laufwerk, das auf Notebooks mittlerweile zur Rarität geworden ist. Zur Not können Sie sich aber auch mit einem externen Laufwerk behelfen.

Sound Juicer greift beim Erstellen der Audio-Dateien über das GStreamer-System auf externe Programme zurück, beispielsweise auf lame für MP3-Dateien, faac für AAC-Dateien (\*.m4a) etc. Standardmäßig erzeugt Sound Juicer Ogg-Dateien, deren Namen sich so zusammensetzen:

Musik/Gruppe/CD-Name/NN - Track-Titel.ogg

Die Verzeichnishierarchie und das gewünschte Audio-Format stellen Sie in den Einstellungen ein (siehe [Abbildung 7.26](#)).

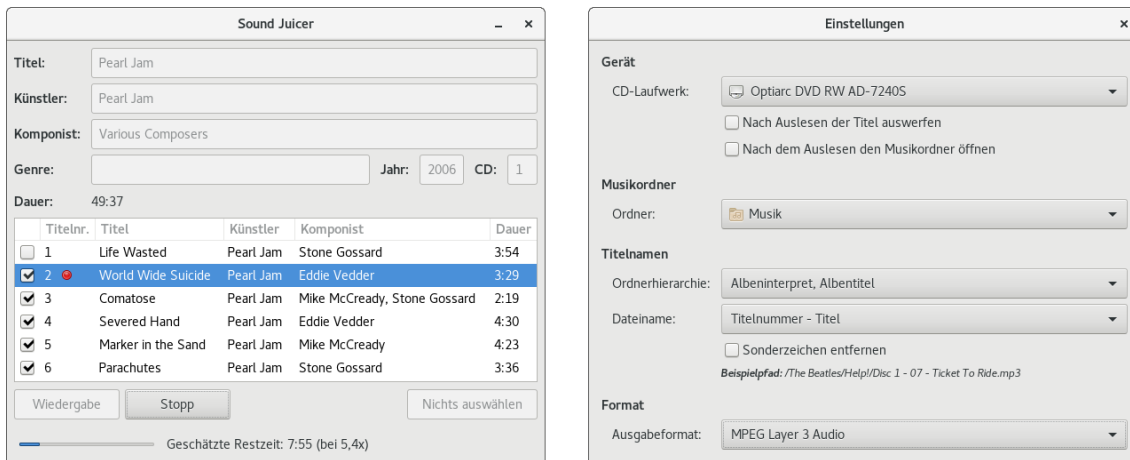


Abbildung 7.26 CD-Tracks mit Sound Juicer auslesen

## HandBrake

Mit dem Programm HandBrake (siehe [Abbildung 7.27](#)) können Sie DVDs rippen, also aus einer DVD eine Video-Datei erstellen. Dazu vorweg eine Leseempfehlung: *Brother Johns Encodingwissen* fasst fachlich fundiert und sprachlich unterhaltsam zusammen, was man wissen sollte, wenn man DVDs auslesen und daraus MPEG-4-Video-Dateien erzeugen möchte, also umgangssprachlich DVD-Ripping betreibt:

<https://encodingwissen.de>

### Hinweis

HandBrake liest DVDs aus. Das ist natürlich nur für DVDs zulässig, die keinen fremden Copyrights unterliegen – also z. B. für Ihre nicht verschlüsselte DVD mit einem Hochzeits- oder Kinder-Video (»Leos erste Schritte«). Keinesfalls dürfen Sie diese Werkzeuge verwenden, um irgendwelche Blockbuster zu kopieren oder der Video-Sammlung Ihres privaten Media-Centers hinzuzufügen. Welcher meiner Leser würde auf derart verwerfliche Ideen kommen? Lesen Sie lieber ein Buch!

**Grundlagen** Wenn Sie eine DVD unterwegs ansehen möchten, Ihr Notebook aber kein DVD-Laufwerk besitzt, übertragen Sie den Film am besten auf die Festplatte. Um Platz zu sparen, erzeugen Sie dabei eine neue Filmdatei, die die Video- und Audio-Daten enthält. Dieser Vorgang wird als DVD-Ripping bezeichnet.



Es gibt schier unendlich viele Varianten, wie das Ripping durchgeführt wird. Wichtige Parameter sind:

- ▶ der Audio-Codec (z. B. MP3, Ogg Vorbis, AAC, AC-3)
- ▶ der MPEG-4-Codec (z. B. DivX, H264, Ogg Theora, WebM, Xvid)
- ▶ das Container-Format (z. B. AVI, MKV, MOV, MP4, OGM)
- ▶ das Untertitel-Format (z. B. SRT, VobSub)
- ▶ Qualitätsfaktoren und Komprimierung

Bleibt noch die Qualitätsfrage: Wie groß darf die resultierende Datei maximal werden? Wenn Sie die Originalqualität einer DVD erhalten wollen, müssen Sie bis zu 1 GiB pro Stunde Filmlänge veranschlagen.

DVD-Ripping per Kommandozeile ist nur etwas für hartgesottene Linux-Anwender. Die Mühe lohnt nicht, zumal es Benutzeroberflächen gibt, die bei der Einstellung der vielen Parameter helfen.

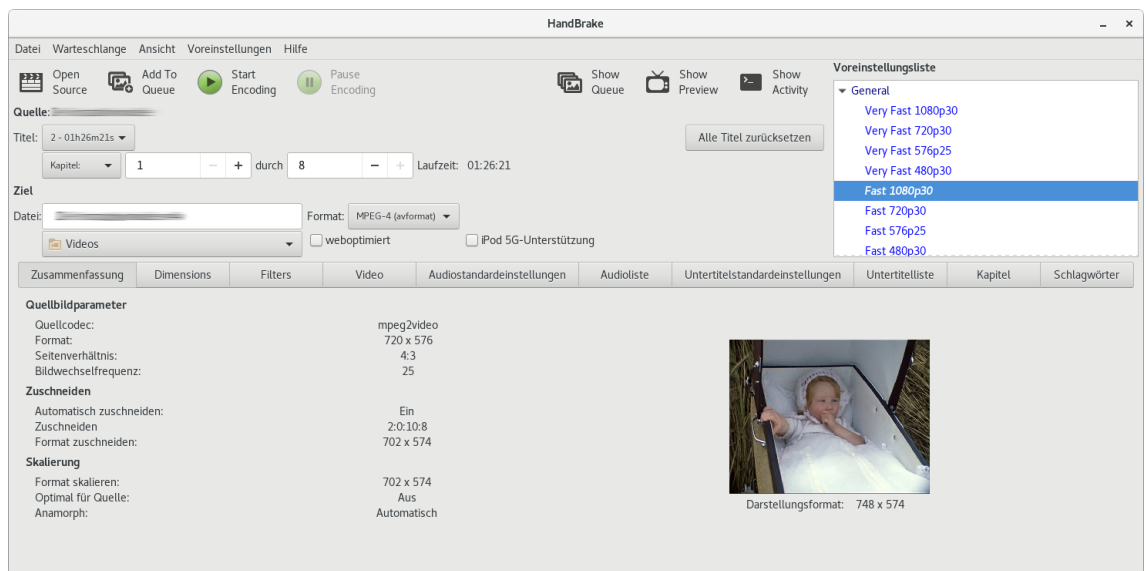


Abbildung 7.27 DVDs rippen mit HandBrake

Das Programm HandBrake (siehe [Abbildung 7.27](#)) war lange Zeit unter macOS-Anwendern bekannter als unter Linux. Das hat sich mittlerweile geändert, nicht zuletzt deswegen, weil viele andere DVD-Ripper für Linux nicht mehr gewartet werden.

**HandBrake**

Umso mehr kann HandBrake überzeugen: Das Programm stellt unter anderem einige vordefinierte Konvertierungsprofile zur Auswahl, die für das Abspielen der Filme auf verschiedenen Apple-Geräten optimiert sind. Beachten Sie bei der Installation,

dass Sie auch das Paket `handbrake-gui` mit der Benutzeroberfläche installieren. Bei manchen Distributionen enthält `handbrake` nur die Kommandozeilenversion des Programms.

Die Bedienung ist einfach: Sie geben eine Filmquelle (DVD, Film- oder ISO-Datei) an, legen fest, unter welchem Dateinamen der recodierte Film gespeichert werden soll, und wählen ein vordefiniertes Einstellungsprofil aus. Mit dem `START`-Button beginnen Sie dann die Recodierung.

## 7.16 Screenshots und Screencasts

Ein Screenshot ist ein Abbild des aktuellen Bildschirm- oder Fensterinhalts in einer Grafikdatei. Als »Screencast« bezeichnet man den Video-Mitschnitt des Bildschirm-inhalts. In der Regel kann auch ein Audio-Kanal mit aufgezeichnet werden. Screencasts eignen sich hervorragend, um Programmfunktionen zu dokumentieren. Die folgenden Abschnitte stellen kurz einige Programme vor, mit denen Sie Screenshots und Screencasts erstellen können.

**Wayland-Probleme** Aktuell sind nahezu alle Screenshot- und Screencast-Tools inkompatibel zu Wayland. Das hat damit zu tun, dass Wayland aus Sicherheitsgründen Programmen nicht erlaubt, den Bildschirminhalt anderer Programme auszulesen. Aktuell können nur in den Wayland Compositor integrierte Funktionen diese Schranke überwinden – also z. B. die Screenshot-Funktionen von Gnome.

### Screenshots erstellen

- Gnome** Unter Gnome erstellt `gnome-screenshot` Bildschirmabbildungen. Wie unter Windows führt `Druck` zu einem Screenshot des gesamten Bildschirms und `Alt+Druck` zu einer Abbildung des gerade aktiven Fensters. Die Bilder werden als PNG-Dateien im Verzeichnis `Bilder` gespeichert (Dateiname `Bildschirmfoto von yyyy-mm-dd hh-mm-ss.png`). Die Tastenkürzel zum Erzeugen von Screenshots können Sie im Modul `TASTATUR` der Systemeinstellungen festlegen.
- KDE** Um unter KDE Bildschirmabbildungen zu erstellen, starten Sie das Programm `ksnaps-hot` und stellen den gewünschten Aufnahmemodus ein (`VOLLBILD`, `FENSTER` oder `BEREICH`). Mit dem Button `NEUES BILDSCHIRMFOTO` erstellen Sie den Screenshot. Beim Speichern wird je nach Dateikennung automatisch das entsprechende Format verwendet.
- GIMP** Auch mit dem Bildverarbeitungsprogramm GIMP können Sie mit `DATEI • ERSTELLEN • SCREENSHOT` eine Bildschirmabbildung erstellen. Das ist praktisch, wenn Sie das Bild anschließend ohnedies mit GIMP weiterbearbeiten möchten.

Wenn Sie viele Screenshots erstellen, werden Sie vermutlich Shutter mögen: Dieses Programm bietet unzählige Zusatzfunktionen, um den gewünschten Bildausschnitt auszuwählen, das aufgenommene Bild mit Wasserzeichen zu versehen etc. Die Entwicklung von Shutter ist aber offensichtlich eingeschlafen: Die letzte Version stammt von Ende 2014.

Shutter

<http://shutter-project.org>

## Screencasts

Unter Linux herrscht kein Mangel an Programmen zur Aufnahme von Screencasts, ganz im Gegenteil: Es gibt zu viele! Die folgende Seite zählt gleich zehn verschiedene Programme bzw. Scripts auf:

<https://wiki.ubuntuusers.de/Screencasts>

Aber wenn man sich die Programme näher ansieht, bemerkt man rasch, dass die große Auswahl nicht wirklich weiterhilft. Viele Programme sind uralt und werden nicht mehr gewartet; die Grafikeffekte der Gnome Shell können nicht aufgezeichnet werden, in der resultierenden Video-Datei sind Bild und Ton nicht synchron etc. Kurzum, das Testen ist eine frustrierende Angelegenheit.

Bei aktuellen Gnome-Versionen ist eine Screencast-Funktion direkt in die Gnome Shell eingebaut.  $\boxed{\diamond} + \boxed{\text{Strg}} + \boxed{\text{Alt}} + \boxed{\text{R}}$  startet die Aufnahme. Wenn Sie diese Tastenkombination ein zweites Mal drücken, endet die Aufzeichnung. Die resultierende Datei im WebM-Format wird im Verzeichnis Videos gespeichert.

Screencasts unter Gnome

Leider gibt es zu dieser Screencast-Funktion keinerlei Konfigurationsmöglichkeiten. Standardmäßig wird kein Audio-Signal mitaufgezeichnet. Das lässt sich nur durch eine direkte Veränderung der `dconf`-Einstellungen für `org.gnome.shell.recorder` beheben. Details können Sie hier nachlesen:

<https://askubuntu.com/questions/112473>

Das bekannteste Screencasting-Programm ist recordMyDesktop. Es kann auf den meisten Distributionen installiert werden und wird direkt aus einem Terminalfenster heraus gestartet. Es gibt keine Benutzeroberfläche.

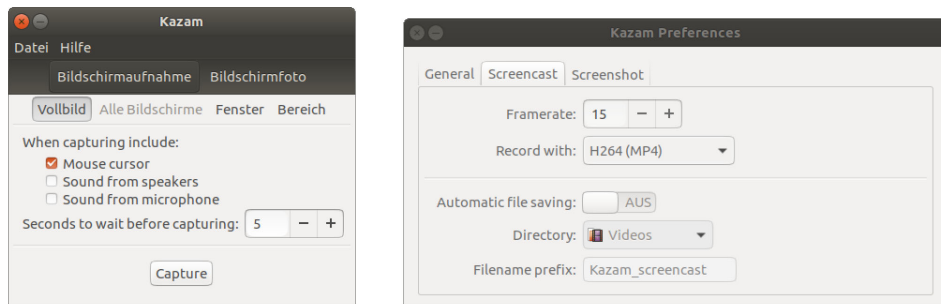
recordMyDesktop

An sich funktioniert das Programm zufriedenstellend. Der größte Mangel ist der Umstand, dass das Programm ausschließlich freie Formate unterstützt (Ogg-Container, Theora-Video-Codec, Vorbis-Audio-Codec). An sich ist das natürlich lobenswert, aber wenn die Aufnahme später in einem kommerziellen Umfeld weiterverarbeitet werden muss (sprich unter Windows oder macOS), macht man sich mit diesen Codecs wenig Freunde. Gegen recordMyDesktop spricht auch der Umstand, dass die

aktuelle Version aus dem Jahr 2008 stammt. Das Programm wird zwar anscheinend noch gewartet, aber nicht mehr weiterentwickelt.

**Kazam** (siehe [Abbildung 7.28](#)) ist im Vergleich zu recordMyDesktop ein wesentlich jüngerer Projekt, seine Entwicklung steht seit 2015 aber ebenfalls still. Kurz die wichtigsten Features:

- ▶ Kazam unterstützt die Aufnahmeformate H264 (MP4), VP8 (WebM), RAW (AVI), HUFFYAV (AVI) sowie Lossless JPEG (AV).
- ▶ Es können zwei Audio-Kanäle aufgezeichnet werden.
- ▶ Die Framerate ist frei einstellbar (standardmäßig 15 Bilder/Sekunde).
- ▶ Der aufzuzeichnende Bildschirmbereich ist frei einstellbar. Bei einer Dual-Screen-Konfiguration ist es möglich, nur den Inhalt eines Monitors aufzuzeichnen.



**Abbildung 7.28** Kazam samt Konfigurationsdialog

Leider steht das Programm momentan nur unter Ubuntu standardmäßig als Paket zur Verfügung. Wer mit anderen Distributionen arbeitet, muss sich mit einer manuellen Installation plagen.

# Kapitel 8

## Raspberry Pi

Der Raspberry Pi zählt seit mehreren Jahren zu den beliebtesten Minicomputern in der Maker- und Elektronikbastelszene. Die Grundfläche des Standardmodells ist etwas größer als eine Kreditkarte; in ein Gehäuse verpackt, hat der Computer das Volumen von zwei Smartphones. Das eigentliche Grundgerät kostet je nach Händler rund 40 EUR. Zusätzlich brauchen Sie in der Regel ein Netzteil, ein Gehäuse, eine SD-Speicherkarte und eventuell ein paar Kabel. Die Gesamtinvestition liegt also deutlich unter 100 EUR.

Dafür erhalten Sie einen vollwertigen, Linux-basierten Computer mit einer ARM-CPU, den Sie zur Steuerung elektrischer Geräte, für Versuchsaufbauten, als Mini-Server z. B. für den VPN-Zugang zu Ihrem Netzwerk zu Hause oder als kleines Multimedia-Center in der Art des Apple TV oder des Google Chromecast einsetzen können. Dieses Kapitel beschreibt, worauf Sie bei der Inbetriebnahme des Raspberry Pi achten müssen, gibt Konfigurationstipps und umreißt einige Anwendungsfälle.

Zur Inbetriebnahme des Raspberry Pi benötigen Sie einen »richtigen« Computer. Mit ihm beschreiben Sie die SD-Karte mit einer Linux-Distribution für den Raspberry Pi. Grundsätzlich können Sie das auch mit einem Windows- oder Apple-Computer tun, aber in diesem Buch nehme ich natürlich an, dass Sie unter Linux arbeiten.

### Hinweis

Es gibt viele Möglichkeiten, den Raspberry Pi zu nutzen: als Mini-PC, als Medien-Center, als Steuerungs- und Bastelplattform, als Server etc. Insofern ist die Einordnung dieses Kapitels in das Buch schwierig.

Generell arbeiten Sie mit dem Raspberry Pi oft deutlich systemnäher, als dies auf einem Notebook oder PC mit einer typischen Distribution der Fall ist. Linux-Einsteiger werden feststellen, dass dieses Kapitel teilweise Know-how voraussetzt, das ich erst in späteren Kapiteln vermitteln werde. Insofern müsste ich dieses Kapitel eigentlich in der Mitte oder beinahe am Ende des Buchs platzieren. Das erschien mir aber nicht wünschenswert.

## 8.1 Grundlagen

### Hardware

**Raspberry Pi 3, Modell B** Der Minicomputer Raspberry Pi besteht aus einer einzigen Platine in der Größe einer Kreditkarte (siehe [Abbildung 8.1](#)). Im Sommer 2017 trug das populärste der diversen Raspberry-Pi-Modelle die Bezeichnung »Raspberry Pi 3, Modell B«. Dieses Modell zeichnet sich durch die folgenden Eckdaten aus:

- ▶ ein Broadcom BCM2837 System-on-a-Chip (SoC), das aus vier 64-Bit-CPU-Cores in ARMv8-Architektur mit 1,2 GHz sowie einem Broadcom Video-Core IV mit H.264 Encoder/Decoder besteht
- ▶ 1 GiB RAM
- ▶ einen Micro-USB-Anschluss zur Stromversorgung
- ▶ vier USB-2-Anschlüsse für Tastatur, Maus und andere USB-Geräte
- ▶ einen HDMI-Ausgang für Bild und Ton, Auflösung bis zu 1920 × 1200 Pixel
- ▶ einen kombinierten Audio/Video-Ausgang für einen vierpoligen 3,5-mm-Klinkestecker
- ▶ einen Micro-SD-Karten-Slot (SDHC/SDXC)
- ▶ einen Ethernet-Anschluss (10/100 Mbit/s)
- ▶ eine GPIO-Steckerleiste mit 40 Pins für allgemeine Input/Output-Zwecke (General Purpose Input/Output inklusive UART, I<sup>2</sup>C-Bus, SPI-Bus, I<sup>2</sup>S-Audio)

Der Raspberry Pi weist damit ähnliche Eckdaten auf wie ein Mittelklasse-Smartphone. Natürlich fehlen die Telefonfunktionen und das Display, dafür bekommen Sie aber Netzwerk-, Monitor- sowie allgemeine I/O-Anschlüsse.

Vielleicht fragen Sie sich, warum der Raspberry Pi keinen Gigabit-Ethernet-Anschluss hat. Das liegt daran, dass der Ethernet-Adapter intern als USB-2-Device angebunden wird. USB 2 ist aber zu langsam, um die Datenmengen eines Gigabit-Netzwerks zu verarbeiten.

Generell ist das USB-System eine der größten Schwachstellen des Raspberry Pi: Alle vier USB-Anschlüsse und der Ethernet-Anschluss teilen sich nämlich *einen* USB-Kanal. Insofern ist der Raspberry Pi keine ideale Plattform für Eigenbau-NAS-Geräte oder Server, weil die Anbindung zur Außenwelt zu langsam ist – ganz egal, ob es sich um externe Datenträger oder um das Netzwerk handelt.

**Raspberry Pi Zero W** Eine interessante Alternative ist das Modell Raspberry Pi Zero W. Es ist nur halb so groß, verfügt nur über halb so viel Speicher und über eine etwas ältere CPU mit nur einem Core. Dafür begnügt sich die Miniausgabe des Raspberry Pi aber mit weniger als einem Watt Leistung – und bietet dennoch WLAN- und Bluetooth-Adapter.

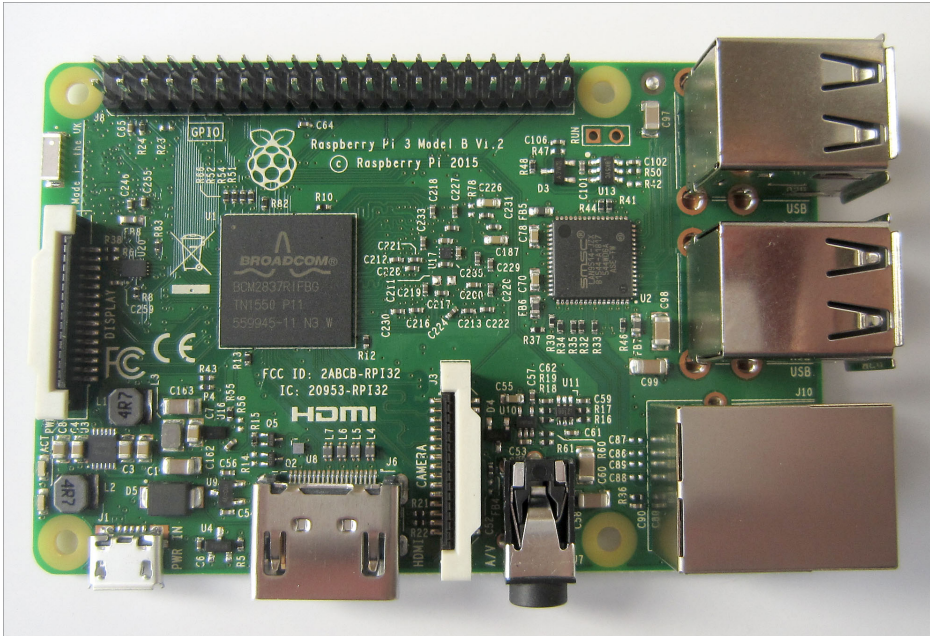


Abbildung 8.1 Der Raspberry Pi 3 (Modell B)

Das Zero-Modell ist damit ideal für winzige Bastelprojekte mit geringem Energieverbrauch geeignet.

Bevor Sie den Raspberry Pi erwerben, sollten Sie sich darüber klar werden, was Sie sonst noch alles brauchen: **Zubehör**

- ▶ ein Netzteil
- ▶ eventuell ein Gehäuse
- ▶ eine Micro-SD-Speicherkarte (16 GiB sind empfehlenswert, für manche Anwendungen reicht aber auch weniger Speicherplatz.)
- ▶ Tastatur, Maus, Kabel sowie einen Monitor oder Fernseher

### Das Netzteil ist entscheidend für die Stabilität

Achten Sie beim Kauf des Netzteils darauf, dass dieses ausreichend leistungsstark ist. Typische Handy-Netzteile haben zwar den richtigen Stecker, liefern aber zu wenig Leistung. Für den Raspberry Pi 3, Modell B, muss das Netzteil zumindest 5 Watt Leistung liefern können, damit der Raspberry Pi im Dauerbetrieb stabil läuft. Sie können über den Raspberry Pi auch USB-Geräte mit Strom versorgen – dann kann sich die Leistungsaufnahme für das Netzgerät auf über 10 Watt erhöhen.

Über ein HDMI-Kabel können Sie den Raspberry Pi an jeden gängigen Fernseher sowie an viele Computer-Monitore anschließen, sofern diese über einen HDMI-Eingang verfügen.

**Gehäuse** Falls Sie den Raspberry Pi in ein Gehäuse verpacken, sollten Sie darauf achten, dass es Belüftungsschlitze hat. Der Raspberry Pi läuft mangels Lüfter und anderer bewegter Teile vollkommen lautlos, produziert aber durchaus Abwärme. In einem Gehäuse ohne Luftzirkulation riskieren Sie ein vorzeitiges Ableben Ihres neuen Gadgets!

**Uhr** Der Raspberry Pi enthält keine integrierte Uhr. Die Uhrzeit muss deswegen nach jedem Start neu gestellt werden, idealerweise über eine Netzwerkverbindung mit NTP.

**Alternativen zum Raspberry Pi** Der Raspberry Pi ist zwar momentan der erfolgreichste, aber keineswegs der einzige Linux-taugliche Minicomputer. Schon seit vielen Jahren gibt es diverse Embedded-Linux-Systeme, die aber deutlich teurer und primär für den industriellen Einsatz gedacht sind. Eine Menge Informationen zu diesem Thema finden Sie auf der schon erwähnten Website <http://elinux.org>.

Daneben gibt es mittlerweile eine Menge Geräte, die sich wie der Raspberry Pi speziell an computer- und elektronikbegeisterte Bastler richten. Dazu zählen unter anderem der Banana Pi sowie diverse BeagleBoard- und Odroid-Geräte. Bei der Auswahl eines Geräts sollten Sie freilich nicht nur auf den Preis und die technischen Daten achten. Das momentan wichtigste Argument für den Raspberry Pi sind die exzellente Software-Unterstützung, die riesige Community und die unzähligen im Web verfügbaren Erweiterungen und Anleitungen.

**Weiterführende Informationen** Zum Raspberry Pi existieren mittlerweile unzählige Websites. Anbei einige Links zu den wichtigsten Seiten, auf denen Sie über dieses Kapitel hinausgehende Informationen finden:

<https://www.raspberrypi.org>

[http://elinux.org/RPi\\_Hub](http://elinux.org/RPi_Hub)

Natürlich will ich Ihnen hier nicht verschweigen, dass ich zusammen mit Christoph Scherbeck und Charly Kühnast ein 1000-seitiges Buch speziell für den Raspberry Pi verfasst habe. Infos und einen Blog zum Buch finden Sie hier:

<https://pi-buch.info>

### Software

Der Raspberry Pi enthält anfangs (fast) gar keine Software. Wenn Sie den Minirechner mit der Stromversorgung und einem Monitor verbinden, erhalten Sie nicht einmal ein Bild! Es gibt kein BIOS, EFI oder eine vergleichbare eingebaute Software, die verrät, ob der Raspberry Pi prinzipiell funktioniert.



Um den Raspberry Pi ausprobieren zu können, müssen Sie zuerst ein Betriebssystem auf eine SD-Karte schreiben. Wie Sie dabei im Detail vorgehen müssen, wird in [Abschnitt 8.2](#) beschrieben.

Geeignete Betriebssysteme für den Raspberry Pi müssen vor allem zwei Voraussetzungen erfüllen: Sie müssen für die ARM-CPU-Architektur kompiliert sein, und sie müssen in zwei getrennten Partitionen auf die SD-Karte geschrieben werden. Die erste Partition im FAT-Format enthält den Boot-Code, die Konfigurationsdatei `config.txt` und den Kernel; die zweite Partition enthält das eigentliche Betriebssystem.

Raspbian ist nicht nur die von den Raspberry-Pi-Entwicklern empfohlene Linux-Distribution, sondern auch die bei Weitem die populärste. Raspbian basiert aktuell auf Debian 8. Sein »Pixel-Desktop« basiert auf LXDE. Wenn es keine guten Gründe für eine andere Distribution gibt, sollten Sie für Ihre ersten Experimente unbedingt mit Raspbian arbeiten. Raspbian wird regelmäßig mit Updates versorgt und bietet eine solide Grundlage für jede erdenkliche Anwendung des Raspberry Pi.

Raspbian

Neben Raspbian gibt es eine ganze Palette anderer universeller Linux-Distributionen, die für den Raspberry Pi adaptiert wurden. Dazu zählen unter anderem Arch Linux und Ubuntu MATE. Eine eigene Gruppe von Distributionen ist speziell für den Multimedia-Einsatz optimiert: Dazu zählen LibreELEC, OSMC, RasPlex, Volumio und XBian. Fans alter Computer-Spiele werden an Lakka und RecalboxOS ihre Freude haben: Diese beiden Distributionen machen aus dem Raspberry Pi eine Retro-Spielkonsole, auf der Sie diverse alte Video-Spiele ausführen können. Dazu brauchen Sie außer einem Monitor lediglich einen USB-Joystick oder -Gamecontroller. Leider gibt es für diese Distributionen nur ganz wenige Spiele, die legal verfügbar sind.

Andere Linux-Distributionen

Obwohl in diesem Buch Linux im Vordergrund steht, sei nicht verheimlicht, dass auf dem Raspberry Pi auch andere Betriebssysteme laufen. In die Schlagzeilen gebracht hat es insbesondere Windows 10 IoT, wobei das Kürzel IoT für »Internet of Things« steht. Es handelt sich also nicht um eine Desktop-Version von Windows, sondern mehr um eine Windows-basierte Laufzeitumgebung, in der auf dem Raspberry Pi mit Visual Studio entwickelte Programme ausgeführt werden können.

Exoten

### Raspberry Pi versus Notebook/PC

Im Prinzip ist ein Raspberry Pi in Kombination mit einer geeigneten Linux-Distribution ein vollwertiger Computer mit grafischer Benutzeroberfläche, Webbrowser etc. Kann der Raspberry Pi also Ihren Desktop-PC ersetzen?

Er kann, aber Sie müssen bereit sein, beim Arbeitskomfort Abstriche zu machen. Die CPU des Raspberry Pi 3 kann nicht mit aktuellen CPUs von Intel oder AMD mithalten, der Zugriff auf SD-Karten oder USB-Sticks ist vergleichsweise langsam, und das eine

Gigabyte RAM ist für den intensiven Desktop-Einsatz zu wenig. Von der Geschwindigkeit abgesehen, machen auch so triviale Dinge wie etwa die fehlende eingebaute Uhr oder der nicht existente Ein/Aus-Schalter den Desktop-Betrieb des Raspberry Pi unpraktisch.

Seine Stärken spielt der Raspberry Pi eher als Steuerungsrechner oder Media-Center aus, wo er nur *eine* spezifische Aufgabe erledigt.

## 8.2 Raspbian installieren und konfigurieren

Raspbian ist die dominierende Distribution für den Raspberry Pi. Der oft vermittelte Eindruck, Raspbian sei einfach ein für die ARM-CPU kompiliertes Debian, täuscht: Raspbian ist in seiner Software-Defaultauswahl speziell für den Minirechner optimiert. Die Distribution enthält zudem eine Menge Raspberry-Pi-spezifischer Zusatzpakete, die z. B. zur Ansteuerung exotischer Hardware-Erweiterungen dienen.

Raspbian verwendet nicht den gewöhnlichen Debian-Kernel, sondern einen von der Raspberry Pi Foundation zur Verfügung gestellten Kernel. Außerdem werden zusammen mit Raspbian einige Programme mitgeliefert, die sonst nicht frei erhältlich sind, unter anderem eine Vollversion der Computer-Algebra-Software Mathematica.

**Versionen** Die im Frühjahr 2017 aktuelle Raspbian-Version basiert auf Debian 8 alias »Jessie«. Vermutlich wird es ein paar Monate nach der Fertigstellung von Debian 9 auch eine neue Raspbian-Version geben (»Raspbian Stretch«).

Raspbian liegt in zwei Varianten vor: Die Standardvariante ist mit einem Installationsumfang von rund 4 GiB sehr umfangreich und enthält eine komplette grafische Benutzeroberfläche. Alternativ dazu ist Raspbian Lite auf das Minimum reduziert und läuft nur im Textmodus. Raspbian Lite ist ideal für Server-Aufgaben sowie für Bastelprojekte ohne Bildschirm geeignet.

**Installationsvarianten** Die »Installation« einer Linux-Distribution auf einen Raspberry Pi sieht vollkommen anders aus als die Installation einer gewöhnlichen Linux-Distribution auf ein Notebook. Genau genommen installieren Sie nämlich gar nichts, sondern kopieren Dateien bzw. ein vorkonfiguriertes Image auf eine SD-Karte. Dazu benötigen Sie einen Computer mit einem SD-Slot oder mit einem externen SD-Kartenleser.

Für Raspbian gibt es zwei Installationsvarianten, die ich Ihnen im Folgenden beide näher erläutern werde:

- **NOOBS-Installation:** Bei der besonders einfachen NOOBS-Installation packen Sie eine ZIP-Datei auf Ihrem Linux-Rechner aus und kopieren sie auf eine zuvor formatierte SD-Karte.

- **Image-Installation:** Alternativ können Sie Raspbian auch als Image herunterladen und mit dem Kommando `dd` blockweise auf eine SD-Karte übertragen. Dieses Verfahren erfordert zwar ein wenig mehr Linux-Know-how, dafür können Sie die einmal erlernte Vorgehensweise später auch für diverse andere Raspberry-Pi-Distributionen anwenden, für die es die NOOBS-Variante nicht gibt.

### Die richtige SD-Karte

Aktuelle Raspberry-Pi-Modelle erwarten eine Micro-SD-Karte im Standardformat. Die SD-Karte muss dem SDHC-Standard entsprechen und darf offiziell maximal 32 GiB groß sein. Größere SD-Karten funktionieren aber ebenfalls, allerdings müssen Sie dann eine Image-Installation durchführen (nicht die NOOBS-Variante). Für erste Experimente mit Raspbian sind 16 GiB aber vollkommen ausreichend.

Wenn Sie Wert auf einen schnellen Start des Raspberry Pi legen bzw. häufig größere Datenmengen lesen oder schreiben möchten, sollten Sie eine möglichst schnelle SD-Karte verwenden (z. B. Class 10).

Nicht funktionierende SD-Karten zählen zu den häufigsten Fehlerursachen im Betrieb des Raspberry Pi. Vermeiden Sie Billigprodukte. Werfen Sie bei Problemen auch einen Blick auf die folgende Seite:

[http://elinux.org/RPi\\_SD\\_cards](http://elinux.org/RPi_SD_cards)

### NOOBS-Installation

Die Durchführung der NOOBS-Installation ist besonders einfach: Sie laden von der folgenden Webseite die NOOBS-ZIP-Datei herunter, packen das Archiv aus und kopieren dann alle Dateien des Archivs direkt auf die zuvor formatierte SD-Karte. Beachten Sie, dass die SD-Karte vorher VFAT-formatiert werden muss! Sie darf weder ein Linux-Dateisystem noch ein ExFAT-Dateisystem enthalten. Das Linux-Dateisystem wird erst eingerichtet, wenn Sie den Raspberry Pi mit der vorbereiteten SD-Karte starten.

<https://www.raspberrypi.org/downloads>

Die NOOBS-Installation ist besonders einfach durchzuführen, wenn Sie über einen Computer verfügen, der unter Windows oder macOS läuft. Dann empfiehlt es sich, zum Formatieren der SD-Karte das Formatierprogramm der *SD Association* einzusetzen, das Sie von der folgenden Seite kostenlos herunterladen können:

<https://www.sdcard.org/downloads>

Unter Linux bereitet das Formatieren hingegen oft Probleme, besonders dann, wenn die SD-Karte bereits für eine andere Raspberry-Pi-Installation verwendet wurde und daher Linux-Partitionen enthält.

**Device-Namen  
der SD-Karte  
feststellen**

Als Erstes müssen Sie den Device-Namen der SD-Karte feststellen – also den Namen einer speziellen Datei, die direkten Zugriff auf die SD-Karte gibt. Dazu führen Sie zuerst `lsblk` aus. Danach schieben Sie die SD-Karte in den SD-Slot, und anschließend wiederholen Sie `lsblk`:

```
user$ lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda   8:0    0 119,2G  0 disk
  sda1 8:1    0 143,1M  0 part /boot/efi
  sda2 8:2    0   2,8G  0 part [SWAP]
  sda3 8:3    0   93,1G  0 part /
... (SD-Karte einschieben)
user$ lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda   8:0    0 119,2G  0 disk
  sda1 8:1    0 143,1M  0 part /boot/efi
  sda2 8:2    0   2,8G  0 part [SWAP]
  sda3 8:3    0   93,1G  0 part /
sdb   8:16   1   7,4G  0 disk
  sdb1 8:17   1    56M  0 part /media/xxx
  sdb2 8:18   1   7,4G  0 part /media/yyy
```

Die SD-Karte wird also über die Device-Datei `/dev/sdb` angesprochen. Die meisten Linux-Distributionen binden beim Verbinden externer Datenträger automatisch alle dort gefundenen Partitionen ein. Es ist ganz wichtig, dass Sie alle Partitionen der SD-Karte mit `umount` aus dem Verzeichnisbaum lösen! Mit `lsblk` können Sie sich vergewissern, dass es für `/dev/sdb?` nun keinen Mountpoint mehr gibt.

```
root# umount /media/xxx
root# umount /media/yyy
```

**Verwenden Sie nicht »Sicher auswerfen«!**

Wenn Sie unter Linux mit einer grafischen Benutzeroberfläche arbeiten, liegt es nahe, anstelle von `umount /dev/sdb?` den Button SICHER AUSWERFEN oder SICHER ENTFERNEN Ihres Dateimanagers zu verwenden.

Damit wird die betreffende Partition aber nicht nur aus dem Verzeichnisbaum gelöst, manche Dateimanager deaktivieren die SD-Karte bei der Gelegenheit gleich ganz. Das gilt z. B. für aktuelle Versionen des Gnome-Dateimanagers Nautilus. Die SD-Karte kann dann überhaupt nicht mehr angesprochen werden, die Device-Datei `/dev/sdb` existiert nicht. Abhilfe: Entfernen Sie die SD-Karte aus dem Slot, fügen Sie sie neuerlich ein, und verwenden Sie dann `umount`!

**SD-Karte  
formatieren**

Danach richten Sie mit `parted` zuerst eine neue Partitionstabelle und dann eine neue Windows-Partition ein, die nahezu die ganze SD-Karte ausfüllt. `mkfs.vfat` richtet darin ein Windows-Dateisystem ein.

```

root# parted /dev/sdb mklabel msdos
root# parted /dev/sdb 'mkpart primary fat32 1MiB -1MiB'
root# mkfs.vfat -F 32 /dev/sdb
root# mkdir /media/sd-card
root# mount /dev/sdb /media/sd-card

```

Anstelle von `/dev/sdb` müssen Sie den Device-Namen Ihrer SD-Karte angeben. Je nachdem, wie viele Datenträger Ihr Linux-Rechner verwendet, kann das auch `/dev/sdc` oder `/dev/sdd` sein! Verwenden Sie auf keinen Fall `/dev/sda`! Das ist immer die primäre Festplatte oder SSD Ihres Computers, deren Inhalt Sie durch die folgenden Kommandos zerstören würden!

Auf die so vorbereitete SD-Karte kopieren Sie nun den Inhalt des NOOBS-ZIP-Archivs. Unter KDE oder Gnome verwenden Sie dazu am besten einen Dateimanager. Wenn Sie im Terminal arbeiten, packen Sie das ZIP-Archiv mit `unzip` aus und kopieren dann die Dateien. Stellen Sie sicher, dass die NOOBS-Dateien `recovery.*` direkt auf der SD-Karte gespeichert werden, nicht in einem Unterverzeichnis!

NOOBS-Dateien  
kopieren

```

user$ unzip NOOBS_n_n.zip
root# cp NOOBS_n_n/* /media/sd-card
root# umount /media/sd-card

```

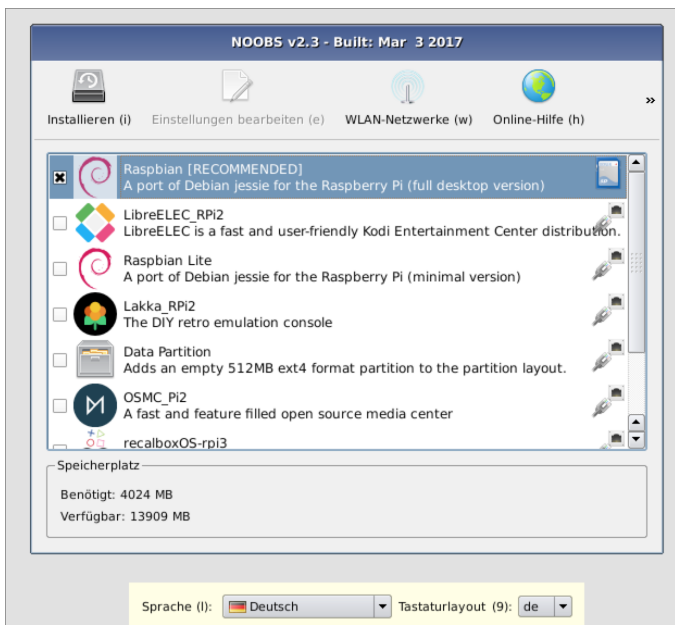


Abbildung 8.2 NOOBS-Installation

Nachdem Sie die SD-Karte aus dem Dateisystem gelöst haben, können Sie sie in den Slot des Raspberry Pi stecken und den Minicomputer starten. Wenn möglich, verbind-

den Sie den Raspberry Pi außerdem über ein Netzkabel mit einem Switch/Hub im lokalen Netzwerk. Wenige Sekunden nach dem Einschalten des Raspberry Pi erscheint das NOOBS-Fenster, in dem Sie die Sprache, das Tastaturlayout und das zu installierende Betriebssystem auswählen (siehe [Abbildung 8.2](#)).

In der Regel werden Sie im NOOBS-Menü nur den ersten Eintrag, RASPBIAN, auswählen. Grundsätzlich ist es auch möglich, mehrere Distributionen auf einmal zu installieren. In diesem Fall erscheint jedes Mal beim Start des Raspberry Pi ein Boot-Menü, in dem Sie das Betriebssystem, das Sie jeweils starten wollen, auswählen. Parallelinstallationen haben allerdings den Nachteil, dass sich alle Betriebssysteme den Platz auf der Festplatte teilen. Es ist nachträglich nicht ohne Weiteres möglich, ein Betriebssystem zu entfernen und den freien Platz einem anderen Betriebssystem zuzuweisen. Besser ist es in der Regel, für jede Distribution eine eigene SD-Karte zu verwenden.

Die Installation von Raspbian dauert einige Minuten. Anschließend erscheint auf dem Bildschirm die Nachricht OS(ES) INSTALLED SUCCESSFULLY. Sobald Sie diese Meldung mit OK bestätigen, wird der Raspberry Pi neu gestartet.

### Image-Installation

Für die zweite Raspbian-Installationsvariante laden Sie von der erwähnten NOOBS-Webseite nicht die NOOBS-ZIP-Datei herunter, sondern die Raspbian-Image-Datei. Das Image ist ebenfalls in ein ZIP-Archiv verpackt. Mit `unzip` packen Sie das Archiv aus:

```
user$ unzip nn-nn-raspbian.zip
Archive:  nn-nn-raspbian.zip
inflating: nn-nn-raspbian.img
```

Als Nächstes müssen Sie herausfinden, welchen Device-Namen die SD-Karte hat. Dieser Schritt erfolgt wie bei der NOOBS-Installation. Mit `umount` lösen Sie gegebenenfalls aktive Dateisysteme der SD-Karte aus dem Verzeichnisbaum.

#### Image-Datei kopieren

Der letzte Schritt besteht jetzt darin, die Image-Datei auf die SD-Karte zu kopieren. Das geht am einfachsten mit dem Kommando `dd`, das Sie mit `root`-Rechten ausführen müssen. Achten Sie darauf, dass Sie den richtigen Device-Namen angeben! Wenn Sie hier irrtümlich `/dev/sda` angeben, sind alle Daten auf Ihrer Festplatte bzw. SSD unwiderruflich zerstört!

```
root# dd if=nn-nn-raspbian.img of=/dev/sdb bs=4M
```

`dd` gibt leider kein optisches Feedback zum Kopiervorgang, der mehrere Minuten dauert. Mehr Komfort bietet in dieser Hinsicht das Kommando `dcfldd`, das Sie aber zumeist extra installieren müssen. Eine wirklich präzise Fortschrittsanzeige kann aber auch `dcfldd` nicht bieten, weil die Ergebnisse durch das I/O-Caching verfälscht werden: Zuerst scheint alles ganz schnell zu gehen, dann dauert es aber doch recht lange, bis der Vorgang abgeschlossen ist.

```
root# dcfldd if=nn-nn-raspbian.img of=/dev/sdb bs=4M \
      statusinterval=10
```

Um zu überprüfen, ob alles funktioniert hat, müssen Sie die SD-Karte entfernen und neu einfügen. Die Karte sollte nun zwei Partitionen enthalten: eine kleine FAT-Boot-Partition und eine rund 4 GiB große Systempartition, die beim ersten Start von Raspbian so vergrößert wird, dass sie die SD-Karte vollständig ausfüllt.

Wenn Ihnen das Kommando `dd` unheimlich ist, können Sie zum Beschreiben der SD-Karte auch die grafische Benutzeroberfläche Etcher (<https://etcher.io>) verwenden. Etcher

## Konfiguration

Beim ersten Start von Raspbian erscheint die grafische Benutzeroberfläche – vorerst in englischer Sprache und mit US-Tastaturlayout. Mit dem Programm RASPBERRY PI CONFIGURATION ändern Sie nun die Grundeinstellungen von Raspbian (siehe [Abbildung 8.3](#)). Dieser Abschnitt stellt die wichtigsten Parameter und Optionen kurz vor.

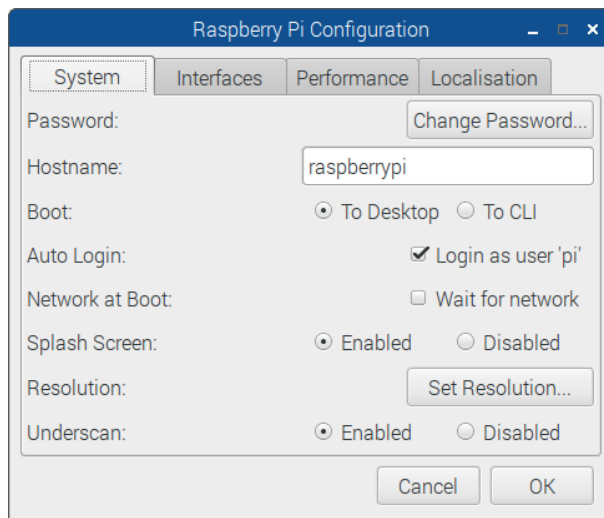


Abbildung 8.3 Konfiguration im Grafikmodus in Raspbian Jessie

- ▶ **PASSWORD:** Standardmäßig ist in Raspbian der Benutzer `pi` mit dem Passwort `raspberry` eingestellt. Dieses Passwort sollten Sie mit `CHANGE PASSWORD` unbedingt ändern!
- ▶ **BOOT TO DESKTOP / TO CLI:** Mithilfe der beiden Radiobuttons legen Sie fest, wie Raspbian gestartet werden soll. Standardmäßig wird die grafische Benutzeroberfläche gestartet, alternativ kann Raspbian auch im Textmodus hochgefahren werden (CLI steht für *Command Line Interface*).
- ▶ **UNDERSCAN:** Diese Option steuert, ob in das HDMI-Signal rund um den eigentlichen Bildschirminhalt ein schwarzer Rand eingebaut werden soll. Standardmäßig ist das der Fall, weil bei manchen Monitoren sonst kein stabiles Bild zustande kommt. Bei vielen Monitoren führt die Option aber dazu, dass dieser schwarze Rand tatsächlich sichtbar ist, wodurch sich die nutzbare Monitorfläche vermindert.
- ▶ **INTERFACES:** In diesem Dialog können Sie diverse Funktionen des Raspberry Pi ein- bzw. ausschalten. Dazu zählen die Kamera, verschiedene Bussysteme (I<sup>2</sup>S, SPI) und der SSH-Dienst.

Hinter den Kulissen werden die Bussysteme durch Veränderungen in `/boot/config.txt` aktiviert bzw. deaktiviert. Damit das Kamera-Erweiterungsmodul genutzt werden kann, fügt das Konfigurationsprogramm die beiden Anweisungen `start_x=1` und `gpu_mem=128` in `config.txt` ein.

- ▶ **PERFORMANCE:** Auf dem dritten Dialogblatt können Sie die Taktfrequenz älterer Raspberry-Pi-Modelle erhöhen. Bei aktuellen Modellen wird eine Erhöhung des Takts hingegen nicht empfohlen, im Konfigurationsprogramm fehlen dann die entsprechenden Optionen.  
Gleichzeitig können Sie hier einstellen, welcher Bereich des RAMs für das Grafiksystem reserviert werden soll. Für die meisten Anwendungen sind die standardmäßig vorgesehenen 64 MiB mehr als ausreichend. Wenn Sie aber Multimedia-Programme ausführen oder das Kameramodul verwenden möchten, sind zumindest 128 MiB erforderlich.
- ▶ **LOCALISATION:** Das vierte Dialogblatt ist anfänglich das wichtigste. Hier stellen Sie die Sprache, das Tastaturlayout und die Zeitzone ein.

Manche Einstellungen werden erst nach einem Neustart wirksam.

**raspi-config** Das »alte« Konfigurationsprogramm `raspi-config` steht Ihnen weiterhin zur Verfügung (siehe [Abbildung 8.4](#)). Es kann in Textkonsolen bzw. im Terminal ausgeführt werden. `raspi-config` ist gut geeignet, um die Konfiguration von Raspbian Lite durchzuführen.



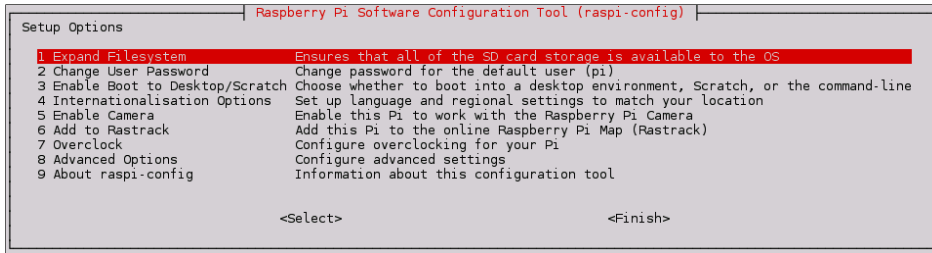


Abbildung 8.4 Das traditionelle Konfigurationsprogramm raspi-config

## Erste Schritte

Nach dem Start erscheint ohne Login der Pixel-Desktop auf der Basis von LXDE (*Light-weight X11 Desktop Environment*). Wenn Sie sich aber ab- und neu anmelden, müssen Sie den Login-Namen `pi` auswählen und Ihr Passwort angeben. Wenn Sie das Passwort nicht im Rahmen der Erstkonfiguration verändert haben, lautet es `raspberrypi`.

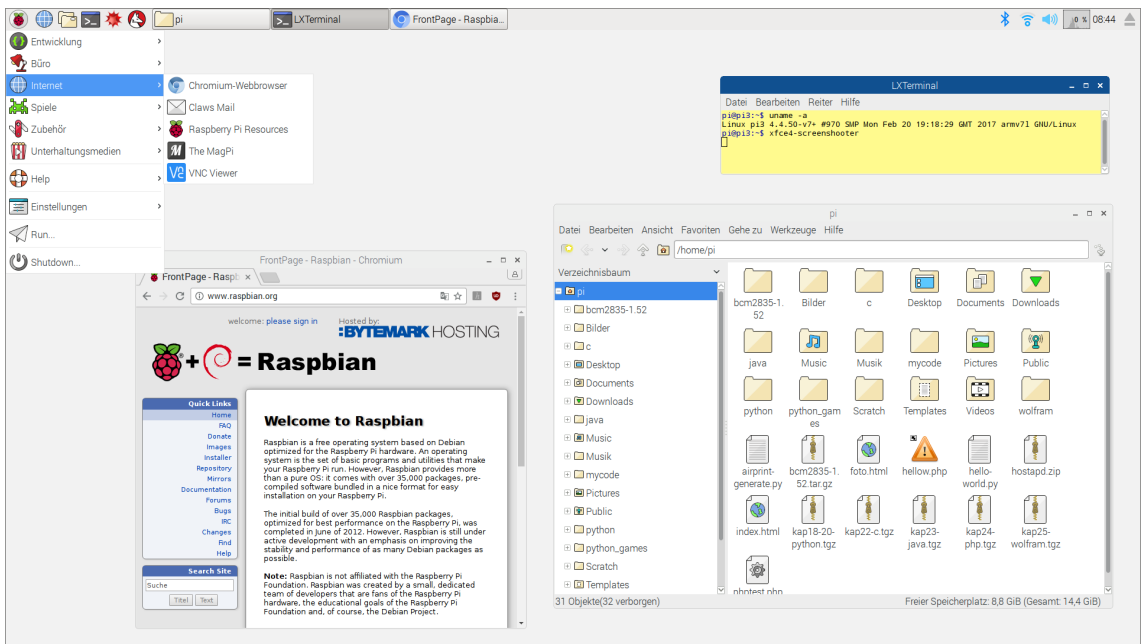


Abbildung 8.5 Der Pixel-Desktop auf der Basis von LXDE

Wenn Sie den Desktop durch eine Login-Box absichern möchten, deaktivieren Sie den Auto-Login, indem Sie in `/etc/lightdm.conf` der Zeile `autologin-user=pi` ein Kommentarzeichen voranstellen. Gleichzeitig sollten Sie auch die Einstellung für `greeter-hide-users` von `true` auf `false` stellen:

```
/etc/lightdm/lightdm.conf
...
greeter-hide-users=false
# autologin-user=pi
```

- sudo** Administratorarbeiten führen Sie mit `sudo` aus (siehe auch [Abschnitt 12.3](#), »Prozesse unter einer anderen Identität ausführen (sudo)«). Der Benutzer `pi` darf `sudo` ohne Passwort benutzen. Wenn Sie `sudo`-Aktivitäten durch ein Passwort absichern möchten, stellen Sie mit einem Editor der letzten Zeile von `/etc/sudoers` ein Kommentarzeichen `#` voran. Wegen der weiter oben enthaltenen Regel für die Gruppe `sudo` kann `pi` weiterhin `sudo` nutzen, allerdings muss nun nochmals das eigene Passwort angegeben werden. (Um die Änderungen durchzuführen, müssen Sie zuerst `sudo -s` ausführen und `/etc/sudoers` dann in einem Editor bearbeiten.)

```
# /etc/sudoers
...
%sudo  ALL=(ALL:ALL) ALL
...
# pi ALL=(ALL) NOPASSWD: ALL
```

#### Ein- und ausschalten

Der Raspberry Pi ist für den Dauerbetrieb gedacht – es gibt keinen Schalter zum Ein- und Ausschalten. Um das Gerät sicher auszuschalten, müssen Sie es zuerst herunterfahren. Dazu führen Sie im Startmenü `SHUTDOWN` oder im Textmodus `sudo halt` aus. Sobald das Bild auf dem Bildschirm verschwindet und keine der LEDs des Geräts mehr blinkt, können Sie das Kabel zur Stromversorgung lösen. Sobald Sie das Gerät wieder anstecken, wird es automatisch neu gestartet.

#### Achtung

Vermeiden Sie es, einfach im laufenden Betrieb die Stromversorgung zu trennen! Ihr Raspberry Pi kann dann das Dateisystem nicht ordentlich herunterfahren. Normalerweise passiert nichts, aber Sie riskieren nicht nur einzelne defekte Dateien, sondern sogar ein inkonsistentes Dateisystem auf der SD-Karte. Im schlimmsten Fall müssen Sie Raspbian oder eine andere Linux-Distribution neu auf die SD-Karte schreiben und verlieren alle Ihre Daten!

- Update** Sobald Sie eine Netzwerkverbindung hergestellt haben, sollten Sie ein Update von Raspbian durchführen:

```
user$ sudo apt update
user$ sudo apt full-upgrade
```

Das Update ist ein guter Stabilitätstest für Ihr frisch installiertes System. Es beansprucht alle Komponenten des Computers. Wenn das Update fehlerfrei abgeschlossen wird, können Sie zuversichtlich sein, dass Ihr Minicomputer stabil läuft.

Raspbian stellt nur sehr selten offizielle Kernel-Updates zur Verfügung. Parallel zum offiziellen Kernel gibt es aber einen inoffiziellen Test-Kernel, den Sie bei Bedarf mit dem Kommando `rpi-update` installieren können. Das ist nur dann zweckmäßig, wenn Sie neu in den Kernel eingeflossene Hardware-Treiber nutzen möchten.

Kernel-Update

Die Kernel- und Firmware-Dateien werden in der ersten Partition der SD-Karte gespeichert. Unter Raspbian können Sie den Inhalt dieser Partition im Verzeichnis `/boot` ansehen. Der Kernel befindet sich in der Datei `kernel*.img`, die Firmware in den Dateien `*start.elf`. Die Dateien werden unmittelbar nach dem Einschalten geladen.

Das eigentliche Update ist mit dem Update-Tool `rpi-update` schnell erledigt: Sie führen das Kommando aus, bestätigen die Rückfragen und starten Ihren Minicomputer schließlich neu:

```
pi$ sudo rpi-update
pi$ sudo reboot
```

Sollten Sie je in die Verlegenheit kommen, dass Sie ein Kernel-Update rückgängig machen möchten, besteht der einfachste Weg darin, dass Sie auf der folgenden Webseite nach einer älteren Kernel-Version suchen:

<https://github.com/Hexxeh/rpi-firmware/commits/master>

Klicken Sie die gewünschte Version an. Sie gelangen so auf eine Detailseite, die den eine lange hexadezimale Zahl als Commit-Code enthält. Diesen Code übergeben Sie an `rpi-update`:

```
pi$ rpi-update 52241088c1da59a359110d39c1875cda56496764
```

In Raspbian sind der grafische Editor Leafpad sowie eine Auswahl von Editoren für das Terminal installiert, unter anderem `vi` und `nano`. Auch hier gilt: Den Lieblingseditor Ihrer Wahl installieren Sie einfach mit `apt`.

Editor

Nach zehn Minuten ohne Tastatur- und Mausaktivität aktiviert sich der Bildschirmschoner. Damit wird der Bildschirminhalt zwar schwarz, der Monitor läuft aber weiter. In dieser Form ist der Bildschirmschoner unbrauchbar. Damit der Bildschirmschoner richtig funktioniert, müssen Sie am Ende der Datei `/boot/config.txt` eine Zeile hinzufügen:

Bildschirmschoner und Energiesparfunktionen

```
# am Ende von /boot/config.txt
...
hdm_i_b_lanking=1
```

Wenn Sie umgekehrt den Bildschirmschoner ganz deaktivieren möchten, besteht die einfachste Lösung darin, die Systemkonfigurationsdatei `/etc/lightdm/lightdm.conf` zu ändern. Dazu laden Sie die Datei in einen Editor, suchen nach dem Abschnitt, der mit `[SeatDefaults]` eingeleitet wird, und fügen dort die Zeile `xserver-command=...` ein:

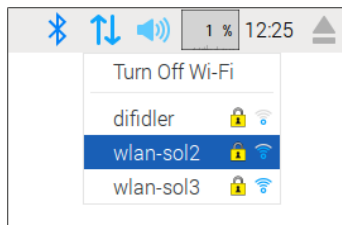
```
# in der Datei /etc/lightdm/lightdm.conf
...
[SeatDefaults]
xserver-command=X -s 0 -dpms
```

Die Einstellung bewirkt, dass der Grafik-Server X, der für die Darstellung der grafischen Benutzeroberfläche verantwortlich ist, in Zukunft gar keine Energiesparfunktionen verwendet.

## WLAN-Verbindung herstellen

Aktuelle Raspberry-Pi-Modelle verfügen über einen internen WLAN-Adapter. Bei älteren Modellen müssen Sie einen USB-WLAN-Adapter einsetzen. Vor dem Kauf eines USB-WLAN-Steckers sollten Sie ein wenig recherchieren, damit Sie ein Modell erhalten, das von Raspbian auf Anhieb unterstützt wird. Das trifft z. B. für die weitverbreiteten USB-Stecker mit RTL81xx-Chips zu (z. B. EDIMAX-7811).

Die eigentliche WLAN-Konfiguration ist denkbar einfach: Ein Klick auf das WLAN-Icon im Panel listet alle verfügbaren Funknetze auf (siehe [Abbildung 8.6](#)). Nach der Auswahl eines Funknetzes muss nur noch das dazugehörige Passwort angegeben werden – fertig!



**Abbildung 8.6** Das WLAN-Menü im Panel von Raspbian


Hinter den Kulissen ist für den WLAN-Verbindungsaufbau das Programm `wpa_supplicant` zuständig. Die Verbindungsparameter werden in der Datei `/etc/wpa_supplicant/wpa_supplicant.conf` gespeichert, inklusive der Passwörter im Klartext.

### Den Raspberry Pi als WLAN-Access-Point einsetzen

Wenn Sie Ihren Raspberry Pi als Access-Point verwenden möchten, brauchen Sie einen WLAN-Adapter, der den Access-Point-Modus unterstützt. Das ist nicht bei allen Modellen der Fall! Außerdem müssen Sie das Paket `hostapd` installieren und konfigurieren. Eine Anleitung zum Umgang mit diesem Programm finden Sie in [Kapitel 27](#), »Internet-Gateway«.

## Bluetooth

Mit der Bluetooth-Unterstützung sieht es ähnlich aus wie mit dem WLAN: Nur aktuelle Modelle enthalten einen Bluetooth-Adapter. Bei älteren Raspberry Pis müssen Sie für wenige Euro einen USB-Bluetooth-Stecker erwerben. Die meisten Modelle funktionieren auf Anhieb – recherchieren Sie dennoch *vor* dem Kauf!

Raspbian verfügt über eigene Bluetooth-Konfigurationswerkzeuge. Um ein neues Bluetooth-Gerät einzurichten, klicken Sie auf das Bluetooth-Icon im Panel und führen ADD DEVICE aus. Der Dialog zeigt nach einiger Zeit alle in Funkreichweite befindlichen Bluetooth-Geräte an (natürlich nur, soweit diese nicht mit anderen Computern verbunden sind). Mit dem Button PAIR versucht das Konfigurationsprogramm nun, eine Verbindung zum Gerät herzustellen. Bei manchen Geräten (z. B. Tastaturen) müssen Sie dabei einen Pairing-Code sowie  eingeben.

Einmal gekoppelte Geräte sollte der Raspberry Pi auch nach dem nächsten Neustart korrekt wiedererkennen. Bei unseren Tests dauerte es aber oft mehrere Sekunden, bis diverse Bluetooth-Geräte auf neue Eingaben reagierten. So naheliegend die Verwendung von Bluetooth beim Raspberry Pi ist – in unserer Praxis haben sich USB-Tastaturen und -Mäuse als wesentlich zuverlässiger erwiesen.

## 8.3 Kodi und LibreELEC

Zu den beliebtesten Anwendungen des Raspberry Pi zählt der Einsatz als Multimedia-Center. Dabei kommt in der Regel das Programm Kodi (ehemals XBMC) zum Einsatz. Das ist eine Sammlung von Open-Source-Programmen zur Verwendung eines Linux-Computers als Multimedia-Center. Je nach Hardware-Voraussetzungen können Sie mit Kodi Folgendes machen:

- ▶ auf lokalen Datenträgern oder im lokalen Netzwerk verfügbare Audio- und Video-Dateien abspielen (SMB, NFS, DNLS)
- ▶ CDs und DVDs abspielen (erfordert ein externes Laufwerk)
- ▶ Live fernsehen (erfordert eine Karte zum TV-Empfang)
- ▶ TV-Sendungen aufzeichnen
- ▶ YouTube, diverse Mediatheken, Internet-Radios und -Fernsehstationen nutzen (erfordert die Installation von Add-ons)
- ▶ Fotos ansehen
- ▶ den Wetterbericht lesen

Das Abspielen von CDs/DVDs sowie der Empfang traditioneller Fernsehprogramme ist auf einem Raspberry Pi aufgrund der fehlenden Hardware-Voraussetzungen nicht

möglich. Üblicherweise wird ein Raspberry Pi primär zum Abspielen von Audio- und Video-Dateien oder Streams verwendet, die im Internet oder im lokalen Netzwerk zur Verfügung stehen. Ideal lässt sich Kodi in Kombination mit einem NAS-Gerät nutzen, das die Foto-, Audio- und Video-Sammlung des Haushalts enthält. Die Bedienung des Geräts kann wie bei einem Computer mit Tastatur und Maus, aber auch durch eine IR-Fernbedienung oder durch ein Smartphone erfolgen.

Ich konzentriere mich in diesem Abschnitt auf die Raspberry-Pi-spezifischen Besonderheiten von Kodi. Auf die eigentliche Bedienung von Kodi gehe ich hingegen nur kurz ein – die lässt sich rasch durch Ausprobieren erlernen. Außerdem gibt es ein umfassendes, als Wiki organisiertes Handbuch:

*<http://kodi.wiki>*

#### Kodi-Distributionen für den Raspberry Pi

Theoretisch wäre es möglich, ein vorhandenes Raspbian-System durch die Installation diverser Pakete Kodi-tauglich zu machen. Das ist wegen des damit verbundenen Overheads aber unüblich. Stattdessen gibt es zurzeit gleich vier dezidierte Kodi-Distributionen, die ausschließlich die für den Kodi-Betrieb erforderlichen Programme enthalten:

- ▶ OpenELEC: *<http://openelec.tv>*
- ▶ LibreELEC (Fork von OpenELEC): *<https://libreelec.tv>*
- ▶ OSMX (ehemals Raspbmc): *<https://osmc.tv>*
- ▶ XBian: *<http://xbian.org>*

In den vergangenen Jahren war OpenELEC die populärste Kodi-Distribution für den Raspberry Pi. Allerdings kam es im Frühjahr 2016 zu einem Streit innerhalb der Entwicklergemeinde, der zu einer Spaltung des Projekts führte. Mittlerweile hat LibreELEC die Nachfolge von OpenELEC übernommen. OpenELEC existiert zwar auch noch, hinkt aber mit neuen Versionen und Updates meilenweit hinter LibreELEC her. Dieser Abschnitt bezieht sich auf LibreELEC 8 mit Kodi 17.1.

#### Rasplex

Als wären vier Kodi-Distributionen nicht genug, gibt es mit Rasplex eine weitere Multimedia-Distribution für den Raspberry Pi. Rasplex basiert allerdings nicht auf Kodi, sondern auf dem Plex-Projekt. Die Zielsetzung von Plex ist ähnlich wie bei Kodi, der Ansatz aber ein anderer: Plex trennt Client- und Server-Funktionen. Damit Sie Rasplex nutzen können, benötigen Sie in Ihrem Haushalt einen Plex-Server. Diese Rolle kann z. B. ein NAS-Gerät übernehmen. Plex erfordert anfänglich mehr Aufwand für die Konfiguration, hat dafür aber den Vorteil, dass später unkompliziert weitere Clients hinzugefügt werden können.

*<http://rasplex.com>*

## LibreELEC installieren und konfigurieren

LibreELEC (*Libre Embedded Linux Entertainment Center*) ist keine abgespeckte Raspbian-Variante, sondern eine vollkommen eigenständige Distribution – und zwar eine mit durchaus bemerkenswerten Eigenschaften: So gibt es in LibreELEC keine Paketverwaltung. Stattdessen befindet sich die gesamte LibreELEC-Distribution in einer nur rund 130 MiB großen komprimierten Datei `/flash/SYSTEM`. Diese Datei wird im Read-only-Modus über die Device-Datei `/dev/loop0` als squashfs-Dateisystem genutzt.

LibreELEC agiert als Samba-Server und ist somit für Linux-, Windows- und macOS-Rechner im lokalen Netzwerk sichtbar. Das macht die Übertragung von Video-Dateien auf die SD-Karte des Raspberry Pi besonders einfach.

Um LibreELEC zu installieren, laden Sie auf Ihrem regulären Computer das aktuelle Disk-Image herunter. Achten Sie darauf, dass Sie die richtige Version herunterladen! LibreELEC unterstützt diverse Hardware-Plattformen. Selbst für den Raspberry Pi gibt es zwei Images: eines für alte Raspberry Pis (Version 1) und ein zweites für neue Modelle mit ARMv7- oder ARMv8-CPU. Die Downloads sind hier zu finden:

Installation auf  
eine SD-Karte

<https://libreelec.tv/downloads>

Die komprimierte Datei hat die Endung `.img.gz`. Sie müssen die Datei nun zuerst dekomprimieren und dann auf eine SD-Karte übertragen. Details lesen Sie bitte in [Abschnitt 8.2, »Raspbian installieren und konfigurieren«](#), nach.

```
user$ gunzip libreelec-n.n.img.gz
root# dd if=libreelec-n.n.img of=/dev/xxx bs=16M
```

Sofern Sie die abzuspielenden Filme aus dem Internet oder von einem NAS-Gerät beziehen, reicht eine SD-Karte mit einer Größe von nur einem Gigabyte für den Betrieb von LibreELEC vollkommen aus. Nur wenn Sie vorhaben, auf der SD-Karte Audio- oder Video-Dateien zu speichern, sollten Sie eine möglichst große SD-Karte wählen.

Mit der SD-Karte starten Sie nun den Raspberry Pi. Dort werden zuerst die Partitionen auf der SD-Karte an deren Größe angepasst. Nach circa einer halben Minute erscheint das Programm *Welcome to LibreELEC*, das bei der Erstkonfiguration hilft. Im ersten Schritt können Sie den Hostnamen verändern. Standardmäßig kommt der Name `libreelec` zum Einsatz. Sofern Ihr Minicomputer über einen WLAN-Adapter verfügt, zeigt der nächste Konfigurationsdialog die in Reichweite befindlichen Funknetzwerke. Sie können nun eines davon auswählen und das dazugehörige Passwort angeben. Die WLAN-Konfiguration kann aber selbstverständlich auch später erfolgen.

LibreELEC-  
Erstkonfiguration

Im nächsten Dialog geht es darum, welche Netzwerkdienste unter LibreELEC standardmäßig laufen. Zur Auswahl stehen SSH und Samba, wobei SSH normalerweise

nicht aktiv ist, Samba schon. Dazu ein paar Hintergrundinformationen: Bis zum Abschluss der Konfiguration ist es empfehlenswert, beide Dienste zu aktivieren. Wenn Sie die Dienste später nicht mehr benötigen, können Sie sie im Programm über **OPTIONEN • LIBREELEC** unkompliziert abschalten.

**Sprache, Tastatur und Zeitzone** Anfänglich zeigt Kodi alle Menüs in englischer Sprache an, und für die Tastatur gilt das US-Tastaturlayout. Um die deutsche Sprache zu aktivieren, klicken Sie das Zahnrad-Icon an, das zu den Systemeinstellungen führt. Die Einstellungen für Sprache und Zeitzone finden Sie im Modul **INTERFACE SETTINGS** bzw. im Modul **BENUTZEROBERFLÄCHE**, sobald Sie die Sprache verändert haben.

Dieses Modul sieht auch die Möglichkeit vor, das Tastaturlayout zu verändern – bei meinen Tests hat das aber nicht funktioniert. Wechseln Sie daher in das Einstellungsmodul **LIBREELEC**: Dessen Dialogblatt **SYSTEM** bietet ebenfalls die Möglichkeit, das Tastaturlayout zu verändern – und dort funktioniert es auch.

**Audio-Ausgang wählen** Standardmäßig erfolgt die Audio-Ausgabe über das HDMI-Kabel. Falls Sie zur Wiedergabe einen Computer-Monitor ohne Audio-Funktionen verwenden, können Sie die Tonausgabe auch über den Analog-Audio-Ausgang des Raspberry Pi leiten. Dazu öffnen Sie das Dialogblatt **AUDIO** des Einstellungsmoduls **SYSTEM** und verändern dort das Audio-Ausgabegerät. In diesem Konfigurationsdialog können Sie auch gleich **KLANGSCHEMA WIEDERGEHEN** auf **NIE** stellen. Damit setzen Sie den lästigen Bling-Tönen ein Ende, die bei allen möglichen Aktionen erklingen.

## Netzwerkkonfiguration

**Netzwerk- und WLAN-Konfiguration** Wenn Ihr Raspberry Pi über ein Netzwerkkabel mit dem lokalen Netzwerk verbunden ist, erübrigt sich die Netzwerkkonfiguration: Wie Raspbian bezieht auch LibreELEC die IP-Adresse und andere Parameter vom DHCP-Server des LANs.

Beinahe ebenso unkompliziert erfolgt die WLAN-Konfiguration. Im Einstellungsmodul **LIBREELEC** im Dialogblatt **VERBINDUNGEN**. Dort werden alle in Funkreichweite befindlichen Netze aufgelistet (siehe [Abbildung 8.7](#)). Sobald Sie ein Netzwerk auswählen, können Sie das WLAN-Passwort eingeben. Die Netzwerkeinstellungen werden hier gespeichert:

```
/storage/.cache/connman/wifi_xxx_managed_psk/settings
```

**SSH nutzen** Manche Administrationsarbeiten können Sie nur durchführen, indem Sie sich via SSH von einem anderen Computer aus in das LibreELEC-System einloggen. Voraussetzung dafür ist natürlich, dass unter LibreELEC der SSH-Server läuft. Falls Sie die entsprechende Option während der Konfiguration nicht aktiviert haben, können Sie dies jetzt nachholen. Die entsprechende Option finden Sie im Einstellungsmodul **LIBREELEC** im Dialogblatt **DIENSTE**. Als Login-Passwort ist `libreelec` voreingestellt.



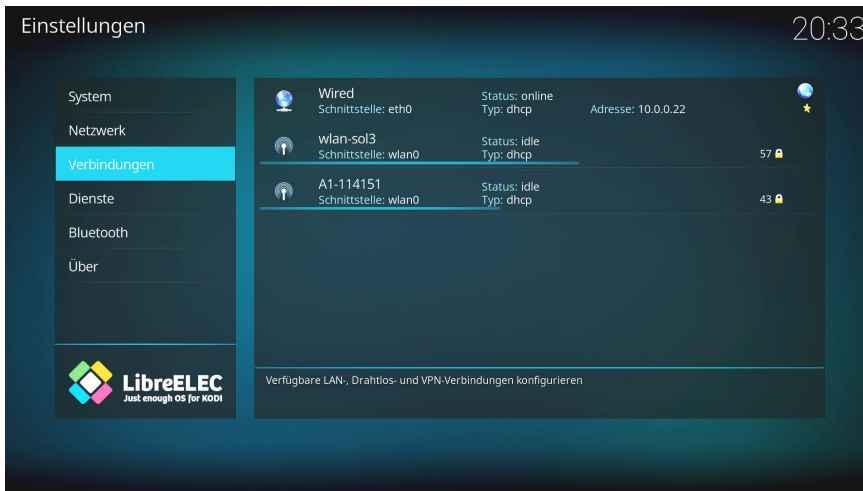


Abbildung 8.7 Übersicht über alle Netzwerkverbindungen in LibreELEC

```
user$ ssh root@libreelec
root@libreelec's password: *****
```

Sollte der Hostname `libreelec` in Ihrem lokalen Netzwerk nicht bekannt sein, geben Sie stattdessen die IP-Adresse Ihres Raspberry Pi an, also z. B. `ssh root@10.0.0.8`. Sie wird im Einstellungsmodul SYSTEMINFORMATIONEN angezeigt.

Unter LibreELEC können Sie das Root-Passwort nicht verändern: Zum einen steht das Kommando `passwd` gar nicht zur Verfügung, zum anderen wird das Dateisystem, in dem die Passwortdatei `/etc/shadow` gespeichert ist, aus Sicherheitsgründen im Read-only-Modus verwendet. Wenn Sie dennoch SSH verwenden möchten, empfiehlt die LibreELEC-Dokumentation, die Authentifizierung auf Schlüssel umzustellen.

SSH absichern

Dazu öffnen Sie auf Ihrem Linux-Arbeitsrechner ein Terminalfenster und überprüfen, ob die Dateien `.ssh/id_rsa` und `.ssh/id_rsa.pub` bereits existieren. Ist das nicht der Fall, führen Sie das Kommando `ssh-keygen` aus. Es erzeugt diese beiden Schlüsseldateien. Der Zugriff auf den Schlüssel wird standardmäßig durch ein Passwort abgesichert. Wenn Sie statt der Passwordeingabe einfach  drücken, entfällt dieses Sicherheitsmerkmal. Sofern Sie den Schlüssel nur für die LibreELEC-Administration verwenden, gehen Sie damit kein großes Risiko ein. Mit `ssh-copy-id` übertragen Sie nun den Schlüssel auf den Raspberry Pi. Weitere Details zu `ssh-copy-id` können Sie in [Abschnitt 31.4, »Authentifizierung mit Schlüsseln«](#), nachlesen.

```
user$ ssh-copy-id root@openelec
root@libreelec's password: *****
```

Von nun an können Sie sich von Ihrem Linux-Rechner aus *ohne* Login direkt auf dem LibreELEC-System anmelden. Sobald das funktioniert, können Sie den SSH-Server von

LibreELEC so konfigurieren, dass eine Authentifizierung per Passwort nicht mehr zulässig ist. Dazu führen Sie im Kodi-Menü **OPTIONEN • LIBREELEC • SERVICES** aus und setzen die Option **SSH KENNWORT DEAKTIVIEREN**.

**Samba absichern** LibreELEC macht mit dem Programm Samba einige Verzeichnisse im lokalen Netzwerk zugänglich. Allerdings hat standardmäßig jeder im lokalen Netzwerk Lese- und Schreibrechte – und das ohne Passwort! Wenn Ihnen das zu unsicher erscheint, aktivieren Sie im Dialog **OPTIONEN • LIBREELEC • SERVICES** die Option **SAMBA KENNWORTAUTHENTIFIZIERUNG BENUTZEN** und geben einen Benutzernamen und das dazugehörige Passwort an.

### Hardware-Decodierung

Für die weiteren Konfigurationsarbeiten ist es erforderlich, den Verzeichnisbaum von LibreELEC zu kennen. Dieser ist anders organisiert, als Sie es von anderen Linux-Distributionen kennen:

- ▶ Das Verzeichnis `/flash` enthält nicht nur die für den Boot-Vorgang erforderlichen Dateien (die bei anderen Raspberry-Pi-Distributionen in `/boot` liegen), sondern auch die fast 100 MiB große Datei `SYSTEM`. Diese Datei enthält das gesamte LibreELEC-System inklusive Kodi.
- ▶ Sowohl `/flash` als auch das Root-Dateisystem `/` werden im Read-only-Modus genutzt. Damit wird unbeabsichtigten Veränderungen vorgebeugt.
- ▶ Alle veränderlichen Daten befinden sich im Verzeichnis `/storage`. Das zugrunde liegende Dateisystem füllt beinahe die gesamte SD-Karte aus. Dort können z. B. eigene Filme, Audio-Dateien und Bilder gespeichert werden. Die entsprechenden Unterverzeichnisse dienen gleichzeitig als Netzwerkverzeichnisse. Daher können Sie unkompliziert von einem anderen Computer Dateien auf die SD-Karte des Raspberry Pi übertragen.

#### Hardware-Decodierung aktivieren

Der Grafikprozessor des Raspberry Pi kann zur Hardware-Decodierung mancher Video-Codecs eingesetzt werden. Wenn Sie MPEG-2- oder VC-1-Videos auf Ihrem Raspberry Pi ansehen möchten, sind diese Schlüssel zwingend erforderlich. Der VC-1-Codec wird z. B. in vielen WMV-Dateien verwendet.

Die Hardware-Decodierung muss durch einen Lizenzschlüssel freigeschaltet werden. Solche Schlüssel können Sie auf der Website <http://www.raspberrypi.com> erwerben. Beide Schlüssel zusammen kosten momentan etwas mehr als 4 EUR. Zum Bezahlen müssen Sie PayPal verwenden.

Beim Kauf müssen Sie die Seriennummer Ihres Raspberry Pi angeben. Diese können Sie dem Dialogblatt **OPTIONEN • SYSTEMINFO • HARDWARE** oder der Datei `/proc/cpuinfo` entnehmen:

```
pi$ grep Serial /proc/cpuinfo
Serial: 0000000013579bdf
```

Sie erhalten den Freischaltcode nach einer Weile per E-Mail. Der Raspberry-Pi-Store verspricht eine Zusendung innerhalb von 72 Stunden, bei mir hat es aber nur eine Stunde gedauert. Der Code ist mit der Seriennummer verknüpft und gilt somit nur für Ihren Raspberry Pi.

Sobald Sie den bzw. die Schlüssel erhalten haben, können Sie ihn bzw. sie in der Datei `/flash/config.txt` eintragen. Die erforderlichen Arbeiten müssen Sie via SSH durchführen. Dazu loggen Sie sich von einem anderen Computer aus als `root` in LibreELEC ein. Danach stellen Sie das `/flash`-Dateisystem vom Read-only- in den Read/Write-Modus um und laden `config.txt` in den Editor `nano`. Nach der Änderung starten Sie das System mit `reboot` neu:

```
user$ ssh root@openelec
root# mount -o remount,rw /flash
root# nano /flash/config.txt
root# reboot
```

In `nano` fügen Sie am Ende von `config.txt` zwei Zeilen ein, die wie im folgenden Listing aussehen – aber natürlich mit Ihren eigenen Freischaltcodes:

```
# Datei /flash/config.txt
...
decode_MPG2=0x12345678
decode_WVC1=0x9abcdef0
```

Ob alles funktioniert hat, können Sie nach dem Neustart via SSH mit dem Kommando `vcgencmd` verifizieren:

```
root# vcgencmd codec_enabled MPG2
MPG2=enabled
root# vcgencmd codec_enabled WVC1
WVC1=enabled
```

### Freischaltcodes unter Raspbian

Dieselben Freischaltcodes gelten selbstverständlich auch für alle anderen Raspberry-Pi-Distributionen. Dort finden Sie `config.txt` allerdings nicht im Verzeichnis `/flash`, sondern im Verzeichnis `/boot`. Änderungen in `config.txt` sind auf Anhieb möglich, d. h., Sie können auf das `mount`-Kommando verzichten.

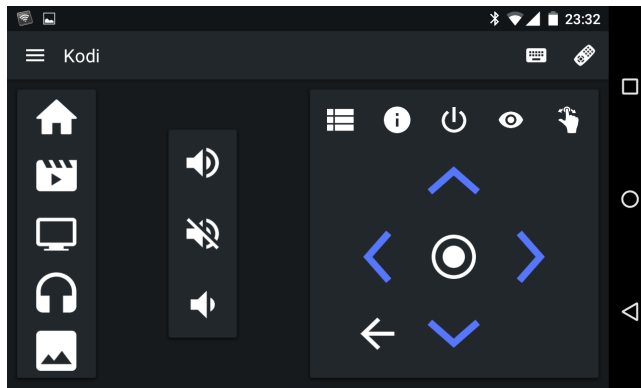
## Fernbedienung

Normalerweise gibt es im Betrieb eines Raspberry Pi als Medien-Center zwei Phasen: In der ersten Phase konfigurieren Sie das Gerät. Während dieser Phase sind Tastatur und Maus natürlich praktisch.

Ist die Konfiguration abgeschlossen, beginnt Phase zwei – die Nutzung des Geräts: Im einfachsten Fall reicht hierfür die Fernbedienung Ihres Fernsehgeräts aus. Dessen Signale werden nämlich bei modernen TV-Geräten via HDMI an den Raspberry Pi weitergeleitet. Alternative Steuerungsgeräte sind eine eigene Fernbedienung für den Raspberry Pi (erfordert einen zusätzlichen IR-Empfänger für den Raspberry Pi) oder die Bedienung durch eine Kodi-App auf dem Smartphone oder Tablet. Dieser Abschnitt gibt Tipps zur Konfiguration und Anwendung verschiedener Eingabegeräte.

**CEC-Fernbedienung** Im Idealfall können Sie Kodi direkt mit der Fernbedienung Ihres TV-Geräts steuern. Die meisten modernen Fernseher leiten Signale der Fernbedienung, die nicht für das TV-Gerät bestimmt sind, über das HDMI-Kabel via CEC (*Consumer Electronics Control*) an den Raspberry Pi weiter. Dieser verarbeitet dann die Signale. Laut Forenberichten funktioniert dieses Verfahren bei vielen Fernsehern auf Anhieb. Weniger Glück hatte ich mit meinem schon etwas älteren Sony-Fernseher: Obwohl das Gerät CEC an sich unter dem Sony-Markennamen »Bravia Sync« unterstützt, erkannte es den Raspberry Pi nicht als CEC-taugliches HDMI-Gerät.

**Smartphone-Fernbedienung** Zur Steuerung Ihres Media-Centers können Sie auch eine XBMC- oder Kodi-App für Android- oder iOS-Geräte verwenden (siehe [Abbildung 8.8](#)). Entsprechende Apps sind kostenlos im Google Play Store oder im App Store von Apple erhältlich. Sofern die Fernbedienung via CEC nicht funktioniert, ist ein Smartphone der unkomplizierteste Weg, um Kodi zu bedienen.



**Abbildung 8.8** Kodi-Fernbedienung mit der Android-App »Yatse«

Damit Kodi Signale der Apps verarbeitet, müssen Sie im Einstellungsmodul DIENSTE die Option FERNSTEUERUNG DURCH ANWENDUNGEN AUF ANDEREN RECHNERN ZULASSEN aktivieren.

Die Kodi-Steuerung durch das Smartphone oder Tablet erfolgt über WLAN. Bei der Konfiguration der Kodi-App müssen Sie den Hostnamen oder die IP-Adresse Ihres

Raspberry Pi angeben. Der Kodi-Webserver von LibreELEC verwendet standardmäßig den Port 8080 und den Benutzernamen `kodi` (ohne Passwort).

Anstelle einer eigenen App können Sie auch einfach den Webbrowser Ihres Smartphones oder Tablets verwenden. Damit ersparen Sie sich die Installation einer App. Die entsprechenden Kodi-Einstellungen finden Sie in den Systemeinstellungen im Dialogblatt **DIENTE • STEUERUNG**, wobei unter LibreELEC die Fernsteuerung per Webbrowser bereits standardmäßig aktiviert ist.

Wenn die TV-Fernbedienung nicht funktioniert und Sie auch nicht ein Smartphone zur Bedienung verwenden möchten, können Sie Ihren Raspberry Pi (wie in [Abschnitt 8.4](#), »Hardware-Basteleien«, beschrieben) selbst um einen IR-Empfänger erweitern. Das ist nicht schwierig, selbst Einsteiger in die Welt der Elektronik werden dabei nicht überfordert.

GPIO-IR-  
Empfänger

Wesentlich mühsamer und nervenaufreibender ist die LIRC-Konfiguration. LIRC steht für *Linux Infrared Remote Control*. `lirc` ist ein Paket von Programmen, das IR-Signale verarbeitet.

LIRC-  
Konfiguration


Um LIRC einzustellen, müssen Sie eine SSH-Verbindung zu LibreELEC herstellen. Danach werfen Sie einen Blick in die Datei `/etc/lirc/lircd.conf.rpi`. Wenn Sie Glück haben, finden Sie dort die erforderliche Konfiguration für Ihre Fernbedienung. In diesem Fall kopieren Sie die Datei nach `/storage/.config/lircd.conf` und löschen dann mit einem Editor alle Zeilen, die *nicht* zu Ihrer Fernbedienung passen. Zum Schluss darf es nur noch einen Block geben, der mit `begin remote` beginnt und mit `end remote` endet. Damit die Konfiguration wirksam wird, starten Sie LibreELEC mit `reboot neu`.

Wenn die Fernbedienung weiterhin nicht funktioniert, ist Handarbeit erforderlich. Dazu stoppen Sie zuerst den LIRC-Dämon:

```
root# killall lircd
```

Nun löschen Sie die eventuell vorhandene Datei `/storage/.config/lircd.conf` und erstellen mit dem Programm `irrecord` eine neue Konfigurationsdatei, die genau für Ihre Fernbedienung passt. Während das Programm läuft, müssen Sie zuerst wahllos 160-mal eine Taste Ihrer Fernbedienung drücken. Jede Taste sollte mindestens einmal gedrückt werden.

```
root# irrecord -d /dev/lirc0 /storage/.config/lircd.conf
irrecord - application for recording IR-codes for usage
with lirc
...
Now start pressing buttons on your remote control.
Don't stop pressing buttons until two lines of dots (2x80)
have been generated.
...
```

In der zweiten Phase müssen Sie jeweils zuerst einen LIRC-Tastennamen angeben und dann die entsprechende Taste auf Ihrer Fernbedienung drücken. Mit  schließen Sie diese überaus mühsame Konfiguration schließlich ab.

```
Now enter the names for the buttons.
Please enter the name for the next button (press <ENTER> to finish recording)
KEY_UP
Now hold down button "KEY_UP".
<Pfeil hinauf der Fernbedienung>

Please enter the name for the next button (press <ENTER> to finish recording)
KEY_DOWN
Now hold down button "KEY_DOWN".
<Pfeil hinunter der Fernbedienung>
...
```

Welche LIRC-Tastennamen es gibt, ermitteln Sie am besten im Voraus mit dem Befehl `irrecord -l`:

```
root# irrecord -l | grep KEY
KEY_0
KEY_102ND
KEY_1
KEY_2
...
```

Ein prinzipielles Problem besteht darin, dass die KEY-Namen nicht immer eindeutig sind. Sollen Sie der Pfeiltaste nach oben `KEY_UP` oder `KEY_VOLUMEUP` zuordnen? Hier hilft leider nur Experimentieren oder eine Suche im Internet. Das folgende Listing zeigt ein Beispiel für die Konfiguration der Apple-Fernbedienung, deren größter Vorteil darin besteht, dass sie aus nur sieben Tasten besteht:

```
# Datei /storage/.config/lircd.conf
# mit irrecord aufgezeichnete Konfiguration für die
# Apple-Aluminium-Fernbedienung
begin remote
  name          lircd.conf
  bits          8
  flags         SPACE_ENC
  eps           30
  aeps         100
  header        9099 4462
  one           587 1646
  zero          587 535
  ptrail        587
  pre_data_bits 16
  pre_data      0x77E1
  post_data_bits 8
```

```

post_data      0x30
gap            38524
toggle_bit_mask 0x0
begin codes
  KEY_UP      0xD0
  KEY_DOWN    0xB0
  KEY_LEFT    0x10
  KEY_RIGHT   0xE0
  KEY_MENU    0x40
  KEY_PLAY    0x7A 0x20
  KEY_OK      0xBA 0x20
end codes
end remote

```

Um die selbst erzeugte Konfigurationsdatei zu aktivieren, starten Sie LibreELEC abermals neu. Nach dem Neustart können Sie mit `irw` überprüfen, ob `lircd` funktioniert. Dazu starten Sie das Programm und drücken ein paar Tasten der Fernbedienung:

```

root# irw
 6a 0 KEY_RIGHT devinput
 6a 0 KEY_RIGHT_UP devinput
 69 0 KEY_LEFT devinput
 69 0 KEY_LEFT_UP devinput
<Strg>+<C>

```

### LibreELEC-Updates

Wenn LibreELEC Updates erkennt, weist es in der Kodi-Benutzeroberfläche auf diese Möglichkeit hin, unternimmt standardmäßig aber nichts. Um ein Update manuell zu starten, öffnen Sie das Einstellungsmodul LIBREELEC. Manuelle Updates sind beispielsweise bei Hauptversionssprüngen erforderlich, z. B. von LibreELEC 8 auf LibreELEC 9. Eine genauere Beschreibung des Update-Prozesses sowie Tipps zur Durchführung manueller Updates finden Sie auf der folgenden Webseite:

[https://wiki.libreelec.tv/index.php?title=HOW\\_TO:Update\\_LibreELEC](https://wiki.libreelec.tv/index.php?title=HOW_TO:Update_LibreELEC)

### Kodi-Grundfunktionen

Nachdem ich nun seitenlang alle möglichen Konfigurationsdetails beschrieben habe, wird es Zeit für die eigentliche Anwendung von Kodi – also für das Anhören von Musik und das Abspielen von Video-Dateien. Während die Konfiguration teilweise LibreELEC-spezifisch ist, gelten die weiteren Ausführungen für jede Distribution, die Kodi enthält, also z. B. auch für OpenELEC oder OSMC.

Vorweg ein Überblick über das Bedienungskonzept von Kodi: Auf dem Startbildschirm befindet sich das Kodi-Menü. Es besteht aus den Einträgen FILME, SERIEN,

MUSIK, MUSIKVIDEOS, VIDEOS etc. (siehe [Abbildung 8.9](#)). Die Auswahl des gewünschten Menüpunkts erfolgt per Tastatur, Maus oder Fernbedienung. Anfänglich führen die Menüs allerdings ins Leere. Das liegt daran, dass es im lokalen Dateisystem von Kodi noch keine Multimedia-Dateien gibt und Kodi auch keine Netzwerkquellen für derartige Dateien kennt.

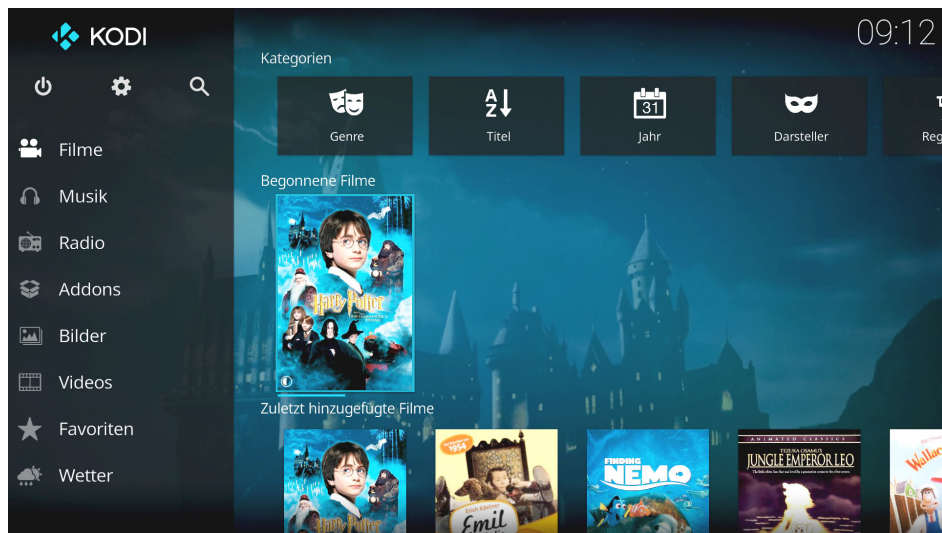
### Filme versus Videos

Kodi unterscheidet zwischen Filmen und Videos. Auf den ersten Blick scheint es, als würden beide Begriffe dasselbe meinen. Der Unterschied besteht darin, dass Kodi bei »Filmen« auch Kontextinformationen kennt, also den Namen, die Länge des Films, eine Inhaltsangabe usw. Kodi verwaltet eine eigene Datenbank, in der es diese Informationen speichert.

Ein »Video« ist dagegen eine Datei, bei der diese Zusatzinformationen fehlen. Abspielen können Sie sowohl Videos als auch Filme, aber Filme können von Kodi ansprechender präsentiert werden, nach Schlagwörtern durchsucht werden etc.

Noch mehr Details zu dieser Unterscheidung und zur Kodi-Datenbank, in der diese Metadaten gespeichert werden, finden Sie hier:

<https://www.kodinerds.net/index.php/Thread/16483>



**Abbildung 8.9** Der Kodi-Startbildschirm mit Menü

#### Lokale Videos abspielen

Um den ersten Film abzuspielen, suchen Sie im Dateimanager eines beliebigen Rechners im lokalen Netzwerk den Hostnamen des Raspberry Pi (normalerweise libreelec, wenn Sie LibreELEC verwenden). Sollte der Rechner nicht sichtbar sein, stellen Sie die



Verbindung manuell her und geben als Netzwerkadresse `smb://<hostname>` an. Nun kopieren Sie eine oder mehrere Video-Dateien in das Verzeichnis `Videos`.

Zurück auf dem Raspberry Pi navigieren Sie nun durch die Menüs `FILME • ZU DATEIEN • VIDEOS`. Dort finden Sie die Video-Dateien, die Sie auf die SD-Karte kopiert haben, und können sie abspielen (siehe [Abbildung 8.10](#)). Sollte die Wiedergabe nicht funktionieren, vergewissern Sie sich, dass Sie die Lizenzschlüssel zur Hardware-Decodierung der Formate MPEG-2 und VC-1 korrekt eingerichtet haben. Grundsätzlich kann Kodi keine DRM-geschützten Video-Dateien abspielen.



**Abbildung 8.10** Videos abspielen in Kodi

Kodi zeigt die in einem Verzeichnis enthaltenen Dateien standardmäßig als Liste mit Dateinamen an. Über ein Menü am linken Bildschirmrand, das Sie per Fernbedienung mit der linken Pfeiltaste aufrufen, können Sie zwischen verschiedenen Darstellungsformen wechseln, z. B. `LISTE`, `THUMBNAIL` oder `POSTER`.

Die SD-Karte Ihres Raspberry Pi ist für erste Experimente gut geeignet. Eine mögliche Alternative zur lokalen Speicherung von Videos besteht darin, die Dateien auf eine externe Festplatte oder einen USB-Stick zu kopieren und dieses Gerät dann mit dem Raspberry Pi zu verbinden. Führen Sie nun `VIDEOS • DATEIEN • DATEIEN HINZUFÜGEN` aus, erscheint der Name des Datenträgers in der Liste der zur Auswahl stehenden Datenquellen. Dort können Sie die Video-Dateien auswählen.

**Video-Dateien  
eines USB-Sticks  
abspielen**

Am elegantesten funktioniert der Kodi-Betrieb, wenn Sie über das lokale Netzwerk oder WLAN auf Audio- und Video-Dateien zugreifen, die ein anderer Computer oder ein NAS-Gerät bereithält. Kodi unterstützt die meisten gängigen Protokolle für Netzwerkverzeichnisse, unter anderem `SMB` (Windows/Samba), `DLNA`, `AFP` und `NFS`.

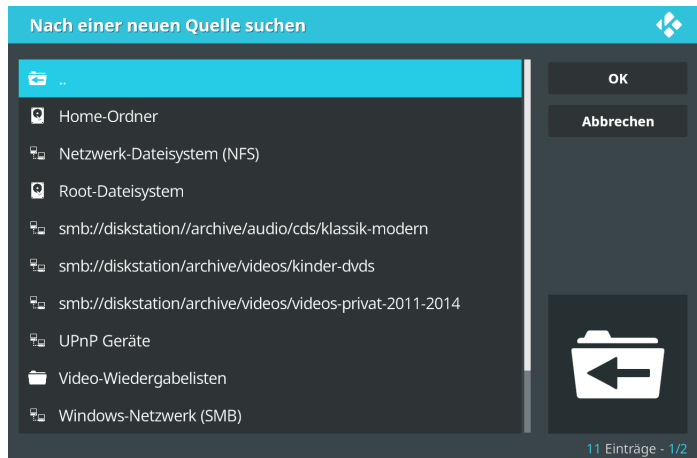
**Video-Dateien  
aus dem lokalen  
Netzwerk  
abspielen**

### Richten Sie Medienverzeichnisse per Tastatur und Maus ein!

Es ist außerordentlich umständlich, Medienquellen per Fernbedienung einzurichten. Führen Sie diese Konfigurationsarbeiten daher mit Tastatur und Maus aus. Sind die Quellen einmal korrekt konfiguriert, ist die Auswahl von Filmen mit einer Fernbedienung ein Kinderspiel.

Wenn der Medien-Server bzw. das NAS-Gerät DLNA und UPnP unterstützt, ist die Kodi-Konfiguration am einfachsten. Die Kürzel DLNA und UPnP stehen für *Digital Living Network Alliance* und *Universal Plug and Play* und bezeichnen Standards zur unkomplizierten Vernetzung und Nutzung von Audio- und Video-Geräten. Viele NAS-Geräte unterstützen DLNA/UPnP – suchen Sie gegebenenfalls nach der entsprechenden Option in der Benutzeroberfläche Ihres NAS-Geräts!

Um in Kodi auf ein DLNA-Gerät zuzugreifen, führen Sie VIDEOS • DATEIEN • VIDEOS HINZUFÜGEN aus. Im Dialog VIDEOQUELLE HINZUFÜGEN führt der Button DURCHSUCHEN in den UnterdialoG NACH EINER NEUEN QUELLE SUCHEN (siehe [Abbildung 8.11](#)). Dort wählen Sie UPnP GERÄTE aus. Kodi zeigt dann eine Liste aller DLNA-Geräte im lokalen Netzwerk an. Anschließend wählen Sie ein Verzeichnis des DLNA-Geräts aus und geben diesem bei Bedarf einen eigenen Namen.



**Abbildung 8.11** Auswahl einer neuen Medienquelle

Auch der Zugriff auf Video-Dateien in einem Windows-Netzwerkverzeichnis ohne Passwortschutz ist einfach: Sie beginnen wieder mit VIDEOS • DATEIEN • VIDEOS HINZUFÜGEN • DURCHSUCHEN • NACH EINER NEUEN QUELLE SUCHEN. Im Auswahldialog finden Sie den Eintrag WINDOWS-NETZWERK (SMB) (siehe [Abbildung 8.11](#)). Damit können Sie Windows-Server in der Arbeitsgruppe WORKSTATION auswählen.

Sollten Ihre Netzwerkverzeichnisse durch ein Passwort abgesichert sein, führt die Auswahl eines Servers durch WINDOWS-NETZWERK (SMB) allerdings zur wenig aussagekräftigen Fehlermeldung *Operation not permitted*. Lassen Sie sich davon nicht irritieren, starten Sie vielmehr die Auswahl des Netzwerkverzeichnisses mit VIDEOS • DATEIEN • VIDEOS HINZUFÜGEN • DURCHSUCHEN • NACH EINER NEUEN QUELLE SUCHEN neu. Nun wählen Sie aber den Eintrag NETZWERKFREIGABE HINZUFÜGEN. Dadurch gelangen Sie in einen Dialog, in dem Sie alle Parameter des Netzwerkverzeichnisses frei eingeben können: das Protokoll, den Namen des Servers oder NAS-Geräts, das Freigabeverzeichnis, den Benutzernamen und das Passwort.

ADD-ONS • VIDEO-ADDONS • ZUM ADDON-BROWSER führt in eine Liste von Add-ons (siehe [Abbildung 8.12](#)), mit denen Sie Filme aus diversen Internet-Angeboten auf Ihrem Media-Center ansehen können. Unter anderem gibt es Add-ons für Apple iTunes-Podcasts, die ARD Mediathek, Arte+7, Netzkino, die ORF TVthek, Spiegel Online, YouTube sowie für die ZDF Mediathek. Leider können viele Angebote nur im jeweiligen Land uneingeschränkt genutzt werden, Arte+7 also in Deutschland und Frankreich, die ORF TVthek nur in Österreich etc. Die Verwendung der Add-ons aus anderen Ländern funktioniert entweder gar nicht oder nur mit einem reduzierten Video-Angebot.

Videos aus dem Internet

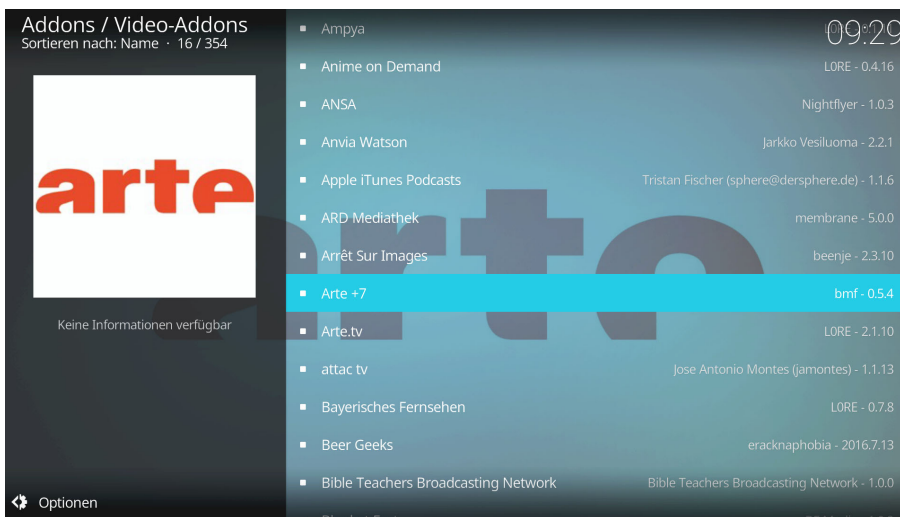


Abbildung 8.12 Video-Add-ons für Kodi

Das Abspielen von Musik folgt demselben Muster wie die Wiedergabe von Videos: Sie müssen zuerst eine Musikquelle definieren (ein lokales Verzeichnis, ein Netzwerkverzeichnis, einen DLNA-Server etc.) und können dann einen Titel zur Wiedergabe auswählen. Kodi zeigt automatisch CD-Cover an, wenn diese in die MP3-Dateien eingebettet sind (siehe [Abbildung 8.13](#)).

Audio

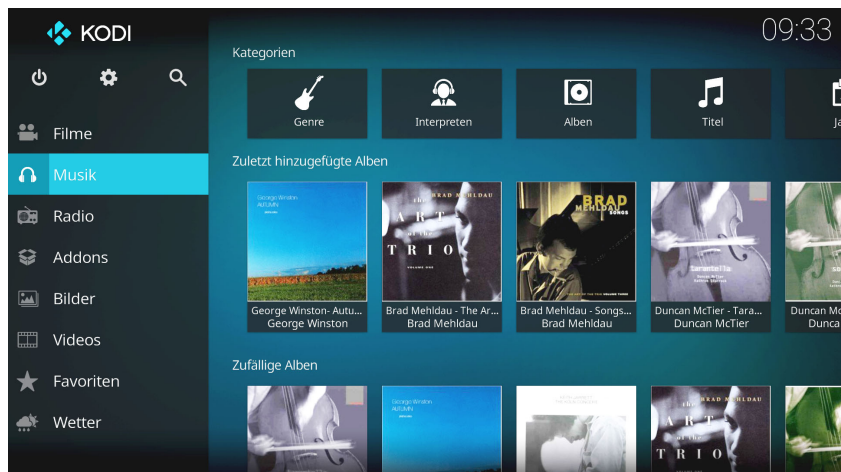


Abbildung 8.13 Kodi als Audio-Player

## 8.4 Hardware-Basteleien

Die eigentliche Besonderheit des Raspberry Pi ist weder seine winzige Größe noch sein Preis – die riesige Faszination für den Raspberry Pi geht vielmehr von 40 Pins (elektrischen Kontakten) aus, die zur Messung und Steuerung elektronischer Geräte verwendet werden können. Sowohl Elektronikbastler als auch Embedded-Linux-Profis bekommen mit dem Raspberry Pi ein Spielzeug bzw. Werkzeug in die Hand, das die Entwicklung computergesteuerter Geräte so einfach wie selten zuvor macht.

### Zerstören Sie Ihren Raspberry Pi nicht durch Unachtsamkeit!

Auch wenn ich mich in diesem Abschnitt an Elektronikeinsteiger und nicht an Embedded-Linux-Profis wende, gehe ich im Weiteren davon aus, dass Sie elementare Grundregeln im Umgang mit elektronischen Komponenten kennen:

- ▶ Durch elektrostatische Ladungen können Sie Ihren Raspberry Pi zerstören! Dabei reicht es, bloß einen elektrischen Kontakt zu berühren. Verwenden Sie ein Antistatikband (ESD-Armband).
- ▶ Auch versehentliche Kurzschlüsse, die falsche Beschaltung von Pins und dergleichen können Ihrem Minicomputer den Garaus machen.
- ▶ Schalten Sie Ihren Raspberry Pi immer aus, wenn Sie Veränderungen an der Schaltung durchführen.
- ▶ Beachten Sie schließlich, dass die meisten GPIO-Pins eine maximale Spannung von 3,3 Volt erwarten. Die für viele andere elektronische Bauteile üblichen 5 Volt sind zu hoch und können den Raspberry Pi ebenfalls kaputt machen.

## Das Layout der 40-Pin-Steckerleiste

Die Platine des Raspberry Pi enthält in einer Ecke eine Steckerleiste mit  $2 \times 20$  Kontakten in einem Rasterabstand von 2,54 mm. Beim Raspberry Pi 2 hat diese Steckerleiste die offizielle Bezeichnung »J8-Header«. Bei älteren Modellen wurde die damals kleinere GPIO-Leiste mit 26 Pins »P1-Header« genannt.

Diese Steckerleiste stellt neben einigen allgemein verwendbaren Kontakten (*General Purpose Input/Output* = GPIO) auch zwei Versorgungsspannungen (3,3 V bzw. 5 V) sowie die Masse (also 0 V) zur Verfügung. [Tabelle 8.1](#) erklärt die wichtigsten Abkürzungen, die zur Beschreibung von GPIO-Pins verwendet werden. Viele weitere Details können Sie auf der folgenden Seite nachlesen:

[http://elinux.org/RPi\\_BCM2835\\_GPIOs](http://elinux.org/RPi_BCM2835_GPIOs)

| Abkürzung        | Bedeutung  |
|------------------|--|
| GND              | Ground (Masse, also 0 V)   |
| GPIO             | General Purpose Input Output   |
| GPCLK            | General Purpose Clock (einstellbarer Taktgeber)                      |
| I <sup>2</sup> S | Inter-IC Sound Interface (Übertragung von Audio-Daten)               |
| I <sup>2</sup> C | Inter-Integrated Circuit (serieller Datenbus)                        |
| PWM              | Pulse Width Modulation (für SPI)                                     |
| SD1              | Secondary memory Data bus  |
| SPI              | Serial Peripheral Interface (serieller Datenbus)                     |
| SPI-MOSI         | Master out, Slave in (für SPI)                                       |
| SPI-MISO         | Master in, Slave out (für SPI)                                       |
| SPI-SCLK         | Serial Clock (für SPI)   |
| UART             | Universal Asynchronous Receiver Transmitter (serielle Schnittstelle) |

**Tabelle 8.1** Abkürzungen

[Abbildung 8.14](#) fasst die Belegung der 40 Pins des J8-Headers zusammen (Quelle: [http://elinux.org/RPi\\_Low-level\\_peripherals](http://elinux.org/RPi_Low-level_peripherals), ohne Gewähr!). Viele Pins erfüllen je nach Programmierung alternative Funktionen. Beispielsweise können die Pins 3 und 5 nicht nur als GPIO-Kontakte verwendet werden, sondern auch zum Anschluss einer elektronischen Komponente mit I<sup>2</sup>C-Bus. Pin-Belegung

Diese Seite enthält eine umfassende Beschreibung aller GPIO-Pins des Broadcom BCM2835 System-on-a-Chip. Dieser Bauteil, der auch die CPU und die GPU enthält, war das Kernstück der Raspberry-Pi-Platine der ersten Generation. Der aktuelle Raspberry Pi 3 enthält den kompatiblen Nachfolge-Chip BCM2837. Leider gibt es zum BCM2837 momentan weit weniger umfassende Dokumentation als zum Vorgängermodell BCM2835.

| Name   | BCM | Pin | Header J8 |   | Pin | BCM | Name   |
|--------|-----|-----|-----------|---|-----|-----|--------|
| 3,3 V  | –   | 1   | □         | ○ | 2   | –   | 5 V    |
| SDA    | 2   | 3   | ○         | ○ | 4   | –   | 5 V    |
| SCL    | 3   | 5   | ○         | ○ | 6   | –   | 0 V    |
| GPIO7  | 4   | 7   | ○         | ○ | 8   | 14  | TxD    |
| 0 V    | –   | 9   | ○         | ○ | 10  | 15  | RxD    |
| GPIO0  | 17  | 11  | ○         | ○ | 12  | 18  | GPIO1  |
| GPIO2  | 27  | 13  | ○         | ○ | 14  | –   | 0 V    |
| GPIO3  | 22  | 15  | ○         | ○ | 16  | 23  | GPIO4  |
| 3,3 V  | –   | 17  | ○         | ○ | 18  | 24  | GPIO5  |
| MOSI   | 10  | 19  | ○         | ○ | 20  | –   | 0 V    |
| MISO   | 9   | 21  | ○         | ○ | 22  | 25  | GPIO6  |
| SCLK   | 11  | 23  | ○         | ○ | 24  | 8   | CE0    |
| 0 V    | –   | 25  | ○         | ○ | 26  | 7   | CE1    |
| ID_SD  | 0   | 27  | ○         | ○ | 28  | 12  | ID_SC  |
| GPIO21 | 5   | 29  | ○         | ○ | 30  | –   | 0 V    |
| GPIO22 | 6   | 31  | ○         | ○ | 32  | 12  | GPIO26 |
| GPIO23 | 13  | 33  | ○         | ○ | 34  | –   | 0 V    |
| GPIO24 | 19  | 35  | ○         | ○ | 36  | 16  | GPIO27 |
| GPIO25 | 26  | 37  | ○         | ○ | 38  | 20  | GPIO28 |
| 0 V    | –   | 39  | ○         | ○ | 40  | 21  | GPIO29 |

Abbildung 8.14 Schematischer Plan des Raspberry Pi mit Pin-Nummerierung

Wirrwarr bei  
der GPIO-Num-  
merierung

Leider gibt es verschiedene Möglichkeiten, die Pins der GPIO-Leiste zu bezeichnen:

- ▶ Wirklich eindeutig ist nur die Pin-Nummer, bei der die Pins der Steckerleiste durchnummeriert werden.
- ▶ Die Broadcom-CPU des Raspberry Pi stellt eine Menge GPIO-Kontakte zur Verfügung – weit mehr, als auf der Raspberry-Pi-Platine zugänglich sind. Diese GPIOs sind durch die vom Hersteller vorgegebene BCM-Nummern bezeichnet (siehe [Abbildung 8.14](#)).
- ▶ Schließlich gibt es noch die Raspberry-Pi-spezifische Benennung der GPIO-Pins (Spalte *Name* in [Abbildung 8.14](#)). Teilweise spiegelt der Name einfach die Funktion wieder (z. B. SCLK), teilweise verwendet man aber auch Nummern (z. B. GPIO2). Unbegreiflicherweise sind das aber andere Nummern als die BCM-Nummern!

Der Kontakt des J8-Headers mit der Pin-Nummer 11 entspricht »GPIO 17« in der BCM-Nomenklatur, aber »GPIO 0« in der Raspberry-Pi-Nomenklatur. In den weiteren Beispielen in diesem Buch beziehe ich mich immer auf die Pin-Nummer. Wenn Sie aber Anleitungen aus dem Internet folgen, müssen Sie immer hinterfragen, welche Nomenklatur zur Anwendung kommt und welcher Pin nun wirklich gemeint ist.

Pin 1 und 17 stellen eine Spannung von 3,3 V zur Verfügung und werden oft dazu verwendet, externe Schaltungen mit Strom zu versorgen. Die beiden Pins dürfen allerdings *zusammen* maximal mit 50 mA belastet werden.

Maximaler  
Ausgangsstrom

Die Pins 2 und 4 mit einer Versorgungsspannung von 5 V werden über eine selbst-rückstellende Sicherung (*Poly Fuse*) geleitet. Fließt hier zu viel Strom, schaltet sich der Raspberry Pi für eine Weile ab. Mit etwas Glück kommt es daher zu keinen bleibenden Schäden.

Wenn Sie GPIO-Kontakte zur Steuerung verwenden (Konfiguration als Output) und auf HIGH stellen, beträgt die Spannung am betreffenden GPIO-Pin 3,3 V. Der Steuerungsstrom pro Pin sollte 8 mA nicht überschreiten (bzw. 50 mA für *alle* GPIOs sowie Pin 1 und 17). Verwenden Sie also geeignete Vorwiderstände!

Soweit Sie GPIO-Kontakte als Eingänge verwenden, dürfen Sie eine Spannung von 3,3 V nicht überschreiten! 5 V sind definitiv zu viel und können Ihren Raspberry Pi zerstören.

Maximale Ein-  
gangsspannung

Vor jedem Projekt müssen Sie sich die Frage stellen: »Welche der vielen GPIO-Pins setze ich ein?« Solange es nur darum geht, erste Experimente durchzuführen und ein paar Leuchtdioden ein- und auszuschalten, können Sie dazu jeden der 17 GPIO-*n*-Pins verwenden. Diverse Spezialfunktionen stehen allerdings auf ausgewählten Pins zur Verfügung. Im folgenden Überblick gelten – wie schon erwähnt – die Pin-Nummern auf den J8-Header des Raspberry Pi:

Welcher GPIO für  
welchen Zweck?

- ▶ **Pin 3 und 5** sind erforderlich für I<sup>2</sup>C-Komponenten. Die beiden Pins sind mit einem 1,8-k $\Omega$ -Pull-up-Widerstand verbunden und eignen sich auch gut als Signaleingänge (z. B. für Schalter/Taster).
- ▶ **Pin 7** wird vom 1-Wire-Kerneltreiber verwendet oder kann als Taktgeber eingesetzt werden.
- ▶ **Pin 8 und 10** werden beim Booten des Raspberry Pi standardmäßig als serielle Schnittstelle konfiguriert. Dort werden normalerweise die Kernelmeldungen ausgegeben. Wenn Sie die Pins für allgemeine I/O-Aufgaben nutzen möchten, müssen Sie sie umprogrammieren, z. B. mit dem Kommando `gpio` aus der WiringPi-Bibliothek.
- ▶ **Pin 11, 12 und 13** können zum Anschluss von SPI-Komponenten verwendet werden (SPI-Kanal 1).



- ▶ **Pin 12** wird standardmäßig vom LIRC-Kerneltreiber verwendet und eignet sich daher gut als Signaleingang für einen IR-Empfänger. Dieser Pin kann auch als PWM-Ausgang verwendet werden. Vorsicht: Wenn Sie Audio-Signale über den Kopfhörerausgang ausgeben, wird dieses Signal auch über Pin 12 geleitet.
- ▶ **Pin 19, 21, 23, 24 und 26** können zum Anschluss von SPI-Komponenten verwendet werden (SPI-Kanal 0).
- ▶ **Pin 27 und 28** bilden die Schnittstelle zum I<sup>2</sup>C-Bus 0. Dieser Bus ist zur Kommunikation mit EEPROMs reserviert, die sich auf standardisierten Erweiterungs-Boards (sogenannten »HATs«) befinden.

Noch mehr Informationen finden Sie auf der folgenden Seite, die sich allerdings *nicht* auf die Pin-Nummern des J8-Headers bezieht, sondern BCM-Nummern verwendet:

[http://elinux.org/RPi\\_BCM2835\\_GPIOs](http://elinux.org/RPi_BCM2835_GPIOs)

#### Kontakt zu GPIO-Pins herstellen

Bevor Sie Ihr erstes Bastelprojekt beginnen, müssen Sie sich überlegen, wie Sie den elektronischen Kontakt zu einem der 26 Pins herstellen. Für kleine Versuchsaufbauten auf einem Steckboard sind kurze Kabel mit Stecker und Buchse ideal (siehe Abbildung 8.15). Die Kabel werden in diversen Raspberry-Pi-Shops angeboten, oft zusammen mit einem Steckboard als Starter-Kit. Suchen Sie gegebenenfalls im Internet nach *breadboard jumper wire male female*.

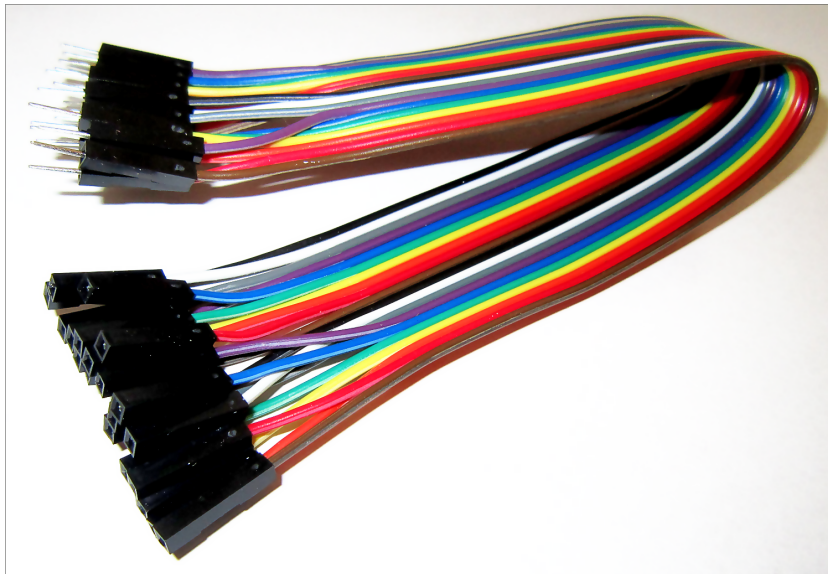


Abbildung 8.15 Steckboard-Kabel

Mit ein wenig Erfahrung im Löten und einer Buchsenleiste im 2,54-mm-Raster (erhältlich in jedem Elektronikmarkt) können Sie sich selbst passende Stecker herstellen.



Eine andere Alternative ist ein 40-Pin-Stecker mit einem Flachbandkabel, dessen Drähte Sie dann trennen. Manche Raspberry-Pi-Händler bieten auch spezielle *Cobbler* an, um alle 40 Pins des J8-Headers über ein Flachbandkabel mit den Kontaktreihen eines Steckboards zu verbinden.

Löten Sie auf keinen Fall die Kabel direkt an die Stecker-Pins! Die unvermeidlichen Lötreste machen es nahezu unmöglich, später einen Flachbandstecker zu verwenden.

In Ergänzung zu den 40 Pins des J8-Headers enthält die Platine des Raspberry Pi einige weitere Kontaktstellen – die P2-, P3-, P5- und P6-Header. Diese Kontaktstellen sind nicht mit Steckern verbunden. Wenn Sie diese Kontakte nutzen möchten, müssen Sie Ihre elektronischen Bauteile, Kabel oder Stecker dort anlöten.

Noch mehr  
Kontakte/GPIOs  
(P2, P3, P5, P6)

Die acht Kontakte des P5-Headers befinden sich je nachdem, welches Modell des Raspberry Pi Sie einsetzen, an unterschiedlichen Orten auf der Platine. Im weiteren Verlauf dieses Kapitels beziehe ich mich ausschließlich auf die 40 Pins des J8-Headers! Eine vollständige Hardware-Beschreibung des Raspberry Pi finden Sie hier:

[http://elinux.org/RPi\\_Hardware](http://elinux.org/RPi_Hardware)

### LEDs ein- und ausschalten

Sozusagen als *Hello World!*-Projekt zeige ich Ihnen, wie Sie mit Ihrem Raspberry Pi eine Leuchtdiode (LED) ein- und ausschalten. Die erste Variante besteht darin, die LED direkt an die 3,3-V-Spannungsversorgung anzuschließen. Sie leuchtet dann immer.

LED im  
Dauerbetrieb

Im Datenblatt Ihrer LED lesen Sie nach, wie groß der Spannungsabfall an der Diode ist und welchen Strom die Diode erwartet – z. B. 2 V und 10 mA. Die Größe des erforderlichen Vorwiderstands ergibt sich aus der Restspannung  $3,3 \text{ V} - 2 \text{ V} = 1,3 \text{ V}$  und der Formel  $R = U / I = 1,3 \text{ V} / 10 \text{ mA}$  mit  $130 \Omega$ . Wenn Sie den nächstgrößeren Widerstand verwenden, den Sie finden, kann nichts passieren. Die LED leuchtet dann entsprechend weniger hell.

Da derselbe Schaltungsaufbau später über einen GPIO-Pin mit einem maximalen Ausgangsstrom von 8 mA gesteuert werden soll, ist es besser, den Widerstand gleich entsprechend größer zu dimensionieren ( $1,3 \text{ V} / 8 \text{ mA} = 163 \Omega$ ). Ich habe für meine Experimente mit  $330\text{-}\Omega$ -Widerständen gearbeitet, womit sich ein Strom von 4 mA ergibt.

Auf einem Steckboard bauen Sie nun die Schaltung gemäß Abbildung 8.16 auf und verbinden die Schaltung mit den Pins 1 (3,3 V) und 25 (GND) des Raspberry Pi. Achten Sie auf die richtige Polung der LED. Der längere Draht der LED verbindet die Anode (Plus), der kürzere die Kathode (Minus).

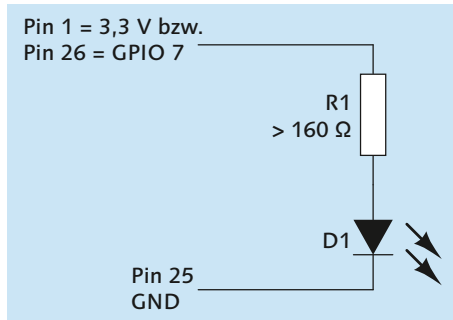


Abbildung 8.16 Simple LED-Schaltung

**LED manuell ein- und ausschalten**

Nachdem Sie sich überzeugt haben, dass die obige Schaltung prinzipiell funktioniert, verwenden Sie nun anstelle von Pin 1 (3,3 V) einen GPIO-Kontakt, z. B. Pin 26 für GPIO 7.

**Tipp**

GPIO-Pins sind zur Steuerung, nicht zur Stromversorgung gedacht. Wenn Sie ein elektronisches Bauteil mit mehr als 8 mA bei 3,3 V versorgen möchten, verwenden Sie zur Stromversorgung Pin 2 oder 4 (5 V) und steuern den Stromfluss durch einen GPIO-Ausgang über einen Transistor. Einen entsprechenden Schaltungsaufbau finden Sie hier:

<http://www.raspberrypi-spy.co.uk/2012/06/control-led-using-gpio-output-pin>

Beim Einschalten des Raspberry Pi wird die LED nun nicht mehr leuchten. Vielmehr können Sie die LED jetzt durch ein Python-Programm steuern. Den erforderlichen Quellcode geben Sie mit einem Editor ein:

```
#!/usr/bin/python3
import RPi.GPIO as GPIO
import time

# Pin-Nummern verwenden (nicht GPIO-Nummern!)
GPIO.setmode(GPIO.BOARD)

# Pin 26 (= GPIO 7) zur Datenausgabe verwenden
GPIO.setup(26, GPIO.OUT)

# Pin 26 einschalten
GPIO.output(26, GPIO.HIGH)

# Pin 26 nach fünf Sekunden wieder ausschalten
time.sleep(5)
GPIO.output(26, GPIO.LOW)
```

```
# alle vom Script benutzten GPIOs/Pins wieder freigeben
GPIO.cleanup()
```

Der Programmcode sollte auch ohne Python-Erfahrung auf Anhieb verständlich sein. `chmod` macht die Script-Datei ausführbar:

```
pi$ chmod a+x led1.py
pi$ ./led1.py
```

Das obige Beispiel hat den Einsatz der `RPi.GPIO`-Bibliothek für Python gezeigt. Diese Bibliothek ist die am häufigsten eingesetzte Bibliothek zur GPIO-Programmierung.

**RPi.GPIO und  
gpiozero**

In den letzten Jahren hat sich als Alternative dazu die `gpiozero`-Bibliothek etabliert. Sie ist stärker objektorientiert konzipiert und etwas einsteigerfreundlicher. Weitere Informationen sowohl zur `RPi.GPIO`-Bibliothek als auch zu `gpiozero` finden Sie hier:

<https://sourceforge.net/projects/raspberry-gpio-python>  
[http://elinux.org/RPi\\_Low-level\\_peripherals#GPIO\\_Code\\_examples](http://elinux.org/RPi_Low-level_peripherals#GPIO_Code_examples)  
<https://gpiozero.readthedocs.io/en/v1.3.2/recipes.html>

Nicht immer ist es praktisch, für jede Veränderung eines GPIO-Pins gleich ein Python-Script zu verfassen. Die Bibliothek `WiringPi` stellt das Kommando `gpio` zur Verfügung, mit dem Sie einzelne GPIO-Pins direkt im Terminal manipulieren können. Bei aktuellen Raspbian-Versionen steht das Kommando standardmäßig zur Verfügung (Paket `wiringpi`).

**GPIO-Steuerung  
im Terminal  
(WiringPi)**

Bei älteren Versionen ist die Installation von `WiringPi` aus einem Git-Repository in wenigen Minuten erledigt:

```
pi$ sudo apt install git
pi$ git clone git://git.drogon.net/wiringPi
pi$ cd wiringPi
pi$ ./build
```

Mit dem Kommando `gpio` können Sie nun ebenfalls die LED ein- und ausschalten. Das Kommando erfordert keine `root`-Rechte. Die Option `-1` (eins, nicht L) bewirkt dabei, dass `gpio` physische Pin-Nummern des J8-Headers als Parameter erwartet.

```
pi$ gpio -1 mode 26 out
pi$ gpio -1 write 26 1    (LED ein)
pi$ gpio -1 write 26 0    (LED aus)
```

`gpio readall` verrät den aktuellen Status aller GPIO-Pins. Dabei gibt die erste/letzte Spalte die BCM-Bezeichnung des Pins an, die zweite/vorletzte Spalte die `WiringPi`-Bezeichnung, die dritte/drittletzte den Raspberry-Pi-Namen:

```

pi$ gpio readall
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|BCM|wPi|  Name  |Mode| V | Physical | V | Mode| Name  | wPi| BCM|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|   |   | 3.3v   |    |   | 1  | 2  |   |   | 5v   |   |   |
| 2 | 8 | SDA.1  | IN  | 1 | 3  | 4  |   |   | 5V   |   |   |
| 3 | 9 | SCL.1  | IN  | 1 | 5  | 6  |   |   | 0v   |   |   |
| 4 | 7 | GPIO.7 | IN  | 1 | 7  | 8  | 1 | ALTO| TxD  | 15 | 14 |
|   |   | 0v     |    |   | 9  | 10 | 1 | ALTO| RxD  | 16 | 15 |
|17 | 0 | GPIO.0 | IN  | 0 | 11 | 12 | 0 | IN  | GPIO.1| 1  | 18 |
|27 | 2 | GPIO.2 | IN  | 0 | 13 | 14 |   |   | 0v   |   |   |
|22 | 3 | GPIO.3 | IN  | 0 | 15 | 16 | 0 | IN  | GPIO.4| 4  | 23 |
|   |   | 3.3v   |    |   | 17 | 18 | 0 | IN  | GPIO.5| 5  | 24 |
|10 |12 | MOSI   | IN  | 0 | 19 | 20 |   |   | 0v   |   |   |
| 9 |13 | MISO   | IN  | 1 | 21 | 22 | 0 | OUT | GPIO.6| 6  | 25 |
|11 |14 | SCLK   | IN  | 0 | 23 | 24 | 0 | OUT | CEO   |10  | 8  |
|   |   | 0v     |    |   | 25 | 26 | 1 | OUT | CE1   |11  | 7  |
| 0 |30 | SDA.0  | IN  | 1 | 27 | 28 | 1 | IN  | SCL.0 |31  | 1  |
| 5 |21 | GPIO.21| OUT | 1 | 29 | 30 |   |   | 0v   |   |   |
| 6 |22 | GPIO.22| IN  | 1 | 31 | 32 | 0 | IN  | GPIO.26|26  | 12 |
|13 |23 | GPIO.23| IN  | 0 | 33 | 34 |   |   | 0v   |   |   |
|19 |24 | GPIO.24| IN  | 0 | 35 | 36 | 0 | IN  | GPIO.27|27  | 16 |
|26 |25 | GPIO.25| IN  | 0 | 37 | 38 | 0 | IN  | GPIO.28|28  | 20 |
|   |   | 0v     |    |   | 39 | 40 | 0 | IN  | GPIO.29|29  | 21 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|BCM|wPi|  Name  |Mode| V | Physical | V | Mode| Name  | wPi| BCM|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

### LED-Überwachung der CPU-Temperatur

Um das Beispiel ein wenig interessanter zu machen, zeigt [Abbildung 8.17](#) eine Schaltung mit nunmehr drei Dioden (grün, gelb und rot), die durch die GPIOs 7, 8 und 25 gesteuert werden (Pin 26, 24 und 22). Das Ziel besteht darin, die LEDs in Abhängigkeit von der CPU-Temperatur ein- und auszuschalten: Die grüne LED soll leuchten, wenn die CPU-Temperatur zwischen 30 und 45 Grad beträgt, die gelbe LED bei Temperaturen zwischen 45 und 60 Grad und die rote LED bei Temperaturen darüber.

Nachdem Sie die Schaltung aufgebaut haben, testen Sie, ob alle drei LEDs funktionieren:

```

pi$ sudo -s
pi$ gpio -1 mode 26 out
pi$ gpio -1 write 26 1 (grüne LED ein)
pi$ gpio -1 write 26 0 (grüne LED aus)
pi$ gpio -1 mode 24 out
pi$ gpio -1 write 24 1 (gelbe LED ein)
pi$ gpio -1 write 24 0 (gelbe LED aus)
pi$ gpio -1 mode 22 out
pi$ gpio -1 write 22 1 (rote LED ein)
pi$ gpio -1 write 22 0 (rote LED aus)

```

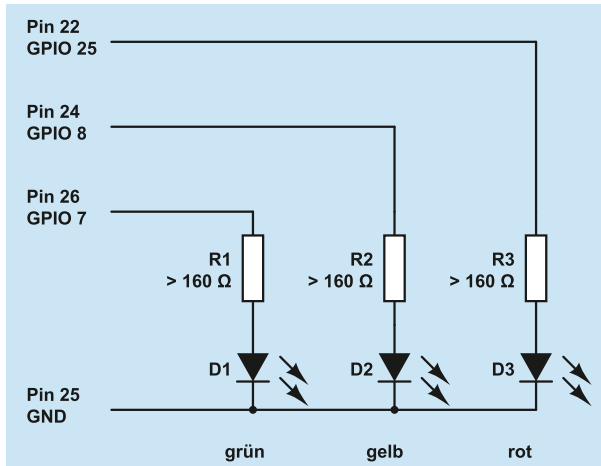


Abbildung 8.17 LED-Anzeige der CPU-Temperatur

Ein Shell-Script zum Ein- und Ausschalten der drei LEDs in Abhängigkeit von der CPU-Temperatur ist rasch geschrieben. Wenn Ihnen Python lieber ist, können Sie den Code natürlich ebenso gut damit formulieren.

```
#!/bin/bash
# Datei /home/pi/led-rot-gelb-gruen
greenpin=26
yellowpin=24
redpin=22
gpio -1 mode $greenpin out
gpio -1 mode $yellowpin out
gpio -1 mode $redpin out

# CPU-Temperatur
temp=$(cat /sys/class/thermal/thermal_zone0/temp)
echo $temp

# grüne LED ansteuern
if [[ "$temp" -ge 30000 && "$temp" -le 45000 ]]; then
    gpio -1 write $greenpin 1
else
    gpio -1 write $greenpin 0
fi
# gelbe LED ansteuern
if [[ "$temp" -ge 45000 && "$temp" -le 60000 ]]; then
    gpio -1 write $yellowpin 1
else
    gpio -1 write $yellowpin 0
fi
```

```
# rote LED ansteuern
if [[ "$temp" -ge 60000 ]]; then
  gpio -1 write $redpin 1
else
  gpio -1 write $redpin 0
fi
```

Diese Datei muss ausführbar sein:

```
root# chmod a+x /home/pi/led-rot-gelb-gruen
```

Jetzt geht es nur noch darum, dieses Script einmal pro Minute aufzurufen. Dazu legen Sie im Verzeichnis `/etc/cron.d` die folgende Datei an (siehe auch [Abschnitt 12.6](#), »Prozesse automatisch starten (Cron)«):

```
# Datei /etc/cron.d/cpu-temp
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
*/1 * * * * root /home/pi/led-rot-gelb-gruen
```

Von nun an werden die drei LEDs minütlich je nach CPU-Temperatur ein- und ausgeschaltet. Vergessen Sie nicht, `/etc/cron.d/cpu-led` zu löschen, wenn Sie die Pins 22, 24 und 26 später für eine andere Schaltung verwenden möchten!

### Eine LED mit einem Taster ein- und ausschalten

Die einfachste Form, an den Raspberry Pi *Eingaben* weiterzuleiten, ist ein simpler Taster, der den Stromkreis schließt, solange er gedrückt ist. Während umgangssprachlich oft alles, worauf man drücken kann, ein »Schalter« ist, unterscheidet die Elektrotechnik zwischen Schaltern, die den Zustand beibehalten (wie ein Lichtschalter), und Tastern, die zurückspringen, wenn man sie loslässt (wie bei Ihrer Tastatur). Für dieses Beispiel benötigen Sie also einen Taster. Wenn Sie zum Testaufbau ein Steckboard verwenden, fragen Sie in Ihrem Elektronikgeschäft nach einem *Print Taster*.

Das Ziel dieses Abschnitts ist eine Schaltung, bei der Sie durch einen kurzen Druck auf eine Taste eine LED einschalten. Drücken Sie nochmals, soll die LED wieder ausgehen. Die Aufgabenstellung klingt erdenklich trivial, aber Sie werden sehen, dass dieser Eindruck täuscht.

**GPIO-Input** Bevor Sie den Taster aber mit einem GPIO-Pin verbinden, müssen Sie sich Gedanken darüber machen, wie der Raspberry Pi Eingaben verarbeitet. Es ist möglich, einen GPIO-Pin als *Input* zu konfigurieren. Die Ausgangsspannung dieses Pins ist damit undefiniert. Wenn von außen eine Spannung nahe 0 V angelegt wird, wird das als *low* = 0 interpretiert. Ist die angelegte Spannung hingegen nahe 3,3 V, wird das Signal als *high* = 1 interpretiert. Als Input verwendete GPIOs können nicht zwischen anderen Zuständen unterscheiden und können somit nicht zur Messung der angelegten Spannung verwendet werden.

Ein Input-Pin soll nie unbeschaltet sein, weil seine Spannung dann undefiniert ist (*floating*). Gleichzeitig ist es nicht empfehlenswert, den Pin direkt mit der Masse oder mit der Versorgungsspannung (3,3 V) zu verbinden: Sollte der GPIO-Pin irrtümlich als Output-Pin programmiert sein, würden unter Umständen große Ströme fließen, die Ihren Raspberry Pi mit etwas Pech zerstören. Die Lösung für dieses Problem sind Pull-up- oder Pull-down-Widerstände in der Größenordnung von circa 1 k $\Omega$  bis 10 k $\Omega$ , um den Signaleingang für beide möglichen Zustände des Tasters mit der Masse bzw. mit 3,3 V zu verbinden. Hintergrundinformationen zu Pull-up- und Pull-down-Widerständen können Sie in der Wikipedia nachlesen:

Pull-up- und  
Pull-down-  
Widerstände

[https://de.wikipedia.org/wiki/Open\\_circuit#Beschaltung\\_der\\_Signalleitungen](https://de.wikipedia.org/wiki/Open_circuit#Beschaltung_der_Signalleitungen)

Bei der Beschaltung des Raspberry Pi können Sie sich Pull-up- und Pull-down-Widerstände unter Umständen sparen: Zum einen sind die Pins 3 und 5 des J8-Headers standardmäßig mit 1,8 k $\Omega$  externen Pull-up-Widerständen verbunden; zum anderen lassen sich alle GPIOs im Input-Modus so programmieren, dass CPU-interne Pull-up- oder Pull-down-Widerstände aktiviert werden. Der interne Schaltungsaufbau ist auf der folgenden Seite gut beschrieben:

<http://mosaic-industries.com/embedded-systems/microcontroller-projects/raspberry-pi/gpio-pin-electrical-specifications>

Dennoch ist es empfehlenswert, Signaleingänge grundsätzlich mit einem externen Pull-up- oder Pull-down-Widerstand zu versehen. Sie vermeiden damit Probleme, wenn ein GPIO-Pin versehentlich falsch konfiguriert ist oder während der Initialisierung des Raspberry Pi einen anderen Zustand einnimmt, als Ihre Schaltung voraussetzt. Gefahrlos auf Pull-up-Widerstände können Sie nur verzichten, wenn Sie Ihren Taster mit den Pins 3 oder 5 verbinden. Diese beiden Pins stehen aber nur zur freien Verfügung, wenn Ihre Schaltung keine I<sup>2</sup>S-Komponenten enthält.

Abbildung 8.18 zeigt den Aufbau der Schaltung. Soweit es die Leuchtdiode betrifft, gibt es keine Veränderung im Vergleich zum vorigen Abschnitt – wenn man einmal davon absieht, dass diesmal Pin 23 zur Ansteuerung verwendet wird.

Schaltung

Der Taster ist direkt mit der Masse und über den Pull-up-Widerstand R2 mit der Versorgungsspannung 3,3 V verbunden. Im Normalzustand lautet der Signalzustand an Pin 21 also High, beim Drücken Low. Der Widerstand R3 ist eine zusätzliche Sicherheitsmaßnahme. Er verhindert einen Kurzschluss für den zugegebenermaßen unwahrscheinlichen Fall, dass Pin 21 irrtümlich als Output programmiert wird, auf High gestellt und gleichzeitig der Taster gedrückt ist. Ohne R3 gäbe es dann eine direkte Verbindung zwischen 3,3 V an Pin 21 und der Masse; es würde mehr Strom fließen, als der Raspberry Pi liefern kann. Dank R3 ist der Strom selbst in diesem Fall auf 3,3 mA begrenzt.

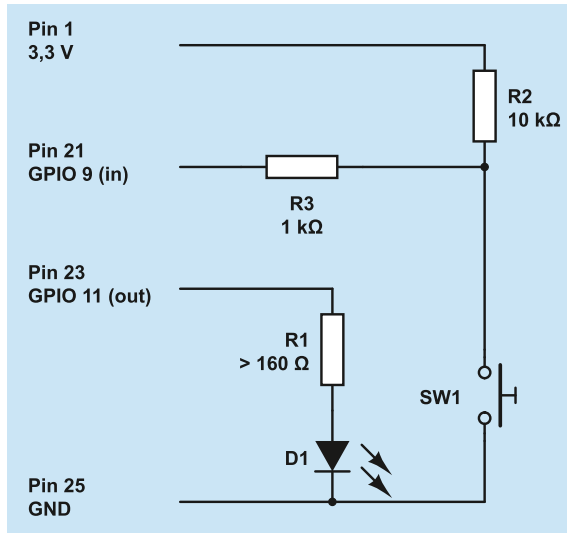


Abbildung 8.18 LED-Steuerung per Taster mit Pull-up-Widerstand

**Test** Bevor Sie sich an die Programmierung machen, sollten Sie kurz überprüfen, ob der Schaltungsaufbau funktioniert. Zuerst schalten Sie versuchsweise die LED ein/aus:

```
pi$ gpio -1 mode 23 out
pi$ gpio -1 write 23 1    (LED ein)
pi$ gpio -1 write 23 0    (LED aus)
```

Danach testen Sie den Signaleingang an Pin 21, wobei Sie einmal die Taste gedrückt halten:

```
pi$ gpio -1 mode 21 in
pi$ gpio -1 read 21    (Normalzustand)
1
pi$ gpio -1 read 21    (Taste gedrückt)
0
```

Wenn Sie ein direktes Feedback wünschen, können Sie Pin 21 mit einem kleinen Python-Script kontinuierlich abfragen:

```
#!/usr/bin/python3
import RPi.GPIO as GPIO
import time

# Pin-Nummern verwenden (nicht GPIO-Nummern!)
GPIO.setmode(GPIO.BOARD)

# GPIO 21 = Input
GPIO.setup(21, GPIO.IN)
```



```
while True:
    input = GPIO.input(21)
    print("Zustand: " + str(input))
    time.sleep(0.01)
```

Sobald Sie dieses Programm starten, gibt es den aktuellen Signaleingang von Pin 21 regelmäßig im Terminal aus – bis Sie das Programm mit `[Strg]+[C]` wieder stoppen.

Die Überwachung von Signalzuständen durch eine Schleife ist selten ein optimales Konzept. Wenn die `sleep`-Zeit kurz ist, verursacht das Programm eine Menge unnötige CPU-Last; verwenden Sie eine längere Zeit, steigt die Reaktionszeit des Programms, und im ungünstigsten Fall übersieht es einen kurzen Impuls ganz. Wesentlich intelligenter ist es, das Python-Programm so zu formulieren, dass es einfach auf einen Signalwechsel wartet und erst dann durch ein Event aktiv wird.

**Signalüberwachung durch Events**

Das folgende Python-Programm demonstriert diese Vorgehensweise: Mit `def` wird eine Callback-Funktion definiert, die immer dann aufgerufen werden soll, wenn die Taste gedrückt wird. Genau darum kümmert sich `add_event_detect`: Damit wird Pin 23 überwacht. Dank `add_event_callback` wird immer, wenn dessen Signalpegel von High auf Low fällt, die Funktion `switch_on` aufgerufen.

Das Programm soll laufen, bis es durch `[Strg]+[C]` beendet wird. Das ist der Zweck der Endlosschleife am Ende des Programms.

```
#!/usr/bin/python3
import RPi.GPIO as GPIO
import time, sys

# Pin-Nummern verwenden (nicht GPIO-Nummern!)
GPIO.setmode(GPIO.BOARD)
ledStatus = 0

# GPIO 21 = Input, 23 = Output
GPIO.setup(21, GPIO.IN)
GPIO.setup(23, GPIO.OUT)
GPIO.output(23, ledStatus)

# Funktion definieren, um bei Tastendruck den LED-Zustand zu ändern
def switch_on(pin):
    global ledStatus
    ledStatus = not ledStatus
    GPIO.output(23, ledStatus)
    return

# switch_on-Funktion aufrufen, wenn Signal von HIGH auf LOW wechselt
GPIO.add_event_detect(21, GPIO.FALLING)
GPIO.add_event_callback(21, switch_on)
```

```
# mit minimaler CPU-Lastung auf das Programmende durch Strg+C warten
try:
    while True:
        time.sleep(5)
except KeyboardInterrupt:
    GPIO.cleanup()
    sys.exit()
```

**Taster entprellen** Wenn Sie Schaltung und Programm nun ausprobieren, werden Sie feststellen, dass das Ein- und Ausschalten recht unzuverlässig funktioniert. Schuld daran ist ein Verhalten aller mechanischen Taster und Schalter: Diese prellen, d. h., ein Metallblättchen schlägt *mehrfach* gegen einen Kontaktpunkt und löst deswegen ganz rasch hintereinander *mehrere* Pegelwechsel am Input-Pin aus.

Für das Problem gibt es zwei einfache Lösungen: Entweder bauen Sie in Ihre Schaltung einen Kondensator ein, der während seiner Ladezeit das Prellen verhindert, oder Sie entscheiden sich für eine Software-Lösung und warten nach jedem Input-Event 200 ms, bevor Sie wieder Eingaben entgegennehmen. Die Software-Lösung ist in der GPIO-Bibliothek für Python bereits vorgesehen. Sie geben einfach als zusätzlichen Parameter bei `add_event_detect` die gewünschte Entprellzeit in Millisekunden an:

```
GPIO.add_event_detect(21, GPIO.FALLING, bouncetime=200)
```

Bei meinen Tests hat das leider nicht zuverlässig funktioniert. Eine entsprechende Funktion lässt sich aber mit wenig Aufwand selbst programmieren: Bei jedem Tastendruck merken Sie sich in `switch_on` die gerade aktuelle Zeit (Variable `lastTime`). Die nächste Veränderung des LED-Zustands führen Sie erst durch, wenn zumindest 200 ms vergangen sind.

```
# Änderungen im Programmcode
import datetime, time, sys
lastTime=datetime.datetime.now()
...
# bei Tastendruck LED-Zustand ändern
def switch_on( pin ):
    global ledStatus, lastTime
    now = datetime.datetime.now()
    if(now-lastTime > datetime.timedelta(microseconds=200000)):
        ledStatus = not ledStatus
        GPIO.output(23, ledStatus)
        lastTime = now
    return
```

## Temperatur messen

Das Ziel dieses Abschnitts ist die Messung der Umgebungstemperatur mit einem Temperatursensor. Der Raspberry Pi verfügt über keine Analog-Eingänge, an denen

die Spannung mit einem Analog/Digital-Wandler gemessen werden kann. Alle GPIO-Inputs sind digitale Eingänge, die nur zwischen 0 und 1 unterscheiden können. Deswegen empfiehlt es sich, zur einfachen Temperaturmessung ein Bauelement mit einem integrierten A/D-Wandler zu verwenden.

Bewährt hat sich für diese Aufgabe das Bauelement DS1820, das oft auch als 1-Wire-Thermometer angepriesen wird. Dieser Name ergibt sich daraus, dass diese Komponente nur über drei Anschlüsse verfügt: Zwei dienen zur Stromversorgung und der dritte dient zur Signalübertragung in Form eines binären Datenstroms. Der DS1820 kann sogar ohne explizite Versorgungsspannung betrieben werden und bezieht den Strom dann über die Signalleitung. Auf diese Schaltungsvariante gehe ich hier aber nicht ein. DS1820

Der DS1820 misst Temperaturen in einem Messbereich zwischen  $-55\text{ °C}$  und  $+125\text{ °C}$ . Die Temperatur wird als 9- oder 12-Bit-Zahl übertragen. Da jeder DS1820 mit einer eindeutigen Seriennummer ausgestattet ist, können mehrere Elemente parallel geschaltet und getrennt ausgewertet werden (über einen einzigen GPIO-Pin!). Beim Auslesen der Thermometer hilft ein eigenes Linux-Kernelmodul.

Es existieren verschiedene Varianten zum originalen DS1820: Am leichtesten erhältlich ist zumeist das Bauteil DS18S20, das fast vollständig kompatibel zum Original ist und als Grundlage für diesen Abschnitt diene. Ebenfalls populär ist die Variante DS18B20, bei dem die gewünschte Messgenauigkeit über ein Register programmiert werden kann. Eine kleinere Genauigkeit ermöglicht schnellere Messungen und reduziert den Stromverbrauch. Einige DS1820-Varianten werden zudem in einer wasserdichten Ausführung angeboten, die aber dieselben elektrischen Eigenschaften aufweist. Ein ausführliches Datenblatt sowie eine Beschreibung der Unterschiede zwischen den verschiedenen Varianten finden Sie hier:

<https://datasheets.maximintegrated.com/en/ds/DS18S20.pdf>  
<https://www.maximintegrated.com/app-notes/index.mvp/id/4377>

Abbildung 8.19 zeigt den Schaltungsaufbau. Ähnlich wie bei mechanischen Schaltern muss auch beim DS1820 ein Pull-up-Widerstand verwendet werden. Beachten Sie, dass Sie – im Gegensatz zu den bisherigen Schaltungen – den Signaleingang nicht frei wählen können. Pin 7 = GPIO 4 habe ich hier deswegen verwendet, weil der 1-Wire-Kerneltreiber diesen Signaleingang standardmäßig benutzt! Andere GPIOs sind möglich, müssen dann aber in `config.txt` explizit angegeben werden. Schaltung

#### Welcher Pin des DS1820 ist Pin 1?

Die Belegung der Pins des DS1820 geht aus dem Datenblatt hervor. Dabei müssen Sie beachten, dass das Bauelement in der Sicht von unten (*bottom view*) dargestellt ist.

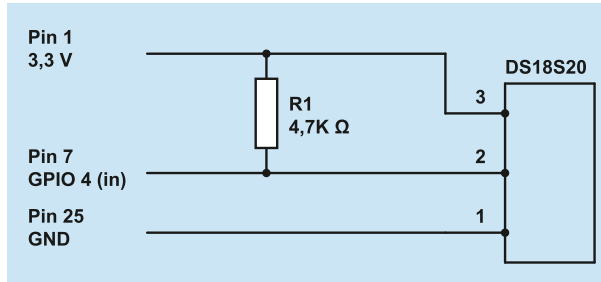


Abbildung 8.19 Schaltungsaufbau zur Messung der Umgebungstemperatur

### 1-Wire-Treiber aktivieren

Raspbian verwendet sogenannte Device Trees zur Beschreibung der Hardware des Raspberry Pi. Der Device Tree wird beim Starten des Minicomputers geladen. In Abhängigkeit vom Device Tree entscheidet der Linux-Kernel dann, welche Treiber (Module) er braucht. Für die Standardkomponenten des Raspberry Pi funktioniert dies aufgrund von Konfigurationsdateien im Verzeichnis `/boot` automatisch. Wenn Sie aber wie für dieses Beispiel zusätzliche Hardware nutzen möchten, dann müssen Sie den Device Tree entsprechend modifizieren.

Zu diesem Zweck sind die Parameter `dtparam=xxx` bzw. `dtoverlay=xxx` vorgesehen, die Sie in die Datei `/boot/config.txt` einfügen können. Eine Zusammenfassung der gängigsten Einstellungen gibt die Datei `/boot/overlays/README`.

In unserem Fall wollen wir den vorgegebenen Device Tree um den 1-Wire-Treiber ergänzen. Die erforderliche Zeile in `config.txt` sieht so aus:

```
# Ergänzung in /boot/config.txt
...
dtoverlay=w1-gpio-pullup
```

Nach einem Neustart des Raspberry Pi können Sie sich mit `lsmod` davon überzeugen, dass die 1-Wire-Kernelmodule geladen sind:

```
pi$ lsmod | grep w1
w1_therm    3584  0
w1_gpio     3657  0
wire        25219  2 w1_gpio,w1_therm
```

Anstatt `/boot/config.txt` manuell zu verändern, können Sie den Treiber auch mit **EINSTELLUNGEN • RASPBERRY-PI-KONFIGURATION • SCHNITTSTELLEN** aktivieren (Option **EINDRAHT-BUS**).

### Temperatur auslesen

Bevor Sie die Temperatur auslesen können, müssen Sie Pin 7 als Signaleingang konfigurieren:

```
root# gpio -1 mode 7 in
```

Der Datei `/sys/devices/w1_bus_master1/w1_master_slaves` können Sie nun die IDs aller angeschlossenen DS1820-Sensoren entnehmen. In diesem Beispiel gibt es nur einen Sensor, dessen ID-Code mit 10 beginnt:

```
pi$ cat /sys/devices/w1_bus_master1/w1_master_slaves
...
10-000802ae1551
```

Die Messdaten jedes Sensors liegen in einer Textdatei vor. Interessant ist die zweite Zeile: `t=nnn` gibt die Temperatur in Tausendstel Grad an, auch wenn die Messgenauigkeit geringer ist. Zum Messzeitpunkt betrug die Umgebungstemperatur also ca. 20,6 °C.

```
pi$ cat /sys/devices/w1_bus_master1/10-000802ae1551/w1_slave
 29 00 4b 46 ff ff 02 10 0c : crc=0c YES
 29 00 4b 46 ff ff 02 10 0c t=20625
```

## IR-Empfänger

Wenn Sie Ihren Raspberry Pi als Media-Center verwenden und dieses mit einer IR-Fernbedienung steuern möchten, benötigen Sie einen IR-Empfänger. Das gängigste Bauteil hierfür hat die Bezeichnung TSOP4838. Es ist Mitglied einer ganzen Familie von IR-Empfängern, die für unterschiedliche Frequenzen optimiert sind. Das Bauteil TSOP4838 ist auf 38 kHz abgestimmt, also auf den Frequenzbereich typischer TV-Fernbedienungen. Technische Details können Sie im Datenblatt nachlesen. Beachten Sie, dass die Belegung der Pins je nach TSOP-Variante unterschiedlich ist!

<https://www.maximintegrated.com/en/app-notes/index.mvp/id/4377>

Die Schaltung in [Abbildung 8.20](#) entspricht dem Vorschlag aus dem Datenblatt. Sowohl der Widerstand als auch der Kondensator sind optional; sie verbessern lediglich die elektrische Stabilität der Schaltung. Sie können also auch direkt die Pins 1, 2 und 3 des TSOP4838 mit den Pins 25 (GND), 12 und 1 (3,3 V) des J8-Headers des Raspberry Pi verbinden. Warum dient gerade Pin 12 als Signaleingang? Weil der mit ihm verbundene Eingang GPIO 18 standardmäßig vom `lirc`-Kerntreiber verwendet wird.

Schaltung

Damit der Kernel des Raspberry Pi weiß, dass er den IR-Treiber laden muss, ist wieder eine Veränderung in `/boot/config.txt` erforderlich:

IR-Treiber  
aktivieren

```
# Ergänzung in /boot/config.txt
...
dtoverlay=lirc-rpi
```

Nach einem Reboot vergewissern Sie sich mit `lsmod`, dass die IR-Treiber tatsächlich geladen wurden:

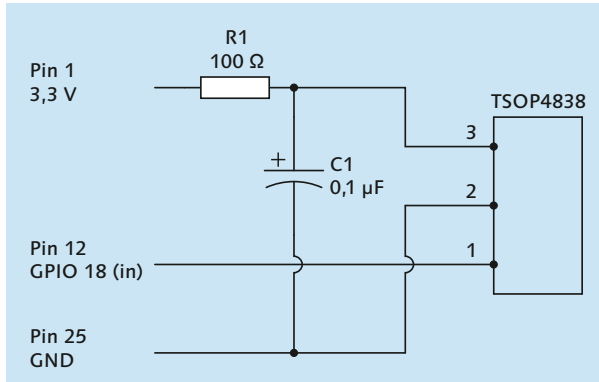


Abbildung 8.20 Ein einfacher IR-Empfänger

```
pi$ lsmod | grep lirc
lirc_rpi          6638  0
lirc_dev         8169  1 lirc_rpi
rc_core         16948  1 lirc_dev
```

#### IR-Signale verarbeiten

Jetzt können Sie im Textmodus mit dem Kommando `mode2` Signale der Device-Datei der IR-Schnittstelle auslesen. Danach brauchen Sie nur noch eine Fernbedienung auf den IR-Empfänger richten und einige Tasten drücken.

```
pi$ sudo apt install lirc
pi$ mode2 -d /dev/lirc0
space 1613
pulse 584
space 537
pulse 593
space 524
pulse 603
space 530
...
<Strg>+<C>
```

#### Hinweis

Sollte auf Ihrem Raspberry Pi der Dämon `lircd` laufen, müssen Sie ihn vor dem Test der Fernbedienung durch `mode2` beenden: `kill $(pidof lircd)`

`mode2` ist nur dazu gedacht, die Fernbedienung auszuprobieren. In aller Regel werden Sie den IR-Empfänger in Kombination mit Kodi und dem Hintergrundprogramm `lircd` verwenden. Tipps zur richtigen Konfiguration Ihrer Fernbedienung finden Sie in [Abschnitt 8.3](#), »Kodi und LibreELEC«.

## 8.5 Interna und Backups

Zum Abschluss des Raspberry-Pi-Kapitels fasst dieser Abschnitt einige technische Details zur Funktionsweise des Raspberry Pi zusammen und gibt Tipps zur Durchführung von Backups.

### Der Inhalt der Boot-Partition

Damit der Boot-Prozess des Raspberry Pi funktioniert, muss die erste Partition der SD-Karte eine VFAT-Partition sein und mehrere Dateien enthalten, deren Inhalt in [Tabelle 8.2](#) zusammengefasst ist. Im laufenden Betrieb finden Sie diese Dateien im Verzeichnis `/boot`. Normalerweise sollten Sie diese Dateien nicht anrühren. Die einzige Ausnahme ist die Datei `config.txt`, die im nächsten Abschnitt näher beschrieben wird.

| Datei                      | Inhalt   |
|----------------------------|--|
| <code>bootcode.bin</code>  | der Bootloader   |
| <code>start.elf</code>     | die Firmware des Grafikprozessors                                    |
| <code>*.dtb</code>         | Device-Tree-Blobs für verschiedene Raspberry-Pi-Modelle              |
| <code>overlay/*.dtb</code> | Device-Tree-Blobs für diverse Hardware-Komponenten                   |
| <code>config.txt</code>    | Textdatei zur Hardware-Konfiguration                                 |
| <code>kernel.img</code>    | der Linux-Kernel für den Raspberry Pi 1                              |
| <code>kernel7.img</code>   | der Linux-Kernel für den Raspberry Pi 2 und 3 (ARMv7- und ARMv8-CPU) |
| <code>cmdline.txt</code>   | Textdatei mit Parametern, die an den Kernel übergeben werden         |

**Tabelle 8.2** Die Boot-Dateien des Raspberry Pi in der Reihenfolge ihrer Verarbeitung

Der Boot-Vorgang verläuft auf dem Raspberry Pi vollkommen anders als auf einem gewöhnlichen PC. Es gibt kein BIOS, kein EFI und keinen GRUB. Wenn der Raspberry Pi eingeschaltet wird, ist vorerst nur der GPU-Core aktiv, also der Grafikteil der CPU. Die GPU lädt den ersten Teil des Bootloaders aus einem ROM (*1st stage bootloader*). Mit diesem Miniprogramm kann die CPU auf die SD-Karte zugreifen und dort `bootcode.bin` in den Cache der CPU lesen. Diese Datei enthält den restlichen Bootloader (*2nd stage bootloader*). Der Bootloader lädt nun `start.elf`. Das darin enthaltene Programm wertet die Device-Tree-Dateien sowie `config.txt` aus, liest `cmdline.txt` und `kernel.img` bzw. `kernel7.img` und startet schließlich den Kernel.

**Boot-Prozess**

## Die Konfigurationsdatei »config.txt«

Unabhängig davon, welches Betriebssystem Sie auf Ihrem Raspberry Pi installiert haben, bestimmt die Datei `config.txt` in der ersten Partition der SD-Karte viele Eckdaten der Konfiguration. Die Datei wird direkt beim Booten des Raspberry Pi ausgewertet. Dieser Abschnitt fasst die wichtigsten Einstellmöglichkeiten dieser Datei zusammen. Noch mehr `config.txt`-Details können Sie hier nachlesen:

[http://elinux.org/RPi\\_config.txt](http://elinux.org/RPi_config.txt)

**Parameter auslesen** Veränderungen an `config.txt` werden erst mit dem nächsten Neustart wirksam. Viele `config.txt`-Parameter können Sie im laufenden Betrieb mit `vcgencmd get_config` auslesen (siehe den folgenden Abschnitt).

**Speicher für das Grafiksystem** Der Raspberry Pi 2 und 3 verfügt jeweils insgesamt über 1 GiB RAM. Das RAM wird zwischen der CPU und dem Grafikprozessor geteilt. Die Aufteilung muss beim Start endgültig mit dem Parameter `gpu_mem` festgelegt werden, wobei Sie dem Grafiksystem 16, 64, 128 oder 256 MiB zuweisen können. 16 MiB sind für den normalen Betrieb ausreichend. Für grafikintensive Anwendungen (3D-Grafik, HD-Filme abspielen etc.) benötigt das Grafiksystem hingegen 128 MiB Speicher. Auch die Nutzung der Kamera erfordert ein Minimum von 128 MiB Grafikspeicher:

```
gpu_mem=128
```

**HDMI-Einstellungen** Normalerweise funktioniert die Grafik via HDMI ohne weitere Konfiguration. Nur wenn es Probleme gibt, können Sie mit diversen `hdmi_XXX`-Parametern bestimmte Einstellungen erzwingen. Das folgende Listing gibt hierfür einige Beispiele:

```
# HDMI-Ausgang verwenden, auch wenn kein Monitor erkannt wird
hdmi_force_hotplug=1

# HDMI-Auflösung 1400*1050 @ 60 Hz
# alle zulässigen hdmi_mode-Werte: siehe http://elinux.org/RPi_config.txt
hdmi_group=2
hdmi_mode=42

# Display-Drehung korrigieren
# 1 = 90 Grad, 2 = 180 Grad, 3 = 270 Grad
display_rotate=1

# HDMI-Signalstärke
# 0 = normal, 7 = maximal (Vorsicht!)
config_hdmi_boost=4

# HDMI-Signal im Energiesparmodus wirklich ausschalten
hdmi_blanking=1
```



Wenn Sie nicht wissen, welche Video-Modi Ihr Monitor unterstützt, führen Sie das Kommando `tvservice` aus. Die aufgelisteten Modi mit der Option `-m CEA` gelten für `hdmi_group=1`, die Modi mit der Option `-m DMT` für `hdmi_group=2`. Die folgenden Ergebnisse sind auf einem Monitor mit einer Auflösung von 1920 × 1200 Pixel entstanden:

```
pi$ vservice -m CEA
Group CEA has 7 modes:
    mode 1: 640x480 @ 60Hz 4:3, clock:25MHz progressive
    mode 2: 720x480 @ 60Hz 4:3, clock:27MHz progressive
    ...
    (native) mode 16: 1920x1080 @ 60Hz 16:9, clock:148MHz progressive
pi$ tvservice -m DMT
Group DMT has 13 modes:
    mode 4: 640x480 @ 60Hz 4:3, clock:25MHz progressive
    ...
    mode 68: 1920x1200 @ 60Hz 16:10, clock:154MHz progressive
```

Zum Raspberry Pi gibt es eine kleine Kamera, die über ein Flachbandkabel mit dem Minicomputer verbunden wird. Damit Sie diese Kamera nutzen können, sind zwei Ergänzungen in `config.txt` erforderlich. Diese Einstellungen können Sie auch mit dem Programm *Raspberry Pi Configuration* bzw. mittels `raspi-config` vornehmen.

Kamera nutzen

```
# Datei /boot/config.txt
# Kameramodul aktivieren
start_x=1
# 128 MiB RAM für den Video-Speicher reservieren
gpu_mem=128
# optional: LED der Kamera deaktivieren
disable_camera_led=1
```

Zusammen mit Raspbian werden zwei Kommandos installiert, um die Kamera zu nutzen: `raspistill -o name.jpg` macht ein Foto und speichert das Bild in der angegebenen Datei. `raspivid -o name.h264 -t 10000` erstellt ein 10 Sekunden langes Video und speichert es in einer H264-Datei. Beide Kommandos lassen sich durch diverse Optionen steuern, die in den `man`-Seiten dokumentiert sind.

Sogenannte Device Trees (siehe [Abschnitt 25.2](#)) beschreiben die Hardware des Raspberry Pi – inklusive der Erweiterungen, die Sie selbst für ein Bastelprojekt hinzugefügt haben. Damit der Kernel über Ihre Erweiterungen Bescheid weiß, müssen Sie in `config.txt` angeben, welche Komponenten oder Bussysteme Sie zusätzlich nutzen wollen.

Device-Tree-Einstellungen

Beachten Sie dass Sie Optionen ohne Leerzeichen aneinanderreihen müssen. Die folgenden Zeilen geben Syntaxbeispiele für die am häufigsten vorkommenden Anwendungen:

```

# Datei /boot/config.txt
# Beispiele zur Steuerung des Device-Tree-Systems
# (weitere Details siehe /boot/overlays/README)

# Audio-System aktivieren (lädt snd_bcm2835)
dtparam=audio=on

# SPI-Bus aktivieren
dtparam=spi=on

# I2C-Bus aktivieren
dtparam=i2c_arm=on

# HiFi-Berry DAC+ verwenden
dtoverlay=hifiberry-dacplus

# 1-Wire-Temperatursensor mit Standardeinstellungen verwenden
dtoverlay=w1-gpio-pullup

# 1-Wire-Temperatursensor verwenden, der mit
# GPIO X verbunden ist (per Default GPIO 4), und
# dabei den internen Pull-up-Widerstand aktivieren
dtoverlay=w1-gpio-pullup,gpiopin=X,pullup=y

# Echtzeituhr-Modell ds1307 verwenden
dtoverlay=rtc-i2c,ds1307

# IR-Empfänger verwenden
dtoverlay=lirc-rpi

```

**Overclocking** In der Vergangenheit war das Overclocking gewissermaßen ein Volkssport unter den Raspberry-Pi-Freaks. Das lag daran, dass bei den ersten Raspberry-Pi-Modellen die CPU-Frequenz sehr konservativ voreingestellt war und damit genug Spielraum nach oben bot. Bei aktuellen Raspberry-Pi-Modellen ist das aber nicht mehr der Fall. Das Overclocking ist zwar weiterhin möglich, aber definitiv nicht empfehlenswert.

Welche Werte für das Overclocking zweckmäßig sind, hängt von der Version Ihres Raspberry Pi ab:

- ▶ Die Modelle der Version 1 verwenden die CPU BCM2835 mit einer Grundfrequenz von bis zu 700 MHz. Ein Overclocking bis ca. 1 GHz ist sinnvoll.
- ▶ Der Raspberry Pi Zero sowie der Zero W verwenden dieselbe CPU, aber bereits eine Grundfrequenz von 1 GHz. Overclocking wird nicht empfohlen.
- ▶ Der Raspberry Pi 2, Modell B, verwendet die CPU BCM2836 mit einer Taktfrequenz von bis zu 900 MHz. Hier ist ein moderates Overclocking bis ca. 1 GHz denkbar.

- Der Raspberry Pi 3, Modell B, setzt schließlich die CPU BCM2837 ein. Die Taktfrequenz beträgt bis zu 1,2 GHz. Eine Steigerung darüber hinaus ist nicht empfehlenswert.

Weitere Details und Tipps können Sie auf den folgenden Seiten nachlesen:

<https://raspberrypi.stackexchange.com/questions/38897>

<http://www.jackenhack.com/raspberry-pi-3-overclocking>

Aus technischer Sicht ist das Overclocking unkompliziert. Sie müssen lediglich einige Zeilen in `config.txt` ändern und Ihren Raspberry Pi neu starten. Am einfachsten führen Sie diese Änderungen mit dem Raspberry-Pi-Konfigurationsprogramm durch. Die folgenden Einstellungen sind für einen Raspberry Pi 2 gedacht und bewirken, dass er bei Bedarf mit einer Taktfrequenz von 1 GHz läuft statt normalerweise mit 900 MHz:

```
# Overclocking-Einstellungen in /boot/config.txt
# für einen Raspberry Pi 2

# Maximale Taktfrequenz der CPU in MHz. Default 900 MHz.
arm_freq=1000

# Taktfrequenzen der Grafik- und Encoding-Cores. Default 250 MHz.
# GPU = Graphics Processing Unit.
core_freq=500

# RAM-Taktfrequenz. Default 400 MHz.
sdram_freq=500

# Die Spannung um n*0.025 V anheben. Nur bei starkem Overclocking erforderlich.
# Maximal 6 mit force_turbo=0 (entspricht 0.150 V),
# maximal 8 mit force_turbe=1 (entspricht 0.2V).
over_voltage=2
over_voltage_sdram=0
```

Das Overclocking erfolgt normalerweise dynamisch, d. h. nur dann, wenn die CPU- oder GPU-Leistung tatsächlich benötigt wird. In den Ruhezeiten laufen CPU und GPU in den Defaulttaktfrequenzen. `force_turbo=1` verhindert die Taktabsenkung und bewirkt, dass die CPU/GPU immer mit der angegebenen Frequenz getaktet wird. Das ist stabiler, die CPU/GPU wird aber schneller heiß. Die gerade aktuelle CPU-Frequenz und -Temperatur können Sie wie folgt auslesen (in kHz bzw. in Milligrad):

```
pi$ cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq
600000
pi$ cat /sys/class/thermal/thermal_zone0/temp
39007
```

**Video-Codex** Die Encoding-Komponenten der GPU können die Video-Decodierung unterstützen. Für manche Codex ist das standardmäßig der Fall, für andere muss diese Funktion mit einem kostenpflichtigen Lizenzschlüssel freigeschaltet werden. Momentan sind zwei derartige Schlüssel vorgesehen: je einer für den MPG-2- und für den VC-1-Decoder. Die folgenden Werte sind natürlich nur Muster. Die Schlüssel müssen zur ID Ihrer CPU passen.

```
decode_MPG2=0x12345678
decode_WVC1=0x9abcdef0
```

**Boot-Vorgang** Die folgenden Parameter modifizieren den Boot-Vorgang:

```
# Parameter für den Kernel (anstelle der Datei cmdline.txt)
cmdline=xxx
# Name der Kernel-Datei (default: kernel.img)
kernel=filename
# Wartezeit, bevor der Kernel geladen wird (in Sekunden, default 1).
boot_delay=2
```

### Das Kommando »vcgencmd«

Mit dem Kommando `vcgencmd` können Sie diverse Statusinformationen der CPU auslesen. `vcgencmd commands` liefert eine Liste aller bekannten Kommandos. Die folgenden Beispiele zeigen einige Anwendungen:

```
pi$ vcgencmd measure_clock arm           (CPU-Frequenz)
frequency(45)=600000000

pi$ vcgencmd measure_clock core         (Frequenz der Grafik-Cores)
frequency(1)=250000000

pi$ vcgencmd measure_volts core        (Spannung der Grafik-Cores)
volt=1.20V

pi$ vcgencmd measure_temp              (CPU-Temperatur)
temp=47.1°C

pi$ vcgencmd codec_enabled H264        (Steht Codec xy zur Verfügung?)
H264=enabled

pi$ vcgencmd get_config int            (Liste aller aktiven Integer-Optionen)
disable_overscan=1
temp_limit=85
force_pwm_open=1

pi$ vcgencmd get_config str            (Liste aller aktiven String-Optionen)
```

Noch mehr Anwendungsbeispiele finden Sie hier:

[http://elinux.org/RPI\\_vcgencmd\\_usage](http://elinux.org/RPI_vcgencmd_usage)

## Backups

Grundsätzlich gelten für den laufenden Betrieb eines Minicomputers dieselben Backup-Strategien wie für einen gewöhnlichen Computer (siehe [Kapitel 36](#)): Wenn Sie auf Ihrem Minicomputer veränderliche Daten speichern, sollten Sie diese regelmäßig sichern. Im Idealfall hat der Minicomputer eine Verbindung zu einem NAS-Gerät; dann bietet sich die Programmierung eines kleinen Backup-Scripts an, das einmal täglich alle relevanten Daten in einem Netzwerkverzeichnis sichert.

Davon losgelöst ist es zweckmäßig, hin und wieder eine Sicherungskopie der ganzen SD-Karte zu erstellen. Dieses Backup kann dann jederzeit auf eine neue, zumindest gleich große SD-Karte übertragen werden. Die folgenden Beispiele gehen wieder davon aus, dass das Device der SD-Karte `/dev/sdb` lautet. Passen Sie die Device-Angaben entsprechend an!

SD-Karte sichern

```
root# umount /dev/sdb?
root# dd if=/dev/sdb of=backup.img bs=4M
```

Das Auslesen großer SD-Karten dauert leider ziemlich lange. Wenn Sie in dieser Zeit ein Feedback wünschen, setzen Sie statt `dd` das Kommando `dcfldd` ein. Optional können Sie die zu sichernden Daten auch gleich komprimieren:

```
root# dcfldd if=/dev/sdb bs=4M | gzip > backup.img.gz
```

### Riesen-Backup, obwohl die SD-Karte halb leer ist?

Die Backup-Datei wird möglicherweise trotz Komprimierung größer ausfallen als erwartet – auch dann, wenn nur ein kleiner Teil der SD-Karte tatsächlich mit Daten gefüllt ist. Das liegt daran, dass in jedem Fall der gesamte Datenträger blockweise ausgelesen wird, egal ob diese Blöcke vom Dateisystem genutzt werden oder nicht. Oft enthalten die Blöcke Zufallsdaten, z. B. Überreste einer früheren Nutzung der Karte in einer Digitalkamera, die schwer zu komprimieren sind.

In die umgekehrte Richtung sehen die Kommandos wie folgt aus. Beachten Sie, dass dabei der gesamte Inhalt des Datenträgers `/dev/sdb` überschrieben wird!

SD-Karte wiederherstellen

```
root# umount /dev/sdb?
root# dd if=backup.img of=/dev/sdb bs=4M
```

Wenn Sie das Backup-Image komprimiert haben, gehen Sie so vor:

```
root# gunzip -c backup.img.gz | dd of=/dev/sdb bs=4M
```

## 8.6 Wenn es Probleme gibt

Nicht immer klappt alles auf Anhieb. Wenn Sie Pech haben, stürzt Ihr Raspberry Pi nach wenigen Sekunden ab, bleibt hängen, zeigt unverständliche Fehlermeldungen an oder – was sicherlich der unangenehmste Fall ist – liefert am Bildschirm überhaupt kein Bild. Dann ist eine Diagnose natürlich besonders schwierig. Dieser Abschnitt fasst einige Tipps zusammen, was Sie in solchen Fällen tun können.

**Stromversorgung** Wenn man Forenberichten glauben darf, ist eine unzureichende Stromversorgung die bei Weitem häufigste Fehlerursache. Der Raspberry Pi 3 benötigt laut Spezifikation an sich rund 1000 mA Strom – das ist wesentlich mehr, als typische Handy-Netzteile mit Micro-USB-Kabel liefern können. Sparen Sie daher nicht beim Netzteil, sondern kaufen Sie eines, das zumindest 1 A Strom liefern kann. Wenn Ihr Raspberry Pi diverse USB-Geräte mit Strom versorgen muss, kann der Strombedarf sogar auf über 2,5 A ansteigen.

**SD-Karte** An zweiter Stelle in der Hitliste der Probleme mit dem Raspberry Pi stehen SD-Karten. Es gibt Modelle, die nicht zum Raspberry Pi kompatibel sind, auch wenn diese Karten in einer Kamera oder im Kartenslot eines Notebooks problemlos funktionieren. Versuchen Sie es einfach mit einem anderen Modell, und werfen Sie vor dem Kauf einen Blick auf die folgende Seite des Embedded Linux Wikis:

[http://elinux.org/RPi\\_SD\\_cards](http://elinux.org/RPi_SD_cards)

Nicht immer ist die Karte an sich schuld. Eine mögliche Fehlerursache kann auch sein, dass Sie das Linux-Image nicht fehlerfrei auf die SD-Karte übertragen haben. Das Problem äußert sich in der Regel dadurch, dass der Boot-Prozess von diversen *Authentication*-Warnungen unterbrochen wird und schließlich ganz stoppt.

Abhilfe: Kopieren Sie die Image-Datei nochmals auf die SD-Karte. Vergleichen Sie vorher die SHA1-Prüfsumme der ZIP-Datei mit dem auf der Download-Seite angegebenen Wert. Achten Sie unbedingt darauf, dass keine Partition der SD-Karte in den Verzeichnisbaum eingebunden ist! Wenn die SD-Karte den Device-Namen `/dev/sdb` hat, müssen Sie also vor dem `dd`-Kommando `umount /dev/sdb?` ausführen.

### SD-Karten sind keine Festplatten!

Generell sind SD-Karten – unabhängig von ihrem Preis – leider oft Billigprodukte, deren Lebensdauer und Stabilität selten mit Festplatten oder SSDs mithalten kann. Überlegen Sie sich eine Backup-Strategie, vermeiden Sie nach Möglichkeit stark I/O-lastige Anwendungen, bzw. speichern Sie Ihre Daten auf einem NAS-Speichergerät.

Besonders schwierig ist die Fehlersuche, wenn Ihr Monitor oder Fernseher gar kein Bild zeigt. Klären Sie zuerst die naheliegenden Fragen: Funktioniert die Stromversorgung? Wenn im Raspberry Pi nicht zumindest eine rote Diode leuchtet, bekommt der Computer keinen bzw. zu wenig Strom. Ist das Kabel oder der Bildschirm schuld? Wenn möglich, versuchen Sie es mit einem anderen HDMI-Kabel bzw. mit einem anderen Monitor/Fernseher.

Wenn das alles nichts hilft, sollten Sie versuchen, in der Datei `config.txt` auf der ersten Partition der SD-Karte Veränderungen vorzunehmen. Diese Datei wird vom Raspberry Pi unmittelbar nach dem Start gelesen und enthält unter anderem einige Parameter, die das HDMI-Signal und die Grafikauflösung betreffen. Standardmäßig enthält diese Datei nur eine einzige für das Grafiksystem relevante Anweisung:

```
# Datei config.txt (Defaulteinstellung)
disable_overscan=1
```

Bei Display-Problemen sollten Sie es mit dieser Einstellung versuchen:

```
# Datei config.txt
hdmi_force_hotplug=1
config_hdmi_boost=4
hdmi_group=2
hdmi_mode=4
disable_overscan=0
```

Ihr Raspberry Pi verwendet nun eine Auflösung von nur  $640 \times 480$  Pixel, wobei der tatsächlich nutzbare Bereich wegen eines schwarzen Overscan-Bereichs an den Rändern noch etwas kleiner ist. Wirklich zufriedenstellend arbeiten können Sie so nicht, aber immerhin lässt sich auf diese Weise sicherstellen, dass Ihr Minicomputer an sich funktioniert.

### config.txt sicher ändern

Zur Veränderung der Datei `config.txt` unterbrechen Sie die Stromversorgung zum Raspberry Pi und stecken die SD-Karte in den Slot Ihres regulären Computers. Dort können Sie die Datei `config.txt` mit einem beliebigen Editor ändern. Speichern Sie die Veränderungen, werfen Sie die SD-Karte im Dateimanager aus, stecken Sie sie wieder in den Raspberry Pi, und stellen Sie dessen Stromversorgung wieder her.

Sobald Ihr Raspberry Pi läuft, können Sie die Datei `config.txt` auch im laufenden Betrieb ändern. Sie finden die Datei im `/boot`-Verzeichnis. Änderungen werden erst nach einem Neustart wirksam.

Wenn auf dem Monitor nach dem Einschalten des Raspberry Pi nur ein buntes Farbmuster zu sehen ist (links oben Rot, rechts unten Hellblau), deutet das darauf hin, dass die Datei `start.elf` von der ersten Partition der SD-Karte gelesen werden konnte, dass aber der Linux-Kernel aus der Datei `kernel.img` nicht gelesen oder ausgeführt werden kann.

**Status-LEDs** Auf dem Raspberry Pi befinden sich einige Leuchtdioden, die verraten, in welchem Zustand sich der Minicomputer gerade befindet. Die Anzahl und der Ort der LEDs variieren je nach Modell. Für den Raspberry Pi 2 und 3 gilt:

- ▶ **Ethernet-LEDs:** Es gibt zwei Leuchtdioden für den Ethernet-Status. Diese befinden sich direkt in der Ethernet-Buchse. Die grüne LED leuchtet, wenn eine Netzwerkverbindung besteht; sie flackert, wenn gerade Daten übertragen werden. Die gelbe LED gibt an, dass der 100-Mbit/s-Modus aktiv ist.
- ▶ **Stromversorgungs-LED (rot):** Diese LED befindet sich am Beginn der GPIO-Leiste nahe Pin 1. Sie leuchtet, wenn der Raspberry Pi mit der Stromversorgung verbunden ist – und zwar nur dann, wenn die erforderliche Versorgungsspannung von 4,7 V zur Verfügung steht.
- ▶ **Status-LED (grün):** Diese LED befindet sich neben der gerade erwähnten Stromversorgungs-LED. Sie leuchtet, wenn Daten von oder zur SD-Karte übertragen werden. Falls diese LED nach dem Einschalten nur schwach leuchtet, findet der Raspberry Pi auf der SD-Karte die zum Booten erforderlichen Dateien nicht bzw. kann überhaupt nicht mit der SD-Karte kommunizieren.

Wenn der Raspberry Pi nur einen Teil der Boot-Dateien lesen kann, blinkt die grüne LED in einem speziellen Muster:

- Dreimal Blinken bedeutet, dass die Datei `start.elf` nicht gefunden wurde.
- Viermal Blinken bedeutet, dass `start.elf` nicht ausgeführt werden kann.
- Siebenmal Blinken bedeutet, dass `kernel.img` nicht gefunden wurde.

**Andere Probleme** Als *die* ultimative Referenz bei Hardware-Problemen mit dem Raspberry Pi gilt die folgende Website:

[http://elinux.org/R-Pi\\_Troubleshooting](http://elinux.org/R-Pi_Troubleshooting)



TEIL III

# Linux-Grundlagen



# Kapitel 9

## Terminalfenster und Konsolen

Bis jetzt habe ich Ihnen Linux in erster Linie als Desktop-System präsentiert. Sie haben diverse Programme kennengelernt, die vielleicht ein wenig anders aussehen als unter Windows oder macOS, aber letztlich denselben Zweck erfüllen und ähnlich zu bedienen sind. Der Umgang mit Linux endet allerdings nicht an dieser Stelle. Es gibt quasi noch eine andere Seite von Linux, die auf den ersten Blick abschreckend wirken mag: Erfahrene Linux-Anwender führen in Terminalfenstern bzw. Textkonsolen Kommandos aus und erhalten die Resultate wiederum in Textform. Die Maus spielt nur noch eine Nebenrolle, grafische Benutzeroberflächen sind passé.

Vom Desktop ins  
Terminal

Wenn Sie einmal mit der Arbeit in Terminalfenstern vertraut sind, können Sie dort viele Aufgaben effizient ausführen: Sie können Linux-Kommandos miteinander verknüpfen, im Hintergrund ausführen, automatisch ausführen, in kleinen Programmen (Scripts) automatisieren etc. All diese Möglichkeiten stehen Ihnen auch dann zur Verfügung, wenn Sie nicht lokal am Rechner sitzen, sondern nur über eine Netzwerkverbindung verfügen.

Reine Büroanwender werden für Terminalfenster seltener Verwendung finden als Programmierer oder Netzwerkadministratoren. Auf jeden Fall gehört die Arbeit im Terminal zum elementaren Handwerkszeug jedes Anwenders, der Linux richtig kennenlernen will. Das merken Sie spätestens dann zum ersten Mal, wenn das Grafiksystem wegen einer Fehlkonfiguration nicht funktioniert oder wenn Sie Ihren externen Root-Server administrieren möchten.

Dieses Kapitel gibt lediglich einen ersten Überblick über Arbeitstechniken in Terminalfenstern bzw. Konsolen. Für die Ausführung der Programme im Terminal ist eine sogenannte Shell verantwortlich. Unter Linux stehen mehrere Shells zur Auswahl. Am häufigsten kommt die `bash` zum Einsatz, deren Grundfunktionen Thema des nächsten Kapitels sind.

Querverweise

Die weiteren Kapitel stellen dann diverse Linux-Kommandos näher vor. Diese dienen beispielsweise zur Verwaltung des Dateisystems (`ls`, `cp`, `mv`, `ln`, `rm` etc.), zur Suche nach Dateien (`find`, `grep`, `locate`), zur Steuerung von Netzwerkfunktionen (`ping`, `ip`, `ssh`) etc. Nebenbei werden Sie eine Menge Linux-Grundlagen lernen.

## 9.1 Textkonsolen und Terminalfenster

**Textkonsolen** Microsoft Windows nutzen Sie ausschließlich im Grafikmodus. Linux können Sie dagegen auch in sogenannten Textkonsolen nutzen. Bei den meisten Distributionen stehen sechs Textkonsolen zur Verfügung. Der Wechsel zwischen diesen Textkonsolen erfolgt mit `[Alt]+[F1]` für die erste Konsole, `[Alt]+[F2]` für die zweite etc. Wenn der Rechner bereits im Grafikmodus läuft, führt `[Strg]+[Alt]+[F1]` in die erste Textkonsole und `[Alt]+[F7]` zurück in den Grafikmodus. Bei einigen Distributionen, unter anderem bei CentOS und Fedora, sind statt der siebten die erste Konsole bzw. die ersten zwei Konsolen für den Grafikmodus reserviert.

Bevor Sie in einer Textkonsole arbeiten können, müssen Sie sich einloggen. Wenn Sie mit der Arbeit fertig sind oder wenn Sie sich unter einem anderen Namen anmelden möchten, müssen Sie sich wieder ausloggen. Dazu drücken Sie einfach `[Strg]+[D]`.

Sie können in der einen Konsole ein Kommando starten, und während dieses Kommando läuft, können Sie in der zweiten Konsole etwas anderes erledigen. Sie können sich auch in einer Konsole als `root` anmelden, um administrative Aufgaben zu erledigen, während Sie in der anderen Konsole unter Ihrem normalen Login-Namen eine Datei editieren. Jede Konsole läuft also vollkommen unabhängig von den anderen.

| Tastenkürzel  | Funktion   |
|---|--|
| <code>[Strg]+[Alt]+[Fn]</code>                      | vom Grafikmodus in die Textkonsole <i>n</i> wechseln   |
| <code>[Alt]+[Fn]</code>                             | von einer Textkonsole in eine andere Textkonsole <i>n</i> wechseln   |
| <code>[Alt]+[F7]</code>                             | zurück in den Grafikmodus wechseln ( <code>[Alt]+[F1]</code> bei CentOS bzw. <code>[Alt]+[F2]</code> bei Fedora) |
| <code>[Alt]+[→]</code> / <code>[Alt]+[←]</code>     | in die vorige/nächste Textkonsole wechseln   |
| <code>[⇧]+[Bild↑]</code> / <code>[⇧]+[Bild↓]</code> | vorwärts/rückwärts blättern  |
| <code>[Strg]+[Alt]+[Entf]</code>                    | Linux beenden (nur in Textkonsolen, führt shutdown aus, Vorsicht!)   |

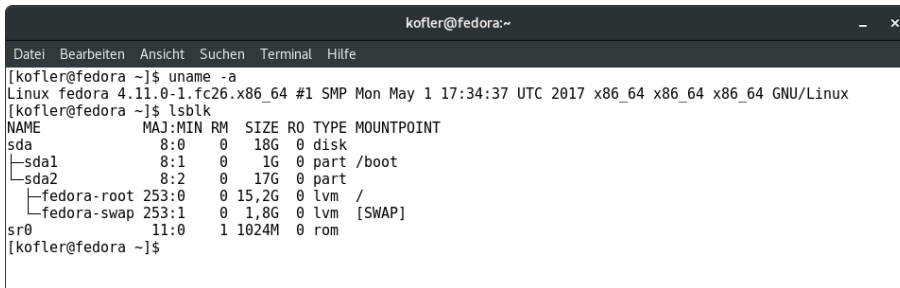
**Tabelle 9.1** Tastenkürzel zum Aktivieren von Textkonsolen

Mit `[⇧]+[Bild↑]` und `[⇧]+[Bild↓]` scrollen Sie den Bildschirminhalt einer Textkonsole auf und ab. Auf diese Weise können Sie die Ergebnisse der zuletzt ausgeführten Programme nochmals ansehen, auch wenn sie bereits aus dem sichtbaren Bildschirmbereich hinausgeschoben wurden.

### Terminalfenster (Shell-Fenster)

Natürlich müssen Sie aus dem Grafikmodus nicht in eine Textkonsole wechseln, nur um Kommandos auszuführen. Für diesen Zweck reicht ein in einem Fenster ausgeführtes Terminalprogramm vollkommen aus (siehe [Abbildung 9.1](#)). Zur Bezeichnung

eines Terminals sind mitunter auch die Begriffe »Konsolenfenster« oder »Shell-Fenster« gebräuchlich.



```

kofler@fedora:~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
[kofler@fedora ~]$ uname -a
Linux fedora 4.11.0-1.fc26.x86_64 #1 SMP Mon May 1 17:34:37 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
[kofler@fedora ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0  18G  0 disk
├─sda1       8:1    0   1G  0 part /boot
├─sda2       8:2    0  17G  0 part
├─fedora-root 253:0  0 15,2G 0 lvm  /
├─fedora-swap 253:1  0  1,8G  0 lvm  [SWAP]
sr0         11:0    1 1024M  0 rom
[kofler@fedora ~]$

```

Abbildung 9.1 Ein Terminalfenster

Je nach Distribution und Desktop-System stehen unterschiedliche Terminalfenster zur Auswahl, beispielsweise gnome-terminal (Gnome), konsole (KDE) oder xterm. Auch das Menükommando zum Starten eines Terminalfensters variiert je nach Distribution – hier ein paar Beispiele:

CentOS/RHEL (Gnome): ANWENDUNGEN • HILFSPROGRAMME • TERMINAL

Fedora (Gnome):  terminal




openSUSE (KDE): ANWENDUNGEN • SYSTEM • TERMINAL


Ubuntu (Unity):  terminal

In Terminalfenstern können Sie wie in einer Textkonsole arbeiten. Der einzige Unterschied besteht darin, dass Sie dank einer Bildlaufleiste bequemer durch die bisherigen Ausgaben scrollen können.

Innerhalb von Textkonsolen bzw. Terminalfenstern helfen diverse Tastenkürzel bei der effizienten Eingabe von Kommandos. [Tabelle 9.2](#) fasst die wichtigsten Kürzel zusammen. Sie gelten nur, wenn Sie die bash in der Standardkonfiguration als Shell verwenden, was bei den meisten Distributionen der Fall ist. Wenn Sie unter Gnome in einem Terminalfenster arbeiten, sollten Sie BEARBEITEN • TASTENKOMBINATIONEN ausführen und die Option ALLE MENÜKÜRZELBUCHSTABEN AKTIVIEREN deaktivieren.

Wichtige  
Tastenkürzel

Insbesondere die Kommandoerweiterung mit  spart eine Menge Tipparbeit. Sie brauchen nur die ersten Buchstaben eines Kommandos oder einer Datei angeben. Anschließend drücken Sie . Wenn der Dateiname bereits eindeutig erkennbar ist, wird er vollständig ergänzt, sonst nur so weit, bis sich mehrere Möglichkeiten ergeben. Ein zweimaliges Drücken von  bewirkt, dass eine Liste aller Dateinamen angezeigt wird, die mit den bereits eingegebenen Anfangsbuchstaben beginnen. Im Detail ist dieser Mechanismus in [Abschnitt 10.3](#), »Kommandoeingabe«, beschrieben.

| Tastenkürzel  | Funktion   |
|---|--|
| <b>Strg</b> + <b>A</b>  | Cursor an den Zeilenanfang (wie <b>Pos1</b> )      |
| <b>Strg</b> + <b>C</b>  | Programm abbrechen                                 |
| <b>Strg</b> + <b>E</b>  | Cursor an das Ende der Zeile (wie <b>Ende</b> )    |
| <b>Strg</b> + <b>K</b>  | Zeile ab Cursor löschen                            |
| <b>Strg</b> + <b>Y</b>  | zuletzt gelöschten Text wieder einfügen            |
| <b>Strg</b> + <b>Z</b>  | Programm unterbrechen (Fortsetzung mit fg oder bg) |
|  | Datei- und Kommandonamen vervollständigen          |
| <b>↑</b> / <b>↓</b>   | durch die bisher ausgeführten Kommandos blättern   |


**Tabelle 9.2** Tastenkürzel zur Kommandoeingabe in der bash

**Maus** Die Maus spielt in Textkonsolen bzw. in Terminalfenstern nur eine untergeordnete Rolle. Sie können sie *nicht* dazu verwenden, um die aktuelle Cursorposition zu verändern! Das gelingt nur mit den Cursortasten. Die Funktion der Maus beschränkt sich darauf, mit der linken Maustaste Text zu kopieren und diesen dann mit der mittleren Maustaste an der aktuellen Cursorposition wieder einzufügen.

Wenn Sie mit einer Maus oder einem Touchpad ohne mittlere Taste arbeiten, müssen Sie wie in anderen Betriebssystemen den Text zum Einfügen zuerst mit einem Tastenkürzel in die Zwischenablage kopieren und mit einem zweiten Kürzel einfügen. Wie umständlich! Jetzt verstehen Sie vielleicht, warum viele Linux-Freaks beim Kauf der Maus bzw. eines Notebooks darauf achten, dass das Mousrad als Taste genutzt werden kann bzw. dass das Touchpad über eine dritte Taste verfügt.

### Die Maus in Textkonsolen

In Textkonsolen, also beim Arbeiten *ohne* grafische Benutzeroberfläche, kann eigentlich keine Maus verwendet werden – es sei denn, Sie installieren und starten das Programm `gpm`. Dann können Sie die Maus auch in Textkonsolen zum Kopieren von Text nutzen.

**Kommandos ausführen** Zum Ausführen von Kommandos geben Sie in der Textkonsole oder im Shell-Fenster einfach den Kommandonamen, eventuell einige Parameter und schließlich  ein. Das Kommando `ls` liefert eine Liste der Dateien und Unterverzeichnisse im aktuellen Verzeichnis.

```
user$ ls -l
-rw----- 1 user users 17708403 19. Mai 10:35 20060519_DN.pdf
-rw----- 1 user users 506614 29. Jun 12:11 anbot-katzbauer.pdf
```

```
drwxrwxr-x 3 user users 4096 13. Apr 11:31 bak
drwxrwxr-x 2 user users 4096 18. Jul 15:03 bin
-rw-r--r-- 1 user users 243571 3. Jul 09:14 DB20078.jpg
drwxr-xr-x 2 user users 4096 7. Apr 10:59 Desktop
...
```

Aus dem obigen Beispiel geht auch hervor, wie in diesem Buch die Kommando­eingabe und das Ergebnis dargestellt werden: `user$` am Beginn der ersten Zeile bedeutet, dass das Kommando von einem gewöhnlichen Benutzer ausgeführt wurde. Wenn in der ersten Textspalte stattdessen `root#` angegeben ist, wurde das Kommando hingegen von `root` ausgeführt (also vom Systemadministrator). `user$` bzw. `root#` gilt als Eingabeprompt. Diese Zeichen werden am Beginn jeder Eingabezeile automatisch angezeigt. Sie dürfen diese Zeichen *nicht* mit eingeben! Generell gilt, dass nur die fett hervorgehobenen Zeichen einzugeben sind! Auf Ihrem Rechner wird statt `user$` bzw. `root#` möglicherweise ein anderer Text angezeigt, der oft das aktuelle Verzeichnis und/oder den Rechnernamen enthält. Auf diese Angaben verzichte ich in diesem Buch aus Gründen der Übersichtlichkeit.

Manchmal reicht der Platz in diesem Buch nicht aus, um ein Kommando in einer einzigen Zeile abzdrukken. In solchen Fällen wird das Kommando über mehrere Zeilen verteilt, die durch das Zeichen `\` getrennt sind. Das sieht dann beispielsweise so aus:

```
user$ gconftool-2 --set "/apps/panel/toplevels/top_panel_screen0/monitor" \
      --type integer "0"
```

Sie können dieses Kommando nun ebenfalls zweizeilig eingeben – dann müssen Sie die erste Zeile wie im Buch mit `\` abschließen. Sie können die zwei Zeilen aber auch einfach zusammenziehen: Dann entfällt das Zeichen `\`.

Sie können Kommandos auch im Hintergrund ausführen. Das bedeutet, dass Sie nicht auf das Programmende zu warten brauchen, sondern sofort weiterarbeiten können. Dazu geben Sie am Ende der Kommandozeile das Zeichen `&` an. Diese Vorgehensweise empfiehlt sich vor allem, wenn Sie aus einer Konsole heraus ein Programm mit grafischer Benutzeroberfläche starten (z. B. `firefox &`).

**Kommandos im Hintergrund ausführen**

Es ist unter Linux unüblich, als `root` zu arbeiten, also mit Systemadministratorrechten. Wenn Sie als gewöhnlicher Benutzer eingeloggt sind, gibt es verschiedene Wege, einzelne Kommandos mit `root`-Rechten auszuführen. Bei vielen Distributionen führen Sie im Terminalfenster einfach `su -l` aus und loggen sich so vorübergehend als `root` ein. Dazu müssen Sie natürlich das `root`-Passwort kennen. Nun können Sie als `root` textorientierte Kommandos ausführen. `exit` oder `Ctrl+D` führt wieder zum ursprünglichen User zurück. Unter Ubuntu kommt statt `su` das Kommando `sudo` zum Einsatz.

**Arbeiten als root**

Tipps dazu, wie Sie die Kommandoausführung vom Vordergrund in den Hintergrund verschieben, wie Sie eine Liste aller aktiven Kommandos (Prozesse) ermitteln etc., folgen in [Kapitel 12](#), »Prozessverwaltung«.

## 9.2 Textdateien anzeigen und editieren

**less** Unter KDE oder Gnome können Sie Textdateien direkt im Dateimanager lesen. Mit der rechten Maustaste können Sie die Datei auch in einem Editor öffnen. Wenn Sie hingegen in einer Textkonsole oder in einem Terminalfenster arbeiten, verwenden Sie zum Betrachten von Dateien am besten das Kommando `less`. Sie können das Kommando auch hinter andere Kommandos stellen, um deren oft sehr lange Ausgaben in Ruhe seitenweise zu lesen oder darin nach Text zu suchen (siehe [Tabelle 9.3](#)):

```
user$ less datei      (seitenweise Anzeige der Datei)
user$ ls -l | less    (seitenweise Anzeige des Dateiverzeichnisses)
user$ ps ax | less    (seitenweise Anzeige der Prozessliste)
```

| Tastenkürzel                                     | Funktion                                      |
|--|---|
| Cursortasten                                     | Text nach oben oder unten verschieben         |
| <code>Pos1</code> , <code>Ende</code>            | an den Beginn / das Ende des Textes springen  |
| <code>G</code> , <code>⇧</code> + <code>G</code> | an den Beginn / das Ende des Textes springen  |
| <code>/</code> muster <code>↵</code>             | vorwärts suchen                               |
| <code>?</code> muster <code>↵</code>             | rückwärts suchen                              |
| <code>N</code>                                   | Suche vorwärts wiederholen ( <i>next</i> )    |
| <code>⇧</code> + <code>N</code>                  | Suche rückwärts wiederholen                   |
| <code>Q</code>                                   | beenden ( <i>quit</i> )                       |
| <code>H</code>                                   | Hilfetext mit weiteren Tastenkürzeln anzeigen |

**Tabelle 9.3** less-Tastenkürzel

Auf manchen Mini-Linux-Systemen, z. B. in Embedded-Geräten wie NAS-Festplatten, fehlt das Kommando `less`. Möglicherweise ist stattdessen der `less`-Vorgänger `more` installiert. Andernfalls können Sie mit `cat` den gesamten Inhalt einer Textdatei ausgeben, also ohne seitenweises Blättern. Wenn Sie nur die letzten Zeilen lesen möchten, z. B. bei einer Logging-Datei, verwenden Sie `tail`.

**Präprozessor** Bei den meisten Distributionen kann `less` nicht nur einfache Textdateien anzeigen, sondern auch komprimierte Dateien, den Inhalt von `tar`-Archiven etc. Damit das funktioniert, analysiert ein Präprozessor die zu verarbeitenden Dateien und leitet das Ergebnis an `less` weiter. Im Detail ist die Vorgehensweise distributionsabhängig:



- ▶ Bei Fedora und Red Hat ist die Umgebungsvariable LESSOPEN so voreingestellt, dass less zuerst das Script `/usr/bin/lesspipe.sh` ausführt und dessen Ergebnis anzeigt. Bei SUSE verweist LESSOPEN auf das Script `/usr/bin/lessopen.sh`.
- ▶ Bei Debian und Ubuntu sind auch vergleichbare Scripts bzw. Kommandos installiert (`lessfile` und `lesspipe.sh`). Der Unterschied zwischen den beiden Varianten besteht darin, dass `lesspipe` seine Ergebnisse sofort an `less` weiterleitet, während `lessfile` eine temporäre Datei erzeugt. Das ist langsamer, hat aber den Vorteil, dass `less` sofort die Anzahl der Zeilen und die prozentuale Position im Text kennt. Bei Ubuntu ist `lesspipe` standardmäßig aktiv, bei Debian ist eine entsprechende Zeile in `.bashrc` zwar ebenfalls vorgesehen, aber auskommentiert.

### Das Terminal zeigt nur noch merkwürdige Zeichen ...

Wenn Sie in einer Textkonsole eine Datei anzeigen, die statt Text binäre Daten enthält, kann es passieren, dass die Daten als Sonderzeichen interpretiert werden und die Konsole dabei durcheinanderkommt. In diesem Fall werden nur noch seltsame Zeichen auf dem Bildschirm bzw. im Terminalfenster angezeigt, d. h., die Zuordnung des Zeichensatzes stimmt nicht mehr. Abhilfe schafft zumeist das Kommando `reset`.

## Texteditoren

Unter KDE oder Gnome stehen mit `kate` oder `gedit` komfortable Texteditoren mit intuitiver Bedienung zur Verfügung. In einer Textkonsole sind diese Programme aber nicht verwendbar – Sie brauchen einen Editor, der im Textmodus läuft. Dieser Abschnitt stellt die populärsten Vertreter dieser Zunft vor. Welcher der Editoren bei Ihnen standardmäßig installiert ist, hängt von Ihrer Distribution ab.

Eine Sonderrolle unter den Editoren nimmt der GNU Emacs ein (Start mit `emacs`). Dieser Editor enthält unglaublich viele Funktionen und ersetzt für viele Programmierer eine ganze Entwicklungsumgebung. Daher habe ich [Kapitel 16](#) ausschließlich diesem Editor gewidmet.

Emacs, jove, jed,  
jmacs

Vorweg fasst [Tabelle 9.4](#) die elementarsten Kommandos zusammen. Die Kommandos gelten auch für die Editoren `jove`, `jed` und `jmacs`. Dabei handelt es sich um Minimalversionen des Emacs, die in den Grundfunktionen kompatibel sind.

Ebenfalls ein Urgestein der Unix-Geschichte ist der Editor Vi, der unter Linux zumeist durch das dazu kompatible Programm Vim vertreten ist, seltener durch den ebenfalls kompatiblen Editor Elvis. Der Original-Vi ist aus urheberrechtlichen Gründen nicht Teil von Linux. Das Kommando `vi` kann aber dennoch ausgeführt werden und führt dann zum Start von Vim oder Elvis.

Vi, Vim und Elvis

| Tastenkürzel                                  | Funktion   |
|---|--|
| <code>Strg + X</code> , <code>Strg + F</code> | lädt eine neue Datei.                            |
| <code>Strg + X</code> , <code>Strg + S</code> | speichert die aktuelle Datei.                    |
| <code>Strg + X</code> , <code>Strg + W</code> | speichert die Datei unter einem neuen Namen.     |
| <code>Strg + G</code>                         | bricht die Eingabe eines Kommandos ab.           |
| <code>Strg + K</code>                         | löscht eine Zeile.                               |
| <code>Strg + X</code> , <code>U</code>        | macht das Löschen rückgängig (Undo).             |
| <code>Strg + X</code> , <code>Strg + C</code> | beendet den Emacs (mit Rückfrage zum Speichern). |

Tabelle 9.4 Emacs-Tastenkürzel

Der Vi bietet fast genauso viele Funktionen wie der Emacs, die Bedienung ist aber noch schwieriger zu erlernen. Dafür ist der Vi vergleichsweise kompakt und steht zumeist auch auf Notfallsystemen zur Verfügung. Außerdem werden Sie den Vi auf praktisch allen anderen Unix-Systemen vorfinden. Das Programm stellt insofern einen inoffiziellen Unix/Linux-Standard dar und wird von diversen Programmen automatisch als Editor aufgerufen.

Der wichtigste fundamentale Unterschied zu anderen Editoren besteht darin, dass der Vi zwischen verschiedenen Modi unterscheidet. Die Texteingabe ist nur im Insert-Modus möglich (siehe [Tabelle 9.5](#)). Die Eingabe der meisten Kommandos erfolgt im Complex-Command-Modus, der mit `:` aktiviert wird (siehe [Tabelle 9.6](#)). Vorher muss gegebenenfalls der Insert-Modus durch `Esc` verlassen werden. Die Cursorbewegung ist natürlich auch mit den Cursortasten möglich. Dem Vi in seiner Erscheinungsform Vim ist [Kapitel 15](#) gewidmet.

joe ist ein sehr einfacher Editor. Die Tastenkürzel sind dem aus DOS-Zeiten stammenden Textverarbeitungsprogramm Wordstar nachempfunden (siehe [Tabelle 9.7](#)). Eine umfassende Beschreibung aller Kommandos erhalten Sie, wenn Sie in einer Konsole `man joe` ausführen. Das Programm kann auch unter den Namen `jmacs` oder `jpico` gestartet werden. Es gelten dann andere Tastenkürzel, die zum Emacs bzw. zu Pico kompatibel sind.

Ebenfalls bescheiden im Befehlsumfang, aber dafür einfach zu bedienen, ist `nano` bzw. `pico`. Bei diesem Editor geben die beiden unteren Bildschirmzeilen eine Übersicht der zur Verfügung stehenden Kommandos (siehe [Abbildung 9.2](#)). Bei den meisten aktuellen Distributionen ist lediglich `nano` installiert. `pico` war früher stärker verbreitet; seine nicht vollständig open-source-kompatible Lizenz hat aber dazu geführt, dass der Editor nun unter Linux nicht mehr zum Einsatz kommt. `nano` ist zu `pico` kompatibel, leidet aber nicht unter Lizenzproblemen.

| Tastenkürzel | Funktion   |
|--------------|--|
| [I]          | wechselt in den Insert-Modus.                              |
| [Esc]        | beendet den Insert-Modus.                                  |
| [H] / [L]    | bewegt den Cursor nach links/rechts.                       |
| [J] / [K]    | bewegt den Cursor ab/auf.                                  |
| [X]          | löscht ein Zeichen.  |
| [D] [D]      | löscht die aktuelle Zeile.                                 |
| [P]          | fügt die gelöschte Zeile an der Cursorposition wieder ein. |
| [U]          | macht die letzte Änderung rückgängig (Undo).               |
| [:           | wechselt in den Complex-Command-Modus.                     |

Tabelle 9.5 Vi-Tastenkürzel

| Tastenkürzel | Funktion                                    |
|--------------|---|
| :w name      | speichert den Text unter einem neuen Namen. |
| :wq          | speichert und beendet den Vi.               |
| :q!          | beendet den Vi, ohne zu speichern.          |
| :help        | startet die Online-Hilfe.                   |

Tabelle 9.6 Vi-Kommandos im Complex-Command-Modus

Einige Programme starten zum Ansehen oder Editieren von Dateien selbstständig einen Editor, standardmäßig zumeist den Editor Vi. Wenn Sie einen anderen Editor wünschen, müssen Sie in `/etc/profile` oder `.profile` die Umgebungsvariablen `EDITOR` und `VISUAL` einstellen:

**Standardeditor  
einstellen**

```
# Ergänzung in /etc/profile oder ~/.profile
export EDITOR=/usr/bin/jmacs
export VISUAL=$EDITOR
```

Ergänzend dazu bieten viele Distributionen die Möglichkeit, mit dem Kommando `alternatives` ein Defaultprogramm bei mehreren von der Funktion her gleichartigen Programmen (Editoren, Java-Interpreter etc.) einzustellen – siehe [Abschnitt 19.11](#), »Verwaltung von Parallelinstallationen (alternatives)«.

## 9.3 man und info

Kommandos wie `ls`, `cp` oder `top`, die Sie üblicherweise in einem Terminalfenster ausführen, reagieren weder auf `[F1]` noch verfügen sie über ein HILFE-Menü. Es gibt aber

| Tastenkürzel | Funktion   |
|--------------|--|
| Strg + K, H  | blendet das Hilfefenster ein/aus.                  |
| Strg + K, E  | lädt eine neue Datei.                              |
| Strg + K, D  | speichert die Datei (wahlweise unter neuem Namen). |
| Strg + Y     | löscht eine Zeile.                                 |
| Strg + ↵ + - | macht das Löschen rückgängig (Undo).               |
| Strg + C     | beendet joe (mit Rückfrage zum Speichern).         |

Tabelle 9.7 joe-Tastenkürzel



Abbildung 9.2 Der Editor nano in einem KDE-Terminalfenster

natürlich auch für diese Kommandos Hilfetexte, die durch verschiedene Kommandos gelesen werden können:

- ▶ kommando `--help` liefert bei sehr vielen Kommandos eine Liste aller Optionen samt einer kurzen Erklärung zu ihrer Bedeutung.
- ▶ man kommando zeigt bei vielen Kommandos den man-Hilfetext an. Durch den meist mehrseitigen Text können Sie mit den Cursortasten blättern. Mit `Q` beenden Sie die Hilfe.
- ▶ help kommando funktioniert nur bei sogenannten Shell-Kommandos, z. B. `cd` oder `alias`.
- ▶ info kommando ist eine Alternative zu man. Das info-System eignet sich vor allem für sehr umfangreiche Hilfetexte. Ob der Hilfetext im man- oder info-System vorliegt, hängt ganz einfach davon ab, für welches Hilfesystem sich die Programmentwickler entschieden haben. man ist aber deutlich populärer.

man ist ein Kommando zur Anzeige der Dokumentation vieler elementarer Kommandos wie `ls` oder `cp`. man wird in der Form `man kommando` aufgerufen, um den Hilfetext zu kommando zu lesen.

Die optionale Angabe eines Bereichs (`man bereich kommando`) schränkt die Suche nach man-Texten auf einen Themenbereich ein. Beispielsweise liefert `man 3 printf` die Syntax der C-Funktion `printf`. Diese Einschränkung ist dann notwendig, wenn mehrere gleichnamige man-Texte in unterschiedlichen Themenbereichen existieren. man zeigt in diesem Fall nur den ersten gefundenen man-Text an.

Wenn Sie alle gleichnamigen man-Texte aus allen Bereichen lesen möchten, müssen Sie man mit der Option `-a` verwenden. Sobald Sie den Text gelesen haben und man mit `q` beenden, erscheint der man-Text zum nächsten Abschnitt.

In vielen Unix- und Linux-Büchern werden zusammen mit den Kommandos gleich die man-Nummern angegeben – etwa `find(1)`. Damit wissen Sie sofort, wie Sie man aufrufen müssen. man kennt üblicherweise die Themenbereiche 1 bis 9 und n (siehe [Tabelle 9.8](#)). Manchmal werden die Kommandos von Programmiersprachen in zusätzlichen Bereichen mit anderen Buchstaben eingeordnet.

Die Darstellung der Hilfetexte erfolgt intern durch das Programm `less`. Deswegen gelten für die Navigation im Hilfetext die Tastenkürzel aus [Tabelle 9.3](#). Aus welchen Verzeichnissen man die Hilfetexte liest, können Sie wahlweise durch die Steuerungsdatei `/etc/manpath.config` oder durch die Umgebungsvariable `MANPATH` einstellen.

|   | Thema                               |   | Thema                              |
|---|-------------------------------------|---|------------------------------------|
| 1 | Benutzerkommandos                   | 6 | Spiele                             |
| 2 | Systemaufrufe                       | 7 | Diverses                           |
| 3 | Funktionen der Programmiersprache C | 8 | Kommandos zur Systemadministration |
| 4 | Dateiformate, Device-Dateien        | 9 | Kernelfunktionen                   |
| 5 | Konfigurationsdateien               | n | neue Kommandos                     |

**Tabelle 9.8** man-Themengruppen

Unter KDE und Gnome können Sie man-Texte auch mit den jeweiligen Help- oder Webbrowsern lesen. Die folgenden Beispiele zeigen, wie Sie die man-Seite zu `ls` und ein Inhaltsverzeichnis aller man-Seiten anzeigen können:

```
user$ gnome-help man:ls
user$ khelpcenter man:ls
user$ khelpcenter 'man:(index)'
```

- help** Zu manchen Kommandos erhalten Sie Hilfe nicht mit `man`, sondern mit `help`. Das betrifft alle Kommandos, die direkt von der Shell ausgeführt werden. (Die Shell ist der Kommandointerpreter, der Ihre Eingaben entgegennimmt. Ausführliche Informationen zur Linux-Standard-Shell `bash` finden Sie im nächsten Kapitel.)
- info** `man`-Hilfetexte haben den Nachteil, dass sie nur schwer strukturierbar sind. Das alternative `info`-Format bietet hier deutlich bessere Möglichkeiten, weswegen vor allem umfangreiche Hilfetexte häufig nur in diesem Format vorliegen.

`info` wird üblicherweise in der Form `info kommando` aufgerufen. Wird das Kommando ohne Parameter gestartet, zeigt das Programm eine Übersicht der verfügbaren Hilfetemen an.

Leider erweist sich der Vorteil der klareren Strukturierung rasch als Nachteil: Die Navigation in `info`-Texten ist unübersichtlich, außerdem fehlt ein Suchmechanismus, der über die gerade aktuelle Seite hinausreicht.

| Tastenkürzel | Funktion   |
|--------------|--|
| Leertaste    | Text nach unten scrollen   |
| ←            | Text nach oben scrollen  |
| B, E         | zum Anfang/Ende der Info-Einheit springen ( <i>beginning/end</i> ) |
| ↔            | Cursor zum nächsten Querverweis bewegen                            |
| ↵            | Querverweis zu anderer Info-Einheit verfolgen                      |
| N            | nächste Info-Einheit derselben Hierarchiestufe ( <i>next</i> )     |
| P            | vorige Info-Einheit derselben Hierarchiestufe ( <i>previous</i> )  |
| U            | eine Hierarchieebene nach oben ( <i>up</i> )                       |
| L            | zurück zum zuletzt angezeigten Text ( <i>last</i> )                |
| H            | ausführliche Bedienungsanleitung ( <i>help</i> )                   |
| ?            | Kommandoübersicht  |
| Q            | beendet <code>info</code> ( <i>quit</i> ).                         |

**Tabelle 9.9** `info`-Tastenkürzel

Statt `info` können Sie auch den Editor Emacs starten und mit `[Alt]+[X]` `info` `↵` oder mit `[Strg]+[H]`, `[I]` in den `info`-Modus wechseln. Dort werden alle Querverweise farbig hervorgehoben und können durch einen Klick mit der mittleren Maustaste bequem verfolgt werden. Eine andere komfortable Alternative zu `info` ist das Programm `pinfo`. Unter KDE bzw. Gnome lesen Sie `info`-Texte am besten mit dem jeweiligen Hilfesystem.

# Kapitel 10

## bash (Shell)

Im Mittelpunkt dieses Kapitels steht die Bourne Again Shell (kurz `bash`). Dieses Programm ermöglicht die Ausführung von Kommandos in einem Terminalfenster bzw. in einer Textkonsole. Eine Shell ist also ein Kommandointerpreter, der eine Menge Zusatzfunktionen bietet, z. B. die Kombination mehrerer Kommandos oder die Speicherung der Ergebnisse eines Kommandos in einer Datei. Gleichzeitig enthält die `bash` eine eigene Programmiersprache, die zur Erstellung von Shell-Programmen (Shell-Scripts) verwendet werden kann.

Dieses Kapitel behandelt die Verwendung der `bash` sowohl als Kommandointerpreter als auch zur Programmierung. Wesentliche Themen dieses Kapitels sind eine Einführung in den Umgang mit der `bash`, die Ein- und Ausgabeumleitung, die Kommunikation zwischen mehreren Prozessen (Pipes, Kommandosubstitution) und die Verwaltung von Shell-Variablen. Wenn Sie sich für die `bash`-Programmierung interessieren, finden Sie in [Abschnitt 10.8](#), »`bash`-Script-Beispiele«, einen Überblick über die wichtigsten Sprachelemente und diverse Beispiele. Das Kapitel endet mit einer Tabelle aller Sonderzeichen der `bash`.

### 10.1 Was ist eine Shell?

*Bourne Again Shell* ist ein englisches Wortspiel: Die `bash` ist somit die wiedergeborene Bourne-Shell, die neben der Korn-Shell und der C-Shell zu den drei klassischen Unix-Shells zählt. Unter Linux sind alle drei Shells und noch einige weitere verfügbar, standardmäßig wird aber zumeist die `bash` eingerichtet.

Was ist nun eine Shell? In erster Linie wird die Shell zum Aufruf von Linux-Kommandos und Programmen eingesetzt. Sie stellt damit eine Art Kommandointerpreter dar, vergleichbar in etwa mit `cmd.exe` aus der Windows-Welt. Eine Shell wird in jedem Terminalfenster und in jeder Textkonsole nach dem Login ausgeführt. Gleichzeitig stellt die Shell eine Programmiersprache zur Verfügung, mit der Arbeitsabläufe automatisiert werden können. Mit speziellen Shell-Kommandos können Sie innerhalb dieser Programme Variablen verwenden, Abfragen und Schleifen bilden etc. Die resultierenden Programme werden je nach den Präferenzen des Autors als »Stapeldateien«,

»Batch-Dateien«, »Scripts«, »Shell-Prozeduren« oder so ähnlich bezeichnet. In jedem Fall handelt es sich dabei um einfache Textdateien, die von der Shell ausgeführt (interpretiert) werden.

**Version** Dieses Kapitel beschreibt die bash-Version 4.n. Wenn Sie nicht wissen, mit welcher Shell(-Version) Sie arbeiten, führen Sie die folgenden Kommandos aus:

```
user$ echo $0
-bash
user$ bash --version
GNU bash, Version 4.4.12(1)-release (x86_64-redhat-linux-gnu)
```

**Dokumentation** Zur bash existieren ein umfangreicher man-Text und eine ebenso umfangreiche info-Datei. Denselben Text können Sie auch im Webbrowser lesen:

<https://gnu.org/software/bash/manual/bash.html>

## Andere Shells

Die bash gilt bei nahezu allen Linux-Distributionen als Standard-Shell für die Arbeit in Konsolen oder Terminalfenstern. Mit dem Paketverwaltungssystem Ihrer Distribution können Sie unzählige weitere Shells installieren. Bei Linux-Profis ist insbesondere die Z-Shell zsh beliebt. Andere Varianten sind die Korn-Shell (ksh oder pksh) und die C-Shell (csh oder tcsh). Um eine dieser Shells nach der Installation auszuprobieren, starten Sie ein Terminalfenster und führen darin den jeweiligen Shell-Namen aus. `exit` führt zurück in die zuletzt aktive Shell.

```
user$ zsh
hostname% ls      (Kommandos in der zsh ausführen)
...
hostname% exit   (zurück zur vorigen Shell)
user$
```

**Standard-Shell verändern** Für jeden Linux-Benutzer ist eine eigene Standard-Shell vorgesehen. Diese Shell wird ausgeführt, wenn Sie ein Terminalfenster öffnen bzw. wenn Sie sich in einer Textkonsole anmelden. Die Standard-Shell ist in der Datei `/etc/passwd` gespeichert. Die Shell wird als letzter Eintrag in der Zeile jedes Anwenders genannt. Um eine andere Standard-Shell einzustellen, führen Sie das Kommando `chsh` (*change shell*) aus. Die Shell-Programme sind im Verzeichnis `/bin` gespeichert. Sie müssen also beispielsweise `chsh /bin/csh` angeben, wenn Sie in Zukunft mit der C-Shell arbeiten möchten. Eine Liste der verwendbaren Shells befindet sich in `/etc/shells`.

**zsh** Die vielleicht interessanteste Shell für Linux-Freaks und Entwickler ist die Z-Shell (zsh). Eine gute Einführung gibt dieser Artikel:

<https://pro-linux.de/artikel/2/1186>



## 10.2 Basiskonfiguration

Die Tastaturkonfiguration der bash wird global in der Datei `/etc/inputrc` bzw. individuell durch `~/.inputrc` eingestellt. Falls Sie keine deutschen Sonderzeichen eingeben können oder die Tasten `[Entf]`, `[Pos1]` und `[Ende]` nicht wie erwartet funktionieren, müssen Sie `inputrc` wie folgt einstellen. Alle gängigen Distributionen sind standardmäßig so konfiguriert, wobei es oft noch diverse weitere Einstellungen gibt.

Funktionstasten  
in der bash

```
# Datei /etc/inputrc bzw. ~/.inputrc
set meta-flag on
set convert-meta off
set output-meta on
"\e[1~": beginning-of-line
"\e[3~": delete-char
"\e[4~": end-of-line
```

Diese Datei steuert die Funktion `readline`, die bash-intern zur Verarbeitung von Tastatureingaben verwendet wird. Durch die drei ersten Anweisungen wird erreicht, dass erstens 8-Bit-Zeichen bei der Eingabe erkannt werden, dass sie zweitens nicht in andere Zeichen konvertiert werden und dass sie drittens auch tatsächlich ausgegeben werden. Die nächsten drei Zeilen steuern die Reaktion auf das Drücken der Tasten `[Pos1]`, `[Entf]` und `[Ende]`.

Die Veränderungen werden erst nach einem Neustart der Shell wirksam. In einer Textkonsole loggen Sie sich aus und dann wieder ein. In Desktop-Systemen starten Sie ein neues Terminalfenster.

In der Shell wird am Beginn jeder Eingabezeile je nach Distribution der Name des Rechners, des Benutzers und/oder des aktuellen Verzeichnisses angezeigt. Diese Zeichenkette wird »Prompt« genannt.

Eingabeprompt

Der Inhalt des Prompts wird durch die Umgebungsvariable `PS1` festgelegt, systemweit oft in der Datei `/etc/bash.bashrc`, bei Red Hat/Fedora in `/etc/bashrc`. Um die Variable `PS1` individuell einzustellen, ändern Sie je nach Distribution die Datei `.profile` oder `.bashrc`. (Hintergründe zum Umgang mit Umgebungsvariablen behandelt [Abschnitt 10.7](#), »Shell-Variablen«.) Die folgende Zeile bewirkt, dass als Prompt nur das aktuelle Verzeichnis angezeigt wird:

```
# Veränderung in ~/.profile oder ~/.bashrc
PS1="\w \ $"
```

Dabei ist `\u` ein Platzhalter für den Benutzernamen, `\h` für den Hostnamen, `\w` für das gesamte aktuelle Verzeichnis, `\W` für den letzten Teil des aktuellen Verzeichnisses und `\$` für den Promptabschluss (`$` oder `#` für `root`).

**Farbiger Prompt** Mit `\[\e[0;nnm\]` können Sie in PS1 den Formatierungscode `nn` einbetten. Eine umfassende Anleitung zur Promptkonfiguration inklusive einer Auflistung aller ANSI-Farbcodes finden Sie im folgenden HOWTO-Dokument:

*<http://tldp.org/HOWTO/Bash-Prompt-HOWTO>*

Auf meinen Rechnern verwende ich die folgende Einstellung:

```
PS1='\[\e[0;34m\]\u@h:\W\$\[\e[0;39m\] '
```

Damit wird ein blauer Prompt in der Form `benutzername@rechnername:verzeichnis` angezeigt. Der Prompt zeigt aber nicht den gesamten Pfad an, sondern nur den letzten Teil, also z. B. `nautilus`, wenn das aktuelle Verzeichnis `/usr/lib/nautilus` lautet. Das spart Platz, wenn Sie sich in einem mehrteiligen Verzeichnis befinden. Ergänzend zu PS1 kann auch die Variable `PROMPT_COMMAND` eingestellt werden. Diese Variable enthält ein Kommando, das jedes Mal ausgeführt wird, bevor PS1 angezeigt wird.

### 10.3 Kommandoeingabe

Normalerweise nutzen Sie die `bash` einfach durch die Eingabe ganz gewöhnlicher Kommandos. Die `bash` unterstützt Sie dabei durch eine Menge praktischer Tastenkürzel und Sondertasten. Insbesondere können Sie mit den Cursorstasten `↑` und `↓` die zuletzt eingegebenen Kommandos wieder bearbeiten, was eine Menge Tipparbeit spart. Beim Ausloggen aus einer Shell werden die zuletzt eingegebenen Kommandos in einer Datei `~/.bash_history` gespeichert und stehen so auch nach dem nächsten Einloggen wieder zur Verfügung.

Kommandozeilen können wie in einem Texteditor verändert werden, das heißt, Sie können an beliebigen Stellen Zeichen einfügen und löschen. Die Tastaturbelegung der `bash` ist praktisch vollständig konfigurierbar. Außerdem können Sie zwischen dem `emacs`- und dem `vi`-Modus umschalten. Damit gelten für alle grundlegenden Edit-Kommandos dieselben Tastenkürzel wie im jeweils ausgewählten Editor. Die Standardeinstellung ist in der Regel der `emacs`-Modus. In diesem Kapitel werden alle Tastenkürzel ebenfalls für diesen Modus angegeben.

#### Expansion von Kommando- und Dateinamen

Mit der automatischen Expansion von Kommando- und Dateinamen hilft die `bash` Ihnen, den Tippaufwand zu minimieren. Dazu geben Sie zuerst die Anfangsbuchstaben des Kommandos oder des Dateinamens ein und drücken dann `⏪`. Wenn der Name bereits eindeutig identifizierbar ist, wird er vollständig ergänzt. Wenn es mehrere Namen gibt, die mit den gleichen Buchstaben beginnen, wird der Name nur so

weit erweitert, wie die Namen übereinstimmen. Außerdem erklingt in diesem Fall ein Signalton, der darauf hinweist, dass der Dateiname möglicherweise noch nicht vollständig ist.

Am leichtesten ist die Expansion von Dateinamen anhand eines Beispiels zu verstehen. Die Eingabe


```
user$ ema  ba 
```

wird auf meinem Rechner im Verzeichnis mit den Dateien für dieses Buch automatisch zu

```
user$ emacs bash.tex
```


erweitert. Dabei ist `emacs` der Name meines Lieblingseditors und `bash.tex` der Name der  $\text{\LaTeX}$ -Datei dieses Kapitels. Zur Vervollständigung von `em` durchsucht `bash` alle in der `PATH`-Variablen angegebenen Verzeichnisse nach ausführbaren Programmen. Zur Vervollständigung des Dateinamens wird dagegen nur das aktuelle Verzeichnis berücksichtigt.



Die Expansion funktioniert auch bei Dateinamen, denen mehrere Verzeichnisse vorangestellt sind. Wenn Sie

```
user$ ls /usr/sh 
```

eingeben, erweitert `bash` diese Eingabe zu:

```
user$ ls /usr/share/
```

Wenn eine eindeutige Erweiterung nicht möglich ist (Signalton), können Sie einfach nochmals  drücken. `bash` zeigt dann in den Zeilen unterhalb der aktuellen Eingabezeile alle möglichen Ergänzungen an. Die Eingabe

```
user$ e  
```

führt zur Ausgabe einer fast endlosen Liste aller Kommandos und Programme, die mit dem Buchstaben `e` beginnen. Anschließend kann die Eingabe fortgesetzt werden.

### Programme bzw. Scripts im lokalen Verzeichnis starten

Programme und Kommandos im gerade aktuellen Verzeichnis werden bei der Kommandoexpansion nur dann berücksichtigt, wenn das aktuelle Verzeichnis in der `PATH`-Variablen enthalten ist. Den Inhalt von `PATH` können Sie sich mit `echo $PATH` ansehen. Das aktuelle Verzeichnis wird durch `».«` abgekürzt.

Bei allen gängigen Linux-Distributionen fehlt aus Sicherheitsgründen das aktuelle Verzeichnis in `PATH`. Um Programme aus dem aktuellen Verzeichnis auszuführen, müssen Sie daher `./name` eingeben.

**Pfad zum Programm ermitteln**

Die automatische Kommandoexpansion verschleiert, wo sich ein Programm nun wirklich befindet. Um das herauszufinden, gibt es mehrere Möglichkeiten:

- ▶ `whereis name` durchsucht alle Standardverzeichnisse.
- ▶ `which name` durchsucht alle in `PATH` enthaltenen Verzeichnisse und ermittelt das Programm, das bei der Eingabe des Kommandos ohne Pfad ausgeführt würde. `which` ist dann interessant, wenn es mehrere Versionen eines Programms gibt, die sich in unterschiedlichen Verzeichnissen befinden.
- ▶ `type name` funktioniert ähnlich wie `which`, berücksichtigt aber auch Kommandos, die in der `bash` integriert sind oder als Alias definiert sind.

Die `bash` bietet analoge Expansionsmechanismen auch für die Namen von Heimatverzeichnissen und für Variablennamen an: `~ko` liefert auf meinem Rechner `~kofler/`, `$PAT` ergibt `$PATH`.

**Programmspezifische Expansion**

Bei der Ausführung des Kommandos `latex name.tex` kommen als mögliche Dateien nur solche infrage, die mit `*.tex` enden. Wenn Sie `man name` ausführen, sind nur Einträge relevant, zu denen tatsächlich `man`-Texte existieren. Analog gibt es zahlreiche weitere Kommandos und Programme, bei denen die Auswahl der möglichen Dateien oder Parameter von vornherein eingeschränkt ist. Da ist es natürlich praktisch, wenn bei der Expansion nur solche Dateien bzw. Parameter berücksichtigt werden, die zum Kommando passen.

Genau darum kümmert sich das `bash`-Kommando `complete`. Viele Distributionen sind mit einer umfangreichen `complete`-Konfiguration ausgestattet, die aber teilweise extra installiert werden muss (Paket `bash-completion`). Die Konfiguration erfolgt in der Regel durch eine der folgenden Dateien:

```
/etc/bash_completion
/etc/bash_completion.d/*
/etc/profile.d/complete.bash
/etc/profile.d/bash_completion.sh
```

Zur Definition eigener Expansionsregeln müssen Sie sich in die recht unübersichtliche Syntax von `complete` einarbeiten. Eine knappe Beschreibung geben `help complete` und `man bash` (suchen Sie nach *Programmable Completion*). Weitere Tipps zur Konfiguration des Expansionsmechanismus finden Sie unter:

<https://pro-linux.de/artikel/2/153>

**Wichtige Tastenkürzel**

Tabelle 10.1 fasst die wichtigsten Tastenkürzel der `bash` zusammen. Die Tabelle geht davon aus, dass `bash` für den `emacs`-Modus konfiguriert ist, wie dies bei nahezu allen Distributionen der Fall ist.

| Kürzel         | Bedeutung   |
|----------------|---|
| ↑, ↓           | durch die zuletzt eingegebenen Kommandos scrollen       |
| ←, →           | Cursor zurück- bzw. vorbewegen                          |
| Strg+A, Strg+E | Cursor an den Beginn bzw. an das Ende der Zeile bewegen |
| Alt+B, Alt+F   | Cursor um ein Wort rückwärts bzw. vorwärts bewegen      |
| Alt+D          | Wort löschen  |
| Strg+K         | bis zum Ende der Zeile löschen                          |
| Strg+Y         | zuletzt gelöschten Text wieder einfügen                 |
| Strg+T         | die beiden vorangehenden Zeichen vertauschen            |
| Alt+T          | die beiden vorangehenden Wörter vertauschen             |
| ↵              | Expansion des Kommando- oder Dateinamens                |
| Strg+L         | den Bildschirm löschen                                  |
| Strg+R         | Suche nach früher eingegebenen Kommandos                |
| Alt+.          | den zuletzt verwendeten Parameter einfügen              |
| Strg+_         | letzte Änderung rückgängig machen (Undo)                |

Tabelle 10.1 Die wichtigsten bash-Tastenkürzel

Die Funktion des Tastenkürzels `Alt+.` ist nur anhand eines Beispiels zu verstehen. Nehmen wir an, Sie haben gerade eine Datei kopiert (`cp name1 name2`). Nun wollen Sie im nächsten Kommando die Kopie wieder löschen. Statt `rm name2` geben Sie `rm` und dann `Alt+.` ein. `bash` fügt automatisch den zuletzt verwendeten Befehlsparameter ein. Durch das mehrfache Drücken von `Alt+.` können Sie auch auf alle weiteren Parameter zurückgreifen, also auf `name1` durch zweimaliges Drücken.

Letzten  
Parameter  
einfügen

Auch das Tastenkürzel `Strg+R` bedarf einer ausführlicheren Erklärung: Damit ist es möglich, bereits eingegebene Kommandos zu suchen: Drücken Sie am Beginn der Zeile `Strg+R`, und geben Sie dann die ersten Zeichen der gesuchten Kommandozeile ein. `bash` zeigt daraufhin automatisch das zuletzt verwendete Kommando mit diesen Anfangsbuchstaben an. Mehrmaliges Drücken von `Strg+R` wechselt zwischen verschiedenen passenden Möglichkeiten. `Strg+S` funktioniert wie `Strg+R`, durchläuft die Liste passender Kommandos aber in umgekehrter Richtung. `←`, `↵` und die Cursortasten brechen die Suche ab und führen das gefundene Kommando aus bzw. ermöglichen das Editieren der gefundenen Zeile.

Kommandosuche

Manche Konsolen betrachten `Strg+S` als Anweisung, die Ausgabe vorübergehend zu stoppen. Erst `Strg+Q` setzt die Ausgabe wieder fort. Wenn Ihre Konsole so auf `Strg+S` reagiert, können Sie die Kommandosuche nur mit `Strg+R` durchführen.

Die bash-Tastenkürzel stammen eigentlich von der `readline`-Bibliothek, die von bash zur Verarbeitung von Eingaben genutzt wird. Noch mehr Kürzel finden Sie mit `man readline`.

## Alias-Abkürzungen

Mit dem Kommando `alias` können Sie sich bei der Eingabe von Kommandos in der Shell einige Tipparbeit ersparen. Mit diesem Kommando werden Abkürzungen definiert. Bei der Verarbeitung der Kommandozeile wird überprüft, ob das erste Wort eine Abkürzung enthält. Wenn das der Fall ist, wird die Abkürzung durch den vollständigen Text ersetzt.

Abkürzungen für eine bestimmte Kombination von Optionen oder für Dateinamen sind nicht möglich, weil die bash die weiteren Parameter eines Kommandos nicht nach Abkürzungen durchsucht. Die bash erkennt aber Sonderfälle, bei denen in einer Kommandozeile mehrere Programme genannt werden (Pipes, Kommandosubstitution, sequenzielle Ausführung von Kommandos mit `»;`), und durchsucht alle vorkommenden Kommandonamen auf Abkürzungen.

```
user$ alias cdb='cd ~/kofler/linuxbuch'
```

Durch das obige Kommando wird die Abkürzung `cdb` definiert, mit der ich rasch in das von mir oft benötigte Verzeichnis `/home/kofler/linuxbuch` wechseln kann.

`alias`-Aufrufe können auch verschachtelt eingesetzt werden. Beachten Sie, dass `alias`-Abkürzungen Vorrang gegenüber gleichnamigen Kommandos haben. Das kann dazu genutzt werden, um den unerwünschten Aufruf eines Kommandos zu vermeiden:

```
user$ alias more=less
```

Von nun an führt jeder Versuch, das Kommando `more` aufzurufen, zum Start des leistungsfähigeren Programms `less`. Sollten Sie aus irgendeinem Grund dennoch `more` benötigen, müssen Sie den gesamten Pfadnamen angeben (`/bin/more`) oder einen Backslash voranstellen (`\more`). Der Backslash verhindert in diesem Fall die Alias-Auswertung.

`alias`-Abkürzungen können mit `unalias` wieder gelöscht werden. Ansonsten gelten sie bis zum Verlassen der Shell (also spätestens bis zum Logout). Wenn Sie bestimmte Abkürzungen immer wieder benötigen, sollten Sie die `alias`-Anweisungen in die Dateien `/etc/bashrc` oder `.bashrc` in Ihrem Heimatverzeichnis aufnehmen.

Bei vielen Distributionen sind diverse `alias`-Abkürzungen vordefiniert. Wenn also beispielsweise `rm` ständig fragt, ob die Datei wirklich gelöscht werden soll, ist meist der vordefinierte Alias `rm=rm -i` schuld. Eine Liste mit allen gerade gültigen Abkürzungen liefert das Kommando `alias`. Die folgenden Zeilen geben an, an welchen Orten Debian, Fedora, SUSE und Ubuntu `alias`-Definitionen berücksichtigen:

```
Debian, Fedora, Ubuntu: /etc/bashrc      /etc/profile.d/*.sh ~/.bashrc
SUSE:                   /etc/bash.bashrc /etc/profile.d/*.sh ~/.bashrc ~/.alias
```

Eine ähnliche Wirkung wie Abkürzungen können auch Shell-Programme haben. Shell-Scripts haben zudem den Vorteil, dass sie mit Parametern (\$1, \$2 etc.) zurecht kommen und flexibler eingesetzt werden können.

## 10.4 Ein- und Ausgabeumleitung

Bei der Ausführung von Kommandos in der `bash` existieren drei sogenannte Standarddateien. Der Begriff »Datei« stiftet dabei ein wenig Verwirrung: Es handelt sich eigentlich nicht um richtige Dateien, sondern um Dateideskriptoren, die auf Betriebssystemebene wie Dateien behandelt werden.

- ▶ **Standardeingabe:** Das gerade ausgeführte Programm, z. B. die `bash` oder ein beliebiges von dort gestartetes Kommando, liest alle Eingaben von der Standardeingabe. Als Standardeingabequelle gilt normalerweise die Tastatur.
- ▶ **Standardausgabe:** Dorthin werden alle Ausgaben des Programms geleitet – etwa die Auflistung aller Dateien durch `ls`. Als Standardausgabe gilt normalerweise das Terminalfenster.
- ▶ **Standardfehler:** Auch Fehlermeldungen werden üblicherweise im aktuellen Terminal angezeigt.

An sich ist das alles selbstverständlich – woher sonst als von der Tastatur sollten die Eingaben kommen, wo sonst als auf dem Bildschirm sollten Ergebnisse oder Fehler angezeigt werden? Bemerkenswert ist aber die Möglichkeit, die Standardeingabe oder -ausgabe umzuleiten.

Beispielsweise kann der Fall auftreten, dass das Inhaltsverzeichnis des aktuellen Verzeichnisses nicht auf dem Bildschirm angezeigt, sondern in einer Datei gespeichert werden soll. Die Standardausgabe soll also in eine echte Datei umgeleitet werden. Das erfolgt in der `bash` durch das Zeichen `>`:

```
user$ ls *.tex > inhalt
```

In der Textdatei `inhalt` befindet sich jetzt eine Liste aller `*.tex`-Dateien im aktuellen Verzeichnis. Diese Form der Ausgabeumleitung ist sicherlich die häufigste Anwendung. Daneben existieren aber viele weitere Varianten: `>> datei` leitet alle Fehlermeldungen in die angegebene Datei. `>& datei` bzw. `&> datei` leiten sowohl die Standardausgabe als auch alle Fehlermeldungen in die angegebene Datei. Wenn statt `>` die Verdoppelung `>>` verwendet wird, dann werden die jeweiligen Ausgaben an das Ende einer bereits bestehenden Datei angehängt. Diese und weitere Codes zur Ein- und Ausgabeumleitung sind in [Tabelle 10.2](#) zusammengefasst.

| Kommando              | Funktion   |
|-----------------------|--|
| kommando > datei      | leitet Standardausgaben zur angegebenen Datei.           |
| kommando < datei      | liest Eingaben aus der angegebenen Datei.                |
| kommando 2> datei     | leitet Fehlermeldungen zur angegebenen Datei.            |
| kommando >& datei     | leitet Ausgaben <i>und</i> Fehler um.                    |
| kommando &> datei     | leitet ebenfalls Ausgaben <i>und</i> Fehler um.          |
| kommando >> datei     | hängt Standardausgaben an die vorhandene Datei an.       |
| kommando &>> datei    | hängt Ausgaben und Fehler an die Datei an (ab bash 4.0). |
| kommando1   kommando2 | leitet Ausgaben von Kommando 1 an Kommando 2 weiter.     |
| kommando   tee datei  | zeigt die Ausgaben an und speichert zugleich eine Kopie. |

**Tabelle 10.2** Ein- und Ausgabeumleitung

Eine Eingabeumleitung erfolgt mit `< datei`: Kommandos, die Eingaben von der Tastatur erwarten, lesen diese damit aus der angegebenen Datei.

### Achtung

Es ist nicht möglich, eine Datei zu bearbeiten und gleichzeitig das Ergebnis wieder in diese Datei zu schreiben!

`sort dat > dat` oder auch `sort < dat > dat` führt dazu, dass `dat` gelöscht wird!

**Pipes** Pipes werden mit dem Zeichen `|` gebildet. Dabei wird die Ausgabe des ersten Kommandos als Eingabe für das zweite Kommando verwendet. In der Praxis werden Sie Pipes oft zusammen mit dem Kommando `less` bilden, wenn Sie längere Ausgaben seitenweise betrachten möchten.

```
user$ ls -l | less
```

Durch das obige Kommando wird das Inhaltsverzeichnis des aktuellen Verzeichnisses ermittelt und in eine Pipe geschrieben. Von dort liest das parallel ausgeführte Kommando `less` seine Eingaben und zeigt sie auf dem Bildschirm an.

Pipes eignen sich auch hervorragend dazu, unterschiedliche Kommandos zu kombinieren. So liefert das folgende Kommando eine sortierte Liste aller installierten RPM-Pakete:

```
user$ rpm -qa | sort
```

Statt Pipes können zur Ein- und Ausgabeumleitung auch sogenannte FIFO-Dateien verwendet werden. FIFO steht für *First In First Out* und realisiert die Idee einer Pipe in



Form einer Datei. FIFOs sind bei der Eingabe viel umständlicher als Pipes, sie machen aber deutlich, was das Zeichen | eigentlich bewirkt. In der Praxis werden sie verwendet, damit zwei voneinander unabhängige Programme miteinander kommunizieren können.

```
user$ mkfifo fifo
user$ ls -l > fifo &
user$ less < fifo
```

Durch die drei obigen Kommandos wird zuerst eine FIFO-Datei eingerichtet. Anschließend wird `ls` als Hintergrundprozess gestartet. Er schreibt seine Ausgaben in die Datei. Von dort liest `less` die Daten wieder aus und zeigt sie auf dem Bildschirm an.

Zur Formulierung einer Pipe eignen sich nur solche Kommandos, die die zu verarbeitenden Kommandos aus dem Standardeingabekanal lesen. Wenn das nicht der Fall ist, können Sie ähnliche Effekte durch eine Kommandosubstitution oder durch das Kommando `xargs` erzielen. Diese und andere Substitutionsmechanismen sind Thema von [Abschnitt 10.6](#).

### Ausgabevervielfachung mit »tee«

Gelegentlich kommt es vor, dass die Ausgaben eines Programms zwar in einer Datei gespeichert werden sollen, dass Sie aber dennoch parallel am Bildschirm den Programmverlauf verfolgen wollen. In diesem Fall ist eine Verdoppelung der Ausgabe erforderlich, wobei eine Kopie auf dem Bildschirm angezeigt und die zweite Kopie in einer Datei gespeichert wird. Diese Aufgabe übernimmt das Kommando `tee`:

```
user$ ls | tee inhalt
```

Das Inhaltsverzeichnis des aktuellen Verzeichnisses wird auf dem Bildschirm angezeigt und gleichzeitig in der Datei `inhalt` gespeichert. Dabei erfolgt zuerst eine Weiterleitung der Standardausgabe an das Kommando `tee`. Dieses Kommando zeigt standardmäßig die Standardausgabe auf dem Terminal an und speichert die Kopie davon in der angegebenen Datei. Dass es sich wirklich um eine Vervielfachung der Ausgabe handelt, bemerken Sie, wenn Sie auch die Standardausgabe von `tee` in eine Datei weiterleiten:

```
user$ ls | tee inhalt1 > inhalt2
```

Das Ergebnis sind zwei identische Dateien, `inhalt1` und `inhalt2`. Das obige Kommando hat reinen Beispielcharakter. Etwas schwieriger zu verstehen, dafür aber sinnvoller, ist das folgende Beispiel:

```
user$ ls -l | tee inhalt1 | sort +4 > inhalt2
```

In `inhalt1` befindet sich wiederum das »normale« Inhaltsverzeichnis, das von `ls` automatisch nach Dateinamen sortiert wurde. Die Kopie dieser Ausgabe wurde an `sort` weitergegeben, dort nach der Dateigröße (fünfte Spalte, also Option `+4`) sortiert und in `inhalt2` gespeichert.

## 10.5 Kommandos ausführen

Üblicherweise starten Sie Kommandos einfach durch die Eingabe des Kommandonamens. Daneben gibt es aber einige Möglichkeiten, um mehrere Kommandos hintereinander auszuführen (siehe [Tabelle 10.3](#)).

| Kommando                                    | Funktion  |
|---|---|
| <code>kommando1; kommando2</code>           | führt die Kommandos nacheinander aus.                         |
| <code>kommando1 &amp;&amp; kommando2</code> | führt Kommando 2 aus, wenn Kommando 1 erfolgreich war.        |
| <code>kommando1    kommando2</code>         | führt Kommando 2 aus, wenn Kommando 1 einen Fehler liefert.   |
| <code>kommando &amp;</code>                 | startet das Kommando im Hintergrund.                          |
| <code>kommando1 &amp; kommando2</code>      | startet Kommando 1 im Hintergrund, Kommando 2 im Vordergrund. |
| <code>(kommando1 ; kommando2)</code>        | führt beide Kommandos in der gleichen Shell aus.              |

**Tabelle 10.3** Kommandoausführung

### Hintergrundprozesse

Das wichtigste und am häufigsten benötigte Sonderzeichen ist `&`. Wenn es am Ende der Kommandozeile eingegeben wird, startet `bash` dieses Programm im Hintergrund. Das ist vor allem bei zeitaufwendigen Programmen sinnvoll, weil sofort weitergearbeitet werden kann.

```
user$ find / -name '*sh' > ergebnis &
[1] 3345
```

Das obige Kommando durchsucht das gesamte Dateisystem nach Dateien, die mit den Buchstaben »sh« enden. Die Liste der Dateien wird in die Datei `ergebnis` geschrieben. Da das Kommando im Hintergrund ausgeführt wird, können Sie sofort weiterarbeiten. Die Ausgabe `[1] 3345` bedeutet, dass der Hintergrundprozess die PID-Nummer 3345 hat. PID steht dabei für »Prozessidentifikation«. Die PID-Nummer ist dann von Interesse, wenn der Prozess vorzeitig durch `kill` beendet werden soll. Die Nummer in eckigen Klammern gibt die Nummer des Hintergrundprozesses an, der in `bash` gestartet wurde, und ist im Regelfall nicht von Interesse.

Wenn Sie beim Start eines Kommandos das `&`-Zeichen vergessen, brauchen Sie weder zu warten noch müssen Sie das Programm mit `Strg+C` gewaltsam stoppen. Vielmehr sollten Sie das Programm mit `Strg+Z` unterbrechen und mit `bg` als Hintergrundprozess fortsetzen.

Nach dem `&`-Zeichen kann auch ein weiteres Kommando angegeben werden. In diesem Fall wird das erste Kommando im Hintergrund, das zweite dagegen im Vordergrund ausgeführt. Im folgenden Beispiel wird nochmals das obige `find`-Kommando im Hintergrund gestartet. Gleichzeitig wird aber mit `ls` das aktuelle Inhaltsverzeichnis ausgegeben:

**Ausführung  
mehrerer  
Kommandos**

```
user$ find / -name '*sh' > ergebnis & ls
```

Wenn Sie statt des `&`-Zeichens ein Semikolon angeben, führt `bash` die Kommandos nacheinander und im Vordergrund aus:

```
user$ ls; date
```

Das obige Kommando zeigt zuerst das aktuelle Inhaltsverzeichnis an und gibt anschließend das aktuelle Datum aus. Wenn die Gesamtheit dieser Informationen mit `>` in eine Datei umgeleitet werden soll, müssen Sie beide Kommandos in runde Klammern stellen. Dadurch werden beide Kommandos von ein und derselben Shell ausgeführt.

```
user$ (ls; date) > inhalt
```

In der Datei `inhalt` befinden sich nun die von `ls` erstellte Dateiliste sowie das mit `date` ermittelte aktuelle Datum. Die runden Klammern bewirken, dass die beiden Kommandos innerhalb derselben Shell ausgeführt werden und daher auch ein gemeinsames Ergebnis liefern. (Normalerweise wird bei der Ausführung jedes Kommandos eine neue Shell gestartet.)

Mit den Zeichenkombinationen `&&` und `||` können Sie Kommandos bedingt ausführen, d. h. in Abhängigkeit vom Ergebnis eines anderen Kommandos:

```
user$ kommando1 && kommando2
```

führt Kommando 1 aus. Nur wenn dieses Kommando erfolgreich war (kein Fehler, Rückgabewert 0), wird anschließend auch Kommando 2 ausgeführt.

```
user$ kommando1 || kommando2
```

führt Kommando 1 aus. Nur wenn bei der Ausführung dieses Kommandos ein Fehler auftritt (Rückgabewert ungleich 0), wird anschließend auch Kommando 2 ausgeführt.

Weitere Möglichkeiten zur Bildung von Bedingungen und Verzweigungen bietet das Shell-Kommando `if`, das allerdings nur für die Shell-Programmierung von Interesse ist.

## 10.6 Substitutionsmechanismen

Der Begriff *Substitutionsmechanismus* klingt abstrakt und kompliziert. Die Grundidee besteht darin, dass mit Sonderzeichen gebildete Kommandos durch ihre Ergebnisse ersetzt werden. Im einfachsten Fall bedeutet das, dass bei der Auswertung des Kommandos `ls *.tex` die Zeichenkombination `*.tex` durch die Liste der passenden Dateien – etwa `buch.tex command.tex` – ersetzt wird. Das Kommando `ls` bekommt also nicht `*.tex` zu sehen, sondern eine Liste mit realen Dateinamen.

| Kommando                        | Funktion   |
|---------------------------------|--|
| <code>?</code>                  | genau ein beliebiges Zeichen   |
| <code>*</code>                  | beliebig viele (auch null) beliebige Zeichen (aber keine <code>.*</code> -Dateien!)                                  |
| <code>**</code>                 | alle Dateien und Verzeichnisse, auch aus allen Unterverzeichnissen (ab bash 4.0 mit <code>shopt -s globstar</code> ) |
| <code>[abc]</code>              | eines der angegebenen Zeichen  |
| <code>[a-f]</code>              | ein Zeichen aus dem angegebenen Bereich  |
| <code>[!abc]</code>             | keines der angegebenen Zeichen   |
| <code>[^abc]</code>             | wie oben   |
| <code>~</code>                  | Abkürzung für das Heimatverzeichnis  |
| <code>.</code>                  | aktuelles Verzeichnis  |
| <code>..</code>                 | übergeordnetes Verzeichnis   |
| <code>ab{1,2,3}</code>          | liefert <code>ab1 ab2 ab3</code> .   |
| <code>a{1..4}</code>            | liefert <code>a1 a2 a3 a4</code> .   |
| <code>\$(3*4)</code>            | arithmetische Berechnungen   |
| <code>`kommando`</code>         | ersetzt das Kommando durch sein Ergebnis.  |
| <code>\$(kommando)</code>       | wie oben, alternative Schreibweise   |
| <code>kommando "zeichen"</code> | verhindert die Auswertung aller Sonderzeichen außer <code>\$</code> .  |
| <code>kommando 'zeichen'</code> | wie oben, aber noch restriktiver (keine Variablensubstitution)   |

**Tabelle 10.4** Substitutionsmechanismen

Das Ziel dieses Abschnitts ist es, die wichtigsten Mechanismen bei der Interpretation der Kommandozeile vorzustellen (siehe [Tabelle 10.4](#)): Jokerzeichen dienen zur Bildung von Dateinamen, geschweifte Klammern zum Zusammensetzen von Zeichenketten, eckige Klammern zur Berechnung arithmetischer Klammern, umgekehrte Apostrophe zur Kommandosubstitution etc.

Ein Substitutionsmechanismus wird an dieser Stelle unterschlagen, nämlich die sogenannte Parametersubstitution. Damit können Sie in Variablen gespeicherte Zeichenketten analysieren und verändern. Die generelle Syntax lautet `${var__text}`, wobei `var` der Name einer Variablen ist, `__` für ein oder zwei Sonderzeichen steht und `text` das Suchmuster oder eine Defaulteinstellung enthält. Details zum Umgang mit Variablen sowie zu dem Substitutionsmechanismus, der in diesem Zusammenhang eingesetzt wird, folgen in [Abschnitt 10.10](#), »Variablen in bash-Scripts«.

Wenn Sie `rm *.bak` eingeben und das Kommando `rm` tatsächlich alle Dateien löscht, die mit `.bak` enden, dann ist dafür die `bash` verantwortlich. Die Shell durchsucht das aktuelle Verzeichnis nach passenden Dateien und ersetzt `*.bak` durch die entsprechenden Dateinamen.

**Dateinamen-  
bildung mit  
\* und ?**

Als Jokerzeichen sind `?` (genau ein beliebiges Zeichen) und `*` (beliebig viele (auch null) beliebige Zeichen) erlaubt. Die Zeichenkette `[a,b,e-h]*` steht für Dateinamen, die mit einem der Zeichen `a`, `b`, `e`, `f`, `g` oder `h` beginnen. Wenn als erstes Zeichen innerhalb der eckigen Klammern `^` oder `!` angegeben wird, dann sind alle Zeichen außer den angegebenen Zeichen zulässig. `~` kann als Abkürzung für das Heimatverzeichnis verwendet werden.

Die Funktion von Sonderzeichen können Sie einfach mit dem folgenden `echo`-Kommando testen. Das erste Kommando liefert alle Dateien und Verzeichnisse im Wurzelverzeichnis. Das zweite Kommando schränkt die Ausgabe auf Dateien und Verzeichnisse ein, die mit den Buchstaben `a-f` beginnen:

```
user$ echo /*
/bin /boot /dev /etc /home /lib /lost+found /media /misc /mnt /net /opt
/proc /root /sbin /selinux /srv /sys /tmp /usr /var
user$ echo /[a-f]*
/bin /boot /dev /etc
```

Da die Bildung der Dateinamen nicht durch das jeweilige Programm, sondern durch die `bash` erfolgt, sehen die Resultate manchmal anders aus, als Sie es wahrscheinlich erwarten würden. So kann `ls *` zu einer schier endlosen Liste von Dateien führen, auch wenn sich im aktuellen Verzeichnis nur wenige Dateien befinden. Dem Kommando `ls` wird nach der Expansion von `*` eine Liste aller Dateien und Verzeichnisse übergeben.

`ls` wiederum zeigt bei Verzeichnissen nicht einfach deren Namen, sondern den ganzen Inhalt dieser Verzeichnisse an. Wenn Sie nur eine einfache Liste aller Dateien und Verzeichnisse haben möchten, müssen Sie die Option `-d` verwenden. Sie verhindert, dass der Inhalt der Verzeichnisse angezeigt wird, die in der Parameterzeile stehen.

Wenn Sie ein Feedback haben möchten, wie die `bash` intern funktioniert, können Sie `set -x` ausführen. Die `bash` zeigt dann vor der Ausführung jedes weiteren Kommandos an, wie die Kommandozeile ausgewertet wird (mit allen eventuell voreingestellten Optionen und mit den expandierten Dateinamen).

Standardmäßig berücksichtigt `*` keine Dateien oder Verzeichnisse, die mit einem Punkt beginnen (also »verborgen« sind). Wenn Sie diese auch erfassen möchten, müssen Sie mit `shopt` die `bash`-Option `dotglob` setzen:

```
user$ shopt -s dotglob
user$ echo *
...
user$ shopt -u dotglob    (dotglob wieder deaktivieren)
```

#### Dateinamenbildung mit `**`

Die Zeichenkombination `**` erfasst rekursiv alle Dateien und Verzeichnisse. Um die Kompatibilität zu älteren `bash`-Versionen zu wahren, ist diese in Version 4 eingeführte Neuerung standardmäßig nicht aktiv. Wenn Sie sie nutzen möchten (z. B. in einem Script), müssen Sie mit `shopt -s` die `bash`-Option `globstar` setzen.

```
user$ shopt -s globstar
user$ echo **
...
```

#### Zeichenkettenbildung mit `{}`

`bash` setzt aus Zeichenketten, die in geschweiften Klammern angegeben werden, alle denkbaren Zeichenkettenkombinationen zusammen. Die offizielle Bezeichnung für diesen Substitutionsmechanismus lautet *Klammererweiterung* (Brace Expansion). Aus `teil{1,2a,2b}` wird `teil1 teil2a teil2b`. Klammererweiterungen können den Tippaufwand beim Zugriff auf mehrere ähnliche Dateinamen oder Verzeichnisse reduzieren. Gegenüber Jokerzeichen wie `*` und `?` haben sie den Vorteil, dass auch noch nicht existierende Dateinamen gebildet werden können (etwa für `mkdir`).

```
user$ echo {a,b}{1,2,3}
a1 a2 a3 b1 b2 b3
```

```
user$ echo {ab,cd}{123,456,789}-{I,II}
ab123-I ab123-II ab456-I ab456-II ab789-I ab789-II
cd123-I cd123-II cd456-I cd456-II cd789-I cd789-II
```

Aufzählungen können Sie elegant in der Schreibweise `{a..b}` formulieren, wobei `a` und `b` wahlweise Zahlen oder Buchstaben sein dürfen. Die folgenden Beispiele erklären die Funktionsweise besser als jede Beschreibung:

```
user$ echo {1..5}
1 2 3 4 5
```

```
user$ echo {z..t}
z y x w v u t
```

bash ist normalerweise nicht in der Lage, Berechnungen auszuführen. Wenn Sie `2+3` eingeben, weiß die Shell nicht, was sie mit diesem Ausdruck anfangen soll. Wenn Sie innerhalb der Shell eine Berechnung ausführen möchten, müssen Sie den Ausdruck in eckige Klammern setzen und ein `$`-Zeichen voranstellen:

```
user$ echo ${2+3}
5
```

Berechnung  
arithmetischer  
Ausdrücke mit []

Innerhalb der eckigen Klammern sind die meisten aus der Programmiersprache C bekannten Operatoren erlaubt: `+` `-` `*` `/` für die vier Grundrechenarten, `%` für Modulo-Berechnungen, `==` `!=` `<` `<=` `>` `>=` für Vergleiche, `<<` und `>>` für Bitverschiebungen, `!` `&&` und `||` für logisches NICHT, UND und ODER etc. Alle Berechnungen werden für 32-Bit-Integerzahlen ausgeführt (Zahlenbereich zwischen +/-2147483648). Wenn einzelne Werte aus Variablen entnommen werden sollen, muss ein `$`-Zeichen vorangestellt werden.

Eine alternative Möglichkeit, Berechnungen durchzuführen, bietet das Kommando `expr`. Dabei handelt es sich um ein eigenständiges Linux-Kommando, das unabhängig von `bash` funktioniert.

Die Kommandosubstitution ermöglicht es, ein Kommando innerhalb der Kommandozeile durch dessen Ergebnis zu ersetzen. Dazu muss dieses Kommando zwischen zwei ```-Zeichen eingeschlossen werden. Eine alternative Schreibweise lautet `$(kommando)`. Diese Schreibweise ist vorzuziehen, weil sie erstens die Verwirrung durch die Verwendung von drei verschiedenen Anführungszeichen mindert (`"`, `'` und ```) und weil sie zweitens verschachtelt werden kann.

Kommando-  
substitution

Das so gekennzeichnete Kommando wird also durch sein Ergebnis ersetzt. Diese Substitution ermöglicht den verschachtelten Aufruf mehrerer Kommandos, wobei ein Kommando sein Ergebnis an das andere Kommando übergibt. Die beiden folgenden gleichwertigen Kommandos verdeutlichen diesen sehr leistungsfähigen Mechanismus:

```
user$ ls -lgo `find /usr/share -name '*README*`
user$ ls -lgo $(find /usr/share -name '*README*')
```

Durch das obige Kommando wird zuerst `find /usr/share -name '*README*'` ausgeführt. Das Ergebnis dieses Kommandos ist eine Liste aller Dateien im Verzeichnis `/usr/share`, in denen die Zeichenkette `README` vorkommt. Diese Liste wird nun anstelle des `find`-Kommandos in die Kommandozeile eingesetzt. Die Kommandozeile lautet dann beispielsweise:

```
user$ ls -lgo /usr/share/a2ps/ppd/README \
> /usr/share/a2ps/README ...
```

Dieses Kommando führt zum folgenden Ergebnis:

```
-rw-r--r-- 1 301 15. Feb 12:30 /usr/share/a2ps/ppd/README
-rw-r--r-- 1 1029 15. Feb 12:30 /usr/share/a2ps/README
...
```

Dieses Ergebnis wäre durch eine einfache Pipe mit dem `|`-Zeichen nicht möglich. `ls` erwartet keine Eingaben über die Standardeingabe und ignoriert daher auch die Informationen, die `find` über die Pipe liefert. Das folgende Kommando zeigt daher nur einfach den Inhalt des aktuellen Verzeichnisses an. Die Ergebnisse von `find` werden nicht angezeigt!

```
user$ find /usr/share -name '*README*' | ls -l (funktioniert nicht!)
```

**xargs** Es gibt aber eine andere Lösung, die ohne Kommandosubstitution auskommt: Durch die Zuhilfenahme des Kommandos `xargs` werden Daten aus der Standardeingabe an Kommando weitergeleitet, das Sie nach `xargs` angeben.

```
user$ find /usr/share -name '*README*' | xargs ls -l
```

Ein wesentlicher Vorteil von `xargs` besteht darin, dass es kein Größenlimit für die zu verarbeitenden Daten gibt. Gegebenenfalls ruft `xargs` das Kommando mehrfach auf und übergibt die aus der Standardeingabe kommenden Daten in mehreren Schritten. Die Kommandosubstitution ist hingegen durch die maximale Größe einer Kommandozeile – üblicherweise mehrere Tausend Zeichen – begrenzt.

Die Weitergabe von Dateinamen führt zu Problemen, wenn die Dateinamen Leerzeichen enthalten. Diese Probleme können Sie umgehen, indem Sie an `find` die Option `-print0` übergeben und an `xargs` die Option `-null`. Das folgende Kommando setzt bei allen Verzeichnissen das `execute`-Bit:

```
user$ find -type d -print0 | xargs --null chmod a+x
```

#### Sonderzeichen in Zeichenketten

Da in der `bash` praktisch jedes Zeichen mit Ausnahme der Buchstaben und Ziffern irgendeine besondere Bedeutung hat, scheint es so gut wie unmöglich zu sein, diese Zeichen in Zeichenketten oder Dateinamen zu verwenden. Das Problem kann auf zwei Arten gelöst werden: Entweder wird dem Sonderzeichen ein Backslash `\` vorangestellt oder die gesamte Zeichenkette wird in Apostrophe oder Anführungszeichen gestellt. Durch die Angabe von Apostrophen können Sie also beispielsweise eine Datei mit dem Dateinamen `ab* $cd` löschen:

```
user$ rm 'ab* $cd'
```

Beachten Sie bitte den Unterschied zwischen `'` zur Kennzeichnung von Zeichenketten und ``` zur Kommandosubstitution!

Anführungszeichen haben eine ähnliche Wirkung wie Apostrophe. Sie sind allerdings weniger restriktiv und ermöglichen die Interpretation einiger Sonderzeichen wie `$` `\`



und `\`. In Zeichenketten, die in Anführungszeichen gestellt sind, werden daher Shell-Variablen mit vorangestelltem `$`-Zeichen ausgewertet:

```
user$ echo "Das ist der Zugriffspfad: $PATH"
```

Das Kommando liefert als Ergebnis die Zeichenkette »Das ist der Zugriffspfad:«, gefolgt vom Inhalt der Shell-Variablen `PATH`. Wenn statt der Anführungszeichen einfache Apostrophe verwendet werden, wird die gesamte Zeichenkette unverändert durch `echo` ausgegeben.

## 10.7 Shell-Variablen

Die Funktionalität der `bash` und die vieler anderer Linux-Programme wird durch den Zustand sogenannter Shell-Variablen gesteuert. Shell-Variablen sind mit Variablen einer Programmiersprache vergleichbar, können allerdings nur Zeichenketten speichern. Die Zuweisung von Shell-Variablen erfolgt durch den Zuweisungsoperator `=`. Der Inhalt einer Shell-Variablen kann am einfachsten durch `echo` angezeigt werden, wobei dem Variablennamen ein `$`-Zeichen vorangestellt werden muss:

```
user$ var=abc
user$ echo $var
abc
```

Bei Variablenzuweisungen dürfen Sie zwischen dem Variablennamen und dem Zuweisungsoperator `=` kein Leerzeichen angeben. `var = abc` ist syntaktisch falsch und funktioniert nicht!

Wenn Shell-Variablen Leerzeichen oder andere Sonderzeichen enthalten sollen, muss bei der Zuweisung die gesamte Zeichenkette in einfache oder doppelte Hochkomma gestellt werden:

```
user$ var='abc efg'
```

Bei der Zuweisung können mehrere Zeichenketten unmittelbar aneinandergereiht werden. Im folgenden Beispiel wird der Variablen `a` eine neue Zeichenkette zugewiesen, die aus ihrem alten Inhalt, der Zeichenkette »xxx« und nochmals dem ursprünglichen Inhalt besteht:

```
user$ a=3
user$ a=$a'xxx'$a
user$ echo $a
3xxx3
```

Im folgenden Beispiel wird die vorhandene Variable `PATH` mit einer Liste aller Verzeichnisse, die nach ausführbaren Programmen durchsucht werden, um das `bin`-Verzeichnis im Heimatverzeichnis ergänzt. Damit können nun auch alle Kommandos

ausgeführt werden, die sich in diesem Verzeichnis befinden, ohne den Pfad vollständig anzugeben.

```
user$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
user$ PATH=$PATH:/home/kofler/bin
user$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/home/kofler/bin
```

Berechnungen mit Variablen können in der bereits vorgestellten Schreibweise mit eckigen Klammern durchgeführt werden:

```
user$ a=3
user$ a=${a*4}
user$ echo $a
12
```

Wenn das Ergebnis eines Kommandos in einer Variablen gespeichert werden soll, muss die ebenfalls bereits beschriebene Kommandosubstitution mit  $\$(\text{kommando})$  durchgeführt werden. Im folgenden Beispiel wird das aktuelle Verzeichnis in `a` gespeichert:

```
user$ a=$(pwd)
user$ echo $a
/home/kofler
```

Die Inhalte von Variablen werden nur innerhalb der Shell gespeichert. Sie gehen beim Verlassen der Shell wieder verloren. Wenn bestimmte Variablen immer wieder benötigt werden, sollten die Zuweisungen in der Datei `/etc/profile` bzw. in `.profile` im Heimatverzeichnis durchgeführt werden. Diese beiden Dateien werden (sofern vorhanden) beim Start der `bash` automatisch ausgeführt.

Wenn Sie den Inhalt einer Variablen in einer Datei speichern möchten, führen Sie am einfachsten `echo` mit einer Ausgabeumleitung durch:

```
user$ echo $var > datei
```

### Lokale und globale Variablen (Umgebungsvariablen)

Die Begriffe »lokal« und »global« zur Beschreibung von Variablen sind aus der Welt der Programmiersprachen entlehnt. Bei Shell-Variablen gilt eine Variable dann als global, wenn sie beim Start eines Kommandos oder eines Shell-Programms weitergegeben wird. Globale Variablen werden oft auch als Umgebungsvariablen (*Environment Variables*) bezeichnet.

Beachten Sie bitte, dass alle durch eine einfache Zuweisung entstandenen Variablen nur als lokal gelten! Um eine globale Variable zu definieren, müssen Sie `export` oder `declare -x` aufrufen.

Zur Variablenverwaltung innerhalb der Shell existieren zahlreiche Kommandos, wobei es funktionelle Überlappungen gibt. Zur Definition einer globalen Variablen können Sie beispielsweise sowohl `export` als auch `declare -x` verwenden. Die folgenden Beispiele versuchen, die Verwirrung durch ähnliche Kommandos ein wenig zu mindern:

|                             |   |
|-----------------------------|---|
| <code>a=3</code>            | Kurzschreibweise für <code>let</code> , <code>a</code> ist lokal.                     |
| <code>declare a=3</code>    | weist der lokalen Variablen <code>a</code> einen Wert zu (wie <code>let</code> ).     |
| <code>declare -x a=3</code> | weist der globalen Variablen <code>a</code> einen Wert zu (wie <code>export</code> ). |
| <code>export</code>         | zeigt alle globalen Variablen an.   |
| <code>export a</code>       | macht <code>a</code> zu einer globalen Variablen.                                     |
| <code>export a=3</code>     | weist der globalen Variablen <code>a</code> einen Wert zu.                            |
| <code>let a=3</code>        | weist der lokalen Variablen <code>a</code> einen Wert zu.                             |
| <code>local a=3</code>      | definiert <code>a</code> als lokal (nur in Shell-Funktionen).                         |
| <code>printenv</code>       | zeigt wie <code>export</code> alle globalen Variablen an.                             |
| <code>set</code>            | zeigt <i>alle</i> Variablen an (lokale und globale).                                  |
| <code>unset a</code>        | löscht die Variable <code>a</code> .  |

Wenn Sie Variablen einrichten, die das Verhalten von anderen Linux-Kommandos steuern sollen, müssen diese Variablen immer global sein! Damit Sie einerseits die Substitutionsmechanismen der Shell ausnutzen und andererseits globale Variablen definieren können, sollten Sie Variablen zuerst mit `x=...` zuweisen und anschließend mit `export x` als global definieren.

Variablenzuweisungen gelten immer nur für *eine* Shell. Wenn Sie in mehreren Terminals bzw. Terminalfenstern arbeiten, laufen darin jeweils eigenständige und voneinander unabhängige Shells. Die Veränderung einer Variablen in einer Shell hat keinerlei Einfluss auf die anderen Shells. Sie können aber oft benötigte Variablenzuweisungen in der Datei `.profile` festlegen, die automatisch beim Start jeder Shell ausgeführt wird.

### Wichtige Shell-Variablen

Prinzipiell können Sie beliebig viele neue Variablen einführen und nach Gutdünken benennen und verwenden. Dabei sollten Sie aber versuchen, bereits vorhandene Variablen zu vermeiden, da diese zumeist von der `bash` und häufig auch von anderen Linux-Kommandos ausgewertet werden. Eine unkontrollierte Veränderung dieser Variablen kann zur Folge haben, dass die Verarbeitung von Kommandos nicht mehr richtig funktioniert, dass Linux plötzlich Dateien nicht mehr findet etc. Dieser Abschnitt beschreibt die wichtigsten Shell-Variablen in alphabetischer Reihenfolge:

#### BASH

enthält den Dateinamen der `bash`.

**HOME**

enthält den Pfad des Heimatverzeichnisses, beispielsweise `/home/mk`.

**LOGNAME**

enthält den Login-Namen (User-Namen).

**HOSTNAME**

enthält den Hostnamen (Rechnernamen).

**MAIL**

enthält den Pfad des Verzeichnisses, in dem ankommende Mail gespeichert wird (nur, wenn ein lokaler Mail-Server installiert ist).

**OLDPWD**

enthält den Pfad des zuletzt aktiven Verzeichnisses.

**PATH**

enthält eine Liste von Verzeichnissen. Wenn die `bash` ein Kommando ausführen soll, durchsucht sie alle in `PATH` aufgezählten Verzeichnisse nach dem Kommando. Die Verzeichnisse sind durch Doppelpunkte voneinander getrennt.

Die Einstellung von `PATH` erfolgt distributionsspezifisch an verschiedenen Stellen während des Startprozesses (Init-V, Upstart). Der beste Ort, um eigene Änderungen durchzuführen, ist `/etc/profile` bzw. (wenn Ihre Distribution dies vorsieht) eine Datei im Verzeichnis `/etc/profile.d`. Dort fügen Sie ein Kommando nach dem folgenden Muster ein:

```
# Ergänzung in /etc/profile oder in /etc/profile.d/myown.sh
PATH=$PATH:/myown/bin
```

Aus Sicherheitsgründen (um das unbeabsichtigte Ausführen von Programmen im aktuellen Verzeichnis zu vermeiden) fehlt in `PATH` das lokale Verzeichnis. Wenn Sie Programme im gerade aktuellen Verzeichnis ohne vorangestelltes `./` ausführen möchten, müssen Sie `PATH` um `.` erweitern.

**PROMPT\_COMMAND**

kann ein Kommando enthalten, das jedes Mal ausgeführt wird, bevor die `bash` den Kommandoprompt anzeigt.

**PS1**

enthält eine Zeichenkette, deren Inhalt am Beginn jeder Eingabezeile angezeigt wird (Prompt). Innerhalb dieser Zeichenkette sind unter anderem folgende Zeichenkombinationen vorgesehen: `\t` für die aktuelle Zeit, `\d` für das Datum, `\w` für das aktuelle Verzeichnis, `\W` für den letzten Teil des aktuellen Verzeichnisses (also `X11` für `/usr/bin/X11`), `\u` für den User-Namen, `\h` für den Hostnamen (Rechnernamen) sowie `\$` für das Promptzeichen (`$` für normale Anwender, `#` für `root`).

**PS2**

wie PS1, allerdings wird die Zeichenkette nur bei mehrzeiligen Eingaben angezeigt, also wenn die erste Zeile mit `\` abgeschlossen wurde. Eine typische Einstellung lautet `">"`.

**PWD**

enthält den Pfad des aktuellen Verzeichnisses.

Neben den hier beschriebenen Variablen sind normalerweise zahlreiche weitere Umgebungsvariablen definiert, die Funktionen der Shell sowie diverser anderer Programme steuern. Eine Liste aller definierten Variablen erhalten Sie mit `printenv | sort`.

**RANDOM**

liefert bei jedem Auslesen eine neue, zufällige Zahl zwischen 0 und 32767.

## 10.8 bash-Script-Beispiele

Shell-Programme sind einfache Textdateien mit einigen Linux- und/oder bash-Kommandos. Nach dem Start eines Shell-Programms werden diese Kommandos der Reihe nach ausgeführt. Dem Shell-Programm können Parameter wie einem normalen Kommando übergeben werden. Diese Parameter können innerhalb des Programms ausgewertet werden.

Da die einfache sequenzielle Ausführung einiger Kommandos wenig Spielraum für komplexe Aufgabenstellungen lässt, kennt die bash Kommandos zur Bildung von Verzweigungen, Schleifen und Funktionen. Damit steht Ihnen eine echte Programmiersprache zur Verfügung, für die Sie weder einen Compiler noch C-Kenntnisse benötigen.

Typische Anwendungen für Shell-Programme sind die Automatisierung von oft benötigten Kommandofolgen zur Installation von Programmen, zur Administration des Systems, zur Durchführung von Backups, zur Konfiguration und Ausführung einzelner Programme etc.

Die folgenden Seiten geben nur eine erste Einführung in die Programmierung mit der bash. Unzählige weitere Informationen und Beispiele finden Sie auf der ausgezeichneten Website <http://bash-hackers.org>.

Unter Linux wimmelt es nur so von Beispielen für die bash-Programmierung, auch wenn Sie bisher möglicherweise nichts davon bemerkt haben. Viele Kommandos, die Sie während der Installation, Konfiguration und Administration von Linux ausführen, sind in Wirklichkeit bash-Programme.

Scripts suchen

Das folgende `find/grep`-Kommando durchsucht das Verzeichnis `/etc/` nach shell-Programmen. Dabei werden alle Dateien erkannt, die als ausführbar gekennzeichnet sind und die die Zeichenkette `\#! ... sh` enthalten. Die Ausführung des Kommandos nimmt einige Zeit in Anspruch, weil das gesamte Dateisystem durchsucht wird.

```
user$ find /etc -type f -perm +111 -exec grep -q '#!.*sh' {} \; -print
```

### Beispiel 1: grepall

Angenommen, Sie verwenden häufig die Kommandos `grep` und `find`, um im gerade aktuellen Verzeichnis und allen Unterverzeichnissen nach Dateien zu suchen, die eine bestimmte Zeichenkette enthalten. Das richtige Kommando sieht so aus:

```
user$ find . -type f -exec grep -q suchtext {} \; -print
```

Wenn Sie wie ich jedes Mal neu rätseln, welche Kombination der Optionen dazu erforderlich ist, liegt es nahe, das neue Kommando `grepall` zu definieren, das eben diese Aufgabe übernimmt. Dazu starten Sie Ihren Lieblingseditor, um die Textdatei `grepall` zu schreiben. Die Datei besteht aus nur zwei Zeilen, wobei die erste den Programmnamen des Interpreters angibt, der die Script-Datei ausführen soll.

```
#!/bin/bash
find . -type f -exec grep -q $1 {} \; -print
```

#### Kleine Textdateien ohne Editor erstellen

Wenn Sie sich den Editoraufruf sparen möchten, können Sie die Datei auch mit `cat` erstellen: Geben Sie das Kommando `cat > grepall` ein. Das Kommando erwartet jetzt Daten aus der Standardeingabe (Tastatur) und schreibt diese in die Datei `grepall`. Geben Sie nun das Kommando mit all seinen Optionen ein. Anschließend beenden Sie `cat` mit `[Strg]+[D]` (das entspricht EOF, also *end of file*). Die resultierende Datei können Sie mit `cat grepall` ansehen.

**Shebang** Die erste Zeile des bash-Scripts wird *Shebang* genannt und beginnt mit der Zeichenkombination `#!`. Danach folgt der Pfad zum Interpreter, der den folgenden Code ausführen soll. Für dieses und die folgenden Beispiele ergibt sich damit die Zeile `#!/bin/bash`.

Anstelle von `/bin/bash` können Sie auch `/bin/sh` angeben. Bei vielen Distributionen wird das Script dann ebenfalls von der `bash` ausgeführt, weil `/bin/sh` einfach ein Link auf die `bash` ist. Vorsicht: Unter Debian und Ubuntu zeigt `/bin/sh` aber auf den Shell-Interpreter `dash`. Dieses Programm ist auf höhere Geschwindigkeit optimiert, aber nicht vollständig kompatibel zur `bash`.

Der Versuch, die gerade erstellte Datei `grepall` auszuführen, endet mit der Fehlermeldung *permission denied*. Der Grund für diese Meldung besteht darin, dass bei neuen Dateien generell die Zugriffsbits (x) zum Ausführen der Datei deaktiviert sind. Das können Sie aber rasch mit `chmod` ändern. `grepall abc` liefert jetzt die gewünschte Liste aller Dateien, die die Zeichenkette »abc« enthalten:

Execution-Bit

```
user$ ./grepall abc
bash: ./grepall: Permission denied
user$ chmod a+x grepall
user$ ./grepall abc
./bashprg.tex
```

Damit Sie das Kommando `grepall` ausführen können, ohne jedesmal das Verzeichnis anzugeben, in dem der Code gespeichert ist, müssen Sie die Datei in ein Verzeichnis verschieben, das in `$PATH` enthalten ist. Hierfür bietet sich `/usr/local/bin` an. Dort ist das Kommando allen Benutzern zugänglich.

```
root# mv grepall /usr/local/bin
```

### Beispiel 2: stripcomments

Auch das zweite Beispiel ist ein Einzeiler. Sie übergeben an das Kommando `stripcomments` eine Textdatei. Die drei verschachtelten `grep`-Kommandos eliminieren nun alle Zeilen, die mit den Zeichen `#` oder `;` beginnen bzw. ganz leer sind. Kommentare werden auch dann entfernt, wenn sich vor den Zeichen `#` oder `;` Leer- oder Tabulatorzeichen befinden. Das Kommando eignet sich ausgezeichnet dazu, um bei Konfigurationsdateien alle Kommentarzeilen zu entfernen und nur die tatsächlich gültigen Einstellungen anzuzeigen.

```
#!/bin/bash
grep -Ev '^[[:space:]]*#|^[[:space:]]*|^\$' $1
```

Kurz zur Erklärung: Das Muster `^[[:space:]]*#\` findet Zeilen, die mit `#` beginnen, wobei zwischen dem Zeilenanfang (`^`) und `#` beliebig viele Leer- und Tabulatorzeichen sein dürfen. Analog erfasst der Ausdruck `^[[:space:]]*;` alle Zeilen, die mit `;` beginnen. Das dritte Muster gilt für leere Zeilen, die nur aus Zeilenanfang und Zeilenende (`$`) bestehen.

Die Option `-v` invertiert die übliche Funktion von `grep`: Statt die gefundenen Zeilen zu extrahieren, liefert `grep` nun alle Zeilen, auf die das Muster *nicht* zutrifft. Die Option `-E` aktiviert die erweiterte `grep`-Syntax, die die Kombination mehrerer Suchausdrücke mit dem Zeichen `|` erlaubt.

### Beispiel 3: `applysedfile`

Die beiden obigen Beispiele zeigen zwar gut, wie Sie sich etwas Tipp- und Denkarbeit ersparen können, deuten die weitreichenden Möglichkeiten der Script-Programmierung aber noch nicht einmal an. Schon mehr bietet in dieser Hinsicht das nächste Beispiel: Nehmen Sie an, Sie stehen vor der Aufgabe, in einem ganzen Bündel von Dateien eine Reihe gleichartiger Suchen-und-Ersetzen-Läufe durchzuführen. Das kommt immer wieder vor, wenn Sie in einem über mehrere Dateien verteilten Programmcode einen Variablen- oder Prozedurnamen verändern möchten. Ich stand bei der Überarbeitung dieses Buchs für die fünfte Auflage aufgrund der neuen Rechtschreibung vor einem ähnlichen Problem: In Dutzenden von `*.tex`-Dateien sollte »daß« durch »dass«, »muß« durch »muss« etc. ersetzt werden.

Das Script-Programm `applysedfile` hilft bei derartigen Aufgaben. Der Aufruf dieses Scripts sieht folgendermaßen aus:

```
user$ applysedfile *.tex
```

Das Programm erstellt nun von allen `*.tex`-Dateien eine Sicherheitskopie `*.bak`. Anschließend wird das Unix-Kommando `sed` verwendet, um eine ganze Liste von Kommandos für jede `*.tex`-Datei auszuführen. Diese Kommandos müssen sich in der Datei `./sedfile` befinden, die von `applysedfile` automatisch benutzt wird. Der Code von `applysedfile` sieht folgendermaßen aus:

```
#!/bin/bash
# Beispiel applysedfile
# Verwendung: applysedfile *.tex
#          wendet ./sedfile auf die Liste der übergebenen Dateien an
for i in $*
do
    echo "process $i"
    # make a backup of old file
    cp $i ${i%.*}.bak
    # build new file
    sed -f ./sedfile < ${i%.*}.bak > $i
done
```

Kurz einige Anmerkungen zur Funktion dieses kleinen Programms: Bei den vier ersten Zeilen handelt es sich um Kommentare, die mit dem Zeichen `#` eingeleitet werden.

`for` leitet eine Schleife ein. Für jeden Schleifendurchgang wird ein Dateiname in die Variable `i` eingesetzt. Die Liste der Dateinamen stammt aus `$*`. Diese Zeichenkombination ist ein Platzhalter für alle an das Programm übergebenen Parameter und Dateinamen.

Der Schleifenkörper gibt den Namen jeder Datei aus. Mit `cp` wird eine Sicherungskopie der Datei erstellt. (Dabei werden zuerst alle Zeichen ab dem ersten Punkt im



Dateinamen gelöscht. Anschließend wird `.bak` angehängt.) Schließlich wird das Kommando `sed` für die Datei ausgeführt, wobei die Steuerungsdatei `sedfile` aus dem lokalen Verzeichnis verwendet wird.

Für die Umstellung auf die neue Rechtschreibung sahen die ersten Zeilen dieser Datei wie folgt aus:

```
s.daß.dass.g
s.muß.muss.g
s.paßt.passt.g
s.läßt.lässt.g
```

Dabei handelt es sich bei jeder Zeile um ein `sed`-Kommando, das die erste Zeichenkette durch die zweite ersetzt (Kommando `s`). Der nachgestellte Buchstabe `g` bedeutet, dass das Kommando auch mehrfach innerhalb einer Zeile ausgeführt werden soll (falls »daß« oder »muß« mehrere Male innerhalb einer Zeile auftreten sollte).

#### Beispiel 4: Backup-Script

Das folgende Script wird jede Nacht automatisch auf meinem root-Server ausgeführt. Als Erstes wird die Variable `m` initialisiert, die den aktuellen Monat als Zahl enthält. Das Kommando `date` liefert das aktuelle Datum samt Uhrzeit. Die Formatzeichenkette `+%m` extrahiert daraus den Monat.

Nun erstellt `tar` ein Backup des Verzeichnisses `/var/www`. Das Archiv wird nicht direkt in einer Datei gespeichert, sondern mittels `|` an das Kommando `curl` weitergeleitet. `curl` überträgt die Daten auf einen FTP-Server (Benutzername `kofler`, Passwort `xxxx`, IP-Adresse `1.2.3.4`). Auf dem FTP-Server wird das Backup unter dem Namen `www-MM.tgz` gespeichert, wobei `MM` den Monat angibt (01 bis 12).

Auf diese Weise entstehen im Verlauf eines Jahres monatliche Backup-Versionen, sodass ich zur Not auch einen alten Zustand meiner Website rekonstruieren kann, sollte das erforderlich sein. Gleichzeitig ist der Platzbedarf der Backup-Dateien gering. Zu jedem Zeitpunkt gibt es maximal 12 Versionen, also `www-01.tgz` bis `www-12.tgz`.

Das Kommando `mysqldump` erstellt ein Backup der MySQL-Datenbank `cms`, in der das Content-Management-System (CMS) meiner Website alle Seiten und unzählige andere Daten speichert. Abermals wird das Backup mittels `|` an `curl` weitergegeben und auf meinem FTP-Server gespeichert.

```
#!/bin/bash
m=$(date +%m)
cd /var
tar czf - www | curl -T - -u kofler:xxxx ftp://1.2.3.4/www-$m.tgz
mysqldump -u cms -pxxxx cms | curl -T - -u kofler:xxxx ftp://1.2.3.4/cms-$m.sql
```

Das gesamte Script habe ich unter dem Dateinamen `/etc/myscripts/backup` gespeichert. Um den täglichen Aufruf kümmert sich Cron (siehe [Abschnitt 12.6](#), »Prozesse automatisch starten (Cron)«). Die dazu passende Konfigurationsdatei `/etc/cron.d/backup` sieht so aus:

```
# jeden Sonntag um 3:15
15 3 * * 0 root /etc/myscripts/backup
```

### Beispiel 5: Thumbnails erzeugen

Als »Thumbnails« werden verkleinerte Versionen von Bilddateien bezeichnet. Das folgende Script wird in der Form `makethumbs *.jpg` aufgerufen. Es erzeugt das Unterverzeichnis `400x400` und speichert dort verkleinerte Kopien der ursprünglichen Bilder. Die Maximalgröße der neuen Bilder beträgt  $400 \times 400$  Pixel, wobei die Proportionen des Originalbilds erhalten bleiben. Bilder, die kleiner sind, bleiben unverändert, werden also nicht vergrößert.

Das Script wendet das `convert`-Kommando aus dem Paket Image Magick an. Für die Verkleinerung ist die Option `-resize` verantwortlich. `-size` bewirkt lediglich eine schnellere Verarbeitung.

```
#!/bin/bash
# Verwendung: makethumbs *.jpg
if [ ! -d 400x400 ]; then # Unterverzeichnis erzeugen
    mkdir 400x400
fi
for filename do          # alle Dateien verarbeiten
    echo "processing $filename"
    convert -size 400x400 -resize 400x400 $filename 400x400/$filename
done
```

### Beispiel 6: Studenten-Accounts einrichten

Das folgende Script habe ich benötigt, als ich zu Beginn einer Linux-Lehrveranstaltung für alle Studenten und Studentinnen einen Account einrichten wollte. Ausgangspunkt war die Datei `students.txt`, die zeilenweise die Nachnamen der Studenten enthielt:

```
huber
mueller
schmidt
...
```

Nun hätte ich jeden Studenten einzeln einrichten können, entweder mit einer grafischen Benutzeroberfläche oder durch die manuelle Ausführung von `useradd`, `passwd` und `chage`. Stattdessen habe ich das folgende Mini-Script verfasst:

```
#!/bin/bash
while read s; do
    pw="$s"-1234"
    useradd $s
    echo -e "$pw\n$pw" | passwd --stdin $s
    chage -d 0 -E 2017-12-31 $s
done < studenten.txt
```

Das Script durchläuft in der `while`-Schleife die Namen aller Studenten. Der gerade aktuelle Name befindet sich in der Variablen `s`. Die Variable `pw` enthält das Startpasswort, das sich aus dem Namen plus `-1234` zusammensetzt, also beispielsweise `huber-1234`. `useradd` richtet den Account ein. `echo` gibt zweimal das Passwort aus, getrennt durch ein Zeilenendezeichen. Diese Ausgabe wird freilich nicht angezeigt, sondern mit `|` an das `passwd`-Kommando weitergeleitet. Auf diese Weise wird das Passwort eingerichtet. (Das `passwd`-Kommando erwartet aus Sicherheitsgründen eine Wiederholung des Passworts. Deswegen muss `$pw` zweimal an `passwd` übergeben werden.)

`chage -d 0` bewirkt, dass jeder Student unmittelbar nach dem ersten Login sein Passwort ändern muss. Die Option `-E 2017-12-31` hat zur Folge, dass die Accounts Ende Dezember 2017 auslaufen und dann nicht mehr genutzt werden können. Um die Accounts samt der zugeordneten `/home`-Verzeichnisse dann wieder zu löschen, gibt es ein weiteres Script:

```
#!/bin/bash
while read s; do
    userdel -r $s
done < studenten.txt
```

### Beispiel 7: Mehrere MySQL/MariaDB-Datenbanken ändern

In meiner Funktion als Datenbankadministrator muss ich gelegentlich an einer Menge gleichartiger Datenbanken Änderungen durchführen – z. B. bei 50 in ihrer Struktur gleichartigen Kundendatenbanken eine Spalte zu einer Tabelle hinzufügen. Manuell müsste ich dazu mit `mysql dbname` eine Verbindung zu jeder Datenbank herstellen und dann `ALTER TABLE tblname ADD ...` ausführen. Drei Zeilen `bash`-Code sorgen für mehr Effizienz:

```
#!/bin/bash
for db in $(cat /etc/mydbs.txt); do
    mysql $db < updates.sql
done
```

Das Script durchläuft alle in der Datei `/etc/mydbs.txt` zeilenweise aufgelisteten Datenbanken. Für jede Datenbank führt `mysql` nun die in der Datei `updates.sql` gespeicherten SQL-Kommandos aus. Die einzige Voraussetzung besteht darin, dass der Benutzer,

der das Script ausführt, ohne explizite Passwortangabe eine Verbindung zu allen Datenbanken herstellen kann. Damit das funktioniert, müssen sich in der MySQL-spezifischen Datei `.my.cnf` (oder ab MySQL 5.6 in `.mylogin.cnf`) die erforderlichen Authentifizierungsdaten befinden.

## 10.9 bash-Script-Grundregeln

**Shebang** Wie ich bereits bei der Beschreibung der Beispiele aus dem vorigen Abschnitt erwähnt habe, beginnen `bash`-Scripts mit der Zeile `#!/bin/bash`. Diese Zeile, der sogenannte *Shebang*, gibt an, durch welchen Interpreter das Script ausgeführt wird – in diesem Kapitel eben durch die `bash`.

### Vorsicht mit Sonderzeichen

In der ersten Zeile eines Scripts dürfen keine deutschen Sonderzeichen verwendet werden, auch nicht in Kommentaren. Die `bash` weigert sich sonst, die Datei auszuführen, und liefert die Meldung *cannot execute binary file*.

In Shell-Script-Dateien dürfen die Zeilen nicht durch die Windows-typische Kombination aus Carriage Return und Linefeed getrennt sein. Das kann z. B. passieren, wenn die Dateien unter Windows erstellt und dann nach Linux kopiert wurden. In diesem Fall liefert `bash` die wenig aussagekräftige Fehlermeldung *bad interpreter*. Bei Unicode-Dateien (UTF8) sorgt das folgende Kommando für die richtige Zeilentrennung:

```
recode u8/cr-lf..u8 < windowsdatei > \ linuxdatei
```

**Verhalten im Fehlerfall** In den meisten Programmiersprachen gilt: Sobald ein unbehandelter Fehler auftritt, wird das Programm beendet. Wesentlich entspannter geht es in `bash`-Scripts zu: Dass bei der Kommandoausführung Fehler passieren, dass also ein Kommando mit einem Fehlercode ungleich 0 endet, wird als etwas Alltägliches hingenommen. Die Ausführung des Scripts wird einfach mit der nächsten Anweisung fortgesetzt.

Das gilt allerdings nicht für offensichtliche Syntaxfehler, z. B. für eine `for`-Schleife ohne `done` oder für eine Zeichenkette, die mit `"` beginnt, bei der das zweite `"`-Zeichen aber fehlt: In solchen Fällen kann die `bash` die weitere Programmstruktur nicht mehr entschlüsseln: Sie zeigt eine Fehlermeldung an und bricht die Codeausführung ab. Anweisungen bis zum ersten Syntaxfehler werden aber auch in diesem Fall ausgeführt.

Auch wenn die hohe Fehlertoleranz der `bash` oft bequem ist – manchmal wollen Sie, dass ein Script beim ersten Fehler abbricht und nicht womöglich noch mehr Folge-

fehler produziert. Das erreichen Sie, wenn Sie in den Code `set -e` einbauen. Ab dieser Position führen Fehler zum sofortigen Ende. Wenn diese strikte Fehlerkontrolle für das ganze Script gelten soll, geben Sie die Option `-e` gleich in der ersten Zeile an:

```
#!/bin/bash -e
# dieses Script wird beim ersten Fehler abgebrochen
...
```

Bei miteinander verknüpften Kommandos gelten `set -e` bzw. die bash-Option `-e` nur für das Gesamtergebnis. Wenn im folgenden Beispiel `cmd1` scheitert – und sei es daran, dass es das Kommando gar nicht gibt –, dann wird `cmd2` ausgeführt. Nur wenn auch dieses Kommando zu einem Fehler führt, wird das Script abgebrochen.

```
#!/bin/bash -e
cmd1 || cmd2
```

Shell-Scripts können nur ausgeführt werden, wenn die Zugriffsbits für den Lesezugriff (`r`) und die Ausführung (`x`) gesetzt sind (`chmod ug+rx datei`). Falls sich Scripts auf externen Datenträgern bzw. Partitionen befinden, müssen Sie sicherstellen, dass das Dateisystem mit der `exec`-Option in den Verzeichnisbaum eingebunden ist.

Execute-Bit

Wenn Sie eine Sammlung eigener Shell-Script-Programme für den täglichen Gebrauch schreiben, ist es sinnvoll, diese an einem zentralen Ort zu speichern. Als Verzeichnis bietet sich `~/bin` an. Wenn Sie anschließend folgende Änderung in `.profile` vornehmen, können diese Script-Programme ohne eine komplette Pfadangabe ausgeführt werden. (Bei manchen Distributionen ist das gar nicht notwendig, dort ist `~/bin` immer Bestandteil von `PATH`.)

bin-Verzeichnis

```
# Ergänzung in ~/.profile bzw. in ~/.bashrc
PATH=$PATH:~/bin'
```

## 10.10 Variablen in bash-Scripts

Erste Informationen zum Umgang mit Variablen habe ich bereits in [Abschnitt 10.7](#), »Shell-Variablen«, gegeben. Dort ist unter anderem der Unterschied zwischen normalen Shell-Variablen und Umgebungsvariablen beschrieben. In diesem Abschnitt werden weitere Aspekte der Variablenverwaltung behandelt, die besonders für die Shell-Programmierung relevant sind. Im Detail geht es um den Gültigkeitsbereich von Variablen, um einige in der bash vordefinierte Variablen (z. B. `$*` oder `$?`), um den Mechanismus der Parametersubstitution zur Analyse und Verarbeitung von Zeichenketten in Variablen und schließlich um die Eingabe von Variablen in Shell-Programmen.

## Der Gültigkeitsbereich von Variablen

Um die Feinheiten der Variablenverwaltung bei der Ausführung von Shell-Programmen zu verstehen, sind Grundkenntnisse über die Mechanismen beim Start von Kommandos und Shell-Programmen erforderlich.

Zur Ausführung eines Kommandos oder eines Programms erzeugt die `bash` einen neuen Prozess mit einer eigenen PID-Nummer. Das ist eine Linux-interne Nummer zur Identifizierung und Verwaltung des Prozesses. Von den Shell-Variablen werden nur jene an den neuen Prozess weitergegeben, die als Umgebungsvariablen deklariert wurden (`export` oder `declare -x`). Wenn ein Kommando im Vordergrund gestartet wird, tritt die `bash` während der Ausführung in den Hintergrund und wartet auf das Ende des Kommandos. Andernfalls laufen beide Programme parallel, also die `bash` und das im Hintergrund gestartete Programm.

Einen Sonderfall stellt der Start eines Shell-Programms dar. Die Abarbeitung des Shell-Programms erfolgt nämlich nicht in der laufenden Shell, sondern in einer eigens dazu gestarteten Subshell. Es laufen nun also zwei Instanzen der `bash` – die eine als ihr Kommandointerpreter und die zweite zur Ausführung des Shell-Programms. Wenn innerhalb dieses Programms ein weiteres Shell-Programm gestartet wird, wird dazu eine dritte `bash`-Instanz gestartet usw. Die Ausführung eigener Subshells für Shell-Programme ist erforderlich, damit mehrere Shell-Programme parallel und ohne gegenseitige Beeinflussung gegebenenfalls auch im Hintergrund ausgeführt werden können.

Das Konzept der Subshells wirkt sich insofern auf die Variablenverwaltung aus, als jede (Sub-)Shell eigene Variablen besitzt. Der Subshell werden wie beim Start jedes beliebigen anderen Programms nur die Variablen der interaktiven Shell übergeben, die als Umgebungsvariablen deklariert waren. Anschließend sind die Variablen in den beiden Shells unabhängig voneinander: Die Veränderung von Variablen in der einen Shell hat keinerlei Einfluss auf Variablen der anderen Shell.

Manchmal möchte man mit einem Shell-Programm neue Variablen deklarieren bzw. vorhandene Variablen bleibend verändern. Um das zu ermöglichen, können Sie Shell-Programme auch innerhalb der aktuellen `bash` ausführen, also ohne den automatischen Start einer Subshell. Dazu müssen Sie vor den Dateinamen des Shell-Programms einen Punkt und ein Leerzeichen stellen. Das entspricht der Kurzschreibweise des Shell-Kommandos `source`.

Dazu ein Beispiel: Sie möchten ein Shell-Programm schreiben, das die `PATH`-Variable um den Pfad des gerade aktuellen Verzeichnisses erweitert. Das erforderliche Programm `addpwd` ist ganz einfach:

```
#!/bin/bash
# Shell-Programm addpwd ergänzt den Pfad um das aktuelle Verzeichnis
#
PATH=$PATH:"$(pwd)
```

In der Variablen `PATH` werden also der bisherige Inhalt dieser Variablen, ein Doppelpunkt und schließlich via Kommandosubstitution das Ergebnis des Kommandos `pwd` gespeichert. Der folgende Testlauf beweist, dass sich der Inhalt der `PATH`-Variablen in der aktuellen Shell erst dann ändert, wenn `addpwd` mit einem vorangestellten Punkt gestartet wird. Innerhalb der Subshell, die beim ersten Aufruf von `addpwd` gestartet wurde, wird `PATH` natürlich auch geändert – aber diese Änderung gilt nur, solange `addpwd` läuft.

```
user$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
user$ addpwd
user$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
user$ . addpwd
user$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/home/user
```

### Durch die Shell vordefinierte Variablen

Innerhalb von Shell-Programmen kann auf einige von der `bash` vordefinierte Variablen zugegriffen werden. Diese Variablen können nicht durch Zuweisungen verändert, sondern nur gelesen werden. Der Name der Variablen wird durch verschiedene Sonderzeichen gebildet. In [Tabelle 10.5](#) werden die Variablen gleich mit dem vorangestellten `$`-Zeichen angegeben.

| Variable                  | Bedeutung  |
|---------------------------|--|
| <code>\$?</code>          | Rückgabewert des letzten Kommandos   |
| <code>#!</code>           | PID des zuletzt gestarteten Hintergrundprozesses   |
| <code>\$\$</code>         | PID der aktuellen Shell  |
| <code>\$0</code>          | Dateiname des gerade ausgeführten Shell-Scripts (oder des symbolischen Links, der auf die Datei zeigt) |
| <code>\$#</code>          | Anzahl der dem Shell-Programm übergebenen Parameter  |
| <code>\$1 bis \$9</code>  | Parameter 1 bis 9  |
| <code>\$* oder \$@</code> | Gesamtheit aller übergebenen Parameter   |

**Tabelle 10.5** `$`-Variablen

Noch einige Anmerkungen zur Anwendung dieser Variablen: \$0 bis \$9, \$# und \$\* dienen zur Auswertung der Parameter, die dem Batch-Programm übergeben wurden. Beinahe jedes Script-Beispiel in diesem Kapitel zeigt dafür Anwendungsmöglichkeiten.

Im Zusammenhang mit der Auswertung von Parametern ist das `bash`-Kommando `shift` interessant. Dieses Kommando schiebt die übergebenen Parameter quasi durch die zehn Variablen \$0 bis \$9. Wenn Sie `shift 9` ausführen, gehen die ersten neun dem Programm übergebenen Parameter verloren, dafür können jetzt aber die nächsten neun bequem angesprochen werden. `shift` ohne weitere Angaben verschiebt die Parameterliste um einen Parameter.

`$?`  kann zur Bildung von Bedingungen verwendet werden, um den weiteren Programmverlauf vom Ergebnis des letzten Kommandos abhängig zu machen. Prinzipiell ist es auch möglich, ein Kommando direkt als Bedingung in `if` anzugeben. Die Variable  `$?`  hat den Vorteil, dass allzu lange und unübersichtliche Anweisungen vermieden werden können.

Die Variable  `$$`  enthält die PID (*Process Identification Number*). Dieser Zahlenwert wird Linux-intern zur Verwaltung der Prozesse verwendet. Die PID ist eindeutig, d. h., im ganzen System existiert mit Sicherheit kein zweiter Prozess mit derselben Nummer. Deswegen eignet sich dieser Wert hervorragend zur Bildung einer temporären Datei. Beispielsweise speichern Sie mit `ls > tmp.$$` eine Liste aller Dateien in der Datei `tmp.nnn`. Selbst wenn dieselbe Stapeldatei gleichzeitig in einem anderen Terminal läuft, wird es wegen der unterschiedlichen PIDs der beiden Shells mit Sicherheit zu keinem Namenskonflikt kommen.

## Felder (Arrays)

Neben einfachen Variablen kennt die `bash` auch Felder (Arrays). Bis einschließlich Version 3 muss der Index eine Zahl sein. Beachten Sie die von C abweichende Syntax  `${feld[n]}`  für den Zugriff auf das `n`-te Element.

```
x=()                # Definition eines leeren Arrays
x[0]='a'           # Array-Elemente zuweisen
x[1]='b'
x[2]='c'
x=('a' 'b' 'c')    # Kurzschreibweise für die obigen vier Zeilen
echo ${x[1]}       # ein Array-Element lesen
echo ${x[@]}       # alle Array-Elemente lesen
```

Die für Programmierer wahrscheinlich wichtigste Neuerung in `bash` 4.0 ist die Unterstützung assoziativer Arrays. Dazu müssen Sie die Feldvariable explizit mit `declare -A` als assoziativ deklarieren! Andernfalls wird die Variable als normales Feld betrachtet.



Die im Index verwendeten Zeichenketten werden zu 0 ausgewertet, und Sie bekommen ein gewöhnliches Array, das aus nur einem einzigen Element besteht (Index 0).

```
declare -A y                # Definition eines leeren assoziativen Arrays
y[abc] = 123                # Element eines assoziativen Arrays zuweisen
y[efg] = xxx
y=( [abc]=123 [efg]=xxx )   # Kurzschreibweise für die obigen zwei Zeilen
echo ${y[abc]}             # ein Array-Element lesen
```

Eine weitere Neuerung in Version 4 besteht darin, dass Sie mit `mapfile` eine Textdatei zeilenweise in die Elemente eines gewöhnlichen Arrays einlesen können:

```
mapfile z < textdatei
```

### Parametersubstitution

Die `bash` stellt unter dem Begriff Parametersubstitution einige Kommandos zur Verfügung, mit denen in Variablen gespeicherte Zeichenketten bearbeitet werden können. Beachten Sie, dass der Variablenname *ohne* vorangestelltes `$`-Zeichen angegeben wird. Wenn hingegen das Vergleichsmuster aus einer Variablen gelesen werden soll, muss dort ein `$`-Zeichen verwendet werden.

`${var:-default}`

Wenn die Variable leer ist, liefert die Konstruktion die Defaulteinstellung als Ergebnis, andernfalls den Inhalt der Variablen. Die Variable wird nicht geändert.

`${var:=default}`

Wie oben, es wird aber gleichzeitig der Inhalt der Variablen geändert, wenn diese bisher leer war.

`${var:+neu}`

Wenn die Variable leer ist, bleibt sie leer. Wenn die Variable dagegen bereits belegt ist, wird der bisherige Inhalt durch eine neue Einstellung ersetzt. Die Konstruktion liefert den neuen Inhalt der Variablen.

`${var:?fehlermeldung}`

Wenn die Variable leer ist, werden der Variablenname und die Fehlermeldung ausgegeben, und das Shell-Programm wird anschließend beendet. Andernfalls liefert die Konstruktion den Inhalt der Variablen.

`${#var}`

liefert die Anzahl der in der Variablen gespeicherten Zeichen als Ergebnis (0, falls die Variable leer ist). Die Variable wird nicht geändert.

`${var#muster}`

vergleicht den Anfang der Variablen mit dem angegebenen Muster. Wenn das Muster erkannt wird, liefert die Konstruktion den Inhalt der Variablen abzüglich des kürzestmöglichen Textes, der dem Suchmuster entspricht. Wird das Muster dagegen nicht

gefunden, wird der ganze Inhalt der Variablen zurückgegeben. Im Suchmuster können die zur Bildung von Dateinamen bekannten Joker-Zeichen verwendet werden (\* ? [abc]). Die Variable wird in keinem Fall verändert:

```
user$ dat=/home/mk/buch/buch.tar.gz
user$ echo ${dat#*/}
home/mk/buch/buch.tar.gz
user$ echo ${dat#*.}
tar.gz
```

#### **`${var##muster}`**

Wie oben, allerdings wird jetzt die größtmögliche Zeichenkette, die dem Muster entspricht, eliminiert:

```
user$ dat=/home/mk/buch/buch.tar.gz
user$ echo ${dat##*/}
buch.tar.gz
user$ echo ${dat##*.}
gz
```

#### **`${var%muster}`**

Wie `${var#muster}`, allerdings erfolgt der Mustervergleich jetzt am Ende des Variableninhalts. Es wird die kürzestmögliche Zeichenkette vom Ende der Variablen eliminiert. Die Variable selbst bleibt unverändert:

```
user$ dat=/home/mk/buch/buch.tar.gz
user$ echo ${dat%/*}
/home/mk/buch
user$ echo ${dat%.*}
/home/mk/buch/buch.tar
```

#### **`${var%muster}`**

Wie oben, allerdings wird die größtmögliche Zeichenkette eliminiert:

```
user$ dat=/home/mk/buch/buch.tar.gz
user$ echo ${dat%/*}
-- keine Ausgabe --
user$ echo ${dat%.*}
/home/mk/buch/buch
```

#### **`${var/find/replace}`**

ersetzt das erste Auftreten des Musters `find` durch `replace`:

```
user$ x='abcdeab12ab'
user$ echo echo ${x/ab/xy}
xycdeab12ab
```

#### **`${var//find/replace}`**

ersetzt jedes Auftreten des Musters `find` durch `replace`:


```
user$ x='abcdeab12ab'
user$ echo echo ${x//ab/xy}
xycdexy12xy
```

```
${!var}
```

liefert den Inhalt der Variablen, deren Name in `var` als Zeichenkette enthalten ist:

```
user$ abc="123"
user$ efg=abc
user$ echo ${!efg}
123
```

### Variablen mit »read« einlesen

Mit dem `bash`-Kommando `read` können Sie Benutzereingaben verarbeiten. In der Regel geben Sie dazu zuerst mit `echo` einen kurzen Text aus, in dem Sie den Anwender darüber informieren, welche Eingabe Sie erwarten, beispielsweise `y/n`, einen numerischen Wert etc. Dabei ist die Option `-n` sinnvoll, damit die Eingabe unmittelbar hinter dem `echo`-Text und nicht in der nächsten Zeile erfolgt. Bei der Ausführung des anschließenden `read`-Kommandos wartet die `bash` so lange, bis der Anwender eine Zeile eingibt und diese mit  abschließt.

Im folgenden Beispielprogramm wird die `while`-Schleife so lange ausgeführt, bis die Zeichenkette in der Variablen `a` nicht mehr leer ist:

```
#!/bin/bash
# Beispiel readvar: numerischen Wert einlesen
a= # a löschen
while [ -z "$a" ]; do
  echo -n "Geben Sie eine Zahl ein: "
  read a
  a=${a##*[^0-9,' ','-']*} # Zeichenketten eliminieren, die
                          # irgendwelche Zeichen außer 0-9, dem
                          # Minuszeichen und dem Leerzeichen
                          # enthalten
  if [ -z "$a" ]; then
    echo "Ungültige Eingabe, bitte Eingabe wiederholen"
  fi
done
echo $a
```

Nach der Eingabe durch `read` wird der gesamte Inhalt der Variablen via Parametersubstitution gelöscht, wenn darin irgendein Zeichen außer einer Ziffer, einem Minuszeichen oder einem Leerzeichen vorkommt. Diese Kontrolle ist zwar nicht vollkommen (die Zeichenketten `"12-34-5"` und `"12 34"` sind demnach beide gültig), aber schon recht wirkungsvoll. Informationen zu `while` finden Sie im folgenden Abschnitt.

Ein Testlauf demonstriert die Funktion des kleinen Programms:

```
user$ readvar
Geben Sie eine Zahl ein: a
Ungültige Eingabe, bitte Eingabe wiederholen
Geben Sie eine Zahl ein: 12
12
```

## 10.11 Codestrukturierung in bash-Scripts

Dieser Abschnitt stellt einige Verfahren vor, mit denen Sie den Code in bash-Scripts strukturieren können: Verzweigungen mit `if` und `case`, Schleifen mit `for`, `while` und `until`, Funktionen etc.

### if-Verzweigungen

In der Shell-Datei `iftst` wird durch eine `if`-Abfrage getestet, ob zwei Parameter übergeben wurden. Wenn das nicht der Fall ist, wird eine Fehlermeldung ausgegeben. Das Programm wird durch `exit` mit einem Rückgabewert ungleich 0 (Indikator für Fehler) beendet. Andernfalls wird der Inhalt der beiden Parameter auf dem Bildschirm angezeigt.

```
#!/bin/bash
# Beispiel iftst
if test $# -ne 2; then
    echo "Dem Kommando müssen genau zwei Parameter übergeben werden!"
    exit 1
else
    echo "Parameter 1: $1, Parameter 2: $2"
fi
```

Ein kurzer Testlauf demonstriert das Verhalten des Programms:

```
user$ iftst a
Dem Kommando müssen genau zwei Parameter übergeben werden!
user$ iftst a b
Parameter 1: a, Parameter 2: b
```

Als Kriterium für die Verzweigung gilt der Rückgabewert des letzten Kommandos vor `then`. Die Bedingung ist erfüllt, wenn dieses Kommando den Rückgabewert 0 liefert. Wenn `then` noch in derselben Zeile angegeben wird (und nicht erst in der nächsten), dann muss das Kommando mit einem Semikolon abgeschlossen werden.

### Verkehrte Logik

Beachten Sie, dass in der `bash` die Wahrheitswerte für wahr (0) und falsch (ungleich 0) umgekehrt definiert sind als in den meisten anderen Programmiersprachen! Kommandos, die ordnungsgemäß beendet werden, liefern den Rückgabewert 0. Jeder Wert ungleich 0 deutet auf einen Fehler hin. Manche Kommandos liefern je nach Fehlertyp unterschiedliche Fehlerwerte.

Im obigen Beispiel wurde die Bedingung unter Zuhilfenahme des `bash`-Kommandos `test` gebildet. Der Operator `-ne` steht dabei für ungleich (*not equal*). `test` kommt immer dann zum Einsatz, wenn zwei Zeichenketten oder Zahlen miteinander verglichen werden sollen, wenn getestet werden soll, ob eine Datei existiert etc. Das Kommando wird im nächsten Abschnitt beschrieben.

Das obige Programm könnte auch anders formuliert werden: Statt des `test`-Kommandos kann eine Kurzschreibweise in eckigen Klammern verwendet werden. Dabei muss nach `[` und vor `]` jeweils ein Leerzeichen angegeben werden!

Außerdem kann das zweite `echo`-Kommando aus der `if`-Struktur herausgelöst werden, weil wegen der `exit`-Anweisungen alle Zeilen nach `fi` nur dann ausgeführt werden, wenn die Bedingung erfüllt ist.

```
#!/bin/bash
# Beispiel iftst, 2. Variante
if [ $# -ne 2 ]; then
    echo "Dem Kommando müssen genau zwei Parameter übergeben werden!"
    exit 1
fi
echo "Parameter 1: $1, Parameter 2: $2"
```

### Formulierung von Bedingungen mit »test«

In der `bash` ist es nicht möglich, Bedingungen – etwa den Vergleich einer Variablen mit einem Wert – direkt anzugeben. Zum einen basiert die ganze Konzeption der `bash` darauf, dass alle Aktionen über ein einheitliches Kommandokzept durchgeführt werden, zum anderen sind Sonderzeichen wie `>` und `<` bereits für andere Zwecke vergeben. Aus diesem Grund müssen Sie zur Formulierung von Bedingungen in Schleifen und Verzweigungen das `bash`-Kommando `test` verwenden. (`test` existiert übrigens auch als eigenständiges Kommando außerhalb der `bash`. Es wurde aber auch in die `bash` integriert, um eine höhere Verarbeitungsgeschwindigkeit zu erzielen.)

`test` liefert als Rückgabewert 0 (wahr), wenn die Bedingung erfüllt ist, oder 1 (falsch), wenn die Bedingung nicht erfüllt ist. Um den Schreibaufwand zu verringern, ist eine Kurzschreibweise in eckigen Klammern vorgesehen.

Zahlen,  
Zeichenketten,  
Dateien

test wird in drei Aufgabenbereichen eingesetzt: zum Vergleich zweier Zahlen, zum Vergleich von Zeichenketten und zum Test, ob eine Datei existiert und bestimmte Eigenschaften aufweist. Die folgenden Beispiele zeigen einige mögliche Anwendungsfälle:

```
test "$x"
```

überprüft, ob `x` belegt ist. Das Ergebnis ist falsch, wenn die Zeichenkette 0 Zeichen aufweist, andernfalls ist es wahr.

```
test $x -gt 5
```

testet, ob die Variable `x` einen Zahlenwert größer 5 enthält. Wenn `x` keine Zahl enthält, kommt es zu einer Fehlermeldung. Statt `-gt` (*greater than*) können auch die folgenden Vergleichsoperatoren verwendet werden: `-eq` (*equal*), `-ne` (*not equal*), `-lt` (*less than*), `-le` (*less equal*) und `-ge` (*greater equal*).

```
test -f $x
```

testet, ob eine Datei mit dem in `x` angegebenen Namen existiert.

Wenn `test` interaktiv in der Shell ausgeführt werden soll, muss nach dem `test`-Kommando die Variable `?` (Rückgabewert des letzten Kommandos) mit `echo` gelesen werden:

```
user$ a=20
user$ test $a -eq 20; echo $?
0
user$ test $a -gt 20; echo $?
1
```

### case-Verzweigungen

`case`-Konstruktionen werden mit dem Schlüsselwort `case` eingeleitet, dem der zu analysierende Parameter zumeist in einer Variablen folgt. Nach dem Schlüsselwort `in` können dann mehrere mögliche Musterzeichenketten angegeben werden, mit denen der Parameter verglichen wird. Dabei sind die gleichen Jokerzeichen wie bei Dateinamen erlaubt. Das Muster wird mit einer runden Klammer `)` abgeschlossen, also etwa mit `--*)` zur Erkennung von Zeichenketten, die mit zwei Minuszeichen beginnen. Mehrere Muster können durch `|` voneinander getrennt werden. In diesem Fall werden beide Muster getestet. Beispielsweise dient `*.c|*.h` zur Erkennung von `*.c`- und `*.h`-Dateien im selben Zweig.

Die der Klammer folgenden Kommandos müssen durch zwei Semikola abgeschlossen werden. Wenn ein `else`-Zweig benötigt wird, dann muss als letztes Muster `*` angegeben werden – alle Zeichenketten entsprechen diesem Muster. Bei der Abarbeitung einer `case`-Konstruktion wird nur der erste Zweig berücksichtigt, bei dem der Parameter dem angegebenen Muster entspricht.

Das folgende Beispiel `casetst` zeigt die Anwendung von `case` zur Klassifizierung der übergebenen Parameter in Dateinamen und Optionen. Die Schleife für die Variable `i` wird für alle der Shell-Datei übergebenen Parameter ausgeführt. Innerhalb dieser Schleife wird jeder einzelne Parameter mit `case` analysiert. Wenn der Parameter mit einem Bindestrich beginnt, wird der Parameter an das Ende der Variablen `opt` angefügt, andernfalls an das Ende von `dat`. Beispiel

```
#!/bin/bash
# Beispiel casetst
for i do # Schleife für alle übergebenen Parameter
  case "$i" in
    -* ) opt="$opt $i";;
    * ) dat="$dat $i";;
  esac
done # Ende der Schleife
echo "Optionen: $opt"
echo "Dateien: $dat"
```

Ein Beispiellauf der Shell-Datei beweist die Wirkungsweise dieser einfachen Fallunterscheidung. Die in ihrer Reihenfolge wahllos übergebenen Parameter werden in Optionen und Dateinamen untergliedert:

```
user$ casetst -x -y dat1 dat2 -z dat3
Optionen: -x -y -z
Dateien: dat1 dat2 dat3
```

Nach demselben Schema können `case`-Verzweigungen auch zur Klassifizierung von bestimmten Dateikennungen verwendet werden, indem im Suchmuster `*.abc` angegeben wird. Wenn Sie sich eingehender mit `case`-Analysen beschäftigen möchten, sollten Sie sich die Shell-Datei `/usr/bin/gnroff` ansehen. Die Datei bereitet die in der Syntax von `roff` übergebenen Parameter so auf, dass das verwandte Kommando `groff` damit zurechtkommt.

## for-Schleifen

Die `bash` kennt drei Kommandos zur Bildung von Schleifen: `for` führt eine Schleife für alle Elemente einer angegebenen Liste aus. `while` führt eine Schleife so lange aus, bis die angegebene Bedingung nicht mehr erfüllt ist, `until` führt sie dagegen so lange aus, bis die Bedingung zum ersten Mal erfüllt ist. Alle drei Schleifen können mit `break` vorzeitig verlassen werden. `continue` überspringt den restlichen Schleifenkörper und setzt die Schleife mit dem nächsten Schleifendurchlauf fort.

Im ersten Beispiel werden der Variablen `i` der Reihe nach die Zeichenketten `a`, `b` und `c` zugewiesen. Im Schleifenkörper wird zwischen `do` und `done` der Inhalt der Variablen ausgegeben. Beachten Sie, dass sowohl am Ende der Liste als auch am Ende des

echo-Kommandos ein Strichpunkt erforderlich ist. Auf diese Strichpunkte kann nur verzichtet werden, wenn die Eingabe auf mehrere Zeilen verteilt wird (was in Script-Dateien häufig der Fall ist).

```
user$ for i in a b c; do echo $i; done
a
b
c
```

Die äquivalente mehrzeilige Formulierung des obigen Kommandos in einer Script-Datei würde so aussehen:

```
#!/bin/bash
for i in a b c; do
    echo $i
done
```

#### Schleife über Dateien

Die Liste für `for` kann auch mit Jokerzeichen für Dateinamen oder mit `{..}`-Konstruktionen zur Bildung von Zeichenketten erstellt werden. Im folgenden Beispiel werden alle `*.tex`-Dateien in `*.tex~`-Dateien kopiert. Das Zeichen `~` am Ende eines Dateinamens bezeichnet unter Unix/Linux üblicherweise eine Backup-Datei. Beim `cp`-Kommando ist `$file` jeweils in Anführungszeichen gestellt, damit auch Dateinamen mit Leerzeichen korrekt behandelt werden.

```
user$ for file in *.tex; do cp "$file" "$file~"; done
```

#### Schleife über alle Zeilen einer Textdatei

Oft benötigen Sie Schleifen, um eine Textdatei Zeile für Zeile abzuarbeiten. Kein Problem: Übergeben Sie an das Schlüsselwort `in` einfach das Ergebnis von `cat datei!` Das folgende Miniprogramm erstellt für alle Datenbanken, die in der Datei `dbs.txt` zeilenweise genannt sind, ein komprimiertes Backup in der Datei `db.sql.gz`:

```
#!/bin/bash
# Schleife über
for db in $(cat dbs.txt); do
    mysqldump $db | gzip -c > $db.sql.gz
done
```

#### Schleife über alle Parameter

Wenn `for`-Schleifen ohne `in ...` gebildet werden, dann werden der Schleifenvariablen der Reihe nach alle beim Aufruf übergebenen Parameter übergeben (das entspricht also `in $*`). Ein Beispiel für so eine Schleife finden Sie bei der Beschreibung von `case`.

Wenn an das `case`-Beispiel Dateinamen mit Leerzeichen übergeben werden, kommt es allerdings zu Problemen: Die `bash` interpretiert das Leerzeichen als Trennzeichen und verarbeitet die Teile des Dateinamens getrennt. Abhilfe schafft die folgende Konstruktion:



```
#!/bin/bash
# Schleife über alle Parameter, kommt mit Leerzeichen in den Dateinamen zurecht
for i in "$@"; do
    ls -l "$i"
done
```

Schleifen über einen numerischen Zahlenbereich formulieren Sie am einfachsten in der folgenden Form:

**Numerische  
for-Schleifen**

```
for i in {1..12}; do
    echo "$i"
done
# Ausgabe 1, 2, 3, ... ,12
```

Wenn Sie führende Nullen benötigen, geben Sie diese im ersten Parameter in {n1..n2} an:

```
for i in {01..12}; do
    echo "$i"
done
# Ausgabe 01, 02, 03, ... ,12
```

Für Schleifen mit einer vorgegebenen Schrittweite verwenden Sie die Schreibweise {n1..n2..step}, also zum Beispiel:

```
for i in {1..12..3}; do
    echo "$i"
done
# Ausgabe 1, 4, 7, 10
```

Zu guter Letzt können Sie auch auf die klassische C-Syntax zurückgreifen:

```
for ((i=1;i<=12;i++)); do
    echo "$i"
done
# Ausgabe 1, 2, 3, ... ,12
```

Beachten Sie, dass die bash keine Fließkommaarithmetik vorsieht. Alle hier vorgestellten Konstrukte funktionieren also nur für ganze Zahlen.

## while-Schleifen

Im folgenden Beispiel wird der Variablen *i* der Wert 1 zugewiesen. Anschließend wird die Variable im Schleifenkörper zwischen *do* und *done* so oft um 1 erhöht, bis der Wert 5 überschritten wird. Beachten Sie, dass Bedingungen wie bei *if*-Verzweigungen mit dem Kommando *test* bzw. mit dessen Kurzschreibweise in eckigen Klammern angegeben werden müssen.

```
user$ i=1; while [ $i -le 5 ]; do echo $i; i=$((i+1)); done
1
2
3
4
5
```

Die folgende Schleife verarbeitet alle Dateinamen, die sich aus dem Kommando `ls *.jpg` ergeben:

```
ls *.jpg | while read file
do
    echo "$file"
done
```

### until-Schleifen

Der einzige Unterschied zwischen `until`-Schleifen und `while`-Schleifen besteht darin, dass die Bedingung logisch negiert formuliert wird. Das folgende Kommando ist daher zur obigen `while`-Schleife äquivalent. Dabei wird `-gt` zur Formulierung der Bedingung `i>5` (*greater than*) verwendet.

```
user$ i=1; until [ $i -gt 5 ]; do echo $i; i=$((i+1)); done
1
2
3
4
5
```

### Funktionen

Das Schlüsselwort `function` definiert eine Subfunktion, die wie ein neues Kommando aufgerufen werden kann. Der Code der Funktion muss zwischen geschwungene Klammern gesetzt werden. Innerhalb der Funktion können mit `local` lokale Variablen definiert werden. Funktionen können rekursiv aufgerufen werden. Funktionen müssen *vor* ihrem ersten Aufruf deklariert werden!

An Funktionen können Parameter übergeben werden. Anders als bei vielen Programmiersprachen werden die Parameter nicht in Klammern gestellt. Innerhalb der Funktion können die Parameter den Variablen `$1`, `$2` entnommen werden, d. h., eine Funktion verarbeitet Parameter auf die gleiche Art und Weise wie ein `bash`-Script. Das folgende Mini-Script gibt *Hello World, Linux!* aus:

```
#!/bin/bash
function myfunc {
    echo "Hello World, $1!"
}
myfunc "Linux"
```

Das Schlüsselwort `function` ist optional. Wenn auf `function` verzichtet wird, müssen dem Funktionsnamen allerdings zwei runde Klammern folgen. Somit ist das folgende Programm gleichwertig zum obigen Beispiel:

```
#!/bin/bash
myfunc() {
    echo "Hello World, $!"
}
myfunc "Linux"
```

## exit

`exit` beendet das laufende Script. Damit gibt das Script den Rückgabewert des zuletzt ausgeführten Kommandos zurück. Wenn Sie das nicht wünschen, können Sie an `exit` explizit einen Wert übergeben. Der Wert 0 signalisiert ein fehlerfreies Programmende. 1 weist auf einen allgemeinen Fehler hin, 2 auf einen Fehler in den übergebenen Parametern. Alle anderen Fehlercodes sind anwendungsspezifisch.

## Heredoc-Syntax

Die Heredoc-Syntax ermöglicht es, mehrzeilige Konstrukte zu formulieren, die mit einer frei gewählten Zeichenkette enden. Besonders häufig wird diese Syntax verwendet, um mehrzeilige Dateien zu erzeugen oder mehrzeilige Variablen zu definieren:

```
# schreibt einen mehrzeiligen Text in die Datei myfile
cat <<EOF > myfile
Zeile 1
Zeile 2
Zeile 3
EOF

# speichert einen mehrzeiligen Text in der Variablen $myvar
read -r -d '' myvar <<EOF
Zeile 1
Zeile 2
Zeile 3
EOF
echo "$myvar"
```

Anstelle von EOF können Sie eine beliebige andere Zeichenkette verwenden. Es ist üblich, diese Zeichenkette in Großbuchstaben anzugeben. Beim `read`-Kommando bewirkt die Option `-r`, dass die Zeichenkette unverändert übernommen wird (*raw*). `-d ''` (*delimiter*) bewirkt, dass das Zeilenende nicht als Ende der Eingabe zählt. Bei der Ausgabe mehrzeiliger Zeichenketten durch `echo` ist es wichtig, die Variable in Anführungszeichen zu setzen – andernfalls gehen die Zeilenumbrüche verloren.

## 10.12 Referenz wichtiger bash-Sonderzeichen

Sowohl bei der Eingabe von Kommandos als auch bei der Shell-Programmierung können Sie eine unüberschaubare Fülle von Sonderzeichen für diverse Aktionen verwenden. [Tabelle 10.6](#) fasst alle Sonderzeichen zusammen, die in diesem Kapitel behandelt wurden.

| Zeichen      | Bedeutung   |
|--------------|---|
| ;            | trennt mehrere Kommandos.                                   |
| :            | Shell-Kommando, das nichts tut                              |
| .            | Shell-Code hier einfügen (. datei entspricht source datei)  |
| #            | leitet einen Kommentar ein.                                 |
| #!/bin/bash  | identifiziert die gewünschte Shell für das Shell-Programm.  |
| &            | führt das Kommando im Hintergrund aus (kom &).              |
| &&           | bedingte Kommandoausführung (kom1 && kom2)                  |
| &>           | Umleitung von Standardausgabe und -fehler (entspricht >&)   |
|              | bildet Pipes (kom1   kom2).                                 |
|              | bedingte Kommandoausführung (kom1    kom2)                  |
| *            | Jokerzeichen für Dateinamen (beliebig viele Zeichen)        |
| ?            | Jokerzeichen für Dateinamen (ein beliebiges Zeichen)        |
| [abc]        | Jokerzeichen für Dateinamen (ein Zeichen aus abc)           |
| [ ausdruck ] | Kurzschreibweise für test ausdruck                          |
| (...)        | Kommandos in derselben Shell ausführen ((kom1; kom2))       |
| {...}        | Kommandos gruppieren  |
| { , , }      | Zeichenketten zusammensetzen (a{1,2,3} → a1 a2 a3)          |
| {a..b}       | Zeichenketten zusammensetzen (b{4..6} → b4 b5 b6)           |
| ~            | Abkürzung für das Heimatverzeichnis                         |
| >            | Ausgabeumleitung in eine Datei (kom > dat)                  |
| >>           | Ausgabeumleitung; an vorhandene Datei anhängen              |
| >&           | Umleitung von Standardausgabe und -fehler (entspricht &>)   |
| &>>          | Standardausgabe und -fehler an Datei anhängen (ab bash 4.0) |
| 2>           | Umleitung der Standardfehlerausgabe                         |
| <            | Eingabeumleitung aus einer Datei (kom < dat)                |
| << ende      | Eingabeumleitung aus der aktiven Datei bis zu ende          |

Tabelle 10.6 bash-Sonderzeichen

| Zeichen      | Bedeutung   |
|--------------|---|
| \$           | Kennzeichnung von Variablen (echo \$var)                      |
| #!           | PID des zuletzt gestarteten Hintergrundprozesses              |
| \$\$         | PID der aktuellen Shell                                       |
| \$0          | Dateiname des gerade ausgeführten Shell-Scripts               |
| \$1 bis \$9  | die ersten neun dem Kommando übergebenen Parameter            |
| \$#          | Anzahl der dem Shell-Programm übergebenen Parameter           |
| *\$ oder @\$ | Gesamtheit aller übergebenen Parameter                        |
| \$?          | Rückgabewert des letzten Kommandos (0 = OK oder Fehlernummer) |
| \$(...)      | Kommandosubstitution (echo \$(ls))                            |
| \${...}      | diverse Spezialfunktionen zur Bearbeitung von Zeichenketten   |
| `\${...}`    | arithmetische Auswertung (echo `\${2+3}`)                     |
| "..."        | Auswertung der meisten Sonderzeichen verhindern               |
| '...'        | Auswertung aller Sonderzeichen verhindern                     |
| `...`        | Kommandosubstitution (echo `ls`)                              |

Tabelle 10.6 bash-Sonderzeichen (Forts.)



# Kapitel 11

## Dateien und Verzeichnisse

Dieses Kapitel beschreibt den Umgang mit Dateien. Im Detail behandle ich die folgenden Themen:

- ▶ Dateien, Verzeichnisse und Links
- ▶ Dateien kopieren, verschieben, löschen und suchen
- ▶ Zugriffsrechte von Dateien (inklusive ACLs)
- ▶ Linux-Verzeichnisstruktur
- ▶ Device-Dateien

Gewissermaßen die Fortsetzung dieses Kapitels ist [Kapitel 22](#), »Administration des Dateisystems«. Dort geht es dann um Fragen, die überwiegend für Systemadministratoren interessant sind: Welche Dateisysteme gibt es? Wie erfolgt der Zugriff auf externe Festplatten und USB-Sticks? Wie werden Dateisysteme in den Verzeichnisbaum integriert (/etc/fstab, mount-Optionen)? Wie kann die Partitionierung einer Festplatte oder SSD verändert werden? Wie kann ein Software-RAID-System eingesetzt werden? Was ist LVM? Wie kann ein ganzes Dateisystem verschlüsselt werden?

Administration  
des Dateisystems

Das Thema Backups behandle ich in [Kapitel 36](#). Dort stelle ich Ihnen nicht nur Benutzeroberflächen zur Durchführung von Backups vor, sondern auch eine breite Palette von Kommandos, mit denen Sie Dateien komprimieren, in Archive bündeln, synchronisieren, verschlüsseln etc.

Backup

### 11.1 Umgang mit Dateien und Verzeichnissen

Ganz kurz die wichtigsten Fakten zu Dateinamen:

- ▶ Unter Linux sind Dateinamen mit einer Länge von bis zu 255 Zeichen zulässig.
- ▶ Es wird zwischen Groß- und Kleinschreibung unterschieden!
- ▶ Leerzeichen in Dateinamen sind erlaubt, führen aber bei der Verarbeitung durch Scripts oft zu Problemen. Dateinamen mit Leer- oder Sonderzeichen müssen in Hochkommata gestellt werden, beispielsweise "a b".

- ▶ Internationale Zeichen im Dateinamen sind zulässig, können aber zu Problemen führen, wenn unterschiedliche Zeichensätze zum Einsatz kommen, z. B. in einem Netzwerk. Alle gängigen Linux-Distributionen verwenden Unicode UTF-8 als Standardzeichensatz. Aus der Sicht des Linux-Kernels ist der Dateiname einfach eine Bytefolge, in der lediglich das Zeichen / und der Code 0 nicht vorkommen dürfen. Wie diese Bytefolge interpretiert wird, hängt vom gerade gültigen Zeichensatz ab.
- ▶ Dateinamen dürfen beliebig viele Punkte enthalten. `README.bootutils.gz` ist ein ganz normaler Dateiname, der andeutet, dass es sich um eine komprimierte README-Datei zum Thema Boot-Utilities handelt.
- ▶ Dateien, die mit einem Punkt beginnen, gelten als versteckte Dateien. Versteckte Dateien werden durch `ls` bzw. durch diverse Dateimanager normalerweise nicht angezeigt.

Die Größe von Dateien ist bei aktuellen Linux-Distributionen nahezu unbeschränkt und liegt je nach Dateisystem im Terabyte-Bereich.

## Verzeichnisse

**Verzeichnisbaum** Der Verzeichnisbaum von Linux beginnt im Wurzelverzeichnis /. Laufwerksangaben wie `C:` sind unter Linux nicht möglich. Innerhalb dieses Buchs gelten alle weiteren Verzeichnisse als *untergeordnet*: Das Wurzelverzeichnis steht also – bildlich gesehen – ganz oben. In manchen Büchern ist die Nomenklatur gerade umgekehrt, was zwar dem Baumbild (Wurzel unten, Verästelung oben) besser entspricht, aber nicht mit dem üblichen Sprachgebrauch übereinstimmt.

Linux-Einsteiger tun sich oft schwer, Dateien im weit verästelten Verzeichnissystem zu finden. Abhilfe: Lesen Sie [Abschnitt 11.4](#), »Dateien suchen (find, grep, locate)«, sowie [Abschnitt 11.8](#), »Die Linux-Verzeichnisstruktur«! Dort lernen Sie einerseits diverse Suchwerkzeuge kennen und erfahren andererseits, wie der Verzeichnisbaum von Linux strukturiert ist.

**Heimatverzeichnis** In Textkonsolen bzw. Terminalfenstern ist anfänglich automatisch das sogenannte Heimat- oder Home-Verzeichnis aktiv. Alle darin enthaltenen Dateien und Unterverzeichnisse gehören Ihnen. Andere Benutzer mit der Ausnahme von `root` dürfen diese Dateien weder verändern noch löschen, und je nach Einstellung der Zugriffsrechte nicht einmal lesen.

Das Heimatverzeichnis befindet sich im Linux-Verzeichnisbaum üblicherweise an der Stelle `/home/<loginname>/`. Nur bei `root` heißt das Heimatverzeichnis `/root`. Da es umständlich wäre, `/home/<loginname>` immer auszuschreiben, kann das eigene Heimatverzeichnis mit der Tilde `~` abgekürzt werden. Für den Zugriff auf die Heimatverzeichnisse anderer Benutzer ist außerdem die Schreibweise `~<loginname>` möglich.



In jedem Verzeichnis existieren zwei besondere Unterverzeichnisse, die zur formalen Verwaltung der Verzeichnishierarchie dienen: Das Verzeichnis mit dem Namen `.` ist ein Verweis auf das aktuelle Verzeichnis, das Verzeichnis `..` ein Verweis auf das übergeordnete Verzeichnis.

Die Verzeichnisse `.` und `..`

Die beiden folgenden Kopierkommandos zeigen, wie Sie diese Verzeichnisse nutzen. Das erste Kommando kopiert die Datei `/etc/fstab` in das gerade aktuelle Verzeichnis. Wenn das aktuelle Verzeichnis `/home/name` lautet, dann hat die neue Datei den Namen `/home/name/fstab`.

```
user$ cp /etc/fstab .
```

Das zweite Beispiel aktiviert zuerst mit `cd` das Verzeichnis `linuxbuch` im Heimatverzeichnis. Das Kopierkommando `cp` erstellt dann eine Sicherheitskopie der Datei `fileuse.tex`, die den Text dieses Kapitels enthält. Die Sicherheitskopie hat den Namen `~/fileuse.tex.bak`.

```
user$ cd ~/linuxbuch
user$ cp fileuse.tex ../fileuse.tex.bak
```

Wenn das Heimatverzeichnis `/home/name` lautet, dann haben Sie also gerade eine Kopie von `/home/name/linuxbuch/fileuse.tex` erstellt. Die Sicherheitskopie hat den vollständigen Namen `/home/name/fileuse.tex.bak`. Weitere `cp`-Beispiele folgen im nächsten Abschnitt.

| Zeichen         | Bedeutung  |
|-----------------|--|
| <code>~</code>  | Heimatverzeichnis                                    |
| <code>.</code>  | aktuelles Verzeichnis                                |
| <code>..</code> | übergeordnetes Verzeichnis zum aktuellen Verzeichnis |

Tabelle 11.1 Sonderzeichen für Verzeichnisse

## Elementare Kommandos zur Bearbeitung von Dateien und Verzeichnissen

Obwohl unter KDE und Gnome moderne Dateimanager zur Verfügung stehen, verwenden erfahrene Linux-Anwender gerne textorientierte Kommandos. [Tabelle 11.2](#) fasst die allerwichtigsten Kommandos zusammen.

Mit dem Kommando `cd` wechseln Sie in ein anderes Verzeichnis. `cd -` wechselt zurück in das zuletzt aktive Verzeichnis, `cd ..` wechselt in das übergeordnete Verzeichnis, `cd` ohne weitere Parameter wechselt in das Heimatverzeichnis.

Verzeichnis mit `cd` wechseln


```
user$ cd /etc/samba
```

| Kommando | Funktion   |
|----------|--|
| cd       | wechselt das aktuelle Verzeichnis.               |
| cp       | kopiert Dateien.                                 |
| j        | wechselt in ein zuletzt verwendetes Verzeichnis. |
| less     | zeigt Textdateien seitenweise an.                |
| ls       | zeigt alle Dateien eines Verzeichnisses an.      |
| mkdir    | erzeugt ein neues Verzeichnis.                   |
| mv       | verschiebt Dateien bzw. ändert ihren Namen.      |
| rm       | löscht Dateien.                                  |
| rmdir    | löscht Verzeichnisse.                            |

**Tabelle 11.2** Elementare Kommandos zum Umgang mit Dateien und Verzeichnissen




#### Verzeichnis mit j wechseln

Bei einigen Distributionen können Sie zum Verzeichniswechsel auch das Kommando `j` aus dem Paket `autojump` verwenden. In der Regel müssen Sie dieses Paket zuerst installieren. Autojump merkt sich, in welchen Verzeichnissen Sie am häufigsten arbeiten. `jumpstats` liefert Ihnen bei Bedarf die dazugehörigen statistischen Daten.

Wenn Sie in ein schon früher genutztes Verzeichnis wechseln möchten, führen Sie `j abc` aus, wobei `abc` die ersten Buchstaben des Verzeichnisnamens sind. Die Eingabe des oft langen Pfads zum Verzeichnis ist nicht erforderlich! Sofern es mehrere passende Verzeichnisse gibt, wechselt `j` in das Verzeichnis, das Sie zuletzt am häufigsten verwendet haben. Zudem können Sie mit  zwischen den zur Auswahl stehenden Verzeichnissen wählen.

Die Autojump-Website bezeichnet `j` als ein mitlernendes `cd`-Kommando – eine durchaus zutreffende Beschreibung. Es dauert nur kurze Zeit, sich an `j` zu gewöhnen; danach möchte man das Kommando nicht mehr missen.

<https://github.com/joelthelion/autojump/wiki>

`j` kann `cd` allerdings nicht ersetzen, sondern nur ergänzen: Die Vervollständigung des Pfadnamens durch  funktioniert nur für Verzeichnisse, die sich bereits in der Autojump-Datenbank befinden. Wenn Sie also `cd /e`  eingeben, wird `/e` in der Regel zu `/etc/` ergänzt. Die analoge Vervollständigung von `j /e`  funktioniert hingegen erst, wenn sich `/etc` bereits in der Autojump-Datenbank befindet. Mit anderen Worten: Um in ein Verzeichnis zu wechseln, das Autojump nicht kennt, ist es weiterhin besser, `cd` zu verwenden. (Sie gelangen auch mit `j` in das gewünschte Verzeichnis, aber dazu müssen Sie den ganzen Verzeichnisnamen eintippen.) Schade – noch eleganter wäre es, wenn `j` als vollständiger `cd`-Ersatz verwendet werden könnte!

ls liefert eine Liste aller Dateien im aktuellen Verzeichnis. Wenn Sie auch verborgene Dateien sehen möchten, geben Sie zusätzlich die Option `-a` an. Wenn Sie sich nicht nur für den Dateinamen, sondern auch für die Dateigröße, den Besitzer und andere Details interessieren, hilft Ihnen die Option `-l` weiter. Standardmäßig ist die Ausgabe von `ls` alphabetisch geordnet. Um die Dateiliste nach dem Zeitpunkt der letzten Änderung, der Dateigröße bzw. der Dateikennung zu sortieren, verwenden Sie die Optionen `-t`, `-S` bzw. `-X`. `-r` dreht die Sortierordnung um. Das folgende Kommando zeigt alle `*.tex`-Dateien im Verzeichnis `linuxbuch`, geordnet nach ihrer Größe (die größte Datei zuerst):

Dateien auflisten

```
user$ ls -l -S linuxbuch/*.tex
...
-rw-r--r-- 1 kofler kofler 30113 2017-05-11 09:09 linuxbuch/intro.tex
-rw-r--r-- 1 kofler kofler 63173 2017-01-29 08:05 linuxbuch/kde.tex
-rw-r--r-- 1 kofler kofler 76498 2017-06-08 15:43 linuxbuch/kernel.tex
...
```

Kurz einige Anmerkungen zur Interpretation des `ls`-Ergebnisses: Die zehn Zeichen am Beginn der Zeile geben den Dateityp und die Zugriffsbits an. Als Dateityp kommen infrage: der Bindestrich `-` für eine normale Datei, `d` für ein Verzeichnis (*Directory*), `b` oder `c` für eine Device-Datei (*Block* oder *Char*) oder `l` für einen symbolischen Link. Die nächsten drei Zeichen (`rw`) geben an, ob der Besitzer die Datei lesen, schreiben und ausführen darf. Analoge Informationen folgen für die Mitglieder der Gruppe sowie für alle anderen Systembenutzer.

Die Zahl im Anschluss an die zehn Typ- und Zugriffszeichen gibt an, wie viele Hardlinks auf die Datei verweisen. (Was Links sind, wird in [Abschnitt 11.2](#) beschrieben. Details zur Zugriffsverwaltung von Linux-Dateien folgen in [Abschnitt 11.5](#), »Zugriffsrechte, Benutzer und Gruppenzugehörigkeit«.) Die weiteren Spalten geben den Besitzer und die Gruppe der Datei an (hier jeweils `kofler`), die Größe der Datei, das Datum und die Uhrzeit der letzten Änderung und zuletzt den Dateinamen.

Bei den meisten Distributionen ist `ls` so konfiguriert, dass es Dateien und Verzeichnisse je nach Typ in unterschiedlichen Farben darstellt. Sollte das bei Ihrer Distribution nicht der Fall sein, erzielen Sie diesen Effekt mit der Option `--color`. Um umgekehrt die farbige Darstellung zu verhindern, führen Sie `l --color=none` aus.

ls-Farben

ls berücksichtigt nur die Dateien des aktuellen Verzeichnisses. Um auch die Dateien aus Unterverzeichnissen einzuschließen, verwenden Sie die Option `-R`. Diese Option steht übrigens auch bei vielen anderen Kommandos zur Verfügung. Das folgende Kommando listet sämtliche Dateien in allen Unterverzeichnissen auf. Diese Liste wird normalerweise recht lang. Daher leitet `| less` das Resultat von `ls` an `less` weiter, sodass Sie durch das Ergebnis blättern können.

Rekursives ls

```
user$ ls -lR | less
```

**Dateien kopieren** `cp name1 name2` kopiert die Datei `name1`. Die Kopie hat den Namen `name2`. Um mehrere Dateien zu kopieren, rufen Sie das Kommando in der Form `cp name1 name2 ... zielverzeichnis` auf. Die folgenden Kommandos machen `linuxbuch` zum aktiven Verzeichnis, erzeugen `bak` als Unterverzeichnis und kopieren alle `*.tex`-Dateien dorthin:

```
user$ cd linuxbuch
user$ mkdir bak
user$ cp *.tex bak/
```

**Verzeichnisse kopieren** Um ganze Verzeichnisse samt ihrem Inhalt zu kopieren, verwenden Sie `cp -r`. Die Option `-r` bewirkt, dass der gesamte Inhalt des Quellverzeichnisses rekursiv verarbeitet wird (inklusive versteckter Dateien). Wenn Sie möchten, dass beim Kopieren die Zugriffsrechte und -zeiten erhalten bleiben, verwenden Sie statt `-r` die Option `-a`.

Etwas diffizil ist die Frage, ob das Quellverzeichnis selbst oder nur sein Inhalt kopiert wird. Wenn es das Zielverzeichnis bereits gibt, wird darin das neue Unterverzeichnis `quellverzeichnis` erzeugt und der gesamte Inhalt des Quellverzeichnisses dorthin kopiert. Wenn es das Zielverzeichnis hingegen noch nicht gibt, wird es erzeugt. In diesem Fall wird nur der *Inhalt* des Quellverzeichnisses in das neu erzeugte Zielverzeichnis kopiert, nicht aber das Quellverzeichnis selbst. Die folgenden Beispiele verdeutlichen den Unterschied:

```
user$ mkdir test
user$ touch test/a
user$ mkdir test/b
user$ touch test/b/c
user$ mkdir ziel1
user$ cp -r test ziel1      (Das Verzeichnis ziel1/ existiert schon.)
user$ ls ziel1
test
user$ cp -r test ziel2     (Das Verzeichnis ziel2/ existiert noch nicht.)
user$ ls ziel2
a b
```

**Dateien und Verzeichnisse löschen** `rm datei` löscht die angegebene Datei unwiderruflich. `rm` kann normalerweise nur für Dateien, nicht aber für Verzeichnisse verwendet werden. Für Verzeichnisse ist das Kommando `rmdir verzeichnis` vorgesehen, das allerdings nur funktioniert, wenn das Verzeichnis leer ist. In der Praxis werden Sie zum Löschen von Verzeichnissen oft `rm` mit der Option `-rf` verwenden. Das bedeutet, dass rekursiv alle Unterverzeichnisse und Dateien ohne Rückfrage gelöscht werden. Es sollte Ihnen klar sein, dass `rm -rf` ein sehr gefährliches Kommando ist!

```
user$ rm -rf linuxbuch-bak/ (Backup-Verzeichnis löschen)
```

## Platzbedarf von Dateien und Verzeichnissen ermitteln

`ls -l` verrät Ihnen zwar, wie groß eine Datei ist. Oft wollen Sie aber wissen, wie viel Platz die Dateien im gesamten Verzeichnis beanspruchen, wie viel Platz auf der Festplatte noch frei ist etc. Dabei helfen die beiden Kommandos `df` und `du`.

`df` zeigt für alle in das Dateisystem eingebundenen Partitionen bzw. Datenträger an, wie viel Speicher insgesamt zur Verfügung steht und wie viel davon noch frei ist.

Freie Festplattenkapazität ermitteln

Im folgenden Beispiel liefert `df` Ergebnisse für vier Partitionen bzw. Datenträger. Die Option `-h` bewirkt, dass sämtliche Kapazitätsangaben in lesbaren Zahlen in KiB, MiB bzw. GiB angegeben werden. `df` zeigt normalerweise auch diverse temporäre Dateisysteme an, die nur zur internen Verwaltung dienen und nicht zum Speichern regulärer Dateien gedacht sind. Die Aufnahme solcher Dateisysteme in das `df`-Ergebnis vermeiden Sie mit der Option `-x tmpfs`.

```
user$ df -h -x tmpfs
Dateisystem      Größe Benut  Verf Ben% Eingehängt auf
/dev/sda3         14G  4,7G  8,5G  36% /
/dev/sda2        942M   47M  849M   6% /boot
/dev/sda6         28G  7,7G  19G  30% /home
```

Mit `df` können Sie auch feststellen, in welcher Partition sich ein Verzeichnis physikalisch befindet. Im folgenden Beispiel befindet sich das Verzeichnis `/home/kofler` in der Partition `/dev/sda6`, die an der Stelle `/home` in den Verzeichnisbaum eingebunden ist:

```
user$ df -h /home/kofler/
/dev/sda6         28G  7,7G  19G  30% /home
```

`du` ermittelt den Platzbedarf für das aktuelle Verzeichnis sowie für alle darin enthaltenen Unterverzeichnisse. Die Option `-h` bewirkt wiederum, dass das Ergebnis in lesbarer Form (nicht in KiB-Blöcken) angezeigt wird. Es gibt keine Optionen, um die `du`-Ergebnisse zu sortieren. Unter Gnome können Sie die Größe von Verzeichnissen mit dem Programm `baobab` grafisch veranschaulichen; bei KDE ist eine vergleichbare Ansicht in den Dateimanager Konqueror integriert.

Verzeichnisgröße ermitteln

```
user$ du -h fotos/2017
74M   fotos/2017/2017-04-ostern
162M  fotos/2017/2017-08-korsika
66M   fotos/2017/2017-11-diverse
...
2,0G  fotos/2017
```

## Jokerzeichen

Im täglichen Umgang mit Dateien werden Sie häufig ganze Gruppen von Dateien bearbeiten – etwa alle Dateien mit der Endung `.pdf`. Um das zu ermöglichen, sind bei der Eingabe von Linux-Kommandos sogenannte Jokerzeichen vorgesehen.

| Zeichen                                      | Bedeutung                                    |
|--|--|
| <code>?</code>                               | genau ein beliebiges Zeichen                 |
| <code>*</code>                               | beliebig viele (auch null) beliebige Zeichen |
| <code>[abc]</code>                           | genau eines der angegebenen Zeichen          |
| <code>[a-f]</code>                           | ein Zeichen aus dem angegebenen Bereich      |
| <code>[!abc]</code> oder <code>[^abc]</code> | keines der angegebenen Zeichen               |

**Tabelle 11.3** Jokerzeichen für Dateinamen

**\* und ?** `?` dient zur Spezifikation *eines* beliebigen Zeichens, und `*` dient zur Spezifikation beliebig vieler (auch null) Zeichen. Wer sich noch mit DOS auskennt, wird auf den ersten Blick keinen Unterschied erkennen. Dieser Eindruck täuscht aber:

- ▶ `*` erfasst fast alle Zeichen, also auch Punkte, sofern diese nicht am Beginn des Dateinamens stehen. Wenn Sie alle Dateien bearbeiten möchten, heißt es unter Linux `*` und nicht `*.*!` (Anmerkungen zu versteckten Dateien folgen weiter unten.)
- ▶ Auch mehrere Jokerzeichen bringen Linux nicht aus dem Gleichgewicht. Sie können beispielsweise mit `*graf*` alle Dateien suchen, die `graf` in ihrem Namen enthalten – also etwa `grafik.doc`, `apfelgraf` und `README.graf`.

**[ ] und [!]** Wenn Ihnen die Jokerzeichen `?` und `*` zu allgemein sind, können Sie eine stärkere Einschränkung durch die Angabe eckiger Klammern erreichen. `[abc]` steht als Platzhalter für einen der drei Buchstaben `a`, `b` oder `c`. Wenn innerhalb der eckigen Klammern ein Bindestrich zwischen zwei Buchstaben oder Ziffern angegeben wird, dann ist ein Zeichen dazwischen gemeint:

- ▶ `[a-f]*` erfasst demnach alle Dateien, die mit einem Buchstaben zwischen `a` und `f` beginnen.
- ▶ `*[_.-]*` meint alle Dateien, die irgendwo in ihrem Dateinamen zumindest einen Punkt, Unterstrich oder Bindestrich enthalten.
- ▶ Durch ein Ausrufezeichen kann der Ausdruck negiert werden: `[!a-z]*` meint alle Dateien, die mit einem Großbuchstaben oder mit einem Sonderzeichen beginnen.
- ▶ `*.[hc]` erfasst alle Dateien, die mit `.c` oder `.h` enden.

Die Jokerzeichen können auch für Verzeichnisse verwendet werden. `*/*.jpg` erfasst alle `*.jpg`-Dateien, die sich in Unterverzeichnissen des aktuellen Verzeichnisses befinden (nur eine Ebene darunter, also nicht auch Dateien in Unter-Unterverzeichnissen). `/usr/*bin/*` erfasst alle Dateien in den Verzeichnissen `/usr/bin` und `/usr/sbin`.

\* für  
Verzeichnisse

Für die Auswertung der Jokerzeichen ist nicht das jeweils aufgerufene Kommando zuständig, sondern die Shell, aus der das Kommando aufgerufen wird. Die `bash`, die unter Linux gebräuchlichste Shell, kennt neben den gerade beschriebenen Jokerzeichen eine Menge weiterer Sonderzeichen, die bei der Ausführung eines Kommandos eine besondere Wirkung haben (siehe [Abschnitt 10.6](#), »Substitutionsmechanismen«).

Das folgende Kommando kopiert alle `*.c`-Dateien aus dem Verzeichnis `projekt` in das aktuelle Verzeichnis:

Beispiel

```
user$ cp projekt/*.c .
```

### Komplikationen bei der Verwendung von Jokerzeichen

Der Umgang mit Jokerzeichen sieht auf den ersten Blick einfacher aus, als er in Wirklichkeit ist. Wenn Sie Schwierigkeiten mit Jokerzeichen haben, sollten Sie einfach einige Experimente mit `echo jokerzeichen` durchführen. Dieses Kommando zeigt einfach alle durch eine Jokerzeichen-Kombination erfassten Dateinamen auf dem Bildschirm an, ohne die Dateinamen zu verändern.

Ein Problem besteht darin, dass `*` nicht nur Dateien, sondern auch Verzeichnisse erfasst. `ls *` zeigt aus diesem Grund nicht nur alle Dateien im aktuellen Verzeichnis an, sondern auch den Inhalt aller Unterverzeichnisse, die über `*` erfasst werden. Beim Kommando `ls` kann dieses Problem durch die Option `-d` umgangen werden; bei anderen Kommandos steht diese Option aber nicht zur Verfügung.

Wenn Sie alle Verzeichnisse, nicht aber normale Dateien bearbeiten möchten, hilft die Jokerzeichenkombination `*/.` weiter: Mit ihr werden alle »Dateien« erfasst, die als Unterverzeichnis einen Verweis auf sich selbst enthalten, und das ist eben nur bei Verzeichnissen der Fall. (Verzeichnisse gelten intern als eine Sonderform einer Datei – daher die Anführungszeichen.)

Verzeichnisse mit  
\*/. bearbeiten

```
user$ echo */.
```

Die Tatsache, dass nicht das jeweilige Programm, sondern schon die Shell für die Verarbeitung der Jokerzeichen zuständig ist, hat nicht nur Vorteile. So ist es etwa unmöglich, mit `ls -R *.tex` nach `*.tex`-Dateien auch in Unterverzeichnissen zu suchen. Die Option `-R` für das Kommando `ls` bewirkt eigentlich ein rekursives Durchsuchen von Unterverzeichnissen.

Probleme mit  
\*.endung

Der Grund, warum das Kommando dennoch nicht funktioniert, ist einfach: Die Shell erweitert das Muster `*.tex` für das *aktuelle* Verzeichnis und übergibt die Liste der gefundenen Dateien an `ls`. Dieses Kommando zeigt Informationen zu diesen Dateien an. Wenn Sie keine Verzeichnisse mit der Endung `.tex` haben, ist `ls` damit am Ende – auch die Option `-R` kann daran nichts mehr ändern. Rekursiv durchsucht werden nämlich nur Verzeichnisse, die als Parameter übergeben werden.

Zum Suchen nach Dateien stellt Linux deshalb das sehr viel flexiblere Kommando `find` zur Verfügung. Im Beispiel unten wird eine Liste aller `*.tex`-Dateien im aktuellen Verzeichnis und in allen untergeordneten Verzeichnissen angezeigt. Grundlagen und weitere Beispiele zu `find` folgen in [Abschnitt 11.4](#), »Dateien suchen (`find`, `grep`, `locate`)«.

```
user$ find . -name '*.tex'
```

#### Dateien umbenennen

Leider ist es in Linux nicht möglich, mit dem Kommando `mv *.x *.y` alle `*.x`-Dateien in `*.y`-Dateien umzubenennen. Der Grund für diese Einschränkung ist wieder derselbe wie oben beschrieben: Die Shell ersetzt `*.x` durch die Liste aller Dateien, die diesem Muster entsprechen. Für `*.y` gibt es keine gültigen Dateinamen. An das Kommando `mv` werden daher eine Liste mehrerer Dateien und der Ausdruck `*.y` übergeben – und `mv` weiß dann nicht, was es mit diesen Argumenten tun soll.

Dazu ein konkretes Beispiel: Angenommen, im aktuellen Verzeichnis befinden sich nur die Dateien `markus.x`, `peter.x` und `ulrike.x`. Wenn Sie `mv *.x *.y` ausführen, ersetzt die Shell das Muster `*.x` durch die drei genannten Dateien. Die Shell findet keine passenden Dateien für `*.y` und übergibt das Muster so, wie es ist. Erst jetzt wird das Kommando `mv` gestartet. Es bekommt folgende Parameter, mit denen es erwartungsgemäß nichts anfangen kann:

```
user$ mv markus.x peter.x ulrike.x *.y
```

Selbst wenn an `mv` als Parameterliste `markus.x peter.x ulrike.x markus.y peter.y ulrike.y` übergeben würde, wäre die Wirkung nicht die erwünschte. `mv` ist prinzipiell nicht in der Lage, mehrere Dateien umzubenennen. Entweder werden also *mehrere* Dateien in ein anderes Verzeichnis verschoben oder es wird nur *eine* Datei umbenannt.

#### Umbenennen mit `sed` und `for`

Unix-Experten haben natürlich auch für dieses Problem eine Lösung gefunden: Sie verwenden den Streameditor `sed`. Wegen der eher komplizierten Bedienung von `sed` eignen sich Beispiele wie das folgende eigentlich nur zur Shell-Programmierung.

Kurz zur Funktionsweise: `ls` liefert die Liste der Dateien, die umbenannt werden sollen, und gibt sie an `sed` weiter. `sed` bildet daraus mit dem Kommando `s` (reguläres Suchen und Ersetzen) eine Liste von `cp`-Kommandos und gibt diese wiederum an eine neue Shell `sh` weiter, die die Kommandos schließlich ausführt. Durch die Zeile unten werden alle `*.x`-Dateien in `*.y`-Dateien kopiert:

```
user$ ls *.x | sed 's/\(.*\)\.x$/cp & \1.y/' | sh
```



Eine andere Lösung besteht darin, eine kleine Schleife zu formulieren. Das folgende Kommando bildet zu allen \*.tex-Dateien Backup-Kopien mit der Endung tex~. (Die Endung ~ wird häufig zur Kennzeichnung von Sicherheitskopien verwendet.)

```
user$ for i in *.tex; do cp $i $i~; done
```

### Versteckte Dateien und Verzeichnisse

Unter Linux gelten Dateien und Verzeichnisse, deren Name mit einem Punkt beginnt, als »versteckt«. Das Jokerzeichen \* berücksichtigt gar nicht alle Dateien in einem Verzeichnis: Dateien, die mit einem Punkt beginnen (häufig Konfigurationsdateien, die unsichtbar sein sollen), werden ignoriert.

Wenn Sie nun glauben, Sie könnten unsichtbare Dateien mit .\* erfassen, wird alles noch schlimmer: Damit sind nämlich nicht nur unsichtbare Dateien gemeint, die mit . beginnen, sondern auch die Verzeichnisse . und .. (also das aktuelle und das übergeordnete Verzeichnis). Wenn das jeweilige Kommando in der Lage ist, ganze Verzeichnisse zu bearbeiten, können die Folgen fatal sein.

Das Problem kann mit dem Suchmuster `.[!..]*` umgangen werden. Damit werden alle Dateinamen erfasst, deren erstes Zeichen ein Punkt ist, die mindestens ein weiteres Zeichen aufweisen, das kein Punkt ist, und die beliebig viele (auch null) weitere Zeichen haben.

```
user$ echo .[!..]*
```

Beim Kommando `ls` kann die Option `-a` verwendet werden. Sie führt dazu, dass alle Dateien angezeigt werden, auch unsichtbare. Allerdings dürfen bei dieser Verwendung von `ls` keine Masken (etwa `*rc*`) angegeben werden. `-a` funktioniert nur dann, wenn `ls` sich die Dateien selbst suchen darf und nicht die Shell diese Aufgabe übernimmt.

Wirklich universell funktioniert auch in diesem Fall nur `find`. Das folgende Kommando findet alle versteckten Dateien im aktuellen Verzeichnis:

```
user$ find -maxdepth 1 -type f -name '.*'
```

### Sonderformen von Dateien (Links, Devices etc.)

Neben gewöhnlichen Dateien kennt Linux eine Reihe von Sonderformen, z. B. Verzeichnisse, Links sowie Device-Dateien zum Zugriff auf Hardware-Komponenten. `ls -l` kennzeichnet derartige Sonderformen durch das erste Zeichen der Ausgabe (siehe [Tabelle 11.4](#)).

| Zeichen | Bedeutung                                    |
|---------|--|
| -       | normale Datei                                |
| d       | Verzeichnis                                  |
| l       | symbolischer Link                            |
| c       | zeichenorientiertes Gerät (Character Device) |
| b       | blockorientiertes Gerät (Block Device)       |
| s       | Socket (Inter-Prozess-Kommunikation)         |
| p       | Pipe, First In First Out (FIFO)              |

**Tabelle 11.4** Identifizierung von Spezialdateien in der ersten Spalte des »ls«-Ergebnisses

```
user$ ls -l /dev/s*
brw-rw----. 1 root disk ... /dev/sda          (Block-Device)
brw-rw----. 1 root disk ... /dev/sda1        (Block-Device)
crw-rw----. 1 root disk ... /dev/sg0         (Character-Device)
lrwxrwxrwx. 1 root root ... /dev/stderr -> /proc/self/fd/2 (Link)
...
```

Links lernen Sie gleich im nächsten Abschnitt näher kennen, Device-Dateien in [Abschnitt 11.9](#). FIFO-Dateien ermöglichen es, dass ein Prozess in die Datei schreibt und ein zweiter Prozess die Daten von dort wieder ausliest. Auch Socket-Dateien dienen zur Kommunikation zwischen Prozessen, sind aber intern wesentlich effizienter implementiert.

## 11.2 Links

### Feste Links (Hardlinks)

Links sind Verweise auf Dateien oder Verzeichnisse. Durch Links können Sie von verschiedenen Orten in der Verzeichnisstruktur auf ein- und dieselbe Datei zugreifen, ohne dass diese Datei physikalisch mehrfach gespeichert werden muss. Links sind damit ein wichtiges Hilfsmittel zur Vermeidung von Redundanzen. Im Linux-Dateisystem kommen Links besonders häufig in `/bin-` und `/lib-`Verzeichnissen vor. (Sehen Sie sich beispielsweise das Ergebnis des Kommandos `ls -l /usr/bin` an!)

Am einfachsten sind Links anhand eines Beispiels zu verstehen: Angenommen, im Verzeichnis `test` befindet sich die Datei `abc`; durch das Kommando `ln abc xyz` wird scheinbar eine neue Datei `xyz` erstellt. In Wahrheit sind aber `abc` und `xyz` nur zwei Verweise auf ein und dieselbe Datei. Die einzige Möglichkeit, das zu überprüfen, bietet das Kommando `ls` mit der Option `-l`. Es gibt in der zweiten Spalte an, wie viele Links auf eine bestimmte Datei zeigen – im vorliegenden Beispiel also 2. Wenn zusätzlich die Option `-i` verwendet wird, gibt `ls` auch den Inode der Datei an, der bei Links identisch ist. Inodes sind interne Identifikationsnummern des Dateisystems.

```

user$ ls -li
59293 -rw-r--r-- 1 root    root    1004 Oct  4 16:40 abc
user$ ln abc xyz
user$ ls -li
59293 -rw-r--r-- 2 root    root    1004 Oct  4 16:40 abc
59293 -rw-r--r-- 2 root    root    1004 Oct  4 16:40 xyz

```

Wenn Sie nun eine der beiden Dateien verändern (egal welche), ändert sich automatisch auch die andere Datei – weil es ja in Wirklichkeit nur eine einzige Datei gibt! Wenn Sie eine der beiden Dateien löschen, reduzieren Sie dadurch nur die Anzahl der Links.

### Backup-Dateien von verlinkten Dateien sind problematisch

Wenn Sie fest verlinkte Dateien mit einem Texteditor bearbeiten, treten bisweilen seltsame Ergebnisse auf: Der Link zeigt nach dem ersten Speichern auf die Backup-Datei und beim zweiten Speichern ins Leere.

Der Grund: Manche Editoren erzeugen beim Speichern eine Backup-Datei, indem sie die vorhandene Datei umbenennen, also beispielsweise abc in abc~. Die geänderte Datei wird vollkommen neu angelegt, erhält einen neuen Inode und ist damit frei von Links. Abhilfe: Verwenden Sie symbolische Links.

Linux kennt zwei Formen von Links. Das obige Beispiel hat feste Links (Hardlinks) vorgestellt, wie sie standardmäßig durch das Kommando `ln` erzeugt werden. Wird `ln` dagegen mit der Option `-s` verwendet, erzeugt das Kommando symbolische Links. Symbolische Links werden manchmal auch weiche Links oder Softlinks genannt. Sie haben den Vorteil, dass sie innerhalb des Dateisystems von einer physikalischen Festplatte auf eine andere verweisen können und dass sie nicht nur auf Dateien, sondern auch auf Verzeichnisse angewandt werden können. Beides ist mit festen Links normalerweise nicht möglich. Einen Sonderfall stellen feste Links auf Verzeichnisse dar, die zwar möglich sind, aber nur von `root` erstellt werden können.

Symbolische  
Links

Durch `ls` wird bei symbolischen Links angezeigt, wo sich die Ursprungsdatei befindet. Es wird allerdings kein Zähler verwaltet, der angibt, von wie vielen Stellen auf die Ursprungsdatei verwiesen wird.

Intern besteht der Unterschied zwischen festen und symbolischen Links darin, dass im einen Fall der Inode, im anderen Fall der Dateiname oder (bei Links über ein Verzeichnis hinaus) die Pfadangabe gespeichert wird.

```

user$ ln -s abc efg
user$ ls -li
59293 -rw-r--r-- 2 root    root    1004 Oct  4 16:40 abc
59310 lrwxrwxrwx 1 root    root      3 Oct  4 16:52 efg -> abc
59293 -rw-r--r-- 2 root    root    1004 Oct  4 16:40 xyz

```

**Tipp**

Bevor Sie einen symbolischen Link einrichten, sollten Sie immer in das Verzeichnis wechseln, das den Link enthalten wird. Andernfalls kann es passieren, dass der Link nicht dorthin zeigt, wohin Sie es erwarten.

Symbolische Links verhalten sich ein wenig anders als feste Links. Das Löschen der Ursprungsdatei (also z. B. abc aus dem vorigen Beispiel) verändert den Link auf diese Datei nicht, `efg` verweist jetzt aber auf eine gar nicht vorhandene Datei. Wird dagegen der symbolische Link gelöscht, hat das keinen Einfluss auf die Ursprungsdatei.

Symbolische Links können nicht nur für Dateien, sondern auch für Verzeichnisse erstellt werden. Das kann einige Verwirrung stiften, weil durch einen symbolischen Link ganze Verzeichnisbäume scheinbar verdoppelt werden. In Wirklichkeit stellt der Verzeichnis-Link aber nur einen zusätzlichen Pfad zu denselben Dateien und Unterverzeichnissen dar.

Generell sollten Sie versuchen, möglichst keine absoluten, sondern nur relative Pfadangaben in Links zu verwenden. Damit vermeiden Sie Probleme, die sich beim Mounten von Verzeichnissen per NFS oder beim Verschieben von Verzeichnissen ergeben können.

**Feste Links versus symbolische Links**

Sowohl symbolische als auch feste Links haben Vorteile. Symbolische Links sind einfacher in der Handhabung. Dafür verbrauchen feste Links weniger Speicher und sind schneller.

**11.3 Dateitypen (MIME)**

Sie klicken in einem Webbrowser oder Dateimanager auf einen Link, der auf eine MP3-Datei verweist – und die MP3-Datei wird automatisch in einem Audio-Player abgespielt. Wenn das funktioniert, ist MIME korrekt konfiguriert.

- MIME** MIME steht für »Multipurpose Internet Mail Extensions«. Ursprünglich bezog sich MIME auf E-Mail-Attachments. Wenn mit einer E-Mail beispielsweise eine PDF- oder JPEG-Datei mitgesandt wird, dann sollte der E-Mail-Client wissen, mit welchem Programm diese Datei betrachtet bzw. bearbeitet werden kann. Damit das funktioniert, ist die MIME-Konfiguration erforderlich.

Mittlerweile reicht die Anwendung von MIME aber viel weiter: Wenn Sie im Dateimanager oder Webbrowser einen Link auf eine Datendatei verfolgen, sollte auch dieses Programm wissen, wie es mit diesen Daten umgehen soll. Die Bedeutung einer korrekten MIME-Konfiguration erstreckt sich also auf alle Programme, die mit unterschiedlichen Datentypen zurecht kommen müssen.

Linux wäre nicht Linux (oder Unix), wenn es *einen* zentralen Ort für die MIME-Konfiguration gäbe. Stattdessen gibt es eine ganze Menge Speicherorte. Die MIME-Daten für KDE-Programme, Gnome-Programme, diverse Webbrowser, für das Drucksystem CUPS etc. werden jeweils separat verwaltet. Außerdem gibt es noch eine zentrale MIME-Konfiguration für alle Programme, die keine eigenen MIME-Konfigurationsdateien verwalten.

MIME-  
Konfiguration

Die Aufteilung der MIME-Konfiguration auf mehrere Orte hat natürlich gute Gründe: Sowohl KDE als auch Gnome verwenden ein Konzept, das Komponenten zur Bearbeitung verschiedener Datentypen vorsieht. Wenn im KDE-Dateimanager eine PNG-Bilddatei angezeigt werden soll, wird einfach die entsprechende Komponente geladen und ausgeführt. Da die KDE- und Gnome-Bibliotheken in der Regel zueinander inkompatibel sind, wäre es fatal, wenn der KDE-Dateimanager versuchen würde, eine Gnome-Komponente auszuführen (oder umgekehrt). Um das zu vermeiden, verwenden KDE und Gnome jeweils ihre eigene MIME-Datenbank. Ähnlich ist die Argumentation auch bei allen anderen Programmen mit eigener MIME-Konfiguration.

Bei vielen MIME-Konfigurationsdateien muss darüber hinaus zwischen der globalen und der individuellen Konfiguration unterschieden werden, also zwischen der Grundeinstellung für alle Anwender und den benutzerspezifischen Einstellungen. Im Folgenden wird nur die MIME-Grundkonfiguration von Linux präsentiert. Anwendungsspezifische MIME-Details sind in anderen Kapiteln beschrieben: die KDE-MIME-Konfiguration also im KDE-Kapitel etc.

Die allgemeinen MIME-Konfigurationsdateien werden nur von den Programmen berücksichtigt, die keine eigenen MIME-Dateien verwalten. Die Einstellungen sind auf zwei Dateien verteilt, von denen es jeweils eine globale und eine benutzerspezifische Version gibt (siehe [Tabelle 11.5](#)).

Allgemeine  
MIME-  
Konfiguration

| Datei           | Bedeutung                            |
|-----------------|--------------------------------------|
| /etc/mime.types | globale Konfiguration für Dateitypen |
| /etc/mailcap    | globale Konfiguration für Programme  |
| .mime.types     | lokale Konfiguration für Dateitypen  |
| .mailcap        | lokale Konfiguration für Programme   |

**Tabelle 11.5** MIME-Konfigurationsdateien

`mime.types` enthält eine Liste, die die Zuordnung zwischen Dateitypen (erste Spalte) und Dateikennungen (alle weiteren Spalten) herstellt. Die erste Beispielzeile ordnet dem Typ `application/pdf` die Kennung `*.pdf` zu. In `mime.types` wird zum Teil zwischen Text- und X-Applikationen unterschieden, weswegen Sie Dateitypen wie `application/x-name` finden werden:

```
# in /etc/mime.types
...
application/pdf    pdf
```

`mailcap` gibt an, welches Programm zur Anzeige bzw. Bearbeitung eines bestimmten Dateityps verwendet werden soll. Die folgende Zeile besagt, dass zur Anzeige von PDF-Dateien das Programm `evince` verwendet werden soll. Im Gegensatz zu `mime.types` müssen die Spalten in `mailcap` durch Semikola getrennt werden. `%s` ist ein Platzhalter für den Dateinamen.

```
# in /etc/mailcap
application/pdf; evince %s
```

#### Magic-Dateien zur Erkennung des Dateityps

MIME ist für die Zuordnung zwischen dem Dateityp und den dazu passenden Programmen zuständig. Aber wie wird der Dateityp überhaupt festgestellt? Der Normalfall besteht darin, dass die Dateikennung den Dateityp angibt. Die Dateikennung `*.ps` deutet beispielsweise auf eine PostScript-Datei hin.

Bei Dateien ohne Kennung versuchen das Programm `file` bzw. entsprechende KDE- oder Gnome-Äquivalente den Dateityp aus dem Inhalt der ersten Bytes bzw. anhand von charakteristischen Zeichenketten zu erkennen, die in der Datei enthalten sind. Das Erkennungsverfahren basiert auf in das Kommando `file` einkompilierten Informationen darüber, welche Byte- und Zeichenmuster eine Datei enthalten kann. Bei einigen Distributionen kann die Standardkonfiguration durch die Dateien `/etc/magic` bzw. in `.magic` verändert werden.

## 11.4 Dateien suchen (`find`, `grep`, `locate`)

Linux bietet eine Menge Möglichkeiten, um nach Dateien zu suchen (siehe [Tabelle 11.6](#)). Welches Kommando am besten geeignet ist, hängt davon ab, um welche Art von Datei es sich handelt (Textdatei, Programm etc.) und welche Informationen bekannt sind – z. B. Teile des Dateinamens oder Suchbegriffe für den Inhalt.

| Kommando | Funktion  |
|----------|---|
| grep     | sucht Text in einer Textdatei.                  |
| find     | sucht Dateien nach Name, Datum, Größe etc.      |
| locate   | sucht Dateien nach ihrem Namen.                 |
| whereis  | sucht Dateien in vordefinierten Verzeichnissen. |
| which    | sucht Programme in PATH-Verzeichnissen.         |

**Tabelle 11.6** Kommandos zur Dateisuche

### which und whereis

`which` sucht nach dem angegebenen Kommando. Es liefert den vollständigen Namen des Kommandos, das ausgeführt werden würde, wenn der Kommandoname ohne Pfadinformationen aufgerufen würde.

`which` durchsucht lediglich die in `PATH` angegebenen Verzeichnisse und arbeitet daher außerordentlich schnell. `PATH` enthält eine Liste von Verzeichnissen, in denen sich Programme befinden. Beachten Sie aber, dass `PATH` für `root` mehr Verzeichnisse enthält als für gewöhnliche Benutzer. Wenn Sie also Systemkommandos suchen, müssen Sie sich als `root` einloggen.

```
user$ which emacs
/usr/bin/emacs
```

`whereis` durchsucht alle üblichen Pfade für Binärdateien, Konfigurationsdateien, man-Seiten und Quellcode nach dem angegebenen Dateinamen. `whereis` erfasst damit mehr Verzeichnisse als `which` und beschränkt sich nicht nur auf Programme. Es versagt allerdings für Dateien, die sich nicht in den für `whereis` vordefinierten Verzeichnissen befinden (siehe `man whereis`).

```
user$ whereis fstab
fstab: /etc/fstab /usr/include/fstab.h /usr/share/man/man5/fstab.5.gz
```

### locate

`locate` `muster` findet Dateien, bei denen das angegebene Suchmuster im vollständigen Dateinamen vorkommt, also im Pfad plus Dateinamen. Die Suche ist sehr schnell: `locate` durchsucht nämlich nicht das Dateisystem, sondern greift auf eine Datenbank zurück, die eine Liste aller Dateinamen des Dateisystems enthält. Je nach Distribution zeigt `locate` nur solche Dateien an, auf die der Benutzer tatsächlich Zugriff hat. Führen Sie `locate` gegebenenfalls als `root` aus, wenn Sie nach Systemdateien suchen. `locate` kann nur benutzt werden, wenn das entsprechende Paket installiert ist, was nicht bei allen Distributionen standardmäßig der Fall ist.

**Beispiele** Das folgende Kommando sucht die X-Konfigurationsdatei `xorg.conf`:

```
user$ locate xorg.conf
/usr/share/X11/xorg.conf.d
/usr/share/X11/xorg.conf.d/10-evdev.conf
/usr/share/X11/xorg.conf.d/10-quirks.conf
/usr/share/X11/xorg.conf.d/11-evdev-quirks.conf
...
```

Die Suche nach `dvips` liefert (sofern dieses Paket sowie  $\text{\LaTeX}$  installiert ist) sehr viele Treffer, weil der Suchbegriff in mehreren Verzeichnisnamen vorkommt. Anstatt alle Suchergebnisse anzuzeigen, werden diese mit `wc` gezählt:

```
user$ locate dvips | wc -l
421
```

Die Anzahl der Ergebnisse wird wesentlich kleiner, wenn Sie nur nach Dateien suchen, die mit `dvips` enden:

```
user$ locate '*dvips'
/usr/bin/dvips
/usr/bin/odvips
/usr/bin/opdvips
/usr/bin/pdvips
/usr/local/texmf/dvips
/usr/local/texmf/fonts/map/dvips
...
```

**updatedb** Die Qualität der Suchergebnisse steht und fällt mit der Aktualität der Datenbank für `locate`. Bei den meisten Distributionen wird die `locate`-Datenbank einmal täglich durch das Kommando `updatedb` aktualisiert. `updatedb` kann natürlich jederzeit auch manuell ausgeführt werden. Das erfordert aber `root`-Rechte.

**Distributions-spezifische Details** Je nach Distribution sind `locate` und `updatedb` unterschiedlich implementiert. Bei Debian, Fedora und Ubuntu stellt das standardmäßig installierte Paket `mlocate` die Kommandos `locate` und `updatedb` zur Verfügung. Die Dateidatenbank befindet sich in der Datei `/var/lib/mlocate/mlocate.db` und wird einmal täglich durch den Cron-Job `/etc/cron.daily/mlocate` aktualisiert. Die Konfigurationsdatei `/etc/updatedb.conf` bestimmt, welche Verzeichnisse und Dateisysteme nicht berücksichtigt werden (z. B. Netzwerkdateisysteme).

Bei openSUSE steht `locate` standardmäßig nicht zur Verfügung. Bevor Sie das Suchkommando nutzen können, müssen Sie das Paket `findutils-locate` installieren und als `root` einmalig `updatedb` ausführen. In Zukunft wird das Kommando einmal täglich durch den Cron-Job `/etc/cron.daily/suse.de-updatedb` aktualisiert. Die Konfiguration erfolgt durch `/etc/sysconfig/locate`.



## find und grep

`find` ist ein ebenso leistungsfähiges wie komplexes Kommando zur Suche nach Dateien. Es berücksichtigt verschiedene Suchkriterien: ein Muster für den Dateinamen, die Dateigröße, das Datum der Erstellung oder des letzten Zugriffs etc. Eine vollständige Referenz aller Optionen gibt man `find`. Die folgenden Beispiele führen aber wohl am besten in den Umgang mit `find` ein. Beachten Sie, dass `find` ein vergleichsweise langsames Kommando ist, weil es das Dateisystem Verzeichnis für Verzeichnis durchsucht.

Ohne weitere Parameter liefert `find` eine Liste aller Dateien im aktuellen Verzeichnis `find` und in allen Unterverzeichnissen:

```
user$ find
...
```

Das folgende Kommando sucht alle Dateien im aktuellen Verzeichnis und in allen Unterverzeichnissen, die mit `.e` beginnen:

```
user$ find -name '.e*'
./.evolution
./.emacs
./.emacs~
./.esd_auth
...
```

`find` sucht ausgehend vom Verzeichnis `/usr/share/texmf` alle `*.tex`-Dateien in einem Verzeichnis, das mit `latex` endet:

```
user$ find /usr/share/texmf -path '*latex/*.tex'
/usr/share/texmf/ptex/platex/base/plnews03.tex
/usr/share/texmf/ptex/platex/base/kinsoku.tex
...
```

Im nächsten Beispiel sucht `find` alle Verzeichnisse innerhalb von `/etc/`. Gewöhnliche Dateien in `/etc` werden dagegen nicht angezeigt. Die Ergebnisliste wird durch `sort` alphabetisch geordnet, was standardmäßig nicht der Fall ist.

```
root# find /etc -type d | sort
/etc
/etc/acpi
/etc/acpi/actions
...
```

Im Folgenden sucht `find` alle Dateien in den (Unter-)Verzeichnissen von `/home`, die Benutzern der Gruppe `users` gehören und deren Inhalt in den letzten fünf Tagen in irgendeiner Form verändert wurde:

```
root# find /home -group users -mtime -5
...
```

`find -mtime +5` findet Dateien, die vor *mehr* als fünf Tagen verändert wurden, und `-mtime 5` liefert solche Dateien, die vor *genau* fünf Tagen verändert wurden. `find` rechnet dabei in Vielfachen von 24 Stunden vom aktuellen Zeitpunkt aus. Wenn Sie statt `-mtime` die Option `-ctime` verwenden, gilt die *inode change time* als Änderungszeitpunkt. Dieser Zeitpunkt verändert sich beispielsweise auch dann, wenn nicht der Inhalt, sondern z. B. die Zugriffsrechte verändert werden.

Das folgende Kommando löscht alle Backup-Dateien im aktuellen Verzeichnis und in allen Unterverzeichnissen. Dabei wird die Liste aller infrage kommenden Dateien mit `find` gebildet und durch Kommandosubstitution `$(kommando)` an `rm` weitergeleitet.

```
user$ rm $(find . -name '*~')
```

Falls es sich um *sehr* viele Dateien handelt, tritt bei der Ausführung des obigen Kommandos ein Fehler auf: Die Kommandozeile mit allen `*~`-Dateien wird so lang, dass sie die maximale Kommandozeilenlänge überschreitet. In solchen Fällen müssen Sie entweder die `-exec`-Option des `find`-Kommandos oder das Kommando `xargs` zu Hilfe nehmen.

- grep** Das Kommando `grep` durchsucht eine Textdatei nach einem Suchmuster. Je nach Einstellung der Optionen zeigt das Kommando anschließend die gefundenen Textpassagen an oder gibt einfach nur an, in wie vielen Zeilen das Suchmuster gefunden wurde. Das Suchmuster ist ein sogenannter regulärer Ausdruck.

Das folgende Kommando durchsucht alle `*.tex`-Dateien des aktuellen Verzeichnisses nach der Zeichenkette »emacs«. Die Liste aller gefundenen Zeilen, denen jeweils der Dateiname vorangestellt ist, wird im Terminal angezeigt.

```
user$ grep emacs *.tex
...
```

`grep` ermittelt hier, wie oft die Funktion `arctan` in den angegebenen `*.c`-Dateien verwendet wird:

```
user$ grep -c arctan\(.*\) *.c
```

`grep -v` liefert als Ergebnis alle Zeilen, die das Suchmuster nicht enthalten. Im folgenden Beispiel entfernt `grep` aus `configfile` alle Zeilen, die mit dem Zeichen `#` beginnen – also alle Kommentare. Das nachgestellte `cat`-Kommando eliminiert außerdem alle leeren Zeilen. Das Endergebnis wird in der Datei `nocomments` gespeichert. Die Anweisung ist praktisch, wenn wenige Konfigurationszeilen in Hunderten oder Tausenden von Kommentarzeilen untergehen.

```
user$ grep -v '^#' configfile | cat -s > nocomments
```

Sie können `find` und `grep` auch kombinieren, um besonders wirkungsvolle Suchen durchzuführen. Im folgenden Beispiel durchsucht `find` alle `*.tex`-Dateien daraufhin, ob in ihnen die Zeichenkette »emacs« vorkommt. Wenn das der Fall ist, wird der Dateiname auf dem Bildschirm ausgegeben. Beachten Sie, dass die Option `-print` nicht vor `-exec` angegeben werden darf. Im Gegensatz zum obigen Beispiel `grep emacs *.tex` berücksichtigt dieses Beispiel auch `*.tex`-Dateien in beliebig tief verschachtelten Unterverzeichnissen.

**find und grep kombinieren**

```
user$ find -name '*.tex' -type f -exec grep -q emacs {} \; -print
...
```

Das folgende Kommando durchsucht alle Dateien im aktuellen Verzeichnis, die kleiner als 10 KiB sind, nach dem regulären Ausdruck `case.*in`. Die Liste der gefundenen Dateien wird in der Datei `ergebnis` gespeichert. Durch die Einschränkung der Dateigröße auf 10 KiB wird versucht, die zumeist erheblich größeren binären Dateien aus der Suche auszuschließen.

```
user$ find -name '*' -maxdepth 1 -size -10k -exec grep -q \
    case.*in {} \; -print > ergebnis
```

## 11.5 Zugriffsrechte, Benutzer und Gruppenzugehörigkeit

Linux ist als Multiuser-System konzipiert und benötigt daher Mechanismen, die steuern, wer auf welche Dateien zugreifen darf, wer sie ändern darf etc. Die Basis des Zugriffssystems stellt die Verwaltung von Benutzern und Gruppen dar, die in [Abschnitt 18.5](#), »Benutzer und Gruppen, Passwörter«, beschrieben wird. Vorerst gehen wir einfach einmal davon aus, dass jeder Benutzer unter Linux einem Benutzer-Account und einer oder mehreren Gruppen zugeordnet ist. Diese Minimalvoraussetzung reicht aus, um das Konzept der Zugriffsrechte von Dateien zu verstehen.

### Zugriffsrechte für Dateien

Mit jeder Datei bzw. mit jedem Verzeichnis werden folgende Informationen gespeichert:

**Zugriffsrechte pro Datei**

- ▶ der Besitzer (Owner) der Datei
- ▶ eine Gruppe, der die Datei zuzuordnen ist
- ▶ neun Zugriffsbits (`rwxrwxrwx` für Read/Write/Execute für den Besitzer, für alle Gruppenmitglieder und für den Rest der Welt)
- ▶ einige weitere Zusatzbits für Spezialfunktionen

Der Besitzer (Owner) einer Datei ist in der Regel die Person, die die Datei erzeugt hat. Als Gruppe wird normalerweise die primäre Gruppe des Besitzers verwendet – als die Defaultgruppe des Besitzers.

Die Zugriffsinformationen *r*, *w* und *x* steuern, wer die Datei lesen (*read*), schreiben (*write*) und ausführen darf (*execute*). Diese Informationen werden getrennt für den Besitzer, für die Gruppe und für alle anderen Benutzer gespeichert. Das ermöglicht es, dem Besitzer mehr Rechte zu geben als anderen Benutzern. Die Informationen werden meist »Zugriffsbits« genannt, weil sie intern als Zahl mit bitweiser Codierung gespeichert werden.

### Wer darf eine Datei löschen?

Die Zugriffsrechte einer Datei haben *keinen* Einfluss darauf, wer eine Datei löschen darf. Darüber entscheidet einzig und allein derjenige, der Zugriff auf das *Verzeichnis* hat, in dem sich die Datei befindet! Eine Datei darf löschen, wer für das Verzeichnis die Rechte *w* und *x* hat. Mehr Informationen zu den Zugriffsrechten für Verzeichnisse folgen im nächsten Abschnitt.

Die Zugriffsbits, der Besitzer sowie die Gruppenzugehörigkeit einer Datei können mit `ls -l` betrachtet werden. Für eine typische Textdatei liefert `ls` das folgende Ergebnis:

```
michael$ ls -l datei.txt
-rw-r----- 1 michael users      3529 Oct  4 15:43 datei.txt
```

Kurz die Interpretation: Das erste Zeichen gibt den Dateityp an. Das Zeichen `-` bedeutet, dass es sich um eine normale Datei handelt. Andere Möglichkeiten sind `d` für ein Verzeichnis (*Directory*), `l` für einen symbolischen Link etc.

Die drei Zeichen `rw-` geben an, dass die Datei vom Besitzer `michael` gelesen und verändert werden kann. Da es sich um eine Textdatei handelt, ist das erste `x`-Bit deaktiviert, die Datei kann also nicht ausgeführt werden.

Die folgenden drei Zeichen `r--` geben an, dass alle Mitglieder der Gruppe `users` die Datei lesen, aber nicht verändern dürfen.

Aus den letzten drei Zeichen `---` geht hervor, dass andere Benutzer – die also weder `michael` noch Mitglieder der Gruppe `user` sind – die Datei weder lesen noch verändern dürfen.

Wenn `michael` möchte, dass diese Datei von allen Anwendern gelesen werden kann, dann muss er das letzte `r`-Bit aktivieren. Dazu verwendet er das Kommando `chmod o+r`:

```
michael$ chmod o+r datei.txt
michael$ ls datei.txt -l
-rw-r--r-- 1 michael users      3529 Oct  4 15:43 datei.txt
```

Manchmal sollen *zwei* oder mehr Benutzer die Möglichkeit bekommen, die Datei zu verändern. Dazu kann eine neue Gruppe gebildet werden, der diese Benutzer angehören. Wenn `michael` und `kathrin` das Dokumentationsteam einer Firma bilden, wäre als Gruppenname etwa `dokuteam` sinnvoll. Anschließend wird die Gruppenzugehörigkeit mit `chgrp` geändert:

```
michael$ chgrp dokuteam datei.txt
michael$ chmod g+rw datei.txt
michael$ ls datei.txt -l
-rw-rw-r-- 1 michael dokuteam 3529 Oct 4 15:43 datei.txt
```

Tatsächlich ist die gemeinsame Bearbeitung von Dateien noch ein wenig diffiziler: Es muss auch sichergestellt werden, dass alle Benutzer Zugriff auf das *Verzeichnis* haben, in dem sich die Dateien befinden. Mehr Details zu diesem Thema folgen gleich.

Statt in der Schreibweise `rw-rw-rw-` werden die neun Zugriffsbits sowie drei weitere Spezialbits oft auch oktal dargestellt. Das ist wahrscheinlich die populärste Anwendung, die dieses Zahlensystem auf der Basis der Zahl 8 bis heute hat.

Oktale  
Schreibweise

Den Zugriffsbits für den Benutzer, die Gruppe und alle anderen ist jeweils eine Ziffer zugeordnet (siehe [Tabelle 11.7](#)). Jede Ziffer ist aus den Werten 4, 2 und 1 für `r`, `w` und `x` zusammengesetzt. `660` bedeutet daher `rw-rw----`, `777` steht für `rw-rw-rw-`. Die `setuid`-, `setgid`- und `sticky`-Bits, die in [Abschnitt 11.6](#), »Spezialbits und die `umask`-Einstellung«, vorgestellt werden, haben die Oktalwerte `4000`, `2000` und `1000`.

| Code  | Bedeutung   |
|---|---|
| 4000 = s = setuid<br>2000 = s = setgid<br>1000 = t = sticky | Spezialbits   |
| 400 = r = read<br>200 = w = write<br>100 = x = execute      | Zugriffsbits für den Besitzer (u = user in <code>chmod</code> ) |
| 40 = r<br>20 = w<br>10 = x                                  | Zugriffsbits für Gruppenmitglieder (g = group)                  |
| 4 = r<br>2 = w<br>1 = x                                     | Zugriffsbits für alle anderen (o = others)                      |

**Tabelle 11.7** Oktalcodes für die Zugriffsbits

Mit dem Kommando `chmod` können Sie die Zugriffsbits auch oktal einstellen, was viele erfahrene Benutzer wegen des geringeren Tippaufwands vorziehen:

```
user$ chmod 640 datei.txt
```

Erstaunlicherweise ist `ls` aber nicht in der Lage, die Zugriffsbits oktal darzustellen. Abhilfe schafft das Kommando `stat`:

```
user$ stat -c "%a %n" *
755 php53-beispiele
550 Private
755 samples
...
```

#### Zugriffsrechte auf Devices

Der Zugriff auf Hardware-Komponenten wie Festplatten, DVD-Laufwerke, Schnittstellen etc. erfolgt in Linux über sogenannte Devices (siehe [Abschnitt 11.9](#)). Um gezielt steuern zu können, welcher Benutzer auf welche Devices zugreifen darf, sind den Devices unterschiedliche Benutzergruppen zugeordnet. Beispielsweise sind die Devices `/dev/ttyS*` für die seriellen Schnittstellen unter Debian und Ubuntu der Gruppe `dialout` zugeordnet:

```
root# ls -l /dev/ttyS1
crw-rw---- 1 root dialout 5, 65 Jul 18 /dev/ttyS1
```

Wenn der Systemadministrator möchte, dass der User `hubert` die serielle Schnittstelle nutzen darf, fügt er `hubert` zur Gruppe `dialout` hinzu:

```
root# usermod -a -G dialout hubert
```

### Zugriffsrechte für Verzeichnisse

#### r-, w- und x-Zugriff auf Verzeichnisse

Die neun Zugriffsbits haben im Prinzip auch bei Verzeichnissen Gültigkeit, allerdings besitzen sie dort eine etwas abweichende Bedeutung:

- ▶ Das `r`-Bit erlaubt Ihnen, die Liste der Dateinamen zu ermitteln (Kommando `ls`).
- ▶ Das `w`-Bit gibt Ihnen das Recht, den Inhalt eines Verzeichnisses zu ändern, also z. B. eine neue Datei zu erzeugen oder eine vorhandene Datei umzubenennen oder zu löschen.
- ▶ Mit dem `x`-Bit können Sie in ein Verzeichnis wechseln (Kommando `cd`). Sie können aber nur auf Dateien zugreifen, deren Namen Sie kennen. Erst die Kombination `rx` ermöglicht es, ein Verzeichnis richtig zu bearbeiten, also z. B. mit `ls -l` eine Liste aller Dateinamen samt detaillierter Informationen zu jeder Datei zu ermitteln. Wenn sowohl `x` als auch `w` gesetzt sind, dürfen im Verzeichnis neue Dateien erzeugt werden.

Die ein wenig merkwürdige Interpretation der `r`- und `x`-Zugriffsrechte hat damit zu tun, dass Verzeichnisse vom Dateisystem als ein Sonderfall einer Datei betrachtet werden; der Inhalt der Verzeichnis-»Datei« ist eine Auflistung der Namen der Dateien, die sich im Verzeichnis befinden, sowie von deren Inode-Nummern.

Tabelle 11.8 fasst zusammen, welche Zugriffsrechte für ein Verzeichnis und die darin enthaltene Datei erforderlich sind, um bestimmte Aktionen durchzuführen. Das Zeichen – in der Spalte *Datei* gibt an, dass die Zugriffsrechte auf die Datei nicht relevant sind. Wie üblich gelten diese Regeln nur für gewöhnliche Benutzer. *root* darf unabhängig von den eingestellten Zugriffsrechten alles!

Wer darf was?

| Aktion                      | Kommando                                    | Datei | Verzeichnis |
|-----------------------------|---|-------|-------------|
| In Verzeichnis wechseln     | <code>cd verzeichnis</code>                 | –     | x           |
| Liste der Dateien ermitteln | <code>ls verzeichnis/*</code>               | –     | r           |
| Dateiinformatoren lesen     | <code>ls -l verzeichnis/*</code>            | –     | rx          |
| Neue Datei erzeugen         | <code>touch verzeichnis/neuedatei</code>    | –     | wx          |
| Datei lesen                 | <code>less verzeichnis/datei</code>         | r     | x           |
| Vorhandene Datei ändern     | <code>cat &gt;&gt; verzeichnis/datei</code> | w     | x           |
| Datei löschen               | <code>rm verzeichnis/datei</code>           | –     | wx          |
| Programm ausführen          | <code>verzeichnis/programm</code>           | x     | x           |
| Script-Datei ausführen      | <code>verzeichnis/script</code>             | rx    | x           |

**Tabelle 11.8** Erforderliche Zugriffsrechte für Standardaktionen

Bei verschachtelten Verzeichnissen ist für die Basisverzeichnisse vor allem das *x*-Bit entscheidend. Ist dieses nicht gesetzt, können die Unterverzeichnisse nicht genutzt werden. In der Praxis ist es zumeist zweckmäßig, für Basisverzeichnisse auch das *r*-Bit zu setzen. Fehlt das Leserecht, muss der Anwender den Namen des Unterverzeichnisses wissen.

Verschachtelte Verzeichnisse

Welche Operationen im Unterverzeichnis erlaubt sind, hängt ausschließlich von den *rwX*-Bits dieses Verzeichnisses ab. Wenn für das Unterverzeichnis die Rechte *rwX* gesetzt sind, können somit Dateien gelesen, erzeugt, verändert und gelöscht werden – selbst dann, wenn in den Basisverzeichnissen die Rechte *r* und *w* fehlen!

Betrachten Sie zum besseren Verständnis das Verzeichnis `/`, das Verzeichnis `/home` und ein darin enthaltenes Benutzerverzeichnis:

```
root# ls -ld /
drwxr-xr-x ... root root ... /home/
root# ls -ld /home/
drwxr-xr-x ... root root ... /home/
root# ls -ld /home/kofler/ (unter Debian, Ubuntu)
drwxr-xr-x ... kofler kofler ... /home/kofler/
```

Für `/` und `/home` gilt: Jeder darf die Namen der in diesem Verzeichnis enthaltenen Dateien und Unterverzeichnisse ermitteln sowie detaillierte Informationen darüber

abfragen, also `ls -l /home` ausführen. Aber nur `root` darf mit `mkdir /home/neuer-` Benutzer neue Heimatverzeichnisse einrichten.

Für `/home/kofler` gilt: Nur der Benutzer `kofler` darf in diesem Verzeichnis neue Dateien anlegen. Alle anderen Benutzer dürfen in das Verzeichnis reinsehen (`ls -l`), aber nichts verändern. Es bleibt dem Besitzer des Heimatverzeichnisses überlassen, die Zugriffsrechte für eigene Dateien und Unterverzeichnisse gegebenenfalls so einzustellen, dass auch ein Lesezugriff unmöglich ist.

Beachten Sie, dass `kofler` selbstverständlich in seinem Heimatverzeichnis Dateien anlegen, verändern und löschen darf, obwohl er keine Schreibrechte in `/` und `/home` hat!

Einige Linux-Distributionen, wie Fedora oder Red Hat, stellen die Zugriffsrechte für die Heimatverzeichnisse restriktiver ein und erlauben wirklich nur dem Besitzer einen Blick in das Verzeichnis:

```
root# ls -ld /home/kofler/      (unter Fedora, RHEL, CentOS)
drwx----- ... kofler kofler ... /home/kofler/
```

#### Verzeichnisse gemeinsam benutzen

Die gerade erwähnten Heimatverzeichnisse sind ein Beispiel für Verzeichnisse, die nur für einen bestimmten Benutzer gedacht sind. Wie aber richten Sie Verzeichnisse ein, die mehrere Benutzer gemeinsam nutzen können – z. B. ein Projektverzeichnis für die Benutzer `sebastian` und `matthias`, sodass beide in dem Verzeichnis Dateien lesen *und* verändern dürfen?

Die Lösung für derartige Probleme sind Gruppen: Sie legen eine neue Gruppe `projektxy` an und ordnen `sebastian` und `matthias` dieser Gruppe zu. Nun brauchen Sie noch ein Projektverzeichnis, das dieser Gruppe zugeordnet ist:

```
root# addgroup projektxy
root# usermod -a -G projektxy sebastian
root# usermod -a -G projektxy matthias
root# mkdir -p /projekte/xy
root# chgrp projektxy /projekte/xy
root# chmod 755 /projekte
root# chmod 2770 /projekte/xy
root# ls -ld /projekte/xy
drwxrws--- ... root projektxy ... /projekte/xy
```

Die Einstellung der Zugriffsrechte für `/projekte` ist wie für das `/home`-Verzeichnis: Nur `root` darf darin neue Dateien und Verzeichnisse anlegen. Alle anderen dürfen die Verzeichnisse benutzen.

Im Verzeichnis `/projekte/xy` haben alle Mitglieder der Gruppe `projektxy` Schreib- und Leserechte, also in diesem Beispiel `sebastian` und `matthias`.



Die einzige Besonderheit ist das Setgid-Bit für dieses Verzeichnis (Oktalcode 2000). Es bewirkt, dass in diesem Verzeichnis eingerichtete Dateien und Verzeichnisse automatisch der Gruppe `projektxy` zugeordnet werden, nicht der Gruppe desjenigen Benutzers, der die Datei erzeugt. Damit wird der Fall vermieden, dass `sebastian` eine neue Datei erzeugt und `matthias` diese zwar sieht und lesen, aber nicht verändern kann. Hintergrundinformationen zum Setgid-Bit folgen im nächsten Abschnitt.

## 11.6 Spezialbits und die umask-Einstellung

Das Prinzip der Zugriffsrechte ist Ihnen nun bekannt. Jetzt fehlen noch zwei Feinheiten: Zum einen gibt es noch drei weitere Zugriffsbits mit den merkwürdigen Namen »Setuid«, »Setgid« und »Sticky«, deren Bedeutung ich Ihnen gleich erkläre. Zum anderen ist noch die Frage zu klären, wem neu erzeugte Dateien gehören. Hier spielt die sogenannte `umask`-Einstellung eine große Rolle.

### Setuid-, Setgid- und Sticky-Bit

Die Bedeutung der drei mal drei Zugriffsbits `rxwxrwxrwx` ist leicht zu verstehen. Darüber hinaus können bei den Zugriffsinformationen von Dateien und Verzeichnissen noch drei weitere Informationen gespeichert werden: das Setuid-Bit, das Setgid-Bit und das Sticky-Bit. Im Regelfall müssen nur Systemadministratoren diese Spezialbits kennen.

Das Setuid-Bit wird oft verkürzt Suid-Bit genannt. Es bewirkt, dass Programme immer so ausgeführt werden, als hätte der Besitzer selbst das Programm gestartet. Oft ist der Besitzer von Programmen `root`; dann kann jeder das Programm ausführen, als wäre er selbst `root`. Intern wird für die Ausführung des Programms die User-Identifikationsnummer des Besitzers der Datei und nicht die UID des aktuellen Benutzers verwendet.

Setuid-Bit  
(Suid-Bit)

Das Bit wird eingesetzt, um gewöhnlichen Besitzern zusätzliche Rechte zu geben, die nur bei der Ausführung dieses Programms gelten. Ein Beispiel für die Anwendung des Setuid-Bits ist das Kommando `/usr/bin/passwd`. Es ermöglicht jedem Benutzer, sein eigenes Passwort zu verändern. Die Passwörter werden aber in der Datei `/etc/shadow` gespeichert, auf die nur `root` Lese- und Schreibzugriff hat. Daher muss `passwd` mit `root`-Rechten ausgeführt werden.

`ls -l` zeigt bei derartigen Programmen bei den Benutzer-Zugriffsbits, also in der ersten `rxwx`-Gruppe, den Buchstaben `s` oder `S` statt des `x` an: ein kleines `s`, wenn das Execute-Bit auch gesetzt ist (der Normalfall), ein großes `S`, wenn nur das Setuid-Bit, nicht aber das Execute-Bit gesetzt ist. Der Oktalwert dieses Bits für `chmod` beträgt 4000.

```
user$ ls -l /bin/mount
-rwsr-xr-x  1 root  root   68508 Feb 25  01:11 /bin/mount
user$ ls -l /usr/bin/passwd
-rwsr-xr-x ... root root ... /usr/bin/passwd
```

Wenn Sie alle Programme finden möchten, bei denen das Setuid-Bit gesetzt ist, verwenden Sie `find` mit der Option `-perm -4000`. Bei einer Ubuntu-Defaultinstallation findet das Kommando rund 15 Setuid-Programme:

```
user find /usr/bin /usr/sbin/ /bin /sbin -perm -4000
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/sudo
...
```

#### Achtung, Sicherheitsrisiko!

Das Setuid-Bit kann leicht zu einem Sicherheitsrisiko werden – insbesondere dann, wenn während der Ausführung des Programms weitere Programme gestartet werden. Aus diesem Grund sollten Sie die Anwendung des Setuid-Bits nach Möglichkeit vermeiden.

Bei Script-Dateien (bash, Python, Tcl etc.) wird das Setuid-Bit unter Linux generell ignoriert! Es ist also unmöglich, einem Script durch ein Setuid-Bit root-Rechte zu geben.

#### Setgid-Bit für Dateien

Das Setgid-Bit hat bei Programmen eine ähnliche Wirkung wie Setuid. Allerdings wird nun während der Ausführung des Programms die Gruppen-Identifikationsnummer der Datei verwendet, nicht die GID des aktuellen Benutzers.

`ls -l` zeigt bei derartigen Programmen für die Gruppen-Zugriffsbits, also in der zweiten `rwx`-Gruppe, den Buchstaben `s` oder `S` anstelle des `x` an. Der Oktalwert dieses Bits beträgt 2000.

#### Setgid-Bit für Verzeichnisse

Bei Verzeichnissen hat das Setgid-Bit eine ganz andere Bedeutung: Wenn es gesetzt ist, wird neu erzeugten Dateien innerhalb dieses Verzeichnisses die Gruppe des Verzeichnisses zugeordnet – anstatt, wie sonst üblich, die Gruppe desjenigen, der die Datei erzeugt.

In der Praxis wird das Setgid-Bit eingesetzt, wenn mehrere Benutzer ein Verzeichnis gemeinsam benutzen sollen: Dann ist es nämlich zweckmäßig, dass neue Dateien der gemeinsamen Gruppe zugeordnet werden und nicht der gerade aktiven Gruppe desjenigen Benutzers, der die Datei erzeugt. Aus diesem Grund wurde das Setgid-Bit im vorigen Abschnitt für das Verzeichnis `/projekte/xy` verwendet.

Das Sticky-Bit bewirkt bei Verzeichnissen, in denen alle die Dateien ändern dürfen, dass jeder nur seine *eigenen* Dateien löschen darf und nicht auch Dateien anderer Benutzer. Das Bit ist beispielsweise beim /tmp-Verzeichnis gesetzt. In diesem Verzeichnis darf jeder Benutzer temporäre Dateien anlegen. Es soll aber vermieden werden, dass auch jeder Benutzer nach Belieben fremde Dateien umbenennen oder löschen kann.

Sticky-Bit für Verzeichnisse

ls -l zeigt bei derartigen Programmen für alle gültigen Zugriffsbits den Buchstaben t anstelle des x an. Der Oktalwert dieses Bits beträgt 1000. Die Bedeutung des Sticky-Bits ist Linux-spezifisch! Bei anderen Unix-Varianten kann das Bit eine andere oder gar keine Bedeutung haben.

```
user$ ls -ld /tmp/
drwxrwxrwt ... root root ... /tmp/
```

Am einfachsten gelingt die Einstellung der Spezialbits durch chmod, wenn Sie mit Oktalcodes arbeiten – also z. B. chmod 2770 verzeichnis. Aber natürlich ist auch eine Veränderung in der chmod-üblichen Syntax owner+-bit möglich:

Veränderung der Spezialbits durch chmod

```
root# chmod u+s datei (setzt das Setuid-Bit)
root# chmod u-s datei (löscht das Setuid-Bit)
root# chmod g+s datei (setzt das Setgid-Bit)
root# chmod g-s datei (löscht das Setgid-Bit)
root# chmod +t datei (setzt das Sticky-Bit)
root# chmod -t datei (löscht das Sticky-Bit)
```

Die Interpretation der Ergebnisse von ls -l ist nicht ganz einfach:

Darstellung der Spezialbits durch ls

- ▶ Das Setuid-Bit wird durch die Buchstaben s/S anstelle des x in der ersten rwx-Gruppe angezeigt.
- ▶ Das Setgid-Bit wird durch die Buchstaben s/S anstelle des x in der zweiten rwx-Gruppe angezeigt.
- ▶ Das Sticky-Bit wird durch die Buchstaben t/T anstelle des x in der dritten rwx-Gruppe angezeigt.

Die Großbuchstaben S und T kommen nur zur Anwendung, wenn das entsprechende Execute-Bit *nicht* gesetzt ist. In der Regel ist das ein Hinweis darauf, dass die Spezialbits falsch verwendet werden.

Linux-intern werden zusammen mit den Zugriffsbits und den Spezialbits auch die Informationen darüber gespeichert, welche Funktion eine Datei hat. Es kann sich beispielsweise um eine normale Datei handeln, um ein Verzeichnis, um einen Link, um ein Block-Device etc.

Dateifunktion bzw. -typ

Die meisten Programme zur Dateiverwaltung verbergen diese Zusatzinformation. Es gibt aber einige wenige Programme, die diese Information als Zahlenwert anzeigen. Bei einer gewöhnlichen Datei lautet die komplette Spezifikation dann 100000 (Kennzeichnung für eine gewöhnliche Datei) plus N000 (Spezialbits) plus xxx (Zugriffsbits), also beispielsweise 100760. Die Zahlencodes für die Dateitypen erhalten Sie mit `man 2 stat`.

### Besitzer und Gruppe neuer Dateien

Dieser Abschnitt beschäftigt sich mit der Frage, welche Faktoren die Zugriffsinformationen *neuer* Dateien bestimmen. Um das einfach auszuprobieren, verwenden Sie das Kommando `touch`. Dieses Kommando erzeugt eine neue, leere Datei, falls die angegebene Datei noch nicht existiert.

**Beispiel** Der Benutzer `michael` erzeugt die neue Datei `myFile1`. Es sollte nicht überraschen, dass diese Datei wieder dem Benutzer `michael` gehört – er hat sie ja gerade selbst erzeugt. Als Gruppenzugehörigkeit wurde automatisch `michael` verwendet. `michael` ist die primäre Gruppe des Benutzers `michael`. (Manche Distributionen weisen nicht jedem Benutzer eine eigene Gruppe zu, sondern allen Benutzern die Gruppe `users`.)

```
michael$ touch myFile1
michael$ ls -l myFile1
-rw-r--r--  1 michael  michael           0 Jun 14 16:45 myFile1
```

`michael` gehört einer Reihe weiterer Gruppen an (Kommando `groups`). Um eine Datei zu erzeugen, die nicht der primären Gruppe angehört, muss zuerst die aktive Gruppe gewechselt werden (Kommando `newgrp`):

```
michael$ groups
michael adm admin cdrom dokuteam dialout lpadmin plugdev sambashare
michael$ newgrp dokuteam
michael$ touch myFile2
michael$ ls -l myFile2
-rw-r--r--  1 michael  dokuteam           0 Jun 14 17:02 myFile2
```

Natürlich hätte `myFile2` auch ohne vorheriges `newgrp` erzeugt werden können. Dann hätte die Gruppenzugehörigkeit nachträglich mit `chgrp` verändert werden müssen. `newgrp` ist dann praktisch, wenn mehrere neue Dateien erzeugt werden, die automatisch einer bestimmten Gruppe angehören sollen.

#### Besitzer und Gruppenzugehörigkeit

Aus den zwei Beispielen oben geht hervor, dass neue Dateien automatisch dem Benutzer gehören, der sie erzeugt. Als Gruppenzugehörigkeit wird normalerweise die primäre Gruppe des Benutzers verwendet. Allerdings gibt es hier zwei Ausnahmen:

- Wenn der Benutzer mit `newgrp` eine andere seiner Gruppen zur aktuellen Gruppe gemacht hat, gehört die neue Datei dieser Gruppe.

- ▶ Wenn in einem Verzeichnis das Setgid-Bit gesetzt ist (siehe den vorigen Abschnitt), dann erhalten darin erzeugte Dateien automatisch dieselbe Gruppe wie das Verzeichnis. Die aktive Gruppe des Benutzers wird nicht berücksichtigt.

### Zugriffsbits neuer Dateien (umask)

Bei den Zugriffsbits ist die Sache etwas komplizierter. Linux sieht eigentlich vor, dass neue Dateien die Zugriffsbits `rw-rw-rw` (oktal 666) bekommen, also von jedem gelesen und verändert werden dürfen. Neue Verzeichnisse und Programmdateien, die von einem Compiler erzeugt werden, bekommen automatisch die Zugriffsbits `rxwxrwxrwx` (777), können also auch von jedem ausgeführt werden.

Zugriffsbits

Für die praktische Arbeit mit mehreren Benutzern wäre diese Grundeinstellung allerdings zu freizügig. Deswegen sehen alle Linux-Shells die sogenannte `umask`-Einstellung vor. Dabei handelt es sich um einen Zahlenwert, der die Bits angibt, die von den Standardzugriffsbits abgezogen werden. Die aktuelle Einstellung des `umask`-Werts können Sie mit dem gleichnamigen Kommando feststellen und bei Bedarf auch verändern. Die folgenden Kommandos zeigen die Defaulteinstellungen für einige Distributionen:

umask-Arithmetik

```
michael$ umask (Debian, SUSE)
0022
michael$ umask (CentOS, Fedora, RHEL, Ubuntu)
0002
```

Viele Linux-Distributionen verwenden also den `umask`-Wert `022` (`---w--w-`). Daher bekommen neue Dateien die Zugriffsbits  $666 - 022 = 644$  (`rw-r--r--`), neue Verzeichnisse und Programme die Zugriffsbits  $777 - 022 = 755$  (`rxwxr-xr-x`).

Der bei Ubuntu und Red-Hat-Derivaten übliche `umask`-Wert `002` ist liberaler: Neue Dateien erhalten die Zugriffsbits  $666 - 002 = 664$  (`rw-rw-r--`), neue Verzeichnisse  $777 - 002 = 775$  (`rxwxrwxr-x`).

Für den Benutzer `root` gilt bei allen von mir getesteten Distributionen die Einstellung `0022`:

umask für root

```
root# umask
0022
```

Wie muss `umask` eingestellt werden, damit neue Dateien die Zugriffsrechte `rw-r-----` = 640 erhalten, neue Verzeichnisse `rxwxr-x---` = 750? Die Antwort ergibt sich aus der Subtraktion von 777 minus dem oktalen Zielwert für Verzeichnisse:  $777 - 750 = 027$ .

Welche umask für welchen Zweck?

```
michael$ umask 27
michael$ touch neue-datei
michael$ mkdir neues-verzeichnis
```

```
michael$ ls -ld neu*
-rw-r----- ... michael michael ... neue-datei
drwxr-x--- ... michael michael ... neues-verzeichnis
```

#### umask-Konfigurationsdateien

Die Einstellung des `umask`-Werts erfolgt in den Konfigurationsdateien der Shells. Für die `bash` wird `umask` meist in `/etc/profile` oder `/etc/bashrc` eingestellt.

Bei neueren Distributionen kümmert sich PAM (siehe [Abschnitt 18.6](#), »PAM, NSS und `nsd`«) um die Einstellung von `umask`. Das PAM-Modul `pam_umask` wertet unter anderem den optionalen Eintrag `ulimit=xxx` in der dritten Spalte (GECOS) von `/etc/passwd` aus. Außerdem werden die Einstellungen aus `/etc/default/login` und `/etc/login.defs` berücksichtigt. Die letztere Datei enthält z. B. unter Ubuntu die `umask`-Defaulteinstellung.

Einzelne Benutzer können bei den meisten Distributionen eine davon abweichende Einstellung in der Datei `~/.bashrc` vornehmen. Wenn Sie beispielsweise möchten, dass von Ihnen erzeugte Dateien nur von den Gruppenmitgliedern, nicht aber von anderen Benutzern gelesen bzw. ausgeführt werden dürfen, verwenden Sie folgende Einstellung:

```
# in ~/.bashrc
umask 027
```

#### Wer darf Zugriffsinformationen ändern?

Bei einer einmal erzeugten Datei werden weder der Besitzer noch die Zugriffsbits geändert, wenn sie von einem anderen Benutzer bearbeitet wird. Nur der Besitzer darf die Gruppenzugehörigkeit und Zugriffsbits ändern. Und nur `root` darf den Besitzer einer Datei verändern. Damit ist es also nicht möglich, dass der Besitzer einer Datei diese einem anderen gleichsam schenkt.

## 11.7 Access Control Lists und Extended Attributes

#### Access Control Lists (ACLs)

Die Unix-typische Verwaltung von Benutzern und Gruppen sowie die darauf aufbauenden Zugriffsrechte für Verzeichnisse und Dateien haben sich seit Jahrzehnten bewährt. Das Konzept ist so einfach, dass man es nach ein paar Stunden versteht. Es gibt allerdings Fälle, in denen dieses einfache System unzureichend ist.

Aus diesem Grund wurde ein feinmaschigeres System zur Verwaltung von Zugriffsrechten entwickelt, das auf sogenannten Access Control Lists (ACLs) basiert. ACLs ermöglichen es, für jede Datei bzw. für jedes Verzeichnis beliebig viele Regeln aufzustellen, welche Benutzer und Gruppen die Datei bzw. das Verzeichnis lesen oder verändern dürfen und wer das – abweichend von den Unix-Zugriffsrechten – *nicht*

darf. ACLs wirken also ergänzend zu den Standardzugriffsrechten und können zusätzliche Rechte einräumen oder vorhandene Rechte entziehen.

ACLs stehen unter Linux standardmäßig zur Verfügung. Bei den Dateisystemen `btrfs` und `xfs` sind ACLs in jedem Fall aktiv. Bei den `ext`-Dateisystemen muss dagegen die `mount`-Option `acl` verwendet werden, um ACLs zu aktivieren.

Der Datei-Server Samba ist das bei Weitem wichtigste Programm, das wirklich von ACLs profitiert. Es ist dank ACLs in der Lage, Windows-Zugriffsrechte unter Linux nachzubilden.

Nur weil ACLs mehr Möglichkeiten bieten, lösen sie die herkömmliche Rechteverwaltung keineswegs ab! Für erfahrene Administratoren großer Netzwerke mögen ACLs zusätzliche Sicherheit bringen oder zumindest die Verwaltung vereinfachen, für die meisten Linux-Anwender ist die gewöhnliche Rechteverwaltung aber absolut ausreichend. Wer das komplexe ACL-System nicht korrekt anwendet, wird womöglich zusätzliche Sicherheitslöcher aufreißen. Daher gibt es momentan kaum Distributionen, die ACLs standardmäßig nutzen.

Einschränkungen

Ein zentrales Problem besteht darin, dass viele Linux-Kommandos und -Programme ACLs nicht korrekt verarbeiten. Da kann es schon einmal passieren, dass einer kopierten Datei plötzlich die ACL-Informationen des Originals fehlen. Auch die meisten Dateimanager können ACLs weder richtig anzeigen noch verändern. Der KDE-Dateimanager Dolphin ist eine positive Ausnahme.

Eng verwandt mit ACLs sind Extended Attributes (EAs). Sie ermöglichen es, zu jeder Datei zusätzliche Attribut-Wert-Paare zu speichern. Sie können einer Textdatei also beispielsweise das Attribut `charset` mit der Einstellung `utf8` zuordnen, um so den benutzten Zeichensatz zu speichern. Das bringt freilich nur dann Vorteile mit sich, wenn es auch Programme gibt, die diese Informationen auswerten. Je nachdem, welches Dateisystem Sie einsetzen, müssen auch EAs durch eine entsprechende `mount`-Option aktiviert werden, beim `ext`-Dateisystem beispielsweise durch `user_xattr`.

Extended Attributes (EAs)

Weitere Hintergrundinformationen und Details zur Anwendung von ACLs und EAs finden Sie in den `man`-Seiten zu `acl`, `getfacl`, `setfacl`, `attr(5)`, `getfattr` und `getsatr`.

In den folgenden Beispielen gehe ich davon aus, dass das Paket `attr` mit den Kommandos `attr`, `getfattr` und `setfattr` installiert ist und dass Sie mit einem Dateisystem arbeiten, in dem ACLs und EAs aktiviert sind. Wenn es sich um ein `ext4`-Dateisystem handelt, sollte das Ergebnis von `mount` so aussehen:

Voraussetzungen

```
user$ mount
...
/dev/sdc5 on /test type ext4 (rw,acl,user_xattr)
...
```

Sollte das nicht der Fall sein, erhalten Sie bei den folgenden Beispielen Fehler der Art *Operation wird nicht unterstützt*. Abhilfe schafft die Veränderung der mount-Optionen in `/etc/fstab` und ein Neueinbinden des Dateisystems. Werfen Sie gegebenenfalls einen Blick in [Kapitel 22](#), »Administration des Dateisystems«. Informationen speziell zur Veränderung der mount-Optionen in `/etc/fstab` finden Sie in [Abschnitt 22.8](#).

### Access Control Lists

**getfacl** Auch bei einem Dateisystem mit ACLs gelten normalerweise die Standardzugriffsrechte, die oft auch als »minimale ACL« bezeichnet werden. `getfacl` zeigt diese Rechte in ACL-Form an:

```
user$ touch datei1
user$ getfacl datei1
# file: datei1
# owner: kofler
# group: kofler
user::rw-
group::r--
other::r--
user$ ls -l datei1
-rw-r--r-- 1 kofler kofler ... datei2
```

**setfacl** Mit `setfacl` definieren Sie nun zusätzliche Zugriffsregeln. Die folgenden Kommandos geben der Benutzerin `gabi` sowie allen Mitgliedern der Gruppe `docuteam` Schreib- und Lesezugriff auf die Datei, verbieten aber der Benutzerin `kathrin` jeglichen Zugriff:

```
user$ setfacl -m gabi:rw datei1
user$ setfacl -m g:docuteam:rw datei1
user$ setfacl -m kathrin:- datei1
```

Die Rechtestliste von `getfacl` ist nun schon etwas länger. `ls` zeigt nun bei den Zugriffsrechten für Gruppenmitglieder die ACL-Maske an. Den Zugriffsbuchstaben folgt das Zeichen `+`, um darauf hinzuweisen, dass es ACL-Regeln gibt.

```
user$ getfacl datei1
# file: datei1
# owner: kofler
# group: kofler
user::rw-
user:gabi:rw-
user:kathrin:---
group::r--
group:docuteam:rw-
mask::rw-
other::r--
user$ ls -l datei1
-rw-rw-r--+ 1 kofler kofler ... datei1
```



Eine typische Anwendung von ACLs besteht darin, dass Sie einem bestimmten Benutzer Zugriff auf Ihre Dateien geben möchten, ohne die Dateien aber gleich allen anderen Benutzern (einer bestimmten Gruppe) zugänglich zu machen. Normalerweise müssten Sie nun den Administrator bitten, dass er eine neue Gruppe einrichtet, der Sie und die weiteren Benutzer angehören, mit denen Sie die Dateien gemeinsam bearbeiten möchten. Mit ACL führen Sie einfach `setfacl -m benutzer:rw datei` aus.

Die ACL-Maske limitiert die Rechte, die durch ACL-Regeln gegeben werden. Wenn Sie die ACL-Maske beispielsweise auf `r` stellen, kann keine ACL-Regel einem Benutzer Schreib- oder Ausführrechte geben. Die ACL-Maske hat also Vorrang gegenüber den ACL-Regeln. Sie hat allerdings keinen Einfluss auf die Rechte, die sich durch die herkömmlichen Zugriffsrechte für den Besitzer der Datei bzw. für Gruppenmitglieder der Datei ergeben.

ACL-Maske

Bei jeder Änderung einer ACL-Regel durch `setfacl` wird die Maske automatisch so neu berechnet, dass alle anderen ACL-Regeln erfüllt werden können. Diese Maske wird von `getfacl` angezeigt und auch bei `ls -l` berücksichtigt.

Sie können die Maske durch `setfacl -m m:RWX datei` explizit einstellen und so die ACL-Rechte limitieren. Dabei ersetzen Sie `RWX` durch die gewünschten Zugriffsbits. Beachten Sie aber, dass Ihre eigene Maske nur so lange gilt, bis Sie eine neue ACL-Regel definieren. Dadurch wird die ACL-Maske automatisch neu berechnet (es sei denn, Sie verhindern das durch die Option `-n`).

Für Verzeichnisse können Sie einen zweiten Satz Regeln für die Standard-ACL festlegen. Die Standard-ACL steuert nicht den Zugriff auf das Verzeichnis, sondern gilt als Muster für neue Dateien. Jede Datei, die innerhalb des Verzeichnisses neu erzeugt wird, erbt gewissermaßen die Standard-ACL des Verzeichnisses. Bei vielen ACL-Anwendungen dient ein neues Verzeichnis mit einer geschickt gewählten Standard-ACL als Ausgangspunkt.

Standard-ACL

Das größte Hindernis für die weitere Verbreitung von ACLs besteht darin, dass viele Standardkommandos und nahezu alle Anwendungsprogramme ACLs einfach ignorieren. Wenn Sie eine Datei mit ACL-Regeln mit `cp` einfach kopieren, hat die Kopie alle ACL-Regeln verloren. Dasselbe gilt, wenn Sie die Datei mit einem Editor, mit LibreOffice oder mit GIMP öffnen und unter einem anderen Namen speichern. Bei `cp` schafft die Option `-p` Abhilfe, aber bei den meisten anderen Kommandos und Programmen fehlen vergleichbare Optionen bzw. ein ACL-konformes Verhalten.

ACL-Kompatibilität

Problematisch sind auch Backups. `tar` und `rsync` eliminieren ACL-Regeln. Das Dateisystem von CDs und DVDs sieht keine ACLs vor, sodass diese Informationen auch dort verloren gehen. Es bestehen zwei Auswege: Entweder setzen Sie statt `tar` die ACL-kompatible Variante `star` ein, oder Sie erzeugen vor dem Backup eine zusätzli-

che Textdatei, die die ACL-Regeln aller Dateien enthält. Nach dem Backup stellen Sie die ACL-Regeln anhand dieser Datei wieder her:

```
user$ getfacl -R --skip-base . > acl-backup    (ACL-Regeln speichern)
user$ setfacl --restore=acl-backup            (ACL-Regeln wiederherstellen)
```

### Extended Attributes

**setfattr und getfattr** Die folgenden Beispiele zeigen, wie Sie mit `setfattr` Attribute speichern und diese mit `getfattr` auslesen:

```
user$ touch datei2
user$ setfattr -n user.language -v de datei2
user$ setfattr --name=user.charset --value=utf8 datei2
user$ getfattr -d datei2
# file: datei2
user.charset="utf8"
user.language="de"
```

`getfattr` liefert normalerweise nur Attribute, deren Name mit »user.« beginnt. Wenn Sie andere Attribute sehen möchten, müssen Sie deren Namen durch `-n` oder deren Muster durch `-m` angeben.

```
user$ getfattr -n security.selinux -d tst
# file: tst
security.selinux="user_u:object_r:user_home_t:s0^000"
```

**EA-Kompatibilität** Es gibt momentan leider kaum Programme, die Extended Attributes beim Kopieren, Archivieren etc. erhalten. Selbst `cp -p` ignoriert die Attribute. Bei Backups gehen Sie am besten ähnlich wie bei ACLs so vor, dass Sie vor dem Backup eine Datei mit allen EAs erstellen. Anhand dieser Datei können Sie die EAs später wiederherstellen.

```
user$ getfattr -R . > ea-backup    (Attribute speichern)
user$ setfattr --restore=ea-backup (Attribute wiederherstellen)
```

### Capabilities

Zu den interessantesten Anwendungen von Extended Attributes gehört die Möglichkeit, bei ausführbaren Dateien anzugeben, welche Operationen für das Programm zulässig sind, also welche »Capabilities« das Programm hat. Das würde es erlauben, weniger Programme mit dem Setuid-Bit zu kennzeichnen und auf diese Weise die Sicherheit von Linux-Distributionen zu erhöhen. Leider gibt es zurzeit keine gängige Distribution, die von diesen Möglichkeiten auch Gebrauch macht.

Damit Capabilities funktionieren, müssen zwei Voraussetzungen erfüllt sein:

Voraussetzungen

- ▶ Die Bibliothek `libcap` muss installiert sein (`/lib/libcap*` oder `/lib64/libcap*`).
- ▶ Das Dateisystem muss EAs unterstützen, weil die Capability-Daten in Form von EAs gespeichert werden. Bei `ext`-Dateisystemen muss daher die `mount`-Option `user_xattr` verwendet werden.

Bei den meisten Distributionen sind die ersten zwei Voraussetzungen standardmäßig erfüllt. Der dritte Punkt erfordert in der Regel eine Änderung von `/etc/fstab`. Weitere Grundlagen zu Capabilities können Sie hier nachlesen:

<https://lwn.net/Articles/313047>

<https://friedhoff.org/posixfilecaps.html>

Um Capabilities zu administrieren, benötigen Sie die Kommandos `getcap` und `setcap`. Sie müssen bei vielen Distributionen extra installiert werden (Paket `libcap-ng-utils`). Das folgende Beispiel demonstriert die Anwendung von Capabilities: Das Netzwerkkommando `ping` ist bei den meisten Distributionen mit dem `setuid`-Bit ausgestattet, sodass es von gewöhnlichen Benutzern verwendet werden kann. Sobald Sie dieses Bit löschen, kann nur noch `root` mit `ping` arbeiten:

`getcap` und  
`setcap`

```
root# chmod u-s /bin/ping
user$ ping yahoo.de
ping: icmp open socket: Die Operation ist nicht erlaubt
```

Anstatt nun das unsichere `setuid`-Bit wieder zu setzen, reicht es auch, dem Kommando `ping` mit `setcap` den Zugriff auf Netzwerkfunktionen des Kernels zu geben. Mit `getcap` können Sie nachsehen, welche Capabilities ein Kommando hat:

```
root# setcap cap_net_raw=ep /bin/ping
root# getcap /bin/ping
/bin/ping = cap_net_raw+ep
```

## 11.8 Die Linux-Verzeichnisstruktur

Ein typisches Unix-System besteht aus Tausenden von Dateien. Während der Entwicklung von Unix haben sich bestimmte Regeln herauskristallisiert, in welchen Verzeichnissen welche Dateien normalerweise gespeichert werden. Diese Regeln wurden an die Besonderheiten von Linux angepasst und in einem eigenen Dokument zusammengefasst: dem Filesystem Hierarchy Standard (FHS). Die meisten Linux-Distributionen halten sich bis auf wenige Ausnahmen an diesen Standard.

Filesystem  
Hierarchy  
Standard

<https://refspecs.linuxfoundation.org/fhs.shtml>

Die in diesem Abschnitt zusammengefassten Informationen geben eine erste Orientierungshilfe. Dabei wurde nicht nur der FHS berücksichtigt, sondern auch die Gepflogenheiten populärer Linux-Distributionen.

Das Dateisystem beginnt mit dem Wurzelverzeichnis. Dort befinden sich normalerweise keine Dateien, sondern nur Verzeichnisse:

- `/bin` enthält elementare Linux-Kommandos zur Systemverwaltung, die von allen Benutzern ausgeführt werden können. Weitere Programme befinden sich in `/usr/bin`. Bei modernen Distributionen ist `/bin` einfach ein Link auf `/usr/bin`; die Trennung zwischen `/bin` und `/usr/bin` wurde damit aufgehoben.
- `/boot` enthält Dateien, die zum Booten des Systems (im Regelfall durch GRUB) verwendet werden. Bei den meisten Distributionen befindet sich hier auch der Kernel.
- `/dev` enthält alle Device-Dateien. Auf fast alle Hardware-Komponenten – etwa die serielle Schnittstelle oder eine Festplattenpartition – wird über sogenannte Device-Dateien zugegriffen. Diese werden dynamisch durch das `udev`-System eingerichtet (siehe [Abschnitt 11.9](#), »Device-Dateien«). Bei den meisten Distributionen befindet sich das `/dev`-Verzeichnis in einer RAM-Disk, d. h., der Inhalt des Verzeichnisses bleibt bei einem Neustart des Rechners nicht erhalten.
- `/etc` enthält Konfigurationsdateien für das ganze System. Innerhalb von `/etc` gibt es eine Menge Unterverzeichnisse, die die Konfigurationsdateien in Gruppen ordnen – z. B. `/etc/apt` für Dateien des Paketverwaltungssystems `apt`. Viele Dateien aus `/etc` sind in den Konfigurationskapiteln dieses Buchs beschrieben. Werfen Sie auch einen Blick in das Stichwortverzeichnis (Buchstabe E)!
- `/home` enthält die Heimatverzeichnisse aller regulären Linux-Anwender. Das Heimatverzeichnis ist jenes Verzeichnis, in dem sich der Anwender nach dem Einloggen automatisch befindet und auf dessen Dateien er uneingeschränkte Zugriffsrechte hat. Ein Sonderfall ist wie so oft `root`: Dessen Heimatverzeichnis lautet `/root`.
- `/lib[64]` enthält einige gemeinsame Bibliotheken (Shared Libraries) oder symbolische Links darauf. Die Dateien werden zur Ausführung von Programmen benötigt. `/lib/modules` enthält Kernelmodule, die im laufenden Betrieb dynamisch aktiviert bzw. deaktiviert werden. Weitere Bibliotheken befinden sich in `/usr/lib[64]`. Das Verzeichnis `/lib/firmware` enthält die Firmware diverser Hardware-Komponenten (z. B. WLAN-Controller).

Bei aktuellen Distributionen ist `/lib` ein Link auf `/usr/lib`. Damit werden alle Bibliotheken zentral im `/usr`-Verzeichnis abgelegt.

`/lost+found` gibt es nur in `ext`-Dateisystemen. Das Verzeichnis ist normalerweise leer. Enthält es doch Dateien, dann handelt es sich um Dateifragmente, die beim Versuch, das Dateisystem durch `fsck` zu reparieren, nicht mehr zugeordnet werden konnten. Mit anderen Worten: Es wurden Sektoren gefunden, aber es ist unklar, zu welcher Datei der Sektor einmal gehört hat. Anstatt derartige Dateifragmente einfach zu löschen, kopiert `fsck` diese in das `lost+found`-Verzeichnis.

`fsck` wird automatisch während des Systemstarts ausgeführt, wenn Linux nicht ordnungsgemäß beendet wurde (Stromausfall, Absturz etc.) oder wenn das Dateisystem längere Zeit nicht mehr überprüft wurde. Das Ziel von `fsck` ist es, das Dateisystem wieder in einen klar definierten Zustand zu bringen.

`/media` enthält Unterverzeichnisse wie `cdrom` oder `<usb-stick-name>`, an deren Stelle externe Dateisysteme eingebunden werden. Traditionell war hierfür `/mnt` üblich, in den vergangenen Jahren hat sich stattdessen zuerst `/media` und schließlich das `/run/media/<benutzername>/<datenträgername>` durchgesetzt.

`/opt` ist für Zusatzpakete vorgesehen, wird von den gängigen Distributionen aber nur selten genutzt – vermutlich deswegen, weil unklar ist, wie sich Zusatzpakete von normalen Paketen unterscheiden.

`/proc` enthält Unterverzeichnisse für alle laufenden Prozesse. Es handelt sich hierbei nicht um echte Dateien! Das `/proc`-Verzeichnis spiegelt lediglich die Linux-interne Verwaltung der Prozesse wider.

`/root` enthält die Dateien des Benutzers `root`, also des Systemadministrators.

`/run` enthält bei vielen aktuellen Distributionen Dateien mit den Prozess-IDs sowie weiteren Informationen von manchen Systemdiensten. In der Vergangenheit wurden diese Dateien im Verzeichnis `/var/run` gespeichert.

Das Unterverzeichnis `/run/lock/` enthält Locking-Dateien. Bei älteren Distributionen finden Sie die Locking-Dateien stattdessen in `/var/lock`.

Bei vielen Distributionen werden entweder das gesamte `/run`-Verzeichnis oder zumindest einzelne `/run`-Unterverzeichnisse in einer RAM-Disk abgelegt. Die überwiegend sehr kleinen Dateien in `/run` werden somit nie physikalisch auf einer Festplatte oder SSD gespeichert und gehen beim Neustart des Rechners verloren.

|                     |  |
|---------------------|--|
| <code>/sbin</code>  | enthält Kommandos zur Systemverwaltung. Ein gemeinsames Merkmal aller darin gespeicherten Programme ist, dass sie nur von <code>root</code> ausgeführt werden dürfen. Bei modernen Distributionen ist <code>/sbin</code> ein Link auf <code>/usr/sbin</code> ; alle Kommandos zur Systemverwaltung befinden sich nun in <code>/usr/sbin</code> .   |
| <code>/share</code> | enthält manchmal architekturunabhängige Dateien, also Dateien, die unabhängig vom Prozessor sind. Der korrekte Ort ist eigentlich <code>/usr/share</code> .  |
| <code>/srv</code>   | enthält bei einigen Distributionen (Fedora, RHEL) Daten für Server-Prozesse, z. B. in <code>/srv/www</code> Dateien des Webservers oder in <code>/srv/ftp</code> Dateien des FTP-Servers.  |
| <code>/sys</code>   | enthält das <code>sysfs</code> -Dateisystem. Es liefert wie das <code>proc</code> -Dateisystem Informationen über den Zustand des Rechners.  |
| <code>/tmp</code>   | enthält temporäre Dateien. Oft werden temporäre Dateien aber auch in <code>/var/tmp</code> gespeichert.  |
| <code>/usr</code>   | enthält alle Anwendungsprogramme, das komplette X-System, die Quellcodes zu Linux etc. Der Inhalt dieses Verzeichnisses ändert sich normalerweise nur bei Paketinstallationen und Updates. Für veränderliche Dateien ist das Verzeichnis <code>/var</code> vorgesehen. <a href="#">Tabelle 11.9</a> gibt aber eine kurze Beschreibung der wichtigsten Unterverzeichnisse von <code>/usr</code> .   |
| <code>/var</code>   | enthält veränderliche Dateien. Wichtige Unterverzeichnisse sind z. B. <code>docker</code> (Docker-Dateien), <code>lock</code> (Locking-Dateien zum Zugriffsschutz auf Devices), <code>log</code> (Logging-Dateien), <code>mail</code> (E-Mail-Dateien, oft auch in <code>/var/spool/mail</code> ), <code>mysql</code> (MySQL-Datenbankdateien), <code>run</code> (Dateien mit Prozess-IDs von manchen Systemdiensten) und <code>spool</code> (zwischen gespeicherte Druckdateien). |

Die grundsätzliche Struktur der Verzeichnisse auf Wurzelebene ist also recht gut zu verstehen. Die Probleme beginnen erst mit der Unterteilung von `/usr` und `/var` in zahlreiche Unterverzeichnisse. Prinzipiell werden dabei viele Verzeichnisse genauso benannt wie in der Wurzel-Ebene – etwa `bin` für ausführbare Programme.

Dabei tritt das Problem auf, dass es mehrere Gruppen ausführbarer Programme gibt: textorientierte Kommandos, X-Programme etc. In der Vergangenheit gab es für diese Programmgruppen einzelne Verzeichnisse, z. B. `/usr/bin/X11` für Programme mit grafischer Benutzeroberfläche. Mittlerweile bemühen sich die meisten Distributionen, möglichst alle Programme in *ein* Verzeichnis zu installieren, also nach `/usr/bin`. Symbolische Links stellen die Kompatibilität zu vergangenen Standards her.

| Verzeichnis               | Inhalt  |
|---------------------------|---|
| <code>/usr/bin</code>     | ausführbare Programme   |
| <code>/usr/games</code>   | Spiele; evtl. Link auf <code>/usr/share/games</code>  |
| <code>/usr/include</code> | C-Include-Dateien   |
| <code>/usr/lib[64]</code> | diverse Libraries, außerdem zahllose Unterverzeichnisse für C-Compiler, diverse andere Programmiersprachen, große Programmpakete wie emacs oder $\LaTeX$ etc. |
| <code>/usr/local</code>   | Anwendungen und Dateien, die nicht unmittelbar zur Linux-Distribution gehören oder später installiert wurden  |
| <code>/usr/sbin</code>    | nur von root ausführbare Programme  |
| <code>/usr/share</code>   | architekturunabhängige Daten (z. B. Emacs-Lisp-Dateien, Ghostscript-Zeichensätze etc.), Dokumentation ( <code>/usr/share/doc</code> )                         |
| <code>/usr/src</code>     | Quellcode zu Linux und eventuell zu anderen Programmen)   |

Tabelle 11.9 /usr-Verzeichnisse

## 11.9 Device-Dateien

Im Linux-Dateisystem werden nicht nur Dateien und Verzeichnisse verwaltet, sondern auch sogenannte Devices. Dabei handelt es sich um speziell gekennzeichnete Dateien, in denen keine Daten gespeichert werden, sondern die vielmehr eine Verbindung zum Linux-Kernel herstellen.

Devices ermöglichen den Zugriff auf viele Hardware-Komponenten des Rechners, also etwa auf Festplatten/SSDs, serielle und parallele Schnittstellen, den Arbeitsspeicher (RAM) etc. Devices sind durch drei Informationen charakterisiert: die Major Device Number, die Minor Device Number und den Typ des Zugriffs (block- oder zeichenorientiert).

**Major und Minor  
Device Number**

Die Major Device Number gibt an, welcher Treiber des Linux-Kernels für die Verwaltung zuständig ist. Die meisten Treiber sind mit ihrer Major Device Number auf der folgenden Seite aufgelistet:

<https://www.kernel.org/doc/Documentation/admin-guide/devices.txt>

Bei vielen Treibern dient die Minor Device Number zur Differenzierung zwischen verschiedenen (verwandten) Einzelgeräten – etwa beim Treiber für Festplatten zwischen unterschiedlichen Partitionen.

Der Zugriffstyp gibt an, ob die Geräte gepuffert sind (das ist bei allen blockorientierten Geräten wie Festplatten etc. der Fall) oder nicht (zeichenorientierte Geräte wie serielle oder parallele Schnittstellen).

Wenn Sie mit `ls -l` das Inhaltsverzeichnis von `/dev` betrachten, werden statt der Dateigröße die Device-Nummern (Major und Minor) ausgegeben. Das erste Zeichen der Zugriffsbits lautet `b` oder `c` (block- oder zeichenorientiert).

```
user$ ls -l /dev/sda?
brw-rw---- 1 root root 8, 1 ... /dev/sda1
brw-rw---- 1 root root 8, 2 ... /dev/sda2
...
```

**Interna** Linux-intern befinden sich im `/dev`-Verzeichnis nur sogenannte Inodes: Das sind die kleinsten Verwaltungseinheiten eines Dateisystems, aber keine richtigen Dateien mit Inhalt. Neue Device-Dateien können mit dem Kommando `mknod` eingerichtet werden. In der Praxis ist das aber selten notwendig, weil sich das `udev`-System automatisch darum kümmert. Die Major und Minor Device Number werden zu einer 64-Bit-Zahl zusammengesetzt.

Auf viele Devices dürfen aus Sicherheitsgründen nur `root` bzw. die Mitglieder einer bestimmten Gruppe zugreifen. Um auch anderen Benutzern Zugriff auf diese Devices zu ermöglichen, fügen Sie den Benutzer dieser Gruppe hinzu.

Einige Device-Dateien haben eine besondere Funktion: So dient `/dev/null` als »schwarzes Loch«, an das Daten gesendet werden können, die dort für immer verschwinden – etwa zur Umleitung von Kommandoausgaben, die nicht angezeigt werden sollen. `/dev/zero` ist eine unerschöpfliche Quelle von 0-Bytes, die manchmal dazu verwendet wird, Dateien bis zu einer vorgegebenen Größe mit Nullen zu füllen. `/dev/random` und `/dev/urandom` liefern zufällige Zahlen.

**udev-System** In der Vergangenheit erzeugten Distributionen während der Installation Tausende von Device-Dateien. Tatsächlich genutzt werden höchstens ein paar Hundert Dateien; nur sind es auf jedem Rechner – je nach Hardware-Ausstattung – unterschiedliche Device-Dateien.

Abhilfe schafft das `udev`-System. Das Hintergrundprogramm `udev` bzw. bei aktuellen Distributionen `systemd-udev` erkennt alle mit dem Rechner verbundenen Hardware-Komponenten und erzeugt die erforderlichen Device-Dateien nach Bedarf. `udev` bzw. `systemd-udev` wird durch den Init-Prozess gestartet. Die Konfiguration erfolgt durch die Dateien des Verzeichnisses `/etc/udev`.

Im Zuge der Bemühungen, Linux schneller zu starten, wurde `udev` um das `devtmpfs`-Dateisystem ergänzt. Dieses temporäre Dateisystem bildet das `/dev`-Verzeichnis ab. Während des Bootprozesses kann `devtmpfs` ohne den Overhead des vollständigen `udev`-Systems genutzt werden.



| Device       | Bedeutung   |
|--------------|---|
| /dev/cdrom   | Link auf das CD-ROM-Device                                |
| /dev/console | das gerade aktive virtuelle Terminal                      |
| /dev/disk/*  | zusätzliche Links auf Festplatten- und Partitions-Devices |
| /dev/dri/*   | Direct Rendering Infrastructure (3D-Grafik mit X)         |
| /dev/dsp*    | Zugang zur Soundkarte (Digital Sampling Device)           |
| /dev/fb*     | Frame Buffer (Grafikkarte)                                |
| /dev/input/* | Maus  |
| /dev/kbd     | Tastatur (PS/2)   |
| /dev/kmem    | Speicher (RAM) im Core-Format (für Debugger)              |
| /dev/mapper  | Mapping-Dateien für LVM, Krypto-Container etc.            |
| /dev/md*     | Meta-Devices (RAID etc.)                                  |
| /dev/mem     | Speicher (RAM)  |
| /dev/mixer*  | Zugang zur Soundkarte                                     |
| /dev/port    | IO-Ports  |
| /dev/pts/*   | virtuelle Terminals gemäß Unix 98                         |
| /dev/ptyp*   | virtuelle Terminals unter X (Master)                      |
| /dev/ram     | RAM-Disk  |
| /dev/raw1394 | direkter Zugriff auf Firewire-Geräte                      |
| /dev/sd*     | SCSI/SATA/USB/Firewire-Festplatten                        |
| /dev/scd*    | SCSI/SATA/USB/Firewire-CD/DVD-Laufwerke                   |
| /dev/shm     | POSIX Shared Memory                                       |
| /dev/snd     | ALSA-Sound (Link auf /proc/asound/dev)                    |
| /dev/st*     | SCSI/SATA/USB/Firewire-CD/DVD-Laufwerke                   |
| /dev/tty*    | virtuelle Terminals im Textmodus                          |
| /dev/ttyp*   | virtuelle Terminals unter X (Slave)                       |
| /dev/ttyS*   | serielle Schnittstellen (Modem, Maus etc.)                |
| /dev/usb/*   | USB-Geräte (siehe auch /proc/bus/usb)                     |

Tabelle 11.10 Wichtige Device-Dateien



# Kapitel 12

## Prozessverwaltung

Dieses Kapitel beschreibt, wie Linux mit Prozessen umgeht. Im Verlauf dieses Kapitels lernen Sie,

- ▶ welche Möglichkeiten es gibt, Programme zu starten und wieder zu beenden (zur Not auch gewaltsam),
- ▶ wie Sie ein Programm mit root-Rechten ausführen,
- ▶ was Dämonen sind und
- ▶ wie Sie Programme zu bestimmten Zeiten durch Cron oder systemd automatisch starten können.

### 12.1 Prozesse starten, verwalten und stoppen

In diesem Kapitel ist überwiegend von Prozessen die Rede. Ein Prozess ist auf Betriebssystemebene für die Ausführung eines Programms oder Kommandos verantwortlich. Das klingt nach einer eher trivialen Aufgabe; da aber eine Menge Programme und Hintergrunddienste parallel laufen, ist es gar nicht so einfach, die Rechenzeit zwischen allen Programmen gerecht bzw. sinnvoll zu verteilen.

Programme,  
Kommandos,  
Prozesse, Tasks

#### Programme und Kommandos

Ein Programm bzw. ein Kommando ist eigentlich nur eine ausführbare Datei. Eine Programmdatei unterscheidet sich von anderen Dateien also dadurch, dass das Execute-Bit `x` gesetzt ist.

Linux-intern gibt es keine Unterscheidung zwischen einem Programm wie Firefox oder einem Kommando wie `ls`. Umgangssprachlich werden textorientierte Programme wie `ls` aber oft als Kommandos bezeichnet.

Erst durch den Start einer gleichsam leblosen Programmdatei wird diese zu einem lebendigen Prozess, der vom Linux-Kernel verwaltet wird. So gesehen müsste die Überschrift dieses Abschnitts eigentlich lauten: *Programme und Kommandos starten, Prozesse verwalten und stoppen*.

**\*.exe-Dateien** Hin und wieder taucht die Frage auf, wo denn unter Linux die \*.exe-Dateien sind. Bis vor einigen Jahren hieß die richtige Antwort: Es gibt keine \*.exe-Dateien. Ausführbare Programme sind durch das Zugriffsbit `x` gekennzeichnet; die von Windows bekannte Dateikennung \*.exe ist somit überflüssig.

Mittlerweile ist diese Antwort insofern nicht mehr ganz richtig, als es unter Linux tatsächlich vereinzelt \*.exe-Dateien geben kann. Dabei handelt es sich um Programme, die in der Programmiersprache C# entwickelt wurden und die zur Ausführung auf die Mono-Bibliothek zurückgreifen. Die Mono-Bibliothek ist wiederum eine Open-Source-Implementierung des .NET Frameworks von Microsoft.

### Programme starten

**Programmstart unter X** Im Grafikmodus starten Sie Programme im Regelfall über ein Menü oder durch das Anklicken eines Icons. Desktop-Systeme wie KDE, Gnome oder Unity bieten mit den Tastenkürzeln `[Alt]+[F2]` oder `[⌘]` eine zusätzliche Möglichkeit, Programme rasch zu starten.

**Textkonsole, Terminalfenster** Alternativ können Sie Programme auch in einem Terminalfenster oder in einer Textkonsole starten. Dazu geben Sie einfach den Namen des Programms ein und drücken `[↵]`. Gerade Linux-Profis wählen oft diesen Weg, weil es schneller geht, ein paar Buchstaben einzutippen, als das Programm in verzweigten Menüs zu suchen.

Normalerweise reicht es aus, wenn Sie einfach den Namen des Programms angeben. Der Shell-Interpreter sucht das Programm dann in allen Verzeichnissen, die in der Umgebungsvariablen `PATH` angegeben sind. Die folgenden Zeilen zeigen eine typische Einstellung dieser Variablen:

```
user$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:
/usr/local/games:/snap/bin
```

Wenn Sie ein Programm starten möchten, das sich in keinem dieser Verzeichnisse befindet, müssen Sie den vollständigen Pfad angeben. Das gilt auch für Programme im gerade aktuellen Verzeichnis! Hier wird der Pfad einfach durch einen Punkt angegeben, also beispielsweise `./meinprogramm`.

### Vordergrund- und Hintergrundprozesse

Wenn Sie im Startmenü Ihres Desktops Programme starten, laufen diese selbstverständlich als sogenannte Hintergrundprozesse, also ohne sich gegenseitig zu behindern. Sie können weitere Programme starten, ohne auf das Ende der bisher gestarteten Programme warten zu müssen.

Ganz anders ist das Verhalten, wenn Sie ein Programm in einer Textkonsole bzw. einem Terminal ausführen. Das Programm wird als Vordergrundprozess gestartet. Bevor Sie im Terminal das nächste Kommando eingeben können, müssen Sie auf das Ende des zuletzt gestarteten Programms warten.

Aber auch in Textkonsolen oder Terminalfenstern können Sie Programme im Hintergrund starten. Dazu geben Sie einfach am Ende des Kommandos das Zeichen `&` an:

```
user$ firefox &
```

Wenn Sie `&` vergessen haben, können Sie das Programm auch nachträglich in einen Hintergrundprozess umwandeln. Unterbrechen Sie die Programmausführung mit `[Strg]+[Z]`, und setzen Sie das Programm mit `fg` fort:

Vom Vordergrund  
in den  
Hintergrund

```
user$ firefox
<Strg>+<Z>
+ Stopped firefox
user$ bg
+ firefox &
```

Wenn Sie statt `bg` das Kommando `fg` verwenden, wird das Programm als Vordergrundprozess fortgesetzt.

Bei manchen Kommandos stören diverse Textausgaben bei der Hintergrundauführung. Diese können Sie aber leicht unterdrücken, indem Sie sie nach `/dev/null` umleiten. Beispielsweise wird durch das folgende Kommando ein Dateisystem im Hintergrund eingerichtet:

```
root# mkfs.ext4 /dev/sdc1 > /dev/null &
```

### Liste aller laufenden Prozesse (`ps`, `top`)

Eine Liste der zurzeit laufenden Prozesse können Sie sehr einfach mit `ps` erzeugen. Ohne Optionen zeigt `ps` nur Ihre eigenen Prozesse an – und nur solche, die aus Textkonsolen bzw. Shell-Fenstern gestartet wurden. `ps` kann durch zahlreiche Optionen gesteuert werden, wobei viele Optionen ohne das sonst übliche vorangestellte Minuszeichen angegeben werden. Wenn der Prozessname in eckigen Klammern steht, handelt es sich um einen Prozess des Kernels. Im folgenden Beispiel wurde die Liste der Prozesse aus Platzgründen stark gekürzt. Auf einem typischen Linux-System mit grafischer Benutzeroberfläche laufen normalerweise deutlich mehr als 100 Prozesse zugleich.

```
user$ ps ax
  PID TTY          STAT TIME  COMMAND
    1 ?           Ss    0:00  init [2]
    2 ?           S      0:00  [kthreadd]
    3 ?           S      0:00  [ksoftirqd/0]
```

```

...
3064 pts/2    S      0:39 emacs command.tex
3151 pts/2    S+    1:23 /bin/sh ./lvauto
3735 pts/4    S      0:00 su -l
3740 pts/4    S+    0:00 -bash

```

- top** Praktischer als `ps` ist oft `top`: Dieses Kommando ordnet die Prozesse danach, wie sehr sie die CPU belasten, und zeigt die gerade aktiven Prozesse zuerst an. Das Programm gibt auch einen Überblick über den aktuellen Speicherbedarf etc. Die Prozessliste wird alle paar Sekunden aktualisiert, bis das Programm mit `q` beendet wird. Die folgenden Zeilen zeigen einen Webserver nahezu im Leerlauf:

```

top - 20:50:38 up 11 days, 12:18,  1 user,  load average: 0.10, 0.09, 0.08
Tasks: 114 total,  1 running, 113 sleeping,  0 stopped,  0 zombie
Cpu(s):  2.6%us,  0.2%sy,  0.0%ni, 96.8%id,  0.4%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   4049808k total, 1580060k used, 2469748k free,  203396k buffers
Swap:   521212k total,    0k used,  521212k free,  804400k cached

```

| PID   | USER     | PR | NI | VIRT  | RES  | SHR  | S | %CPU | %MEM | TIME+   | COMMAND |
|-------|----------|----|----|-------|------|------|---|------|------|---------|---------|
| 32601 | www-data | 20 | 0  | 346m  | 32m  | 4296 | S | 5    | 0.8  | 0:00.15 | apache2 |
| 32592 | www-data | 20 | 0  | 344m  | 31m  | 4324 | S | 3    | 0.8  | 0:00.47 | apache2 |
| 851   | mysql    | 20 | 0  | 1403m | 53m  | 7948 | S | 0    | 1.4  | 6:40.10 | mysqld  |
| 1     | root     | 20 | 0  | 24336 | 2184 | 1272 | S | 0    | 0.1  | 0:01.07 | init    |

...

Der Wert in der PID-Spalte gibt die Prozessnummer an. Wenn Sie diese Nummer kennen, können Sie außer Kontrolle geratene Programme oder Hintergrundprozesse mit dem Kommando `kill` gewaltsam stoppen.

Prozesse können verschiedene Zustände annehmen. Die zwei häufigsten Zustände sind R (*running*) und S (*sleeping*, das Programm hat also gerade nichts zu tun und wartet auf Eingaben). Programme können auch vorübergehend unterbrochen werden und weisen dann den Zustand T (*stopped*) auf.

`top` nimmt auch interaktiv Kommandos entgegen. Damit können Sie Prozesse stoppen (`K`, `kill`) oder ihre Priorität verändern (`R`, `renice`).

- htop** Eine wesentlich komfortablere Alternative zu `top` ist das Kommando `htop`, das bei den meisten Distributionen separat installiert werden muss. Es erlaubt unter anderem ein horizontales und vertikales Scrollen in der Prozessliste.
- iotop** Wenn Sie nicht die CPU- und Speicherauslastung, sondern die Zugriffe auf Festplatten und andere Datenträger verfolgen möchten, starten Sie statt `top` das Kommando `iotop`. Mit der Option `-o` schränken Sie die Ausgabe auf Prozesse ein, die tatsächlich IO-Aktivität verursachen. `-u` beschränkt die Ausgabe auf eigene Prozesse. `iotop` ist Bestandteil des gleichnamigen Pakets, das in der Regel extra installiert werden muss.

Wenn Sie den Programmnamen wissen und die dazugehörige Prozessnummer (PID) ermitteln möchten, hilft `pidof`. Wenn es mehrere Prozesse mit dem gleichen Namen gibt, liefert `pidof` eine ganze Liste von Nummern:

Prozessnummer  
ermitteln

```
root# pidof nscd
1777 1776 1775 1774 1765 1763 1753
```

Manchmal ist es auch nützlich, festzustellen, welche Programme auf eine bestimmte Datei oder ein Verzeichnis zugreifen. Die entsprechenden Prozessnummern können Sie mit `fuser` feststellen. Ein Verzeichnis gilt auch dann als benutzt, wenn darin ein Programm gestartet wurde. Das folgende Kommando zeigt, dass die Shell `bash` das Verzeichnis `/media/dvd` nutzt:

```
root# fuser -v /media/dvd
          USER      PID ACCESS COMMAND
/media/dvd  kofler    2183 ..c..  bash
          root      Kernel mount  /media/dvd
```

Beachten Sie, dass ein Dateizugriff nur festgestellt werden kann, wenn das Programm die Datei wirklich geöffnet hat. Das ist bei einem Texteditor beispielsweise nicht der Fall: Der Editor hat die Datei zum Laden geöffnet, dann aber wieder geschlossen. Zum Speichern öffnet er die Datei wiederum nur für kurze Zeit.

Manche Hintergrundprozesse speichern im Verzeichnis `/var/run` eine PID-Datei, z. B. `/var/run/httpd.pid`. Diese Datei enthält in der ersten Zeile die Prozessnummer; weitere Zeilen können Zusatzinformationen enthalten, z. B. die Netzwerkschnittstelle. PID-Dateien ermöglichen das gezielte Beenden eines bestimmten Prozesses durch das Init-System, und zwar auch dann, wenn es mehrere gleichnamige Prozesse gibt.

PID-Dateien

Zu den Kommandos `top` und `htop` gibt es auch grafische Alternativen, z. B. `ksysguard` (KDE) oder `gnome-system-monitor` (Gnome). Ungleich mehr Darstellungs- und Konfigurationsmöglichkeiten bietet `Conky`. Im Internet finden Sie `Conky`-Konfigurationen, die wahre Kunstwerke sind und aus einem faden Desktop-Hintergrund eine originelle und nützliche Informationszentrale machen. Leider ist das manuelle Einrichten einer `Conky`-Konfiguration mit viel Arbeit verbunden. Viele der im Internet angebotenen Konfigurationsdateien sind veraltet oder passen nur zu einer speziellen Distribution.

Grafische  
Varianten zu `top`

Wenn Sie nicht den eigenen Rechner überwachen möchten, sondern mehrere externe Server, greifen Sie am besten auf Tools wie `Nagios` oder `Munin` zurück. Deren Konfiguration ist aber aufwendig und erfordert Routine bei der Server-Administration.

Server-  
Überwachung

Eine einfachere Alternative für einen einzelnen Server ist das Projekt *Linux Dash*. Nach der unkomplizierten Installation fasst eine im Webbrowser abrufbare Seite die wichtigsten Statusinformationen zusammen. Eine gute Einführung gibt dieser Heise-Artikel:

<https://heise.de/-2407522>

## Prozeshierarchie

Intern wird mit jedem Prozess auch die PID-Nummer des Elternprozesses gespeichert. Diese Information ermöglicht die Darstellung eines Prozessbaums, an dessen Spitze immer der Init-Prozess steht. Das ist das erste Programm, das unmittelbar nach dem Laden des Kernels gestartet wird (siehe [Kapitel 24](#), »Das Init-System«). In [Abbildung 12.1](#) ist das das Programm `systemd`, d. h., die Abbildung wurde unter einer Distribution erstellt, die `systemd` als Init-System verwendet.



Abbildung 12.1 Prozessübersicht mit `pstree`

Das Kommando `pstree`, das mitunter extra installiert werden muss, zeigt die Hierarchie des Prozessbaums. Mit der Option `-h` werden die Elternprozesse zum gerade laufenden Prozess fett hervorgehoben. In [Abbildung 12.1](#) ist das aber nicht zu sehen: Die so gekennzeichneten Prozesse folgen erst weiter unten, ich wollte in der Abbildung aber die Spitze des Prozessbaums zeigen.



## Prozesse gewaltsam beenden (kill, xkill)

Normalerweise läuft ein Prozess bis zum regulären Programmende. Aber leider kommt es auch unter Linux vor, dass Programme Fehler enthalten, sich nicht mehr stoppen lassen und womöglich immer mehr Speicher und CPU-Kapazität beanspruchen. In solchen Fällen muss der Prozess gewaltsam beendet werden. Bei textorientierten Kommandos hilft in den meisten Fällen einfach `[Strg]+[C]`. Damit wird das Programm sofort beendet.

Das Kommando `kill` versendet Signale an einen laufenden Prozess, der durch die PID-Nummer spezifiziert wird. (Diese Nummer können Sie mit `top` oder `ps` ermitteln.) Um ein Programm »höflich« zu beenden, wird das Signal 15 verwendet. (`kill` verwendet dieses Signal per Default.) Hilft das nicht, muss das Signal 9 eingesetzt werden (hier für den Prozess 2725):

```
user$ kill -9 2725
```

`kill` kann nur für eigene Prozesse verwendet werden. Nur `root` darf auch fremde Prozesse beenden.

Auch mit `top` können Sie Prozesse beenden: Geben Sie einfach `[K]` und anschließend die Prozessnummer und das gewünschte Signal ein! `top`

`killall` ist insofern bequemer, als keine Prozessnummer, sondern der Programmname angegeben werden kann. Allerdings werden nun *alle* Prozesse dieses Namens beendet. `killall`

```
root# killall -9 firefox
```

Unter X geht es noch bequemer. Starten Sie in einem Shell-Fenster `xkill`, und klicken Sie einfach das Fenster des Programms an, das Sie beenden wollen. An den Prozess wird wiederum das Signal 9 gesandt. `xkill` funktioniert nur, wenn Sie X als Grafiksystem verwenden, ist also nicht Wayland-kompatibel. `xkill`

Unter KDE können Sie `xkill` auch mit `[Strg]+[Alt]+[Esc]` starten. Wenn das irrtümlich passiert, können Sie `xkill` mit `[Esc]` abbrechen.

Manchmal wird durch `xkill` zwar das Fenster geschlossen, der Prozess oder Teile davon laufen aber weiter. Vergewissern Sie sich mit `top` bzw. mit `ps`, dass das Programm wirklich beendet ist. Zur Not müssen Sie mit `kill -9 <pid>` nachhelfen. Hartnäckige Fälle

Wirklich unangenehm wird es, wenn ein Desktop-Programm nicht nur hängen bleibt, sondern dabei auch den Tastatur- und Maus-Fokus an sich reißt oder das Grafiksystem sonstwie blockiert. Der Rechner reagiert dann auf keine Eingaben mehr. In solchen Fällen hilft manchmal die magische Tastenkombination `[Strg]+[Alt]+[Fn]` weiter, mit der der Wechsel in die Textkonsole `n` erfolgt. Dort können Sie sich einloggen und das betreffende Programm mit `top` suchen und beenden. Blockierte Tastatur oder Maus

Wenn die Tastatur vollständig blockiert ist, besteht immer noch die Möglichkeit, sich über ein Netzwerk via `ssh` einzuloggen und `kill` auf diese Weise auszuführen. Diese Variante ist natürlich nur möglich, wenn Sie in einem lokalen Netz arbeiten und auf dem lokalen Rechner `sshd` läuft.

**Prozessgröße beschränken** Bei Programmen, die über eine Shell gestartet werden (etwa bei allen Kommandos, die in einem Shell-Fenster ausgeführt werden), können Sie mit dem Shell-Kommando `ulimit` den maximalen Speicherverbrauch, die maximale Größe erzeugter Dateien etc. begrenzen. `ulimit` wird üblicherweise in `/etc/profile` eingestellt.

### Verteilung der Rechenzeit (`nice`, `renice`, `ionice`)

Im alltäglichen Betrieb von Linux ist die Rechenkapazität meist mehr als ausreichend, um alle laufenden Prozesse ohne Verzögerungen auszuführen. Wenn Linux aber gerade mit rechenaufwendigen Prozessen beschäftigt ist – z. B. während des Kompilierens eines umfangreichen Programms –, versucht es, die zur Verfügung stehende Rechenzeit gerecht an alle Prozesse zu verteilen.

In manchen Fällen ist es sinnvoll, einem Prozess bewusst mehr oder weniger Rechenzeit zuzuteilen. Dazu dient das Kommando `nice`, mit dem Programme mit reduzierter oder erhöhter Priorität gestartet werden können. Dazu wird an `nice` die gewünschte Priorität übergeben, die von 19 (ganz niedrig) bis -20 (ganz hoch) reicht. Per Default werden Prozesse mit der Priorität 0 gestartet. Im folgenden Beispiel wird ein Backup-Programm mit niedrigerer Priorität gestartet, damit es keine anderen Prozesse beeinträchtigt. (Es ist ja egal, ob das Backup ein paar Sekunden länger dauert.)

```
user$ nice -n 10 ./my-backup-script
```

**renice** Mit `renice` kann auch die Priorität von bereits laufenden Prozessen geändert werden. Als Parameter muss die Prozess-ID angegeben werden, die vorher mit `top` oder `ps` ermittelt wurde. Details zu `renice` finden Sie auf der `man`-Seite. Auch `top` ist in der Lage, interaktiv die Priorität eines Prozesses zu verändern; dazu drücken Sie einfach `R`. Allerdings kann nur `root` Programme mit einer höheren Priorität als 0 starten bzw. die Priorität eines bereits laufenden Prozesses erhöhen.

**ionice** Oft ist nicht die CPU, sondern der Datenträger der limitierende Faktor bei der Ausführung von Programmen. Wenn Sie vermeiden möchten, dass beispielsweise ein Backup-Script die gesamte I/O-Kapazität des Rechners für sich beansprucht und damit andere, vielleicht zeitkritischere Prozesse bremst, können Sie es mit `ionice` mit reduzierter I/O-Priorität ausführen. Das folgende Kommando liest ein Logical Volume aus, komprimiert seinen Inhalt und speichert ihn in einer Image-Datei:

```
root# ionice -c 3 cat /dev/vg1/snap | lzop -c > /backup/image.lzo
```

## Ein- und Ausgabeumleitung, Pipe

Fast alle textorientierten Kommandos erwarten Eingaben über den sogenannten Standardeingabekanal (per Default die Tastatur) und senden Ausgaben an den Standardausgabekanal; in einem Terminal wird der resultierende Text einfach angezeigt. Sowohl die Ein- als auch die Ausgabe lassen sich umleiten, wodurch sich viele Möglichkeiten ergeben. Beispielsweise speichert das folgende Kommando die Liste aller Dateien des Verzeichnisses `xy` in der Datei `z`:

```
user$ ls xy > z
```

Durch sogenannte Pipes kann die Ausgabe eines Kommandos als Eingabe für das nächste Kommando verwendet werden. Beim folgenden Beispiel filtert `egrep` aus der Liste aller installierten Pakete diejenigen heraus, die die Zeichenketten »mysql« oder »mariadb« in beliebiger Groß- und Kleinschreibung enthalten. `sort` sortiert diese Liste schließlich.

Mit anderen Worten: Die Ausgaben des Kommandos `rpm` werden dank des ersten `|`-Zeichens an `grep` weitergeleitet, dessen Ausgaben mit dem dritten `|`-Zeichen an `sort`. (Das zweite `|`-Zeichen wird von `egrep` im Sinne von *oder* interpretiert.) Mehr Details und Beispiele zur Ein- und Ausgabeumleitung finden Sie in [Abschnitt 10.4](#), »Ein- und Ausgabeumleitung«.

```
user$ rpm -qa | egrep -i 'mysql|mariadb' | sort
mariadb-5.5.41-2.el7_0.x86_64
mariadb-libs-5.5.41-2.el7_0.x86_64
mariadb-server-5.5.41-2.el7_0.x86_64
php-mysql-5.4.16-36.el7_1.x86_64
```

## 12.2 Prozesse unter einer anderen Identität ausführen (su)

Bei der Programmausführung durch gewöhnliche Benutzer gibt es zwei Einschränkungen:

- ▶ Gewöhnliche Benutzer dürfen nur die Prozesse ausführen, bei denen die Zugriffsrechte (Besitzer, Gruppe, `r`- und `x`-Zugriffsbits) dies zulassen. Bei gewöhnlichen Programmen ist das keine Einschränkung. Es gibt aber beispielsweise im Verzeichnis `/usr/sbin` einige Kommandos zur Systemadministration, die nur von `root` gestartet werden können.
- ▶ Prozesse gehören gleichsam dem Benutzer, der sie gestartet hat. Das bedeutet, dass der Prozess auf die gleichen Dateien zugreifen darf wie der Benutzer. Umgekehrt formuliert: Dateien, die Sie als Benutzer nicht verändern dürfen, dürfen auch nicht von Programmen verändert werden, die Sie starten. Vom Prozess neu erzeugte Dateien gehören ebenfalls dem Benutzer, der das Programm gestartet hat (siehe auch [Abschnitt 11.6](#), »Spezialbits und die `umask`-Einstellung«).

Als gewöhnlicher Benutzer können Sie aus diesen Gründen viele administrative Arbeiten nicht durchführen. Die offensichtlich einfachste Lösung besteht darin, sich als `root` einzuloggen. Ich habe in diesem Buch aber schon mehrfach darauf hingewiesen, dass es keine gute Idee ist, ständig als `root` zu arbeiten: Die Gefahr ist einfach zu groß, dass Sie irrtümlich Schaden anrichten. Aus diesem Grund sperren manche Distributionen den `root`-Login vollständig. So ist unter Ubuntu ein direkter `root`-Login unmöglich.

Dieser Abschnitt beschreibt, wie Sie mit `su` bzw. `ssh` dennoch administrative Tätigkeiten durchführen können, ohne sich als gewöhnlicher Benutzer auszuloggen. [Abschnitt 12.3](#) zeigt eine alternative Vorgehensweise mit `sudo`, die sich vor allem unter Ubuntu bewährt hat. [Abschnitt 12.4](#) präsentiert schließlich das Programm `PolicyKit`, das ganz neue Wege bei der Ausführung von `root`-Aufgaben geht.

### Das `su`-Kommando

In vielen Fällen geht es nur darum, rasch ein Kommando als `root` auszuführen. Da wäre es unpraktisch, das gerade aktive Desktop-System bzw. Terminalfenster zu verlassen und sich neu als `root` einzuloggen.

Die einfachste Möglichkeit, innerhalb eines Terminalfensters den Benutzer zu ändern, bietet das Kommando `su name`. Wenn Sie das Kommando nicht als `root` ausführen, werden Sie nach dem Passwort des jeweiligen Anwenders gefragt. Innerhalb des Terminals können Sie jetzt Kommandos unter dem geänderten Namen ausführen, bis Sie durch `exit` oder `[Strg]+[D]` zurück in den Normalmodus wechseln.

Die folgenden Zeilen zeigen, wie ein gewöhnlicher Benutzer sich kurz als `root` anmeldet, als `root` eine Festplattenpartition in den Verzeichnisbaum einbindet und sich dann als `root` wieder ausloggt und normal weiterarbeitet:

```
user$ su -l root
Password: *****
root# mount -t ext2 /test /dev/sda7
root# <Strg>+<D>
logout
user$ ls /test
```

Damit `su` ein vollwertiger Ersatz für einen `root`-Login ist, müssen Sie die Option `-l` verwenden! Damit erreichen Sie, dass alle Login-Startdateien eingelesen werden, was unter anderem zur korrekten Definition von `PATH` notwendig ist.

**pkexec** Je nach Konfiguration und Grafiksystem (Wayland) können Desktop-Programme nicht in einer `su`-Session ausgeführt werden. Vielmehr greifen die meisten Distributionen auf das Kommando `pkexec` zurück, um in der Desktop-Umgebung Programme

mit Administratorrechten zu starten. Hinter den Kulissen greift `pkexec` auf PolicyKit zurück (siehe [Abschnitt 12.4](#)).

Die `Setuid`- und `Setgid`-Zugriffsbits stellen eine weitere Möglichkeit dar, bestimmte Programme so zu kennzeichnen, dass jeder sie ausführen kann, als wäre er bzw. sie `root` oder ein anderer Benutzer oder Mitglied einer anderen Gruppe. Damit entfällt aber ein wichtiger Schutzmechanismus: Das `Setuid`-Bit gilt für *alle* Benutzer, ganz egal, ob diese das `root`-Passwort kennen (für `su`) oder einer `sudo`-Gruppe angehören. Weitere Informationen zu den `Setuid`- und `Setgid`-Zugriffsbits finden Sie in [Abschnitt 11.6](#), »Spezialbits und die `umask`-Einstellung«.

Setuid- und  
Setgid-  
Zugriffsbits

## 12.3 Prozesse unter einer anderen Identität ausführen (sudo)

`sudo` verfolgt einen ganz anderen Ansatz als die oben beschriebenen `su`-Varianten. Das Programm ermöglicht nach entsprechender Konfiguration bestimmten Benutzern die Ausführung bestimmter Programme mit `root`-Rechten. Zur Sicherheit muss nochmals das *eigene* Passwort angegeben werden, also eben *nicht* das `root`-Passwort.

`sudo` führt diese Programme dann so aus, als wären sie von einem anderen Benutzer gestartet worden (Default: `root`). Damit können einzelne Benutzer administrative Aufgaben übernehmen bzw. systemkritische Kommandos ausführen, ohne dazu das `root`-Passwort kennen zu müssen. `sudo` protokolliert alle ausgeführten Kommandos sowie gescheiterte Versuche üblicherweise in `/var/log/messages`.

Je nach Konfiguration bleibt die `sudo`-Authentifizierung für einige Minuten gültig. Wenn Sie innerhalb dieser Zeit ein weiteres Kommando mit `sudo` ausführen, werden Sie nicht neuerlich nach dem Passwort gefragt. Die Zeitspanne kann in `/etc/sudoers` mit dem Schlüsselwort `timestamp_timeout` verändert werden.

Die Konfiguration von `sudo` erfolgt durch die Datei `/etc/sudoers`. Vereinfacht ausgedrückt, beschreibt die Datei in drei Spalten, welche Benutzer von welchem Rechner aus welche Programme ausführen dürfen. Die folgende Zeile bedeutet, dass die Benutzerin `kathrin` am Rechner `uranus` das Kommando `/sbin/fdisk` ausführen darf. Das Schlüsselwort `ALL` bedeutet, dass `kathrin` das Kommando unter jedem beliebigen Account ausführen darf, also als `root`, als `news`, als `lp` etc.

Konfiguration

```
# in /etc/sudoers
kathrin uranus=(ALL) /sbin/fdisk
```

Wenn der ersten Spalte von `sudoers` das Zeichen `%` vorangestellt wird, gilt der Eintrag für alle Mitglieder der angegebenen Gruppe. Diverse weitere Syntaxvarianten beschreibt man `sudoers`.

### Ändern Sie `/etc/sudoers` mit `visudo`!

Aus Sicherheitsgründen sollte `/etc/sudoers` ausschließlich mit dem Kommando `visudo` editiert werden (siehe auch dessen `man`-Seite)! `visudo` führt vor dem Speichern einen Syntaxtest durch und stellt so sicher, dass Sie sich nicht durch eine fehlerhafte `sudoers`-Datei selbst von weiteren Administrationsarbeiten ausschließen. Besonders wichtig ist das bei Distributionen wie Ubuntu, die keinen `root`-Login vorsehen.

`visudo` verwendet, wie der Name vermuten lässt, normalerweise den Editor `vi`; wenn Sie einen anderen Editor vorziehen, müssen Sie dessen exakten Pfad vor der Ausführung von `visudo` in der Umgebungsvariablen `EDITOR` angeben – also z. B. durch `export EDITOR=/usr/bin/jmacs`.

Die Konfiguration von `/etc/sudoers` bietet viel mehr syntaktische Möglichkeiten, als hier angedeutet wurde. Lesen Sie die `man`-Seiten zu `sudo` und zu `sudoers`! Noch mehr Details sind auf der `sudo`-Homepage nachzulesen:

<https://sudo.ws>

**Anwendung** Kathrin kann nun `fdisk` folgendermaßen ausführen:

```
kathrin$ sudo /sbin/fdisk /dev/sda
Password: xxxxxx
```

Als Passwort muss das Passwort der Benutzerin `kathrin` angegeben werden. Bei `fdisk` muss der vollständige Pfad angegeben werden, falls sich `fdisk` nicht in einem der `PATH`-Verzeichnisse von `kathrin` befindet. `fdisk` wird automatisch im Account `root` ausgeführt. Ein anderer Account kann mit `sudo -u account` gewählt werden.

**sudo ohne Passwort** Es besteht die Möglichkeit, einem bestimmten Benutzer das Ausführen von `sudo` ohne Passwortangabe zu erlauben. Dazu fügen Sie in `sudoers` eine Zeile nach dem folgenden Muster ein:

```
kofler ALL=(ALL) NOPASSWD: ALL
```

Das ist natürlich ein Sicherheitsrisiko, aber wer oft Administratoraufgaben ausführen muss, wird die so gewonnene Bequemlichkeit schätzen. Beachten Sie, dass das `NOPASSWD`-Tag nur gültig ist, wenn es keine anderen `sudoers`-Zeilen gibt, die vom selben Benutzer ein Passwort verlangen. Das gilt auch für Gruppeneinträge, also z. B. `%admin`.

**Ein-/Ausgabeumleitung** Wenn Sie in einem mit `sudo` ausgeführten Kommando Ein- und Ausgabeumleitung verwenden, wird diese direkt von der Shell ausgeführt, nicht von `sudo` – und somit nur mit Ihren lokalen Rechten, nicht mit `root`-Rechten. Das führt zu Fehlern:

```
user$ sudo ls /etc > /etc/ls-out.txt
-bash: /etc/ls-out.txt: Keine Berechtigung
```

Dieses Problem können Sie umgehen, indem Sie an `sudo` eine Shell übergeben und in dieser das gewünschte Kommando ausführen:

```
user$ sudo sh -c 'ls /etc > /etc/ls-out.txt'
```

Mitunter kommt es vor, dass Sie mehrere Kommandos automatisiert durch ein Script per `sudo` ausführen möchten. Die korrekte Syntax sieht dann so aus:

Shell-Scripts mit  
sudo ausführen

```
user$ sudo bash script.sh
user$ sudo ./script.sh      (wenn das Script mit Shebang beginnt)
```

Bei umfangreicheren Administrationsaufgaben wird es zunehmend lästig, jedem Kommando `sudo` voranzustellen. Eleganter ist es, mit `sudo -s` in den `root`-Modus zu wechseln. Alle weiteren Kommandos werden wie von `root` ausgeführt. Sie beenden diesen Modus mit `exit` oder `[Strg]+[D]`.

Kommandos  
interaktiv mit  
sudo ausführen

```
user$ sudo -s      (in den root-Modus wechseln)
root# cmd1
root# cmd2
root# cmd3
user$ exit        (in den root-Modus wechseln)
```

Auf den meisten aktuellen Linux-Distributionen können Sie mit `sudo` auch grafische Programme ausführen. Das gilt allerdings nicht für Distributionen, deren Grafiksystem mit Wayland realisiert ist! Dort ist `sudo` auf Kommandos beschränkt, die im Textmodus in einem Terminalfenster ausgeführt werden.

sudo und  
Wayland

## sudo bei Ubuntu

Bei Ubuntu und einigen anderen Distributionen wird der Benutzer `root` standardmäßig ohne gültiges Passwort eingerichtet. Ein `root`-Login ist damit unmöglich! Auch `su` oder `ssh -l root` funktionieren nicht. Die einzige Möglichkeit zur Ausführung administrativer Kommandos bietet somit `sudo`. Die Datei `/etc/sudoers` enthält nur wenige Zeilen:

```
# Defaultkonfiguration in /etc/sudoers bei Ubuntu
Defaults    env_reset
Defaults    mail_badpass
Defaults    secure_path=\
    "/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
root       ALL=(ALL) ALL
%admin     ALL=(ALL) ALL
%sudo     ALL=(ALL:ALL) ALL
```

`Defaults env_reset` bewirkt, dass beim Benutzerwechsel alle Umgebungsvariablen zurückgesetzt werden. `Defaults mail_badpass` führt dazu, dass nach einem fehlerhaften Login-Versuch eine Warn-E-Mail an den Administrator versandt wird. `Defaults secure_path` legt den Inhalt der `PATH`-Umgebungsvariablen für `sudo`-Kommandos fest.

Die vierte Zeile gibt `root` uneingeschränkten Zugriff auf alle Programme. Die Zeile ist unter Ubuntu eigentlich zwecklos, weil der `root`-Login gesperrt ist. Am wichtigsten sind die letzten zwei Zeilen: Sie erlauben allen Mitgliedern der Gruppen `sudo` und `admin` den Aufruf sämtlicher Programme.

Normalerweise ist unter Ubuntu nur der erste Benutzer, also der, der während der Installation eingerichtet wurde, Mitglied der `sudo`-Gruppe. Weitere Benutzer können ebenfalls dieser Administratoren-Gruppe zugeordnet werden. Bei älteren Ubuntu-Versionen wurde anstelle der `sudo`-Gruppe die `admin`-Gruppe verwendet.

Die folgende zusätzliche Zeile in `/etc/sudoers` erlaubt es dem Benutzer `kofler`, die Kommandos `apt` und `apt-get` sowie das Programm `Synaptic` ohne Passwort auszuführen. Damit können Updates und Paketinstallationen ohne Passwortheingabe durchgeführt werden.

```
# Ergänzung in /etc/sudoers bei Ubuntu
kofler ALL=NOPASSWD: /usr/sbin/synaptic, /usr/bin/apt-get, /usr/bin/apt
```

### sudo bei Raspbian

Auch bei der für den Raspberry Pi optimierten Distribution Raspbian gibt es standardmäßig keinen `root`-Login. Als Defaultbenutzer ist `pi` eingerichtet. Ungewöhnlich liberal ist der `sudo`-Eintrag für diesen Benutzer:

```
# /etc/sudoers
...
pi ALL=(ALL) NOPASSWD: ALL
```

`pi` darf damit jedes beliebige Kommando mit `sudo` ohne Passwort ausführen. Sicherer ist es, dieser Zeile das Kommentarzeichen `#` voranzustellen. `pi` darf dann weiter `sudo` nutzen, weil er der `sudo`-Gruppe angehört, muss sich aber durch sein Passwort authentifizieren.

### sudo bei CentOS, Fedora und Red Hat

Bei aktuellen Fedora-Distributionen und Red-Hat-Derivaten können Sie während der Installation einen neuen Benutzer einrichten und diesen zum Administrator machen. Das bedeutet, dass er der Gruppe `wheel` zugeordnet wird. `/etc/sudoers` enthält für diese Gruppe die folgende Zeile:

```
# in /etc/sudoers bei Fedora
...
%wheel    ALL=(ALL) ALL
```



Ansonsten verhält sich `sudo` weitgehend wie bei Ubuntu, d. h., `wheel`-Gruppenmitglieder müssen ihr eigenes Passwort angeben, um `sudo`-Kommandos ausführen zu dürfen. Losgelöst von `sudo` gibt es weiterhin den Benutzer `root` mit einem eigenen Passwort.

### sudo bei SUSE

`sudo` ist auch bei SUSE-Distributionen standardmäßig eingerichtet. Die Konfiguration weicht aber deutlich von der von Ubuntu, Fedora & Co. ab:

```
# Defaultkonfiguration in /etc/sudoers bei openSUSE
Defaults always_set_home # verhindert die Ausführung von X-Programmen
Defaults env_reset      # verhindert die Ausführung von X-Programmen
Defaults env_keep = "LANG LC_ADDRESS LC_CTYPE ..."
Defaults targetpw      # sudo fragt nach dem Passwort des Zielbenutzers
..
ALL ALL=(ALL) ALL      # mit dem richtigen Passwort darf jeder alles
root ALL=(ALL) ALL
```

Die ersten drei Zeilen verhindern aus Sicherheitsgründen die Ausführung von X-Programmen mit `sudo`. Wenn `sudo` den direkten Start solcher Programme unterstützen soll, müssen Sie diese Zeilen löschen bzw. ein `#`-Zeichen voranstellen.

`Defaults targetpw` bedeutet, dass grundsätzlich das Passwort für den Account angegeben werden muss, in dem das Kommando ausgeführt werden soll, in der Regel also das `root`-Passwort. Der größte Vorteil von `sudo`, dass nämlich nicht mehrere Benutzer mit Administrationsaufgaben gemeinsam das `root`-Passwort kennen müssen, geht damit verloren. Die Zeile `ALL ALL=(ALL) ALL` erlaubt schließlich allen Benutzern, jedes Kommando auszuführen, sofern das richtige Passwort für den Ziel-Account bekannt ist.

## 12.4 Prozesse unter einer anderen Identität ausführen (PolicyKit)

Die Grundidee des PolicyKits besteht darin, Programme in zwei Komponenten zu zerlegen: Der eine Teil enthält die Benutzeroberfläche und läuft mit gewöhnlichen Benutzerrechten. Der zweite Teil des Programms, der in der Nomenklatur des PolicyKits als *mechanism* bezeichnet wird, ist für Systemeingriffe zuständig und läuft mit `root`-Rechten. Diese Trennung hat den fundamentalen Vorteil, dass nicht mehr ein riesiges Programm mit `root`-Rechten laufen muss, sondern nur noch kleine Teile. Das reduziert mögliche Sicherheitsrisiken. Außerdem besteht theoretisch die Möglichkeit, dass verschiedene Benutzeroberflächen (z. B. ein Gnome- und ein KDE-Programm) auf ein einheitliches Set von Mechanismen zurückgreifen.

Konzept

Die Kommunikation zwischen den beiden Komponenten erfolgt durch ein sogenanntes Bussystem, in der Regel über den »D-Bus«. Ob ein bestimmter Mechanismus ausgeführt werden darf oder nicht, entscheiden Funktionen der PolicyKit-Bibliothek, die auf eine zentrale Rechtedatenbank zurückgreifen. Für die Entscheidung werden drei Kriterien berücksichtigt:

- ▶ **Subjekt:** Wer bzw. welcher Benutzer will Systemänderungen durchführen?
- ▶ **Objekt:** Welches Objekt soll verändert werden (z. B. eine Datei, eine Partition oder eine Netzwerkverbindung)?
- ▶ **Aktion:** Was soll gemacht werden (z. B. eine Partition in das Dateisystem einbinden)?

**Benutzersicht** In vielen Fällen bemerkt der Benutzer gar nichts vom PolicyKit. Beispielsweise erlaubt die Standardkonfiguration bei den meisten aktuellen Distributionen dem Dateimanager, externe Datenträger in das Dateisystem einzubinden. Dazu ist keine weitere Authentifizierung erforderlich: Der Vorgang erfolgt automatisch, sobald der Datenträger angeschlossen wird.

Eine zweite Variante besteht darin, dass die PolicyKit-Regeln eine Autorisierung verlangen – beispielsweise zur Durchführung eines Updates mit dem PackageKit. In diesem Fall erscheint ein Authentifizierungsdialog. Bemerkenswert ist, dass sich das PolicyKit bei entsprechender Konfiguration die Authentifizierung merkt und in \*.auths-Dateien in /var/lib/PolicyKit/ speichert. Wenn ein Benutzer sich also ein einziges Mal für einen bestimmten Vorgang authentifiziert hat, fragt PolicyKit in Zukunft nicht mehr nach.

Auf eine dritte Variante stoßen Sie bei diversen Gnome-Administrationswerkzeugen: Hier führt ein mit einem Vorhängeschloss gekennzeichnetener Button zum Authentifizierungsdialog. Erst nach der Angabe des root- oder Benutzerpassworts können Systemveränderungen durchgeführt werden.

**Konfiguration** Die Konfiguration des PolicyKits erfolgt an folgenden drei Orten:

|                                     |   |
|-------------------------------------|---|
| /etc/polkit-1/*                     | (globale Konfiguration, Voreinstellungen) |
| /usr/share/polkit-1/action/*.policy | (Aktionen)                                |
| /var/lib/polkit-1/*                 | (Rechte)                                  |

Bei der Grundkonfiguration gibt es distributionsspezifische Besonderheiten. Beispielsweise gibt /etc/polkit-1/localauthority.conf.d/51-ubuntu-admin.conf bei Ubuntu-Systemen allen Benutzern der admin- und sudo-Gruppen Administratorrechte. Unter Fedora gilt eine analoge Regel für wheel-Gruppenmitglieder.

**pkexec** Ein Bestandteil des PolicyKit-Pakets ist das bereits erwähnte Kommando pkexec. Es ist für die eigentliche Ausführung von Kommandos verantwortlich und wertet dabei die Policy-Konfiguration aus.

Für im Terminal auszuführende Kommandos funktioniert `pkexec` ohne weitere Konfiguration. Freilich wäre es noch einfacher, in diesem Fall `su` oder `sudo` zu verwenden.

```
user$ pkexec nano
```

Bei nicht für `pkexec` konfigurierten grafischen Programmen scheitert der Aufruf aber mit der Fehlermeldung *cannot open display*:

```
user$ pkexec gedit
Unable to init server ..., cannot open display
```

Sofern das Grafiksystem unter X läuft (nicht unter Wayland, siehe [Kapitel 21](#), »Grafiksystem«), reicht es aus, eine neue Konfigurationsdatei mit dem Namen `*.policy` im Verzeichnis `/usr/share/polkit-1/actions` einzurichten. Als Muster kann die PolicyKit-Datei des Partitionseditors `gparted` dienen. Die Datei sieht unter Fedora so aus:

Eigene  
Policy-Dateien (X)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE policyconfig PUBLIC
"-//freedesktop//DTD PolicyKit Policy Configuration 1.0//EN"
"http://www.freedesktop.org/standards/PolicyKit/1/policyconfig.dtd">
<!-- Datei
     /usr/share/polkit-1/actions/org.fedoraproject.pkexec.run-gparted.policy -->

<policyconfig>
  <action id="org.freedesktop.policykit.pkexec.run-gparted">
    <description>Run GParted</description>
    <message>Authentication is required to run GParted</message>
    <defaults>
      <allow_any>no</allow_any>
      <allow_inactive>no</allow_inactive>
      <allow_active>auth_admin_keep</allow_active>
    </defaults>
    <annotate key="org.freedesktop.policykit.exec.path">
      /usr/sbin/gparted
    </annotate>
    <annotate key="org.freedesktop.policykit.exec.allow_gui">
      TRUE
    </annotate>
  </action>
</policyconfig>
```

Die äquivalente Datei `myown.gedit.policy` für `gedit` sieht dann so aus:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ... wie oben>
<!-- Datei /usr/share/polkit-1/actions/myown.gedit.policy -->
```

```

<policyconfig>
  <action id="myown.gedit">
    <description>Run the gedit program</description>
    <message>
      Authentication is required to run gedit with root privileges
    </message>
    <icon_name>gedit</icon_name>
    <defaults>
      <allow_any>auth_admin</allow_any>
      <allow_inactive>auth_admin</allow_inactive>
      <allow_active>auth_admin</allow_active>
    </defaults>
    <annotate key="org.freedesktop.policykit.exec.path">
      /usr/sbin/gedit
    </annotate>
    <annotate key="org.freedesktop.policykit.exec.allow_gui">
      true
    </annotate>
  </action>
</policyconfig>

```

Eine derartige Datei ist rasch eingerichtet und erlaubt es in der Folge, `gedit` in einem Terminalfenster mit `root`-Rechten zu starten. Dabei erscheint natürlich weiterhin zuerst ein Dialog zur Eingabe des `sudo`- oder `root`-Passworts:

```
user$ pkexec gedit
```

**Wayland** Ich habe es bereits erwähnt: Der obige einfache Lösungsweg funktioniert nur unter X. Wenn das Grafiksystem unter Wayland läuft, müsste der Code von `gedit` vollkommen neu konzipiert werden, um nur gewisse Operationen (das Laden und Speichern von Dateien) mit `root`-Rechten zu erledigen, die eigentliche Benutzeroberfläche aber weiterhin mit gewöhnlichen Rechten auszuführen. So lange, bis sich die `gedit`-Entwickler diese Mühe machen, können Sie `gedit` unter Wayland nur mit gewöhnlichen Rechten nutzen. Die andere Alternative besteht darin, als Grafiksystem eben weiterhin X zu verwenden. Auch wenn sich Wayland nach und nach durchsetzt, wird X sicher noch etliche Jahre unterstützt.

## 12.5 Systemprozesse (Dämonen)

Als Dämonen (englisch *daemons*) werden Hintergrundprozesse zur Systemverwaltung bezeichnet. Diese Prozesse werden normalerweise während des Hochfahrens des Rechners im Rahmen des `Init`-Prozesses gestartet. Wenn Sie mit der Windows-Diktion vertraut sind, entsprechen Linux-Dämonen einfach Diensten. [Tabelle 12.1](#) beschreibt ganz kurz die Aufgaben der wichtigsten Dämonen. Soweit die Programme in diesem Buch beschrieben werden, werden die betreffenden Abschnitte angegeben.

| Prozess        | Bedeutung  |
|----------------|--|
| afpd           | Server für das Apple Filing Protocol ( <a href="#">Abschnitt 29.2</a> )              |
| apache2        | Webserver ( <a href="#">Abschnitt 32.1</a> )   |
| atd            | startet andere Programme zu vorgegebenen Zeiten (ähnlich wie cron).                  |
| avahi-daemon   | automatische Netzwerkkonfiguration (ZeroConf)  |
| bluetoothd     | Bluetooth-Verwaltung   |
| cron           | startet andere Programme zu vorgegebenen Zeiten ( <a href="#">Abschnitt 12.6</a> ).  |
| cupsd          | Drucker-Spooler ( <a href="#">Abschnitt 30.2</a> )                                   |
| dbus-daemon    | D-BUS-Kommunikation ( <a href="#">Abschnitt 18.8</a> )                               |
| dhclient       | DHCP-Client ( <a href="#">Abschnitt 26.4</a> )                                       |
| dhcpcd         | weist anderen Rechnern die IP-Netzwerkadresse zu ( <a href="#">Abschnitt 27.6</a> ). |
| dhcpcd         | ermittelt die eigene IP-Netzwerkadresse ( <a href="#">Abschnitt 26.4</a> ).          |
| dovecot        | IMAP- und POP-Server ( <a href="#">Abschnitt 34.5</a> )                              |
| gdm            | Gnome-Login-Manager ( <a href="#">Abschnitt 21.5</a> )                               |
| httpd          | Webserver (z. B. Apache)   |
| lightdm        | Login-Manager unter Raspbian und Ubuntu ( <a href="#">Abschnitt 21.5</a> )           |
| lockd          | NFS-3-Locking  |
| lpd            | herkömmlicher Drucker-Spooler auf der Basis von BSD-LPD                              |
| mdnsd          | automatische Netzwerkkonfiguration (ZeroConf, Rendezvous, Bonjour)                   |
| mysqld         | MySQL-Datenbank-Server ( <a href="#">Kapitel 33</a> )                                |
| named          | Domain-Nameserver  |
| NetworkManager | NetworkManager ( <a href="#">Abschnitt 26.1</a> )                                    |
| nmbd           | Nameserver für Windows/Samba ( <a href="#">Abschnitt 28.1</a> )                      |
| nscd           | Cache für Benutzer-, Gruppen- und Rechnernamen ( <a href="#">Abschnitt 18.6</a> )    |
| ntpd           | Zeiteinstellung mit dem Network Time Protocol ( <a href="#">Abschnitt 18.4</a> )     |
| postfix        | Mail-Server zum Versenden von E-Mails ( <a href="#">Abschnitt 34.2</a> )             |
| rsyslogd       | protokolliert Systemmeldungen ( <a href="#">Abschnitt 18.9</a> ).                    |
| sddm           | Simple Desktop Display Manager (KDE) ( <a href="#">Abschnitt 21.5</a> )              |
| sendmail       | Mail-Server zum Versenden von E-Mails  |
| smartd         | SMART-Festplattenüberwachung ( <a href="#">Abschnitt 22.19</a> )                     |
| smbd           | Datei-Server für Windows/Samba ( <a href="#">Kapitel 28</a> )                        |
| sshd           | Secure-Shell-Server ( <a href="#">Kapitel 31</a> )                                   |
| spamd          | erkennt Spam-Mails ( <a href="#">Abschnitt 34.7</a> ).                               |

Tabelle 12.1 Wichtige Systemprozesse

| Prozess          | Bedeutung  |
|------------------|--|
| systemd-journald | protokolliert Systemmeldungen (Abschnitt 18.10). |
| systemd-udev     | Device-Verwaltung (Abschnitt 11.9)               |
| udev             | Device-Verwaltung (Abschnitt 11.9)               |
| vsftpd           | FTP-Server (Abschnitt 32.8)                      |
| xinetd           | startet andere Netzwerkdämonen (Abschnitt 24.7). |

Tabelle 12.1 Wichtige Systemprozesse (Forts.)

### Kernel-Threads

Neben gewöhnlichen Server-Diensten wie httpd (Apache) gibt es Hintergrundprozesse, bei denen es sich aber nicht um richtige Programme handelt, sondern um Teilprozesse (Threads) des Kernels. Sie erkennen diese Prozesse daran, dass ps axu ihre Namen in eckige Klammern stellt. Manchen dieser Teilprozesse ist eine Nummer hintangestellt, die auf die CPU hinweist. kblockd/0 verwaltet somit den Block-Device-Buffer für die erste CPU, kblockd/1 den Buffer für die zweite CPU etc.

| Kernel-Thread | Bedeutung  |
|---------------|--|
| kacpid        | ACPI-Funktionen  |
| kblockd       | verwaltet den Block-Device-Buffer.                     |
| kdevtmpfs     | verwaltet das temporäre /dev-Dateisystem.              |
| khelperd      | lädt bzw. entfernt Kernelmodule für Benutzerprogramme. |
| knfsd         | NFS-Server   |
| kthread       | verwaltet Threads.                                     |
| nfsd          | NFS-Server   |
| ksoftirqd     | Hardware-Interrupt-Verwaltung                          |
| kswapd        | Swapping   |
| kworker       | Kernel-Funktionen ausführen (»arbeiten«)               |
| lockd         | NFS-Locking  |
| md            | RAID-Funktionen  |
| migration     | bestimmt, welche Prozesse auf welcher CPU laufen.      |
| rpciod        | NFS  |
| scsi_eh       | verwaltet SCSI-Fehler und -Timeouts.                   |
| watchdog      | überwacht, ob das System noch reagiert.                |

Tabelle 12.2 Ausgewählte Kernel-Threads

Die meisten Kernel-Threads betreffen Low-Level-Aufgaben des Betriebssystems (Speicherverwaltung, Prozessverwaltung, CPU-Steuerung etc.). Sie werden überwiegend bereits während der Systeminitialisierung zu Beginn des Systemstarts gestartet. Für normale Anforderungen ist keine spezielle Konfiguration erforderlich. Die Funktion der wichtigsten Kernel-Threads ist in [Tabelle 12.2](#) zusammengefasst. Wenn Sie mit `ps` ausschließlich Kernel-Threads anzeigen möchten, führen Sie das Kommando mit den folgenden Optionen aus:

```
user$ ps -f -p 2 --ppid 2
```

### Systemdienste starten und beenden

Die in [Tabelle 12.1](#) aufgezählten Dämonen werden über das Init-System gestartet, das ich in [Kapitel 24](#) im Detail beschreibe. Bei nahezu gängigen Distributionen wird `systemd` als Init-System verwendet.

An dieser Stelle finden Sie lediglich eine kurze Zusammenfassung, wie Sie einen Systemdienst manuell starten bzw. stoppen, was Sie tun müssen, damit der Dämon beim Systemstart automatisch gestartet wird, bzw. wie Sie den automatischen Start vermeiden. Diese Informationen werden Sie insbesondere beim Einrichten und Konfigurieren von Netzwerkdiensten häufig benötigen. Beachten Sie, dass nicht nur das Kommando, sondern auch der Dienstname je nach Distribution variieren kann. Beispielsweise heißt das Script zum Starten des Webservers Apache bei Debian, Ubuntu und SUSE `apache2`, bei Fedora und Red Hat dagegen `httpd`.

```
root# systemctl start name (einmalig starten)
root# systemctl stop name (stoppen)

root# systemctl status name (Status ermitteln)
root# systemctl restart name (neu starten)
root# systemctl reload name (Konfiguration neu laden)

root# systemctl enable name (in Zukunft automatisch starten)
root# systemctl disable name (in Zukunft nicht mehr starten)
```

Unter Debian und Ubuntu werden neu installierte Server-Dienste sofort und in Zukunft bei jedem Neustart automatisch ausgeführt. Bei anderen Distributionen müssen Sie, sobald Sie die Konfiguration abgeschlossen haben, einmalig diese beiden Kommandos ausführen:

```
root# systemctl start name (einmalig starten)
root# systemctl disable name (in Zukunft nicht mehr starten)
```

Anstelle von `systemctl` funktioniert auf den meisten Distributionen auch das ältere Kommando `service`, bei dem allerdings die auszuführende Aktion und der Dienstname zu vertauschen sind – also beispielsweise:

```
root# service name reload
```

**Init-V-Scripts** Soweit es in Ihrer Distribution noch traditionelle Init-V-Scripts gibt, können Sie diese direkt ausführen:

```
root# /etc/init.d/name reload
```

Bei Init-V-Scripts entscheiden Links in den Verzeichnissen `/etc/rc.d/rcN.d`, welche Dienste automatisch gestartet werden. Statt mit `systemctl enable` können Sie diese Links auch mit distributionsspezifischen Kommandos einrichten: Bei älteren Debian- und SUSE-Versionen steht dazu das Kommando `inserv` zur Verfügung, unter CentOS und RHEL das Kommando `chkconfig`. Sollte das Init-V-Script keine Angaben dazu enthalten, in welchen Runleveln der Dienst normalerweise gestartet werden soll, müssen Sie dies mit der Option `--level` angeben. `--level 35` meint die Runlevel 3 und 5.

```
root# inserv name (alte Debian- und SUSE-Versionen)
root# chkconfig --add name (alte RHEL- und CentOS-Versionen)
root# chkconfig --level 35 name on (alte RHEL- und CentOS-Versionen)
```

Die folgenden Kommandos verhindern in Zukunft den automatischen Start beim Hochfahren:

```
root# inserv -r name (alte Debian- und SUSE-Versionen)
root# chkconfig --del name (alte RHEL- und CentOS-Versionen)
```

## 12.6 Prozesse automatisch starten (Cron)

Wenn Ihr Rechner plötzlich – scheinbar unvermittelt – damit beginnt, die Festplatte zu durchsuchen, Ihnen E-Mails zusendet etc., dann ist die Ursache fast immer der automatische Start von Prozessen durch den Dämon Cron. Dieses Programm wird beim Rechnerstart durch den Init-Prozess automatisch gestartet. Es wird einmal pro Minute aktiv, analysiert alle `crontab`-Dateien und startet die dort angegebenen Programme. Cron wird in erster Linie für Wartungsarbeiten verwendet – um Logging-Dateien zu komprimieren und zu archivieren, um temporäre Dateien zu löschen, um Verzeichnisse zu aktualisieren, Backups durchzuführen etc.

Die globale Konfiguration von Cron erfolgt durch die Datei `/etc/crontab`. Darüber hinaus dürfen Benutzer Ihre eigenen Cron-Jobs in den benutzerspezifischen Dateien `/var/spool/cron/[tabs/]username` definieren.

Das Recht der benutzerspezifischen Cron-Steuerung kann mit den beiden Dateien `/var/spool/cron/allow` und `/deny` eingestellt werden. Wenn `allow` existiert, dürfen nur die hier eingetragenen Benutzer Cron-Kommandos ausführen. Wenn `deny` existiert, sind die hier eingetragenen Benutzer ausgeschlossen. Existiert keine dieser Dateien, hängt es von der Kompilation von Cron ab, ob irgendwelche Benutzer außer `root` Cron verwenden dürfen.



### Minimalinstallationen ohne Cron

Bei gewöhnlichen Linux-Installationen wird Cron automatisch installiert. Beachten Sie aber, dass Cron bei Minimalinstallationen, wie sie im Server- und Virtualisierungsbereich üblich sind, mitunter fehlt! Abhilfe: Führen Sie `yum install crontab` (RHEL) bzw. `apt-get install cron` (Debian/Ubuntu) aus!

Die Datei `/etc/crontab` bzw. die Dateien in `/etc/cron.d` enthalten zeilenweise Einträge für die auszuführenden Programme. Die Syntax sieht so aus: `/etc/crontab`

```
# in /etc/crontab
min hour day month weekday user command
```

| Spalte  | Bedeutung  |
|---------|--|
| min     | gibt an, in welcher Minute (0–59) das Programm ausgeführt werden soll.   |
| hour    | gibt die Stunde an (0–23).   |
| day     | gibt den Tag im Monat an (1–31).   |
| month   | gibt den Monat an (1–12).  |
| weekday | gibt den Tag der Woche an (0–7, 0 und 7 bedeuten jeweils Sonntag).       |
| user    | gibt an, für welchen Benutzer das Kommando ausgeführt wird (meist root). |
| command | enthält schließlich das auszuführende Kommando.                          |

**Tabelle 12.3** crontab-Spalten

Wenn in den ersten fünf Feldern statt einer Zahl ein `*` angegeben wird, wird dieses Feld ignoriert. `15 * * * *` bedeutet beispielsweise, dass das Kommando immer 15 Minuten nach der ganzen Stunde ausgeführt werden soll, in jeder Stunde, an jedem Tag, in jedem Monat, unabhängig vom Wochentag. `29 0 * * 6` bedeutet, dass das Kommando an jedem Samstag um 0:29 Uhr ausgeführt wird.

Für die Zeitfelder ist auch die Schreibweise `*/n` erlaubt. Das bedeutet, dass das Kommando jede  $n$ -te Minute/Stunde etc. ausgeführt wird. `*/15 * * * *` würde also bedeuten, dass das Kommando viertelstündlich ( $n:00$ ,  $n:15$ ,  $n:30$  und  $n:45$ ) ausgeführt wird. Um die globale Cron-Konfiguration zu verändern, können Sie `/etc/crontab` bzw. die Dateien in `/etc/cron.d/*` direkt mit einem Editor bearbeiten.

Anstelle der fünf Zeitspalten, die in [Tabelle 12.3](#) zusammengefasst sind, dürfen auch die in [Tabelle 12.4](#) aufgezählten `@`-Kürzel verwendet werden. Eine weitere Zusatzregel besagt, dass ein Minuszeichen am Beginn der ersten Spalte verhindert, dass Syslog die Kommandoausführung protokolliert. Das ist allerdings nur erlaubt, wenn die sechste Spalte `root` enthält.

| Kürzel    | Code      | Bedeutung                   |
|-----------|-----------|-----------------------------|
| @reboot   | –         | nach jedem Reboot ausführen |
| @yearly   | 0 0 1 1 * | einmal im Jahr ausführen    |
| @annually | 0 0 1 1 * | wie @yearly                 |
| @monthly  | 0 0 1 * * | einmal pro Monat ausführen  |
| @weekly   | 0 0 * * 0 | einmal pro Woche ausführen  |
| @daily    | 0 0 * * * | einmal pro Tag ausführen    |
| @hourly   | 0 * * * * | einmal pro Stunde ausführen |

**Tabelle 12.4** crontab-Intervallkürzel ersetzen die ersten fünf Spalten.

### Die Crontab-Syntax erfordert einen Zeilenumbruch nach der letzten Zeile

Achten Sie darauf, dass alle Cron-Konfigurationsdateien mit einem Zeilenumbruch enden müssen – andernfalls wird die letzte Zeile ignoriert!

**Crontab-Beispiel** Die folgenden Zeilen geben ein paar Beispiele für Einträge in `/etc/crontab`:

```
# Backup durchführen jede Nacht um 1:45
45 1 * * * root /myscripts/backup-site
# Aufräumarbeiten am 1. jedes Monats um 6:00
0 6 1 * * root /myscripts/cleanup
# Netzwerkverbindung alle 10 Minuten protokollieren
*/10 * * * * root /myscripts/log-ping-result
```

**Fehler-E-Mail** Wenn ein Cron-Job Ausgaben oder Fehlermeldungen liefert, werden diese automatisch an `root@localhost` versendet. Durch die Einstellung der Variablen `MAILTO` in `crontab` können Sie auch eine andere E-Mail-Adresse einstellen. Beachten Sie aber, dass das Versenden nicht lokaler E-Mails einen konfigurierten E-Mail-Server voraussetzt.

**Benutzerspezifische crontab-Dateien** Die Dateien `/var/spool/cron/[tabs/]user` haben dasselbe Format wie `crontab`. Der einzige Unterschied besteht darin, dass die `user`-Spalte fehlt, weil diese Information ja bereits aus dem Namen der `crontab`-Datei hervorgeht. Um benutzerspezifische Cron-Einträge zu verändern, sollten Sie dazu das Kommando `crontab -e` einsetzen. Führen Sie vorher `export EDITOR=emacs` aus, wenn Sie nicht mit dem `vi` arbeiten möchten. `man cron` und `man crontab` geben weitere Informationen.

## **/etc/cron.hourly, .daily, .weekly, .monthly**

Bei allen gängigen Distributionen sind die folgenden vier Verzeichnisse eingerichtet:

```
/etc/cron.hourly/    (enthaltene Scripts stündlich ausführen)
/etc/cron.daily/     (enthaltene Scripts täglich ausführen)
/etc/cron.weekly/   (enthaltene Scripts wöchentlich ausführen)
/etc/cron.monthly/  (enthaltene Scripts monatlich ausführen)
```

Sie können in diesen Verzeichnissen also eigene Scripts speichern. Vergessen Sie nicht, das *execute*-Bit zu setzen (`chmod a+x datei`)! Die Scripts werden dann einmal pro Stunde, Tag, Woche oder Monat mit *root*-Rechten ausgeführt. Die Verwendung der */etc/cron.xxx-ly*-Verzeichnisse erspart Ihnen das Nachdenken über die korrekte Crontab-Syntax. Außerdem haben diese Verzeichnisse im Vergleich zu herkömmlichen Crontab-Einträgen den Vorteil, dass so definierte Scripts selbst dann zuverlässig ausgeführt werden, wenn der Rechner nicht ständig läuft.

Die Distributionen unterscheiden sich erheblich in der Art und Weise, *wie* die in den vier Verzeichnissen enthaltenen Scripts ausgeführt werden: Die meisten Distributionen setzen voraus, dass das im nächsten Abschnitt beschriebene Programm Anacron installiert ist. Anacron kümmert sich dann um die Ausführung.

Bei Ubuntu funktioniert die Ausführung der */etc/cron.xxx-ly*-Scripts aber auch ohne Anacron. Das gelingt durch die folgende Crontab-Konfiguration:

```
# /etc/crontab bei Ubuntu
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || \
    ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || \
    ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || \
    ( cd / && run-parts --report /etc/cron.monthly )
```

Im Klartext bedeutet das, dass Cron

- ▶ 17 Minuten nach jeder vollen Stunde alle Script-Dateien des Verzeichnisses */etc/cron.hourly* ausführt,
- ▶ täglich um 6:25 Uhr alle Script-Dateien des Verzeichnisses */etc/cron.daily* ausführt,
- ▶ wöchentlich am Sonntag um 6:47 Uhr alle Script-Dateien des Verzeichnisses */etc/cron.weekly* ausführt und
- ▶ an jedem Ersten des Monats um 6:52 Uhr alle Script-Dateien des Verzeichnisses */etc/cron.monthly* ausführt.

Sollte Anacron installiert sein, dann liefert `test -x` ein positives Ergebnis und es kommt zu keinem Aufruf von `run-parts`. Das ist auch gar nicht notwendig, denn nun übernimmt Anacron die Kontrolle über die Scripts der vier `/etc/cron.xxx-ly`-Verzeichnisse.

### Regeln für Dateinamen in `/etc/cron.daily`, `-.weekly` und `-.monthly`

Bei Debian und Ubuntu dürfen die Dateinamen eigener Scripts in `cron.daily`, `cron.weekly` und `cron.monthly` ausschließlich aus Zahlen, Buchstaben und Binde- und Unterstrichen bestehen! Sobald der Dateiname auch nur einen Punkt enthält, wird das Script von `run-parts` ignoriert! Diese Regeln sollen vermeiden, dass bei der Veränderung einer Script-Datei entstehende Backup-Dateien ebenfalls ausgeführt werden.

Wenn Sie sich vergewissern möchten, welche Scripts aus `run.daily` täglich ausgeführt werden, führen Sie `run-parts --test /etc/cron.daily` aus.

Auch SUSE-Distributionen sind von Anacron unabhängig. Hier kümmert sich das Script `run-crons` um die `/etc/cron.xxx-ly`-Verzeichnisse. Dieses Script wird viertelstündlich ausgeführt:

```
# /etc/crontab bei SUSE
-*/15 * * * * root test -x /usr/lib/cron/run-crons && \
                /usr/lib/cron/run-crons >/dev/null 2>&1
```

`run-crons` ist nicht ganz so pedantisch wie Ubuntu's `run-part` bei der Überprüfung der Dateinamen. Nicht ausgeführt werden lediglich Scripts, deren Namen wie Backup-Dateien oder Überbleibsel der Kommandos `rpm` oder `svn` aussehen.

## Anacron

Cron setzt voraus, dass der Rechner ständig läuft – also so, wie es bei Servern der Fall ist. Ganz anders sieht es bei Notebooks oder Desktop-PCs aus, die häufig ein- und wieder ausgeschaltet werden: Um auch in diesem Fall sicherzustellen, dass täglich, wöchentlich oder monatlich auszuführende Aufgaben erledigt werden, wurde Anacron entwickelt.

Anacron speichert den Ausführungszeitpunkt in Dateien des Verzeichnisses `/var/spool/anacron`. Damit wird ausgeschlossen, dass ein für die tägliche Ausführung bestimmtes Script am selben Tag zweimal ausgeführt wird. Anacron wird durch `/etc/anacrontab` gesteuert. Bei den meisten Distributionen sieht die Defaultkonfiguration wie folgt aus:

```
# Datei /etc/anacrontab (Debian, CentOS, Fedora, Red Hat, Ubuntu)
1      5  cron.daily      nice run-parts /etc/cron.daily
7      25 cron.weekly    nice run-parts /etc/cron.weekly
@monthly 45 cron.monthly    nice run-parts /etc/cron.monthly
```

Dabei gibt die erste Spalte an, wie oft die Aufgabe erledigt werden soll (1 = täglich, 7 = einmal pro Woche). Die zweite Spalte nennt eine Verzögerungszeit in Minuten, die Anacron verstreichen lässt, bevor es `run-parts` ausführt. Diese Zeitspanne soll verhindern, dass tägliche, wöchentliche und monatliche Aufgaben zugleich erledigt werden.

Wenn viele Cron-Jobs zur gleichen Zeit starten, kann dies zu einer starken CPU- oder I/O-Belastung für das System führen. Eine Möglichkeit besteht, die Jobs eben konsequent so einzurichten, dass dies nicht vorkommt. Eine andere Möglichkeit bietet die Variable `RANDOM_DELAY`, die in `anacrontab` definiert werden kann. Sie gibt eine zufällige Verzögerungszeit in Minuten an. `RANDOM_DELAY=20` bewirkt, dass jeder durch Anacron ausgeführte Job um eine zufällige Zeitspanne von 0 bis 20 Minuten verzögert wird.

Zufällige  
Verzögerung von  
Cron-Jobs

## 12.7 Prozesse automatisch starten (systemd-Timer)

Immer mehr Distributionen verwenden `systemd` als Init-System, also zum automatischen Starten und Beenden von Prozessen beim Hoch- bzw. Herunterfahren des Rechners. Eine detaillierte Vorstellung von `systemd` in dieser Funktion folgt in [Abschnitt 24.1](#). `systemd` kann sich aber ähnlich wie Cron auch um die regelmäßige Ausführung von Prozessen kümmern. Dieser Aspekt von `systemd` steht hier im Vordergrund.

Die meisten Distributionen machen momentan nur recht sparsam von den `systemd`-Timer-Funktionen Gebrauch. Insofern ist `systemd` momentan nicht als Ersatz für Cron zu betrachten, sondern eher als Ergänzung. Sofern die Linux-Distributoren mitspielen, hat `systemd` aber durchaus das Potenzial, Cron längerfristig abzulösen. Ein Vorreiter ist in dieser Hinsicht wie üblich Fedora – aber selbst in der getesteten Version 23 gibt es standardmäßig nur vier aktive Timer. Diesen stehen ebenso viele vordefinierte Cron-Jobs in `/etc/cron.*` gegenüber.

Die zeitgesteuerte Ausführung von Programmen wird in `systemd` durch Timer-Unit-Dateien gesteuert, also durch Dateien mit der Endung `.timer`. Einen Überblick über aktive Timer gibt `systemctl`, wobei die folgende Ausgabe aus Platzgründen gekürzt dargestellt ist:

systemd-Timer

```
user$ systemctl list-timers
NEXT          LEFT          LAST          UNIT
Don 2017-08-20 09:33 18min left   Mit 2017-08-19 09:33 systemd-tmpfiles.timer
Don 2017-08-20 09:42 26min left   Don 2017-08-20 08:42 dnf-makecache.timer
Don 2017-08-20 11:19 2h 3min left Mit 2017-08-19 11:15 dnf-automatic.timer
Fre 2017-08-21 00:00 14h left     Don 2017-08-20 00:00 mlocate-updatedb.timer
Fre 2017-08-21 00:00 14h left     Don 2017-08-20 00:00 unbound-anchor.timer
```

**.timer-Syntax** Die `.timer`-Dateien liegen in einem leicht verständlichen Textformat vor. Die folgenden Zeilen zeigen die Datei `dnf-automatic.timer`, die für den täglichen Start des Scripts `dnf-automatic` verantwortlich ist. Dieses Script aus dem gleichnamigen Paket aktualisiert die Paketdatenbank und installiert bei entsprechender Konfiguration alle verfügbaren Updates.

```
# Datei /usr/lib/systemd/system/dnf-automatic.timer
[Unit]
Description=dnf-automatic timer

[Timer]
OnBootSec=1h
OnUnitInactiveSec=1d

[Install]
WantedBy=basic.target
```

`OnBootSec=1h` bewirkt, dass der Prozess eine Stunde nach einem Neustart des Rechners erstmalig ausgeführt wird. `OnUnitInactiveSec=1d` bedeutet, dass der Dienst 24 Stunden nach dem Ende des letzten Ablaufs wieder gestartet werden soll. Wenn die Zeitangabe ohne Einheit erfolgt, sind Sekunden gemeint. Die Syntax der Zeitangaben ist in `man systemd.time` dokumentiert. Zulässig ist z. B. `2h 15min` oder `2weeks` oder `4months`.

Anstelle dieser etwas vagen Vorgaben können Sie die Zeitangaben im `[Timer]`-Abschnitt auch absolut eintragen. Die folgenden Einstellungen bewirken, dass die Aufgabe jeden Tag um 8:15 erledigt wird. `Persistent=true` bewirkt, dass versäumte Jobs beim nächsten Rechnerstart sofort nachgeholt werden.

```
# in einer .timer-Datei
[Timer]
OnCalendar=8:15
Persistent=true
```

Für absolute Zeitangaben sieht `systemd` eine komplexe Syntax vor, die in `man systemd.time` im Abschnitt »Calendar Events« ausführlich dokumentiert ist. So bewirkt `OnCalendar=Sun 2017-**-* 17:15` beispielsweise, dass ein Job an jedem Sonntag des Jahres 2017 um 17:15 ausgeführt wird.

`systemd` versucht, eingerichtete Timer-Jobs standardmäßig in einem Zeitfenster von einer Minute nach der vorgesehenen Zeit auszuführen. Wenn Sie möchten, dass sich `systemd` exakter an Ihre Vorgaben hält, geben Sie im `[Timer]`-Abschnitt mit `AccuracySec` eine kleinere Zeitspanne an, z. B. von `1s` (eine Sekunde) bis hin zu `1us` (theoretisch eine millionstel Sekunde, ganz so exakt wird der Start in der Praxis aber nicht immer gelingen).

Jeder `name.timer`-Datei muss eine entsprechende `name.service`-Datei gegenüberstehen, die Details zum auszuführenden Prozess enthält. Für das obige `dnf-automatic`-Beispiel sieht die `.service`-Datei wie folgt aus:

```
# Datei /usr/lib/systemd/system/dnf-automatic.service
[Unit]
Description=dnf automatic

[Service]
Type=oneshot
Nice=19
IOSchedulingClass=2
IOSchedulingPriority=7
Environment="ABRT_IGNORE_PYTHON=1"
ExecStart=/usr/bin/dnf-automatic /etc/dnf/automatic.conf --timer
```

Neu eingerichtete Timer müssen wie üblich durch `systemctl start` erstmalig gestartet und durch `systemctl enable` dauerhaft (also über den nächsten Neustart hinweg) aktiviert werden. Analog beenden Sie den Aufruf durch `systemctl stop` und `systemctl disable`. Damit Änderungen in bereits aktiven Timern wirksam werden, führen Sie am einfachsten das folgende Kommando aus:

```
root# systemctl reenable --now name.timer
```

Nach dem Einrichten neuer Timer-Jobs sollten Sie sich immer mit `systemctl list-timers` vergewissern, dass `systemd` Ihre `.timer`-Angaben korrekt interpretiert hat.

Das folgende Beispiel zeigt, wie Sie mit `systemd` ein simples Script zur Überprüfung der Netzwerkverbindung alle 10 Minuten ausführen. Das Script `ping-kofler` testet durch ein dreimaliges `ping`, ob und wie schnell der Server `kofler.info` auf Netzwerkfragen reagiert. Die gerade aktuelle Zeit und die `ping`-Ausgaben werden in `/var/log/ping-kofler.log` protokolliert. `&>>` bewirkt, dass sowohl die Standardausgabe als auch eventuell auftretende Fehler zu einer vorhandenen Datei hinzugefügt werden. Dieser Operator setzt die `bash`-Version 4 voraus. Vergessen Sie nicht, das Script mit `chmod a+x` ausführbar zu machen.

```
#!/bin/bash
# Datei /usr/local/bin/ping-kofler
date &>> /var/log/ping-kofler.log
ping -c 3 kofler.info &>> /var/log/ping-kofler.log
```

Für den Aufruf des Scripts ist die folgende `.service`-Datei zuständig:

```
# Datei /etc/systemd/system/ping-kofler.service
[Unit]
Description=simpler ping-Test

[Service]
ExecStart=/usr/local/bin/ping-kofler
```

Die Einstellungen für den automatischen Start befinden sich in der folgenden `.timer`-Datei. Das Script soll 30 Sekunden nach `systemctl start` erstmalig ausgeführt werden und anschließend alle 10 Minuten. Der `[Install]`-Abschnitt ist erforderlich, damit der Timer mit `systemctl enable` dauerhaft aktiviert werden kann.

```
[Unit]
Description=ping-kofler timer
```

```
[Timer]
OnActiveSec=30s
OnUnitActiveSec=10m
```

```
[Install]
WantedBy=basic.target
```

Um die Script-Ausführung zu starten und dauerhaft einzurichten, sind nun noch zwei `systemctl`-Kommandos notwendig:

```
root# systemctl start ping-kofler.timer
root# systemctl enable ping-kofler.timer
```

Die zwei Kommandos können mit der Option `--now` zu einem kombiniert werden:

```
root# systemctl enable --now ping-kofler.timer
```

### Cron/Anacron versus systemd

Das Beispiel macht sofort klar, dass das Einrichten eines `systemd`-Timers mit wesentlich mehr Aufwand verbunden ist als die eine `crontab`-Zeile, die für die periodische Cron-Ausführung erforderlich wäre.

Die Vorteile von `systemd` bestehen darin, dass `systemd`-Jobs exakter gesteuert werden können (eigenes Environment, siehe `man systemd.exec`, `cgroups`-Regeln etc.) und dass ihr Aufruf im `systemd`-Journal protokolliert wird.

Es gibt aber auch Nachteile: So wird bei einem Fehler nicht automatisch eine E-Mail versendet. Außerdem gibt es keine zufällige Verzögerung (`RANDOM_DELAY` in `anacrontab`), um die gleichzeitige Ausführung vieler Jobs zu vermeiden.

**Dokumentation** Weitere Informationen zur `systemd`-Timer-Konfiguration liefern `man systemd.timer` sowie die folgenden Seiten:

<https://wiki.archlinux.org/index.php/Systemd/Timers>

<https://blog.higgsboson.tk/2013/06/09/use-systemd-as-a-cron-replacement>



# Kapitel 13

## Konverter für Grafik, Text und Multimedia

Linux stellt zahllose Kommandos zur Verfügung, mit denen Sie Bilder, Texte und andere Dateien von einem Format in das andere konvertieren können: von GIF zu JPEG, von Latin1- zu Unicode-Text, von PostScript zu PDF, von HTML zu einfachem Text, von MP3 zu WAV etc.

Dieses Kapitel stellt eine Auswahl derartiger Kommandos vor und zeigt einige Anwendungsbeispiele. Wenn das eine oder andere Kommando aus diesem Kapitel bei Ihnen nicht zur Verfügung steht, müssen Sie das entsprechende Paket suchen und installieren – denn nur selten sind alle Pakete, die Sie brauchen, schon installiert.

### 13.1 Grafik-Konverter

Unter den vielen Grafik-Konvertern für Linux stechen zwei Pakete heraus: Image Magick und Netpbm. Beide Pakete kommen jeweils mit unzähligen Grafikformaten zurecht und bieten auch Kommandos bzw. Optionen zur einfachen Bildverarbeitung an (Bildgröße ändern, Bildausschnitt ändern, Kontrast verbessern, Farbzahl reduzieren etc.). Im Folgenden finden Sie eine kurze Vorstellung dieser beiden Pakete sowie einiger anderer Kommandos bzw. Bibliotheken zur Konvertierung von Bild-dateien.

Das Programmpaket Image Magick besteht aus mehreren Einzelkommandos, deren wichtigstes `convert` ist. Es erzeugt aus einer vorhandenen Bilddatei eine neue und ändert dabei das Format. Quell- und Zielformat gehen dabei einfach aus den Dateinamen hervor. Das folgende Kommando erzeugt also die Datei `bild.png` im PNG-Format:

Image Magick

```
user$ convert bild.jpg bild.png
```

Durch über 100 Optionen können diverse Bildparameter verändert werden (Größe, Farbzahl, Kompressionsgrad etc.):

```
user$ convert -resize 100x100 bild.jpg bild.png
user$ convert -type Grayscale bild.jpg bild.eps
user$ convert -quality 80 bild.bmp bild.jpg
```

`mogrify` funktioniert so ähnlich wie `convert`, verändert aber die vorhandene Datei (anstatt eine neue zu erzeugen):

```
user$ mogrify -resize 50% test.jpg
```

Abschließend noch kurz ein Überblick über weitere Kommandos: `compare` vergleicht zwei Bilder. `conjure` führt Bildverarbeitungskommandos der Magick Scripting Language (MSL) aus. `identify` liefert eine Beschreibung der Bilddatei, also Format, Größe etc. `import` erstellt einen Screenshot und speichert das Bild in einer Datei. `montage` setzt mehrere Bilder zu einem neuen zusammen.

```
user$ identify -verbose bild.png
Image: bild.png
  Format: PNG (Portable Network Graphics)
  Geometry: 85x100
  Type: TrueColor
  ...
```

Ein Beispiel für ein kleines Shell-Script, das von allen als Parameter übergebenen Dateien Thumbnails (verkleinerte Bilder) erzeugt, finden Sie in [Abschnitt 10.8](#), »bash-Script-Beispiele«. Eine umfassende Dokumentation zu allen Kommandos finden Sie auf der folgenden Website:

<https://imagemagick.org>

Wenn Sie die Funktionen von Image Magick mit einer grafischen Benutzeroberfläche nutzen möchten, bietet sich hierfür das Programm `Converseen` an:

<http://converseen.fasterland.net>

**Netpbm** Einen ähnlichen Ansatz wie Image Magick verfolgt auch das `Netpbm`-Paket, ehemals *Portable Bitmap Utilities*. Allerdings muss hier jede Datei zuerst in das interne `Pnm`- oder `Pbm`-Format umgewandelt werden. Das folgende Beispiel zeigt die Konvertierung einer TIFF-Datei in eine EPS-Datei, wobei der Farbraum des Bilds gleichzeitig normalisiert wird (`pnmnorm`):

```
user$ tiff2pnm bild.tif | pnmnorm | pnmtops -noturn -rle -scale 0.5 > bild.eps
```

Eine Beschreibung der ca. 200 `Netpbm`-Kommandos finden Sie hier:

<http://netpbm.sourceforge.net/doc>

|  |                    |
|--|--------------------|
| Das libtiff-Paket enthält die gleichnamige Bibliothek sowie diverse Kommandos zur Bearbeitung und Konvertierung von TIFF-Dateien. Zu den wichtigsten Konvertierungskommandos zählen <code>bmp2tiff</code> , <code>gif2tiff</code> , <code>tiff2pdf</code> und <code>tiff2ps</code> . Bei der Manipulation von TIFF-Dateien helfen unter anderem <code>tiffcp</code> , <code>tiffinfo</code> und <code>tiffsplit</code> . | libtiff-Bibliothek |
| Das libwmf-Paket enthält die gleichnamige Bibliothek sowie einige Kommandos zur Verarbeitung von WMF- und EMF-Dateien (Windows Metafile bzw. Enhanced Metafile). Wichtige Konvertierkommandos sind <code>wmf2eps</code> , <code>wmf2svg</code> sowie <code>wmf2gd</code> (Konvertierung in die Formate JPEG und PNG).  | libwmf-Bibliothek  |
| Je nach Distribution enthält das Paket <code>librsvg2</code> oder <code>librsvg2-bin</code> die Kommandos <code>rsvg</code> und <code>rsvg-convert</code> , um SVG-Dateien (Scalable Vector Graphics) in Bitmap-Dateien umzuwandeln.   | SVG-Konverter      |
| Ebenfalls je nach Distribution stehen verschiedene Bibliotheken und Kommandos zur Verarbeitung der EXIF-Daten in Fotos zur Auswahl, beispielsweise <code>exif</code> , <code>exiftran</code> oder <code>exiv2</code> .   | EXIF               |
| Bessere Digitalkameras bieten die Möglichkeit, Fotos ohne Qualitätsverlust in herstellereigenen RAW-Dateien zu speichern. Bei der Umwandlung derartiger Dateien in gewöhnliche Bilddateien hilft das Kommando <code>dcrw</code> aus dem gleichnamigen Paket.   | RAW-Dateien        |

## 13.2 Audio- und Video-Konverter

Tabelle 13.1 gibt einen Überblick über die wichtigsten Kommandos, um Audio-Dateien von der CD zu lesen bzw. um Audio- und Video-Dateien von einem Format in ein anderes umzuwandeln. Soweit der Paketname nicht ohnedies aus dem Kommando hervorgeht, ist er in Klammern angegeben. Der Paketname kann allerdings von Distribution zu Distribution variieren.

Im Folgenden finden Sie einige Zusatzinformationen und Beispiele zu ausgewählten Kommandos. Auf eine Referenz der Optionen dieser Kommandos verzichte ich aus Platzgründen. (Bei Bedarf hilft man `kommandoname` weiter.)

Beim CD-Ripper geht es nicht um *Jack the Ripper*, sondern um die Kunst, Audio-Tracks möglichst effizient und ohne Qualitätsverlust von einer CD auf die Festplatte zu übertragen. Zu den populärsten derartigen Kommandos zählen `cdparanoia` und `icedax` (ehemals `cdda2wav`). `cdparanoia` hat den Ruf, auch bei Problemfällen wie zerkratzten CDs besonders gute Ergebnisse zu liefern. Beide Kommandos werden mit einer Unzahl von Parametern gesteuert.

Hier zwei Beispiele: Das erste Kommando liest Track 3 einer CD im ersten CD/DVD-Laufwerk. Die resultierende Datei bekommt den Namen `audio.wav`:

```
root# icedax -D /dev/scd0 -t 3
```

| Format                    | Kommando (Paket)                          |
|---------------------------|---|
| CD → WAV                  | icedax, cdparanoia                        |
| MP3 → WAV                 | mpg123, mpg321, madplay                   |
| WAV → MP3                 | lame                                      |
| OGG → WAV                 | oggdec (vorbis-tools)                     |
| WAV → OGG                 | oggenc (vorbis-tools)                     |
| MP3 → OGG                 | mp32ogg                                   |
| AAC → WAV                 | faad                                      |
| WAV → AAC                 | faac                                      |
| WAV ↔ FLAC                | flac                                      |
| Audio ↔ Audio             | sox                                       |
| Audio ↔ Audio             | sfconvert (audiofile)                     |
| Audio/Video ↔ Audio/Video | ffmpeg (Audio/Video-Konverter)            |
| Audio/Video ↔ Audio/Video | mencoder (noch ein Audio/Video-Konverter) |

**Tabelle 13.1** Audio- und Video-Konverter

Das folgende Kommando liest Track 4 von einer CD im selben Laufwerk. Das Ergebnis ist die Datei `cdda.wav` im lokalen Verzeichnis:

```
root# cdparanoia -d /dev/scd0 "4"
```

**MP3-Encoder** Da viele wichtige MP3-Patente 2017 auslaufen, sind MP3-Encoder mittlerweile in vielen Distributionen standardmäßig verfügbar. Sollte das bei Ihrer Distribution nicht der Fall sein, finden Sie die Programme problemlos im Internet als Zusatzpakete. Das bekannteste Programm ist `lame`:

<http://lame.sourceforge.net>

Die Anwendung ist denkbar einfach: `lame input.wav output.mp3` erzeugt aus der Ausgangsdatei im WAV-Format eine entsprechende MP3-Datei. Der Vorgang und insbesondere die gewünschte Qualität der MP3-Datei werden durch zahlreiche Optionen gesteuert.

**MP3 → OGG** Die in [Tabelle 13.1](#) aufgezählten Kommandos können kombiniert werden, um beispielsweise eine MP3-Datei in das Ogg-Vorbis-Format umzuwandeln. Beachten Sie aber, dass mehrstufige Umwandlungen mit Qualitätsverlusten behaftet sein können und möglichst vermieden werden sollten!

```
user$ mpg321 -s in.mp3 -w - | oggenc - -o out.ogg
```

Durch das obige Kommando gehen auch die Info-Tags (ID3) verloren.

FLAC steht für *Free Lossless Audio Codec*. FLAC-Dateien sind zwar größer als MP3- oder Ogg-Dateien, aber wesentlich kleiner als WAV-Dateien. Der wesentliche Vorteil im Vergleich zu MP3 oder Ogg besteht darin, dass die Audio-Daten verlustfrei codiert werden. Zum Codieren und Decodieren verwenden Sie das Kommando `flac`.

FLAC

SoX steht für *Sound Exchange* und bietet mit dem Kommando `sox` eine weitere Möglichkeit, Audio-Dateien von einem Format in ein anderes umzuwandeln. `sox` kennt mehr Formate als `sfconvert` (siehe unten). Zu `sox` gibt es die grafischen Oberflächen `xsox` sowie `gsox`.

SoX

Das Paket `audiofile` implementiert die wichtigsten Funktionen der gleichnamigen Audio-File-Bibliothek des Computerherstellers SGI. Das interessanteste Kommando ist `sfconvert`: Es konvertiert Audio-Dateien zwischen den Formaten `aiff`, `aifc`, `next` und `wave`. `sfinfo` versucht zu ermitteln, welches Format eine Audio-Datei nutzt.

Audio File  
Bibliothek

Wenn Sie bei der Konvertierung von Audio-Dateien mehr Komfort wünschen, werden Sie vielleicht am Gnome-Programm `Sound Converter` Gefallen finden (siehe [Abbildung 7.24](#) auf Seite 259).

Sound Converter

Das Kommando `ffmpeg` aus dem gleichnamigen Paket konvertiert Audio- und Video-Dateien von einem Format in ein anderes. Die Liste der unterstützten Formate ist lang und kann mit `ffmpeg -formats` ermittelt werden.

ffmpeg  
(Video-Konverter)

Bei der Angabe von Optionen müssen Sie beachten, dass diese für die als Nächstes angegebene Datei gelten. Die Reihenfolge der Optionen ist daher entscheidend für die korrekte Funktion des Kommandos. Soweit Sie keine abweichenden Einstellungen vornehmen, verwendet `ffmpeg` für die Ergebnisdatei dieselben Codecs und Einstellungen wie in der Quelldatei und vermeidet so nach Möglichkeit Qualitätsverluste. Im folgenden Beispiel erstellt `ffmpeg` eine Filmdatei in DVD-Auflösung:

```
user$ ffmpeg -i in.avi out.mpg
user$ ffmpeg -i in.avi -y -target pal-dvd out.avi
```

`ffmpeg` eignet sich auch dazu, Audio- und Video-Daten zum Brennen einer eigenen DVD aufzubereiten. Ein entsprechendes Beispiel gibt man `ffmpeg`. Wenn Sie sich die vielen `ffmpeg`-Optionen nicht merken wollen, können Sie zur Umwandlung von Video-Dateien auch eine grafische Benutzeroberfläche verwenden, z. B. `winff`.

Einige Versionen von Debian und Ubuntu bieten anstelle von `ffmpeg` das Paket `libav-tools` an. Das darin enthaltene Kommando `avconv` ist zu `ffmpeg` weitestgehend kompatibel. `libav` ist eine Abspaltung des `ffmpeg`-Projekts, zu der es nach einem Streit unter `ffmpeg`-Entwicklern gekommen war. Eine Weile war `libav` die besser gepflegte Bibliothek. Mittlerweile ist Ubuntu wieder in das `ffmpeg`-Lager zurückgekehrt. Debian 9 stellt Pakete von beiden Konvertern zur Verfügung.

ffmpeg versus  
libav

**mencoder** Eine Alternative zu `ffmpeg/avconv` ist das Kommando `mencoder`. Es verwendet dieselbe Code-Basis wie der Video-Player MPlayer und wird zusammen mit diesem installiert. `mencoder` kommt mit allen Audio- und Video-Formaten zurecht, die auch von MPlayer unterstützt werden.

### 13.3 Textkonverter (Zeichensatz und Zeilentrennung)

Dieser Abschnitt stellt die Kommandos `recode`, `iconv`, `unix2dos` und `dos2unix` vor. Sie dienen dazu, den Zeichensatz bzw. die Zeilentrennzeichen von reinen Textdateien zu ändern. Das ist dann erforderlich, wenn Sie Textdateien zwischen Systemen mit unterschiedlichen Zeichensätzen bzw. Textformatkonventionen austauschen.

**recode** `recode` führt eine Zeichensatzkonvertierung von Zeichensatz 1 nach Zeichensatz 2 durch. Das folgende Kommando konvertiert die DOS-Datei `dosdat` in eine Linux-Datei mit dem Latin-1-Zeichensatz:

```
user$ recode ibmpc..latin1 < dosdat > linuxdat
```

Wie das folgende Beispiel beweist, kann `recode` auch das Zeilentrennzeichen verändern. Das Kommando ersetzt in der Datei `windowsdat` alle Zeilenenden (CR plus LF, also *Carriage Return* und *Line Feed*) durch das unter Linux übliche Zeilenende (nur LF). Der eigentliche Zeichensatz wird nicht geändert. Die resultierende Datei wird in `linuxdat` gespeichert.

```
user$ recode latin1/cr-lf..latin1 < windowsdat > linuxdat
```

`recode` liest die im Zeichensatz Latin-1 codierte Textdatei `latin1dat` und speichert sie als UTF-8-Datei (Unicode):

```
user$ recode latin1..u8 < latin1dat > utf8dat
```

**iconv** Eine populäre Alternative zu `recode` ist das Kommando `iconv`. Dieses Kommando ist allerdings nicht in der Lage, die Zeilentrennungszeichen zu verändern. Das folgende Beispiel erzeugt abermals aus einer Latin-1-codierten Textdatei eine entsprechende UTF-8-Datei:

```
user$ iconv -f latin1 -t utf-8 latin1dat > utf8dat
```

**dos2unix, unix2dos** Die Kommandos `dos2unix` und `unix2dos` aus dem Paket `dos2unix` ändern die Zeilentrennungszeichen zwischen dem DOS/Windows-typischen Format (CR plus LF) und dem Unix/Linux-typischen Format (nur LF). Die Kommandos eignen sich nur für Textdateien mit Ein-Byte-Zeichensätzen (z. B. ASCII, Latin-1), nicht für Unicode-Dateien!

```
user$ dos2unix datei.txt
```

## 13.4 Dateinamenkonverter (Zeichensatz)

Bis ungefähr zur Jahrtausendwende war es unter Linux üblich, Dateinamen im Zeichensatz Latin-1 darzustellen. Mittlerweile gilt hingegen Unicode (UTF-8) als Standard. Bei der Umstellung der Dateinamen von einem Zeichensatz auf einen anderen hilft das Kommando `convmv`. Dieses Kommando steht allerdings selten standardmäßig zur Verfügung. Bei einigen Distributionen kann es mühelos in Form eines gleichnamigen Pakets installiert werden. Sollte für Ihre Distribution kein entsprechendes Paket existieren, müssen Sie das Perl-Script von der folgenden Seite herunterladen:

<https://j3e.de/linux/convmv>

Um rekursiv alle Dateien eines Verzeichnisses vom Zeichensatz Latin-1 auf UTF-8 umzustellen (mit Rückfrage für jede einzelne Änderung), rufen Sie `convmv` so auf:

```
user$ convmv -r -i --notest -f iso-8859-1 -t utf8 verzeichnisname
```

`convmv` verändert nur den Namen, nicht den Inhalt der Dateien! Bei ersten Tests ist es empfehlenswert, auf die Option `--notest` zu verzichten: `convmv` zeigt dann nur die geplanten Änderungen an, ohne diese tatsächlich auszuführen.

`convmv` versucht, selbstständig Dateinamen zu erkennen, die bereits den UTF-8-Zeichensatz nutzen, und verzichtet in diesem Fall auf eine neuerliche Namensänderung. Diesen Schutz können Sie durch die Option `--nosmart` deaktivieren.

## 13.5 Dokumentkonverter (PostScript, PDF, HTML, LaTeX)

Dieser Abschnitt stellt Kommandos vor, die bei der Bearbeitung und Konvertierung von Dokumenten in den Formaten PostScript, PDF, HTML etc. helfen. [Tabelle 13.2](#) gibt einen ersten Überblick.

### Text → PostScript

Wenn Sie Textdateien mit `lpr datei` direkt ausdrucken, kümmert sich das Drucksystem normalerweise automatisch um die Formatierung des Texts. Wenn Sie allerdings besondere Wünsche haben, wie der resultierende Ausdruck formatiert werden soll, empfiehlt sich eine manuelle Konvertierung und ein anschließender Ausdruck der PostScript-Datei. Dazu eignen sich unter anderem die Kommandos `a2ps`, `enscript` und `mpage`. Die drei Kommandos bieten dieselben Grundfunktionen, unterscheiden sich aber durch diverse Formatierungsoptionen.

| Format  | Kommando                          |
|---|-----------------------------------|
| Text → PostScript                                 | a2ps, enscript, mpage             |
| HTML → Text, PostScript                           | html2text, html2ps                |
| PostScript ↔ PDF                                  | ps2pdf, epstopdf, pdf2ps, pdftops |
| PostScript, PDF → Bitmap, Druckerformat           | gs                                |
| PostScript → PostScript (Seiten extrahieren etc.) | psutils                           |
| PDF → PDF (Bilder/Seiten extrahieren etc.)        | pdftk, pdfnup, pdfjoin, pdftedit  |
| PDF → Text  | pdftotext                         |
| LaTeX → DVI, PostScript, PDF                      | latex, pdflatex, dvips, dvipdf    |
| Markdown → HTML                                   | markdown                          |
| Markdown ↔ diverse Formate                        | pandoc                            |

Tabelle 13.2 Dokument-Konverter

- a2ps** a2ps steht für *Any to PostScript* und kann beispielsweise auch Texinfo-Dateien in das PostScript-Format umwandeln. Die folgenden Beispiele beschränken sich aber auf reine Textdateien. Beachten Sie, dass die Textdateien einen Latin-Zeichensatz nutzen müssen (nicht Unicode!). Standardmäßig formatiert das Kommando den Text in einer zweiseitigen Seite im Querformat.

```
user$ a2ps text.txt -o postscript.ps
```

Das folgende Kommando verarbeitet mehrere Textdateien und formatiert die Ausgabe mit vier kleinen Seiten pro Blatt. Wenn a2ps den Text als Programmcode erkennt, führt es automatisch eine Syntaxhervorhebung durch (Schlüsselwörter fett, Kommentare kursiv etc.).

```
user$ a2ps datei1.c datei2.c datei3.h -4 -o postscript.ps
```

- enscript** enscript konvertiert Textdateien in die Formate PostScript, HTML und RTF. Das Kommando erwartet die Textdatei im Zeichensatz Latin-1. Das folgende Kommando erzeugt DIN-A4-Seiten im Querformat mit drei Spalten pro Seite:

```
user$ enscript -M A4 --landscape -3 text.txt -p postscript.ps
```

- mpage** Auch mpage konvertiert Textdateien in das PostScript-Format, per Default mit vier Seiten pro Blatt und im Letter-Format. Das folgende Kommando erzeugt DIN-A4-Seiten im Querformat mit zwei Seiten pro Blatt:

```
user$ mpage -2 -bA4 text.txt > postscript.ps
```



Leider kommt keines der oben vorgestellten Kommandos dieser Programme mit Unicode-Text (UTF-8) zurecht. Wenn Sie Unicode-Dokumente ausdrucken möchten, nehmen Sie am besten einen Unicode-Editor zu Hilfe. Eine einfache Konvertierung in das PostScript-Format können Sie auch mit `cnprint` vornehmen:

<http://ywpw.com/cai/software>

### HTML → Text, PostScript

`html2text` konvertiert HTML-Dokumente in reine Textdateien. Das ist dann praktisch, wenn HTML-Dateien in einer Form weitergegeben werden sollen, die ein bequemes Lesen ohne Webbrowser möglich macht. html2text

```
user$ html2text datei.html > text.txt
```

`html2text` liefert Latin-1-Text (nicht Unicode!). Die Formatierung des Texts wird durch einige Optionen sowie durch `/etc/html2textrc` bzw. durch `.html2textrc` gesteuert (siehe `man html2textrc`). Zur Konvertierung von HTML in Text können Sie auch textbasierte Webbrowser wie Lynx, ELinks oder `w3m` einsetzen.

Für die automatische Konvertierung vom HTML- in das PostScript-Format eignet sich das Perl-Script `html2ps`. Die Verwendung ist denkbar einfach: `html2ps -D name.html > name.ps`. Die Option `-D` bewirkt, dass `html2ps` DSC-konforme Kommentare in die PostScript-Datei einbaut, was deren Weiterverarbeitung sehr erleichtert. html2ps

Sie sollten sich freilich keine Hoffnungen machen, dass `html2ps` mit modernen HTML-Seiten mit JavaScript-Code, CSS-Formatierung etc. zurechtkommt. Zufriedenstellende Ergebnisse liefert `html2ps` nur bei sehr simplen HTML-Dokumenten. Für die manuelle Umwandlung von HTML zu PostScript oder PDF können Sie die Druckfunktionen Ihres Webbrowsers einsetzen. Sie werden aber feststellen, dass selbst das nur bescheidene Ergebnisse liefert, insbesondere was den Seitenumbruch betrifft.

### PostScript ↔ PDF

`ps2pdf` `quelle.ps` `ziel.pdf` erzeugt aus einer beliebigen PostScript-Datei eine PDF-Datei. Das Kommando erfüllt damit im Prinzip dieselbe Funktion wie das kommerzielle Programm Adobe Distiller. Es basiert auf Ghostscript (`gs`). ps2pdf

`ps2pdf` erzeugt momentan Dateien, die zum PDF-Format 1.2 für den Acrobat Reader 3.*n* kompatibel sind. Die Dokumentation weist aber darauf hin, dass sich das Defaultverhalten in Zukunft ändern kann. Wenn Sie die Kompatibilität zu einer bestimmten PDF-Version sicherstellen möchten, sollten Sie die Kommandos `ps2pdf12`, `ps2pdf13` und `ps2pdf14` einsetzen.

Die Qualität der PDF-Dateien hängt stark davon ab, welche Schriftarten im PostScript-Dokument verwendet werden. Bei nicht unterstützten Schriften müssen die Zeichen durch Bitmaps ersetzt werden, was die Darstellungsqualität stark mindert.

Das Verhalten von `ps2pdf` wird durch unzählige Optionen gesteuert. Eine vollständige Dokumentation war zuletzt hier zu finden:

<https://ghostscript.com/doc/current/Ps2pdf.htm>

- pdf2ps** Die Umkehrung zu `ps2pdf` ist `pdf2ps quelle.pdf ziel.ps`. Auch `pdf2ps` greift auf `gs` zurück.
- pdftops** `pdftops` erfüllt zwar prinzipiell dieselbe Aufgabe wie `pdf2ps`, ist intern aber anders implementiert und bietet wesentlich mehr Optionen zur Beeinflussung der resultierenden PostScript-Dateien. Beispielsweise können Sie den gewünschten PostScript-Level, die Papiergröße etc. angeben.
- epstopdf** Wenn Sie aus einer EPS-Abbildung eine PDF-Datei erstellen möchten, bietet sich `epstopdf` an. EPS steht für *Encapsulated PostScript* und bezeichnet PostScript-Dateien, bei denen durch eine sogenannte Bounding Box die exakte Bildgröße angegeben ist. EPS-Dateien eignen sich gut zum Einbetten in andere Dokumente (z. B. mit  $\LaTeX$  oder LibreOffice).

`epstopdf` befindet sich üblicherweise im `tetex`-Paket, das  $\TeX$ ,  $\LaTeX$  und andere  $\TeX$ -spezifische Programme enthält. Im Unterschied zu `ps2pdf` berücksichtigt `epstopdf` die Größe des Bildes. Leider ist `epstopdf` nicht in der Lage, in der EPS-Datei enthaltene Bitmaps unverändert in die PDF-Datei zu übertragen, auch nicht mit der Option `--nocompress`.

### PostScript/PDF → Druckerformat/Bitmap

- gs** Das Kommando `gs`, bekannter unter dem Namen *Ghostscript*, konvertiert PostScript- und PDF-Dokumente in diverse Bitmap- und Druckerformate. Ghostscript ist ein wichtiger Baustein des Linux-Drucksystems (siehe [Kapitel 30](#), »CUPS«), weil es den Ausdruck von PostScript-Dokumenten auf Druckern ohne PostScript-Funktionen ermöglicht. Das Programm wird aber auch von diversen PostScript-Viewern und -Konvertern eingesetzt.

`gs` greift auf die auf dem Rechner installierten Schriftarten sowie auf eine Sammlung eigener Fonts zurück, die sich üblicherweise im Paket `ghostscript-fonts` befinden. Diese Fonts sind erforderlich, um PostScript-Schriften in eine Bitmap-Darstellung umzuwandeln.

Ghostscript ist in verschiedenen Versionen erhältlich. In den meisten Linux-Distributionen kommt GNU Ghostscript oder dessen Variante ESP Ghostscript zum Einsatz. Diese beiden Versionen unterstehen der GPL. ESP Ghostscript ist speziell für die Zusammenarbeit mit CUPS optimiert. ESP steht dabei für den Firmennamen *Easy Software Products*.

Daneben gibt es kommerzielle Ghostscript-Versionen, wie Artifex Ghostscript, die beispielsweise an Druckerhersteller verkauft werden. Weitere Informationen zu den verschiedenen Ghostscript-Versionen finden Sie hier:

<https://ghostscript.com>

<https://artifex.com>

Eine Menge Druckertreiber sind direkt in Ghostscript integriert. Daneben wurden aber diverse Treiber außerhalb des Ghostscript-Projekts entwickelt. Das wichtigste derartige Treiberprojekt ist Gutenprint (ehemals GIMP-Print). Weitere Informationen finden Sie hier:

Externe  
Druckertreiber  
(Gutenprint)

<http://gimp-print.sourceforge.net>

An dieser Stelle ebenfalls erwähnenswert ist HPLIP (HP Linux Imaging and Printing). In diesem Projekt stellt die Firma HP Open-Source-Treiber für viele ihrer Drucker und Scanner zur Verfügung. Das HPLIB-Projekt hat allerdings nichts mit Ghostscript zu tun und wird in Kombination mit dem Drucksystem CUPS genutzt.

Wegen der guten Integration von Ghostscript in das Drucksystem und in diverse andere Programme wird `gs` nur selten manuell eingesetzt. Damit `gs` korrekt funktioniert, müssen mindestens zwei Optionen angegeben werden: `-sOutputFile=` zur Angabe der Datei, in die das Ergebnis geschrieben werden soll, sowie `-sDEVICE=name` oder `@name.upp` zur Einstellung des Ausgabeformats. In der Regel ist es sinnvoll, auch die Option `-dNOPAUSE` zu verwenden. Falls Sie auf DIN-A4-Papier drucken möchten, sollten Sie schließlich noch `-sPAPERSIZE=a4` angeben.

Manueller Aufruf

Die folgende Anweisung übersetzt `test.ps` in das Format des HP-Laserjet 3. Die resultierende Datei `out.hp` kann auf nahezu jedem Laser-Drucker ausgegeben werden, selbst auf sehr alten Modellen.

```
user$ gs -sDEVICE=ljet3 -sOutputFile=out.hp -sPAPERSIZE=a4 \
        -dNOPAUSE -dBATCH test.ps
```

Das zweite Beispiel wandelt eine PostScript- in eine PDF-Datei um. Auch `ps2pdf` ist in Wirklichkeit nichts anderes als ein Script, das `gs` aufruft.

```
user$ gs -dNOPAUSE -dBATCH -sDEVICE=pdfwrite -sOutputFile=out.pdf test.ps
```

Zu guter Letzt sehen Sie hier ein Kommando, das eine EPS-Datei in eine PNG-Datei umwandelt:

```
user$ gs -dNOPAUSE -dBATCh -sDEVICE=png16m -sOutputFile=out.png \
        -dEPSCrop -r100 bild.eps
```

### PostScript-Utilities

**psutils-Paket** Bei der Bearbeitung von PostScript-Dateien helfen die Kommandos des psutils-Pakets (siehe [Tabelle 13.3](#)). Dabei handelt es sich teils um eigenständige Programme, teils um bash- oder Perl-Script-Dateien.

| Kommando    | Funktion  |
|-------------|---|
| epsffit     | passt die Größe einer EPS-Datei an.   |
| extractres  | analysiert die Datei und liefert %IncludeResource-Kommentare für alle benötigten Fonts, Dateien etc.  |
| fixfmps     | passt FrameMaker-Dateien an die psutils-Konventionen an.  |
| fixmacps    | passt Macintosh-Dateien an die psutils-Konventionen an.   |
| fixscribeps | passt Scribe-Dateien an die psutils-Konventionen an.  |
| fixtpps     | passt Troff/Tpscript-Dateien an die psutils-Konventionen an.  |
| fixwfwps    | passt MS-Word-Dateien an die psutils-Konventionen an.   |
| fixwpps     | passt WordPerfect-Dateien an die psutils-Konventionen an.   |
| fixwwps     | passt MS-Write-Dateien an die psutils-Konventionen an.  |
| getafm      | erzeugt AFM-Dateien zur Beschreibung von Fonts.   |
| includeres  | fügt die mit extractres erzeugten Kommentare in eine PostScript-Datei ein.  |
| psbook      | ordnet die Seiten eines Textes so an, dass ganze Bögen (etwa mit je 16 Seiten) gedruckt werden können.  |
| psnup       | ordnet mehrere verkleinerte Seiten auf einem Blatt an.  |
| psresize    | verändert die erforderliche Papiergröße eines Dokuments. Das Kommando löst das regelmäßig auftretende Problem des Ausdrucks von PostScript-Dokumenten, die für das US-Letter-Format erzeugt wurden. |
| psselect    | extrahiert einzelne Seiten aus einer PostScript-Datei.  |
| pstops      | ordnet die Seiten eines Dokuments in einer neuen Reihenfolge.   |

**Tabelle 13.3** psutils-Kommandos

Das folgende Beispiel zeigt, wie eine mit L<sup>A</sup>T<sub>E</sub>X und DVIPS erzeugte PostScript-Datei mit dem Manuskript dieses Buches in eine Darstellung mit 64 Seiten pro Blatt umgewandelt wird. Damit erscheint jede Seite nur noch briefmarkengroß. Das ermöglicht

anschließend in einem PostScript-Viewer eine rasche, übersichtsartige Kontrolle des Seitenlayouts (ähnlich wie die Druckvorschau bei Microsoft Word mit dem kleinstmöglichen Zoomfaktor):

```
user$ psnup -b-0.4cm -64 -q < linux.ps > vorschau.ps
```

Die obigen Kommandos funktionieren nur dann, wenn die PostScript-Dateien DSC-konforme Kommentare enthalten. (DSC steht für *Document Structuring Conventions*.) Die Kommentare werden nicht ausgedruckt, enthalten aber wichtige Informationen über die Größe einer Seite, über den Beginn und das Ende von Seiten etc.) EPS-Dateien sind einseitige PostScript-Dateien, die spezielle Kommentare zur Einbettung in andere Dokumente enthalten (insbesondere Bounding-Box-Angaben über die Größe des Ausdrucks).

Um zwei oder mehrere PostScript-Dateien aneinanderzufügen, setzen Sie am einfachsten das Ghostscript-Kommando `gs` ein. Zufriedenstellende Ergebnisse erzielen Sie allerdings nur dann, wenn `gs` alle Font-Dateien findet.

PS-Dateien  
zusammenfügen

```
user$ gs -sDEVICE=pswrite -sOutputFile=out.ps -dNOPAUSE -dBATCH in1.ps in2.ps ...
```

## PDF-Utilities

Das Kommando `pdftk` (PDF-Toolkit) bietet für PDF-Dokumente ähnliche Funktionen wie `psutils` für PostScript-Dateien. Sie können damit Seiten extrahieren, mehrere PDF-Dokumente zusammenführen, eine unverschlüsselte Version eines verschlüsselten PDF-Dokuments erstellen (vorausgesetzt, Sie kennen das Passwort), PDF-Formulare ausfüllen etc. Ausführliche Informationen finden Sie auf der folgenden Website:

pdftk

<https://pdlabs.com/tools/pdftk-the-pdf-toolkit>

Das folgende Kommando liest die Seiten 10 bis 20 sowie 30 bis 40 aus `in.pdf` und schreibt sie in die neue Datei `out.pdf`:

```
user$ pdftk in.pdf cat 10-20 30-40 output out.pdf
```

Auch um mehrere PDF-Dateien aneinanderzufügen, verwenden Sie das Kommando `cat`:

```
user$ pdftk in1.pdf in2.pdf in3.pdf cat output out.pdf
```

Das folgende Beispiel erzeugt für jede einzelne Seite in `in.pdf` eine eigene PDF-Datei mit dem Namen `pg_N`, wobei `N` die Seitennummer ist. Wenn Sie andere Dateinamen wünschen, müssen Sie eine Zeichenkette in `printf`-Syntax an `output` übergeben, z. B. `output seite-%02d.pdf`.

```
user$ pdftk in.pdf burst
```

Das nächste Beispiel erzeugt eine verschlüsselte PDF-Datei. Die Datei kann zwar ohne das Passwort `xxx` gelesen, nicht aber ausgedruckt oder sonstwie bearbeitet werden. Wenn Sie selbst das Lesen der Datei schützen möchten, verwenden Sie statt `owner_pw` das Kommando `user_pw`.

```
user$ pdftk in.pdf output encrypted.pdf owner_pw xxx
```

- Poppler** Poppler ist eine Sammlung von Kommandos zur Umwandlung von PDF-Dokumenten in andere Formate (Text, Bitmap, PostScript etc.). Poppler wird unter Linux von vielen PDF-Viewern eingesetzt. Das Programm befindet sich üblicherweise im Paket `poppler-utils`.
- xpdf-utils** Das Paket `xpdf-utils` enthält unter anderem die Kommandos `pdftops` (erzeugt PostScript-Dateien aus PDF-Dokumenten), `pdfinfo` (extrahiert die PDF-Dokument-Eigenschaften), `pdfimages` (extrahiert Bilder aus PDF-Dateien) und `pdftotext` (extrahiert den Text aus einer PDF-Datei).
- pdfedit** Das Paket `pdfedit` enthält diverse Werkzeuge und eine Benutzeroberfläche, um PDF-Dateien zu verändern.
- pdfjam** Das Paket `pdfjam` enthält die Kommandos `pdfnup`, `pdfjoin` und `pdf90`. Damit können Sie PDF-Dateien aneinanderfügen und rotieren.
- GUIs** Auch wer Kommandos und ihre Optionen verabscheut und sich stattdessen nach einer grafischen Benutzeroberfläche sehnt, findet eine reiche Auswahl von häufig Java-basierten Open-Source-Programmen. Neben dem schon erwähnten Programm `PDFedit` sind vor allem `PDF-Shuffler`, `Bookbinder`, `JPDF Tweak` sowie `PDF Split and Merge (PDF Sam)` interessant.

## LaTeX & Co.

LaTeX ist ein System zum Setzen (Layouten) wissenschaftlicher Texte. Dieser Abschnitt beschreibt ganz kurz die wichtigsten Kommandos, um LaTeX-Dateien (\*.tex) in andere Formate umzuwandeln, ohne aber auf die LaTeX-Syntax einzugehen.

- latex** Das Kommando `latex name.tex` erzeugt aus der LaTeX-Datei eine DVI-Datei. Diese Datei enthält alle Anweisungen für das Seitenlayout in einer drucker- bzw. device-unabhängigen Sprache.
- dvips** Sobald die DVI-Datei vorliegt, kann sie mit den Programmen `xdvi` oder `kdvi` betrachtet werden. `dvips` wandelt die DVI-Datei in das PostScript-Format um. Das folgende Kommando zeigt die prinzipielle Syntax des Kommandos:

```
user$ dvips [optionen] -o name.ps name.dvi
```

Oft möchte man  $\LaTeX$ -Dokumente als PDF-Datei weitergeben. Dazu gibt es viele Möglichkeiten: dvi2pdf

- ▶ Sie wandeln die  $\LaTeX$ -Datei mit `pdflatex` direkt in eine PDF-Datei um.
- ▶ Sie erzeugen zuerst mit `dvips` eine PostScript-Datei und wandeln diese dann mit `ps2pdf` oder mit dem Adobe Distiller in eine PDF-Datei um. Adobe Distiller ist Teil des kommerziellen Programmpakets Adobe Acrobat, von dem es zurzeit leider keine Linux-Version gibt.
- ▶ Sie wandeln die DVI-Datei mit `dvi2pdf` oder `dvi2pdfm` in eine PDF-Datei um. `dvi2pdf` entspricht dabei dem obigen Punkt, weil als Zwischenschritt ebenfalls eine PostScript-Datei erzeugt wird.

Als Ergebnis erhalten Sie eine PDF-Datei, die wie die äquivalente PostScript-Datei aussieht. Ob auch PDF-Zusatzfunktionen (Inhaltsverzeichnis, anklickbare Links etc.) genutzt werden können, hängt vom Umwandlungsweg und von den im  $\LaTeX$ -Dokument eingesetzten Zusatzpaketen ab:

- ▶ `pdflatex`: Dieses Programm sieht eine Reihe zusätzlicher  $\LaTeX$ -Kommandos vor, um die PDF-Funktionen zu steuern. Sofern Sie nicht `pdflatex`-inkompatible Pakete einsetzen, ist diese Lösung vorzuziehen.
- ▶ `dvips/ps2pdf` bzw. `dvi2pdf`: PDF-Funktionen können durch das  $\LaTeX$ -Paket `hyperref` genutzt werden.
- ▶ `dvi2pdfm`: Hier müssen Sie zusätzliche `\special`-Kommandos in das  $\LaTeX$ -Dokument einfügen.

## 13.6 Markdown und Pandoc

In den letzten Jahren ist das Markdown-Format immer populärer geworden: Es erlaubt das Schreiben formatierter Texte in einer für das menschliche Auge besonders angenehmen Form, also ohne die Fülle von Formatierungscodes, die  $\LaTeX$  und HTML-Code auszeichnen. Das folgende Listing gibt dafür ein Beispiel und bedarf keiner weiteren Erklärung:

Markdown-Syntax

Die Markdown-Syntax  
=====

```
Markdown-Dokumente sind simple Textdateien. Die
Textformatierung erfolgt durch Auszeichnungselemente,
die den Textfluss kaum stören. Ein paar Beispiele:
*Kursiver Text*, **fetter Text**, `Text mit Sonderzeichen`
*in* Listing-Schrift`.
```

Einfache Links werden zwischen ``<`` und ``>`` gestellt:  
`<http://de.wikipedia.org/wiki/Markdown>`.

Alternativ kann auch zwischen dem dargestellten Text und dem Link differenziert werden:  
`[Wikipedia](http://de.wikipedia.org/)`.

`>` Eingerückter Text wird wie in E-Mails mit dem Zeichen ``>>`` eingeleitet. Die Einrückung kann natürlich über mehrere Zeilen reichen.

Aufzählungen

-----

Nicht nummerierte Aufzählungspunkte werden durch Sterne markiert:

- \* Der erste Aufzählungspunkt
- \* Der zweite Aufzählungspunkt
- \* Der dritte Aufzählungspunkt

Listings/Programmcode

-----

Programmlistings müssen im Markdown-Quelltext um vier Zeichen eingerückt werden. Hier können alle Sonderzeichen verwendet werden. `<>()*~!@$%&*'``

**markdown-Editoren** Die meisten besseren Editoren unter Linux unterstützen die Markdown-Syntax und zeigen Markdown-Text mit farbigem Syntax-Highlighting an. Besonders gut gelingt dies den Editoren Atom und VSCode (siehe [Kapitel 17](#)). Als Dateikennung für Markdown-Dokumente ist `*.md` üblich.

**markdown-Kommando** Der Markdown-Erfinder John Gruber hat das Perl-Script `markdown` entwickelt, um Markdown-Texte in das HTML-Format umzuwandeln (siehe [Abbildung 13.1](#)). Das Kommando steht bei vielen Linux-Distributionen im gleichnamigen Paket zur Verfügung.

```
user$ markdown input.md > output.html
```

Die optische Gestaltung der resultierenden HTML-Seite können Sie bei Bedarf mit einer ergänzenden CSS-Datei steuern.

**Pandoc** Die Markdown-Idee wurde von vielen anderen Entwicklern aufgegriffen. Mittlerweile gibt es eine ganze Fülle von Markdown-ähnlichen Formaten und Konvertierungswerkzeugen. Am weitesten geht das Pandoc-Projekt:



- ▶ Das Kommando `pandoc` unterstützt alle gängigen Markdown-Dialekte.
- ▶ Es kann Markdown-Texte nicht nur in das HTML-Format umwandeln, sondern auch in diverse andere Formate, darunter PDF,  $\text{\LaTeX}$ , EPUB, Microsoft Word und LibreOffice Writer.
- ▶ Mit erheblichen Einschränkungen ist sogar eine Konvertierung in die umgekehrte Richtung möglich. Sie können also z. B. aus einer  $\text{\LaTeX}$ -Datei eine Markdown-Datei machen.



Abbildung 13.1 Dieses HTML-Dokument wurde aus dem vorhin abgedruckten Markdown-Code erstellt.

Viele Distributionen stellen Pandoc im gleichnamigen Paket zur Verfügung. Sollte das bei Ihrer Distribution nicht der Fall sein, müssen Sie Pandoc von der Projektwebseite herunterladen und installieren:

*<https://pandoc.org/installing.html>*

Als Autor hat mich die Markdown-Syntax in Kombination mit Pandoc vollständig überzeugt. Ich verfasse seit Jahren alle neuen Bücher und eBooks, Unterrichtsfolien und Webbeiträge in der Markdown-Syntax. (Dieses Buch ist allerdings eine Ausnahme: Seit ich die erste Auflage vor mehr als zwanzig Jahren in  $\text{\LaTeX}$  verfasst habe, gibt es kein Zurück mehr. Eine automatisierte Umwandlung in andere Formate scheitert an den vielen  $\text{\LaTeX}$ -Eigenheiten, die das Manuskript nutzt.)



# Kapitel 14

## Netzwerk-Tools

Dieses Kapitel stellt Kommandos zur Benutzung, Steuerung und Analyse elementarer Netzwerkdienste vor. Sie lernen hier, wie Sie sich mit `ssh` auf einem anderen Rechner im Netzwerk einloggen und mit `wget` Dateien übertragen. Mit den Programmen `Lynx` und `Mutt` können Sie im Textmodus sogar Webseiten besuchen und Mails lesen und verfassen.

Weitere Kommandos zur Analyse des Netzwerkstatus sowie zur Suche nach offenen Ports auf fremden Rechnern, `netstat` und `nmap`, stelle ich Ihnen in [Kapitel 37](#), »Firewalls«, vor.

### 14.1 Netzwerkstatus ermitteln

Dieser Abschnitt gibt einen Überblick über Kommandos zum Test der Grundfunktionen des Netzwerks. Weitere Informationen zu den hier vorgestellten Kommandos folgen in [Abschnitt 26.4](#), »Manuelle LAN- und WLAN-Konfiguration«.

Das Kommando `ip addr` liefert eine Liste aller bekannten Netzwerkschnittstellen:

**Netzwerk-  
schnittstellen  
ermitteln**

```
root# ip addr
ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:1c:42:55:4f:0e brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.42/24 brd 10.0.0.255 scope global eth0
    inet6 fe80::21c:42ff:fe55:4f0e/64 scope link
        valid_lft forever preferred_lft forever
```

Typische Schnittstellen sind `ethN` oder `enpNsM` (Ethernet), `wlanN` (WLAN) und `pppN` (Internetzugang via UMTS-Modem, ADSL oder VPN). Bei den meisten gängigen Distributionen fließt in den Namen der Ethernet-Schnittstelle die interne Bus-Nummer ein – z. B. `enp0s1`. Auf PCs und Notebooks mit nur einer Ethernet-Schnittstelle wirkt der

Name umständlich. Aber die bus-spezifische Nummerierung stellt sicher, dass sich auf Servern mit vielen Netzwerkschnittstellen die Nummerierung auch dann nicht ändert, wenn weitere Netzwerkadapter hinzugefügt werden.

Eine Sonderrolle nimmt die Schnittstelle `lo` ein: Sie ermöglicht es lokalen Programmen, über das Netzwerkprotokoll zu kommunizieren. Das funktioniert selbst dann, wenn ein Rechner nicht nach außen hin mit einem Netzwerk verbunden ist.

Wenn `ip addr` nur bei der Schnittstelle `lo` eine IP-Adresse angibt, wurde noch keine Netzwerkschnittstelle aktiviert. Abhilfe schafft das von Ihrer Distribution vorgesehene Werkzeug zur Netzwerkkonfiguration. Sie können die Netzwerkschnittstelle mit dem `ip`-Kommando auch manuell aktivieren. Details dazu sowie zur IPv6-Konfiguration finden Sie in [Kapitel 26](#), »Netzwerkkonfiguration«.

#### Erreichbarkeit von localhost testen

`ping` sendet einmal pro Sekunde ein kleines Netzwerkpaket an die angegebene Adresse. Wenn sich dort ein Rechner befindet, sendet dieser eine Antwort, es sei denn, eine Firewall verhindert das. `ping` läuft so lange, bis es mit `Strg+C` beendet wird. `ping localhost` überprüft, ob das Loopback-Interface und damit die elementaren Netzwerkfunktionen des eigenen Rechners funktionieren:

```
user$ ping localhost
PING localhost (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=255 time=0.152 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=255 time=0.114 ms
...
```

#### Erreichbarkeit des lokalen Netzes testen

Indem Sie an `ping` statt `localhost` die IP-Nummer eines anderen Rechners im lokalen Netz übergeben, testen Sie, ob das lokale Netz funktioniert. `-c 2` bewirkt, dass `ping` nicht endlos läuft, sondern nach zwei Paketen endet:

```
user$ ping -c 2 192.168.0.99
PING 192.168.0.99 (192.168.0.99): 56 data bytes
64 bytes from 192.168.0.99: icmp_seq=0 ttl=255 time=0.274 ms
64 bytes from 192.168.0.99: icmp_seq=1 ttl=255 time=0.150 ms
...
```

Wenn es im lokalen Netz einen Nameserver gibt, der der IP-Nummer `192.168.0.99` einen Namen zuordnet, oder wenn die Datei `/etc/hosts` diese Aufgabe übernimmt, können Sie bei `ping` statt der IP-Nummer den Rechnernamen angeben:

```
user$ ping -c 2 mars
PING mars.sol (192.168.0.99) 56(84) bytes of data.
64 bytes from mars.sol (192.168.0.99): icmp_seq=1 ttl=64 time=0.281 ms
64 bytes from mars.sol (192.168.0.99): icmp_seq=2 ttl=64 time=0.287 ms

--- mars.sol ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.281/0.284/0.287/0.003 ms
```

Als Nächstes können Sie testen, ob die Verbindung zum Internet gelingt. Das folgende Kommando testet gleichzeitig zwei Aspekte der Netzwerkkonfiguration: die Erreichbarkeit des Nameservers und die Funktion des Gateways.

Internetzugang  
testen

```
user$ ping -c 2 www.yahoo.com
PING www.yahoo-ht2.akadns.net (209.73.186.238) 56(84) bytes of data.
64 bytes from f1.www.vip.re3.yahoo.com (209.73.186.238): icmp_seq=1 time=122 ms
64 bytes from f1.www.vip.re3.yahoo.com (209.73.186.238): icmp_seq=2 time=123 ms

--- www.yahoo-ht2.akadns.net ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 122.731/123.256/123.782/0.631 ms
```

Wenn das nicht funktioniert, sind mehrere Ursachen denkbar:

- ▶ Vielleicht ist der Server von Yahoo gerade unerreichbar, oder der Server hat aus Sicherheitsgründen die Antwort auf ping deaktiviert. Probieren Sie eine andere bekannte Internetadresse aus.
- ▶ Für die Ermittlung der IP-Adresse zu yahoo.com ist der Nameserver verantwortlich. Wenn Sie die Fehlermeldung *unknown host yahoo.com* erhalten, gibt es Probleme mit dem Nameserver. Überprüfen Sie, ob `/etc/resolv.conf` dessen Adresse enthält.
- ▶ Das Gateway ist dafür zuständig, IP-Pakete aus dem lokalen Netzwerk an das Internet weiterzuleiten. Wenn das nicht funktioniert, erhalten Sie die Fehlermeldung *connect: Network is unreachable*. Die Gateway-Konfiguration können Sie mit `ip route` überprüfen. Das Kommando liefert normalerweise mehrere Zeilen. Die Gateway-Adresse befindet sich in der dritten Spalte der Zeile, die mit `default` beginnt:

```
user$ ip route
default via 10.0.0.138 dev eth0
10.0.0.0/24 dev eth0 proto kernel scope link src 10.0.0.42
```

- ▶ Falls Sie in einem lokalen Netz einen eigenen Rechner als Gateway eingerichtet haben, besteht die Möglichkeit, dass Sie die Masquerading-Funktion vergessen haben. In diesem Fall würde der Internetzugang für das gesamte lokale Netzwerk nicht funktionieren. Eine detaillierte Anleitung zur Konfiguration eines eigenen LAN- oder WLAN-Routers finden Sie in [Kapitel 27](#), »Internet-Gateway«.

Mit `traceroute` finden Sie heraus, welchen Weg ein Netzwerkpaket von Ihrem Rechner zu einem anderen Rechner nimmt und wie viele Millisekunden die Laufzeit bis zur jeweiligen Zwischenstation beträgt. Standardmäßig unternimmt das Kommando drei Versuche und liefert daher entsprechend drei Zeiten. Das Kommando funktioniert nicht, wenn sich auf einer der Zwischenstationen eine Firewall befindet, die den von `traceroute` genutzten UDP-Port 33434 blockiert. In diesem Fall liefert `traceroute` für diese und alle weiteren Stationen nur noch drei Sterne.

Den Weg von  
IP-Paketen  
verfolgen

Die folgenden Zeilen zeigen den Weg von meinem Arbeitsrechner zu google.at. Zeile 1 beschreibt mein Internet-Gateway (den Rechner mars.sol), Zeile 2 den ADSL-Router und Zeile 3 das Gateway meines Internet-Providers.

```
user$ traceroute google.at
traceroute to google.at (66.102.9.104), 30 hops max, 40 byte packets
 1 mars.sol.0.168.192.in-addr.arpa (192.168.0.1) 0.277 ms ...
 2 192.168.1.1 (192.168.1.1) 0.373 ms ...
 3 N704P030.adsl.highway.telekom.at (62.47.31.254) 8.598 ms ...
 4 172.19.90.193 (172.19.90.193) 11.864 ms ...
 ...
14 66.102.9.104 (66.102.9.104) 52.741 ms ...
```

### Firewalls umgehen

Mitunter behindern Sicherheitseinstellungen und Firewalls die Arbeit von traceroute. Anstelle von IP-Adressen zeigt das Kommando dann nur \* \* \* an. In solchen Fällen können Sie versuchen, mit den Optionen -T oder -I andere Verfahren zu verwenden, um den Weg von Paketen zu verfolgen. Beide Optionen erfordern root-Rechte.

**mtr** Das Kommando `mtr` sendet regelmäßig Netzwerkpakete zum angegebenen Host und analysiert die Antworten. Die Ergebnisliste kombiniert Daten von ping und traceroute. Beachten Sie, dass es zwei Versionen dieses Programms gibt: das hier beschriebene Textkommando sowie eine Variante mit grafischer Benutzeroberfläche. Bei Desktop-Installationen von Debian und Ubuntu ist standardmäßig die GTK-Variante installiert. Um stattdessen die Textversion zu installieren, führen Sie `apt-get install mtr-tiny` aus.

```
user$ mtr -c 10 -r google.de
HOST: michael's-computer      Loss%  Snt  Last  Avg  Best  Wrst StDev
 1 |-- speedtouch.lan         0.0%   10  42.6  48.5   6.0  95.9  28.9
 2 |-- 178-191-207-254.adsl.high 0.0%   10  18.9  20.4  18.6  23.2   1.9
 3 |-- 195.3.74.129           0.0%   10  19.4  18.8  17.9  19.4   0.5
 4 |-- AUX10-GRAZBC10.highway.te 0.0%   10  21.2  21.3  20.7  22.0   0.3
 5 |-- 195.3.70.154           0.0%   10  21.2  27.3  20.9  81.2  18.9
 6 |-- 62.47.120.150          0.0%   10  25.3  25.5  24.9  26.0   0.4
 7 |-- 209.85.243.119         0.0%   10  25.6  25.9  25.2  28.2   0.8
 8 |-- 216.239.46.88          0.0%   10  25.8  26.3  25.8  27.7   0.6
 9 |-- bud01s08-in-f23.1e100.net 0.0%   10  25.7  25.8  25.0  26.8   0.5
```

**gnome-nettool** Wer unter Gnome arbeitet, kann einen Großteil der oben aufgezählten Informationen ganz komfortabel mit dem Programm `gnome-nettool` ermitteln (siehe [Abbildung 14.1](#)). Bei einigen Distributionen steht das Programm aus Platzgründen standardmäßig nicht zur Verfügung – dann müssen Sie das gleichnamige Paket zuerst installieren.

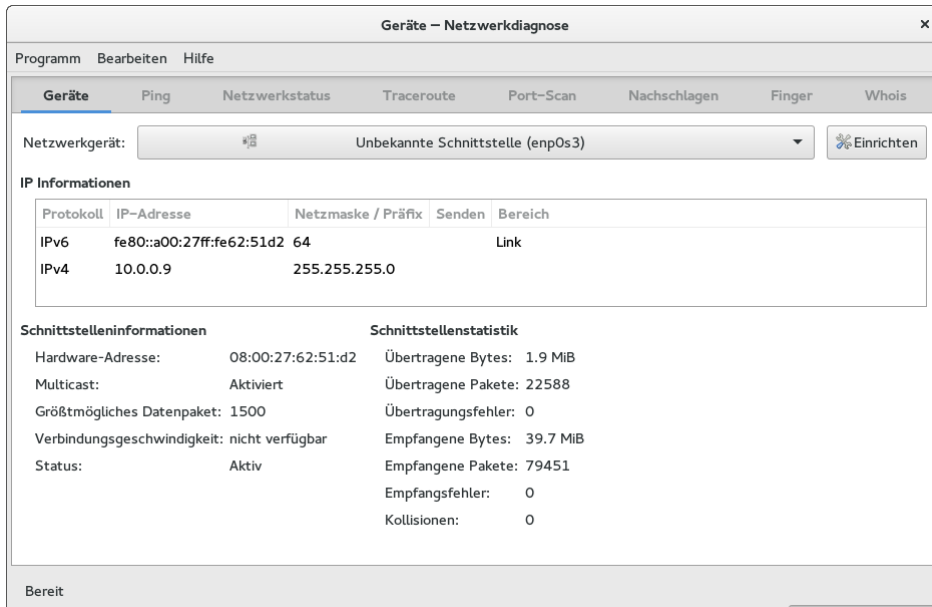


Abbildung 14.1 Netzwerkdiagnose unter Gnome

## 14.2 Auf anderen Rechnern arbeiten (SSH)

Die Programme `telnet`, `rlogin` und `ssh` ermöglichen es, so auf einem anderen Rechner zu arbeiten, als stünde er vor Ihnen. Das funktioniert sowohl für kommandoorientierte Programme als auch für X-Programme. Dieser Abschnitt beschränkt sich auf die Beschreibung von `ssh` (Secure Shell). Die älteren Programme `telnet` und `rlogin` sollten aus Sicherheitsgründen nicht mehr eingesetzt werden. Sie übertragen die Login-Informationen inklusive des Passworts unverschlüsselt.

Die Grundvoraussetzung für die Anwendung von `ssh` besteht darin, dass auf dem zweiten Rechner ein SSH-Server läuft, also das Programm `sshd`. Bei manchen Linux-Distributionen ist dies standardmäßig der Fall, bei anderen muss das Programm (zumeist als Paket `openssh-server`) zuerst installiert werden. Wenn auf den Rechnern Firewalls laufen, dürfen diese den Port 22 nicht blockieren.

### Einigen eigenen SSH-Server einrichten

Informationen zur Installation, Konfiguration und Absicherung eines SSH-Servers folgen in [Kapitel 31, »Secure Shell \(SSH\)«](#). Dort erfahren Sie auch, wie Sie den SSH-Server absichern.

**Gewöhnliche Shell-Session**

Wenn Sie auf dem Rechner `uranus` arbeiten und nun eine Shell-Session auf dem Rechner `mars` starten möchten, führen Sie zum Verbindungsaufbau das folgende Kommando aus:

```
user@uranus$ ssh mars
user@mars's password: *****
```

Beim ersten Verbindungsaufbau zu einem neuen Rechner erscheint eine Warnung nach dem folgenden Muster:

```
The authenticity of host 'mars (192.168.0.10)' can't be established.
RSA1 key fingerprint is 1e:0e:15:ad:6f:64:88:60:ec:21:f1:4b:b7:68:f4:32.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'mars,192.168.0.10' (RSA1) to the list
of known hosts.
```

Das bedeutet, dass `ssh` sich nicht sicher ist, ob es dem Rechner `mars` mit der IP-Adresse `192.168.0.10` vertrauen darf. Es könnte sein, dass ein fremder Rechner vortäuscht, `mars` zu sein. Wenn Sie die Rückfrage mit `YES` beantworten, speichert `ssh` den Namen, die Adresse und den RSA-Fingerprint (einen Code zur eindeutigen Identifizierung des Partnerrechners) in `~/.ssh/known_hosts`.

Falls Sie auf `mars` unter einem anderen Login-Namen als auf `uranus` arbeiten möchten (z. B. als `root`), geben Sie den Namen mit der Option `-l` an:

```
user@uranus$ ssh -l root mars
root@mars's password: *****
```

### SSH-Authentifizierung mit Schlüsseln

Wesentlich sicherer als ein Login mit Passwort ist die Authentifizierung durch einen Schlüssel. Die Vorgehensweise wird im Detail in [Abschnitt 31.4](#) beschrieben. Die Verwendung von Schlüsseln ermöglicht es auch, auf SSH basierende Kommandos und Scripts automatisch per Script auszuführen.

**Kommandos ausführen**

Statt `ssh` interaktiv zu nutzen, können Sie auf dem entfernten Rechner auch einfach nur ein Kommando ausführen. Das Kommando und seine Parameter werden einfach als weitere Parameter an `ssh` übergeben. `ssh` endet nach diesem Kommando.

```
user@uranus$ ssh mars kommando optionen
user@mars's password: *****
```

Aus dieser scheinbar trivialen Funktion ergeben sich weitreichende Möglichkeiten: Sie können nun beispielsweise auf dem entfernten Rechner `tar` starten, das damit erstellte Archiv an die Standardausgabe weiterleiten (geben Sie dazu einen Bindestrich - nach der Option `-f` ein, also `-f -`) und die Standardausgabe mit `|` als Eingabe



für ein zweites tar-Kommando verwenden, das lokal läuft. Damit können Sie einen ganzen Verzeichnisbaum sicher via SSH kopieren.

Das folgende Kommando zeigt, wie ich den gesamten `/var/www`-Verzeichnisbaum meines Webservers `kofler.info` in das lokale Verzeichnis `~/bak` kopiere. Das Kommando setzt dabei voraus, dass alle Dateien in `/var/www` vom Benutzer `username` gelesen werden können.

```
user$ ssh -l username kofler.info tar -cf - /var/www | tar -xC ~/bak/ -f -
username@kofler.info's password: *****
```

Wenn Sie in einem Script mehrere Kommandos via SSH ausführen möchten, verwenden Sie am besten die Heredoc-Syntax (siehe auch [Abschnitt 10.11](#), »Code-Strukturierung in bash-Scripts«):

Mehrere Kommandos ausführen (Scripts)

```
#!/bin/bash
pw=strengeheim
...
ssh -T root@host <<ENDSSH
echo root:$pw | chpasswd
rm -f /etc/file1
cp /root/file2 /userxy/file3
ENDSSH
...
```

Damit führt `ssh` alle Kommandos aus, bis im Script die mit `ENDSSH` markierte Zeile erreicht wird. Die Option `-T` verhindert dabei, dass SSH versucht, ein Pseudo-Terminal zu öffnen. Das ist hier unerwünscht, weil die Kommandoausführung nicht interaktiv erfolgen soll.

Beim ersten SSH-Verbindungsaufbau zu einem neuen Host fragt `ssh`, ob Sie dem Host vertrauen. Normalerweise ist diese Rückfrage sinnvoll. Wenn Sie aber mit `ssh` automatisiert auf mehreren Hosts (oft in virtuellen Maschinen) Arbeiten durchführen möchten, stört die Rückfrage. Abhilfe schafft in solchen Fällen die Option `-o StrictHostKeyChecking=no`.

Rückfrage bei erstem Verbindungsaufbau verhindern

Diese und andere Optionen können Sie auch global in `/etc/ssh/ssh_config` oder individuell für einen Benutzer in `.ssh/config` einstellen. Verwechseln Sie aber `/etc/ssh/ssh_config` nicht mit `/etc/ssh/sshd_config`! Die erste Datei enthält SSH-Client-Optionen, die zweite Datei Optionen für den SSH-Server.

Sofern als Grafiksystem X und nicht Wayland verwendet wird (sowohl auf dem Client als auch auf dem Server!), können Sie in einer SSH-Verbindung, die Sie mit `ssh -X` initiiert haben, auch Grafikprogramme ausführen. Die Option `-X` ist erforderlich, damit sich `ssh` um die korrekte Einstellung der `DISPLAY`-Variablen kümmert.

SSH und X

```
user@localhost$ ssh -X otheruser@otherhost
otheruser@otherhost$ firefox & (Firefox läuft extern, wird aber lokal angezeigt)
```

**Dateien sicher kopieren mit scp**

Um eine Datei via SSH über das Netzwerk zu kopieren, gibt es das Kommando `scp`. Die Syntax sieht so aus:

```
user$ scp [[user1@]host1:]filename1 [[user2@]host2:][filename2]
user2@host2's password: *****
```

Damit wird die Datei `filename1` vom Rechner `host1` zum Rechner `host2` übertragen und dort in der Datei `filename2` gespeichert. Einige Anmerkungen zu den vielen optionalen Bestandteilen der Kopieranweisung:

- ▶ `host1` und `host2` müssen nicht angegeben werden, wenn der lokale Rechner (also `localhost`) gemeint ist.
- ▶ `user1` muss nicht angegeben werden, wenn der aktive Benutzer gemeint ist.
- ▶ `user2` muss nicht angegeben werden, wenn auf dem Rechner `host2` der aktuelle Benutzername von `host1` bzw. `user1` verwendet werden soll.
- ▶ `filename1` darf auch ein Verzeichnis sein. Sie müssen dann die Option `-r` angeben, damit das gesamte Verzeichnis mit allen Unterverzeichnissen übertragen wird.
- ▶ `filename2` muss nicht angegeben werden, wenn der Dateiname unverändert bleiben soll. Die Datei wird dann in das Home-Verzeichnis von `user2` kopiert.

Statt `filename2` kann auch das Zielverzeichnis angegeben werden, wobei wie üblich `~` für das Home-Verzeichnis von `user2` verwendet wird.

Zum Abschluss noch ein Beispiel: Nehmen Sie an, die Benutzerin `gabi` arbeitet auf dem Rechner `uranus`. Sie will die Datei `abc.txt` in das Verzeichnis `~/efg` auf dem Rechner `mars` übertragen. Das `scp`-Kommando sieht so aus:

```
gabi@uranus$ scp abc.txt mars:~/efg/
gabi@mars's password: *****
```

Falls Sie beim `scp`-Kommando eine IPv6-Adresse angeben wollen, müssen Sie diese in eckige Klammern stellen. Andernfalls kommt `scp` bei den vielen Doppelpunkten durcheinander.

```
user$ scp kofler@[2001:1234:5678::1]:datei.txt .
```

**SFTP** SFTP (*Secure FTP*) ist eine auf SSH basierende sichere Variante zum Protokoll FTP. Details zu SFTP folgen im nächsten Abschnitt, der die Übertragung von Dateien via FTP und HTTP zum Thema hat.

**SSH-Tunnel** Eine SSH-Anwendungsmöglichkeit für fortgeschrittene Linux-Anwender ist der Tunnelbau. Derartige Tunnel eignen sich zwar nicht als Transportmöglichkeit für Autos oder Züge, sie ermöglichen aber die Übertragung aller IP-Pakete, die an einen bestimmten Port gerichtet sind. SSH-Tunnel bieten damit einen sicheren Weg, um IP-Pakete zwischen zwei Rechnern zu übertragen – und das selbst dann, wenn sich

zwischen den beiden Rechnern eine Firewall befindet, die den Port eigentlich blockiert. Eine Einführung in die Welt der IP-Pakete und eine Erklärung des Begriffs *Port* finden Sie in [Kapitel 37](#), »Firewalls«.

Wenn der Tunnelbau vom Client-Rechner aus erfolgt, kommt die Option `-L localhost:remoteport` zum Einsatz. Beispielsweise bewirkt das folgende Kommando, dass der Port 3306 des Rechners `mars` über den Port 3307 des lokalen Rechners zugänglich ist. Durch das Kommando wird gleichzeitig eine SSH-Session gestartet, was Sie durch `-N` aber verhindern können (wenn Sie nur den Tunnel, aber keine Shell benötigen). Falls der Login bei `mars` unter einem anderen Namen erfolgen soll, müssen Sie den Login-Namen wie üblich durch `-l name` oder durch `name@remotehost` angeben.

```
user@uranus$ ssh -L 3307:localhost:3306 username@mars
user@mars's password: *****
```

Der Tunnel bleibt so lange offen, bis die SSH-Session mit `[Strg]+[D]` beendet wird. Falls Sie `ssh` mit der Option `-N` gestartet haben, muss das Programm mit `[Strg]+[C]` gestoppt werden.

3306 ist der übliche Port von MySQL. Sie können nun auf dem Rechner `uranus` über dessen Port 3307 auf den MySQL-Server zugreifen, der auf `mars` läuft. Beim `mysql`-Kommando müssen Sie den Port 3307 und den Hostname `127.0.0.1` angeben, damit der SSH-Tunnel tatsächlich benutzt wird. Standardmäßig stellt `mysql` lokale Verbindungen über eine Socket-Datei her.

```
user@uranus$ mysql -u mysqllogin -P 3307 -h 127.0.0.1 -p
Enter password: *****
```

Damit der MySQL-Login funktioniert, müssen zwei Voraussetzungen erfüllt sein:

- ▶ Erstens muss der MySQL-Server auf dem Rechner `mars` grundsätzlich IP-Verbindungen akzeptieren. Der MySQL-Server kann aus Sicherheitsgründen auch so konfiguriert sein, dass Verbindungen nur über eine Socket-Datei möglich sind. Dann hilft ein Tunnel nicht weiter, weil ein Tunnel nur Ports verbinden kann.
- ▶ Zweitens muss der MySQL-Server die Kombination aus Login-Name und Hostname akzeptieren. Als Hostname wird der Name des Rechners verwendet, zu dem `ssh` den Tunnel errichtet hat – hier also `mars` bzw. `mars.sol`, wenn die Domain `sol` lautet.

Es gibt noch weit mehr und oft viel komplexere Anwendungsmöglichkeiten für SSH-Tunnel. Beispielsweise können Sie die Tunnel dazu verwenden, um ein Virtual Private Network zu bilden. Weiterführende Dokumentation finden Sie z. B. hier: VPN

<http://www.tldp.org/HOWTO/VPN-HOWTO>

**SSH-Dateisystem** Mit dem Kommando `sshfs`, das sich bei vielen Distributionen im gleichnamigen Paket befindet, können Sie das Dateisystem eines externen Rechners in den lokalen Verzeichnisbaum integrieren. Das kann beispielsweise die Durchführung von Backups vereinfachen.

```
root# mkdir /media/ext-host
root# sshfs user@hostname /media/ext-host
root# ...
root# umount /media/ext-host
```

Beachten Sie aber, dass Sie im SSH-Dateisystem wegen der Verschlüsselung aller Daten zumeist einen geringeren Durchsatz als mit Samba oder NFS erzielen werden. Das SSH-Dateisystem ist deswegen für den Einsatz in lokalen Netzwerken nur bedingt geeignet. Ich habe zudem die Erfahrung gemacht, dass `sshfs` auf kurzzeitige Netzwerk-ausfälle allergisch reagiert und dann hängen bleibt. Ich bin deswegen vom Einsatz dieses an sich praktischen Dateisystems wieder abgekommen.

## telnet

Ein Vorgänger von `ssh` war `telnet`. Da `telnet` keine Daten verschlüsselt, sollte das Kommando auf keinen Fall dazu verwendet werden, um auf externen Rechnern zu arbeiten. Aktuelle Linux-Distributionen lassen dies standardmäßig ohnedies nicht zu, aber man stößt immer wieder auf Router, ADSL-Modems etc., die diese Art der Kommunikation zulassen.

Der Grund, warum ich Ihnen hier `telnet` überhaupt präsentiere, ist ein anderer: `telnet` eignet sich gut dazu, um zu überprüfen, ob auf einem externen Rechner auf einem bestimmten Port ein Netzwerkdienst läuft und auf einen Verbindungsaufbau wartet. Beispielsweise können Sie mit `telnet` sicherstellen, dass der zuvor eingerichtete Mail-Server tatsächlich läuft. Dazu übergeben Sie an `telnet` den Namen oder die IP-Adresse des Servers sowie die Port-Nummer:

```
user$ telnet kofler.info 25
Trying 5.9.22.29...
Connected to kofler.info.
Escape character is '^]'.
220 kofler.info ESMTP Postfix (Ubuntu)
  helo kofler.info
250 kofler.info
^] (Verbindung mit Strg+] beenden)
```

## 14.3 Dateien übertragen (FTP)

FTP steht für *File Transfer Protocol* und bezeichnet ein recht altes Verfahren zur Übertragung von Dateien über ein Netzwerk. Seine große Popularität verdankt FTP der Spielart Anonymous FTP: Viele große Internet-Server bieten allen Anwendern Zugang zu sogenannten FTP-Archiven. Dieser Zugang ist (im Gegensatz zum sonstigen FTP) nicht durch ein Passwort versperrt.

Grundlagen

Ein großer Nachteil von FTP besteht darin, dass beim Login-Prozess der Benutzername und das Passwort unverschlüsselt übertragen werden. Eine sichere Alternative ist SFTP (Secure FTP) auf der Basis von SSH (siehe [Kapitel 31](#), »Secure Shell (SSH)«). Auch HTTP, also das Protokoll zur Übertragung von Webseiten, wird oft als Alternative zu FTP eingesetzt.

In diesem Kapitel geht es nur um die Nutzung von FTP, also um die Client-Sichtweise. Damit FTP funktioniert, muss auf der Gegenstelle ein FTP-Server laufen. Dessen Konfiguration ist in [Abschnitt 32.8](#), »FTP-Server (vsftpd)«, beschrieben.

Der Urahn aller FTP-Clients ist das interaktive Textkommando `ftp`. Da es Dateien normalerweise aus dem aktuellen Verzeichnis bzw. in das aktuelle Verzeichnis überträgt, sollten Sie vor dem Start von `ftp` mit `cd` in das gewünschte Arbeitsverzeichnis wechseln. Die FTP-Sitzung wird dann mit dem Kommando `ftp user@ftpservername` oder einfach `ftp ftpservername` eingeleitet. Falls Sie Anonymous FTP nutzen möchten, geben Sie als Benutzernamen `anonymous` ein.

FTP-Kommando

Nach dem Verbindungsaufbau und der Eingabe des Passworts kann es losgehen: Mit den Kommandos `cd`, `pwd` und `ls`, die dieselbe Bedeutung wie unter Linux haben, können Sie sich durch die Verzeichnisse des FTP-Archivs bewegen. Um eine Datei vom FTP-Archiv in das aktuelle Verzeichnis Ihres Rechners zu übertragen, führen Sie `get datei` aus. Der Dateiname bleibt dabei unverändert.

Umgekehrt können Sie mit `put` eine Datei aus Ihrem aktuellen Verzeichnis in ein Verzeichnis des FTP-Archivs übertragen. Das geht freilich nur dann, wenn Sie eine Schreiberlaubnis für das Verzeichnis haben. Bei Anonymous FTP ist das zumeist nur für ein Verzeichnis mit einem Namen wie `/pub/incoming` der Fall. Die FTP-Sitzung wird mit dem Kommando `quit` oder `bye` beendet. Eine Referenz der wichtigsten FTP-Kommandos finden Sie in [Tabelle 14.1](#).

### Text- versus Binärmodus

Bevor Sie eine Datei übertragen, müssen Sie mit `binary` in den Binärmodus umschalten. Im Textmodus interpretiert FTP die Dateien als Texte und versucht, diese in das Format des jeweiligen Rechners zu konvertieren. Binärdateien werden durch so eine Konvertierung unbrauchbar. Die meisten FTP-Server sind glücklicherweise so konfiguriert, dass `binary` als Grundeinstellung gilt.

| Kommando      | Funktion   |
|---------------|--|
| ?             | zeigt eine Liste aller FTP-Kommandos an.   |
| !             | ermöglicht die Ausführung von Shell-Kommandos.   |
| ascii         | wechselt in den Textmodus.   |
| binary        | wechselt in den Binärmodus.  |
| bye           | beendet FTP.   |
| cd verz       | wechselt in das angegebene FTP-Verzeichnis.  |
| close         | beendet die Verbindung zum FTP-Server.   |
| get datei     | überträgt die Datei vom FTP-Archiv in das aktuelle Verzeichnis.                                  |
| help kommando | zeigt eine kurze Info zum angegebenen Kommando an.   |
| lcd verz      | wechselt das aktuelle Verzeichnis auf dem lokalen Rechner.                                       |
| ls            | zeigt die Liste der Dateien auf dem FTP-Server an.   |
| lls           | zeigt die Liste der Dateien auf dem lokalen Rechner an.  |
| mget *.muster | überträgt alle passenden Dateien vom FTP-Archiv in das aktuelle Verzeichnis (siehe auch prompt). |
| open          | stellt die Verbindung zum fremden Rechner her (wenn es beim ersten Versuch nicht geklappt hat).  |
| prompt        | aktiviert/deaktiviert die automatische Rückfrage vor der Übertragung jeder Datei durch mget.     |
| put datei     | überträgt die Datei in das FTP-Archiv ( <i>upload</i> ).   |
| quit          | beendet FTP.   |
| reget datei   | setzt die Übertragung einer bereits teilweise übertragenen Datei fort.                           |
| user          | ermöglicht einen neuen Login.  |

Tabelle 14.1 ftp-Kommandos

#### Andere FTP-Programme

Das Kommando `ftp` ist nicht komfortabel zu bedienen. Zum Glück gibt es unzählige Alternativen:

- ▶ Webbrowser, Dateimanager: Alle unter Linux verfügbaren Webbrowser und Dateimanager können auch zum FTP-Download verwendet werden. Manche Programme ermöglichen sogar einen komfortablen Upload.
- ▶ Grafische FTP-Clients: Programme wie `gftp` (Gnome) sind speziell für typische FTP-Aufgaben optimiert. Sie bieten Spezialfunktionen wie Bookmark- und Passwortverwaltung, die parallele Übertragung mehrerer Dateien, die Synchronisation von Verzeichnissen etc.

- ▶ `ncftp`: Diese Alternative zu `ftp` hat zwar eine textbasierte Benutzeroberfläche, ist aber komfortabler als das Original zu bedienen.
- ▶ `sftp`: Dieses Programm ist ähnlich minimalistisch wie `ftp`, aber dafür deutlich sicherer. Allerdings muss an der Gegenstelle ein SSH-Server laufen (kein FTP-Server). `sftp` wird im folgenden Abschnitt beschrieben.
- ▶ `wget`, `curl`, `lftp`: Diese Kommandos helfen bei der automatisierten Übertragung von Dateien bzw. ganzer Verzeichnisbäume via FTP.

Wenn Sie das Protokoll FTP nicht als Benutzer `anonymous` nutzen möchten, sondern sich mit Name und Passwort anmelden können, gilt bei den meisten FTP-Clients die folgende Syntax:

FTP-Adresse mit  
Passwort

```
ftp://benutzername:password@servername
```

Manche FTP-Clients funktionieren nicht richtig, wenn sich zwischen Ihrem Rechner und dem FTP-Server eine Firewall befindet oder wenn Sie in einem lokalen Netzwerk arbeiten, das mittels Masquerading mit dem Internet verbunden ist. In solchen Fällen hilft es fast immer, den Client in einen sogenannten passiven Modus zu versetzen. Leider gibt es dafür kein einheitliches Kommando – werfen Sie also einen Blick in die Dokumentation! Die meisten Clients erkennen derartige Situationen selbstständig und aktivieren den passiven Modus automatisch.

Passiver Modus

### SFTP (Secure FTP)

Das Kommando `sftp` ist Teil des `openssh`-Pakets. `sftp` verwendet intern ein ganz anderes Protokoll als `ftp` und kann wie `ssh` nur eingesetzt werden, wenn auf der Gegenstelle ein SSH-Server läuft. Anonymous FTP ist mit `sftp` nicht möglich. Davon abgesehen, erfolgt die Bedienung des Programms wie die von `ftp`. Mit `sftp -b batchdatei` können Sie SFTP-Downloads automatisieren.

Vielen ist `sftp` zu spartanisch. Die Auswahl komfortablerer SFTP-Clients ist allerdings kleiner als bei FTP. Außerdem ist manchmal etwas Überredungskunst erforderlich, bis der Verbindungsaufbau klappt:

SFTP-Alternative

- ▶ `gftp`: `gftp` bietet vielseitige SFTP-Konfigurationsmöglichkeiten (FTP • OPTIONEN • SSH). Wenn es Probleme gibt, achten Sie darauf, dass Sie den richtigen Port verwenden (22 für SSH, nicht 21 wie bei FTP). Häufig müssen Sie außerdem VERWENDE SSH2 SFTP FUNKTIONEN im Optionsdialog aktivieren.
- ▶ KDE, Gnome: Mit Dolphin oder Nautilus initiieren Sie eine SFTP-Verbindung, indem Sie die Adresse `sftp://user@servername` eingeben. Nach der Passwortabfrage zeigen die Programme das FTP-Verzeichnis wie ein lokales Verzeichnis an. Beide Dateimanager unterstützen auch direkt das SSH-Protokoll, das selbst dann funktioniert, wenn `sftp` nicht zur Verfügung steht. Dazu geben Sie die Adresse in der Form `fish://user@servername` an.

## wget

Der interaktive Ansatz des Kommandos `ftp` ist zur Automatisierung von Downloads – beispielsweise in einem Script – ungeeignet. Auch sonst ist `ftp` reichlich inflexibel. Beispielsweise ist es unmöglich, einen unterbrochenen Download selbstständig wieder aufzunehmen. Abhilfe schafft das Kommando `wget`, das speziell zur Durchführung großer Downloads bzw. zur Übertragung ganzer Verzeichnisse konzipiert ist. `wget` unterstützt gleichermaßen die Protokolle FTP, HTTP und HTTPS.

**Beispiele** In der Grundform lädt `wget` die angegebene Datei einfach herunter:

```
user$ wget ftp://myftpserver.de/name.abc
```

Wenn der Download aus irgendeinem Grund unterbrochen wird, kann er mit `-c` ohne Umstände wieder aufgenommen werden:

```
user$ wget -c ftp://myftpserver.de/name.abc
```

Downloads von großen Dateien, beispielsweise von ISO-Images von Linux-Distributionen, dauern bei einem nicht so guten Internetzugang mehrere Stunden. Da bietet es sich an, den Download über Nacht durchzuführen. Das folgende Kommando stellt nahezu sicher, dass sich die Datei am nächsten Morgen tatsächlich auf dem Rechner befindet. Wegen `-t 20` wird der Download nach einem Verbindungsabbruch bis zu 20-mal neu aufgenommen. `--retry-connrefused` bewirkt, dass selbst nach dem Fehler *connection refused* ein neuer Versuch gestartet wird. Das ist dann zweckmäßig, wenn der Download-Server bekanntermaßen unzuverlässig ist und immer wieder für kurze Zeit unerreichbar ist.

```
user$ wget -t 20 --retry-connrefused http://mydownloadserver.de/name.iso
```

Das folgende Kommando lädt sämtliche Dateien herunter, die notwendig sind, um die angegebene Webseite später in unverändertem Zustand offline zu lesen. Kurz zur Bedeutung der Optionen: `-p` lädt auch CSS-Dateien und Bilder herunter. `-k` verändert in den heruntergeladenen Dateien die Links, sodass diese auf lokale Dateien verweisen. `-E` fügt heruntergeladenen Script-Dateien (ASP, PHP etc.) die Kennung `.html` hinzu. `-H` verfolgt auch Links auf externe Websites.

```
user$ wget -p -k -E -H http://mywebsite.de/seite.html
```

Wenn Sie eine ganze Website offline lesen möchten, hilft das folgende rekursive Download-Kommando (Option `-r`). Die Rekursionstiefe wird durch `-l 4` auf vier Ebenen limitiert.

```
user$ wget -r -l 4 -p -E -k http://mywebsite.de
```



## curl

Das Kommando `curl` hilft dabei, Dateien von oder zu FTP-, HTTP- oder sonstigen Servern zu übertragen. Die `man`-Seite listet eine beeindruckende Palette von Protokollen auf, die `curl` beherrscht. In diesem Abschnitt beschränke ich mich allerdings auf FTP-Uploads. Für die Script-Programmierung besonders praktisch ist, dass `curl` auch Daten aus der Standardeingabe verarbeiten bzw. zur Standardausgabe schreiben kann. Sie müssen also nicht zuerst eine `*.tar.gz`-Datei erstellen und diese dann zum FTP-Server übertragen, sondern können beide Operationen mittels einer Pipe gleichzeitig ausführen.

Das folgende Kommando überträgt die angegebene Datei zum FTP-Server `backupserver` und speichert sie im Verzeichnis `verz`:

```
user$ curl -T datei -u username:password ftp://backupserver/verz
```

Um Daten aus dem Standardeingabekanal zu verarbeiten, geben Sie mit `-T` als Dateinamen einen Bindestrich an. Das folgende Kommando speichert das aus dem `tar`-Kommando resultierende Ergebnis direkt in der Datei `name.tgz` auf dem FTP-Server:

```
user$ tar czf - verz/ | curl -T - -u usern:pw ftp://bserver/name.tgz
```

## lftp

`lftp` ist ein komfortabler interaktiver FTP-Client. Das Kommando eignet sich aber auch gut, um FTP-Uploads oder andere Kommandos in einem Script auszuführen. Dazu können Sie an `lftp` entweder mit `-c` mehrere durch Strichpunkte getrennte FTP-Kommandos übergeben oder mit `-f` eine Datei angeben, die diese Kommandos zeilenweise enthält. Das erste Kommando wird dabei immer `user benutzername,password servername` lauten, um die Verbindung zum FTP-Server herzustellen. Das folgende Kommando demonstriert einen Datei-Upload:

```
root# lftp -c "open -u username,password backupserver; put www.tgz"
```

Wenn Sie der Datei auf dem FTP-Server einen anderen Namen geben möchten, geben Sie zusätzlich die Option `-o <neuerName>` an. `lftp` zeigt während des Uploads den aktuellen Fortschritt an.

Um statt einer Datei ein ganzes Verzeichnis zum Backup-Server zu übertragen, verwenden Sie das Kommando `mirror -R`. (`mirror` kopiert normalerweise Verzeichnisse vom FTP-Server auf den lokalen Rechner. `-R` dreht die Übertragungsrichtung um.) Auch hierzu ein Beispiel:

```
root# lftp -c "open -u usern,passw bserver; mirror -R verzeichnis"
```

Im Unterschied zu anderen FTP-Clients unterstützt `lftp` das Kommando `du`, mit dem Sie feststellen können, wie viel Speicherplatz Ihre Backup-Dateien bereits belegen.

Das ist dann wichtig, wenn Ihr Speicherplatz auf dem Backup-Server streng limitiert ist. Das folgende Kommando zeigt, wie Sie ohne interaktiven Eingriff den bereits belegten Speicherplatz ermitteln. Die Option `-s` gibt an, dass Sie nur an der Endsumme interessiert sind. `-m` bewirkt, dass als Maßeinheit MiB verwendet wird.

```
user$ lftp -c "open -u username,password bserver; du -s -m"
2378 .
```

Wenn Sie das Ergebnis für eine Berechnung verwenden möchten, stört die zweite Spalte (also der Punkt, der angibt, dass sich der Zahlenwert auf das aktuelle Verzeichnis bezieht). Stellen Sie dem Kommando einfach `cut -f 1` hinten, um die erste Spalte zu extrahieren:

```
user$ lftp -c "open -u usern,passw bserver; du -s -m" | cut -f 1
2378
```

### rsync, mirror, sitecopy

`rsync` hilft dabei, ganze Verzeichnisbäume zu kopieren bzw. zu synchronisieren. Eine ausführliche Beschreibung dieses Kommandos finden Sie in [Abschnitt 36.4](#), »Verzeichnisse synchronisieren (rsync)«. Sofern auf dem Partnerrechner weder ein SSH- noch ein `rsync`-Server läuft, können Sie anstelle von `rsync` auf die Kommandos `mirror` oder `sitecopy` zurückgreifen. Das Perl-Skript `mirror` aus dem gleichnamigen Paket kopiert ganze Verzeichnisbäume von einem FTP-Server auf den lokalen Rechner. Das Kommando `sitecopy` ist hingegen dahingehend optimiert, einen Verzeichnisbaum auf einen Webserver hochzuladen, wobei der Datentransfer wahlweise via FTP oder WebDAV erfolgt.

## 14.4 Lynx

### Webbrowser im Textmodus

Webbrowser wie Firefox oder Chrome sind in einer Textkonsole oder in einem Terminalfenster unbrauchbar. Um dennoch auch im Textmodus rasch eine Webseite zu besuchen oder ein HTML-Dokument zu lesen, helfen Programme wie ELinks, Lynx oder `w3m`. Nebenbei können Sie mit diesen Programmen einfache HTML-Dokumente in reinen Text umwandeln. Alle drei Programme sind ähnlich zu bedienen. Zahlreiche Optionen sowie Tastenkürzel sind in den `man`-Seiten bzw. im integrierten Hilfesystem dokumentiert. Aus Platzgründen stelle ich hier nur exemplarisch das bekannteste Programm Lynx näher vor.

**Lynx** Die Bedienung von Lynx ist einfach: Sie starten das Programm im Regelfall dadurch, dass Sie eine WWW-Adresse oder den Namen einer HTML-Datei als Parameter angeben. Lynx lädt das Dokument und zeigt die erste Seite an, wobei Überschriften und Links durch unterschiedliche Farben gekennzeichnet sind. Wenn Sie Lynx mit der

Option `-use_mouse` starten, können Sie das Programm auch per Maus bedienen: Mit der linken Taste folgen Sie einem Link, die mittlere Taste zeigt ein Kontextmenü an, und die rechte Taste führt zur vorherigen Seite zurück.

Lynx verwendet zur Ausgabe standardmäßig den Latin-1-Zeichensatz. Damit Sonderzeichen in Unicode-Konsolen richtig dargestellt werden, geben Sie die Option `-display_charset=utf-8` an. Das folgende Kommando zeigt, wie Sie Lynx als Konverter von HTML in reinen Text einsetzen:

```
user$ lynx -dump quelle.html > ziel.txt
```

## 14.5 Mutt

Zum Lesen lokaler E-Mails bietet sich das textbasierte E-Mail-Programm Mutt an (siehe [Abbildung 14.2](#)). Vor dem ersten Einsatz muss das zumeist gleichnamige Paket installiert werden. In einem Konsolenfenster führen Sie zuerst `su -l` aus, um sich als `root` anzumelden, und starten das Programm dann mit dem Kommando `mutt`.

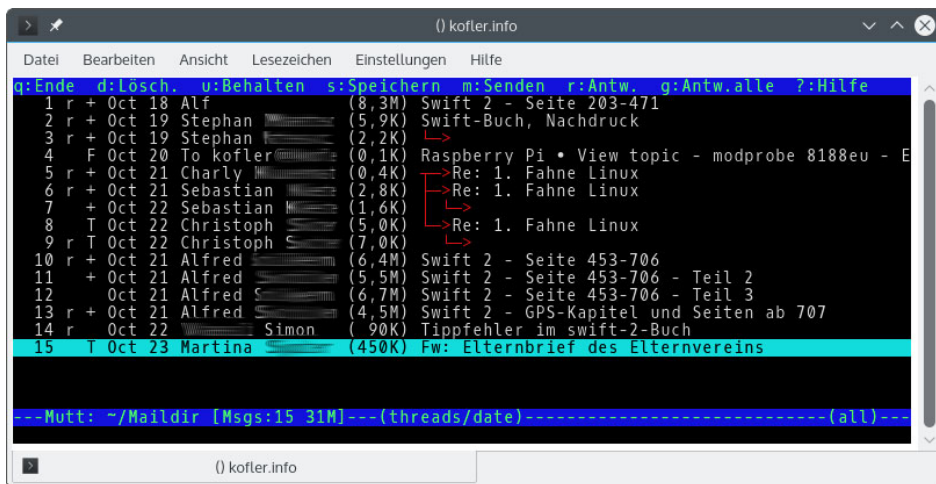


Abbildung 14.2 Lokale E-Mails mit Mutt lesen

Das Programm zeigt auf der Startseite die Titelzeilen aller E-Mails an. Wenn der aktive Benutzer noch keine einzige E-Mail empfangen hat, beklagt sich Mutt darüber, dass es die Datei `/var/mail/benutzer` noch nicht gibt. Diese Warnung können Sie ignorieren. Sie tritt nicht mehr auf, sobald die erste E-Mail eingetroffen ist.

Mit den Cursortasten bewegen Sie sich durch die Inbox. `←` zeigt den Text der ausgewählten E-Mail an. Mit der Leertaste blättern Sie durch die Nachricht. `↓` führt zur nächsten Nachricht, `↑` zurück in die Inbox. `?` zeigt einen Hilfetext mit allen wichtigen Tastenkürzeln an.

Um eine neue E-Mail zu verfassen, drücken Sie **[M]** und geben den Empfänger und die Subject-Zeile an. Anschließend startet Mutt den Editor, den Sie mit der Umgebungsvariable `$EDITOR` oder mit dem Link `/etc/alternatives/editor` ausgewählt haben. Dort schreiben Sie den Nachrichtentext, speichern ihn und verlassen den Editor. Anschließend versenden Sie die E-Mail in Mutt durch **[Y]**.

**[Q]** beendet das Programm. Beim Verlassen stellt Mutt zwei Fragen: Sollen mit **[D]** als gelöscht markierte E-Mails endgültig gelöscht werden? Und sollen gelesene Nachrichten nach `/home/username/mbox` verschoben werden? Wenn Sie vorhaben, die E-Mails später noch mit einem anderen Programm zu bearbeiten, sollten Sie beide Fragen mit **[N]** beantworten. Besonders die zweite Frage ist kritisch: In der lokalen `mbox`-Datei findet nur noch Mutt die E-Mails, nicht aber ein externes Programm wie z. B. der POP-Server Dovecot.

**Konfiguration** Mutt funktioniert auf Anhieb, wenn sich Ihre E-Mail in einer `mbox`-Datei im Verzeichnis `/var/mail/name` befindet. Wenn Ihre E-Mails hingegen im Maildir-Format im Verzeichnis `Maildir` gespeichert werden, müssen Sie die Konfigurationsdatei `.muttrc` mit dem folgenden Inhalt einrichten:

```
# Datei .muttrc
set mbox_type=Maildir
set folder=~/.Maildir"
set mask="!^\.[^.]"
set mbox=~/.Maildir"
set record="+.Sent"
set postponed="+.Drafts"
set spoolfile=~/.Maildir"
```

Weitere Maildir-Konfigurationstipps für diverse Spezialfälle finden Sie hier:

<https://dev.mutt.org/trac/wiki/MuttFaq/Maildir>

<https://eising.wordpress.com/mutt-maildir-mini-howto>

TEIL IV

## **Text- und Code-Editoren**



# Kapitel 15

## Vim

Im Mittelpunkt dieses Kapitels stehen der Editor Vi und dessen Open-Source-Implementierung Vim (*Vi Improved*). Diese Editoren sind – ebenso wie der im nächsten Kapitel vorgestellte Editor Emacs – relativ schwer zu erlernen. Dieser Aufwand lohnt sich nur, wenn Sie ständig Text, Programmcode, HTML-Dokumente etc. bearbeiten, wenn ein Texteditor also ein ständiges und unverzichtbares Werkzeug für Sie ist. Wenn Sie zu dieser Zielgruppe gehören, bieten Vi und Emacs Ihnen schier unendlich viele Spezialfunktionen.

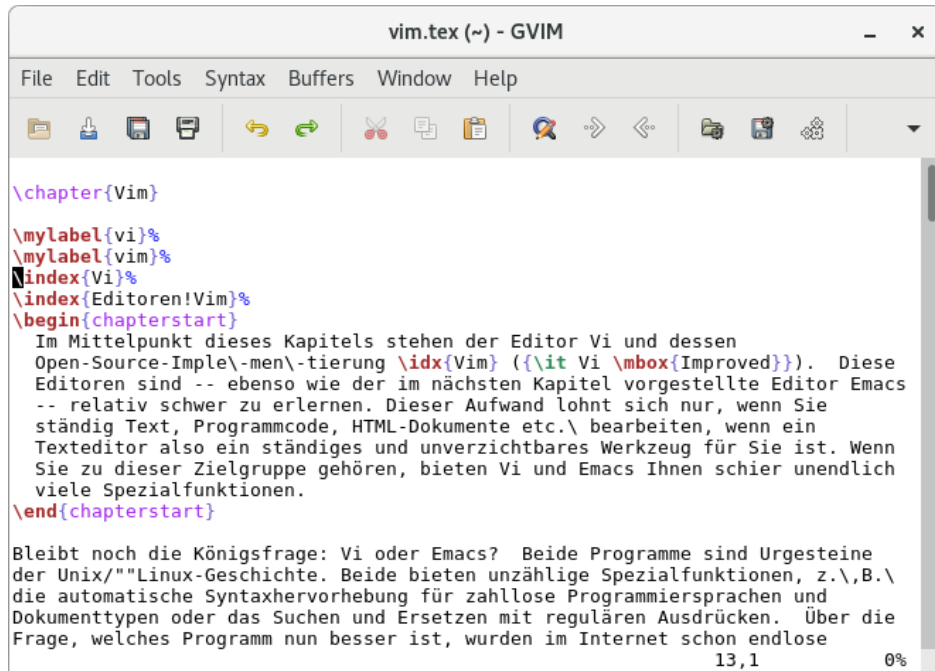
Bleibt noch die Königsfrage: Vi oder Emacs? Beide Programme sind Urgesteine der Unix/Linux-Geschichte. Beide bieten unzählige Spezialfunktionen, z. B. die automatische Syntaxhervorhebung für zahllose Programmiersprachen und Dokumenttypen oder das Suchen und Ersetzen mit regulären Ausdrücken. Über die Frage, welches Programm nun besser ist, wurden im Internet schon endlose Diskussionen geführt. Wirklich objektiv kann auch ich die Frage nicht beantworten: Da ich sämtliche Auflagen dieses Buchs mit Emacs-Varianten verfasst habe (dieses Kapitel natürlich ausgenommen, so viel Vi muss sein!), ist mir der Emacs viel vertrauter als irgendwelche Vi-Varianten.

Persönlich erscheint mir der Editor Emacs intuitiver zu bedienen und einfacher zu erlernen. Beim Vi treibt einen die Unterscheidung zwischen dem Standard- und dem Einfügemodus anfänglich leicht zum Wahnsinn. Für Vi & Co. spricht andererseits, dass das Programm ein De-facto-Standard unter Unix/Linux ist. Es beansprucht wesentlich weniger Ressourcen und steht selbst auf minimalen Rescue-Systemen zur Verfügung. Echte Unix/Linux-Freaks sollten ohnedies beide Editoren in ihren Grundfunktionen beherrschen – und viel mehr vermittele ich in diesem Buch nicht.

Der ursprüngliche Editor Vi ist ein kommerzielles Programm und steht daher unter Linux nicht zur Verfügung. Vim ist dagegen ein Open-Source-Programm, das zu Vi kompatibel ist und darüber hinaus zahllose Verbesserungen und Erweiterungen bietet. Das Programm kann wahlweise mit den Kommandos `vi` oder `vim` gestartet werden.

### Vim im Grafikmodus

Grundsätzlich wird Vim in einer Textkonsole bzw. in einem Konsolenfenster ausgeführt. Wenn Sie ein richtiges Menü und ordentliche Bildlaufleisten bevorzugen, sollten Sie einen Blick auf `gvim` werfen (siehe [Abbildung 15.1](#)). Diese grafische Variante zu Vim muss eventuell extra installiert werden, wobei der Paketname meist `vim-X11` oder `vim-gnome` lautet. (Die X11-Variante läuft selbstverständlich auch unter Wayland.)



**Abbildung 15.1** Die grafische Variante des Editors vim

**Links** Aus Platzgründen kann dieses Kapitel nur eine Einführung zu Vim geben. Für Einsteiger sehr hilfreich ist das Tutorial, das Sie durch das Kommando `vimtutor` starten. (Dadurch wird `vim` gestartet und ein deutschsprachiger Hilfetext geladen, der eine Einführung sowie Beispiele zum Ausprobieren enthält.) Die folgenden Links verweisen auf Seiten mit weiterführenden Informationen:

|   |                         |
|---|-------------------------|
| <a href="http://www.vim.org">http://www.vim.org</a>   | (Homepage)              |
| <a href="http://vimdoc.sourceforge.net/vimfaq.html">http://vimdoc.sourceforge.net/vimfaq.html</a> | (FAQ)                   |
| <a href="https://fprintf.net/vimCheatSheet.html">https://fprintf.net/vimCheatSheet.html</a>       | (Tastenkürzel)          |
| <a href="http://truth.sk/vim/vimbook-OPL.pdf">http://truth.sk/vim/vimbook-OPL.pdf</a>             | (500-seitiges Vim-Buch) |

### Vim ist Charityware

Der Hauptentwickler Bram Moolenaar bezeichnet Vim als »Charityware«: Vim ist kostenlos unter einer GPL-kompatiblen Lizenz verfügbar. Wer Vim regelmäßig nutzt, wird aber gebeten, sich in Form einer Spende zu bedanken, die einer Kinderhilfsorganisation in Uganda zugutekommt. Weitere Informationen liefert das Vim-Kommando `[Esc] : help iccf [←]`.



## 15.1 Schnelleinstieg

Sie starten Vim üblicherweise in der Form `vim dateiname` innerhalb einer Textkonsole oder in einem Konsolenfenster. Die zu ändernde Datei wird direkt in der Konsole angezeigt.

Bevor Sie darauflosschreiben können, müssen Sie sich allerdings mit einer Eigenheit auseinandersetzen: Das Programm unterscheidet zwischen unterschiedlichen Bearbeitungsmodi (siehe [Tabelle 15.1](#)). Der Standardmodus dient nicht zur Eingabe von Text, sondern zur Ausführung von Kommandos. Wenn Sie im Standardmodus beispielsweise `[L]` eingeben, bewegen Sie damit den Cursor um ein Zeichen nach links. `[D]` `[W]` löscht ein Wort, `[P]` fügt es an der aktuellen Cursorposition wieder ein etc.

Standardmodus

Um Text einzugeben, müssen Sie mit `[I]` (*insert*) oder `[A]` (*append*) in den Einfügemodus wechseln. `vim` zeigt nun in der untersten Zeile ganz links den Text `-- EINFÜGEN --` an. Im Einfügemodus können Sie Text eingeben, den Cursor bewegen und einzelne Zeichen löschen (`[Entf]` und `[←]`). Der Unterschied zwischen `[I]` und `[A]` besteht darin, dass die Eingabe bei `[I]` an der aktuellen Cursorposition beginnt, bei `[A]` beim Zeichen dahinter.

Einfügemodus

Bevor Sie wieder ein Kommando eingeben können, müssen Sie mit `[Esc]` zurück in den Standardmodus wechseln. Dieser Modus wird nicht extra gekennzeichnet. Der linke Teil der letzten Zeile ist jetzt also leer.

### Moduswechsel ohne Änderung der Cursorposition

Beim Wechsel vom Einfüge- in den Standardmodus bewegt sich der Cursor um ein Zeichen nach links – es sei denn, er steht bereits am Beginn einer Zeile. Dieses merkwürdige Verhalten ist laut Vim-FAQ beabsichtigt und kann nicht verhindert werden. Um ein einzelnes Kommando auszuführen, ohne den Einfügemodus zu verlassen – und damit auch ohne die aktuelle Cursorposition zu verändern –, leiten Sie die Kommandoeingabe mit `[Strg]+[O]` ein.

Im Einfügemodus können Sie mit `[Entf]` und `[←]` wie üblich einzelne Zeichen löschen. Wenn Sie Wörter, Zeilen oder ganze Bereiche löschen möchten, wechseln Sie zuerst mit `[Esc]` in den Standardmodus. Anschließend löscht `[D]`, `[W]` ein Wort und `[D]`, `[D]` eine ganze Zeile. Wenn Sie eine Zahl voranstellen, wird das Löschkommando entsprechend oft wiederholt. `[5]`, `[D]`, `[D]` löscht also fünf Zeilen. `[.]` wiederholt das zuletzt ausgeführte Kommando.

Text löschen

| Tastenkürzel                      | Funktion   |
|-----------------------------------|--|
| <b>I</b>                          | aktiviert den Einfügemodus.  |
| <b>A</b>                          | aktiviert den Einfügemodus. Die Texteingabe beginnt beim nächsten Zeichen. |
| <b>Esc</b>                        | aktiviert den Standardmodus bzw. bricht die Kommandoingabe ab.             |
| <b>Kommandos im Standardmodus</b> |  |
| <b>D</b> , <b>W</b>               | löscht ein Wort.   |
| <b>D</b> , <b>D</b>               | löscht die aktuelle Zeile.   |
| <i>n</i> <b>D</b> , <b>D</b>      | löscht <i>n</i> Zeilen.  |
| <b>P</b>                          | fügt den zuletzt gelöschten Text hinter der Cursorposition ein.            |
| <b>⇧</b> + <b>P</b>               | fügt den zuletzt gelöschten Text vor der Cursorposition ein.               |
| <b>.</b>                          | wiederholt das letzte Kommando.  |
| <b>U</b>                          | macht die letzte Änderung rückgängig (Undo).                               |
| <b>⇧</b> + <b>U</b>               | widerruft alle Änderungen in der aktuellen Zeile.                          |
| <b>Strg</b> + <b>R</b>            | macht Undo rückgängig (Redo, ab Vim 7).                                    |
| <b>:</b> <b>w</b>                 | speichert die Datei.   |
| <b>:</b> <b>q</b>                 | beendet vim.   |
| <b>:</b> <b>q!</b>                | beendet vim auch dann, wenn es nicht gespeicherte Dateien gibt.            |
| <b>Kommandos im Einfügemodus</b>  |  |
| <b>Strg</b> + <b>O</b> kommando   | führt das Kommando aus, ohne den Einfügemodus zu verlassen.                |

Tabelle 15.1 Elementare Kommandos

**P** (*put*) fügt den zuletzt gelöschten Text hinter der aktuellen Cursorposition ein, **⇧** + **P** davor. **U** (*undo*) widerruft die letzten Änderungen, **Strg** + **R** (*redo*) stellt die Änderungen wieder her. (Vim 6 kann nur die letzte Änderung widerrufen, ein nochmaliges **U** stellt die Änderung wieder her.)

#### Speichern und beenden

Um die geänderte Datei zu speichern, wechseln Sie mit **Esc** in den Standardmodus und geben dann das Kommando **:** **w** **↵** (*write*) ein. **:** **q** **↵** (*quit*) beendet den Editor, sofern alle offenen Dateien gespeichert sind. Mit **:** **q!** **↵** erzwingen Sie ein Ende selbst dann, wenn es nicht gespeicherte Änderungen gibt. **:** **wq** **↵** kombiniert das Speichern und das Programmende.

## Hilfe

Vim stellt eine umfassende Online-Hilfe in englischer Sprache zur Verfügung. Zur Startseite des Hilfesystems gelangen Sie von jedem Modus aus mit `[F1]`. Alternativ führen im Standardmodus `:` `help` bzw. `:` `help thema` zur Hilfe. Wenn Sie wissen möchten, welche Hilfethemen es gibt, die das Schlüsselwort *abc* enthalten, geben Sie `:` `help abc` `[Strg]` `[D]` ein.

Der Hilfetext wird in einem eigenen Teilbereich von vim angezeigt (einem sogenannten Fenster, auch wenn es sich dabei nicht um ein eigenständiges Fenster im Sinne des Linux-Grafiksystems handelt). Dieses Fenster schließen Sie mit `:` `q` wieder. Sie können das Hilfefenster aber auch geöffnet lassen und im ursprünglichen Text weiterarbeiten. Dazu wechseln Sie mit `[Strg]` `[W]`, `[W]` das gerade aktive Fenster. Mehr Informationen zum Umgang mit vim-Fenstern, -Puffern und zur Bearbeitung mehrerer Dateien folgen in [Abschnitt 15.5](#), »Mehrere Dateien gleichzeitig bearbeiten«.

Das Hilfefenster

Im Hilfetext sind Verweise auf andere Hilfethemen hervorgehoben (in der von mir getesteten Version hellblau). Um zu diesem Thema zu springen, bewegen Sie den Cursor auf das Schlüsselwort und führen `[Strg]` `[J]` aus. Noch einfacher geht es, wenn die Maus aktiviert ist (siehe [Abschnitt 15.7](#), »Tipps und Tricks«): Dann reicht ein Doppelklick auf das Hilfethema, um dorthin zu springen. `[Strg]` `[T]` führt zur ursprünglichen Seite zurück.

Navigation  
in der Hilfe

## 15.2 Cursorbewegung

Die Cursortasten funktionieren sowohl im Standardmodus als auch im Einfügemodus. Außerdem können Sie die Cursorposition durch diverse Tastenkombinationen im Standardmodus ändern (siehe [Tabelle 15.2](#)). Vi-Freaks bewegen sich damit effizienter durch den Text als mit den Cursortasten.

Eine Eigenheit von vim besteht darin, dass `[←]` am Beginn einer Zeile den Cursor nicht an das Ende der vorherigen Zeile stellt. Analog funktioniert auch `[→]` am Ende einer Zeile nicht wie gewohnt. Um das übliche Verhalten anderer Editoren zu erzielen, führen Sie im Standardmodus `:` `set whichwrap=b,s,<,>,[,]` aus bzw. fügen dieses `set`-Kommando in `.vimrc` ein.

`[M]` buchstabe speichert die aktuelle Cursorposition in einem Positionsmarker. Mit `[']` buchstabe bewegen Sie den Cursor zurück an die so gespeicherte Position.

Cursorpositionen  
speichern

Vim merkt sich die Cursorposition, an der eine neue Cursorbewegung beginnt. `[']` `[']` führt zurück zu dieser Position. Nochmals `[']` `[']` bewegt den Cursor wieder an die letzte Position. `[']` `[[` bzw. `[']` `][` bewegen den Cursor an den Beginn bzw. das Ende des zuletzt veränderten Textabschnitts.

| Tastenkürzel                              | Funktion   |
|---|--|
| Cursortasten                              | Die Cursortasten haben die übliche Bedeutung.                      |
| <b>H</b> / <b>L</b>                       | bewegt den Cursor nach links/rechts.                               |
| <b>J</b> / <b>K</b>                       | bewegt den Cursor nach unten/oben.                                 |
| <b>↵</b> + <b>H</b> / <b>↵</b> + <b>L</b> | bewegt den Cursor an den Beginn bzw. das Ende der aktuellen Seite. |
| <b>↵</b> + <b>M</b>                       | bewegt den Cursor in die Mitte der aktuellen Seite.                |
| <b>B</b> / <b>W</b>                       | bewegt den Cursor um ein Wort nach links/rechts.                   |
| <b>E</b>                                  | bewegt den Cursor an das Ende des Worts.                           |
| <b>G</b> , <b>E</b>                       | bewegt den Cursor an den Anfang des Worts.                         |
| <b>(</b> , <b>)</b>                       | bewegt den Cursor an den Beginn des aktuellen/nächsten Satzes.     |
| <b>{</b> , <b>}</b>                       | bewegt den Cursor an den Beginn des aktuellen/nächsten Absatzes.   |
| <b>^</b> , <b>\$</b>                      | bewegt den Cursor an den Beginn bzw. das Ende der Zeile.           |
| <b>↵</b> + <b>G</b>                       | bewegt den Cursor an das Ende der Datei.                           |
| <b>G</b> , <b>G</b>                       | bewegt den Cursor an den Beginn der Datei.                         |
| <i>n</i> <b>↵</b> + <b>G</b>              | bewegt den Cursor in die Zeile <i>n</i> .                          |
| <i>n</i> <b> </b>                         | bewegt den Cursor in die Spalte <i>n</i> .                         |
| <b>%</b>                                  | bewegt den Cursor zum korrespondierenden Klammerzeichen ()[]{}.    |

Tabelle 15.2 Tastenkürzel zur Cursorbewegung im Standardmodus

**Wo bin ich?** Im Einfügemodus zeigt Vim rechts in der Statuszeile die aktuelle Zeilen- und Spaltennummer sowie eine Prozentzahl an, die angibt, in welchem Abschnitt des Texts Sie sich befinden (z. B. 92 % – also in den letzten 10 Prozent). Mit **Strg** + **G** zeigt Vim in der Statuszeile auch den Namen der Datei, ihren Zustand (z. B. *Verändert*), die gesamte Länge in Zeilen und die relative Position im Text in Prozent an.

### 15.3 Text bearbeiten

**Textzeichen mehrfach einfügen** Um ein Textzeichen mehrfach einzufügen, geben Sie im Standardmodus die Anzahl, das Kommando **A** (*append*), das gewünschte Zeichen und schließlich **Esc** ein. Um also 50-mal das Zeichen = einzugeben, geben Sie **5**, **0**, **A**, **=**, **Esc** ein. Nach dem Kommando befinden Sie sich wieder im Standardmodus.

**Tippfehler** Vim hilft auch bei der Korrektur typischer Tippfehler: **~** ändert die Groß- und Kleinschreibung des aktuellen Buchstabens. **X**, **P** vertauscht die folgenden zwei Buchstaben.

Tabelle 15.3 gibt einen Überblick über die wichtigsten Kommandos zum Löschen von Text. Wenn Sie vor dem Löschkommando eine Zahl eingeben, wird das Löschkommando entsprechend oft wiederholt. Wie für alle anderen Vim-Kommandos gilt: `.` wiederholt das letzte Kommando, `n` `.` wiederholt es  $n$ -mal.

Text löschen

| Tastenkürzel                           | Funktion im Einfügemodus   |
|--|--|
| <code>[Entf]</code> , <code>[←]</code> | Diese Tasten haben die übliche Bedeutung.  |
| <b>Funktion im Standardmodus</b>       |  |
| <code>[X]</code>                       | löscht das Zeichen an der Cursorposition bzw. den markierten Text.   |
| <code>[↵] + [X]</code>                 | löscht das Zeichen vor dem Cursor.   |
| <code>[D]</code> , <code>[D]</code>    | löscht die aktuelle Zeile.   |
| <code>[D]</code> cursorkommando        | löscht den Text entsprechend dem Kommando zur Cursorbewegung (siehe Tabelle 15.2). Beispiele: <code>[D]</code> , <code>[\$]</code> löscht bis zum Ende der Zeile. <code>[D]</code> , <code>[B]</code> löscht das vorige Wort. <code>[D]</code> , <code>[W]</code> löscht das nächste Wort. |

Tabelle 15.3 Text löschen

Text wird grundsätzlich in ein Kopierregister gelöscht. Der zuletzt gelöschte Text kann von dort mit `[↵] + [P]` an der aktuellen Cursorposition bzw. mit `[P]` hinter der Cursorposition wieder in den Text eingefügt werden.

Eine eigentümliche Art, Text zu löschen und dann durch neuen Text zu ersetzen, bieten die *C*-Kommandos (*change*): Beispielsweise löscht `[C]`, `[W]` das aktuelle Wort und aktiviert den Einfügemodus. Sie geben nun das neue Wort ein und schließen die Eingabe mit `[Esc]` ab. Analog funktioniert `[C]` auch für andere Cursorkommandos.

Sie können Text auch in das Kopierregister einfügen, ohne ihn zu löschen. Tabelle 15.4 fasst die entsprechenden Kommandos zusammen. Alle gelten für den Standardmodus.

Text kopieren

| Tastenkürzel                        | Funktion   |
|-------------------------------------|--|
| <code>[Y]</code>                    | kopiert den markierten Text in das Kopierregister.   |
| <code>[Y]</code> , <code>[Y]</code> | kopiert die aktuelle Zeile in das Kopierregister.  |
| <code>[Y]</code> cursorkommando     | kopiert den durch die Cursorbewegung erfassten Text. Beispiel: <code>[Y]</code> , <code>[J]</code> kopiert den Text bis zum Ende des Absatzes. |

Tabelle 15.4 Text in das Kopierregister kopieren

**Text markieren** Einige (Löschen-)Kommandos setzen voraus, dass Sie zuerst einen Textausschnitt markieren. Vim sieht dazu drei verschiedene Markierungsmodi vor, die Sie mit `[V]`, `[⇧]+[V]` bzw. `[Strg]+[V]` am Startpunkt der Markierung aktivieren bzw. ebenso wieder deaktivieren. Während einer dieser Modi aktiv ist, enthält die unterste Vim-Zeile den Text `-- VISUELL --`. Sie bewegen den Cursor nun zum Endpunkt der Markierung oder erweitern die Markierung durch einige spezielle Markierungskommandos (siehe [Tabelle 15.5](#)). Solange der Markierungsmodus aktiv ist, stehen Ihnen diverse Kommandos zur Bearbeitung des markierten Texts zur Auswahl (siehe [Tabelle 15.6](#)).

| Tastenkürzel              | Funktion  |
|---------------------------|---|
| <code>[V]</code>          | (de)aktiviert den Zeichenmarkierungsmodus.                        |
| <code>[⇧]+[V]</code>      | (de)aktiviert den Zeilenmarkierungsmodus.                         |
| <code>[Strg]+[V]</code>   | (de)aktiviert den Blockmarkierungsmodus.                          |
| <code>[A], [W]</code>     | vergrößert die Markierung um ein Wort.                            |
| <code>[A], [S]</code>     | vergrößert die Markierung um einen Satz.                          |
| <code>[A], [P]</code>     | vergrößert die Markierung um einen Absatz.                        |
| <code>[A], [B]</code>     | vergrößert die Markierung um eine <code>()</code> -Ebene.         |
| <code>[A], [⇧]+[B]</code> | vergrößert die Markierung um eine <code>{}</code> -Ebene.         |
| <code>[G], [V]</code>     | markiert den zuletzt markierten Text nochmals.                    |
| <code>[O]</code>          | wechselt die Cursorposition zwischen Markierungsanfang und -ende. |

**Tabelle 15.5** Text markieren

| Tastenkürzel                | Funktion  |
|-----------------------------|---|
| <code>[X]</code>            | löscht den markierten Text.   |
| <code>[Y]</code>            | kopiert den markierten Text in das Kopierregister.                            |
| <code>[~]</code>            | ändert die Groß-/Kleinschreibung.   |
| <code>[J]</code>            | fügt die markierten Zeilen zu einer langen Zeile zusammen.                    |
| <code>[G], [Q]</code>       | führt einen Zeilenumbruch durch (für Fließtext).                              |
| <code>[&gt;], [&lt;]</code> | rückt den Text um eine Tabulatorposition ein oder aus.                        |
| <code>[=]</code>            | rückt den Text dem aktuellen <code>indent</code> -Modus entsprechend neu ein. |
| <code>!sort</code>          | sortiert die Zeilen mit dem externen Kommando <code>sort</code> .             |

**Tabelle 15.6** Markierten Text bearbeiten

Gerade beim Editieren von Code ist das richtige Einrücken von Zeilen wichtig. Vim hilft dabei auf vielfältige Weise. Die elementarsten Kommandos sind `>`, `>` bzw. `<`, `<`. Sie rücken die aktuelle Zeile um eine Tabulatorposition ein oder aus. Wenn Sie vorher mehrere Zeilen Text markieren, können Sie die Kommandos auf einen ganzen Block anwenden. Dabei reicht die einfache Eingabe von `>` bzw. `<`. `: set shiftwidth=N` verändert die Einrücktiefe (normalerweise 8 Zeichen).

Zeilen einrücken

Vim kann auch versuchen, neue Zeilen schon während der Eingabe automatisch einzurücken. Dazu aktivieren Sie einen Einrückmodus, beispielsweise durch `: set cindent`. Im Folgenden sind die Grundfunktionen der wichtigsten Vim-Einrückmodi kurz zusammengefasst:

- ▶ **autoindent:** Rückt die folgende Zeile genauso weit ein wie die vorherige.
- ▶ **smartindent:** Funktioniert wie `autoindent`, berücksichtigt aber zusätzlich `{}`-Klammerebenen. Damit Vim die schließenden Klammern richtig erkennt, sollten diese am Beginn einer neuen Zeile angegeben werden. Das Ausmaß der Einrückung je nach Klammerebene steuern Sie durch die Option `shiftwidth`. Zuvor markierter Text kann durch `=` neu eingerückt werden.
- ▶ **cindent:** Funktioniert wie `smartindent`, berücksichtigt aber auch diverse Codestrukturen von C bzw. C++. Der Einrückmechanismus kann durch verschiedene Optionen den persönlichen Vorlieben angepasst werden (siehe dazu `: help C-indenting`).

Vim ist so vorkonfiguriert, dass das Verfassen von Code bzw. das Ändern von Konfigurationsdateien möglichst gut funktioniert. Aus diesem Grund führt Vim keinen automatischen Zeilenumbruch durch (d. h., Sie müssen neue Zeilen selbst mit `↵` beginnen). Sie können Vim aber selbstverständlich auch zum Verfassen gewöhnlichen Texts einsetzen (etwa für E-Mails). [Tabelle 15.7](#) fasst einige spezielle Kommandos und Optionen zusammen, die dabei helfen:

Fließtext

| Tastenkürzel  | Funktion   |
|---|--|
| <code>↵ + J</code>  | verbindet die aktuelle Zeile mit der folgenden.  |
| <code>n ↵ + J</code>  | verbindet <i>n</i> Zeilen zu einer langen Zeile.   |
| <code>G</code> , <code>Q</code> , <code>A</code> , <code>P</code> | umbricht den aktuellen Absatz neu und stellt den Cursor an den Beginn des nächsten Absatzes. |
| <code>G</code> , <code>W</code> , <code>A</code> , <code>P</code> | wie oben, aber belässt den Cursor am aktuellen Ort.  |
| <code>: set textwidth=nn</code>                                   | automatischer Zeilenumbruch nach maximal <i>nn</i> Zeichen (normalerweise: 0 = deaktiviert)  |

Tabelle 15.7 Fließtext bearbeiten

Die `G`-Kommandos berücksichtigen automatisch den `autoindent`-Modus sowie die Einstellung von `textwidth`. Wenn `textwidth` 0 enthält, beträgt die maximale Zeilenlänge 79 Zeichen. Eine Menge Konfigurationsmöglichkeiten für eine besonders komfortable Fließtexteingabe bietet die Option `formatoptions` (siehe den dazugehörigen Hilfetext).

#### Wortergänzungen

Das Eintippen langer Wörter und von Funktions- und Variablenamen ist mühsam und fehleranfällig. Vim hilft Ihnen dabei auf geniale Weise: Sie geben lediglich die ersten Buchstaben ein und drücken dann `Strg]+P`. Wenn das Wort bereits eindeutig bestimmt ist, wird es sofort vervollständigt. Andernfalls können Sie mit `Strg]+P` bzw. den Cursortasten das gewünschte Wort auswählen. Vim berücksichtigt bei der Wortergänzung alle Wörter aller geladenen Dateien, wobei Wörter aus der aktuellen Datei und dabei wiederum Wörter in der Nähe der Cursorposition bevorzugt werden.

## 15.4 Suchen und Ersetzen

#### Text suchen

Im Standardmodus bewegt `/` `suchtext` `↵` den Cursor zum gesuchten Text. `N` wiederholt die Suche, `↵+N` wiederholt die Suche rückwärts. Um von vornherein rückwärts zu suchen, beginnen Sie die Suche mit `?` `suchausdruck`. [Tabelle 15.8](#) erläutert die wichtigsten Sonderzeichen, um nach Mustern zu suchen.

| Zeichen | Bedeutung  |
|---------|--|
| .       | ein beliebiges Zeichen                                     |
| ^ \$    | Zeilenanfang/Zeilenende                                    |
| \< \>   | Wortanfang/Wortende  |
| [a-e]   | ein Zeichen zwischen <i>a</i> und <i>e</i>                 |
| \s, \t  | ein Leerzeichen bzw. ein Tabulatorzeichen                  |
| \( \)   | fasst ein Suchmuster als Gruppe zusammen.                  |
| \=      | Der Suchausdruck muss 0- oder einmal auftreten.            |
| *       | Der Suchausdruck darf beliebig oft (auch 0-mal) auftreten. |
| \+      | Der Suchausdruck muss mindestens einmal auftreten.         |

**Tabelle 15.8** Sonderzeichen im Suchausdruck

#### Groß- und Kleinschreibung

Vim unterscheidet bei der Suche standardmäßig zwischen Groß- und Kleinschreibung. Wenn Sie das nicht möchten, leiten Sie das Suchmuster mit `/c` ein (gilt nur für diese Suche) oder führen `:` `set ignorecase` aus (gilt für alle weiteren Suchen).



Mit `:set incsearch` aktivieren Sie die sogenannte inkrementelle Suche: Bereits während der Eingabe des Suchtexts durch `/` suchausdruck bewegt Vim den Cursor zum ersten passenden Ort. `↵` beendet die Suche, `Esc` bricht sie ab. Nach der Suche bleiben alle Übereinstimmungen im Text markiert, bis Sie eine neue Suche durchführen oder `:nohlsearch` ausführen.

Inkrementelle  
Suche

Um alle Vorkommen des Texts `abc` ohne Rückfrage durch `efg` zu ersetzen, führen Sie im Standardmodus `:%s/abc/efg/g` aus. `'` `'` führt anschließend zurück an den Beginn der Suche. [Tabelle 15.9](#) stellt einige Varianten des Suchen-und-Ersetzen-Kommandos vor. Beim Suchen und Ersetzen mit Rückfrage können Sie mit `Y` oder `N` für jeden gefundenen Suchausdruck angeben, ob dieser durch den neuen Text ersetzt werden soll oder nicht. `Q` bricht den Vorgang ab, `A` ersetzt alle weiteren Vorkommen. Im Ersetzen-Ausdruck können Sie sich mit `\n` auf die  $n$ -te Gruppe im Suchmuster beziehen. Eine Menge weiterer Tipps zum Suchen und Ersetzen sowie zahlreiche Beispiele finden Sie in der Online-Hilfe (`:help substitute`).

Suchen und  
Ersetzen

| Tastenkürzel                 | Funktion  |
|------------------------------|---|
| <code>:%s/abc/efg/g</code>   | ersetzt ohne Rückfrage alle Vorkommen von <code>abc</code> durch <code>efg</code> . |
| <code>:%s/abc/efg/gc</code>  | ersetzt mit Rückfrage alle Vorkommen von <code>abc</code> durch <code>efg</code> .  |
| <code>:%s/abc/efg/gci</code> | ersetzt ohne Berücksichtigung der Groß- und Kleinschreibung.                        |

Tabelle 15.9 Suchen und Ersetzen

## 15.5 Mehrere Dateien gleichzeitig bearbeiten

Im Standardmodus lädt `:e dateiname` eine neue Datei. Die neue Datei ersetzt die momentan bearbeitete Datei, die Sie vorher speichern müssen – andernfalls bricht Vim den Vorgang ab. Sie können das Laden der Datei mit `:e! dateiname` zwar erzwingen, verlieren dann aber alle durchgeführten Änderungen an der zuletzt aktuellen Datei.

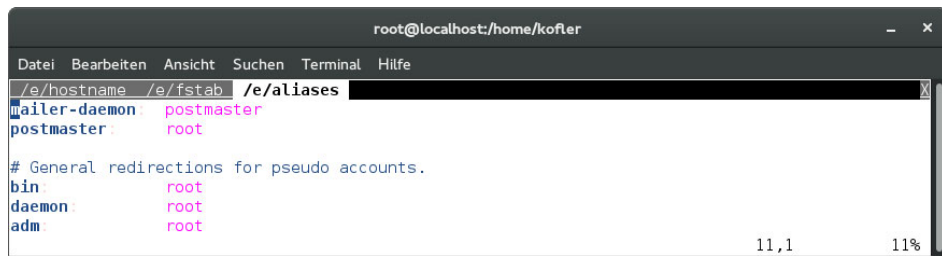
Selbstverständlich können Sie in Vim auch mehrere Dateien gleichzeitig bearbeiten. Vorher sollten Sie allerdings das nicht besonders intuitive Konzept verstehen, wie Vim intern Texte verwaltet und anzeigt. Jeder im Editor dargestellte Text befindet sich intern in einem sogenannten Puffer. Das gilt sowohl für Dateien als auch für Hilfetexte. Solange es nur einen Puffer gibt, wird dieser auf der gesamten Vim-Arbeitsfläche angezeigt. Um mehrere Puffer gleichzeitig anzuzeigen, wird die Arbeitsfläche in mehrere sogenannte Fenster aufgeteilt, wie dies auch bei der Anzeige von Hilfetexten der Fall ist. Ein derartiges Fenster ist kein eigenständiges Fenster im Sinne des Linux-Grafiksystems, sondern nur ein Teilbereich der Arbeitsfläche.

Puffer und  
Fenster

Die Puffer veränderter Dateien sind immer in einem Fenster sichtbar. Puffer von Dateien, die seit dem letzten Speichern nicht mehr geändert wurden, können dagegen ausgeblendet werden. Die Puffer bleiben dabei im Speicher, gelten nun aber als nicht mehr aktiv. (Vorsicht: Wenn Sie ein Fenster mit einer noch nicht gespeicherten Datei schließen, gehen alle Änderungen verloren! Der inaktive Puffer der Datei bleibt zwar verfügbar, enthält aber die Datei zum Zeitpunkt der letzten Speicherung.)

Vim ist auch in der Lage, eine Datei gleichzeitig in mehreren Fenstern darzustellen, z. B. um unterschiedliche Teile eines sehr langen Texts zu bearbeiten.

**Tabbed-Fenster** Vim erleichtert die Bearbeitung mehrerer Dateien mit Dialogblättern im Textmodus (siehe [Abbildung 15.2](#)). Wirklich komfortabel funktioniert das, wenn Sie die Maus aktiviert haben (siehe [Abschnitt 15.7](#), »Tipps und Tricks«): Dann können Sie bequem mit der Maus die gerade aktive Datei auswählen bzw. mit dem X-Button rechts oben schließen. Weitere Informationen zu diesen *Tabbed Pages* erhalten Sie mit `: help tabpage`.



**Abbildung 15.2** Drei Dateien in drei Tabbed-Fenstern

**Eine neue Datei laden** Je nachdem, ob Sie mit Fenstern oder Tabbed-Fenstern arbeiten möchten, laden Sie neue Dateien mit `: new dateiname` oder `: tabnew dateiname`. Wenn Sie beim Programmstart mehrere Dateien übergeben, also beispielsweise `vim datei1 datei2 datei3`, wird lediglich die erste Datei angezeigt; die anderen werden in unsichtbare Puffer geladen. Um jede Datei in einem Fenster bzw. in einem Tabbed-Fenster zu öffnen, müssen Sie zusätzlich die Option `-o` bzw. `-p` übergeben.

[Tabelle 15.10](#) fasst die wichtigsten Kommandos zusammen, um Dateien zu laden und zu speichern, zwischen (Tabbed-)Fenstern zu wechseln etc.

| Tastenkürzel               | Funktion                                 |
|----------------------------|--|
| <code>: e dateiname</code> | lädt eine Datei in den aktuellen Puffer. |
| <code>: w</code>           | speichert die aktuelle Datei.            |
| <code>: wall</code>        | speichert alle offenen Dateien.          |

**Tabelle 15.10** Dateien, Puffer und Fenster

| Tastenkürzel                      | Funktion   |
|-----------------------------------|--|
| <code>: wq</code>                 | speichert und schließt den Puffer.   |
| <code>: e datei</code>            | lädt eine Datei in den aktuellen Puffer.   |
| <code>: w</code>                  | speichert die aktuelle Datei.  |
| <code>: wall</code>               | speichert alle offenen Dateien.  |
| <code>: wq</code>                 | speichert und schließt den Puffer.   |
| <code>: q</code>                  | schließt den aktuellen Puffer und beendet Vim, wenn keine weiteren Puffer mehr offen sind. |
| <code>: q!</code>                 | schließt den Puffer auch mit ungesicherten Änderungen.                                     |
| <code>: qall</code>               | schließt alle Puffer und beendet Vim.  |
| <code>: split</code>              | teilt das Fenster und zeigt in beiden Teilen denselben Text an.                            |
| <code>: new</code>                | erzeugt einen leeren Puffer und zeigt ihn in einem Fenster an.                             |
| <code>: new datei</code>          | lädt eine Datei in einen neuen Puffer.   |
| <code>: only</code>               | maximiert das aktuelle Fenster und schließt die anderen Puffer.                            |
| <code>: all</code>                | zeigt alle Puffer in entsprechend verkleinerten Fenstern an.                               |
| <code>: buffers</code>            | liefert die Liste aller Puffer.  |
| <code>: buffer n</code>           | zeigt den Puffer <i>n</i> an und löscht den aktuellen Puffer.                              |
| <code>: buffer datei</code>       | zeigt den Puffer mit der Datei im aktuellen Fenster an.                                    |
| <code>: tabnew</code>             | erzeugt einen Puffer und zeigt ihn in einem Tabbed-Fenster an.                             |
| <code>: tabnew datei</code>       | lädt eine Datei und zeigt sie in einem Tabbed-Fenster an.                                  |
| <code>: tabnext</code>            | wechselt in das nächste Tabbed-Fenster.  |
| <code>: tabprevious</code>        | wechselt in das vorige Tabbed-Fenster.   |
| <code>Strg + Bild↑ / Bild↓</code> | wechselt in das nächste/vorige Tabbed-Fenster.   |
| <code>: tabclose</code>           | schließt das aktuelle Tabbed-Fenster.  |
| <code>: tabonly</code>            | schließt alle anderen Tabbed-Fenster.  |

Tabelle 15.10 Dateien, Puffer und Fenster (Forts.)

## 15.6 Interna

Auf den vorigen Seiten wurden immer wieder besondere Funktionen von Vim durch Optionen aktiviert oder verändert, und dieser Abschnitt stellt eine Menge weiterer Optionen vor. Beachten Sie, dass manche Optionen nur lokal für die aktuelle Datei bzw. den aktuellen Puffer gelten. `: set` verändert in diesem Fall nur die lokale Ein-

Optionen

stellung. Um die Option global zu verändern, verwenden Sie `: setglobal`. [Tabelle 15.11](#) fasst die wichtigsten Kommandos zur Bearbeitung von Optionen zusammen.

| Tastenkürzel                                  | Funktion   |
|---|--|
| <code>: help options</code>                   | liefert allgemeine Informationen sowie eine Optionsreferenz. |
| <code>: help 'option'</code>                  | liefert Informationen zur angegebenen Option.                |
| <code>: set</code>                            | liefert alle Optionen, die nicht im Grundzustand sind.       |
| <code>: set &lt;boolescheoption&gt;</code>    | aktiviert die boolesche Option.                              |
| <code>: set no&lt;boolescheoption&gt;</code>  | deaktiviert die boolesche Option.                            |
| <code>: set inv&lt;boolescheoption&gt;</code> | invertiert den aktuellen Zustand der Option.                 |
| <code>: set option?</code>                    | zeigt die Einstellung der Option an.                         |
| <code>: set option=wert</code>                | stellt die Option neu ein.                                   |
| <code>: set option+=wert</code>               | verändert die Option.  |
| <code>: set option-=wert</code>               | verändert die Option.  |
| <code>: set option&amp;</code>                | setzt die Option auf den Grundzustand zurück.                |

**Tabelle 15.11** Umgang mit Optionen

#### Konfiguration (vimrc)

Sämtliche Optionseinstellungen gehen beim Beenden von Vim verloren. Um Optionen bleibend einzustellen, verändern Sie die Vim-Konfigurationsdatei `.vimrc`. Beachten Sie, dass Kommentare durch das Zeichen `"` eingeleitet werden!

Die folgenden Zeilen geben ein einfaches Beispiel, wie `.vimrc` aussehen kann. Alle dort verwendeten Optionen wurden bzw. werden in diesem Kapitel vorgestellt. Wenn Sie im Internet nach `vimrc` suchen, finden Sie zahllose weitere Beispiele.

```
" Beispiel für ~/.vimrc
" Cursorposition mit der Maus festlegen
set mouse=a
" <Cursor links> am Zeilenanfang bewegt den Cursor an das Ende der vorigen Zeile,
" <Cursor rechts> am Zeilenende bewegt den Cursor an den Beginn der nächsten
" Zeile
set whichwrap=b,s,<,>[,]
" Backups (name~) beim Speichern erzeugen
set backup
" inkrementelle Suche aktivieren
set incsearch
" generell Leerzeichen statt Tabs einfügen
set expandtab
```

Die Konfigurierbarkeit von Vim geht aber noch viel weiter: Sie können Vim-Optionen für unterschiedliche Dateitypen unterschiedlich einstellen, neue Funktionen selbst programmieren etc. Erfinden Sie aber nicht das Rad neu! Unter der folgenden Adresse finden Sie zahllose fertige Lösungen, die Sie mit wenig Aufwand nutzen können. In der Regel reicht es aus, die betreffende Datei in `.vimrc` durch `source dateiname` einzubinden.

<http://www.vim.org/scripts>

Unabhängig von der Backup-Einstellung aktualisiert Vim während des Schreibens regelmäßig eine sogenannte Swap-Datei. Der Name dieser Datei ergibt sich aus einem vorangestellten Punkt, dem aktuellen Dateinamen sowie der Endung `.swp` (also beispielsweise `.mycode.c.swp`, wenn Sie gerade `mycode.c` bearbeiten). Diese Datei enthält in einem Binärformat alle Änderungen, die Sie seit dem letzten Speichern durchgeführt haben. Sie wird automatisch aktualisiert, wenn Sie mehr als 200 Zeichen neuen Text eingegeben haben oder vier Sekunden lang keine Eingaben durchgeführt haben. Die Swap-Datei wird beim regulären Beenden von Vim gelöscht.

Sollte der Strom ausfallen, Linux oder Vim abstürzen etc., können Sie Ihre ungesicherte Arbeit beim nächsten Vim-Start wiederherstellen. Vim bemerkt beim Öffnen der Datei, dass eine Swap-Datei existiert, und stellt verschiedene Optionen zur Auswahl. Im Regelfall werden Sie sich für `[W]` (Wiederherstellen) entscheiden und die so gerettete Datei anschließend speichern. Anschließend sollten Sie Vim verlassen und die Swap-Datei explizit löschen (das erfolgt nicht automatisch!).

Vim kommt standardmäßig mit den meisten wichtigen Zeichensätzen zurecht, unter anderem mit diversen Latin-Varianten, Unicode (utf-8, utf-16, ucs-2, ucs-2l, ucs-4 etc.) sowie einigen asiatischen 2-Byte-Zeichensätzen (z. B. `euc-kr`, also Koreanisch).

**Zeichensatz**

Vim ermittelt beim Start den Standardzeichensatz des Betriebssystems (Option `encoding`). Beim Lesen einer neuen Datei versucht Vim, auch deren Zeichensatz zu erkennen (Option `fileencoding`). Dabei werden der Reihe nach alle in der Option `fileencodings` aufgezählten Zeichensätze ausprobiert, bis ein Zeichensatz zur fehlerfreien Darstellung des gesamten Texts gefunden wird. (Die Grundeinstellung für `fileencodings` lautet oft `utf-8,latin1`.)

### Setzen Sie Vim möglichst in einer UTF8-Umgebung ein!

Wenn `encoding` und `fileencoding` nicht übereinstimmen, führt Vim beim Laden und Speichern automatisch eine Konvertierung durch. Falls der `encoding`-Zeichensatz weniger Zeichen darstellen kann als `fileencoding`, kann es dabei zu Verlusten kommen. Um das zu vermeiden, sollte Vim nach Möglichkeit in einer UTF-8-Umgebung eingesetzt werden. Unter Linux ist das standardmäßig der Fall.

Um herauszufinden, welchen Zeichensatz die gerade bearbeitete Datei nutzt, führen Sie `: set fileencoding?` aus. Mit `: set fileencoding=<neuerZeichensatz>` verändern Sie den Zeichensatz. Wenn Sie die Datei nun speichern, wird sie im neuen Zeichensatz gespeichert!

## 15.7 Tipps und Tricks

**:-Kommandos effizient eingeben** Bei der Eingabe von Kommandos, die mit `:` beginnen, gibt es einige Eingabehilfen: Mit den Cursortasten können Sie durch die zuletzt benutzten Kommandos blättern. (Vim speichert die Kommandos in `.viminfo` und merkt sich die Kommandos somit auch nach dem Programmende.) Weiters können Sie mit `⌘` Schlüsselwörter vervollständigen (z. B. bei der Eingabe von Optionen). Und zu guter Letzt können Sie viele `:`-Kommandos abkürzen (z. B. `: tabn` statt `: tabnext`).

**Zeilennummern anzeigen** `: set number` zeigt neben jeder Zeile die Zeilennummer an. `: set nonumber` deaktiviert diesen Modus wieder.

**Backups** Standardmäßig erstellt Vim beim Speichern kein Backup (also keine Kopie der ursprünglichen Datei). Wenn Sie das wünschen, führen Sie im Standardmodus `: set backup` aus. Die Backup-Datei erhält den Namen `altername~`. Um Backups generell zu aktivieren, fügen Sie `set backup` in `.vimrc` ein.

**Maus aktivieren** Wenn Sie Vim in einer Textkonsole oder in einem Konsolenfenster verwenden, dann ist die Funktion der Maus auf ihre Grundfunktionen unter X beschränkt: Sie können damit zwar Text kopieren und an der aktuellen Cursorposition einfügen, Sie können aber nicht die aktuelle Cursorposition verändern etc.

Um der Maus in Vim mehr Funktionen zu geben, verwenden Sie entweder die grafische Variante `gvim`, oder Sie führen im Standardmodus `: set mouse=a` aus. Sie können nun den Cursor durch einen Mausklick neu positionieren, das aktive Vim-Fenster auswählen, mit dem Mousrad durch den Text scrollen etc.

Dieser Mausmodus hat allerdings einen Nachteil: Die mittlere Maustaste fügt nun den zuletzt in Vim gelöscht Text ein. Die Maus kann nicht mehr zum Kopieren von Text zwischen Vim und anderen Programmen genutzt werden. Abhilfe ist aber einfach: Die herkömmlichen Mausfunktionen sind weiterhin verfügbar, wenn Sie zusätzlich die `⌘`-Taste drücken. Achten Sie aber darauf, dass sich Vim beim Einfügen von Text tatsächlich im Einfügemodus befindet! Andernfalls wird der per Maus eingefügte Text als Kommando interpretiert, und das kann schiefgehen.

**Leerzeichen statt Tabulatoren** Damit Vim in Ihren Text grundsätzlich Leerzeichen statt Tabulatorzeichen einfügt, führen Sie `: set expandtab` aus bzw. fügen die Anweisung in `.vimrc` ein. Um in der vorhandenen Datei alle Tabulatorzeichen durch die entsprechende Anzahl von

Leerzeichen zu ersetzen, führen Sie anschließend `:` `retab` aus. Um umgekehrt Leerzeichen durch Tabulatoren zu ersetzen, führen Sie `:` `set unexpandtab` und dann `:` `retab!` aus.

`.` wiederholt das letzte Kommando – so viel wissen Sie schon. Wenn Sie aber eine ganze Abfolge von Kommandos mehrfach ausführen möchten, definieren Sie ein Makro. Dazu starten Sie im Standardmodus mit `Q` den Makromodus. Das nächste Zeichen gibt den Namen des Makros an (genau genommen den Namen des Registers, in dem das Makro gespeichert wird). Alle weiteren Kommandos werden im Makro gespeichert, bis Sie die Eingabe abermals durch `Q` beenden. Das so aufgezeichnete Makro können Sie nun mit `@` `makroname` ausführen. (Wenn Sie Vim verlassen, gehen alle gespeicherten Makros verloren.)

Makros

Ein Beispiel: Die folgende Tastensequenz zeichnet das Makro `a` auf, das am Beginn und am Ende eines Wortes das Anführungszeichen `"` einfügt:

`Q` `A` `I` `"` `Esc` `E` `A` `"` `Esc` `Q`

Wenn der Cursor nun innerhalb eines Wortes steht und Sie `@`, `A` ausführen, wird dieses Wort in Anführungszeichen gestellt. `@`, `@` wiederholt das letzte Registerkommando, ohne dass Sie sich an den Makro- bzw. Registernamen erinnern müssen.

Wenn Sie in Vim ein Linux-Kommando ausführen möchten, ohne Vim zu verlassen, führen Sie im Standardmodus `:` `!kommandoname` aus (also beispielsweise `!ls`, um die Liste der Dateien im aktuellen Verzeichnis zu ermitteln). Vim zeigt das Ergebnis des Kommandos an. Mit `↵` gelangen Sie zurück in den Editor. Zur Ausführung mehrerer Kommandos öffnen Sie mit `:` `sh` eine neue Shell. Von dort gelangen Sie mit `Strg`+`D` zurück in den Editor.

Linux-Kommandos ausführen

Wenn Sie sich nicht an die verschiedenen Vim-Modi gewöhnen können, auf Vim aber nicht mehr verzichten möchten, starten Sie das Programm am besten mit `vim -y` bzw. führen `:` `set insertmode` aus. Damit verbleibt der Editor immer im Einfügemodus. Sie müssen nun jedes Kommando mit `Strg`+`0` einleiten. Um mehrere Kommandos auf einmal auszuführen, ist es auch möglich, mit `Strg`+`L` für längere Zeit in den Standardmodus zu wechseln und diesen mit `Esc` zu verlassen.

Vim im Easy-Modus verwenden

Noch ähnlicher zu anderen Editoren verhält sich Vim, wenn Sie `evim` starten: Damit wird der Editor in der grafischen Benutzeroberfläche `gvim` gestartet. Textmarkierungen können mit `⇧` und den Cursortasten durchgeführt werden. Texte werden mit `Strg`+`C` kopiert, mit `Strg`+`X` gelöscht und mit `Strg`+`V` wieder eingefügt. `evim` bezeichnet den Easy-Modus als »Vim for gumbies« (was auch immer ein *gumby* ist ...): Sie verlieren damit so viel von den Grundeigenschaften von Vim, dass es besser ist, gleich einen anderen Editor einzusetzen.





# Kapitel 16

## Emacs

Emacs einfach nur als Editor zu bezeichnen, greift zu kurz: Das Programm eignet sich nicht nur zur Bearbeitung von Texten, sondern auch als komplette Entwicklungsumgebung, als E-Mail-Programm etc. Für manche Anwender ist der Emacs gleichsam ein Ersatzbetriebssystem für alle Funktionen der alltäglichen Arbeit. Wenn Sie mit dem Emacs erst einmal umgehen können, werden Sie nie wieder einen anderen Editor benötigen. Es versteht sich eigentlich von selbst, dass ich fast das gesamte Buch seit der ersten Auflage mit dem Emacs geschrieben habe ...

Wo viel Licht ist, da gibt es bekanntlich auch Schatten: Die Bedienung des Emacs sieht auf den ersten Blick ein wenig abschreckend aus. Es wimmelt nur so von `[Strg]`- und `[Alt]`-Sequenzen, mit denen die zahllosen Kommandos aufgerufen werden. Nicht umsonst behaupten Spötter, der Name Emacs stünde für *Escape Meta Alt Control Shift*. Auch die Benutzeroberfläche (Menü, Symbolleiste) wirkt antiquiert, deutschsprachige Menüs fehlen überhaupt, die Konfiguration ist umständlich etc. Kurz und gut: Der Emacs ist ein Editor für Profis, die bereit sind, Zeit für die Einarbeitung zu investieren, und die sich nicht an Äußerlichkeiten stören.

### 16.1 Schnelleinstieg

Dieses Kapitel bezieht sich auf den GNU Emacs (siehe [Abbildung 16.1](#)) in der Version 25.n. Der Emacs-Fork XEmacs, der eine Weile eine größere Verbreitung hatte, spielt im Linux-Alltag keine nennenswerte Rolle mehr. Deutlich interessanter sind »kleine« Emacs-Varianten wie `jed`, `jmacs` (aus dem Paket `joe`), `jove` oder `zile`: Ihr Hauptvorteil besteht darin, dass ihr Ressourcenbedarf viel geringer ist. Damit eignen sich diese Programme ideal für Notfallsysteme oder für ältere Rechner mit langsamen CPUs und wenig Speicher etc.

Versionen

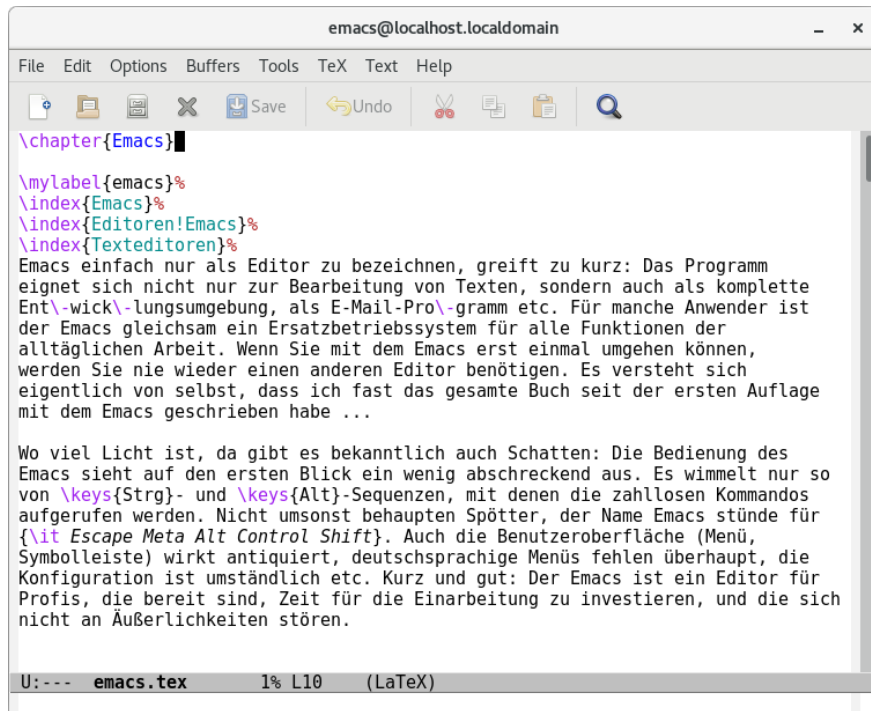


Abbildung 16.1 Der GNU Emacs

Links Weitere Informationen zum Emacs finden Sie hier:

<https://www.gnu.org/software/emacs/emacs.html>

<https://www.emacswiki.org>

### Texte laden und speichern, Programm beenden

Der Emacs wird durch die Eingabe von `emacs` gestartet. Wenn Sie beim Start des Programms einen oder mehrere Dateinamen angeben, werden diese Dateien automatisch geladen. Dabei sind auch Suchmuster erlaubt: `emacs Makefile *.ch` lädt die Datei `Makefile` sowie alle `*.c`- und `*.h`-Dateien des aktuellen Verzeichnisses. Sobald das Programm einmal läuft, laden Sie weitere Dateien mit `[Strg]+[X]`, `[Strg]+[F]` Dateiname `[↵]`.

Mit `[Strg]+[X]`, `[Strg]+[S]` speichern Sie die geänderte Datei. `[Strg]+[X]`, `[Strg]+[C]` beendet das Programm. Wenn der Emacs dabei irgendwelche noch nicht gespeicherten Dateien entdeckt, erscheint eine Sicherheitsabfrage, ob Sie den Emacs tatsächlich ohne zu speichern verlassen möchten. Antworten Sie auf diese Abfrage durch die Eingabe von `yes` `[↵]`, falls Sie die Änderungen tatsächlich verwerfen möchten. Um

eine Datei unter einem anderen Namen zu speichern, geben Sie `Strg+X`, `Strg+W` Dateiname `↵` ein.

| Tastenkürzel   | Funktion  |
|--|---|
| <code>Strg+X</code> , <code>Strg+F</code> datei <code>↵</code> | lädt eine Datei (Find).                               |
| <code>Strg+X</code> , <code>I</code>                           | fügt eine Datei in den vorhandenen Text ein (Insert). |
| <code>Strg+X</code> , <code>Strg+S</code>                      | speichert eine Datei (Save).                          |
| <code>Strg+X</code> , <code>S</code>                           | speichert alle Dateien (mit Rückfrage).               |
| <code>Strg+X</code> , <code>S</code> , <code>!</code>          | speichert alle offenen Dateien (ohne Rückfrage).      |
| <code>Strg+X</code> , <code>Strg+W</code> datei <code>↵</code> | speichert unter einem neuen Namen (Write).            |
| <code>Strg+X</code> , <code>Strg+C</code>                      | beendet den Editor.                                   |

**Tabelle 16.1** Dateien laden und speichern, Emacs beenden

Wenn Sie den Emacs in einer Textkonsole verwenden, können Sie das Programm mit `Strg+Z` vorübergehend verlassen. Mit `fg` nehmen Sie die Arbeit wieder auf. Im Grafikmodus bewirkt `Strg+Z` lediglich die Verkleinerung in ein Icon.

Der Emacs erstellt beim Speichern automatisch eine Sicherheitskopie `name~`, in der der ursprüngliche Text enthalten ist. Außerdem speichert der Emacs in regelmäßigen Abständen den aktuellen Zustand des Textes in der Datei `#name#`. Auf diese Datei können Sie zurückgreifen, wenn während des Arbeitens der Strom ausgefallen ist oder wenn Sie aus irgendeinem anderen Grund den Emacs nicht ordnungsgemäß verlassen konnten.

Sicherheitskopien

Beachten Sie, dass die #-Dateien im Emacs-internen Zeichensatz gespeichert werden und nicht in dem Zeichensatz, in dem Sie Ihre Datei bearbeitet haben. Aus diesem Grund sollten Sie zur Wiederherstellung der Dateien `Alt+X` `recover-session` einsetzen. Alternativ können Sie auch direkt die #-Dateien laden und mit `Strg+X`, `↵`, `F` Zeichensatz deren Zeichensatz verändern.

## Elementare Kommandos

Üblicherweise bewegen Sie den Cursor mit den Cursortasten sowie mit `Bild↑` bzw. `Bild↓`. Sollte das nicht funktionieren (z. B. wenn Sie den Emacs über ein schlecht funktionierendes Terminalprogramm gestartet haben), klappt es auf jeden Fall mit den Kommandos, die in [Tabelle 16.2](#) zusammengefasst sind.

Sie können an jeder beliebigen Stelle neuen Text eingeben. Mit `Entf` und `←` löschen Sie einzelne Zeichen. Alternativ existiert das Tastaturkommando `Strg+D` zum Löschen des Zeichens an der Cursorposition (*delete*).

| Tastenkürzel        | Funktion   |
|---------------------|--|
| <code>Strg+F</code> | bewegt den Cursor ein Zeichen nach links (Forwards).   |
| <code>Strg+B</code> | bewegt den Cursor ein Zeichen nach rechts (Backwards). |
| <code>Strg+P</code> | bewegt den Cursor eine Zeile nach oben (Previous).     |
| <code>Strg+N</code> | bewegt den Cursor eine Zeile nach unten (Next).        |
| <code>Strg+V</code> | bewegt den Text eine Seite nach oben.                  |
| <code>Alt+V</code>  | bewegt den Text eine Seite nach unten.                 |

**Tabelle 16.2** Tastenkürzel, falls die Cursortasten versagen

**Undo** Mit `Strg+X`, `U` (*undo*) oder mit `Strg+_`, im deutschen Tastaturlayout also `Strg+⬠+⬡`, widerrufen Sie die letzten Änderungen. Diese Undo-Funktion funktioniert für beliebig komplexe Kommandos und praktisch unbegrenzt!

Wenn Ihnen während der Eingabe eines Kommandos ein Fehler unterläuft, können Sie die Kommandoingabe mit `Strg+G` abbrechen. Das ist besonders dann praktisch, wenn Sie irrtümlich `Esc` drücken.

### Online-Hilfe

Der Emacs stellt zahlreiche Kommandos zum Aufruf der englischsprachigen Online-Hilfe zur Verfügung. Das für den Einstieg wichtigste Kommando lautet `F1`, `T` (Tutorial). Mit `Strg+X`, `B`, `↵` gelangen Sie in den ursprünglichen Text zurück.

Wenn nach der Ausführung eines Hilfe-Kommandos mehrere Textabschnitte (Fenster) übrig bleiben, können Sie mit `Strg+X`, `0` (»Oh«) den Textcursor in das jeweils nächste Fenster stellen. `Strg+X`, `0` (»Null«) entfernt das aktuelle Fenster; `Strg+X`, `1` löscht alle Fenster außer dem aktuellen Fenster. Mit den drei Kommandos können Sie also zwischen dem Hilfe- und dem Textfenster hin- und herspringen und schließlich das Hilfefenster wieder entfernen.

Wird der Hilfetext dagegen seitenfüllend angezeigt, können Sie mit `Strg+X`, `B`, `↵` zurück in Ihren eigentlichen Text springen. Intern wird die Verwaltung mehrerer Texte – also beispielsweise Ihres Textes und des Hilfetextes – durch sogenannte Puffer realisiert (siehe [Abschnitt 16.8](#), »Puffer und Fenster«).

Die wichtigste Informationsquelle zum Emacs ist das interne `info`-System, das offiziell als Emacs-Handbuch gilt. Bei manchen Distributionen wird dieses Handbuch auch im HTML-Format mitgeliefert, sodass es noch komfortabler gelesen werden kann.

| Tastenkürzel  | Funktion  |
|---|---|
| <code>F1</code> , <code>F1</code>                         | Übersicht über vorhandene Hilfefunktionen   |
| <code>F1</code> , <code>A</code> text <code>↵</code>      | Übersicht über alle Kommandos, die text enthalten (Apropos)   |
| <code>F1</code> , <code>B</code>                          | Übersicht über alle Tastenkürzel (Bindings)   |
| <code>F1</code> , <code>C</code> tastenkürzel             | Kurzbeschreibung des zugeordneten Kommandos (Command)   |
| <code>F1</code> , <code>F</code> kommando <code>↵</code>  | Kurzbeschreibung des Kommandos (Function)   |
| <code>F1</code> , <code>↕</code> + <code>F</code>         | Emacs-FAQ (Frequently Asked Questions)  |
| <code>F1</code> , <code>I</code>                          | startet das info-System zur Anzeige hierarchischer Hilfetexte (zur Bedienung siehe <a href="#">Abschnitt 9.3</a> , »man und info«). |
| <code>F1</code> , <code>N</code>                          | Zusammenfassung der Neuerungen in der aktuellen Version im Vergleich zu den früheren Versionen                                      |
| <code>F1</code> , <code>T</code>                          | Einführung in die Bedienung von Emacs (Tutorial)  |
| <code>F1</code> , <code>Strg</code> + <code>F</code> name | startet das info-System und zeigt Informationen zum angegebenen Kommando an.  |
| <code>F1</code> , <code>Strg</code> + <code>P</code>      | Informationen über die Idee freier Software   |

Tabelle 16.3 Online-Dokumentation nutzen

Die eingebaute Online-Hilfe des Emacs liegt im Info-Format vor. Beim Lesen von info-Texten wird im Emacs ein eigener info-Modus aktiviert. Querverweise bzw. Menüeinträge können Sie einfach durch Klicken mit der mittleren Maustaste verfolgen. Mit `L` gelangen Sie zur zuletzt sichtbaren Seite zurück.

Info-Modus

## 16.2 Grundlagen

Der Emacs kennt verschiedene Bearbeitungsmodi, in denen zusätzliche Kommandos zur Bearbeitung spezieller Dateien zur Verfügung stehen. Dabei wird zwischen Haupt- und Nebenmodi unterschieden: Es kann immer nur ein Hauptmodus aktiv sein. Dieser kann aber durch mehrere Nebenmodi ergänzt werden.

Bearbeitungsmodi

Zu den wichtigsten Hauptmodi zählen solche für fast alle gängigen Programmiersprachen (C, C++, Java etc.) sowie der  $\text{\LaTeX}$ -Modus zur Bearbeitung von  $\text{\LaTeX}$ -Dateien. Der Emacs aktiviert beim Laden einer Datei automatisch den Modus, der ihm passend erscheint (z. B. den C-Modus, wenn der Dateiname auf `.c` endet). Wenn der Emacs keinen passenden Modus erkennen kann, wählt er den Fundamental-Modus als Grundeinstellung.

Zu den wichtigsten Nebenmodi gehören der Fill-Modus zur Bearbeitung von Fließtext mit Absätzen über mehrere Zeilen und der Abbrev-Modus zur automatischen Auflösung von Abkürzungen.

Die elementaren Emacs-Kommandos funktionieren in allen Modi gleich, weswegen Sie sich mit den Bearbeitungsmodi vorläufig noch nicht beschäftigen müssen. Wenn Sie Eigenmächtigkeiten des Emacs aufgrund eines bestimmten Modus deaktivieren möchten (z. B. das automatische Einrücken von Programmzeilen im C-Modus), schalten Sie einfach mit `[Alt]+[X]` `fundamental-mode` `[←]` in den Grundmodus um. Genauere Informationen zu den Bearbeitungsmodi finden Sie in [Abschnitt 16.9](#).

#### Tastaturkonventionen

Generell gibt es drei Möglichkeiten zur Eingabe von Emacs-Kommandos: das Menü, die Verwendung von Tastenkürzeln (zumeist eine Kombination mit `[Strg]` oder `[Alt]`) oder die Eingabe des gesamten Kommandonamens. Die dritte Variante wird mit `[Alt]+[X]` eingeleitet, also etwa `[Alt]+[X]` `delete-char` `[←]`.

Die Eingabe von Kommandos und anderen Parametern wird durch zwei Mechanismen erleichtert:

- ▶ Während der Eingabe können Sie den Kommandonamen wie bei der Kommando-eingabe im Shell-Terminal mit `[↵]` ergänzen. Der Emacs unterscheidet dabei zwischen Groß- und Kleinschreibung. In gleicher Weise können auch Dateinamen ergänzt werden. Wenn mehrere Möglichkeiten bestehen, zeigt der Emacs diese auf dem Bildschirm an.
- ▶ Auf früher bei `[Alt]+[X]` angegebene Kommandos können Sie (nach der Einleitung des neuen Kommandos durch `[Alt]+[X]`) mit `[Alt]+[P]` (*Previous*) und `[Alt]+[N]` (*Next*) zurückgreifen.

In diesem Buch werden die Tastenfolgen so angegeben, wie sie auf einer deutschen Tastatur bei korrekter Installation eingegeben werden können. Dabei bedeutet ein Plus-Zeichen, dass mehrere Tasten gleichzeitig gedrückt werden müssen, während ein Komma darauf hinweist, dass die Tasten nacheinander gedrückt werden. Buchstaben werden immer als Großbuchstaben angegeben, obwohl die `[⇧]`-Taste dabei nicht gedrückt werden muss! `[Alt]+[X]` bedeutet also, dass Sie die Tasten `[Alt]` und `[X]` gleichzeitig drücken sollen, nicht aber `[⇧]`!

In der Dokumentation zum Emacs werden Tastenkürzel etwas abweichend dargestellt: DEL bedeutet nicht `[Entf]`, sondern `[←]`! C steht für Control (gemeint ist `[Strg]`) und M für `[Meta]`.

Eine direkte Entsprechung der Meta-Taste existiert auf einer Standard-PC-Tastatur nicht. M-x kann auf einer PC-Tastatur auf zwei Weisen nachgebildet werden: durch `[Esc]` und `[X]` (nacheinander) oder durch `[Alt]+[X]`. In diesem Buch wird generell die bequemere `[Alt]`-Tastenkombination angegeben.

Bei manchen Emacs-kompatiblen Programmen bzw. bei der Verwendung des Emacs in einer Textkonsole gibt es allerdings Probleme mit der Taste `[Alt]`. Statt `[Alt]+[X]` müssen Sie dort `[Esc]`, `[X]` benutzen. Beachten Sie, dass der Emacs zwischen `[Strg]+[X]`, `[Strg]+[B]` und der ähnlich aussehenden Kombination `[Strg]+[X]`, `[B]` unterscheidet! Es ist also nicht egal, wie lange Sie die `[Strg]`-Taste gedrückt halten.

Feinheiten

In Emacs gelten die unter X üblichen Konventionen, d. h., Sie markieren Text mit der Maus und fügen ihn dann mit der mittleren Maustaste wieder ein. Wenn Sie im Emacs mehrere Texte gleichzeitig anzeigen, können Sie auch die Trennleiste zwischen den Textbereichen mit der linken Maustaste verschieben. Mit der rechten Maustaste stellen Sie den Endpunkt des gerade markierten Textbereichs ein, der danach bearbeitet werden kann. Die Maustasten in Kombination mit `[⇧]` bzw. `[Strg]` dienen zur Ausführung diverser Kommandos (z. B. zur Auswahl des Fonts, der zur Darstellung des Texts verwendet wird).

Mausunterstützung

Beim Start des Emacs unter X können Sie durch Kommandozeilenoptionen zahlreiche Einstellungen für Farben, Zeichensätze etc. vornehmen. [Tabelle 16.4](#) zählt die wichtigsten Optionen auf. Eine vollständige Beschreibung finden Sie in der Manual-Seite zum Emacs.

Startoptionen

| Option            | Bedeutung  |
|-------------------|--|
| -nw               | Textversion des Emacs im Shell-Fenster starten (No Window)   |
| -fg farbe         | Vordergrundfarbe (Textfarbe; normalerweise Schwarz)  |
| -bg farbe         | Hintergrundfarbe (normalerweise Weiß)  |
| -cr farbe         | Farbe des Textcursors (normalerweise Schwarz)  |
| -geometry bxh+x+y | Größe (Breite mal Höhe) und Position des Emacs-Fensters voreinstellen; alle Angaben in Textzeichen |
| -fn Zeichensatz   | startet Emacs mit dem angegebenen Zeichensatz.   |

Tabelle 16.4 Kommandozeilenoptionen

## 16.3 Cursorbewegung

Neben den Cursortasten kennt Emacs eine Menge Tastenkürzel zur Cursorbewegung. Die wichtigsten Kürzel sind in [Tabelle 16.5](#) zusammengefasst.

Der Emacs ist in der Lage, ein beliebiges Kommando mehrfach hintereinander auszuführen. Dazu müssen Sie zuerst `[Alt]+n` eingeben, wobei  $n$  eine beliebige Zahl ist. Die Ziffern müssen vom alphanumerischen Tastaturteil stammen (nicht vom Zehnerblock im rechten Teil der Tastatur). Während der gesamten Zahleneingabe müssen

Cursorkommandos mehrfach ausführen

Sie `Alt` gedrückt halten. Anschließend geben Sie das gewünschte Kommando an. Beispielsweise wird der Text durch `Alt+n`, `Bild↓` um  $n$  Seiten nach unten gescrollt. Dieses Verfahren kann auch zur Eingabe von Textzeichen verwendet werden. Beispielsweise zeichnet `Alt+60`, `-` eine Linie.

| Tastenkürzel   | Funktion   |
|--|--|
| <code>Alt+F</code> / <code>Alt+B</code>                                | bewegt den Cursor ein Wort vor bzw. zurück (For-/Backwards).       |
| <code>Strg+A</code> / <code>Strg+E</code>                              | stellt den Cursor an den Beginn bzw. das Ende der Zeile.           |
| <code>Alt+A</code> / <code>Alt+E</code>                                | stellt den Cursor an den Beginn bzw. das Ende des Absatzes.        |
| <code>Strg+V</code> / <code>Alt+V</code>                               | bewegt den Text eine Seite nach unten bzw. oben.                   |
| <code>Alt+&lt;</code> / <code>Alt+↶+&gt;</code>                        | bewegt den Cursor an den Beginn bzw. das Ende des Textes.          |
| <code>Strg+L</code>  | scrollt den Text so, dass der Cursor in der Bildmitte steht.       |
| <code>Alt+G n ↵</code>   | stellt den Cursor in Zeile $n$ .                                   |
| <code>Strg+X</code> , <code>R</code> , Leertaste <code>z ↵</code>      | speichert die aktuelle Cursorposition im Register <code>z</code> . |
| <code>Strg+X</code> , <code>R</code> , <code>J</code> <code>z ↵</code> | springt zu der im Register <code>z</code> gespeicherten Position.  |

Tabelle 16.5 Cursorbewegung

**Wo bin ich?** Wenn Sie wissen möchten, in welcher Zeile Sie sich gerade befinden, geben Sie `Alt+X` `what-line ↵` ein. Der Emacs zeigt jetzt die aktuelle Zeilennummer in der untersten Bildschirmzeile an. Noch praktischer ist es, mit `Alt+X` `line-number-mode ↵` eine ständige Anzeige der Zeilennummer zu aktivieren. Leider funktioniert diese Anzeige bei sehr langen Texten (im MiB-Bereich) nicht mehr. Natürlich kann auch die Spaltennummer angezeigt werden – aktivieren Sie den `column-number-mode`!

#### Cursorpositionen in Registern speichern

In einem längeren Text wünscht man sich oft, rasch zwischen verschiedenen Stellen im Text hin- und herspringen zu können. Zu diesem Zweck kann die aktuelle Cursorposition mit einem Kommando in einem sogenannten Register gespeichert werden (siehe die vorletzte Zeile in [Tabelle 16.5](#)). Ein Register ist ein Speicherplatz, der durch ein Textzeichen (Buchstabe oder Ziffer) gekennzeichnet wird. Zu einem späteren Zeitpunkt können Sie durch die Angabe dieses Registers wieder an den ursprünglich gespeicherten Ort springen. Beachten Sie bitte, dass Register beim Verlassen des Emacs nicht gespeichert werden.



## 16.4 Text markieren, löschen und einfügen

Die Tasten `[Entf]` oder `[Strg]+[D]` sowie `[←]` zum Löschen einzelner Zeichen haben Sie schon kennengelernt. Um größere Textmengen zu löschen, setzen Sie die in [Tabelle 16.6](#) zusammengefassten Kommandos ein. Wenn Sie die dort aufgezählten Löschkommandos mehrmals unmittelbar hintereinander ausführen, fügt `[Strg]+[Y]` den gesamten gelöschten Text wieder ein. `[Strg]+[Y]` kann mehrfach und an beliebigen Stellen im Text ausgeführt werden. Das Kommando ermöglicht es daher, den gelöschten Text an eine andere Stelle zu verschieben bzw. zu kopieren.

| Tastenkürzel                                     | Funktion  |
|--|---|
| <code>[Alt]+[D]</code>                           | löscht das nächste Wort bzw. das Ende des Wortes ab dem Cursor.                         |
| <code>[Alt]+[←]</code>                           | löscht das vorige Wort bzw. den Beginn des Wortes bis zum Cursor.                       |
| <code>[Strg]+[K]</code>                          | löscht das Zeilenende ab der Cursorposition.  |
| <code>[Alt]+[0]</code> , <code>[Strg]+[K]</code> | löscht den Zeilenanfang vor der Cursorposition.   |
| <code>[Alt]+[M]</code>                           | löscht den nächsten Absatz.   |
| <code>[Alt]+[Z]</code> , x                       | löscht alle Zeichen bis zum nächsten Auftreten von x (das Zeichen x wird mit gelöscht). |
| <code>[Strg]+[Y]</code>                          | fügt den zuletzt gelöschten Text an der Cursorposition wieder ein.                      |

**Tabelle 16.6** Text löschen und wieder einfügen

Die obigen Kommandos sind relativ unflexibel, weil die zu löschende Textmenge starr vorgegeben ist. Wenn Sie einen beliebigen Textausschnitt löschen möchten, markieren Sie diesen zuvor. Dazu führen Sie zuerst am Anfang oder am Ende des Bereichs `[Strg]+Leertaste` aus. Diese Markierung bleibt unsichtbar, der Emacs zeigt aber die Meldung »Mark set« an. Als markierter Bereich gilt von nun an der Text zwischen dem markierten Punkt und der aktuellen Position des Textcursors.

| Tastenkürzel                                      | Funktion  |
|---|---|
| <code>[Strg]+Leertaste</code>                     | setzt einen (unsichtbaren) Markierungspunkt.                                    |
| <code>[Strg]+[W]</code>                           | löscht den Text zwischen dem Markierungspunkt und der aktuellen Cursorposition. |
| <code>[Strg]+[Y]</code>                           | fügt den gelöschten Text wieder ein.  |
| <code>[Strg]+[X]</code> , <code>[Strg]+[X]</code> | vertauscht Cursorposition und Markierungspunkt.                                 |

**Tabelle 16.7** Text markieren

Wenn Sie sich nicht an die Bereichsmarkierung mit `Strg+X` gewöhnen möchten, können Sie im Emacs auch die unter Windows übliche Form der Markierung mit `⇧` aktivieren. Ab Emacs 23.1 steht diese Markiermethode standardmäßig zur Verfügung. Bei älteren Emacs-Versionen müssen Sie vorher `Alt+X` `pc-selection-mode` ausführen bzw. die Emacs-Konfiguration entsprechend verändern.

**CUA-Modus** Mit `Alt+X` `cua-mode` können Sie den Common-User-Access-Modus aktivieren. Sofern Sie mit `⇧` Text markiert haben, können Sie diesen wie in nahezu jedem anderen Programm mit `Strg+C` in die Zwischenablage kopieren bzw. mit `Strg+X` ausschneiden. `Strg+V` fügt den Inhalt der Zwischenablage an der aktuellen Cursorposition wieder ein. Wenn kein Text markiert ist, leitet `Strg+X` wie bisher diverse Emacs-Kommandos ein.

## 16.5 Text bearbeiten

**Text einfügen bzw. überschreiben** Der Emacs befindet sich normalerweise im Einfügemodus. Das heißt, neu eingegebener Text wird an der aktuellen Cursorposition in den vorhandenen Text eingefügt. Wenn Sie stattdessen den vorhandenen Text überschreiben möchten, wechseln Sie mit `Alt+X` `overwrite-mode` `↵` in den Überschreibmodus. Die nochmalige Ausführung des Kommandos schaltet den Modus wieder aus. Bei einer korrekten Konfiguration der Tastatur können Sie den Modus auch mit `Einfg` umschalten.

Zur Veränderung der Groß- und Kleinschreibung bereits geschriebener Wörter bietet der Emacs die Kommandos an, die in [Tabelle 16.8](#) zusammengefasst sind.

**Tippfehler** Ein häufiger Tippfehler ist das Vertauschen zweier Buchstaben. Mit `Strg+T` können Sie solche Vertauschungen bequem korrigieren. Der Cursor muss dabei auf dem zweiten der beiden betroffenen Buchstaben stehen, im Wort »vertauschcen« also auf »c«.


Analog können mit `Alt+T` zwei Wörter vertauscht werden. Wenn der Cursor dabei am Beginn eines Wortes steht, wird dieses Wort mit dem vorangegangenen vertauscht. Steht der Cursor dagegen irgendwo im Wort, dann wird das Wort mit dem folgenden Wort vertauscht. Das mehrfache Ausführen von `Alt+T` führt dazu, dass das erste der beiden Wörter immer weiter nach vorn bewegt wird.




Mit `Strg+X`, `Strg+T` vertauschen Sie schließlich die aktuelle Zeile mit der vorherigen Zeile. Die mehrfache Ausführung des Kommandos führt dazu, dass die Zeile oberhalb des Curors immer weiter nach unten rutscht.


| Tastenkürzel  | Funktion   |
|---|--|
| <b>A</b> l <b>t</b> + <b>C</b>                                    | Buchstabe an der Cursorposition groß, alle weiteren Buchstaben des aktuellen Wortes klein ( <i>capitalize</i> )                    |
| <b>A</b> l <b>t</b> + <b>L</b>                                    | alle Buchstaben des Wortes ab Cursorposition klein ( <i>lower</i> )  |
| <b>A</b> l <b>t</b> + <b>U</b>                                    | alle Buchstaben des Wortes ab Cursorposition groß ( <i>upper</i> )   |
| <b>E</b> sc, <b>-</b> , <b>A</b> l <b>t</b> + <b>C</b>            | erster Buchstabe groß, Rest klein. Wenn der Cursor am Beginn eines Wortes steht, wird das vorige Wort verändert.                   |
| <b>E</b> sc, <b>-</b> , <b>A</b> l <b>t</b> + <b>L</b>            | alle Buchstaben des Wortes bis zur Cursorposition klein; wenn der Cursor am Beginn eines Wortes steht, dann das vorige Wort klein. |
| <b>E</b> sc, <b>-</b> , <b>A</b> l <b>t</b> + <b>U</b>            | alle Buchstaben des Wortes bis zur Cursorposition groß; wenn der Cursor am Beginn eines Wortes steht, dann das vorige Wort groß.   |
| <b>S</b> tr <b>g</b> + Leertaste                                  | Markierungspunkt setzen  |
| <b>S</b> tr <b>g</b> + <b>X</b> , <b>S</b> tr <b>g</b> + <b>L</b> | Bereich zwischen Markierungspunkt und Cursor klein   |
| <b>S</b> tr <b>g</b> + <b>X</b> , <b>S</b> tr <b>g</b> + <b>U</b> | Bereich zwischen Markierungspunkt und Cursor groß  |

Tabelle 16.8 Groß- und Kleinschreibung ändern

## Tabulatoren

In der Grundeinstellung und bei der Bearbeitung eines normalen ASCII-Texts wird durch  ein Tabulatorzeichen eingefügt. Tabulatoren sind nicht sichtbar. Ob an einer Stelle ein Tabulatorzeichen oder mehrere Leerzeichen stehen, merken Sie erst, wenn Sie den Cursor darüber bewegen. Bei Tabulatoren bewegt sich der Cursor in Sprüngen.

Je nachdem, welchen Text (z. B. eine \*.tex-Datei) Sie mit dem Emacs bearbeiten, wird automatisch ein dazu passender Bearbeitungsmodus aktiviert (siehe [Abschnitt 16.9](#), »Besondere Bearbeitungsmodi«). Bei manchen dieser Modi werden einzelne Tasten undefiniert. Dies betrifft insbesondere auch die Taste . Im C-Modus bewirkt die Taste beispielsweise, dass der Zeilenanfang entsprechend der Programmstruktur eingerückt wird. Im  $\text{\LaTeX}$ -Modus hat die -Taste gar keine Wirkung. Wenn  also nicht so funktioniert, wie Sie erwarten, ist zumeist der Bearbeitungsmodus schuld. Es bestehen mehrere Möglichkeiten, dennoch Tabulatoren einzugeben:

- ▶ Mit **S**tr**g** + **Q**,  können Sie unabhängig von allen Modi ein Tabulatorzeichen in den Text einfügen.
- ▶ Mit **A**l**t** + **I** können Sie unabhängig vom Bearbeitungsmodus ein Tabulatorzeichen oder entsprechend viele Leerzeichen einfügen (je nach Einstellung von `indent-tabs-mode`, siehe unten).

- ▶ Mit `[Alt]+[X]` fundamental-mode können Sie den gerade aktuellen Bearbeitungsmodus deaktivieren. Dann funktioniert `[Esc]` wie in anderen Programmen gewohnt, allerdings verlieren Sie gleichzeitig auch alle Spezialfunktionen des bisher gültigen Bearbeitungsmodus.

**Tabulatorweite** Als Tabulatorweite gelten normalerweise acht Zeichen. Mit `[Alt]+[X]` set-variable tab-width können Sie aber auch eine andere Tabulatorweite einstellen. Wenn Sie generell mit vier statt mit acht Zeichen pro Tabulator arbeiten möchten, können Sie diese Einstellung auch in der Konfigurationsdatei `~/ .emacs` vornehmen.

**Wechsel zwischen Leerzeichen und Tabulatoren** In einigen Bearbeitungsmodi ersetzt der Emacs automatisch lange Folgen von Leerzeichen durch Tabulatoren. Mit den beiden Kommandos `[Alt]+[X]` tabify können Sie im vorher markierten Bereich alle Leerzeichenserien durch Tabulatorzeichen ersetzen. `[Alt]+[X]` untabify funktioniert genau umgekehrt und ersetzt Tabulatoren durch eine ausreichende Anzahl von Leerzeichen.

**indent-tabs-mode** Wenn Sie in die Konfigurationsdatei `.emacs` die folgende Zeile einbauen, dann fügt der Emacs generell statt Tabulatorzeichen Leerzeichen ein:

```
(setq-default indent-tabs-mode nil)
```

### Text manuell ein- und ausrücken

Das Ein- und Ausrücken von Text ist insbesondere in Programmlistings zur Strukturierung des Codes erforderlich. Das wichtigste Kommando wird mit `[Strg]+[X]`, `[Esc]` aufgerufen (siehe [Tabelle 16.9](#)). Es rückt den Text zwischen dem Markierungspunkt (`[Strg]+Leertaste`) und der aktuellen Cursorposition um ein Leerzeichen ein. Wenn Sie vor diesem Kommando `[Alt]+n` ausführen, wird der markierte Textbereich um  $n$  Zeichen eingerückt. Durch ein vorangestelltes `[Esc]`, `[_]` wird der Text aus- statt eingerückt.

| Tastenkürzel  | Funktion   |
|---|--|
| <code>[Strg]+Leertaste</code>   | Markierungspunkt setzen  |
| <code>[Strg]+[X]</code> , <code>[Esc]</code>  | Text zwischen Markierungspunkt und Cursorposition um ein Zeichen einrücken |
| <code>[Esc]</code> , <code>[_]</code> , <code>[Strg]+[X]</code> , <code>[Esc]</code>                        | Text um ein Zeichen ausrücken  |
| <code>[Alt]+n</code> , <code>[Strg]+[X]</code> , <code>[Esc]</code>   | Text um $n$ Zeichen einrücken  |
| <code>[Esc]</code> , <code>[_]</code> , <code>[Alt]+n</code> , <code>[Strg]+[X]</code> , <code>[Esc]</code> | Text um $n$ Zeichen ausrücken  |

**Tabelle 16.9** Text ein- und ausrücken

**Rechteck-Kommandos** Wenn Sie rechteckige Textblöcke innerhalb von Zeilen einfügen oder löschen möchten (etwa bei der Bearbeitung von Tabellen oder zum Ein- oder Ausrücken von Kommentaren am Ende von Programmzeilen), müssen Sie mit den sogenannten

Rechteck-Kommandos arbeiten (siehe [Tabelle 16.10](#)). Als Rechteck gelten dabei alle Zeichen im Bereich zwischen dem Markierungspunkt und der Cursorposition.

Noch komfortabler lassen sich rechteckige Textblöcke im CUA-Modus bearbeiten: Wenn dieser Modus aktiv ist, beginnen Sie die Rechteckmarkierung mit `Strg + ⌘`. Anschließend können Sie mit `Entf` Zeichen in allen Zeilen löschen bzw. mit allen anderen Tasten neuen Text einfügen.

| Tastenkürzel   | Funktion   |
|--|--|
| <code>Strg</code> + Leertaste  | Markierungspunkt setzen  |
| <code>Strg</code> + <code>X</code> , <code>R</code> , <code>O</code> | rechteckigen Bereich öffnen ( <i>Rectangle Open</i> ), d. h., in den rechteckigen Bereich Leerzeichen einfügen |
| <code>Strg</code> + <code>X</code> , <code>R</code> , <code>K</code> | rechteckigen Bereich löschen ( <i>Rectangle Kill</i> )   |
| <code>Strg</code> + <code>X</code> , <code>R</code> , <code>Y</code> | gelöschten rechteckigen Bereich an der Cursorposition einfügen ( <i>Rectangle Yank</i> )                       |
| <code>Alt</code> + <code>X</code> string-rectangle                   | einen Text vor jede Zeile des markierten Bereichs einfügen   |
| <code>Strg</code> + <code>⌘</code>                                   | Rechteck-Markierung im CUA-Modus   |

**Tabelle 16.10** Rechteck-Kommandos

Der Emacs kennt darüber hinaus einige Bearbeitungsmodi, in denen Einrückungen automatisch durchgeführt werden. So werden im C-Modus Programmzeilen bei jeder geschweiften Klammer { oder } um einige Leerzeichen ein- oder ausgerückt (siehe [Abschnitt 16.9](#), »Besondere Bearbeitungsmodi«).

## 16.6 Fließtext

Bisher habe ich angenommen, dass Sie mit dem Emacs Programmcode, Konfigurationsdateien etc. bearbeiten. Ein wenig anders sieht der Umgang mit dem Emacs aus, wenn Sie Fließtext bearbeiten möchten. Der Emacs führt normalerweise keinen automatischen Umbruch durch. Wenn Zeilen länger sind als die Bildschirm- oder Fensterbreite, dann wird am linken Ende ein \-Zeichen dargestellt und der Text in der nächsten Zeile fortgesetzt.

Wenn Sie eine einzelne längere Zeile umbrechen möchten, führen Sie das Kommando `Alt + Q` aus: Damit werden an geeigneten Stellen Leerzeichen durch Zeilenumbrüche ersetzt. Aus einer langen Zeile werden so mehrere kurze Zeilen. Dabei betrachtet der Emacs alle Zeilen, die nicht explizit durch eine vollkommen leere Zeile von anderen Zeilen getrennt sind, als einen Absatz. Bei einem Programmlisting sind die Folgen dieses Kommandos natürlich fatal! Führen Sie mit `Strg + X`, `U` ein Undo durch.

### Zeilenumbruch in $\LaTeX$ -Dokumenten

Wenn Sie  $\TeX$ - oder  $\LaTeX$ -Dateien bearbeiten, gilt für  $\text{Alt}+\text{Q}$  eine Besonderheit: Zeilen, die mit einem  $\backslash$ -Zeichen beginnen, gelten als Absatzgrenze und werden nicht umbrochen. Um einen Umbruch dennoch durchzuführen, müssen Sie diese Zeile manuell mit der vorhergehenden Zeile verbinden und nochmals  $\text{Alt}+\text{Q}$  ausführen. Noch bequemer ist es, das AUC-TEX-Paket zu installieren und zu aktivieren: Dann versteht der Emacs  $\LaTeX$  besser und führt den Zeilenumbruch intelligenter durch.

Bei der Eingabe eines neuen Textes ist es natürlich lästig, ständig  $\text{Alt}+\text{Q}$  zu drücken. Daher existiert ein eigener Fließtextmodus, der mit  $\text{Alt}+\text{X}$  `auto-fill-mode`  $\leftarrow$  aktiviert wird (siehe [Tabelle 16.11](#)). Wenn sich der Emacs in diesem Modus befindet, werden alle Neueingaben automatisch umbrochen. Bereits vorhandener Text wird durch diesen Modus nicht verändert. Auch das Löschen von Text führt nicht zu einem automatischen Umbruch, weswegen nach Änderungen in einem bereits vorhandenen Fließtext häufig ein manueller Umbruch mit  $\text{Alt}+\text{Q}$  erzwungen werden muss. Der Umbruch erfolgt normalerweise spätestens nach 70 Zeichen. Sie können die Umbruchspalte mit dem folgenden Kommando verändern:  $\text{Alt}+\text{X}$  `set-variable`  $\leftarrow$  `fill-column`  $\leftarrow$   $n$   $\leftarrow$ .

| Tastenkürzel   | Funktion  |
|--|---|
| $\text{Alt}+\text{Q}$  | führt einen manuellen Zeilenumbruch durch.                  |
| $\text{Alt}+\text{X}$ <code>auto-fill-mode</code> $\leftarrow$ | aktiviert den Fließtextmodus (automatischer Zeilenumbruch). |

**Tabelle 16.11** Fließtext umbrechen

Wenn Sie mehrere Absätze eingerückten Textes eingeben möchten, können Sie die erste gültige Spalte voreinstellen. Dazu müssen Sie so viele Leer- oder Tabulatorzeichen in einer sonst leeren Zeile eingeben, wie Ihr Text eingerückt werden soll. Anschließend führen Sie  $\text{Strg}+\text{X}$ , `.` aus, also  $\text{Strg}+\text{X}$ , `Punkt` (siehe [Tabelle 16.12](#)). Das Programm rückt jetzt ab der zweiten Zeile eines Absatzes alle Zeilen bis zur Einrückspalte ein.

Zum Neuformatieren größerer Textmengen, die unterschiedlich stark eingerückt sind, eignet sich das Kommando  $\text{Alt}+\text{x}$  `fill-individual-paragraphs`  $\leftarrow$ . Dieses Kommando formatiert den gesamten Bereich zwischen dem Markierungspunkt ( $\text{Strg}+\text{Leertaste}$ ) und der aktuellen Cursorposition. Dabei werden die aktuellen Einrückungen beibehalten.

| Tastenkürzel  | Funktion  |
|---|---|
| <code>Strg</code> + <code>X</code> , <code>.</code>           | definiert die Einrückspalte durch die aktuelle Cursorposition. Der Cursor muss dazu in einer leeren (!) Zeile stehen. |
| <code>Alt</code> + <code>M</code>                             | bewegt den Cursor an den Beginn einer eingerückten Zeile (ähnlich wie <code>Strg</code> + <code>A</code> ).           |
| <code>Strg</code> + Leertaste                                 | setzt den Markierungspunkt.   |
| <code>Alt</code> + <code>x</code> fill-individ <code>↵</code> | formatiert den Bereich zwischen Markierungspunkt und Cursorposition neu und behält die aktuellen Einrückungen bei.    |

Tabelle 16.12 Fließtext einrücken

Wenn Sie sehr viel mit Einrückungen arbeiten, ist der Textmodus bequemer als die oben beschriebene Vorgehensweise. Diesen Modus aktivieren Sie mit `Alt` + `X` text-mode `↵` (siehe Tabelle 16.13). Um in diesem Modus Fließtext zu bearbeiten, aktivieren Sie außerdem den dafür vorgesehenen Nebenmodus mit `Alt` + `X` auto-fill-mode `↵`. Nebenmodi definieren einige zusätzliche Kommandos, die parallel zu einem beliebigen Hauptmodus verwendet werden können (siehe auch [Abschnitt 16.9](#), »Besondere Bearbeitungsmodi«).

Die einzige wesentliche Neuerung des Textmodus besteht darin, dass der Emacs beim Zeilenumbruch jede neue Zeile automatisch so weit einrückt wie die vorhergehende Zeile. Auch `Alt` + `Q` für den manuellen Umbruch orientiert sich jetzt automatisch an der Einrückung der ersten Zeile.

| Tastenkürzel                                       | Funktion   |
|--|--|
| <code>Alt</code> + <code>X</code> text-mode        | aktiviert den Textmodus.   |
| <code>Alt</code> + <code>X</code> auto-fill-mode   | aktiviert den Nebenmodus für Fließtext.  |
| <code>Alt</code> + <code>Q</code>                  | führt einen manuellen Umbruch durch und orientiert sich dabei an der Einrückung der aktuellen Zeile. |
| <code>Alt</code> + <code>S</code>                  | zentriert die aktuelle Zeile.  |
| <code>Alt</code> + <code>⇧</code> + <code>S</code> | zentriert den aktuellen Absatz.  |

Tabelle 16.13 Textmodus

Wenn Sie Zeilen oder Absätze zentrieren möchten, ohne deswegen in den Textmodus zu wechseln, können Sie die entsprechenden Kommandos in den anderen Modi mit `Alt` + `X` center-line `↵` bzw. mit `Alt` + `X` center-paragraph `↵` aufrufen.

**Abkürzungen** Eine Besonderheit des Emacs besteht darin, dass Sie ohne Vorarbeit Abkürzungen verwenden können. Dazu geben Sie die ersten Buchstaben eines Wortes ein und drücken `[Alt]+[Z]`. Der Emacs sucht daraufhin zuerst im vorangehenden, dann im nachfolgenden Text und schließlich in allen geöffneten Dateien nach Wörtern, die mit diesen Zeichen beginnen. Wenn Sie an dieser Stelle im Text um `[Alt]+[Z]` eingeben, ersetzt der Emacs »Um« durch »Umgebung«. Wenn Sie `[Alt]+[Z]` öfter drücken, bietet der Emacs weitere mögliche Ergänzungen an, etwa »Umgang« und »Umgehen«.

Dynamische Erweiterungen funktionieren nur, wenn sich ein Wort bereits im Text einer geladenen Datei befindet (es muss nicht die aktuelle Datei sein) und wenn die Anfangsbuchstaben übereinstimmen.

## 16.7 Suchen und Ersetzen

Am schnellsten finden Sie Text mit `[Strg]+[S]` `suchtext`. Das Kommando weist gegenüber den Suchkommandos anderer Programme eine Besonderheit auf: Es beginnt die Suche sofort nach der Eingabe des ersten Zeichens. Wenn Sie also »Nebenmodus« suchen und `[Strg]+[S]` `Neb` eingeben, dann springt der Cursor bereits zum ersten Wort, das mit »Neb« beginnt. Anstatt die weiteren Buchstaben einzugeben, können Sie jetzt durch das abermalige Drücken von `[Strg]+[S]` zum nächsten Wort springen, das auch mit »Neb« beginnt. (Wenn Sie nur Kleinbuchstaben eingeben, wird nicht zwischen Groß- und Kleinschreibung unterschieden.)

Wenn Sie jetzt auf die Idee kommen, dass Sie eigentlich nach »Neuigkeit« suchen, löschen Sie das »b« mit `[←]`. Der Emacs springt zum ersten Wort zurück (ausgehend von der Position beim Beginn der Suche), das mit »Ne« beginnt. Mit der Eingabe von `[U]` springt Emacs weiter zum ersten Wort, das mit »Neu« beginnt. Probieren Sie es einfach einmal aus – Sie werden von diesem Konzept sofort begeistert sein!

Sobald Sie `[↵]` oder eine Cursortaste drücken, nimmt das Programm an, dass die Suche beendet ist, und setzt den Cursor an die gefundene Stelle. Der Beginn der Suche wird dabei durch einen Markierungspunkt gespeichert. Daher können Sie mit `[Strg]+[X]`, `[Strg]+[X]` den Cursor mühelos wieder dorthin zurückstellen, wo er zu Beginn der Suche stand. Ein abermaliges `[Strg]+[X]`, `[Strg]+[X]` führt Sie wieder an die Stelle des Suchtextes.

Durch zweimaliges Drücken von `[Strg]+[S]` können Sie die Suche wieder aufnehmen und zum nächsten Auftreten des Suchtextes springen. Wenn Sie rückwärts suchen möchten, drücken Sie einfach `[Strg]+[R]` statt `[Strg]+[S]`.



| Tastenkürzel                                    | Funktion  |
|---|---|
| <b>Strg</b> + <b>S</b>                          | inkrementelle Suche vorwärts  |
| <b>Strg</b> + <b>R</b>                          | inkrementelle Suche rückwärts   |
| <b>Alt</b> + <b>P</b>                           | wählt einen früher verwendeten Suchtext aus ( <i>Previous</i> ).                    |
| <b>Alt</b> + <b>N</b>                           | wählt einen später verwendeten Suchtext aus ( <i>Next</i> ).                        |
| <b>Strg</b> + <b>G</b>                          | Abbruch der Suche   |
| <b>Strg</b> + <b>X</b> , <b>Strg</b> + <b>X</b> | vertauscht den Markierungspunkt (Beginn der Suche) und die aktuelle Cursorposition. |
| <b>Strg</b> + <b>Alt</b> + <b>S</b>             | inkrementelle Mustersuche vorwärts  |
| <b>Strg</b> + <b>Alt</b> + <b>R</b>             | inkrementelle Mustersuche rückwärts   |
| <b>Alt</b> + <b>%</b>                           | Suchen und Ersetzen ohne Muster   |
| <b>Alt</b> + <b>X</b> query-replace-r <b>↵</b>  | Suchen und Ersetzen mit Muster  |

**Tabelle 16.14** Kommandos zum Suchen und Ersetzen

Wenn Sie zu einem späteren Zeitpunkt nach einem Text suchen möchten, den Sie früher schon einmal gesucht haben, können Sie nach **Strg** + **S** mit **Alt** + **P** (*Previous*) und **Alt** + **N** (*Next*) einen Text aus der gespeicherten Liste der Suchtexte auswählen.

### Suche nach Mustern (mit regulären Ausdrücken)

Die inkrementelle Suche findet Texte, die exakt dem Suchtext entsprechen. Häufig ist es aber wünschenswert, nach Texten zu suchen, die einem bestimmten Muster entsprechen. Eine derartige Suche starten Sie mit **Strg** + **Alt** + **S** bzw. + **R**.

Im Suchtext wird zwischen Groß- und Kleinschreibung unterschieden. Zur Syntax der Musterzeichenkette (siehe [Tabelle 16.15](#)) folgen jetzt noch einige erklärende Beispiele:

- ▶ `\<[Dd]ie\>` sucht nach dem Artikel »die«, egal ob er klein- oder großgeschrieben ist. Wortzusammensetzungen mit »die« (also etwa »dieser«) werden ignoriert.
- ▶ `[Dd]ie[a-z]+` sucht nach Wortzusammensetzungen, die mit »Die« oder »die« beginnen und denen mindestens ein weiterer Buchstabe folgt. Der Cursor bleibt jeweils am Ende des Wortes stehen (beim ersten Zeichen, das kein Buchstabe zwischen a und z ist).
- ▶ `[Dd]ie[a-zäöüß]+` funktioniert wie oben beschrieben, findet aber auch Wortzusammensetzungen, die deutsche Sonderzeichen enthalten.

Die Zeichenpaare \`(` und \`)` haben keinen Einfluss auf die eigentliche Suche. Die Zeichen im gesuchten Text, die den in der Gruppe enthaltenen Zeichen entsprechen, können dann aber zum Bilden des Ersetzen-Textes wiederverwendet werden (siehe unten).

| Suchmuster            | Funktion  |
|-----------------------|---|
| <code>\&lt;</code>    | Anfang eines Wortes   |
| <code>&amp;</code>    | Ende eines Wortes   |
| <code>^</code>        | Anfang der Zeile  |
| <code>\$</code>       | Ende der Zeile  |
| <code>.</code>        | ein beliebiges Zeichen mit Ausnahme eines Zeilenumbruchs  |
| <code>*</code>        | beliebig viele (auch 0) beliebige Zeichen (wie * in Dateinamen)   |
| <code>+</code>        | beliebig viele (aber mindestens ein) beliebige(s) Zeichen   |
| <code>?</code>        | kein oder ein beliebiges Zeichen  |
| <code>[abc..]</code>  | eines der aufgezählten Zeichen  |
| <code>[^abc..]</code> | keines der aufgezählten Zeichen   |
| <code>\(</code>       | Beginn einer Gruppe (siehe unten »Suchen und Ersetzen«)   |
| <code>\)</code>       | Ende einer Gruppe   |
| <code>\x</code>       | Sonderzeichen <i>x</i> (z. B. <code>\\</code> zur Suche nach einem <code>\</code> -Zeichen oder <code>\.</code> zur Suche nach einem Punkt) |
| <code>\&amp;</code>   | Platzhalter im Ersetzen-Muster für den gesamten gefundenen Text   |
| <code>\1</code>       | Platzhalter im Ersetzen-Muster für die erste <code>\(...\)</code> -Gruppe im Suchtext ( <code>()</code> )                                   |

Tabelle 16.15 Aufbau eines regulären Suchmusters

## Suchen und Ersetzen

Auch beim Suchen und Ersetzen unterscheidet der Emacs zwischen dem normalen Kommando und der erweiterten Version mit Mustersuche. Bei der normalen Variante mit `Alt+%` wird die Groß- und Kleinschreibung bei der Suche ignoriert. Beim Ersetzen (siehe [Tabelle 16.16](#)) bleiben die Anfangsbuchstaben von Wörtern so erhalten, wie sie bisher waren, wenn der Ersetzen-Text vollständig kleingeschrieben ist. Das Suchen- und Ersetzen-Kommando kann nicht für mehrzeilige Texte verwendet werden, weil die Joker-Zeichen `*` und `+` nicht über eine Zeile hinaus wirksam sind.

Das Suchen und Ersetzen mit Mustern starten Sie mit `Alt+X` `query-replace-r` `↵`. In der Ersetzen-Zeichenkette können Sie mit `\&` und `\n` Platzhalter angeben, die dem ganzen Suchmuster bzw. einem Teil davon entsprechen (siehe [Tabelle 16.15](#)).

Damit lassen sich sehr komplexe Operationen effizient durchführen. Zur Veranschaulichung folgt ein Beispiel:

Sie ersetzen `funktion(\([^\,]*\),\([^\,]*\))` durch `funktion(\2,\1)`: Bei jedem Aufruf von `funktion` werden die beiden Parameter vertauscht. Aus `funktion(a+b,2*e)` wird daher `funktion(2*e,a+b)`. Einzige Bedingung: In den Parametern der Funktion dürfen keine Kommata auftreten. Beim Vertauschen der Parameter in `funktion(f(a,b),g(x,y))` versagt das Kommando.

Verwenden Sie das Kommando zum Suchen und Ersetzen mit Mustern zunächst mit Vorsicht, und speichern Sie zuvor Ihren Text. Gerade bei den ersten Versuchen kommt es häufig vor, dass mit dem Suchmuster ganz andere (oft viel größere) Texte erfasst werden, als Sie geplant haben. `Strg+X`, `Strg+U` macht fehlerhafte Ersetzen-Kommandos bei Bedarf wieder rückgängig.

| Tastenkürzel                       | Funktion   |
|------------------------------------|--|
| Leertaste oder <code>Y</code>      | ersetzen, Suche fortsetzen   |
| <code>,</code>                     | ersetzen, aber Cursor stehen lassen, damit das Ergebnis kontrolliert werden kann. Wenn alles in Ordnung ist, kann das Kommando mit der Leertaste fortgesetzt werden. |
| <code>←</code> oder <code>N</code> | nicht ersetzen, Suche fortsetzen   |
| <code>Esc</code>                   | nicht ersetzen, Kommando abbrechen   |
| <code>!</code>                     | alle weiteren Ersetzungen ohne Rückfrage durchführen   |
| <code>Strg+R</code>                | Kommando vorläufig unterbrechen, um an der aktuellen Cursorposition eine manuelle Korrektur vorzunehmen (Recursive Edit)   |
| <code>Strg+Alt+R</code>            | Ersetzen-Kommando wieder aufnehmen   |

Tabelle 16.16 Tastenkürzel zur Bearbeitung des gefundenen Texts

## 16.8 Puffer und Fenster

Bei der Bearbeitung mehrerer Texte verwaltet Emacs jeden Text in einem sogenannten Puffer. Selbst wenn Sie mit nur einem Text arbeiten, existieren mehrere Puffer: einer für den Text (der Name des Puffers stimmt mit dessen Dateinamen überein), einer für ein irgendwann geöffnetes Info- oder Hilfefenster (Puffername `*info*` oder `*help*`), einer für die zuletzt angezeigte Liste mit möglichen Kommandos, die durch `☞` ergänzt wurden (`*completions*`) etc.

Neben dem Begriff des Puffers kennt der Emacs auch Fenster: Ein Fenster ist ein Bereich innerhalb des Emacs, in dem ein Puffer angezeigt wird. Normalerweise wird

nur ein einziges Fenster verwendet, das den gesamten zur Verfügung stehenden Raum nutzt. Bei der Ausführung mancher Kommandos (z. B. zur Anzeige von Hilfe- oder anderen Emacs-internen Informationen) wird der Bildschirm horizontal in zwei Fenster geteilt. Auch eine Unterteilung in mehrere horizontale oder vertikale Streifen ist möglich. Dabei kann in jedem Bereich (Fenster) ein anderer Puffer angezeigt werden.

Es besteht auch die Möglichkeit, in zwei Fenstern denselben Puffer darzustellen. Das ist vor allem bei sehr langen Texten praktisch: Sie können so zwei unterschiedliche Abschnitte des Textes bearbeiten, ohne ständig umständliche Cursorbewegungen durchführen zu müssen.

### Verschiedene »Fenster«-Arten

Der Fensterbegriff in Emacs hat nichts mit einem herkömmlichen Fenster unter Gnome oder KDE zu tun, sondern meint nur einen Teilbereich innerhalb des Emacs-Fensters. Wenn Sie tatsächlich ein zweites Emacs-Fenster benötigen, etwa um zwei Programmlistings bequem nebeneinander zu bearbeiten, führen Sie `FILE • NEW FRAME` aus.

Die Kommandos in [Tabelle 16.17](#) beziehen sich auf das gerade aktuelle Fenster (also auf das Fenster, in dem der Cursor steht). Die Kommandos wechseln den Puffer, der in diesem Fenster angezeigt wird.

| Tastenkürzel                             | Funktion   |
|--|--|
| <code>(Strg) + [X], [B], [↵]</code>      | aktiviert den zuvor verwendeten Puffer.  |
| <code>(Strg) + [X], [B], name [↵]</code> | aktiviert den angegebenen Puffer.  |
| <code>(Strg) + [X], (Strg) + [B]</code>  | zeigt in einem Fenster die Liste aller möglichen Puffer an. Dieses Fenster kann mit <code>(Strg) + [X], [I]</code> wieder gelöscht werden. |
| <code>(Strg) + [X], [K] name [↵]</code>  | löscht den angegebenen Puffer. Wenn der Puffer eine noch nicht gespeicherte Datei enthält, erscheint eine Sicherheitsabfrage.              |

**Tabelle 16.17** Pufferkommandos

Die Kommandos in [Tabelle 16.18](#) wirken sich nur auf die Anzeige der Puffer in verschiedenen Bildschirmbereichen (Fenstern) aus. Die Trennlinie zwischen den Fenstern kann mit der Maus bewegt werden. Die Puffer werden durch das Löschen eines Fensters nicht berührt. Sie werden zwar unsichtbar, bleiben aber weiterhin im Speicher und können jederzeit wieder angezeigt werden.

| Tastenkürzel                | Funktion   |
|-----------------------------|--|
| <code>Strg + X, 0</code>    | springt zum nächsten Fenster (»Oh«).                     |
| <code>Strg + X, 0</code>    | löscht das aktuelle Fenster (»Null«).                    |
| <code>Strg + X, 1</code>    | löscht alle Fenster außer dem, in dem der Cursor steht.  |
| <code>Strg + X, 2</code>    | teilt das aktuelle Fenster in zwei horizontale Bereiche. |
| <code>Strg + X, 3</code>    | teilt das aktuelle Fenster in zwei vertikale Bereiche.   |
| <code>Strg + X, &lt;</code> | verschiebt den Fensterinhalt nach links.                 |
| <code>Strg + X, &gt;</code> | verschiebt den Fensterinhalt nach rechts.                |

Tabelle 16.18 Fensterkommandos

## 16.9 Besondere Bearbeitungsmodi

Zahlreiche Bearbeitungsmodi verändern die Funktionalität des Editors und stellen zusätzliche Spezialkommandos zur Verfügung. Damit wird der Emacs optimal an einen Texttyp angepasst. Je nach Modus werden außerdem Schlüsselwörter und Kommentare farblich hervorgehoben.

Der Emacs unterscheidet zwischen Haupt- und Nebenmodi (siehe [Tabelle 16.19](#) und [Tabelle 16.20](#)). Es kann immer nur ein Hauptmodus aktiv sein. Dieser Modus wird automatisch entsprechend der Kennung des Dateinamens und nach Schlüsselwörtern im Text gewählt. Der Hauptmodus kann durch Nebenmodi ergänzt werden. Für jede im Emacs bearbeitete Datei (für jeden Puffer) gilt eine eigene Moduseinstellung. Die manuelle Veränderung des Modus wirkt sich immer nur auf den gerade aktuellen Puffer aus. Durch den Wechsel in einen anderen Hauptmodus wird der bisherige Modus deaktiviert. Das Ein- oder Ausschalten eines Nebenmodus verändert den Hauptmodus nicht.

Eine Übersicht über alle verfügbaren Modi gibt `F1`, `A` mode `↩`. Informationen zum gerade aktiven Hauptmodus erhalten Sie mit `F1`, `M`.

Das vielleicht attraktivste Merkmal der Bearbeitungsmodi ist das sogenannte Syntax-Highlighting. Dabei werden Kommandos, Kommentare etc. durch Farben oder Schriftattribute gekennzeichnet. Programmcode,  $\LaTeX$ -Dokumente etc. gewinnen dadurch erheblich an Übersichtlichkeit.

Syntax-  
hervorhebung

Unbegreiflicherweise enthält der Emacs standardmäßig keinen PHP-Modus. Auch ein Modus zur Bearbeitung von Markdown-Dokumenten fehlt. Wie Sie diese und andere Emacs-Erweiterungen unkompliziert nachinstallieren können, erfahren Sie in [Abschnitt 16.11](#), »MELPA«.

PHP- und  
Markdown-  
Modus

| Tastenkürzel                                       | Funktion  |
|--|---|
| <code>Alt+X</code> <code>fundamental-mode</code> ↵ | Standardmodus (Grundeinstellung)                |
| <code>Alt+X</code> <code>text-mode</code> ↵        | Modus zur bequemen Einrückung von Text          |
| <code>Alt+X</code> <code>c-mode</code> ↵           | C-Modus   |
| <code>Alt+X</code> <code>c++-mode</code> ↵         | C++-Modus                                       |
| <code>Alt+X</code> <code>emacs-lisp-mode</code> ↵  | Emacs-Lisp-Dateien bearbeiten (z. B. ~/ .emacs) |
| <code>Alt+X</code> <code>html-mode</code> ↵        | HTML-Modus                                      |
| <code>Alt+X</code> <code>java-mode</code> ↵        | Java-Modus                                      |
| <code>Alt+X</code> <code>latex-mode</code> ↵       | LaTeX-Modus                                     |
| <code>Alt+X</code> <code>sh-mode</code> ↵          | Modus zur Bearbeitung von Shell-Scripts         |

Tabelle 16.19 Wichtige Emacs-Hauptmodi

| Tastenkürzel                                       | Funktion  |
|--|---|
| <code>Alt+X</code> <code>auto-fill-mode</code> ↵   | Fließtextmodus (automatischer Wortumbruch)                                      |
| <code>Alt+X</code> <code>cua-mode</code> ↵         | CUA-Modus ( <code>Strg+C</code> , <code>Strg+X</code> und <code>Strg+V</code> ) |
| <code>Alt+X</code> <code>font-lock-mode</code> ↵   | farbige Syntaxmarkierung  |
| <code>Alt+X</code> <code>iso-accents-mode</code> ↵ | Eingabe fremdsprachiger Sonderzeichen   |
| <code>Alt+X</code> <code>abbrev-mode</code> ↵      | Abkürzungsmodus (automatische Auflösung von Abkürzungen)                        |

Tabelle 16.20 Wichtige Emacs-Nebenmodi

### Automatische und explizite Moduseinstellung

Der Emacs versucht beim Laden einer Datei aus der Dateikennung und dem Inhalt der ersten Zeilen automatisch zu erkennen, um welchen Dateityp es sich handelt, und aktiviert dann den entsprechenden Modus. Nur wenn das nicht klappt, müssen Sie den Modus wie oben beschrieben manuell aktivieren. Wenn die automatische Aktivierung nicht funktioniert, können Sie auch in der ersten Zeile der Datei einen Kommentar einfügen, der die Zeichen  `-*- name -*-` enthält. Statt `name` müssen Sie den Namen des gewünschten Modus angeben (also etwa  `-*- html -*-`).

## 16.10 Konfiguration

Wohl kein anderer Editor bietet mehr Konfigurationsmöglichkeiten als der Emacs. Dieser Abschnitt gibt zuerst einen Überblick über die Konfigurationsdateien und beschreibt dann einige elementare Konfigurationsschritte.

Wenn sich Ihre Emacs-Version anders verhält, als in diesem Buch beschrieben wird, dann ist oft die Konfiguration Ihrer Linux-Distribution verantwortlich. Die Einstellungen können sich sowohl in den persönlichen als auch in den globalen Konfigurationsdateien (`site-start.el`) befinden. Beachten Sie insbesondere, dass bei vielen Distributionen beim Anlegen neuer Linux-Benutzer automatisch Konfigurationsdateien aus `/etc/skel` in das Benutzerverzeichnis kopiert werden!

Die benutzerspezifische Konfiguration kann wahlweise durch Menükommandos (OPTIONS-Menü) oder durch eine direkte Veränderung der Konfigurationsdateien `.emacs` durchgeführt werden.

Persönliche  
Konfiguration

Neben den persönlichen Konfigurationsdateien gibt es auch globale Konfigurationsdateien, deren Einstellungen für alle Benutzer gelten. Diese Dateien enthalten je nach Distribution diverse Voreinstellungen.

Globale  
Konfiguration

|   |                   |
|---|-------------------|
| <code>/usr/share/emacs/site-lisp/site-start.el</code>     |                   |
| <code>/usr/share/emacs/site-lisp/debian-startup.el</code> | (Debian, Ubuntu)  |
| <code>/usr/share/emacs/site-lisp/site-start.d/*</code>    | (Red Hat, Fedora) |
| <code>/usr/share/emacs/site-lisp/site-start.el</code>     | (SUSE)            |

Beim Emacs können Sie einige elementare Einstellungen direkt über Einträge des OPTIONS-Menüs vornehmen. Damit beispielsweise der Schiebepfeil wie in allen anderen Programmen auf der rechten Seite angezeigt wird, führen Sie `OPTIONS • SHOW/HIDE • SCROLLBAR • ON THE RIGHT` aus. Die Einstellungen werden sofort wirksam, gehen aber verloren, wenn Sie den Emacs verlassen. Um geänderte Einstellungen bleibend in `.emacs` zu speichern, müssen Sie `OPTIONS • SAVE OPTIONS` ausführen!

Konfiguration  
per Mausclick

Alle weitergehenden Einstellmöglichkeiten, von denen es Tausende gibt, sind über `OPTIONS • CUSTOMIZE EMACS • TOP-LEVEL CUSTOMIZATION GROUP` erreichbar (siehe [Abbildung 16.2](#)). Dieses Kommando öffnet einen neuen Emacs-Puffer, der wie ein Dialog aussieht. Die Buttons dieser Seite führen zu weiteren Dialogen für verschiedene Gruppen von Optionen. Auf jeder Seite können Sie durch Buttons alle durchgeführten Änderungen nur bis zum Programmende oder bleibend in `.emacs` speichern (`SET FOR CURRENT SESSION` bzw. `SAVE FOR FUTURE SESSIONS`).

Die Bedienung der Dialogoptionen ist nicht schwierig. Das Problem besteht aber darin, dass die Dialoge tief verschachtelt sind und es nicht immer ganz einfach ist, den richtigen Dialog für eine bestimmte Option zu finden.

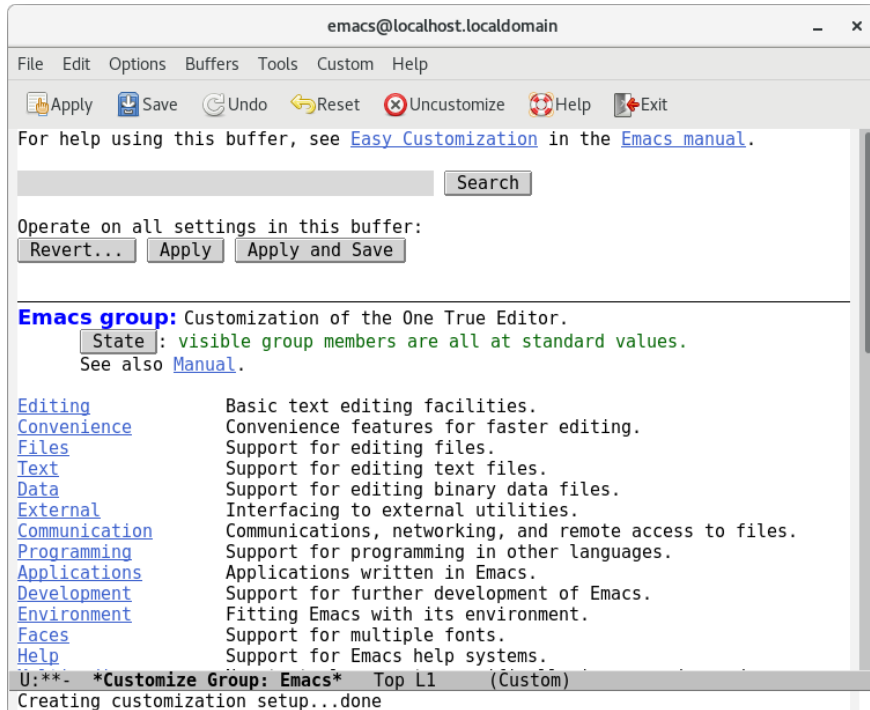



Abbildung 16.2 Emacs-Konfiguration

Schriftart und  
-größe einstellen

Am schnellsten kann die Schriftart hier mit  und der linken Maustaste eingestellt werden. Allerdings besteht keine Möglichkeit, diese Einstellung auch zu speichern. Alternativ können Sie die Standardschrift auch mit `OPTIONS • CUSTOMIZE • SPECIFIC FACE` einstellen, wobei Sie die Schrift `default` auswählen. Im nun erscheinenden Dialog können Sie sowohl die Schriftfamilie als auch die Schriftgröße (Height) einstellen. Falls dem Namen der Schrift deren Hersteller vorangestellt wird, muss dies mit einem Bindestrich erfolgen (z. B. Adobe-Courier). An dieser Stelle können Sie auch die Hintergrundfarbe des Emacs einstellen: Geben Sie einfach im `BACKGROUND`-Feld der Schriftart `default` den Namen der gewünschten Farbe an (z. B. Lightgrey).

Mit `SET FOR CURRENT SESSION` können Sie Ihre Einstellungen ausprobieren. Der Schriftwechsel dauert einige Sekunden und führt dazu, dass auch die Fenstergröße von Emacs verändert wird. Mit `SAVE FOR FUTURE SESSIONS` wird die Einstellung in `.emacs` gespeichert.

Beispiel für  
.emacs

Anstatt sich durch verschachtelte Dialoge zu klicken, können Sie die Konfiguration auch direkt in `.emacs` durchführen. Der Platz reicht hier nicht für eine ausführliche Beschreibung aus. Stattdessen gibt das folgende kommentierte Listing einige Beispiele für beliebte Optionen und Einstellungen. Bei manchen Linux-Distributionen gelten einige dieser Einstellungen standardmäßig.



```

; Beispiel für .emacs
; kein Begrüßungsbildschirm
(setq inhibit-startup-message t)

; Zeilen- und Spaltennummer in der Statuszeile anzeigen
(line-number-mode 1)
(column-number-mode 1)

; markierten Textbereich sichtbar machen
(setq-default transient-mark-mode t)

; mit <Tab> Leerzeichen statt Tabulatoren einfügen
(setq-default indent-tabs-mode nil)

; letzte Zeile automatisch mit Newline-Code abschließen
(setq require-final-newline t)

; Syntaxhervorhebung automatisch aktivieren
(global-font-lock-mode t)

; Cursorposition speichern
(require 'saveplace)
(setq-default save-place t)

; AUC TEX aktivieren (das ist ein erweiterter LaTeX-Modus;
; das auctex-Paket muss separat installiert werden; der Paket-
; name lautet bei vielen Distributionen emacs-auctex
(require 'tex-site)

; eigene Funktion; vertauscht den Buchstaben
(defun swap-char()          ;zwei Buchstaben an der Cursor-Position
  (interactive)           ;vertauschen:
  (save-excursion
    (forward-char)
    (transpose-chars 1)))

; ein paar Tastaturkürzel:
(global-set-key [f2] 'switch-to-buffer) ;Buffer wechseln
(global-set-key [f3] 'goto-line)      ;zur Zeile n springen
(global-set-key [f4] 'advertised-undo) ;Undo-Funktion
(global-set-key [f5] 'swap-char)      ;Buchstaben vertauschen

```

Noch viel mehr Konfigurationsmöglichkeiten haben Sie, wenn Sie sich auf die Emacs-Lisp-Programmierung einlassen. Damit können Sie in der Konfigurationsdatei `.emacs` eigene Kommandos, Tastenkürzel etc. definieren. Schauen Sie sich dazu <http://dotemacs.de> an.

**Emacs-Lisp-  
Programmierung**

## 16.11 MELPA

MELPA (<https://melpa.org>) steht für »Milkypostman's Emacs Lisp Package Archive« und ist das vermutlich am besten gewartete Archiv von Emacs-Erweiterungen. Im Frühjahr 2017 befanden sich in diesem Archiv über 3500 verschiedene Erweiterungen. Um das Archiv zu aktivieren, fügen Sie in die Datei `.emacs` die folgenden Zeilen ein, die Sie zur Vermeidung von Tippfehlern am besten von der MELPA-Webseite kopieren:

```
(require 'package)
(add-to-list 'package-archives
  ("melpa" . "http://melpa.org/packages/"))
(when (< emacs-major-version 24)
  ;; For important compatibility libraries like cl-lib
  (add-to-list 'package-archives '("gnu" . "http://elpa.gnu.org/packages/")))
(package-initialize)
```

Nach einem Neustart des Emacs können Sie das Erweiterungsarchiv verwenden. `Alt+X` list-packages führt nun in die schier endlose Liste aller Erweiterungen (siehe [Abbildung 16.3](#)). Wie üblich können Sie den Text mit `Strg+S` rasch durchsuchen. Ein Mausklick auf ein Paket zeigt Detailinformationen und einen INSTALL-Button an. Zu meinen Favoriten zählen die Erweiterungen `php-mode` und `markdown-mode`. Sie erleichtern die Bearbeitung von PHP- und Markdown-Dateien erheblich.

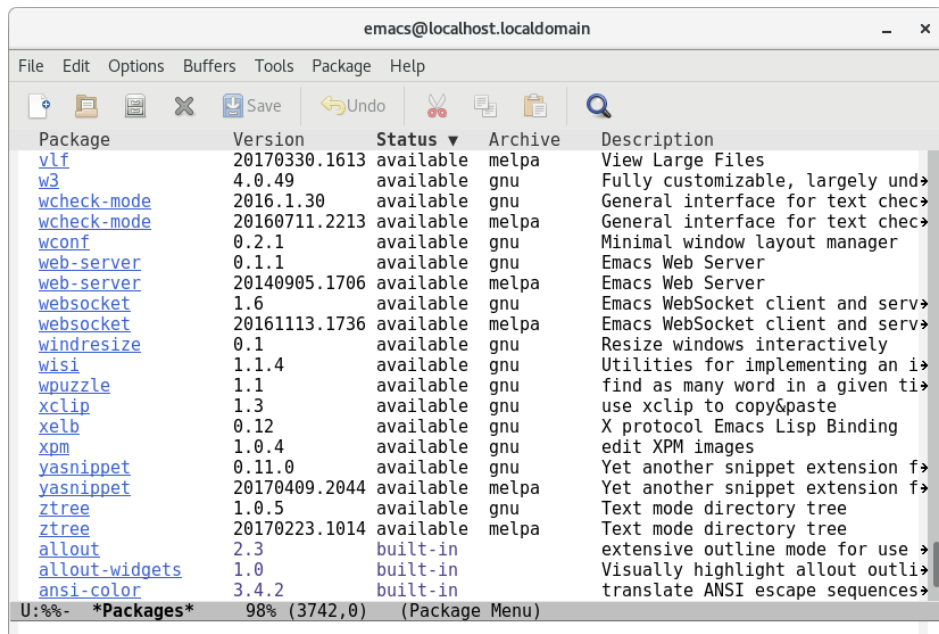


Abbildung 16.3 MELPA ist ein Archiv unzähliger Emacs-Erweiterungen.

## 16.12 Unicode

Emacs kommt dank der Mule-Erweiterung (*Multilingual Enhancement*) mit den meisten gängigen Zeichensätzen zurecht. Mule-Kommandos können über das Menü `OPTIONS • MULE` ausgeführt werden. Mit Version 23 wurde die interne Textverwaltung vollständig auf Unicode umgestellt (*Native Unicode Support*); ältere Emacs-Versionen verwenden ein eigenes Textformat. Für den Anwender ergeben sich durch diese Umstellung aber keine Änderungen.

Der Emacs erkennt den Zeichensatz von Dateien in vielen Fällen selbstständig bzw. verwendet einfach den Standardzeichensatz Ihrer Distribution. In manchen Fällen ist es aber notwendig, den Zeichensatz explizit zu bestimmen. Dazu geben Sie mit dem Kommando `(Strg)+[X]`, `(←)`, `[C]` Codierung an, welcher Zeichensatz bei der Ausführung des nächsten Kommandos gelten soll (siehe [Tabelle 16.21](#)). Die zur Auswahl stehenden Codierungen ermitteln Sie dabei bequem mit `(⇐)`.

Zeichensatz  
explizit angeben

| Kurzbezeichnung | Bedeutung                                   |
|-----------------|---|
| iso-8859-n      | ISO-8859- <i>n</i> -Dateien                 |
| iso-latin-n     | ISO-Latin- <i>n</i> -Dateien                |
| utf-8           | UTF-8-Dateien                               |
| utf-8-dos       | UTF-8-Dateien mit DOS/Windows-Zeilenkennung |
| utf-8-unix      | UTF-8-Dateien mit Unix/Linux-Zeilenkennung  |
| binary          | Binärdatei                                  |

**Tabelle 16.21** Häufig eingesetzte Codierungen

Um die Codierung eines bereits geladenen Buffers zu verändern, führen Sie `(Strg)+[X]`, `(←)`, `[F]` Codierung aus. Welcher Zeichensatz gerade benutzt wird, geht aus den ersten Zeichen der Statuszeile hervor. -U bedeutet beispielsweise, dass ein Unicode-Text vorliegt.

`(Strg)+[H]`, `(⇐)+[C]`, `(←)` beschreibt die Codierung des aktuellen Buffers. `(Strg)+[U]`, `(Strg)+[X]`, `[=]` beschreibt den Code des Zeichens unter dem Cursor.

Hilfsfunktionen

### Eingabe fremdsprachiger Sonderzeichen

Mit den oben beschriebenen Kommandos bzw. Funktionen sollte es Ihnen gelingen, Unicode-Dateien korrekt zu laden, darzustellen und wieder zu speichern. Meist wollen Sie derartige Texte aber auch selbst ändern. Aber was tun Sie, wenn das gewünschte Zeichen nicht auf der Tastatur zu finden ist?

**Unicode-Zeichen** Jedes Unicode-Zeichen kann durch seinen hexadezimalen Code eingegeben werden. Dazu führen Sie `[Alt]+[X]` `ucs-insert n` oder kürzer `[Alt]+[X]` `[8]` `[↵]` `hexcode` aus. Das Euro-Zeichen geben Sie beispielsweise mit `[Alt]+[X]` `[8]` `[↵]` `20ac` ein. Das Kommando `ucs-insert` akzeptiert auch die Namen von Unicode-Zeichen. Daher fügt auch `[Alt]+[X]` `[8]` `[↵]` `euro sign` das Euro-Zeichen ein. Weitere Tipps zur Eingabe von Unicode-Zeichen finden Sie hier:

*[http://ergoemacs.org/emacs/emacs\\_n\\_unicode.html](http://ergoemacs.org/emacs/emacs_n_unicode.html)*

**Latin-9-Prefix-Modus** Mit `[Alt]+[X]` `set-input-method` `[↵]` `latin-9-prefix` aktivieren Sie einen speziellen Modus. Er hilft bei der Eingabe von Zeichen aus dem Latin-Zeichensatz, die durch Akzente, Striche, Kreise oder anders modifiziert sind. Beispiele sind etwa à, á, â, ã, å, ä, ø oder ç. Der Modus ist auch dann praktisch, wenn Sie die Zeichen äöüß auf einer Tastatur mit US-Layout eingeben möchten.

Die Tasten `[~]`, `[_]`, `[^]`, `[/]`, `[']` und `[`]` haben jetzt eine neue Bedeutung: Wird direkt anschließend ein passender Buchstabe eingegeben, verbindet der Emacs die beiden Zeichen zu einem neuen Buchstaben. Die Eingabe `[~]`, `[O]` liefert also den Buchstaben Ö, `[_]`, `[s]` ein ß, `[^]`, `[c]` ein ç, `[/]`, `[a]` ein å, `[/]`, `[e]` ein æ, `[/]`, `[o]` ein ø. Um ein " einzugeben, müssen Sie nun allerdings `[~]` und danach die Leertaste oder `[Strg]+[Q]`, `[~]` tippen.

**Andere Eingabemodi** Mit `[Alt]+[X]` `set-input-method` `[↵]`, `[⇧]` können Sie zwischen rund 50 weiteren Modi auswählen. Diese Modi helfen z.B. bei der Eingabe chinesischer, japanischer und koreanischer Zeichen. Der gewählte Eingabemodus kann durch die Tastenkombination `[Strg]+[V]` jederzeit deaktiviert bzw. anschließend wieder aktiviert werden.

# Kapitel 17

## Atom und VSCode

Wenn Sie die beiden vorigen Kapitel gelesen haben, glauben Sie vielleicht, Linux wäre in der Steinzeit stehen geblieben. Aber keine Sorge, unter Linux laufen natürlich auch unzählige Editoren mit einer modernen Oberfläche. Zwei davon möchte ich Ihnen in diesem Kapitel näher vorstellen: Atom und Visual Studio Code, im Folgenden kurz VSCode.

Die zwei Editoren haben etliche Gemeinsamkeiten: Sie basieren auf Electron, einem Framework, das sich aus Teilen des Webbrowsers Chromium und der JavaScript-Bibliothek Node.js zusammensetzt. Beide Editoren laufen auf allen gängigen Plattformen, also unter Linux, Windows und macOS. Beide Editoren sind noch recht jung: Die ersten stabilen Versionen von Atom bzw. VSCode wurden im Februar 2016 bzw. im April 2016 veröffentlicht. Beide Editoren lassen sich unkompliziert durch Plugins erweitern. Und bei beiden Editoren steht der Quellcode unter einer Open-Source-Lizenz zur Verfügung.

Unterschiede gibt es in der Herkunft:

- ▶ Das Atom-Projekt wurde von der Firma GitHub initiiert – quasi als Git-kompatibler Editor für Entwickler. Atom kommt mit nahezu jeder Art von Code- und Textformaten zurecht, ganz egal, ob die Dateien unter der Versionskontrolle von Git stehen oder nicht.
- ▶ VSCode wurde hingegen maßgeblich von einem Team rund um den Microsoft-Mitarbeiter Erich Gamma (Co-Autor des Buchs *Entwurfsmuster*) entwickelt. Lassen Sie sich übrigens nicht vom Namen *Visual Studio Code* täuschen: Der Editor hat weder funktionell noch in seiner Codebasis etwas mit den Visual-Studio-Entwicklungsumgebungen von Microsoft zu tun. Und ja, obwohl VSCode ein Microsoft-Projekt ist, untersteht es einer Open-Source-Lizenz!

Welcher Editor ist nun besser, werden Sie vielleicht fragen. Eine klare Antwort muss ich Ihnen schuldig bleiben. Beide Editoren haben ihren Charme, mit beiden machen Sie nichts verkehrt. Persönlich zieht es mich eher zu Atom, was aber vielleicht nur daran liegt, dass ich dieses Programm zuerst ausprobiert habe. Wenn Sie eine Entscheidungshilfe brauchen, suchen Sie im Internet nach *atom vs vscode*! Sie werden unzählige Vergleichstests finden.

Natürlich habe ich den ersten Teil dieses Kapitels mit Atom, den zweiten dann mit VSCode verfasst. Dabei habe ich jeweils die  $\LaTeX$ -Erweiterung für den jeweiligen Editor aktiviert. Grundsätzlich war das Schreiben in beiden Editoren problemlos. Aber ich kam nie in Versuchung, von meinem seit 25 Jahren geliebten Editor Emacs Abschied zu nehmen :-)

**Alternativen** Das Editor-Angebot hat sich in den vergangenen Jahren erfreulich vergrößert. Ein interessanter Editor mit starkem Fokus auf Webentwickler ist *Adobe Brackets*.

Unter professionellen Entwicklern sehr beliebt ist das kommerzielle Programm *Sublime Text*. Im Gegensatz zu den anderen genannten Programmen basiert er nicht auf Web-Frameworks, sondern wurde in C++ entwickelt und hat damit einen spürbaren Geschwindigkeitsvorteil. Es mangelt nicht an cleveren Funktionen, für die Sie aber aktuell ca. EUR 70 investieren müssen. Immerhin erwerben Sie mit der Lizenz das Recht, den Editor auf allen Ihren Rechnern auszuführen, was dem Linux-Entwickleralltag mit vielen Parallel- und Neuinstallationen entgegenkommt. Sie können Sublime Text unbegrenzt kostenlos testen, aber das Programm wird Sie immer wieder daran erinnern, dass eine Lizenz fällig ist.

#### Vi und Emacs bleiben wichtig!

Weder die elegante Oberfläche noch die einfache Erweiterbarkeit von Atom, VSCode und Co. machen Vi und Emacs obsolet. Der entscheidende Vorteil dieser Unix-Urgesteine besteht darin, dass Vi und Emacs auch im Textmodus, in Minimalinstallationen und via SSH laufen. Außerdem gehen Vi und selbst Emacs vergleichsweise sparsam mit den Ressourcen um, was man von Atom & Co. leider wirklich nicht behaupten kann.

## 17.1 Atom

### Installation und Updates

Die meisten Distributionen bieten aktuell noch keine Atom-Pakete an. Die Installation ist dennoch unkompliziert: Auf der Webseite <https://atom.io> finden Sie RPM- und Debian-Pakete zum Download. Bei den meisten Distributionen wird direkt aus dem Webbrowser heraus ein Paketmanager zur Installation gestartet. Funktioniert das nicht, verwenden Sie `rpm -i paketname` bzw. `dpkg -i paketname`.

Die Installation ohne Paketquelle hat den offensichtlichen Nachteil, dass Sie sich selbst um Updates kümmern müssen – und die gibt es reichlich: in den vergangenen Jahren alle zwei bis drei Monate. Für ein Update beenden Sie Atom, laden das gerade aktuelle RPM- oder Debian-Paket von <https://atom.io> herunter und wiederholen die Installation. Keine Angst, Ihre Konfiguration bleibt dabei erhalten. Einstellungen, Tastenkürzel etc. werden im Verzeichnis `.item` gespeichert.

Atom verfügt über zwei eingebaute Update-Funktionen: Zum einen können Sie mit EDIT • PREFERENCES • UPDATES alle installierten Atom-Erweiterungspakete aktualisieren. Zum anderen gibt es in den Einstellungen im Dialogblatt CORE die Option AUTOMATICALLY UPDATE: Unter Windows und macOS sorgt diese Option dafür, dass Atom selbst automatisch aktualisiert wird. Unter Linux funktioniert diese Option aber nicht.

### Inoffizielle Atom-Paketquellen

Es gibt zwar keine offiziellen Atom-Paketquellen, aber einige Atom-Fans haben sich die Mühe gemacht, derartige Repositories selbst einzurichten. Wie immer bei derartigen Initiativen gibt es keine Garantien, wie lange die Paketquellen gewartet werden. Unter Ubuntu kann Atom auch als Snap-Paket installiert werden (`sudo snap install atom --classic`).

<https://launchpad.net/~webupd8team/+archive/ubuntu/atom>

<https://copr.fedorainfracloud.org/coprs/mosquito/atom>

## Erste Schritte

Das Atom-Fenster ist beim ersten Start in zwei Teile (*Panes*) geteilt: Links heißt Atom Sie im 21. Jahrhundert willkommen, rechts enthält der WELCOME GUIDE einige Links zu wichtigen Funktionen (siehe [Abbildung 17.1](#)). Wenn Sie beide Texte schließen, bleibt das Dokument *untitled* übrig, in dem Sie Atom ausprobieren können. Den WELCOME GUIDE können Sie bei Bedarf über das HELP-Menü öffnen.

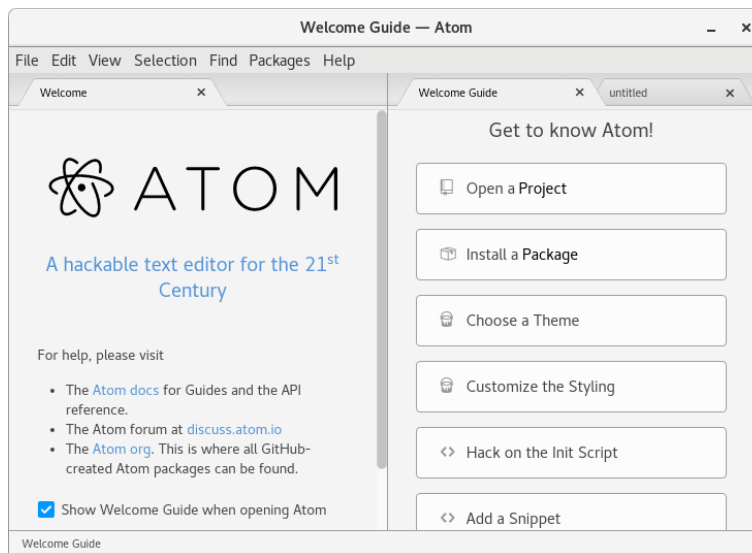


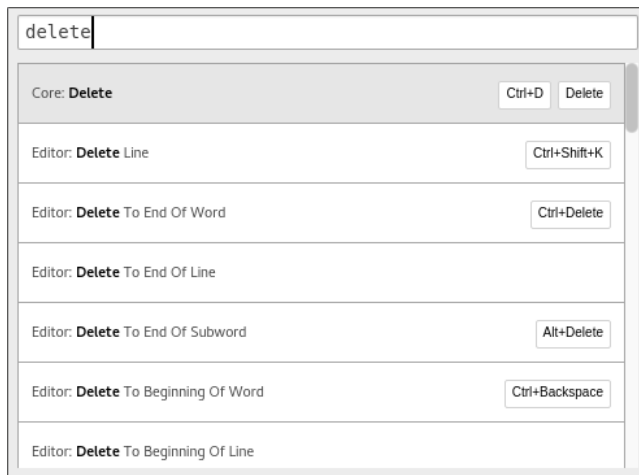
Abbildung 17.1 Die Benutzeroberfläche von Atom

**Projekte** Wie in jedem anderen Editor können Sie im FILE-Menü einzelne Dateien zur Bearbeitung öffnen. Wenn Sie mehrere Dateien bearbeiten möchten, die sich in einem Verzeichnis befinden, ist FILE • OPEN FOLDER die bessere Wahl: Damit wird links der Verzeichnisbaum eingeblendet. Sie können dann per Mausklick zwischen den Dateien des Projekts wechseln. Wenn Ihnen der Verzeichnisbaum zu viel Platz wegnimmt, können Sie ihn mit VIEW • TOGGLE TREE VIEW ein- und ausblenden.

Aus Atom-Sicht gilt jedes einmal geöffnete Verzeichnis als Projekt. Mit FILE • REOPEN PROJECT wechseln Sie zwischen in der Vergangenheit bearbeiteten Projekten, wobei Atom für jedes Projekt ein neues Fenster öffnet.

**Menü und Command View**

Die wichtigsten Atom-Kommandos finden Sie wie üblich im Menü. Darüber hinaus gibt es aber unzählige weitere Kommandos, die aus Gründen der Übersichtlichkeit im Menü fehlen. Eine Referenz aller Kommandos gibt die COMMAND VIEW. Das ist ein Auswahldialog, den Sie mit `[Strg]+[⇧]+[P]` öffnen (siehe [Abbildung 17.2](#)). Durch die Eingabe einiger Zeichen können Sie die dort angezeigte Kommandoliste filtern. Mit *delete* als Filterbegriff finden Sie also diverse Kommandos, um Text zu löschen. Soweit die Kommandos mit Tastenkürzeln verbunden sind, werden diese in der Kommandoreferenz ebenfalls angezeigt.



**Abbildung 17.2** Auf der Suche nach dem richtigen Kommando

**Tabs und Panes** Mehrere zugleich geöffnete Dateien werden als Tabs dargestellt, zwischen denen Sie mit `[Strg]+[Tab]` wechseln können. Atom gliedert das Fenster außerdem in »Panee«, also verschiebbare Fensterbereiche, die wiederum Tabs aufnehmen können. Die Administration der Panes erfolgt wahlweise über Menükommandos (VIEW • PANES), über Tastenkürzel, die zumeist mit `[Strg]+[K]` beginnen, oder über weitere Kommandos, die Sie mit `[Strg]+[⇧]+[P]` und dem Suchbegriff *panes* in der Kommandoreferenz finden.



## Grundeinstellungen

Nach der Installation zeigt sich Atom von seiner dunklen Seite, d. h., Text wird in weißer/heller Schrift auf schwarzem Hintergrund angezeigt. Das mag modern sein, ist tagsüber in einem hellen Büro aber nicht ergonomisch. Abhilfe: Führen Sie `EDIT • PREFERENCES` aus, wechseln Sie in das Dialogblatt `THEMES`, und stellen Sie dort eines der `Light Themes` für die Oberfläche und für das `Syntax-Highlighting` ein. Alle Bildschirmabbildungen in diesem Kapitel wurden der besseren Lesbarkeit wegen im `Light Theme` erstellt.

Light oder Dark Theme

Die `Line-Wrapping-Einstellungen` steuern, wie Atom mit zu langen Zeilen umgeht. Standardmäßig werden diese am Fensterrand einfach abgeschnitten; Sie müssen also den Fensterinhalt nach links schieben, wenn Sie das Ende einer langen Zeile sehen möchten. Mit der Option `SOFT WRAP` im Dialogblatt `EDIT • PREFERENCES • EDITOR` erreichen Sie, dass derartige Zeilen am Bildschirm (nicht aber in der resultierenden Datei) umbrochen werden. Die Einstellung `SOFT WRAP HANGING INDENT` gibt an, wie weit der umbrochene Text eingerückt werden soll (relativ zur Einrückung des Zeilenbeginns).

Line Wrapping

Wenn Sie mit Atom Fließtext und nicht Programmcode verfassen, dann ist es oft wünschenswert, dass Atom den Text wirklich über mehrere Zeilen verteilt (»Hard Wrap«). Das gelingt manuell mit `EDIT • REFLOW SELECTION` bzw. `[Strg] + [Q] + [Q]`. Wenn Sie diese Funktion oft brauchen, sollten Sie ihr ein anderes Tastenkürzel zuweisen; andernfalls ist die Gefahr groß, dass Sie Atom versehentlich beenden.

Ich habe keine Möglichkeit gefunden, Atom so einzustellen, dass er lange Zeilen bereits bei der Eingabe umbricht (*Hard Wrap*).

Die `Autocomplete-Funktion` schlägt nach der Eingabe einiger Buchstaben passende Vervollständigungen vor, wobei Wörter aus allen geladenen Dateien berücksichtigt werden. Standardmäßig funktioniert das nur für ausgewählte Programmiersprachen bzw. Textformate auf Anhieb. Wenn keine Vervollständigungen angezeigt werden, schafft zumeist die Installation eines passenden `Autocomplete-Pakets` für Ihre Programmiersprache Abhilfe. Allerdings gibt es nicht für alle Sprachen geeignete Pakete. Die größten Probleme hatte ich erstaunlicherweise mit simplen Text- sowie mit `LATEX`-Dateien, in denen die `Autocomplete-Funktion` komplett versagte.

Autocomplete

Im Dialogblatt `EDIT • PREFERENCES • EDITOR` können Sie im Listenfeld `TABTYPE` einstellen, ob `[Tab]` einen `Tabulatorcode (HARD)` oder `Leerzeichen (SOFT)` in Ihre Datei einfügen soll. Die `Defaulteinstellung` lautet `AUTO`: Je nachdem, ob bei der ersten eingerückten Zeile ein `Leerzeichen` oder ein `Tabulatorzeichen` verwendet wird, passt sich Atom entsprechend für den Rest der Datei an.

Tabulatoren

## Paketverwaltung

In Atom ist der Großteil aller Funktionen in Form von Paketen realisiert. Standardmäßig sind in Atom bereits rund 80 sogenannte Core-Pakete aktiv. Diese Pakete kümmern sich z. B. um das Syntaxhighlighting in diversen Sprachen, um die automatische Erstellung von Backups aller offenen Dateien oder um die vorhin beschriebene Autocomplete-Funktion.

Einen Überblick über alle installierten Pakete gibt EDIT • PREFERENCES • PACKAGES. In diesem Dialog können Sie Pakete auch konfigurieren, vorübergehend deaktivieren oder ganz entfernen (siehe [Abbildung 17.3](#)). Zur Suche nach neuen Paketen sowie zu deren Installation wechseln Sie in das Dialogblatt INSTALL. Dort haben Sie aktuell die Wahl zwischen mehr als 5000 Atom-Erweiterungen! Im Dialogblatt UPDATES können Sie die installierten Pakete auf den neuesten Stand bringen.

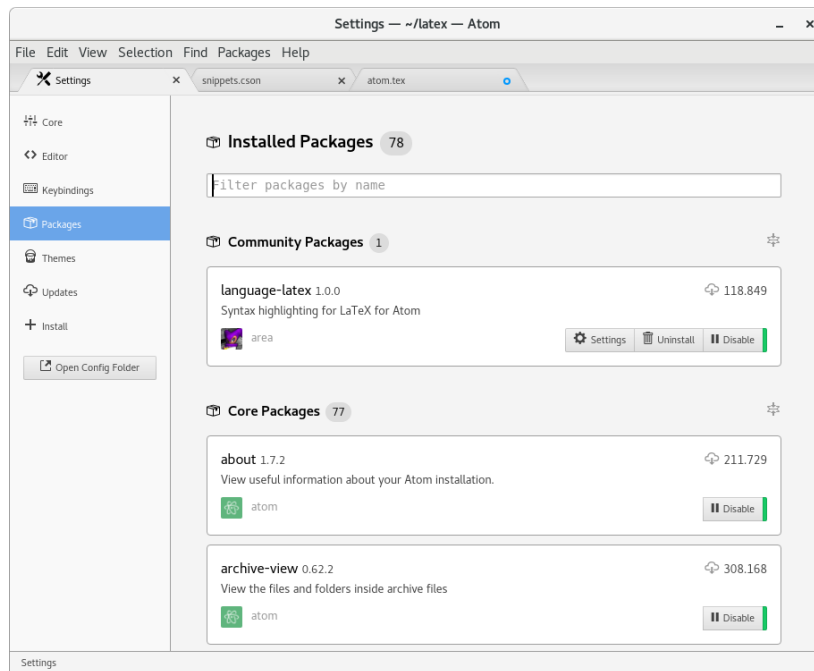


Abbildung 17.3 Atom verfügt über eine eigene Paketverwaltung.

Manche Pakete erweitern das Atom-Menü. Paketspezifische Kommandos finden Sie unter PACKAGES • PAKETNAME.

**apm-Kommando** Interessanterweise können Sie die Paketverwaltung auch außerhalb von Atom mit dem Kommando `apm` durchführen. `apm list` zählt alle installierten Pakete auf, `apm install name` installiert das gewünschte Paket etc. Das funktioniert auch, wenn Atom aktuell nicht läuft. `atom help` gibt eine kurze Beschreibung des Kommandos, und `atom help subcmd` liefert ausführlichere Informationen zu einem Subkommando.

## Tastatur

Es ist in Atom relativ einfach, eigene Tastenkürzel zu definieren. Dazu öffnen Sie mit EDIT • KEYMAP die Datei `.atom/keymap.cson` und fügen dort Anweisungen entsprechend dem folgenden Muster ein: keymap.cson

```
'atom-text-editor':
  'ctrl-a': 'editor:move-to-beginning-of-line'
  'ctrl-e': 'editor:move-to-end-of-screen-line'
```

Die in `keymap.cson` erforderliche Syntax ist in den ersten Zeilen der Datei dokumentiert. Durchgeführte Änderungen werden wirksam, sobald Sie die Datei speichern. Die Einstellungen haben Vorrang gegenüber den Tastenkürzeln, die in Paketen definiert sind. Atom passt übrigens die Menüeinträge an die gerade geltenden Tastenkürzel an!

Bei der Suche nach den Namen der auszuführenden Atom-Kommandos hilft einerseits das KEYBINDINGS-Dialogblatt der SETTINGS und andererseits die Kommandopalette, die Sie mit `Strg+⌘+P` öffnen.

### Emacs-Tastenkürzel

Für Emacs-Fans gibt es gleich mehrere Pakete, die Emacs-Tastenkürzel unter Atom nachbilden. Am populärsten ist `atomic-emacs`, weniger umfassend (im positiven Sinne) ist `emacs-core-keys`.

Mit EDIT • SNIPPETS gelangen Sie in die Datei `.atom/snippets.cson`. In dieser Datei können Sie Textbausteine definieren. Die folgenden Zeilen geben zwei Beispiele für derartige Bausteine: snippets.cson

```
'.text':
  'mfg':
    'prefix': 'mfg'
    'body': 'Mit freundlichen Grüßen,\n\n          Michael Kofler'

'.text.tex.latex':
  'latex vl':
    'prefix': 'vl'
    'body': '\\\\begin{verbatim}\n\n\\\\end{verbatim}'
```

Der erste Baustein gilt in allen Textdateien, aber nicht in Codedateien. Wenn Sie `mfg` und dann `↵` eingeben, macht Atom daraus die Floskel *Mit freundlichen Grüßen*. Der zweite Baustein gilt nur in  $\LaTeX$ -Dateien und hilft bei der Eingabe eines Codeblocks. Die vierfachen Backslashes sind notwendig, damit ein Backslash korrekt in den Text eingefügt wird.

## Markdown-Dateien bearbeiten

Das für mich seit etlichen Jahren wichtigste Textformat ist Markdown (siehe auch [Abschnitt 13.6](#), »Markdown und Pandoc«). Atom kommt auf Anhieb mit \*.md-Dateien zurecht. Die Tastenkombination `[Strg] + [⇧] + [M]` öffnet neben dem Text in einer neuen Pane eine Preview-Ansicht, die für die meisten Zwecke ausreicht. Damit übertrifft Atom quasi aus dem Stand und ohne jede Konfigurationsarbeit den Funktionsumfang vieler Markdown-Editoren.

### Pandoc-Integration

Standardmäßig geht die Markdown-Preview davon aus, dass Sie Markdown in der GitHub-Variante nutzen. Wenn Sie die erweiterten Formatierungsmöglichkeiten von Pandoc nutzen, müssen Sie das standardmäßig aktive Package `markdown-preview` deaktivieren (Button `DISABLE`) und stattdessen das Paket `markdown-preview-plus` installieren. Die Tastenkombination für die Vorschau bleibt mit `[Strg] + [⇧] + [M]` unverändert.

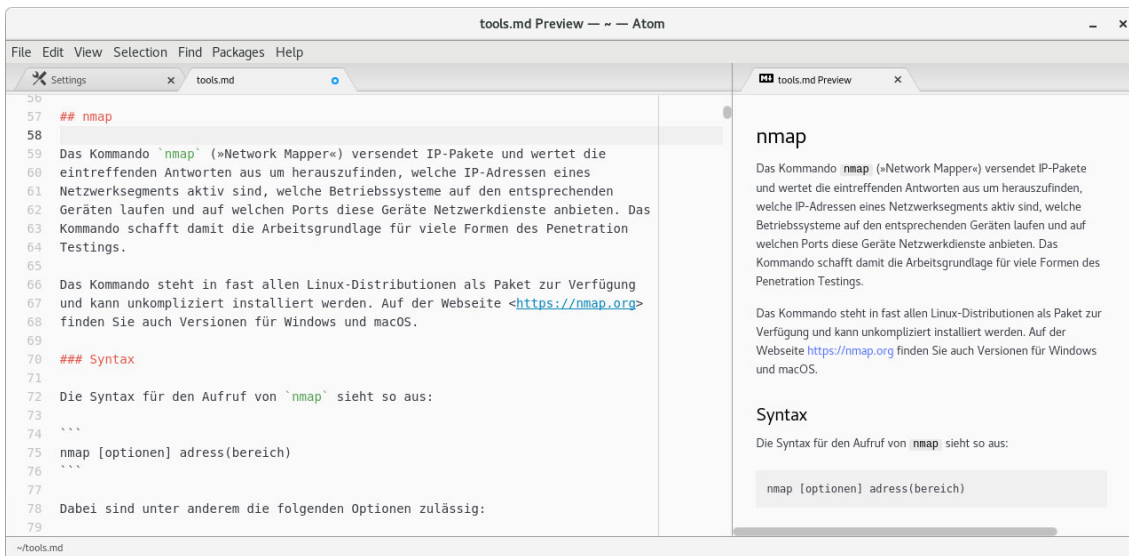


Abbildung 17.4 Links ein Markdown-Text, rechts die Vorschau

Der entscheidender Vorteil von `markdown-preview-plus` besteht darin, dass es anstatt der simplen GitHub-Markdown-Engine auch Pandoc aufrufen kann. Dazu müssen Sie einige Optionen des Packages konfigurieren:

- ▶ Sie aktivieren die Option `ENABLE PANDOC PARSER`.
- ▶ Bei der Option `PANDOC OPTIONS: PATH` geben Sie den Pfad zum `pandoc`-Kommando ein, also das Ergebnis von `which pandoc`.

- ▶ Mit `PANDOC OPTIONS: COMMANDLINE ARGUMENTS` können Sie die Optionen angeben, die beim Aufruf an `pandoc` übergeben werden sollen. Nicht zulässig ist hier allerdings die Option `-c my.css`: Eine eigene CSS-Datei für die Vorschau bringt leider die gesamte Atom-Oberfläche durcheinander.
- ▶ Mit `PANDOC OPTIONS: MARKDOWN FLAVOR` geben Sie an, welche Markdown-Variante `pandoc` verwenden soll. Standardmäßig gilt `markdown-raw_tex+tex_math_single_backslash`. Oft reicht einfach `markdown`, also der Standard-Markdown-Dialekt von Pandoc.

## 17.2 VSCode

RPM- und Debian-Pakete für VSCode finden Sie auf der Projektwebseite zum Download:

Installation und Updates

<https://code.visualstudio.com>

Bei den meisten Distributionen können Sie die Installation direkt aus dem Webbrowser heraus starten. Gelingt das nicht, verwenden Sie `rpm -i paketname` bzw. `dpkg -i paketname`.

Da es für VSCode aktuell keine offiziellen Paketquellen gibt, müssen Sie sich um eventuell veröffentlichte Updates selbst kümmern und gegebenenfalls den Download und die Installation wiederholen. Alternativ können Sie im Internet nach inoffiziellen Paketquellen suchen. Eine ist z. B. <https://github.com/tagplus5/vscode-ppa> für Ubuntu.

### Erste Schritte

In der Seitenleiste von VSCode können Sie über fünf Icons einen Verzeichnis-Browser, die Suchfunktion, ein Git-Menü, einen Debugger sowie die Extension-Verwaltung ein- und durch nochmaliges Anklicken wieder ausblenden. Für diese Aktionen gibt es jeweils auch `[Strg]+[Icon]`-Tastenkombinationen (siehe das ANZEIGEN-Menü).

Seitenleiste

Mehrere offene Dateien werden in Form von Dialogblättern (*Tabs*) angezeigt (siehe [Abbildung 17.5](#)). Außerdem können Sie den Editor vertikal in mehrere Gruppen unterteilen, z. B., um mehrere Dateien (oder Ausschnitte derselben Datei) nebeneinander darzustellen. Mit `[Strg]+[Tab]` wechseln Sie zwischen den Tabs der aktiven Gruppen, mit `[Strg]+[1]`, `[Strg]+[2]` etc. zwischen den Gruppen. Weitere Kommandos zum Wechseln zwischen Dateien und Gruppen finden Sie im Menü GEHE ZU.

Tabs und Gruppen

VSCode verwendet wie Atom die Tastenkombination `[Strg]+[Icon]+[P]` zum Aufruf eines kleinen Dialogs, in dem Sie alle verfügbaren Kommandos suchen und ausführen kön-

Menüs und Kommandos

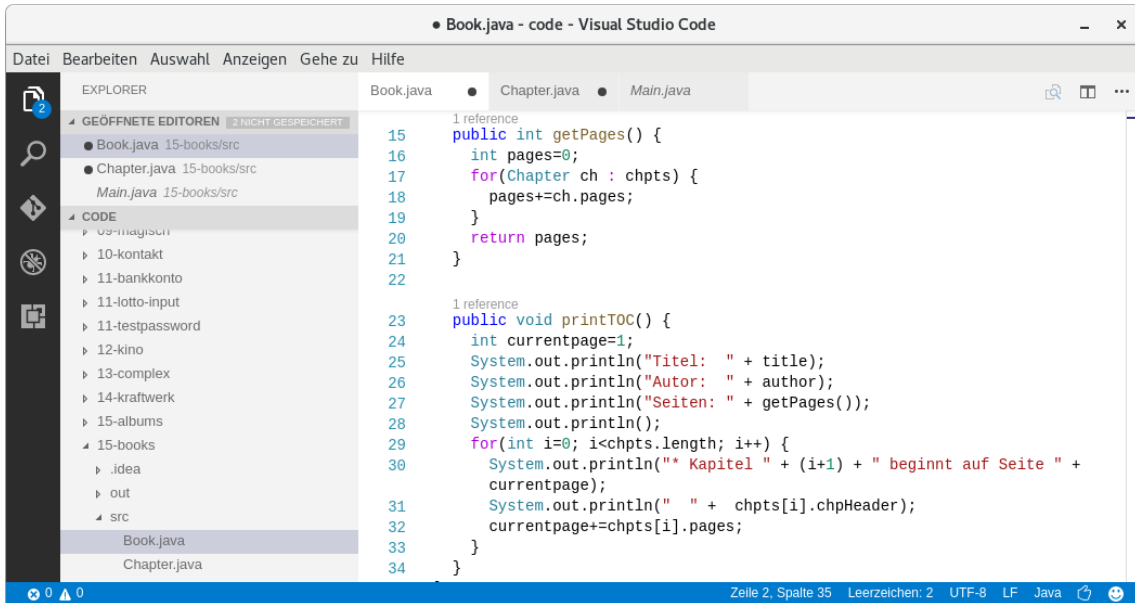


Abbildung 17.5 VSCode mit Seitenleiste und Tabs für mehrere geöffnete Java-Dateien

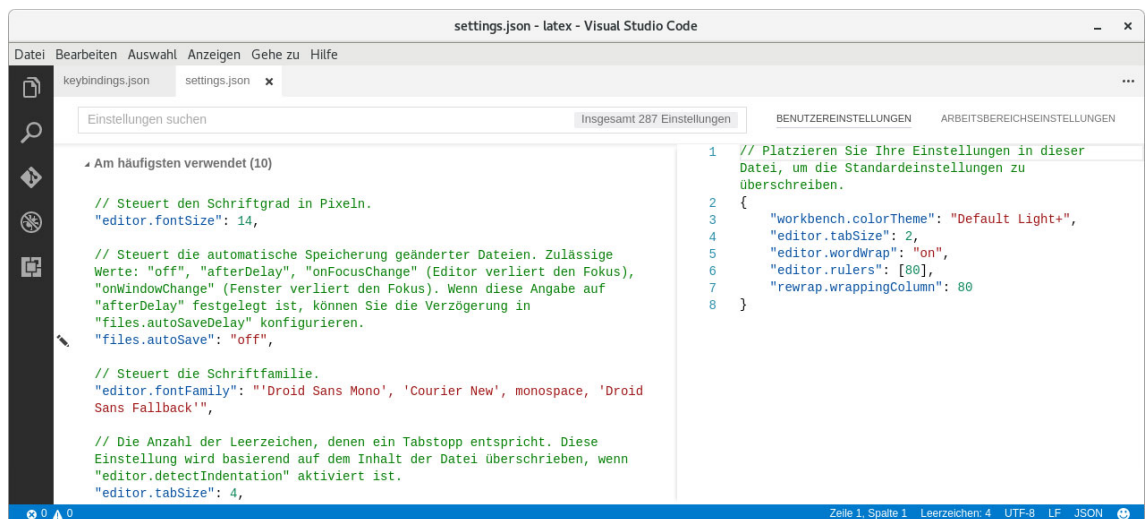
nen. Dabei müssen Sie allerdings nach den englischen Kommandonamen suchen, was angesichts deutschsprachiger Menüs inkonsequent ist.

Persönlich habe ich sowieso Zweifel, ob lokalisierte Menüs bei einem Editor für Programmentwickler eine gute Idee sind. Wenn Sie englische Menüs vorziehen, sollten Sie mit `[Strg]+[⇧]+[P]` `language` die englische Lokalisierung einstellen. (Sprachänderungen werden erst nach einem Neustart des Editors wirksam.)

- Ordner** Wenn Sie `DATEI • ORDNER ÖFFNEN` ausführen, entspricht dies einem Projektwechsel. VSCode schließt die aktuell geöffneten Dateien und wechselt in das neue Verzeichnis. Wenn Sie parallel Dateien aus mehreren Verzeichnissen bearbeiten möchten, müssen Sie vorher mit `[Strg]+[⇧]+[N]` ein weiteres VSCode-Fenster öffnen.
- Zen-Mode** Mit `[Strg]+[K],[Z]` aktivieren Sie den Zen-Modus. In diesem Modus wechselt VSCode in den Vollbildmodus und zeigt nur den aktuellen Text an, um jede Ablenkung zu vermeiden. Zur Rückkehr in den Standardmodus drücken Sie zweimal `[Esc]`.
- Terminal** VSCode enthält ein integriertes Terminal, das Sie mit `ANZEIGEN • TERMINAL` öffnen. Standardmäßig ist im Terminal das Verzeichnis aktiv, in dem sich die aktuell bearbeitete Datei befindet.

## Grundeinstellungen

Zur Veränderung der Grundeinstellungen führen Sie DATEI • EINSTELLUNGEN • EINSTELLUNGEN aus. VSCode zeigt dann links einen gut organisierten Text mit den Defaulteinstellungen an. Daraus kopieren Sie die für Sie relevanten Zeilen wahlweise in das Dialogblatt BENUTZER- oder ARBEITSBEREICHEINSTELLUNGEN und passen die Einstellungen Ihren Wünschen entsprechend an (siehe [Abbildung 17.6](#)). Das erste Dialogblatt enthält Einstellungen, die generell für VSCode gelten, das zweite solche, die nur für Dateien des aktuellen Verzeichnisses (»Projekts«) gelten. Die Einstellungen werden in `.config/Code/User/settings.json` (relativ zum Heimatverzeichnis) bzw. in `.vscode/settings.json` (relativ zum Projektverzeichnis) gespeichert.



**Abbildung 17.6** VSCode-Einstellungen werden nicht in Konfigurationsdialogen, sondern im JSON-Format in Konfigurationsdateien verändert.

VSCode ist durch sogenannte Extensions erweiterbar. ANSICHT • EXTENSIONS zeigt eine Liste aller aktuell installierten Extensions an und bietet die Möglichkeit, nach weiteren Extensions zu suchen und diese zu installieren. In bester Microsoft-Manier erfordern viele Extensions nach der Installation einen Neustart des Editors, um wirksam zu werden. Nach geeigneten Extensions können Sie auch auf der folgenden Webseite suchen:

**Extensions**

<https://marketplace.visualstudio.com/VSCode>

VSCode kann wie die meisten anderen Editoren zu lange Zeilen automatisch umbrechen (*Soft Wrap*, Parameter `editor.wordWrap`). Es gibt aber erstaunlicherweise keine Funktion für einen *Hard Wrap*, also für das Einfügen von echten Zeilenumbrüchen in mehrzeilige Texte oder Kommentare. Abhilfe sollte die Extension `rewrap` schaffen.

**Line Wrapping**

Bei meinen Tests bin ich damit aber nicht restlos glücklich geworden. Trotz der Einstellung von `rewrap.wrappingColumn` hat die Funktion bei jedem Aufruf wieder gefragt, wie lang die maximale Zeilenlänge sein darf.

**Light Theme** Wie Atom verwendet VSCode standardmäßig dunkle Farben. Wenn Sie ein helleres Erscheinungsbild vorziehen, wählen Sie mit DATEI • EINSTELLUNGEN • FARBDESIGN ein anderes Farbschema aus. Mit dem Menükommando wird automatisch eine geeignete Zeile in `settings.json` eingefügt.

## Konfiguration der Tastatur und andere Eingabeerleichterungen

**Tastenkürzel** Zur Veränderung von Tastenkürzeln führen Sie DATEI • EINSTELLUNGEN • TASTENKOMBINATIONEN aus. Wie bei den Grundeinstellungen zeigt VSCode dann links den Code für alle schon definierten Kürzel an. Um ein Kürzel zu ändern, kopieren Sie es in das anfangs leere rechte Dialogblatt und passen den Code dann an. Die folgenden Zeilen geben einige Beispiele:

```
[
  { "key": "ctrl+e", "command": "cursorEnd",
    "when": "editorTextFocus" },
  { "key": "ctrl+a", "command": "cursorHome",
    "when": "editorTextFocus" },
  { "key": "ctrl+d", "command": "deleteRight",
    "when": "editorTextFocus && !editorReadOnly" },
  { "key": "f4", "command": "rewrap.rewrapCommentAt",
    "when": "editorTextFocus && !editorReadOnly" }
]
```

Mit `Strg+S` speichern Sie die Einstellungen in `.config/Code/User/keybindings.json`. Mit dem Speichern wird die geänderte Konfiguration sofort wirksam.

### Keymaps anderer Editoren

DATEI • EINSTELLUNGEN • ERWEITERTE TASTENZUORDNUNGEN hilft bei der Installation von Extensions, die die Keymaps anderer Editoren nachbilden (Atom, Eclipse, Emacs, Sublime, Vim etc.).


**IntelliSense** Die automatische Vervollständigung von (Schlüssel-)Wörtern heißt in Microsoft-Produkten schon seit Jahrzehnten »IntelliSense«. Tatsächlich hat die Funktion bei meinen Tests in VSCode einen intelligenteren und vor allem problemloseren Eindruck als bei Atom hinterlassen. IntelliSense funktioniert ohne Konfigurationsärger auf Anhieb. Noch zielsicherer sind die IntelliSense-Vorschläge freilich, wenn Sie eine Extension mit spezifischen VSCode-Erweiterungen für die Programmiersprache Ihrer Wahl installieren.



Sie können häufig benötigte Codestrukturen oder Texte mit einer Abkürzung versehen. Dazu führen Sie DATEI • EINSTELLUNGEN • BENUTZERCODEAUSSCHNITTE aus und wählen eine Sprache oder ein Textformat aus. Damit gelangen Sie in eine Snippet-Datei für die jeweilige Sprache. Dort können Sie nun wie im folgenden Beispiel für die Programmiersprache Java Codeblöcke samt einer dazugehörigen Abkürzung (hier `syso`) definieren:

```
{
  "println": {
    "prefix": "syso",
    "body": [ "System.out.println();" ],
    "description": "Print to console"
  }
}
```

Wenn Sie nun `syso` eingeben, schlägt VSCode als Vervollständigung den Textblock (body) vor. VSCode-Snippets sind immer sprachabhängig. Es gibt aktuell noch keine Möglichkeit, einen gemeinsamen Textblock für mehrere Code- oder Texttypen zu definieren.

VSCode unterstützt auch Emmets. Das ist eine spezielle Syntax zur Eingabe von HTML- und CSS-Code. Wenn Sie in einer HTML-Datei beispielsweise `nav>ul>li` und dann  eingeben, macht VSCode daraus den folgenden Codeblock:

```
<nav>
  <ul>
    <li></li>
  </ul>
</nav>
```

Eine Menge Beispiele für die Emmet-Syntax finden Sie hier:

<https://docs.emmet.io/cheat-sheet>



TEIL V

# **Systemkonfiguration und Administration**



# Kapitel 18

## Basiskonfiguration

Dieses Kapitel ist das erste einer ganzen Reihe von Kapiteln zur Linux-Systemkonfiguration. Nach einigen einleitenden Informationen geht es in diesem Kapitel um elementare Funktionen:

- ▶ Konfiguration von Textkonsolen
- ▶ Einstellung von Datum und Uhrzeit
- ▶ Benutzerverwaltung
- ▶ Internationalisierung, Zeichensatz, Unicode
- ▶ Überblick über die Hardware-Konfiguration
- ▶ Logging-Dateien

Die weiteren Kapitel behandeln dann die Paketverwaltung, die Verwaltung der Systembibliotheken, die Konfiguration des Grafiksystems (X/Wayland), die Administration des Dateisystems, den Systemstart (GRUB, Init-System) und den Umgang mit dem Kernel und seinen Modulen.

### 18.1 Einführung

Dieses und die folgenden Kapitel geben Ihnen einen Blick hinter die Kulissen der Linux-Konfigurationsprogramme. Sie sollen verstehen, was wie wo gesteuert und voreingestellt wird. Daher werden Sie hier eine Menge Hintergrundinformationen darüber finden, wie das Gesamtsystem funktioniert.

Leider unterscheiden sich die diversen Distributionen bei der Konfiguration in vielen kleinen Details. In diesem Buch versuche ich, den gemeinsamen Nenner möglichst vieler Linux-Systeme zu beschreiben. Dennoch kann es vorkommen, dass gerade bei Ihrer Distribution einzelne Details ein wenig anders gelöst sind. In solchen Fällen bleibt Ihnen ein Blick in die Dokumentation bzw. eine Internet-Suche nicht erspart.

Auch wenn das für Sie vielleicht hin und wieder unangenehm ist, bin ich dennoch überzeugt, dass der allgemeingültige Ansatz der bessere ist als ein Buch zur Red-Hat-Administration, ein weiteres zur SUSE-Administration etc. – nicht zuletzt deswegen,

weil sich ja auch die einzelnen Distributionen von Version zu Version ändern. Über kurz oder lang müssen Sie also in jedem Fall lernen, selbst die oft englischsprachigen Manuals, Hilfeseiten etc. zu lesen und zu verstehen. Dieses Buch will keine Originalhandbücher oder Schritt-für-Schritt-Anleitungen ersetzen, sondern Grundlagenwissen vermitteln!

**Wer ist hier der Systemadministrator?**

Bisher war der sogenannte Systemadministrator vielleicht irgendeine fremde Person, die Ihnen – oft unwillig und überarbeitet – zu Hilfe kam. Wenn Sie nicht in einem großen Betrieb arbeiten, dann ist der Administrator wohl überhaupt nur ein abstrakter Begriff aus vielen Büchern, so etwa nach dem Motto: »Wenn's nicht mehr weitergeht, fragen Sie den Systemadministrator ...«

Indem Sie Linux selbst auf Ihren Rechner installiert haben, ändert sich dieses Bild: Nun sind *Sie* der Systemadministrator! Erschrecken Sie nicht vor diesem Begriff – der Systemadministrator ist einfach die Person, die sich um die Konfiguration des Rechners kümmert. Solange es um die Linux-Grundfunktionen geht, kann das jeder!

**Konfigurations- und Administrationswerkzeuge**

Einige Distributionen bieten komfortable Konfigurationsprogramme an, die sowohl während als auch nach der Installation verwendet werden können. Besonders zeichnet sich hier SUSE mit YaST aus. Auch Desktop-Systeme wie KDE und Gnome enthalten Konfigurationswerkzeuge, deren Wirkung über den Desktop hinausgeht. Diese Werkzeuge sollten bei grundlegenden Konfigurationsproblemen immer die erste Wahl sein!

Neben den mitgelieferten Konfigurationsprogrammen gibt es auch externe Werkzeuge bzw. eigene Linux-Distributionen mit zusätzlichen Administrationswerkzeugen. Viele davon lassen sich über eine Webschnittstelle bedienen. [Tabelle 18.1](#) gibt einen Überblick über einige populäre Werkzeuge. Beachten Sie bitte, dass der Einsatzzweck und der Funktionsumfang dieser Werkzeuge sehr weit variiert. Ein Teil der aufgezählten Werkzeuge ist speziell zur Wartung vieler gleichartiger Linux-Installationen gedacht; andere sind nur zur Server-Konfiguration vorgesehen.

Ein Bullet in der Spalte FREI bedeutet, dass es sich um Open-Source-Software handelt, die auch in Unternehmen kostenlos genutzt werden kann. Auch bei den meisten kommerziellen Produkten gibt es freie Varianten mit reduziertem Funktionsumfang.

### Vermeiden Sie Abhängigkeiten!

Mit kommerziellen Administrationswerkzeugen rutschen Sie leicht in neue Abhängigkeiten. Zudem habe ich in der Vergangenheit schon eine Menge Administrationswerkzeuge kommen und wieder gehen gesehen. Entscheiden Sie sich nicht leichtfertig für externe Administrationswerkzeuge!

| Link  | Funktion                                | frei |
|---|---|------|
| <a href="http://webmin.com">http://webmin.com</a>                                   | System- und Netzwerkadministration      | •    |
| <a href="http://fai-project.org">http://fai-project.org</a>                         | Software-Verteilung und -Installation   | •    |
| <a href="http://m23.sourceforge.net">http://m23.sourceforge.net</a>                 | Software-Verteilung und -Administration | •    |
| <a href="http://directory.fedoraproject.org">http://directory.fedoraproject.org</a> | LDAP-Benutzeroberfläche für RHEL/Fedora | •    |
| <a href="https://sso.redhat.com">https://sso.redhat.com</a>                         | Red-Hat-Administration                  |      |
| <a href="https://landscape.canonical.com">https://landscape.canonical.com</a>       | Ubuntu-Administration                   |      |
| <a href="https://univention.de/produkte/ucs">https://univention.de/produkte/ucs</a> | LAN- und Mail-Server-Konfiguration      |      |
| <a href="http://zentyal.org">http://zentyal.org</a>                                 | LAN-Server-Konfiguration                |      |
| <a href="https://cpanel.net">https://cpanel.net</a>                                 | Root- und Webserver-Administration      |      |
| <a href="https://plesk.com">https://plesk.com</a>                                   | Root- und Webserver-Administration      |      |

**Tabelle 18.1** Ausgewählte Administrationswerkzeuge

Ausgefeilte Konfigurationswerkzeuge mit schönen Benutzeroberflächen nehmen Ihnen die Mühe ab, Linux-Konfigurationsdateien direkt zu verändern. Gerade für Linux-Einsteiger ist dies zweifellos praktisch. Es gibt aber eine ganze Reihe von Gründen, sich dennoch mit den Konfigurationsdateien und damit mit den Interna von Linux auseinanderzusetzen:

**Konfigurationsdateien**

- ▶ Die Konfigurationsdateien lassen sich mit jedem beliebigen Texteditor verändern, auch in einer Textkonsole oder über eine SSH-Verbindung.
- ▶ Sobald Sie einmal verstanden haben, wie die Konfiguration einer bestimmten Linux-Funktion erfolgt, können Sie dieses Wissen bei beinahe jeder anderen Linux-Distribution anwenden.
- ▶ Nur durch die direkte Veränderung der Konfigurationsdateien können Sie alle Aspekte einer Systemfunktion steuern. Konfigurationswerkzeuge beschränken sich dagegen oft auf einige besonders wichtige Details.
- ▶ Konfigurationsdateien lassen sich leicht von einem Rechner zum anderen kopieren. Das kann eine Menge Zeit sparen, wenn Sie einen Distributionswechsel durchführen, Linux auf einem anderen Rechner neu installieren etc.
- ▶ Je besser Sie verstehen, wie die Konfigurationsdateien aufgebaut sind und welche Steuerungsmöglichkeiten sie bieten, desto besser verstehen Sie Linux und desto weniger ist Ihr Rechner die sprichwörtliche »Black Box«, in die keiner hineinblicken kann.

**Das Zeilenende kann entscheidend sein!**

Achten Sie beim Editieren von Konfigurationsdateien darauf, dass auch die letzte Zeile mit `↵` abgeschlossen wird. Manche Linux-Programme bearbeiten Dateien nicht korrekt, wenn in der letzten Zeile das Zeilenende fehlt.

**/etc-Verzeichnis** Fast alle Linux-Konfigurationsdateien befinden sich im `/etc`-Verzeichnis. Eine Referenz aller im Buch behandelten Konfigurationsdateien finden Sie daher im Stichwortverzeichnis unter dem Buchstaben E. Zusammengehörende Konfigurationsdateien größerer Programme sind oft in eigenen Unterverzeichnissen organisiert (siehe [Tabelle 18.2](#)).

| Verzeichnis                          | Inhalt   |
|--------------------------------------|--|
| <code>/etc/cron*</code>              | Cron-Dateien (siehe <a href="#">Abschnitt 12.6</a> )     |
| <code>/etc/default</code>            | distributionsspezifische Dateien (Debian, Ubuntu)        |
| <code>/etc/init.d, /etc/rc*.d</code> | Init-V-System (siehe <a href="#">Abschnitt 24.2</a> )    |
| <code>/etc/systemd</code>            | systemd (siehe <a href="#">Abschnitt 24.1</a> )          |
| <code>/etc/sysconfig</code>          | distributionsspezifische Dateien (Fedora, Red Hat, SUSE) |
| <code>/etc/X11</code>                | Grafiksystem   |

**Tabelle 18.2** Wichtige `/etc`-Verzeichnisse

**/etc-Backup** Es ist eine gute Idee, eine Sicherheitskopie des gesamten `/etc`-Verzeichnisses anzulegen. Damit können Sie nach Änderungen jederzeit rasch feststellen, wie der ursprüngliche Zustand einer bestimmten Konfigurationsdatei war.

```
root# mkdir /etc-backup
root# cp -a /etc/* /etc-backup
```

**Konfigurationsdateien suchen** Wenn Sie eine Konfigurationsdatei in Ihrer Distribution nicht finden, kann das mehrere Ursachen haben: Möglicherweise sind die zugrunde liegenden Programmpakete gar nicht installiert, oder die Konfigurationsdateien befinden sich bei Ihrer Distribution an einem anderen Ort. Verwenden Sie zur Suche die Kommandos `locate`, `find` und `grep`. Das folgende Kommando zeigt, wie Sie in `/etc` und allen Unterverzeichnissen nach Dateien suchen können, deren Inhalt (nicht der Dateiname) das Wort `abcde` in beliebiger Groß- und Kleinschreibung enthält:

```
root# cd /etc
root# grep -r -i abcde
```



Bei manchen Programmen werden Änderungen an den Konfigurationsdateien erst wirksam, wenn Sie das Programm neu starten bzw. es explizit dazu auffordern, die Konfigurationsdateien neu einzulesen. Dazu ist `systemctl reload` vorgesehen. Einige wenige Dienste unterstützen `reload` nicht – dann müssen Sie stattdessen `restart` verwenden.

Neue  
Konfiguration  
aktivieren

```
root# systemctl reload funktionsname      (Distributionen mit systemd)
root# systemctl restart funktionsname
```

Anders als unter Windows ist es fast nie erforderlich, den Rechner neu zu starten. Ausnahmen sind nur Veränderungen am Kernel sowie einige hardware-spezifische Einstellungen, die nur unmittelbar beim Systemstart durchgeführt werden können.

## 18.2 Konfiguration der Textkonsolen

Bei modernen Distributionen startet Linux direkt das Grafiksystem, und Linux-Einsteiger wissen oft gar nicht, dass es auch Textkonsolen gibt. Hin und wieder kommt es freilich vor, dass die Konfiguration für das Grafiksystem fehlerhaft ist oder aus anderen Gründen kein grafisches System zur Verfügung steht. Bei Server-Installationen bzw. in virtuellen Maschinen wird oft bewusst auf das Grafiksystem verzichtet. In solchen Fällen müssen Sie sich mit den Textkonsolen anfreunden. Für elementare Einstellungen wie das Tastaturlayout und die Schriftart ist je nach Distribution entweder das `kbd`-System oder das neuere `console`-System verantwortlich. Im Detail sieht die Konfiguration bei jeder Distribution ein wenig anders aus.

### Tastaturlayout

Unter Debian und Ubuntu kümmern sich die Programme des Pakets `console-setup` um das Tastaturlayout. Die Konfigurationsdatei `/etc/default/keyboard` steuert das Tastaturlayout.

Debian, Ubuntu

```
# /etc/default/keyboard
# Tastatur
XKBMODEL="pc105"
XKBLAYOUT="de"
XKBVARIANT=""
XKBOPTIONS="lv3:ralt_switch"
```

Egal, ob Debian oder Ubuntu: Veränderungen am Tastaturlayout sollten Sie nach Möglichkeit nicht durch eine direkte Veränderung der Konfigurationsdateien vornehmen, sondern mit dem folgenden Kommando:

```
root# dpkg-reconfigure keyboard-configuration
```

Um die Auswertung der Dateien kümmert sich `systemd` (Service `keyboard-setup`). Unter Debian führt `systemd` das Script `/lib/console-setup/keyboard-setup.sh` aus, unter Ubuntu verzweigt `systemd` dagegen in das Init-V-Script `/etc/init.d/keyboard-setup.sh` (zumindest bis Version 17.04).

**Fedora sowie CentOS und RHEL**

Bei Fedora sowie CentOS und RHEL ab Version 7 sind das `kbd`-Paket und `systemd` für die Einstellung des Tastaturlayouts zuständig. Die Konfiguration wird in zwei Dateien gespeichert: in `/etc/vconsole.conf` für den Textmodus und in `/etc/X11/xorg.conf.d/00-keyboard.conf` für das Grafiksystem X.

```
# Datei /etc/vconsole.conf
KEYMAP="de"
FONT="eurlatgr"
```

Das Kommando `localectl list-keymaps` liefert eine Liste aller möglichen Tastaturlayouts. Zur Konfiguration können Sie `localectl set-keymap` verwenden. `localectl` versucht die Einstellung auch für den Grafikmodus zu übernehmen, was aber nicht immer gelingt. Im folgenden Beispiel werden zuerst alle deutschen Tastaturlayouts ermittelt, und dann wird die Variante für Apples Mac-Tastatur eingestellt:

```
root# localectl list-keymaps | egrep '^de.*|de-'
de
de-deadacute
de-mac
...
root# localectl set-keymap de-mac
```

Für die Auswertung der Konfiguration ist der `systemd`-Service `systemd-localed` zuständig. Bei CentOS 7 und RHEL 7 (aber nicht bei Fedora!) kümmert sich darüber hinaus auch GRUB um das Tastaturlayout, wobei die GRUB-Einstellungen in Textkonsolen Vorrang haben. Zur Veränderung der Tastaturlayouts müssen Sie deswegen in der Datei `/etc/default/grub` den Parameter `GRUB_CMDLINE_LINUX` neu einstellen:

```
# in der Datei /etc/default/grub
...
GRUB_CMDLINE_LINUX="... vconsole.keymap=de-latin1 ..."
```

Damit hier geänderte Einstellungen wirksam werden, müssen Sie `grub2-mkconfig` ausführen und den Rechner neu starten:

```
root# grub2-mkconfig -o /boot/grub2/grub.cfg
root# reboot
```

**SUSE** Auch SUSE verwendet das `kbd`-Paket. Die Konfiguration befindet sich in `/etc/sysconfig/keyboard`. Sie wird bei aktuellen SUSE-Versionen durch `systemd` ausgewertet (Service `systemd-vconsole-setup`). Die Einstellungen gelten nur für die Konsole, nicht für den Grafikmodus.

## Schriftart

Konsolen sind grundsätzlich Unicode-kompatibel. Allerdings ist die Maximalanzahl der möglichen Zeichen in Konsolenschriften sehr klein (256 bzw. 512). Daher können Konsolenschriften immer nur einen winzigen Bruchteil aller Unicode-Zeichen abbilden.

Die Konfigurationseinstellungen befinden sich in `/etc/default/console-setup`. Die Datei wird durch den `systemd`-Service `console-setup` ausgewertet. **Debian, Ubuntu**

```
# /etc/default/console-setup
...
# Schriftart
CHARMAP="UTF-8"
CODESET="Lat15"
FONTFACE="VGA"
FONTSIZE="16"
```

Fedora sowie CentOS und RHEL ab Version 7 speichern die gewünschte Konsolenschrift in der Datei `/etc/vconsole.conf`. Diese Datei wird durch `systemd` ausgewertet (Service `systemd-vconsole-setup`). **Fedora sowie CentOS und RHEL**

Bei aktuellen SUSE-Versionen wird die Konsolenschrift in `/etc/sysconfig/console` eingestellt. `systemd` wertet diese Datei aus (Service `systemd-vconsole-setup`). Standardmäßig kommt die Schrift `eurlatgr.psfu` zum Einsatz, die neben den Latin-1-Zeichen auch das Euro-Symbol enthält. **SUSE**

## gpm-Konfiguration (Maus)

Die Verwendung der Maus ist eigentlich nur im Grafiksystem vorgesehen. Das Programm `gpm` erlaubt eine eingeschränkte Benutzung der Maus aber auch in Textkonsolen: Insbesondere können Sie nun mit der linken Maustaste Text markieren und ihn mit der mittleren oder rechten Maustaste an der aktuellen Cursorposition einfügen. Beachten Sie aber, dass Sie bei den meisten Konsolenprogrammen die Cursorposition nicht mit der Maus verändern können.

Sofern `gpm` installiert ist, wird es durch das Init-System gestartet, je nach Distribution durch ein altes Init-V-Script oder durch eine `systemd`-Service-Datei. Die Konfiguration erfolgt durch `/etc/gpm.conf` oder `/etc/sysconfig/mouse`, wobei es selten erforderlich ist, die Standardeinstellungen zu verändern.

### 18.3 Datum und Uhrzeit

Wegen der internationalen Vernetzung von Rechnern ist die Verwendung einer weltweit einheitlichen Uhrzeit erforderlich, nämlich der Greenwich Mean Time. Auf Unix-Rechnern ist die GMT das Maß aller Dinge bzw. der Zeit. Anstelle von GMT ist als zweite Abkürzung auch UTC üblich (Universal Time, Coordinated).

Wenn Sie eine Datei speichern, dann wird nicht die aktuelle Ortszeit gespeichert, sondern eine auf diesen internationalen Standard umgerechnete Zeit. Wenn Sie die Datei anschließend mit `ls -l` ansehen, wird die Uhrzeit wieder auf die Ortszeit am Standort des Rechners zurückgerechnet. Dieses Verfahren ermöglicht es festzustellen, welche Datei aktueller ist: eine um 18:00 Uhr Ortszeit in München gespeicherte Datei oder eine um 12:30 Uhr Ortszeit in New York gespeicherte Datei.

**Zeiteinstellung  
während des  
Rechnerstarts**

Die Zeiteinstellung während des Rechnerstarts erfolgt zumeist in zwei Phasen: Zuerst wird die Uhr des Mainboards ausgelesen. Ihr Hauptnachteil ist die relativ geringe Genauigkeit. Sobald eine Internetverbindung besteht, kann Linux seine Zeit mit anderen Servern im Internet synchronisieren (siehe [Abschnitt 18.4](#)).

Zum Auslesen der Hardware-Uhr, die oft auch CMOS-Uhr oder *Real Time Clock* (RTC) genannt wird, dient das Kommando `hwclock`. Bei modernen Distributionen wird es durch `systemd` ausgeführt (Service `systemd-timedated`).

**Lokale Zeit oder  
GMT?**

Die CMOS-Uhr kann auf die lokale Uhrzeit oder auf GMT gestellt sein. Für Linux-Rechner ist die GMT-Variante empfehlenswert, nicht zuletzt deswegen, um dem halbjährlichen Ärger mit der Zeitumstellung auf Sommer- bzw. Winterzeit aus dem Weg zu gehen.

Windows geht allerdings standardmäßig davon aus, dass die CMOS-Uhr die lokale Zeit enthält. Linux kommt auch mit diesem Fall zurecht, muss aber wissen, welche Zeit die Hardware-Uhr nun enthält. Diese Information können Sie mit `timedatectl` fixieren:

```
root# timedatectl set-local-rtc 0 (Hardware-Uhr enthält GMT)
root# timedatectl set-local-rtc 1 (Hardware-Uhr enthält lokale Zeit)
```

`timedatectl set-local-rtc` verändert die dritte Zeile der Datei `/etc/adjtime`, die entweder UTC oder LOCAL enthält (siehe auch `man hwclock`).

Bei Dual-Boot-Systemen mit Windows und Linux haben Sie zwei Möglichkeiten: Entweder belassen Sie in der CMOS-Uhr die lokale Zeit und teilen dies Linux mit (manche Distributionen wie Ubuntu erkennen diesen Fall automatisch während der Installation), oder aber Sie stellen die CMOS-Uhr auf GMT und teilen dies Windows mit. Dazu müssen Sie lediglich einen Eintrag in der Registry ändern. Das folgende Kommando ist in einem Eingabeaufforderungsfenster mit Administratorrechten auszuführen.

Das Kommando ist hier aus Platzgründen auf drei Zeilen verteilt, Sie müssen es aber in einer einzigen langen Anweisung eingeben.

```
> reg.exe add
"HKKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\TimeZoneInformation"
/v RealTimeIsUniversal /d 1 /t REG_DWORD /f
```

(Quelle: [https://wiki.archlinux.org/index.php/Time#UTC\\_in\\_Windows](https://wiki.archlinux.org/index.php/Time#UTC_in_Windows))

Damit Kommandos wie `ls` oder die Dateimanager von KDE und Gnome die GMT-Zeit in die lokale Zeit umrechnen und entsprechend anzeigen können, muss jedes Programm wissen, in welcher Zeitzone es läuft. Fast alle Linux-Programme greifen dazu auf Funktionen der `glibc`-Bibliothek zurück. Diese Bibliothek wertet die Datei `/etc/localtime` aus. Bei dieser Datei handelt es sich um den Link zu einer Zeitzonendatei aus dem Verzeichnis `/usr/share/zoneinfo` oder um eine Kopie dieser Datei. Die Zeitzonendatei enthält auch Informationen dazu, ob in der betreffenden Zeitzone ein bestimmtes Sommerzeitschema gilt (z. B. Zeitzone MESZ für die mitteleuropäische Sommerzeit).

Einstellung der  
Zeitzone

Die aktuelle Einstellung ermitteln Sie am komfortabelsten mit `timedatectl status`:

```
user$ timedatectl status
Local time: Sat 2017-06-03 07:08:55 CEST
Universal time: Sat 2017-06-03 05:08:55 UTC
RTC time: Mon 2017-05-08 10:49:43
Time zone: Europe/Berlin (CEST, +0200)
Network time on: no
NTP synchronized: yes
RTC in local TZ: no
```

Zur Änderung der Zeitzone ermitteln Sie mit `timedatectl list-timezones` alle zur Auswahl stehenden Zeitzonen und stellen die gewünschte Zeitzone dann mit `timedatectl set-timezone` ein:

```
user$ timedatectl list-timezones | grep Europe
Europe/Amsterdam
Europe/Andorra
Europe/Astrakhan
Europe/Athens
...
root# timedatectl set-timezone Europe/Vienna
```

Hinter den Kulissen ändert `timedatectl set-timezone` einfach den Link von `/etc/localtime`:

```
root# ln -s -f /usr/share/zoneinfo/Europe/Vienna /etc/localtime
```

Bei Debian und Ubuntu gibt auch die Textdatei `/etc/timezone` die Zeitzone an. Diese Datei wird aber nicht unmittelbar von der `glibc`-Bibliothek ausgewertet, sondern nur von den Werkzeugen zur Neueinstellung der `localtime`-Datei.

**Konfigurationswerkzeuge** Je nach Distribution, Desktop- bzw. Init-System können Sie ein Konfigurationsprogramm verwenden, um die Zeitzone sowie Datum und Uhrzeit der Rechner-Uhr zu verändern:

|                 |   |
|-----------------|---|
| Gnome:          | Systemeinstellungsmodul DATUM UND ZEIT              |
| KDE:            | Kontrollzentrum-Modul SYSTEMVERWALTUNG • DATUM/ZEIT |
| systemd:        | timedatectl   |
| Debian, Ubuntu: | dpkg-reconfigure tzdata                             |
| SUSE:           | YaST-Modul SYSTEM • ZEITZONE                        |

## 18.4 Datum und Uhrzeit via NTP synchronisieren

Sobald ein Rechner über eine Internetverbindung verfügt, ist es sinnvoll, die lokale Uhrzeit regelmäßig mit Zeit-Servern im Internet abzugleichen. Dazu ist das *Network Time Protocol* (NTP) vorgesehen. Mit NTP gelingt es, dass die lokale Zeit auf dem Computer nur minimal (im Mikrosekundenbereich) von der durch Atomuhren vorgegebenen Zeit abweicht. Ein weiterer Vorteil besteht darin, dass die nun (fast) vollkommen exakte Zeit auch dazu verwendet werden kann, hin und wieder die CMOS-Uhr nachzujustieren.

Zur Nutzung von NTP bestehen diverse Möglichkeiten, und selbstredend verwendet nahezu jede Distribution einen anderen Mechanismus:

- ▶ Bei Distributionen mit systemd (siehe auch [Abschnitt 24.1](#)) kann das Target `time-sync` aktiv sein. Das führt zum Start des Dämons `systemd-timesyncd`, der sich um die Zeiteinstellung via NTP kümmert.
- ▶ Das Kommando `ntpdate` bezieht *einmal* die exakte Zeit und stellt die Uhr des Rechners. Bei Rechnern, die häufig ein- und ausgeschaltet werden, ist das ausreichend genau.
- ▶ Gewissermaßen die klassischen NTP-Programme sind `ntpdate`, um Datum und Uhrzeit einmalig einzustellen, sowie `ntpd`, um die Uhrzeit dauerhaft mit externen NTP-Servern zu synchronisieren und die aktuelle Zeit bei Bedarf selbst an andere Clients weiterzugeben. Auch wenn Sie auf einem Rechner `ntpd` einsetzen, ist `ntpdate` praktisch, um die Uhrzeit erstmalig exakt einzustellen. `ntpd` funktioniert nämlich nur dann, wenn die anfängliche Abweichung zwischen der exakten und der lokalen Uhrzeit kleiner als eine Minute ist.

**Links** Weitere Informationen zur Verwaltung von Datum und Uhrzeit finden Sie auf den folgenden Seiten:

<http://ntp.org>

<http://tldp.org/HOWTO/TimePrecision-HOWTO>

[https://wiki.archlinux.org/index.php/Network\\_Time\\_Protocol\\_daemon](https://wiki.archlinux.org/index.php/Network_Time_Protocol_daemon)

### Abrupte Zeitänderungen sind gefährlich

Es gibt eine ganze Menge Programme, die plötzliche Zeitänderungen schlecht vertragen. Dazu zählen beispielsweise das Authentifizierungssystem Kerberos, der POP3- und IMAP-Server Dovecot sowie Datenbank-Server. Es empfiehlt sich, solche Programme vor der Ausführung von `ntpdate` bzw. vor der manuellen Veränderung der Uhrzeit herunterzufahren und anschließend neu zu starten. Dovecot endet automatisch, wenn es entdeckt, dass die Uhrzeit zurückgestellt wurde.

### systemd-timesyncd (Debian, Ubuntu)

Auf aktuellen Debian- und Ubuntu-Distributionen läuft in der Regel der Hintergrundprozess `systemd-timesyncd`. Es wird als Teil des `systemd`-Targets `time-sync` ausgeführt. Die `systemd`-Dokumentation sieht die Aktivierung dieses Targets eigentlich nur vor, wenn `systemd` Init-V-Scripts ausführen muss, die auf eine exakte Zeit angewiesen sind (siehe `man systemd.special`). Unter Debian und Ubuntu wird der Service `systemd-timesyncd` aber auch im Rahmen des `sysinit`-Targets aktiviert.

Der Hintergrunddienst `systemd-timesyncd` greift auf NTP-Server zu, die bereits beim Kompilieren fixiert wurden. Diese können Sie wie folgt ermitteln:

```
root# strings /lib/systemd/systemd-timesyncd | grep ntp
sd_network_get_ntp
ntp.ubuntu.com
```

Abweichend davon können in `/etc/systemd/timesyncd.conf` andere NTP-Server genannt werden.

### Chrony (Fedora, CentOS und RHEL)

Unter Fedora wurden die klassischen NTP-Tools schon vor mehreren Jahren durch das Programm Chrony ersetzt. Es eignet sich besonders gut für Notebooks und virtuelle Maschinen, die nicht ständig mit dem Internet verbunden sind und deren Zeit nach längeren Offline-Perioden oft deutlich korrigiert werden muss.

Die Konfiguration erfolgt durch `/etc/chrony.conf`. Standardmäßig bewirkt dort der Parameter `pool`, dass Chrony die Zeit von mehreren öffentlichen NTP-Servern des Projekts `http://pool.ntp.org` bezieht. Alternativ können Sie mit `server` explizit eigene NTP-Server angeben (siehe `man chrony.conf`).

Konfiguration

```
# Datei /etc/chrony.conf
pool 2.fedora.pool.ntp.org iburst
...
```

Sollte die automatische Neueinstellung der Uhrzeit nach einer längeren Offline-Periode nicht korrekt funktionieren, starten Sie Chrony mit dem folgenden Kommando neu:

```
root# systemctl restart chronyd
```

**Status ermitteln** Chrony besteht aus zwei Programmen: dem Hintergrunddienst (Dämon) `chronyd` und dem Kommando `chronyc`. Letzteres ermöglicht die Einstellung diverser Parameter sowie die Überwachung des Chrony-Status. `chronyc help` gibt einen Überblick über die zur Auswahl stehenden Kommandos. `chrony sources` fasst ähnlich wie `ntpq -p` zusammen, welche NTP-Server Chrony als Zeitquelle verwendet und wie groß die Zeitabweichungen gerade sind.

```
root# chronyc sources
210 Number of sources = 4
MS Name/IP address          Stratum Poll Reach LastRx Last sample
=====
^* ntp.cnh.at                2   8   377   458   -129us[ -163us] +/-  49ms
^+ mail.hoco.at              2   9   377   270   +107us[ +107us] +/-  86ms
^+ 193.170.62.252            2  10   377   134   -421us[ -421us] +/-  62ms
^+ ns5.nosuchhost.net        2   9   377   337   +1123us[+1123us] +/-  42ms
```

**Chrony unter RHEL 7 / CentOS 7** Die Desktop-Installationen von RHEL und CentOS verwenden ab Version 7 wie Fedora Chrony. Die Pakete `ntp` und `ntpdate` sind zwar ebenfalls installiert, aber nicht aktiv.

Gerade umgekehrt sieht eine Minimalinstallation für den Server-Betrieb bzw. in einer virtuellen Maschine aus: Dort sind standardmäßig weder Chrony noch `ntp/ntpdate` installiert. Wenn Sie NTP nutzen möchten, ist es Ihnen freigestellt, ob Sie die traditionellen NTP-Pakete oder das modernere Chrony bevorzugen. Die folgenden Kommandos zeigen die Installation und Inbetriebnahme von Chrony:

```
root# yum install chrony
root# systemctl start chrony
root# systemctl enable chrony
```

Weitere Informationen zu Chrony können Sie hier nachlesen:

<https://chrony.tuxfamily.org>

### NTP-Tools (SUSE, Raspbian)

**SUSE** SUSE-Distributionen greifen standardmäßig auf die klassischen NTP-Tools zurück. Die Konfiguration erfolgt im YaST-Modul `SYSTEM • DATUM UND ZEIT`; bei einer Konfigurationsänderung wird einmalig `ntpdate` ausgeführt. Um selbst einen NTP-Server einzurichten, starten Sie das YaST-Modul `NETZWERKDIENTE • NTP-EINRICHTUNG` und aktivieren die Option `STARTE NTP-DIENST JETZT UND BEIM BOOTEN`. Entscheidend ist,



dass Sie der Konfiguration einen oder mehrere NTP-Server hinzufügen. Standardmäßig wird nur die lokale Uhr verwendet, was unzureichend ist. `ntpd` wird durch `systemd` gestartet.

Mit `ntpq -p` überzeugen Sie sich davon, dass `ntpd` funktioniert. Entscheidend ist die `offset`-Spalte der Ausgabe dieses Kommandos: Sie gibt die Differenz zwischen der lokalen Uhr und der Uhr verschiedener Referenz-Server in Millisekunden an. Die Differenz sollte möglichst klein sein. Damit `ntpq -p` brauchbare Ergebnisse liefert, muss `ntpd` einige Zeit laufen (zumindest mehrere Minuten lang). Beachten Sie, dass `ntpd` die Zeit bei kleinen Abweichungen nicht einfach korrigiert, sondern die Uhr eine Weile etwas schneller oder etwas langsamer laufen lässt, bis die korrekte Zeit erreicht wird. Dadurch werden abrupte Zeitänderungen vermieden.

```
root# ntpq -p
      remote           refid      st t when poll reach  delay  offset  jitter
=====
+83.68.137.76    131.130.251.107  2 u  789 1024  377  11.468   9.437 102.141
*ntp.kurios.at  131.130.251.107  2 u   69 1024  177  11.125   9.991  25.342
+ntp1.wtnet.de  10.129.40.211   2 u  823 1024  377  33.246   9.874 102.921
+time.reisenbaue 17.253.52.125   2 u  925 1024  377  16.150  10.011  82.466
```

Sollte die Zeitabweichung größer als eine Sekunde sein (das entspricht einem Wert größer 1000 in der `Offset`-Spalte), muss die Uhrzeit manuell mit `ntpdate` eingestellt werden:

```
root# systemctl stop ntpd
root# ntpdate de.pool.ntp.org
root# systemctl start ntpd
```

Auch ältere Debian-Versionen und Raspbian Jessie verwenden die NTP-Tools. Auf dem Raspberry Pi ist NTP besonders wichtig, weil der Minirechner über keine interne Uhr verfügt. Der Start von `ntpd` erfolgt zwar durch `systemd`, es wird aber ein altes `Init-V-Script` verwendet (`/etc/init.d/ntp`).

Raspbian

### Einen eigenen NTP-Server betreiben

Ergänzend zu einer NTP-Client-Konfiguration können Sie auf einem Linux-Rechner auch einen eigenen NTP-Server betreiben. Angesichts der guten Erreichbarkeit vieler im Internet verfügbaren öffentlichen NTP-Server besteht dazu selten eine Notwendigkeit. Ein eigener NTP-Server kann aber z. B. in einem größeren lokalen Netz einer Firma zweckmäßig sein, wenn es einen zentralen Login-Dienst wie Kerberos gibt, dessen Authentifizierungsmechanismen voraussetzen, dass die Zeit aller Clients exakt mit der des Kerberos-Servers übereinstimmt.

Trotz der inzwischen großen Auswahl an NTP-Clients ist das gängigste Programm zur Implementierung eines NTP-Servers unverändert `ntpd`. Bei der Konfiguration

ist darauf zu achten, dass nur Clients aus dem lokalen Netzwerk auf den eigenen NTP-Server zugreifen können. Die Konfiguration führen Sie in `/etc/ntp.conf` durch. Detaillierte Informationen finden Sie hier:

<http://support.ntp.org/bin/view/Support/AccessRestrictions>

Achten Sie auch darauf, dass der Port 123 (TCP und UDP) auf dem NTP-Server nicht blockiert ist!

## 18.5 Benutzer und Gruppen, Passwörter

Zugriffsrechte steuern unter Linux, wer auf welche Dateien zugreifen darf, wer welche Programme ausführen darf und wer auf welche Hardware-Komponenten bzw. Device-Dateien zugreifen darf (siehe auch [Abschnitt 11.5](#), »Zugriffsrechte, Benutzer und Gruppenzugehörigkeit«). Diesen Zugriffsrechten liegt die Benutzerverwaltung zugrunde: Standardmäßig werden bei der Installation von Linux diverse Benutzer eingerichtet. Jeder Benutzer ist zumindest einer, möglicherweise aber auch mehreren Gruppen zugeordnet. Gruppen dienen dazu, mehreren Benutzern den Zugriff auf gemeinsame Dateien bzw. Programme zu ermöglichen.

Naturgemäß bietet die Benutzerverwaltung die Möglichkeit, auf einem Rechner mehrere Benutzer einzurichten, die sich dann einloggen und isoliert voneinander arbeiten können. Die Benutzerverwaltung ist aber auch von Bedeutung, wenn Sie auf Ihrem Linux-Rechner alleine arbeiten: Aus Sicherheitsgründen werden viele Systemdienste nämlich nicht mit `root`-Rechten, sondern ebenfalls in speziellen System-Accounts ohne Passwort ausgeführt. Diese Accounts sind also für einen gewöhnlichen Login gesperrt und nur für die Ausführung von Programmen gedacht.

Dieser Abschnitt beschreibt, welche Werkzeuge Ihnen dabei helfen, Benutzer und Gruppen einzurichten, und welche Konfigurationsdateien hinter den Kulissen für die Zuordnung von Benutzern, Gruppen und Passwörtern verantwortlich sind.

### Konfigurationsprogramme

Prinzipiell können Sie als `root` die Benutzerverwaltung weitgehend manuell durchführen, indem Sie die in diesem Abschnitt beschriebenen Dateien direkt ändern. Komfortabler und sicherer ist es, die mit den meisten Distributionen mitgelieferten Werkzeuge zur Benutzer- und Gruppenverwaltung einzusetzen. Beachten Sie, dass bei Red Hat, Fedora & Co. das erwähnte Programm `system-config-users` zuerst installiert werden muss.

|                  |  |
|------------------|--|
| Gnome:           | Systemeinstellungsmodul BENUTZER   |
| KDE:             | Systemeinstellungsmodul BENUTZERVERWALTUNG                               |
| Debian, Ubuntu:  | Gnome- oder KDE-Werkzeuge  |
| Red Hat, Fedora: | <code>system-config-users</code> (siehe <a href="#">Abbildung 18.1</a> ) |
| SUSE:            | YaST-Modul SICHERHEIT • BENUTZER UND GRUPPEN                             |

| Benutzername      | Benutzer-ID | Bevorzugte Gruppe | Vollständiger Name        | Anmelde-Shell    | Persönliches Verzeichnis |
|-------------------|-------------|-------------------|---------------------------|------------------|--------------------------|
| openvpn           | 991         | openvpn           | OpenVPN                   | /sbin/nologin    | /etc/openvpn             |
| saslauth          | 992         | saslauth          | Saslauthd user            | /sbin/nologin    | /run/saslauthd           |
| colord            | 993         | colord            | User for colord           | /sbin/nologin    | /var/lib/colord          |
| geoclue           | 994         | geoclue           | User for geoclue          | /sbin/nologin    | /var/lib/geoclue         |
| polkitd           | 995         | polkitd           | User for polkitd          | /sbin/nologin    | /                        |
| systemd-bus-proxy | 996         | systemd-bus-proxy | systemd Bus Proxy         | /sbin/nologin    | /                        |
| systemd-resolve   | 997         | systemd-resolve   | systemd Resolver          | /sbin/nologin    | /                        |
| systemd-network   | 998         | systemd-network   | systemd Network Manager   | /sbin/nologin    | /                        |
| systemd-timesync  | 999         | systemd-timesync  | systemd Time Synchronizat | /sbin/nologin    | /                        |
| <b>kofler</b>     | <b>1000</b> | <b>kofler</b>     | <b>Michael Kofler</b>     | <b>/bin/bash</b> | <b>/home/kofler</b>      |
| nfsnobody         | 65534       | nfsnobody         | Anonymous NFS User        | /sbin/nologin    | /var/lib/nfs             |

Abbildung 18.1 Benutzerverwaltung unter Fedora

Wenn Sie auf komfortable Benutzeroberflächen verzichten können oder die Benutzerverwaltung in Scripts automatisieren möchten, können Sie auf die Kommandos zurückgreifen, die in [Tabelle 18.3](#) zusammengefasst sind. Das folgende Beispiel zeigt, wie Sie den neuen Benutzer `testuser` anlegen und ihm ein Passwort zuweisen:

Kommandos

```
root# useradd -m testuser
root# passwd testuser
New passwd: xxxxxxxx
Re-enter new passwd: xxxxxxxx
```

Es wird Ihnen sicherlich auffallen, dass es für manche Aufgaben gleich zwei Kommandos gibt, z. B. `adduser` und `useradd`. Bei `adduser`, `addgroup`, `deluser` und `delgroup` handelt es sich um Debian-spezifische Erweiterungen zu den herkömmlichen Kommandos `useradd`, `groupadd` etc. Unter Debian und Ubuntu berücksichtigen diese Kommandos die in `/etc/adduser.conf` und `/etc/deluser.conf` definierten Regeln.

Für Konfusion sorgen Red Hat und Fedora: Dort stehen die Kommandos `adduser`, `addgroup`, `deluser` und `delgroup` ebenfalls zur Verfügung. Allerdings handelt es sich dabei nicht um die von Debian vertrauten Kommandos, sondern um Links auf `useradd`, `groupadd`, `userdel` und `groupdel`. Aus diesem Grund hat `adduser` unter Fedora dieselbe Syntax wie `useradd`, aber eine andere Syntax als `adduser` unter Debian!

| Kommando | Funktion   |
|----------|--|
| adduser  | richtet einen neuen Benutzer ein (Debian).             |
| addgroup | richtet eine neue Gruppe ein (Debian).                 |
| chage    | steuert, wie lange ein Passwort gültig bleibt.         |
| chgrp    | ändert die Gruppenzugehörigkeit einer Datei.           |
| chmod    | ändert die Zugriffsbits einer Datei.                   |
| chown    | ändert den Besitzer einer Datei.                       |
| chsh     | verändert die Standard-Shell eines Benutzers.          |
| delgroup | löscht eine Gruppe (Debian).                           |
| deluser  | löscht einen Benutzer (Debian).                        |
| groupadd | richtet eine neue Gruppe ein.                          |
| groupdel | löscht eine Gruppe.                                    |
| groupmod | verändert Gruppeneigenschaften.                        |
| groups   | zeigt die Gruppen des aktuellen Benutzers an.          |
| id       | zeigt die aktuelle Benutzer- und Gruppen-ID-Nummer an. |
| newgrp   | ändert die aktive Gruppe eines Benutzers.              |
| newusers | richtet mehrere neue Benutzer ein.                     |
| passwd   | verändert das Passwort eines Benutzers.                |
| useradd  | richtet einen neuen Benutzer ein.                      |
| userdel  | löscht einen Benutzer.                                 |
| usermod  | verändert Benutzereigenschaften.                       |

**Tabelle 18.3** Kommandos zur Benutzer- und Gruppenverwaltung

## Benutzerverwaltung

Unter Linux gibt es drei Typen von Benutzern:

- ▶ **Super-User alias Systemadministrator alias root:** Dieser Benutzer hat üblicherweise den Namen root. Wer das root-Passwort kennt und sich als root anmeldet, hat uneingeschränkte Rechte: Er oder sie darf alle Dateien ansehen, verändern, löschen, alle Programme ausführen etc. Derart viele Rechte sind nur zur Systemadministration erforderlich. Alle anderen Aufgaben sollten Sie aus Sicherheitsgründen nicht als root ausführen!
- ▶ **Gewöhnliche Benutzer:** Diese Benutzer verwenden Linux, um damit zu arbeiten. Sie haben vollen Zugriff auf ihre eigenen Dateien, aber nur eingeschränkten Zugriff auf den Rest des Systems. Als Login-Name wird oft der Vor- oder Nachname des Anwenders verwendet z. B. kathrin oder hofer.

- **Systembenutzer für Dämonen und Server-Dienste:** Schließlich gibt es eine Reihe von Benutzern, die nicht für die interaktive Arbeit am Computer vorgesehen sind, sondern zur Ausführung bestimmter Programme dienen. Beispielsweise wird der Webserver Apache nicht vom Benutzer `root` ausgeführt, sondern von einem eigenen Benutzer, der je nach Distribution `apache` oder `wwwrun` oder `httpd` oder so ähnlich heißt. Diese Vorgehensweise wird gewählt, um eine möglichst hohe Systemsicherheit zu erzielen.

Die Liste aller Benutzer wird in der Datei `/etc/passwd` gespeichert. Dort werden für jeden Benutzer der Login-Name, der vollständige Name, die UID- und GID-Nummer, das Heimatverzeichnis und die Shell gespeichert. Dabei gilt folgendes Format: `/etc/passwd`

*Login:Passwort:UID:GID:Name:Heimatverzeichnis:Shell*

Die folgenden Zeilen zeigen einige Benutzerdefinitionen in `/etc/passwd` unter Ubuntu Linux:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
...
kofler:x:1000:1000:Michael Kofler,,,:/home/kofler:/bin/bash
huber:x:1001:1001:Herbert Huber,,,:/home/huber:/bin/bash
```

Der Name `passwd` lässt vermuten, dass in der Datei auch die Passwörter gespeichert werden. Das war früher tatsächlich der Fall. Heute enthält `/etc/passwd` anstelle der Passwortinformationen nur das Zeichen `x`. Der verschlüsselte Passwort-Hash wird in der separaten Datei `/etc/shadow` gespeichert, die ich Ihnen gleich vorstellen werde (siehe den Abschnitt »Passwörter«).

Der Login-Name sollte nur aus Kleinbuchstaben (US-ASCII-Buchstaben und Zahlen) bestehen und nicht länger als acht Zeichen sein. Zwar sind sowohl Nicht-ASCII-Zeichen als auch mehr als acht Zeichen prinzipiell zulässig, es kann aber passieren, dass Probleme in Kombination mit manchen Programmen auftreten. Für den getrennt gespeicherten vollständigen Namen gelten diese Einschränkungen nicht. **Login-Name**

Die UID-Nummer (*User Identification*) dient zur internen Identifizierung des Benutzers. Die Nummer wird insbesondere als Zusatzinformation zu jeder Datei gespeichert, sodass klar ist, wem die Datei gehört. **UID und GID**

Für die Vergabe von UID-Nummern gibt es Regeln: `root` hat immer `UID=0`. Für Server-Dienste und Dämonen sind bei den meisten Distributionen UID-Nummern zwischen 1 und 999 vorgesehen. Für gewöhnliche Benutzer sind dementsprechend Nummern ab 1000 vorgesehen.

Die GID-Nummer (*Group Identification*) gibt an, zu welcher Gruppe der Anwender gehört. Mehr Details zu Gruppen folgen im nächsten Abschnitt.

#### Heimatverzeichnis

Das Heimatverzeichnis ist der Ort, an dem der Benutzer seine privaten Daten speichern kann. Bei gewöhnlichen Benutzern wird dazu üblicherweise der Pfad `/home/<loginname>` verwendet. Im Heimatverzeichnis werden auch die persönlichen Konfigurationseinstellungen des Benutzers für diverse Programme gespeichert. Beispielsweise enthält die Datei `.emacs` die Konfigurationseinstellungen für den Editor Emacs. Da die Namen derartiger Konfigurationsdateien meistens mit einem Punkt beginnen, sind sie unsichtbar. Sie können mit dem Kommando `ls -la` angezeigt werden.

Damit bei neuen Benutzern sofort sinnvolle Standardeinstellungen für die wichtigsten Programme vorliegen, sollten beim Anlegen eines neuen Benutzers alle Dateien aus `/etc/skel` in das neu erzeugte Heimatverzeichnis kopiert werden. Viele Programme zum Anlegen neuer Benutzer erledigen diesen Schritt automatisch. Der Inhalt von `/etc/skel` stellt damit die Ausgangseinstellung für jeden neuen Benutzer dar.

**Shell** Die Shell ist ein Interpreter, mit dem der Benutzer nach dem Login Kommandos ausführen kann. Da unter Linux mehrere Shells zur Auswahl stehen, muss in der `passwd`-Datei angegeben werden, welche Shell zum Einsatz kommen soll. Unter Linux ist dies meistens die Shell `bash`. In der `passwd`-Datei muss der vollständige Dateiname der Shell gespeichert werden, also beispielsweise `/bin/bash`.

## Gruppenverwaltung

Der Sinn von Gruppen besteht darin, mehreren Benutzern den gemeinsamen Zugriff auf Dateien zu ermöglichen. Dazu wird jeder Benutzer einer primären Gruppe (*Initial Group*) zugeordnet. Außerdem kann ein Benutzer beliebig vielen weiteren Gruppen (*Supplementary Groups*) zugeordnet werden, also Mitglied mehrerer Gruppen sein.

Die Datei `/etc/group` enthält die Liste aller Gruppen. Die folgenden Zeilen zeigen einige Gruppeneinstellungen in `/etc/group`. Es gilt folgendes Format:

*Gruppenname:Passwort:GID:Benutzerliste*

Die folgenden Zeilen stammen aus der `group`-Datei eines Ubuntu-Systems:

```
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,kofler
cdrom:x:24:kofler
...
kofler:x:1000:
huber:x:1001:
...
```

Die Zuordnung zwischen Benutzer und Gruppe erfolgt auf zwei Weisen:

- ▶ Die primäre Gruppe eines Benutzers wird in `/etc/passwd` gespeichert. Beim Benutzer `kofler` lautet die primäre Gruppe ebenfalls `kofler`; das geht aus dem Wert `1000` in der GID-Spalte in der `/etc/passwd`-Datei hervor.
- ▶ Die Zugehörigkeit zu weiteren Gruppen wird gespeichert, indem der Name des Benutzers in der letzten Spalte der Datei `/etc/group` angegeben wird. So gehört `kofler` beispielsweise auch zu den Gruppen `adm` und `cdrom`. Das erlaubt dem Benutzer `kofler`, auf Ubuntu-Systemen Administrationsarbeiten durchzuführen und auf CDs/DVDs zuzugreifen.

Bei den GID-Nummern ist `0` für `root` vorgesehen, `1` bis `99` für Systemdienste. `GID=100` ist normalerweise für die Gruppe `users` reserviert. GIDs größer `1000` werden dann für benutzerspezifische Gruppen verwendet bzw. dürfen für eigene Zwecke definiert werden.

Für die Zuordnung zwischen Benutzern und ihren primären Gruppen gibt es zwei gängige Strategien: Primäre Gruppe

- ▶ Beim herkömmlichen Verfahren, das seit vielen Jahren unter Unix/Linux zum Einsatz kommt, sind alle gewöhnlichen Benutzer der primären Gruppe `users` zugeordnet. SUSE ist bis heute ein Anhänger dieser sehr einfachen Strategie.
- ▶ Debian-basierte Distributionen sowie Red Hat und Fedora setzen auf ein anderes Verfahren: Jeder Benutzer bekommt seine eigene primäre Gruppe. In diesem Fall gibt es für die Benutzer `kofler` und `huber` jeweils eine gleichnamige Gruppe. Die Gruppe `users` spielt keine Rolle mehr.

Das Verfahren hat unter bestimmten Umständen Vorteile – etwa dann, wenn mehrere Mitglieder einer sekundären Gruppe gemeinsame Dateien erzeugen. Diese Vorteile kommen aber nur bei einer entsprechenden Systemadministration zum Tragen. Im Detail sind die Unterschiede auf der folgenden Debian-Wiki-Seite erläutert:

<https://wiki.debian.org/UserPrivateGroups>

## Passwörter

Linux-Passwörter bestehen üblicherweise nur aus ASCII-Zeichen. Internationale Sonderzeichen sind zwar grundsätzlich erlaubt, können aber leicht zu Problemen führen, z. B. bei der Eingabe auf einer noch nicht korrekt konfigurierten Tastatur. Aus Sicherheitsgründen sollten Passwörter sowohl Groß- als auch Kleinbuchstaben sowie mindestens eine Ziffer enthalten.

Passwörter werden unter Linux in Form sogenannter Hash-Codes gespeichert, die eine Kontrolle, aber keine Rekonstruktion von Passwörtern ermöglichen. Die Passwörter dürfen beliebig lang sein. Aktuelle Distributionen verwenden den als sicher geltenden Hash-Algorithmus SHA512 in Kombination mit einem zufälligen Initialisierungscode für das Passwort, dem sogenannten Salt. Dieses »Salz« bewirkt, dass für ein- und dasselbe Passwort jedes Mal ein anderer Hash-Code gespeichert wird. Damit ist in `/etc/shadow` nicht erkennbar, ob mehrere Benutzer dasselbe Passwort haben. Der Hash-Algorithmus wird durch die Variable `ENCRYPT_METHOD` in `/etc/login.defs` festgelegt.

Um potenziellen Angreifern das Leben zu erschweren, werden die verschlüsselten Passwortcodes nicht direkt in `/etc/passwd` gespeichert, sondern in der getrennten Datei `/etc/shadow`. Der Vorteil besteht darin, dass diese Datei nur von `root` gelesen werden kann. `/etc/passwd` und `/etc/group` sind hingegen für alle Benutzer des Systems lesbar, weil sie elementare Verwaltungsinformationen enthalten. Bei `/etc/shadow` reicht es dagegen aus, wenn nur die Programme zur Passwortverifizierung und -änderung darauf zugreifen dürfen. Ein potenzieller Angreifer muss daher zuerst `root`-Zugang erhalten, bevor er auch nur die verschlüsselten Passwort-Codes lesen kann.

Für die `shadow`-Datei gilt das folgende Format:

*Login:Passwort-Code:d1:d2:d3:d4:d5:d6:reserved*

Die folgenden Zeilen zeigen einen Ausschnitt aus einer `shadow`-Datei:

```
root:$6$Ecrkix...:14391:0:99999:7:::
daemon*:14391:0:99999:7:::
bin*:14391:0:99999:7:::
...
kofler:$6$TZR7...:14391:0:99999:7:::
```

Bei den meisten Systembenutzern wie `daemon` oder `bin` wird statt eines Passworts nur ein Stern oder ein Ausrufezeichen gespeichert. Das bedeutet, dass es kein gültiges Passwort gibt, ein Login also unmöglich ist. Die System-Accounts können dennoch verwendet werden: Programme, die zuerst mit `root`-Rechten gestartet werden, können später ihren Besitzer wechseln und dann als `bin`, `daemon`, `lp` etc. fortgesetzt werden. Genau das ist bei den meisten Systemprozessen der Fall: Sie werden während des Systemstarts von `root` gestartet und wechseln dann aus Sicherheitsgründen sofort den Besitzer.

**d1 bis d6** Die Felder `d1` bis `d6` können optionale Zeitangaben enthalten:

- ▶ `d1` gibt an, wann das Passwort zum letzten Mal geändert wurde. (Die Angabe erfolgt in Tagen, die seit dem 1.1.1970 vergangen sind.)
- ▶ `d2` gibt an, in wie vielen Tagen das Passwort geändert werden darf.



- ▶ `d3` gibt an, in wie vielen Tagen das Passwort spätestens geändert werden muss, bevor es ungültig wird. (Details zu den Feldern erhalten Sie mit `man 5 shadow`.)
- ▶ `d4` gibt an, wie viele Tage vor dem Ablauf des Passworts der Benutzer gewarnt wird.
- ▶ `d5` gibt an, nach wie vielen Tagen ein abgelaufener Account ohne gültiges Passwort vollständig deaktiviert wird.
- ▶ `d6` gibt an, seit wann ein Account deaktiviert ist.

Normalerweise werden für `d1` bis `d3` Standardwerte verwendet, sodass das Passwort jederzeit geändert werden kann und unbeschränkt gültig bleibt. `d1` bis `d6` können aber auch dazu verwendet werden, die Gültigkeit von Passwörtern zu beschränken, Login-Accounts zeitlich automatisch zu deaktivieren etc., etwa zur Verwaltung von Studenten-Accounts an einer Universität oder Schule.

Die Einstellung der Felder `d1` bis `d6` erfolgt durch das Kommando `chage` (*change age*). Im folgenden Beispiel wird der Benutzer gezwungen, sein Passwort sofort nach dem ersten Login zu ändern. Außerdem muss er in Zukunft sein Passwort alle 100 Tage ändern. Schließlich steht das Konto nur bis zum 31.12.2018 zur Verfügung. Danach wird jeder Login blockiert:

```
root# chage -d 0 -M 100 -E 2018-12-31 loginname
```

Eine Menge Parameter zur internen Administration von Passwörtern und Logins befinden sich in der Datei `/etc/login.defs`. Die Einstellungen gelten für `useradd`, `groupmod` etc. In `logins.def` ist beispielsweise festgeschrieben, wie viele Tage Passwörter standardmäßig gelten, welcher Wertebereich für neue UIDs und GIDs verwendet wird etc.

Die PAM-Konfiguration (siehe [Abschnitt 18.6](#)) definiert darüber hinaus Regeln für die Passwortüberprüfung. Die PAM-Regeln werden unter anderem vom Kommando `passwd` sowie bei jedem Login berücksichtigt.

Um Ihr eigenes Passwort zu verändern, führen Sie das Kommando `passwd` aus. Sie werden jetzt zuerst nach Ihrem alten Passwort gefragt und dann zweimal hintereinander aufgefordert, ein neues Passwort einzutippen. Nur wenn beide Eingaben übereinstimmen, wird das neue Passwort akzeptiert. Ab jetzt müssen Sie bei jedem Einloggen das neue Passwort verwenden.

Passwörter  
ändern

Während normale Benutzer nur ihr eigenes Passwort ändern können, darf `root` auch die Passwörter fremder Anwender verändern:

```
root# passwd hofer
New password: *****
Re-enter new password: *****
Password changed.
```

**Passwortqualität** Bei den meisten Distributionen stellen PAM-Erweiterungen eine minimale Passwortqualität sicher. Oft müssen Passwörter zumindest 8 oder 9 Zeichen lang sein, sich von Wörterbucheinträgen unterscheiden, Zahlen enthalten etc. Wie diese Qualitätskontrolle erfolgt, ist distributionsabhängig:

- ▶ Bei Fedora sowie RHEL und CentOS ab Version 7 stellt `pam_pwquality` die Passwortqualität sicher. Die Konfiguration erfolgt durch `/etc/security/pwquality.conf`. Eine gute Einführung in die Funktionsweise finden Sie hier:

*[http://deer-run.com/~hal/linux\\_passwords\\_pam.html](http://deer-run.com/~hal/linux_passwords_pam.html)*

- ▶ SUSE verwendet stattdessen die Bibliothek `pam_cracklib`. Wie Sie von den Defaulteinstellungen abweichende Regeln definieren, ist hier ausführlich dokumentiert:

*[http://deer-run.com/~hal/sysadmin/pam\\_cracklib.html](http://deer-run.com/~hal/sysadmin/pam_cracklib.html)*

- ▶ Debian und Ubuntu greifen auf die in das PAM-Modul `pam_unix` integrierten Passworttests zurück. Details zur Funktionsweise können Sie mit `man pam_unix` nachlesen, wobei insbesondere die Beschreibung des Parameters `obscure` relevant ist. Dieser Parameter ist nämlich sowohl in Debian als auch in Ubuntu aktiv (Datei `/etc/pam.d/common-password`). Gegebenenfalls können Sie die minimale Passwortlänge durch den weiteren Parameter `minlen` beeinflussen.

Die Passwortregeln gelten nur für gewöhnliche Benutzer. `root` darf mit dem `passwd`-Kommando sowohl für sich selbst als auch für jeden anderen Benutzer beliebig schlechte Passwörter wählen, obwohl dies natürlich nicht empfehlenswert ist.

### Sichere Passwörter generieren

Wenn Sie häufig Benutzerkonten einrichten, ist es zweckmäßig, automatisch generierte Passwörter zu verwenden. Dabei helfen die Programme `mypasswd` aus dem Fedora- bzw. RHEL-Paket `expect` bzw. `makepasswd` aus dem gleichnamigen Debian- bzw. Ubuntu-Paket:

```
root# makepasswd
aGjoQK1Ezo
```

**root-Passwort vergessen** Was tun Sie, wenn Sie Ihr `root`-Passwort vergessen haben? Bei den meisten Distributionen benötigen Sie ein Live- oder Rescue-System, das Sie von einem entsprechenden USB-Stick oder einer CD/DVD starten können. Anschließend binden Sie die Systempartition `/dev/xxx` Ihres Linux-Systems in ein beliebiges Verzeichnis ein. Mit `chroot` machen Sie dieses Verzeichnis zum neuen Root-Verzeichnis. Nun können Sie mit `passwd` das `root`-Passwort neu einstellen:

```
root# mkdir /rescue
root# mount -t auto /dev/xxx /rescue
```

```

root# chroot /rescue
root# passwd
root# <Strg>+<D>
root# reboot

```

Wenn Sie andere Benutzer daran hindern möchten, auf die gerade beschriebene Weise das root-Passwort zu verändern, müssen Sie im BIOS/EFI Ihres Rechners alle Boot-Medien außer der ersten Festplatte deaktivieren und das BIOS/EFI selbst durch ein Passwort absichern. Dieses Passwort sollten Sie dann aber wirklich nicht vergessen! Einen perfekten Schutz bietet dieses Verfahren aber nicht – ein Angreifer könnte ganz einfach die Festplatte/SSD Ihres Rechners ausbauen und in einem eigenen Rechner ansprechen. Dort funktioniert dann wiederum die oben beschriebene Vorgehensweise.

Eine wesentlich effektivere Möglichkeit zur Absicherung eines Linux-Systems besteht darin, die ganze Systempartition zu verschlüsseln. Das entsprechende Passwort muss dann bei jedem Boot-Vorgang eingegeben werden, weswegen diese Vorgehensweise vor allem bei Desktop-Installationen zweckmäßig ist. Für den Server-Betrieb ist eine Verschlüsselung nur dann sinnvoll, wenn der Server für Sie als Administrator leicht zugänglich ist! Denn auch in diesem Fall müssen Sie das Passwort bei jedem Neustart angeben. Bei Servern, die in einem externen Rechenzentrum stehen, ist dies unmöglich.

Bei vielen Distributionen werden fehlerhafte Login-Versuche in `/var/log/faillog` protokolliert. Im Gegensatz zu vielen anderen Logging-Dateien kommt dabei ein binäres Format zum Einsatz. Die Konfiguration der `faillog`-Datei erfolgt in `/etc/login.defs`. faillog

Mit `faillog -u name` stellen Sie fest, wie viele fehlerhafte Login-Versuche bei einem bestimmten Benutzer seit der letzten gültigen Anmeldung aufgetreten sind.

Mit `faillog -u name -m max` kann eine Maximalanzahl fehlerhafter Login-Versuche für einen bestimmten Benutzer fixiert werden. Wird diese Zahl überschritten, wird der Login blockiert, bis root den Login durch das Kommando `faillog -u name -r` wieder erlaubt. (Damit wird der Login-Zähler zurückgesetzt.)

Sie können die maximale Login-Anzahl durch `faillog -m max` auch generell festlegen. Allerdings sollten Sie dann immer auch `faillog -u root -m 0` ausführen, damit root von dieser Schutzmaßnahme ausgeschlossen ist. Andernfalls könnte es passieren, dass Sie sich selbst als root nicht mehr einloggen können, nachdem ein anderer Benutzer mehrere vergebliche root-Login-Versuche durchgeführt hat.

Unter Fedora, RHEL und CentOS steht `faillog` nicht zur Verfügung. Eine vergleichbare Funktion bieten die PAM-Erweiterung `pam_faillock` und das dazugehörige Kommando `faillock`. Diese Komponenten sind standardmäßig installiert, aber nicht aktiv. faillock

Ein Beispiel für eine entsprechende Konfiguration können Sie auf der folgenden Seite nachlesen:

<https://unix.stackexchange.com/questions/182091>

#### Gruppenpasswörter

Wie bei Benutzern können auch bei Gruppen Passwörter definiert werden (Kommando `gpasswd`). Aber während bei Benutzern Passwörter unbedingt zu empfehlen sind, sind Gruppenpasswörter unüblich. Ihr Hauptnachteil besteht darin, dass alle Gruppenmitglieder das Passwort kennen müssen, was die Administration erschwert und die Anwendung inhärent unsicher macht.

Falls tatsächlich Gruppenpasswörter zum Einsatz kommen sollen, werden diese in der Datei `/etc/gshadow` gespeichert. Ein Gruppenpasswort muss dann eingegeben werden, wenn ein Benutzer mit dem Kommando `newgrp` seine gerade aktive Gruppe wechselt.

### Zusammenspiel der Konfigurationsdateien

Die folgenden Zeilen fassen nochmals zusammen, wie die drei Dateien `passwd`, `group` und `shadow` zusammenspielen. Für jeden Anwender enthält `passwd` eine Zeile nach folgendem Muster:

```
# eine Zeile in /etc/passwd
kofler:x:1000:1000:Michael Kofler:/home/kofler:/bin/bash
```

Dabei ist `kofler` der Login-Name. `1000` ist gleichermaßen UID und GID, `Michael Kofler` der vollständige Name, `/home/kofler` sein Benutzerverzeichnis und `/bin/bash` seine Shell. Die UID muss eine eindeutige Nummer sein, die für die Verwaltung der Zugriffsrechte von Dateien wichtig ist.

Die dazugehörige Zeile in `/etc/shadow` mit den Passwortinformationen sieht so aus:

```
# eine Zeile in /etc/shadow
kofler:$6$9dk0$. . . :13479:0:99999:7:::
```

Die Zeichenkette nach `kofler:` ist das verschlüsselte Passwort. Wenn auf die Zeichenkette verzichtet wird, kann der Login ohne Passwort verwendet werden. Wenn statt der Zeichenkette ein `*` oder `!` eingetragen ist, ist der Login gesperrt.

Die GID-Nummer in `/etc/passwd` muss mit einer Gruppe aus `/etc/group` übereinstimmen. Bei vielen Distributionen ist jedem gewöhnlichen Benutzer eine gleichnamige Gruppe zugeordnet:

```
# eine Zeile aus /etc/group
kofler:x:1000:
```

## Benutzerverwaltung im Netzwerk

Wenn Sie mehrere Linux-Rechner miteinander vernetzen und mit NFS einen gegenseitigen Zugriff auf Dateien ermöglichen möchten, dann müssen Sie darauf achten, dass die UID- und GID-Nummern auf allen Rechnern einheitlich sind. Das allein wird bei mehreren Rechnern schon recht aufwendig. Wenn Sie nun auch noch möchten, dass sich jeder Benutzer auf jedem Rechner einloggen kann, und zwar natürlich immer unter dem gleichen Login-Namen und mit dem gleichen Passwort, dann müssen Sie alle `/etc/passwd`-Dateien ständig synchronisieren. Der Administrationsaufwand ist dann riesig.

Um diesen Aufwand zu vermeiden, wird in solchen Fällen meist ein zentraler Server zur Benutzerverwaltung eingesetzt. Zur Authentifizierung der Clients stehen eine ganze Menge alternativer Verfahren bzw. Protokolle zur Auswahl, deren Beschreibung in diesem Buch aber aus Platzgründen leider nicht möglich ist:

- ▶ Samba bzw. die Windows-Benutzerverwaltung (Active Directories)
- ▶ LDAP (Lightweight Directory Access Protocol)
- ▶ Kerberos
- ▶ NIS (Network Information Service, veraltet)

## 18.6 PAM, NSS und nscd

Nachdem ich Ihnen im vorigen Abschnitt die Prinzipien der Benutzer- und Gruppenverwaltung präsentiert habe, folgen hier einige weitere technische Details. Hinter den drei Abkürzungen der Überschrift verstecken sich die folgenden Verfahren:

- ▶ **PAM:** Die *Pluggable Authentication Modules* sind eine zentrale Bibliothek, die diversen Kommandos Authentifizierungsfunktionen zur Verfügung stellt.
- ▶ **NSS:** Der *Name Service Switch* entscheidet, welche Quellen zur Auflösung von Benutzer-, Gruppen- und Hostnamen herangezogen werden.
- ▶ **nscd:** Der *Name Service Caching Daemon* ist ein Zwischenspeicher für Namensabfragen. Er kann die zuletzt benötigten Benutzer-, Gruppen- und Hostnamen besonders schnell neuerlich zur Verfügung stellen.

### PAM

Die Pluggable Authentication Modules (PAM) sind eine Bibliothek, deren Funktionen bei Authentifizierungsaufgaben helfen. Wenn Sie auf einem Linux-Rechner einen Login durchführen oder sich auf andere Weise authentifizieren bzw. Ihr Passwort verändern, greifen die jeweiligen Programme auf die PAM-Bibliothek zurück. Auch Cron

und PolicyKit nutzen PAM. Mit `ldd` können Sie feststellen, ob ein bestimmtes Kommando auf PAM-Bibliotheken zurückgreift:

```
root# ldd /usr/bin/passwd | grep libpam
libpam.so.0 => /lib64/libpam.so.0 (0x00007f607587c000)
libpam_misc.so.0 => /lib64/libpam_misc.so.0 (0x00007f6075677000)
```

Eine umfassende Dokumentation zu PAM finden Sie hier:

<http://linux-pam.org>

PAM ist standardmäßig so konfiguriert, dass es die lokalen Passwortdateien auswertet, also z. B. `/etc/shadow`. Wenn ergänzend auch ein anderes Authentifizierungsverfahren genutzt werden soll (z. B. LDAP), muss die PAM-Konfiguration entsprechend verändert werden. Dabei helfen je nach Distribution unterschiedliche Werkzeuge:

CentOS, Fedora, Red Hat: `authconfig`

SUSE: `YaST: SICHERHEIT • BENUTZER- UND GRUPPENVERWALTUNG`

Ubuntu: `pam-auth-update`

**pam.conf** Die Konfigurationsdateien befinden sich im Verzeichnis `/etc/pam.d/`. Darüber hinaus wird auch die Datei `/etc/pam.conf` ausgewertet. Die Konfigurationsdateien enthalten zeilenweise Regeln. Jede Regel besteht aus mindestens drei Teilen bzw. Spalten:

Typ Reaktion PAM-Modul [Modulargumente]

Einträgen in `pam.conf` muss zudem der Name des Service vorangestellt werden, z. B. `login` bzw. `other` für Standardeinträge. Bei den Dateien in `/etc/pam.d` ergibt sich der Service-Name aus dem Dateinamen.

**Regeltyp  
(erste Spalte)**

PAM unterscheidet zwischen vier Regeltypen. Bei Debian, SUSE und Ubuntu enthalten die Dateien `common-account`, `common-auth`, `common-password` und `common-session` Standardregeln für diese vier Typen:

- ▶ `account`: ermöglicht die Limitierung von Diensten je nach Tageszeit, Auslastung, Login-Ort (z. B. Konsole) etc.
- ▶ `auth`: betrifft die Autorisierung, also die Passwortabfrage und -überprüfung, sowie die anschließende Zuweisung von Privilegien, z. B. Gruppenzugehörigkeiten.
- ▶ `password`: betrifft den Mechanismus zur Änderung des Passworts.
- ▶ `session`: ermöglicht es, Aktionen vor oder nach dem eigentlichen Dienst auszuführen: Logging, Dateisysteme einbinden/lösen, Status der Mailbox anzeigen etc.

Wenn dem Regeltyp ein Minuszeichen vorangestellt ist, kommt es zu keiner Fehlermeldung, wenn ein in der Regel angegebenes Modul nicht verfügbar ist. Diese Syntaxvariante erlaubt die Definition von Regeln, die erst dann aktiv werden, wenn das entsprechende PAM-Modul tatsächlich installiert wird.

Die zweite Spalte in den Konfigurationsdateien gibt an, wie PAM reagieren soll, wenn eine Regel erfüllt bzw. nicht erfüllt ist. Es gibt zwei Möglichkeiten, die Reaktion zu beschreiben: entweder durch ein einfaches Schlüsselwort (z. B. `required`, `requisite`) oder durch ein in eckige Klammern gesetztes Wert/Ergebnis-Paar (z. B. `[success=1 new_authtok_reqd=done default=ignore]`). Die Bedeutung der vier wichtigsten Schlüsselwörter der einfachen Syntaxvariante ist in [Tabelle 18.4](#) erklärt.

Reaktion  
(zweite Spalte)

| Schlüsselwort           | Reaktion   |
|-------------------------|--|
| <code>requisite</code>  | Bei einem Regelverstoß liefert die PAM-Funktion sofort ein negatives Ergebnis, und die weiteren Regeln werden nicht mehr abgearbeitet.   |
| <code>required</code>   | Bei einem Regelverstoß liefert die PAM-Funktion ein negatives Ergebnis; weitere Regeln werden abgearbeitet, ihr Ergebnis wird aber nicht berücksichtigt.   |
| <code>sufficient</code> | Bei Einhaltung der Regel liefert PAM sofort ein positives Ergebnis (es sei denn, es liegt bereits ein Verstoß gegen eine vorangegangene <code>requisite</code> -Regel vor); weitere Regeln werden nicht mehr berücksichtigt. Bei einem Regelverstoß setzt PAM mit der nächsten Regel fort. |
| <code>optional</code>   | Das Ergebnis der Regel ist nur dann relevant, wenn es sich um die einzige Regel für einen bestimmten Regeltyp und einen bestimmten Service (z. B. <code>su</code> ) handelt.   |

**Tabelle 18.4** Reaktion auf PAM-Regelverstöße

Bei der zweiten Syntaxvariante geben Sie mehrere Wert/Ergebnis-Paare in der Form `[value1=result1 value2=result2 ...]` an. Für `value` gibt es eine ganze Reihe vordefinierter Schlüsselwörter, die das Ergebnis einer Regel ausdrücken. `result` kann entweder eine Zahl sein, die angibt, wie viele weitere Regeln nun übersprungen werden sollen, oder ein Schlüsselwort, das das gewünschte PAM-Ergebnis angibt (`ignore`, `bad`, `die`, `ok`, `done` oder `reset`). Das folgende Listing gibt an, wie die Schlüsselwörter der ersten Syntaxvariante in der zweiten Schreibweise ausgedrückt werden:

```
requisite = [success=ok new_authtok_reqd=ok ignore=ignore default=die]
required  = [success=ok new_authtok_reqd=ok ignore=ignore default=bad]
sufficient = [success=done new_authtok_reqd=done default=ignore]
optional  = [success=ok new_authtok_reqd=ok default=ignore]
```

Die dritte Spalte gibt den Namen des PAM-Moduls an, das die Regel auswertet. Das Verhalten des Moduls kann durch Optionen beeinflusst werden. Leider gibt es keine zentrale Dokumentation der zulässigen Optionen und ihrer Bedeutung.

Modul und  
Optionen (dritte  
Spalte)

Das folgende Listing fasst die Einstellungen von Fedora zusammen. Beachten Sie, dass die Standardeinstellungen je nach Distribution stark variieren und oft über mehrere Dateien verteilt sind, z. B. auf `common-xxx` bei Debian, SUSE und Ubuntu.

Standard-  
konfiguration

```

# Datei /etc/pam.d/password-auth (Fedora)
auth      required      pam_env.so
auth      sufficient    pam_unix.so nullok try_first_pass
auth      requisite    pam_succeed_if.so uid >= 1000 quiet_success
auth      required      pam_deny.so

account   required      pam_unix.so
account   sufficient    pam_localuser.so
account   sufficient    pam_succeed_if.so uid < 1000 quiet
account   required      pam_permit.so

password  requisite     pam_pwquality.so try_first_pass local_users_only \
          retry=3 authtok_type=
password  sufficient    pam_unix.so sha512 shadow nullok try_first_pass \
          use_authtok
password  required      pam_deny.so

session   optional      pam_keyinit.so revoke
session   required     pam_limits.so
-session  optional      pam_systemd.so
session   [success=1 default=ignore] \
          pam_succeed_if.so service in crond quiet use_uid
session   required     pam_unix.so

```

### PAM-Konfiguration bei Fedora und RHEL

Unter Fedora und RHEL sollten Sie es vermeiden, Dateien in `/etc/pam.d` direkt zu verändern. Besser ist es, die Konfiguration in `/etc/sysconfig/authconfig` durchzuführen. Das Kommando `authconfig` wertet diese Datei aus und überschreibt dann die betreffenden Dateien in `/etc/pam.d`. Leider bietet `authconfig` nur wenige Einstellmöglichkeiten und ist nicht besonders gut dokumentiert. Am ehesten hilft `authconfig --help` weiter.

### Name Service Switch (NSS)

Bei einem Login, bei der Verwaltung der Zugriffsrechte auf Dateien, bei Netzwerkzugriffen etc. sind alle möglichen Informationen über Benutzer- und Gruppennamen, UIDs und GIDs, Hostnamen, Ports von Netzwerkdiensten etc. erforderlich. Diese Daten befinden sich standardmäßig in den Dateien `/etc/passwd`, `/etc/group`, `/etc/hosts`, `/etc/services` etc.

In der Unix- bzw. Linux-Nomenklatur wird der Zugriff auf diese Daten unter dem Begriff *Name Services* zusammengefasst. Zuständig für diese Aufgabe ist der Name Service Switch (NSS), eine Sammlung von Funktionen der `libc`-Bibliothek. Ähnlich



wie bei der Authentifizierung lässt sich die Datenquelle für NSS einstellen, beispielsweise wenn ein LDAP-Server die Daten zur Verfügung stellt. Die entscheidende Konfigurationsdatei ist `/etc/nsswitch.conf`. Die folgenden Zeilen zeigen die Standardkonfiguration unter Ubuntu. Nahezu identisch sieht die Konfiguration von Debian aus.

```
# Datei /etc/nsswitch.conf (Ubuntu)
passwd:    compat
group:     compat
shadow:    compat
hosts:     files mdns4_minimal [NOTFOUND=return] resolve [!UNAVAIL=return] dns
networks:  files
protocols: db files
...
```

Die erste Spalte in `nsswitch.conf` bezeichnet die Datenbank bzw. Datei. Nach dem Doppelpunkt beschreiben Schlüsselwörter die Zugriffsmethode auf die Daten sowie optional in eckigen Klammern die Reaktion auf Nachschlageergebnisse. Die folgende Liste erklärt die wichtigsten Schlüsselwörter für die Zugriffsmethoden. Wenn eine Zeile mehrere Zugriffsmethoden nennt, werden diese der Reihe nach angewendet, bis eine Methode erfolgreich ist. Weitere Syntaxdetails verrät man `nsswitch.conf`.

- ▶ `files`: greift auf die traditionellen Konfigurationsdateien zurück (`/etc/passwd` und `/etc/group`).
- ▶ `compat`: wie `files`, wobei den Benutzer- und Gruppenangaben die Zeichen `+` und `-` vorangestellt werden können. Das erhöht die Kompatibilität zu NIS. `compat` kann nicht mit anderen Schlüsselwörtern kombiniert werden.
- ▶ `db`: liest die Daten aus einer BDB-Datenbankdatei (BDB = *Berkeley Database*).
- ▶ `dns`: kontaktiert einen Nameserver.
- ▶ `mdns`: verwendet Multicast DNS (alias Zeroconf bzw. Apple Bonjour/Rendezvous).
- ▶ `sss`: kontaktiert den System Security Services Daemon (`sssd`). Das ist ein lokaler Dienst, der als Zwischenspeicher für externe Authentifizierungsdienste wie Kerberos oder LDAP agiert.

Die Zugriffsmethoden setzen voraus, dass die entsprechende Bibliothek installiert ist, beispielsweise `libnss_db` für `db`. Fehlt die Bibliothek, wird das entsprechende Schlüsselwort einfach ignoriert, ohne dass ein Fehler gemeldet wird.

### nscd (Name Service Caching Daemon)

Bei SUSE wird standardmäßig das Programm `nscd` installiert und während des Hochfahrens des Rechners aktiviert. Bei den meisten anderen Distributionen ist eine optionale Installation möglich. SUSE

`nscd` steht für *Name Service Caching Daemon*. Das Programm merkt sich bei entsprechender Konfiguration Login-, Gruppen- und Hostnamen sowie deren IP-Nummern. Im Unterschied zu einem Nameserver stellt `nscd` diese Informationen aber nur dem lokalen Rechner zur Verfügung, nicht anderen Rechnern im Netzwerk. Sinnvoll ist der Einsatz von `nscd` vor allem dann, wenn die Benutzerverwaltung durch einen Netzwerkdienst erfolgt (z. B. LDAP). `nscd` dient dann als Cache für Informationen und vermeidet unnötige LDAP-Anfragen, die oft nur vergleichsweise langsam beantwortet werden.

`/etc/nscd.conf` Die Konfiguration von `nscd` erfolgt durch `/etc/nscd.conf`. Üblicherweise enthält die Datei Einträge in drei Gruppen: `passwd` für Login-Namen, `group` für Gruppen und `host` für Hostnamen. Die Einstellungen jeder Gruppe legen fest, wie lange die Daten gespeichert werden, wie viele Einträge maximal verwaltet werden sollen etc.

### sssd (System Security Services Daemon)

CentOS, Fedora, RHEL, Fedora und Red Hat bevorzugen anstelle von `nscd` den System Security Services Daemon. `/etc/nsswitch` sieht wie folgt aus:

```
# Datei /etc/nsswitch (CentOS, Fedora, RHEL)
passwd: files sss
shadow: files sss
group: files sss
```

`sssd` ist zwar kein vollständiger Ersatz für `nscd`, bietet dafür aber andere Zusatzfunktionen. In jedem Fall ist der Einsatz von `nscd` und/oder `sssd` nur zweckmäßig, wenn Ihre Linux-Clients auf ein zentrales Authentifizierungssystem zugreifen. Hilfreich bei der Differenzierung zwischen den beiden Cache-Systemen ist der folgende Blog-Beitrag:

<http://geekdom.wesmo.com/2014/05/16/embracing-sssd-in-linux>

## 18.7 Spracheinstellung, Internationalisierung, Unicode

In diesem Abschnitt geht es um zwei Dinge:

- ▶ **Lokalisierung bzw. Spracheinstellung:** Diese Einstellung bestimmt, in welcher Sprache Fehlermeldungen, Menüs, Dialoge, Hilfetexte etc. angezeigt werden. Auch die Formatierung von Datum, Uhrzeit, Währungsbeträgen etc. ändert sich dadurch entsprechend.
- ▶ **Zeichensatz:** Der Zeichensatz bestimmt, welche Codes zur Speicherung von Buchstaben verwendet werden. Hier herrscht generelle Einigkeit nur für 7-Bit-ASCII. Beinahe jeder Zeichensatz verwendet den Code 65 für den Buchstaben A. Abweichungen gibt es hingegen bei internationalen Zeichen: Daher gelten für den

Buchstaben Ä je nach Zeichensatz unterschiedliche Codes. Es gibt sogar Zeichensätze, in denen Ä überhaupt nicht vorkommt, um so Platz für andere Zeichen, z. B. kyrillische oder hebräische Buchstaben, zu schaffen.

### Was bedeutet i10n und i18n?

Im Zusammenhang mit der Lokalisierung von Programmen werden Sie immer wieder auf die merkwürdigen Kürzel i18n und i10n stoßen: Dabei handelt es sich um die Kurzschreibweise für »Internationalization« (*i* plus 18 Buchstaben plus *n*) bzw. »Localization«. Diese Kürzel eignen sich auch gut als Suchbegriff, falls Sie im Internet nach weiteren Informationen suchen möchten.

## Zeichensatz-Grundlagen

Ein Zeichensatz (*character set*) beschreibt die Zuordnung zwischen Zahlencodes und Buchstaben. Bekannte Zeichensätze sind ASCII (7 Bit), ISO-Latin-*n* (8 Bit) und Unicode (16 bzw. 32 Bit).

- ▶ **ASCII:** Der ASCII-Zeichensatz beschreibt lediglich 127 Zeichen, darunter die Buchstaben a–z bzw. A–Z, die Ziffern 0–9 sowie diverse Interpunktionszeichen.
- ▶ **ISO-8859, Latin-Zeichensätze:** Die ISO-Zeichensätze enthalten neben den 127 ASCII-Zeichen bis zu 128 zusätzliche Sonderzeichen für verschiedene Sprachregionen. Beispielsweise enthält ISO-8859-1 = Latin-1 alle in Westeuropa üblichen Zeichen, ISO-8859-2 = Latin-2 die in Zentral- und Osteuropa wichtigsten Zeichen etc. Der Zeichensatz ISO-8859-15 = Latin-9 entspricht Latin-1, enthält aber zusätzlich das Euro-Zeichen. Unter Windows werden Zeichensätze *Code Pages* genannt. Code Page 1252 stimmt weitgehend mit Latin-1 überein.
- ▶ **Unicode:** Um das Durcheinander verschiedenster 8-Bit-Zeichensätze zu lösen, wurde der Zeichensatz Unicode (ISO-10646) entworfen. Damit können nicht nur alle europäischen Sonderzeichen codiert werden, sondern auch die meisten asiatischen Zeichen. In den ersten Unicode-Versionen waren für jedes Zeichen 16 Bit vorgesehen, d. h., der Zeichensatz bot Platz für über 65.000 Zeichen. Selbst diese Zahl stellte sich als zu klein heraus, weswegen aktuelle Unicode-Versionen auch 32-Bit-Codes erlauben.

Unicode regelt nur, welcher Code welchem Zeichen zugeordnet ist, nicht aber, wie die Codes gespeichert werden. Die einfachste Lösung ist es, jedes Zeichen einfach durch 2 oder 4 Byte darzustellen. Diese Formatierung wird UTF-16 bzw. UTF-32 genannt (Unicode Transfer Format). Sie hat allerdings zwei Nachteile: Erstens verdoppelt bzw. vervierfacht sich der Speicherbedarf, und zwar auch in solchen Fällen, in denen überwiegend europäische Zeichen oder sogar nur US-ASCII-Zeichen gespeichert werden sollen. Zweitens tritt der Bytecode 0 an beliebigen Stellen in

Unicode-Zeichenketten auf. Viele C-Programme, E-Mail-Server etc. setzen aber voraus, dass das Byte 0 das Ende einer Zeichenkette markiert.

Deswegen gibt es auch andere Möglichkeiten, Unicode-Texte zu repräsentieren. Am populärsten ist UTF-8. Dabei werden die US-ASCII-Zeichen (7 Bit) wie bisher durch ein Byte dargestellt, deren oberstes Bit 0 ist. Alle anderen Unicode-Zeichen werden durch unterschiedlich lange Byte-Ketten dargestellt. Der offensichtliche Nachteil dieses Formats besteht darin, dass es keinen unmittelbaren Zusammenhang zwischen der Byteanzahl und der Anzahl der Zeichen eines Dokuments gibt. Dennoch hat sich UTF-8 unter Unix/Linux sowie im Internet als De-facto-Standard etabliert. Alle gängigen Linux-Distributionen verwenden standardmäßig UTF-8.

#### Auswirkungen des Zeichensatzes

Der aktive Zeichensatz entscheidet darüber, wie Zeichen in Textdateien bzw. in Dateinamen codiert werden. Die Dateisysteme von Linux kommen mit jedem Zeichensatz zurecht. Als Dateiname gilt jede Zeichenkette, die mit dem Bytecode 0 endet. Je nachdem, welcher Zeichensatz gerade gültig ist, kann die Bytefolge und -anzahl für einen Dateinamen wie `äöü.txt` aber ganz unterschiedlich sein! Wenn der aktuelle Zeichensatz Latin-1 lautet, kann dieser Name durch 7 Byte (plus ein 0-Byte) ausgedrückt werden. Wenn als Zeichensatz dagegen Unicode/UTF-8 verwendet wird, ist der Dateiname 10 Byte lang, weil zur Darstellung von `ä`, `ö` und `ü` jeweils zwei Byte benötigt werden.

Es gibt eine Reihe von Programmen, die unabhängig vom Zeichensatz funktionieren bzw. mit mehreren Zeichensätzen gleichzeitig zurechtkommen: Beispielsweise können E-Mail-Programme und Webbrowser auch E-Mails bzw. Webseiten darstellen, die nicht den gerade aktiven Zeichensatz verwenden. Damit das funktioniert, enthält jede E-Mail bzw. jede Webseite Informationen über den eingesetzten Zeichensatz. Moderne Textverarbeitungsprogramme speichern den Text zumeist in einem Unicode-Zeichensatz oder unter Verwendung eines eigenen Codes. Auch Editoren wie Emacs oder XEmacs sind grundsätzlich in der Lage, Textdateien in verschiedenen Codierungen zu verarbeiten bzw. zu speichern.

#### Zeichensatzprobleme

Probleme treten am häufigsten auf, wenn Sender und Empfänger beim Austausch von (Text-)Dateien einen unterschiedlichen Zeichensatz verwenden. Beispielsweise verfasst ein Benutzer einer Linux-Distribution mit Unicode-Zeichensatz mit einem Editor eine Textdatei mit internationalen Sonderzeichen. Nun soll ein Benutzer eines anderen Betriebssystems mit Latin-Zeichensatz die Datei weiterbearbeiten. Dieser Benutzer stellt zu seiner Verwunderung fest, dass alle Nicht-ASCII-Zeichen falsch dargestellt werden. Derartige Probleme lassen sich mit den Kommandos `recode` bzw. `iconv` zumeist leicht lösen. Diese Kommandos habe ich in [Abschnitt 13.3](#), »Textkonverter (Zeichensatz und Zeilentrennung)«, beschrieben.

Dieselben Probleme betreffen auch Dateinamen, insbesondere im Zusammenspiel mit NFS3: Wenn Sie auf einem Rechner mit UTF8-Zeichensatz die Datei äöü.txt erzeugen und ein anderer Rechner mit Latin-Zeichensatz via NFS auf diese Datei zugreift, sieht der Dateiname so ähnlich wie ÃÃÃ.txt aus. Abhilfe schaffen die Verwendung eines einheitlichen Zeichensatzes auf allen Rechnern des Netzwerks oder der Einsatz von NFS4. Wenn Sie den Zeichensatz für die Namen zahlreicher schon vorhandener Dateien ändern möchten, hilft das in [Abschnitt 13.4](#), »Dateinamenkonverter (Zeichensatz)«, beschriebene Kommando `convmv` weiter.

Die Schriftart darf nicht mit einem Zeichensatz verwechselt werden. Sie ist dafür zuständig, wie ein bestimmtes Zeichen auf dem Bildschirm angezeigt wird. Dazu gibt es verschiedene Schriftarten (z. B. Arial, Courier, Helvetica, Palatino, um einige bekannte zu nennen).

Schriftart (Font)

Natürlich haben Schriftarten und Zeichensätze miteinander zu tun: Bevor ein Zeichen mit dem Code 234 korrekt auf dem Bildschirm dargestellt werden kann, muss klar sein, welcher Zeichensatz für die Codierung verwendet wurde. Manche alten Schriftarten waren auf 256 Zeichen beschränkt und standen daher in getrennten Versionen für verschiedene Zeichensätze zur Verfügung. Neuere Schriften enthalten hingegen mehr Zeichen und sind Unicode-kompatibel.

### Lokalisation und Zeichensatz einstellen

Je nach Distribution bzw. Desktop-System können Sie verschiedene Werkzeuge zur Konfiguration der Sprache einsetzen. Als Zeichensatz kommt fast immer UTF-8 zum Einsatz. Nur wenige Distributionen bieten noch die Möglichkeit, einen 8-Bit-Zeichensatz einzustellen. Bei allen Distributionen müssen Sie sich neu einloggen, damit veränderte Spracheinstellungen wirksam werden. Gnome berücksichtigt die Spracheinstellung des Systems und bietet hierfür selbst keine Konfigurationswerkzeuge an.

Konfigurationswerkzeuge

|                    |   |
|--------------------|---|
| Debian:            | <code>dpkg-reconfigure locales</code> oder <code>localectl</code> |
| Fedora:            | <code>localectl</code>  |
| Gnome:             | Systemeinstellungsmodul REGION UND SPRACHE                        |
| KDE:               | Systemeinstellungsmodul PERSÖNLICHES • LAND/REGION                |
| RHEL 7 / CentOS 7: | <code>localectl</code>  |
| SUSE:              | YaST-Modul SYSTEM • SPRACHE                                       |
| systemd:           | <code>localectl</code>  |
| Ubuntu:            | <code>gnome-language-selector</code>                              |

Außerdem bieten manche Display-Manager im Login-Dialog für das Desktop-System die Möglichkeit, für die nächste Sitzung die gewünschte Sprache auszuwählen.

**Konfigurationsdateien** Die Konfigurationseinstellungen werden an unterschiedlichen Orten gespeichert:

```
Debian, Ubuntu:           /etc/default/locale
Fedora, RHEL 7 und CentOS 7, systemd: /etc/locale.conf
SUSE:                     /etc/sysconfig/language
```

`/etc/locale.conf` ist der vom `systemd`-Entwickler vorgeschlagene neue Ort, vermutlich werden nach und nach auch andere Distributionen diese Datei verwenden. Um eine neue Standardsprache einzustellen, können Sie bei Distributionen mit einer aktuellen `systemd`-Version das Kommando `localectl set-locale` verwenden. Eine Liste aller möglichen Einstellungen liefert `localectl list-locales`. Viele Distributionen berücksichtigen darüber hinaus benutzerspezifische Einstellungen in der Datei `.i18n` im Heimatverzeichnis.

**LC- und LANG-Variablen**

Intern wird sowohl die Lokalisation als auch der Zeichensatz durch Umgebungsvariablen wie `LC_CTYPE` und `LANG` gesteuert. Für die Auswertung dieser Variablen ist die `glibc`-Bibliothek verantwortlich, die in fast allen Linux-Programmen zum Einsatz kommt. Die Lokalisation kann kategorieweise durchgeführt werden. Damit ist es möglich, beispielsweise für Datums- und Zeitangaben das in Deutschland übliche Format zu verwenden, Fehlermeldungen aber dennoch in Englisch anzuzeigen. [Tabelle 18.5](#) zählt die wichtigsten Variablen auf.

| Variable                 | Bedeutung  |
|--------------------------|--|
| <code>LANG</code>        | bestimmt den Standardwert für alle nicht eingestellten LC-Variablen. |
| <code>LC_CTYPE</code>    | bestimmt den Zeichensatz.  |
| <code>LC_COLLATE</code>  | bestimmt die Sortierordnung.   |
| <code>LC_MESSAGES</code> | bestimmt die Darstellung von Nachrichten, Fehlermeldungen etc.       |
| <code>LC_NUMERIC</code>  | bestimmt die Darstellung von Zahlen.                                 |
| <code>LC_TIME</code>     | bestimmt die Darstellung von Datum und Uhrzeit.                      |
| <code>LC_MONETARY</code> | bestimmt die Darstellung von Geldbeträgen.                           |
| <code>LC_PAPER</code>    | bestimmt die Papiergröße.  |
| <code>LC_ALL</code>      | überschreibt alle individuellen LC-Einstellungen.                    |

**Tabelle 18.5** Wichtige Lokalisationsvariablen

Natürlich berücksichtigt nicht jedes Programm alle Kategorien; manche Programme ignorieren die `LC_`-Variablen sogar vollständig. Wenn einzelne Kategorien nicht eingestellt sind, verwenden Programme als Standardwert `C` bzw. `POSIX`. Das bedeutet, dass Fehlermeldungen auf Englisch erscheinen, Daten und Zeiten im amerikanischen Format dargestellt werden etc.

Anstatt alle Variablen einzeln einzustellen, können Sie einfach die Variable `LANG` einstellen. Damit wird für alle undefinierten Variablen der `LANG`-Standardwert verwendet. Einzig bei `LC_COLLATE` bleibt die Grundeinstellung `POSIX`. Bei den meisten Distributionen erfolgt die gesamte Spracheinstellung über die `LANG`-Variable.

Noch stärker als `LANG` wirkt `LC_ALL`. Wenn diese Variable gesetzt wird, gilt für alle Kategorien diese Einstellung, egal wie `LANG` oder andere `LC_`-Variablen eingestellt sind.

Bei den meisten Programmen befinden sich Fehlermeldungen und andere Texte für jede Sprache separat in eigenen Verzeichnissen, z. B. in `/usr/share/locale*/sprache/LC_MESSAGES`. Weitere Hintergrundinformationen zum Thema *Locales and Internationalization* finden Sie mit dem Kommando `man locale` sowie auf der folgenden Website:

<http://www.gnu.org/software/libc/manual>

Den aktuellen Zustand der Lokalisationseinstellung können Sie am einfachsten mit dem Kommando `locale` ermitteln. Dieses Kommando wertet auch `LANG` und `LC_ALL` aus und ermittelt daraus die resultierenden Einstellungen. Das folgende Beispiel zeigt die Einstellung auf meinem Rechner:

Lokalisation  
testen

```
user$ locale
LANG=de_DE.UTF-8
LC_CTYPE="de_AT.UTF-8"
LC_TIME="de_AT.UTF-8"
...
LC_ALL=
```

Zum Testen der Lokalisation können Sie ein beliebiges Kommando fehlerhaft ausführen. Die Fehlermeldung erscheint in der eingestellten Sprache. Wenn `LANG` auf `de_DE` eingestellt ist, sieht die Fehlermeldung des `mount`-Kommandos wie folgt aus:

```
user$ mount /xy
mount: Konnte /xy nicht in /etc/fstab oder /etc/mtab finden
```

Wenn Sie ein einzelnes Kommando mit einer anderen Spracheinstellung ausführen möchten, ohne gleich die gesamte Konfiguration zu ändern, verwenden Sie am besten das Kommando `env`. Dieses Kommando erwartet eine Reihe von Variablenzuweisungen und schließlich das eigentliche Kommando, das unter Berücksichtigung der eingestellten Variablen ausgeführt wird:

env

```
user$ env LANG=C mount /xy
mount: can't find /xy in /etc/fstab or /etc/mtab
```

Falls die Fehlermeldung trotz geänderter `LANG`-Einstellung noch immer in der jeweiligen Landessprache (statt in Englisch) erscheint, versuchen Sie, auch `LANGUAGE` zurückzusetzen:

```
user$ env LANG=C LANGUAGE=C mount /xy
mount: can't find /xy in /etc/fstab or /etc/mtab
```

Um LANG für den gesamten Verlauf einer Sitzung einzustellen, führen Sie export LANG=C aus.

#### Zulässige LC/LANG-Einstellungen

Eine Liste aller möglichen Einstellungen ermitteln Sie mit locale -a. Üblicherweise wird die Schreibweise x\_y verwendet, wobei x durch zwei Buchstaben die Sprache und y durch zwei Buchstaben das Land bezeichnet. Im deutschen Sprachraum sollten Sie de\_DE verwenden. Für die englische Standardeinstellung ist die Kurzschreibweise C erlaubt. Neuere glibc-Versionen verstehen auch Einstellungen wie deutsch oder german. Die Datei /usr/share/locale/locale.alias enthält eine Tabelle, die die zulässigen Kurzschreibweisen dem vollständigen Lokalisationsnamen zuordnet.

#### Lokalisierungs-pakete

Ob Menüs, Dialoge, Fehlermeldungen, Hilfetexte etc. tatsächlich in der richtigen Sprache angezeigt werden, hängt davon ab, ob die dazu erforderlichen Lokalisierungsdateien installiert sind. Aus Platzgründen ist dies oft nur für eine oder zwei Sprachen, z. B. Englisch und Deutsch, der Fall. Wenn Sie Ihre Distribution auch in französischer Sprache nutzen möchten, müssen Sie für Gnome, KDE, OpenOffice, Firefox etc. entsprechende Zusatzpakete installieren. Bei SUSE und Ubuntu helfen Ihnen dabei die in der Einleitung dieses Kapitels aufgezählten Konfigurationswerkzeuge, bei anderen Distributionen ist hier aber Handarbeit erforderlich.

Nicht jedes Linux-Programm ist für jede Sprache lokalisiert. Besonders große Lücken gibt es bei der Online-Dokumentation, also bei man-Seiten, Handbüchern und Hilfetexten. Wenn geeignete Lokalisierungsdateien fehlen, zeigt Linux englische Texte an.

#### Einstellung des Zeichensatzes

Zusammen mit der Lokalisation wird auch der Zeichensatz eingestellt. Der Zeichensatz folgt dem Ländercode nach einem Punkt, z. B. de\_DE.ISO-8859-1 oder de\_DE.utf8.

## 18.8 Hardware-Referenz

In diesem Buch gibt es kein eigenes Hardware-Kapitel. Die richtige Konfiguration von Hardware-Komponenten wird stattdessen in den dazu passenden Kapiteln behandelt: Wenn Sie also beispielsweise Probleme mit einer Netzwerkkarte haben, ist [Kapitel 26](#), »Netzwerkkonfiguration«, der richtige Startpunkt.

Dieser Abschnitt hat somit zwei Aufgaben: Zum einen soll er die Suche nach weiteren Informationen zu bestimmten Hardware-Komponenten erleichtern. Zum anderen finden Sie hier kurze Informationen zu Hardware-Themen, die im Rest des Buchs zu kurz kommen. Natürlich gibt es eine Menge Hardware-Komponenten, die in diesem Buch aus den verschiedensten Gründen *nicht* beschrieben sind. Das liegt primär



daran, dass ich nicht die Testmöglichkeiten habe, über die beispielsweise eine Computerzeitschrift verfügt.

### Zuerst recherchieren, dann kaufen!

Erkundigen Sie sich *vor dem Kauf*, ob Ihre neue Hardware Linux-kompatibel ist! Führen Sie im Internet eine Suche mit den Begriffen *linux <modellname>* durch. Auch Linux-orientierte Zeitschriften sind für diesen Zweck naturgemäß eine aktuellere Informationsquelle als Bücher.

Die meisten Hardware-Komponenten werden über sogenannte Devices angesprochen – z. B. `/dev/sda` für eine SATA-Festplatte. Die Device-Dateien werden dynamisch bei Bedarf durch das `udev`-System erzeugt. Eine Liste mit den wichtigsten Linux-Device-Dateien finden Sie in [Abschnitt 11.9](#).

Device-Dateien

Die Treiber zu zahllosen Hardware-Komponenten befinden sich in Kernelmodulen. Ein Teil dieser Module wird während des Systemstarts geladen, die restlichen Module erst bei Bedarf. Wenn das automatische Laden von Modulen nicht funktioniert, sollten Sie einen Blick in die Dateien `/etc/mmodprobe.conf` bzw. `/etc/modprobe.conf.d/*` werfen. Der Umgang mit Modulen und die Funktion dieser Dateien werden in [Abschnitt 25.1](#) beschrieben.

Kernelmodule

Was beim Laden von Modulen geschieht und ob die Hardware erfolgreich initialisiert werden kann, geht aus den Kernelmeldungen hervor. Diese lesen Sie mit dem Kommando `dmesg`.

Bei vielen Komponenten geben virtuelle Dateisysteme in den Verzeichnissen `/proc` und `/sys` detaillierte Informationen. Einen Überblick über solche Hardware-Dateien finden Sie in [Abschnitt 25.7](#).

`/proc-`  
`/sys-Dateien`

Um einen Überblick über die laufende Hardware zu erlangen, führen Sie die Kommandos `lsblk`, `lspci` und `lsusb` aus. Auch ein Blick in die Kernelmeldungen mit `dmesg` ist oft aufschlussreich.

Hardware-  
Überblick

## CPU und Speicher

Welche CPUs in Ihrem Rechner laufen, geht aus der Datei `/proc/cpuinfo` hervor. Die folgende, stark gekürzte Ausgabe entstand auf einem Rechner mit einem Intel-i7-Prozessor. Linux betrachtet die Cores wie eigenständige Prozessoren. Dabei enthält die Zeile `model name` die maximale Taktfrequenz.

CPU

```

user$ cat /proc/cpuinfo
cat /proc/cpuinfo
processor      : 0
model name    : Intel(R) Core(TM) i7 CPU           860 @ 2.80GHz
...
processor      : 1
model name    : Intel(R) Core(TM) i7 CPU           860 @ 2.80GHz
...

```

Bei Prozessoren mit variabler Taktfrequenz sind `cpufreq`-Module für die energie-sparende Reduzierung der Frequenz zuständig. Gute Hintergrundinformationen zu diesem System geben die folgenden zwei Seiten:

<http://www.kernel.org/doc/Documentation/cpu-freq>  
[https://wiki.archlinux.org/index.php/CPU\\_Frequency\\_Scaling](https://wiki.archlinux.org/index.php/CPU_Frequency_Scaling)

Daten zum aktuellen Zustand des Systems sowie Steuerungsmöglichkeiten bieten die Dateien des folgenden Verzeichnisses:

```
/sys/devices/system/cpu/cpuN/cpufreq/
```

Die Anzahl der CPU-Cores ermitteln Sie am einfachsten mit dem Kommando `nproc`.

#### CPU-Frequenz limitieren

Um zu vermeiden, dass die CPU schnell heiß läuft, kann man die maximale CPU-Frequenz limitieren. Dazu verwenden Sie das Kommando `cpufreq-set` aus dem Paket `cpufrequtils`. Das folgende Kommando beschränkt die Frequenz auf 800 MHz:

```
root# cpufreq-set -r -max 0.8GHz
```

Über den Nutzen dieser Anweisung kann man allerdings geteilter Meinung sein: Rechenintensive Aufgaben dauern nun einfach länger, und dabei wird in der Regel noch mehr Wärme freigesetzt.

#### CPU-Temperatur überwachen

Wenn Sie wissen möchten, welche Temperatur die CPU Ihres Rechners gegenwärtig aufweist, installieren Sie das Paket `lm-sensors` oder `lm_sensors`. Nach der Installation führen Sie als `root` das Kommando `sensors-detect` aus. Es stellt fest, welche Hardware-Komponenten Informationen über ihren Zustand liefern. Neben der CPU können das auch die Festplatte, die Grafikkarte oder diverse Lüfter sein, die ihre Drehzahl melden. Zum Schluss werden die erforderlichen Kernelmodule nach einer Rückfrage in der Konfigurationsdatei `/etc/sysconfig/lm_sensors` gespeichert:

```
# Datei /etc/sysconfig/lm_sensors
HWMON_MODULES="coretemp nct6775"
```

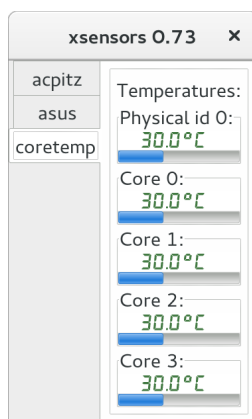
Nach diesen Vorbereitungsarbeiten liefert das Kommando `sensors` eine aktuelle Temperaturliste:

```

root# sensors
...
coretemp-isa-0000
Adapter: ISA adapter
Physical id 0: +34.0°C (high = +80.0°C, crit = +98.0°C)
Core 0:        +31.0°C (high = +80.0°C, crit = +98.0°C)
Core 1:        +33.0°C (high = +80.0°C, crit = +98.0°C)
...

```

Es gibt diverse Programme, die diese Daten eleganter darstellen. Für erste Experimente bietet sich `xsensors` an, das die Daten aller Sensoren in einem Fenster anzeigt (siehe [Abbildung 18.2](#)). Unter Ubuntu installieren Sie das Paket `psensor`.



**Abbildung 18.2** Temperaturanzeige mit `xsensors`

Informationen über den verfügbaren Speicher erhalten Sie mit dem Kommando `free`. Wenn Sie vermuten, dass Ihr Rechner defekte Speicherbausteine hat, bietet das Programm `Memtest86` eine gute Möglichkeit, das RAM zu testen. Bei fast allen Distributionen kann das Programm komfortabel während des Systemstarts gestartet werden. Sollte das bei Ihnen nicht funktionieren, finden Sie auf der folgenden Website ein ISO-Image, um eine boot-fähige CD zu brennen:

Speicher (RAM)

<http://memtest86.com>

## Energieverwaltung

ACPI steht für *Advanced Configuration and Power Interface* und steuert die Energieverwaltungsfunktionen aller marktüblichen PCs und Notebooks. ACPI wird von Linux unterstützt, wovon Sie sich mit `dmesg | grep -i acpi` überzeugen können.

ACPI

ACPI unterstützt verschiedene Schlafmodi, in denen der Rechner wenig (Bereitschaft, Stand-by-Modus) bzw. gar keinen Strom verbraucht (Suspend-Modus, Hibernation).

Suspend

Modus, Ruhezustand). Bei den meisten Distributionen bzw. Desktop-Systemen versetzen Sie den Rechner über das Systemmenü in den gewünschten Schlafmodus.

Im Bereitschaftsmodus wird die CPU in einen Ruhemodus versetzt, in dem sie nur wenig Strom braucht. Das RAM wird weiter mit Strom versorgt, Displays und Datenträger werden aber abgeschaltet.

Im Ruhezustand wird dagegen der aktuelle RAM-Inhalt in der Swap-Partition der Festplatte gespeichert und der Rechner dann vollständig ausgeschaltet. Er braucht dann überhaupt keinen Strom mehr. Das setzt voraus, dass die Swap-Partition ausreichend groß ist! Beim Aufwachen wird der Speicher wieder von der Festplatte oder SSD gelesen. Außerdem müssen sämtliche Hardware-Komponenten neu initialisiert werden. Dieser Prozess ist sehr komplex und erfordert ein optimales Zusammenspiel des Linux-Kernels, seiner Module und des ACPI-Systems.


#### Aus der Ruhe erwachen ...

Meine persönlichen Erfahrungen mit dem Ruhezustand sind leider überwiegend negativ. Soweit nicht schon beim Versuch der Aktivierung ein Fehler auftrat, gelang es oft nicht mehr, den Rechner aus dem Schlafmodus wieder aufzuwecken. Aus diesem Grund gehen manche Distributionen dazu über, den Suspend-Modus in den Systemmenüs gar nicht mehr anzubieten. Wenn doch, sollten Sie die Suspend-Funktionen anfangs mit Vorsicht testen: Sichern Sie vorher alle Daten, führen Sie sync aus, und lösen Sie alle nicht benötigten Dateisysteme aus dem Verzeichnisbaum!


Mehr Glück hatte ich auf meinen Testrechnern mit dem Bereitschaftsmodus, der in der Regel zuverlässig funktionierte.

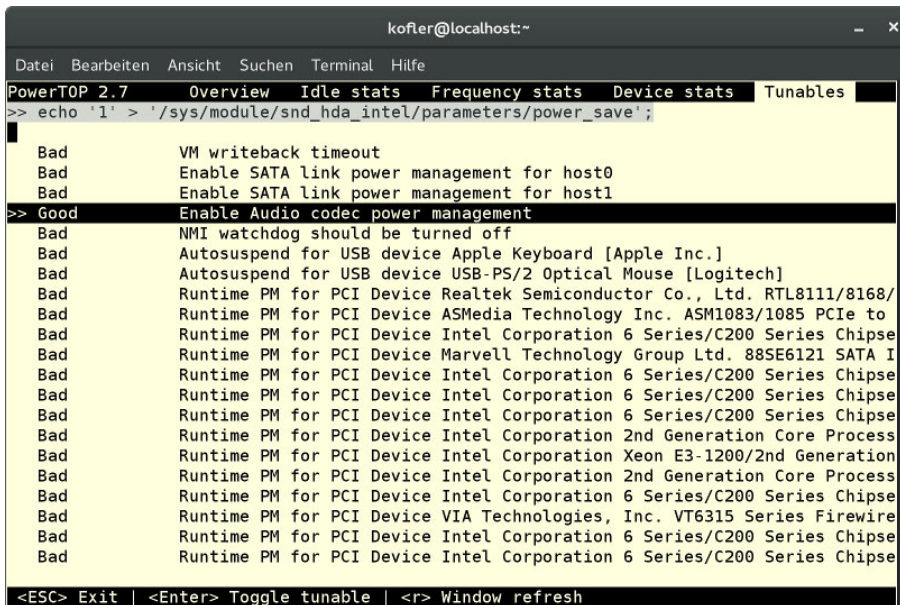
**powertop** Das Kommando `powertop` aus dem gleichnamigen Paket hilft bei der Suche nach Programmen, die einen hohen Stromverbrauch verursachen bzw. die die Energiesparfunktionen der CPU torpedieren. Das Programm gibt Tipps, wie der Energieverbrauch minimiert werden kann. Im folgenden Kommando bewirkt `LC_ALL=c`, dass `powertop` englische Menüs und Statustexte anzeigt. Das ist vor allem dann hilfreich, wenn Sie im Internet nach Details zu einzelnen Parametern suchen möchten.

```
root# LC_ALL=c sudo powertop
```

Nach dem Start können Sie mit  zwischen mehreren Statusseiten wechseln. Diese zeigen an, welche Prozesse die CPU wie oft aus einem Ruhezustand wecken, wie oft sich die CPU in welchen Ruhezuständen befindet, wie oft die CPU in welcher Taktfrequenzstufe läuft, welche Devices wie stark genutzt werden etc.

Die aus der Sicht der Energiesparfunktionen interessanteste Seite heißt TUNABLES (siehe [Abbildung 18.3](#)). Dort zeigt `powertop` eine Liste von Einstellungen an, deren aktueller Zustand BAD bzw. GOOD sein kann. Mit den Cursortasten können Sie nun

einzelne Punkte auswählen und durch  umstellen. `powertop` zeigt dabei an, welches Kommando es ausführt.



```

kofler@localhost:~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
PowerTOP 2.7 Overview Idle stats Frequency stats Device stats Tunables
>> echo '1' > '/sys/module/snd_hda_intel/parameters/power_save';
Bad VM writeback timeout
Bad Enable SATA link power management for host0
Bad Enable SATA link power management for host1
>> Good Enable Audio codec power management
Bad NMI watchdog should be turned off
Bad Autosuspend for USB device Apple Keyboard [Apple Inc.]
Bad Autosuspend for USB device USB-PS/2 Optical Mouse [Logitech]
Bad Runtime PM for PCI Device Realtek Semiconductor Co., Ltd. RTL8111/8168/
Bad Runtime PM for PCI Device ASMedia Technology Inc. ASM1083/1085 PCIe to
Bad Runtime PM for PCI Device Intel Corporation 6 Series/C200 Series Chipse
Bad Runtime PM for PCI Device Marvell Technology Group Ltd. 88SE6121 SATA I
Bad Runtime PM for PCI Device Intel Corporation 6 Series/C200 Series Chipse
Bad Runtime PM for PCI Device Intel Corporation 6 Series/C200 Series Chipse
Bad Runtime PM for PCI Device Intel Corporation 6 Series/C200 Series Chipse
Bad Runtime PM for PCI Device Intel Corporation 2nd Generation Core Process
Bad Runtime PM for PCI Device Intel Corporation Xeon E3-1200/2nd Generation
Bad Runtime PM for PCI Device Intel Corporation 2nd Generation Core Process
Bad Runtime PM for PCI Device Intel Corporation 6 Series/C200 Series Chipse
Bad Runtime PM for PCI Device VIA Technologies, Inc. VT6315 Series Firewire
Bad Runtime PM for PCI Device Intel Corporation 6 Series/C200 Series Chipse
Bad Runtime PM for PCI Device Intel Corporation 6 Series/C200 Series Chipse
<ESC> Exit | <Enter> Toggle tunable | <R> Window refresh

```

Abbildung 18.3 Die Tunables-Ansicht von `powertop`

Sie können nun schrittweise versuchen, einzelne Einstellungen zu ändern, und dann testen, welche Auswirkungen dies hat: Sinkt der mit einem Strommessgerät erfasste Energieverbrauch des zuvor aufgeladenen Notebooks tatsächlich spürbar? Verursachen die deaktivierten Funktionen Probleme? Lassen sich USB-Geräte weiterhin nutzen, funktioniert das Ein- und Ausschalten des WLAN-Adapters weiterhin, funktioniert das Audio-System ohne Störgeräusche etc.?

Die mit `powertop` durchgeführten Änderungen gelten nur bis zum nächsten Neustart des Rechners. Um die Energiesparmaßnahmen dauerhaft zu aktivieren, tragen Sie die von `powertop` angezeigten Kommandos (z. B. `echo '1' > /sys/xxx`) in eine Datei ein, die bei jedem Systemstart ausgeführt wird. Bei den meisten Distributionen eignet sich dazu `/etc/rc.d/rc.local`. Gegebenenfalls müssen Sie diese Datei erzeugen und mit `chmod a+x` als ausführbar kennzeichnen.

Eine radikale Lösung besteht darin, in `rc.local` anstelle einzelner Tuning-Kommandos `powertop --auto-tune` einzutragen. Dann führt `powertop` einfach alle bekannten Optimierungsmaßnahmen aus. Leider schießt `powertop` damit oft über das Ziel hinaus: Was nützt es, wenn das Notebook eine Stunde länger als bisher läuft, aber die Netzwerkverbindung nur noch unzuverlässig funktioniert?

### Liste aller Tuning-Kommandos erstellen

Eine Liste aller möglichen Tuning-Kommandos erhalten Sie, wenn Sie `powertop` mit der Option `--html` ausführen. Das Kommando erzeugt dann nach einer Messzeit von ca. 20 Sekunden die HTML-Datei `powertop.html`, die neben diversen statistischen Daten auch eine Zusammenfassung aller Tuning-Parameter enthält.

## Lüftersteuerung

Nichts nervt mehr, als wenn bei einem an sich leisen Notebook ständig der Lüfter heult. Lässt sich dagegen etwas machen? Nach meinen Erfahrungen eher nicht – achten Sie schon beim Kauf darauf, ein Modell auszuwählen, das leise ist. Für alle, die den Krach doch per Software mindern möchten, folgen hier einige Empfehlungen.

**Fan Control** Für manche Notebooks gibt es spezielle Programme zur Steuerung des Lüfters. Diese Programme setzen voraus, dass zuvor das Paket `lm-sensors` installiert und konfiguriert wurde. Es ist notwendig, damit die CPU-Temperatur überwacht werden kann. Sofern diese Voraussetzung erfüllt ist, können Sie einstellen, ab welcher Temperatur welcher Lüfter in welcher Drehzahl laufen soll.

Eines der populärsten Programme zur Lüftersteuerung hat den Namen `tpfanco`. Es ist zu fast allen Notebooks von IBM und Lenovo kompatibel. Aktuelle Debian-Pakete gibt es hier zum Download:

<https://github.com/tpfanco>

Nach der Installation und dem manuellen Start des `tpfan`-Dämons kann die Lüftersteuerung mit `tpfan-admin` kontrolliert werden (erfordert `root`-Rechte). In dieser grafischen Benutzeroberfläche kann die Einschaltsschwelle für jeden der im System erkannten Temperatursensoren individuell eingestellt werden. Ich habe das Programm auf dem Lenovo-Notebook E320 einige Tage lang ausprobiert. Ich habe zwar erreicht, dass der Lüfter weniger häufig als zuvor lief, ganz verhindern konnte ich seinen Betrieb aber nicht.

### Vorsicht

Naturgemäß erfolgt die manuelle Lüftersteuerung auf eigene Gefahr. Wenn die CPU oder andere Komponenten des Rechners regelmäßig zu heiß sind, verringert sich deren Lebensdauer! Setzen Sie Programme zur Lüftersteuerung daher mit Vorsicht ein, und recherchieren Sie vorher im Internet, welche Erfahrungen andere Benutzer gemacht haben.

## Schnittstellen und Bussysteme

Unter Linux sind serielle bzw. parallele Schnittstellen über die Device-Dateien `/dev/ttyS<n>` bzw. `/dev/lp<n>` zugänglich. Am ehesten treffen Sie auf diese im PC-Sektor nicht mehr üblichen Schnittstellen bei Mini-Computern wie dem Raspberry Pi.

Serielle und  
parallele  
Schnittstelle

Interne Festplatten und SSDs, CD- und DVD-Laufwerke sowie diverse andere Datenträger sind in der Regel über die Bussysteme IDE (darauf stoßen Sie am ehesten in virtuellen Maschinen), SATA oder SCSI mit dem Rechner verbunden (siehe auch [Abschnitt 22.3](#), »Device-Namen für Festplatten und andere Datenträger«). Aktuelle Linux-Versionen kommunizieren mit IDE-, SATA- und SCSI-Geräten über das SCSI-System des Kernels. Informationen über den Zustand der IDE- und SCSI-Systeme und aller damit verwalteten Geräte geben das Kommando `lsscsi` sowie die folgenden Dateien:

IDE, SATA, SCSI

```
/sys/bus/scsi/*
/proc/scsi/*
```

Der *Universal Serial Bus* (USB) wird zur Verbindung zwischen dem Computer und diversen externen Geräten eingesetzt – von der Maus bis zum Scanner. Die erforderlichen USB-Kernelmodule werden automatisch geladen. USB-Datenträger (also USB-Festplatten, Memorysticks, externe DVD-Laufwerke) etc. werden wie SCSI-Geräte behandelt.

USB

Die Verzeichnisse `/dev/bus/usb` und `/sys/bus/usb` enthalten Informationen über alle angeschlossenen USB-Geräte. Eine übersichtliche Liste aller USB-Schnittstellen und -Geräte liefert `lsusb -v` (Paket `usbutils`).

Informationen über PCI-Komponenten in Ihrem Rechner ermitteln Sie am besten mit dem Kommando `lspci`. Die Dateien in `/proc/bus/pci/` und `/sys/bus/pci/` enthalten dieselben Informationen, sind aber wesentlich schwieriger zu interpretieren. Die folgende Ausgabe ist aus Platzgründen stark gekürzt:

PCI

```
root# lspci
00:00.0 Host bridge: Intel Corporation 2nd Generation Core Processor
          Family DRAM Controller
00:02.0 VGA compatible controller: Intel Corporation 2nd Generation
          Core Processor Family Integrated Graphics Controller
...
05:00.0 Ethernet controller: Realtek Semiconductor Co., Ltd.
          RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller
06:00.0 PCI bridge: ASMedia Technology Inc. ASM1083/1085 PCIe to PCI Bridge
```

Die Verwendung der Grafikkarten erfordert eigene Treiber. Deren Konfiguration ist Thema von [Kapitel 21](#), »Grafiksystem«.

Grafik (X)

**Netzwerk-schnittstellen** Detaillierte Informationen zur Konfiguration von LAN- und WLAN-Schnittstellen sowie zum Umgang mit ADSL- und UMTS-Modems finden Sie in Kapitel 26, »Netzwerk-konfiguration«.

## Bluetooth

Bluetooth ist ein Verfahren zur Kommunikation von Hardware-Geräten per Funk. Bluetooth wird überwiegend in elektronischen Kleingeräten eingesetzt (Tastaturen, Mäuse, Lautsprecher, Smartphones etc.). Unter Gnome helfen die Werkzeuge aus dem Paket `gnome-bluetooth` bei der Bluetooth-Konfiguration. Bei aktuellen KDE-Versionen kommt stattdessen das Programmpaket `bluedevil` zum Einsatz. Raspbian verwendet seit Frühjahr 2016 eine eigene Bluetooth-Oberfläche, die in den sogenannten »Pixel Desktop« integriert ist. Eher eine Notlösung ist hingegen der Bluetooth-Manager, der in früheren Raspbian-Versionen sowie einigen anderen älteren Distributionen verwendet wurde.

Hinter den Kulissen ist für die Implementierung des Bluetooth-Stacks der Dämon `bluetoothd` im Zusammenspiel mit `udev` für die Verwaltung der Bluetooth-Geräte verantwortlich. Die entsprechenden Konfigurationsdateien befinden sich im Verzeichnis `/etc/bluetooth`. Weitere Informationen zu Bluetooth unter Linux finden Sie hier:

<http://www.bluez.org>

**bluetoothctl** Nur in seltenen Fällen ist es erforderlich, Bluetooth auf Kommandozeilenebene zu konfigurieren. Ein mögliches Szenario ist die Verwendung des Raspberry Pi ohne Monitor, wenn die Steuerung und Kommunikation ausschließlich über SSH erfolgen soll. In solchen Fällen verwenden Sie am besten das Kommando `bluetoothctl`. Es ist zur interaktiven Nutzung gedacht. Nach dem Start gelangen Sie in einen Kommandomodus, in dem Sie diverse Befehle ausführen können: `list` listet alle Bluetooth-Controller auf (z. B. einen USB-Adapter). `devices` listet alle Geräte in Funkreichweite auf. Mit `exit` oder `[Strg]+[D]` gelangen Sie zurück in den Eingabemodus des Terminals.

Um ein neues Bluetooth-Gerät mit Ihrem Computer zu verbinden, gehen Sie innerhalb einer `bluetoothctl`-Sitzung wie folgt vor:

- ▶ Sie aktivieren mit `pairable on` den Kuppelungsmodus.
- ▶ Sie aktivieren mit `scan on` den Scan-Modus. Das Programm listet nun alle erkannten Geräte in Funkreichweite auf. Dieser Vorgang kann geraume Zeit dauern, einzelne Geräte werden dabei immer wieder aufgelistet. Wenn Sie das gewünschte Gerät gefunden haben, schalten Sie den Modus mit `scan off` einfach wieder aus.



- ▶ Sie aktivieren mit `agent on` einen sogenannten Bluetooth-Agenten. Er kümmert sich um die Autorisierung neuer Geräte (bei Tastaturen: Passworteingabe).
- ▶ Mit `pair xx:xx:xx` initiieren Sie den Verbindungsaufbau zu einem Gerät. Bei einer Tastatur werden Sie nun dazu aufgefordert, einen sechsstelligen Code einzutippen. Vergessen Sie nicht, die Eingabe mit `↵` abzuschließen! Die erfolgreiche Kuppelung erkennen Sie an der Meldung *pairing successful*. Bei Geräten ohne Eingabemöglichkeit (Tastatur, Lautsprecher etc.) gelingt das Pairing zum Glück auch ohne Codeeingabe.
- ▶ Und mit `trust xx:xx:xx` machen Sie dem Bluetooth-System klar, dass Sie dem Gerät wirklich vertrauen.
- ▶ Mit `connect xx:xx:xx` geben Sie an, dass Sie das Gerät tatsächlich nutzen möchten. (Das hätte sich `bluetoothctl` mittlerweile eigentlich denken können ...) Wenn alles klappt, lautet die Reaktion *connection successful*. Das Gerät kann jetzt verwendet werden!
- ▶ `info xx:xx:xx` zeigt den Verbindungsstatus und diverse weitere Informationen zum Gerät an.

Wichtig ist, dass Sie bei Bluetooth-Geräten, die Sie neu konfigurieren wollen, immer wieder auf den Bluetooth-Knopf (Pairing-Knopf) drücken, damit das Gerät so der Umwelt signalisiert, dass es zum Verbindungsaufbau bereit ist. Bei vielen Geräten blinkt dann eine blaue Leuchtdiode. Wenn es keinen derartigen Knopf gibt, können Sie auch versuchen, das Gerät aus- und wieder einzuschalten.

Die folgenden Zeilen zeigen etwas gekürzt die Ein- und Ausgaben zur Konfiguration einer Bluetooth-Tastatur. Alle Ein- und Ausgaben erfolgen in einem Terminalfenster:

```
user$ bluetoothctl
[bluetooth]# agent on
[bluetooth]# pairable on
[bluetooth]# scan on
Discovery started
[CHG] Controller 00:1A:7D:DA:71:13 Discovering: yes
[NEW] Device 70:10:00:1A:92:20 70-10-00-1A-92-20
[CHG] Device 70:10:00:1A:92:20 Name: Bluetooth 3.0 Keyboard
...
[bluetooth]# scan off
[bluetooth]# pair 70:10:00:1A:92:20
Attempting to pair with 70:10:00:1A:92:20
[CHG] Device 70:10:00:1A:92:20 Connected: yes
[agent] PIN code: 963064
[CHG] Device 70:10:00:1A:92:20 Paired: yes
Pairing successful
[bluetooth]# trust 70:10:00:1A:92:20
[bluetooth]# connect 70:10:00:1A:92:20
```

```
[bluetooth]# info 70:10:00:1A:92:20
Device 70:10:00:1A:92:20
    Name: Bluetooth 3.0 Keyboard
    Paired: yes
    Trusted: yes
    Connected: yes
...
[bluetooth]# exit
```

Die Bluetooth-Konfiguration für ein bestimmtes Gerät wird in `/var/lib/bluetooth/id1/id2/info` gespeichert. Dabei ist `id1` der ID-Code des Bluetooth-Controllers und `id2` der ID-Code des Bluetooth-Geräts.

#### Verbindungs- aufbau erzwingen

Das Bluetooth-Gerät sollte nun auch beim nächsten Start korrekt erkannt werden. Bei meinen Tests auf einem Raspberry Pi bin ich an diesem Punkt aber mit manchen Bluetooth-Geräten gescheitert. Abhilfe schuf das Ausführen der Anweisung `bluez-test-input connect` zum Ende des Boot-Prozesses:

```
# Datei /etc/rc.local
...
echo -e "connect FC:58:FA:A0:4D:E7\nquit" | bluetoothctl
exit 0
```

### Hotplug-System

Bei modernen Rechnern können im laufenden Betrieb Festplatten, USB-Sticks und andere Geräte angeschlossen bzw. wieder entfernt werden. Linux muss auf die geänderte Hardware-Situation rasch und möglichst automatisch reagieren. Diese Aufgabe übernimmt das Hotplug-System, dessen Komponenten im Verlauf der letzten Jahre immer wieder verändert wurden. Zuletzt wurde der als ineffizient bekannte *Hardware Abstraction Layer* (HAL) aus den meisten Distributionen entfernt. Die aktuelle Vorgehensweise sieht so aus:

- ▶ **Kernel:** Der Kernel stellt Veränderungen an der Hardware fest, z. B. dass der Benutzer einen USB-Stick angesteckt hat.
- ▶ **udev:** Der Kernel erzeugt via `udev` neue Device-Dateien (siehe [Abschnitt 11.9](#)) und startet geeignete Programme, um die neuen Geräte zu verwalten bzw. um Benachrichtigungen an das Desktop-System zu versenden. Dabei werden Regeldateien aus den Verzeichnissen `/lib/udev/rules.d` sowie `/etc/udev/rules.d` ausgewertet. Tipps zum Verfassen bzw. Verändern dieser Regeln finden Sie hier:

[http://reactivated.net/writing\\_udev\\_rules.html](http://reactivated.net/writing_udev_rules.html)

- ▶ **DeviceKit:** Für manche Geräte bzw. Komponenten hilft das sogenannte DeviceKit bei der Verwaltung. Es besteht aus den Bibliotheken `libudisks2` und `libgudev`, die üblicherweise in gleichnamige Pakete verpackt sind. Für Festplatten(partitionen)

und externe Datenträger sind die Programme und Scripts des Pakets `udisks2` verantwortlich. Um die Energieverwaltung kümmern sich die Regeln und Programme des Pakets `upower`. Weitere Informationen finden Sie hier:

<https://freedesktop.org/wiki/Software/DeviceKit>

<https://freedesktop.org/wiki/Software/udisks>

<https://upower.freedesktop.org>

- ▶ **Desktop:** In KDE ist das Framework *Solid* für die Verarbeitung von D-Bus-Nachrichten zuständig. Unter Gnome kümmert sich Nautilus im Zusammenspiel mit PolicyKit (siehe Abschnitt 12.4) um die externen Datenträger. Die Konfiguration erfolgt in den Systemeinstellungen im Modul DETAILS • WECHSELMEDIEN.
- ▶ **D-Bus:** Zur Kommunikation zwischen den verschiedenen Ebenen des Hotplug-Systems wird das D-Bus-Kommunikationssystem (kurz D-Bus) verwendet. Auf der Basis der Bibliothek `libdbus` kann die Kommunikation direkt zwischen zwei Programmen erfolgen. Wenn Nachrichten zwischen mehreren Programmen ausgetauscht werden sollen, kommt als zentrale Vermittlungsstelle das Hintergrundprogramm `dbus-daemon` zum Einsatz. D-Bus ist außerhalb des Kernels implementiert. Pläne, das Kommunikationssystem im Rahmen des Projekts `kdbus` in den Kernel zu integrieren, sind gescheitert.

## Audio-System (ALSA)

ALSA steht für *Advanced Linux Sound Architecture* und ist im Kernel für die Ansteuerung von Soundkarten auf unterster Ebene verantwortlich. Bei vom Kernel unterstützten Audio-Controllern wird das erforderliche ALSA-Modul automatisch geladen. Die Namen aller ALSA-Module beginnen mit `snd`. Der Befehl `lsmod | grep snd` liefert daher einen raschen Überblick über alle aktiven ALSA-Module. Der Zugriff auf diverse Soundfunktionen erfolgt über Dateien im Verzeichnis `/proc/asound`. ALSA

Sie konfigurieren das ALSA-System mithilfe der Dateien `/etc/alsa/*`, `/etc/asound.conf` sowie `.asoundrc`. Beim Herunterfahren des Rechners bzw. beim nächsten Neustart speichert das Init-System die Lautstärkeinstellungen bzw. stellt sie wieder her.

Bei einer gewöhnlichen Nutzung des Audio-Systems besteht keine Notwendigkeit, die ALSA-Konfiguration zu verändern. Die Hardware-Erkennung sollte automatisch gelingen. Wer besondere Audio-Anforderungen hat (Musiker), zwischen mehreren Audio-Karten differenzieren will oder andere Sonderwünsche hat, der findet auf der folgenden Website und dem dazugehörigen Wiki umfassende Hintergrundinformationen zu ALSA und zu seiner Konfiguration:

<https://alsa-project.org>

**ALSA-Tools** Zur direkten Nutzung von ALSA stehen diverse Kommandos zur Auswahl (Paket `alsa-utils`), von denen hier die wichtigsten kurz vorgestellt werden: `alsactl` speichert bzw. lädt alle ALSA-Einstellungen, also z. B. die zuletzt eingestellte Lautstärke. `alsamixer` verändert die Lautstärke bzw. den Eingangspegel diverser ALSA-Audio-Kanäle. `aplay` spielt eine Audio-Datei ab. `arecord` nimmt eine Audio-Datei auf.

### Fehlersuche im Audio-System

Wenn die Lautsprecher still bleiben, ist die Ursache oft nur ein auf 0 gestellter Lautstärkereger. Für gewöhnliche Anwendungen sind drei Kanäle wichtig: Die Master-Lautstärke steuert die Lautstärke des Gesamtsignals. Die PCM-Lautstärke gibt an, wie laut von Audio- und Video-Playern erzeugte Audio-Daten in das Gesamtsignal eingespeist werden. (PCM steht für *Pulse Code Modulation*.) Die CD-Lautstärke gibt schließlich an, wie laut die direkt vom CD/DVD-Laufwerk kommenden Daten in das Gesamtsignal einfließen, wenn das Laufwerk und die Audio-Karte mit einem Kabel verbunden sind.

Bei modernen Distributionen fehlen bisweilen grafische Benutzeroberflächen, um die Audio-Eingänge und -Ausgänge einzeln einzustellen. Abhilfe: Starten Sie `alsamixer` in einer Textkonsole. Nun können Sie mit den Cursortasten die Kanäle auswählen und deren Pegel justieren. `M` schaltet einen Kanal ganz ein bzw. wieder aus (*mute*).

Hilfreich für erste Tests des Audio-Systems ist das ALSA-Kommando `speaker-test`. Wenn es mit den folgenden Parametern ausgeführt wird, sollte wechselweise im linken und im rechten Lautsprecher der Text *front left* bzw. *front right* zu hören sein:

```
speaker-test -t wav -c2
```

**Audio-Bibliotheken** Viele Audio-Programme verwenden ALSA nicht direkt, sondern greifen auf Soundbibliotheken, Sound-Server etc. zurück. Diese Zwischenschicht zwischen dem Low-Level-System ALSA und den eigentlichen Audio-Anwendungen soll die Programmierung vereinfachen, Audio-Anwendungen netzwerktauglich machen und die konfliktfreie Kooperation gleichzeitig laufender Audio-Programme sicherstellen.

Das Problem besteht nun darin, dass es momentan keine einheitliche Audio-Architektur oberhalb von ALSA gibt: KDE und Gnome gehen jeweils eigene Wege. Anspruchsvolle Audio-Anwendungen, für die die vorhandenen Audio-Bibliotheken unzureichend sind, implementieren elementare Audio-Funktionen selbst neu. Es ist daher schwierig, Audio-Programme zu entwickeln, die unabhängig vom Desktop-System einfach funktionieren.

Die folgenden Punkte stellen einige gängige Audio-Systeme kurz vor:

- **GStreamer:** Die GStreamer-Bibliothek ist ein umfassendes Multimedia-Framework, das von vielen Gnome-Programmen eingesetzt wird. Dank einer Plugin-Architektur ist es sehr modular und kann gut erweitert werden. Auch Codecs zur

Verarbeitung verschiedener Audio- und Video-Formate sind als Plugins verfügbar. Die GStreamer-Bibliothek enthält keinen eigenen Sound-Dämon; das Zusammenführen mehrerer Audio-Signale übernimmt direkt ALSA. Weitere Informationen finden Sie hier:

<https://gstreamer.freedesktop.org>

- ▶ **Phonon:** Das Multimedia-Fundament von KDE heißt Phonon. Die Bibliothek bietet eine einheitliche Programmierschnittstelle zur Nutzung von Audio- und Video-Funktionen, die auf vorhandene Multimedia-Bibliotheken zurückgreift (oft GStreamer oder VLC, fallweise auch Xine). Phonon wird auch von der Qt-Bibliothek als Multimedia-Schnittstelle verwendet. Weitere Details verrät die Phonon-Website:

<http://phonon.kde.org>

- ▶ **PulseAudio:** PulseAudio ist ein netzwerkfähiger Sound-Server, der von den meisten Distributionen verwendet wird. Alle Audio-Streams können mit dem Programm `pavucontrol` getrennt gesteuert und unterschiedlichen Audio-Karten bzw. -Ausgabegeräten zugewiesen werden. PulseAudio sollte auch zusätzliche Audio-Hardware (z. B. USB-Boxen) automatisch erkennen und aktivieren. Weitere Details verrät die folgende Seite:

<https://freedesktop.org/wiki/Software/PulseAudio>

Zu diesen Audio-Systemen gesellen sich diverse Programme, die selbst umfassende Audio- bzw. Multimedia-Bibliotheken enthalten und diese auch anderen Programmen zur Verfügung stellen. Ein prominentes Beispiel ist der Video-Player Xine auf Basis der `xinelib`. Man kann sich leicht ausrechnen, dass Inkompatibilitäten zwischen verschiedenen Audio-Programmen und -Bibliotheken wortwörtlich vorprogrammiert sind.

Für Musiker bzw. professionelle Audio-Anwender gibt es eigene Distributionen, die speziell in Hinblick auf die optimale und störungsfreie Nutzung der Audio-Programme zusammengestellt sind. Am populärsten ist zurzeit Ubuntu Studio (<http://ubuntustudio.org>).

## 18.9 Logging (Syslog)

Der Kernel, diverse administrative Werkzeuge (PAM, APT, dpkg) und die meisten Netzwerkdienste protokollieren Ereignisse und Fehler in zahllose Dateien in `/var/log`. Diese Logging-Dateien sind während der Inbetriebnahme eines neuen Dienstes ausgesprochen praktisch, um Konfigurationsfehler zu finden. Im laufenden Betrieb eines Servers können die Logging-Dateien Hinweise auf Sicherheitsprobleme geben.

**Syslog** Damit nicht jedes Programm eigene Logging-Funktionen implementieren muss, greifen der Kernel sowie eine Menge administrativer Werkzeuge und Server-Dienste auf zentrale Logging-Funktionen zurück, die üblicherweise als Syslog bezeichnet werden. Es gibt verschiedene Implementierungen von Syslog; die populärste ist momentan `rsyslogd`.

Allerdings nutzen nicht alle Netzwerkdienste Syslog. Insbesondere die »großen« Server-Dienste, beispielsweise Apache, CUPS, MySQL und Samba, verwenden jeweils ihre eigenen, in das Programm integrierten Logging-Funktionen. Sie entziehen sich damit der globalen Syslog-Konfiguration. Die Logging-Parameter werden vielmehr in den jeweiligen Konfigurationsdateien des Programms eingestellt.

**Journal** Aktuelle Fedora- und SUSE-Distributionen verwenden anstelle von `rsyslogd` die Journal-Funktion von `systemd` für alle Logging-Aufgaben. Im Unterschied zu `rsyslogd` verwendet das Journal ein binäres Dateiformat, das gegen nachträgliche Veränderungen geschützt ist.

Bei einigen weiteren Distributionen kommen `rsyslogd` und das Journal parallel zum Einsatz. Beispielsweise verwenden RHEL 7 bzw. CentOS 7 das Journal zur temporären Protokollierung der Kernel- und `systemd`-Nachrichten (bis zum nächsten Reboot), während `rsyslogd` weiterhin alle Syslog-Nachrichten dauerhaft aufzeichnet. Mehr Details zum Journal folgen in [Abschnitt 18.10](#), »Logging (Journal)«.

## **rsyslogd**

**Konfiguration** Bei vielen Distributionen (CentOS, Debian, RHEL, Ubuntu) werden die Syslog-Dienste durch das Programm `rsyslogd` realisiert. Dessen Konfiguration erfolgt durch die Dateien `/etc/rsyslogd.conf` und `/etc/rsyslog.d/*.conf`. Im Folgenden beschreibe ich exemplarisch die Konfiguration unter Ubuntu. Dort befinden sich die meisten Einstellungen in `/etc/rsyslog.d/50-default.conf`.

Die Syslog-Konfigurationsdateien enthalten Regeln, die aus zwei Teilen bestehen:

- ▶ **Selektor:** Der erste Teil jeder Regel gibt an, was protokolliert werden soll.
- ▶ **Aktion:** Der zweite Teil steuert, was mit der Meldung geschehen soll.

Regeln können mit dem Zeichen `\` über mehrere Zeilen verteilt werden. Es ist möglich, dass auf eine Meldung mehrere Regeln zutreffen. In diesem Fall wird die Meldung mehrfach protokolliert bzw. weitergegeben.

**Selektor** Jeder Selektor besteht aus zwei durch einen Punkt getrennten Teilen: *dienst.prioritätsstufe*. Es ist erlaubt, mehrere durch einen Strichpunkt separierte Selektoren anzugeben. Des Weiteren können in *einem* Selektor mehrere Dienste durch Kommas

getrennt werden. Alle Linux-Programme, die Syslog verwenden, müssen ihren Meldungen einen Dienst und eine Priorität zuordnen.

Syslog unterscheidet bei der Protokollierung zwischen den in [Tabelle 18.6](#) zusammengefassten Diensten (»Selektoren«). `auth` und `authpriv` gelten für Nachrichten des Authentifizierungssystems auf Benutzer- bzw. auf Systemebene, also z. B. für die Benachrichtigung über einen fehlerhaften Login. In der Praxis ist zumeist nur `authpriv` relevant. Die so protokollierten Nachrichten sind allerdings sicherheitstechnisch sehr sensibel. Die resultierenden Logging-Dateien werden in der Regel so eingerichtet, dass sie nur von `root` gelesen werden können.

| Selektor                                    | Bedeutung                                  |
|---|--|
| <code>auth</code>                           | Authentifizierung                          |
| <code>authpriv</code>                       | Authentifizierung (privilegiert)           |
| <code>daemon</code>                         | diverse Hintergrunddienste                 |
| <code>ftp</code>                            | FTP  |
| <code>kern</code>                           | Kernel-Nachrichten                         |
| <code>lpr</code>                            | Drucksystem (CUPS)                         |
| <code>mail</code>                           | Mail-Server                                |
| <code>news</code>                           | Usenet-Server                              |
| <code>syslog</code>                         | Nachrichten über den Syslog-Dienst selbst  |
| <code>user</code>                           | Default, wenn kein Selektor angegeben wird |
| <code>uucp</code>                           | UUCP (Unix to Unix Copy, veraltet)         |
| <code>local0</code> bis <code>local7</code> | zur freien Verwendung                      |
| <code>*</code>                              | gilt für alle Dienste                      |

**Tabelle 18.6** Syslog-Selektoren

Syslog kennt außerdem diese Prioritätsstufen (in steigender Wichtigkeit): `debug`, `info`, `notice`, `warning` = `warn`, `err` = `error`, `crit`, `alert` und `emerg` = `panic`. Die Schlüsselwörter `warn`, `error` und `panic` gelten als veraltet – verwenden Sie stattdessen `warning`, `err` und `emerg`. Das Zeichen `*` umfasst alle Prioritätsstufen. Das Schlüsselwort `none` gilt für Nachrichten, denen keine Priorität zugeordnet ist.

Die Angabe einer Prioritätsstufe schließt alle höheren (wichtigeren) Prioritätsstufen mit ein. Der Selektor `mail.err` umfasst also auch `crit`-, `alert`- und `emerg`-Meldungen des Mail-Systems. Wenn Sie explizit nur Nachrichten einer bestimmten Priorität wünschen, stellen Sie das Zeichen `=` voran (also etwa `mail.=err`).

**Aktion** Als Aktion wird normalerweise der Name einer Logging-Datei angegeben. Normalerweise werden Logging-Dateien nach jeder Ausgabe synchronisiert. Wenn dem Dateinamen ein Minuszeichen vorangestellt ist, verzichtet Syslog auf die Synchronisierung. Das ist wesentlich effizienter, allerdings gehen dann bei einem Absturz noch nicht physikalisch gespeicherte Meldungen verloren.

Syslog kann Nachrichten auch an FIFO-Dateien (First In First Out) oder Pipes weiterleiten. In diesem Fall stellen Sie dem Dateinamen das Zeichen | voran. Die Datei `/dev/xconsole`, die im folgenden Listing vorkommt, ist eine besondere FIFO-Datei zur Weitergabe von Meldungen an das Grafiksystem X.

Das Zeichen \* bedeutet, dass die Nachricht an alle in Konsolen bzw. via SSH eingeloggtten Benutzer gesendet wird. Da das sehr störend ist, wird diese Form des Loggings standardmäßig nur für kritische Meldungen verwendet. Weitere Details zur Syntax von `rsyslog.conf` finden Sie auf der gleichnamigen `man`-Seite.

**Beispiel** Die folgenden Zeilen geben die Syslog-Standardkonfiguration von Ubuntu leicht gekürzt und etwas übersichtlicher formatiert wieder:

```
# Datei /etc/rsyslog.d/50-default.conf bei Ubuntu (gekürzt)
# Selektor                                Aktion
auth,authpriv.*                          /var/log/auth.log
*.*;auth,authpriv.none                   -/var/log/syslog
kern.*                                    -/var/log/kern.log
mail.*                                    -/var/log/mail.log
mail.err                                  /var/log/mail.err
*.emerg                                   :omusrmsg:*
daemon.*;mail.*;\
    news.err;\
    *.=debug;*.=info;\
    *.=notice;*.=warn                    |/dev/xconsole
```

Im Klartext bedeutet die obige Konfiguration:

- ▶ `/var/log/auth` enthält Authentifizierungsmeldungen aller Prioritätsstufen. Dazu zählen gescheiterte und erfolgreiche Login-Versuche (auch via SSH), PAM-Meldungen, `sudo`-Kommandos etc. Als einzige Logging-Datei wird `auth` bei jeder Meldung sofort synchronisiert.
- ▶ `/var/log/syslog` enthält *alle* via Syslog protokollierten Meldungen (inklusive Authentifizierungsmeldungen, denen keine Priorität zugewiesen ist). Der allumfassende Ansatz ist zugleich ein Vor- und ein Nachteil. Einerseits können Sie so aus einer einzigen Datei alle erdenklichen Informationen extrahieren. Andererseits ist es in diesem Sammelsurium natürlich besonders schwierig, relevante Einträge zu finden.
- ▶ `/var/log/kern.log` enthält alle Kernelmeldungen.



- ▶ Die Nachrichten des Mail-Systems (z. B. Postfix, Dovecot, SpamAssassin) werden über zwei Dateien verteilt. In `mail.log` werden *alle* Nachrichten gespeichert, in `mail.err` nur Fehlermeldungen.
- ▶ Kritische Systemmeldungen, z. B. über einen bevorstehenden Shutdown oder über Kernelfehler, werden durch `:omusrmsg:*` an alle Benutzer weitergeleitet, genau genommen an alle Terminalfenster und Konsolen. `omusrmsg` ist ein `rsyslog`-Modul, um Nachrichten an Benutzer zu senden.
- ▶ Außerdem werden diverse Warnungen und Fehlermeldungen an das X-System weitergeleitet. Um diese Meldungen unter X zu verfolgen, starten Sie das Programm `xconsole`. Es sieht wie ein kleines Terminalfenster aus, erlaubt aber keine Eingaben.

Damit Änderungen an der Syslog-Konfiguration wirksam werden, muss der Syslog-Dienst neu gestartet werden:

```
root# systemctl restart rsyslog
```

Mit dem Kommando `logger` können Sie selbst in einem Script Syslog-Nachrichten aufzeichnen oder neue Syslog-Regeln testen. Normalerweise verwenden Sie das Kommando wie folgt:

```
user$ logger -t mydaemon -p authpriv.info "xxx has a new password"
```

Anstelle von `mydaemon` können Sie ein beliebiges Schlüsselwort angeben, das Ihnen später bei der Suche in den Logging-Dateien hilft. Mit `-p` geben Sie den Selektor an. Syslog fügt Ihrer Nachricht automatisch Zeitinformationen hinzu. In der betreffenden Logging-Datei sieht der vorhin erzeugte Eintrag dann so aus:

```
Jun 16 07:55:03 localhost mydaemon: xxx has a new password
```

Meldungen des Kernels werden in einen 16 KiB großen Ringpuffer im RAM geschrieben. Wenn dieser Puffer voll ist, werden alte Nachrichten gelöscht, um Platz für neue Nachrichten zu schaffen. Den Inhalt dieses Ringpuffers können Sie mit `dmesg` ansehen. Wenn Sie dabei die Option `-c` angeben, wird der Ringpuffer gleichzeitig geleert.

Alle Kernelnachrichten werden außerdem in die virtuelle Datei `/proc/kmsg` geschrieben. Diese Datei dient zur Weitergabe der Kernelnachrichten an Syslog.

Den `dmesg`-Kernelmeldungen ist oft eine Zeitangabe in der Form `[nnn.nnnnnn]` vorangestellt. Die Zahl vor dem Komma gibt die Anzahl der Sekunden seit dem Systemstart an, die weiteren sechs Stellen präzisieren die Zeitangabe auf millionstel Sekunden. Bei der Speicherung der Kernelmeldungen in einer Logging-Datei wird diese Zeitangabe in der Regel durch die absolute Zeit ergänzt.

logger

Kernel-Logging

**Init-Meldungen** Die Bildschirmausgaben des Init-Systems (siehe [Kapitel 24](#)) werden bei manchen Distributionen in der Datei `/var/log/boot.log` aufgezeichnet.

Die meisten modernen Distributionen verwenden `systemd` als Init-System. Alle `systemd`-Ausgaben werden dann außerdem im Journal protokolliert (siehe [Abschnitt 18.10](#)). Diese Ausgaben sind detaillierter, aber oft auch unübersichtlicher als die herkömmliche `boot.log`-Datei.

## logrotate

Logging-Dateien werden nach und nach immer größer. Um den Speicherbedarf der Logging-Dateien unter Kontrolle zu behalten, bietet sich `logrotate` an. Dieses Programm wird bei vielen Distributionen einmal täglich durch das Cron-Script `/etc/cron.daily/logrotate` aufgerufen. Es verarbeitet dann alle Logging-Dateien, die in den Konfigurationsdateien in `/etc/logrotate.d` beschrieben sind. Wie `logrotate` mit den Logging-Dateien umgeht, hängt im Detail von der jeweiligen Programmkonfiguration ab. Die prinzipielle Vorgehensweise ist aber immer dieselbe und sieht so aus:

- ▶ `Logrotate` benennt die aktuelle Logging-Datei um. Aus `name` wird `name.0`.
- ▶ `Logrotate` erzeugt eine neue, leere Logging-Datei `name`.
- ▶ Bei vielen Server-Diensten fordert `Logrotate` den Dämon durch `service name reload` dazu auf, die Konfiguration neu einzulesen. Bei dieser Gelegenheit erkennt der Dämon, dass es eine neue, leere Logging-Datei gibt, und verwendet nun diese.
- ▶ `Logrotate` komprimiert `name.0` oder `name.1` (Option `delaycompress`). `delaycompress` vermeidet Konflikte zwischen dem Dämon, der vielleicht noch in `name.0` schreibt, und dem Komprimierkommando.
- ▶ `Logrotate` benennt bereits vorhandene Logging-Archive um. Aus `name.4.gz` wird `name.5.gz`, aus `name.3.gz` wird `name.4.gz` etc. Dieser Vorgang wird »rotieren« genannt und gibt dem Paket seinen Namen.
- ▶ Wenn es mehr als eine vorgegebene Maximalanzahl von Logging-Archiven gibt, werden die ältesten Archivdateien gelöscht.

**Konfiguration** `/etc/logrotate.conf` enthält einige Defaulteinstellungen für `Logrotate`. Diese Einstellungen gelten nur, soweit die programmspezifischen Konfigurationsdateien keine abweichenden Daten enthalten.

`/etc/logrotate.d` enthält Detailsinstellungen zu diversen Programmen, die Logging-Dateien produzieren. Diese Dateien stammen nicht aus dem `Logrotate`-Paket, sondern aus den Paketen des jeweiligen Programms. Das `samba`-Paket stellt also beispielsweise `/etc/logrotate.d/samba` zur Verfügung. Das stellt sicher, dass die Dateien zur jeweils installierten Programmversion passen und dass `Logrotate` den jeweiligen Server-Dienst über das Umbenennen der Logging-Dateien informiert bzw. neu startet.

Die folgenden Zeilen zeigen als Beispiel die logrotate-Konfiguration für Apache unter Ubuntu: Logrotate bearbeitet die Logging-Dateien täglich. Die Logging-Dateien werden umbenannt und komprimiert. Dabei werden auch die Zugriffsrechte neu eingestellt, sodass die Logging-Dateien nur noch von root sowie von Mitgliedern der adm-Gruppe gelesen werden können. Das Archiv ist auf 14 Dateien limitiert, d. h., Sie können bei Bedarf nur auf die Logging-Daten der letzten zwei Wochen zurückgreifen. Sofern Apache läuft, wird er durch reload davon informiert, dass es neue Logging-Dateien gibt.

```
# Datei /etc/logrotate.d/apache2 (Ubuntu 17.04)
/var/log/apache2/*.log {
    daily
    missingok
    rotate 14
    compress
    delaycompress
    notifempty
    create 640 root adm
    sharedscripts
    postrotate
        if /etc/init.d/apache2 status > /dev/null ; then \
            /etc/init.d/apache2 reload > /dev/null; \
        fi;
    endscript
    prerotate
        if [ -d /etc/logrotate.d/httpd-prerotate ]; then \
            run-parts /etc/logrotate.d/httpd-prerotate; \
        fi; \
    endscript
}
```

## logwatch

Das manuelle Lesen der Logging-Dateien mag die ersten Tage oder während der Suche nach einem Fehler ganz interessant sein. Nach kurzer Zeit werden Sie aber wie jeder andere Server-Administrator die Lust am Logging-Studium verlieren und Ihre Logging-Dateien mehr und mehr vernachlässigen. Automatisierte Tools zur Logging-Auswertung sollen in diesem Fall Abhilfe schaffen. Exemplarisch stelle ich im Folgenden das Programm logwatch vor, das bei den meisten Distributionen als Paket mitgeliefert wird.

Nach der Installation wird Logwatch einmal täglich durch `/etc/cron.daily/*logwatch` ausgeführt. Es wertet die Einträge der Logging-Dateien für die letzten 24 Stunden aus, fasst die relevanten Informationen in einer E-Mail zusammen und versendet diese an root.

Die Logging-Zusammenfassung ist üblicherweise etliche Seiten lang – eigentlich zu lang, um sie täglich vollständig zu lesen. Der Textumfang hängt davon ab, wie viele Server-Dienste installiert sind und wie viele Ereignisse während des letzten Tages protokolliert worden sind. Die folgenden Zeilen zeigen eine aus Platzgründen stark gekürzte Logwatch-E-Mail:

```
Logwatch 7.4.0
Date Range Processed: yesterday

----- Dovecot Begin -----
Dovecot IMAP and POP3 Successful Logins: 159
Dovecot disconnects: 146

----- fail2ban-messages Begin -----
Banned services with Fail2Ban: Bans:Unbans
  ssh:                               [145:145]

----- httpd Begin -----
Connection attempts using mod_proxy:
  111.248.98.148 -> vip163mx01.mxmail.netease.com:25: 1 Time(s)

A total of 65 sites probed the server
  123.125.71.110
  123.125.71.112
  123.125.71.114
  ...

----- pam_unix Begin -----
dovecot:
  Authentication Failures:
    info@xxxxx: 3 Time(s)
    accounts@xxxxx: 1 Time(s)
    advent@xxxxx: 1 Time(s)
    backup@xxxxx: 1 Time(s)
    ...

sshd:
  Authentication Failures:
    root (43.229.53.51): 368 Time(s)
    unknown (gw2.tibc.ru): 257 Time(s)
    root (218.65.30.61): 32 Time(s)
    ...

----- Disk Space Begin -----
Filesystem      Size  Used Avail Use% Mounted on
/dev/md1        488M   69M  394M  15% /boot
/dev/md2        1.8T   71G  1.7T   5% /
```

### Logging-Updates per E-Mail

Logwatch setzt voraus, dass auf Ihrem Rechner ein E-Mail-Server läuft, um die Logging-Zusammenfassung zu versenden!

Wenn unter Debian oder Ubuntu während der Logwatch-Installation Postfix installiert wird und Sie nur lokale E-Mails verarbeiten möchten, also keinen richtigen E-Mail-Server einrichten wollen, wählen Sie die Konfigurationsoption `NUR LOKAL`. Standardmäßig werden `root`-E-Mails an den Standardbenutzer weitergeleitet, der in der Datei `/etc/aliases` angegeben ist. Lokale E-Mails lesen Sie am einfachsten mit `Mutt`. Details zur Mail-Server-Konfiguration folgen in [Kapitel 34](#), »Postfix und Dovecot«.

Vielleicht wundern Sie sich, warum Logwatch wie von Zauberhand ohne jede Konfiguration funktioniert. Der Grund ist einfach: Zusammen mit Logwatch wird eine Standardkonfiguration in das Verzeichnis `/usr/share/logwatch/default.conf` installiert. Die dort befindliche Datei `logwatch.conf` enthält einige globale Grundeinstellungen. Dort ist beispielsweise eingestellt, an wen die Zusammenfassung gesendet wird (`MailTo = root`), für welchen Zeitraum die Logging-Dateien ausgewertet werden sollen (`Range = yesterday`), wie detailliert die Zusammenfassung sein soll (`Detail = Low`) etc.

Konfiguration

Außerdem enthalten die Dateien `default.conf/services/*.conf` Konfigurationseinstellungen für zahlreiche Server-Dienste, z. B. für Apache, Postfix, ClamAV, Dovecot, Sendmail, SSH etc. Diese Konfigurationsdateien werden natürlich nur wirksam, sofern die betreffenden Dienste tatsächlich laufen. Relevante Ergebnisse erhalten Sie zudem nur, wenn Sie die Orte der Standard-Logging-Dateien nicht verändert haben.

Das Verzeichnis `/usr/share/logwatch/dist.conf` ist für distributionsspezifische Änderungen gegenüber der Standardkonfiguration vorgesehen. Hier durchgeführte Einstellungen haben Vorrang gegenüber der Standardkonfiguration.

Um die Konfiguration selbst zu ändern, kopieren Sie die betreffende Datei in das entsprechende Verzeichnis in `/etc/watchlog` und modifizieren sie dort. Es reicht aus, wenn diese Datei nur die Änderungen gegenüber dem Original enthält. Zum Ausprobieren führen Sie anschließend Logwatch manuell aus:

```
root# logwatch --mailto name@host
```

## 18.10 Logging (Journal)

Das Init-System `systemd`, das in [Abschnitt 24.1](#) beschrieben wird, enthält eigene Logging-Funktionen, das sogenannte »Journal«. Für die Protokollierung ist der Hintergrundprozess `systemd-journald` verantwortlich. Im Vergleich zu traditionellen Syslog-Diensten gibt es drei fundamentale große Unterschiede:

- ▶ Das Journal wird in einem binären Format gespeichert. Das spart Platz und somit Zeit. Gleichzeitig ist dieses binäre Format aber auch der größte Nachteil des Journals: Unzählige Tools setzen voraus, dass die Logging-Dateien im Textformat vorliegen und unkompliziert mit `grep` ausgewertet werden können.
- ▶ Das Journal ist gegen nachträgliche Änderungen geschützt. Damit ist es für einen Einbrecher unmöglich, seine Spuren zu beseitigen:  
*<https://plus.google.com/115547683951727699051/posts/g1E6AxVKtyc>*
- ▶ Das gesamte Journal wird an einem Ort gespeichert. Die bei Syslog übliche Trennung in mehrere Logging-Dateien entfällt (und damit auch die ganze entsprechende Konfiguration). Um bestimmte Nachrichten aus dem Journal zu filtern, müssen Sie entsprechende Optionen an `journalctl` übergeben. (Details dazu folgen gleich.)

Davon abgesehen ist das Journal aber Syslog-kompatibel. Alle Dienste, die bisher Syslog zum Protokollieren verwendet haben, können ohne Änderungen das Journal nutzen. Auch das im vorigen Abschnitt beschriebene Kommando `logger` kooperiert problemlos mit dem Journal.

Mit der zunehmenden Verbreitung von `systemd` als Init-System ersetzt das Journal auf immer mehr Distributionen `rsyslogd`. Beachten Sie aber, dass das Journal nicht zwingend auf jeder `systemd`-Distribution aktiviert werden muss. Es besteht auch die Möglichkeit, `rsyslogd` und das Journal parallel auszuführen. Je nach Konfiguration werden dann gewisse Daten einfach doppelt protokolliert. Dies ist z. B. bei aktuellen Versionen von CentOS, Debian, RHEL und Ubuntu der Fall. Nur in Fedora und openSUSE hat das Journal `rsyslogd` schon vollständig ersetzt.

**Logging-Dateien** Wenn das Verzeichnis `/var/log/journal` existiert, speichert das Journal seine Protokolle dort. Existiert dieses Verzeichnis hingegen nicht, kommt `/run/log/journal` als Speicherort zur Anwendung. Auf den ersten Blick scheint das kein großer Unterschied zu sein, allerdings ist `/run` üblicherweise ein temporäres Dateisystem. Dort gespeicherte Protokolle gehen daher mit jedem Neustart verloren!

**journalctl** Die binären Protokolldateien werden mit dem Kommando `journalctl` ausgelesen. Standardmäßig können Sie damit wie mit `less` durch *alle* protokollierten Nachrichten blättern, wobei die älteste Nachricht zuerst angezeigt wird. Wie bei `less` können Sie mit `>` zum Ende des Protokolls, also zur neuesten Nachricht springen.

Mithilfe von Optionen können Sie die von `journalctl` präsentierten Nachrichten einschränken:

- ▶ `-b` zeigt nur die Nachrichten seit dem letzten Neustart des Rechners.
- ▶ `-f` startet `journalctl` im Dauerbetrieb, wobei ständig die gerade eintreffenden Nachrichten angezeigt werden. `Strg+C` beendet das Kommando.
- ▶ `-k` zeigt nur Kernelnachrichten.

- ▶ `-n N` zeigt nur die letzten `N` Zeilen.
- ▶ `-t suchbegriff` zeigt nur Nachrichten für das angegebene Syslog-Stichwort (Tag, wie bei `logger -t`).
- ▶ `-u name` zeigt nur Nachrichten für den angegebenen `systemd`-Dienst (Unit, z. B. `avahi-daemon`).

Das folgende Kommando filtert aus allen Logging-Nachrichten seit dem letzten Reboot jene heraus, in denen der Suchbegriff `systemd` vorkommt:

```
root# journalctl -b | grep systemd
...
Jun 06 ... systemd[1]: Started Network Manager Script Dispatcher Service.
Jun 06 ... systemd[1]: squid.service: Unit cannot be reloaded because it is
        inactive.
Jun 06 ... systemd[1]: Starting dnf makecache...
Jun 06 ... systemd[1]: Started dnf makecache.
```

Das Journal wird durch `/etc/systemd/journal.conf` sowie durch die folgenden Konfigurationsdateien gesteuert: Konfiguration

```
/etc/systemd/journald.conf.d/*.conf
/run/systemd/journald.conf.d/*.conf
/usr/lib/systemd/journald.conf.d/*.conf
```

Details zu den dort zulässigen Einstellungen können Sie mit `man journald.conf` nachlesen. Ich konzentriere mich hier auf einige ausgewählte Parameter:

- ▶ `ForwardToSyslog` gibt an, ob durch das Journal protokollierte Nachrichten auch an einen traditionellen Syslog-Dienst weitergegeben werden sollen.
- ▶ `MaxFileSec` gibt an, nach welcher Zeit spätestens eine neue Logging-Datei gestartet werden soll. Die Defaulteinstellung sieht einen Monat vor. Diese Einstellung ist nur relevant, wenn die Logging-Dateien langsamer wachsen, als die Limits `SystemMaxUse`, `SystemMaxFileSize` und `SystemKeepFree` vorgeben.
- ▶ `MaxLevelStore` gibt die Prioritätsstufe an, bis zu der Nachrichten im Journal gespeichert werden. Die Defaulteinstellung lautet `debug`.
- ▶ `SystemMaxUse` gibt an, wie viel Prozent des Dateisystems die Logging-Dateien maximal beanspruchen können. Bevor dieses Limit überschritten wird, werden alte Logging-Dateien gelöscht. Die Defaulteinstellung beträgt 10 Prozent.
- ▶ `SystemMaxFileSize` gibt die maximale Größe einer Logging-Datei an. Die Defaulteinstellung beträgt ein Achtel von `SystemMaxUse`. Das führt zu einem automatischen »Rotating«, wobei neben der aktuellen Logging-Datei maximal sieben ältere Dateien entstehen.
- ▶ `SystemKeepFree` gibt an, wie viel Prozent des Dateisystems frei bleiben müssen. Die Defaulteinstellung beträgt 15 Prozent.





# Kapitel 19

## Software- und Paketverwaltung

Dieses Kapitel beschreibt, wie Sie unter Linux Software bzw. Pakete installieren und aktualisieren und welche Techniken dabei zum Einsatz kommen. Zentrale Themen dieses Kapitels sind die Paketformate RPM und DEB, die Paketverwaltungskommandos `apt-get`, `aptitude`, `dnf`, `dpkg`, `rpm`, `yum` und `zypper` sowie schließlich die zwei neuen Paketsysteme Snap und Flatpak.

### 19.1 Einführung

Unter Windows ist es üblich, neue Programme aus einer MSI-Datei (Microsoft Installer) oder durch das Ausführen von `setup.exe` zu installieren. Das Setup-Paket enthält alle Dateien, die für das Programm erforderlich sind.

Linux verfolgt einen ganz anderen Ansatz: Mit einem Paketverwaltungssystem wird eine Datenbank verwaltet, die Informationen über alle bereits installierten Software-Pakete enthält. Neue Programme werden durch die Kommandos des Paketverwaltungssystems installiert und dabei von zentralen Paketquellen aus dem Internet heruntergeladen. Dieses Konzept hat eine Menge Vorteile:

- ▶ Größere Programme können in voneinander unabhängige Pakete mit Bibliotheken und Sprachdateien (Lokalisierung) aufgeteilt werden. Das vermindert die Redundanz und ermöglicht es, nur die wirklich erforderlichen Komponenten zu installieren.
- ▶ Die Paketverwaltung berücksichtigt Abhängigkeiten und Konflikte zwischen Software-Paketen. Wenn beispielsweise ein Programm A die Bibliothek B voraussetzt, lässt das Paketverwaltungssystem die Installation von A erst zu, nachdem B installiert worden ist.
- ▶ Anhand der Paketdatenbank lässt sich jederzeit nachvollziehen, zu welchem Paket eine bestimmte Datei gehört und ob sich diese Datei noch im ursprünglichen Zustand befindet.

- ▶ Alle auf einem Rechner installierten Pakete können gemeinsam aktualisiert werden. Dieses zentrale Update-System ist der vermutlich größte Vorteil im Vergleich zu Microsoft Windows, wo nahezu jedes größere Programm sein eigenes Update-Programm pflegt.

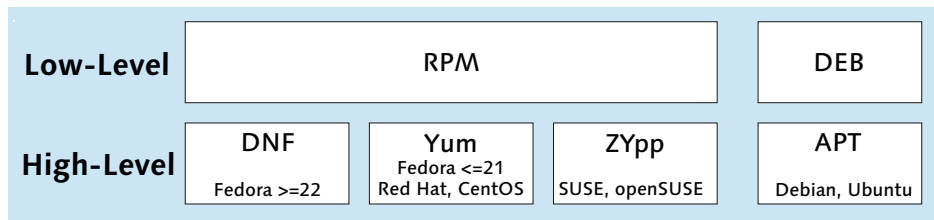
**Paketformate** Zwei Paketverwaltungssysteme dominieren den Linux-Markt:

- ▶ **RPM:** Red Hat, Fedora, Mandriva, SUSE sowie zahllose weitere Distributionen verwenden das von Red Hat entwickelte Paketformat RPM.
- ▶ **DEB:** Debian und alle davon abgeleiteten Distributionen nutzen dagegen das Paketformat DEB.

Die Kommandos zur Installation, Deinstallation und zum Update dieser Pakete (`rpm`, `dpkg` etc.) sind allerdings relativ primitiv. Sie können weder Pakete aus Paketquellen herunterladen noch Paketabhängigkeiten auflösen.

**Paket-  
verwaltungs-  
systeme**

Deswegen entstanden aufbauend auf `rpm` bzw. `dpkg` neue Paketverwaltungssysteme mit einer Menge Zusatzfunktionen. Dazu zählen die automatische Installation abhängiger Pakete, die Durchführung von Updates für das gesamte System und die Berücksichtigung von Paketquellen aus dem Internet. Beispiele für derartige Paketverwaltungssysteme sind DNF, Yum und ZYpp für RPM-Pakete sowie APT und Aptitude für DEB-Pakete (siehe [Abbildung 19.1](#)).



**Abbildung 19.1** Low- und High-Level-Paketverwaltungssysteme

**Distributions-  
spezifische  
Werkzeuge**

Ergänzend zu diesen Standardprogrammen gibt es bei manchen Distributionen bzw. Desktop-Systemen eigene Programme zur Paketverwaltung und zur Durchführung von Updates:

|                       |  |
|-----------------------|--|
| Gnome:                | Software ( <code>gnome-software</code> ) |
| KDE:                  | Discover sowie diverse andere Programme  |
| CentOS, Fedora, RHEL: | PackageKit                               |
| Debian, Ubuntu:       | <code>update-manager</code>              |
| SUSE:                 | YaST-Module der Gruppe SOFTWARE          |

Nicht nur die Paketverwaltungswerkzeuge unterscheiden sich von Distribution zu Distribution, auch sonst gibt es trotz gemeinsamer Standards viele distributionspezifische Eigenheiten. Diese sind in [Abschnitt 19.13](#) zusammengefasst.

#### Vermeiden Sie es, Pakete unterschiedlicher Distributionen zu mischen

Die Pakete einer Linux-Distribution sind aufeinander abgestimmt. Das bedeutet, dass sie einheitliche Bibliotheken nutzen, mit demselben Compiler kompiliert wurden etc. Als Linux-Einsteiger sind Sie deshalb gut beraten, nur Pakete zu installieren, die für Ihre Distribution gedacht sind. Nicht zu empfehlen ist die Installation eines Red-Hat-Pakets unter SUSE (oder umgekehrt). Die dabei auftretenden Probleme, wie fehlende Bibliotheken oder nicht erfüllte Paketabhängigkeiten, lassen sich – wenn überhaupt – nur von Linux-Profis beheben.

Wenn Sie für 50, 100 oder 1000 Linux-Rechner verantwortlich sind, wird die Administration und Paketverwaltung trotz der in diesem Kapitel vorgestellten Werkzeuge zur Qual. Sie benötigen ein Werkzeug, um zentral auf allen oder auf zuvor ausgewählten Rechnern ein Update durchzuführen, ein neues Programm zu installieren oder die Konfiguration zu verändern. Je nach Distribution bieten sich hierfür Red Hat Network, ZENworks (SUSE), m23 (diverse Distributionen) oder Landscape (Ubuntu) an:

**Zentrale  
Administration**

<https://www.microfocus.com/de-de/products/zenworks>

<http://m23.sourceforge.net>

<https://landscape.canonical.com>

Das Kommando `tar` hilft dabei, eine Menge Dateien zu einem Archiv zusammenzufassen bzw. ein derartiges Archiv wieder auszupacken. In den Anfangszeiten von Linux, noch bevor es die Paketformate RPM und DEB gab, verwendeten die meisten Distributionen `tar`-Archive wie Pakete. Selbst heute gibt es noch Distributionen, die `tar`-Archive als Paketersatz verwenden, z. B. Slackware.

**tar-Pakete**

Aber auch für alle anderen Distributionen spielen `tar`-Archive eine gewisse Rolle im Alltag fortgeschrittener Linux-Anwender: Viele Software-Entwickler, die keine Lust dazu haben, RPM- oder DEB-Pakete zu erzeugen, stellen stattdessen einfache `tar`-Archive mit allen erforderlichen Dateien zur Verfügung. Das Archiv muss lediglich am richtigen Ort mit `tar xzf name.tgz` ausgepackt werden – schon ist das Programm installiert. Allerdings führt `tar` an der Paketverwaltung Ihrer Distribution vorbei! Ein so installiertes Programm lässt sich schwer aktualisieren, nicht deinstallieren und kann Konflikte verursachen. Die Installation von `tar`-Paketen ist daher nur zu empfehlen, wenn Sie wissen, was Sie tun, und wenn das gewünschte Programm in keiner anderen Form verfügbar ist.

## Nachteile der traditionellen Paketverwaltung

Die Paketverwaltung gängiger Distributionen funktioniert gut, ist aber auch mit Nachteilen verbunden:

- ▶ Die bei vielen Distributionen beinahe täglichen Updates verunsichern Anwender, die von Windows oder macOS größere Update-Zeitspannen gewohnt sind.
- ▶ Der Download-Bedarf für die Updates ist groß. Mehrere Hundert MiB pro Monat sind für eine Desktop-Distribution nicht unüblich.
- ▶ Viele Linux-Anwender würden eine Distribution gerne länger verwenden, dabei aber einige wenige Programme aktualisieren – oft Desktop-Anwendungen wie LibreOffice oder GIMP. Genau das machen gängige Distributionen so gut wie unmöglich. Es werden zwar Sicherheits-Updates angeboten, nicht aber grundlegende neue Versionen.

Wenn Sie in einer älteren Distribution die neueste Version von LibreOffice nutzen möchten, müssen Sie eine manuelle Installation durchführen oder auf nicht offizielle Paketquellen zurückgreifen. Beides ist nur fortgeschrittenen Benutzern zu empfehlen. Unter Windows oder macOS ist es ungleich einfacher, die gerade aktuelle LibreOffice-Version zu installieren.

Es gibt technische Gründe, warum das so ist: Die meisten Linux-Programme verwenden unzählige Bibliotheken. Ein Versions-Update von LibreOffice setzt voraus, dass auch einige Bibliotheken aktualisiert werden müssen. Das kann wiederum Inkompatibilitäten mit anderen Programmen auslösen, die ebenfalls auf diese Bibliothek zurückgreifen.

Ein möglicher Ausweg besteht darin, bei wichtigen Programmen die dazugehörigen Bibliotheken zu integrieren. Google Chrome hat diesen Weg von Anfang an beschritten, und auch die Firefox- und Thunderbird-Pakete werden mittlerweile so gewartet. Aber auch diese Vorgehensweise ist mit Nachteilen verbunden: Aufgrund der nun unvermeidlichen Redundanzen steigen der Platzbedarf auf der Festplatte, das Download-Volumen bei jedem Update und der RAM-Bedarf bei der gleichzeitigen Ausführung mehrerer Programme. Tritt in einer Bibliothek ein Sicherheitsproblem auf, kann es nicht mehr zentral behoben werden. Vielmehr müssen alle Programme, die diese Bibliothek verwenden, aktualisiert werden.

## Die Zukunft der Paketverwaltung

Aus heutiger Sicht erscheint es ziemlich sicher, dass uns RPM- und DEB-Pakete noch eine Weile als Basis für die Paketverwaltung gewöhnlicher Distributionen erhalten bleiben. Parallel dazu etablieren sich aber zunehmend alternative Konzepte. Sie sind einerseits dazu gedacht, RPM/DEB zu ergänzen und deren Schwachstellen zu

kompensieren; sie können aber andererseits in Spezialfällen die herkömmliche Paketverwaltung ganz ersetzen.

Das von Red Hat bzw. Fedora favorisierte *Flatpak* und das von Canonical/Ubuntu entwickelte *Snap* erleichtern es, aktuelle Programme distributionsunabhängig in aktuellen Versionen anzubieten. Beide Paketsysteme agieren parallel zur herkömmlichen Paketverwaltung. Aus Anwendersicht erleichtern sie die Installation von (Desktop-)Programmen abseits des offiziellen Paketangebots. Gleichzeitig mindert ein Sandbox-System mögliche Sicherheitsprobleme. Eine ausführliche Beschreibung dieser beiden Paketsysteme folgt in [Abschnitt 19.12](#).

Flatpak und Snap

Ähnliche Ziele wie Flatpak und Snap verfolgen die Projekte *Zero Install* und *AppImage*. Insbesondere AppImage wirkt ausgereift und hat sogar den Segen des sonst so kritischen Kernelentwicklers Linus Torvalds erhalten: »This is just very cool.« Ohne die Unterstützung großer Distributionen ist es aber unwahrscheinlich, dass sich diese Systeme durchsetzen können.

Zero Install und AppImage

<http://Oinstall.net>

<http://appimage.org>

## 19.2 RPM-Paketverwaltung

Das Kommando `rpm` installiert und verwaltet RPM-Pakete. Es hilft dabei,

- ▶ im Rahmen einer Installation automatisch Änderungen in schon vorhandenen Dateien durchzuführen (etwa in Script-Dateien).
- ▶ ein Programm durch eine aktuellere Version zu ersetzen, wobei von geänderten Dateien automatisch Backups erstellt werden.
- ▶ alle Dateien eines Programms wieder zu entfernen.
- ▶ sicherzustellen, dass vor der Installation eines Programms alle Voraussetzungen erfüllt sind – dass also alle erforderlichen Bibliotheken in der richtigen Version zur Verfügung stehen.
- ▶ zu überprüfen, ob eine Datei seit der Installation des Pakets verändert wurde.
- ▶ festzustellen, zu welchem Paket eine bestimmte Datei gehört.

Die erforderlichen Verwaltungsinformationen befinden sich in jedem RPM-Paket. Bei der Installation werden diese Informationen in eine Datenbank eingetragen, deren Dateien sich im Verzeichnis `/var/lib/rpm` befinden.

## Grundlagen

Die meisten RPM-Pakete werden in zwei Varianten zur Verfügung gestellt: als Binärpaket und als Quellcodepaket. Das Binärpaket enthält die zur Ausführung des Programms notwendigen Dateien. Das Quellcodepaket ist allerdings nur für Entwickler interessant. Es enthält den Quellcode, der erforderlich war, um das Binärpaket zusammenzustellen.

Der Paketname enthält ziemlich viele Informationen: `abc-2.0.7-1.x86_64.rpm` bezeichnet beispielsweise das Paket `abc` mit der Versionsnummer `2.0.7` und der Release-Nummer `1`. Falls bei der Zusammenstellung eines Pakets ein Fehler aufgetreten ist, zusätzliche Online-Dokumentation beigefügt wurde oder andere Änderungen durchgeführt wurden, entstehen Release-Ziffern größer als `1` für eine bestimmte Versionsnummer. Die Versionsnummer bezieht sich also auf das eigentliche Programm, die Release-Nummer auf die `rpm`-Zusammenstellung.

Die Kennung `x86_64` weist darauf hin, dass das Paket Binärdateien für Intel/AMD-kompatible 64-Bit-Prozessoren enthält. Wenn das Paket `abc` Script- oder Textdateien enthält, die von der CPU-Architektur unabhängig sind, wird statt der CPU-Kennung das Kürzel `noarch` verwendet. Wenn das Paket den Quellcode enthält, ist stattdessen das Kürzel `src` üblich.

**Metadaten** Die Paketdatei enthält neben den zu installierenden Dateien zahlreiche Verwaltungsinformationen: eine kurze Paketbeschreibung, abermals Informationen über Versionsnummern, die Einordnung in die Gruppenhierarchie, Abhängigkeiten von anderen Paketen etc. Abhängigkeiten bestehen dann, wenn ein Paket eine bestimmte Programmiersprache, wie Perl, oder eine bestimmte Library voraussetzt. In diesem Fall müssen zuerst diese Pakete installiert werden.

`rpm` verwaltet eine Datenbank mit Informationen über alle installierten Binärpakete. Diese Datenbank wird in diversen Dateien im Verzeichnis `/var/lib/rpm` gespeichert. Die Datenbank enthält nur Informationen zu Binärpaketen. Eventuell auch installierte Pakete mit Quellcode werden nicht in die Datenbank aufgenommen.

Damit die RPM-Datenbank mit der tatsächlichen Installation übereinstimmt, dürfen Pakete nicht einfach durch Löschen der Dateien, sondern müssen durch ein Deinstallieren (`rpm -e`) entfernt werden!

**Delta-RPM-Pakete** Um ein RPM-Paket zu aktualisieren, wird oft das gesamte neue Paket heruntergeladen. Gerade bei Sicherheits-Updates, bei denen oft nur winzige Änderungen an wenigen Dateien erforderlich sind, ist das ineffizient. Aus diesem Grund gibt es Delta-RPM-Pakete, die nur die Änderungen gegenüber einer bestimmten Version des Pakets enthalten.

Bei der Anwendung von Delta-RPMs erzeugt das Kommando `applydeltarpm` aus dem Delta-RPM und dem Original-Paket bzw. dessen installierten Dateien das neue, aktualisierte RPM-Paket. Dieses wird dann ganz normal installiert. `applydeltarpm` ist Teil des Pakets `deltarpm`.

`applydeltarpm` setzt voraus, dass momentan eine ganz bestimmte Version des Pakets installiert ist. Ist das nicht der Fall bzw. wurden deren Dateien nach der Installation verändert, ist zur Durchführung des Updates die Original-RPM-Datei erforderlich.

Bei 64-Bit-Distributionen kann es vorkommen, dass `rpm -qi name` trotz eines eindeutigen Paketnamens Informationen zu *zwei* Paketen auflistet. Das ist kein Fehler; vielmehr handelt es sich um zwei gleichnamige Pakete mit den Dateien der 32- und der 64-Bit-Variante eines Programms bzw. einer Bibliothek.

32/64-Bit-Probleme

SUSE vermeidet gleichnamige Pakete mit unterschiedlichem Inhalt, indem es die 32-Bit-Varianten im Paketnamen mit dem Anhang `32bit` kennzeichnet. `rpm -qa | grep 32bit` liefert dort eine Liste aller 32-Bit-Pakete, die aus Kompatibilitätsgründen erforderlich sind.

### RPM-Datenbank reparieren

In seltenen Fällen passiert es, dass die RPM-Datenbank inkonsistente Daten enthält. Das äußert sich darin, dass das `rpm`-Kommando nicht mehr verwendet werden kann bzw. Fehlermeldungen wie *cannot open packages database* liefert. Abhilfe schaffen meistens die folgenden Kommandos:

```
root# rm -f /var/lib/rpm/__db*
root# db_verify /var/lib/rpm/Packages
root# rpm --rebuilddb
root# yum/dnf clean all
```

Damit wird die RPM-Datenbank neu erzeugt. Das dauert allerdings eine Weile.

### Das rpm-Kommando

Es mag auf den ersten Blick überraschend wirken, aber Sie werden mit dem `rpm`-Kommando selten ein Paket installieren oder wieder entfernen. Dazu setzen Sie in aller Regel `dnf`, `yum`, `zypper` oder eine grafische Benutzeroberfläche ein, und `rpm` kommt nur hinter den Kulissen zum Einsatz.

Der praktische Nutzen des `rpm`-Kommandos besteht heute primär darin, die Paketdatenbank auszulesen und daraus Informationen zu extrahieren, die `yum` oder `zypper` Ihnen gar nicht oder nur viel umständlicher gibt. [Tabelle 19.1](#) fasst die wichtigsten `rpm`-Kommandos zusammen. Die folgenden Beispiele zeigen die praktische Anwendung.

| Aufgabe   | Kommando                         |
|---|----------------------------------|
| Paket installieren  | <code>rpm -i datei.rpm</code>    |
| Paket aktualisieren   | <code>rpm -U datei.rpm</code>    |
| Paketinstallation überprüfen (verify)                           | <code>rpm -V datei.rpm</code>    |
| Paket entfernen   | <code>rpm -e paketname</code>    |
| Alle installierten Pakete ermitteln                             | <code>rpm -qa</code>             |
| Paket ermitteln, das diese Datei zur Verfügung stellt           | <code>rpm -qf datei</code>       |
| Paketbeschreibung anzeigen                                      | <code>rpm -qi paketname</code>   |
| Liste aller Dateien des Pakets ermitteln                        | <code>rpm -ql paketname</code>   |
| Liste aller Konfigurationsdateien des Pakets ermitteln          | <code>rpm -qc paketname</code>   |
| Informationen zu einem noch nicht installierten Paket ermitteln | <code>rpm -qpli datei.rpm</code> |

Tabelle 19.1 Wichtige rpm-Kommandos

**Beispiele** Nehmen Sie an, Sie entdecken im `/etc`-Verzeichnis eine Datei, die Ihnen bisher noch nie aufgefallen ist und von der Sie wissen möchten, welchen Zweck sie hat. `rpm -qf` verrät, zu welchem Paket sie gehört. `rpm -qi` liefert eine kurze Beschreibung des Pakets, und `rpm -ql` zeigt alle anderen Dateien, die aus diesem Paket stammen:

```
user$ rpm -qf /etc/login.defs
shadow-utils-4.2.1-2.fc23.x86_64
user$ rpm -qi shadow-utils
Name      : shadow-utils
Summary   : Utilities for managing accounts and shadow password files
...
user$ rpm -ql shadow-utils
/etc/default/useradd
/etc/login.defs
/usr/bin/chage
...
```

Vielleicht möchten Sie wissen, welche perl-Pakete installiert sind. `rpm -qa` liefert eine Liste aller installierten Pakete. Mit `grep` filtern Sie daraus die interessanten Pakete heraus; `sort` sortiert die Liste:

```
user$ rpm -qa | grep perl | sort
perl-5.22.0-349.fc23.x86_64
perl-Carp-1.36-346.fc23.noarch
perl-constant-1.33-347.fc23.noarch
...
```

Perl macht Schwierigkeiten, und Sie sind sich nicht sicher, ob das Perl-Paket korrekt installiert ist. Sind noch alle installierten Dateien dieses Pakets im Originalzustand?



Die Antwort gibt `rpm -V`. Es listet alle Dateien auf, die sich geändert haben. In der Regel sollte das Ergebnis wie im folgenden Beispiel leer sein oder nur Konfigurationsdateien enthalten.

```
user$ rpm -V perl
```

Unter Fedora oder RHEL können Sie das Paket nun mit `yum reinstall perl` reparieren:

```
root# yum reinstall perl
```

## 19.3 Yum

Yum ist ein Programm, das die Verwaltung von RPM-Paketen vereinfacht. Es wird von vielen RPM-basierten Kommandos zur Paketverwaltung verwendet, z. B. von CentOS, RHEL sowie von älteren Fedora-Versionen. Yum bietet im Vergleich zu `rpm` eine Menge Zusatzfunktionen:

- ▶ Als Datenquelle (*Repository*) dienen Yum-Archive im Internet. Ein Repository ist eine Sammlung von RPM-Paketen, zu denen im Verzeichnis `repodata` zusätzliche Metadaten gespeichert sind. Sie geben Informationen über den Inhalt und die Abhängigkeiten aller Pakete.
- ▶ Yum kann mehrere Mirrors für eine Paketquelle verwalten und versucht, den gerade schnellsten Mirror zu verwenden.
- ▶ Yum löst Paketabhängigkeiten auf, lädt alle erforderlichen Pakete und installiert sie. Wenn Sie beispielsweise ein Paket aus der Paketquelle A installieren, kann es sein, dass Yum nach einer Rückfrage abhängige Pakete aus den Quellen B und C herunterlädt und ebenfalls installiert.
- ▶ Yum kann alle bereits installierten Pakete mit einem einzigen Kommando aktualisieren. Dazu wird für jedes Paket getestet, ob es in einer der registrierten Paketquellen eine neuere Version des Pakets gibt. Wenn das der Fall ist, werden die entsprechenden Pakete heruntergeladen und installiert. Natürlich werden auch dabei alle Paketabhängigkeiten aufgelöst.

Aktuelle Fedora-Versionen verwenden anstelle von Yum das modernere Werkzeug DNF. Vermutlich wird auch die Version RHEL 8 auf DNF wechseln. Aus Anwendersicht ändert sich dadurch wenig: Das Kommando `dnf` ist weitestgehend kompatibel zu `yum`; die Verwaltung der Paketquellen hat sich überhaupt nicht geändert. Insofern gelten die folgenden Informationen auch für DNF. Einige spezifische Details von DNF erläutere ich dann in [Abschnitt 19.4](#).

Yum versus DNF

Es ist nicht zulässig, mehrere Yum-Instanzen parallel auszuführen. Wenn bereits ein Yum-Kommando oder -Programm läuft, führt ein neuerlicher Start zur Fehlermeldung *another copy is running*.

Locking-Konflikte

## Konfiguration

Die Grundkonfiguration von Yum erfolgt durch die Datei `/etc/yum.conf`. Die folgenden Zeilen zeigen auszugsweise die Konfiguration von CentOS 7:

```
# Datei /etc/yum.conf
[main]
cachedir=/var/cache/yum/$basearch/$releasever
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
exactarch=1
obsoletes=1
ggpcheck=1
plugins=1
installonly_limit=5
distroverpkg=centos-release
```

Kurz eine Erläuterung zu den wichtigsten Einstellungen: `keepcache=0` bewirkt, dass heruntergeladene Pakete nach der Installation nicht archiviert werden. In der Regel ist das eine zweckmäßige Einstellung, weil der Platzbedarf für die Pakete im Laufe der Zeit recht groß wird und normalerweise kein Grund besteht, die Pakete ein zweites Mal zu installieren. Es kann allerdings passieren, dass `yum` während der Installation ein Problem feststellt und die Installation abbricht. Auch in diesem Fall werden die heruntergeladenen Pakete gelöscht. Wenn Sie das Problem beheben können und anschließend das Update wiederholen, müssen alle Pakete neuerlich heruntergeladen werden. Diese Situation vermeiden Sie mit `keepcache=1`. Um die heruntergeladenen Pakete in `/var/cache/yum` explizit zu löschen, führen Sie `yum clean packages` aus.

`exactarch=1` bewirkt, dass Yum nur Updates berücksichtigt, bei denen die Architektur mit dem bereits installierten Paket übereinstimmt. i386-Pakete können also nicht durch neuere x86\_64-Pakete ersetzt werden.

`ggpcheck=1` bewirkt, dass Yum mit einem Schlüssel die Authentizität der Pakete sicherstellt. `ggpcheck` kann außerdem abweichend von der Einstellung in `yum.conf` auch individuell für jede Paketquelle eingestellt werden. `plugins` entscheidet, ob Yum Plugins berücksichtigt.

Es gibt Pakete, die Yum installieren, aber nicht aktualisieren soll. Dazu zählen insbesondere Kernelpakete: Bei einem Kernel-Update wird das neue Kernelpaket zusätzlich installiert, ohne das alte Kernelpaket anzurühren. Mit der Variablen `installonlypkgs` werden die Namen derartiger Pakete eingestellt. Standardmäßig hat diese Variable die Einstellung `kernel`, `kernel-smp`, `kernel-bigmem`, `kernel-enterprise`, `kernel-debug`, `kernel-unsupported`. Die in `yum.conf` enthaltene Variable `installonly_limit` steuert schließlich, wie viele Versionen derartiger Pakete parallel installiert werden. Die Standardein-

stellung 5 bewirkt, dass immer nur die aktuellsten drei Kernelversionen installiert bleiben. Ältere Kernelpakete werden entfernt.

Jede Paketquelle wird in einer eigenen \*.repo-Datei im Verzeichnis /etc/yum.repos.d definiert. Die folgenden Zeilen zeigen die Paketquelle für die Basispakete von CentOS 7:

**Paketquellen  
einrichten**

```
# Datei /etc/yum.repos.d/CentOS-Base.repo (gekürzt)
[base]
name=CentOS-$releasever - Base
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&
arch=$basearch&repo=os&infra=$infra
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

[updates]
[updates]
[centosplus]
... analog für Updates und Zusatzpakete
```

Die Adresse der Paketquelle kann wahlweise absolut mit `baseurl=...` oder mit `mirrorlist=...` in Form einer Mirror-Datei angegeben werden. Diese Datei enthält eine Liste von Mirror-Servern. Yum entscheidet sich selbstständig für einen der Mirrors. Yum ersetzt in der Konfigurationsdatei die Variablen `$releasever`, `$arch`, `$basearch` etc. durch die Versionsnummer der Linux-Distribution, deren Architektur und andere Angaben. Kurz zur Herkunft der drei wichtigsten Variablen:

- ▶ `$arch` liefert die Architektur des Rechners. Bei einer 64-Bit-Installation ist das ein wenig überraschend `ia32e`. Dieses Kürzel ist eine alte Intel-Schreibweise zur Kennzeichnung von 32-Bit-CPUs mit 64-Bit-Erweiterung.
- ▶ `$basearch` ist die `$arch` zugrunde liegende Basisarchitektur, beispielsweise `x86_64`.
- ▶ `$releasever` ergibt sich aus der Versionsnummer des Pakets, dessen Name in `yum.conf` mit dem Schlüsselwort `distroverpkg` angegeben wurde. Fehlt diese Einstellung, verwendet Yum die Versionsnummer des Pakets `redhat-release`.

Wenn Sie möchten, dass bestimmte Pakete von Yum nicht angetastet und beim Vorliegen einer neuen Version auch nicht aktualisiert werden, fügen Sie in `yum.conf` oder in die \*.repo-Datei der Paketquelle eine Zeile mit `exclude name1 name2 name3` ein. In den Paketnamen können Sie Jokerzeichen verwenden, d. h., auch `exclude emacs*` ist erlaubt.

**Pakete sperren**

Yum lässt sich durch Plugins erweitern und bietet dann noch mehr Funktionen. Die Konfiguration der Plugins erfolgt durch Dateien im Verzeichnis `/etc/yum/pluginconf.d`.

**Plugins**

**Presto** Presto ermöglicht die Verwendung von Delta-RPMs für Updates. Der Download-Umfang von Updates sinkt um 60 bis 80 Prozent, allerdings ist die CPU-Belastung während der Anwendung der Updates höher. Presto war bei älteren Yum-Versionen ein Plugin, mittlerweile ist es direkt in Yum integriert.

**Fastest Mirror** Yum versucht, immer den »besten« Mirror zum Download von Paketen zu verwenden. Das gelingt allerdings nicht immer. Gegebenenfalls können Sie die Fastest-Mirror-Logik manuell optimieren. Dazu erzeugen Sie die Datei `/etc/yum/pluginconf.d/fastestmirror.conf` und schließen darin einzelne besonders lahme Server mit `exclude` aus:

```
# in /etc/yum/pluginconf.d/fastestmirror.conf
...
exclude=lame-duck.com
```

### Das Yum-Kommando

Tabelle 19.2 fasst die wichtigsten Yum-Kommandos zusammen. Wenn Sie `yum` zum ersten Mal ausführen, werden Metainformationen zu allen eingerichteten Paketquellen heruntergeladen, was eine Weile dauern kann. Alle weiteren Kommandos werden dann sofort ausgeführt, bis das nächste Update der Metainformationen ansteht.

| Aufgabe  | Kommando                                    |
|--|---|
| Paket installieren   | <code>yum install name</code>               |
| Lokale Paketdatei installieren   | <code>yum localinstall datei.rpm</code>     |
| Liste der verfügbaren Updates ermitteln  | <code>yum check-update</code>               |
| Ein Paket aktualisieren  | <code>yum update name</code>                |
| Alle Pakete aktualisieren  | <code>yum update</code>                     |
| Paket entfernen  | <code>yum remove name</code>                |
| Liste aller installierten Pakete ermitteln   | <code>yum list installed</code>             |
| Liste aller verfügbaren Pakete ermitteln, deren Name mit <code>abc</code> beginnt  | <code>yum list available 'abc*'</code>      |
| Pakete suchen, die den Begriff <code>abc</code> in der Paketbeschreibung enthalten | <code>yum search 'abc'</code>               |
| Paketgruppen bearbeiten  | <code>yum grouplist/groupinstall/...</code> |
| Liste der letzten Yum-Aktionen anzeigen  | <code>yum history</code>                    |
| Details zur Aktion <code>n</code> ermitteln  | <code>yum history info n</code>             |

**Tabelle 19.2** Wichtige Yum- und DNF-Kommandos

Die folgenden Kommandos demonstrieren die Anwendung von Yum, wobei die Ausgaben aus Platzgründen gekürzt sind: Beispiele

```
root# yum check-update
firefox.x86_64           52.1.0-2.el7.centos    updates
firewall-config.noarch  0.4.3.2-8.1.el7_3.2    updates
firewalld.noarch        0.4.3.2-8.1.el7_3.2    updates
firewalld-filesystem.noarch 0.4.3.2-8.1.el7_3.2    updates
gdm.x86_64              1:3.14.2-20.el7_3      updates
...
root# yum install mariadb-server
...
Installieren:
mariadb-server          x86_64  1:5.5.41-2.el7_0  base  11 M
Als Abhängigkeiten installiert:
perl-Compress-Raw-Bzip2 x86_64  2.061-3.el7       base  32 k
perl-Compress-Raw-Zlib  x86_64  1:2.061-4.el7     base  57 k
perl-DBD-MySQL          x86_64  4.023-5.el7       base  140 k
...
Installieren 1 Paket (+8 Abhängige Pakete)
Gesamte Downloadgröße: 12 M
Installationsgröße: 59 M
Is this ok [y/d/N]: y
```

Wenn Sie nach einem Paket suchen, dessen Namen Sie nicht exakt kennen, führen Sie am besten `yum list available` aus. Dieses Kommando liefert eine Liste aller Pakete, die installiert werden können. Mit `grep` können Sie das Ergebnis filtern:

```
user$ yum list available | grep kde
```

Yum bietet Paketgruppen, die Ihnen dabei helfen, mit wenig Aufwand alle erforderlichen Pakete für eine bestimmte Aufgabe zu installieren. Eine Liste der verfügbaren Paketgruppen samt der englischsprachigen Gruppen-IDs liefert `yum grouplist -v`. Der Befehl `yum groupinfo <gname>` verrät, welche Pakete zu einer Gruppe gehören. `yum groupinfo` unterteilt die Pakete in drei Kategorien: *mandatory*, *default* und *optional*. `yum groupinstall gname` installiert alle *mandatory*- und *default*-Pakete. Yum kennt keine Option, um auch die optionalen Pakete zu installieren. Wenn Sie das möchten, müssen Sie die folgende Änderung in `yum.conf` durchführen:

```
# Ergänzung in /etc/yum.conf
group_package_types = mandatory default optional
```

Um eine Paketgruppe zu aktualisieren bzw. zu entfernen, verwenden Sie `yum group-update` bzw. `yum groupremove`.

Yum ist von sich aus nicht in der Lage, Quellcodepakete zu installieren. Diese Aufgabe übernimmt stattdessen das Kommando `yumdownloader`, das sich im Paket `yum-utils`

**Paketgruppen**

**Quellcodepakete**

befindet. Das folgende Kommando lädt das Quellcodepaket des Editors `gedit` in das lokale Verzeichnis. Dabei werden die normalerweise nicht aktiven `source`-Quellen in den `*.repo`-Dateien automatisch aktiviert.

```
user$ yumdownloader --source gedit
```

- Yum Extender (Yumex)** Yumex (Paketname `yumex`) ist eine einfache und funktionelle grafische Benutzeroberfläche zu Yum. Beim Start aktualisiert Yumex die lokalen Metadaten zu allen Paketquellen. Anschließend können Sie nach Paketen suchen, diese zur Installation markieren und die Installation schließlich durchführen.
- Automatische Updates** Wenn Sie möchten, dass Yum einmal täglich ein automatisches Update durchführt, installieren Sie `yum-cron`. Damit tatsächlich Updates durchgeführt werden, tragen Sie in `/etc/yum/yum-cron.conf` die Einstellung `apply_updates = yes` ein. Um den Update-Aufruf kümmert sich die Cron-Datei `/etc/cron.daily/0yum-daily.cron`, die Sie nicht verändern müssen.

## 19.4 DNF

**DNF** Die Fedora-Entwickler waren mit einigen Aspekten von Yum nicht zufrieden und haben deswegen einen weiteren Paketmanager entwickelt: DNF. Das Kommando `dnf` ist in der Bedienung kompatibel zu `yum`, basiert aber auf anderen Bibliotheken:

- ▶ Die ursprünglich von openSUSE entwickelte Bibliothek `libsolv` ist für die korrekte Auflösung der Paketabhängigkeiten verantwortlich.
- ▶ `hawkey` kümmert sich um die Kommunikation mit RPM und `libsolv`. RPM bleibt also auch für DNF das Fundament.

Im Vergleich zu Yum soll DNF einen besser strukturierten Code aufweisen, weniger Speicher beanspruchen und schneller sein. Davon abgesehen, ändert sich wenig, d. h., [Tabelle 19.2](#) gilt auch für das Kommando `dnf`. Zur Installation eines Pakets führen Sie also `dnf install name` aus, ein Update führen Sie mit `dnf update` durch etc. Bei meinen Tests hat sich DNF wie Yum verhalten, d. h., ich konnte weder Vor- noch Nachteile feststellen. Eine Zusammenfassung der wenigen Bedienungsunterschiede gibt man `yum2dnf`. Weitere Details können Sie auf der folgenden Seite nachlesen:

<https://fedoraproject.org/wiki/Features/DNF>

DNF ersetzt seit Fedora 22 Yum vollständig. RHEL 7 und CentOS 7 basieren weiterhin auf Yum, hier ist die Umstellung auf DNF wohl mit Version 8 zu erwarten.

- Konfiguration** Die DNF-spezifischen Konfigurationsdateien befinden sich im Verzeichnis `/etc/dnf`. Die dort enthaltene Datei `dnf.conf` enthält nur wenige Zeilen:

```
# /etc/dnf/dnf.conf
[main]
gpgcheck=1
installonly_limit=3
clean_requirements_on_remove=true
```

`gpgcheck=1` verlangt wie bei Yum signierte Pakete. `installonly_limit=3` begrenzt die Anzahl alter Kernel-Versionen auf 3. `clean_requirements_on_remove=true` bewirkt, dass bei der Deinstallation eines Pakets automatisch auch alle abhängigen Pakete entfernt werden (es sei denn, diese werden mittlerweile auch von anderen Paketen benötigt).

Nach einigen Diskussionen innerhalb der Fedora-Entwicklergemeinde verwendet DNF standardmäßig Delta-RPMs. Das spart Bandbreite, kostet aber bei einer sehr schnellen Internetverbindung mehr Zeit, weil die Anwendung der Delta-Pakete sehr CPU-intensiv ist. Wenn Sie möchten, können Sie Delta-RPMs durch die Ergänzung `del-tarpm=false` in `dnf.conf` deaktivieren.

Bei den Paketquellen hat sich nichts geändert: Diese werden wie bei Yum in den Dateien `/etc/yum.repos.d/*.repo` gespeichert.

Copr ist ein automatisiertes Build-System. Entwickler können damit aus eigenem Quellcode RPM-Pakete generieren und diese dann unkompliziert anderen CentOS/Fedora/RHEL-Anwendern als Paketquelle anbieten. Copr hat damit Ähnlichkeiten mit dem bekannteren PPA-Konzept von Ubuntu. Copr

<https://developer.fedoraproject.org/deployment/copr/copr-cli.html>

Ich gehe an dieser Stelle nur auf die Nutzung derartiger Copr-Paketquellen ein. Das ist denkbar einfach: Sie richten die Paketquelle mit `dnf copr enable` ein und installieren dann das gewünschte Paket – hier z. B. ein Paket zur Steuerung von Gnome-Shell-Erweiterungen in den Webbrowsern Chrome und Firefox:

```
root# dnf copr enable region51/chrome-gnome-shell
root# dnf install chrome-gnome-shell
```

Das Zusatzpaket `dnf-automatic` kümmert sich um die automatische Installation von Paketen. Die dazugehörige Konfigurationsdatei ist `/etc/dnf/automatic.conf`. Dort müssen Sie `apply_updates` auf `yes` stellen. Wenn Sie möchten, können Sie auch diverse `email`-Parameter einstellen, um zu steuern, wer nach jedem Update automatisch eine E-Mail erhält. Automatische Updates

```
# Datei /etc/dnf/automatic.conf
...
apply_updates = yes
```

Um die automatischen Updates zu aktivieren, sind schließlich noch zwei Kommandos erforderlich:

```
root# systemctl enable dnf-automatic.timer
root# systemctl start dnf-automatic.timer
```

Es kümmert sich also nicht Cron um die automatische Ausführung des Update-Scripts, sondern systemd. Wenn Sie wissen möchten, wann das nächste Update ansteht, führen Sie `systemctl list-timers` aus:

```
root# systemctl list-timers '*dnf*'
NEXT                LEFT                LAST                PASSED
Die 2017-08-15 10:18 CEST 24h ago           Mit 2017-08-16 11:15 CEST 2min 42s ago
Mit 2017-08-16 11:35 CEST 17min left       Mit 2017-08-16 10:35 CEST 42min ago
```

## 19.5 ZYpp

SUSE verwendet wie Fedora und Red Hat RPM-Pakete. Die auf RPM aufbauende Paketverwaltung ZYpp ist allerdings eine SUSE-Eigenentwicklung. ZYpp steht für *ZENworks, YaST, Packages and Patches*, wobei die Verwendung von ZENworks optional und nur in SUSE-Enterprise-Distributionen vorgesehen ist.

**libzypp** Hinter den Kulissen stellt die Bibliothek `libzypp` die ZYpp-Grundfunktionen zur Verfügung. `libzypp` kommt sowohl mit YaST- als auch mit Yum-Paketquellen zurecht. Sämtliche Konfigurations-, Datenbank- und Cache-Dateien befinden sich im Verzeichnis `/var/lib/zypp`. Sowohl YaST als auch `PackageKit` greifen unter openSUSE auf `libzypp` zurück.

### Updates versus Patches

`zypper` unterscheidet zwischen Updates und Patches! Updates sind gewöhnliche RPM-Pakete, die in einer neueren Version als der installierten zur Verfügung stehen. Patches sind dagegen Ergänzungs- bzw. Aktualisierungspakete (Delta-RPMs).

SUSE verwendet zur Aktualisierung seiner eigenen Pakete Patches in Form von Delta-RPMs. Externe Paketquellen, wie `Packman`, stellen neue Paketversionen dagegen in Form von Updates zur Verfügung.

**Paketquellen** Paketquellen werden in Textdateien im Verzeichnis `/etc/zypp/repos.d` gespeichert. Wenn Sie diese Dateien mit einem Editor verändern, müssen Sie darauf achten, anschließend alle Sicherheitskopien zu löschen. Andernfalls bekommen Sie Doppelgänger in der Liste der Paketquellen. Die folgenden Zeilen zeigen die Definition der Open-Source-Paketquelle für openSUSE:

```
# Datei /etc/zypp/repos.d/repo-oss.repo
[repo-oss]
name=openSUSE-Leap-Oss
```



```

enabled=1
autorefresh=1
baseurl=http://download.opensuse.org/distribution/leap/42.1/repo/oss/
path=/
type=yast2
keeppackages=0

```

## Das zypper-Kommando

zypper ist eine Kommandoschnittstelle zu libzypp. zypper ist damit das SUSE-Gegenstück zu yum, dnf bzw. apt-get. Sie können damit Pakete suchen, installieren, aktualisieren und entfernen sowie Paketquellen verwalten (siehe [Tabelle 19.3](#)). zypper muss von root ausgeführt werden.

| Aufgabe                                       | Kommando                       |
|---|--------------------------------|
| Metadaten der Paketquellen neu einlesen       | zypper refresh                 |
| Paket installieren                            | zypper install name            |
| Paket entfernen                               | zypper remove name             |
| Liste aller Updates ermitteln                 | zypper -t package list-updates |
| Alle Pakete aktualisieren                     | zypper -t package update       |
| Distributions-Update durchführen              | zypper dup                     |
| Informationen zu einem Paket ermitteln        | zypper info name               |
| Pakete suchen, deren Paketname abc enthält    | zypper search abc              |
| Pakete suchen, deren Beschreibung abc enthält | zypper search -d abc           |
| Zwischengespeicherte Pakete (Cache) löschen   | zypper clean                   |
| Liste aller Paketquellen ermitteln            | zypper repos                   |
| Neue Paketquelle einrichten                   | zypper addrepo uri name        |

**Tabelle 19.3** Wichtige zypper-Kommandos

Die folgenden Beispiele zeigen die Anwendung von zypper. Das erste Kommando listet die Paketquellen auf, das zweite aktualisiert die Quellen, und das dritte installiert den Editor emacs.

**Beispiele**

```

root# zypper repos
# Alias      Name                               Aktiviert GPG      Aktualis.
1 openSUSE-42.2-0 openSUSE-42.2-0                    Nein      ----      ----
2 repo-non-oss openSUSE-Leap-42.2-Non-Oss        Ja       (r ) Ja      Ja
3 repo-oss    openSUSE-Leap-42.2-Oss            Ja       (r ) Ja      Ja
...

```

```

root# zypper refresh
All repositories have been refreshed.
root# zypper install emacs
The following NEW packages are going to be installed:
  emacs emacs-info emacs-x11 etags libXaw3d8 libm17n0 libotf0 m17n-db
8 new packages to install.
Overall download size: 23.6 MiB. Already cached: 0 B After the operation,
additional 103.5 MiB will be used.
Continue? [y/n/?] (y): y

```

**Paketgruppen** Um alle erforderlichen Pakete für eine bestimmte Aufgabe zu installieren, etwa zur Verwendung des Rechners als Datei-Server, kennt ZYpp sogenannte *pattern*. `zypper search -t pattern` ermittelt eine Liste aller derartigen Paketgruppen. `zypper info -t pattern name` verrät, welche Pakete zu einer Paketgruppe gehören. Mit `zypper install -t pattern name` installieren Sie alle Pakete einer Paketgruppe.

**history** Die Datei `/var/log/zypp/history` enthält eine ausgesprochen praktische Referenz darüber, wann welches Paket aus welcher Paketquelle installiert oder entfernt wurde und welche Konfigurationsarbeiten dabei durchgeführt wurden.

**Distributions-Updates** Mit `zypper dup` führen Sie ein Distributions-Update im laufenden Betrieb durch. Um die dazu erforderliche Änderung der Paketquellen müssen Sie sich allerdings selbst kümmern.

```

root# zypper update      (Update für die bisherige Version)
root# ...                (Paketquellen auf die neue Version umstellen)
root# zypper dup        (alte Pakete durch neue ersetzen)
root# reboot            (Neustart)

```

## 19.6 Debian-Paketverwaltung (dpkg)

Die Verwaltung von Debian-Paketen erfolgt auf zwei Ebenen: Dieser Abschnitt beschreibt das Kommando `dpkg`, das auf der unteren Ebene für die Installation und Verwaltung von Paketen verantwortlich ist. `dpkg` ist mit `rpm` vergleichbar. Das Kommando kann einzelne Pakete installieren, aktualisieren, entfernen und dabei testen, ob alle Paketabhängigkeiten erfüllt sind.

Ähnlich wie `rpm` scheitert auch `dpkg` daran, nicht erfüllte Paketabhängigkeiten selbst aufzulösen oder Pakete selbstständig von Paketquellen zu laden. Genau diese Aufgaben erfüllt `APT` (*Advanced Package Tool*, siehe [Abschnitt 19.7](#)). Es baut auf `dpkg` auf und bietet ähnliche Funktionen wie die gerade vorgestellten Systeme Yum, DNF und ZYpp. Zur eigentlichen Paketverwaltung stehen drei Kommandos zur Wahl: `apt`, `apt-get` und `aptitude`. Für die interaktive Arbeit bietet `apt` am meisten Komfort. Die Unterschiede zwischen den drei Kommandos sind aber gering.

Auch wenn in diesem und dem folgenden Abschnitt von Debian-Paketen die Rede ist, gelten die Informationen für alle Linux-Distributionen, die dieses Paketformat nutzen. Neben Debian sind das beispielsweise die Ubuntu-Familie, Raspbian und Linux Mint. Wenn Sie von einer RPM-basierten Distribution auf eine Distribution mit Debian-Paketen umsteigen, finden Sie auf dieser Seite eine ausgezeichnete Übersicht über rpm-Kommandos sowie dazu äquivalente dpkg- und apt-Kommandos:

<https://help.ubuntu.com/community/SwitchingToUbuntu/FromLinux/RedHatEnterpriseLinuxAndFedora>

dpkg verwaltet zu allen Paketen umfassende Metainformationen (eine Paketbeschreibung, eine Liste aller Dateien des Pakets, Abhängigkeitsdaten etc.). Diese Daten liegen im dctrl-Format (*Debian control*) vor. Das Paket dctrl-tools enthält diverse Kommandos, um Abfragen in den dctrl-Daten durchzuführen. `man grep-dctrl` gibt eine ausführliche Beschreibung dieser Kommandos und konkrete Anwendungsbeispiele.

Metadaten

## Das dpkg-Kommando

Tabelle 19.4 gibt einen Überblick über die wichtigsten dpkg-Optionen. In der Praxis werden Sie dpkg zumeist einsetzen, um Informationen über installierte oder verfügbare Pakete zu ermitteln.

| Aufgabe  | Kommando                                |
|--|---|
| Paket installieren bzw. aktualisieren                | <code>dpkg --install datei.deb</code>   |
| Paket konfigurieren                                  | <code>dpkg --configure datei.deb</code> |
| Paket entfernen                                      | <code>dpkg --remove paketname</code>    |
| Paket vollständig entfernen (auch geänderte Dateien) | <code>dpkg --purge paketname</code>     |
| Alle installierten Pakete ermitteln                  | <code>dpkg --list</code>                |
| Pakete suchen, deren Paketbeschreibung abc enthält   | <code>dpkg --list abc</code>            |
| Liste aller Dateien des Pakets ermitteln             | <code>dpkg --listfiles paketname</code> |

**Tabelle 19.4** Wichtige dpkg-Kommandos

Die folgenden Beispiele verdeutlichen die Anwendung von dpkg in Standardsituationen:

Beispiele

```
root# dpkg --install test.deb
root# dpkg --search /etc/sensors3.conf
libsensors4:adm64 /etc/sensors3.conf
```

```

root# dpkg --listfiles libsensors4
/.
/etc
/etc/sensors.d
/etc/sensors.d/.placeholder
/etc/sensors3.conf
/usr
/usr/lib
/usr/lib/x86_64-linux-gnu
/usr/lib/x86_64-linux-gnu/libsensors.so.4.3.2
...

```

`dpkg --list` liefert eine Liste aller installierten Pakete. Die Ausgaben des folgenden Beispiels habe ich zur besseren Lesbarkeit etwas gekürzt:

```

root# dpkg --list | grep cups
ii cups 1.7.5-11+deb8u1 amd64 CUPS - driver, web interface
ii cups-browsed 1.0.61-5+deb8u1 amd64 OpenPrinting CUPS Filters
ii cups-bsd 1.7.5-11+deb8u1 amd64 CUPS - BSD commands
ii cups-client 1.7.5-11+deb8u1 amd64 CUPS - client programs (SysV)
ii cups-common 1.7.5-11+deb8u1 all CUPS - common files
ii cups-core-drivers 1.7.5-11+deb8u1 amd64 CUPS - PPD-less printing
...

```

`dpkg --list` zeigt einen Statuscode an, der aus zwei oder drei Buchstaben besteht (siehe [Tabelle 19.5](#)). Die beiden häufigsten Statuscodes sind `ii` für ein korrekt installiertes Paket sowie `rc` für ein entferntes Paket, bei dem die Konfigurationsdateien noch verfügbar sind. Um `rc`-Pakete vollständig zu entfernen, führen Sie `dpkg --purge name` aus. Weitere Details zum Paketstatus und zur Behebung von Problemen gibt man `dpkg`.

### Paket neu konfigurieren

Bei der Installation von Debian-Paketen werden automatisch Installations- und Konfigurations-Skripts ausgeführt. Bei einigen wenigen Programmen gibt es darüber hinaus interaktive Setup-Programme, die bei der individuellen Konfiguration des Pakets helfen, z. B. bei der Grundkonfiguration des E-Mail-Servers Postfix. Wenn Sie die Konfiguration später wiederholen möchten, führen Sie `dpkg-reconfigure paketname` aus.

### get-selections und set-selections

Um rasch eine sortierte Liste aller installierten Pakete zu ermitteln, führen Sie `dpkg --get-selections` aus. Die Paketliste enthält weniger Detailinformationen als jene von `dpkg --list` und ist daher übersichtlicher. Wenn Sie die Liste in eine Textdatei umleiten und speichern, können Sie alle Pakete später auf einem anderen Rechner mit `dpkg --set-selections` installieren.

```
root# dpkg --get-selections
accountsservice          install
acl                      install
...
```

Der Status *hold* bedeutet, dass ein Paket bei einem Update nicht aktualisiert werden soll. Die beiden folgenden Kommandos zeigen, wie Sie ein Paket in den *hold*-Status bringen bzw. diesen Status wieder aufheben: hold-Status

```
root# echo "paketname hold" | dpkg --set-selections
root# echo "paketname install" | dpkg --set-selections
```

| Code | Bedeutung   |
|------|---|
|      | <b>Erster Buchstabe:</b> gewünschter Zustand                  |
| u    | unbekannt (unknown)   |
| i    | zu installieren (install)                                     |
| r    | zu löschen (remove)   |
| p    | komplett zu löschen (purge)                                   |
| h    | unverändert lassen (hold)                                     |
|      | <b>Zweiter Buchstabe:</b> tatsächlicher Zustand               |
| n    | nicht installiert (not)                                       |
| h    | teilweise installiert (half installed)                        |
| u    | ausgepackt, aber nicht konfiguriert (unpacked)                |
| f    | installiert, aber nicht konfiguriert (failed config file)     |
| i    | vollständig installiert (install)                             |
| c    | Nur die Konfigurationsdateien sind installiert (config file). |
| t    | wartet auf den Trigger eines anderen Pakets (trigger).        |
|      | <b>Dritter Buchstabe (optional):</b> Fehlercode               |
| h    | Paket soll nicht geändert werden (hold).                      |
| r    | Neuinstallation erforderlich (reinstall required)             |

**Tabelle 19.5** Buchstabencodes des dpkg-list-Ergebnisses

## 19.7 APT

APT (*Advanced Packaging Tool*) ist für Debian-Pakete das, was DNF, Yum bzw. Zypp für RPM-Pakete ist: ein High-Level-Paketverwaltungssystem, das Pakete selbstständig von Paketquellen herunterlädt und Paketabhängigkeiten automatisch auflöst. Die Kombination aus Debian-Paketen und APT ergibt momentan das wohl ausgereifteste Paketverwaltungssystem für Linux. Es wird unter anderem von Ubuntu und Debian als Standardsystem zur Paketverwaltung eingesetzt.

Wie Yum, DNF und Zypp erfordert auch APT spezielle Paketquellen, die neben den DEB-Paketen auch Metainformationen über den Inhalt der Pakete und deren Abhängigkeiten zur Verfügung stellen.

**apt versus  
apt-get versus  
aptitude**

Zur eigentlichen Paketverwaltung stehen gleich drei alternative Kommandos zur Auswahl: `apt`, `apt-get` und `aptitude`. Die Kommandos sind einander sehr ähnlich und weisen bei vielen Operationen sogar dieselbe Syntax auf. `apt` und `aptitude` sind speziell für die interaktive Benutzung optimiert, während `apt-get` speziell zur Script-Programmierung bzw. zur automatisierten Ausführung von Paketverwaltungskommandos geeignet ist.

Bei aktuellen Debian- und Ubuntu-Systemen sind `apt` und `apt-get` standardmäßig installiert; `aptitude` muss bei Bedarf nachinstalliert werden. Ubuntu empfiehlt für die interaktive Paketverwaltung das Kommando `apt`. Dessen ungeachtet ist `apt-get` nach wie vor populärer, und sei es nur, weil so viele Anleitungen im Internet `apt-get` verwenden. Aber Sie machen auch mit `apt` und `aptitude` nichts verkehrt. Es ist auch kein Problem, mal das eine und dann wieder das andere Kommando zu verwenden.

Die folgenden drei Kommandos sind gleichwertig. Sie laden das angegebene Paket und alle davon abhängigen Pakete herunter und installieren sie:

```
root# apt      install paketname
root# apt-get install paketname
root# aptitude install paketname
```

## Konfiguration

Die Konfiguration von APT erfolgt durch die beiden Dateien `apt.conf.d/*` und `sources.list` im Verzeichnis `/etc/apt`. Weitere Definitionen von Paketquellen können sich im Verzeichnis `sources.list.d` befinden.

`apt.conf.d/*` enthält in der Regel nur wenige Basiseinstellungen, die Sie zumeist so belassen, wie sie von Ihrer Distribution vorgegeben sind. Schon interessanter ist `sources.list`. Diese Datei enthält zeilenweise die APT-Paketquellen. Die Syntax jeder Zeile sieht so aus:

```
pakettyp uri distribution [komponente1] [komponente2] [komponente3] ...
```

Der Pakettyp lautet `deb` für gewöhnliche Debian-Pakete bzw. `deb-src` für Quellcodepakete. Die zweite Spalte gibt das Basisverzeichnis der Paketquelle an. Neben HTTP- und FTP-Verzeichnissen unterstützt APT auch gewöhnliche Verzeichnisse, RSH- oder SSH-Server sowie CDs bzw. DVDs.

Die dritte Spalte bezeichnet die Distribution. Im folgenden Listing handelt es sich dabei um Ubuntu 17.04 mit dem Codenamen »Zesty Zapus«. Alle weiteren Spal-

ten geben die Komponenten der Distribution an, die berücksichtigt werden können. Die Komponentennamen sind von der Distribution und von der Paketquelle abhängig! Beispielsweise unterscheidet Ubuntu zwischen *main*-, *restricted*-, *universe*- und *multiverse*-Paketen, während Debian zwischen den Komponenten *main*, *contrib*, *non-free* etc. differenziert.

Die zuerst genannten Paketquellen werden bevorzugt: Wenn ein bestimmtes Paket also in mehreren Quellen zum Download zur Verfügung steht, lädt APT es von der ersten Quelle herunter. Das folgende Listing verdeutlicht die Syntax. Aus Platzgründen wurde dabei jeder Eintrag über zwei Zeilen verteilt.

```
# Datei /etc/apt/sources.list (Ubuntu)
deb http://de.archive.ubuntu.com/ubuntu/ zesty          \
      main restricted universe multiverse
deb http://de.archive.ubuntu.com/ubuntu/ zesty-updates  \
      main restricted universe multiverse
deb http://security.ubuntu.com/ubuntu   zesty-security \
      main restricted universe multiverse
```

Veränderungen an `sources.list` führen Sie am einfachsten mit einem Texteditor durch. Alternativ können Sie auch eine grafische Benutzeroberfläche zu Hilfe nehmen, z. B. `software-properties-gtk`.

Bei den meisten APT-Quellen im Internet sind die Metadateien zur Beschreibung der Paketquellen durch einen Schlüssel signiert. Weiters enthalten die APT-Inhaltsverzeichnisse Prüfsummen für alle Pakete. Mit diesem Kontrollmechanismus kann sichergestellt werden, dass kein Paket nachträglich verändert wurde. Diese Kontrolle funktioniert aber nur, wenn APT den öffentlichen Teil des Schlüssels kennt und somit die Authentizität des Paketarchivs feststellen kann. Um einen Schlüssel für APT einzurichten, verwenden Sie das Kommando `apt-key`:

APT-Schlüssel  
installieren

```
root# apt-key add schlüsseldatei.gpg
```

## Das apt-Kommando

Die eigentliche Paketverwaltung führen Sie wahlweise mit dem Kommando `apt` oder dessen Alternativen `apt-get` bzw. `aptitude` durch (siehe die folgenden Abschnitte). Ubuntu empfiehlt für die interaktive Benutzung explizit `apt`. Für Debian habe ich keine diesbezügliche Information gefunden, aber auch dort gilt, dass `apt` benutzerfreundlicher ist und Kommandos enthält, die in `apt-get` fehlen und für die Sie ansonsten `apt-cache` oder `dpkg` aufrufen müssen. Die wichtigsten `apt`-Kommandos sind in [Tabelle 19.6](#) zusammengefasst.

| Aufgabe   | Kommando                            |
|---|-------------------------------------|
| Metadaten aus den Paketquellen aktualisieren    | <code>apt update</code>             |
| Paket installieren                              | <code>apt install name</code>       |
| Alle Pakete aktualisieren                       | <code>apt upgrade</code>            |
| Wie oben, aber bei Bedarf Pakete deinstallieren | <code>apt full-upgrade</code>       |
| Paket entfernen                                 | <code>apt remove name</code>        |
| Nicht mehr benötigte Pakete deinstallieren      | <code>apt autoremove</code>         |
| Zwischengespeicherte Pakete aus Cache löschen   | <code>apt autoclean</code>          |
| Paket entfernen                                 | <code>apt remove name</code>        |
| Alle verfügbaren Pakete auflisten               | <code>apt list</code>               |
| Alle installierten Pakete auflisten             | <code>apt list --installed</code>   |
| Paket suchen                                    | <code>apt search suchbegriff</code> |
| Infos zu Paket anzeigen                         | <code>apt show paketname</code>     |

Tabelle 19.6 Wichtige apt-Kommandos

**Beispiele** Bevor Sie Pakete installieren, sollten Sie `apt-get update` ausführen und damit die neuesten Informationen aus den Paketquellen herunterladen. Dadurch werden weder Pakete installiert noch aktualisiert; es geht hier nur um die Paketbeschreibungen, also um die Metadaten! Die meisten anderen Paketverwaltungssysteme (Yum, Zypp) aktualisieren diese Metadaten bei Bedarf selbstständig – aber eben nicht bei `apt-get` und `aptitude`. Anschließend können Sie ein neues Paket mit `apt-get install` herunterladen und installieren:

```
root# apt update
root# apt install dovecot-imapd
...
Die folgenden zusätzlichen Pakete werden installiert:
  dovecot-core
Vorgeschlagene Pakete:
  ntp dovecot-gssapi dovecot-sieve dovecot-pgsql dovecot-mysql
  dovecot-sqlite dovecot-ldap dovecot-pop3d dovecot-lmtpd
  dovecot-managesieved dovecot-solr
Die folgenden NEUEN Pakete werden installiert:
  dovecot-core dovecot-imapd
0 aktualisiert, 2 neu installiert, 0 zu entfernen und 33 nicht aktualisiert.
Es müssen 2.796 kB an Archiven heruntergeladen werden.
Nach dieser Operation werden 9.284 kB Plattenplatz zusätzlich benutzt.
Möchten Sie fortfahren? [J/n]  J
```

`apt remove paketname` entfernt das angegebene Paket. Ursprünglich zusammen mit dem Paket installierte abhängige Pakete bleiben davon aber unberührt. Das können



Sie mit `apt autoremove` beheben: Dieses Kommando entfernt alle nicht mehr benötigten Pakete.

Das richtige Kommando zur Durchführung von Updates ist in aller Regel `apt full-upgrade`. Wenn es aufgrund geänderter Paketabhängigkeiten erforderlich ist, werden dadurch auch zusätzliche Pakete installiert bzw. vorhandene Pakete entfernt. `apt upgrade` führt zwar auch ein Update durch, rührt aber Pakete nicht an, wenn aufgrund von geänderten Abhängigkeiten gleichzeitig andere Pakete entfernt werden müssen.

```
root# apt full-upgrade
```

`apt source paketname` installiert den Quellcode des gewünschten Pakets in das aktuelle Verzeichnis. Weitere Details finden Sie in [Abschnitt 20.2](#), »Programme selbst kompilieren«.

Quellcode  
installieren

Um ein Update auf die nächste Version von Debian oder Ubuntu durchzuführen, passen Sie zuerst die Paketquellen in `/etc/apt/sources.list` entsprechend an. Insbesondere müssen Sie dort den Codenamen der jeweiligen alten Version durch den der neuen Version ersetzen, also z. B. bei einem Update von Debian 8 auf Version 9 `jessie` durch `stretch`. Anschließend führen Sie `apt-get dist-upgrade` aus. Der Download und die Installation der Pakete wird je nach Installationsumfang circa eine halbe Stunde dauern. Danach starten Sie Ihren Rechner neu – fertig!

Distributions-  
Updates

Leider sind Release-Updates trotz des ausgezeichneten Debian-Paketverwaltungssystems eine heikle Angelegenheit. Dass nach dem Update wirklich alle Programme und Server-Dienste wie bisher funktionieren, ist eher ein Glücks- als der Regelfall. Deswegen ziehe ich eine Neuinstallation zumeist vor.

## Das `apt-get`-Kommando

Die Subkommandos des `apt-get`-Kommandos stimmen überwiegend mit denen von `apt` überein (siehe [Tabelle 19.7](#)). Anstelle von `full-upgrade` heißt es ein wenig irreführend `dist-upgrade`, obwohl sich die Version der Distribution nicht ändert. Die Sub-Kommandos `list`, `search` und `show` stehen nicht zur Verfügung. Vergleichbare Funktionen bieten stattdessen die Kommandos `dpkg` und `apt-cache`.

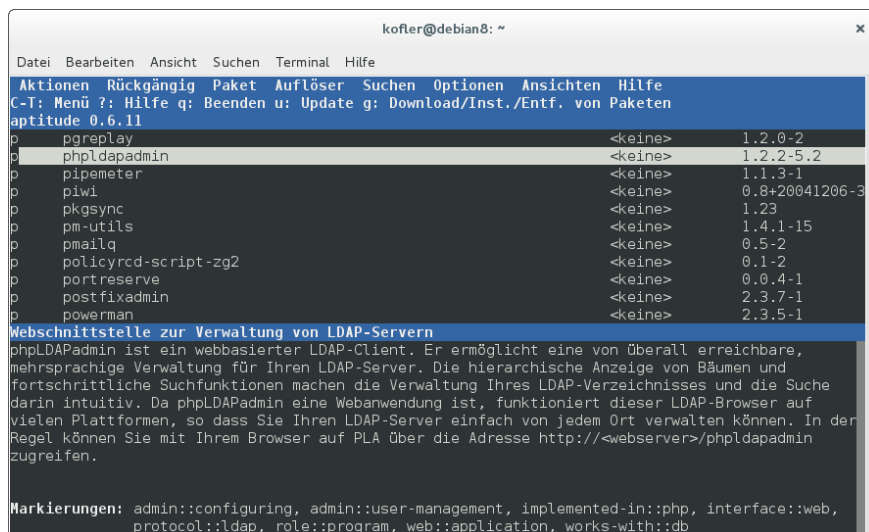
## Das `aptitude`-Kommando

Das Kommando `aptitude` baut ebenfalls auf APT auf. Wenn Sie das Programm kommandoorientiert einsetzen (`aptitude install paketname`), ist es weitgehend syntaxkompatibel zu `apt-get`. Die meisten in [Tabelle 19.7](#) aufgezählten Kommandos – mit der Ausnahme von `autoremove` – stehen in gleicher Form für `aptitude` zur Verfügung. Das Pendant zu `apt-get dist-upgrade` lautet allerdings `aptitude full-upgrade`.

| Aufgabe  | Kommando             |
|--|----------------------|
| Metadaten aus den Paketquellen aktualisieren                       | apt-get update       |
| Paket installieren   | apt-get install name |
| Alle Pakete aktualisieren  | apt-get upgrade      |
| Wie oben, aber bei Bedarf auch neue, abhängige Pakete installieren | apt-get dist-upgrade |
| Paket entfernen  | apt-get remove name  |
| Nicht mehr benötigte Pakete deinstallieren                         | apt-get autoremove   |
| Zwischengespeicherte Pakete aus Cache löschen                      | apt-get autoclean    |

Tabelle 19.7 Wichtige apt-get-Kommandos

Alternativ können Sie das Programm auch mit einer Textbenutzeroberfläche in einer Konsole nutzen (siehe [Abbildung 19.2](#)), indem Sie es einfach ohne weitere Parameter starten. Zur Menüauswahl verwenden Sie die Tastenkombination `[Strg]+[T]`. Wirklich intuitiv ist aptitude trotz Menü nicht zu bedienen; die meisten Anwender verwenden aptitude daher wie apt-get einfach zum Ausführen einzelner Kommandos.



```

kofter@debian8: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
Aktionen Rückgängig Paket Auflöser Suchen Optionen Ansichten Hilfe
C-T: Menü ?: Hilfe q: Beenden u: Update g: Download/Inst./Entf. von Paketen
aptitude 0.6.11
p pcreplay <keine> 1.2.0-2
p phpldapadmin <keine> 1.2.2-5.2
p pipemeter <keine> 1.1.3-1
p piwi <keine> 0.8+20041206-3
p pkgsync <keine> 1.23
p pm-utils <keine> 1.4.1-15
p pmailq <keine> 0.5-2
p policyrcd-script-zg2 <keine> 0.1-2
p portreserve <keine> 0.0.4-1
p postfixadmin <keine> 2.3.7-1
p powerman <keine> 2.3.5-1
Webschnittstelle zur Verwaltung von LDAP-Servern
phpLDAPadmin ist ein webbasierter LDAP-Client. Er ermöglicht eine von überall erreichbare,
mehrsprachige Verwaltung für Ihren LDAP-Server. Die hierarchische Anzeige von Bäumen und
fortschrittliche Suchfunktionen machen die Verwaltung Ihres LDAP-Verzeichnisses und die Suche
darin intuitiv. Da phpLDAPadmin eine Webanwendung ist, funktioniert dieser LDAP-Browser auf
vielen Plattformen, so dass Sie Ihren LDAP-Server einfach von jedem Ort verwalten können. In der
Regel können Sie mit Ihrem Browser auf PLA über die Adresse http://<webserver>/phpldapadmin
zugreifen.
Markierungen: admin::configuring, admin::user-management, implemented-in::php, interface::web,
protocol::ldap, role::program, web::application, works-with::db

```

Abbildung 19.2 Paketverwaltung mit aptitude

aptitude bietet im Vergleich zu apt und apt-get einen entscheidenden Vorteil: Das Programm merkt sich, welche abhängigen Pakete installiert wurden, und entfernt diese bei einer Deinstallation automatisch wieder. Wenn Sie beispielsweise das Programm xyz installieren und dieses fünf weitere Pakete (lib-abc, lib-efg etc.) voraussetzt, dann werden diese Pakete bei der Deinstallation von xyz wieder entfernt, sofern mittler-

weile kein anderes Paket davon abhängig ist. Wenn Sie xyz dagegen mit apt-get oder Synaptic entfernen, bleiben die abhängigen Pakete lib-abc, lib-efg etc. erhalten. Nach einer Weile weiß niemand mehr, warum die Pakete installiert worden sind. Debian empfiehlt deswegen explizit die Verwendung von aptitude anstelle von apt-get zur Paketverwaltung.

Die beiden folgenden Kommandos installieren zuerst das Paket mysql-server und entfernen es dann wieder. Bemerkenswert ist, dass beim zweiten Kommando die zahlreichen zusammen mit mysql-server installierten abhängigen Pakete weitestgehend wieder entfernt werden. Beispiel

```
root# aptitude install mysql-server
Die folgenden NEUEN Pakete werden zusätzlich installiert:
  libaio1a libdbd-mysql-perla libdbi-perla libhtml-template-perla
  libmysqlclient18a libterm-readkey-perla mysql-client-5.5a mysql-commona
  mysql-server mysql-server-5.5a mysql-server-core-5.5a
0 Pakete aktualisiert, 11 zusätzlich installiert, 0 werden entfernt.
root# aptitude remove mysql-server
Die folgenden Pakete werden ENTFERNT:
  libaio1u libdbd-mysql-perlu ... mysql-server-core-5.5u
0 Pakete aktualisiert, 0 zusätzlich installiert, 10 werden entfernt
```

### APT-Zusatzkommandos

Zur Installation von Paketgruppen greifen Debian und Ubuntu in der Regel auf Metapakete zurück: Das sind leere Pakete, die lediglich eine Menge Paketabhängigkeiten definieren. Beispielsweise wird zusammen mit dem Metapaket build-essential eine ganze Sammlung von Paketen mit grundlegenden Entwicklungswerkzeugen installiert (Compiler, make etc.). tasksel

Daneben gibt es noch einen zweiten Mechanismus zur Definition von Paketgruppen, der auf dem Kommando tasksel basiert. Dieser Mechanismus ist vor allem dazu gedacht, während der Installation der Distribution auf einfache Weise Paketgruppen auszuwählen. tasksel kann aber natürlich auch später im laufenden Betrieb verwendet werden: Eine Liste aller verfügbaren Paketgruppen liefert tasksel --list-task. Zur Installation von Paketgruppen verwenden Sie tasksel install gruppenname. Wenn tasksel ohne Optionen ausgeführt wird, erscheint ein Dialog zur Auswahl der gewünschten Paketgruppen.

apt-cache ermittelt diverse Daten zu den verfügbaren bzw. zu bereits installierten Paketen: apt-cache

```
root# apt-cache show apache2
Package: apache2
Description: next generation, scalable, extendable web server
...
```

```
root# apt-cache search scribus | sort
lprof - Hardware Color Profiler
scribus - Open Source Desktop Publishing
scribus-template - additional scribus templates
```

**apturl** Unter Ubuntu ist standardmäßig das Paket `apturl` mit dem Programm `apturl-gtk` installiert. Es ermöglicht nach einer Rückfrage die Installation von Paketen einfach durch das Anklicken spezieller `atp:`-Links im Webbrowser. Auf derartige Links stoßen Sie vor allem auf Ubuntu-Wikis und -Foren.

**History** Alle durch das APT-System durchgeführten Operationen (egal, ob nun `apt`, `apt-get` oder ein anderes Kommando zum Einsatz kommt) werden in `/var/log/apt/history.log` protokolliert. Wenn diese Datei nicht existiert oder leer ist, wurde sie vielleicht »rotiert«, also umbenannt und komprimiert. Dekomprimieren Sie gegebenenfalls `history.log.1.gz`, `history.log.2.gz` etc. und werfen Sie einen Blick in diese Dateien.

### Updates automatisieren

**unattended-upgrades** Das Paket `unattended-upgrades` kümmert sich bei richtiger Konfiguration darum, automatisch Updates durchzuführen. Unter Ubuntu ist das Paket standardmäßig installiert; unter Debian und Raspbian müssen Sie die Installation selbst durchführen.

Ausgangspunkt für die Download- und Update-Automatik ist das Programm Cron, das einmal täglich das Script `/etc/cron.daily/apt` ausführt. Es wertet die Konfigurationsdatei `/etc/apt/apt.conf.d/*` aus und führt bei Bedarf das Upgrade-Kommando `unattended-upgrade` aus.

**Basis-konfiguration** Leider sind die Voreinstellungen in den Konfigurationsdateien je nach Distribution recht unterschiedlich. Grundvoraussetzungen sind die folgenden drei Parameter, die Sie gegebenenfalls richtigstellen oder einer eigenen Konfigurationsdatei hinzufügen müssen. Ich verwende oft `99myown`. Weil der Dateiname mit `99` beginnt, wird die Datei zum Schluss verarbeitet und überschreibt, falls notwendig, in anderen Dateien eingestellte Optionen.

```
// Datei /etc/apt/apt.conf.d/99myown
// Script unattended-upgrade aktivieren
APT::Periodic::Unattended-Upgrade "1";

// Paketliste einmal täglich aktualisieren
APT::Periodic::Update-Package-Lists "1";

// Updates tatsächlich durchführen
APT::Periodic::Unattended-Upgrade "1";
```

Nun müssen Sie noch klarstellen, welche Pakete automatisch aktualisiert werden sollen. Unter Ubuntu ist dazu der Parameter `Allowed-Origins` in der Datei `50unattended-upgrades` so vorkonfiguriert, dass normale Updates sowie Sicherheits-Updates installiert werden – eine durchaus sinnvolle Einstellung:

**Welche Pakete aktualisieren?**

```
// Datei /etc/apt/apt.conf.d/50 (Ubuntu)
Unattended-Upgrade::Allowed-Origins {
    "${distro_id}:${distro_codename}";
    "${distro_id}:${distro_codename}-security";
}
```

Debian und Raspbian steuern die zu aktualisierenden Pakete dagegen durch `Origin-Patterns`. Die Defaultkonfiguration von Debian sieht nur die Installation sicherheitskritischer Updates vor. Wenn Sie auch normale Updates installieren möchten, entfernen Sie die `label`-Angabe aus der Konfigurationsdatei:

```
// Datei /etc/apt/apt.conf.d/50unattended-upgrades (Debian)
Unattended-Upgrade::Origins-Pattern {
    "origin=Debian,codename=${distro_codename},label=Debian-Security";
};
```

Unter Raspbian enthält `Origins-Pattern` nur auskommentierte Beispieleinträge. Selbst wenn Sie alle anderen Optionen korrekt eingestellt haben, führt `unattended-upgrade` deswegen keine Updates durch! Abhilfe: Verwenden Sie die folgende Konfiguration, um alle Updates aus allen Paketquellen zu installieren:

```
// Datei /etc/apt/apt.conf.d/50unattended-upgrades (Raspbian)
Unattended-Upgrade::Origins-Pattern {
    "origin=*";
};
```

Mit `Package-Blacklist` können Sie, falls erwünscht, einzelne Pakete vom automatischen Update ausschließen:

```
// folgende Pakete nicht aktualisieren
Unattended-Upgrade::Package-Blacklist {
    "vim";
    ...
};
```

Im Zuge der automatischen Updates werden natürlich bei Bedarf auch neue Kernelversionen installiert. Derartige Updates werden aber erst wirksam, wenn der Rechner neu gestartet wird. Standardmäßig ist dies nicht der Fall, aber die folgenden Einstellungen in `50unattended-upgrades` oder einer eigenen Konfigurationsdatei schaffen Abhilfe:

**Kernel-Updates**

```
# wenn erforderlich, automatischer Reboot in der nächsten Nacht um 2:00
Unattended-Upgrade::Automatic-Reboot "true";
Unattended-Upgrade::Automatic-Reboot-Time "02:00";
```

Wenn Sie manuell feststellen möchten, ob ein Neustart erforderlich ist, testen Sie einfach, ob die Datei `/var/run/reboot-required` existiert. `/var/run/reboot-required.pkgs` listet auf, aufgrund welcher Pakete ein Neustart notwendig ist. In der Regel sind das der Kernel oder systemnahe Bibliotheken. Alternativ können Sie auch das Kommando `checkrestart` aus dem Paket `debian-goodies` aufrufen.

**Volle  
Boot-Partition  
vermeiden**

Normale Pakete werden bei Updates einfach ersetzt. Kernel-Updates werden hingegen parallel installiert, sodass Sie bei Problemen immer noch auf die vorige Kernel-Version zurückgreifen können. Gerade bei Server-Installationen ist es üblich, dass es für das Verzeichnis `/boot` eine eigene Partition gibt. Dort landen der Kernel sowie einige dazugehörige Dateien – und führen früher oder später dazu, dass die Boot-Partition vollläuft. Im ungünstigsten Fall ist dann ein nicht mehr boot-fähiges System die Folge. (Unter CentOS/Fedora/RHEL tritt dieses Problem nicht auf, weil Yum bzw. DNF standardmäßig die Anzahl der parallel installierten Kernelversionen auf drei limitiert.)

Abhilfe schafft die regelmäßige Ausführung von `apt autoremove`. Noch besser ist es freilich, auch diese Aufgabe zu automatisieren. Die Datei `50unattended-upgrades` sieht dazu bereits die Option `Remove-Unused-Dependencies` vor. Sie müssen die Option lediglich auf `true` setzen. Es stellt sich nur die Frage, warum diese Option nicht standardmäßig aktiv ist.

```
# in /etc/apt/apt.conf.d/50unattended-upgrades ändern
...
// nach Update apt-get autoremove ausführen
Unattended-Upgrade::Remove-Unused-Dependencies "true";
```

**Logging** `unattended-upgrades` protokolliert seine Bemühungen in der folgenden Datei. Die Datei ist insbesondere dann eine große Hilfe, wenn `unattended-upgrades` nicht funktioniert und Sie herausfinden möchten, woran dies liegt.

```
/var/log/unattended-upgrades/unattended-upgrades
```

Außerdem legt `/etc/cron.daily/apt` im Verzeichnis `/var/lib/apt/periodic/` Timestamp-Dateien an, aus denen hervorgeht, wann bestimmte Operationen zum letzten Mal durchgeführt wurden.

**Interna** Abschließend noch einige Interna zum Upgrade-Prozess: `/etc/cron.daily/apt` enthält ein `sleep`-Kommando, das das Script für eine zufällige Anzahl von bis zu 1800 Sekunden anhält, also für bis zu einer halben Stunde. Diese Zwangspause vermeidet, dass Tausende Rechner aufgrund des Cron-Jobs gleichzeitig auf die Paketquellen zugreifen.

`unattended-upgrade` aktualisiert keine Pakete, für die ein sogenannter `conffile` prompt besteht, d. h., deren Konfigurationsdatei manuell verändert wurde. Leider geht das

nur aus den Logging-Dateien hervor, nicht aus der Status-E-Mail. Da dieser Fall in der Praxis immer wieder vorkommt, müssen Sie trotz der automatisierten Updates regelmäßig kontrollieren, ob es nicht doch Updates gibt, die manuell durchzuführen sind.

### Das Für und Wider automatischer Updates

Automatische Updates mindern das Risiko, dass ein Angreifer eine bereits behobene Sicherheitslücke ausnutzt. Aber sie können auch fatale Folgen haben, wenn einzelne Server-Funktionen aufgrund eines defekten Updates plötzlich nicht mehr richtig laufen. Eine Alternative zu automatischen Updates ist ein Cron-Script, das einmal täglich die Verfügbarkeit neuer Updates überprüft (z. B. durch `apt-get dist-upgrade --simulate`) und das Ergebnis als E-Mail an den Administrator sendet.

## Synaptic

Die populärste grafische Benutzeroberfläche zur Administration von Debian-Paketen auf der Basis der APT-Kommandos hieß lange Zeit Synaptic. Aktuelle Versionen von Debian und Ubuntu sehen zwar mittlerweile standardmäßig die modernere PackageKit-Benutzeroberfläche `gpk-application` bzw. das Gnome-Programm *Software* zur Installation von Programmen vor (siehe [Abbildung 5.13](#) auf Seite 171); persönlich ziehe ich aber weiterhin das stärker technisch orientierte Programm Synaptic vor, um gezielt nach Paketen zu suchen und diese zu installieren (siehe [Abbildung 19.3](#)).

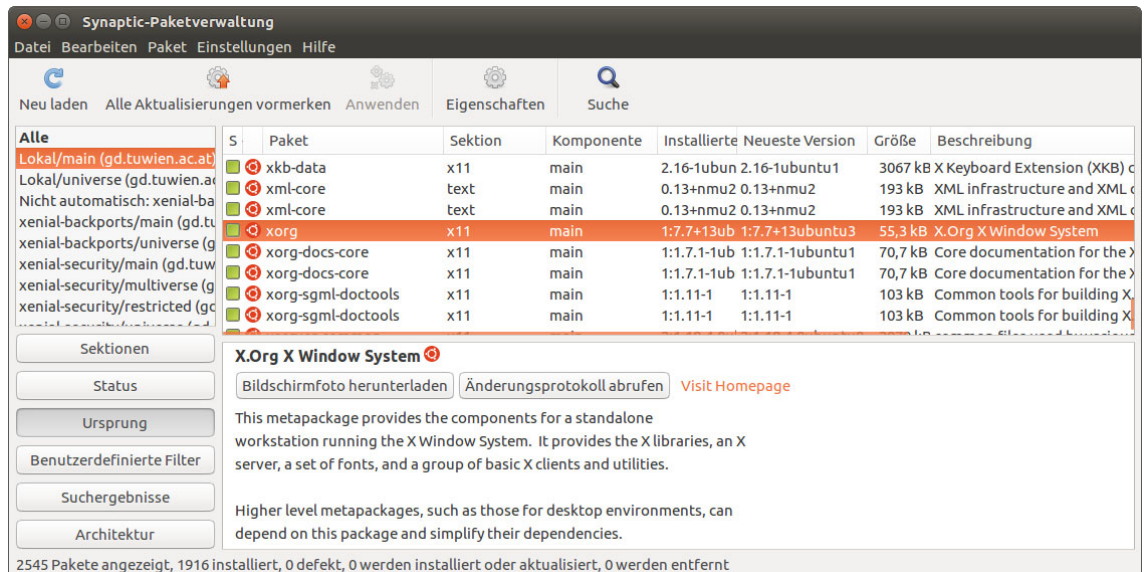


Abbildung 19.3 Paketverwaltung mit Synaptic

- Installation** Um ein bestimmtes Paket zu installieren, wählen Sie es per Doppelklick zur Installation aus. Wenn das Paket von anderen Paketen abhängig ist, erscheint ein Dialog mit allen weiteren Paketen, die ebenfalls installiert werden müssen. Die eigentliche Installation beginnt mit dem Button ANWENDEN, wobei Sie noch eine Zusammenfassung aller geplanten Aktionen bestätigen müssen.
- Die Liste aller zur Installation vorgemerkten Pakete sehen Sie, wenn Sie zuerst den Button BENUTZERDEFINIERTER FILTER und dann den Listeneintrag VORGEMERKTE ÄNDERUNGEN anklicken. Synaptic merkt sich alle Installationsvorgänge. Sie können diese jederzeit mit DATEI • CHRONIK nachvollziehen.
- Paketquellen verwalten** Die Verwaltung der Paketquellen erfolgt im Dialog EINSTELLUNGEN • PAKETQUELLEN. Dort werden alle bekannten Paketquellen angezeigt. Durch einen Klick auf das Auswahlhäkchen können Sie einzelne Paketquellen rasch aktivieren bzw. deaktivieren. Mit BEARBEITEN verändern Sie die Eigenschaften vorhandener Paketquellen, mit HINZUFÜGEN richten Sie eine neue Paketquelle ein.
- Defekte Pakete** Ein Paket gilt als »defekt«, wenn während der Installation oder Deinstallation ein Problem auftritt und der Vorgang nicht korrekt abgeschlossen werden kann. Synaptic und andere Paketverwaltungswerkzeuge verweigern ihren Dienst, bis dieses Problem gelöst ist.
- Zur Abhilfe klicken Sie in Synaptic in der Seitenliste auf den Button BENUTZERDEFINIERTER FILTER und dann auf den Eintrag DEFEKT. Synaptic zeigt nun eine Liste aller defekten Pakete an. Markieren Sie alle Pakete durch `[Strg]+[A]`, klicken Sie die Liste mit der rechten Maustaste an, und wählen Sie den Eintrag ZUM ERNEUTEN INSTALLIEREN VORMERKEN. Anschließend führen Sie die Neuinstallation durch ANWENDEN aus. Sollten dabei abermals Probleme auftreten, markieren Sie die betreffenden Pakete zum Entfernen.

### Locking-Probleme

Es kann immer nur ein Paketverwaltungsprogramm laufen. Beim Versuch, zwei Paketverwaltungsprogramme gleichzeitig auszuführen, erscheint die Warnung *unable to get exclusive lock*. Das bedeutet, dass das Programm nicht allein auf die internen Paketverwaltungsdateien zugreifen kann. Abhilfe: Beenden Sie eines der beiden Programme.

In seltenen Fällen tritt die *lock*-Warnung auch dann auf, wenn augenscheinlich kein anderes Paketverwaltungsprogramm mehr läuft. Die Ursache ist zumeist, dass ein Programm die *lock*-Datei beim Programmende nicht ordnungsgemäß entfernt hat. Gegebenenfalls löschen Sie die *lock*-Datei ganz einfach: `rm /var/lib/dpkg/lock`.



## 19.8 PackageKit

PackageKit ist eine Bibliothek zur Kommunikation mit diversen Paketverwaltungssystemen. Die Besonderheit von PackageKit besteht darin, dass es zu mehreren Paketverwaltungssystemen kompatibel ist, unter anderem zu APT, DNF, Yum und Zypper. PackageKit schafft so eine gemeinsame, distributionsunabhängige Basis für darauf aufbauende Benutzeroberflächen zur Paketverwaltung bzw. zur Durchführung von Updates. Beispielsweise greifen `gnome-software` oder Discover (KDE) auf PackageKit zurück.

PackageKit wird von allen gängigen Distributionen eingesetzt. In openSUSE beschränkt sich seine Aufgabe auf die Durchführung von Updates. PackageKit greift zur Erlangung von `root`-Rechten auf PolicyKit zurück (siehe [Abschnitt 12.4](#)).

PackageKit wird durch die Dateien `/etc/PackageKit/*` konfiguriert. In der Datei `PackageKit.conf` kann bei Bedarf mit `DefaultBackend` eingestellt werden, auf welches Paketverwaltungssystem PackageKit zurückgreifen soll. Im Regelfall erkennt PackageKit das Backend aber automatisch.

Interna und  
Konfiguration

Der Dämon `packagekitd` ist für die Koordination der PackageKit-Operationen erforderlich. Das Programm wird bei Bedarf automatisch von den PackageKit-Kommandos bzw. -Benutzeroberflächen gestartet – und auch wieder beendet, wenn es nicht mehr benötigt wird. Der Dämon läuft also nicht ständig als Hintergrunddienst.

Mit dem Kommando `pkcon` können Sie sämtliche Paketoperationen auch in einer Konsole ausführen oder durch ein Script automatisieren. Beachten Sie, dass das Kommando nicht von `root` ausgeführt werden darf! Sie müssen das Kommando als gewöhnlicher Benutzer starten. Soweit erforderlich, verwendet das Kommando PolicyKit, um `root`-Rechte zu erlangen. Falls Sie in einer Konsole verfolgen möchten, was PackageKit gerade macht, führen Sie `pkmon` aus.

Im Laufe der Zeit speichert PackageKit in `/var/cache/PackageKit` unzählige Dateien, die später nicht mehr benötigt werden. Der Speicherbedarf für dieses Verzeichnis kann mehrere GiB betragen. Abhilfe schafft das folgende Kommando, das den Cache aktualisiert, dabei aber gleichzeitig das maximale Alter der Dateien auf 0 reduziert (Option `-c -1`):

Cache aufräumen

```
root# pkcon refresh force -c -1
```

## 19.9 tar

Mit dem Kommando `tar` können Sie ähnlich wie unter Windows mit WinZIP komprimierte Dateiarhive erstellen und auspacken. Eine genauere Beschreibung des Kommandos und speziell seiner Anwendung für Backup-Aufgaben finden Sie in [Abschnitt 36.3](#), »Dateien komprimieren und archivieren«.

Vor allem bei erfahrenen Linux-Anwendern kommt es häufig vor, dass Linux-Software installiert werden soll, die nicht in Form eines Pakets einer bestimmten Distribution organisiert ist. Auch dabei kommt zumeist das `tar`-Format zur Anwendung.

gzip-  
komprimierte  
Archive

Mit `gzip` komprimierte Archive weisen üblicherweise die Dateikennungen `*.tgz` oder `*.tar.gz` auf. Die Archive müssen mit dem Programm `tar` auf Ihrem Rechner installiert werden. Beachten Sie beim Auspacken, dass `tar` relativ zum gerade aktuellen Verzeichnis arbeitet. Stellen Sie also sicher, dass Sie sich im richtigen Verzeichnis befinden!

```
root# tar -tzf archiv.tar.gz          (Inhalt des Archivs anzeigen)
root# tar -xzf archiv.tar.gz          (Dateien auspacken)
root# tar -xzf archiv.tar.gz "*.tex" (nur *.tex-Dateien auspacken)
root# tar -xzf archiv.tar.gz -C verz  (in ein Verzeichnis auspacken)
```

bzip2-  
komprimierte  
Archive

Immer häufiger wird zur Komprimierung der Archive das leistungsstärkere Programm `bzip2` verwendet. Sie erkennen derartige Archive an der Kennung `*.tar.bz2`. Zum Anzeigen bzw. Dekomprimieren müssen Sie nun statt `-z` die `tar`-Option `-j` verwenden, also beispielsweise `tar -tjf archiv.tar.bz2`.

Quellcode-Pakete

In vielen Fällen liegt die Software nur im Quellcode vor und muss vor ihrer Verwendung noch kompiliert werden. Das setzt voraus, dass Sie die wichtigsten Entwicklungswerkzeuge sowie alle erforderlichen Bibliotheken installiert haben (`gcc`, `make`, `devel`-Pakete etc., siehe [Abschnitt 20.2](#), »Programme selbst kompilieren«).

### Achtung

Die Installation von Software-Paketen durch `tar` umgeht die Paketverwaltung des jeweiligen Systems. Die RPM- bzw. DEB-Datenbanken wissen daher nichts von den Programmen, die Sie installiert haben. Daher sollten Sie es immer vorziehen, Pakete zu installieren, die speziell für die jeweilige Distribution vorbereitet wurden.

## 19.10 Umwandlung zwischen Paketformaten (alien)

Was können Sie tun, wenn ein Paket nur im RPM-Format zu finden ist, Sie aber unter Debian oder Ubuntu arbeiten? Was tun Sie, wenn Sie aus einem RPM-Paket nur eine einzige Datei ansehen möchten? Die Antwort lautet: Verwenden Sie `alien`. Dieses

Kommando wandelt Pakete zwischen verschiedenen Formaten um (RPM, DEB und tar-Archiv).

`alien` funktioniert leider nur bei einfachen Paketen problemlos. Wenn ein Paket dagegen Installations-Scripts oder andere spezifische Eigenheiten des jeweiligen Paketformats nutzt, wird die Installation des zuvor umgewandelten Pakets meist scheitern. Generell ist `alien` eher ein Werkzeug für Linux-Profis.

Das gewünschte Paketformat wird durch die Optionen `--to-deb` (Debian), `--to-rpm` (RPM) oder `--to-tgz` (tar-Archiv) angegeben. `alien` muss von `root` ausgeführt werden, damit die Besitzer und Zugriffsrechte der neuen Pakete richtig eingestellt werden. Das folgende Kommando wandelt ein Debian-Paket in ein RPM-Paket um:

```
root# alien --to-rpm paket.deb
```

Die folgenden Kommandos zeigen, wie Sie eine einzelne Datei aus einem RPM-Paket extrahieren. Dazu wandeln Sie das Paket zuerst in ein tar-Archiv um und verwenden dann `tar`, um die Datei daraus zu extrahieren und dann mit `less` anzuzeigen. (Statt `tar` können Sie natürlich auch den Dateimanager Konqueror oder Archivprogramme wie `ark` oder `file-roller` einsetzen. Diese Programme zeigen den Inhalt des Archivs in einer ansprechenden Benutzeroberfläche an.)

```
root# alien --to-tgz paket.rpm
root# tar -xzf paket.tgz ./usr/share/doc/packages/paket/TODO
root# less ./usr/share/doc/packages/paket/TODO
```

## 19.11 Verwaltung von Parallelinstallationen (alternatives)

Unter Linux stehen oft mehrere alternative Programme zur Auswahl, die denselben Zweck erfüllen und manchmal sogar denselben Kommandonamen nutzen: Drucksysteme, Editoren, Java-Umgebungen etc. In manchen Situationen ist es zweckmäßig, mehrere Varianten bzw. sogar mehrere Versionen ein- und desselben Programms parallel zu installieren. Sofern dabei jede Programmversion in einem eigenen Verzeichnisbaum landet, ist die Installation an sich ohne Konflikte möglich. Welche Programmversion kommt aber zum Einsatz, wenn der Anwender ein bestimmtes Kommando ausführt?

Zur Beantwortung dieser Frage verwenden viele gängige Distributionen ein zuerst von Debian entwickeltes Konzept, das auf symbolischen Links im Verzeichnis `/etc/alternatives` basiert. Die folgende Liste gibt an, in welchem Paket das `alternatives`-Verzeichnis und das dazugehörige Verwaltungskommando `update-alternatives` enthalten sind:

Debian, Ubuntu: Paket dpkg  
 Red Hat, Fedora: Paket chkconfig  
 SUSE: Paket update-alternatives

Am einfachsten ist das Konzept anhand eines Beispiels zu verstehen. Nehmen wir an, auf einem Rechner sind zwei Java-Versionen installiert. Java-Programme werden mit `java klasse` ausgeführt. Nun ist `/usr/bin/java` als Link auf `/etc/alternatives/java` realisiert. `/etc/alternatives/java` ist ein weiterer Link, der auf die gewünschte Java-Version verweist:

```
user$ ls -l /usr/bin/java
... /usr/bin/java -> /etc/alternatives/java
user$ ls -l /etc/alternatives/java
... /etc/alternatives/java ->
    /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
```

Die Verwaltung der Links erfolgt in der Regel automatisch durch Scripts bei der Paketinstallation. Dabei kommt das Kommando `update-alternatives` zur Anwendung. Unter Red Hat/Fedora ist das Kommando auch unter dem Namen `alternatives` verfügbar.

#### Alternativen auflisten

Mit `update-alternatives --display` stellen Sie fest, welche Versionen eines bestimmten Programms verfügbar sind und welche Version standardmäßig gilt. Die folgenden Zeilen zeigen das Ergebnis für `editor` auf einem Debian-System mit mehreren installierten Texteditoren. Die `slave`-Zeilen betreffen Kommandos, die dem eigentlichen Programm untergeordnet sind, und `man`-Seiten. `update-alternatives` aktualisiert bei einer Veränderung des Kommando-Links automatisch auch alle `slave`-Links.

```
root# update-alternatives --display editor
editor - Auto-Modus
  Link verweist zurzeit auf /usr/bin/joe
/bin/nano - Priorität 40
  Slave editor.1.gz: /usr/share/man/man1/nano.1.gz
/usr/bin/jmacs - Priorität 50
  Slave editor.1.gz: /usr/share/man/man1/jmacs.1.gz
  Slave editorrc: /etc/joe/jmacsrc
/usr/bin/joe - Priorität 70
  Slave editor.1.gz: /usr/share/man/man1/joe.1.gz
  Slave editorrc: /etc/joe/joerc
...
Gegenwärtig »beste« Version ist »/usr/bin/joe«.
```

Normalerweise erfolgt die Link-Verwaltung im Automatikmodus: Jedes installierte Paket enthält eine Prioritätsnummer. `update-alternative` aktiviert bei jeder (De-)Installation die Alternative mit der höchsten Priorität.

`update-alternatives -config` bestimmt die in Zukunft aktive Variante. Das Kommando liefert die Liste der zur Auswahl stehenden Alternativen, von denen Sie dann eine aktivieren. `update-alternatives` aktualisiert nun die Links. `update-alternatives -auto` führt bei Bedarf zurück in den Automatikmodus. Im folgenden Beispiel wird `jmacs` als `Defaulteditor` eingestellt:

Andere  
Alternative  
auswählen

```
root# update-alternatives --config editor
Es gibt 7 Auswahlmöglichkeiten für die Alternative editor,
welche /usr/bin/editor bereitstellen.
```

| Auswahl | Pfad           | Priorität | Status          |
|---------|----------------|-----------|-----------------|
| * 0     | /usr/bin/joe   | 70        | Auto-Modus      |
| 1       | /bin/nano      | 40        | manueller Modus |
| 2       | /usr/bin/jmacs | 50        | manueller Modus |

...

Drücken Sie die Eingabetaste, um die aktuelle Wahl[\*] beizubehalten, oder geben Sie die Auswahlnummer ein: **2**

```
update-alternatives: /usr/bin/jmacs wird verwendet, um
/usr/bin/editor (editor) im manuellen Modus bereitzustellen
```

Interne Verwaltungsinformationen zu den Links werden je nach Distribution im Verzeichnis `/var/lib/dpkg/alternatives` oder `/var/lib/rpm/alternatives` gespeichert.

## 19.12 Flatpak und Snap

In der Einleitung dieses Kapitels habe ich es bereits erwähnt: Die auf DEB- oder RPM-Paketen basierenden Paketverwaltungssysteme funktionieren gut, sind aber auch mit Einschränkungen verbunden. Software-Anbieter suchen seit Jahren nach einer Möglichkeit, Desktop-Anwendungen distributionsübergreifend so anbieten zu können, dass sie von Linux-Anwendern ohne umfassende Systemkenntnisse unkompliziert installiert werden können. Nur die Distribution als Basis verbleibt bis auf Weiteres unter der Kontrolle herkömmlicher Paketverwaltungssysteme.

Wie so oft in der Linux-Geschichte zeichnet sich auch dieses Mal ab, dass es nicht eine, sondern mehrere Lösungen gibt:

- ▶ Red Hat favorisiert *Flatpak* (<http://flatpak.org>), das im Gnome-Umfeld entwickelt wurde und zunächst `xdg-Apps` hieß. Als Testvehikel dient wie üblich Fedora, das schon seit Version 24 die Flatpak-Infrastruktur inkludiert.
- ▶ Canonical setzt auf die Eigenentwicklung *Snap* (<https://snapcraft.io>) und liefert die erforderlichen Pakete standardmäßig seit Ubuntu 16.04 aus. Snap wurde ursprünglich für Ubuntu Core entwickelt, eine Variante von Ubuntu für das Internet of Things. Snaps sind daher, anders als Flatpaks, auch für den Server-Einsatz

geeignet. Die Weitergabe von Snaps erfolgt über den Snap Store, der aktuell (Mitte 2017) aber nur kostenlose Snaps enthält.

Dieser Abschnitt behandelt Flatpak und Snap auf der Basis von Fedora 26 bzw. Ubuntu 17.04. Beide Systeme können optional auch auf den meisten gängigen Distributionen installiert werden.

**Sandboxing** Auch wenn es viele Unterschiede zwischen Flatpak und Snap gibt, so finden sich doch auch gemeinsame Nenner. Beide Systeme erlauben die Installation neuer Anwendungen auf Benutzerebene, also ohne die für `apt/dnf/zypper` erforderlichen `root`-Rechte. Gleichzeitig laufen die so installierten Programme in einer »Sandbox«, also mit eingeschränkten Rechten. Das soll Sicherheitsprobleme minimieren. Detaillierte Beschreibungen der Sandbox-Konzepte von Flatpak und Snap finden Sie hier:

<https://github.com/flatpak/flatpak/wiki/Sandbox>

<https://developer.ubuntu.com/en/snappy/guides/security>

Beachten Sie, dass das Sicherheitskonzept von Snap eine Kernel-Erweiterung voraussetzt, die erst seit Version 4.11 in den offiziellen Kernel aufgenommen wurde. In den Ubuntu-Kerneln ist die Erweiterung natürlich schon länger integriert, aber wenn Sie Snaps unter einer anderen Distribution verwenden, gelten dort laxere Sicherheitsregeln als unter Ubuntu.

**Platzbedarf** Flatpak und Snap haben derzeit aber einen großen Nachteil: Der Platzbedarf von so installierten Programmen ist immens, oft um ein Mehrfaches größer als bei einer herkömmlichen Installation. Das hat damit zu tun, dass die distributionsüberschreitende Ausführung der Programme nur gewährleistet werden kann, wenn die Programme alles erforderlichen Bibliotheken in der richtigen Version mitbringen.

Besonders extrem ist das bei Flatpak: Dort erfordern viele Anwendungen quasi nebenbei ein halbes Gnome-System – auch wenn Ihre Distribution ohnedies Gnome nutzt (womöglich sogar in der richtigen Version)! Wenn Sie Glück haben, werden die in eigenen Paketen organisierten Bibliotheken zumindest von mehreren Flatpaks gemeinsam genutzt. Mit etwas Pech verwendet das eine Flatpak Gnome 3.24 als Fundament, das nächste aber Gnome 3.26 – dann müssen alle Bibliotheken nochmals installiert werden.

Beachten Sie, dass der große Platzbedarf nicht nur Ihren Datenträger betrifft: Da die Bibliotheken nicht mit dem Rest des Systems geteilt werden können, erfordert die Ausführung von Anwendungen auch mehr Arbeitsspeicher als bei herkömmlichen Installationen.

### Persönliche Einschätzung

Flatpak und Snap versuchen, ein dringendes Problem der Linux-Welt zu lösen: Sie machen es einfacher, aktuelle Desktop-Anwendungen unabhängig von den offiziellen Paketquellen zu installieren.

Die aktuellen Implementierungen sind aber nicht überzeugend: Bei Flatpak stört vor allem die absurde Vergeudung von Ressourcen. (Eine Spotify-Installation erfordert 1 GiB Platz auf der Festplatte!)

Bei Snap wiederum hakt die Desktop-Integration an allen Ecken und Enden. Außerdem irritiert der Snap-Store-Zwang, zumal sich der »Store« noch in einem erbärmlichen Zustand befindet. Es gibt aktuell nicht einmal eine ordentliche Webseite.

Bevor sich eine der Lösungen als Standard etablieren kann, bleibt also noch viel zu tun. Die besseren Chancen gebe ich Flatpak, und zwar losgelöst von allen technischen Details. Canonical ist mit seiner eigenbrötlerischen Strategie schon mehrfach gescheitert (Upstart, Unity, Mir), und es sieht aktuell nicht so aus, als hätte es aus den Fehlern der Vergangenheit gelernt.

## Flatpak

Sofern die Flatpak-Infrastruktur zur Verfügung steht, starten Sie die Installation einer als Flatpak verpackten Anwendung im Webbrowser, indem Sie einen Link anklicken, der auf eine `.flatpakref`-Datei verweist. Eine Sammlung aktuell verfügbarer Flatpaks finden Sie hier:

Anwendung

<http://flatpak.org/apps.html>

Die Installation müssen Sie nun im Gnome-Programm *Software* bestätigen. Dort werden zuerst einige Details zur Anwendung angezeigt. (*Software* bietet aber aktuell keine Möglichkeit, nach Flatpaks zu suchen.) Anschließend können Sie das Programm wie ein gewöhnliches Programm in den Gnome-Aktivitäten suchen und starten und bei Bedarf zu den Favoriten hinzufügen (also im Dock verankern).

Alternativ können Sie Flatpaks mit dem gleichnamigen Kommando auch im Terminal installieren. Die folgenden Zeilen zeigen die Installation und Ausführung von Skype:

flatpak-Kommando

```
user$ flatpak install --from \
    https://s3.amazonaws.com/alexlarsson/skype-repo/skype.flatpakref
user$ flatpak run com.skype.Client
```

Bei vielen Anwendungen werden mehrere Flatpaks installiert: neben der eigentlichen Anwendung auch solche für die erforderlichen Bibliotheken (z. B. das Gnome-System). Das hat den Vorteil, dass mehrere Flatpaks manchmal Bibliotheken gemeinsam nutzen können.

Beachten Sie, dass `flatpak install` im Gegensatz zum Gnome-Programm *Software* die Installation standardmäßig auf Systemebene durchführt. Für eine Installation innerhalb des Benutzerverzeichnisses müssen Sie zusätzlich die Option `--user` angeben.

Das Kommando `flatpak` hilft auch bei der weiteren Administration aller installierten Flatpaks: `flatpak list` liefert eine Liste der Installationen, `flatpak update` aktualisiert alle Flatpaks, soweit es in der Quelle des jeweiligen Pakets Updates gibt etc. Mit `flatpak uninstall` können Sie einzelne Flatpaks wieder entfernen.

**Interna** Flatpak-Benutzerinstallationen landen im Verzeichnis `.local/share/flatpak`, Installationen auf Systemebene im Verzeichnis `/var/lib/flatpak`. Die unzähligen dort angelegten Verzeichnisse und Dateien verwenden häufig UUIDs als Namen, also lange hexadezimale Codes.

Für die Installation von Spotify wurden insgesamt über 46.000 Verzeichnisse, Dateien und Links eingerichtet. Der Platzbedarf betrug knapp ein GiB. Es ist nicht lange her, da reichte das für eine ganze Linux-Distribution aus!

Die Ausführung von Flatpak-Programmen setzt voraus, dass der Hintergrundprozess `flatpak-session-helper` läuft. Dieser Prozess, der beim Login durch `systemd` mit den Rechten des jeweiligen Benutzers gestartet wird, kommuniziert über DBUS mit den Flatpak-Kommandos und -Programmen.

## Snap

**Anwendung** Unter Ubuntu können Sie Snaps direkt im Gnome-Programm *Software* installieren. Das ist natürlich sehr anwenderfreundlich, aber zum einen ist das dort verfügbare Snap-Angebot aktuell minimal, zum anderen gibt *Software* keinerlei Informationen darüber, ob das zu installierende Programm ein gewöhnliches Paket oder ein Snap-Paket ist. Das merken Sie aber sofort, wenn Sie auf **INSTALLIEREN** klicken: Vor der ersten Installation müssen Sie auf <https://login.ubuntu.com> einen Ubuntu-One-Login einrichten und sich damit anmelden (siehe [Abbildung 19.4](#)).

Nach der Installation können Sie das Programm direkt aus *Software* heraus starten. Unter Unity finden Sie das Programm auch im Startmenü (Launcher) und können es im Dock verankern. In der Gnome-Version funktioniert die Integration mit Snap allerdings noch nicht: Die Suchfunktion der Gnome-Aktivitäten findet die Snap-Anwendung nicht, und laufende Snap-Anwendungen können nicht zu den Favoriten hinzugefügt und so im Dock verankert werden. (Beide Einschränkungen werden sicher in künftigen Ubuntu-Versionen behoben.) Sie müssen zum Start das im Folgenden beschriebene Kommando `snap run` verwenden.



**Abbildung 19.4** Vor der ersten Installation eines Snap-Paketes müssen Sie sich mit Ihrem Ubuntu-One-Konto anmelden.

Im Terminal administrieren Sie Snap durch das gleichnamige Kommando. `snap list` zählt alle installierten Snap-Anwendungen auf, `snap run` startet eine von ihnen:

```
user$ snap list
Name Version Rev Developer Notes
core 16-2 1689 canonical -
vlc daily 4 videolan -
user$ snap run vlc
```

`snap find` liefert einen aktuell recht kleinen Katalog weiterer verfügbarer Snaps. Mit `snap install` können Sie nun eines dieser Programme installieren. Eigentlich ist vorgesehen, dass Sie sich vorher einmal mit `snap login` anmelden. Bei meinen Tests mit zwei verschiedenen Ubuntu-One-Accounts und unter verschiedenen Ubuntu-Versionen ist dies allerdings durchgängig gescheitert. Laut man `snap` sollte ein Token für die Zugriffsberechtigung in `.snap/auth.json` gespeichert werden.

Wenn ein Snap-Login nicht möglich ist, müssen Sie administrative `snap`-Kommandos mit `sudo` ausführen:

```
user$ snap find
Name Version Developer Summary
anbox-installer 1 morphis Installer for the ...
vlc daily videolan The ultimate media player
docker 1.13.1-0 docker-inc The docker app deployment ..
lxd 2.13 canonical System container manager ...
rocketchat-server 0.55.1 rocketchat Group chat server for 100s, ...
keepassxc 2.1.4 keepassxreboot community driven port of ...
nextcloud 11.0.3snap1 nextcloud Nextcloud Server
hugo 0.20.6 hugo-authors Fast and Flexible Static ...
conjure-up 2.1.5 canonical Package runtime for ...
user$ sudo install hugo
```

snap refresh aktualisiert alle Snap-Anwendungen, soweit Updates zur Verfügung stehen. snap remove name löscht eine Snap-App.

**Mehr Snaps** snap find listet nur ganz wenige, offiziell unterstützte Apps auf. Tatsächlich enthält der App-Store ungleich mehr kostenlose Snap-Anwendungen, darunter aber auch unzählige Hello-World- und Test-Apps. Diese finden Sie mit uappexplorer-cli. Standardmäßig liefert aber auch dieses Programm nur 30 Ergebnisse. Abhilfe schafft die Option --size. Mit --search können Sie gezielt nach Snaps suchen und so z. B. LibreOffice entdecken:

```
user$ sudo snap install uappexplorer-cli
user$ uappexplorer-cli --type snap --size 10000
hello-world      This is a simple hello world example.
beagleboneblack  This package contains a example OEM part
APM:Plane        This snap provides the files needed to fly a plane ...
OS.js            A snap of OS.js, a nodejs based desktop inside ...
...
user$ uappexplorer-cli --type snap --search libre
libreoffice      LibreOffice is a powerful office suite ...
librealsense     librealsense ...
librealsense-ch... librealsense
emacs-tealeg     An extensible, customizable, free/libre ...
```

Alternativ können Sie auch auf der nicht offiziellen Seite <https://uappexplorer.com> nach Snaps suchen. Die Installation aus dem Webbrowser heraus führt aktuell aber noch zu Fehlermeldungen, auch wenn die Installation bei meinen Tests zumindest in manchen Fällen durchgeführt wurde.

**Interna** Die interne Verwaltung von Snaps erfolgt ganz anders als bei Flatpak. Snap-Anwendungen werden in Form von großen \*.snap-Dateien in /var/lib/snapd/snaps gespeichert:

```
user$< ls -lh /var/lib/snapd/snaps
-rw-r--r-- 1 root root 80M Mai 3 16:16 core_1689.snap
-rw-r--r-- 1 root root 11M Mai 4 08:07 hugo_75.snap
drwxr-xr-x 2 root root 4,0K Apr 6 10:32 partial
-rw-r--r-- 1 root root 116M Mai 3 16:18 vlc_4.snap
```

Der im Hintergrund laufende Snap-Dämon snapd bindet diese Dateien als squashfs-Dateisysteme an der Stelle /snap/xxx in den Verzeichnisbaum ein und macht die Anwendungen so zugänglich:

```
user$ df -h -t squashfs
Dateisystem  Größe Benutzt Verf. Verw% Eingehängt auf
/dev/loop0   80M    80M    0 100% /snap/core/1689
/dev/loop1   116M   116M    0 100% /snap/vlc/4
/dev/loop2   11M    11M    0 100% /snap/hugo/75
```

## 19.13 Distributionsspezifische Eigenheiten

Bei diesem Kapitel ist es schwierig, die allgemeine Beschreibung von Paketverwaltungswerkzeugen von den spezifischen Besonderheiten einzelner Distributionen zu trennen. In den bisherigen Abschnitten habe ich mich bemüht, Ihnen die Grundlagen und Kommandos zu beschreiben, die für mehrere Distributionen gemeinsam gelten. In diesem Abschnitt folgen nun einige Besonderheiten, die nur für eine bestimmte Distribution gelten.

### CentOS und RHEL

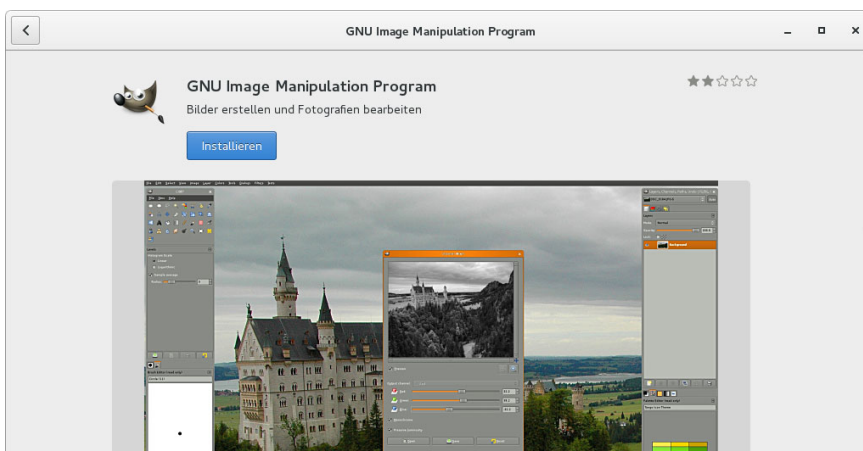
Die Paketauswahl in CentOS bzw. RHEL ist im Vergleich zu anderen Distributionen bescheiden. Abhilfe schafft das Einrichten der EPEL-Paketquelle (Extra Packages for Enterprise Linux), die eine riesige Sammlung gut gewarteter Zusatzpakete enthält. Bei CentOS 7 richten Sie EPEL mühelos mit dem folgenden Kommando ein: EPEL

```
root# yum install epel-release
```

Unter RHEL sowie bei älteren CentOS- oder RHEL-Installationen müssen Sie das `epel-release`-Paket von der EPEL-Webseite herunterladen und mit `rpm -i` installieren. RPM-Links zum Anklicken finden Sie hier:

<https://fedoraproject.org/wiki/EPEL>

Als grafische Benutzeroberfläche zur Installation von Paketen bzw. zur Durchführung von Updates ist wie in vielen anderen Distributionen mit einem Gnome-Desktopsystem das Programm *Software* vorgesehen (siehe [Abbildung 19.5](#)). »Software«



**Abbildung 19.5** Das Gnome-Programm »Software« lädt zur Installation von GIMP ein. »Software« ist hübsch anzusehen, aber bestenfalls Linux-Einsteigern zu empfehlen.

Das Programm ist einfach zu bedienen, aber einer Enterprise-Distribution unwürdig. So liefert die Suche nach *apache* keine Treffer. *Software* ist nur zur Installation von Desktop-Programmen gedacht. Verwenden Sie *yum*!

#### Update-Philosophie

Momentan gibt es keine Linux-Distribution, die über einen längeren Zeitraum Updates verspricht als RHEL bzw. CentOS. Mit den Updates aktualisieren Sie Ihre Distribution schrittweise von Version 7.0 auf 7.1, 7.2, 7.3 etc. Ein Update auf die nächste Major-Version, also auf 8.0, ist aber nicht vorgesehen.

Versprechen Sie sich von den Updates nicht zu viel: RHEL sieht fast ausschließlich Sicherheits-Updates vor. Hin und wieder machen auch Gnome oder wichtige Desktop-Anwendungen wie LibreOffice einen Versionsprung, aber das ist die Ausnahme von der Regel.

Interessanterweise bleibt auch die Kernelversion über die ganze Lebensdauer der Distribution unverändert. In diesem Punkt sollten Sie sich aber nicht von der Versionsnummer täuschen lassen: Die Red-Hat-Entwickler bauen ständig wichtige Sicherheits-Updates und fallweise auch neue Funktionen in den Kernel ein. Red Hat versucht auf diese Weise, maximale Stabilität mit etwas Modernität zu verbinden.

## Debian

**Paketgruppen** Debian-Pakete sind in drei Gruppen eingeteilt:

- ▶ *Main*: Das sind die Basispakete von Debian. Der Quellcode dieser Pakete ist unter einer Lizenz verfügbar, die den strengen Regeln des Debian-Projekts entspricht. Das garantiert, dass die Nutzung und Weitergabe wirklich frei im Sinne der Open-Source-Idee ist.
- ▶ *Contrib*: Pakete dieser Gruppe sind ebenfalls samt Quellcode frei verfügbar. Die Pakete können allerdings nur in Kombination mit *Non-Free*-Paketen verwendet werden. Das betrifft z. B. alle Programme, die auf Bibliotheken aufbauen, deren Lizenz in irgendeiner Weise Einschränkungen unterliegt.
- ▶ *Non-Free*: Pakete dieser Gruppe sind zwar kostenlos, ihre Lizenz entspricht aber nicht dem Open-Source-Ideal des Debian-Projekts. Zu vielen *Non-Free*-Paketen steht überhaupt kein öffentlicher Quellcode zur Verfügung.

Zudem unterscheidet Debian zwischen *Stable*-, *Testing*- und *Unstable*-Paketen:

- ▶ Als *Stable* gelten nur die Pakete, die Bestandteil der aktuellen, offiziellen Debian-Distribution sind. Diese Pakete sind in der Regel stabil und sicher, aber nicht besonders aktuell.
- ▶ Aktuellere Versionen können Sie installieren, wenn Sie die *Unstable*-Paketquellen einrichten. Wie der Name bereits ausdrückt, setzen Sie damit zu einem gewissen Grad die Stabilität Ihres Systems aufs Spiel. (Aber auch Ubuntu greift überwiegend

auf *Unstable*-Pakete zurück – zu viel Angst ist also nicht angebracht.) Die Summe der *Unstable*-Pakete stellt den aktuellen Debian-Entwicklungsstand dar. Für den *Unstable*-Zweig sind keine offiziellen Updates vorgesehen. Bekannte Fehler werden einfach durch die Veröffentlichung einer neuen Version behoben.

- Sozusagen als Übergangsstadium zwischen *Stable* und *Unstable* sind die *Testing*-Pakete gedacht. *Unstable*-Pakete, bei denen zehn Tage lang keine kritischen Fehler entdeckt werden, landen automatisch in *Testing* (allerdings nur, wenn auch alle abhängigen Pakete frei von kritischen Fehlern sind!).

Die drei Zweige haben jeweils Debian-interne Codenamen: Mit der Freigabe von Debian 9 steht »Stretch« für *Stable* (also Debian 9), »Buster« für *Testing* und »Sid« für *Unstable*. Der Codename für *Unstable* bleibt immer gleich. Wenn Debian 10 fertig wird, bekommt es den Namen »Buster«, und der *Testing*-Zweig erhält einen neuen Namen (voraussichtlich *Bullseye*).

Je nach Entwicklungsstand kann es vorübergehend auch *Experimental*-Pakete geben, um fundamental neue Konzepte auszuprobieren.

Damit das APT-System Zugriff auf alle Pakete samt Updates hat, sollte `sources.list` wie im folgenden Beispiel aussehen: `sources.list`

```
deb http://debian.inode.at/debian/          stretch          \
    main contrib non-free
deb http://security.debian.org/debian-security stretch/updates \
    main contrib non-free
```

Die beiden Paketquellen, deren Definition jeweils mit `deb` beginnt, sind hier nur aus Platzgründen über zwei Zeilen verteilt. `debian.inode.at` sollten Sie durch einen geografisch nahe liegenden Mirror-Server ersetzen.

Beachten Sie, dass nach einer Debian-Neuinstallation `non-free` nicht enthalten ist. Wenn Sie also Zugang zu *Non-Free*-Paketen wünschen, müssen Sie `non-free` selbst hinzufügen! Falls Sie auch Quellcodepakete installieren möchten, kopieren Sie die obigen Anweisungen und ersetzen jeweils `deb` durch `deb-src`.

Eine mögliche Alternative zur *testing*-Paketquelle ist die *backports*-Paketquelle. Die dort enthaltenen Pakete gelten als etwas stabiler, dafür ist die Auswahl deutlich geringer. Um die Backports-Paketquelle zu nutzen, fügen Sie die folgende Zeile zu `/etc/apt/sources.list` hinzu. Dabei ersetzen Sie wieder `debian.inode.at` durch einen nahegelegenen Mirror-Server. `Backports`

```
# in /etc/apt/sources.list
deb http://debian.inode.at/debian/ stretch-backports main contrib non-free
```

Um zu vermeiden, dass beim nächsten Update alle installierten Pakete durch neuere Backport-Versionen ersetzt werden, sind die Backports-Pakete durch die Einstellung `NotAutomatic: yes` in der Release-Datei der Paketquelle so gekennzeichnet, dass

sie eine geringere Priorität als normale Pakete haben. Deswegen müssen Sie bei der Installation von Backports-Paketen mit `aptitude` explizit die Option `-t stretch-backports` angeben.

```
root# aptitude -t stretch-backports install paketname
```

**DVD-Paketquelle** Nach einer Installation von einer DVD verbleibt in `sources.list` eine Zeile, die mit `deb cdrom:xxx` beginnt. Sollten Sie gerade über keinen Internetzugang verfügen, ist das praktisch – dann können Sie nämlich Pakete direkt von der DVD installieren. Ansonsten stört diese Zeile aber eher, weil Debian bei jeder Installation nach der DVD fragt, wenn diese gerade nicht im Laufwerk ist. Abhilfe: Kommentieren Sie die Zeile einfach aus, indem Sie das Zeichen `#` voranstellen.

## Fedora

**Zusätzliche Paketquellen** Zur Installation von proprietären Treibern, Multimedia-Codern etc. müssen Sie die Paketquelle RPM Fusion einrichten (siehe [Abschnitt 3.3](#), »Fedora«).

**Distributions-Updates im laufenden Betrieb** Fedora können Sie mit dem DNF-Modul `dnf-plugin-system-upgrade` von einer Version zur nächsten aktualisieren. Dazu führen Sie die folgenden Kommandos aus:

```
root# dnf install dnf-plugin-system-upgrade
root# dnf update
root# dnf repolist --releasever 26
root# dnf config-manager --set-disabled repo-name
root# dnf system-upgrade download --releasever 26
root# dnf system-upgrade reboot
```

Anstelle von `26` geben Sie natürlich die entsprechende Version an. Mit dem Kommando `dnf repolist` stellen Sie fest, ob alle von Ihnen in der vorigen Fedora-Version eingerichteten Paketquellen auch für die aktuelle Fedora-Version zur Verfügung stehen. Ist das nicht der Fall, müssen Sie die Paketquellen deaktivieren (`enabled=0` in `/etc/yum.repos.d/*.repo`). Sollten beim Update Probleme auftreten, können Sie versuchen, die betreffenden Pakete vorübergehend zu deinstallieren. Anschließend räumen Sie mit `dnf system-upgrade clean` auf und starten dann mit `dnf system-upgrade reboot` einen neuen Versuch. Weitere Details können Sie hier nachlesen:

[https://fedoraproject.org/wiki/DNF\\_system\\_upgrade](https://fedoraproject.org/wiki/DNF_system_upgrade)

Beachten Sie, dass Distributions-Updates fehleranfällig sind. Ich hatte schon derart oft Probleme damit, dass ich generell davon abrate. Führen Sie eine Neuinstallation durch!

## Linux Mint

Die Paketverwaltung von Linux Mint entspricht im Wesentlichen der von Ubuntu. Es gibt aber einige Unterschiede:

- ▶ Als Benutzeroberfläche zur Installation von Updates kommt die Mint-Eigenentwicklung `mintupdate` zum Einsatz. Bitte beachten Sie, dass dieses Programm standardmäßig nur Updates durchführt, die von den Mint-Entwicklern empfohlen werden. Das kann dazu führen, dass aus Sicherheitsgründen notwendige Updates unterdrückt werden. Wie Sie `linuxmint` sicherer konfigurieren, können Sie in [Abschnitt 3.4, »Linux Mint«](#), nachlesen.
- ▶ Für die restliche Paketverwaltung, d. h. zur Installation neuer Pakete bzw. zum Entfernen vorhandener Pakete, steht die Benutzeroberfläche `mintinstall` zur Verfügung.
- ▶ `/etc/apt/sources.list.d/official-package-repositories.list` enthält neben den Ubuntu-Paketquellen auch eine Paketquelle für Linux-Mint-spezifische Pakete.

## openSUSE

Unter openSUSE können Sie zur Paketverwaltung YaST verwenden. Dieses Konfigurationsprogramm stellt in der Gruppe SOFTWARE gleich fünf Module zur Auswahl. Gerade SUSE-Einsteigern fällt es manchmal schwer, unter den ähnlich lautenden Einträgen den richtigen zu finden. Der folgende Überblick gibt eine erste Orientierungshilfe. Die wichtigsten Module werden im Anschluss genauer vorgestellt.

Paketverwaltung  
mit YaST

- ▶ MEDIEN-ÜBERPRÜFUNG testet, ob eine Installations-CD oder -DVD frei von Fehlern ist.
- ▶ ONLINE-AKTUALISIERUNG startet YOU (YaST Online Update), um aktualisierte Pakete oder Sicherheits-Updates herunterzuladen. Wenn Sie unter KDE oder Gnome arbeiten, werden Updates aber mit desktopspezifischen Programmen durchgeführt, die mit PackageKit zusammenarbeiten.
- ▶ SOFTWARE INSTALLIEREN UND LÖSCHEN führt zum zentralen Paketverwaltungsmodul, mit dem Sie neue SUSE-Pakete installieren, vorhandene entfernen oder aktualisieren können etc. Dieses Modul werden Sie vermutlich am häufigsten einsetzen.
- ▶ SOFTWARE REPOSITORIES hilft bei der Verwaltung der Paketquellen. Mit HINZUFÜGEN • COMMUNITY/GEMEINSCHAFTS-REPOSITORIES können Sie populäre Paketquellen mit wenigen Mausklicks einrichten.
- ▶ ZUSATZ-PRODUKTE ermöglicht die Installation zumeist kommerzieller Programme, die auf einer SUSE-kompatiblen CD oder in SUSE-Paketquellen im Internet angeboten werden. Dieses Modul ist *nicht* dazu gedacht, um gewöhnliche, zu SUSE gehörende Pakete zu installieren – dazu verwenden Sie SOFTWARE INSTALLIEREN!

**Software  
installieren**

Für das YaST-Modul SOFTWARE • SOFTWARE INSTALLIEREN UND LÖSCHEN gibt es zwei grundverschiedene Implementierungen: Die KDE-Variante entspricht dem, was SUSE-Anwender seit vielen Jahren gewöhnt sind. Die neuere Gnome-Variante bietet dieselben Funktionen, ist aber etwas einfacher zu bedienen. Im Folgenden beziehe ich mich auf die KDE-Version.

Im Hauptfenster können Sie mit dem Button ANZEIGEN verschiedene Ansichten (Dialogblätter) öffnen und im weiteren Verlauf zwischen ihnen wechseln. In [Abbildung 19.6](#) ist nur ein Teil dieser Dialogblätter offen.

- ▶ **SUCHEN:** In dieser Ansicht können Sie nach Paketen suchen, deren Namen oder deren Funktion Sie kennen.
- ▶ **RPM-GRUPPEN:** Hier werden die Pakete in einer baumartigen Gruppenstruktur dargestellt (z. B. alle Pakete der Gruppe ENTWICKLUNG • WERKZEUGE • BUILDING). Die Orientierung in dem verzweigten Baum ist allerdings schwierig.
- ▶ **INSTALLATIONSZUSAMMENFASSUNG:** In dieser Ansicht sehen Sie, welche Pakete momentan zur Installation, zum Update oder zum Entfernen markiert sind.
- ▶ **PAKETGRUPPEN:** In dieser Ansicht werden Pakete angezeigt, die inhaltlich zusammenpassen, z. B. alle Spiele.
- ▶ **SCHEMATA:** Die Ansicht zeigt Pakete an, die funktionell zusammengehören, z. B. alle Pakete zum Einrichten eines Webservers (siehe [Abbildung 19.6](#)). Damit können Sie rasch und bequem alle Pakete zur Erfüllung einer bestimmten Aufgabe zur Installation auswählen. Im Prinzip verfolgen SCHEMATA und PAKETGRUPPEN dieselbe Idee, einzig die Gruppierungslogik ist anders.
- ▶ **SPRACHEN:** Diese Ansicht zeigt alle Lokalisierungspakete für eine bestimmte Sprache.
- ▶ **INSTALLATIONQUELLEN:** Diese Ansicht zeigt alle Pakete einer ausgewählten Paketquelle. Optional kann die oft sehr lange Paketliste mit einem zweiten Filter reduziert werden.

Der Paketmanager überprüft bei jeder Installation die Paketabhängigkeiten und aktiviert gegebenenfalls weitere Pakete zur automatischen Installation bzw. zum Update. Falls Abhängigkeitskonflikte auftreten, zeigt YaST verschiedene Vorschläge an, wie das Problem zu beheben ist.

Der Status von Paketen wird durch Symbole ausgedrückt. Der Zustand kann per Maus (verwenden Sie gegebenenfalls das Kontextmenü mit der rechten Maustaste!) oder per Tastatur verändert werden. Eine vollständige Beschreibung aller Symbole erhalten Sie mit HILFE • SYMBOLE.



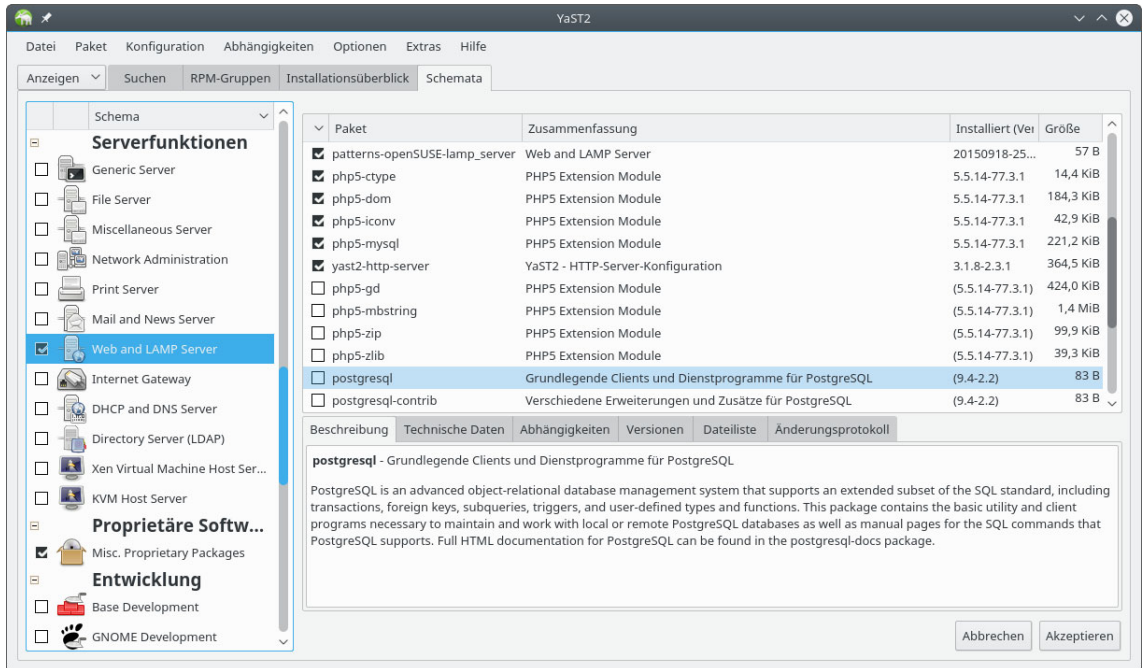


Abbildung 19.6 Installation von Software-Paketen mit YaST unter KDE

Das Verfahren *One-Click-Install* ermöglicht es, dass ein Klick im Webbrowser auf eine YMP-Datei die angeführten Paketquellen bleibend einrichtet und alle angegebenen Pakete installiert. Hinter den Kulissen kümmert sich ein YaST-Modul um diese Arbeiten. Vor dem Beginn der Installation müssen Sie natürlich das `root`-Passwort angeben. Bei den YMP-Dateien handelt es sich um einfache XML-Dateien, die alle erforderlichen Informationen (Paketquellen, Paketnamen etc.) enthalten. YMP steht dabei für *Yast Meta Package*. One-Click-Links lassen sich von `root` auch durch das Kommando `OCICLI` installieren:

One-Click-Install  
(\*ymp-Dateien)

```
root# OCICLI "http://eine-website.de/ein-tooles-programm.ymp"
```

Die ZYpp-Paketverwaltung greift auf die Installations-DVD sowie auf Paketquellen aus dem Internet zurück. Beim Einrichten weiterer Paketquellen hilft das YaST-Modul `SOFTWARE REPOSITORIES`. Falls Sie die Installations-DVD nach der Installation *nicht* weiter als Paketquelle verwenden möchten, können Sie die DVD-Paketquelle in diesem Modul explizit deaktivieren. Damit verschwinden die lästigen Fragen nach dem Einlegen der DVD bei jeder Paketinstallation.

Zusätzliche  
Paketquellen

`HINZUFÜGEN • COMMUNITY/GEMEINSCHAFT-REPOSITORIES` hilft beim Einrichten populärer Paketquellen. Die wichtigsten zwei sind *Packman* (aktuelle Multimedia-Pakete) und *nVidia* (proprietäre Grafiktreiber).

**Updates** openSUSE leistet sich den Luxus, zwei Update-Manager mitzuliefern: PackageKit und das YaST-Modul ONLINE-AKTUALISIERUNG. PackageKit ist dafür zuständig, aus dem jeweiligen Desktop-System heraus Updates durchzuführen, wobei es für Gnome und KDE jeweils eigene Oberflächen gibt. YOU (*Yast Online Update*) ist hingegen ein Relikt aus früheren SUSE-Zeiten. Sein Einsatz empfiehlt sich nur, wenn die KDE- oder Gnome-Update-Werkzeuge Probleme bereiten oder wenn Sie einen KDE- und Gnome-freien Desktop nutzen. Alternativ können Sie Updates auch in einem Terminalfenster durchführen:

```
root# zypper update
```

**Sprachpakete** Um die Lokalisierungspakete für eine bestimmte Sprache zu installieren, starten Sie das YaST-Modul SYSTEM • SPRACHE und wählen dort die gewünschten Sprachen aus.

**Distributions-Updates** SUSE bietet zwei Möglichkeiten, ein Distributions-Update durchzuführen:

- ▶ **Mit einer Installations-DVD:** Diese Update-Variante verläuft ähnlich wie eine Neuinstallation. Sie starten den Rechner mit der Installations-DVD neu. Das Installationsprogramm erkennt die vorhandene openSUSE-Version und bietet deren Aktualisierung als Option an. Diese Vorgehensweise ist gut ausgereift und wird weiterhin unterstützt. Sie hat aber einen großen Nachteil: Das System ist während des Updates, das circa eine halbe Stunde dauert, *offline*.
- ▶ **Im laufenden Betrieb:** Um ein Update im laufenden Betrieb durchzuführen, müssen Sie zuerst die vorhandenen Repositories löschen und durch neue Repositories für die gerade aktuelle openSUSE-Version ersetzen. Anschließend führen Sie in einer Konsole `zypper dup` aus. Danach ist ein Neustart erforderlich.

### openSUSE Tumbleweed (Rolling Release)

Linux-Experten und -Entwickler, die nach Möglichkeit immer die neuesten Software-Versionen einsetzen möchten, können mit dem Projekt Tumbleweed openSUSE zu einer Rolling-Release-Distribution machen. Das bedeutet, dass neue Programmversionen im Rahmen von täglichen Updates installiert werden und keine Notwendigkeit besteht, immer wieder eine neue openSUSE-Version zu installieren bzw. ein entsprechendes Distributions-Update durchzuführen.

Tumbleweed versucht, neue Software-Versionen zu aktivieren, sobald diese einigermaßen stabil sind. Dennoch sind beim Einsatz von Tumbleweed natürlich gelegentlich Probleme zu erwarten, wenn eine neue Software-Version doch noch Fehler enthält oder Inkompatibilitäten mit anderen Komponenten verursacht. Die Tumbleweed-Projektseite warnt vor dem Einsatz von Tumbleweed, wenn Sie proprietäre Treiber benötigen (NVIDIA, ATI), zusätzliche Paketquellen außer *Oss*, *Non-Oss* und *Update* aktiviert haben oder openSUSE in einer virtuellen Maschine ausführen.

Es bestehen zwei Möglichkeiten, Tumbleweed zu installieren. Die eine besteht darin, dass Sie direkt eine ISO-Datei von der Tumbleweed-Seite herunterladen und diese als Quelle für eine Neuinstallation verwenden. Bei der anderen Variante verwenden Sie eine gewöhnliche openSUSE-Installation als Ausgangspunkt, ändern dann die Paketquellen und führen schließlich ein Distributions-Update durch. Das ist umständlicher, erspart aber unter Umständen eine Neuinstallation. Die erforderlichen Kommandos sind ebenfalls auf der Tumbleweed-Seite zusammengefasst:

<https://en.opensuse.org/Portal:Tumbleweed>

## Ubuntu

Es gibt vier Ubuntu-Paketgruppen, die alle standardmäßig in `/etc/apt/sources.list` **Paketquellen** aktiviert sind:

- ▶ **Uneingeschränkt unterstützt (main):** Diese Pakete sind Bestandteil von Ubuntu, sind frei verfügbar und können ohne Lizenzprobleme frei weitergegeben werden. `main`-Pakete werden vom Ubuntu-Team gewartet und mit Updates versorgt.
- ▶ **Eingeschränktes Copyright (restricted):** `restricted`-Pakete enthalten Programme, die für die Funktion von Ubuntu Linux wichtig sind, die aber nicht als Open-Source-Software vorliegen. Dabei handelt es sich insbesondere um Hardware-Treiber für Grafik- und WLAN-Karten. Auch die `restricted`-Pakete werden offiziell von Ubuntu unterstützt und gewartet. Bei Sicherheits-Updates ist das Ubuntu-Team allerdings auf die Unterstützung der Firmen angewiesen, die die jeweiligen Programme zur Verfügung stellen.
- ▶ **Von der Gemeinschaft verwaltet (universe):** `universe`-Pakete enthalten Open-Source-Programme, die nicht vom Ubuntu-Team gewartet werden. Stattdessen kümmern sich Mitglieder der Ubuntu-Community um diese Pakete.
- ▶ **Unfrei (multiverse):** `multiverse`-Pakete enthalten Programme oder Daten, die nicht unter einer Open-Source-Lizenz stehen bzw. die nicht den Debian-Regeln für eine freie Verbreitung entsprechen. Die Pakete werden wie `universe`-Pakete nicht von Ubuntu gewartet.

Die `partner`-Paketquelle wird von der Firma Canonical gewartet. Sie enthält kommerzielle Programme, die kostenlos weitergegeben werden dürfen – zuletzt z. B. Skype. In der Vergangenheit wurden die Pakete der `partner`-Paketquelle leider nur schlecht gewartet; im Laufe der Zeit waren die wenigen verfügbaren Pakete auch noch veraltet. Am einfachsten aktivieren Sie die `partner`-Paketquelle im gerade erwähnten System-einstellungsmodul im Dialogblatt `ANDERE PROGRAMME`. **Partner-Pakete**

**Personal Package Archive** PPA steht für *Personal Package Archive* und ist eine Möglichkeit für Ubuntu-Entwickler, aktuelle Versionen von diversen Programmen zur Verfügung zu stellen, ohne diese offiziell in die Ubuntu-Paketquellen zu integrieren. PPAs bieten oft den schnellsten Weg, um neue Versionen von Grafiktreibern, LibreOffice, GIMP etc. relativ gefahrlos in Ubuntu zu integrieren. Weitere Informationen über PPAs finden Sie hier:

<https://launchpad.net/ubuntu/+ppas>

Um eine PPA-Paketquelle einzurichten und daraus ein Paket zu installieren, führen Sie einfach die folgenden Kommandos aus:

```
user$ sudo add-apt-repository ppa:name
user$ sudo add-get update
user$ sudo add-get install paketname
```

**Support-Status** Das attraktivste Merkmal der Ubuntu-LTS-Distributionen besteht in der fünfjährigen Update-Garantie. Dabei ist aber zu beachten, dass der lange Wartungszeitraum nur für Pakete aus den Gruppen *main* und *restricted* gilt! In der Praxis ist es aber oft notwendig, auch Pakete aus den Gruppen *universe* oder *multiverse* bzw. aus vollkommen anderen Paketquellen zu installieren. Damit wird es aber immer schwieriger, einen Überblick zu bewahren, für welche Pakete es noch Updates gibt.

Eine große Hilfe ist diesbezüglich das Kommando `ubuntu-support-status`. Bei einem Aufruf ohne weitere Parameter liefert es einen Überblick, wie viele Pakete über welchen Zeitraum gewartet werden:

```
root# ubuntu-support-status
Sie haben 9 Pakete (1.3%), die bis April 2019 (3y) unterstützt werden
Sie haben 638 Pakete (91.4%), die bis April 2021 (5y) unterstützt werden
Sie haben 8 Pakete (1.1%), die bis Mai 2022 (5y) unterstützt werden
Sie haben 3 Pakete (0.4%), die nicht/nicht mehr heruntergeladen werden können
Sie haben 40 nicht unterstützte Pakete (5.7%)
```

Noch mehr Details erhalten Sie, wenn Sie an das Kommando die Optionen `--show-unsupported`, `--show-supported` oder `--show-all` übergeben. Das Kommando listet alle Pakete auf, die in eine bestimmte Gruppe gehören.

```
root# ubuntu-support-status --show-unsupported
Nicht mehr herunterzuladen:
  libmysqlclient18 php5-common php5-gd
```

```
Nicht unterstützt:
  fail2ban fonts-lyx glances joe letsencrypt libgsasl7 ...
  spamass-milter ttf-bitstream-vera
```

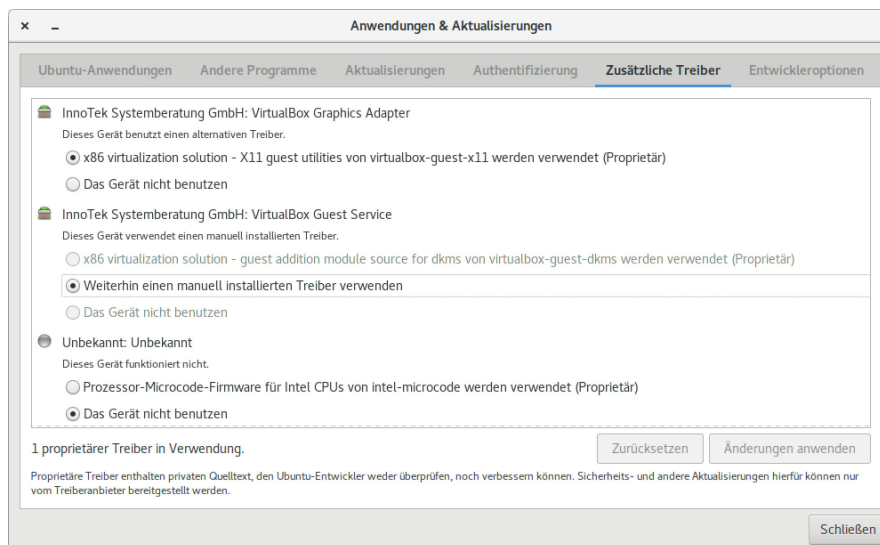
Hintergrundinformationen zur Langzeitpflege von Ubuntu-Paketen und Kritik an der mitunter unklaren Informationspolitik von Canonical finden Sie in meinem Blog sowie auf *heise.de*:

<https://kofler.info/die-lts-frage>

<https://heise.de/-3179960>

Das Programm *Anwendungen & Aktualisierungen* (software-properties) hilft bei der Installation von proprietären Treibern. Nach dem Start des Programms wechseln Sie in das Dialogblatt ZUSÄTZLICHE TREIBER. Das Programm analysiert nun Ihre Hardware und bietet gegebenenfalls passende Treiber zur Installation an (siehe Abbildung 19.7).

Proprietäre  
Treiber



**Abbildung 19.7** Installation von proprietären Treibern unter Ubuntu

Im Notfall, also beispielsweise bei einem Versagen des Grafiksystems, können Sie die Treiberinstallation mit dem Kommando `ubuntu-drivers` auch im Textmodus durchführen:

- ▶ `ubuntu-drivers devices` zeigt an, für welche Hardware-Komponenten welche Treiber zur Auswahl stehen.
- ▶ `ubuntu-drivers list` zeigt die aktuell aktiven Treiber an.
- ▶ `ubuntu-drivers autoinstall` führt eine automatische Installation bzw. Aktualisierung aller geeigneten Treiber durch.
- ▶ `ubuntu-drivers debug` zeigt den Status der Treiber und gegebenenfalls Fehlermeldungen an.

Das Kommando bietet allerdings keine Möglichkeit, um manuell Treiber auszuwählen. Um einen Treiber abweichend von `ubuntu-drivers autoinstall` zu installieren bzw. wieder zu entfernen, müssen Sie die betreffenden Pakete mit `apt` selbst installieren:

```
root# ubuntu-drivers devices
model    : MCP89 [GeForce 320M]
vendor   : NVIDIA Corporation
modalias : pci:v000010DEd000008A4sv0000106Bsd000000C0bc03sc00i00
driver   : xserver-xorg-video-nouveau - distro free      builtin
driver   : nvidia-304                  - distro non-free
driver   : nvidia-304-updates          - distro non-free
driver   : nvidia-340                  - distro non-free recommended
driver   : nvidia-340-updates          - distro non-free
...
root# apt install nvidia-340
```

**Sprachpakete** Mit dem Programm *Sprachen* (Kommando `gnome-language-selector` aus dem Paket `language-selector-gnome`) können Sie Sprachpakete für die aktuelle oder für eine weitere Sprache installieren bzw. vervollständigen und die Standardsprache einstellen. Die Änderung der Standardsprache wird beim nächsten Login wirksam. Das Programm kümmert sich allerdings nur um direkt zu Ubuntu gehörige Programme. Wenn Sie auch KDE-Programme installiert haben, müssen Sie für deren Lokalisierung das Paket `kde-l10n-de` installieren.

**Updates** Wenn Ubuntu Software-Updates feststellt, startet es automatisch das Programm *Aktualisierungsverwaltung* (Kommando `update-manager`). Es zeigt die verfügbaren Updates an und installiert diese gegebenenfalls.

**Distributions-Updates** Durch das Update-System werden normalerweise nur einzelne Programme aktualisiert, nicht aber die ganze Distribution. Sobald das Update-System eine neue Ubuntu-Version erkennt, fragt es, ob es ein vollständiges Distributions-Update durchführen soll. Antworten Sie nicht leichtfertig mit JA! Distributions-Updates dauern relativ lange und sind häufig mit Problemen verbunden.

Bei Bedarf können Sie das Distributions-Update auch manuell mit `do-release-update -m desktop` (für Desktop-Systeme) bzw. `-m server` (für Server) starten. Bei Ubuntu LTS-Versionen sind Release-Updates nur für die nächste LTS-Version vorgesehen, also z. B. von 16.04 auf 18.04. Wenn Sie ein Update auf eine Nicht-LTS-Version durchführen möchten, müssen Sie vorher in `/etc/update-manager/release-upgrades` die Variable `Prompt` von `lts` auf `normal` stellen.

# Kapitel 20

## Bibliotheken und Java

Im Mittelpunkt dieses Kapitels stehen Bibliotheken, die zur Ausführung von Programmen erforderlich sind. Die meisten Linux-Programme stehen in kompilierter Form zur Verfügung und greifen auf diverse Bibliotheken zurück, die dynamisch bei Bedarf geladen werden. Der erste Abschnitt dieses Kapitels führt in die unter Linux übliche Bibliotheksverwaltung ein.

Wenn Sie mit gängigen Distributionen arbeiten, installieren Sie zumeist nur fertig kompilierte Programme in Form von sogenannten Binärpaketen. Wenn Sie allerdings ganz neue Programmversionen oder selten benutzte Programme einsetzen möchten, kann es sein, dass Sie keine vorkompilierte Version des Programms zum Download finden. In solchen Fällen müssen Sie den Quellcode (meist in den Sprachen C oder C++) herunterladen und das Programm selbst kompilieren. [Abschnitt 20.2](#) gibt dazu einige einführende Tipps, ohne aber im Detail auf das unerschöpfliche Thema *Programmentwicklung unter Linux* einzugehen. (Einen ungleich tieferen Einblick bietet das 1400-seitige Buch *Linux-Unix-Programmierung*, das im Rheinwerk Verlag erschienen ist.)

Das Kapitel beschreibt auch, wie unter Linux Java-Programme ausgeführt werden. Dazu muss eine Java-Laufzeitumgebung installiert werden.

Scripts, die von einem Interpreter ausgeführt werden, sind nicht Thema dieses Kapitels. Unter Linux sind diverse Script-Sprachen üblich, unter anderem Perl, Python, PHP sowie die Shell `bash`. Dieses Buch geht allerdings nur auf die `bash` ausführlich ein (siehe [Kapitel 10](#)).

### 20.1 Bibliotheken

Praktisch alle Linux-Programme verwenden dieselben Standardfunktionen, beispielsweise zum Zugriff auf Dateien, zur Ausgabe am Bildschirm, zur Unterstützung von X etc. Es wäre sinnlos, wenn jedes noch so kleine Programm all diese Funktionen unmittelbar im Code enthalten würde – riesige Programmdateien wären die Folge. Stattdessen bauen die meisten Linux-Programme auf sogenannten *Shared Libraries*

auf: Bei der Ausführung eines Programms werden automatisch die erforderlichen Bibliotheken geladen. Der Vorteil: Wenn mehrere Programme Funktionen derselben Bibliothek nutzen, muss die Bibliothek nur einmal geladen werden.

Bibliotheken spielen eine zentrale Rolle dabei, ob und welche Programme auf Ihrem Rechner ausgeführt werden können. Fehlt auch nur eine einzige Bibliothek bzw. steht sie in einer zu alten Version zur Verfügung, kommt es beim Programmstart zu einer Fehlermeldung. Damit Sie in solchen Fällen nicht ganz hilflos den Tiefen der Linux-Internas ausgeliefert sind, vermittelt dieser Abschnitt einige Grundlageninformationen zu Bibliotheken.

**glibc** Zu den wichtigsten und grundlegendsten Linux-Bibliotheken zählt die GNU C Library (glibc), die mitunter auch als libc 6 bezeichnet wird. Im Sommer 2017 lag die glibc in der Version 2.25 vor. Normalerweise gibt es pro Jahr zwei Versions-Updates.

#### Dynamisch und statisch gelinkte Programme

Gewöhnliche Programme greifen wie oben beschrieben auf Bibliotheken zurück. Diese Bibliotheken werden erst zur Laufzeit dynamisch geladen und sind vom Konzept her mit Windows-DLLs (Dynamic Link Libraries) zu vergleichen.

Beim Kompilieren eines Programms besteht auch die Möglichkeit, Libraries statisch zu linken. Das bedeutet, dass die Library-Funktionen direkt in den Programmcode integriert werden. Die Programmdatei wird dadurch größer, ist aber nicht mehr von irgendwelchen Libraries abhängig. Das ist praktisch, um Programme unkompliziert weiterzugeben.

### Bibliotheken automatisch laden

Sofern Sie Linux als Anwender und nicht als Programmierer nutzen, werden Sie mit Bibliotheken nur dann konfrontiert, wenn diese fehlen. Meistens treten solche Probleme auf, wenn Sie manuell, also ohne Paketverwaltungswerkzeuge, ein neues Programm installieren. Beim Versuch, das Programm zu starten, erscheint eine Fehlermeldung, in der das Fehlen einer bestimmten Library angezeigt wird.

#### Bibliothekskliste feststellen

Dem Kommando `ldd` wird als Parameter der vollständige Dateiname des Programms übergeben. Als Reaktion listet `ldd` alle Libraries auf, die das Programm benötigt. Außerdem wird angegeben, wo sich eine passende Library befindet und welche Libraries fehlen bzw. nur in einer veralteten Version zur Verfügung stehen:

```
user$ ldd /bin/cp
linux-vdso.so.1 (0x00007ffc479e000)
libselinux.so.1 => /lib64/libselinux.so.1 (0x00007f5533dd1000)
libacl.so.1 => /lib64/libacl.so.1 (0x00007f5533bc8000)
libattr.so.1 => /lib64/libattr.so.1 (0x00007f55339c3000)
libc.so.6 => /lib64/libc.so.6 (0x00007f55335f2000)
libpcre.so.1 => /lib64/libpcre.so.1 (0x00007f553380000)
```



```
libdl.so.2 => /lib64/libdl.so.2 (0x00007f553317a000)
/lib64/ld-linux-x86-64.so.2 (0x000055cb0d47e000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f5532f5c000)
```

Bei KDE- und Gnome-Programmen ist die Bibliotheksliste wesentlich länger. Das ist auch der Grund, warum der Start dieser Programme etwas länger dauert. Wenn `ldd` hingegen das Ergebnis *not a dynamic executable* liefert, handelt es sich um ein Programm, das alle erforderlichen Bibliotheken bereits enthält, also um ein statisch gelinktes Programm:

```
root# ldd /usr/bin/firefox
not a dynamic executable
```

Kurz einige Informationen zur Nomenklatur der Libraries: Das Kürzel `.so` weist darauf hin, dass es sich um eine Shared Library handelt (im Gegensatz zu `.a` für statische Libraries). Die folgende Ziffer gibt die Hauptversionsnummer an. `cp` benötigt also Version 6 der `libc`-Bibliothek.

**Bibliotheks-  
namen**

In den Verzeichnissen `/lib`, `/lib64`, `/usr/lib` etc. befinden sich oft Links von der Hauptversion auf die tatsächlich installierte Version. So benötigt `cp` die Bibliothek `ld-linux-x86-64` in der Version 2. Tatsächlich ist aber die dazu kompatible Version 2.25 installiert:

```
user$ ls -l /lib64/ld-*
... /lib64/ld-2.25.so
... /lib64/ld-linux-x86-64.so.2 -> ld-2.25.so
```

Beim Start eines Programms ist der sogenannte *Runtime Linker* `ld.so` dafür zuständig, alle Bibliotheken zu finden und zu laden. Dazu wertet der Linker die Datei `/etc/ld.so.cache` aus. Diese Binärdatei enthält alle relevanten Bibliotheksdaten, also Versionsnummern, Zugriffspfade etc. Der Zweck dieser Datei besteht darin, dem Linker eine langwierige Suche nach den Bibliotheken zu ersparen.

**Programmstart**

`/etc/ld.so.cache` wird vom Programm `ldconfig` erzeugt. `ldconfig` wertet seinerseits `/etc/ld.so.conf` aus. Diese Datei enthält eine Liste mit Pfadangaben bzw. Querverweise auf andere Konfigurationsdateien.

Die Verzeichnisse `/lib` und `/usr/lib` werden auf jeden Fall berücksichtigt und fehlen daher zumeist in `ld.so.conf` bzw. in den weiteren Konfigurationsdateien. Wenn außer `/lib` und `/usr/lib` keine weiteren Verzeichnisse zu berücksichtigen sind, kann `ld.so.conf` auch ganz fehlen.

Bei manchen Distributionen wird `ldconfig` bei jedem Rechnerneustart ausgeführt, um so sicherzustellen, dass die Cache-Datei auf dem aktuellsten Stand ist. `ldconfig` muss unbedingt ausgeführt werden, wenn neue Bibliotheken manuell installiert werden! Andernfalls sind die Bibliotheken für das System nicht sichtbar. Falls sich

die Bibliotheken in einem neuen Verzeichnis befinden, muss außerdem die Datei `/etc/ld.so.conf` entsprechend ergänzt werden. Bei der Installation von Bibliothekspaketen kümmert sich in der Regel der Paketmanager um diese Aufgaben.

`ld.so` berücksichtigt zusätzlich alle in der Umgebungsvariablen `LD_LIBRARY_PATH` enthaltenen Verzeichnisse. Die Verzeichnisse müssen durch Doppelpunkte getrennt sein. Bei vielen Distributionen ist diese Umgebungsvariable leer.

### 32- und 64-Bit-Bibliotheken

Die meisten gängigen Distributionen gibt es momentan in zumindest zwei Ausführungen: für Intel/AMD-kompatible 32-Bit-Prozessoren und für Intel/AMD-kompatible 64-Bit-Prozessoren. Bei 32-Bit-Distributionen gibt es naturgemäß nur 32-Bit-Bibliotheken. Dies gilt aber leider nicht analog für 64-Bit-Distributionen: Nach wie vor gibt es Programme, die sich nicht für 64-Bit-Systeme kompilieren lassen.

**/lib und /lib64** Zur Ausführung von 32-Bit-Programmen auf 64-Bit-Distributionen sind 32-Bit-Bibliotheken erforderlich. Um Konflikte zu vermeiden, werden die Bibliotheken in unterschiedliche Verzeichnisse installiert. Im Linux-Jargon heißt diese Vorgehensweise *Multi Architecture (Multiarch)* bzw. exakter *Bi-architecture*, weil mehrere bzw. zwei Prozessorarchitekturen parallel unterstützt werden. Bei den meisten Distributionen sind die Verzeichnisse `/lib` und `/lib64` üblich, um zwischen 32- und 64-Bit-Bibliotheken zu differenzieren. Diese Doppelgleisigkeit ist natürlich mit Nachteilen verbunden: Die doppelte Installation von Bibliotheken erfordert zusätzlichen Platz auf der Festplatte und macht die Wartung komplizierter.

**CPU-spezifische Verzeichnisse** Eine andere Lösung besteht darin, innerhalb von `/usr/lib` Unterverzeichnisse für die jeweilige CPU-Architektur zu schaffen, z. B. `/usr/lib/x86_64-linux-gnu` (Debian ab Version 7, Ubuntu ab Version 12.04). Der Vorteil dieses Ansatzes besteht darin, dass er über die Intel/AMD-Welt hinaus auch auf ganz andere CPU-Architekturen übertragbar ist.

Die Multiarch-Idee erstreckt sich über Verzeichnispfade hinweg auch auf die Paketverwaltung: Es muss zum Beispiel möglich sein, ein und dasselbe Paket bei Bedarf mehrfach zu installieren, einmal in der 32-Bit- und einmal in der 64-Bit-Version. Zur klareren Unterscheidung kann dem Paketnamen die Architektur mit einem Doppelpunkt angehängt werden, z. B. `gvfs:i386` (bezeichnet die 32-Bit-Version des `gvfs`-Pakets). Auch die Abhängigkeitsüberprüfung muss nun doppelläufig arbeiten (also für 32- und 64-Bit-Abhängigkeiten). Das `dpkg`-Kommando unterstützt den Multiarch-Ansatz bereits gut.

## Prelinking

Beim Start eines Programms, das auf dynamische Bibliotheken zurückgreift, muss eine Verbindung zwischen dem Programm und den Bibliotheken hergestellt werden. Dieser Vorgang wird als *Linking* bezeichnet. Er beansprucht bei komplexen Programmen einige Zeit.

Das Programm `prelink` kann die erforderlichen Link-Informationen im Voraus ermitteln, was beim ersten Mal ebenfalls lange dauert. Bei diesem Vorgang müssen alle ausführbaren Programme durchsucht werden. Welche Verzeichnisse für Programme und Bibliotheken `prelink` berücksichtigt, wird durch die Datei `/etc/prelink.conf` konfiguriert. Weitere Optionen können Sie in `/etc/sysconfig/prelink` bzw. `/etc/default/prelink` (Debian, Ubuntu) einstellen.

In der Folge kann jedes so präparierte Programm schneller auf seine Bibliotheken zugreifen. Die Prelinking-Informationen müssen allerdings jedes Mal aktualisiert werden, wenn eine Bibliothek aktualisiert wird.

Prelinking ist allerdings auch mit diversen Nachteilen verbunden. Wegen der Veränderung ausführbarer Dateien ist es anschließend nicht mehr möglich, die Integrität dieser Dateien zu kontrollieren. Dieser und andere Gründe haben dazu geführt, dass Prelinking bei den meisten Distributionen nicht standardmäßig aktiv ist. Weitere Informationen finden Sie unter:

**Nachteile**

<https://pagure.io/fesco/issue/1183>

Um Prelinking unter Debian und Ubuntu zu nutzen, müssen Sie das Paket `prelink` installieren und in `/etc/default/prelink` die Einstellung `PRELINKING=yes` vornehmen. `prelink` wird täglich durch einen Cron-Job ausgeführt.

**Debian, Ubuntu**

Bei aktuellen Fedora-Versionen steht Prelinking nicht mehr zur Verfügung.

**Fedora**

Um Prelinking in RHEL 7 bzw. CentOS 7 auch in Version 7 zu nutzen, müssen Sie das Paket `prelink` installieren. Die Prelinking-Informationen werden regelmäßig aktualisiert (Cron-Job `/etc/cron.daily/prelink`, Konfigurationsdatei `/etc/sysconfig/prelink`).

**RHEL und CentOS**

In aktuellen openSUSE-Distributionen fehlt das `prelink`-Paket. Sie können es aber aus einem Entwickler-Repository manuell installieren:

**openSUSE**

<http://software.opensuse.org/download.html?project=devel:tools&package=prelink>

## 20.2 Programme selbst kompilieren

Es gibt zumeist nur zwei Gründe dafür, Linux-Programme selbst zu kompilieren: Entweder finden Sie für das gewünschte Programm in Ihrer Distribution kein Binärpaket mit dem fertig kompilierten Programm oder Sie möchten das Programm mit einer vom Standard abweichenden Konfiguration kompilieren.

- Voraussetzungen** Bevor Sie zur Tat schreiten, müssen einige Voraussetzungen erfüllt sein:
- ▶ Die *GNU Compiler Collection* (Pakete `gcc` und `gcc-c++`) muss installiert sein. Diese Pakete enthalten Compiler für C und C++.
  - ▶ Hilfswerkzeuge wie `make`, `automake`, `autoconf` etc. müssen installiert sein. Diese Programme sind für die Konfiguration und Durchführung des Kompilationsprozesses erforderlich.
  - ▶ Die Entwicklerversionen diverser Bibliotheken müssen installiert sein. Die Namen der entsprechenden Pakete enden bei Red Hat, Fedora und SUSE üblicherweise auf `-devel`, bei Debian und Ubuntu auf `-dev`. Beispielsweise enthält `glibc-devel` bzw. `libc6-dev` die Entwicklungsdateien für die `glibc`-Basisbibliothek. Welche Entwicklerpakete Sie sonst noch brauchen, hängt von der Natur des Programms ab, das Sie kompilieren möchten. Fehlermeldungen, in denen sich der Compiler oder Linker über fehlende Bibliotheken beklagt, sind ein eindeutiges Indiz dafür, dass Sie ein wichtiges Entwicklerpaket übersehen haben.
- Debian, Ubuntu** Bei Debian und Ubuntu definiert das Metapaket `build-essential` Abhängigkeiten für die wichtigsten Entwicklerpakete. Deswegen führt die Installation von `build-essential` automatisch zur Installation diverser weiterer Pakete, die zusammen die Grundausrüstung für die Programmentwicklung in C/C++ bilden. Unter Ubuntu sind die durch `build-essential` definierten Pakete standardmäßig installiert; unter Debian führen Sie `apt-get install build-essential` aus.
- Fedora** Um die Grundvoraussetzungen für die Programmentwicklung in Fedora zu erfüllen, führen Sie am einfachsten `yum groupinstall development-tools` aus. Auch für die KDE- und Gnome-Programmentwicklung gibt es eigene Paketgruppen: `kde-software-development` und `gnome-software-development`.
- RHEL und CentOS** Unter RHEL und CentOS gibt es wie in Fedora eine Paketgruppe mit Entwicklungswerkzeugen. Ihr Name lautet allerdings einfach `development`.
- openSUSE** Angehende openSUSE-Entwickler installieren in YaST alle Pakete des Paketschemas `GRUNDLEGENDE ENTWICKLUNGSUMGEBUNG`. Falls Sie vorhaben, KDE- oder Gnome-Programme zu entwickeln, installieren Sie auch die Selektionen `KDE-` bzw. `GNOME-ENTWICKLUNG`. Wenn Sie `zypper` vorziehen, führen Sie `zypper install -t pattern devel_basis` bzw. `devel_kde` bzw. `devel_gnome` aus.

## Code auspacken

Im Internet finden Sie den Quellcode zumeist in komprimierten TAR-Archiven. Nach dem Download entpacken Sie den Code in ein lokales Verzeichnis: tar

```
user$ tar xzf name.tar.gz      (für .gz oder .tgz)
user$ tar xjf name.tar.bz2    (für .bz2)
user$ tar xJf name.tar.xz     (für .xz)
user$ cd name
```

Eine Alternative zu den TAR-Archiven sind Quellcodepakete, die exakt den Code enthalten, aus dem ein bestimmtes Programm Ihrer Distribution kompiliert wurde. Die Quellcodepakete finden Sie in der Regel auf dem (FTP-)Server Ihrer Distribution. Bei Distributionen auf Basis von RPM-Paketen befinden sich die Quellcodedateien in SRPM-Paketen mit der Dateikennung `*.src.rpm`. Zur Installation führen Sie wie üblich `rpm -i` aus: SRPM-Pakete

```
root# rpm -i name.src.rpm
```

Es hängt von der Distribution ab, wo der Quellcode nun tatsächlich landet:

```
Fedora, Red Hat:   /usr/src/redhat/
openSUSE:          /usr/src/packages/
```

- ▶ `SOURCES/name.tar.xxx` enthält den eigentlichen Code. Das TAR-Archiv muss wie oben beschrieben entpackt werden.
- ▶ `SOURCES/name-xxx.patch` (Red Hat) oder `SOURCES/name.dif` (openSUSE) enthält distributionsspezifische Veränderungen am ursprünglichen Code. Wenn Sie die Codedateien entsprechend ändern (patchen) möchten, führen Sie das folgende Kommando aus:

```
user$ cd name-quellcodeverzeichnis
user$ patch < name.dif/patch
```

Je nachdem, welches Verzeichnis gerade aktuell ist und wie die Verzeichnisangaben innerhalb der Patch-Datei sind, müssen Sie zusätzlich die Option `-p1` angeben (siehe `man patch`).

- ▶ `SPECS/name.spec` enthält eine Paketbeschreibung, die auch zur Erstellung von RPM-Paketen dient. Wenn Sie aus selbst kompilierten Programmen wieder ein RPM-Paket erstellen möchten, müssen Sie dazu das Kommando `rpmbuild` einsetzen, auf das ich hier aber nicht eingehe. Lesen Sie `man rpmbuild`!

Bei Debian-basierten Distributionen befindet sich der Quellcode in mehreren Dateien, die Sie am besten mit `apt-get source` in das aktuelle Verzeichnis installieren: Debian-  
Quellcodepakete

```
user$ apt-get source paketname
```

Im aktuellen Verzeichnis finden Sie nun drei neue Dateien und ein Verzeichnis:

- ▶ `paketname.dsc` enthält eine Kurzbeschreibung des Pakets.
- ▶ `paketname.orig.tar.gz` enthält ein TAR-Archiv mit dem ursprünglichen Quellcode des Programmentwicklers.
- ▶ `paketname.diff.gz` enthält alle Debian- bzw. Ubuntu-spezifischen Änderungen am Originalquellcode.
- ▶ Das neue Verzeichnis `paketname/` enthält schließlich den bereits extrahierten Inhalt von `paketname.diff.gz`, wobei alle Änderungen aus der `diff`-Datei bereits ausgeführt wurden.

### Programm kompilieren

Zum Kompilieren und Installieren von Programmen sind drei Kommandos erforderlich, die manchmal auch als »Dreischritt« bezeichnet werden: `./configure`, `make` und `make install`. Die drei Kommandos werden im Folgenden näher beschrieben. Dabei setze ich voraus, dass Sie sich im Quellcodeverzeichnis befinden.

**configure** `configure` ist ein Script, das testet, ob alle erforderlichen Programme und Bibliotheken verfügbar sind. Da sich das Script im lokalen Verzeichnis befindet, muss es in der Form `./configure` ausgeführt werden. Das Script adaptiert die Datei `Makefile`, die alle Kommandos enthält, um die diversen Codedateien zu kompilieren und zu linken. Bei manchen (zumeist eher kleineren Programmen) kann es sein, dass es das Script `configure` nicht gibt. In diesem Fall führen Sie sofort `make` aus.

```
user$ ./configure
```

**make** `make` löst die Verarbeitung der Compile- und Link-Kommandos aus. Sie sehen nun (manchmal schier endlose) Nachrichten und Warnungen der verschiedenen Compiler-Läufe über das Konsolenfenster huschen. Solange kein Fehler auftritt, können Sie diese Meldungen getrost ignorieren. Als Ergebnis sollte sich im Quellcodeverzeichnis nun die ausführbare Datei `name` befinden.

```
user$ make
```

In vielen Fällen können Sie das Programm nun sofort starten (Kommando `./name`) und testen. Beachten Sie aber, dass insbesondere Netzwerkdienste eine spezielle Konfiguration erfordern und zumeist nur durch Init-Scripts korrekt gestartet werden!

**make install** Der letzte Schritt besteht darin, das Programm allen Benutzern zugänglich zu machen. Dazu müssen die Programm- und eventuell auch Bibliotheksdateien in öffentlich zugängliche Verzeichnisse kopiert werden. Das erfordert `root`-Rechte. Vor

der Ausführung von `make install` sollten Sie sicherstellen, dass das betreffende Programm nicht schon installiert ist. Wenn das der Fall ist, sollte es vorher deinstalliert werden.

```
root# make install
```

### Mögliche Probleme

Während des Kompilierens können vielfältige Probleme auftreten. Am wahrscheinlichsten ist, dass irgendwelche Compiler-Hilfswerkzeuge oder zum Kompilieren notwendige Entwicklerversionen von Bibliotheken fehlen. Diese Probleme werden in der Regel bereits durch `configure` festgestellt und lassen sich meist relativ leicht beheben, indem das fehlende Paket einfach installiert wird.

Schon schwieriger wird es, wenn `configure` nach Bibliotheken verlangt, die in Ihrer Distribution nicht oder nicht in der erforderlichen Version verfügbar sind: Dann müssen Sie sich im Internet auf die Suche nach der betreffenden Bibliothek machen und eventuell zuerst die Bibliothek kompilieren. Zu komplexen Programmen, wie Apache oder `mpplayer`, finden Sie im Internet richtiggehende Kompilieranleitungen, in denen Schritt für Schritt beschrieben wird, was in welcher Reihenfolge installiert und kompiliert werden muss.

Noch problematischer ist es, wenn während der Kompilierung ein Syntaxfehler auftritt, die Kompilation also mit einer Fehlermeldung abbricht. Schuld daran ist oft nicht ein Programmfehler, sondern eine Inkompatibilität zwischen Ihrem Compiler und dem Code. Manche Programme können nur mit einer bestimmten Version von `gcc` kompiliert werden, wobei es oft *gerade nicht* die neueste Version sein muss. Die Lösung besteht hier darin, die gewünschte Compiler-Version zu installieren. Auch hierzu finden Sie im Internet oder in den README-Dateien zum Quellcode oft genaue Anweisungen.

Selbst kompilierte Programme oder Bibliotheken können die Paketverwaltung durcheinanderbringen. Das Problem besteht darin, dass das selbst kompilierte Programm `abc` zwar nun auf Ihrem System installiert ist, die RPM- oder DEB-Datenbank aber nichts davon weiß. Wenn Sie nun versuchen, das Paket `xyz` zu installieren, das von `abc` abhängt, kommt es zu einer Fehlermeldung wegen scheinbar nicht erfüllter Paketabhängigkeiten. Mit `rpm` können Sie das Paket dank der Optionen `-nodeps` und `-force` dennoch installieren.

Paketverwaltung

Die eleganteste Lösung besteht darin, das Programm nicht mit `make install` zu installieren, sondern zuerst in ein Paket zu verpacken und dieses dann zu installieren. Das setzt voraus, dass Sie sich zuerst mit den Kommandos zum Erstellen von Paketen vertraut machen. Außerdem ist diese Vorgehensweise ziemlich umständlich, insbesondere wenn ein Programm mehrfach getestet und neu kompiliert werden muss.

## Beispiele

**Hello World in C** Der Platz in diesem Buch reicht nicht aus, um auch eine Einführung in die Programmierung in C und C++ zu geben. Da ich aber in der Vergangenheit immer wieder diesbezügliche Fragen erhalten habe, finden Sie hier eine kurze Anleitung, wie Sie das klassische »Hello World«-Programm in C und C++ programmieren und kompilieren. Für die C-Version schreiben Sie mit einem Editor die folgenden Zeilen in die Datei `hello.c`:

```
// hello.c
#include <stdio.h>
int main(void)
{
    printf("Hello World!\n");
}
```

Mit den folgenden Kommandos kompilieren Sie das Programm und führen es aus:

```
user$ gcc -o hello hello.c
user$ ./hello
Hello World!
```

**Hello World in C++** Der vergleichbare Code in C++ sieht so aus:

```
// hello.cpp
#include <iostream>
int main()
{
    std::cout << "Hello World!\n";
    return 0;
}
```

Zum Kompilieren verwenden Sie nun `g++` statt `gcc`:

```
user$ g++ -o hello hello.cpp
user$ ./hello
Hello World!
```

### Builder, KDevelop, VSCode, Emacs oder Vi?

Wenn Sie unter Linux eine komfortable Entwicklungsumgebung zur Programmierung in C oder C++ suchen, sollten Sie KDevelop (KDE) oder Builder (Gnome) ausprobieren. Darüber hinaus gibt es natürlich viele für einzelne Sprachen optimierte Entwicklungsumgebungen, z. B. Eclipse oder IntelliJ IDEA für Java. Und wer einen vollwertigen Editor vorzieht, kann die Programme Vi, Emacs, Atom oder VSCode ausprobieren.



## 20.3 Java

Auf Desktop-Rechnern ist Java primär für diverse LibreOffice-Zusatzfunktionen von Bedeutung. Außerdem setzt eine stetig wachsende Zahl von plattformunabhängigen Programmen Java voraus. Von deutlich größerer Bedeutung für Linux ist Java auf Servern, die in Java programmierte Websites oder Webservices anbieten (Tomcat, WildFly etc.). Interessant ist Java aber natürlich auch für viele Schüler und Studenten, die mit dieser Sprache und oft mit Eclipse als Entwicklungsumgebung programmieren lernen, Projektarbeiten durchführen etc.

Die *Java Standard Edition*, der Java-Compiler *javac*, das *Java Development Kit*, die *Java Virtual Machine* und die Java-Klassenbibliothek stehen als Open-Source-Code gemäß der GPL zur Verfügung. [Tabelle 20.1](#) fasst die wichtigsten Java-Abkürzungen zusammen.

| Abkürzung     | Bedeutung   |
|---------------|---|
| JVM           | Java Virtual Machine (führt Java-Programme aus)   |
| JRE           | Java Runtime Environment zur Ausführung von Java-Programmen; enthält die JVM sowie eine Menge Java-Bibliotheken.                      |
| JDK           | Java Software Development Kit zur Java-Programmentwicklung  |
| Java SE = JSE | Java Standard Edition für Desktop-Anwendungen   |
| Java EE = JEE | Java Enterprise Edition für Server-Anwendungen  |
| JavaFX        | Framework für grafische Benutzeroberflächen und Webapplikationen, mit gewissen Ähnlichkeiten zu Adobe Flash und Microsoft Silverlight |

**Tabelle 20.1** Java-Abkürzungen

Neben den offiziellen Java-Paketen von Oracle, die zwar kostenlos zur Verfügung stehen, aber nicht mit Linux-Distributionen ausgeliefert werden können, gibt es reine Open-Source-Implementierungen von Java: Sie basieren auf dem GPL-Code von Sun, setzen daneben aber auch Komponenten aus anderen Open-Source-Projekten ein, insbesondere IcedTea und OpenJDK.

OpenJDK/IcedTea

Die meisten aktuellen Linux-Distributionen liefern zurzeit Java 8 auf Basis des OpenJDK-Projekts aus. Diese Java-Version ist zu mehr als 99 Prozent kompatibel zum Original von Oracle. Das verbleibende Prozent ist auf Java-Komponenten zurückzuführen, die aufgrund von Lizenzproblemen nicht als Open-Source-Code veröffentlicht werden können und zu denen noch keine Open-Source-Ersatzlösungen verfügbar sind.

Um die auf Ihrem Rechner installierte Java-Version herauszufinden, führen Sie `java -version` aus. Wenn das Kommando `java` nicht zur Verfügung steht, ist Java gar

Installierte  
Java-Version  
feststellen

nicht installiert. Abhilfe schafft bei den meisten Distributionen die Installation des Pakets `java-n.n-openjdk` oder `openjdk-n-jdk`. Die folgenden Zeilen zeigen, dass Java 8 installiert ist. Das heißt, die Java-interne Versionsnummer 1.8 bezeichnet das, was umgangssprachlich Java 8 genannt wird.

```
user$ java -version
openjdk version "1.8.0_131"
OpenJDK Runtime Environment (build 1.8.0_131-b12)
OpenJDK 64-Bit Server VM (build 25.131-b12, mixed mode)
```

Das Kommando `java` ist zum Ausführen von Java-Programmen ausreichend. Wenn Sie aber selbst Java-Programme kompilieren möchten, benötigen Sie auch das Java Development Kit (JDK) mit dem Java-Compiler `javac`. Die Compiler-Version ermitteln Sie auf die gleiche Art und Weise:

```
root# javac -version
javac 1.8.0_131
```

**JDK installieren** Wenn das Kommando `javac` nicht zur Verfügung steht, müssen Sie das JDK installieren. Für Debian 8 steht leider nur die Java-Version 7 in den offiziellen Paketen zur Verfügung.

```
root# dnf install java-1.8.0-openjdk-devel      (CentOS, Fedora, RHEL)
root# apt install openjdk-7-jdk              (Debian 8)
root# apt install openjdk-8-jdk              (Debian 9)
root# zypper install java-1_8_0-openjdk-devel (SUSE)
root# apt install openjdk-8-jdk              (Ubuntu)
```

Oracle hat Java 9 bis zur Fertigstellung dieses Buchs noch nicht freigegeben. In vielen Distributionen gab es im Frühjahr 2017 aber bereits OpenJDK-9-Pakete mit Beta-Versionen von Java 9.

**WebUpd8-Pakete** Alin Andrei, Hauptautor der Webseite *webupd8.org*, betreut ein PPA (Private Package Archive) für Ubuntu. Es enthält unter anderem Pakete mit Scripts, die die Installation und Pflege der offiziellen Oracle-Java-Pakete unter Ubuntu besonders einfach machen:

<http://webupd8.org/p/ubuntu-ppas-by-webupd8.html>

**Eclipse, IDEA, NetBeans** Alle drei »großen« Java-IDEs können Sie von den jeweiligen Projektwebseiten kostenlos in Linux-Versionen herunterladen. Manche Distributoren stellen diese IDEs auch in ihren Paketquellen zur Verfügung. Diese Pakete sind aber häufig veraltet und daher nicht zu empfehlen.

# Kapitel 21

## Grafiksystem

Das Grafiksystem von Linux erlebt gerade einen Umbruch: Nachdem mehrere Jahrzehnte lang das *X Window System* (nicht *Windows*!) als Fundament diente, hat 2017 der Umstieg auf das Nachfolgesystem Wayland begonnen. Als erste Distribution setzt Fedora seit Version 25 Wayland standardmäßig ein. Das X Window System bleibt aber parallel installiert und wird automatisch als Fallback verwendet, sollte es Treiberprobleme geben. Bei einigen anderen Distributionen war die Situation zu Beginn des Sommers 2017 gerade umgekehrt: Standardmäßig kommt X zum Einsatz, die Benutzer können aber während des Logins optional Wayland aktivieren und ausprobieren.

Für den Anwender ist der Wechsel nicht sichtbar: Alles sieht aus wie vorher, und im Idealfall sollte auch alles wie vorher funktionieren. Längerfristig verspricht der Umstieg auf Wayland Performance-Vorteile und eine einfachere Programmierung. Davon ist aber noch nicht viel zu bemerken, vor allem deswegen, weil aktuell das Hauptaugenmerk darauf liegt, bei größtmöglicher Kompatibilität den Umstieg auf Wayland überhaupt zu vollziehen. Erst wenn die Entwickler in ein, zwei Jahren davon ausgehen können, dass auf der Mehrheit der Linux-Systeme Wayland aktiv ist, können in einem nächsten Schritt Desktop-Systeme wie Gnome oder KDE sowie Desktop-Applikationen wie Firefox, GIMP oder LibreOffice explizit für Wayland optimiert werden.

Canonical wollte für Ubuntu und insbesondere dessen Smartphone-Variante ein eigenes Grafiksystem namens Mir etablieren. Im Frühjahr 2017 hat Canonical seine Smartphone-Ambitionen samt Mir begraben und setzt für die Desktop-Varianten von Ubuntu nun ebenfalls auf Wayland.

Obwohl klar ist, dass die Bedeutung von X schwinden wird, ist dieses Kapitel noch immer X-lastig. Das liegt daran, dass das X Window System gegenwärtig auf den meisten Distributionen als Defaultsystem dient. Selbst auf modernen Distributionen wie Fedora müssen Sie noch in vielen Fällen auf X zurückgreifen, wenn es Treiberprobleme gibt oder wenn Sie besondere Konfigurationsvarianten realisieren wollen.

## 21.1 Grundlagen

Für Umsteiger, die bisher mit Windows oder macOS gearbeitet haben, ist die Differenzierung zwischen Grafiksystem und Desktop-Anwendungen oft schwer zu verstehen. Das liegt daran, dass unter Windows bzw. unter macOS nicht mehrere Desktop-Systeme zur Auswahl stehen. Ein Windows-PC oder Mac läuft immer im Grafikmodus. Welche Treiber oder Bibliotheken im Hintergrund dafür verantwortlich sind, dass ein Fenster auf dem Bildschirm erscheint, ist aus Anwendersicht egal.

Solange alles funktioniert, gilt dies natürlich auch für Linux. Wenn Sie aber verstehen möchten, wie das Grafiksystem hinter den Kulissen funktioniert, muss Ihnen die Differenzierung zwischen dem Low-Level-Grafiksystem auf der einen Seite (X, Wayland) und den darauf aufbauenden High-Level-Anwendungen auf der anderen Seite (Gnome, KDE, alle Programme, die im Grafikmodus laufen) bewusst werden. X bzw. Wayland kümmern sich nur um ganz grundlegende Aufgaben wie um die Kommunikation mit dem Kernel oder anderen Treibern oder um das Entgegennehmen von Maus-, Trackpad- oder Tastatureingaben. Das tatsächliche Erscheinungsbild und Verhalten eines Linux-Desktops wird dadurch nicht geprägt – darum kümmern sich erst die auf X oder Wayland aufbauenden Bibliotheken und Programme.

### X Window System

Das X Window System (kurz X) wurde ursprünglich vom Massachusetts Institute of Technology entwickelt. X bezeichnet Basisfunktionen zum Zeichnen von Punkten, Rechtecken etc., aber auch ein Netzwerkprotokoll, das es ermöglicht, ein X-Programm auf Rechner A auszuführen und die Ergebnisse auf Rechner B darzustellen.

**X-Server** Der X-Server stellt die Schnittstelle zwischen dem X Window System und der Hardware her. Der Server ist modularisiert: Das bedeutet, dass der eigentliche Server durch ein Modul mit den spezifischen Funktionen für die jeweilige Grafikkarte ergänzt wird.

**X-Erweiterungs-module** Die Standardfunktionen des X-Servers können durch diverse Zusatzmodule (Extensions) erweitert werden, die beispielsweise für 3D-Grafik, für die Video-Ausgabe etc. verantwortlich sind.

**Window Manager** Der Window Manager ist ein X-Programm, das für die Verwaltung der Fenster zuständig ist. Sie können mit dem Window Manager andere Programme starten, zwischen Fenstern wechseln, Fenster verschieben und schließen etc. – also eigentlich recht triviale Aufgaben ausführen.

Der Window Manager ist auch für die Dekoration der Fenster zuständig, also für die Gestaltung des Rahmens rund um den eigentlichen Fensterinhalt samt der Titelleiste mit Buttons zum Schließen oder Maximieren des Fensters. Es ist wichtig sich vor

Augen zu halten, dass diese Aufgaben vom Window Manager und nicht von X selbst erledigt werden. KDE und Gnome haben jeweils ihren eigenen Window Manager.

## Wayland

Wayland ist kein Programm, sondern »nur« ein Protokoll für die Kommunikation zwischen dem Wayland Compositor (einem Display-Server) und den Anwendungsprogrammen (Clients). Sowohl die vom Gnome-Projekt verwendete GTK-Bibliothek als auch die Bibliothek Qt 5, die von KDE verwendet wird, ist Wayland-kompatibel. Für Programme, die noch nicht selbst Wayland-Bibliotheken nutzen, gibt es die Zwischenschicht XWayland. Sie stellt die Kompatibilität zum X Window System her.

Wayland greift teilweise auf dieselben im Kernel verankerten Mechanismen bzw. Treiber zurück wie X. Dazu zählen die Verarbeitung von Eingaben (*libinput*), die Einstellung der Auflösung (*Kernel Mode Setting*, KMS) und der Zugriff auf den Video-Speicher (*Direct Rendering Manager*, DRM). Insofern erfindet Wayland das Rad nicht neu, sondern greift auf viele Bausteine zurück, die auch unter X eingesetzt werden. Grundsätzlich anders als unter X erfolgt die Kommunikation mit den 3D-Funktionen der Mesa-Bibliothek über die Schnittstelle EGL.

Der Bruch mit X und der Verzicht auf eine vollständige Kompatibilität zu unzähligen alten Spezifikationen hat den Vorteil, dass eine Menge für die Praxis nicht mehr relevanter Code aus dem X Window System nicht weiter gewartet werden muss. Insofern verspricht Wayland ein wesentlich schlankeres System zu werden.

Während unter X ein eigener Window Manager für die Dekoration der Fenster zuständig ist, übernimmt diese Aufgabe der Wayland Compositor – natürlich im Zusammenspiel mit Wayland-kompatiblen Versionen von Gnome, KDE etc. Der Wayland Compositor kümmert sich auch um Aufgaben, die unter X der Xorg-Server übernimmt – z. B. um die Verarbeitung von Eingaben. Insofern gibt es eine Menge Details, die nicht davon abhängig sind, wie Wayland funktioniert, sondern wie der Wayland Compositor für das jeweilige Desktop-System implementiert ist.

Wayland  
Compositor

Wenn Gnome unter Wayland läuft, dann fungiert *Mutter* als Wayland Compositor. Wenn Gnome hingegen unter X ausgeführt wird, dann dient *Mutter* als Window Manager. Insofern ist *Mutter* eine Brücke, die je nach Grafiksystem zwischen Gnome und X bzw. zwischen Gnome und Wayland errichtet wird. *Mutter* wird nicht als eigenständiges Programm ausgeführt, sondern in Form von Bibliotheken, mit denen die Gnome Shell verlinkt ist.

Mutter

### Warum immer Gnome?

Im Wayland-Kontext ist ständig von Gnome die Rede, als gäbe es kein anderes Desktop-System. Das liegt daran, dass Gnome im Frühsommer 2017 tatsächlich das einzige Desktop-System war, dessen Wayland Compositor weitgehend stabil funktionierte. Bei KDE sind die Arbeiten auch schon relativ weit fortgeschritten. Die Neon-Distribution experimentiert gegenwärtig mit Wayland, will Wayland aber frühestens 2018 standardmäßig aktivieren.

Bei einer Internet-Recherche zu Wayland werden Sie auch auf den Begriff *Weston* stoßen: Weston war der erste funktionierende Wayland Compositor. Das Programm war aber mehr eine Designstudie. Seit der Wayland Compositor von Gnome funktioniert, hat Weston keine praktische Relevanz mehr.

### Wayland-Einschränkungen im Vergleich zu X

Aus dem Wayland-Design ergeben sich im Vergleich zu X einige Einschränkungen:

- ▶ Wayland ist aktuell nicht netzwerktauglich. Sie können also nicht wie unter X ein Grafikprogramm auf dem Host 1 ausführen, dieses aber z. B. via SSH auf dem Host 2 anzeigen und bedienen. Es ist unklar, ob Wayland diese Funktionalität später erhalten wird. Einen Implementierungsversuch hat es schon gegeben, aber er ist gescheitert.
- ▶ Remote-Desktop-Programme wie VNC- oder RDP-Server sind nicht Wayland-kompatibel. Sofern auf dem Rechner alle erforderlichen X-Pakete installiert sind, ist es aber möglich, parallel zu Wayland einen VNC- oder RDP-Server einzurichten. Lokales Arbeiten am Computer erfolgt dann via Wayland, der Netzwerkzugriff hingegen via X. Das erfordert zwangsläufig zwei getrennte Desktop-Sessions. Screen-Sharing ist aktuell mit Wayland nicht möglich. (Das gilt auch für das kommerzielle Programm *TeamViewer*.)
- ▶ Die meisten Tools zur Aufnahme von Screenshots und Screencasts funktionieren nicht. Das ist eigentlich ein Sicherheits-Feature (*It's not a bug, it's a feature!*): Es soll verhindern, dass ein Programm Informationen eines anderen Programms auslesen kann. Ausgenommen von dieser Einschränkung sind in das Desktop-System integrierte Werkzeuge, also z. B. die Screenshot-Tools von Gnome.
- ▶ Unter Wayland können Sie nicht einfach ein grafisches Programm mit root-Rechten ausführen (z. B. aus einem Terminal heraus, in dem Sie vorher `sudo` ausgeführt haben). Diese Einschränkung ist gerade für Power-User schwerwiegend und betrifft z. B. das grafische Partitionswerkzeug `gparted`, diverse Installations- und Setup-Programme, Backup-Tools etc. Selbst ein simpler Start eines Editors oder Dateimanagers mit root-Rechten ist unmöglich.

Für das Problem gibt es zwei Lösungen: Die eine besteht darin, eben mit Textkommandos zu arbeiten – vi, emacs, parted etc. Die andere Variante setzt voraus, dass grafische Programme vollkommen neu konzipiert werden müssen: Die Benutzeroberfläche läuft dabei mit gewöhnlichen Benutzerrechten. Nur für die Arbeiten, die wirklich Administrationsrechte erfordern, fordert das Programm über PolicyKit (siehe [Abschnitt 12.4](#), »Prozesse unter einer anderen Identität ausführen (PolicyKit)«) die notwendigen Rechte an. So funktionieren z. B. die Gnome-Programme *Einstellungen* und *Software*.

Kurzfristig ist nicht zu erwarten, dass alle für Power-User relevanten Programme auf PolicyKit umgestellt werden. Die pragmatische Lösung lautet dann eben, X und nicht Wayland als Grafiksystem zu verwenden – solange es X eben noch gibt.

- ▶ Diverse X-spezifische Werkzeuge wie xkill, xrandr oder xmodmap können nicht mehr verwendet werden.
- ▶ Spiele können nicht die Auflösung des Grafiksystems ändern.

## Glossar

In Texten über das Linux-Grafiksystem wimmelt es nur so von Abkürzungen und obskuren Begriffen. Dieser Abschnitt gibt in alphabetischer Reihenfolge eine Orientierungshilfe.

Compiz ist ein Composition- und Fenster-Manager, der 3D-Funktionen unterstützt. Zuletzt wurde Compiz nur noch von Ubuntu für dessen eigenes Desktop-System Unity verwendet. Mit dem Umstieg auf Gnome in Version 17.10 wird es vermutlich keine Linux-Distribution mehr geben, die Compiz standardmäßig nutzt. Compiz

Das *Direct Rendering Interface* (DRI) ermöglicht X die Nutzung der 3D-Funktionen der Grafikkarte – sofern es einen passenden DRI-Treiber für die Karte gibt. DRI baut auf dem im Kernel enthaltenen *Direct Rendering Manager* (DRM) auf. Wayland verwendet auch DRM, allerdings über die EGL-Schnittstelle. DRI und DRM

EGL ist eine Schnittstelle zwischen OpenGL und der Fensterverwaltung von Wayland. EGL ermöglicht es Wayland-Programmen, direkt 3D-Funktionen (OpenGL-Funktionen) zur Darstellung des Fensterinhalts aufzurufen. X-Programme verwenden GLX anstelle von EGL. EGL

Unter X werden die Open-GL-Funktionen über die GLX-Bibliothek genutzt. Diese Bibliothek stellt die Verbindung zwischen dem X Window System und Open GL her. GLX stellt beispielsweise sicher, dass Open-GL-Ausgaben nur im gerade sichtbaren Teil eines Fensters erfolgen und nicht mit anderen Fenstern kollidieren. GLX ist durch ein Modul in X integriert. GLX und libGL

- KMS** *Kernel Mode Setting* (KMS) bedeutet, dass der Linux-Kernel und nicht X den Grafikmodus einstellt. KMS wird von allen wichtigen Open-Source-Treibern unterstützt. KMS ermöglicht es, die gewünschte Grafikauflösung bereits unmittelbar nach dem Rechnerstart einzustellen. Falls der Kernel beim Booten nicht die richtige Auflösung wählt, kann sie mit der Kerneloption `video` eingestellt werden (siehe [Abschnitt 25.8](#), »Kernel-Boot-Optionen«).
- Mesa und Open GL** *Open GL* (oft auch kurz GL genannt) ist eine ursprünglich von SGI entwickelte Bibliothek zur Darstellung von 3D-Grafiken, die auf fast allen Unix/Linux-Rechnern zur Verfügung steht. Nahezu alle unter Linux verfügbaren 3D-Programme und -Spiele bauen auf Open GL auf. Open GL ist also gewissermaßen das Unix/Linux-Gegenstück zu Microsofts DirectX-Bibliothek.
- Da der Code von Open GL ursprünglich nicht frei verfügbar war, ist die dazu kompatible Open-Source-Bibliothek Mesa entstanden. Mesa war anfänglich eine reine Software-Lösung, nutzt aber mittlerweile 3D-Funktionen der Grafikkarte.
- OpenCL** Die *Open Computing Language* (OpenCL) ist eine Schnittstelle, um parallelisierbare Algorithmen zu entwickeln und besonders effizient durch die GPU einer Grafikkarte auszuführen.
- RandR** Die *Resize and Rotate Extension* (RandR) erlaubt es, einige Einstellungen des Grafiksystems im laufenden Betrieb zu ändern. Dazu zählen die Auflösung, die Bildfrequenz und die Bildrotation. Via RandR kann auch ein zweiter Bildschirm aktiviert werden.
- VDPAU** Die *Video Decode and Presentation API for Unix* (VDPAU) ist eine API zur Dekodierung von Videostreamen. VDPAU wurde zwar ursprünglich von NVIDIA entwickelt, wird aber auch von AMD unterstützt.
- Vulkan** Vulkan ist eine erst seit 2016 verfügbare Alternative zu OpenGL. Die 3D-API ist besonders auf hohe Geschwindigkeit optimiert. Das Design ähnelt dem von Direct3D 12 (Microsoft). Aktuelle Linux-Grafiktreiber und Wayland unterstützen Vulkan bereits.
- XRender** Die *X Rendering Extension* (kurz XRender) ist eine Bibliothek zur Erzielung von Transparenz- und Überlagerungseffekten (*Alpha Blending*). Die Bibliothek wird auch zur Textausgabe verwendet. XRender greift aus Geschwindigkeitsgründen auf 3D-Hardware-Funktionen zurück.

## 21.2 Grafiktreiber

Bevor ich in den folgenden Abschnitten die Konfiguration und den Betrieb von X und Wayland beschreibe, möchte ich an dieser Stelle auf das größte Problem beider Grafiksysteme eingehen: die mangelnde Unterstützung moderner Grafikkarten durch Open-Source-Treiber.



Die überwiegende Mehrheit aller aktuellen PCs und Notebooks enthält Grafik-Chips (*Graphical Processing Units*, kurz GPUs) der folgenden drei Firmen: AMD, Intel und NVIDIA, wobei es die Intel-Grafikchips nur in Form von kompletten Chipsätzen gibt, also nicht als eigenständige Grafikkarten.

Zuerst die gute Nachricht: Die Grafiktreiber, die als Teil des Linux-Kernels in Form von Open-Source-Code realisiert sind, funktionieren grundsätzlich in Kombination mit den meisten gängigen Grafikkarten, unabhängig davon, ob Sie nun mit X oder Wayland arbeiten. Sie müssen also keine Angst haben, dass nach dem Start von Linux der Bildschirm einfach schwarz bleibt.

Und nun die schlechte Nachricht: Die mit den Open-Source-Treibern erzielte Geschwindigkeit ist nicht immer optimal, teilweise bleiben 3D-, Zusatz- oder Energiesparfunktionen ungenutzt. Der Grafiktreiber kann also ein Grund sein, warum ein Notebook unter Windows länger läuft als unter Linux.

Bei einigen Grafikkarten können alternativ bzw. ergänzend zu den Open-Source-Treibern von den Herstellern zur Verfügung gestellten Binärtreiber Abhilfe schaffen. Diese Treiber basieren allerdings nicht auf Open-Source-Code und sind deswegen mit diversen Nachteilen verbunden.

Genau genommen gibt es nicht *einen* Treiber, sondern mehrere. Diese bilden zusammen den sogenannten Treiber-Stack, also aufeinander aufbauende Treiber für verschiedene Komponenten des Grafiksystems:

Treiber-Stack

- ▶ Hardware-Treiber auf Kernel-Ebene
- ▶ Direct Rendering Manager (`libdrm`)
- ▶ 3D-Funktionen/OpenGL (Mesa)
- ▶ X-Treiber (`xf86-video-xxx`)

Die vierte Ebene ist nur relevant, wenn X im Spiel ist. Bei Wayland entfällt diese Ebene; davon abgesehen verwendet Wayland aber die gleichen Treiber wie X. Beachten Sie, dass die Namen in den verschiedenen Ebenen nicht konsistent sind. Der Intel-Treiber setzt sich etwa aus dem Kernelmodul `i915`, der Bibliothek `libdrm_intel`, dem Mesa-Treiber `i965_dri` oder `i915_dri` (je nach Chip) sowie dem X-Treiber `intel` zusammen.

Hilfreich zum tieferen Verständnis der Treiberebenen sind der folgende Blog-Beitrag sowie ein achtseitiger Artikel von Thorsten Leemhuis für die Zeitschrift c't. Beide Texte sind allerdings schon mehrere Jahre alt und berücksichtigen Wayland nicht:

<https://blogs.igalia.com/itoral/2014/07/29/a-brief-introduction-to-the-linux-graphics-stack>

<http://heise.de/-2415770> (kostenpflichtig)

Benchmarktests sowie eine Menge Hintergrundinformationen zu den gerade aktuellen Grafiktreibern finden Sie auf der ausgezeichneten Website <https://phoronix.com>.

### Treiber für AMD, Intel und NVIDIA

Die folgenden Absätze fassen die aktuelle Treiber-Situation zusammen, alphabetisch geordnet nach Grafikkartenhersteller.

- AMD** Traditionell gab es für Grafikkarten von AMD (ehemals ATI) über viele Jahre zwei Treiber: den Open-Source-Treiber `radeon` und den Binärtreiber von AMD `fglrx`. Diese Treiber haben aktuell noch eine große Verbreitung: einerseits, weil sie bei älteren Linux-Distributionen standardmäßig zum Einsatz kommen bzw. mit wenig Aufwand nachinstalliert werden können, und andererseits, weil sie diverse ältere GPU-Modelle unterstützen, die in manchem alten PC oder Notebook stecken.

Beide Treiber werden aktuell durch den neuen Open-Source-Treiber `amdgpu` ersetzt, der seit 2015 von AMD in Kooperation mit der Open-Source-Community entwickelt wird. Der Treiber unterstützt nur AMD-GPUs mit der Mikroarchitektur *Graphics Core Next* (GCN), die seit 2011 verwendet wird.

Aus verschiedenen Gründen sieht sich AMD nicht in der Lage, alle Funktionen seiner GPUs durch Open-Source-Treiber zu unterstützen. Zu diesen Funktionen zählen OpenCL, Vulkan und VDPAU. Wenn Sie diese Funktionen unter Linux nutzen wollen, benötigen Sie den `amdgpu-pro`-Treiber. Diese Erweiterung zum `amdgpu`-Treiber steht auf der AMD-Webseite kostenlos zum Download zur Verfügung. Der Treiber ist aber proprietär, der Quellcode steht nicht zur Verfügung:

<http://support.amd.com/en-us/kb-articles/Pages/AMDGPU-PRO-Driver-for-Linux-Release-Notes.aspx>

- Intel** Grundsätzlich arbeitet auch Intel gut mit der Open-Source-Gemeinde zusammen. Der Intel-Treiber mit dem Namen `i915` befindet sich im offiziellen Kernel-Code, proprietäre Treiber gibt es nicht. Insofern bietet der Kauf eines Notebooks oder PCs mit einer Intel-CPU samt integrierter Grafikeinheit beinahe eine Garantie für den unkomplizierten Einsatz unter Linux. Für den gewöhnlichen Desktop-Einsatz reicht die Leistung vollkommen aus.

Freilich ist die Treibersituation auch bei Intel nicht makellos. Insbesondere die schlechte Wartung der X-spezifischen Komponenten des Intel-Treiber-Stacks verärgert Entwickler und Distributoren schon seit Jahren. Debian, Fedora und Ubuntu verwenden seit 2016 bzw. 2017 anstelle des Intel-spezifischen Treibers `xf86-video-intel` den generischen Treiber `xf86-video-modesetting`.

Normalerweise sind Sie gut beraten, den von Ihrer Distribution ausgelieferten Treiber zu verwenden. Nur für den Fall, dass Probleme auftreten oder Sie als Entwickler neue Features ausprobieren möchten, bietet Intel ein Update-Tool an, mit dem Sie den Intel-Treiber-Stack bei einigen ausgewählten Distributionen durch neue Versionen ersetzen können:

<https://01.org/linuxgraphics/downloads>

NVIDIA beharrt bis heute auf seinem Standpunkt, dass Lizenzvereinbarungen mit anderen Unternehmen und Patente die Entwicklung eines Open-Source-Treibers unmöglich machen und eine öffentliche Dokumentation der internen Schnittstellen verhindern würden. Stattdessen stellt NVIDIA den kostenlosen Binärtreiber `nvidia` zur Verfügung. Dessen Qualität war in der Vergangenheit zwar gut, die prinzipiellen Nachteile eines Nicht-Open-Source-Treibers bleiben aber bestehen (siehe den folgenden Abschnitt). Der `nvidia`-Treiber ist zudem nur mit Einschränkungen Wayland-kompatibel.

**NVIDIA**

Trotz des Widerstands von NVIDIA hat die Open-Source-Gemeinde mit `nouveau` einen eigenen Treiber entwickelt, der mittlerweile bei allen Distributionen standardmäßig zum Einsatz kommt und sowohl für X als auch für Wayland recht gut funktioniert. Probleme machen aber neue Grafikkarten sowie die Nutzung der Energiesparfunktionen.

Auf manchen Notebooks und vereinzelt auch auf Desktop-Rechnern befinden sich *zwei* Grafiksysteme: ein energiesparendes System, das in der Regel direkt in die CPU integriert ist, und ein zweites System für hohe 3D-Leistung. Dieser hybride Ansatz versucht, eine hohe Laufzeit mit hoher Grafikleistung zu vereinen – je nachdem, was der Benutzer gerade braucht. Mit den geeigneten Treibern unter Windows oder macOS kann das aktive Grafiksystem im laufenden Betrieb gewechselt werden, ohne dass der Benutzer dies bemerkt.

**Hybrid-Lösungen**

Wenn Sie Pech haben, spricht Linux nur die langsame Chipsatz-Grafik an, ist aber nicht in der Lage, das zweite Grafiksystem in einen Energiesparmodus zu versetzen. Mit anderen Worten: Das Grafiksystem ist langsam *und* verbraucht unnötig viel Strom.

Immerhin lassen sich manche Hybrid-Grafiksysteme mit etwas Mühe so konfigurieren, dass sie unter Linux einigermaßen zufriedenstellend laufen (aktuell allerdings nur mit X, nicht mit Wayland). Besonders einfach gelingt die Konfiguration mit Ubuntu oder Mint. Eine exzellente Zusammenfassung über die Nutzung der NVIDIA-Hybrid-Lösung *Optimus* gibt Thorsten Leemhuis im folgenden, leider kostenpflichtigen und nicht mehr ganz aktuellen c't-Artikel:

<http://heise.de/-2638735> (Artikel vom Mai 2015)

Viele Detailinformationen für technisch versierte Benutzer enthält das Wiki von Arch Linux:

[https://wiki.archlinux.org/index.php/NVIDIA\\_Optimus](https://wiki.archlinux.org/index.php/NVIDIA_Optimus)

<https://wiki.archlinux.org/index.php/PRIME>

Selbst habe ich ein Hybrid-Grafiksystem nie ausprobiert, weil ich mich beim Kauf von Rechnern selbst an die folgenden Empfehlungen halte. Von einem Leser der vorigen Auflage dieses Buchs habe ich das Feedback erhalten, dass meine Einstellung zur Hybrid-Grafik zu negativ sei. Die Erfahrungen des Lesers mit einem NVIDIA-Optimus-System in Kombination mit Ubuntu bzw. Mint waren hervorragend.

**Kaufempfehlung** Seit Intel die Grafikfunktionen in viele CPU-Modelle integriert hat, ist eine Empfehlung recht einfach: Egal ob Notebook oder Desktop-PC, entscheiden Sie sich für einen Rechner mit integrierter Intel-Grafik und ohne zusätzliche Grafikkarte! Die Grafikleistung ist für typische Linux-Anwendungen mehr als ausreichend, der Stromverbrauch ist minimal, und Sie gehen allen Treiberproblemen aus dem Weg. Egal, welche Linux-Distribution Sie installieren – das Grafiksystem läuft ohne umständliche Treiberinstallationen auf Anhieb problemlos!

Grundsätzlich gilt diese Empfehlung auch für AMD-Prozessoren mit integrierter Grafikeinheit. Diese haben aktuell den Vorteil, dass die Grafikleistung meist höher ist als bei preislich vergleichbaren Intel-CPU. Dem steht der Nachteil gegenüber, dass die Unterstützung durch die Open-Source-Treiber `radeon` bzw. `amdgpu` theoretisch zwar ausgezeichnet ist, in der Praxis aber bei neuen AMD-CPU deutlich öfter hakt als bei Intel-CPU.

Wenn Sie wirklich eine dezidierte Grafikkarte brauchen, ist ein nicht ganz neues Modell mit einem AMD-Grafikchip empfehlenswert. Für aktuelle NVIDIA-Grafikkarten sollten Sie sich nur entscheiden, wenn Sie vorher gründlich recherchiert haben, dass der proprietäre NVIDIA-Treiber mit Ihrer Distribution und mit Ihren anderen Anforderungen kompatibel ist.

Um Rechner mit Hybrid-Grafik sollten Sie einen weiten Bogen machen. Computer mit Hybrid-Grafik erkennen Sie oft an Marketingbezeichnungen wie *NVIDIA Optimus*, *AMD Hybrid Graphics* oder *Virtu GPU Virtualization*.

### Probleme nichtfreier Treiber

Vielleicht stehen Sie auf dem Standpunkt, die Unterscheidung zwischen »echten« Open-Source-Treibern und kostenlosen Herstellertreibern (auch proprietäre Treiber, Binärtreiber oder im Englischen *Restricted Driver* genannt) sei Haarspalterei – Hauptsache, es funktioniert. Es gibt aber gute Gründe, die für Open-Source-Treiber und gegen Binärtreiber sprechen:

- ▶ Die Stabilität der Herstellertreiber war in der Vergangenheit nicht immer besonders hoch.
- ▶ Die Grafiktreiber müssen zur X-Version passen. Gerade Fedora-Anwender, deren Distribution oft die allerneuste, erst halb fertige X-Version enthält, wissen davon ein Lied zu singen: In der Vergangenheit dauerte es oft monatelang, bis es kompatible Herstellertreiber gab.
- ▶ Grafiktreiber erfordern eine enge Verzahnung mit dem Linux-Kernel. Dazu befindet sich zwischen dem eigentlichen Treiber (Closed-Source) und dem Kernel (GPL) ein kleines Kernelmodul, das nur als Schnittstelle dient. Viele Linux-Entwickler haben Zweifel daran, dass diese Vorgehensweise GPL-konform ist, und dulden sie nur widerwillig. Die Kernerentwickler bezeichnen den Kernel als *tainted* (makelbehaftet), sobald ein Nicht-GPL-Treiber geladen wird, und verweigern in diesem Fall jegliche Unterstützung bei Problemen.

Die Verzahnung mit dem Kernel hat einen weiteren Nachteil: Nach jedem Kernel-Update muss auch das Kernelmodul des Grafiktreibers aktualisiert werden. Wie kompliziert dieser Vorgang ist, hängt von der Distribution ab. Im Idealfall wird der neue Grafiktreiber vom Paketverwaltungssystem automatisch heruntergeladen und installiert; im ungünstigsten Fall funktioniert nach dem Kernel-Update das Grafiksystem nicht mehr und Sie müssen in einer Textkonsole ein neues Kernelmodul für den Treiber kompilieren.

- ▶ Wegen der oben erwähnten GPL-Konflikte ist die Weitergabe der Binärtreiber schwierig. Sie müssen die Treiber daher nach der Installation der Distribution extra herunterladen und installieren.
- ▶ Wenn im Treiber ein Sicherheitsproblem auftritt, können Linux-Distributoren nur darauf hoffen, dass die Grafikfirmen möglichst rasch ein Update zustande bringen. Bei Open-Source-Code kann die Entwicklergemeinde den Fehler dagegen selbst beheben, was in der Regel schneller geht.
- ▶ Mangels Code ist es unmöglich, die Treiber für andere Betriebssysteme, CPU-Architekturen etc. zu portieren. Die Grafikfirmen entscheiden, welche Systeme unterstützt werden. Beispielsweise gab es lange Zeit keine Treiber für 64-Bit- oder BSD-Systeme.
- ▶ Die Grafikunterstützung unter Linux ist von der Gunst des Herstellers abhängig. Ältere Grafikkarten werden oft nicht besonders lange unterstützt.
- ▶ Wenn Sie UEFI Secure Boot nutzen und Ihre Distribution nur signierte Kernelmodule zulässt, können Sie keine proprietären Grafiktreiber verwenden.

Auf Dauer kann Linux nur dann ein Open-Source-System bleiben, wenn auch die wichtigsten Komponenten frei verfügbar sind. Und dazu zählen zweifelsohne die Grafiktreiber. Suchen Sie Ihren nächsten Rechner bzw. Ihre nächste Grafikkarte auch unter dem Gesichtspunkt aus, ob es dafür freie Treiber gibt!

## 21.3 NVIDIA-Treiberinstallation

Auch wenn NVIDIA mit der Linux-Community schlecht zusammenarbeitet, erzeugt die Firma doch herausragend gute Grafikkarten, die auch unter Linux beliebt sind. In einfachen Fällen reicht der Open-Source-Treiber `nouveau` aus, aber die optimale Nutzung der Grafikkarte erfordert oft den proprietären NVIDIA-Treiber. Je nach Distribution ist die Installation mehr oder weniger umständlich.

**Ubuntu** Besonders komfortabel gelingt die Treiberinstallation unter Ubuntu: Dort starten Sie das Programm ANWENDUNGEN & AKTUALISIERUNGEN. Die für Ihr System passenden Treiber werden im Dialogblatt ZUSÄTZLICHE TREIBER aufgelistet (siehe [Abbildung 21.1](#)). Zur Aktivierung müssen Sie den Rechner neu starten – fertig!

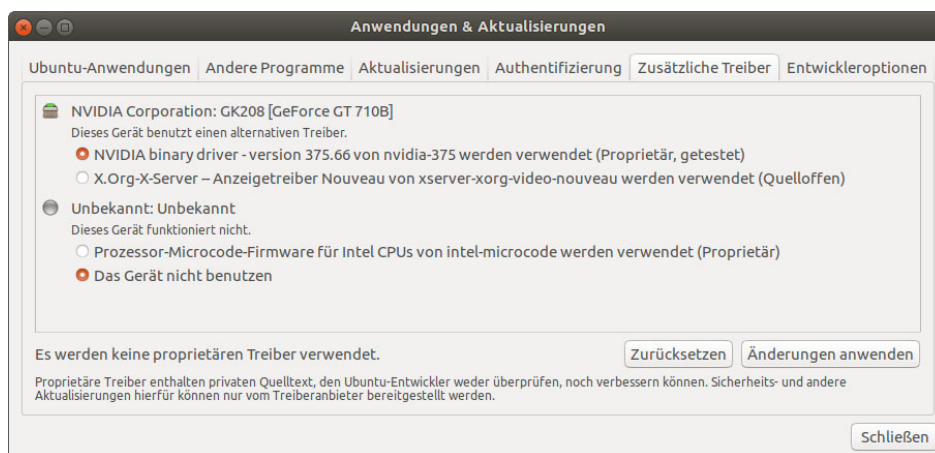


Abbildung 21.1 Installation des NVIDIA-Treibers unter Ubuntu

CentOS, Fedora,  
RHEL

Die aktuell am besten gewarteten NVIDIA-Pakete für Red-Hat-ähnliche Distributionen befinden sich in der relativ neuen Paketquelle *negativo.org*. Eine ausführliche Beschreibung der Pakete und ihrer Verwendung finden Sie hier:

<https://negativo17.org/nvidia-driver>

Ich habe die Installation unter Fedora 26 getestet:

```
root# dnf config-manager --add-repo=\
      https://negativo17.org/repos/fedora-nvidia.repo
root# dnf makecache
root# dnf install nvidia-driver nvidia-settings
root# reboot
```

Der erste Neustart dauert relativ lange, weil dabei die NVIDIA-Kernelmodule kompiliert werden. Drücken Sie `[Esc]`, wenn Sie die Init-Meldungen lesen möchten. Der NVIDIA-Treiber wird ohne weitere Konfigurationsarbeiten aktiv.

Wenn Sie die NVIDIA-Treiber manuell installieren möchten, finden Sie hier ausführliche Anleitungen:

Manuelle  
Installation

<https://wiki.archlinux.org/index.php/NVIDIA>

<https://www.if-not-true-then-false.com/2015/fedora-nvidia-guide>

[https://wiki.ubuntuusers.de/Grafikkarten/Nvidia/Manuelle\\_Treiberinstallation](https://wiki.ubuntuusers.de/Grafikkarten/Nvidia/Manuelle_Treiberinstallation)

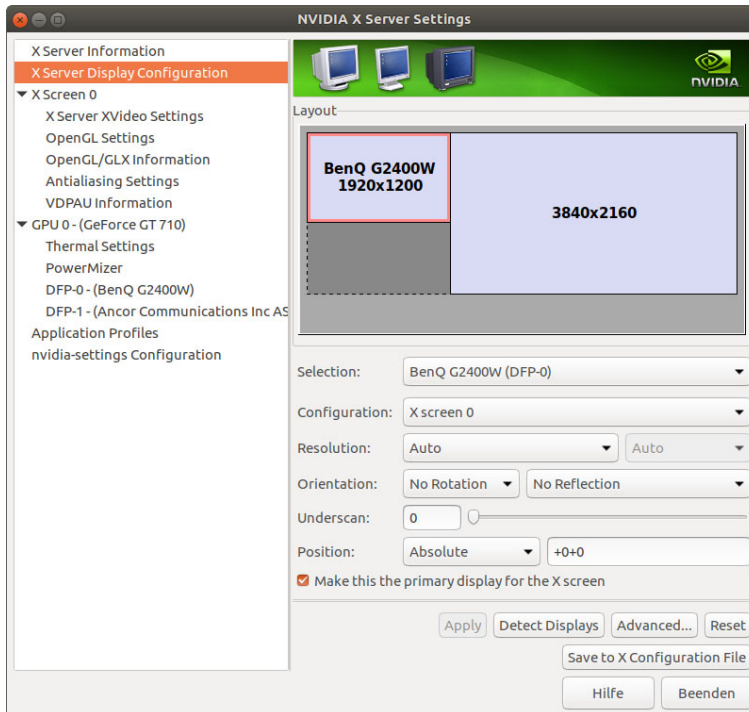


Abbildung 21.2 NVIDIA-Treiberkonfiguration

Wenn der NVIDIA-Treiber richtig installiert ist, wird er beim Start des Grafiksystems automatisch geladen. Sie können sich mit `lspci -k` sowie durch einen Blick in die Datei `/var/log/Xorg.0.log` davon überzeugen.

Anwendung

Änderungen an Konfigurationsdateien sind in der Regel nicht erforderlich. Sollte Linux allerdings weiterhin den `nouveau`-Treiber laden, fügen Sie in die Datei `/etc/modprobe.d/blacklist.conf` die folgenden Zeilen ein, starten den Rechner dann neu und versuchen es noch einmal:

```
# /etc/modprobe.d/blacklist.conf
blacklist nouveau
```

Nur in Ausnahmefällen müssen Sie `xorg.conf` dahingehend verändern, dass zwingend der `nvidia`-Treiber zu verwenden ist:

```
# Datei /etc/X11/xorg.conf
Section "Device"
    Identifier "Device0"
    Driver      "nvidia"
EndSection
```

Sobald der NVIDIA-Treiber grundsätzlich läuft, setzen Sie zur weiteren Konfiguration das Programm `nvidia-settings` ein (siehe [Abbildung 21.2](#)). Unter Ubuntu müssen Sie das Programm vorher explizit mit `apt` installieren. `nvidia-settings` ermöglicht die unmittelbare Veränderung zahlreicher Optionen, also ohne X-Neustart. Wenn das Ergebnis zufriedenstellend ist, können Sie die Änderungen in `xorg.conf` speichern. Dabei müssen Sie dann das `root`- oder Ihr eigenes Passwort für `sudo` angeben.

**Wayland** Aktuell sind Wayland und die NVIDIA-Treiber inkompatibel zueinander. Im Oktober 2016 gab es zwar Berichte darüber, dass an der Lösung des Problems gearbeitet würde, bis zum Frühsommer 2017 war dann aber nichts mehr zu hören:

*<https://www.golem.de/news/fuer-wayland-unterstuetzung-linux-team-von-nvidia-arbeitet-an-einheitlichem-speicher-api-1610-123624.html>*

## 21.4 Status des Grafiksystems feststellen

Es ist gar nicht so einfach, festzustellen, welche Treiber und Komponenten bei einem aktuell laufenden Grafiksystem aktiv sind.

**X oder Wayland?** Ein Blick in die Prozessliste verrät, ob X (genau genommen der X-Server mit dem Programmnamen `Xorg`) oder Wayland läuft:

```
user$ ps ax | egrep -i "xorg|wayland"
  898 tty1      Ssl+  0:00 /usr/libexec/gdm-wayland-session ...
 1058 tty1      Sl+   0:00 /usr/bin/Xwayland :1024 -rootless ...
```

Bei der Interpretation des Ergebnisses müssen Sie aber aufpassen. Das Grafiksystem wird nämlich zweimal gestartet: einmal zur Anzeige der Login-Box (dafür ist der sogenannte Display Manager zuständig) und noch einmal nach dem Login für den Desktop des eingeloggten Benutzers. Manche Distributionen verwenden für den Display Manager, der relativ wenige Anforderungen an das Grafiksystem stellt, Wayland, für die Desktop-Umgebung dann aber X.

Gewissheit gibt das Kommando `loginctl`, das alle Sessions unter der Kontrolle von `systemd` ausgibt:



```
user$ loginctl
SESSION    UID    USER    SEAT    TTY
    c1     42    gdm      seat0   /dev/tty1
    2     1000  kofler   seat0   /dev/tty2
    4     1000  kofler
```

Im zweiten Schritt können Sie nun Details zu einer Session ermitteln und so unter anderem feststellen, ob die Session X, Wayland oder nur eine TTY-Schnittstelle verwendet, wie dies bei SSH-Logins oder beim Arbeiten in einer Textkonsole der Fall ist:

```
root# loginctl show-session -p Type 2
Type=wayland
```

Sofern Ihre Distribution X verwendet, können Sie die Version des X-Servers mit dem folgenden Kommando feststellen. Das Kommando muss in einer SSH-Sitzung oder in einer Textkonsole ausgeführt werden. Auf dem Beispielrechner läuft der X.org-Server in Version 1.19.3:

X-Server-Version

```
user$ X -showconfig
X.Org X Server 1.19.3
Release Date: 2017-03-15
X Protocol Version 11, Revision 0
```

Eine alternative Vorgehensweise bietet das Kommando `xdpinfo`. Das Kommando muss in einem Terminalfenster innerhalb des Grafiksystems ausgeführt werden. Dank der X-Kompatibilität von Wayland funktioniert das Kommando gleichermaßen unter X und unter Wayland.

```
user$ xdpinfo | grep release
vendor release number: 11903000
```

Um herauszufinden, welche Grafikkarte bzw. welchen Grafikadapter Sie haben und durch welchen Kerneltreiber dieser gesteuert wird, werten Sie die Ausgaben des Kommandos `lspci` aus. Die folgenden Zeilen zeigen einmal das Ergebnis von zwei Rechnern. Einer war mit einer Grafikkarte mit AMD-GPU ausgestattet, der andere mit einer NVIDIA-Grafikkarte:

Grafikkarte und Treiber

```
user$ lspci -k | egrep -A3 'VGA|3D|Display'
02:00.0 VGA compatible controller: Advanced Micro Devices, Inc.
  [AMD/ATI] Cape Verde PRO [Radeon HD 7750/8740 / R7 250E]
  Subsystem: PC Partner Limited / Sapphire Technology Device a001
  Kernel driver in use: radeon
  Kernel modules: radeon
user$ lspci -k | egrep -A3 'VGA|3D|Display'
02:00.0 VGA compatible controller: NVIDIA Corporation GK208
  [GeForce GT 710B] (rev a1)
  Subsystem: Micro-Star International Co., Ltd. [MSI] Device 8c93
  Kernel driver in use: nvidia
  Kernel modules: nvidiafb, nouveau, nvidia_375_drm, nvidia_375
```

**3D-Funktionen  
(Mesa)**

Mit dem Programm `glxinfo`, das sich je nach Distribution im Paket `mesa-utils`, `glx-utils` oder `Mesa-demo-x` versteckt, können Sie die 3D-Funktionen Ihres Systems überprüfen. Das Programm liefert eine Menge Detailinformationen über das laufende GLX-System, also über die OpenGL-3D-Erweiterungen des X Window Systems. `glxinfo` kann auch ausgeführt werden, wenn Sie Wayland verwenden.

Mit `grep` filtern Sie die entscheidenden Zeilen heraus:

```
user$ glxinfo | grep render (Intel-Treiber, X)
direct rendering: Yes
OpenGL renderer string: Mesa DRI Intel(R) Sandybridge Desktop
user$ glxinfo | grep render (Radeon-Treiber, Wayland)
direct rendering: Yes
OpenGL renderer string: ATI Radeon R9 M290X OpenGL Engine
user$ glxinfo | grep render (Radeon-Treiber, X)
direct rendering: Yes
OpenGL renderer string: Gallium 0.4 on AMD CAPE VERDE
user$ glxinfo | grep render (Nouveau-Treiber, Wayland/X)
direct rendering: Yes
OpenGL renderer string: Gallium 0.4 on NV106
user$ glxinfo | grep render (NVIDIA-Treiber, X)
direct rendering: Yes
OpenGL renderer string: Geforce GT 710/PCIe/SSE2
```

Wenn dagegen kein 3D-beschleunigter Treiber läuft, sieht die Ausgabe wie in einem der drei folgenden Beispiele aus. Im ersten Beispiel stehen gar keine 3D-Funktionen zur Verfügung, im zweiten Fall werden sie per Software durch die Mesa-Bibliothek nachgebildet und im dritten Fall durch die `llvmpipe`-Bibliothek. Bei der seit 2012 gebräuchlichen `llvmpipe`-Bibliothek kümmert sich die CPU um die 3D-Funktionen. Das erhöht zwar den Rechenaufwand der CPU beträchtlich, ermöglicht aber die Nutzung von 3D-Funktionen auch ohne einen richtigen 3D-Grafiktreiber.

```
user$ glxinfo | grep render
Xlib: extension "GLX" missing on display ":0.0"
user$ glxinfo | grep render
direct rendering: No
OpenGL renderer string: Mesa GLX Indirect
user$ glxinfo | grep render
direct rendering: Yes
OpenGL renderer string: Gallium 0.4 on llvmpipe (LLVM 3.9, 256 bits)
```

**Logging**

Beim Start von X werden zahlreiche Meldungen, Warnungen und eventuell auch Fehlermeldungen in der Datei `/var/log/Xorg.0.log` gespeichert. Dieses Startprotokoll enthält ausführliche Informationen darüber, welche Konfigurationsdatei verwendet wurde, welche Module geladen wurden, welche Probleme dabei aufgetreten sind, welche Grafikmodi aus welchen Gründen verworfen wurden etc. Einträge innerhalb der Logging-Datei sind durch folgende Codes gekennzeichnet:

- (\*\*) Einstellung aus der Konfigurationsdatei
- (++) Einstellung aus der Kommandozeile
- (==) X-StandardEinstellung
- (--) Einstellung, die sich aus erkannter Hardware ergibt
- (!!) Hinweis
- (II) Hinweis
- (WW) Warnung
- (EE) Fehler

Falls es in `/var/log/` mehrere X-Logging-Dateien gibt, halten Sie Ausschau nach der aktuellsten Datei. Aufgrund der Fülle der Informationen in `Xorg.0.log` gleicht die Suche nach wirklich relevanten Daten leider der sprichwörtlichen Suche nach der Nadel im Heuhaufen. Gegebenenfalls senden Sie einfach die gesamte Logging-Datei jemandem, der sich besser damit auskennt, bzw. posten sie in ein Support-Forum.

Wenn die Datei fehlt, verwendet das Grafiksystem vermutlich Wayland. Dort gilt leider das andere Extrem – Wayland verzichtet auf nahezu jede Protokollierung. Das hat in vor allem damit zu tun, dass Wayland gar keinen eigenen Prozess hat, der mit dem X-Server vergleichbar ist. Wayland definiert nur Protokolle, die in Bibliotheken implementiert werden. Wenn dort Fehler auftreten, werden unter Umständen Fehlermeldungen an den Syslog-Dienst weitergegeben. Diese Meldungen können Sie mit `grep` aus dem Journal oder dem zentralen Syslog-Protokoll extrahieren:

```
root# journalctl | grep -i wayland
```

Tipps zur Fehlersuche bei Wayland-Problemen gibt die folgende Seite:

[https://fedoraproject.org/wiki/How\\_to\\_debug\\_Wayland\\_problems](https://fedoraproject.org/wiki/How_to_debug_Wayland_problems)

## 21.5 Start des Grafiksystems

Dieser Abschnitt fasst einige Informationen zum Starten und Stoppen des Grafiksystems zusammen. In der Regel müssen Sie sich darum nur kümmern, wenn der automatische Start beim Hochfahren des Rechners nicht funktioniert bzw. wenn Sie in den Prozess manuell eingreifen möchten.

Desktop-Distributionen sind so konfiguriert, dass standardmäßig das Grafiksystem gestartet wird. Verantwortlich dafür ist das `systemd`-Target `graphical` (siehe auch [Abschnitt 24.1](#), »`systemd`«). Um vom Grafikmodus in den Textmodus bzw. wieder zurück in den Grafikmodus zu wechseln, führen Sie `systemctl isolate` aus. Das Kommando wird unmittelbar wirksam. Beim Wechsel in den Textmodus werden alle gerade laufenden Programme sofort beendet. Speichern Sie also vorher alle offenen Dateien! Wenn nach dem Wechsel in den Textmodus nur ein schwarzer Bildschirm

`systemd`

sichtbar ist, müssen Sie mit `Alt+F2` oder `Alt+F3` in eine andere Textkonsole wechseln.

```
root# systemctl isolate multi-user    (Textmodus aktivieren)
root# systemctl isolate graphical    (Grafikmodus aktivieren)
```

`systemctl isolate` gilt unmittelbar, aber nur bis zum nächsten Neustart des Rechners. Wenn Sie dauerhaft zwischen Text- und Grafikmodus wechseln wollen, müssen Sie das Default-Target von `systemd` ändern:

```
root# systemctl set-default multi-user (in Zukunft im Textmodus starten)
root# systemctl set-default graphical (in Zukunft im Grafikmodus starten)
```

### Wayland oder X?

Aktuelle Linux-Distributionen, soweit sie überhaupt Wayland unterstützen, sind noch dual eingerichtet: Es stehen also die Grafiksysteme X und Wayland zur Auswahl. Wie aber können Sie darauf Einfluss nehmen, welches Grafiksystem verwendet wird?

Fedora 26 ist die erste Distribution, die nach Möglichkeit versucht, Wayland zu verwenden. Nur wenn das nicht klappt, wird automatisch X gestartet. Wenn Sie X verwenden wollen, obwohl sich Wayland prinzipiell starten lässt, klicken Sie beim Login auf das Zahnrad-Menü und wählen den Eintrag GNOME UNTER XORG aus (siehe [Abbildung 21.3](#)). Der für den Login verantwortliche Display-Manager merkt sich Ihre Entscheidung, bis Sie das nächste Mal einen anderen Eintrag auswählen.



**Abbildung 21.3** Der Gnome Display Manager bietet unter Fedora eine Wahlmöglichkeit zwischen Wayland und X sowie zwischen zwei Gnome-Varianten.

Bei den meisten anderen Distributionen, die im Sommer 2017 Wayland unterstützten, war das Setup gerade umgekehrt: Standardmäßig kommt X zum Einsatz. Sie können sich aber beim Login explizit für Wayland entscheiden, wenn Sie das möchten.

## Die Rolle des Display Managers

Was passiert eigentlich, wenn das Init-System das Grafiksystem aktiviert, wenn bei aktuellen Distributionen also `systemd` das Target `graphical` startet? Gestartet wird vorerst einmal ein sogenannter Display Manager. Dieses Programm läuft zwar an sich schon im Grafikmodus, stellt aber lediglich eine Login-Box zur Verfügung. Erst nach dem erfolgreichen Login startet der Display Manager dann das Desktop-System, auf vielen Distributionen also Gnome oder KDE.

Bei manchen Display Managern kann beim Login die Sprache, eines von mehreren Desktop-Systemen sowie das Grafiksystem (X oder Wayland) ausgewählt werden. Ubuntu bietet beim Login außerdem die Möglichkeit, einen Gastmodus zu aktivieren: Damit können Sie Ihren Computer vorübergehend einer anderen Person leihen, die darauf ihre E-Mails liest oder Online-Banking nutzt – und das, ohne dass diese Person Ihre Daten sieht und ohne dass Sie später auf den Browser-Cache des Gasts zugreifen können.

Damit ist auch schon klar, dass es, typisch für Linux, nicht eine Implementierung des Display Managers gibt, sondern mehrere. Am weitesten verbreitet ist der *Gnome Display Manager* (das Programm `gdm`, siehe [Abbildung 21.3](#)). Populäre Alternativen sind der *Simple Desktop Display Manager* (`sddm`, kommt unter KDE zum Einsatz) und der *Light Display Manager* (`lightdm`), der von Raspbian sowie von Ubuntu bis Version 17.04 verwendet wird bzw. wurde. Nur noch von historischer Bedeutung ist das minimalistische Programm `xdm`, das ehemals *der* Display Manager des X Window Systems war.

Dementsprechend sind auch unterschiedliche Service-Dateien dafür verantwortlich, dass `systemd` den jeweiligen Display Manager startet:

|  |                            |
|--|----------------------------|
| <code>/lib/systemd/system/gdm.service</code>                 | (CentOS 7, Fedora, RHEL 7) |
| <code>/lib/systemd/system/gdm3.service</code>                | (Debian 8)                 |
| <code>/lib/systemd/system/gdm.service</code>                 | (Debian 9)                 |
| <code>/lib/systemd/system/sddm.service</code>                | (Kubuntu, Neon)            |
| <code>/lib/systemd/system/lightdm.service</code>             | (Raspbian)                 |
| <code>/usr/lib/systemd/system/display-manager.service</code> | (SUSE)                     |
| <code>/lib/systemd/system/lightdm.service</code>             | (Ubuntu bis 17.04)         |
| <code>/lib/systemd/system/gdm.service</code>                 | (Ubuntu ab 17.10)          |

## Konfiguration des Display Managers

- gdm** Die Konfigurationsdateien von `gdm` befinden sich in `/etc/gdm` oder `/etc/gdm3`. Sie steuern unter anderem, welche Programme, Kommandos und Scripts für verschiedene Funktionen des Display Managers genutzt werden sollen.
- lightdm** Die Konfiguration von `lightdm` erfolgt durch zwei minimalistische Dateien in `/etc/lightdm`. Änderungen an diesen Dateien sind nur selten erforderlich. Mit der Anweisung `allow-guest=false` in `lightdm.conf` können Sie unter Ubuntu Gast-Sitzungen deaktivieren.
- sddm** `sddm` wird durch die Datei `/etc/sddm.conf` konfiguriert. Anstatt die Datei direkt zu verändern, können Sie dazu das Dialogblatt **STARTEN UND BEENDEN • ANMELDEBILDSCHIRM** der Systemeinstellungen verwenden.
- \*.desktop-Dateien** Wenn Sie mehrere Desktop-Systeme installiert haben, können Sie beim Login im Display Manager in einem Menü auswählen, welcher Desktop bzw. Window Manager gestartet werden soll. Die Daten für dieses Menü befinden sich bei den meisten Distributionen in `*.desktop`-Dateien im Verzeichnis `/usr/share/xsession` (Schlüsselwort `SessionDesktopDir` in der `gdm`-Konfiguration). Beispielsweise enthält die Desktop-Datei zum Start von Gnome die folgenden Zeilen:

```
[Desktop Entry]
Name=GNOME
Comment=This session logs you into GNOME
Exec=gnome-session
...
```

## Auto-Login

Auf Desktop-Systemen ist es manchmal erwünscht, dass der Standardbenutzer beim Start des Rechners automatisch eingeloggt wird. Das ist zwar ein Sicherheitsrisiko, dafür aber bequem. Bei vielen Distributionen können Sie den Auto-Login in den Systemeinstellungen konfigurieren, bei Gnome-Systemen und unter Ubuntu in der Benutzerverwaltung. Die folgenden Absätze erklären die manuelle Konfiguration.

- gdm** Wenn Ihre Distribution `gmd` als Display Manager verwendet, fügen Sie die folgenden Zeilen in den `[daemon]`-Abschnitt von `custom.conf` ein:

```
# Datei /etc/gdm[3]/custom.conf
...
[daemon]
  AutomaticLoginEnable=true
  AutomaticLogin=<loginname>
```

Bei `lightdm` steuert die Variable `autologin-user` in `lightdm.conf` den Auto-Login: `lightdm`

```
# Datei /etc/lightdm/lightdm.conf
[SeatDefaults]
autologin-user=<loginname>
...
```

`openSUSE` sieht eigene Konfigurationsdateien für den Auto-Login vor. Direkte Veränderungen in KDE oder Gnome sind nicht zu empfehlen, weil `SuSEconfig` die Einstellungen bei der nächsten Gelegenheit überschreibt. Stattdessen verändern Sie die Datei `/etc/sysconfig/displaymanager`. Um den Auto-Login zu deaktivieren, weisen Sie der `AUTOLOGIN`-Variablen eine leere Zeichenkette zu. Die Änderungen werden erst gültig, wenn Sie das Kommando `SuSEconfig` ausführen. `openSUSE`

```
# /etc/sysconfig/displaymanager
...
DISPLAYMANAGER_AUTOLOGIN=""
```

### Manueller Start des Desktop-Systems

Normalerweise startet also `systemd` den Display Manager und damit auch das Grafiksystem. Mit einem Login gelangen Sie von dort in das Desktop-System.

Sofern Sie X (nicht Wayland) benutzen, gibt es noch eine Alternative: Sie können sich in einer Textkonsole anmelden und dann `startx` ausführen. Damit wird das Desktop-System für den eingeloggten Benutzer ohne den Umweg über den Display Manager gestartet. `startx`

In der Praxis ist dieser Weg selten empfehlenswert. Die Startvariante ist aber dann praktisch, wenn Sie Probleme mit der X-Konfiguration haben und ohne langwierigen Target-Wechsel eine neue Konfiguration rasch ausprobieren möchten.

Ein unmittelbar mit `startx` vergleichbares Kommando für Wayland gibt es zwar nicht, Sie können aber wie folgt Gnome samt Wayland aus einer Textkonsole starten: `dbus-session`

```
user$ dbus-run-session -- gnome-shell --display-server --wayland
```

Innerhalb von Gnome funktioniert jetzt allerdings der Logout nicht. Abhilfe schafft das folgende Kommando, das zwar nicht besonders elegant ist, aber durchaus wirkungsvoll:

```
user$ killall gnome-shell
```

## 21.6 Konfiguration von X (xorg.conf)

**xorg.conf** Zur Konfiguration von X dienen die Dateien `/etc/X11/xorg.conf` und `/etc/X11/xorg.conf.d/*.conf`. In der Vergangenheit spielte die dort gespeicherte Konfiguration eine große Rolle: Der Start von X war unmöglich, wenn `xorg.conf` fehlte. Mittlerweile hat sich das aber radikal geändert – aktuelle X-Versionen funktionieren vollkommen ohne `xorg.conf`: X ermittelt beim Start die aktuelle Hardware (Grafikkarte, Monitor, Maus, Tastatur) und lädt automatisch geeignete Treiber und Module.

Eine manuelle Konfiguration ist nur in Ausnahmefällen erforderlich, wenn die automatische Konfiguration versagt. Dieser Abschnitt führt in die Syntax von `xorg.conf` ein und gibt einige Konfigurationstipps. Beachten Sie, dass Änderungen an `xorg.conf` erst mit einem Neustart von X wirksam werden:

```
root# systemctl isolate multi-user
root# systemctl isolate graphical
```

Fehler in `xorg.conf` können dazu führen, dass X gar nicht mehr gestartet werden kann. In diesem Fall müssen Sie die Korrekturarbeiten in einer Textkonsole durchführen. Machen Sie sich damit vertraut, bevor Sie an `xorg.conf` herumspielen (siehe [Kapitel 9](#), »Terminalfenster und Konsolen«)!

Keine Konfiguration für Wayland

Bei Wayland gibt es keine zu `xorg.conf` vergleichbare Konfiguration! Wayland verlässt sich auf die automatische Erkennung aller Parameter.

### Aufbau der Konfigurationsdatei xorg.conf

Die Datei `/etc/X11/xorg.conf` ist in mehrere Abschnitte gegliedert, die mit Section "name" eingeleitet und mit EndSection abgeschlossen werden (siehe [Tabelle 21.1](#)).

| Abschnitt   | Bedeutung                             |
|-------------|---------------------------------------|
| Monitor     | Monitordaten                          |
| Device      | Konfiguration der Grafikkarte         |
| Screen      | Bildschirmauflösung                   |
| Files       | Dateinamen (z. B. Font-Verzeichnisse) |
| Module      | Zusatzmodule (z. B. freetype, dri)    |
| ServerFlags | verschiedene Server-Optionen          |
| InputClass  | Tastatur, Maus, Touchpad              |

Tabelle 21.1 xorg.conf-Abschnitte



Das folgende Listing zeigt eine Minimalkonfiguration für Notfälle, wenn das Grafiksystem gar nicht funktionieren will. Die hier vorgeschlagene Konfiguration verwendet den VESA-Treiber und bietet damit keine 3D-Unterstützung.

Minimal-  
konfiguration

```
Section "Monitor"
    Identifier "mon0"
    HorizSync 31 - 94
    VertRefresh 60
EndSection

Section "Device"
    Identifier "dev0"
    Driver "vesa"
EndSection

Section "Screen"
    Identifier "screen0"
    Monitor "mon0"
    Device "dev0"
    DefaultDepth 24
    SubSection "Display"
        Depth 24
        Modes "1024x768"
    EndSubSection
EndSection
```

Die in vielen Abschnitten zwingend erforderliche Identifier-Zeile gibt dem Abschnitt einen Namen und ermöglicht Querverweise zwischen den Abschnitten. Beispielsweise verweist der Abschnitt Screen auf das Device dev0 und den Monitor mon0.

Identifier

### Monitor-Abschnitt

Der Monitor-Abschnitt ist im Regelfall überflüssig, weil moderne Monitore ihre Eckdaten an die Grafikkarte übermitteln. Sollte das bei uralten Monitoren oder in einer virtuellen Maschine nicht funktionieren, können Sie den zulässigen Bereich für die horizontale Zeilenfrequenz (in kHz) und für die Bildfrequenz (in Hz) angeben. Die folgenden Angaben gelten für einen Monitor mit einer Auflösung von 1600×1200 Pixeln und einer maximalen Bildfrequenz von 75 Hz:

```
Section "Monitor"
    ...
    HorizSync 30-95 # Zeilenfrequenz 30 bis 95 kHz (Zeilen/sec)
    VertRefresh 58-78 # Bildfrequenz 58 bis 78 Hz (Bilder/sec)
EndSection
```

**Modeline** Optional können Sie mit `Modeline` exakt angeben, in welchem Grafikmodus der Monitor betrieben werden soll. Ein Grafikmodus wird durch seinen Namen und neun Zahlenwerte bestimmt. Die folgende Zeile zeigt ein Beispiel:

```
Modeline "640x480" 25.175 640 664 760 800 480 491 493 525
```

Damit wird ein Grafikmodus mit 640×480 Pixeln beschrieben. Die Zeichenkette "640x480" ist gleichzeitig auch der Name dieses Modus. Der Zahlenwert 25.175 gibt die Pixelfrequenz (Videobandbreite) in MHz an.

Die nächsten vier Werte betreffen das horizontale Timing: Eine einzelne Bildschirmzeile mit 640 *sichtbaren* Pixeln wird in Wirklichkeit aus 800 *virtuellen* Pixeln zusammengesetzt. Die ersten 640 Pixel werden tatsächlich angezeigt. Während der verbleibenden 160 Pixel wird der Elektronenstrahl durch den HSync-Impuls zurück an den Beginn der nächsten Zeile bewegt. Während dieser Zeit hat der Elektronenstrahl die Intensität 0. Die vier Werte kommen also wie folgt zustande:

```
640    640 Bildschirmpixel anzeigen
664    24 weitere Pixel dunkel tasten
760    96 Pixel lang einen HSync-Impuls erzeugen
800    nochmals 40 Pixel dunkel tasten, d. h. insgesamt 800 virtuelle Punkte
```

Ganz analog wie beim horizontalen Timing sind auch die Angaben für das vertikale Timing in Bildschirmzeilen zu interpretieren:

```
480    480 Zeilen anzeigen
491    11 Zeilen dunkel tasten
493    2 Zeilen lang einen VSync-Impuls erzeugen
525    nochmals 32 Zeilen dunkel tasten, d. h. insgesamt 525 virtuelle Zeilen
```

Aus den jeweils letzten Werten der Vierergruppen und der Pixelfrequenz ergeben sich übrigens die horizontale Zeilenfrequenz und die vertikale Bildfrequenz: 25,175 MHz dividiert durch 800 Pixel pro Zeile ergibt eine Zeilenfrequenz von 31,469 kHz. Die Zeilenfrequenz dividiert durch 525 Zeilen pro Bild liefert die vertikale Bildfrequenz von 60 Hz.

**gtf** Die Parameter für eine `Modeline`-Zeile können Sie ganz komfortabel mit dem Kommando `gtf` ermitteln. Dazu übergeben Sie an das Kommando die gewünschte Auflösung und Bildfrequenz:

```
user$ gtf 1600 1200 60
# 1600x1200 @ 60.00 Hz (GTF) hsync: 74.52 kHz; pclk: 160.96 MHz
Modeline "1600x1200_60.00" 160.96 1600 1704 1880 2160 \
1200 1201 1204 1242 -HSync +Vsync
```

Losgelöst von den Modeline-Angaben kann die Grafikauflösung auch im laufenden Betrieb verändert werden. Zuständig dafür ist das RandR-Protokoll, das ich in [Abschnitt 21.7](#), »Dynamische Konfigurationsänderungen mit RandR«, vorstelle.

RandR

Schließlich können Sie mit `DisplaySize` die Breite und Höhe des Monitors in Millimetern angeben. X wertet diese Informationen aus, um den DPI-Wert zu bestimmen.

DisplaySize

```
DisplaySize 336 252
```

### Device-Abschnitt (Grafikkarte)

Das wichtigste Schlüsselwort im Device-Abschnitt ist `Driver`. Es bestimmt, welcher Treiber geladen werden soll. Die zur Auswahl stehenden Grafiktreiber befinden sich im Verzeichnis `/usr/lib[64]/xorg/modules/drivers`. Im Regelfall erkennt X selbst den geeigneten Treiber. Eine explizite Treibereinstellung ist nur bei ganz neuen Grafikkarten erforderlich oder wenn Sie einen binären Herstellertreiber verwenden.

Falls mehrere PCI-Grafikkarten in den Rechner eingebaut sind, können Sie mit `BusID` genau angeben, welche Sie meinen. Die drei Ziffern geben den PCI-Bus, die Device-Nummer und die Funktion an. Die korrekten Werte können Sie herausfinden, indem Sie in einer Textkonsole `X -scanpci` ausführen. X darf zu diesem Zeitpunkt nicht laufen.

```
Section "Device"
    Driver      "radeon"
    BusID      "1:0:0"
EndSection
```

Nahezu jeder Treiber kennt Optionen zur Steuerung von Spezialeinstellungen, zur Umgehung von Problemen bzw. zur Aktivierung besonderer Funktionen. Detaillierte Informationen gibt die jeweilige `man`-Seite (also beispielsweise `man radeon`).

Treiberspezifische Optionen

### Treiber für Notfälle

Wenn Sie eine Grafikkarte nutzen, zu der es keine Treiber gibt, stellen die drei in diesem Abschnitt vorgestellten Treiber eine Notlösung dar. Auch wenn der Bildaufbau vergleichsweise langsam ist und natürlich keinerlei 3D-Funktionen zur Verfügung stehen, ermöglichen die Treiber zumindest überhaupt eine Nutzung des Grafiksystems.

Mit dem VESA-Treiber können Sie alle VESA-Modi Ihrer Grafikkarte nutzen. Kurz einige Hintergrundinformationen: Die *Video Electronics Standard Association* (VESA) hat eine Reihe von Grafikmodi für Standardauflösungen normiert. Jeder Modus ist durch die folgenden Eckdaten bestimmt: Auflösung (z. B. 1280×1024 Pixel), Farbtiefe und Bildfrequenz. Fast alle Grafikkarten unterstützen neben eigenen Grafikmodi auch eine Menge VESA-Modi.

VESA-Treiber

Wie die nächsten Zeilen zeigen, ist die Verwendung des VESA-Treibers denkbar einfach. Sofern die restliche Konfigurationsdatei korrekt ist, werden alle VESA-Modi berücksichtigt, die die Grafikkarte unterstützt und die der Monitor darstellen kann.

```
Section "Device"
    Identifier    "myDevice"
    Driver        "vesa"
EndSection
```

**Framebuffer-Treiber** Der fbdev-Treiber greift direkt auf den Speicher (Framebuffer) der Grafikkarte zu. Der Treiber setzt damit noch eine Ebene tiefer an als der VESA-Treiber. Er sollte mit fast allen Grafikkarten funktionieren, sofern der Linux-Kernel mit Framebuffer-Unterstützung kompiliert wurde. Dass diese Unterstützung vorhanden ist, erkennen Sie daran, dass die Datei `/proc/fb` existiert.

Eine grundlegende Voraussetzung für die Nutzung des Treibers besteht allerdings darin, dass bereits beim Booten des Rechners der richtige VGA-Modus ausgewählt wird. Bis zum Neustart des Rechners kann X nur in dem so festgelegten Grafikmodus betrieben werden. Zur Auswahl des Modus fügen Sie in die GRUB-Konfigurationsdatei die Kerneloption `vga=N` ein. Die richtigen Werte (dezimal) für N finden Sie in der folgenden Tabelle. Unter SUSE können Sie außerdem mit `hwinfo --framebuffer` eine Liste der Framebuffer-Modi ermitteln, die Ihre Grafikkarte unterstützt.

|                | 640x480 | 800x600 | 1024x640 | 1024x768 | 1280x1024 | 1440x900 | 1600x1200 |
|----------------|---------|---------|----------|----------|-----------|----------|-----------|
| 8 bpp          | 769     | 771     | 874      | 773      | 775       | 864      | 796       |
| 16 bpp (5:5:5) | 784     | 787     | 875      | 790      | 793       | 865      | 797       |
| 16 bpp (5:6:5) | 785     | 788     | 876      | 791      | 794       | 866      | 798       |
| 24 bpp         | 786     | 789     | 877      | 792      | 795       | 867      | 799       |
| 32 bpp         | 809     | 814     | 878      | 824      | 829       | 868      | 834       |

In `xorg.conf` müssen Sie lediglich die richtige Driver-Zeichenkette angeben:

```
# in /etc/X11/xorg.conf
...
Section "Device"
    Identifier    "myDevice"
    Driver        "fbdev"
EndSection
```

**vga-Treiber** Der vga-Treiber unterstützt nur  $640 \times 480$  oder  $800 \times 600$  Pixel bei einer Farbtiefe von 4 Bit (also 16 Farben) und ist somit nur die letzte Notlösung. Weitere Details finden Sie mit `man vga`.

### Screen-Abschnitt (Auflösung, Farbanzahl)

Der Screen-Abschnitt verbindet den Monitor und die Grafikkarte und gibt an, in welcher Auflösung und mit wie vielen Farben die Grafikkarte verwendet werden soll. Die

Schlüsselwörter `Device` und `Monitor` verweisen auf die bereits definierte Grafikkarte und den Monitor. `DefaultDepth` gibt an, wie viele Farben zur Verfügung stehen. Die Angabe erfolgt in Bit pro Pixel. Bei 24 Bit stehen je Grundfarbe 8 Bit – also je 256 Rot-, Grün- und Blautöne – zur Verfügung, insgesamt  $2^{24}$  Farben. Bei 16 Bit stehen je Farbton nur 5 Bit zur Verfügung, ein Bit bleibt üblicherweise ungenutzt.

Innerhalb des `Screen`-Abschnitts können mehrere `Display`-Unterabschnitte angegeben werden, je einer für jede Farbkonfiguration (Schlüsselwort `Depth`). Im Beispiel unten ist nur ein Modus mit 24 Bit pro Pixel definiert:

```
Section "Screen"
    Identifier    "Screen0"
    Device       "Videocard0"
    DefaultDepth 24

    SubSection "Display"
        Depth     24
        Modes     "1280x1024"
    EndSubSection
EndSection
```

In der optionalen `Modes`-Zeile kann die gewünschte Auflösung angegeben werden. Wenn die Zeile weggelassen wird, entscheidet sich X automatisch für die bestmögliche Auflösung, die für den Monitor und die Grafikkarte geeignet ist.

### InputClass-Abschnitt (Tastatur, Maus und Trackpad)

Im Laufe der X-Geschichte gab es alle möglichen Treiber, die sich um Eingabegeräte kümmerten: zuerst den `keyboard`-Treiber für Tastaturen, den `mouse`-Treiber für Mäuse und den `synaptic`-Treiber für Track- und Touchpads, dann den generischen `evdev`-Treiber, der universell für alle Eingabegeräte geeignet war.

Aktuelle Versionen von X verwenden keinen dieser Treiber mehr, sondern stattdessen den `libinput`-Treiber, der mit der gleichnamigen Bibliothek zusammenarbeitet. Ein entscheidender Vorteil des `libinput`-Systems besteht darin, dass es kompatibel zu Wayland ist, sodass X und Wayland zumindest in diesem Punkt auf gemeinsame Treiber zurückgreifen können.

**libinput**

Eine Liste aller für `libinput` sichtbaren Geräte können Sie mit `libinput-list-devices` ermitteln. Eine wesentlich kompaktere Zusammenfassung gibt `xinput list`. Mit `xinput` können Sie auch die Eigenschaften einzelner Geräte verändern (siehe `man xinput`).

**xinput**

**InputClass-Abschnitt** Normalerweise verwaltet libinput alle Eingabegeräte, die es erkennt. Manche Distributionen schränken die Wirkung von libinput auf bestimmte Gerätetypen ein. Das verhindert, das z. B. auch Joystick-Ereignisse von libinput verarbeitet werden. Eine derartige Konfiguration ist bei openSUSE zu sehen:

```
# Datei /etc/X11/xorg.conf.d/60-libinput.conf (openSUSE 42.2)
Section "InputClass"
    Identifier "libinput pointer catchall"
    MatchIsPointer "on"
    MatchDevicePath "/dev/input/event*"
    Driver "libinput"
EndSection
Section "InputClass"
    Identifier "libinput keyboard catchall"
    MatchIsKeyboard "on"
    MatchDevicePath "/dev/input/event*"
    Driver "libinput"
EndSection
... analoge Abschnitte auch für Touchpads und Touchscreens
```

Um die Einstellung des Tastaturlayouts sowie um Optionen zur Bedienung von Trackpad und Maus kümmert sich das Desktop-System, also z. B. Gnome. Zur Defaulteinstellung für das Tastaturlayout werden je nach Distribution die Einstellungen einer Datei in `/etc/X11/xorg.conf.d` ausgewertet. Die folgenden Zeilen zeigen die Einstellungen einer Fedora-Installation im deutschen Sprachraum:

```
# Datei /etc/X11/xorg.conf.d/00-keyboard.conf (Fedora 26)
Section "InputClass"
    Identifier "system-keyboard"
    MatchIsKeyboard "on"
    Option "XkbLayout" "de"
    Option "XkbVariant" "nodeadkeys"
EndSection
```

## 21.7 Dynamische Konfigurationsänderungen mit RandR

Die *Resize and Rotate Extension* (RandR) erlaubt es, Teile der Konfiguration des Grafiksystems im laufenden Betrieb zu verändern, soweit der Grafiktreiber dies unterstützt. Aktuelle Desktop-Systeme passen sich automatisch an die neuen Rahmenbedingungen – beispielsweise an die geänderte Bildschirmauflösung – an und müssen nicht neu gestartet werden.

Manuelle Änderungen können Sie mit dem Kommando `xrandr` ausführen. Mehr Komfort bieten die entsprechenden Module der Gnome- bzw. KDE-Systemeinstellungen.

RandR ist eigentlich eine X-spezifische Erweiterung. Wayland bietet aber ähnliche Funktionen. Beispielsweise sind die Gnome-Einstellungen in der Lage, die Grafikauflösung zu verändern. Das im Folgenden vorgestellte Kommando `xrandr` kann allerdings nicht genutzt werden. Für dieses Kommando gibt es gegenwärtig (im Sommer 2017) keine vergleichbare Alternative. Ein Python-Script beweist aber, dass ein derartiges Programm möglich ist, zumindest für den Fall, dass Wayland in Kombination mit Gnome ausgeführt wird: Wayland

<https://github.com/xytovl/display-config>

Wenn Sie `xrandr` ohne Parameter bzw. mit der Option `-q` ausführen, zeigt es den aktuellen Status von X an. Die folgende Ausgabe ist auf einem System mit zwei Monitoren entstanden. Am Signalausgang DFP1 (*Display Port*) war ein 4k-Monitor mit einer Auflösung von 3840 × 2160 Pixel angeschlossen, am Signalausgang DFP5 (HDMI) ein Monitor mit 1920 × 1200 Pixel. Die Namen der Signalausgänge variieren je nach Grafiktreiber und können z. B. auch `VGA-0` oder `DVI-I-0` lauten. xrandr

```
user$ xrandr
Screen 0: minimum 320 x 200, current 5760 x 2160, maximum 16384 x 16384
DFP1 connected primary 3840x2160+0+0
  (normal left inverted right x axis y axis) 621mm x 341mm
  3840x2160    30.0*+
  2560x1600    60.0
  2560x1440    60.0
  ...
DFP5 connected 1920x1200+3840+0
  (normal left inverted right x axis y axis) 519mm x 324mm
  1920x1200    60.0*+
  1920x1080    60.0    59.9    60.1    60.0
  1600x1200    60.0
  ...
```

`xrandr` kann mit gewöhnlichen Benutzerrechten ausgeführt werden. Alle durchgeführten Änderungen gelten aber nur bis zum nächsten Logout. Das folgende Kommando reduziert die Auflösung auf 1280 × 1024 Punkte:

```
user$ xrandr --size 1280x1024
```

Das nächste Kommando aktiviert die Signalausgänge für DVI und VGA. Die Option `--auto` bewirkt, dass jeder Monitor in der für ihn optimalen Auflösung und Bildfrequenz betrieben wird.

```
user$ xrandr --output DVI-I-0 --auto --output VGA-0 --auto
user$ xrandr --output VGA-0 --off (schaltet den VGA-Ausgang wieder ab)
```

Weitere `xrandr`-Beispiele folgen im nächsten Abschnitt zur Dual-Head-Konfiguration.

**Neue  
Auflösungen  
definieren**

Mit `xrandr` können Sie nur solche Auflösungen aktivieren, die der Treiber der Grafikkarte für den angeschlossenen Monitor vorsieht (siehe das Ergebnis von `xrandr`). Wenn Sie eine andere Auflösung wünschen, z. B. zur Aufnahme eines Screenshots, müssen Sie diese zuerst mit `xrandr --newmode` definieren, beispielsweise so:

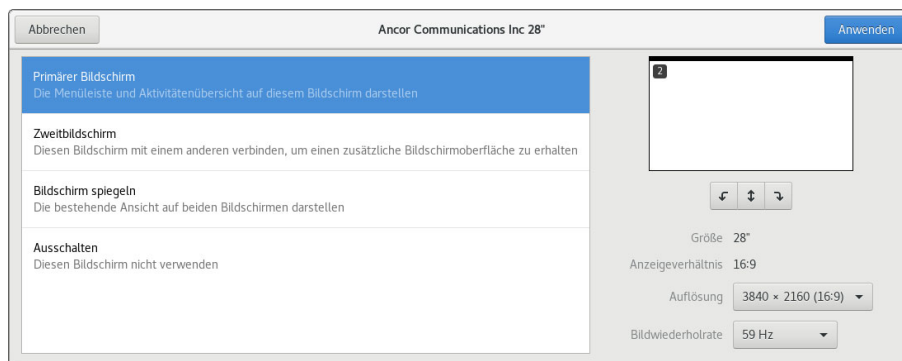
```
user$ xrandr --newmode 1280x720 74.18 1280 1390 1430 1650 720 725 730 750
user$ xrandr --addmode HDMI1 1280x720
user$ xrandr --size 1280x720
```

Beim zweiten `xrandr`-Kommando ist es entscheidend, dass Sie an die Option `--addmode` den Namen des aktiven Grafikausgangs übergeben. Falls Sie die Parameter für den gewünschten Modus nicht kennen (die Syntax ist dieselbe wie bei den `XMode`-Zeilen in `xorg.conf`), finden Sie im Internet in der Regel ein entsprechendes Beispiel. Suchen Sie z. B. nach *modeline 1280x720*. Alternativ können Sie die Parameter auch mit dem Kommando `gtf` errechnen:

```
root# gtf 1280 720 60
# 1280x720 @ 60.00 Hz (GTF) hsync: 44.76 kHz; pclk: 74.48 MHz
Modeline "1280x720_60.00" 74.48 1280 1336 1472 1664 \
    720 721 724 746 -HSync +Vsync
```

**RandR-Einstel-  
lungen in Gnome  
und KDE**

Unter Gnome 3.n verändern Sie die RandR-Konfiguration mit dem Modul **BILDSCHIRME** der Systemeinstellungen. Bei aktuellen Distributionen funktioniert das gleichermaßen unter X und unter Wayland. ([Abbildung 21.4](#) wurde unter Wayland erstellt.)



**Abbildung 21.4** RandR-Einstellungen unter Gnome verändern

Wenn es mehr als einen Monitor gibt, wird standardmäßig auf beiden dasselbe Bild angezeigt (**BILDSCHIRME SPIEGELN**), wobei sich die Bildgröße aus dem jeweils kleineren Wert der horizontalen und vertikalen Auflösung ergibt. Erst wenn Sie bei einem Monitor die Option **PRIMÄRER BILDSCHIRM** oder **ZWEITBILDSCHIRM** angeben, können Sie die beiden Monitore getrennt konfigurieren und ihre Position relativ zueinander verändern. Die Einstellungen werden in `.config/monitors.xml` gespeichert. Sie gelten nur für den aktuellen Benutzer.



Um die Bildschirmeinstellungen auch für den von `gdm` angezeigten Login-Bildschirm zu verändern, kopieren Sie `monitors.xml` in das `gdm`-Verzeichnis:

```
user$ sudo cp ~/.config/monitors.xml /var/lib/gdm/.config
```

Das KDE-Gegenstück zu `gnome-display-properties` ist das Modul ANZEIGE UND MONITOR der Systemeinstellungen. Es bietet im Wesentlichen dieselben Funktionen, speichert seine Einstellungen aber in der Datei `.local/share/kscreen/<id>`, wobei `<id>` der ID-Code des jeweiligen Monitors ist.

Ubuntu mit Unity verwendet eine modifizierte Variante des Gnome-Einstellungsmoduls. Die Syntax und der Speicherort der Konfigurationsdatei sind wie unter Gnome.

Andere Desktop-Systeme stellen zum Teil eigene Benutzeroberflächen zur RandR-Konfiguration zur Verfügung oder greifen auf das Programm `arandr` zurück. Es ist bei vielen Distributionen in den Paketquellen enthalten bzw. kann sonst von der folgenden Webseite heruntergeladen werden: **ARandR**

*<http://christian.amsuess.com/tools/arandr>*



# Kapitel 22

## Administration des Dateisystems

Dieses Kapitel beschreibt verschiedene Facetten der Administration des Dateisystems. Das Kapitel richtet sich an fortgeschrittene Linux-Anwender und geht auf die folgenden Themen ein:

- ▶ **Wie alles zusammenhängt:** Dieser Abschnitt gibt einen ersten Überblick darüber, wie verschiedene Aspekte des Linux-Dateisystems zusammenhängen.
- ▶ **USB-Datenträger formatieren und nutzen:** Für alle Ungeduldigen fasst dieser Abschnitt zusammen, wie Sie auf Kommandoebene einen USB-Stick bzw. eine SD-Karte formatieren bzw. wie Sie eine externe Festplatte dauerhaft in den Verzeichnisbaum integrieren. Die entsprechenden Grundlagen werden dann im weiteren Verlauf des Kapitels erläutert.
- ▶ **Device-Namen:** Linux-intern werden Festplatten und andere Datenträger sowie deren Partitionen über Device-Dateien wie `/dev/sdc3` angesprochen. Dieser Abschnitt fasst das Schema für die Nomenklatur und Nummerierung zusammen.
- ▶ **Partitionierung der Festplatte/SSD:** Die Partitionierung einer Festplatte, SSD oder eines virtuellen Datenträgers (in Virtualisierungssystemen bzw. in der Cloud) ist ein zentraler Bestandteil der Installation von Linux. Manchmal ist es aber auch im Betrieb von Linux erforderlich, eine neue Partition hinzuzufügen. Je nachdem, welche Partitionierungsart für die Festplatte gilt (MBR oder GPT), müssen Sie unterschiedliche Werkzeuge verwenden, um die Partitionierung zu ändern.
- ▶ **Dateisystemtypen:** Nur wenige Betriebssysteme unterstützen so viele Dateisystemtypen wie Linux. Dieser Abschnitt fasst die wichtigsten Varianten zusammen.
- ▶ **Verwaltung des Dateisystems:** Hier erfahren Sie, wie einzelne Datenpartitionen manuell in das Dateisystem eingefügt werden (`mount`) und wie dieser Vorgang automatisiert wird (`/etc/fstab`).
- ▶ **Linux- und Windows-Dateisysteme:** Mehrere Abschnitte geben Tipps und Hinweise zur Nutzung der Dateisysteme `ext4`, `btrfs`, `xfs`, `vfat` und `ntfs`.
- ▶ **CDs/DVDs:** Für Daten-CDs und Daten-DVDs gibt es ebenfalls eigene Dateisystemtypen, die in diesem Abschnitt kurz vorgestellt werden.

- ▶ **Externe Datenträger (USB):** Wenn Sie eine externe Festplatte oder einen USB-Memorystick anschließen, erscheint zumeist automatisch ein Fenster des Dateimanagers. Dieser Abschnitt erklärt, was hinter den Kulissen passiert und wie Sie externe Datenträger bei Bedarf auch manuell nutzen.
- ▶ **Swap-Partitionen oder -Dateien:** Wenn Linux zur Ausführung der Programme zu wenig Arbeitsspeicher hat, lagert es Teile des Speichers in sogenannte Swap-Partitionen oder in Swap-Dateien aus.
- ▶ **RAID:** Mit RAID (*Redundant Array of Inexpensive/Independent Disks*) verknüpfen Sie die Partitionen mehrerer Festplatten miteinander, um auf diese Weise ein zuverlässigeres und/oder schnelleres Gesamtsystem zu erreichen. Dieser Abschnitt geht kurz auf die Grundlagen von RAID ein.
- ▶ **LVM:** Der *Logical Volume Manager* (kurz LVM) ermöglicht eine flexiblere Verwaltung von Partitionen. Mit LVM können Sie beispielsweise Partitionen mehrerer Festplatten zu einer virtuellen Partition vereinen, die Größe von Partitionen im laufenden Betrieb ändern etc.
- ▶ **SMART:** Die *Self-Monitoring Analysis and Reporting Technology* (SMART) ermöglicht es, während des Betriebs von Festplatten statistische Daten zu erfassen und auf diese Weise drohende Zuverlässigkeitsprobleme schon zu erkennen, bevor es zu Datenverlusten kommt.
- ▶ **SSD-TRIM:** Mit dem TRIM-Befehl, den alle aktuellen Solid State Disks unterstützen, kann das Betriebssystem der SSD mitteilen, welche Speicherblöcke des Dateisystems nach dem Löschen einer Datei ungenutzt sind. Die SSD kann dann die interne Nutzung der Speicherzellen optimieren.
- ▶ **Verschlüsselte Dateisysteme:** Wenn Sie vermeiden möchten, dass unbefugte Personen – etwa nach einem Rechnerdiebstahl – Ihre Daten lesen können, müssen Sie Ihre Dateien bzw. Dateisysteme verschlüsseln. Linux stellt hierfür unterschiedliche Verfahren zur Auswahl, wobei das populärste Verfahren momentan auf dem `dm_crypt`-Kernelmodul basiert.

Über die Administration von Dateisystemen ließe sich natürlich noch mehr sagen bzw. schreiben. So viel Platz ist hier aber nicht. Stattdessen müssen hier einige Querverweise genügen:

- ▶ **Nutzung des Dateisystems:** Kommandos zum Kopieren von Dateien oder zum Erstellen von Backups, Hintergründe zu den Zugriffsrechten von Dateien etc. wurden in diesem Buch bereits in [Kapitel 11](#), »Dateien und Verzeichnisse«, vorgestellt.
- ▶ **Netzwerkdateisysteme:** Linux gibt Ihnen die Möglichkeit, Verzeichnisse anderer Rechner in Ihren Verzeichnisbaum zu integrieren. Ich gehe in diesem Buch auf die Dateisysteme CIFS (Windows/Samba, siehe [Abschnitt 28.7](#), »Client-Zugriff«) sowie auf NFS und AFP ein (siehe [Kapitel 29](#)).

- ▶ **Disk-Quotas:** Dabei handelt es sich um ein System, das steuert, wie viel Platz einzelne Benutzer auf der Festplatte beanspruchen dürfen. Wird die Grenze überschritten, können keine neuen Dateien mehr angelegt werden. Disk-Quotas sind praktisch, wenn ein Rechner Speicherplatz für viele Benutzer (Schüler, Studenten etc.) zur Verfügung stellt. Eine gute Einführung finden Sie unter:

<https://wiki.ubuntuusers.de/quota>

- ▶ **Cluster-Dateisysteme:** Cluster-Dateisysteme bzw. globale Dateisysteme verbinden Daten mehrerer Rechner zu einem virtuellen Dateisystem. Damit lassen sich riesige Datenspeicher bilden und von mehreren Rechnern parallel nutzen.

Abermals bietet Linux gleich mehrere Verfahren, um derartige Dateisysteme zusammenzustellen, z. B. mit dem OCFS (*Oracle Cluster Filesystem*), mit GFS (*Global Filesystem*) oder mit Ceph:

<https://oss.oracle.com/projects/ocfs2>

<https://sourceware.org/cluster/gfs>

<https://ceph.com>

## 22.1 Wie alles zusammenhängt

Die vielen Aspekte bei der Verwaltung des Dateisystems sind bisweilen verwirrend. Dieser Abschnitt versucht, die wichtigsten Zusammenhänge kurz und übersichtlich darzustellen. Um den Text möglichst übersichtlich zu halten, beschränke ich mich hier auf eingebaute Festplatten/SSDs und gewöhnliche Linux-Dateisysteme. DVD-Laufwerke, externe Datenträger, LVM- und RAID-Systeme etc. bleiben außen vor.

Den eingebauten Festplatten und SSDs sind unter Linux Device-Dateien zugeordnet. Alle gängigen Distributionen verwenden `/dev/sda` für die erste Festplatte bzw. für den ersten Datenträger, `/dev/sdb` für den zweiten Datenträger etc. Generell schreibe ich in diesem Buch oft einfach von »Festplatten«. Damit meine ich gleichermaßen herkömmliche Festplatten und moderne Solid State Disks ohne bewegliche Teile.

**Festplatten und  
SSDs**

Wenn Sie Linux in Virtualisierungssystemen bzw. in der Cloud verwenden, ist der Datenträger virtuell. Aus Sicht dieses Kapitels ändert das nicht viel. Aber während die Größe einer SSD oder Festplatte unveränderlich ist, kann die Kapazität eines Cloud-Datenspeichers per Mausklick bzw. per Upgrade geändert werden – unter Umständen sogar im laufenden Betrieb.

Um auf einer Festplatte mehrere, voneinander unabhängige Dateisysteme unterzubringen, muss sie in Abschnitte (Partitionen) unterteilt werden. Auch den Partitionen sind Device-Dateien zugeordnet, beispielsweise `/dev/sda1` für die erste Partition der ersten Festplatte. Im Detail ist die Device-Nomenklatur für Partitionen in [Abschnitt 22.3](#) zusammengefasst.

**Partitionen**

- Systempartition** Beim Start von Linux greift der Kernel als Erstes auf die Systempartition (root-Partition) zu. Deren Device-Name oder die UUID (*Universal Unique Identifier*) des darauf enthaltenen Dateisystems wird in einem Kernelparameter in der GRUB-Konfigurationsdatei angegeben.
- Weitere Partitionen** Neben der Systempartition, die unbedingt erforderlich ist, kann es weitere Partitionen geben, deren Dateisysteme bereits beim Start von Linux berücksichtigt werden sollen. Diese Dateien sind in der Datei `/etc/fstab` verzeichnet. Diese Datei muss sich wiederum in der Systempartition befinden. Sie wird im Rahmen des Init-Prozesses ausgewertet.
- Konsistenztest/ Reparatur** Beim Einbinden von Partitionen in den Verzeichnisbaum wird automatisch die Konsistenz der Dateisysteme überprüft. Ist der Rechner zuletzt abgestürzt bzw. wurde er wegen eines Stromausfalls nicht ordnungsgemäß heruntergefahren, kommt es zu einer automatischen Reparatur des Dateisystems oder zu anderen Sicherheitsmaßnahmen, die weitere Konsistenzfehler oder -schäden verhindern sollen. Ein entsprechender Konsistenztest wird aber auch automatisch nach einer bestimmten Nutzungsdauer durchgeführt. Im Detail ist dieser Vorgang von der Distribution, vom Dateisystemtyp und von der individuellen Konfiguration abhängig.
- Verzeichnisbaum statt Laufwerksbuchstaben** Während es unter Windows üblich ist, getrennte Dateisysteme über Laufwerksbuchstaben anzusprechen (C:, D: etc.), werden in Linux sämtliche Dateisysteme in einem Verzeichnisbaum zusammengefasst. Der Zugriff auf die Systempartition erfolgt über das Wurzelverzeichnis `/`. Der Startpunkt aller anderen Dateisysteme kann je nach Distribution und Konfiguration variieren. Üblich sind `/mnt-`, `/media-` oder `/run-`Unterverzeichnisse, beispielsweise `/media/dvd` oder `/run/media/benutzername/dvdname`.
- Dateisysteme hinzufügen** Es ist möglich, im laufenden Betrieb weitere Dateisysteme in den Verzeichnisbaum einzubinden bzw. wieder aus ihm zu lösen. Beim Anstecken eines externen Datenträgers (z. B. eines USB-Sticks) erfolgt das zumeist automatisch. Wenn dieser Automatismus nicht funktioniert bzw. wenn er bewusst deaktiviert wurde, kann `root` mit den Kommandos `mount` und `umount` Dateisysteme auch manuell einbinden bzw. lösen. Die einzige Konstante ist die Systempartition: Sie kann während des Betriebs nicht aus dem Dateisystem gelöst werden. Das ist dann erst beim Herunterfahren des Rechners möglich.
- Dateisystemtypen** Linux unterstützt sehr viele Dateisystemtypen. Die Systempartition muss in einem Linux-Dateisystem vorliegen (z. B. `ext4`, `btrfs` oder `xfs`). Bei den restlichen Partitionen ist die Auswahl noch größer. Infrage kommen beispielsweise auch Windows-, Unix- oder Apple-Dateisysteme.

## 22.2 USB-Datenträger formatieren und nutzen

Sozusagen als Vorgeschmack auf die nachfolgenden Abschnitte fasse ich hier zusammen, wie Sie mit externen Datenträgern unter Linux auf Kommandoebene umgehen. Der Abschnitt richtet sich an Linux-Anwender, die schon ein wenig Erfahrung beim Arbeiten im Terminal haben und nicht immer ein Desktop-System wie KDE oder Gnome nutzen. Details und Grundlagen zu den hier präsentierten Kommandos folgen dann im weiteren Verlauf des Kapitels.

### USB-Stick oder SD-Karte formatieren

Immer wieder kommt es vor, dass Sie unter Linux rasch einen USB-Stick oder eine SD-Karte so formatieren möchten, dass der Datenträger später auch auf einem Windows-Notebook oder in der Kamera verwendet werden kann. Die hier beschriebene Vorgehensweise gilt auch für externe Festplatten.

Zuerst ermitteln Sie mit `lsblk` den Device-Namen des USB-Sticks. Bei der Erkennung des richtigen Devices ist vor allem die `SIZE`-Spalte hilfreich. `/dev/sda` bezeichnet immer die erste, interne Festplatte oder SSD – dieses Device dürfen Sie auf keinen Fall für die weiteren Arbeiten verwenden! Sie würden sonst die Daten Ihrer Festplatte/SSD löschen.

Device ermitteln

Im folgenden Beispiel lautet der richtige Device-Name `/dev/sdb`. Sollte der Datenträger bereits Dateisysteme enthalten und sind diese aktiv (Spalte `MOUNTPOINT`), müssen Sie diese aus dem Verzeichnisbaum lösen. In diesem Beispiel trifft das für das Verzeichnis `/run/media/kofler/Ohne Titel` zu.

Mitunter ist es außerdem notwendig, den USB-Stick oder die SD-Karte neu zu partitionieren. Im folgenden Listing erfolgt dies mit den beiden `parted`-Kommandos. Achtung, mit dem folgenden `parted`-Kommando löschen Sie den gesamten Datenträger! Zuletzt richtet `mkfs.vfat` das neue Dateisystem ein, das hier den Namen (Volume-Label) `FOTOS` erhält:

Partitionieren und formatieren

```
root# lsblk
NAME                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                  8:0    0 223,6G  0 disk
  sda1                8:1    0   95M  0 part /boot/efi
  sda2                8:2    0 170,5G  0 part
    fedora-swap       253:0   0    2,8G  0 lvm  [SWAP]
    fedora-root       253:1   0    28G   0 lvm  /
    fedora-home       253:2   0 139,7G  0 lvm  /home
sdb                  8:16   1    7,4G  0 disk
  sdb1                8:17   1    7,4G  0 part /run/media/kofler/Ohne Titel
root# umount /dev/sdb1
```

```

root# parted /dev/sdb mklabel msdos
  Warnung: Die bestehende Partitionstabelle und alle Daten
  auf /dev/sdb werden gelöscht. Wollen Sie fortfahren?
  Ja/Yes/Nein/No? ja
root# parted /dev/sdb 'mkpart primary fat32 1mib -1mib'
root# mkfs.vfat -F 32 -n FOTOS /dev/sdb1

```

Je nach Verwendungszweck ist es besser, anstelle des VFAT-Dateisystems ein NTFS- oder ein Linux-Dateisystem einzurichten. Dazu gehen Sie wie folgt vor:

```

root# mkfs.ntfs --fast /dev/sdb1 (NTFS)
root# mkfs.ext4 /dev/sdb1 (Linux-Dateisystem)

```

Bevor Sie `mkfs.ntfs` ausführen können, müssen Sie eventuell das Paket `ntfsprogs` installieren.

### USB-Datenträger manuell einbinden

Unter KDE, Gnome oder einem vergleichbaren Desktop-System werden USB-Datenträger nach dem Anstecken automatisch in den Verzeichnisbaum eingebunden; zumeist erscheint auch ein Dateimanager-Fenster auf dem Bildschirm, damit Sie auf die Dateien zugreifen können.

Hinter den Kulissen wird dabei ein `mount`-Kommando ausgeführt – und eben dieses Kommando müssen Sie manuell ausführen. Dazu ermitteln Sie zuerst mit `lsblk` den richtigen Device-Namen. Die meisten USB-Datenträger sind so formatiert, dass sich das Dateisystem auf der ersten Partition befindet (z. B. `/dev/sdb1`). Mitunter wird der ganze Datenträger ohne Partitionierung genutzt – dann würde der Device-Name z. B. `/dev/sdb` lauten. Schließlich kann es sein, dass ein Datenträger mehrere Partitionen und entsprechend mehrere Dateisysteme enthält. Das kommt vor allem bei Festplatten vor.

Sobald Sie wissen, auf welches Device Sie zugreifen möchten, können Sie versuchen, die folgenden Kommandos auszuführen:

```

root# mkdir /media/data
root# mount /dev/sdb1 /media/data

```

`mount` erkennt das Dateisystem selbst und verwendet die Defaulteinstellungen für dieses Dateisystem. Diese Einstellungen reichen aus, damit `root` auf die Dateien zugreifen kann. Bei VFAT-Dateisystemen müssen Sie allerdings zusätzlich die Option `utf8` angeben, damit Dateinamen mit internationalen Zeichen wie `äöü` korrekt angezeigt werden:

```

root# mount /dev/sdb1 /media/data

```



In [Abschnitt 22.8](#), »Verwaltung des Dateisystems (mount und /etc/fstab)«, lernen Sie einige weitere Optionen kennen, die es auch anderen Benutzern als `root` ermöglichen, Dateien auf einem Windows-Dateisystem zu verändern.

### Externe Festplatte automatisch einbinden

Mitunter sind externe Festplatten ständig mit dem Rechner verbunden – z. B. als Backup-Ort oder als zusätzlicher Speicher, weil die interne Festplatte zu klein ist. Wenn Sie ein Desktop-System wie KDE oder Gnome verwenden, bindet dieses den Datenträger automatisch ein, wobei normalerweise das Verzeichnis `/run/media/login-name/dateisystemname` verwendet wird. In diesem Fall ist die hier präsentierte Anleitung hinfällig.

Wenn Sie aber über ein anderes Verzeichnis auf den Datenträger zugreifen möchten oder ohne Desktop-System arbeiten, dann müssen Sie `/etc/fstab` modifizieren. Das Ziel ist es, dass Linux während des Boot-Prozesses den Datenträger dauerhaft in den Verzeichnisbaum einbindet. Zur richtigen `fstab`-Konfiguration müssen Sie drei Dinge wissen:

- ▶ Welches Dateisystem befindet sich auf der Festplatte?
- ▶ Welche UUID-Nummer verwendet das Dateisystem?
- ▶ Welche UID-Nummer hat der Benutzer, der auf die Daten zugreifen soll?

Um die beiden ersten Fragen zu beantworten, verbinden Sie das USB-Kabel der Festplatte mit Ihrem Rechner, ermitteln mit `lsblk` den aktuellen Device-Namen und mit `blkid` die restlichen Daten. Im folgenden Beispiel enthält der Datenträger ein VFAT-Dateisystem, die UUID lautet `F5EC-031F`:

Dateisystemtyp  
und UUID  
ermitteln

```
root# lsblk
...
sdb          8:16   1   7,4G  0 disk
  sdb1       8:17   1   7,4G  0 part
root# blkid /dev/sdb1
/dev/sdb1: LABEL="MYDISK" UUID="F5EC-031F" TYPE="vfat" PARTUUID="fac67ace-01"
```

Die UID-Nummer (User-Identifikation) des ersten auf dem Linux-Rechner eingerichteten Benutzers lautet zumeist 1000. Die UIDs anderer Benutzer entnehmen Sie bei Bedarf der dritten Spalte in `/etc/passwd`.

UID

Der Datenträger soll zukünftig über das Verzeichnis `/media/data` genutzt werden:

Verzeichnis  
einrichten

```
root# mkdir /media/data
```

Damit Linux den Datenträger während des Boot-Vorgangs dort einbindet, fügen Sie mit einem Editor in `/etc/fstab` die folgenden Daten ein. Dabei ist es wichtig, dass alle

`fstab`-Einstel-  
lungen für `vfat`

sechs Spalten in einer Zeile angegeben werden. Das ist hier aus Platzgründen nicht möglich. Achten Sie auch darauf, dass die mit `uid=1000` eingeleiteten Optionen durch Kommata getrennt werden müssen (nicht durch Leerzeichen!).

```
UUID=F5EC-031F /media/data vfat
uid=1000,gid=1000,fmask=0022,dmask=0022,flush,utf8,noexec,nofail 0 0
```

Sie erreichen mit diesen Einstellungen, dass der Benutzer, der die UID 1000 hat, Dateien und Verzeichnisse lesen und verändern darf. Bei den meisten Distributionen ist das der erste eingerichtete Benutzer. Wenn der Zugriff auf die Festplatte durch ein Script erfolgt, das mit `root`-Rechten läuft, können Sie auf die Optionen `uid`, `gid`, `fmask` und `dmask` verzichten; dann hat nur `root` Schreibrechte. Eine detaillierte Beschreibung der vier Optionen gibt man `fstab`.

Alle anderen Benutzer haben Lesezugriff auf den Datenträger. Aus Sicherheitsgründen darf niemand die Programme ausführen, die sich auf dem Datenträger befinden (`noexec`). Sollte die Festplatte beim nächsten Boot-Prozess nicht mit dem Rechner verbunden sein, wird der Boot-Vorgang ohne Fehler fortgesetzt. Bei einem späteren Anstecken der Festplatte muss dann aber manuell `mount` ausgeführt werden:

```
root# mount /media/data
```

**fstab-Einstellungen für ntfs**

Je nach Dateisystemtyp muss die Zeile in `/etc/fstab` angepasst werden. Die folgenden Einstellungen gelten für ein NTFS-Dateisystem:

```
UUID=713052474ADB2774 /media/data ntfs
uid=1000,gid=1000,fmask=0022,dmask=0022,flush,nofail,noexec 0 0
```

**fstab-Einstellungen für ext4**

Bei einem Linux-Dateisystem wie `ext4` gelten dieselben Zugriffsregeln wie bei lokalen Dateisystemen. Die Optionen `uid`, `gid`, `fmask` und `dmask` entfallen daher:

```
UUID=e593f718-27a4-475a-be4a-bec358d38796 /media/data ext4 nofail,noexec 0 0
```

## 22.3 Device-Namen für Festplatten und andere Datenträger

SATA und SCSI sind die zurzeit üblichen Standards, um einen Computer mit seinen Laufwerken zu verbinden. [Tabelle 22.1](#) fasst die Bedeutung dieser und einiger weiterer Abkürzungen zusammen.

**Kernelinterna**

Linux-intern erfolgt der Zugriff auf interne und externe Festplatten und deren Partitionen, auf CD- und DVD-Laufwerke sowie auf andere Datenträger über Device-Dateien. Das sind besondere Dateien, die als Schnittstelle zwischen Linux und der Hardware dienen.

| Abkürzung | Bedeutung  |
|-----------|--|
| ATA       | Advanced Technology Attachment (Schnittstelle zum Anschluss von Festplatten) |
| ATAPI     | ATA Packet Interface (ATA-Erweiterung für CD- und DVD-Laufwerke)             |
| IDE       | Integrated Device Electronics (alternative Bezeichnung für PATA)             |
| PATA      | Parallel ATA (alte ATA-Schnittstelle mit paralleler Datenübertragung)        |
| SATA      | Serial ATA (neue ATA-Schnittstelle mit serieller Datenübertragung)           |
| SCSI      | Small Computer System Interface (Alternative zu IDE/SATA)                    |

Tabelle 22.1 Glossar

Diese Device-Dateien benötigen Sie nur zu Verwaltungszwecken, d. h., wenn Sie die Partitionierung einer Festplatte ändern oder eine bestimmte Partition in das Dateisystem einbinden möchten. Im normalen Betrieb greifen Sie auf das gesamte Dateisystem über Verzeichnisse zu. Dabei bezeichnet / den Start des Dateisystems. Einzelne Partitionen können darin an beliebigen Orten eingebunden werden – eine zusätzliche Linux-Partition etwa unter dem Namen /data, eine Windows-Partition beispielsweise unter dem Namen /media/win.

Im Kernel ist der SCSI-Treiber für alle internen und externen Datenträger verantwortlich, die über die Bussysteme IDE, SATA, SCSI, USB oder Firewire angeschlossen sind. Die Treiberbezeichnung »SCSI« ist historisch begründet: Ursprünglich war der SCSI-Treiber wirklich nur für SCSI-Geräte gedacht, nach und nach wurden aber immer mehr Bussysteme in diesen Treiber integriert.

Alle Datenträger werden mit /dev/sdXN benannt. Die Speichermedien heißen also der Reihe nach /dev/sda, /dev/sdb etc. (siehe [Tabelle 22.2](#)). Bei SATA-Geräten werden der Reihe nach alle genutzten Kanäle mit einem Buchstaben verbunden. Moderne Mainboards sehen zumeist mindestens sechs oder acht Kanäle vor. Wenn beispielsweise zwei Festplatten an die SATA-Kanäle 1 und 3 angeschlossen sind, erhalten diese die Device-Namen /dev/sda und /dev/sdb. Wenn später eine dritte Festplatte an den Kanal 2 angeschlossen wird, ändert sich der Device-Name der zweiten Festplatte von /dev/sdb in /dev/sdc.

Device-Namen

Bei SCSI-Geräten hängt die Reihenfolge von den ID-Nummern der Geräte ab. Löcher in der ID-Reihenfolge werden nicht berücksichtigt. Drei SCSI-Geräte mit den ID-Nummern 0, 2 und 5 bekommen also die Device-Namen /dev/sda bis /dev/sdc. Ähnlich wie bei SATA-Geräten können sich durch eine spätere Konfigurationsänderung die Device-Namen ändern: Wenn ein viertes Gerät mit der ID-Nummer 3 hinzugefügt wird, bekommt dieses den Namen /dev/sdc. Das Gerät mit der ID-Nummer 5 wird jetzt als /dev/sdd angesprochen. Wenn gleichzeitig Geräte verschiedener Bussysteme

angeschlossen sind, hängt es vom BIOS und von den genutzten PCI-Steckplätzen ab, welches Bussystem zuerst berücksichtigt wird.

Externe USB- und Firewire-Geräte werden wie SCSI-Geräte behandelt. Für die Zuweisung der Buchstaben ist die Reihenfolge entscheidend, in der die Geräte angeschlossen werden. CD- und DVD-Laufwerke bekommen eigene Device-Namen, die je nach Distribution `/dev/scdN` oder `/dev/srN` lauten.

| Device  | Bedeutung               |
|---|-------------------------|
| <code>/dev/sda</code>                             | erste Festplatte        |
| <code>/dev/sdb</code>                             | zweite Festplatte       |
| ...   |                         |
| <code>/dev/scd0</code> oder <code>/dev/sr0</code> | erstes CD/DVD-Laufwerk  |
| <code>/dev/scd1</code> oder <code>/dev/sr1</code> | zweites CD/DVD-Laufwerk |
| ...   |                         |

**Tabelle 22.2** Device-Namen

#### Virtuelle Datenträger (virtio)

Wenn Linux in einer virtuellen Maschine ausgeführt wird und dabei der virtio-Treiber zum Einsatz kommt, spricht der Kernel die virtuellen Festplatten über die Device-Namen `/dev/vda`, `/dev/vdb` etc. an. Der virtio-Treiber ermöglicht eine besonders effiziente Kommunikation zwischen dem Virtualisierungssystem und dem Kernel in der virtuellen Maschine. Die Virtualisierungssysteme KVM und Xen unterstützen virtio standardmäßig.

#### Partitionsnummern (MBR)

Das Nummerierungsschema für Partitionen hängt davon ab, wie die Festplatte aufgeteilt ist. Zurzeit sind zwei Partitionierungsvarianten möglich: die klassische MBR-Methode, bei der sich die Partitionierungstabelle im Master Boot Record (MBR) befindet, und die neueren GUID Partition Tables (GPTs), die vor allem bei sehr großen Festplatten sowie auf EFI-Systemen zum Einsatz kommen.

Bei der MBR-Partitionierung sind die Ziffern 1 bis 4 für primäre oder erweiterte Partitionen reserviert und die Ziffern ab 5 für logische Partitionen innerhalb der erweiterten Partitionen. Aus diesem Grund kommt es recht häufig vor, dass es in der Nummerierung Löcher gibt. Wenn die Festplatte beispielsweise eine primäre, eine erweiterte und drei logische Partitionen aufweist, haben diese die Nummern 1, 2, 5, 6 und 7. Tabelle 22.3 gibt einige Beispiele.

Die Anzahl der Partitionen pro Datenträger ist limitiert. Meinen Tests zufolge können Sie auf Festplatten/SSDs mit MBR-Partitionierung bis zu 64 Partitionen einrichten. Mit einer GPT steigt die Anzahl auf 128:

<https://kofler.info/wie-viele-partitionen-pro-festplattessd-unterstuetzt-linux>

| Device     | Bedeutung   |
|------------|---|
| /dev/sda   | die erste SCSI/SATA-Platte                                |
| /dev/sda1  | die erste primäre Partition dieser Festplatte             |
| /dev/sdd3  | die dritte primäre Partition der vierten SCSI/SATA-Platte |
| /dev/sdd5  | die erste logische Partition der vierten SCSI/SATA-Platte |
| /dev/sdd15 | die elfte logische Partition der vierten SCSI/SATA-Platte |

**Tabelle 22.3** Beispiele für die Partitionsnummerierung (MBR)

Wesentlich einfacher ist die Nummerierung bei Festplatten mit einer GPT: Die Unterscheidung zwischen primären, erweiterten und logischen Partitionen entfällt. Die Partitionen werden einfach der Reihe nach durchnummeriert. Das Kernel-Limit von 15 Partitionen bleibt aber bestehen, obwohl eine GPT bis zu 128 Partitionen erlaubt.

Partitions-  
nummern  
(GPT)

#### Unkonventionelle Nummerierung

Es ist möglich, dass die physikalische Reihenfolge der Partitionen von der Nummerierung abweicht! Nehmen Sie an, auf einer Festplatte mit 3 TiB wurden drei Partitionen mit je 1 TiB angelegt (/dev/sda1 bis /dev/sda3). Anschließend wird die mittlere Partition gelöscht. Im freien Bereich werden nun zwei neue Partitionen mit je 500 GiB erzeugt. Diese beiden Partitionen erhalten die Device-Namen /dev/sda2 und /dev/sda4! Bei einer MBR-Partitionierung ist dieser Sonderfall nicht möglich, weil zwischen /dev/sda1 und /dev/sda3 nur *eine* Partition eingefügt werden kann.

Ein ungemein praktisches und wertvolles Werkzeug auf Rechnern mit mehreren Datenträgern ist `lsblk`. Es liefert eine baumartige Liste mit den Device-Namen aller Datenträger und Partitionen inklusive deren Verwendung bzw. `mount`-Punkt. Beim folgenden Beispiel enthält das erste Laufwerk die System- und die Swap-Partition. Auf dem zweiten Laufwerk ist die erste Partition (`sdb1`) ungenutzt. In der zweiten Partition befindet sich ein LVM-System, das wiederum zwei aktive Logical Volumes enthält.

lsblk

```
root# lsblk
NAME                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                  8:0    0    16G  0 disk
  sda1                8:1    0    15G  0 part /
  sda2                8:2    0     1K  0 part
  sda5                8:5    0   1022M  0 part [SWAP]
sdb                  8:16   0     64G  0 disk
  sdb1                8:17   0      2G  0 part
  sdb2                8:18   0     62G  0 part
```

```

vg1-lv1 (dm-0) 252:0    0   10G 0 lvm /backup
vg1-lv2 (dm-1) 252:1    0   15G 0 lvm /data
sr0              11:0     1 1024M 0 rom

```

#### Alternative Device-Namen

Wie ich oben erklärt habe, kann sich durch den nachträglichen Einbau weiterer SCSI- oder SATA-Festplatten der Device-Name der bisherigen Geräte ändern. Unvorhersehbar sind die Device-Namen bei externen Geräten: Sie ergeben sich aus der Reihenfolge, in der die Geräte angeschlossen werden.

Damit Sie trotz variierender Device-Namen einzelne Geräte bzw. Partitionen einheitlich ansprechen können (z. B. in einem Backup-Script), enthält das Verzeichnis `/dev/disk` zusätzliche Links auf alle Datenträger, die nach verschiedenen Kriterien geordnet sind:

- ▶ `/dev/disk/by-id/nnn` verwendet IDs, die sich aus dem Bussystem, dem Geräte-namen und einer Modell- oder Seriennummer zusammensetzen.
- ▶ `/dev/disk/by-label/label` verwendet den Namen, der dem Dateisystem gegeben wurde.
- ▶ `/dev/disk/by-path/xxx` verwendet einen Pfadnamen, der sich aus der PCI-Schnittstelle, dem Bussystem und der Partitionsnummer ergibt. Vorsicht: Wenn ein USB- oder Firewire-Gerät beim nächsten Mal an einen anderen USB-Stecker angeschlossen wird, ändert sich sein Pfadname!
- ▶ `/dev/disk/by-uuid/nnn` verwendet die UUIDs der Dateisysteme. Universal Unique Identifiers sind eindeutige ID-Nummern, die einem Dateisystem beim Formatieren zugeordnet werden. UUIDs ermöglichen eine Identifizierung von Dateisystemen auch nach einer Änderung an der Hardware-Konfiguration.

Die Anzahl der Links in den `/dev/disk`-Verzeichnissen variiert. `/dev/disk/by-label` und `by-uuid` enthalten beispielsweise nur Links auf Partitionen, die benannt sind bzw. eine UUID haben. Für die automatische Erzeugung der Links ist das `udev`-System verantwortlich (siehe [Abschnitt 11.9](#), »Device-Dateien«). Das folgende `ls`-Kommando zeigt ein Beispiel für die Links eines Testsystems mit einer SATA-Festplatte, einem USB-Stick und einer SD-Karte. Um die Lesbarkeit zu erhöhen, habe ich die Zeilen ein wenig eingerückt und die Informationen zu den Zugriffsrechten entfernt.

```

user$ cd /dev/
user$ ls -lR disk/
disk/by-id:
  scsi-SATA_ST3320620AS_5QF194H9          -> ../../sda
  scsi-SATA_ST3320620AS_5QF194H9-part1    -> ../../sda1
  scsi-SATA_ST3320620AS_5QF194H9-part2    -> ../../sda2
  usb-Generic_USB_CF_Reader_058F312D81B   -> ../../sdc
  ...

```

```

disk/by-path:
pci-0000:00:1d.7-usb-0:5:1.0-scsi-0:0:0:1    -> ../../sdc
pci-0000:00:1f.2-scsi-0:0:0:0              -> ../../sda
pci-0000:00:1f.2-scsi-0:0:0:0-part1       -> ../../sda1
pci-0000:00:1f.2-scsi-0:0:0:0-part2       -> ../../sda2

disk/by-uuid:
008f06ef-28be-45c9-acbc-20cda51f712b     -> ../../sda2
366CA8D16CA88D65                          -> ../../sda1

```

## 22.4 Partitionierung der Festplatte oder SSD

An sich ist die Partitionierung einer Festplatte oder SSD nichts Geheimnisvolles. Das Problem besteht aber darin, dass die Grundlagen der Partitionierung zurück in die 80er-Jahre reichen, als eine Festplatte mit 50 MiB *riesig* war und sich noch niemand eine Solid State Disk vorstellen konnte. In den vergangenen drei Jahrzehnten hat die Datenträgertechnologie fast unvorstellbare Fortschritte gemacht, während sich an den Partitionierungsgrundlagen nur ganz zaghafte etwas geändert hat.

Einen wirklich modernen Standard zur Speicherung der Partitionierungsdaten stellen erst GUID Partition Tables (GPTs) dar, die bei internen Festplatten und SSDs inzwischen weit verbreitet sind. Externe Datenträger und insbesondere USB-Sticks und SD-Karten verwenden aber zumeist immer noch die aus DOS-Zeiten stammende MBR-Partitionierung.

Kurzum: Damit Sie bei der Partitionierung Ihrer Festplatte oder SSD keine Fehler machen, müssen Sie Grundlagen und Prinzipien verstehen, die weit in die Vergangenheit zurückreichen. Aus diesem Grund fällt dieser Abschnitt recht umfangreich aus. Einleitende Informationen zu Partitionen und Partitionstypen finden Sie in [Abschnitt 2.6](#), »Grundlagen der Festplattenpartitionierung«. Wie Partitionen unter Linux benannt und nummeriert werden, habe ich im vorigen Abschnitt beschrieben.

### Achtung

Partitionierungsprogramme können den Inhalt Ihrer gesamten Festplatte bzw. SSD zerstören! Lesen Sie diesen Abschnitt vollständig, bevor Sie Partitionierungswerkzeuge einsetzen! Sie dürfen nie eine Partition verändern, die momentan verwendet wird, d. h., deren Dateisystem in den Verzeichnisbaum eingebunden bzw. »gemountet« ist!

Während der Installation bietet beinahe jede Linux-Distribution einfach zu bedienende Werkzeuge zur Partitionierung der Festplatte an. Aber nur bei wenigen Distributionen stehen diese Werkzeuge auch im laufenden Betrieb zur Verfügung – z. B. bei

Partitionierungs-  
werkzeuge

SUSE das YaST-Modul `SYSTEM · PARTITIONIEREN`. Ansonsten haben Sie, wenn Sie nach der Installation Änderungen an der Partitionierung durchführen möchten, die Wahl zwischen einer ganzen Palette von Partitionierwerkzeugen: Zu den wichtigsten Vertretern zählen die textorientierten Kommandos `fdisk` und `parted` sowie das grafische Programm `gparted`.

**LVM** Wenn Sie häufig Änderungen an der Partitionierung durchführen müssen, sollten Sie sich unbedingt mit LVM anfreunden (siehe [Abschnitt 22.18](#)): LVM fügt eine virtuelle Ebene zwischen den physikalischen Partitionen der Festplatte und den für Dateisysteme genutzten Partitionen ein und vereinfacht nachträgliche Änderungen ungemein.

**Auf Partitionen verzichten** Grundsätzlich ist es möglich, auf Partitionen ganz zu verzichten und ein Dateisystem direkt auf dem Datenträger einzurichten. Auf echter Hardware ist das eher unüblich, zumal sich daraus keine Vorteile ergeben. Bei Cloud-Installationen kann es aber durchaus sinnvoll sein, das Dateisystem direkt auf dem Datenträger einzurichten. Im Fall einer späteren Vergrößerung des virtuellen Datenträgers können Sie dann auch das Dateisystem unkompliziert mit `resize2fs` oder `xfs_growfs` vergrößern, ohne umständlich mit Partitionen hantieren zu müssen.

### MBR oder GPT?

Es gibt zwei Verfahren, wie die Partitionierungsdaten gespeichert werden können: im Master Boot Record (MBR) oder als GUID Partition Table (GPT). Das MBR-Verfahren ist seit Jahrzehnten gebräuchlich, aber nur für Datenträger bis zu 2 TiB geeignet. Zwingend ist der Einsatz einer GPT unter Linux in zwei Fällen erforderlich:

- ▶ bei Festplatten oder SSDs, die größer als 2 TiB sind
- ▶ bei Festplatten oder SSDs, die parallel auch zum Start von Windows oder macOS verwendet werden

Auch wenn eine GPT für Linux optional ist, gibt es keinen vernünftigen Grund, heute noch MBR zu verwenden (außer bei USB-Sticks oder SD-Karten). Ich selbst richte auf neuen Festplatten und SSDs generell eine GPT ein, weil ich das Gewurstel mit primären, erweiterten und logischen Partitionen leid bin.

#### Das Partitionierungssystem kann später nur noch schwer geändert werden!

Die Entscheidung für ein Partitionierungssystem ist endgültig. Ein späterer Wechsel ist zwar jederzeit möglich, geht aber mit dem Verlust aller Daten einher!

**Festplatte/SSD initialisieren** Aktuelle Linux-Distributionen kommen sowohl mit MBR- als auch mit GPT-Datenträgern zurecht. Wenn es aber darum geht, die Partitionierung neu einzurichten, fehlt bei vielen Installationsprogrammen eine Wahlmöglichkeit zwischen MBR und GPT. Auf Datenträgern, die kleiner als zwei TiB sind, wird oft ungefragt MBR verwendet.



Wenn Sie den Partitionierungstyp also selbst bestimmen möchten, ist Handarbeit erforderlich, die Sie am besten in einem Live-System durchführen. Dort starten Sie in einem Terminalfenster mit `root`-Rechten das Programm `parted` und führen das entsprechende `mklabel`-Kommando aus. (Um es ein letztes Mal zu betonen: Mit `mklabel xxx` verlieren Sie alle Daten auf Ihrer Festplatte!)

```
root# parted /dev/sda
(parted) mklabel gpt      (für GPT)
(parted) mklabel msdos   (für MBR)
(parted) quit
```

## Grundregeln

Unabhängig vom eingesetzten Werkzeug müssen Sie einige Grundregeln beachten:

- ▶ Es ist unmöglich, im laufenden Betrieb Änderungen an der Systempartition durchzuführen. Wenn Sie beispielsweise die Systempartition vergrößern möchten, starten Sie den Rechner am besten mit einer Live-CD. Besonders gut geeignet sind die für die Festplattenpartitionierung optimierten Minidistributionen GParted-Live-CD, Parted Magic und SystemRescueCd:

<http://gparted.sourceforge.net/livecd.php>

<https://partedmagic.com>

<http://sysresccd.org>

- ▶ Eine Vergrößerung einer Partition ist grundsätzlich nur möglich, wenn hinter der Partition freier Platz ist. Sie können Partitionen nicht »verschieben«.
- ▶ Wenn Sie die Größe einer Partition ändern, verändert sich damit *nicht* automatisch auch die Größe des darauf enthaltenen Dateisystems! Dazu sind weitere Kommandos erforderlich, die je nach Dateisystemtyp variieren.
- ▶ Linux kann grundsätzlich nur 15 Partitionen pro Festplatte ansprechen. Wenn Ihnen das zu wenig ist, müssen Sie LVM einsetzen.

Bei der MBR-Partitionierung ist eine dieser 15 Partitionen die erweiterte Partition, die zwar andere Partitionen, aber nicht direkt Daten bzw. ein Dateisystem aufnehmen kann. Damit reduziert sich die Maximalanzahl der für Dateisysteme geeigneten Partitionen auf 3 primäre und 11 logische Partitionen.

- ▶ Wenn Sie die Partitionierung einer Festplatte ändern, die momentan genutzt wird (z. B. weil eine Partition die Systempartition Ihrer Linux-Distribution ist), fordert das Partitionierungsprogramm Sie unter Umständen dazu auf, den Rechner neu zu starten. Das gilt insbesondere, wenn Sie eine Distribution mit einer alten Kernel-Version verwenden. Bei `fdisk` lautet die Warnung so: *Re-read table failed with error 16: Device or resource busy. Reboot your system to ensure the partition table is updated.*

Der Grund besteht darin, dass ältere Linux-Kernel nicht in der Lage sind, im laufenden Betrieb die Partitionierungstabelle neu einzulesen. Die Änderungen wurden also gespeichert, werden aber für den Kernel erst nach einem Neustart aktiv. Sie *müssen* Linux daher neu starten, bevor Sie die geänderte Partitionierung nutzen können.

### Festplatten und SSDs mit 4-KiB-Sektoren

Handelsübliche Festplatten sowie SSDs verwenden statt der jahrzehntlang üblichen 512-Byte-Sektoren längere Sektoren von 4096 Byte (4 KiB). Das hat viele Vorteile, unter anderem eine höhere Geschwindigkeit und eine höhere Festplattenkapazität. Aus Kompatibilitätsgründen melden aber auch solche Datenträger eine 512-Byte-Sektorgröße an das Betriebssystem.

Um Festplatten mit 4-KiB-Sektoren effizient nutzen zu können, müssen Partitionen so eingerichtet werden, dass die Startposition jeder Partition ein Vielfaches von 4 KiB beträgt. Ist das nicht der Fall und will das Betriebssystem einen 4-KiB-Bereich verändern, muss die Festplatte zwei 4-KiB-Sektoren lesen, modifizieren und schreiben. Das würde Schreibvorgänge massiv bremsen.

Aktuelle Windows- und Linux-Versionen nehmen bei der Installation Rücksicht auf die 4-KiB-Sektorgröße und richten die Partitions Grenzen bei der Installation an Vielfachen von 1 MiB aus.

Wenn Sie selbst Partitionen einrichten und Programme verwenden, die mit 512-Byte-Sektoren rechnen (z. B. alte `fdisk`-Versionen), müssen Sie darauf achten, dass die Partitions Grenzen ein Vielfaches von 8 Sektoren betragen!

### Größenanpassung der erweiterten Partition (nur bei der MBR-Partitionierung)

Die in diesem Kapitel vorgestellten Partitionierungswerkzeuge, aber auch die während einer Linux-Installation eingesetzten Partitionierungshilfen unterscheiden sich in einem entscheidenden Punkt: Manche Werkzeuge belassen die erweiterte Partition so, wie Sie ursprünglich eingerichtet haben. Ich bezeichne diese Programme hier als Typ 1. Dazu zählen `fdisk`, `parted`, die Installationsprogramme von Fedora, Red Hat und SUSE sowie die Partitionierungswerkzeuge von Windows.

Andere Programme passen die Größe der erweiterten Partition dagegen automatisch so an, dass alle logischen Partitionen exakt darin Platz haben (Typ 2). Dazu zählen die Installationsprogramme von Debian und Ubuntu.

Beide Vorgehensweisen sind an sich in Ordnung, problematisch ist nur der Mischbetrieb: Nehmen wir an, Sie erzeugen mit einem Werkzeug des Typs 2 eine neue, logische Partition. Bei dieser Gelegenheit verkleinert das Programm die erweiterte Partition. Wenn Sie anschließend mit einem Werkzeug des Typs 1 versuchen, eine weitere logische Partition anzulegen, meldet dieses, dass die erweiterte Partition schon voll ist. Programme des Typs 1 sind mit der Ausnahme von `parted` nicht in der Lage, die Größe erweiterter Partitionen zu ändern, wenn sich darin bereits logische Partitionen befinden.

Abhilfe: Starten Sie nochmals ein Programm des Typs 2, und ändern Sie die Partitionierung. Sie können auch mit einem Typ-2-Partitionierwerkzeug eine ausreichend große logische Platzhalterpartition erzeugen. Damit vergrößert sich automatisch die erweiterte Partition. Anschließend löschen Sie die Platzhalterpartition mit einem Partitionierungsprogramm des Typs 1 und können nun den freien Platz in der erweiterten Partition nutzen.

Naturgemäß gehen Sie dem Ärger mit erweiterten Partitionen ganz aus dem Weg, wenn Sie sich von vornherein für die GPT-Partitionierung entscheiden.

## 22.5 parted-Kommando

`parted` ist das wichtigste Partitionierungswerkzeug für Linux auf Kommandoebene. Es kommt sowohl mit MBR- als auch mit GPT-Partitionen zurecht. Details zu den vielen Funktionen von `parted` finden Sie unter:

<https://gnu.org/software/parted>

### fdisk – ein Relikt aus der Vergangenheit

Als Alternative zu `parted` ist bis heute auch `fdisk` beliebt. Da sich `fdisk` aber nur für Datenträger mit MBR-Partitionen eignet, gehe ich in diesem Buch auf `fdisk` nicht mehr ein.

### Bearbeiten Sie mit `parted` nur die Partitionen, nicht deren Inhalt!

`parted` kennt diverse Kommandos, die nicht nur die Partitionen bearbeiten, sondern auch das darauf enthaltene Dateisystem. Das Handbuch rät von der Benutzung dieser Kommandos ab, und in zukünftigen `parted`-Versionen sollen diese Kommandos ganz eliminiert werden. Verwenden Sie `parted` *nur* zur Partitionierung! Um Dateisysteme einzurichten oder zu verändern, verwenden Sie besser die dafür vorgesehenen Kommandos außerhalb von `parted`, z. B. `mkfs.ext4` oder `resize2fs`.

**Start** Beim Start von parted geben Sie das Festplatten-Device an. `help` oder auch kurz `h` führt zur Anzeige der zur Auswahl stehenden Kommandos. `h` Kommando liefert einen knappen Hilfetext zu den einzelnen Kommandos. `print` zeigt die Partitionstabelle an – hier für eine 1,5-TiB-Festplatte, die bereits drei Partitionen enthält:

```
root# parted /dev/sda
(parted) print
Modell: ATA WDC WD15EARS-00Z (scsi)
Festplatte /dev/sda: 1500GB

Sektorgröße (logisch/physisch): 512B/512B
Partitionstabelle: msdos

Anzahl  Beginn  Ende    Größe  Typ    Dateisystem  Flags
1       1049kB  50,0GB  50,0GB  primary ext4          boot
2       50,0GB  52,0GB  2000MB  primary linux-swap(v1)
3       52,0GB  84,2GB  32,2GB  primary
```

**MB versus MiB, GB versus GiB** Standardmäßig zeigt parted Partitionsgrenzen und -größen in dezimalen Maßeinheiten an, also in MB =  $10^6$  Byte oder GB =  $10^9$  Byte. Eingaben ohne Maßeinheit werden als dezimale MByte interpretiert. Sie können auch explizit die gewünschte Einheit angeben – beispielsweise 100MiB (binäre MByte) oder 15GB (dezimale GByte).

Mit `unit` können Sie die gewünschte Maßeinheit für alle Ein- und Ausgaben festlegen. Mögliche Einstellungen sind MB und GB, MiB und GiB (binäre Zählweise, also GiB =  $2^{30}$  Byte) sowie % für Prozentangaben relativ zur Größe der Festplatte.

```
(parted) unit MiB
(parted) print
...
Festplatte /dev/sda: 1397GiB
...
Anzahl  Beginn  Ende    Größe  Typ    Dateisystem  Flags
1       1,00MiB  47684MiB  47683MiB  primary ext4          boot
2       47684MiB  49591MiB  1907MiB  primary linux-swap(v1)
3       49591MiB  80311MiB  30720MiB  primary
```

**mklabel** Wenn die Festplatte noch vollkommen ungenutzt ist und keine Partitionstabelle enthält, müssen Sie diese einrichten. Dazu führen Sie `mklabel msdos` für MBR-Partitionen oder `mklabel gpt` für GPT-Partitionen aus. Beachten Sie, dass Sie mit `mklabel` alle bisher vielleicht gespeicherten Daten der Festplatte verlieren!

**mkpart und rm** Mit den Kommandos `mkpart` bzw. `rm` erzeugen bzw. löschen Sie Partitionen. Bei `mkpart` müssen Sie bei MBR-Partitionen den Typ (`primary`, `extended` oder `logical`) sowie die gewünschte Start- und Endposition angeben. Bei GPT-Partitionen wird der erste Parameter als frei wählbarer Partitionsname interpretiert.

```
(parted) mkpart primary 1MiB 10GiB (MBR)
(parted) mkpart part1 1MiB 10GiB (GPT)
```

### Arbeiten ohne Undo-Funktion

Im Gegensatz zu `fdisk`, wo alle durchzuführenden Änderungen nur vorgemerkt und erst mit dem `Write`-Kommando tatsächlich ausgeführt werden, führt `parted` jedes Partitionierkommando sofort aus!

Achten Sie immer darauf, dass Sie die richtige Festplatte bearbeiten. Wenn Sie beim Start von `parted` kein Device angeben, verwendet `parted` automatisch `/dev/sda`. Das ist möglicherweise nicht beabsichtigt.

Ärgerlicherweise macht es `parted` Nutzern extrem schwer, aneinanderliegende Partitionen zu definieren: Das folgende Kommando zeigt den Versuch, der vorhin angelegten ersten Partition eine zweite hinzuzufügen, wobei die zweite Partition dort beginnen soll, wo die erste Partition endet:

```
(parted) mkpart primary 10GiB 15GiB (MBR)
(parted) mkpart part2 10GiB 15GiB (GPT)
Warnung: Sie wollen eine Partition von 10,7GB bis 16,1GB (Sektoren
20971520..31457280). Das Beste, was Parted bieten kann, ist von
10,7GB nach 16,1GB (Sektoren 20971521..31457280).
Ist dies noch akzeptabel für Sie? Ja/Yes/Nein/No? No
```

`parted` quittiert das Kommando mit der Warnung, dass der Start der zweiten Partition um einen Sektor (in der Regel 512 Byte) verschoben werden muss, und fragt, ob das für Sie akzeptabel ist. Das ist es eben gerade nicht, weil Partitionen auf modernen Festplatten und SSDs unbedingt an 4-KiB-Grenzen ausgerichtet sein sollen.

Nachdem Sie das Erzeugen der neuen Partition also abgelehnt haben, versuchen Sie es neuerlich, wobei Sie den Startpunkt diesmal aber in der Einheit `MiB` angeben und dabei zur vorherigen Partition eine Lücke von einem `MiB` frei lassen. Die Startposition `10241` ergibt sich aus der Rechnung  $1024 \times 10 + 1$ , weil jedes `GiB` aus `1024 MiB` besteht.

```
(parted) mkpart primary 10241MiB 15GiB (MBR)
(parted) mkpart part2 10241MiB 15GiB (GPT)
```

Je größer der Datenträger ist, desto mühsamer wird das korrekte Berechnen der Startpunkte. Das können Sie sich ersparen, wenn Sie `MiB` als Einheit vorgeben und mit `print` die aktuelle Partitionstabelle ansehen. Die Spalte `Ende` gibt an, wo die jeweilige Partition endet. Zu diesem Wert addieren Sie `1` – und schon haben Sie den Startpunkt für die nächste Partition. Der Nachteil dieser Einstellung offenbart sich allerdings bei großen Datenträgern mit sechsstelligen und entsprechend unübersichtlichen `MiB`-Werten.

```
(parted) unit MiB
(parted) print
...
Nummer  Anfang  Ende      Größe   Typ      Dateisystem  Flags
1       1,00MiB  10240MiB  10239MiB primary   LBA
2       10241MiB 15360MiB  5119MiB  primary   LBA
```

**Partitionstypen** Normalerweise erzeugt `mkpart` Partitionen, die später ein Linux-Dateisystem aufnehmen. Dieser Partitionstyp ist auch für Software-RAID und LVM geeignet. Wenn Sie hingegen eine Swap- oder eine Windows-Partition erzeugen möchten, müssen Sie in einem zusätzlichen Parameter vor der Startposition den Dateisystemtyp angeben, z. B. `linux-swaps`, `fat32` oder `ntfs`. Weitere mögliche Dateisystemtypen liefert `help mkpart`.

```
(parted) mkpart primary linux-swaps 1MiB 10GiB (MBR)
(parted) mkpart part1 linux-swaps 1MiB 10GiB (GPT)
```

**Erweiterte und logische Partitionen** Bevor Sie eine logische Partition auf einem MBR-Datenträger erzeugen können, müssen Sie eine erweiterte Partition anlegen (`mkpart extended start ende`). Gegebenenfalls können Sie eine schon vorhandene erweiterte Partition mit `resize` vergrößern, sofern am Ende der erweiterten Partition noch freier Platz auf dem Datenträger ist. Das Kommando `resize` kann aber nicht für andere Partitionsarten verwendet werden.

**Partitionen löschen** Um eine Partition zu löschen, ermitteln Sie zuerst mit `print` die Nummern aller Partitionen. Anschließend löschen Sie mit `rm N` die gewünschte Partition, wobei `N` eine Partitionsnummer laut `print` ist.

Beim Löschen einer Partition gehen vorerst noch keine Daten verloren. Deswegen ist es prinzipiell möglich, diesen Schritt rückgängig zu machen, indem Sie die Partition mit exakt demselben Start- und Endpunkt neuerlich erzeugen. Das Problem dabei ist allerdings die Positionsangabe – diese muss auf den Sektor genau stimmen.

**set** Wenn Sie die neue Partition als Teil eines LVM- oder RAID-Systems nutzen möchten, müssen Sie Zusatzattribute (Flags) entsprechend einstellen. Das erforderliche Kommando lautet `set partitionsnummer attributname on/off`. Mögliche Attribute sind unter anderem `boot`, `lvm`, `raid` und `bios_grub`.

```
(parted) set 3 lvm on
```

**EFI-Partition** Um mit `parted` eine EFI-Partition auf einem GPT-Datenträger einzurichten, verwenden Sie als Partitionstyp `fat32` und die Flags `boot` und `esp`:

```
(parted) mkpart part1 fat32 1mib 100mib
(parted) set 1 boot on
(parted) set 1 esp on
```

Wenn Sie eine Partition irrtümlich gelöscht haben, können Sie mit `rescue` versuchen, diese Partition wiederherzustellen. Dabei müssen Sie den ungefähren Start- und Endpunkt der Partition angeben. `parted` durchsucht den Datenträger dann nach bekannten Dateisystemen. Diese Suche gelingt umso schneller, je genauer Start- und Endpunkt sind. Wenn `parted` fündig wird, bietet es an, diese Partition wieder in die Partitionstabelle einzutragen:

Gelöschte  
Partitionen  
wiederherstellen

```
root#
(parted) rescue 1mib 18gib
Suche nach Dateisystemen...
Informationen: Eine ext4 primary-Partition wurde bei
1049kB -> 17,2GB gefunden. Wollen Sie diese Partition
zur Partitionstabelle hinzufügen?
Ja/Yes/Nein/No/Abbrechen/Cancel? ja
Fehler: Der/die Partition(en) auf /dev/sda wurden geschrieben,
aber es war nicht möglich, den Kernel über die Änderungen
zu informieren, weil sie wahrscheinlich in Benutzung sind.
Daher werden die alten Partitionen noch benutzt. Sie sollten
nun den Rechner neu starten, bevor Sie weitere Änderungen
vornehmen.
Ignorieren/Ignore/Abbrechen/Cancel? ignorieren
```

Erwarten Sie keine Wunder vom Rescue-Kommando. Es kann nur dann zum Erfolg führen, wenn das Dateisystem unbeschädigt ist und nicht z. B. durch andere, überlappend platzierte Dateisysteme bereits teilweise überschrieben wurde.

### Beispiel 1 (MBR)

Die folgenden Kommandos richten zuerst eine primäre Partition (500 MiB), dann eine erweiterte Partition (19,5 GiB) und darin eine logische Partition in derselben Größe ein. Beachten Sie, dass der Startpunkt für die logische Partition um 1 MiB größer ist als der Startpunkt der erweiterten Partition. Dadurch entsteht zwar eine kleine, ungenutzte Lücke, dafür sind die Partitionen aber optimal ausgerichtet.

```
root# parted /dev/sdb
(parted) mklabel msdos
(parted) mkpart primary 1mib 500mib
(parted) mkpart extended 500mib 20gib
(parted) mkpart logical 501mib 20gib
(parted) unit GiB
(parted) print
```

| Nummer | Anfang  | Ende    | Größe   | Typ      | Dateisystem | Flags |
|--------|---------|---------|---------|----------|-------------|-------|
| 1      | 0,00GiB | 0,49GiB | 0,49GiB | primary  |             |       |
| 2      | 0,49GiB | 20,0GiB | 19,5GiB | extended |             | LBA   |
| 5      | 0,49GiB | 20,0GiB | 19,5GiB | logical  |             |       |

Nun soll auf der Festplatte eine weitere logische Swap-Partition eingerichtet werden. Dazu muss zuerst die erweiterte Partition vergrößert werden. `resize` zeigt dabei eine vollkommen irreführende Warnung an – eine erweiterte Partition enthält ja ohnedies nie ein Dateisystem:

```
(parted) resize 2 0,49GiB 21GiB
WARNUNG: Sie versuchen parted auf einem Dateisystem (resize) zu verwenden ...
(parted) mkpart logical linux-swap 20GiB 21GiB
(parted) unit GiB
(parted) print
```

| Nummer | Anfang  | Ende    | Größe   | Typ      | Dateisystem | Flags |
|--------|---------|---------|---------|----------|-------------|-------|
| 1      | 0,00GiB | 0,49GiB | 0,49GiB | primary  |             |       |
| 2      | 0,49GiB | 21,0GiB | 20,5GiB | extended |             | LBA   |
| 5      | 0,49GiB | 20,0GiB | 19,5GiB | logical  |             |       |
| 6      | 20,0GiB | 21,0GiB | 1,00GiB | logical  |             |       |

### Beispiel 2 (GPT)

In einem zweiten Beispiel sollen auf einer Festplatte mit GPT zwei Partitionen eingerichtet werden, von denen eine später als Physical Volume für ein LVM-System dienen wird:

```
root#
(parted) mklabel gpt
(parted) mkpart part1 1mib 2000mib
(parted) mkpart part2 2001mib 64000mib
(parted) set 2 lvm on
(parted) print
```

| Nummer | Anfang  | Ende    | Größe   | Dateisystem | Name  | Flags |
|--------|---------|---------|---------|-------------|-------|-------|
| 1      | 0,00GiB | 2,00GiB | 2,00GiB |             | part1 |       |
| 2      | 2,00GiB | 64,0GiB | 62,0GiB |             | part2 | LVM   |

## 22.6 Partitionierungswerkzeuge mit grafischer Benutzeroberfläche

`parted` ist zugegebenermaßen kein Kommando, das viel Komfort bietet. Deswegen stelle ich Ihnen in diesem Abschnitt einige alternative Werkzeuge vor, die allesamt über eine grafische Benutzeroberfläche verfügen. Ein vollständiger Ersatz für `parted` sind diese Programme aber nicht – denn wenn eine manuelle Partitionierung erforderlich ist, steht oft kein Grafiksystem zur Verfügung!

**GParted (MBR, GPT)** GParted ist eine grafische Benutzeroberfläche zu `parted` (siehe [Abbildung 22.1](#)). GParted kann allerdings nur Partitionen verändern, die momentan unbenutzt sind, die also nicht in den Verzeichnisbaum eingebunden sind.



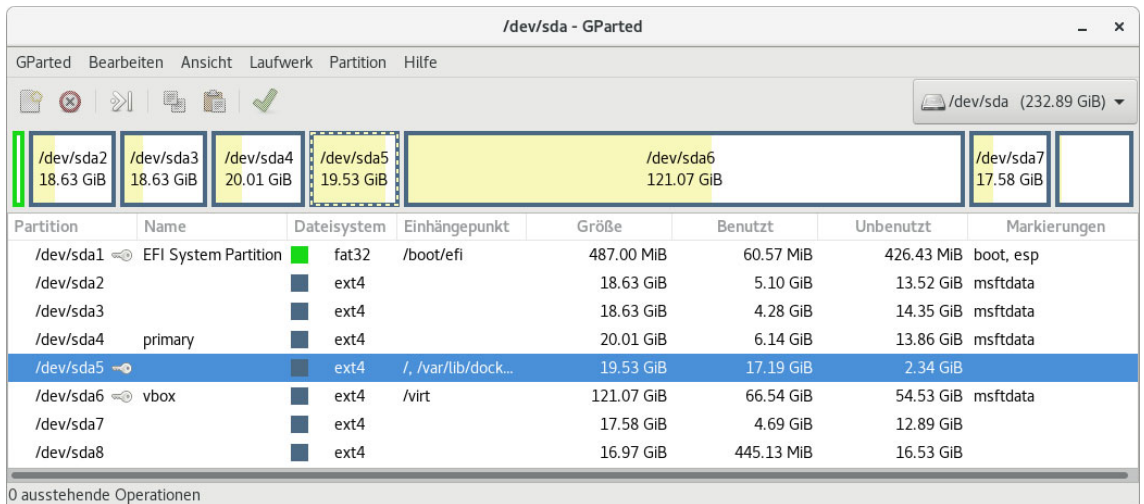


Abbildung 22.1 GParted

Alle benutzten Partitionen werden im Programm mit einem Vorhängeschloss gekennzeichnet; für diese Partitionen sind die Bearbeitungsbuttons gesperrt. Abhilfe: Starten Sie GParted von einer Live-CD oder von einem entsprechenden Image auf einem USB-Stick.

Anders als parted merkt sich GParted alle Aktionen, führt diese aber vorerst nicht aus. BEARBEITEN • RÜCKGÄNGIG widerruft die Aktionen, BEARBEITEN • AUSFÜHREN führt sie endgültig aus.

Um eine Partition zur Verwendung für RAID oder LVM zu kennzeichnen, klicken Sie die Partition mit der rechten Maustaste an und führen MARKIERUNGEN BEARBEITEN aus. Anschließend können Sie diverse Flags setzen, die die Funktion der Partition bezeichnen.

Unter SUSE können Sie als Alternative zu GParted auch das YaST-Modul SYSTEM • PARTITIONIEREN einsetzen. Bei den meisten anderen Distributionen fehlt ein vergleichbares Werkzeug.

Das Gnome-Programm *Laufwerke* (Kommando `gnome-disks`, siehe [Abbildung 22.2](#)) hilft bei der Partitionierung von Festplatten und SSDs, beim Einrichten von Dateisystemen und beim Einbinden von Dateisystemen in den Verzeichnisbaum (inklusive der Veränderung von `/etc/fstab`). Bei einigen Distributionen befindet sich das Programm im Paket `gnome-disk-utils` und muss extra installiert werden.

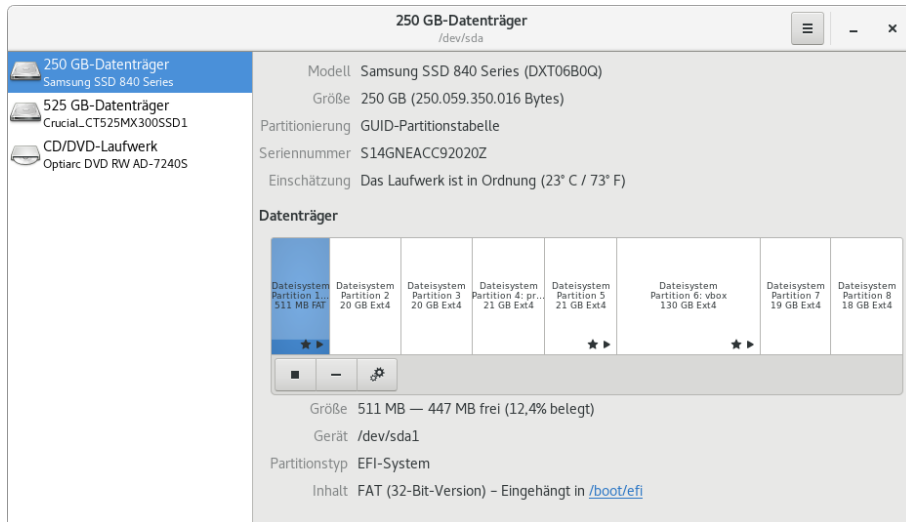


Abbildung 22.2 Das Gnome-Programm »Laufwerke«

Ärgerlich beim Umgang mit dem Programm ist der Umstand, dass `gnome-disks` darauf besteht, auf jeder neuen Partition sofort ein Dateisystem einzurichten. Wenn Sie das nicht möchten, wählen Sie `TYP = BENUTZERDEFINIERT` und geben im Textfeld `DATEISYSTEM` eine ungültige Bezeichnung an (z. B. `none`). Das führt zwar zu einer Fehlermeldung, die Partition wird aber ohne Dateisystem erzeugt. Anschließend öffnen Sie mit dem Zahnrad-Button ein Kontextmenü und wählen `PARTITION BEARBEITEN`. An dieser Stelle haben Sie nun die Möglichkeit, die Partition zur Nutzung für RAID, LVM etc. zu markieren.

## 22.7 Dateisystemtypen

Dieser Abschnitt gibt einen Überblick über die Dateisystemtypen, die unter Linux genutzt werden können. Auf einige besonders wichtige Dateisystemtypen gehe ich im weiteren Verlauf dieses Kapitels dann detaillierter ein: `ext2` bis `ext4`, `btrfs`, `xfs`, `vfat`, `ntfs` und `iso9660`. Welchen bzw. welche Dateisystemtypen Sie zurzeit verwenden, können Sie übrigens ganz leicht mit dem Kommando `df -T` feststellen.

Linux-  
Dateisysteme

»Linux-Dateisysteme« sind zur Installation und zum Betrieb von Linux geeignet. Im Alltagsbetrieb werden Sie gar nicht bemerken, mit welchem der im Folgenden aufgezählten Dateisystemtypen Sie arbeiten. Elementare Kommandos wie `ls` oder `cp`, die Verwaltung der Zugriffsrechte etc. – all das funktioniert unabhängig vom Dateisystem.

Die Dateisysteme unterscheiden sich durch Merkmale, die überwiegend für fortgeschrittene Anwender bzw. für den Server-Einsatz interessant sind: Geschwindigkeit beim Umgang mit sehr großen oder mit sehr vielen eher kleinen Dateien, Effizienz bei Schreib- und Lese-Operationen, CPU-Belastung, Journaling-Funktion (Verhalten nach einem Absturz), Quota-Funktion (die Möglichkeit, den maximalen Speicherverbrauch pro Benutzer einzuschränken), Verwaltungs-Overhead, Snapshot-Funktionen, Verschlüsselungs- und Komprimierfunktionen, TRIM-Unterstützung für SSDs etc.

- ▶ **ext:** ext2 (Extended Filesystem, Version 2) war in den Anfangszeiten von Linux das dominierende Linux-Dateisystem. 2002 hat ext3 seine Nachfolge angetreten, Ende 2008 wurde ext4 vorgestellt. Die maximale Dateisystemgröße beträgt nun ein Exabyte (1.048.576 TiB), was für eine Weile reichen sollte ...
- ▶ **btrfs:** Über viele Jahre wurde btrfs als das Linux-Dateisystem der Zukunft gehandelt. Das mit der Unterstützung von Oracle von Grund auf neu entwickelte Dateisystem beinhaltet Device-Mapper-, Snapshot- und RAID-Funktionen und ist am ehesten mit Suns ZFS zu vergleichen. Leider ist btrfs sehr komplex und noch nicht vollständig ausgereift – eine Aussage, die hier nun schon seit mindestens drei Auflagen steht.
- ▶ **xf:** xfs kam ursprünglich als Dateisystem auf den Workstations der Firma SGI unter dem Betriebssystem IRIX zum Einsatz. xfs wird in aktuellen RHEL- und CentOS-Distributionen standardmäßig verwendet. Im Vergleich zu ext4 gilt xfs für sehr große Dateisysteme mit mehr als 16 TiB als besonders ausgereift – ein Argument, das primär bei großen Server-Installationen relevant ist.

#### Welches ist das beste Linux-Dateisystem?

Das »beste« oder »schnellste« Dateisystem gibt es nicht – jede Wertung hängt vom Verwendungszweck ab. Meine Empfehlung geht in Richtung ext4 sowohl für Desktop- als auch für Server-Installationen. ext4 ist ein vergleichsweise simples, dafür aber sehr robustes Dateisystem.

Wenn Sie an Benchmark-Tests interessiert sind, sollten Sie einen Blick auf die Webseite <https://phoronix.com> werfen. Deren Team führt immer wieder umfassende Vergleiche zwischen den verschiedenen Dateisystemen durch.

ZFS gilt momentan als Maßstab, an dem sich alle Dateisysteme messen müssen. ZFS wurde von Sun für Solaris entwickelt und gehört nun Oracle. Da der ZFS-Code nicht GPL-kompatibel ist, kann er nicht in den Linux-Kernel integriert werden. Es ist aber nicht schwierig, den Treiber als binäres Modul oder als Quellcode zu installieren. Unter Ubuntu steht ZFS sogar als offizielles Paket (`zfsutil-linux`) zur Verfügung. Weitere Informationen finden Sie hier:

<http://zfsonlinux.org>

**Windows, macOS** Die folgenden Dateisysteme helfen beim Datenaustausch mit DOS-, Windows- und macOS-Systemen:

- ▶ **fat32 und vfat:** Dieses Dateisystem, genau genommen eine Kombination aus fat32 und vfat, wird auf den meisten SD-Karten und USB-Sticks verwendet. Ursprünglich kam es auch als Dateisystem für Windows 9x/ME zum Einsatz. Linux kann derartige Dateisysteme lesen und schreiben.
- ▶ **exfat:** Diese Weiterentwicklung des vfat-Dateisystems ist besonders für große Flash-Datenträger geeignet. Es gibt Treiber für Linux, diese müssen aber in der Regel extra installiert werden.
- ▶ **ntfs und ReFS:** ntfs ist in allen aktuellen Windows-Versionen ab Windows NT im Einsatz. Linux kommt mit ntfs gut zurecht und kann Dateien lesen und schreiben. Für das neue *Resilient File System* (ReFS), das aktuell vereinzelt auf Windows-Server-Installationen genutzt wird, gibt es momentan nur einen proprietären Treiber der Firma Paragon.
- ▶ **hfs und hfsplus:** Diese Dateisysteme werden auf Apple-Rechnern eingesetzt. Linux kann derartige Dateisysteme lesen und schreiben. Das Schreiben funktioniert allerdings nur, wenn die Dateisysteme unter macOS *ohne* Journaling-Funktionen eingerichtet wurden.
- ▶ **apfs:** 2016 stellte Apple das neue *Apple File System* (apfs) vor. Seit Mitte 2017 kommt es unter iOS zum Einsatz, ab Herbst 2017 wird es hfs auch unter macOS ablösen. Aktuell gibt es für apfs noch keine Linux-Treiber.

**CD/DVD** Auf Daten-CDs und DVDs werden üblicherweise eigene Dateisysteme verwendet:

- ▶ **iso9660:** Das Dateisystem für CD-ROMs wird durch die ISO-9660-Norm definiert. Diese Norm sieht allerdings nur kurze Dateinamen vor. Lange Dateinamen werden je nach Betriebssystem durch unterschiedliche Erweiterungen unterstützt (Rockridge, Joliet).
- ▶ **udf:** Als Nachfolger zu ISO 9660 hat sich das *Universal Disk Format* etabliert. Es kommt häufig bei DVDs zum Einsatz.

**Netzwerkdateisysteme**

Dateisysteme müssen sich nicht auf der lokalen Festplatte befinden – sie können auch über ein Netzwerk eingebunden werden. Der Linux-Kernel unterstützt diverse Netzwerkdateisysteme, von denen die folgenden fünf am häufigsten zum Einsatz kommen:

- ▶ **nfs:** Das *Network File System* (NFS) ist das unter Unix wichtigste Netzwerkdateisystem.
- ▶ **smbfs/cifs:** Diese Dateisysteme ermöglichen das Einbinden von Windows- oder Samba-Netzwerkverzeichnissen in den Verzeichnisbaum.

- ▶ **sshfs:** Das eher selten eingesetzte Dateisystem `sshfs` ermöglicht es, über SSH erreichbare Verzeichnisse in den lokalen Verzeichnisbaum einzubinden.
- ▶ **coda:** Dieses Dateisystem ist am ehesten mit NFS vergleichbar. Es bietet eine Menge Zusatzfunktionen, ist aber nicht sehr verbreitet.

Unter Linux gibt es eine Reihe von Dateisystemen, die nicht zum Speichern von Daten auf einer Festplatte oder einem anderen Datenträger gedacht sind, sondern lediglich zum Informationsaustausch zwischen dem Kernel und Anwendungsprogrammen. In `/proc/filesystems` sind diese Dateisysteme mit dem Begriff `nodev` gekennzeichnet. Im Folgenden werden nur die wichtigsten derartigen Dateisysteme kurz vorgestellt:

Virtuelle  
Dateisysteme

- ▶ **devpts:** Dieses Dateisystem ermöglicht via `/dev/pts/*` den Zugriff auf Pseudo-Terminals (kurz PTYs) gemäß der Unix-98-Spezifikation. Pseudo-Terminals emulieren eine serielle Schnittstelle und werden z. B. in Terminalfenstern eingesetzt.
- ▶ **proc und sysfs:** Das `proc`-Dateisystem dient zur Abbildung von Verwaltungsinformationen des Kernels bzw. der Prozessverwaltung. Ergänzend dazu bildet das `sysfs`-Dateisystem die Zusammenhänge zwischen dem Kernel und der Hardware ab. Die beiden Dateisysteme sind an den Positionen `/proc` und `/sys` eingebunden.
- ▶ **tmpfs:** Dieses temporäre Dateisystem ermöglicht einen effizienten Datenaustausch zwischen Programmen. Diverse `/run/xxx`-Verzeichnisse werden damit realisiert. In diesen Verzeichnissen gespeicherte Daten gehen beim Neustarten des Rechners verloren.
- ▶ **devtmpfs:** Das `devtmpfs`-Dateisystem bildet die Device-Dateien im `/dev`-Verzeichnis ab.
- ▶ **cgroup:** Die sogenannten Control Groups ermöglichen es, die Nutzung von Ressourcen durch einzelne Prozesse zu steuern bzw. zu limitieren. Das virtuelle *Cgroup Filesystem* hilft dabei, die aktuellen Cgroup-Einstellungen auszulesen oder zu verändern. Die Grundlagen der Control Groups sind hier dokumentiert:

<http://www.kernel.org/doc/Documentation/cgroup-v1/cgroups/cgroups.txt>

Abschließend folgen hier noch einige Dateisysteme bzw. Schlüsselwörter, die sich in die obigen Gruppen nicht einordnen lassen:

Sonstige  
Dateisysteme

- ▶ **auto:** Es gibt kein `auto`-Dateisystem. `auto` darf aber in `/etc/fstab` bzw. bei `mount` zur Angabe des Dateisystems verwendet werden. Linux versucht dann, das Dateisystem selbst zu erkennen. Das funktioniert für die meisten wichtigen Dateisysteme.
- ▶ **autofs:** Auch `autofs` ist kein eigenes Dateisystem, sondern eine Kernelerweiterung, die für die gerade benötigten Dateisysteme automatisch `mount` ausführt. Wird das Dateisystem eine Weile nicht mehr verwendet, wird ebenfalls automatisch `umount` ausgeführt. Dieses Verfahren bietet sich vor allem dann an, wenn von zahlreichen NFS-Verzeichnissen immer nur einige wenige aktiv genutzt werden.

Intern wird dazu beim Systemstart das Programm `automount` gestartet. Die Konfiguration erfolgt durch die Dateien `/etc/auto.master` bzw. `/etc/auto.misc`. Weitere Details finden Sie hier:

<http://tldp.org/HOWTO/Automount.html>

- ▶ **cramfs und squashfs:** Das *Cram Filesystem* und das *Squash Filesystem* sind Read-Only-Dateisysteme. Sie dienen dazu, möglichst viele Daten in komprimierter Form in ein Flash Memory bzw. in ein ROM (Read Only Memory) zu packen. Das Squash-Dateisystem wird auch vom Paketverwaltungssystem Snap (Ubuntu) verwendet, um Snap-Dateien als Read-Only-Dateisysteme in den Verzeichnisbaum einzubinden.
- ▶ **fuse:** Das *Filesystem in Userspace* (FUSE) ermöglicht es, Dateisystemtreiber außerhalb des Kernels zu entwickeln und zu nutzen. FUSE wird also immer zusammen mit einem externen Dateisystemtreiber eingesetzt. FUSE wird beispielsweise vom NTFS-Treiber `ntfs` verwendet.
- ▶ **gfs und ocfs:** Das *Global File System* (GFS) und das *Oracle Cluster File System* (OCFS) ermöglichen den Aufbau riesiger, vernetzter Dateisysteme, auf die mehrere Rechner parallel zugreifen.
- ▶ **loop:** Das *Loopback-Device* ist ein Adapter, der eine gewöhnliche Datei wie ein Block-Device ansprechen kann. Damit können Sie in einer normalen Datei ein beliebiges Dateisystem unterbringen und mit `mount` in den Verzeichnisbaum einbinden. Die dazugehörige Kernelfunktion *Loopback Device Support* ist im Modul `loop` realisiert. Das Loopback-Device wird z. B. für die Erstellung einer Initial-RAM-Disk für GRUB, für die Realisierung von verschlüsselten Dateisystemen oder zum Testen von ISO-Images verwendet.
- ▶ **none:** Naturgemäß ist auch `none` kein Dateisystem. Es besteht aber die selten genutzte Möglichkeit, ein lokales Verzeichnis an einem anderen Ort in den Verzeichnisbaum einzubinden. Dabei geben Sie bei `mount` bzw. in `/etc/fstab` als Dateisystemtyp `none` und als zusätzliche Option `bind` an. Die Wirkung ist ähnlich wie bei einem symbolischen Link, die interne Realisierung aber vollkommen anders. Diese Vorgehensweise ist z. B. bei der Konfiguration eines NFS4-Servers zweckmäßig.
- ▶ **unionfs/aufs/mhddfs/overlay:** Das Konzept von `unionfs` bzw. dessen Variante `aufs` ermöglicht es, mehrere Dateisysteme quasi übereinanderzulegen, wobei das oberste Dateisystem Vorrang hat. `unionfs` und `aufs` kommen bei manchen Live-Systemen zur Anwendung: Linux startet direkt von der DVD. Dem Read-Only-Dateisystem der DVD wird ein RAM-Disk-Dateisystem übergestülpt, in dem Änderungen durchgeführt werden können.

Auch das Dateisystem `mhddfs` fügt Dateien aus mehreren physikalischen Dateisystemen zu einem virtuellen Dateisystem zusammen. Beim Erstellen neuer Dateien

werden die physikalischen Dateisysteme der Reihe nach gefüllt. Neue Dateien werden also zuerst im ersten Dateisystem gespeichert, bis dieses einen vorgegebenen Grenzwert erreicht, dann auf dem zweiten Dateisystem etc.

Die modernste Implementierung eines Overlay-Dateisystems ist aktuell `overlay` bzw. `overlay2`. Dieses Dateisystem wird z. B. von Docker verwendet, um Änderungen gegenüber einem Basis-Image zu speichern.

- ▶ **Verschlüsselte Dateisysteme:** Linux kennt verschiedene Verfahren, um den Inhalt von Dateisystemen zu verschlüsseln. Einige dieser Verfahren basieren direkt auf eigenen Dateisystemen (z. B. `CryptoFS` oder `eCryptfs`). Verbreiteter ist allerdings die Kombination von LVM und Verschlüsselung. Seit Kernel 4.1 enthält das `ext4`-Dateisystem eingebaute Verschlüsselungsfunktionen.

Welche Dateisysteme direkt in den laufenden Kernel integriert bzw. zurzeit als Modul geladen sind, können Sie der Datei `/proc/filesystems` entnehmen. Welche Kernelmodule für weitere Dateisysteme darüber hinaus noch zur Verfügung stehen, sehen Sie im Verzeichnis `/lib/modules/n.n/kernel/fs/`.

## 22.8 Verwaltung des Dateisystems (mount und /etc/fstab)

Nach der Installation von Linux müssen Sie sich normalerweise nicht um die Verwaltung des Dateisystems kümmern: Über diverse Verzeichnisse können Sie auf alle oder zumindest die meisten Datenpartitionen der Festplatte zugreifen. Beim Einlegen von DVDs bzw. beim Anschließen externer Datenträger werden deren Dateisysteme automatisch in den Verzeichnisbaum integriert. Alles funktioniert gleichsam wie von Zauberhand.

Dieser Abschnitt wirft einen Blick hinter die Kulissen und beschreibt die Kommandos `mount` und `umount` sowie die Datei `/etc/fstab`:

- ▶ `mount` bzw. `umount` werden immer dann ausgeführt, wenn ein Dateisystem in den Verzeichnisbaum integriert bzw. wieder aus ihm gelöst wird. Selbstverständlich können Sie diese Kommandos als `root` auch selbst ausführen, wenn die Automatismen versagen bzw. wenn Sie ohne ein grafisches Desktop-System arbeiten.
- ▶ Die Konfigurationsdatei `/etc/fstab` steuert, welche Dateisysteme beim Rechnerstart automatisch in den Verzeichnisbaum integriert werden und welche Optionen dabei gelten. `/etc/fstab` wird während der Linux-Installation vorkonfiguriert. Wenn Sie mit dieser Konfiguration nicht zufrieden sind bzw. wenn sich später Ihre Anforderungen ändern, müssen Sie die Datei mit einem Editor verändern. Dieser Abschnitt beschreibt die Syntax dieser Datei.

Überraschenderweise gibt es nur wenige grafische Konfigurationswerkzeuge, die beim mount-Vorgang bzw. bei einer Änderung von `/etc/fstab` helfen. Zu den wenigen Ausnahmen zählen das im vorigen Abschnitt kurz vorgestellte Programm `gnome-disks` sowie das YaST-Modul `SYSTEM • PARTITIONIEREN (SUSE)`.

Einen Sonderfall stellen externe Datenträger wie USB-Memorysticks oder Firewire-Festplatten dar: Die meisten Distributionen binden solche Datenträger automatisch in das Dateisystem ein, sobald sie mit dem Rechner verbunden werden. Details zum Umgang mit externen Datenträgern folgen in [Abschnitt 22.15](#).

### Aktuellen Zustand des Dateisystems ermitteln

- df** Wenn Sie wissen möchten, wie Ihr Linux-System zurzeit organisiert ist, führen Sie am einfachsten das Kommando `df -h` aus. Dieses Kommando zeigt an, an welcher Stelle im Dateisystem Festplatten, Datenträger etc. eingebunden sind und wie viel Platz auf den einzelnen Festplatten noch frei ist.
- mount** Das Kommando `mount` ohne weitere Optionen liefert noch detailliertere Informationen über die eingebundenen Dateisysteme. Außerdem zeigt das Kommando alle aktiven `mount`-Optionen. Leider gehen die wirklich interessanten Einträge im Ergebnis oft zwischen unzähligen virtuellen Dateisystemen unter. Im folgenden Beispiel wurde die Ausgabe spaltenweise eingerückt, um die Lesbarkeit zu verbessern:

```
user$ mount
/dev/mapper/vg-ubuntu on /                type ext4    (rw,errors=remount-ro)
/dev/mapper/vg-myhome on /myhome    type ext4    (rw)
/dev/mapper/vg-virt  on /virt     type ext4    (rw)
/dev/sda3            on /boot     type ext3    (rw)
cgroup              on /sys/fs/cgroup type tmpfs   (rw,relatime,mode=755)
proc                on /proc     type proc    (rw)
devtmpfs            on /dev      type devtmpfs (rw,mode=0755)
tmpfs               on /run      type tmpfs   (rw,noexec,nosuid,...)
none                on /run/lock type tmpfs   (rw,noexec,nosuid,...)
...
```

Die gleichen Ergebnisse wie `mount` liefert `findmnt`. Der Vorteil dieses Kommandos ist die wesentlich lesefreundlichere Formatierung der Ausgabe. Ähnliche Informationen wie `mount` liefern auch die Dateien `/etc/mtab` und `/proc/mounts`. Sie enthalten jeweils eine Liste aller Datenträger, die momentan eingebunden sind, zusammen mit dem Dateisystemtyp und den verwendeten `mount`-Optionen. `/etc/mtab` ändert sich jedes Mal, wenn ein Dateisystem in den Verzeichnisbaum eingebunden oder aus ihm gelöst wird. Die Syntax in `mtab` ist dieselbe wie in `/etc/fstab`.



## Dateisysteme manuell einbinden und lösen (mount und umount)

Nach der Installation einer aktuellen Linux-Distribution ist das System so konfiguriert, dass Sie `mount` nur sehr selten benötigen: Alle Linux-Dateisysteme sind in den Verzeichnisbaum eingebunden. Beim Einlegen von CDs/DVDs oder beim Anschließen externer Datenträger erscheint automatisch ein neues Fenster des KDE- oder Gnome-Dateimanagers. Auch wenn es vielleicht so aussieht, als würde das Ganze wie von Zauberhand funktionieren, wird hinter den Kulissen immer wieder das Kommando `mount` ausgeführt, um Dateisysteme in den Verzeichnisbaum einzubinden bzw. wieder aus ihm zu lösen.

Die Syntax von `mount` sieht folgendermaßen aus:

```
mount [optionen] device verzeichnis
```

In den Optionen wird unter anderem der Dateisystemtyp angegeben (`-t xxx`). Der Device-Name bezeichnet die Partition bzw. das Laufwerk. Als Verzeichnis kann ein beliebiges Verzeichnis des aktuellen Dateisystems angegeben werden. Dieses Verzeichnis muss bereits existieren. Erzeugen Sie es gegebenenfalls mit `mkdir`!

`mount` kann im Regelfall nur von `root` ausgeführt werden. Es besteht aber die Möglichkeit, dass `/etc/fstab` für einzelne Partitionen allen Benutzern erlaubt, `mount` auszuführen (Option `user` bzw. `users`).

Am einfachsten ist `mount` anhand einiger Beispiele zu verstehen: Das erste Beispiel ermöglicht den Zugriff auf die Daten einer Windows-Partition über das Verzeichnis `/windows`:

Beispiele

```
root# mkdir /windows
root# mount -t ntfs /dev/sda2 /windows
```

Das folgende Kommando bindet eine Daten-CD (ISO-9660-Dateisystem) im Verzeichnis `/media/cdrom` in das Dateisystem ein. Je nach Distribution müssen Sie statt `/dev/sdc0` das Device `/dev/sr0` angeben.

```
root# mount -t iso9660 /dev/scd0 /media/cdrom
```

Mit `mount -o remount` können Sie Optionen eines bereits eingebundenen Dateisystems verändern. Das folgende Kommando aktiviert beispielsweise die `exec`-Option für eine DVD, sodass darauf enthaltene Programme ausgeführt werden können:

remount

```
root# mount /media/dvd -o remount,exec
```

Falls beim Einbinden der Systempartition während des Rechnerstarts Probleme auftreten, wird die Partition nur `read-only` eingebunden. Um die Fehlerursache – etwa einen falschen Eintrag in `/etc/fstab` – zu beheben, ist es aber oft erforderlich, Änderungen im Dateisystem durchzuführen. Dazu führen Sie das folgende Kommando

aus. Mit ihm wird die Systempartition neu eingebunden, wobei jetzt auch Schreibzugriffe möglich sind.

```
root# mount -o remount,rw /
```

**umount** Um ein Dateisystem aus dem Verzeichnisbaum zu lösen, führen Sie `umount` aus:

```
root# umount /media/dvd
```

### Dateisysteme automatisch einbinden (/etc/fstab)

Es wäre sehr mühsam, wenn Sie nach jedem Systemstart diverse Partitionen neu einbinden müssten, bei jedem DVD-Wechsel `mount` mit allen Optionen angeben müssten etc. Der Schlüssel zur Arbeitserleichterung heißt `/etc/fstab`: Diese Datei gibt an, welche Datenträger beim Systemstart in das Dateisystem aufgenommen werden. Auf jeden Fall muss `fstab` die Systempartition sowie alle zur internen Verwaltung notwendigen Dateisysteme enthalten.

**Beispiel** Je nach Distribution kann eine minimale `fstab`-Datei wie folgt aussehen:

```
# zwei Beispielzeilen in /etc/fstab
/dev/sda2 / ext4 defaults 1 1
none /proc proc defaults 0 0
...
```

Durch die erste Zeile wird die zweite Partition der ersten Festplatte als Systemverzeichnis genutzt. Je nachdem, auf welcher Festplattenpartition Sie Linux installiert haben, müssen Sie statt `sda2` natürlich den Device-Namen Ihrer Linux-Partition angeben!

Mit der zweiten Zeile wird das System zur Prozessverwaltung in das Dateisystem eingebunden. Die Dateien und Verzeichnisse des `/proc`-Verzeichnisses existieren nicht tatsächlich auf der Festplatte; es handelt sich nur um ein Abbild von Daten, die kernel-intern verwaltet werden.

### Die Syntax in /etc/fstab

Aus den obigen Beispielen geht bereits das prinzipielle Format von `fstab` hervor: Jede Zeile beschreibt in sechs Spalten einen Datenträger (eine Partition, ein Dateisystem).

**Erste Spalte** Die erste Spalte enthält den Device-Namen des Datenträgers. Statt des Device-Namens können Sie auch den *Volume Name* oder die ID-Nummer des Dateisystems angeben. Die korrekte Syntax lautet in diesem Fall `LABEL=zeichenkette` oder `UUID=nnn-  
nnn`. Mit `blkid` ermitteln Sie den Partitionsnamen und die UUID einer Partition. Um diese Daten zu ändern, setzen Sie je nach Dateisystem unterschiedliche Werkzeuge ein, beispielsweise `tune2fs`.

```
root# blkid /dev/sda9
/dev/sda9: UUID="5a954fc1-00c6-4c25-a943-d4220eff350d" TYPE="ext4"
```

Der Vorteil von Labels oder UUIDs im Vergleich zu Device-Namen besteht darin, dass die Angabe selbst dann noch korrekt ist, wenn sich der Device-Name geändert hat. Das kann insbesondere bei USB-Datenträgern leicht passieren: Je nachdem, welche Datenträger vorher verwendet wurden, kann es durchaus sein, dass die externe Festplatte einmal unter `/dev/sdc` und das nächste Mal unter `/dev/sde` angesprochen wird.

Leider wird `fstab` insbesondere bei der Verwendung von UUIDs sehr unübersichtlich. Probleme kann es auch geben, wenn mehrere Linux-Distributionen parallel installiert werden. In der Regel werden bei jeder Installation einzelne Partitionen neu formatiert. Sie erhalten bei dieser Gelegenheit neue UUIDs. Die bisher installierten Distributionen kennen diese Partitionen nun nicht mehr, und `fstab` muss mühsam an die neuen UUIDs angepasst werden.

Die zweite Spalte gibt an, bei welchem Verzeichnis der Datenträger in den Dateibaum eingebunden wird. Die in der zweiten Spalte angegebenen Verzeichnisse müssen bereits existieren. Die Verzeichnisse müssen nicht leer sein, allerdings können Sie nach dem Einbinden des Dateisystems auf die darin enthaltenen Dateien nicht mehr zugreifen, sondern nur auf die Dateien des eingebundenen Datenträgers.

Zweite Spalte

| Dateisystem  | Verwendung                               |
|--------------|--|
| auto         | Dateisystem automatisch erkennen         |
| btrfs        | btrfs-Dateisystem                        |
| cifs         | Windows-Netzwerkverzeichnis (Samba)      |
| ext2, -3, -4 | ext-Dateisystem Version 2, 3 und 4       |
| iso9660      | Daten-CDs                                |
| nfs          | Unix-Netzwerkverzeichnis (NFS)           |
| ntfs         | Windows-Dateisystem                      |
| proc         | Prozessverwaltung ( <code>/proc</code> ) |
| smbfs        | Windows-Netzwerkverzeichnis (Samba)      |
| swap         | Swap-Partitionen oder -Dateien           |
| sysfs        | Systemverwaltung ( <code>/sys</code> )   |
| tmpfs        | temporäres Dateisystem                   |
| udf          | Universal Disk Format (DVDs, CD-RWs)     |
| vfat         | Windows-9x/ME-Dateisystem                |
| xfs          | XFS-Dateisystem                          |

Tabelle 22.4 Dateisysteme

- Dritte Spalte** Die dritte Spalte gibt das Dateisystem an. Tabelle 22.4 listet in alphabetischer Reihenfolge die wichtigsten Dateisysteme auf. Es ist auch zulässig, mehrere Dateisysteme durch Kommas getrennt anzugeben. Beispielsweise bietet sich `iso9660,udf` für CD- und DVD-Laufwerke an, weil für CDs und DVDs in der Regel nur diese beiden Dateisysteme infrage kommen. `mount` entscheidet sich zwischen den zur Auswahl stehenden Systemen automatisch für das richtige. Die Dateisystemnamen dürfen nicht durch Leerzeichen getrennt werden.
- Vierte Spalte** Die vierte Spalte bestimmt Optionen für den Zugriff auf den Datenträger. Mehrere Optionen werden durch Kommata getrennt. Abermals dürfen keine Leerzeichen eingefügt werden! Tabelle 22.5 zählt die wichtigsten universellen `mount`-Optionen auf. Wenn Sie gar keine Option nutzen möchten, geben Sie `defaults` an (im Plural!).

| Option                | Bedeutung  |
|-----------------------|--|
| <code>defaults</code> | Standardoptionen verwenden   |
| <code>dev</code>      | Kennzeichnung von Character- oder Block-Devices auswerten  |
| <code>discard</code>  | SSD-Trim aktivieren ( <code>ext4</code> , <code>btrfs</code> , <code>xf</code> s und <code>swap</code> )                                   |
| <code>exec</code>     | Programmausführung zulassen (z. B. für CD/DVD-Laufwerke)   |
| <code>noauto</code>   | Datenträger nicht beim Systemstart einbinden   |
| <code>nodev</code>    | Kennzeichnung von Character- oder Block-Devices ignorieren   |
| <code>noexec</code>   | keine Programmausführung erlaubt   |
| <code>nosuid</code>   | Suid- und Guid-Zugriffsbits nicht auswerten  |
| <code>ro</code>       | Read Only (Schreibschutz)  |
| <code>sw</code>       | Swap (Swap-Datei oder -Partition)  |
| <code>suid</code>     | Suid- und Guid-Zugriffsbits auswerten  |
| <code>sync</code>     | Schreibzugriffe nicht puffern (sicherer, aber langsamer)   |
| <code>owner</code>    | Der Besitzer darf ( <code>u</code> ) <code>mount</code> ausführen.   |
| <code>user</code>     | Jeder darf <code>mount</code> ausführen, aber nur der Benutzer des letzten <code>mount</code> -Aufrufs darf <code>umount</code> ausführen. |
| <code>users</code>    | Jeder darf <code>mount</code> und <code>umount</code> ausführen.   |

Tabelle 22.5 `mount`-Optionen

- Fünfte Spalte** Die fünfte Spalte enthält Informationen für das Programm `dump` und wird von Linux ignoriert. Es ist üblich, für die Systempartition 1 und für alle anderen Partitionen oder Datenträger 0 einzutragen.
- Sechste Spalte** Die sechste Spalte gibt an, ob und in welcher Reihenfolge die Dateisysteme beim Systemstart überprüft werden sollen. Oft wird 1 für die Systempartition und 0 für

alle anderen Partitionen eingetragen. Das bedeutet, dass beim Rechnerstart nur die Systempartition auf Fehler überprüft und gegebenenfalls repariert wird.

Falls Sie möchten, dass weitere Partitionen automatisch überprüft werden, geben Sie bei diesen Partitionen die Ziffer 2 an, d. h., die Überprüfung soll nach der Kontrolle der Systempartition erfolgen. Wenn Einträge in der fünften und sechsten Spalte in `/etc/fstab` fehlen, wird 0 angenommen.

## 22.9 Dateisystemgrundlagen

Im Mittelpunkt der folgenden Seiten stehen die Linux-Dateisysteme `ext2`, `ext3`, `ext4`, `btrfs` und `xfs`. Bevor ich deren Einrichtung und Administration beschreibe, gibt dieser Abschnitt einige Grundlageninformationen, die unabhängig vom Dateisystemtyp sind.

Alle gängigen Linux-Dateisysteme unterstützen Journaling-Funktionen. In seiner einfachsten Form bedeutet Journaling, dass der Beginn und das Ende jeder Dateiope- **Journaling** ration in einer speziellen Datei mitprotokolliert werden. Dank des Protokolls kann später geprüft werden, ob eine bestimmte Dateioperation vollständig ausgeführt wurde. Wenn das nicht der Fall ist, kann die Operation widerrufen werden. In der Datenbankwelt spricht man hier von Transaktionen.

Bei fortgeschrittenen Journaling-Systemen besteht die Möglichkeit, die eigentlichen Änderungen an den Dateien im Journal zu protokollieren. Das verlangsamt den gewöhnlichen Betrieb, gibt aber mehr Möglichkeiten zur späteren Rekonstruktion.

Wenn nun eine Dateioperation nicht vollständig abgeschlossen werden kann, geht dies aus dem Protokoll hervor. Bei einfachem Journaling sind die Änderungen zwar verloren, der bisherige Zustand der Datei steht aber zumeist noch zur Verfügung. Versprechen Sie sich also keine Wunder von der Journaling-Funktion!

Der große Vorteil der Journaling-Funktionen besteht darin, dass das Dateisystem beim nächsten Rechnerstart sehr rasch wieder in einen konsistenten Zustand gebracht und beinahe sofort wieder genutzt werden kann. Das ist ein großer Unterschied im Vergleich zu früher, wo nach einem Absturz oder Stromausfall das gesamte Dateisystem systematisch nach eventuellen Fehlern durchsucht werden musste. Das dauerte mehrere Minuten, bei sehr großen Festplatten eventuell sogar Stunden.

Bei einem Stromausfall gibt auch das Journaling keine Garantie für ein konsistentes Dateisystem! Das Problem liegt bei den Festplatten bzw. SSDs: Diese verwenden aus Effizienzgründen beim Schreiben einen internen Zwischenspeicher. Daher kann es passieren, dass das Dateisystem vom Datenträger die Bestätigung erhält, dass er die Daten empfangen und gesichert hat. Tatsächlich kann es danach aber noch Sekunden **Datenverluste trotz Journaling**

dauern, bis die Daten vom Zwischenspeicher physikalisch auf die Festplatte geschrieben werden. Bei SSDs ist diese Zeitspanne viel kürzer, aber das ändert nichts am prinzipiellen Problem.

Tritt in dieser Zeitspanne ein Stromausfall auf, gehen die Daten im Zwischenspeicher verloren. Bei manchen Festplatten lässt sich dieser Cache deaktivieren. Dadurch verringert sich die Geschwindigkeit von Schreiboperationen aber derart, dass in der Praxis zumeist darauf verzichtet wird.

Unabhängig vom Schreib-Cache ist das Verhalten einer Festplatte während eines plötzlichen Stromausfalls undefiniert. Es kann also auch passieren, dass die Festplatte statt Ihrer Daten Zufallsbits schreibt, bevor der Schreibkopf in Sicherheit gebracht wird. Eine Diskussion zu diesem Thema finden Sie hier:

<https://lwn.net/Articles/191352>

Anders formuliert: Journaling-Dateisysteme sind eine feine Sache, schließen einen Datenverlust bei einem Stromausfall aber nicht aus.

#### Automatische Überprüfung des Dateisystems

Wenn Linux beim Starten erkennt, dass der Rechner zuletzt nicht ordnungsgemäß heruntergefahren wurde, führt es für die Systempartition und je nach Konfiguration auch für andere in `/etc/fstab` genannte Partitionen eine Überprüfung des Dateisystems durch. Ob eine Überprüfung stattfindet oder nicht, entscheidet die sechste Spalte in `/etc/fstab`. Dank der Journaling-Funktionen ist diese Überprüfung normalerweise blitzschnell erledigt.

Davon losgelöst sehen einige Dateisysteme (unter anderem `ext` in allen Versionen) eine regelmäßige Überprüfung des Dateisystems auf Konsistenzfehler vor. Diese relativ zeitaufwendigen Tests erfolgen beim Start des Rechners, wenn seit dem letzten Test eine bestimmte Zeitspanne oder eine Anzahl von `mount`-Vorgängen überschritten wurde.

Nach der Einführung der Journaling-Funktionen wurde vielfach argumentiert, der regelmäßige Konsistenztest sei jetzt überflüssig. Das stimmt aber leider nicht ganz: Ein Dateisystem kann auch durch Hardware-Fehler der Festplatte inkonsistent werden – und die Wahrscheinlichkeit solcher Fehler steigt mit der zunehmenden Festplattengröße!

Beispielsweise habe ich im Datenblatt einer 1-TiB-Festplatte die Angabe gefunden, dass die Wahrscheinlichkeit für Bitfehler (*Nonrecoverable Read Errors per Bits Read*) kleiner als  $1 \text{ zu } 10^{15}$  ist. Das klingt wirklich vernachlässigbar. Wenn Sie allerdings in Rechnung stellen, dass auf dieser Festplatte  $8 \times 10^{12}$  Bits Platz finden, wird klar, dass Datenfehler im regulären Betrieb – also ohne irgendwelche Beschädigungen – sehr wohl zu erwarten sind. Ein regelmäßiger Konsistenztest des Dateisystems kann diese

Fehler zwar nicht verhindern, bietet aber eine gewisse Chance, ein Fehlverhalten festzustellen und zu korrigieren, zumindest dann, wenn für die interne Verwaltung des Dateisystems kritische Bereiche betroffen sind.

Wirklich fehlertolerant sind die in diesem Buch vorgestellten Dateisysteme leider alle nicht. Dateisysteme, die durch Prüfsummen Hardware-Fehler erkennen bzw. durch redundante Speicherung derartige Fehler sogar korrigieren können, sind aber ein hochaktuelles Forschungsgebiet.

Eine manuelle Überprüfung können Sie einfach mit dem Kommando `fsck` durchführen. Die betreffende Partition darf während der Kontrolle allerdings nicht verwendet werden, d. h., Sie müssen vorher `umount` ausführen.

Manuelle  
Überprüfung des  
Dateisystems

Die Systempartition können Sie im laufenden Betrieb allerdings nicht überprüfen, weil Sie das Dateisystem nicht mit `umount` abmelden können. Stattdessen führen Sie als `root` das Kommando `touch /forcefsck` aus und starten den Rechner neu. Die Datei `forcefsck` wird auch erzeugt, wenn Sie `shutdown` mit der zusätzlichen Option `-F` ausführen.

Wenn die Datei `/forcefsck` existiert, wird bei fast allen Distributionen beim nächsten Start automatisch eine Überprüfung des Dateisystems durchgeführt. Sollte das nicht funktionieren, fahren Sie den Rechner mit einem Rescue- oder Live-System hoch und führen `fsck` von dort aus.

In der Vergangenheit tauchte immer wieder die Frage auf, wie groß Dateien maximal sein dürfen. Die Antwort hängt davon ab, welchen Kernel, welche CPU-Architektur, welche glibc-Bibliothek und welches Dateisystem Sie verwenden. Aktuelle Distributionen unterstützen durchweg die LFS-Erweiterungen in der glibc-Bibliothek. LFS steht dabei für *Large Filesystem Support*. Die Dateigröße ist dank LFS mit  $2^{63}$  Byte nahezu unbegrenzt. Zum anderen geben aber auch die verschiedenen Dateisystemtypen Limits für die maximale Datei(system)größe vor. [Tabelle 22.6](#) fasst die Daten zusammen. Dabei gilt: 1 TiB = 1024 GiB.

Größenlimits

| Dateisystem | Maximale Dateigröße | Maximale Dateisystemgröße            |
|-------------|---------------------|--------------------------------------|
| btrfs       | 16.777.216 TiB      | 16.777.216 TiB = 16 Exabyte = 16 EiB |
| ext3        | 2 TiB               | 32 TiB (bei 8 KiB Blockgröße)        |
| ext4        | 16 TiB              | 1.048.576 TiB = 1 EiB                |
| xfs         | 9.437.184 TiB       | 9.437.184 TiB                        |
| ZFS         | 16.777.216 TiB      | 16.777.216 TiB                       |

**Tabelle 22.6** Maximale Dateisystemgröße

**Änderung des Dateisystemtyps** Eine Umwandlung des Dateisystemtyps ist in der Regel unmöglich. Der einzige Weg besteht darin, ein neues Dateisystem im gewünschten Typ einzurichten und dann alle Dateien zu kopieren. Zu den wenigen Ausnahmen zählte `btrfs-convert` zur Konvertierung von `ext3/-4` nach `btrfs`. Dieses Kommando wird aber nicht mehr offiziell unterstützt.

## 22.10 Das ext-Dateisystem (ext2, ext3, ext4)

**Versionen** Die verschiedenen `ext`-Versionen dominieren die Welt der Linux-Dateisysteme. Die erste Version wurde nur kurz in der Anfangsphase von Linux eingesetzt. Dann folgten in den Jahren 1993, 2002 und 2008 die Varianten `ext2`, `ext3` und `ext4`. Die maximale Dateisystemgröße wuchs dabei von 2 GiB auf ein Exabyte (1.048.576 TiB). Seit `ext3` stehen Journaling-Funktionen zur Verfügung. Bei allen Versionen wurde auf Kompatibilität geachtet, die eine schrittweise Migration vorhandener Dateisysteme ermöglichte.

Die Kompatibilität der verschiedenen `ext`-Dateisystemversionen drückt sich auch dadurch aus, dass diverse Administrationswerkzeuge weiterhin die Versionsnummer 2 im Kommandonamen haben, obwohl sie auch für neuere Versionen eingesetzt werden können, z. B. `tune2fs` oder `resize2fs`.

Die `ext`-Entwicklung steht seit der Veröffentlichung von `ext4` keineswegs still. Es gibt zwar momentan keine Pläne für `ext5`, es werden aber laufend neue Funktionen in `ext4` eingebaut. Zuletzt wurden in der Kernelversion 4.1 Verschlüsselungsfunktionen direkt in den Dateisystemtreiber integriert.

**/etc/fstab** Einträge für `ext3/ext4`-Dateisysteme in `/etc/fstab` sehen üblicherweise so wie im folgenden Beispiel aus:

```
# /etc/fstab: Linux-Dateisysteme
/dev/sdb8 /          ext4 defaults      1 1
/dev/sdb9 /boot      ext3 defaults      0 0
/dev/sdb9 /data     ext4 acl,user_xattr 0 0
```

### Journaling

Die Dateisysteme `ext3` und `ext4` unterstützen Journaling-Funktionen. Der beim `mount`-Kommando bzw. in `/etc/fstab` definierte `data`-Parameter bestimmt, mit welchem Verfahren das Journaling durchgeführt wird:

- ▶ `data=ordered`: Bei diesem Modus werden im Journal nur Metadaten gespeichert, also Informationen *über* Dateien, aber keine Inhalte. Im Journal werden Dateien erst dann als korrekt (*committed*) gekennzeichnet, wenn sie vollständig auf der



Festplatte gespeichert worden sind. Nach einem Crash kann das Dateisystem sehr rasch wieder in einen konsistenten Zustand gebracht werden, weil alle unvollständig gespeicherten Dateien anhand des Journals sofort erkannt werden. Es ist aber nicht möglich, unvollständig gespeicherte Dateien wiederherzustellen.

Im Modus `data=ordered` wird das Journal alle fünf Sekunden mit der Festplatte synchronisiert. Bei `ext3` hat das zur Folge, dass sämtliche Änderungen an irgendwelchen Dateien innerhalb von fünf Sekunden physikalisch auf der Festplatte gespeichert werden. Dieses Standardverhalten ist zwar nicht besonders effizient, dafür aber sehr sicher: Selbst bei Totalabstürzen und Stromausfällen sind massive Datenverluste äußerst selten. `data=ordered` hat bei `ext3` jedoch eine unerfreuliche Nebenwirkung: Bei jedem Aufruf der `fsync`-Funktion wird nicht nur eine bestimmte Datei, sondern das gesamte Dateisystem synchronisiert. Das kann zu spürbaren Verzögerungen führen.

Bei `ext4` wird das Journal zwar ebenfalls alle fünf Sekunden synchronisiert, die Datenänderungen werden aber aufgrund der *Delayed Allocation* (siehe unten) oft erst viel später gespeichert. Nur ein expliziter Aufruf der `fsync`-Funktion stellt die sofortige physikalische Speicherung einer Datei sicher! Glücklicherweise erfordert `fsync` bei `ext4` nicht, dass das gesamte Dateisystem synchronisiert werden muss. Die Funktion wird daher wesentlich schneller ausgeführt.

- ▶ `data=writeback`: Dieser Modus ähnelt dem `ordered`-Modus. Der Unterschied besteht darin, dass das Journal und die Dateioperationen nicht immer vollständig synchron sind. Das Dateisystem wartet mit den *committed*-Einträgen im Journal nicht auf den Abschluss der Speicheroperation auf der Festplatte. Damit ist das Dateisystem etwas schneller als im `ordered`-Modus. Nach einem Crash ist die Integrität des Dateisystems weiterhin sichergestellt. Allerdings kann es vorkommen, dass veränderte Dateien alte Daten enthalten. Dieses Problem tritt nicht auf, wenn Anwendungsprogramme – wie im POSIX-Standard vorgesehen – den Speichervorgang mit `fsync` abschließen.
- ▶ `data=journal`: Im Gegensatz zu den beiden anderen Modi werden jetzt im Journal auch die tatsächlichen Daten gespeichert. Dadurch müssen alle Änderungen *zweimal* gespeichert werden (zuerst in das Journal und dann in die betroffene Datei). Deswegen ist das Dateisystem in diesem Modus deutlich langsamer. Dafür können nach einem Crash Dateien wiederhergestellt werden, deren Änderungen bereits vollständig in das Journal (aber noch nicht in die Datei) eingetragen worden sind.

Die aus Performance-Sicht wichtigste Neuerung in `ext4` ist die sogenannte *Delayed Allocation* – eine Funktion, die es auch in vielen anderen modernen Dateisystemen gibt, z. B. in `btrfs`, `HFS+`, `xfs` und `ZFS`. *Delayed Allocation* (auch *Allocation on Flush* genannt) bedeutet, dass bei Änderungen die Datenblöcke zur Speicherung von Datei-

**Delayed  
Allocation**

änderungen nicht sofort reserviert werden, sondern erst zu dem Zeitpunkt, zu dem die Daten physikalisch gespeichert werden – und das kann durchaus eine halbe Minute dauern. Das bringt zwei wesentliche Vorteile mit sich: Zum einen können nun Speicheroperationen gebündelt werden, was die Geschwindigkeit erhöht und die Fragmentierung des Dateisystems mindert. Zum anderen kommt es bei temporären Dateien, die nur wenige Sekunden existieren, oft zu gar keiner physikalischen Speicherung.

#### Standardverhalten und Optionen

Sofern der Journaling-Modus und die Allokierung im `mount`-Kommando bzw. in `/etc/fstab` nicht explizit eingestellt werden, gilt das folgende Standardverhalten:

```
ext3 ab Kernel 2.6.36:  data=ordered
ext4:                  data=ordered mit Delayed Allocation
```

Um herauszufinden, welcher Journaling-Modus aktiv ist, müssen Sie die Kernelmeldungen lesen. Im folgenden Beispiel gibt es je eine `ext3`- und eine `ext4`-Partition:

```
root# dmesg | grep EXT
EXT3 FS on sda3, internal journal
EXT3-fs: mounted filesystem with ordered data mode.
...
EXT4-fs (sda4): mounted filesystem with ordered data mode
```

Um einen bestimmten Journaling-Modus explizit auszuwählen, geben Sie bei `mount` oder in `/etc/fstab` die Option `data=xxx` an. Bei `ext4` können Sie zudem die `Delayed Allocation` durch die Option `nodelalloc` deaktivieren.

## Administration

#### Dateisystem einrichten

`ext2`-, `ext3`- und `ext4`-Dateisysteme werden mit `mkfs.ext2`, `mkfs.ext3` oder `mkfs.ext4` formatiert. Im folgenden Beispiel wird in einer 12 GiB großen Partition ein `ext4`-Dateisystem eingerichtet. `mkfs.ext4` entscheidet sich selbstständig für eine Blockgröße von 4 KiB und für 786.432 Inodes.

Das bedeutet, dass Sie im Dateisystem maximal ca. 780.000 Dateien anlegen können. Die durchschnittliche Dateigröße würde dann 16 KiB betragen. Wenn Sie mehr kleinere oder weniger größere Dateien speichern möchten, können Sie mit `-i <n>` angeben, nach wie vielen Bytes jeweils ein Inode vorgesehen werden soll. Wenn die durchschnittliche Dateigröße kleiner ist als `n`, limitiert nicht die Größe der Partition, sondern die Inode-Anzahl das Dateisystem.

Beachten Sie, dass die absolute Anzahl der Inodes nicht mehr verändert werden kann, auch nicht bei einer späteren Vergrößerung des Dateisystems! In den meisten Fällen ist der Vorgabewert von `mkfs.ext4` zweckmäßig.

```

root# mkfs.ext4 /dev/sdb1
mke2fs 1.42.12
Ein Dateisystem mit 3145472 (4k) Blöcken und 786432 Inodes wird erzeugt.
UUID des Dateisystems: 0b1e34fb-a8a9-435e-a186-80d003abf4ee
Superblock-Sicherungskopien gespeichert in den Blöcken:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208
Anfordern von Speicher für die Gruppentabellen: erledigt
Inode-Tabellen werden geschrieben: erledigt
Das Journal (32768 Blöcke) wird angelegt: erledigt
Die Superblöcke und die Informationen über die Dateisystemnutzung werden
geschrieben: erledigt

```

ext-Dateisysteme können beim Rechnerstart regelmäßig auf Fehler überprüft werden, und zwar nach einer bestimmten Anzahl von mount-Vorgängen bzw. nach einer gewissen Zeit, je nachdem, welches Kriterium vorher erfüllt war. Die entsprechenden Parameter legen Sie mit `tune2fs` fest. Dabei geben Sie mit `-c` die maximale mount-Anzahl und mit `-i` das Zeitintervall in Tagen an. Außerdem müssen Sie sicherstellen, dass die sechste Spalte der entsprechenden `fstab`-Zeile einen Wert größer 0 enthält.

**Dateisystem-  
überprüfung**

```

root# tune2fs /dev/sdb1 -c 30 -i 365
tune2fs 1.42.12 (29-Aug-2014)
Die maximale Anzahl von Einhängungen wird auf 30 gesetzt
Der Abstand zwischen den Prüfläufen wird auf 31536000 Sekunden gesetzt

```

Zur Deaktivierung der Funktion geben Sie jeweils die Werte 0 an bzw. setzen die sechste Spalte in `/etc/fstab` auf 0. Die aktuellen Intervalle für die automatische Überprüfung des Dateisystems können Sie mit `tune2fs -l` feststellen:

```

root# tune2fs /dev/sdb1 -l | egrep -i 'check|mount count'
Mount count:          0
Maximum mount count:  30
Last checked:         Tue Sep 15 07:42:35 2015
Check interval:       31536000 (12 months, 5 days)
Next check after:     Wed Sep 14 07:42:35 2016

```

Trotz der Journaling-Funktionen ist eine Überprüfung des Dateisystems hin und wieder sehr zu empfehlen, zumindest einmal pro Jahr. Zum einen werden so eventuelle Hardware-Fehler der Festplatte erkannt. Zum anderen kann es sein, dass die Dateisystemtreiber noch unbekannte Fehler enthalten. Je früher daraus resultierende Fehler korrigiert werden, desto kleiner ist der potenzielle Schaden.

Eine manuelle Überprüfung können Sie jederzeit mit dem Kommando `fsck.ext2/ext3/ext4` durchführen. Die betreffende Partition darf während der Kontrolle allerdings nicht gerade verwendet werden, d. h., Sie müssen gegebenenfalls vorher `umount` ausführen.

```

root# fsck.ext4 -f /dev/sdb1
e2fsck 1.42.12 (29-Aug-2014)
Durchgang 1: Inodes, Blöcke und Größen werden geprüft
Durchgang 2: Verzeichnisstruktur wird geprüft
Durchgang 3: Verzeichnisverknüpfungen werden geprüft
Durchgang 4: Referenzzähler werden überprüft
Durchgang 5: Zusammengefasste Gruppeninformation wird geprüft
/dev/sdb1: 3347/786432 Dateien (0.1% nicht zusammenhängend), 93562/3145472 Blöcke

```

Meist stellt sich bei der Überprüfung heraus, dass alles in Ordnung ist. Andernfalls werden die Reste nicht mehr rekonstruierbarer Dateien im `/lost+found`-Verzeichnis der jeweiligen Partition gespeichert. Falls es sich um Textdateien gehandelt hat, können Sie vielleicht aus den Überresten noch brauchbare Informationen entnehmen.

**Partitionsnamen einstellen** Mit `e2label` können Sie den internen Namen eines `ext`-Dateisystems (*Filesystem Volume Name*) ermitteln bzw. einstellen:

```
root# e2label /dev/sdb1 mylabel
```

Diesen Namen können Sie in der ersten Spalte von `/etc/fstab` statt des Device-Namens angeben.

**UUID einstellen** Beim Einrichten erhält das Dateisystem automatisch eine UUID, die Sie mit `blkid` ermitteln. Bei Bedarf verändern Sie diese Nummer mit `tune2fs -U`. Die Veränderung kann im laufenden Betrieb erfolgen, `umount` ist nicht erforderlich.

```

root# tune2fs -U random /dev/sdb1                (zufällige UUID)
root# tune2fs -U f7c49568-8955-4ffa-9f52-9b2ba9877021 /dev/sdb1 (eigene UUID)

```

**Größe des Dateisystems ändern** Mit `resize2fs` können Sie ein `ext`-Dateisystem vergrößern oder verkleinern. Beachten Sie, dass Sie bei einer Vergrößerung *vorher* die zugrunde liegende Partition, das Logical Volume (LV) oder den virtuellen Datenträger vergrößern müssen, bei einer Verkleinerung die Partition, das LV oder den virtuellen Datenträger aber erst *nachher* verkleinern dürfen! Im folgenden Beispiel wird das LV mit `lvextend` vergrößert. Weitere Details zur LVM-Administration folgen in [Abschnitt 22.18](#), »Logical Volume Manager (LVM)«.

```

root# lvextend -L 40G /dev/mapper/vg1-test
Extending logical volume test to 40,00 GB
Logical volume test successfully resized
root# resize2fs /dev/mapper/vg1-test
Das Dateisystem auf /dev/mapper/vg1-test ist auf /test eingehängt;
Online-Größenveränderung nötig
old desc_blocks = 2, new_desc_blocks = 3
Führe eine Online-Größenänderung von /dev/mapper/vg1-test
auf 10485760 (4k) Blöcke durch.
Das Dateisystem auf /dev/mapper/vg1-test ist nun 10485760 Blöcke groß.

```

Eine Vergrößerung des Dateisystems ist wie im obigen Beispiel im laufenden Betrieb möglich. Allerdings ist eine Vergrößerung über 16 TiB laut verschiedenen Internet-Berichten problematisch. Ich habe den Vorgang mangels eines ausreichend großen RAID-Verbunds nicht selbst testen können. Für eine Verkleinerung muss das Dateisystem vorher aus dem Verzeichnisbaum gelöst werden (umount).

Unter »Fragmentierung« versteht man den Zustand, dass einzelne Dateien nicht in aneinanderliegenden Blöcken, sondern über die ganze Partition verteilt gespeichert werden. Dazu kann es kommen, wenn abwechselnd Dateien gelöscht, neu angelegt, verlängert oder verkürzt werden. Die Fragmentierung kann den Dateizugriff erheblich verlangsamen.

Fragmentierung  
des Datei-  
systems

Die ext2/3/4-Treiber versuchen eine Fragmentierung so gut wie möglich zu vermeiden. Das gelingt allerdings nur, wenn das Dateisystem nie zu mehr als ca. 90 Prozent mit Daten gefüllt ist.

Mit dem relativ neuen Kommando `e4defrag` können Sie ein aktives Dateisystem defragmentieren. Zuerst überprüfen Sie mit `e4defrag -c` für alle Dateien, ob eine Defragmentierung überhaupt sinnvoll ist:

```
root# e4defrag -c /
<Fragmented files>                now/best size/ext
1. /var/log/fail2ban.log.1         16/1      4 KB
2. /home/kofler/Maildir/dovecot.index.log.2  9/1      4 KB
...
Total/best extents                242365/237466
Average size per extent           55 KB
Fragmentation score               1
[0-30 no problem: 31-55 a little bit fragmented: 56- needs defrag]
This directory (/) does not need defragmentation.
Done.
```

`e4defrag` zeigt die fünf am meisten defragmentierten Dateien sowie einen Score für das gesamte Dateisystem an. Im obigen Beispiel ist die Defragmentierung nicht schlimm.

Wenn Sie möchten, können Sie nun in einem zweiten Schritt entweder alle oder auch nur ausgewählte Dateien mit `e4defrag datei/verzeichnis/device defragmentieren`:

```
root# e4defrag /var/log/fail2ban.log.1
ext4 defragmentation for /var/log/fail2ban.log.1
[1/1]/var/log/fail2ban.log.1: 100% [ OK ]
Success: [1/1]
```

Sie können auch unter Windows auf Linux-Partitionen zugreifen. Eine Zusammenstellung der kostenlosen Programme bzw. Treiber finden Sie hier:

Windows-Zugriff  
auf ext4-Datei-  
systeme

[https://wiki.ubuntuusers.de/Linux-Partitionen\\_unter\\_Windows](https://wiki.ubuntuusers.de/Linux-Partitionen_unter_Windows)

Die Firma Paragon bietet einen kommerziellen ext-Treiber für Windows an. Für Privat-anwender gibt es eine kostenlose Version:

<https://paragon-software.com/home/extfs-windows>

## 22.11 Das btrfs-Dateisystem

Seit vielen Jahren heißt es, btrfs sei das Linux-Dateisystem der Zukunft. Strittig ist aber, ob btrfs auch schon tauglich für die Gegenwart ist. Die btrfs- und SUSE-Entwickler sind dieser Meinung. Sowohl openSUSE als auch die aktuellen SUSE-Enterprise-Produkte verwenden btrfs als Standarddateisystem für die Systempartition. Red Hat war 2014 anderer Meinung: RHEL 7 hat deswegen den erwarteten Umstieg auf btrfs nicht vollzogen und ist stattdessen auf xfs als Defaultdateisystem umgestiegen. Nicht einmal die experimentelle Fedora-Distribution verwendet btrfs.

Schon jetzt ist btrfs definitiv das modernste Dateisystem für Linux. Allerdings laufen die unzähligen Funktionen nach wie vor nicht restlos stabil. Ein zweites Problem besteht darin, dass die vielen btrfs-Funktionen zwar toll für Linux-Profis und Server-Administratoren sind, durchschnittliche Linux-Anwender aber oft überfordern. Selbst scheinbar triviale Fragen, nämlich wie viel Platz auf einem Dateisystem noch frei ist, sind in btrfs nur schwierig zu beantworten.

Die Entwicklung von btrfs wurde ursprünglich von Oracle initiiert, erfolgt mittlerweile aber in Kooperation mit zahlreichen anderen Firmen und Kernelentwicklern. Der btrfs-Treiber ist in den Kernel integriert und untersteht wie der gesamte Kernelcode der Lizenz GPL.

**Funktionen** Die folgende Liste fasst die wichtigsten Eigenschaften von btrfs zusammen:

- ▶ Copy on Write: Geänderte Dateiblöcke werden nicht überschrieben, sondern an einer anderen Stelle gespeichert. Das ermöglicht im Zusammenspiel mit Journaling besonders sichere Dateiänderungen.
- ▶ automatische Berechnung von Prüfsummen, um Bitfehler zu entdecken
- ▶ direkte Unterstützung von RAID-0, -1, -5, -6 und -10
- ▶ Snapshots und Subvolumes
- ▶ Komprimierung der Dateien (mount-Option `compress`)
- ▶ SSD-Optimierung (mount-Option `ssd`)
- ▶ Defragmentierung im laufenden Betrieb
- ▶ Deduplizierung, um Dateien gleichen Inhalts nur einmal zu speichern
- ▶ Send/Receive-Nachrichten, die über Änderungen im Dateisystem informieren (ermöglicht effiziente Synchronisierungs- und Backup-Tools)

Mit Subvolumes, Snapshots und RAID bietet btrfs ähnliche Funktionen wie die im Kernel schon enthaltenen *Multi Device* und *Logical Volume Manager* (siehe [Abschnitt 22.17](#), »RAID«, und [Abschnitt 22.18](#), »Logical Volume Manager (LVM)«). An sich sind solche Doppelgleisigkeiten im Kernel unerwünscht, im Falle von btrfs wurden sie aber akzeptiert. Der Grund: Einerseits ermöglicht die direkte Integration von RAID-Funktionen in den Dateisystemtreiber aufgrund der Prüfsummen eine noch höhere Datensicherheit, andererseits haben die btrfs-Entwickler glaubhaft nachweisen können, dass die btrfs-Snapshots wesentlich effizienter sind als die von LVM.

btrfs bietet im Vergleich zu ext4 eine Menge Zusatzfunktionen, ist aber nur in wenigen Fällen schneller. Generell sind Benchmark-Tests zur Beurteilung der Geschwindigkeit eines Dateisystems eine diffizile Angelegenheit. In der Vergangenheit ist btrfs aber selten als Sieger hervorgegangen. Wenn es Ihnen in erster Linie um maximale Geschwindigkeit geht, ist btrfs also nicht unbedingt die erste Wahl.

Benchmarks

Zu btrfs gibt es umfassende Informationen im Internet. Vor dem ersten Einsatz von btrfs empfehle ich Ihnen insbesondere einen Blick auf die folgenden Seiten:

Links

<https://btrfs.wiki.kernel.org/index.php/FAQ>  
<https://btrfs.wiki.kernel.org/index.php/Gotchas>  
[https://btrfs.wiki.kernel.org/index.php/Problem\\_FAQ](https://btrfs.wiki.kernel.org/index.php/Problem_FAQ)

## Administration

Die btrfs-Administration erfolgt überwiegend durch das Kommando btrfs. Dieses Kommando befindet sich zusammen mit mkfs.btrfs je nach Distribution im Paket btrfs-tools, btrfs-progs oder btrfsprogs. Wenn Sie nicht schon während der Installation ein btrfs-Dateisystem eingerichtet haben, müssen Sie dieses Paket in der Regel extra installieren.

Um ein neues btrfs-Dateisystem in einer leeren Partition bzw. einem leeren Logical Volume einzurichten, führen Sie das folgende Kommando aus (wobei Sie natürlich /dev/sdb1 durch Ihren eigenen Device-Namen ersetzen müssen):

btrfs-Dateisystem einrichten

```
root# mkfs.btrfs /dev/sdb1
btrfs-progs v4.5.3+
fs created label (null) on /dev/sdb1
        nodesize 4096 leafsize 4096 sectorsize 4096 size 20.00GB
```

Anschließend binden Sie das Dateisystem in den Verzeichnisbaum ein:

```
root# mkdir /media/btrfs
root# mount /dev/sdb1 /media/btrfs
```

**Dateisystem  
vergrößern/  
verkleinern**

Wenn sich ein `btrfs`-Dateisystem als zu klein herausstellt, ist es am einfachsten, ein weiteres Device (also eine Festplattenpartition oder ein Logical Device) hinzuzufügen. Details dazu folgen im Abschnitt »`btrfs`-Dateisysteme über mehrere Devices verteilen, RAID«.

Es ist aber auch möglich, die Größe eines vorhandenen `btrfs`-Dateisystems im laufenden Betrieb zu erhöhen und sogar zu verringern! In der Praxis funktioniert das am besten, wenn sich das Dateisystem in einem Logical Volume befindet.

Um ein `btrfs`-Dateisystem so zu vergrößern, dass es ein zuvor mit `lvextend` vergrößertes Logical Volume komplett nutzt, führen Sie das folgende Kommando aus:

```
root# btrfs filesystem resize max /mnt-point
```

Statt `max` können Sie auch die neue absolute Größe des Dateisystems angeben oder mit `+` oder `-` die relative Änderung. Dabei sind die Kürzel `k`, `m` und `g` für KiB, MiB und GiB erlaubt. Das folgende Kommando verkleinert das Dateisystem um 2 GiB:

```
root# btrfs filesystem resize -2g /mnt-point
```

**Dateisystem  
überprüfen**

Zur Überprüfung der Konsistenz des Dateisystems gibt es das Kommando `btrfs check`. Es kann nur für nicht eingebundene Dateisysteme ausgeführt werden. Als einzigen Parameter übergeben Sie den Device-Namen der Partition mit dem `btrfs`-Dateisystem:

```
root# btrfs check /dev/sdb1
```

`btrfs check` analysiert das Dateisystem, führt aber keine Änderungen durch. Wenn Sie versuchen möchten, Fehler im Dateisystem zu beheben, müssen Sie zusätzlich die Option `--repair` angeben.

**COW deaktivieren**

`btrfs` basiert auf dem Copy-on-Write-Verfahren (COW): Wenn eine Datei verändert wird, werden diese Veränderungen in leeren Blöcken des Dateisystems gespeichert. Die ursprünglichen Blöcke werden später freigegeben.

In den meisten Fällen ist das ein vernünftiger und sicherer Ansatz, aber in Ausnahmefällen führt COW zu einer unnötigen Fragmentierung der Datei und dadurch zu einer schlechteren Performance. Das betrifft insbesondere Image-Dateien von Virtualisierungsprogrammen sowie binäre Datenbank-Dateien – also sehr große Dateien, in denen häufig blockweise Änderungen durchgeführt werden müssen.

Ich empfehle Ihnen, bei Datenbank- und Virtualisierungsdateien `btrfs` gleich ganz aus dem Weg zu gehen und für das betreffende Verzeichnis besser ein eigenes `ext4`- oder `xfs`-Dateisystem zu verwenden. Wenn Sie die Datenbanken oder Images aber



innerhalb von `btrfs` speichern möchten, dann können Sie für diese Dateien oder auch für ganze Verzeichnisse mit `chattr` das Attribut `C` setzen:

```
root# touch datei
root# chattr +C datei
root# chattr +C verzeichnis
```

`chattr +C` deaktiviert COW für eine Datei bzw. ein ganzes Verzeichnis. Das Attribut gilt allerdings nur für neue Dateien. Um COW bei bereits existierenden Dateien mit einer Größe ungleich 0 Byte zu deaktivieren, müssen Sie die Dateien kopieren:

```
root# mv verz backup
root# mkdir verz
root# chattr +C verz
root# cp -a backup/* verz
root# rm -rf backup
```

### Dateien komprimieren

`btrfs` unterstützt die automatische und transparente Komprimierung von Dateien. Dazu muss das Dateisystem mit der `mount`-Option `compress=zlib` oder `compress=lzo` in den Verzeichnisbaum eingebunden werden. Die Option `compress` gilt nur für neue bzw. geänderte Dateien. Vorhandene Dateien bleiben unverändert, solange sie nur gelesen werden. Die Option gilt für das gesamte Dateisystem, kann also nicht nur für einzelne Verzeichnisse aktiviert werden. Die Komprimierung erfolgt wahlweise durch den besser komprimierenden `zlib`-Algorithmus oder den schnelleren, aber weniger platzsparenden `lzo`-Algorithmus.

`compress` kann bei herkömmlichen Festplatten Dateioperationen beschleunigen. Das mag auf den ersten Blick verwundern, weil die Kompression bzw. Dekompression beim Lesen ja zusätzlichen Aufwand verursacht. Bei einer schnellen CPU ist dieser Aufwand aber gering im Vergleich zu der Ersparnis, die sich dadurch ergibt, dass weniger Datenblöcke der Festplatte oder der SSD gelesen bzw. verändert werden müssen. Als Zusatznutzen kommt natürlich die Platzersparnis hinzu.

Sie können das komprimierte Dateisystem später selbstverständlich auch ohne die Option `compress` nutzen. Neue bzw. veränderte Dateien sind dann nicht mehr komprimiert, bereits vorhandene Dateien bleiben aber komprimiert, solange die Dateien nur gelesen werden.

Die `compress`-Option eignet sich besonders gut für Verzeichnisse, die viele Textdateien enthalten (z. B. `/usr/`, die Platzersparnis beträgt hier fast 50 Prozent!). Nicht empfehlenswert ist die Option hingegen für Ihr Benutzerverzeichnis, wenn sich dort überwiegend bereits komprimierte Dateien befinden, also z. B. Audio-, Video-, PDF- und LibreOffice-Dateien. Eine weitere Komprimierung gelingt dann nicht. Das

erkennt auch der `btrfs`-Treiber und verzichtet bei der betreffenden Datei auf die Komprimierung. Dennoch kostet dieser Test etwas Zeit.

In der Praxis ist es bei Desktop-Systemen zweckmäßig, für die Systempartition `compress` zu verwenden, für die Home-Partition aber häufig nicht. Leider können Sie nicht bei jeder Distribution die `mount`-Optionen bereits bei der Installation einstellen.

Natürlich gibt es Sonderfälle: Wenn Sie z. B. einen MySQL-Datenbank-Server betreiben und dabei den InnoDB-Tabellentreiber einsetzen, der die Tabellen automatisch komprimiert, sollte das MySQL-Datenbankverzeichnis `/var/lib/mysql` nicht in einem `btrfs`-Dateisystem mit `compress`-Option liegen.

## Subvolumes

Von herkömmlichen Dateisystemen kennen Sie die Regel »Eine Partition bzw. ein Logical Volume – ein Dateisystem«. Bei `btrfs` ist das anders: Ein Subvolume ist ein eigenes virtuelles Dateisystem. Innerhalb des »realen« `btrfs`-Dateisystems kann es beliebig viele Subvolumes geben.

Subvolumes können wie Verzeichnisse behandelt werden, sie können aber auch wie Dateisysteme an einem beliebigen Ort in den Verzeichnisbaum eingebettet werden. Subvolumes bilden entsprechend der Verzeichnisstruktur eine Hierarchie. An der Spitze steht das Default-Subvolume, das beim Einrichten eines `btrfs`-Dateisystems automatisch angelegt wird.

### Subvolumes erzeugen

Am einfachsten ist das anhand eines Beispiels zu verstehen. Dabei gehe ich davon aus, dass sich das `btrfs`-Dateisystem in der Partition `/dev/sdb1` befindet und im Verzeichnis `/media/btrfs` eingebunden ist. `btrfs subvolume create` erzeugt nun zwei neue Subvolumes: `sub1` und `data/sub2`.

```
root# btrfs subvolume create /media/btrfs/sub1
root# mkdir /media/btrfs/data
root# btrfs subvolume create /media/btrfs/data/sub2
```

Mit dem Kommando `btrfs subvolume list` können Sie sich einen Überblick über alle Subvolumes und Snapshots Ihres `openSUSE`-Systems verschaffen. Die drei Optionen `-a -p -t` bewirken, dass der Ort des Subvolumes innerhalb der Verzeichnishierarchie exakt angegeben wird, dass zu jedem Subvolume bzw. Snapshot auch das Parent-Subvolume angegeben und das Ergebnis als Tabelle formatiert wird.

```
root# btrfs subvolume list /media/btrfs/ -a -p -t
ID      gen    top level    path
--      ---    -
259     10     5            sub1
260     11     5            data/sub2
```

Beim Erzeugen eines neuen btrfs-Dateisystems wird automatisch ein Default-Subvolume für das Wurzelverzeichnis eingerichtet. Es hat immer die ID 5 und wird nicht extra aufgelistet. Sie sehen aber im obigen Listing, dass die neuen Subvolumes innerhalb dieses Default-Subvolumes erzeugt wurden.

Subvolumes verhalten sich beinahe wie gewöhnliche Verzeichnisse. Sie können darin Dateien anlegen und wieder löschen. Es ist allerdings nicht zulässig, das Subvolume-Verzeichnis selbst zu löschen, auch dann nicht, wenn es leer ist. Die einzige Möglichkeit, das Verzeichnis `sub1` zu löschen, bietet das Kommando `btrfs subvolume delete`, das ich weiter unten beschreibe.

```
root# touch /media/btrfs/sub1/tst
root# rm /media/btrfs/sub1/tst
root# rmdir /media/btrfs/sub1
rmdir: konnte /media/btrfs/sub1 nicht entfernen: Vorgang nicht zulässig
```

Sie können Subvolumes wie eigene Dateisysteme an einem beliebigen Ort im Verzeichnisbaum einbinden. Dazu verwenden Sie `mount` mit der Option `subvolid=n`:

```
root# mkdir /media/sub2
root# mount -o subvolid=260 /dev/sdb1 /media/sub2
```

Das explizite Ausführen von `mount` ist zur Nutzung von Subvolumes erforderlich, die in einem gerade nicht aktiven Subvolume erzeugt wurden (also außerhalb des Default-Subvolumes mit ID=5).

Mit `btrfs subvolume set-default` können Sie das Subvolume festlegen, das beim nächsten `mount`-Kommando standardmäßig verwendet wird, wenn nicht mit der `mount`-Optionen `subvolid` explizit ein anderes Subvolume ausgewählt wird. An `set-default` müssen Sie die Volume-ID übergeben, die Sie vorher mit `btrfs subvolume list` ermitteln.

**Default-Subvolume festlegen**

Mit `btrfs subvolume delete name` löschen Sie ein Subvolume inklusive aller darin enthaltenen Dateien. Das Subvolume muss vorher natürlich aus dem Verzeichnisbaum gelöst werden.

**Subvolumes löschen**

```
root# umount /media/sub2
root# btrfs subvolume delete /media/btrfs/data/sub2
```

Beachten Sie, dass der von Subvolumes beanspruchte Speicher mit der Ausführung von `btrfs subvolume delete` nicht sofort freigegeben wird, sondern erst nach und nach. Ein Kernelprozess kümmert sich im Hintergrund um die erforderlichen Aufräumarbeiten.

## Snapshots

Ein Snapshot ist anfänglich eine virtuelle Kopie eines Subvolumes. »Virtuell« deswegen, weil beim Anlegen eines Snapshots keinerlei Daten kopiert werden. Erst wenn das ursprüngliche Subvolume oder der Snapshot verändert werden, bilden sich zwei getrennte Zweige. Snapshots betreffen immer nur unmittelbar das Basis-Subvolume, nicht aber eventuell vorhandene Unter-Subvolumes (Sub-Subvolumes).

Oft werden Snapshots als eine Art Backup des betreffenden Subvolumes erzeugt. In der Folge wird das Subvolume dann weiterverwendet und verändert, während der Snapshot nur dazu dient, den bisherigen Zustand zu archivieren oder seinen Inhalt einem Backup-Programm zur Verfügung zu stellen. Das ist aber nur eine mögliche Vorgehensweise. `btrfs`-intern ist es durchaus zulässig, Snapshots zu verändern! Das ursprüngliche Subvolume und der daraus erzeugte Snapshot sind zwei Zweige eines Startzustands. (`btrfs` kennt aber auch Read-Only-Snapshots.)

`btrfs`-intern werden Snapshots wie Subvolumes behandelt. Deswegen gelten die meisten `subvolume`-Befehle von `btrfs` gleichermaßen für Subvolumes und Snapshots. Der wesentliche Unterschied zwischen Subvolumes und Snapshots besteht darin, dass Subvolumes anfänglich leer sind, Snapshots dagegen eine virtuelle Kopie des Ausgangsverzeichnisses enthalten.

**btrfs versus LVM** Der Begriff »Snapshot« wird in `btrfs` und LVM (siehe [Abschnitt 22.18](#)) vollkommen unterschiedlich verwendet. In LVM ist ein Snapshot ein unveränderliches Abbild eines Logical Volumes (LV). Sie müssen beim Erzeugen des Snapshots angeben, wie viel Speicherplatz der Snapshot maximal beanspruchen darf. Dieser Speicherplatz dient dazu, bei Bedarf Datenblöcke des ursprünglichen LVs zu archivieren, bevor diese geändert werden. Wenn der Snapshot-Speicherplatz aufgebraucht ist, wird der Snapshot ungültig und kann nicht mehr verwendet werden.

Bei `btrfs` ist der Inhalt des Snapshots dagegen veränderlich. Ein Snapshot ist also ein neuer Zweig eines Verzeichnisbaums, der zum Zeitpunkt der Erstellung des Snapshots eins zu eins mit diesem identisch ist. Ab diesem Zeitpunkt können sich beide Zweige unabhängig voneinander weiterentwickeln. Die beiden Zweige beanspruchen umso mehr Speicherplatz, je mehr Dateien geändert werden. `btrfs` verwendet zur Speicherung der Änderungen einfach den `btrfs`-Speicherpool. Das funktioniert so lange, bis die Kapazität des gesamten Dateisystems erschöpft ist. `btrfs`-Snapshots bieten somit wesentlich mehr Funktionen und Flexibilität als LVM-Snapshots!

**Beispiel** Als Ausgangspunkt für das folgende Beispiel dient ein `btrfs`-Dateisystem in der Partition `/dev/sdb1`. Das Dateisystem ist an der Stelle `/media/btrfs` in den Verzeichnisbaum integriert. `btrfs subvolume snap` erzeugt nun einen Snapshot vom gesamten Dateisystem. Dabei wird zugleich das Verzeichnis `/media/btrfs/snap1` erzeugt. Der Snapshot

kann über dieses Verzeichnis verwendet werden oder mit `mount` wie ein eigenes Dateisystem in den Verzeichnisbaum eingebunden werden. Dabei muss die vom vorigen Abschnitt schon bekannte `mount`-Option `subvol=name` verwendet werden:

```
root# btrfs subvolume snapshot /media/btrfs/ /media/btrfs/snap1
root# mkdir /media/snap1
root# mount -o subvol=snap1 /dev/sdb1 /media/snap1/
```

Sie können nun sowohl im ursprünglichen Dateisystem als auch im Snapshot unabhängig voneinander (also ohne gegenseitige Beeinflussung) Dateien anlegen, verändern und löschen.

`btrfs`-Snapshots sind normalerweise veränderlich. Wenn Sie für Backups einen Read-only-Snapshot wünschen, führen Sie `btrfs subvolume snapshot` mit der Option `-r` aus.

### btrfs-Dateisysteme über mehrere Devices verteilen, RAID

Der `btrfs`-Treiber kann Dateisysteme über mehrere Festplatten bzw. Devices verteilen und unterstützt dabei die RAID-Level 0, 1, 5, 6 und 10, ohne auf den sonst üblichen Linux-RAID-Treiber `mdadm` zurückzugreifen.

Der einfachste Fall von Multi-Device-Dateisystemen entsteht zumeist dann, wenn ein `btrfs`-Dateisystem zu klein wird: Sie können nun ganz einfach ein weiteres Device hinzufügen (also eine leere Festplattenpartition oder ein ungenutztes Logical Volume). Damit wird das Dateisystem entsprechend vergrößert. Sie müssen weder eine Partition neu formatieren noch die Größe des Dateisystems explizit ändern – `btrfs` erledigt all diese Aufgaben selbstständig.

Device  
hinzufügen

```
root# btrfs device add /dev/sdb2 /media/btrfs
```

Anfänglich befinden sich nun alle Daten auf dem ersten Device, während das zweite Device erst nach und nach genutzt wird. Falls sich die Devices auf unterschiedlichen physikalischen Festplatten befinden (und nur dann!), erzielen Sie einen Geschwindigkeitsgewinn, wenn Sie die vorhandenen Dateien mit `btrfs filesystem balance` über alle Devices verteilen. Beachten Sie aber, dass `btrfs filesystem balance` sehr lange dauert und nur selten der Mühe wert ist. Neue bzw. geänderte Dateien verteilen sich ohnedies automatisch über beide Devices.

```
root# btrfs filesystem balance /media/btrfs
```

Es ist auch möglich, ein Device wieder zu entfernen. Die auf dem Device enthaltenen Daten werden dann zuerst auf die anderen Devices übertragen, weswegen die Ausführung des folgenden Kommandos sehr lange dauern kann:

```
root# btrfs device delete /dev/sdb1 /media/btrfs/
```

**mkfs.btrfs mit mehreren Devices**

Sie können ein btrfs-Dateisystem auch von vornherein mit mehreren Devices einrichten, indem Sie an `mkfs.btrfs` mehrere Devices übergeben:

```
root# mkfs.btrfs /dev/sdb1 /dev/sdc1
```

Standardmäßig werden dann die Metadaten des Dateisystems dupliziert (entspricht RAID-1), die eigentlichen Daten aber über alle Devices verteilt. Die Metadaten enthalten die Verwaltungsinformationen des Dateisystems, also z. B. Inode-Listen sowie Bäume zum Suchen nach Dateien. Leider ist beim resultierenden Dateisystem nun weder die Geschwindigkeit optimal (es ist langsamer als ein RAID-0-System wegen der Duplizierung der Metadaten) noch kann es sicherheitstechnisch mit RAID-1 mithalten, weil die eigentlichen Daten nicht redundant gespeichert werden.

Bei der Ausführung von `mount` geben Sie ein beliebiges Device des Dateisystems an. (Nach einem Rechnerneustart muss das Kommando `btrfs device scan` ausgeführt werden, damit btrfs weiß, welche Devices mit btrfs-Dateisystemen es gibt und wie sie zusammengehören.)

```
root# mount -t /dev/sdb1 /media/btrfs
```

Welche RAID-Variante für welche Art von Daten verwendet wird, finden Sie mit `btrfs filesystem df` heraus:

```
root# mkfs.btrfs /dev/sdb1 /dev/sdc1
root# mount /dev/sdb1 /media/btrfs
root# cp -a /etc /media/btrfs
root# btrfs filesystem df /media/btrfs
Data,      RAID0: total=1.80GiB, used=7.07MiB
System,    RAID1: total=8.00MiB, used=16.00KiB
Metadata,  RAID1: total=1.00GiB, used=3.30MiB
...
```

Diese (etwas gekürzte) Ausgabe bedeutet, dass die Daten über beide Devices verteilt werden (RAID-0), dass aber dateisysteminterne Daten (System und Metadata) gespiegelt werden. Sollte auf einem der beiden Datenträger ein Hardware-Fehler auftreten, stehen zumindest von allen Metadaten Duplikate zur Verfügung.

**RAID-0** Wenn Sie ein »richtiges« RAID-System anlegen möchten, bei dem Daten und Metadaten einheitlich behandelt werden, übergeben Sie an `mkfs.btrfs` mit `-d` (für die Daten) und `-m` (für die Metadaten) den gewünschten RAID-Level. Außerdem übergeben Sie an `mkfs.btrfs` die gewünschte Anzahl von Devices. Das folgende Kommando erstellt ein RAID-0-System (Striping):

```
root# mkfs.btrfs -d raid0 -m raid0 /dev/sdb1 /dev/sdc1
```

**RAID-1** Ein RAID-1-Dateisystem wird analog mit diesem Kommando eingerichtet:

```
root# mkfs.btrfs -d raid1 -m raid1 /dev/sdb1 /dev/sdc1
```

Interessant wird es, wenn ein Device ausfällt. Um diesen Fall zu testen, habe ich die Festplatte `/dev/sdc` entfernt. Damit das Dateisystem verwendet werden kann, muss nun die zusätzliche `mount`-Option `degraded` verwendet werden:

```
root# mount -o degraded /dev/sdb1 /media/btrfs
```

Um den RAID-Verbund wiederherzustellen, fügen Sie dem Dateisystem ein neues, möglichst gleich großes Device wieder hinzu. Im folgenden Beispiel ist das wieder `/dev/sdc1`, wobei diese Partition nun aber von einer neuen Festplatte stammt. Um das Dateisystem wieder über beide Devices zu verteilen und somit die RAID-1-Redundanz wiederherzustellen, müssen Sie außerdem `btrfs filesystem balance` ausführen. Bei großen Dateisystemen dauert die Ausführung dieses Kommandos naturgemäß sehr lange. Immerhin kann das Dateisystem in dieser Zeit genutzt werden, wenn auch mit stark verminderter Geschwindigkeit.

```
root# btrfs device add /dev/sdc1 /media/btrfs
root# btrfs filesystem balance /media/btrfs
```

Erst jetzt kann das defekte Device aus dem Dateisystem entfernt werden. Dabei verwenden Sie zur Device-Angabe das Schlüsselwort `missing`:

```
root# btrfs device delete missing /media/btrfs
```

Beim nächsten `mount`-Kommando können Sie nun auf die Option `degraded` verzichten.

### Die Nutzung eines btrfs-Dateisystems ermitteln (df)

Bei anderen Dateisystemen können Sie ganz einfach mit `df -h` feststellen, wie viel Speicherplatz vorhanden ist, wie viel davon belegt ist und wie viel noch frei ist. Bei `btrfs`-Dateisystemen liefert `df` aber mitunter vollkommen irreführende Ergebnisse, insbesondere im Zusammenhang mit RAID. Gründe dafür sind die Trennung zwischen Systemdaten, Metadaten und den eigentlichen Dateien, die Verwendung unterschiedlicher RAID-Level für Daten und Metadaten, eventuell aktive Komprimierfunktionen sowie die Verwendung von Pools, die `btrfs` vorweg für verschiedene Verwendungszwecke reserviert.

Ergänzend zu `df` können Sie auch die `btrfs`-Kommandos `filesystem show` und `filesystem df` ausführen. Das folgende Beispiel soll Ihnen dabei helfen, zumindest die Daten korrekt zu interpretieren, die `btrfs` liefert. Als Ausgangspunkt dient ein kleines `btrfs`-RAID-1-System aus zwei je 8 GiB großen Partitionen. In dieses Dateisystem wurde das gesamte `/usr`-Verzeichnis des Testsystems kopiert (Platzbedarf laut `du` ca. 3,2 GiB).

```
root# mkfs.btrfs -d raid1 -m raid1 /dev/sdb1 /dev/sdc1
root# mount /dev/sdb1 /media/btrfs
root# cp -a /usr /media/btrfs
```

- df** Das Ergebnis von `df` ist hier trotz der Verwendung von RAID-1 nahezu perfekt. Das ist ein großer Fortschritt gegenüber älteren `btrfs`-Versionen, wo `df` im gleichen Szenario meilenweit daneben lag.

```
root# df -h /media/btrfs/
Dateisystem Größe Benutzt Verf. Verw% Eingehängt auf
/dev/sdb1 7,9G 3,5G 3,7G 49% /media/btrfs
```

- filesystem show** `btrfs filesystem show` verrät, dass das Dateisystem aus zwei jeweils 8 GiB großen Devices zusammengesetzt ist und dass beide Devices im gleichen Ausmaß gefüllt sind:

```
root# btrfs filesystem show /dev/sdb1
Label: none uuid: bc942951-41a9-43e7-9d6c-ecee1717ec23
Total devices 2 FS bytes used 3.39GiB
devid 1 size 7.81GiB used 5.03GiB path /dev/sdb1
devid 2 size 7.81GiB used 5.01GiB path /dev/sdc1
```

- filesystem df** `btrfs filesystem df` gibt Auskunft darüber, wie die reservierten Daten verwendet werden. Wirklich relevant sind hierbei die drei Zeilen mit dem Attribut `RAID1`: `btrfs` hat bisher 4 GiB für die eigentlichen Daten reserviert und davon ca. 3,2 GiB tatsächlich genutzt. Weiters hat `btrfs` 1 GiB für Metadaten reserviert und davon rund 220 MiB genutzt. Schließlich hat `btrfs` 8 MiB für Systemdaten reserviert und gerade einmal 16 KiB davon tatsächlich genutzt. Schließlich sind weitere 100 MiB für andere Zwecke reserviert (die Summe aller `single`-Zeilen).

```
root# btrfs filesystem df /media/btrfs/
Data, RAID1:          total=4.00GiB, used=3.17GiB
Data, single:         total=8.00MiB, used=0.00B
System, RAID1:        total=8.00MiB, used=16.00KiB
System, single:       total=4.00MiB, used=0.00B
Metadata, RAID1:      total=1.00GiB, used=219.75MiB
Metadata, single:     total=8.00MiB, used=0.00B
GlobalReserve, single: total=80.00MiB, used=0.00B
```

Wie viel Speicherplatz ist nun wirklich frei? Im Data-Pool sind nur ca.  $4 - 3,17 = 0,83$  GiB frei. Außerdem gibt es auf den Devices jeweils noch ca. 3 GiB unreservierten Speicherplatz. Wenn dieser Platz zur Gänze für Daten genutzt werden kann, dann würde sich 3,83 GiB freier Speicherplatz ergeben, was in etwa dem Ergebnis von `df` entspricht. Sollte `btrfs` aber mehr Platz für Metadaten benötigen, dann sinkt der für Dateien nutzbare Platz.



### Disk-full-Probleme

In der btrfs-Mailing-Liste gibt es regelmäßig Berichte, dass btrfs Platzprobleme feststellt, obwohl die Datenträger eigentlich noch lange nicht voll sind. Daher die Empfehlung: Dimensionieren Sie btrfs-Systeme möglichst großzügig. ext4-Dateisysteme können Sie beinahe bis auf das letzte Byte vollschreiben (auch wenn dies nicht empfehlenswert ist). Bei btrfs sind ordentliche Platzreserven hingegen zwingend notwendig!

### btrfs-Konfiguration in openSUSE

Keine andere Linux-Distribution nutzt btrfs standardmäßig so intensiv wie SUSE. Aktuelle Versionen von openSUSE bzw. der SUSE-Enterprise-Distributionen verwenden btrfs für die Systempartition sowie xfs für die Home-Partition. Innerhalb der Systempartition legt YaST während der Installation diverse Subvolumes für Verzeichnisse wie `/var/xxx`, `/opt` und `/srv` an. Außerdem werden bei administrativen Arbeiten durch YaST je zwei Snapshots angelegt: ein pre-Snapshot (vorher) und ein dazugehöriger pre-post-Snapshot (nachher).

Der Nutzen dieser Konfiguration besteht darin, dass Sie im YaST-Modul SNAPPER (siehe [Abbildung 22.3](#)) Ihr System in einen älteren Zustand zurückversetzen können, ohne dabei auch mit Ihren persönlichen Daten (`/home`-Verzeichnis) einen Sprung in die Vergangenheit zu machen. Alternativ können Sie auch nur einzelne Dateien zurückstellen. Sollte es Boot-Probleme geben, können Sie in GRUB sogar einen älteren Snapshot auswählen und das System in diesem Zustand starten.

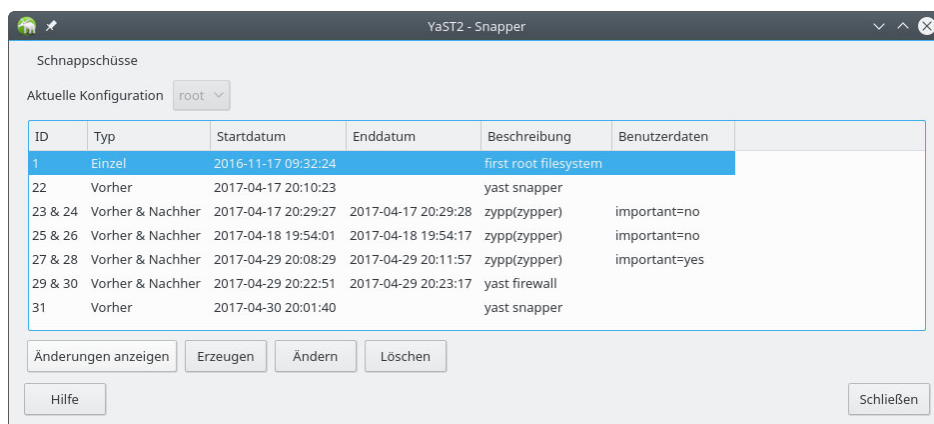


Abbildung 22.3 Verwaltung von Snapshots in YaST

Der Preis für diese Sicherheit vor administrativen Fehlern ist ein btrfs-System, in dem einem vor lauter Subvolumes und Snapshots schwindelig werden kann:

```

root# btrfs subvol list / -p -a -t
ID      gen      parent  top level  path
--      ---      -
257     107      5        5          <FS_TREE>/@
258     3927     257      257        <FS_TREE>/@/.snapshots
259     3929     258      258        <FS_TREE>/@/.snapshots/1/snapshot
260     3688     257      257        <FS_TREE>/@/boot/grub2/i386-pc
261     107      257      257        <FS_TREE>/@/boot/grub2/x86_64-efi
262     3929     257      257        <FS_TREE>/@/opt
263     3771     257      257        <FS_TREE>/@/srv
264     3927     257      257        <FS_TREE>/@/tmp
265     3693     257      257        <FS_TREE>/@/usr/local
266     3849     257      257        <FS_TREE>/@/var/cache
267     107      257      257        <FS_TREE>/@/var/crash
268     1438     257      257        <FS_TREE>/@/var/lib/libvirt/images
269     107      257      257        <FS_TREE>/@/var/lib/machines
270     107      257      257        <FS_TREE>/@/var/lib/mailman
271     107      257      257        <FS_TREE>/@/var/lib/mariadb
272     3803     257      257        <FS_TREE>/@/var/lib/mysql
273     107      257      257        <FS_TREE>/@/var/lib/named
274     107      257      257        <FS_TREE>/@/var/lib/pgsql
275     3929     257      257        <FS_TREE>/@/var/log
276     107      257      257        <FS_TREE>/@/var/opt
277     3928     257      257        <FS_TREE>/@/var/spool
278     3849     257      257        <FS_TREE>/@/var/tmp
318     3784     258      258        <FS_TREE>/@/.snapshots/22/snapshot
...
327     3925     258      258        <FS_TREE>/@/.snapshots/31/snapshot

```

#### Konfiguration und Administration

Die Konfigurationsdateien für das Snapper-System befinden sich in `/etc/snapper`. Aus `/etc/snapper/configs/root` geht hervor, dass standardmäßig Snapshots für die letzten zehn YaST-Aktionen gesichert werden. Ältere Snapshots werden automatisch gelöscht. Optional können Sie in dieser Datei auch stündliche Snapshots aktivieren, wobei auch hier das automatische Löschen älterer Snapshots vorgesehen ist.

Anstelle von YaST können Sie zur Snapper-Konfiguration auch das gleichnamige Kommando verwenden:

- ▶ `snapper list` listet alle verfügbaren Snapshots auf.
- ▶ `snapper create` erzeugt einen neuen Snapshot.
- ▶ `snapper delete n` löscht den angegebenen Snapshot.
- ▶ `snapper delete n1-n2` löscht die Snapshots mit den Nummern zwischen `n1` und `n2`.
- ▶ `snapper cleanup typ` löscht alte Snapshots, wobei verschiedene Aufräumalgorithmen zur Wahl stehen: `number`, `timeline` oder `empty-pre-post`.

- ▶ `snapper undochange n1..n2 [files]` macht die zwischen dem Snapshot `n1` und dem Snapshot `n2` durchgeführten Änderungen rückgängig – wahlweise für alle betroffenen Dateien oder nur für die als Parameter angegebenen Dateien.
- ▶ `snapper rollback [n]` macht den angegebenen bzw. den aktuellsten verfügbaren Snapshot zum neuen Default-Subvolume.

Mit der Frage, wie viel Speicherplatz in einem btrfs-Dateisystem noch frei ist, haben wir uns bereits beschäftigt. Vielleicht wollen Sie aber auch wissen, wie viel Speicherplatz ein einzelnes Subvolume bzw. ein Snapshot beansprucht. Um diese Frage zu beantworten, müssen Sie zuerst die Quota-Funktionen von btrfs aktivieren:

Platzbedarf von Snapshots und Subvolumes ermitteln

```
root# btrfs quota enable /
```

Nach einer Weile (in der Regel nach einigen Minuten) verrät `btrfs qgroup show` dann, welches Subvolume bzw. welcher Snapshot wie viel Speicherplatz beansprucht:

```
root# btrfs qgroup show -e --human-readable --sort=qgroupid /
qgroupid      rfer          excl          max_excl
-----      -
0/5           16.00KiB      16.00KiB      none
0/257         16.00KiB      16.00KiB      none
0/258         288.00KiB     288.00KiB     none
0/259         5.64GiB      2.83MiB       none
0/260         2.38MiB      2.38MiB       none
...
```

Leider ist das Ergebnis in dieser Form nur schwer lesbar: Zum einen werden die Subvolumes bzw. Snapshots nicht durch ihre Namen, sondern nur durch ihre ID-Nummern bezeichnet. Zum anderen bin ich Ihnen noch die Erklärung schuldig, worin der Unterschied zwischen der ersten und der zweiten Spalte besteht: Die erste Spalte gibt den gesamten Speicherbedarf eines Subvolumes/Snapshots an. Die zweite Spalte gibt an, in welchem Ausmaß dieser Speicher exklusiv vom betreffenden Subvolume/Snapshot beansprucht wird. Dieser Speicher wird also nicht mit anderen Subvolumes/Snapshots geteilt.

Zur Deaktivierung der automatischen Snapshots sind drei Maßnahmen erforderlich:

Snapper deaktivieren

- ▶ **YaST-Snapshots bei Administrationsaufgaben:** Für die Erzeugung dieser Snapshots ist YaST verantwortlich. Dementsprechend gibt es hierfür eine Option in der YaST-Konfigurationsdatei, die auf `yes` oder `no` gestellt wird:

```
# Datei /etc/sysconfig/yast2
USE_SNAPPER="no"
```

- ▶ **Snapshots bei Paketinstallationen (Zypper):** Für die Erstellung von Snapshots bei Software-Installationen oder `-Updates` ist ein Zypper-Plugin verantwortlich. Dieses müssen Sie deinstallieren:

```
root# zypper remove snapper-zypp-plugin
```

- ▶ **Stündliche Snapshots:** Stündliche Snapshots sind unter openSUSE standardmäßig ausgeschaltet. Zur Aktivierung bzw. Deaktivierung stellen Sie einfach die Variable `TIMELINE_CREATE` auf `yes` oder `no`:

```
# Datei /etc/snapper/configs/root
TIMELINE_CREATE="no"
```

**Platz schaffen** Wenn das Root-Dateisystem schon aus allen Nähten platzt, löschen Sie mit `snapper delete` oder in YaST nicht mehr benötigte Snapshots und räumen mit `zypper clean` den Cache des Paketverwaltungssystems auf.

### Empfehlungen für SUSE-Installationen

Privaten Anwendern von openSUSE empfehle ich, bei der Installation anstelle von `btrfs` das Dateisystem `ext4` für die Systempartition zu verwenden. Die mit `btrfs` verbundenen Vorteile sind mit einer derartigen Komplexität und für Laien oft unvorhersehbaren Nebenwirkungen verbunden, dass bei Problemen einzig `btrfs`-Experten weiterhelfen können.

Persönlich würde ich vorerst auch im professionellen Einsatz einen Bogen um `btrfs` machen – aber hier kann man zugegebenermaßen auch anderer Meinung sein. Wenn Sie sich also für `btrfs` entscheiden, dann dimensionieren Sie die Systempartition auf jeden Fall doppelt so groß wie bei herkömmlichen Installationen!

## 22.12 Das xfs-Dateisystem

Das `xfs`-Dateisystem wurde ehemals von der Firma SGI für deren Workstations mit dem Unix-ähnlichen Betriebssystem IRIX entwickelt. Später wurde das Dateisystem für Linux portiert und schrittweise immer wieder verbessert. Das Dateisystem gilt als ausgereift, stabil und vor allem im Umgang mit sehr großen Dateien als effizient. Das gilt auch für Dateisysteme, die mehr als 16 TiB umfassen.

Seit 2014 wird `xfs` sowohl von RHEL 7 bzw. CentOS 7 als auch von aktuellen SUSE-Distributionen als Standarddateisystem verwendet. (SUSE differenziert zwischen der Systempartition und den Datenpartitionen: Für die Systempartition ist `btrfs` vorgesehen, für die Datenpartitionen `xfs`.)

Weitere Informationen zum `xfs`-Dateisystem sowie über die Integration diverser neuer Features ab der Kernel-Version 4.8 finden Sie hier:

<https://en.wikipedia.org/wiki/XFS>

[http://xfs.org/index.php/XFS\\_FAQ](http://xfs.org/index.php/XFS_FAQ)

<http://heise.de/-3351436>

[https://blogs.oracle.com/linuxkernel/entry/upcoming\\_xfs\\_work\\_in\\_linux](https://blogs.oracle.com/linuxkernel/entry/upcoming_xfs_work_in_linux)

### XFS-Eigenheiten

Bei der Nutzung von xfs müssen Sie zwei Besonderheiten beachten: Zum einen kann ein xfs-Dateisystem mit `xfs_growfs` vergrößert, aber nicht wieder verkleinert werden.

Zum anderen nutzt das Dateisystem die Partition vom ersten Byte an und lässt im Gegensatz zu den anderen Linux-Dateisystemen keinen Platz für einen Boot-Sektor. Deshalb zerstört die Installation von GRUB in den Boot-Sektor einer xfs-Partition Teile des Dateisystems! Bei BIOS-Rechnern dürfen Sie GRUB ausschließlich in den Master-Boot-Sektor der Festplatte installieren. Bei EFI-Rechnern stellt sich diese Frage nicht; dort werden die GRUB-Dateien ohnedies in der EFI-Partition gespeichert.

Einträge für ein xfs-Dateisystem in `/etc/fstab` sehen üblicherweise wie im folgenden Beispiel aus. Zusätzliche `mount`-Optionen werden nur ganz selten benötigt. Sie sind in man `mount` verzeichnet. `/etc/fstab`

```
# /etc/fstab
/dev/sdb13 /data xfs defaults 0 0
```

Um in einer Partition ein xfs-Dateisystem einzurichten, führen Sie einfach `mkfs.xfs` aus. Falls das Kommando nicht zur Verfügung steht, installieren Sie vorher das Paket `xfsprogs`. Die von `mkfs.xfs` ausgegebenen Parameter können Sie bei Bedarf später mit `xfs_info` neuerlich auslesen. **xfs-Dateisystem einrichten**

```
root# mkfs.xfs /dev/sdb1
meta-data=/dev/sdb1      isize=256    agcount=4, agsize=1048512 blks
                =                sectsz=512   attr=2, projid32bit=1
                =                crc=0         finobt=0
data        =                bsize=4096   blocks=4194048, imaxpct=25
                =                sunit=0       swidth=0 blks
naming     =version 2      bsize=4096   ascii-ci=0 ftype=0
log         =Internes Protokoll bsize=4096   blocks=2560, version=2
                =                sectsz=512   sunit=0 blks, lazy-count=1
realtime   =keine         extsz=4096   blocks=0, rtextents=0
```

Jetzt fehlt nur noch ein `mount`-Kommando, und schon können Sie das Dateisystem nutzen:

```
root# mount -t xfs /dev/sdb1 /test
```

Die Integrität von xfs-Dateisystemen wird bei jedem `mount`-Vorgang automatisch überprüft. Dabei wird aber nur das Journaling-Protokoll ausgewertet. Zur manuellen Überprüfung führen Sie `xfs_check` aus. Das ist nur möglich, wenn das Dateisystem nicht eingebunden ist. Falls das Kommando Fehler entdeckt, können Sie versuchen, diese mit `xfs_repair` zu beheben. **Dateisystem überprüfen**

Um für Kompatibilität zu den anderen Dateisystemen zu sorgen, existiert auch das Kommando `fsck.xfs`. Dieses Kommando erfüllt aber keine Aufgabe und liefert als Ergebnis immer OK.

**Parameter des Dateisystems ändern** `xfs_growfs` vergrößert ein `xfs`-Dateisystem im laufenden Betrieb. Das Dateisystem muss dazu eingebunden sein! Das Kommando setzt voraus, dass die zugrunde liegende Datenpartition vorher vergrößert wurde.

`xfs_admin` verändert diverse Parameter des Dateisystems, beispielsweise den Namen (Label) und die UUID-Nummer. Das Dateisystem muss vorher aus dem Verzeichnisbaum gelöst werden (`umount`).

## 22.13 Windows-Dateisysteme (`vfat`, `ntfs`)

Viele Linux-Anwender haben auf ihrem Rechner parallel eine Windows-Version installiert. Aber auch externe Datenträger nutzen häufig Windows-Dateisysteme (USB-Sticks, Speicherkarten von Digitalkameras etc.). Im Folgenden lernen Sie, wie Sie unter Linux auf Windows-Dateisysteme zugreifen – ganz egal, ob sich diese in einer Partition der internen Festplatte oder auf einem externen Datenträger befinden.

**Varianten** Es gibt drei fundamental unterschiedliche Windows-Dateisysteme:

- ▶ **FAT, VFAT und exFAT:** Vom FAT-Dateisystem gibt es unzählige Varianten. Die historisch ältesten Versionen sind FAT12 für Disketten, FAT16 für Dateisysteme bis 2 GiB sowie FAT32 für Dateisysteme bis 8 TiB und Dateien bis 4 GiB. Mit Windows 95 wurde außerdem VFAT eingeführt, das endlich Dateinamen mit mehr als 8+3 Zeichen erlaubte. Die Kombination aus FAT32 und VFAT ist heute am häufigsten im Einsatz, z. B. auf vielen SD-Karten bis zu 32 GiB.

Die FAT-Variante exFAT wurde speziell für große Flash-Karten entwickelt (z. B. für Digitalkameras). exFAT erlaubt Dateien bis zu 16.777.216 TiB Größe und unterstützt ACLs und Transaktionen.

- ▶ **NTFS:** Das *New Technology File System* wurde mit Windows NT eingeführt und wird von allen aktuellen Windows-Versionen genutzt. Im Vergleich zu FAT bietet NTFS eine höhere Sicherheit (Zugriffsrechte, Journaling etc.) sowie diverse Zusatzfunktionen. Die Dateisystemgröße ist mit 16.777.216 TiB nahezu unbegrenzt.
- ▶ **ReFS:** Seit Windows 8 bzw. Windows Server 2012 kann alternativ zu NTFS auch das *Resilient File System (ReFS)* verwendet werden. Dieses neue Dateisystem ist zwar moderner als NTFS, aber auch mit Einschränkungen verbunden. Es hat bisher keine weite Verbreitung gefunden und kommt bei gewöhnlichen Windows-10-Installationen nicht zum Einsatz.

Nahezu alle Linux-Distributionen können (V)FAT- und NTFS-Dateisysteme auf Anhieb lesen und schreiben. Etwas problematischer ist die exFAT-Unterstützung: Es gibt sowohl einen Open-Source-Treiber (<https://github.com/relan/exfat>) als auch einen kommerziellen Treiber der Firma Tuxera (<https://tuxera.com>). Der Open-Source-Treiber kann aber aufgrund von Patentproblemen nicht in den Kernel integriert werden und muss extra installiert werden:

```
root# apt install exfat-utils exfat-fuse    (Debian, Ubuntu)
root# dnf install exfat-utils fuse-exfat    (Fedora, RPMFusion-Paketquelle)
```

Es gibt noch keinen Open-Source-Treiber für ReFS, wohl aber einen proprietären Treiber der Firma Paragon (<https://paragon-software.com/home/refs-linux>). Mitte 2017 war der Treiber kostenlos erhältlich.

Es ist unter Linux nur selten notwendig, Windows-Dateisysteme einzurichten – aber es ist natürlich möglich. Bevor Sie `mkfs.exfat` bzw. `mkfs.ntfs` ausführen können, müssen Sie gegebenenfalls das Paket `exfat-utils` bzw. `ntfsprogs` installieren.

```
root# mkfs.vfat -F 32 /dev/sdb1 (VFAT)
root# mkfs.exfat /dev/sdb1 (exFAT)
root# mkfs.ntfs --fast /dev/sdb1 (NTFS)
```

Linux-  
Unterstützung

Windows-  
Dateisystem  
einrichten

### Vermeiden Sie Datenverluste beim Zugriff auf Windows-Dateisysteme!

Auf einem Dual-Boot-Rechner, der wahlweise unter Linux oder unter Windows läuft, können unter Linux durchgeführte Schreibzugriffe gleich aus zwei Gründen zu einem inkonsistenten Windows-Dateisystem und zu Datenverlusten führen:

- ▶ Windows wird aus Geschwindigkeitsgründen standardmäßig nicht vollständig heruntergefahren. Stattdessen ist eine Schnellstartfunktion aktiv, die darauf basiert, dass Windows in einen speziellen Ruhezustand versetzt wird. Zuletzt durchgeführte Änderungen am Dateisystem werden in einer speziellen Datei gesichert, die Linux nicht sieht. Bei einem Windows/Linux-Mischbetrieb sollten Sie die Schnellstartfunktion unbedingt deaktivieren. Eine Anleitung finden Sie z. B. hier:

<http://techmixx.de/schnellstart-unter-windows-10-deaktivieren>

Um Windows nur einmal vollständig herunterzufahren, schließen Sie zuerst alle Programme und führen dann in `cmd.exe` das Kommando `shutdown /p` aus.

- ▶ Wenn eine herkömmliche Festplatte mit einem SSD-Cache verbunden ist, sieht Linux normalerweise nur den Inhalt der Festplatte. Auch in diesem Fall kann es sein, dass zuletzt durchgeführte Änderungen am Dateisystem nur im SSD-Cache gespeichert und somit für Linux unsichtbar sind.

Am einfachsten gehen Sie derartigen Problemen aus dem Weg, indem Sie gemeinsame Windows/Linux-Daten extern synchronisieren, z. B. mit einem Dropbox-Verzeichnis.

- Konvertierung von Textdateien** Unabhängig vom Dateisystem bereitet der Textaustausch zwischen Linux und Windows oft Probleme, weil je nach Betriebssystem unterschiedliche Zeichensätze und Kennzeichnungen für das Zeilenende zur Anwendung kommen. Diese Probleme lassen sich mit diversen Konvertierungswerkzeugen lösen (siehe [Kapitel 13](#), »Konverter für Grafik, Text und Multimedia«).
- Zugriffsrechte** VFAT kennt das Konzept von Zugriffsrechten überhaupt nicht. NTFS unterstützt zwar Zugriffsrechte, aber nicht in der Unix/Linux-typischen Art und Weise. Daraus ergibt sich ein Problem: Welcher Linux-Benutzer verfügt über welche Zugriffsrechte auf Windows-Dateien? Die Antwort geben die `mount`-Optionen `uid`, `gid` und `umask/fmask/dmask`. Diese Optionen stellen den Besitzer, die Gruppenzugehörigkeit und die Zugriffsbits für das Windows-Dateisystem ein, und zwar einheitlich für alle Dateien dieses Dateisystems und unabhängig von eventuellen NTFS-Zugriffsrechten.

### Das VFAT-Dateisystem

- Standard-einstellungen** Der `vfat`-Treiber erkennt den FAT-Typ (FAT12/-16/-32) selbstständig. Die Windows-Dateinamen werden unter Linux im Zeichensatz Latin-1 (ISO8859-1) dargestellt. Standardmäßig darf der Benutzer, der `mount` ausführt, alle Dateien und Verzeichnisse lesen und schreiben; alle anderen Benutzer dürfen alles lesen, aber nichts verändern. Selbstverständlich können Sie diese Einstellungen durch Optionen verändern.
- VFAT in `/etc/fstab`** Ein typischer Eintrag in `/etc/fstab` für eine lokale VFAT-Partition auf der Festplatte sieht wie folgt aus:

```
# /etc/fstab
/dev/sda1      /media/win1  vfat      utf8,uid=1000,noexec 0 0
```

Sie erreichen damit, dass der Benutzer mit der Benutzernummer 1000 alle Dateien verändern darf und dass Sonderzeichen in Windows-Dateinamen unter Linux als UTF8-Zeichen dargestellt werden. Aus Sicherheitsgründen dürfen keine Programme ausgeführt werden, die im Windows-Dateisystem gespeichert sind (Option `noexec`).

Die folgende `fstab`-Zeile bindet die Windows-Partition nicht automatisch in den Verzeichnisbaum ein (`noauto`). Dank `users` darf aber jeder Benutzer `mount` ausführen. `gid=users` bewirkt, dass die Gruppenzugehörigkeit der Windows-Dateien durch die Standardgruppe (und nicht die gerade aktuelle Gruppe) des Benutzers bestimmt wird.

```
/dev/sda1      /media/win1  vfat      noauto,users,gid=users,utf8 0 0
```

### Das NTFS-Dateisystem

Der `ntfs`-Treiber von Linux unterstützt Lese- und Schreibzugriffe und kann mit Streams umgehen. Der Treiber kann allerdings keine verschlüsselten Dateien lesen/schreiben und keine komprimierten Dateien erzeugen (wohl aber lesen).



Im Gegensatz zu den meisten anderen Dateisystemtreibern ist `ntfs` nicht als Kernelmodul implementiert, sondern als sogenannter FUSE-Treiber. FUSE steht für *Filesystem in Userspace*. Dabei handelt es sich um ein Kernelmodul, das mit externen Programmen kommuniziert. FUSE ermöglicht es also, den eigentlichen Dateisystemtreiber außerhalb des Kernels zu implementieren.

Um dem Benutzer, der die UID 1000 hat, Schreibrechte auf einen NTFS-Datenträger zu geben brauchen, Sie eine `fstab`-Zeile wie im folgenden Muster: NTFS in `/etc/fstab`

```
# /etc/fstab
/dev/sda1 /media/win ntfs uid=1000,gid=1000 0 0
```

Streams sind eine Besonderheit des NTFS-Dateisystems: Eine NTFS-Datei kann aus mehreren Streams bestehen. Dabei hat jeder Stream dieselbe Funktion wie eine herkömmliche Datei. Beim gewöhnlichen Dateizugriff wird automatisch der Standardstream gelesen bzw. verändert. Streams

Beim `ntfs`-Treiber steuert die Option `streams_interface` den Zugriff auf Streams. In der Standardeinstellung `xattr` werden Streams wie Dateiattribute betrachtet. Der Zugriff auf Streams erfolgt durch die Kommandos `get-` bzw. `setfattr` aus dem Paket `attr` (siehe [Abschnitt 11.7](#), »Access Control Lists und Extended Attributes«). `getfattr -d -e text` liefert eine Liste aller Attribute, wobei deren Inhalt in Textform angezeigt wird.

```
root# mount /dev/sda1 /media/win
root# cd /media/win
root# cat > streamtest
abc <Strg>+<D>

root# setfattr -n user.stream1 -v "efg" streamtest
root# setfattr -n user.stream2 -v "xyz" streamtest
root# cat streamtest
abc
root# getfattr -d -e text streamtest
# file: streamtest
user.stream1="efg"
user.stream2="xyz"

root# cd
root# umount /media/win
```

Alternativ können Sie auch mit `streams_interface=windows` arbeiten. Diese Einstellung aktiviert die Windows-typische Schreibweise in der Form `dateiname:streamname`.

```
root# mount -o streams_interface=windows /dev/sda1 /media/win
root# cd /media/win
root# cat streamtest
abc
root# cat streamtest:stream1
efg
```

**Administration** Das Paket `ntfsprogs` enthält diverse Kommandos, die bei der Administration von NTFS-Dateisystemen helfen. [Tabelle 22.7](#) gibt einen Überblick.

| Kommando                  | Bedeutung  |
|---------------------------|--|
| <code>mkfs.ntfs</code>    | richtet ein NTFS-Dateisystem ein.                |
| <code>ntfsclone</code>    | kopiert ein NTFS-Dateisystem.                    |
| <code>ntfsinfo</code>     | liefert Informationen über ein NTFS-Dateisystem. |
| <code>ntfslabel</code>    | benennt eine NTFS-Partition.                     |
| <code>ntfsresize</code>   | ändert die Größe des NTFS-Dateisystems.          |
| <code>ntfsundelete</code> | versucht, gelöschte Dateien wiederherzustellen.  |

**Tabelle 22.7** Kommandos des `ntfsprogs`-Pakets

## 22.14 CDs und DVDs

CDs und DVDs kommen aus der Mode, ganz ausgestorben sind sie aber noch nicht. Dieser kurze Abschnitt erläutert, wie der Zugriff auf Daten-CDs und -DVDs aus technischer Sicht unter Linux funktioniert.

CD- und DVD-Laufwerke werden im Prinzip wie Festplatten verwaltet. Es gibt aber zwei wesentliche Unterschiede: Erstens ist bei einem CD/DVD-Laufwerk ein Wechsel der CD/DVD möglich, während Sie eine herkömmliche Festplatte im laufenden Betrieb nicht wechseln können. Zweitens verwenden Daten-CDs und -DVDs ein anderes Dateisystem: ISO 9660 oder UDF.

**CD/DVD-Device-Namen** Das CD- oder DVD-Laufwerk ist bei den meisten Linux-Distributionen unter dem Device-Namen `/dev/sr0` zugänglich, mitunter auch unter `/dev/sdc0`.

**ISO 9660 und UDF** ISO 9660 ist das bis heute übliche »Dateisystem« für Daten-CDs. Aufgrund grundlegender Einschränkungen haben sich einige Erweiterungen etabliert: Die Unix-typische Rockridge-Extension erlaubt es, lange Dateinamen und Zugriffsrechte zu speichern. Die Windows-typische Joliet-Erweiterung sieht die Verwendung von Unicode-Zeichen in Dateinamen vor. Die El-Torito-Erweiterung ermöglicht es, einen Rechner direkt von der CD zu starten.

Das *Universal Disk Format* (UDF) ist der Nachfolger zu ISO 9660. Es wird auf vielen DVDs verwendet. DVDs können alternativ aber auch das ISO-9660-Format nutzen. Anders als unter ISO 9660 dürfen unter UDF Dateien größer als 2 GiB werden, Dateinamen können ohne irgendwelche Erweiterungen aus bis zu 255 Unicode-Zeichen bestehen, Read-Write-Medien werden besser unterstützt (Packet Writing) etc.

Wenn Sie in einer Konsole oder mit einem Desktop ohne CD/DVD-Automatismen arbeiten, müssen Sie Ihre CDs/DVDs nach dem Einlegen manuell in den Verzeichnisbaum einbinden. Wie üblich variieren dabei die Device- und Verzeichnisnamen je nach Hardware und Distribution.

Manueller Betrieb

```
root# mount -t iso9660 -o ro /dev/sr0 /media/dvd (ISO-9660-CDs/DVDs)
root# mount -t udf -o ro /dev/sr0 /media/dvd (UDF-DVDs)
```

Standardmäßig sind alle Verzeichnisse und Dateien für alle Benutzer lesbar. Falls Sie Programme, die sich auf der CD bzw. DVD befinden, unmittelbar starten möchten, müssen Sie zusätzlich die Option `exec` angeben. Um internationale Dateinamen korrekt verarbeiten zu können, sollten Sie die Option `iocharset=utf8` bzw. einfach `utf8` verwenden.

Bevor Sie die CD/DVD auswerfen können, müssen Sie explizit `umount` ausführen:

```
root# umount /media/dvd
```

#### umount oder eject

Statt `umount` können Sie auch `eject` ausführen. Durch dieses Kommando wird die CD/DVD nicht nur aus dem Dateisystem gelöst, sondern auch gleich ausgeworfen. Falls es im Rechner mehrere Datenträger gibt, die ausgeworfen werden können, werden diese Möglichkeiten der Reihe nach getestet; der erste gefundene Datenträger wird ausgeworfen. Optional können Sie den gewünschten Datenträger durch den Device-Namen oder Mount-Punkt angeben.

Wenn `umount` den Fehler *device is busy* liefert, bedeutet das, dass ein anderes Programm noch Daten der CD/DVD nutzt. Das ist unter anderem auch dann der Fall, wenn in irgendeiner Shell ein Verzeichnis der CD/DVD geöffnet ist. Führen Sie dort `cd` aus, um in das Heimatverzeichnis zu wechseln. Bei der Suche nach dem Prozess, der die `umount`-Probleme verursacht, kann `fuser` helfen. Führen Sie `fuser -m /cdrom` aus!

Device is busy

`/etc/fstab` enthält normalerweise keine Zeile für das DVD-Laufwerk. Bei Desktop-Systemen wie KDE und Gnome kümmert sich ein Hintergrundprozess darum, eine neu eingelegte CD/DVD automatisch einzubinden. Nur wenn Sie CDs/DVDs häufig manuell in den Verzeichnisbaum einbinden, ist ein `fstab`-Eintrag aber zweckmäßig. Er sieht dann ähnlich wie das folgende Muster aus:

/etc/fstab

```
# /etc/fstab
/dev/sr0 /media/dvd udf,iso9660 users,noauto,ro 0 0
```

Jetzt reichen die Kommandos `mount /media/dvd` bzw. `umount /media/dvd` aus, um eine CD/DVD in den Verzeichnisbaum zu integrieren bzw. aus ihm zu lösen. Jeder Benutzer darf diese Kommandos ausführen.

- Audio-CDs** Audio-CDs werden anders als Daten-CDs behandelt. Sie werden nicht mit `mount` in das Dateisystem eingebunden, sondern mit speziellen Programmen direkt ausgelesen.
- Video-DVDs** Video-DVDs verwenden in der Regel das Dateisystem UDF. Zum Abspielen solcher DVDs benötigen Sie einen Video- oder Multimedia-Player, der die im Dateisystem enthaltenen Dateien korrekt interpretiert und mit den verwendeten Codecs zurechtkommt.
- CDs/DVDs brennen** Um CDs und DVDs zu brennen, verwenden Sie unter KDE das Programm K3B, unter Gnome Brasero bzw. in der Konsole `wodim`.

## 22.15 Externe Datenträger

USB-Sticks, Speicherkarten von Digitalkameras und externe Festplatten haben ein gemeinsames Merkmal: Sie werden im laufenden Betrieb mit dem Computer verbunden und auch wieder gelöst. Intern werden nahezu alle derartigen Laufwerke wie SCSI-Laufwerke behandelt.

- Automatischer Betrieb** Die Desktop-Systeme (KDE, Gnome) nahezu aller Distributionen reagieren beim Einstecken von Datenträgern damit, dass ein neues Fenster des Dateimanagers erscheint, das komfortabel Zugriff auf den Datenträger gibt. Eventuell erscheint auf dem Desktop auch ein Icon, das den Datenträger symbolisiert und Ihnen per Kontextmenü die Möglichkeit gibt, das Dateisystem explizit aus dem Verzeichnisbaum zu lösen.

### Zuerst »umount«, dann Stecker/Kabel lösen!

Achten Sie darauf, sämtliche Partitionen eines Datenträgers explizit aus dem Verzeichnisbaum zu lösen, bevor Sie das Kabel zum Datenträger abziehen! Normalerweise stellt der Dateimanager dazu ein Kommando wie `AUSWERFEN` oder `SICHER ENTFERNEN` zur Verfügung. Hinter den Kulissen wird dann `umount` ausgeführt. Das stellt sicher, dass alle offenen Schreiboperationen ausgeführt werden, bevor die Verbindung zum Laufwerk tatsächlich gekappt wird. Wenn Sie auf diesen Schritt verzichten, riskieren Sie ein beschädigtes Dateisystem und fehlerhafte Dateien!

Unter KDE und Gnome ist es möglich, dass mehrere Benutzer parallel eingeloggt sind. In diesem Fall bekommt in der Regel der zuerst eingeloggte Benutzer Zugriffsrechte auf neu eingesteckte Datenträger. Dieser Sonderfall ist allerdings je nach Distribution unterschiedlich (oder gar nicht) gelöst und kann Probleme verursachen. Vermeiden Sie also Benutzerwechsel, wenn Sie mit externen Datenträgern arbeiten!

- Hotplug-System** Die Hotplug-Verwaltung basiert bei aktuellen Distributionen auf einem Zusammenspiel des Kernels, des `udev`-Systems, des Kommunikationssystems D-Bus und des Programms `PolicyKit`.

Durch die Definition eigener `udev`-Regeln können Sie Scripts erzeugen, die nach der Erkennung eines neuen Datenträgers bzw. vor dem Lösen eines Dateisystems aus dem Verzeichnisbaum ausgeführt werden. Eine kurze Anleitung finden Sie hier:

<https://superuser.com/questions/305723>

Wenn Sie das Dateisystem eines USB-Sticks oder eines anderen externen Datenträgers manuell in den Verzeichnisbaum einbinden möchten, ermitteln Sie zuerst mit `lsblk` den Device-Namen, erzeugen dann mit `mkdir` das Verzeichnis und führen anschließend `mount device verzeichnis` aus. Ein konkretes Beispiel dazu finden Sie in [Abschnitt 22.2](#), »USB-Datenträger formatieren und nutzen«. Dort finden Sie auch eine Anleitung, wie Sie `/etc/fstab` konfigurieren, damit ein externer Datenträger dauerhaft in einem bestimmten Verzeichnis in den Verzeichnisbaum eingebettet wird.

Manueller Betrieb

## 22.16 Swap-Partitionen und -Dateien

Wenn der Arbeitsspeicher zur Ausführung aller Programme nicht ausreicht und Swap-Partitionen oder -Dateien zur Verfügung stehen, lagert Linux Teile des Speichers dorthin aus (Paging). Linux kann auf diese Weise mehr Speicher nutzen, als RAM verfügbar ist.

Das Einrichten einer Swap-Partition erfolgt normalerweise im Rahmen der Installation. Ob und wie viel Swap-Speicher zur Verfügung steht bzw. tatsächlich verwendet wird, überprüfen Sie mit dem Kommando `free`. Im Beispiel unten stehen 1519 MiB RAM und 2000 MiB Swap-Speicher zur Verfügung. Vom RAM werden zurzeit 401 MiB für Programme und Daten verwendet, der Rest wird als Puffer bzw. Cache für Dateien genutzt. Der Swap-Speicher ist momentan ungenutzt.

```
root# free -m
              total        used         free       shared    buffers     cached
Mem:           1519         1479           39           0           67         1010
-/+ buffers/cache:           401         1117
Swap:          2000           0         2000
```

Wenn ein Rechner länger läuft, wird er früher oder später den Swap-Speicher selbst dann nutzen, wenn sehr viel RAM zur Verfügung steht. Der Grund: Der Kernel verwaltet einen Cache für Lesezugriffe auf Dateien. Wird eine Datei später wieder benötigt, kann sie aus dem Cache gelesen werden. Sobald der Cache größer ist als das ansonsten freie RAM, lagert Linux seit langer Zeit nicht mehr genutzte Speicherblöcke in die Swap-Partition aus. Das ist durchaus kein Zeichen dafür, dass zu wenig RAM zur Verfügung steht. Linux versucht lediglich, den verfügbaren Speicher möglichst effizient zu nutzen.

**/etc/fstab** Die folgenden Zeilen zeigen zwei Einträge für Swap-Partitionen in `/etc/fstab`. Die Option `pri` bewirkt, dass die beiden Partitionen von Linux gleichwertig behandelt werden. Das sorgt für eine Geschwindigkeitssteigerung wie beim Striping oder bei RAID-0 (siehe [Abschnitt 22.17](#), »RAID«), sofern sich die Partitionen auf zwei voneinander unabhängigen Festplatten befinden. Wenn es nur eine Swap-Partition gibt, geben Sie statt `pri=0` einfach `defaults` an.

```
# /etc/fstab: Swap-Partitionen
/dev/sda9 swap swap pri=1 0 0
/dev/sdb7 swap swap pri=1 0 0
```

**Swap-Verhalten steuern** Wenn der Speicher im RAM knapp wird, entscheidet der Linux-Kernel nach einem relativ komplexen Algorithmus, ob Cache-Speicher zugunsten anderer Speicheranforderungen freigegeben wird oder ob zuletzt ungenutzte Speicherbereiche in die Swap-Partition ausgelagert werden sollen. Mit dem Kernelparameter `/proc/sys/vm/swappiness` können Sie selbst einstellen, ob der Kernel nach Möglichkeit eher den Cache verkleinert oder Daten auslagert. Was Kernelparameter sind und wie Sie sie einstellen, erfahren Sie in [Abschnitt 25.9](#), »Kernelparameter verändern«.

Die Standardeinstellung für `swappiness` lautet 60, der mögliche Wertebereich reicht von 0 bis 100. Der Wert 0 bedeutet, dass der Kernel Paging nach Möglichkeit vermeidet. 100 bedeutet, dass längere Zeit ungenutzter Speicher möglichst bald in einer Swap-Partition landet. Weitere Details zum `swappiness`-Parameter finden Sie auf der folgenden Seite:

<https://lwn.net/Articles/83588>

In der Praxis werden Sie das Swap-Verhalten am ehesten bemerken, wenn Sie Ihren Rechner über Nacht laufen lassen und in Ihrer Abwesenheit ein Programm auf viele Dateien der Festplatte zugreift, etwa ein Backup-Script oder ein Programm zur Erstellung eines Suchindex. Wegen der vielen Dateizugriffe wächst der Cache stark an. Mit `swappiness=60` oder einem noch höheren Wert wird der Kernel nun seit Stunden nicht genutzten Speicher auslagern. Das könnte beispielsweise die Speicherseiten von Firefox oder GIMP betreffen. Wenn Sie am nächsten Tag in GIMP weiterarbeiten möchten, wird es ein paar Sekunden dauern, um diese Seiten aus der Swap-Partition wieder in den Arbeitsspeicher zu übertragen. Mit `swappiness=0` vermeiden Sie diese Wartezeit.

**Wie viel Swap-Speicher?** In der Vergangenheit lautete die Empfehlung, etwa das Zweifache des RAMs als Swap-Speicher vorzusehen. Mit zunehmender RAM-Größe ist diese Faustregel aber nicht mehr zielführend. Wenn Sie Linux vor allem als Desktop-System verwenden, reicht eine wesentlich kleinere Swap-Partition vollkommen aus (z. B. 1 GiB Swap-Speicher bei 4 GiB RAM).

Es gibt allerdings eine Ausnahme: Wenn Sie bei Notebooks den Ruhezustand (*Suspend to Disk*) nutzen möchten, was nach meinen Erfahrungen leider nur selten stabil

funktioniert, wird der gesamte Arbeitsspeicher in der Swap-Partition gespeichert. Das setzt voraus, dass die Swap-Partition etwas größer ist als der Arbeitsspeicher.

Wiederum andere Anforderungen werden an große Server-Systeme gestellt. Beispielsweise empfiehlt Oracle für seinen Datenbankserver je nach verfügbarem RAM unterschiedliche Faktoren zur Berechnung der Swap-Größe:

|                  |            |
|------------------|------------|
| Bis 2 GiB:       | Faktor 1,5 |
| 2 bis 16 GiB:    | Faktor 1   |
| Mehr als 16 GiB: | 16 GiB     |

Auf 32-Bit-Systemen beträgt die maximale Größe einer Swap-Partition 2 GiB. Sollten Sie mehr Swap-Speicher benötigen, können Sie mehrere Swap-Partitionen verwenden. Noch sinnvoller ist dann aber ein Wechsel auf eine 64-Bit-Distribution.

Immer wieder taucht die Frage auf, ob man ganz auf eine Swap-Partition verzichten kann bzw. sollte, wenn man eine Menge RAM hat. Grundsätzlich funktioniert Linux auch ohne Swap-Speicher; ein Argument spricht aber für eine Swap-Partition: Sollte eines Ihrer Programme außer Kontrolle geraten oder aus anderen Gründen mehr Speicher brauchen als erwartet, ist der verfügbare Speicher irgendwann erschöpft. Das kann zum Absturz des nächsten Prozesses führen, der weiteren Speicher anfordert. Das kann irgendein Prozess sein, nicht unbedingt Ihr außer Kontrolle geratenes Programm.

Linux ohne  
Swap-Partition

Grundsätzlich ändert eine Swap-Partition nichts an diesem Problem – auch der Swap-Speicher ist ja begrenzt. Durch die immer intensivere Nutzung des Swap-Speichers laufen alle Programme aber immer langsamer, sodass Sie rechtzeitig bemerken, dass auf Ihrem Rechner etwas nicht stimmt. Bevor es zu einem Absturz kommt, können Sie das fehlerhafte Programm beenden, zur Not durch `kill`.

Falls sich die Swap-Partition als zu klein herausstellt oder Sie aus anderen Gründen eine weitere Swap-Partition benötigen, richten Sie eine neue Partition ein. Als Partitionstyp geben Sie *Linux swap* an (Code 82 in `fdisk`). Nachdem die Partition mit `mkswap` formatiert worden ist, kann sie mit `swapon` aktiviert werden. Wenn das klappt, ergänzen Sie `/etc/fstab`.

Neue  
Swap-Partition  
einrichten

Aus Geschwindigkeitsgründen sollten Sie möglichst nur eine Swap-Partition pro Festplatte einrichten. Idealerweise sollte sich die Swap-Partition auf einer sonst nicht oder wenig genutzten Festplatte befinden.

Swap-Dateien sind nicht ganz so effizient wie Swap-Partitionen, lassen sich dafür aber leichter in ihrer Größe ändern. (Eine automatische Größenanpassung wie unter Windows oder macOS ist aber leider nicht möglich.) Als aktuell einzige große Distribution richtet Ubuntu seit Version 17.04 bei Desktop-Installationen standardmäßig eine Swap-Datei ein.

Swap-Dateien

Um selbst eine neue Swap-Datei anzulegen, erzeugen Sie mit dem Kommando `dd` eine leere Datei mit einer vorgegebenen Größe. Dabei wird als Datenquelle `/dev/zero` verwendet. Die Größenangabe ergibt sich aus dem Produkt der Parameter `bs` und `count`, hier also 1 GiB. Anschließend wird die Swap-Datei wie eine Swap-Partition mit `mkswap` formatiert und mit `swapon` aktiviert:

```
root# dd bs=1M if=/dev/zero of=/swapfile count=1024
root# mkswap /swapfile 1000
root# sync
root# swapon -v /swapfile
swapon on device /swapfile
```

Um die Größe einer Swap-Datei zu ändern, deaktivieren Sie diese mit `swapoff` und richten dann eine neue Datei in der gewünschten Größe ein. Swap-Dateien werden wie Swap-Partitionen in `fstab` aufgenommen:

```
# Erweiterung zu /etc/fstab
/swapfile none      swap      sw        0 0
```

Beachten Sie, dass das `btrfs`-Dateisystem Swap-Dateien nicht unterstützt!

## 22.17 RAID

Was RAID ist, welche RAID-Level es gibt und wodurch sich Hardware-RAID, BIOS-Software-RAID und Linux-Software-RAID voneinander unterscheiden, habe ich bereits in [Abschnitt 2.7, »RAID, LVM und Verschlüsselung«](#), erklärt. Dieser Abschnitt behandelt ausschließlich die Administration von Linux-Software-RAID auf der Basis von `mdadm`. Aus Platzgründen gehe ich zudem nur auf die RAID-Level 0 und 1 ein.

Falls Sie das Dateisystem `btrfs` nutzen und die RAID-Level 0, 1, 5, 6 oder 10 verwenden möchten, können Sie auf die hier beschriebenen RAID-Funktionen des *Multi Device Drivers* verzichten. `btrfs` enthält selbst RAID-Funktionen.

### Grundlagen

**mdadm** Sofern Sie nicht schon während der Installation einen RAID-Verbund eingerichtet haben, müssen Sie das Paket `mdadm` installieren. Es enthält das gleichnamige Kommando zur RAID-Administration.

`mdadm` empfiehlt auch die Installation eines Mail-Servers (Mail Transfer Agents), damit bei RAID-Problemen eine E-Mail an den Administrator versandt werden kann. Wenn Sie sich noch nicht mit dem Thema E-Mail-Server beschäftigt haben, sollten Sie darauf aber verzichten. Unter Debian und Ubuntu verwenden Sie deswegen bei der Installation mit `apt` die Option `--no-install-recommends`.



Linux-intern ist für Software-RAID der sogenannte *Multi Devices Driver* zuständig. Bei einigen Distributionen ist dieser Treiber direkt in den Kernel integriert, andernfalls wird das Kernelmodul `md_mod` (ehemals einfach `md`) während des Systemstarts automatisch geladen. `dmesg` sollte auf jeden Fall entsprechende Meldungen enthalten. Vergewissern Sie sich auch, dass die Pseudodatei `/proc/mdstat` existiert. Sie gibt Auskunft über den aktuellen Zustand des RAID-Systems.

`md_mod` setzt eine logische Schicht zwischen den Treiber zum Festplattenzugriff (SATA/IDE/SCSI) und den Dateisystemtreiber (z. B. `ext4`). `md_mod` bildet aus mehreren Festplatten-Partitionen ein neues, logisches Device, auf das der Dateisystemtreiber zugreifen kann (`/dev/mdN`). Nach der RAID-Konfiguration verwenden Sie nicht mehr direkt eine Festplattenpartition, sondern eine RAID-Partition `/dev/mdN`, um darauf Ihr Dateisystem einzurichten.

Die zentrale RAID-Konfigurationsdatei ist `/etc/mdadm/mdadm.conf`. Diese Datei sollte neben einigen globalen RAID-Einstellungen Daten über alle aktiven RAID-Verbunde enthalten. Eine vollständig neue Konfigurationsdatei können Sie mit `/usr/share/mdadm/mkconf` erstellen. Das ist dann praktisch, wenn die Konfigurationsdatei verloren gegangen ist oder wenn Sie auf einem Live- oder Rescue-System arbeiten.

Die übliche Vorgehensweise bei der Konfiguration ist ungewöhnlich: Zuerst richten Sie durch die Ausführung von `mdadm`-Kommandos die gewünschten RAID-Verbunde ein oder modifizieren sie. Anschließend erweitern Sie die zumeist schon vorhandene Datei `mdadm.conf` auf der Grundlage der nun vorliegenden Konfiguration. Die Eckdaten der aktiven RAID-Verbunde ermitteln Sie mit `mdadm --examine --scan`, und Sie fügen sie mit `>>` zur existierenden Konfigurationsdatei hinzu:

```
root# mdadm --examine --scan >> /etc/mdadm/mdadm.conf
```

Falls `mdadm.conf` schon vorher RAID-Definitionen enthielt, müssen Sie diese mit einem Editor entfernen, damit kein Verbund doppelt definiert ist. Die folgenden Zeilen zeigen ein Beispiel für den Aufbau von `mdadm.conf`:

```
# Datei /etc/mdadm/mdadm.conf
DEVICE partitions
CREATE owner=root group=disk mode=0660 auto=yes
HOMEHOST <system>
MAILADDR root
ARRAY /dev/md0 level=raid1 num-devices=2 UUID=36c426b0:...
ARRAY /dev/md1 level=raid1 num-devices=2 UUID=71dfc474:...
ARRAY /dev/md2 level=raid1 num-devices=2 UUID=e0f65ea0:...
```

Aktuelle Informationen über den RAID-Status gibt die schon erwähnte Datei `/proc/mdstat`. Im folgenden Beispiel gibt es drei RAID-1-Verbunde, die aus jeweils zwei Partitionen bestehen. Alle drei Verbunde sind aktiv und laufen fehlerfrei: `[UU]` bedeutet,

dass die erste und die zweite Partition des Verbunds *up* ist (also problemlos funktioniert).

```
root# cat /proc/mdstat
Personalities : [raid0] [raid1] [linear] [multipath]
                [raid6] [raid5] [raid4] [raid10]

md0 : active raid1 sda1[0] sdb1
      979840 blocks [2/2] [UU]
md1 : active raid1 sda2[0] sdb2
      1951808 blocks [2/2] [UU]
md2 : active raid1 sda3[0] sdb3
      387730624 blocks [2/2] [UU]

unused devices: <none>
```

Nun ist es natürlich nicht praktikabel, ständig in dieser Datei nachzusehen, ob alles in Ordnung ist. Wesentlich zweckmäßiger ist es, dass `mdadm --monitor` diese Aufgabe übernimmt. Zumeist wird dieses Kommando durch das Init-System beim Hochfahren des Rechners gestartet. Je nach Distribution kann es aber sein, dass Sie `mdadm` vorher entsprechend konfigurieren müssen. Dazu führen Sie beispielsweise unter Ubuntu das folgende Kommando aus:

```
root# dpkg-reconfigure mdadm
```

Vier Dialoge führen nun durch die `mdadm`-Konfiguration: Im ersten Schritt können Sie eine automatische Redundanzüberprüfung aktivieren, die einmal pro Monat (am ersten Sonntag um 1:06 Uhr) die Daten auf den RAID-Partitionen miteinander vergleicht. Diese Kontrolle hilft dabei, Festplattendefekte auch in Bereichen bzw. Dateien festzustellen, die schon länger nicht mehr gelesen oder verändert wurden. Intern erfolgt die Redundanzüberprüfung durch das Kommando `checkarray`, das durch das Cron-Script `/etc/cron.d/mdadm` gestartet wird.

Im zweiten und dritten Schritt aktivieren Sie die Überwachung des RAID-Status und geben an, an welche E-Mail-Adresse eventuell auftretende Warnungen bzw. Fehlerberichte versandt werden sollen. Intern erfolgt die Überwachung durch `mdadm --monitor`. Unter Debian und Ubuntu wird das Kommando vom Init-System gestartet, sofern `/etc/default/mdadm` die Einstellung `START_DAEMON=true` enthält. Die E-Mail-Adresse wird in `/etc/mdadm/mdadm.conf` gespeichert. Sollte ein Problem auftreten, sendet `mdadm` eine Benachrichtigungs-E-Mail an `root`. Damit das funktioniert, muss auf dem Rechner ein Mail-Server installiert sein (siehe [Kapitel 34](#), »Postfix und Dovecot«)! Die E-Mail-Adresse können Sie in `/etc/mdadm/mdadm.conf` mit der Variablen `MAILADDR` einstellen.

Im vierten Schritt geben Sie schließlich an, ob Ihr Server beim Neustart auch dann starten soll, wenn er einen Defekt in einer RAID-Partition feststellt. Bei Root-Servern ist das empfehlenswert.

## Administration

Für RAID-0 benötigen Sie zumindest zwei noch ungenutzte Partitionen. Die Partitionen sollten gleich groß sein, das ist aber nicht unbedingt erforderlich. Je nach RAID-Level führt eine unterschiedliche Größe aber dazu, dass die Geschwindigkeit nicht optimal ist bzw. dass Teile der größeren Partition nicht genutzt werden.

RAID-0-Verbund  
einrichten

Die Partitionen müssen als RAID-Partitionen gekennzeichnet sein. Wenn Sie zum Partitionieren `fdisk` verwenden, stellen Sie die Partitions-ID-Nummer mit dem Kommando `T` auf den hexadezimalen Wert `fd`. Bei `parted` führen Sie `set n raid on` aus, wobei `n` die Partitionsnummer ist.

Im Folgenden werden die Partitionen mit den Device-Namen `/dev/sda3` und `/dev/sdc1` zu einem RAID-0-System verbunden. Die Partitionen müssen für sich nicht formatiert werden. `fdisk -l` zeigt die Beispielkonfiguration:

```
root# fdisk -l /dev/sda /dev/sdc
Disk /dev/sda: 320.0 GB, 320072933376 bytes
  Device Boot      Start         End      Blocks   Id  System
 /dev/sda1            1           973     7815591   83   Linux
 /dev/sda2            974        1034     489982+   82   Linux swap / Solaris
 /dev/sda3           1035        2251     9775552+   fd   Linux raid autodetect
Disk /dev/sdc: 320.0 GB, 320072933376 bytes
  Device Boot      Start         End      Blocks   Id  System
 /dev/sdc1            1          1217     9775521   fd   Linux raid autodetect
```

Ein einziges `mdadm`-Kommando reicht aus, um aus den beiden Partitionen `/dev/sda3` und `/dev/sdc1` einen RAID-0-Verbund zu bilden:

```
root# mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/sda3 /dev/sdc1
mdadm: array /dev/md0 started.
```

Als Nächstes müssen Sie auf der neuen virtuellen Partition `/dev/md0` ein Dateisystem anlegen. Diese Partition kann mit `mount` in das Linux-Dateisystem eingebunden werden. Die Partition wird hier über das Verzeichnis `/striped` angesprochen – selbstverständlich können Sie stattdessen auch einen anderen Namen verwenden.

```
root# mkfs.ext4 /dev/md0
root# mkdir /striped
root# mount /dev/md0 /striped/
```

Wenn alles klappt, sollten Sie die neue Partition in `/etc/fstab` aufnehmen. Bei allen aktuellen Linux-Distributionen wird das RAID-System beim nächsten Systemstart durch das Init-System automatisch initialisiert.

```
# in /etc/fstab
/dev/md0 /striped ext4 defaults 0 0
```

Außerdem müssen Sie die Konfigurationsdatei `mdadm.conf` um eine Zeile erweitern, die den neuen RAID-0-Verbund beschreibt. `mdadm --examine --scan` liefert die Zeile in der vorgeschriebenen Syntax.

**RAID-1-Verbund einrichten** Die Vorgehensweise beim Einrichten eines RAID-1-Verbunds ist exakt dieselbe wie bei RAID-0. Einzig das Kommando zum Einrichten des RAID-Systems sieht ein wenig anders aus und enthält nun `--level=1` statt `--level=0`:

```
root# mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sda3 /dev/sdc1
mdadm: array /dev/md0 started.
root# mkfs.ext4 /dev/md0
```

Falls Sie – wie oben beschrieben – `/dev/md0` vorher probeweise als RAID-0-Partition eingerichtet haben, müssen Sie die Partition mit `mount` aus dem Verzeichnisbaum lösen und mit `mdadm --stop` deaktivieren, bevor Sie `mdadm --create` ausführen können! `mdadm` erkennt dennoch, dass die Partitionen `/dev/sda3` und `/dev/sdc1` bisher anders genutzt wurden, und verlangt eine Bestätigung, dass Sie `/dev/md0` wirklich neu einrichten möchten.

**RAID-1-Verbund testen** Um die Funktionsweise eines RAID-1-Verbunds zu testen – möglichst noch bevor Sie kritische Daten dort gespeichert haben –, markieren Sie eine Partition als defekt:

```
root# mdadm /dev/md0 --fail /dev/sdc1
```

Sofern im Rahmen des Systemstarts `mdadm --monitor` gestartet wurde, sollte `root` auf dem lokalen Rechner nun sofort eine Verständigungs-E-Mail erhalten. Ansonsten können Sie den Verbund weiter nutzen; alle Änderungen werden nun aber nur noch auf der verbleibenden Festplattenpartition gespeichert. `/proc/mdstat` zeigt nun den Status `_U`. Das bedeutet, dass eine Partition läuft (`U` für *up*) und eine fehlt (`_`).

```
root# cat /proc/mdstat
md0 : active raid1 sda3
      979840 blocks [2/1] [_U]
```

Um `/dev/sdc1` wieder zu `/dev/md0` hinzuzufügen, müssen Sie die als defekt gekennzeichnete Partition zuerst explizit entfernen:

```
root# mdadm --remove /dev/md0 /dev/sdc1
root# mdadm --add /dev/md0 /dev/sdc1
```

Es beginnt nun die automatische Resynchronisation der beiden Partitionen, die je nach Größe des Verbunds geraume Zeit dauert (Richtwert: ca. 20 Minuten pro 100 GiB). Immerhin können Sie in dieser Zeit weiterarbeiten. Das Dateisystem wird allerdings langsamer als sonst reagieren.

```
root# cat /proc/mdstat
md0 : active raid1 sda3 sdc1[2]
```

```

485454656 blocks [2/1] [U_]
[>.....] recovery = 3.0% (14577856/485454656)
                               finish=72.8min speed=107724K/sec
root# mdadm --detail /dev/md0 (während die Synchronisation läuft)
...
      State : clean, degraded, recovering
Active Devices : 1
Working Devices : 2
Failed Devices : 0
Spare Devices : 1
Rebuild Status : 75% complete
...
  Number  Major  Minor  RaidDevice State
     0     3     3        0  active sync  /dev/sda3
     1     0     0        -  removed
     2    22     2        1  spare rebuilding /dev/sdc1

```

```

root# mdadm --detail /dev/md0 (nach Abschluss der Synchronisation)
...
      State : clean
Active Devices : 2
Working Devices : 2
Failed Devices : 0
Spare Devices : 0
...
  Number  Major  Minor  RaidDevice State
     0     3     3        0  active sync  /dev/sda3
     1    22     2        1  active sync  /dev/sdc1

```

Allzu oft kommt es ja zum Glück nicht vor – aber wenn tatsächlich eine Festplatte Defekte zeigt und `mdadm` einzelne Partitionen dieser Festplatte als defekt kennzeichnet, sollten Sie alle Partitionen dieser Festplatte explizit aus den jeweiligen RAID-Verbunden entfernen:

**Defekte  
RAID-1-Festplatte  
austauschen**

```

root# mdadm --remove /dev/md0 /dev/sdc1
...

```

Anschließend müssen Sie schleunigst eine Ersatzfestplatte besorgen. Die neue Festplatte muss genug Platz bieten, um auf ihr genauso große Partitionen zu erzeugen wie auf den existierenden Festplatten.

Passen Sie auf, dass Sie wirklich die defekte Festplatte ausbauen und nicht irrtümlich die noch funktionierende! Diese Empfehlung klingt trivial, aber wenn ein Rechner zwei oder mehrere baugleiche Festplatten enthält, ist es gar nicht so einfach, die richtige Festplatte zu finden. Eindeutig ist nur die Seriennummer! Welche Seriennummer mit welchem Device-Namen verbunden ist, verraten `hdparm` oder `smartctl`. Beide Kommandos können nur ausgeführt werden, wenn vorher das gleichnamige Paket installiert wird.

```

root# smartctl -i /dev/sdc
...
Device Model:      SAMSUNG HD403LJ
Serial Number:    SONFJ1MPA07356

root# hdparm -i /dev/sdc
/dev/sdb:
...
Model=SAMSUNG HD403LJ, FwRev=CT100-12,
SerialNo=SONFJ1MPA07356

```

Nach dem Austausch der Festplatte müssen Sie auf der neuen Festplatte neue Partitionen einrichten, die mindestens so groß sind wie die bereits vorhandenen RAID-Partitionen. Die Partitionen müssen als RAID-Partitionen gekennzeichnet werden (hexadezimaler ID-Code `fd`). Nach diesen Vorbereitungsarbeiten ist der Rest ein Kinderspiel: Sie fügen die Partitionen der neuen Festplatte den RAID-Verbunden hinzu:

```

root# mdadm --add /dev/md0 /dev/sdc1
...

```

Der Kernel beginnt nun, die Partitionen der neuen Festplatte mit den vorhandenen RAID-Daten zu synchronisieren. Den Status der Synchronisation verfolgen Sie mit `cat /proc/mdstat`.

### Üben für den Notfall

Ich empfehle Ihnen nachdrücklich, eine RAID-Reparatur auf einem Testsystem in Ruhe auszuprobieren. Einen Festplattendefekt können Sie simulieren, indem Sie eine Partition mit `mdadm --fail` als defekt kennzeichnen oder das Kabel zu einer Festplatte vorübergehend lösen (aber natürlich nicht im laufenden Betrieb!).

**RAID-Verbund deaktivieren** `mdadm --stop` deaktiviert einen RAID-Verbund. Das darauf enthaltene Dateisystem muss vorher mit `umount` aus dem Verzeichnisbaum gelöst werden!

```

root# umount /mount-verzeichnis/
root# mdadm --stop /dev/md0

```

**RAID-Verbund wieder aktivieren** Nur wenn Sie nach `mdadm --stop` keine Veränderungen an den zugrunde liegenden Partitionen durchgeführt haben, können Sie den RAID-Verbund mit `mdadm --assemble` ohne Datenverluste wieder zusammensetzen und aktivieren:

```

root# mdadm --assemble /dev/md0 /dev/sda3 /dev/sdc1
mdadm: /dev/md0 has been started with 2 drives.

```

**Partitionen analysieren** In allen Festplattenpartitionen, die Sie mit `mdadm` zu RAID-Partitionen zusammengefügt haben, wurden in einem speziellen Block Kontextinformationen (Metadaten)

gespeichert. Diese Informationen können Sie mit `mdadm --query` auslesen, beispielsweise um den Status eines unbekanntenen Systems zu ermitteln:

```
root# mdadm --query /dev/sda3
/dev/sda3: is not an md array
/dev/sda3: device 0 in 2 device active raid1 md0. Use mdadm --examine
for more detail.
root# mdadm --query /dev/md0
/dev/md0: 9.32GiB raid1 2 devices, 0 spares. Use mdadm --detail for more detail.
/dev/md0: No md super block found, not an md component.
```

`mdadm --examine` liefert Detailinformationen zu einer Partition, die Teil eines RAID-Verbunds ist:

```
root# mdadm --examine /dev/sda3
/dev/sda3:
    Raid Level : raid1
    Raid Devices : 2
    Total Devices : 2
    ...
Active Devices : 2
Working Devices : 2
    Number  Major  Minor  RaidDevice State
    0       0      3      0      active sync  /dev/sda3
    1       1     22      1      active sync  /dev/sdc1
```

Analog dazu liefert `mdadm --detail` Detailinformationen zu einem RAID-Verbund:

```
root# mdadm --detail /dev/md0
/dev/md0:
    Version : 00.90.03
    Creation Time : Thu Nov 9 16:55:35 2006
    Raid Level : raid1
    Array Size : 9775424 (9.32 GiB 10.01 GB)
    Device Size : 9775424 (9.32 GiB 10.01 GB)
    Raid Devices : 2
    Total Devices : 2
    ...
```

Woher wissen Sie, dass wirklich alle redundant gespeicherten Daten korrekt sind? Normalerweise führt das RAID-System Integritätstests nur durch, wenn es Dateien liest oder schreibt. Viele Dateien werden aber oft monatelang nicht angerührt. Um also mit Sicherheit festzustellen, dass die Festplatten in Ordnung sind, muss das RAID-System sämtliche Datenblöcke lesen und die redundanten Daten vergleichen. Dieser Vorgang wird auch *Scrubbing* genannt.

Kontrolle der  
Dateiintegrität

```
root# echo check > /sys/block/mdn/md<n>/sync_action
```

Wenn dabei Fehler auftreten, können diese repariert werden:

```
root# echo check > /sys/block/mdn/md<n>/sync_action
```

Unter Debian und Ubuntu kümmert sich darum das Script `/usr/share/mdadm/check-array`, das monatlich via `cron` gestartet wird. Dieselbe Funktion erfüllt unter Fedora das `cron`-Script `/etc/cron.weekly/99-raid-check`.

#### RAID-Metadaten löschen

Die Speicherung der RAID-Metadaten in ansonsten ungenutzten Sektoren der RAID-Partition ist normalerweise eine nützliche Sache. Wenn Sie die Festplatte zu einem späteren Zeitpunkt aber anders einsetzen möchten, können die RAID-Metadaten zum Problem werden: Linux-Installationsprogramme und `mdadm` erkennen die Überreste der RAID-Konfiguration und wollen partout nicht einsehen, dass diese Partitionen jetzt anders genutzt werden sollen. Abhilfe schafft das folgende Kommando, das auf alle RAID-Partitionen angewendet werden muss:

```
root# mdadm --zero-superblock /dev/sda3
```

Falls Sie auch mit BIOS-RAID experimentiert haben, können Sie dessen Metadaten auf allen Festplatten mit `dmraid -r -E` löschen.

## 22.18 Logical Volume Manager (LVM)

Der Logical Volume Manager setzt eine logische Schicht zwischen das Dateisystem und die Partitionen der Festplatte. Das Prinzip, die Vorzüge und die Nomenklatur von LVM wurden bereits in [Abschnitt 2.7](#), »RAID, LVM und Verschlüsselung«, erläutert. Dieser Abschnitt konzentriert sich auf die LVM-Administration.

#### Konfigurationshilfen

Manche Distributionen stellen grafische Werkzeuge zur Administration von LVM im laufenden Betrieb zur Verfügung. Bei Fedora und Red Hat hilft Ihnen `system-config-lvm` bei der Konfiguration, bei SUSE das YaST-Modul `SYSTEM • LVM`.

Auch wenn die Programme die LVM-Konfiguration erleichtern, setzen sie doch voraus, dass Sie die LVM-Konzepte verstanden haben. Beachten Sie, dass bei Größenänderungen in der Regel nur die *Logical Volumes* (LVs) geändert werden, nicht aber die darauf enthaltenen Dateisysteme. Deren Größe müssen Sie bei Verkleinerungen vorher, bei Vergrößerungen nachher selbst verändern.

#### dm\_mod

Linux-intern ist für LVM das Kernelmodul `dm_mod` zuständig. Bei manchen Distributionen sind die LVM-Funktionen direkt in den Kernel kompiliert und erscheinen daher nicht im `lsmod`-Ergebnis.

#### RAID

Sie können LVM und RAID kombinieren. Üblicherweise richten Sie dazu zuerst einen RAID-Verbund ein und nutzen dann das resultierende Device `/dev/mdN` als *Physical Volume* (PV).



Ein Sonderfall ist RAID-O. Diese RAID-Variante wird von LVM direkt unterstützt. Um diese Funktion nutzen zu können, müssen Sie auf zwei oder mehr Festplatten jeweils ein PV einrichten. Diese PVs werden zu einer *Volume Group* (VG) vereint. Nun können Sie mit `lvcreate -i N` ein LV einrichten, das die Daten auf mehrere PVs und damit auf N PVs verteilt.

Die LVM-Administration erfolgt durch eine ganze Palette von Kommandos. Die Namen der Kommandos beginnen mit `pv`, `vg` oder `lv`, je nachdem, ob sie zur Bearbeitung von Physical Volumes, Volume Groups oder Logical Volumes gedacht sind. Die wichtigsten Vertreter sind in [Tabelle 22.8](#) aufgezählt. Die Kommandos sind Teil des Pakets `lvm2`, das möglicherweise erst installiert werden muss.

LVM-Kommandos

| Kommando               | Funktion   |
|------------------------|--|
| <code>lvcreate</code>  | richtet ein neues LV in einer VG ein.                |
| <code>lvdisplay</code> | liefert Detailinformationen zu einem LV.             |
| <code>lvextend</code>  | vergrößert ein LV.                                   |
| <code>lvreduce</code>  | verkleinert ein LV.                                  |
| <code>lvremove</code>  | löscht ein LV.                                       |
| <code>lvrename</code>  | gibt dem LV einen neuen Namen.                       |
| <code>lvscan</code>    | listet alle LVs auf.                                 |
| <code>pvcreate</code>  | kennzeichnet eine Partition oder ein Device als PV.  |
| <code>pvdisk</code>    | liefert Detailinformationen zu einem PV.             |
| <code>pvremove</code>  | entfernt die PV-Kennzeichnung eines ungenutzten PVs. |
| <code>pvscan</code>    | listet alle PVs auf.                                 |
| <code>vgchange</code>  | ändert die Attribute einer VG.                       |
| <code>vgcreate</code>  | erzeugt eine neue VG aus einem oder mehreren PVs.    |
| <code>vgdisplay</code> | liefert Detailinformationen zu einer VG.             |
| <code>vgextend</code>  | vergrößert eine VG um ein PV.                        |
| <code>vgmerge</code>   | vereint zwei VGs.                                    |
| <code>vgreduce</code>  | verkleinert eine VG um ein ungenutztes PV.           |
| <code>vgrename</code>  | gibt einer VG einen neuen Namen.                     |
| <code>vgscan</code>    | listet alle VGs auf.                                 |

**Tabelle 22.8** LVM-Kommandoübersicht

Anstelle der Einzelkommandos können Sie die gesamte LVM-Administration auch mit dem Kommando `lvm` ausführen, wobei Sie als ersten Parameter den gewünschten Befehl übergeben. Die Kommandos `lvcreate` und `lvm lvcreate` sind also gleichwertig.

**Beispiele** Die folgenden Beispiele zeigen die Anwendung einiger LVM-Kommandos. Dabei gehe ich davon aus, dass während der Installation kein LVM eingerichtet wurde. Nun soll die zusätzliche Festplatte `/dev/sdc` via LVM genutzt werden. Die Partitionierung der Festplatte sieht so aus:

```
root# fdisk -l /dev/sdc
Disk /dev/sdc: 320.0 GB, 320072933376 bytes
   Device Boot      Start         End      Blocks   Id  System
/dev/sdc1             1         1217     9775521    8e  Linux LVM
/dev/sdc2           1218         2434     977552+    8e  Linux LVM
```

Um LVM zu initialisieren, führen Sie `modprobe` und `vgscan` aus. Sobald ein LVM-System eingerichtet ist, wird das LVM-Kernelmodul automatisch während des Rechnerstarts ausgeführt. Die manuelle Initialisierung ist also nur beim ersten Mal erforderlich:

```
root# modprobe dm_mod
root# vgscan
Reading all physical volumes (this may take a while...)
No volume groups found
```

Aus didaktischen Gründen richte ich LVM zuerst auf der Partition `/dev/sdc1` ein und erweitere das LVM-System später um `/dev/sdc2`. Wenn ohnedies klar ist, dass Sie die gesamte Festplatte für LVM nutzen möchten, ist es natürlich einfacher, gleich eine Partition in Maximalgröße mit `pvcreeate` für die LVM-Nutzung zu kennzeichnen:

```
root# pvcreate /dev/sdc1
Physical volume "/dev/sdc1" successfully created
```

Nun müssen alle PVs zu einer VG zusammengefasst werden. In diesem Beispiel gibt es zwar vorerst nur ein einziges PV, der Schritt ist aber dennoch erforderlich. An das Kommando `vgcreate` muss auch der gewünschte Name der VG übergeben werden. In diesem Beispiel bekommt die VG den Namen `myvg1`:

```
root# vgcreate myvg1 /dev/sdc1
Volume group "myvg1" successfully created
```

`myvg1` stellt jetzt eine Art Datenpool dar, der aber noch ungenutzt ist. Zur Nutzung müssen Sie innerhalb von `myvg1` ein LV einrichten, also eine Art virtueller Partition. Dazu müssen Sie an das Kommando `lvcreate` drei Informationen übergeben: die gewünschte Größe des LVs, den Namen des neuen LVs und den Namen der existierenden VG:

```
root# lvcreate -L 2G -n myvol1 myvg1
Logical volume "myvol1" created
```

Durch das Kommando wird gleichzeitig auch die Datei `/dev/myvg1/myvol1` erzeugt. Dabei handelt es sich um einen Link auf die Datei `/dev/mapper/myvg1-myvol1`. Das LV

kann jetzt unter einem dieser beiden Device-Namen wie eine gewöhnliche Festplattenpartition verwendet werden.

Um in einem Logical Volume ein Dateisystem einzurichten, verwenden Sie beispielsweise `mkfs.ext4` oder `mkfs.xfs`:

```
root# mkfs.ext4 /dev/myvg1/myvol1
```

Mit `mount` können Sie sich davon überzeugen, dass alles geklappt hat:

```
root# mkdir /test
root# mount /dev/myvg1/myvol1 /test
```

Ein Grund dafür, LVM überhaupt zu verwenden, besteht darin, ein Dateisystem nachträglich vergrößern zu können, ohne die Festplatte neu partitionieren zu müssen. Im folgenden Beispiel wird das vorhin eingerichtete Dateisystem (`/dev/myvg1/myvol1` via `/test`) von ursprünglich 2 GiB auf 3 GiB vergrößert. `df` zeigt die Kapazität von `/test` vor der Änderung:

Dateisystem  
vergrößern

```
root# df -h -T /test
Dateisystem  Typ   Größe Benut  Verf Ben% Eingehängt auf
/dev/mapper/myvg1-myvol1
              ext4   2,0G  760M  1,2G  40% /test
```

Dazu muss zuerst das Logical Volume vergrößert werden. Zu diesem Zweck müssen Sie den Device-Namen und die neue Größe an `lvextend` übergeben. Anschließend wird auch das `ext4`-Dateisystem entsprechend vergrößert.

```
root# lvextend -L 3G /dev/myvg1/myvol1
Extending logical volume myvol1 to 3,00 GB
Logical volume myvol1 successfully resized
root# resize2fs /dev/myvg1/myvol1
```

`df` beweist, dass alles funktioniert hat:

```
root# df -h -T /test
Dateisystem  Typ   Größe Benut  Verf Ben% Eingehängt auf
/dev/mapper/myvg1-myvol1
              ext4   3,0G  760M  2,1G  27% /test
```

Grundsätzlich ist auch eine Verkleinerung möglich. Allerdings müssen Sie dazu das betroffene Dateisystem zuerst aus dem Verzeichnisbaum lösen, mit `fsck.ext4` überprüfen und schließlich mit `resize2fs` verkleinern. Erst jetzt dürfen Sie mit `lvreduce` das zugrunde liegende LV verkleinern.

Solange im Speicherpool (in der Volume Group) noch Platz ist, können Logical Volumes leicht vergrößert werden. Aber was tun Sie, wenn auch die VG voll ist? In diesem Fall legen Sie auf einer beliebigen Festplatte Ihres Rechners eine neue Partition an, richten diese Partition als Physical Volume ein und fügen sie mit `vgextend` zur Volume Group hinzu.

Die beiden folgenden Kommandos demonstrieren dies für die Partition `/dev/sdc2`. `myvg1` bekommt damit eine Gesamtkapazität von rund 19 GiB, wovon 16 GiB frei sind:

```
root# pvcreate /dev/sdc2
Physical volume "/dev/sdc2" successfully created
root# vgextend myvg1 /dev/sdc2
Volume group "myvg1" successfully extended
root# vdisplay myvg1
...
VG Size                18,64 GB
Alloc PE / Size       640 / 2,50 GB
Free PE / Size        4132 / 16,14 GB
...
```

**Snapshots** Mit LVM können Sie Snapshots anlegen. Ein Snapshot ist ein unveränderliches Abbild des Dateisystems zu einem bestimmten Zeitpunkt. Der Snapshot kann wie ein eigenes Dateisystem in den Verzeichnisbaum integriert werden. Wenn sich das zugrunde liegende Dateisystem ändert, werden die originalen Daten für den Snapshot archiviert. Sie müssen bereits beim Anlegen des Snapshots angeben, wie viel Speicherplatz LVM für diesen Zweck reservieren soll. Ist dieser Speicherplatz erschöpft, wird der Snapshot ungültig und kann nicht mehr verwendet werden.

LVM-Snapshots bieten wesentlich weniger Funktionen als `btrfs`-Snapshots. LVM-Snapshots werden in der Regel für Backups verwendet. Sie stellen sicher, dass sich die Dateien während des Backups nicht ändern, das Backup also konsistent ist.

Die folgenden Kommandos zeigen, wie Sie zuerst einen Snapshot des LV `myvol1` erstellen, diesen im Verzeichnis `/media/backup` in den Verzeichnisbaum einbinden, ein Backup davon erstellen, den Snapshot wieder aus dem Verzeichnisbaum lösen und schließlich löschen. Während das Backup läuft, kann das LV `myvol1` uneingeschränkt weiterbenutzt werden (z. B. als Speicherplatz für einen Datenbank-Server). Die während des Backups durchgeführten Änderungen dürfen allerdings 100 MiB nicht überschreiten. Während des Backups können Sie mit `lvdisplay /dev/vg1/snap` ermitteln, wie viel Prozent dieses Speicherplatzes bereits in Verwendung sind.

Beachten Sie, dass Sie den Device-Namen des zugrunde liegenden Logical Volumes in der Form `/dev/vgname/lvname` angeben müssen, nicht in der Form `/dev/mapper/vgname-lvname`!

```
root# lvcreate -s -L 100M snap /dev/myvg1/myvol1
Logical volume snap created
root# mkdir /media/backup
root# mount /dev/vg1/snap /media/backup
root# backup-script /media/backup (Backup erstellen)
root# umount /media/backup
root# lvremove /dev/vg1/snap
```

## 22.19 SMART

SMART steht für *Self-Monitoring, Analysis and Reporting Technology* und ist ein Merkmal nahezu aller marktüblichen IDE-, SATA- und SCSI-Festplatten. Dank SMART werden verschiedene Parameter der Festplatte regelmäßig gespeichert. Diese Parameter erlauben einen Rückschluss auf eventuelle Defekte der Festplatte und auf ihre voraussichtliche Lebensdauer. Über eine spezielle Schnittstelle können die SMART-Parameter ausgelesen werden. Die regelmäßige Überwachung der Parameter durch das Betriebssystem ist eine Art Frühwarnsystem. Dieser Abschnitt gibt einen Überblick über die unter Linux verfügbaren Werkzeuge zum Auslesen der SMART-Parameter.

### Traum und Wirklichkeit

Idealerweise werden Sie dank SMART rechtzeitig informiert, bevor ein Datenträger ausfällt. Untersuchungen haben aber gezeigt, dass das in der Praxis selten funktioniert, weder bei herkömmlichen Festplatten noch bei SSDs! Das heißt nicht, dass Sie SMART nicht einsetzen sollen – aber verlassen Sie sich nicht darauf.

Damit SMART genutzt werden kann, müssen einige Voraussetzungen erfüllt sein:

Voraussetzungen

- ▶ Die Festplatte/SSD muss SMART unterstützen. Das können Sie beispielsweise mit `hdparm -I /dev/sdX` feststellen (`sda`, `sdb` etc.).
- ▶ Es muss sich um eine interne Festplatte handeln. Bei externen USB-Festplatten können die SMART-Funktionen leider nicht genutzt werden.
- ▶ Bei Festplatten, die über einen Hardware-RAID-Controller gesteuert werden, können die SMART-Funktionen nur in Einzelfällen genutzt werden. Details gibt man `smartctl` bei der Option `-d`.

Sie können den SMART-Status aber auch über die Kommandozeile ermitteln, was vor allem bei Server-Installationen wichtig ist. Das dazu erforderliche Kommando `smartctl` ist bei den meisten Distributionen Teil des Pakets `smartmontools`, das vielfach extra installiert werden muss. Je nach Distribution wird dabei automatisch auch ein E-Mail-Server (MTA) installiert, um SMART-Benachrichtigungen per E-Mail zu versenden. Auf einem Server ist das zweckmäßig, auf einem Desktop-Rechner hingegen zumeist nicht. Bei Debian und Ubuntu vermeiden Sie die Installation des E-Mail-Servers, wenn Sie an `apt` die Option `--no-install-recommends` übergeben.

`smartctl`

In der einfachsten Form liefert `smartctl` diverse Statusinformationen. Wenn `smartctl -i` in der letzten Zeile *SMART support is Disabled* meldet, aktivieren Sie SMART mit `smartctl -s on`.

```

root# smartctl -i /dev/sdb
Device Model:      SAMSUNG SSD 830 Series
Serial Number:    SOZ3NYAC210778
LU WWN Device Id: 5 002538 043584d30
Firmware Version: CXM03B1Q
User Capacity:    128.035.676.160 bytes [128 GB]
Sector Size:      512 bytes logical/physical
Device is:        Not in smartctl database [for details use: -P showall]
ATA Version is:   8
ATA Standard is:  ACS-2 revision 2
SMART support is: Available - device has SMART capability.
SMART support is: Enabled

```

smartctl -H bzw. smartctl --health gibt an, ob die Festplatte momentan in Ordnung ist und voraussichtlich die nächsten 24 Stunden noch funktionieren wird. Sollte smartctl hier nicht *PASSED* als Ergebnis liefern, sollten Sie *sofort* damit beginnen, ein komplettes Backup durchzuführen!

```

root# smartctl -H /dev/sda
...
SMART overall-health self-assessment test result: PASSED

```

smartctl -A bzw. smartctl --attributes liefert eine Liste von herstellerspezifischen Festplattenattributen. Für diese Attribute existiert kein festgeschriebener Standard, die wichtigsten Attribute werden aber von vielen Festplattenherstellern unterstützt. Bei der Interpretation der Werte sind zwei Spalten entscheidend: *VALUE* gibt den aktuellen Wert an, *THRESH* den Grenzwert. Wenn der aktuelle Wert den Grenzwert unterschreitet, sind Probleme zu erwarten bzw. hat die Festplatte ihre vorgesehene Lebensdauer erreicht.

Die Werte sind auf einen Basiswert von 100 normalisiert. Beispielsweise beginnt *Power\_On\_Hour* bei einer neuen Festplatte mit dem Wert 100. Nach einer bestimmten Anzahl von Betriebsstunden sinkt der Wert auf 99 etc. Die bisher absolvierten Betriebsstunden gehen aus der *RAW\_VALUE*-Spalte hervor. Bei der Testfestplatte lautet der Wert 451, das sind ca. 56 Arbeitstage zu je 8 Stunden. Manche Festplatten messen die Betriebsdauer in Minuten oder Sekunden. In diesem Fall erreichen Sie durch *-v 9,minutes* oder *-v 9,seconds* eine korrekte Anzeige.

Die folgenden, gekürzten Ergebnisse stammen von einer SSD mit gut 10.000 Betriebsstunden. Es gibt keinerlei Anzeichen für Probleme.

```

root# smartctl -A /dev/sda
...
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          VALUE WORST THRESH TYPE      RAW_VALUE
  1 Raw_Read_Error_Rate      100   100   000   Pre-fail  0
  5 Reallocate_NAND_Blk_Cnt  100   100   010   Old_age   0

```

```

 9 Power_On_Hours          100 100 000 Old_age 10802
12 Power_Cycle_Count      100 100 000 Old_age 4
173 Ave_Block-Erase_Count 094 094 000 Old_age 129
174 Unexpect_Power_Loss_Ct 100 100 000 Old_age 2
180 Unused_Reserve_NAND_Blk 000 000 000 Pre-fail 5574
202 Percent_Lifetime_Used 094 094 001 Old_age 6
246 Total_Host_Sector_Write 100 100 000 Old_age 22518536407
247 Host_Program_Page_Count 100 100 000 Old_age 723834105
248 Bckgnd_Program_Page_Cnt 100 100 000 Old_age 3649451105

```

Das zweite Listing wurde für eine Festplatte ermittelt, die bereits Ausfälle gezeigt hat und aus einem RAID-Array entfernt wurde:

```

root# smartctl -A /dev/sda
...
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          VALUE WORST THRESH TYPE      WHEN_FAILED RAW_VALUE
 1 Raw_Read_Error_Rate      105  077  006  Pre-fail -        83304216
 3 Spin_Up_Time             094  094  000  Pre-fail -         0
 4 Start_Stop_Count         100  100  020  Old_age -         7
 5 Reallocated_Sector_Ct    075  051  036  Pre-fail -       32936
 7 Seek_Error_Rate          085  060  030  Pre-fail -     384937741
 9 Power_On_Hours           063  063  000  Old_age -       32533
10 Spin_Retry_Count         100  100  097  Pre-fail -         0
12 Power_Cycle_Count        100  100  020  Old_age -         7
183 Runtime_Bad_Block        098  098  000  Old_age -         2
184 End-to-End_Error         094  094  099  Old_age FAILING_NOW 6
...

```

`smartctl -l error` liefert Informationen über die fünf zuletzt aufgetretenen Fehler. Oft ist das Ergebnis einfach leer (*No Errors Logged*). Vereinzelte Fehler, die sich nicht wiederholen, sind im Regelfall kein Anlass zur Beunruhigung.

```

root# smartctl -l error /dev/sda
SMART Error Log Version: 1
No Errors Logged

```

SMART sieht verschiedene Varianten von Selbsttests vor, um den aktuellen Zustand der Festplatte noch genauer zu ermitteln. Derartige Tests starten Sie mit `smartctl -t short/long`. Ein kurzer Test dauert wenige Minuten, ein ausführlicher Test (`long`) unter Umständen mehrere Stunden. Der Test wird im Hintergrund durchgeführt; Sie können ganz normal weiterarbeiten.

**Selbsttest durchführen**

Nachdem das Testende erreicht ist, sehen Sie sich mit `smartctl -l selftest` das Ergebnis an. Die Spalte *Remaining* besagt dabei, wie weit der Selbsttest bereits ausgeführt ist. Wenn der Wert größer als 0 Prozent ist, läuft der Test noch! *LifeTime* gibt an, wie viele Stunden die Festplatte bereits im Betrieb war. *LBA* gibt den Ort (Sektor) des ersten Fehlers an. Im folgenden Ergebnis wurden drei Selbsttests ausgeführt: einer

unmittelbar nach dem Einbau der Platte nach 40 Betriebsstunden, die anderen beiden nach ca. 2600 Stunden.

```
root# smartctl -t short /dev/sda
root# smartctl -l selftest /dev/sda
Num Test_Description Status Remaining LifeTime LBA
# 1 Extended offline Completed without error 00% 2592 -
# 2 Short offline Completed without error 00% 2591 -
# 3 Short offline Completed without error 00% 40 -
```

#### Automatische Überwachung (smartd)

smartctl ist sicherlich ein interessantes Werkzeug, um Informationen über die Festplatte zu sammeln. Für eine regelmäßige Überwachung aller Festplatten ist das Kommando aber zu unhandlich. Diese Aufgabe übernimmt das Programm smartd. Dabei handelt es sich um einen Dämon (Systemdienst). Die Kommandos für den automatischen Start variieren je nach Distribution und sind in [Abschnitt 12.5](#), »Systemprozesse (Dämonen)«, zusammengefasst.

smartd wird durch `/etc/smartd.conf` gesteuert. Bei einigen Distributionen wertet das Init-Script auch die Dateien `/etc/sysconfig/smartmontools` oder `/etc/default/smartmontools` aus. Diese Dateien enthalten zusätzliche Kommandooptionen für smartd. Bei Debian und Ubuntu müssen Sie in `smartmontools` die Einstellung `start_smartd=yes` vornehmen!

Eine einfache Konfiguration für einen Rechner mit zwei SATA-Festplatten (`/dev/sda` und `/dev/sdb`) sieht folgendermaßen aus:

```
# Datei /etc/smartd.conf
/dev/sda -d sat -H -m root -M test
/dev/sdb -d sat -H -m root -M test
```

Das bedeutet, dass die »Gesundheit« der angegebenen Festplatten alle halbe Stunde überwacht wird (wie durch `smartctl -H`). Wird dabei ein Fehler festgestellt, sendet smartd eine E-Mail an den lokalen Benutzer root. (Das setzt allerdings einen lokalen E-Mail-Server voraus.) `-d sat` kennzeichnet die Festplatten als SATA-Geräte. `-M test` dient zum Testen, ob der E-Mail-Versand prinzipiell funktioniert. Starten Sie smartd:

```
root# systemctl start smartmontools
```

Wenn Sie die Test-E-Mail erhalten haben, entfernen Sie `-M test` aus der Konfiguration. Eine Menge weiterer Konfigurationsbeispiele finden Sie in der Datei `smartd.conf`, die mit `smartmontools` mitgeliefert wird.



## 22.20 SSD-TRIM

Linux läuft auf SSDs (Solid State Disks) vollkommen problemlos – und natürlich viel schneller als auf herkömmlichen Festplatten. Allerdings unterscheidet sich die Nutzung von SSDs in einem Punkt von herkömmlichen Festplatten: Für die interne Optimierung der Speicherzellen ist es erforderlich, dass das Betriebssystem die SSD darüber informiert, welche Speicherblöcke des Dateisystems momentan ungenutzt sind (etwa, weil eine Datei gelöscht wurde). Dieser Vorgang wird wie das zugrundeliegende Kommando »TRIM« genannt. Die technischen Hintergründe können Sie in der Wikipedia nachlesen:

<https://de.wikipedia.org/wiki/Solid-State-Drive>

Nur wenige Linux-Distributionen führen standardmäßig die TRIM-Funktion aus. Das lässt sich damit begründen, dass der daraus resultierende Performance-Gewinn bei modernen SSDs und normaler Nutzung relativ gering ist. Zudem gibt es unterschiedliche Verfahren, das SSD-TRIM durchzuführen – und jede ist mit Vor- und Nachteilen verbunden.

Linux-Profis, denen die optimale Geschwindigkeit ihrer SSD ein Anliegen ist, müssen sich selbst um das SSD-TRIM kümmern. Sie haben die Wahl zwischen zwei Varianten: dem Online-TRIM, bei dem Linux die SSD bei jeder gelöschten Datei sofort benachrichtigt, oder dem Batch-TRIM, das in regelmäßigen Abständen (z. B. einmal pro Woche) durchgeführt wird.

TRIM-Varianten

### Online-TRIM oder Batch-TRIM?

Online-TRIM hat den Nachteil, dass die SSD-internen Aufräumarbeiten gerade dann ausgeführt werden, wenn die SSD ohnedies beschäftigt ist – nämlich während intensiver Schreibvorgänge. Das Online-TRIM verlangsamt *jeden* Schreibvorgang.

Gegen das Batch-TRIM spricht der Umstand, dass dabei alle ausstehenden TRIM-Operationen eines ganzen Tages oder einer ganzen Woche ausgeführt werden. Während dieser Prozess läuft, ist jeder SSD-Zugriff deutlich langsamer.

Persönlich ist mir das Batch-TRIM lieber. Damit kann ich bei Servern den Batch-TRIM in der Nacht ausführen, womit der laufende Betrieb kaum beeinträchtigt wird.

Um das Online-TRIM-Verfahren zu aktivieren, müssen Sie in die Datei `/etc/fstab` für die betroffenen Dateisysteme die Option `discard` einfügen. Diese Option steht für die Dateisysteme `ext4`, `btrfs` und `xfs` zur Verfügung. Achten Sie darauf, dass Sie in `/etc/fstab` keine Syntaxfehler einbauen! Die Liste der `mount`-Optionen darf kein Leerzeichen enthalten.

Online-TRIM

```
# Datei /etc/fstab
UUID=018e... / ext4 errors=remount-ro,user_xattr,discard 0 1
```

**Batch-TRIM** Ein manuelles Batch-TRIM stoßen Sie im Terminalfenster mit dem Kommando `fstrim` an:

```
root# fstrim -v /
/: 6493835264 bytes were trimmed
```

Sie können das TRIM-Kommando auch auf größere Datenblöcke einschränken. Damit werden nur freie Datenblöcke ab der angegebenen Mindestgröße an die SSD gemeldet. `fstrim` kann dann wesentlich schneller ausgeführt werden und erzielt dennoch eine sehr gute Wirkung.

```
root# fstrim -m 64K -v /
```

Um den Prozess zu automatisieren, richten Sie eine neue Cron-Systemdatei ein (optional mit der Option `-m`). Falls Sie Ihr Dateisystem über mehrere Partitionen verteilt haben, müssen Sie `fstrim` für jeden `mount`-Punkt ausführen.

```
# Datei /etc/cron.weekly/fstrim
/sbin/fstrim /
```

#### Einschränkungen und Sonderfälle

TRIM funktioniert auch im Zusammenspiel mit LVM und Software-RAID. Wenn Sie Ihr Dateisystem verschlüsseln, ist die Freigabe ungenutzter Datenblöcke durch TRIM zwar grundsätzlich möglich, die Sicherheit der verschlüsselten Daten wird dadurch aber vermindert.

Grundsätzlich kann die `mount`-Option `discard` auch für die Swap-Partition verwendet werden. Bei Desktop-Systemen ist das nicht erforderlich, weil Linux bei jedem Rechnerstart die gesamte Swap-Partition per TRIM freigibt. Bei einem Server, der über Monate läuft, kann die Option aber unter Umständen zweckmäßig sein.

## 22.21 Verschlüsselung

Notebooks und USB-Sticks können verloren gehen bzw. werden gestohlen. Schlimmer als der eigentliche Verlust des Geräts ist oft der Umstand, dass damit wichtige Daten in fremde Hände geraten: der Zugang zum Online-Banking, Versicherungsnummern, Krankenakten, Firmengeheimnisse, militärisch relevante Informationen etc. Das ist unnötig. Eine relativ simple Verschlüsselung des Dateisystems reicht aus, um die Daten wirksam zu schützen. Dieser Abschnitt gibt einige Hintergrundinformationen zum Umgang mit verschlüsselten Dateien und Dateisystemen.

## Einzelne Dateien verschlüsseln

Eine einzelne Datei verschlüsseln Sie am einfachsten mit dem Kommando `gpg`. `gpg -c` fordert Sie zweimal zur Angabe eines Passworts auf, verschlüsselt dann die angegebene Datei und speichert das Ergebnis unter dem Namen `datei.gpg`. Dabei kommt standardmäßig der Verschlüsselungsalgorithmus CAST5 zur Anwendung. Die ursprüngliche Datei können Sie nun löschen. `gpg -d` stellt die Datei wieder her.

```
user$ gpg -c datei
Geben Sie die Passphrase ein: *******
Geben Sie die Passphrase nochmals ein: *******
user$ gpg -d datei.gpg > datei
Geben Sie die Passphrase ein: *******
```

`gpg` kann zur Codierung bzw. Decodierung auch einen öffentlichen bzw. privaten Schlüssel verwenden, kann Dateien signieren, Schlüssel verwalten etc. Die Beschreibung der unzähligen Optionen in der `man`-Seite ist dementsprechend rund 50 Seiten lang. Die manuelle Verwendung von `gpg` ist aber eher unüblich. Häufiger wird `gpg` von E-Mail-Clients eingesetzt, um (mehr oder weniger automatisch) E-Mails zu signieren oder zu verschlüsseln.

## Ein Dateisystem verschlüsseln (USB-Stick, externe Festplatte)

In der Vergangenheit wurden unzählige Verfahren zur Verschlüsselung von Dateisystemen entwickelt: CryptoFS, eCryptfs, Enc-FS, Loop-AES und LUKS. Ein Teil dieser Verfahren ist noch immer im Einsatz, andere wurden – oft aus Sicherheitsgründen – wieder verworfen. Momentan ist das *Linux Unified Key Setup* (kurz LUKS) die populärste Spielart.

`dm_crypt` und  
LUKS

LUKS basiert auf dem Kernelmodul `dm_crypt`, das den auch für LVM eingesetzten Linux-Device-Mapper um Kryptografiefunktionen erweitert. Das Modul ist eine logische Schicht zwischen den verschlüsselten Rohdaten auf der Festplatte und dem Dateisystem, so wie es der Linux-Anwender sieht. `dm_crypt` unterstützt diverse Verschlüsselungsalgorithmen. `dm_crypt` wird oft mit LVM kombiniert, das ist aber keineswegs notwendig. Sie können `dm_crypt` auch auf einem LVM-freien System einsetzen.

LUKS fügt den verschlüsselten Daten einen Header mit Metainformationen hinzu. Der Header gibt unter anderem an, mit welchem Verfahren die Daten verschlüsselt sind. LUKS vereinfacht die Integration von verschlüsselten Datenträgern in Linux ganz erheblich.

Um verschlüsselte Dateisysteme einzurichten, nehmen Sie das Kommando `cryptsetup` aus dem gleichnamigen Paket zu Hilfe. Die folgenden Zeilen zeigen, wie Sie einen USB-Stick (`/dev/sdh1`) zuerst als Crypto-Device formatieren (`luksFormat`) und das Device dann unter dem willkürlich gewählten Namen `mycontainer` aktivieren

`cryptsetup`

(luksOpen). Naturgemäß sind Ihre Daten nur so sicher wie Ihr Passwort bzw. die aus mehreren Wörtern bestehende Passphrase. Empfohlen wird eine Passwortlänge von zumindest 20 Zeichen.

Anschließend können Sie `/dev/mapper/mycontainer` wie eine Festplattenpartition oder ein LV nutzen – also ein Dateisystem einrichten, dieses in den Verzeichnisbaum einbinden etc. Nach `umount` müssen Sie daran denken, das Crypto-Device wieder zu deaktivieren (`luksClose`), um `/dev/sdh1` freizugeben. Erst jetzt dürfen Sie den USB-Stick ausstecken.

```
root# cryptsetup luksFormat /dev/sdh1
Daten auf /dev/sdh1 werden unwiderruflich überschrieben.
Are you sure? (Type uppercase yes): YES
Enter LUKS passphrase: *****
Verify passphrase: *****
Command successful.
root# cryptsetup luksOpen /dev/sdh1 mycontainer
Enter LUKS passphrase: *****
root# mkfs.ext4 /dev/mapper/mycontainer
root# mount /dev/mapper/mycontainer /test
root# touch /test/xy
root# umount /test/
root# cryptsetup luksClose mycontainer
```

Selbstverständlich können Sie statt eines USB-Sticks auch eine Partition einer internen oder externen Festplatte, ein RAID-Device oder ein Logical Volume Ihres LVM-Systems verwenden. Dazu ersetzen Sie einfach `/dev/sdh1` durch den Device-Namen der Partition bzw. des LVs.

Standardmäßig verwendet `cryptsetup` den Verschlüsselungsalgorithmus AES mit einer Schlüssellänge von 128 Bit. Sie können sich davon mit `cryptsetup luksDump` überzeugen: Dieses Kommando liefert die Crypto-Metainformationen, die LUKS in einem speziellen Sektor des Datenträgers speichert.

```
root# cryptsetup luksDump /dev/sdh1
LUKS header information for /dev/sdh1
Version:          1
Cipher name:      aes
Cipher mode:      cbc-essiv:sha256
Hash spec:        sha1
Payload offset:   1032
MK bits:          128
...
```

Wenn Sie einen anderen Verschlüsselungsalgorithmus oder einen längeren Schlüssel einsetzen möchten, übergeben Sie die gewünschten Daten mit den Optionen `-c` und `-s` an `cryptsetup luksFormat`. Welche Algorithmen zur Auswahl stehen, verrät `cat`

`/proc/crypto`. Als sicher gelten zurzeit die Algorithmen AES und TwoFish. Beachten Sie, dass Verschlüsselungsalgorithmen ein Forschungsgebiet sind, in dem sich vieles schnell ändert: Immer wieder stellen sich Algorithmen als weniger sicher als gedacht heraus. Eine kurze Beschreibung der Algorithmen können Sie im Gentoo-Wiki nachlesen:

[https://wiki.gentoo.org/wiki/Dm-crypt\\_full\\_disk\\_encryption](https://wiki.gentoo.org/wiki/Dm-crypt_full_disk_encryption)

Mit `cryptsetup luksAddKey` können Sie den Zugriff auf ein LUKS-Device durch insgesamt acht verschiedene Passwörter absichern. Das erlaubt die gemeinsame Nutzung eines Datenträgers, bei der jeder Benutzer sein eigenes Passwort verwendet.

`luksFormat` erleichtert das Einrichten einer verschlüsselten Partition oder eines verschlüsselten Datenträgers ein wenig. Das Kommando führt zuerst `cryptsetup luksFormat` und dann `mkfs.vfat` auf. Wenn Sie einen anderen Dateisystemtyp wünschen, müssen Sie ihn mit `-t` angeben.

luksFormat

### Tipp

Nach der Ausführung des Kommandos (oft aber auch dann, wenn Fehler aufgetreten sind) bleibt ein aktives Crypto-Device `/dev/mapper/luksformatN` zurück. Bevor Sie den Datenträger entfernen oder nochmals als Crypto-Device einrichten können, müssen Sie `cryptsetup luksClose luksFormatN` ausführen!

Wenn Sie LUKS-formatierte externe Datenträger mit Ihrem Rechner verbinden und unter Gnome oder KDE arbeiten, wird der Datenträger automatisch als Crypto-Device erkannt. Es erscheint ein Dialog, in dem Sie das Verschlüsselungspasswort angeben müssen. Anschließend wird der Datenträger in das Dateisystem eingebunden. Der Container-Name für `/dev/mapper` lautet `luks_crypto_NNN`. Beim Aushängen wird auch `luksClose` ausgeführt – die Nutzung des verschlüsselten Datenträgers könnte nicht einfacher sein!

Desktop-Nutzung

Wenn Sie ein verschlüsseltes Dateisystem in einer Partition einer lokalen Festplatte eingerichtet haben, wollen Sie vermutlich, dass dieses Dateisystem beim Hochfahren des Rechners in den Verzeichnisbaum eingebunden wird. Zur Automatisierung dieses Vorgangs enthält das Paket `cryptsetup` bereits die erforderlichen `Initrd`- und `Init-V`-Scripts. Diese setzen allerdings voraus, dass das Crypto-Device in die Datei `/etc/crypttab` eingetragen wird.

crypttab

Der Aufbau dieser Datei ist einfach: Die erste Spalte gibt den gewünschten Namen für `/dev/mapper` an, die zweite Spalte den Device-Namen, die dritte Spalte die Datei, aus der der Schlüssel gelesen werden soll (z. B. von einem USB-Stick), oder `none`, wenn das Verschlüsselungspasswort interaktiv eingegeben wird, und die vierte Spalte enthält Optionen.

Im folgenden Beispiel soll das Device `/dev/sda7` unter dem Namen `/dev/mapper/cdisk1` eingerichtet werden. Das Passwort soll während des Rechnerstarts angegeben werden, und das Crypto-Device wurde mit LUKS eingerichtet. Eine Menge weiterer Optionen sind in `man crypttab` beschrieben.

```
# Datei /etc/crypttab
# Mapper-Name Device      Schlüsseldatei  Optionen
cdisk1      /dev/sda7      none           luks
```

Damit das Crypto-Device nicht nur aktiviert, sondern sein Dateisystem auch in den Verzeichnisbaum eingebunden wird, muss auch `/etc/fstab` ergänzt werden. Die folgende Zeile bewirkt, dass das Dateisystem über das Verzeichnis `/media/private-data` genutzt werden kann:

```
# Datei /etc/fstab
...
/dev/mapper/cdisk1 /media/private-data ext4 defaults 0 0
```

Anschließend starten Sie den Rechner neu und testen, ob alles funktioniert.

**Nachteile** Die Verschlüsselung einer Partition ist mit Nachteilen verbunden. Zum einen erfolgen sämtliche Dateioperationen spürbar langsamer als im Normalbetrieb – und umso langsamer, je aufwendiger (und sicherer) das Verschlüsselungsverfahren ist. Verwenden Sie nach Möglichkeit eine schnelle CPU mit mehreren Cores! Zum anderen müssen Sie bei jedem Boot-Vorgang das Passwort eingeben. Das ist nicht nur lästig, sondern macht auch einen Neustart in Abwesenheit unmöglich. Grundsätzlich werden Sie mit verschlüsselten Partitionen zumeist nur auf lokalen PCs arbeiten, nicht oder nur in Ausnahmefällen auf Servern.

**ecryptfs (Ubuntu)** Ubuntu bietet die Möglichkeit, nur das Heimatverzeichnis zu verschlüsseln. Das verschlüsselte Verzeichnis wird beim Login automatisch in das Dateisystem eingebunden und beim Logout wieder entfernt. Intern kommen dabei nicht `dm_crypt` und LUKS zum Einsatz, sondern das Dateisystem `ecryptfs`. Persönlich bin ich kein Fan von diesem Verschlüsselungsverfahren, weil es bei Hardware- oder Boot-Problemen sehr schwierig ist, die verschlüsselten Daten zu retten (selbst dann, wenn Sie Ihren Schlüssel kennen).

**ext4-Verschlüsselung** Der im Kernel enthaltene ext4-Treiber enthält seit Version 4.1 ebenfalls Verschlüsselungsfunktionen, die auf Verzeichnisebene arbeiten. Damit können verschiedene (Heimat-)Verzeichnisse unterschiedlich verschlüsselt werden. Außerdem ist es nicht erforderlich, gleich das ganze Dateisystem zu verschlüsseln.

Die Funktionen wurden primär für den Einsatz in Android entwickelt. Es ist aber zu erwarten, dass die Verschlüsselungsfunktionen früher oder später auch in gewöhnlichen Linux-Distributionen zum Einsatz kommen.

Weitere Informationen können Sie hier nachlesen:

<https://lwn.net/Articles/639427>

<https://askubuntu.com/questions/643577>

<http://blog.quarkslab.com/a-glimpse-of-ext4-filesystem-level-encryption.html>

## Gesamtes System verschlüsseln

Mit der im vorigen Abschnitt vorgestellten Datei `/etc/crypttab` ist es nur noch ein kleiner Schritt von der Verschlüsselung einer lokalen Partition (z. B. `/home`) zur Verschlüsselung des gesamten Systems inklusive der Systempartition. Zwei Details sind wichtig: Zum einen kann GRUB auf die verschlüsselten Daten nicht zugreifen – deswegen ist unbedingt eine eigene, nicht-verschlüsselte Boot-Partition erforderlich. Zum anderen ist zum Zugriff auf die Systempartition die Eingabe des Verschlüsselungspassworts notwendig. Die dafür notwendigen Funktionen müssen als Scripts in die `initrd`-Datei integriert werden. (Das `cryptsetup`-Paket enthält alle erforderlichen Dateien.)

Vorweg ist aber zu klären, wer die Verschlüsselung des gesamten Systems überhaupt braucht: Eigentlich sollte es ja reichen, nur die privaten Dateien in `/home` zu verschlüsseln. Allerdings kann auch die Systempartition für den Notebook-Dieb, so er denn tatsächlich an den Daten interessiert ist, aufschlussreich sein: `/var/cache` oder `/var/tmp` können Überreste von versandten E-Mails, ausgedruckten Dokumenten, gelesenen PDFs etc. enthalten; `/var/log` dokumentiert, wer wann auf dem Computer gearbeitet hat; die Swap-Partition enthält womöglich ausgelagerte Datenblöcke mit sicherheitskritischen Informationen etc. Kurzum: Wenn Sie Ihre Daten bzw. Ihre Privatsphäre am Rechner wirklich vollständig schützen wollen, müssen Sie wohl oder übel das gesamte System verschlüsseln.

Die meisten großen Distributionen bieten im Installationsprogramm eine Option, um das gesamte System zu verschlüsseln. Allerdings müssen Sie bei Distributionen, die mehrere Installationsverfahren oder -medien zur Wahl stellen, in der Regel die traditionelle Variante verwenden. Installationsprogramme, die von einer Live-CD oder -DVD starten, sind teilweise ungeeignet!

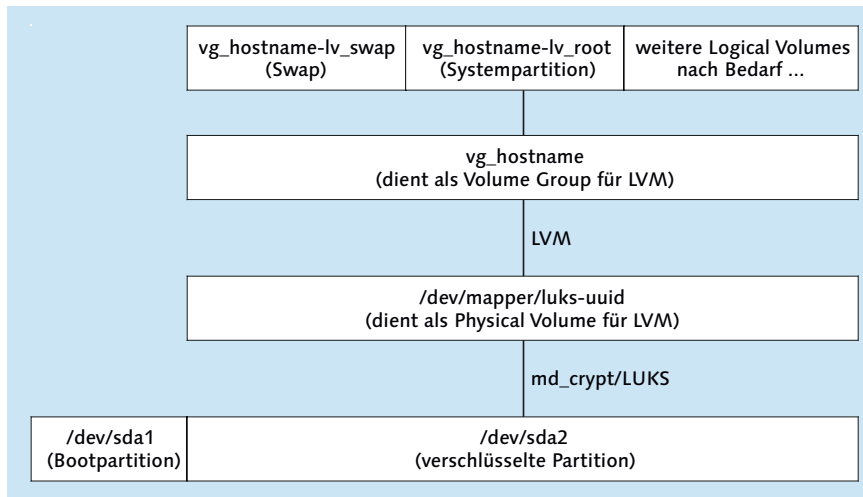
Installation

Bei Debian und Ubuntu wählen Sie die Partitionierungsvariante **VERSCHLÜSSELTES LVM-SYSTEM**, bei Fedora aktivieren Sie im Partitionierungsdialog die Option **VERSCHLÜSSELTES SYSTEM**. Bei openSUSE wählen Sie bei der Partitionierung die Optionen **LVM-BASIERT** und **VERSCHLÜSSELT**.

Der Aufbau des verschlüsselten Systems sieht bei den meisten Distributionen einheitlich aus: Es wird eine unverschlüsselte Boot-Partition für GRUB eingerichtet sowie eine zweite Partition, die verschlüsselt ist und als Physical Volume für LVM dient. Auf

Systemaufbau

diese Weise sind alle via LVM eingerichteten Partitionen (Swap-Partition, Systempartition, Datenpartitionen) automatisch verschlüsselt. Außerdem muss nicht für jede Partition ein eigenes Passwort definiert werden; vielmehr reicht ein zentrales Passwort für das gesamte LVM-System. [Abbildung 22.4](#) zeigt den schematischen Aufbau eines derartigen Systems, wobei ich die Bezeichnung der Devices bzw. LVs von Fedora übernommen habe.



**Abbildung 22.4** Vollständig verschlüsseltes Linux-System

Natürlich gibt es zu dem in [Abbildung 22.4](#) präsentierten Aufbau viele Alternativen. Eine mögliche Variante besteht darin, auf LVM zu verzichten und jede Partition für sich zu verschlüsseln. Für die Swap-Partition kann dabei ein Zufallsschlüssel verwendet werden, der bei jedem Systemstart neu aus `/dev/urandom` erzeugt wird.

Denkbar ist auch eine andere Form der Schlüsselangabe: Statt interaktiv ein Passwort einzugeben, kann der Schlüssel während des Boot-Prozesses aus einer Datei eines USB-Sticks gelesen werden. Der USB-Stick dient dann gewissermaßen als Hardware-Schlüssel, der zum Booten des Rechners erforderlich ist. Auch manche Kartenlesegeräte lassen sich unter Linux nutzen, die Integration in die Verschlüsselungs-Software erfordert aber Handarbeit.

#### SSDs verschlüsseln

Grundsätzlich können Sie die hier beschriebenen Verschlüsselungsmethoden gleichermaßen für herkömmliche Festplatten und für SSDs verwenden. Prinzipbedingt ist die Verschlüsselung eines Dateisystems auf einer SSD aber mit zwei großen Nachteilen verbunden:



- ▶ Aus Sicherheitsgründen sollte immer der gesamte Inhalt des Dateisystems verschlüsselt werden, also auch die Datenblöcke, die momentan gar nicht genutzt werden. Sicherheitstechnisch ist es deswegen nicht zweckmäßig, der SSD per TRIM freie Datenblöcke zu melden. Langfristig wird die Performance der SSD darunter aber leiden.
- ▶ Manche SSDs versuchen, die zu speichernden Daten vorher zu komprimieren. Bei verschlüsselten Daten ist eine Komprimierung aber unmöglich, was bei derartigen SSDs zu Geschwindigkeitseinbußen führt.

Den besten Ausweg aus diesem Dilemma bieten SSDs, bei denen eine Verschlüsselung auf Hardware-Ebene durchgeführt werden kann. Die Kontrolle über die Verschlüsselung erfolgt dann nicht durch Linux, sondern durch das EFI des Mainboards. Es lässt sich allerdings nur sehr schwer beurteilen, wie sicher die in die SSD eingebauten Verschlüsselungsalgorithmen sind.



# Kapitel 23

## GRUB

Das Programm GRUB ist ein sogenannter Bootloader. Es wird als erstes Programm nach dem Einschalten des Rechners gestartet. Bei BIOS-Rechnern können Sie nun in einem Menü zwischen Linux und Windows wählen. Bei EFI-Rechnern spielt diese Wahlfunktion keine große Rolle mehr, GRUB übernimmt aber auch dort die Aufgabe, Linux an sich zu starten und diverse Parameter an den Kernel zu übergeben.

### 23.1 GRUB-Grundlagen

GRUB steht für *Grand Unified Bootloader*. Seit 2012 ist GRUB in der Version 2.0 verfügbar. Dieses Kapitel beschreibt ausschließlich diese Version. **GRUB-Versionen**

Die Vorgängerversion hat die merkwürdige Bezeichnung GRUB 0.97 *legacy*. Der Zusatz *legacy* weist darauf hin, dass diese Version schon seit Jahren nicht mehr weiterentwickelt wird. Und die Versionsnummer 0.97 macht klar, dass Version 1.0 nie erreicht wurde. Auf GRUB 0.97 stoßen Sie heute nur noch in älteren Langzeitdistributionen, z. B. in RHEL 6 bzw. CentOS 6.

Die für GRUB 2 notwendigen Dateien sind oft über mehrere Pakete verteilt: Bei Debian und Ubuntu enthält `grub-common` plattformunabhängige Konfigurationsdateien und Kommandos, und `grub-pc` enthält die BIOS-spezifischen Dateien. Für Rechner, die statt des BIOS auf EFI setzen, ist statt `grub-pc` das Paket `grub-efi-amd64` oder `grub-efi-ia32` erforderlich. Zu guter Letzt enthält `grub-rescue-pc` eine IMG- und eine ISO-Datei, um ein GRUB-Rescue-System auf einem USB-Stick oder einer CD zu speichern. Das ermöglicht es im Notfall, GRUB vom USB-Stick zu starten und dann durch die manuelle Eingabe von GRUB-Kommandos bzw. durch die Veränderung der vorgesehenen Menüeinträge das System zu starten. **Komponenten und Pakete**

Fedora hat die Filetierung in unzählige Pakete vermieden: `grub2` enthält GRUB für BIOS-Rechner, und `grub2-efi` enthält die EFI-kompatible Version. Für die Aktualisierung der GRUB-Konfiguration nach Kernel-Updates sorgen die Fedora-spezifischen Scripts des Pakets `grubby`.

GRUB 2 setzt unabhängig von der Distribution die Installation des Pakets `os-prober` voraus. Das gleichnamige Kommando sucht auf allen erreichbaren Partitionen nach Betriebssystemen. Das Ergebnis von `os-prober` fließt in das automatisch erzeugte GRUB-Menü ein.

## EFI-Systemstart

Bevor die Einzelheiten der GRUB-Installation und -Konfiguration behandelt werden, ist es sinnvoll, sich ein Bild davon zu machen, was während des Boot-Vorgangs passiert. Der Startprozess unterscheidet sich stark, je nachdem, ob auf Ihrem Rechner ein traditionelles BIOS oder das modernere EFI läuft.

### Parallel- installationen

Das *Extensible Firmware Interface* (EFI) vereinfacht die Parallelinstallation mehrerer Betriebssysteme. Während das BIOS grundsätzlich nur die Installation *eines* Betriebssystems vorsah und jede Parallelinstallation einen Bootmanager voraussetzte, unterstützt EFI von sich aus die Installation mehrerer Betriebssysteme: Jedes Betriebssystem kann seinen eigenen Bootloader in ein eigenes Verzeichnis innerhalb der EFI-Partition speichern.

Beim Start des Rechners sucht EFI automatisch den Bootloader des Defaultbetriebssystems und führt ihn aus. Als Defaultbetriebssystem gilt in der Regel das zuletzt installierte Betriebssystem. Wird während des Rechnerstarts eine spezielle Tastenkombination gedrückt, zeigt das EFI ein Menü aller Bootloader an. Dort können Sie das Defaultbetriebssystem einstellen.

In Kombination mit EFI muss GRUB also nur noch Linux starten. Die zweite Funktion von GRUB, nämlich die Auswahl zwischen verschiedenen Betriebssystemen, ist im Zusammenspiel mit EFI eigentlich entbehrlich. Weil sich das GRUB-Menü aber als praktisch erwiesen hat, wird es weiterhin oft angezeigt. Daraus ergeben sich die Startwege, die in [Abbildung 23.1](#) durch gestrichelte Linien angedeutet sind.

### EFI und GRUB

Es gibt für EFI- und für BIOS-Rechner unterschiedliche Varianten von GRUB. Die Installationsprogramme gängiger Distributionen kümmern sich automatisch darum, die richtige Version zu installieren. Das setzt allerdings voraus, dass das Installationsprogramm im EFI-Modus und nicht im BIOS-Modus ausgeführt wird! (Manche Mainboards unterstützen beide Varianten.)

### EFI-Partition

Der GRUB-Code wird bei EFI-Rechnern nicht in den MBR installiert, sondern in ein Verzeichnis der EFI-Partition. Dabei handelt es sich um eine spezielle Partition mit einem VFAT-Dateisystem. Die Partition muss durch eine spezielle UID markiert sein: `OxEF` (MBR) bzw. `C12A7328-F81F-11D2-BA4B-00A0C93EC93B` (GPT).

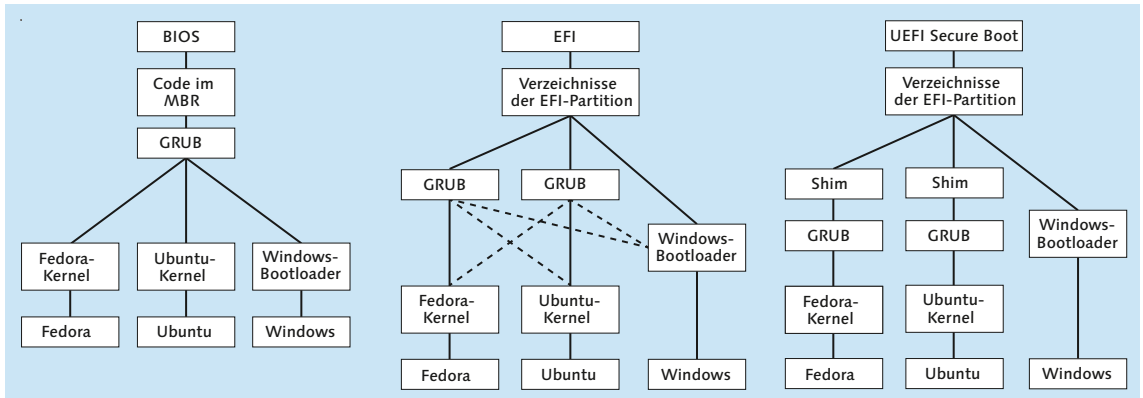


Abbildung 23.1 Boot-Vorgang für drei Betriebssysteme mit BIOS, EFI und UEFI Secure Boot

Wenn Sie Partitionen manuell mit `parted` einrichten, müssen Sie die EFI-Partition mit dem Flag `esp` kennzeichnen (EFI System Partition, `set N esp on`). Außerdem müssen Sie sicherstellen, dass die EFI-Partition im Verzeichnis `/boot/efi` in das Linux-Dateisystem eingebunden wird. (Die Installationsprogramme der meisten Linux-Distributionen kümmern sich selbst um diese Details. Zu den Ausnahmen zählt Arch Linux, wo an dieser Stelle Handarbeit erforderlich ist.)

Microsoft empfiehlt, die EFI-Partition als erste Partition auf der Festplatte einzurichten, obwohl der EFI-Standard dies nicht verlangt. Die Partition muss nicht besonders groß sein, ca. 100 bis 200 MiB reichen.

Zum eigentlichen Start von Linux muss der Bootloader die Linux-Kerneldatei in den Speicher laden und ausführen. Die Kerneldatei hat normalerweise den Dateinamen `/boot/vmlinuz`. Der letzte Buchstabe `z` weist darauf hin, dass der Kernel komprimiert ist. Der Bootloader muss also in der Lage sein, eine vollständige Datei aus einem Linux-Dateisystem zu laden.

Linux-Start

An den Kernel werden meist einige Parameter übergeben, mindestens aber einer: der Device-Name der Systempartition (z. B. `root=/dev/sda3`). Damit weiß der Kernel, welches die Systempartition ist. Sobald der Kernel läuft, gibt er die Kontrolle an das Init-System weiter. Dieses Programm ist für die Initialisierung von Linux zuständig und wird in [Kapitel 24](#), »Das Init-System«, ausführlich beschrieben. Es kümmert sich beispielsweise darum, alle Netzwerkdienste zu starten.

Kernelparameter

Der Linux-Kernel ist modularisiert. Das bedeutet, dass der Kernel an sich nur relativ elementare Funktionen enthält. Zusatzfunktionen zum Zugriff auf bestimmte Hardware-Komponenten, zum Lesen und Schreiben verschiedener Dateisysteme etc. befinden sich dagegen in Modulen, die bei Bedarf aus dem Dateisystem geladen werden und den Kernel so erweitern.

Zugriff auf Kernelmodule

Damit der Startprozess gelingt, muss der Kernel auf die Systempartition zugreifen können. Falls diese Partition in einem Dateisystem vorliegt, das der Kernel nicht direkt unterstützt, oder wenn sich die Partition auf einer SCSI-Festplatte befindet, für die der Kernel keinen Hardware-Treiber enthält, so tritt ein Henne-Ei-Problem auf: Der Kernel kann nicht auf das Dateisystem zugreifen und daher die Module nicht laden, die er benötigen würde, um Dateien des Dateisystems zu lesen ...

**Initrd-Datei** Die Lösung des Problems besteht darin, dass der Bootloader nicht nur den Kernel lädt, sondern auch eine sogenannte Initrd-Datei. Dabei handelt es sich um eine spezielle Datei, die alle für den Startprozess erforderlichen Kernelmodule enthält. Die Datei steht dem Kernel vorübergehend als RAM-Disk zur Verfügung, d. h., der Kernel kann die erforderlichen Module unmittelbar nach dem Start von der RAM-Disk laden. (Initrd ist die Abkürzung für *Initial RAM Disk*.) Die Initrd-Datei hat üblicherweise den Dateinamen `/boot/initrd` oder `/boot/initrd.gz`. Die meisten Distributionen stellen Werkzeuge zur Verfügung, um eine zum Kernel passende `initrd`-Datei zu erzeugen.

**GRUB-Installation und -Konfiguration** Wenn in diesem Buch von »Software-Installation« die Rede ist, dann ist damit üblicherweise die Installation eines Programmpakets auf der Festplatte gemeint. In diesem Kapitel gelten allerdings andere Regeln: Mit der »Installation von GRUB« wird der Prozess bezeichnet, den GRUB-Startcode in die EFI-Partition zu schreiben und einige EFI-Parameter zu verändern. Außerdem werden GRUB-Konfigurations-Scripts im Verzeichnis `/etc/grub.d/` ausgeführt. Diese Scripts erstellen die eigentliche GRUB-2-Konfigurationsdatei `/boot/grub2/grub.cfg`.

### UEFI Secure Boot

Die EFI-Erweiterung *Secure Boot* hat den Boot-Prozess nochmals komplizierter gemacht. Wenn diese EFI-Erweiterung aktiv ist, startet das EFI nur solche Programme, die mit einem dem EFI bekannten Schlüssel signiert sind. Die meisten Mainboards kennen aber nur einen einzigen Schlüssel – den von Microsoft! Glücklicherweise bietet Microsoft für Linux-Distributoren und andere Unternehmen einen Signierdienst an. Damit ist es prinzipiell möglich, einen Bootloader so signieren zu lassen, dass EFI bereit ist, diesen zu starten.

Allerdings sind mit dem Signierdienst Auflagen verbunden: Die Teilnehmer müssen sich dazu verpflichten, das zu startende System gegen Manipulationen abzusichern, sodass Secure Boot seinem Namen gerecht wird und keine Schad-Software lädt, sondern nur den originalen Linux-Kernel der jeweiligen Distribution.

Um den mit dem Microsoft-Schlüssel signierten Code möglichst klein zu halten, haben sich die Distributoren entschlossen, einen Zwischenschritt einzulegen: Das EFI startet zuerst das signierte Programm Shim. Die einzige Aufgabe dieses winzigen Pro-

gramms ist es, die Signaturkette sicherzustellen und GRUB zu starten. GRUB setzt dann wie bisher den Boot-Prozesses fort (siehe [Abbildung 23.1](#) rechts).

Bei aktuellen Versionen von Fedora, SUSE und Ubuntu sind in der Folge GRUB, der Kernel sowie die Kernelmodule jeweils mit distributionseigenen Schlüsseln signiert, sodass sich eine geschlossene Signaturkette vom EFI bis zum letzten Kernelmodul ergibt.

Debian bis einschließlich Version 9 sowie diverse andere Distributionen unterstützen Secure Boot noch gar nicht. Ein Start ist hier nur möglich, wenn im EFI die Secure-Boot-Funktionen deaktiviert werden.

Aus den strengen Secure-Boot-Implementierungen von Fedora, openSUSE und Ubuntu ergeben sich aus Linux-Sicht leider dramatische Einschränkungen:

Der Preis der  
Sicherheit

- ▶ Die Kernelmodule der NVIDIA- und ATI/AMD-Grafiktreiber können nicht geladen werden. Da diese Module direkt von NVIDIA bzw. AMD kompiliert sind, ist es unmöglich, sie mit einem distributionsspezifischen Schlüssel zu signieren. Dieselbe Einschränkung gilt natürlich auch für alle anderen Kernelmodule von proprietären Hardware-Treibern.
- ▶ Es ist nicht ohne Weiteres möglich, einen eigenen Kernel zu kompilieren und einzurichten. Auch dieser Kernel muss ja signiert werden, den dazu erforderlichen Schlüssel hat aber nur der Distributor. Ein Ausweg aus diesem Dilemma ist das von SUSE entwickelte Verfahren *Machine Owner Keys* (MOK). Damit wird in Shim zusätzlich zum SUSE-Schlüssel ein weiterer, von Ihnen selbst zur Verfügung gestellter Schlüssel hinterlegt. Ein selbst kompilierter Kernel und dessen Module können dann ebenfalls mit diesem Schlüssel signiert werden. Das MOK-Konzept haben inzwischen auch andere Distributionen aufgegriffen.
- ▶ GRUB kann nur dann eine andere Linux-Distribution starten, wenn diese mit demselben Schlüssel signiert ist.
- ▶ Der Ruhemodus (*suspend to disk*) funktioniert nicht.
- ▶ Die Kernel-Funktionen `kexec` und `kdump` können nicht genutzt werden.

Am einfachsten umgehen Sie diese Einschränkungen, indem Sie Secure Boot deaktivieren.

### Wogegen schützt Secure Boot?

Secure Boot soll verhindern, dass bereits beim Boot-Prozess Schad-Software geladen wird, die sich in der Folge allen weiteren Sicherheitsmaßnahmen, wie z. B. Viren-Scannern, entzieht. Derartige Angriffe hat es in den letzten Jahrzehnten allerdings nur sehr selten gegeben.

Die unzähligen Sicherheitsprobleme, die Windows und vereinzelt auch Linux in den vergangenen Jahren plagten, waren überwiegend Fehler in einzelnen Anwendungsprogrammen – sei es nun im Internet Explorer oder im Apache Webserver. Diese Probleme wird es weiterhin geben. Secure Boot ändert daran nichts.

**Links** Wenn Sie sich für die technischen Details der Implementierung von Secure Boot unter Linux interessieren, können Sie hier weiterlesen:

<https://mjpg59.dreamwidth.org/12368.html>

<https://lwn.net/Articles/523367>

<https://fedoraproject.org/wiki/Features/SecureBoot>

<https://en.opensuse.org/openSUSE:UEFI>

[https://www.suse.com/releasenotes/x86\\_64/SUSE-SLES/11-SP3](https://www.suse.com/releasenotes/x86_64/SUSE-SLES/11-SP3)

<https://www.suse.com/communities/conversations/uefi-secure-boot-details>

<https://lists.ubuntu.com/archives/ubuntu-devel/2012-June/O35445.html>

<https://askubuntu.com/questions/755238>

## BIOS-Systemstart

Nach dem Einschalten eines älteren Rechners ohne EFI wird das *Basic Input Output System* (BIOS) initialisiert. Während dieses Vorgangs erscheinen meist ein paar Systemmeldungen auf dem Bildschirm. Anschließend lädt das BIOS den Inhalt des ersten Sektors der ersten Festplatte in den Speicher und führt diesen Code aus. Dieser spezielle Sektor der Festplatte heißt *Master Boot Record* (MBR).

### Der Kampf um den Master Boot Record

Es gibt nur einen MBR, aber möglicherweise mehrere Betriebssysteme auf Ihrer Festplatte. Das birgt natürlich Konfliktpotenzial! Sowohl bei Linux- als auch bei Windows-Installationen wird der MBR überschrieben. Während GRUB auch Windows starten kann, nimmt Windows leider keine Rücksicht auf Linux. Deswegen müssen Sie nach einer Windows-Installation GRUB reparieren, wofür Sie am besten ein Live- oder Notfallsystem verwenden. Besser ist es, zuerst Windows und dann Linux zu installieren! Sollte bei einer späteren Windows-Installation der MBR mit GRUB überschrieben werden, finden Sie in den weiteren Abschnitten Notfalltipps.

**Windows-Bootloader** Wenn auf einem Rechner Windows installiert ist, befindet sich im MBR ein winziges Programm. Es sucht die als »aktiv« gekennzeichnete Partition und führt dann den Windows-Bootloader aus, der sich im Boot-Sektor dieser Partition befindet. Falls auf dem Rechner mehrere Windows-Versionen installiert sind, können Sie im Windows-Bootloader zwischen diesen Versionen wählen.



Wenn auf dem Rechner auch Linux installiert ist, wird der MBR üblicherweise durch den Code des Linux-Bootloaders GRUB ersetzt. GRUB kann dann wahlweise Linux starten oder in den Windows-Bootloader verzweigen (siehe [Abbildung 23.1](#)).

Linux-Bootloader

Eine alternative Vorgehensweise besteht darin, den MBR nicht anzurühren, GRUB in den Boot-Sektor der Linux-Systempartition zu installieren und diese Partition als »aktiv« zu markieren. Diese Vorgehensweise würde zwar den MBR-Konventionen entsprechen, ist aber weniger robust und deswegen nicht mehr gebräuchlich. Sie ist zudem inkompatibel mit manchen Dateisystemen, z. B. mit XFS.

Der MBR ist nur 512 Byte groß – zu klein, um das gesamte Bootloader-Programm zu speichern. Deswegen enthält der MBR gerade so viel Code, um den Rest des Bootloaders von der Festplatte zu laden. Dementsprechend ist der GRUB-Code in zwei oder drei Teile zerlegt: `stage1` befindet sich im MBR und hat die Aufgabe, die ersten Sektoren von `stage1_5` oder `stage2` zu laden. `stage1_5` enthält Zusatzcode für den Zugriff auf Dateien in verschiedenen Dateisystemen. `stage2` enthält schließlich den eigentlichen Bootloader.

Sobald der Bootloader läuft, erscheint ein Menü mit einer Auswahl aller Betriebssysteme, die bei der GRUB-Konfiguration definiert wurden. Mit den Cursortasten können Sie nun das gewünschte Betriebssystem auswählen und dann mit  starten. Oft ist GRUB so eingestellt, dass nach einer gewissen Zeit ein Betriebssystem automatisch gestartet wird.

Wenn GRUB einmal läuft, verläuft der eigentliche Linux-Start exakt wie auf einem EFI-Rechner: GRUB lädt den Kernel und startet diesen, wobei es die `initrd`-Datei und Kernelparameter übergibt.

Kernel,  
Initrd-Datei

Auf einem BIOS-Rechner wird bei der Installation von GRUB der GRUB-Startcode in den Boot-Sektor einer Festplatte oder SSD geschrieben. Die weitere Konfiguration erfolgt wie bei einem EFI-Rechner durch die Datei `/boot/grub2/grub.cfg`.

GRUB-  
Installation und  
-Konfiguration

## Initrd-Dateien

Linux verwendet einen modularisierten Kernel. Viele Zusatzfunktionen – z. B. für die Ansteuerung einer SCSI-Karte, für den Zugriff auf bestimmte Dateisysteme, RAID-Verbunde oder LVM-Partitionen – befinden sich nicht direkt im Kernel, sondern in Modulen. Beim Systemstart ist das aber problematisch: Wie soll der Kernel ein Modul laden, wenn er noch gar nicht in der Lage ist, auf das Dateisystem zuzugreifen? Deswegen werden die für den unmittelbaren Start erforderlichen Module in eine Initial RAM Disk verpackt. Die entsprechende `initrd`-Datei übergibt GRUB an den Kernel (Schlüsselwort `initrd` in der GRUB-Konfigurationsdatei).

Henne oder Ei?  
Kernelmodule  
beim  
Boot-Prozess

**Initrd-Interna** Der Kernel und die Initrd-Datei werden im Verzeichnis `/boot` gespeichert. Die Initrd-Datei muss Kernelmodule enthalten, deren Version exakt mit der Version des Kernels übereinstimmt. Aus diesem Grund muss jedes Mal, wenn ein neuer Kernel installiert oder selbst kompiliert wird, auch eine dazu passende Initrd-Datei erstellt werden. Bei einem Kernel-Update erledigt das das Update-Programm. Wenn Sie den neuen Kernel dagegen selbst installieren, müssen Sie sich auch um die Initrd-Datei selbst kümmern.

Die Bezeichnung »Initrd-Datei« ist bei den meisten aktuellen Distributionen eigentlich falsch: Es handelt sich in Wirklichkeit um `initramfs`-Dateien, deren Aufbau etwas weiter unten beschrieben wird. Weil aber sowohl die GRUB-Optionen als auch die Kommandos zum Erzeugen der Dateien den Begriff `initrd` nutzen und der Kernel die Datei trotz der falschen Bezeichnung korrekt interpretiert, bleibe ich in diesem Buch ebenfalls bei dieser Bezeichnung – gewissermaßen wider besseres Wissen.

Die Initrd-Datei ist nicht immer zwingend erforderlich: Wenn Ihr Kernel alle Komponenten enthält, die während des Boot-Prozesses erforderlich sind, gelingt der Start auch ohne Initrd-Datei. Dazu muss der Kernel aber entsprechend kompiliert sein – und genau das ist bei den meisten Distributionen nicht der Fall.

Bedauerlicherweise ist die Erzeugung von Initrd-Dateien nicht standardisiert. Jede Distribution verwendet ihre eigenen Werkzeuge. Die Initrd-Dateien enthalten nicht nur Kernelmodule, sondern auch Scripts zur Hardware-Initialisierung und unter Umständen ein minimales Rescue-System, sodass Rettungsarbeiten selbst dann durchgeführt werden können, wenn das Einbinden der Systempartition nicht gelingt.

**update-initramfs (Debian, Ubuntu)** Unter Debian und Ubuntu ist zur Erzeugung und Administration der Initrd-Dateien das Script `update-initramfs` vorgesehen. Im einfachsten Fall geben Sie einfach nur die Option `-u` an, um die Initrd-Datei der aktuellsten installierten Kernelversion zu aktualisieren. Wenn Sie die Initrd-Datei für eine andere Kernelversion aktualisieren möchten, geben Sie die Versionsnummer mit `-k` an. `-k all` aktualisiert die Initrd-Dateien für alle installierten Kernelversionen.

Mit den Optionen `-c` bzw. `-d` erzeugt `update-initramfs` eine neue Initrd-Datei bzw. löscht eine vorhandene Initrd-Datei. In diesem Fall ist die Angabe der Kernelversion durch `-k` zwingend erforderlich.

```
root# update-initramfs -c -k 4.11.0-7-generic
update-initramfs: Generating /boot/initrd.img-4.11.0-7-generic
```

Hinter den Kulissen greift `update-initramfs` auf das Script `mkinitramfs` zurück, um Initrd-Dateien zu erzeugen. Die Basiskonfiguration erfolgt in `/etc/initramfs-tools/initramfs.conf` sowie durch die Dateien in `/etc/initramfs-tools/conf.d`. Darüber hinaus werden der Initrd-Datei alle in `/etc/initramfs-tools/modules` aufgezählten Module hinzugefügt (ein Modul pro Zeile).

`mkinitramfs` erzeugt in der Standardkonfiguration mit `MODULES=most` in `initramfs.conf` ziemlich große Initrd-Dateien mit unzähligen Kernelmodulen. Sollten Sie `mkinitramfs` direkt aufrufen, müssen Sie zumindest den Namen der neuen Initrd-Datei übergeben (Option `-o`). Wenn die Initrd-Datei nicht für die aktuelle Kernelversion erzeugt werden soll, geben Sie zusätzlich die gewünschte Version an:

```
root# mkinitramfs -o myinitrd 4.11.0-7-generic
```

Fedora, RHEL/CentOS sowie SUSE/openSUSE verwenden `dracut` zur Erzeugung der Initrd-Datei. `dracut` wird bei jedem Kernel-Update automatisch ausgeführt. Das Kommando `dracut` berücksichtigt die Einstellungen aus `/etc/dracut.conf`. Um für einen selbst kompilierten Kernel 4.11.1 in der Datei `/boot/vmlinuz-4.11.1` manuell eine Initrd-Datei zu erzeugen, führen Sie das folgende Kommando aus:

`dracut`  
(Fedora, Red Hat,  
SUSE)

```
root# dracut /boot/initramfs-4.11.1 4.11.1
```

`Dracut` erzeugt kompakte Initrd-Dateien, die nur solche Kernelmodule enthalten, die im laufenden Betrieb aktiv sind. Das kann nach Hardware-Erweiterungen zu Boot-Problemen führen. Aus diesem Grund gibt es für das Rescue-System eine spezielle Initrd-Datei mit viel mehr Modulen. Falls der gewöhnliche Boot-Prozess nach einem Hardware-Umbau scheitern sollte, booten Sie das Rettungssystem und führen dann das folgende Kommando aus:

```
root# dracut --regenerate-all --force
```

Wenn eine Distribution ein Kernel-Update durchführt und sich dadurch der Name der Kerneldatei ändert, muss auch die GRUB-Menüdatei entsprechend geändert und eine zum neuen Kernel passende Initrd-Datei erzeugt werden. Alle gängigen Distributionen erledigen diese Aufgaben im Rahmen der Update-Verwaltung automatisch, sodass beim nächsten Neustart des Rechners automatisch der neue Kernel verwendet wird. Bei vielen Distributionen gibt es für den alten Kernel weiterhin einen GRUB-Menüeintrag, damit bei Update-Problemen eine Möglichkeit besteht, das System mit dem alten Kernel weiterzunutzen.

Kernel-Updates

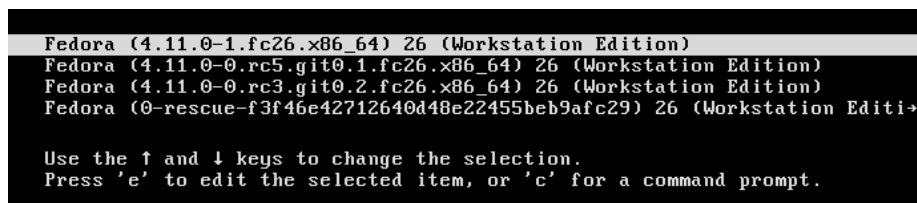
Initrd-Dateien werden intern als `initramfs`-Dateien dargestellt. Die Initrd-Datei ist eine komprimierte Archivdatei (`cpio`-Datei), die aus diversen Verzeichnissen und Dateien zusammengesetzt ist. Wenn Sie sich den Inhalt des Archivs ansehen möchten, gehen Sie so vor:

Initrd-Datei  
ansetzen

```
root# cd /boot
root# cp initrd-n.n initrd-test.gz
root# gunzip initrd-test
root# mkdir test
root# cd test
root# cpio -i < ../initrd-test
root# ls -lR
```

## 23.2 GRUB-Bedienung (Anwendersicht)

Nach einer Linux-Installation erscheint beim Neustarten des Rechners ein Menü zur Auswahl des gewünschten Betriebssystems (siehe [Abbildung 23.2](#)). Das Aussehen von GRUB kann je nach Konfiguration stark variieren. Manche Distributionen starten GRUB im Grafikmodus und nehmen dem Programm damit seinen spartanischen Anstrich. Um diverse Zusatzfunktionen von GRUB nutzen zu können, müssen Sie den Grafikmodus mit `[Esc]` verlassen. Andere Distributionen vermeiden die Anzeige des GRUB-Menüs nach Möglichkeit überhaupt. Wenn beispielsweise Ubuntu als einziges Betriebssystem auf einem Rechner installiert ist, bekommen Sie das GRUB-Menü nur zu sehen, wenn Sie während des Rechnerstarts eine Taste drücken.



```
Fedora (4.11.0-1.fc26.x86_64) 26 (Workstation Edition)
Fedora (4.11.0-0.rc5.git0.1.fc26.x86_64) 26 (Workstation Edition)
Fedora (4.11.0-0.rc3.git0.2.fc26.x86_64) 26 (Workstation Edition)
Fedora (0-rescue-f3f46e42712640d48e22455beb9afc29) 26 (Workstation Editi→

Use the ↑ and ↓ keys to change the selection.
Press 'e' to edit the selected item, or 'c' for a command prompt.
```


Abbildung 23.2 Ein minimalistisches GRUB-Menü

- Passwort** GRUB kann durch ein Passwort abgesichert sein. In diesem Fall können Sie die interaktiven Funktionen von GRUB erst verwenden, nachdem Sie `[P]` gedrückt und dann das Passwort angegeben haben.
- Tastatur** Unter GRUB gilt normalerweise das US-Tastaturlayout. Falls Sie mit einer deutschen Tastatur arbeiten, sind unter anderem `[Y]` und `[Z]` vertauscht. Eine Tabelle zur Eingabe wichtiger Sonderzeichen finden Sie in [Abschnitt 2.11](#), »Probleme beheben«.
- Kernel-Boot-Optionen übergeben** Sofern das GRUB-Menü nicht durch ein Passwort abgesichert ist, können Sie den gerade mit den Cursortasten ausgewählten Eintrag des GRUB-Menüs mit `[E]` (*edit*) verändern. Es werden nun einige Zeilen angezeigt, die in einer recht eigenwilligen Syntax beschreiben, wie das Betriebssystem gestartet werden soll.

Die Editierfunktionen von GRUB werden vor allem dazu verwendet, vor dem Start von Linux zusätzliche Kerneloptionen anzugeben, z. B. zur Umgehung von Hardware-Problemen. Dazu suchen Sie nach einer Zeile, die so ähnlich wie das folgende Muster aussieht:

```
linux /boot/vmlinuz-n.n.n root=/dev/sdb13
```


Am Ende dieser Zeile können Sie Parameter hinzufügen bzw. verändern. `[Strg]+[X]` oder `[F10]` startet dann Linux mit den veränderten Parametern. Die Änderungen werden aber nicht gespeichert!

Vom GRUB-Menü gelangen Sie mit  in einen interaktiven Kommandomodus. Dort können Sie diverse GRUB-Kommandos manuell ausführen. Das bietet die Möglichkeit, ein Linux-Betriebssystem auch dann zu starten, wenn der dazugehörige GRUB-Menüeintrag fehlt oder fehlerhaft ist. Sie müssen dazu nur wissen, auf welcher Partition sich Ihr Linux befindet und wie die entsprechenden GRUB-Kommandos lauten. Details dazu folgen im weiteren Verlauf des Kapitels. Beachten Sie, dass sich die Syntax von GRUB 0.97 und GRUB 2 deutlich unterscheidet. Dieses Kapitel berücksichtigt nur GRUB 2.

Interaktiv  
Kommandos  
ausführen

Mit den folgenden Kommandos wird beispielsweise eine Linux-Distribution gestartet, die sich in der Partition `/dev/sdb13` mit einem `ext4`-Dateisystem befindet. (`insmod ext2` lädt einen Treiber, der gleichermaßen `ext2`, `ext3` und `ext4` unterstützt.)

```
grub> insmod ext2
grub> set root='(hd1,13)'
grub> linux /boot/vmlinuz-n.n.n root=/dev/sdb13 ro
grub> initrd /boot/initrd-n.n.n
grub> boot
```

Bei der Eingabe der Dateinamen vervollständigt GRUB mit  die Dateinamen für das durch `root` bzw. durch eine Laufwerksangabe ausgewählte Dateisystem. Mit `cat` können Sie einzelne Textdateien sogar anzeigen. Daneben bietet der GRUB-Kommandomodus noch viele andere Möglichkeiten, auf die hier aber aus Platzgründen nicht eingegangen wird. `help` liefert die Liste aller Kommandos, `help kommandoname` gibt genauere Informationen zu diesem Kommando.

GRUB liest das Boot-Menü aus der Datei `grub.cfg`, die je nach Distribution in `/boot/grub`, `/boot/grub2` oder `/boot/efi/EFI/distribname` gespeichert wird. Die Datei enthält Kommandos, die die Einträge des GRUB-Menüs beschreiben. Wenn Sie also das GRUB-Menü bleibend verändern möchten, müssen Sie Linux starten und die GRUB-Menüdatei verändern. Bei GRUB 2 verändern Sie dazu zuerst andere Konfigurationsdateien und generieren daraus eine neue Fassung von `grub.cfg`. GRUB berücksichtigt Ihre Änderungen automatisch ab dem nächsten Start. Detaillierte Informationen zum Aufbau der GRUB-Konfigurationsdateien folgen im nächsten Abschnitt.

Menü bleibend  
verändern

## 23.3 GRUB-Konfiguration

Das GRUB-Menü wird durch die Datei `grub.cfg` definiert, die sich je nach Distribution an unterschiedlichen Orten befindet:

`grub.cfg`

|  |  |
|--|--|
| <code>/boot/grub/grub.cfg</code>         | (Debian und Ubuntu auf BIOS- und EFI-Rechnern)     |
| <code>/boot/grub2/grub.cfg</code>        | (Fedora und openSUSE auf BIOS-Rechnern)            |
| <code>/boot/efi/EFI/name/grub.cfg</code> | (die meisten Distributionen auf EFI-Rechnern)      |
| <code>/etc/grub2.cfg</code>              | (Link auf <code>grub.cfg</code> in Fedora)         |
| <code>/etc/grub2-efi.cfg</code>          | (Link auf <code>grub.cfg</code> in Fedora mit EFI) |

Bei einigen Distributionen ist `/boot/efi/EFI/name/grub.cfg` eine kurze Datei, die Informationen darüber enthält, wo sich die tatsächliche Konfigurationsdatei befindet.

`/etc/grub.d/` Manuelle Veränderungen an `grub.cfg` sind *nicht* vorgesehen! Die Zugriffsrechte dieser Datei sind deswegen auf *read-only* gestellt. Wenn Sie das GRUB-Menü modifizieren möchten, verändern Sie die zugrunde liegenden Konfigurationsdateien. Die Orte dieser Dateien sind erfreulicherweise bei Debian, Fedora, openSUSE und Ubuntu identisch:

```
/etc/grub.d/*           (allgemeine GRUB-Konfigurationsdateien)
/etc/default/grub      (distributionsspezifische Ergänzungen)
```

**update-grub** Um nach Änderungen an diesen Dateien oder nach einem Kernel-Update die GRUB-Menüdatei `grub.cfg` neu zu generieren, müssen Sie eines der folgenden Kommandos ausführen:

```
root# update-grub                (Debian und Ubuntu)
root# grub2-mkconfig -o /etc/grub2.cfg      (Fedora mit BIOS)
root# grub2-mkconfig -o /boot/grub2/grub.cfg (Fedora mit EFI)
root# grub2-mkconfig -o /boot/grub2/grub.cfg (openSUSE mit BIOS)
root# grub2-mkconfig -o /boot/grub2-efi/grub.cfg (openSUSE mit EFI)
```

**Beispiel für `grub.cfg`** Die resultierende Datei `grub.cfg` sieht so ähnlich wie das folgende Beispiel aus. Das folgende Listing einer automatisch erzeugten Fassung von `grub.cfg` unter Ubuntu ist aus Platzgründen stark gekürzt. Lassen Sie sich übrigens von der Zeile `insmod ext2` nicht irritieren: Dieses GRUB-Modul ist für alle `ext`-Dateisysteme zuständig, also auch für `ext3` und `ext4`. Weitere Erläuterungen zur Syntax in `grub.cfg` folgen im weiteren Verlauf dieses Abschnitts.

```
# Beispiel für /boot/grub/grub.cfg

# aus /etc/grub.d/00_header
# Code zur Variablenverwaltung ...
# Definition diverser Funktionen: savedefault, recordfail, load_video
# Font-Dateien suchen ...

# aus /etc/grub.d/05_debian_theme
# Farben für den Textmodus einstellen ...

# aus /etc/grub.d/10_linux
# Definition der Funktion gfxmode ...

# Ubuntu starten
menuentry 'Ubuntu' ... {
    recordfail
    load_video
```

```

gfxmode $linux_gfx_mode
insmod gzio
insmod part_msdos
insmod ext2
set root='hd0,msdos1'
search ... --set root 508de33b-b057-4e5f-b936-db8eb1f5ae51
linux /boot/vmlinuz-4.11.0-10-generic \
      root=UUID=508de33b-b057-4e5f-b936-db8eb1f5ae51 \
      ro quiet splash $vt_handoff
initrd /boot/initrd.img-4.11.0-10-generic
}

# Menü mit alten Ubuntu-Versionen und dem Rettungssystem
submenu 'Erweiterte Optionen für Ubuntu' ... {
  menuentry 'Ubuntu, mit Linux 4.11.0-10-generic' ... { ... }
  menuentry 'Ubuntu, with Linux 4.11.0-10-generic (recovery mode)' ... {
    ...
    linux /boot/vmlinuz-4.11.0-10-generic \
          root=UUID=508de33b-b057-4e5f-b936-db8eb1f5ae51 \
          ro recovery nomodeset
    ...
  }
}

# aus /etc/grub.d/20_memtest86+: Speichertest durchführen
menuentry 'Memory test (memtest86+)' {
  insmod part_msdos
  insmod ext2
  set root='hd0,msdos1'
  search --no-floppy --fs-uuid --set=root 0f65...
  knetbsd /boot/memtest86+.elf
}

# aus /etc/grub.d/30_os-prober: in den Windows-Bootloader verzweigen
menuentry 'Windows Boot Manager (auf /dev/sda1)' ... {
  insmod part_gpt
  insmod fat
  set root='hd0,gpt1'
  search ... --set=root 97C2-AD9D
  chainloader /EFI/Microsoft/Boot/bootmgfw.efi
}

```

Unter Debian und Ubuntu erzeugt das Kommando `update-grub` eine neue Version von `grub.cfg`. Unter Fedora und openSUSE führen Sie stattdessen `grub2-mkconfig -o /boot/grub2/grub.cfg` aus. Das Debian/Ubuntu-spezifische Script `update-grub` enthält ebenfalls nur dieses Kommando.

**grub.cfg neu erzeugen**

grub2-mkconfig wertet die im Folgenden beschriebenen Konfigurationsdateien bzw. -Scripts aus. Dabei werden unter anderem GRUB-Menüeinträge für sämtliche Kerneldateien in /boot erzeugt. Außerdem werden alle erreichbaren Partitionen untersucht. Wenn sie andere Betriebssysteme enthalten, werden auch hierfür GRUB-Menüeinträge erzeugt. Aus diesem Grund dauert die Ausführung von update-grub auf Rechnern mit vielen Partitionen eine Weile.

grub2-mkconfig wird automatisch bei jedem Kernel-Update ausgeführt und stellt sicher, dass der neueste Linux-Kernel im Grub-Menü enthalten ist.

**/etc/default/grub** Die Datei /etc/default/grub enthält einige globale GRUB-Einstellungen. Vergessen Sie nicht, dass hier durchgeführte Änderungen erst wirksam werden, wenn Sie grub.cfg neu generieren! In Ubuntu enthält die Konfigurationsdatei die folgenden Einstellungen:

```
# Datei /etc/default/grub
GRUB_DEFAULT=0
GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""
```

Die Standardkonfiguration von Fedora sieht so aus:


```
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="rd.lvm.lv=fedora/root rd.lvm.lv=fedora/swap rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
```

Im Folgenden erkläre ich Ihnen die Bedeutung einiger Parameter, auch solcher, die standardmäßig nicht aktiv sind:

- **GRUB\_DEFAULT** gibt an, welcher GRUB-Menüeintrag standardmäßig ausgewählt werden soll. Die Einstellung "saved" bedeutet, dass der zuletzt ausgewählte Menüeintrag aktiviert wird. Das funktioniert allerdings nur, wenn sich die GRUB-Dateien in einer gewöhnlichen Partition befinden! Ist dagegen LVM oder RAID im Spiel, kann GRUB nach der Menüauswahl keine Umgebungsvariablen speichern.

Eine weitere Möglichkeit besteht darin, GRUB\_DEFAULT die menuentry-Zeichenkette des gewünschten Menüeintrags zuzuweisen. Dabei müssen Sie aber darauf achten, die Schreibweise exakt einzuhalten.



- ▶ Die Variable `GRUB_HIDDEN_TIMEOUT` ist dann von Bedeutung, wenn GRUB während der Installation nur eine einzige Linux-Distribution auf Ihrem Rechner erkennt. In diesem Fall gibt `GRUB_HIDDEN_TIMEOUT` an, wie lange der Benutzer Zeit hat, um mit  das GRUB-Menü anzuzeigen. Während dieser Wartezeit bleibt der Bildschirm schwarz.  
 Unter Ubuntu bewirkt `GRUB_HIDDEN_TIMEOUT=0`, dass GRUB das Betriebssystem sofort startet. Um in den Boot-Vorgang einzugreifen, müssen Sie unmittelbar nach dem Rechnerstart  drücken. Wenn mehrere Betriebssysteme installiert sind, ignoriert GRUB die `GRUB_HIDDEN_TIMEOUT`-Einstellung und zeigt das Menü an.
- ▶ `GRUB_HIDDEN_TIMEOUT_QUIET=true` verhindert, dass während der `GRUB_HIDDEN_TIMEOUT`-Wartezeit ein Countdown-Zähler angezeigt wird. Wenn Sie möchten, dass das GRUB-Menü immer angezeigt wird (auch dann, wenn nur Ubuntu auf dem Rechner installiert ist), stellen Sie den Zeilen `GRUB_HIDDEN_TIMEOUT=...` und `GRUB_HIDDEN_TIMEOUT_QUIET=...` jeweils das Kommentarzeichen `#` voran.
- ▶ `GRUB_TIMEOUT=n` gibt an, wie viele Sekunden GRUB auf die Auswahl eines Menüeintrags wartet. Wenn diese Zeit ohne Benutzereingaben verstreicht, startet GRUB das ausgewählte Betriebssystem. Die hier eingestellte Zeit kommt nur zur Geltung, wenn das GRUB-Menü überhaupt erscheint.
- ▶ Die Variable `GRUB_DISTRIBUTOR` wird von dem Script `10_linux` ausgewertet, das ich weiter unten beschreibe. Sie gibt den Namen der aktuellen Distribution an, also z. B. Ubuntu oder Fedora.
- ▶ Auch `GRUB_CMDLINE_LINUX` und `GRUB_CMDLINE_LINUX_DEFAULT` werden von `10_linux` berücksichtigt und geben an, welche Optionen an den Kernel übergeben werden sollen. Die `GRUB_CMDLINE_LINUX`-Optionen gelten für jeden Start; die `GRUB_CMDLINE_LINUX_DEFAULT`-Optionen werden zusätzlich für den Standardstart hinzugefügt, aber nicht für den Recovery Mode.
- ▶ Standardmäßig wird das GRUB-Menü im Grafikmodus in einer Auflösung von  $640 \times 480$  Pixel angezeigt. Wenn Sie eine höhere Auflösung wünschen, können Sie diese mit `GRUB_GFXMODE` einstellen. Wenn Sie auf den Grafikmodus ganz verzichten möchten, aktivieren Sie die dafür vorgesehene Einstellung `GRUB_TERMINAL=console`. Beide Variablen werden vom Script `00_header` ausgewertet. In der Standardeinstellung gibt es keinen optisch sichtbaren Unterschied zwischen dem Text- und dem Grafikmodus, allerdings können nur im Grafikmodus Unicode-Zeichen angezeigt werden.
- ▶ GRUB übergibt normalerweise das Root-Verzeichnis als UUID-Nummer an den zu startenden Linux-Kernel. Wenn Sie stattdessen die Angabe des Device-Namens (z. B. `/dev/sda1`) vorziehen, aktivieren Sie die Zeile `GRUB_DISABLE_LINUX_UUID=true`. Diese Einstellung gilt nur für den Start der aktiven Distribution (Script `10_linux`), nicht für andere Distributionen.

- ▶ `update-grub` bzw. `grub2-mkconfig` erzeugt normalerweise auch Menüeinträge zum Start von Linux im Recovery Mode. Dabei wird Linux im Single-User-Modus und ohne Anzeige eines Splash-Bildschirms gestartet. Mit `GRUB_DISABLE_RECOVERY="true"` werden keine Recovery-Einträge erzeugt.
- ▶ Mit `GRUB_INIT_TUNE=...` können Sie beim Start von GRUB einen Ton ausgeben lassen.

#### Automatische Erzeugung von `grub.cfg`

Die Grundidee der GRUB-Konfiguration besteht darin, dass Sie selbst nur die Eckdaten der GRUB-Konfiguration vorgeben. Die finale Konfigurationsdatei `grub.cfg` wird unter Berücksichtigung dieser Vorgaben durch ein Script automatisch erzeugt. Der springende Punkt dabei ist, dass die GRUB-Scripts versuchen, *alle* auf dem Rechner installierten Betriebssysteme zu erkennen und entsprechende Menüeinträge in die `grub.cfg` einzubauen. Das gilt auch für alle Betriebssysteme, die zu einem späteren Zeitpunkt hinzugekommen sind.

Das Verzeichnis `/etc/grub.d` enthält dazu mehrere ausführbare Script-Dateien (siehe [Tabelle 23.1](#)). Wenn eine neue Version von `grub.cfg` erzeugt werden soll, führt `update-grub` alle in `grub.d` enthaltenen Scripts der Reihe nach aus. Das Ergebnis, also die Standardausgabe der Scripts, landet in `grub.cfg`. Die script-basierte Vorgehensweise macht die Konfigurationsdateien leider recht unübersichtlich: In den eigentlichen Code sind immer wieder Anweisungen der Form `cat << EOF` eingebettet, die alle folgenden Zeilen bis zum Kürzel EOF an die Standardausgabe leiten. Diese Zeilen enthalten selbst oft Script-Code, der erst von GRUB beim Systemstart ausgewertet wird.

| Datei                        | Funktion  |
|------------------------------|---|
| <code>00_header</code>       | GRUB-Grundeinstellungen   |
| <code>05_debian_theme</code> | farbliche Gestaltung des Menüs (nur unter Debian und Ubuntu)            |
| <code>10_linux</code>        | Menüeinträge zum Start der aktuellen Distribution                       |
| <code>20_linux_xen</code>    | Menüeinträge zum Start virtueller Maschinen                             |
| <code>20_memtest86+</code>   | Menüeintrag zum Start von Memtest86 (nur unter Debian/Ubuntu)           |
| <code>30_os-prober</code>    | Menüeinträge zum Start aller anderen Betriebssysteme                    |
| <code>40_custom</code>       | Muster für eigene Konfigurationsdateien                                 |
| <code>41_custom</code>       | baut in <code>grub.cfg</code> den Text aus <code>custom.cfg</code> ein. |

**Tabelle 23.1** Dateien im Verzeichnis `/etc/grub.d/`

Die beiden interessantesten Scripts sind `10_linux` und `30_os-prober`. `10_linux` liefert für jede Kernelversion im `/boot/`-Verzeichnis zwei GRUB-Menüeinträge: einen zum gewöhnlichen Start und einen zweiten für einen Recovery-Start im Single-User-

Modus und ohne Splash-Bildschirm. Die Menüeinträge sind nach Versionsnummern sortiert, die aktuellste Version steht am Anfang der Liste.

`30_os-prober` ruft das Script `os-prober` auf. Es liefert eine Liste aller Betriebssysteme auf allen zugänglichen Festplattenpartitionen. Für jedes dieser Betriebssysteme werden nun Menüeinträge erzeugt, wobei bei Linux-Distributionen auf die eventuell vorhandene GRUB-Konfiguration zurückgegriffen wird. Dabei kommt es zum Aufruf unzähliger Scripts, die alle Teil des Pakets `os-prober` sind.

Wenn Sie `grub.cfg` um eigene Einträge erweitern möchten, fügen Sie dem Verzeichnis `grub.d` eigene Scripts hinzu. Die Scripts werden in der durch die Startnummer vorgegebenen Reihenfolge ausgeführt, bei gleichlautender Startnummer in alphabetischer Abfolge. Denken Sie daran, das Execute-Bit zu setzen. Beachten Sie auch, dass die Datei nicht einfach unverändert in `grub.cfg` eingebaut wird, sondern dass nur das Ergebnis (die Standardausgabe) in `grub.cfg` einfließt!

`grub.cfg`  
individuell  
erweitern

Die Musterdatei `40_custom` zeigt eine mögliche Vorgehensweise: Dabei wird das `tail`-Kommando auf die Datei angewandt (Parameter `$0`). Die Option `-n +3` bewirkt, dass `tail` die Datei ab der dritten Zeile ausgibt, die ersten zwei Zeilen also überspringt:

```
#!/bin/sh
exec tail -n +3 $0
# This file is an example on how to add custom entries
...
```

Eine andere Variante zeigt `41_custom` auf. Der dort enthaltene Code wird unverändert in `grub.cfg` übertragen und erst zur Laufzeit (also bei der Anzeige des GRUB-Menüs) ausgewertet. Zu diesem Zeitpunkt wird getestet, ob die Datei `/boot/grub/custom.cfg` (Ubuntu) bzw. `/boot/grub2/custom.cfg` (Fedora, openSUSE) existiert. Ist das der Fall, wird sie der GRUB-Konfiguration hinzugefügt. Da `source` zur GRUB-Laufzeit ausgeführt wird, werden Änderungen in `custom.cfg` wirksam, ohne dass die GRUB-Konfigurationsdatei `grub.cfg` neu erzeugt werden muss.

```
#!/bin/sh
cat <<EOF
if [ -f \${prefix}/custom.cfg ]; then
    source \${prefix}/custom.cfg;
fi
EOF
```

## Syntax und Interna

Eine vollständige Syntaxbeschreibung aller in `grub.cfg` erlaubten Schlüsselwörter ist hier aus Platzgründen unmöglich. Ich beschränke mich daher im Folgenden auf die wichtigsten Schlüsselwörter, die in den Beispielen dieses Abschnitts vorkommen. Das

offizielle GRUB-Handbuch (die PDF-Fassung umfasst mehr als 100 Seiten) finden Sie hier:

<http://www.gnu.org/software/grub/manual>

**Variablen** Mit `set varname=wert` führen Sie Variablenzuweisungen durch. Zum Auslesen von Variablen verwenden Sie die Schreibweise `$varname`. Wenn Sie interaktiv GRUB-Kommandos ausführen, zeigt `echo $varname` den Inhalt einer Variablen an, und `set` gibt alle definierten Variablen zurück.

Einige Variablen haben außerdem eine besondere Bedeutung. Dazu zählen z.B. `default`, `timeout`, `color_XXX`, `menu_color_XXX` und insbesondere `root`: Bei sämtlichen Zugriffen auf Dateien wird automatisch die durch `root` definierte Partition gelesen.

GRUB kann Variablen zur Laufzeit bleibend speichern. Dazu muss zuerst im Linux-Dateisystem die Datei `/boot/grub[2]/grub-editenv` eingerichtet werden, was bei den meisten Distributionen standardmäßig der Fall ist:

```
root# grub-editenv /boot/grub[2]/grubenv create
```

GRUB kann nun zur Laufzeit mit `save_env varname` eine Variable in dieser Datei speichern bzw. mit `load_env` alle Variablen aus dieser Datei lesen. Vorher muss die GRUB-Variablen `root` so eingestellt werden, dass sie auf die Partition mit der Environment-Datei verweist.

**Partitionen** GRUB kennt eine eigene Nomenklatur zur Bezeichnung von Festplatten und den darauf enthaltenen Partitionen (siehe [Tabelle 23.2](#)). Inkonsequent ist dabei die Nummerierung: Die erste Festplatte hat die Nummer 0, die erste Partition hingegen die Nummer 1!

| GRUB-Device-Name | Bedeutung   |
|------------------|---|
| (hd0)            | die erste Festplatte/SSD (entspricht <code>/dev/sda</code> )              |
| (hd1)            | die zweite Festplatte/SSD (entspricht <code>/dev/sdb</code> )             |
| (hd0,1)          | die erste Partition der ersten Festplatte/SSD ( <code>/dev/sda1</code> )  |
| (hd2,8)          | die achte Partition der dritten Festplatte/SSD ( <code>/dev/sdc8</code> ) |

**Tabelle 23.2** GRUB-2-Partitionsnamen

Es ist zulässig, der Partitionsnummer ein Kürzel voranzustellen, das das Partitionierungsverfahren angibt: `msdos` für Datenträger mit der Partitionstabelle im MBR, `gpt` für Datenträger mit einer GUID Partition Table. Daraus ergeben sich dann Partitionsnamen wie `(hd0,msdos3)` oder `(hd0,gpt2)`.

In `grub.cfg` kommt mehrfach die folgende Kommandosequenz vor:

Umgang mit  
UUIDs (search)

```
set root=(hd1,1)
search --no-floppy --fs-uuid --set root 724f...
```

Das erste Kommando initialisiert die Variable `root`. Die zweite Anweisung sucht nach einem Dateisystem mit der angegebenen UUID, also z. B. `724f`. Falls die Suche erfolgreich ist, speichert GRUB aufgrund der Option `--set` den entsprechenden Partitionsnamen in der Variablen `root`. Diese Doppelgleisigkeit ist eine Vorsichtsmaßnahme. Sie stellt sicher, dass GRUB die Partition auch dann findet, wenn das Dateisystem inzwischen neu formatiert wurde (andere UUID) oder der Datenträger aufgrund einer anderen Verkabelung eine andere Device-Nummer hat.

Mit `insmod name` lädt GRUB zur Laufzeit Erweiterungsmodule mit Zusatzfunktionen. GRUB sucht nach den Moduldateien `name.mod` im Verzeichnis `/boot/grub[2]` in der durch die Variable `root` eingestellten Partition. Wichtige Module sind unter anderem `part_msdos` und `part_gpt` (Partitionstabellen lesen), `ext2` (Dateisysteme `ext2` bis `ext4`), `raid`, `raid5rec`, `raid6rec` und `mdraid` (Software-RAID), `lvm`, `gfxterm` (grafische Konsole), `vbe` (Grafiksystem) sowie `jpeg`, `tga` und `png` zum Lesen von Grafikdateien. Module

## GRUB-Menüeinträge

GRUB-Menüeinträge werden mit dem Schlüsselwort `menuentry` eingeleitet. Der nachfolgende Text steht in Anführungszeichen und darf internationale Zeichen enthalten. GRUB unterstützt auch Untermenüs. Bei der Bedienung von GRUB gelangen Sie gegebenenfalls mit `[Esc]` aus einem Untermenü zurück in das Hauptmenü.

menuentry und  
submenu

```
menuentry "Linux" {
  startkommandos ...
}
submenu 'Untermenü' {
  menuentry 'Eintrag 1' { ... }
  menuentry 'Eintrag 2' { ... }
}
```

Ein GRUB-2-Menüeintrag zum Start von Linux sieht in der Minimalvariante wie das folgende Beispiel aus: Linux starten

```
menuentry "Linux" {
  set root=(hd0,3)
  linux /boot/vmlinuz-n.n.n root=... ro quiet splash
  initrd /boot/initrd.img-n.n.n
}
```

`set root` gibt die Partition an, in der sich der Kernel- und die `initrd`-Datei befinden. Die Schlüsselwörter `linux` und `initrd` geben relativ zur Partition die Dateinamen an.

Anstelle von `linux` und `initrd` sind auch die Schlüsselwörter `linux16` und `initrd16` zulässig. GRUB greift dann auf ein 16-Bit-Protokoll zurück, um eine bessere Kompatibilität zu ganz alten BIOS-Versionen zu erreichen. Unter anderem kann dann mit der selten erforderlichen Option `vga=n` ein bestimmter Grafikmodus aktiviert werden. Keine Angst, Linux selbst wird natürlich trotzdem als 32- bzw. 64-Bit-System gestartet.

Die mit `linux` angegebenen Parameter werden an den Kernel übergeben. Unbedingt erforderlich sind `root` zur Angabe der Systempartition und `ro`, damit der Zugriff auf die Systempartition anfänglich nur lesend erfolgt. Alle weiteren Parameter sind distributionsabhängig. Einige Beispiele:

```
CentOS:  root=/dev/xxx ro rhgb quiet vconsole.font=... LANG=...
Debian:  root=UUID=xxx ro quiet
Fedora:  root=/dev/xxx ro rhgb quiet LANG=...
openSUSE: root=UUID=xxx resume=/dev/xxx splash=silent showopts quiet
Ubuntu:  root=UUID=xxx ro quiet splash
```

#### linux16 und linuxefi

Anstelle des Schlüsselworts `linux` finden Sie in der automatisch erzeugten Datei `grub.cfg` oft auch `linux16` oder `linuxefi`. Analog wird dann `initrd` durch `initrd16` bzw. `initrdefi` ersetzt.

- ▶ **linux16 und initrd16:** Die `xxx16`-Schlüsselwörter kommen zum Einsatz, wenn ein 16-Bit-Boot-Protokoll verwendet werden soll. Das klingt steinzeitlich, und selbstverständlich wird der Kernel trotzdem als 64-Bit-Programm gestartet; das 16-Bit-Boot-Protokoll ermöglicht es aber, den Grafikmodus während des Boot-Modus durch `vga=nnn` einzustellen. Die Hintergründe können Sie hier nachlesen:

*[https://www.gnu.org/software/grub/manual/html\\_node/linux16.html](https://www.gnu.org/software/grub/manual/html_node/linux16.html)*

- ▶ **linuxefi und initrdefi:** Die EFI-Varianten von `linux` und `initrd` kommen erwartungsgemäß auf EFI-Systemen zum Einsatz. Sie sind unbedingt erforderlich, wenn UEFI Secure Boot zum Einsatz kommt. (Ohne Secure Boot funktionieren auch auf EFI-Systemen die Schlüsselwörter `linux` und `initrd`.)

In den folgenden Beispielen verwende ich der Einfachheit halber immer die Schlüsselwörter `linux` und `initrd`.

#### Eigene Boot-Partition

Viele Linux-Distributionen (z.B. Fedora, CentOS und RHEL) sehen bei der Installation eine eigene Boot-Partition vor. In diesem Fall geben Sie mit `set root` die Boot-Partition an. Dafür entfällt bei `linux` und `initrd` dann die Angabe des Boot-Verzeichnisses:

```
menuentry "Linux - Mit eigener Boot-Partition" {
    set root=(hd0,2)
    linux /vmlinuz-n.n.n    root=... ro quiet splash
    initrd /initrd.img-n.n.n
}
```

Wenn die Partition mit den Kernel- und Initrd-Dateien Teil eines LVM-Systems und/oder eines Software-RAIDs ist, müssen Sie die entsprechenden GRUB-Module laden. Bei RAID-5 bzw. RAID-6 kommt noch das Modul `raid5rec` bzw. `raid6rec` hinzu. In `set root` können Sie nun die Systempartition in der Schreibweise (`lvname`) bzw. (`mdN`) angeben:

Boot-Verzeichnis  
in RAID-  
Partitionen oder  
Logical Volumes

```
menuentry "Linux - Mit Software-RAID" {
    insmod raid mdraid
    set root=(md0)
    linux /boot/vmlinuz-n.n.n root=... ro quiet splash
    initrd /boot/initrd.img-n.n.n
}
menuentry "Linux - Mit LVM" {
    insmod lvm
    set root=(vg1-root)
    linux /boot/vmlinuz-n.n.n root=... ro quiet splash
    initrd /boot/initrd.img-n.n.n
}
menuentry "Linux - LVM auf RAID-5" {
    insmod raid raid5rec mdraid lvm
    set root=(vg1-root)
    linux /boot/vmlinuz-n.n.n root=/dev/mapper/... ro quiet splash
    initrd /boot/initrd.img-n.n.n
}
```

Wenn Sie sich nicht auf Device-Nummern verlassen möchten, können Sie die Systempartition auch durch `search` anhand der UUID suchen. Ist das `search`-Kommando erfolgreich, wird die GRUB-Variablen `root` entsprechend geändert. Das funktioniert auch für LVM- und RAID-Partitionen, sofern vorher die richtigen GRUB-Module geladen wurden.

UUIDs statt  
Devices

```
menuentry "Linux - root-Variablen anhand UUID einstellen" {
    set root=(hd0,3)
    search --no-floppy --fs-uuid --set root 12af...
    linux /boot/vmlinuz-n.n.n root=... ro quiet splash
    initrd /boot/initrd.img-n.n.n
}
```

Um Windows auf einem BIOS-Rechner zu starten, wählen Sie mit `set root` die Windows-Systempartition aus und starten dann mit `chainloader +1` deren Bootloader. Beachten Sie, dass Windows standardmäßig zwei Partitionen einrichtet: eine rund 100 MiB große Boot-Partition mit den Dateien `bootmgr` und `bootsect.bak` sowie eine wesentlich größere Systempartition. In GRUB müssen Sie die Boot-Partition angeben. Das `search`-Kommando ist wie immer optional. Auch auf `drivemap` können Sie in der Regel verzichten. Das Kommando versucht Windows vorzumachen, dass sich Windows auf der ersten Festplatte befindet, selbst wenn das tatsächlich gar nicht der Fall ist. In seltenen Fällen ist das erforderlich, damit Windows startet.

Windows auf  
einem  
BIOS-Rechner  
starten

```

menuentry "Windows" {
    set root=(hd0,1)
    search --no-floppy --fs-uuid --set 12345678...
    drivemap -s (hd0) $root
    chainloader +1
}

```

Wie Sie Windows auf einem EFI-Rechner per GRUB starten, verrät der nächste Abschnitt.

### In einen anderen Bootloader verzweigen

GRUB gibt Ihnen mit dem bereits erwähnten `chainloader`-Kommando die Möglichkeit, in einen anderen Bootloader zu verzweigen. Die entsprechenden Anweisungen betten Sie am besten in eine Konfigurationsdatei ein, die Sie entsprechend dem Muster `/etc/grub.d/40_custom` gestalten.

In einen  
BIOS-Bootloader  
verzweigen

Wenn Sie auf Ihrem Rechner einen weiteren Bootloader in den Startsektor einer Partition installiert haben, können Sie aus dem GRUB-2-Menü in diesen Bootloader verzweigen. Dazu geben Sie dessen Partition mit `set root` an (optional auch mit `search`) und führen `chainloader +1` aus:

```

menuentry "GRUB in /dev/sdb7" {
    set root=(hd1,7)
    search --no-floppy --fs-uuid --set 12345678...
    chainloader +1
}

```

Sollten die obigen Zeilen nicht zum gewünschten Ergebnis führen, können Sie versuchen, auf `set root` zu verzichten und die gewünschte Partition direkt mit `chainloader` anzugeben, also:

```

menuentry "GRUB 0.97 in /dev/sdb7" {
    chainloader (hd1,7)+1 --force
}

```

### GRUB-Installationen in eine Partition

Unter GRUB 0.97 war es nichts Ungewöhnliches, GRUB an den Beginn einer gewöhnlichen Datenpartition zu installieren. Mit GRUB 2 sind solche Installationen aber nicht mehr empfehlenswert, und das Handbuch rät davon ausdrücklich ab! Auf BIOS-Rechnern soll GRUB 2 ausschließlich in den MBR der Festplatte/SSD installiert werden.



Auf EFI-Rechnern können Sie aus GRUB heraus direkt einen anderen EFI-Bootloader starten, auch wenn dies eine eher unübliche Vorgehensweise ist. Besser ist es in der Regel, das gewünschte Betriebssystem direkt in einem EFI-Boot-Menü auszuwählen.

In einen  
EFI-Bootloader  
verzweigen

Die folgenden Beispiele gehen davon aus, dass die EFI-Partition die erste Partition der ersten Festplatte ist. Andernfalls müssen Sie `set root` entsprechend anpassen.

```
menuentry "Windows-Bootloader starten (EFI)" {
    insmod part_gpt
    set root='(hd0,1)'
    chainloader /EFI/Microsoft/Boot/bootmgfw.efi
}

menuentry "Ubuntu-Bootloader starten (EFI)" {
    insmod part_gpt
    set root='(hd0,1)'
    chainloader /EFI/ubuntu/grubx64.efi
}
```

Geeignete EFI-Bootloader können Sie rasch mit diesem Kommando ermitteln:

```
root# find /boot/efi -name '*.efi' | sort
/boot/efi/EFI/Microsoft/Boot/bootmgfw.efi
/boot/efi/EFI/Microsoft/Boot/bootmgr.efi
/boot/efi/EFI/fedora/gcdx64.efi
/boot/efi/EFI/fedora/grubx64.efi
/boot/efi/EFI/fedora/MokManager.efi
/boot/efi/EFI/fedora/shim.efi
/boot/efi/EFI/fedora/shim-fedora.efi
...
```

Bei Windows-Installationen enthält `bootmgfw.efi` den Bootloader. `bootmgr.efi` lässt sich nicht starten.

## Individuelle Konfiguration

GRUB 2 ist so vorkonfiguriert, dass damit möglichst alle auf dem Rechner installierten Betriebssysteme gestartet werden können. Die Standardkonfiguration funktioniert gut und ist sicherlich in den meisten Fällen ausreichend. Dieser Abschnitt richtet sich nur an Linux-Anwender, die GRUB 2 individuell adaptieren möchten. Unter SUSE können Sie zum Einrichten bzw. Konfigurieren von GRUB auch das YaST-Modul `SYSTEM • KONFIGURATION DES BOOTLOADERS` verwenden.

Die Verarbeitung des Scripts `30_os-probe` dauert auf Rechnern mit vielen Partitionen recht lange. Wenn Sie dieses Script nicht benötigen – z. B. weil Sie zum Start anderer Linux-Distributionen das EFI oder selbst definierte GRUB-Menüeinträge verwenden –, tragen Sie die Zeile `GRUB_DISABLE_OS_PROBER=true` in `/etc/default/grub` ein.

os-probe  
deaktivieren

**Grafische Gestaltung**

Sie können das GRUB-Menü mit einem Bild hinterlegen. GRUB versteht die Formate JPG, PNG und TGA. Der erforderliche Code zur Integration einer Hintergrundgrafik ist bei Debian und Ubuntu im Script `05_debian_theme` bereits enthalten. Sie müssen lediglich drei Variablen ändern und dort den Dateinamen Ihres Bilds sowie die gewünschten Farben für den Text angeben:

```
# in /etc/grub.d/05_debian_theme
...
WALLPAPER="/boot/grub/myown.png"
COLOR_NORMAL="white/black"
COLOR_HIGHLIGHT="yellow/black"
```

GRUB 2 enthält auch eine experimentelle Unterstützung von Themen, die neben dem Hintergrund auch verschiedene Schriften und eine vom klassischen GRUB-Menü abweichende Bedienung ermöglichen. Zu den wenigen Distributionen, die davon Gebrauch machen, zählt openSUSE. Das Thema wird durch die Textdatei `/boot/grub2/themes/openSUSE/theme.txt` bestimmt. Für die Integration dieser Datei in die GRUB-Konfiguration ist `/etc/grub.d/00_header` verantwortlich. Tipps zur Realisierung eines vergleichbaren Themas für Ubuntu finden Sie hier:

<http://ubuntuforums.org/showthread.php?t=2081013>

**Defaultbetriebs-system festlegen**

In `/etc/default/grub` können Sie die Nummer des GRUB-Menüeintrags festlegen, der automatisch gestartet wird. In der Praxis bringt das wenig: Wenn Sie beispielsweise möchten, dass standardmäßig Windows gestartet wird und dass dessen Menüeintrag an der zehnten Stelle steht, dann stellen Sie `GRUB_DEFAULT=9` ein, weil die Zählung mit 0 beginnt.

Es kann aber sein, dass das GRUB-Menü beim nächsten Kernel-Update zusätzliche Einträge erhält – und dann ist Ihre Einstellung falsch. Besser ist es, `GRUB_DEFAULT=0` zu belassen und stattdessen den gewünschten GRUB-Menüeintrag vor allen anderen Einträgen einzufügen. Das gelingt am einfachsten durch ein zusätzliches Script in `/etc/grub.d`, wobei der Dateiname mit einer Nummer kleiner 10 beginnt. Die folgenden Zeilen können als Muster dienen:

```
#!/bin/sh
exec tail -n +3 $0
# Datei /etc/grub.d/09_boot-windows-by-default
menuentry "Windows 7" {
    set root=(hd0,1)
    chainloader +1
}
```

Eine andere Möglichkeit besteht darin, den Menüeintrag in `GRUB_DEFAULT` exakt anzugeben, z. B. so:

```
GRUB_DEFAULT='Windows 7 (loader) (on /dev/sda1)'
```

Diese Vorgehensweise ist aber nur dann zweckmäßig, wenn sich der Menüeintrag bei GRUB-Updates nicht ändert. Für Windows funktioniert das gut. Bei Linux enthalten die automatisch generierten Menüeinträge aber oft die Kernelversion – und die ändert sich bei jedem Kernel-Update.

## 23.4 Manuelle GRUB-Installation und Erste Hilfe

Normalerweise wird GRUB 2 während der Installation Ihrer Linux-Distribution korrekt eingerichtet. In der Folge werden Sie zwar vielleicht Änderungen an der GRUB-Konfiguration bzw. `grub.cfg` durchführen, die GRUB-Installation als solche müssen Sie aber nicht mehr anrühren.

Die folgenden Abschnitte für BIOS- und EFI-PCs sind nur dann relevant, wenn Sie aus irgendeinem Grund eine manuelle Installation von GRUB durchführen möchten oder wenn Sie eine defekte GRUB-Installation reparieren müssen – z. B. weil ein anderes Betriebssystem den Inhalt des MBR überschrieben hat.

### BIOS-PCs

Das Script `grub-install`, das unter Fedora sowie openSUSE `grub2-install` heißt, installiert den Bootloader in die ersten Sektoren der angegebenen Festplatte bzw. SSD, also in den MBR sowie in weitere Sektoren, die sich vor dem Beginn der ersten Partition befinden. Als einzigen Parameter übergeben Sie in der Regel den Device-Namen des Datenträgers. Dabei ist sowohl die Linux- als auch die GRUB-Schreibweise zulässig, also `/dev/sda` oder `(hd0)`.

Manuelle  
Installation mit  
`grub-install`

```
root# grub-install /dev/sda
```

Eine Installation in den Startsektor einer Partition ist theoretisch möglich (also z. B. `grub-install /dev/sda3` bzw. `(hd0,3)`), wird aber anders als in GRUB 0.97 nicht mehr empfohlen: Der Platz zum Einbetten des GRUB-Codes ist zu klein, und die deswegen erforderliche Verwendung von Link-Listen mit Querverweisen auf woanders befindliche Datenblöcke gilt als instabil. Wenn diese Argumente Sie nicht überzeugen, können Sie die Installation mit der Option `--force` erzwingen.

Grundsätzlich unterstützt Linux auch auf BIOS-Rechnern die Installation auf eine Festplatte oder SSD mit einer GUID Partition Table (GPT). Allerdings empfiehlt das GRUB-Handbuch in diesem Sonderfall, für die GRUB-Installation eine eigene Partition in der Größe von 1 MiB mit dem Flag `bios_grub` vorzusehen. Diese Partition ist nur für die Installation eines BIOS-kompatiblen Bootloaders gedacht. Diese `bios_grub`-Partition braucht nicht formatiert zu werden. Wenn Sie die Partition manuell ein-

Sonderfall  
BIOS-PC und GPT  
(`bios_grub`-  
Partition)

richten, setzen Sie mit `parted` das Flag `bios_grub`, wobei Sie `N` durch die gewünschte Partitionsnummer ersetzen:

```
root# parted /dev/sda set N bios_grub on
```

### Achtung

Markieren Sie mit dem `bios_grub`-Flag keine Partition, die Daten enthält. Bei der GRUB-Installation wird der Beginn der Partition überschrieben. Ein eventuell auf der Partition befindliches Dateisystem wird dadurch vollständig zerstört!

Wenn GRUB bei der Installation die Existenz einer `bios_grub`-Partition feststellt, installiert es den Beginn der GRUB-Codes wie üblich in den MBR der Festplatte, den restlichen GRUB-Code aber in die gekennzeichnete `bios_grub`-Partition. Diese Vorgehensweise ist bei GRUB-Installationen auf Festplatten mit GPT zwar nicht zwingend erforderlich, gilt aber als wesentlich robuster, vor allem, wenn auch andere Betriebssysteme (Windows) auf dem Rechner installiert werden.

Beachten Sie aber, dass es auch einen Nachteil gibt: Wenn eine derartige Partition existiert, ist es unmöglich, mehrere GRUB-2-Installationen parallel durchzuführen, weil bei jeder neuerlichen GRUB-Installation der Inhalt der `bios_grub`-Partition überschrieben wird. Weitere Informationen zum Thema GRUB 2 und GPT können Sie hier nachlesen:

[http://www.gnu.org/software/grub/manual/html\\_node/BIOS-installation.html](http://www.gnu.org/software/grub/manual/html_node/BIOS-installation.html)

<http://www.wensley.org.uk/gpt>

### GRUB in einem Live-System reparieren (BIOS)

Wenn die GRUB-Installation fehlgeschlagen ist oder durch ein anderes Betriebssystem überschrieben wurde, müssen Sie GRUB von einer Live-CD mit aktuellen GRUB-2-Tools neu installieren. Nach dem Systemstart wechseln Sie in den `root`-Modus (`sudo -s` bei Ubuntu), binden die Systempartition und die aktiven `/dev`-, `/proc`- und `/sys`-Verzeichnisse in das Dateisystem ein und führen dann `chroot` aus. Gegebenenfalls binden Sie nun auch die Boot-Partition in das neue `root`-Dateisystem ein.

Nun aktualisieren Sie die GRUB-Konfiguration und schreiben mit `grub[2]-install` GRUB an den gewünschten Ort, zumeist in den MBR der ersten Festplatte. Wie üblich müssen Sie in den folgenden Kommandos `/dev/sda<n>` durch Ihre eigenen Device-Namen ersetzen!

```
root# mkdir /syspart
root# mount /dev/sda2 /syspart                (Systempartition)
root# mount -o bind /dev /syspart/dev
root# mount -o bind /proc /syspart/proc
root# mount -o bind /sys /syspart/sys
root# chroot /syspart
```

```

root# mount /dev/sda1 /boot           (Boot-Partition, falls vorhanden)
root# update-grub                    (Debian/Ubuntu)
root# grub-install /dev/sda          (Debian/Ubuntu, Forts.)
root# grub2-mkconfig -o /pfad/zu/grub.cfg (Fedora/openSUSE)
root# grub2-install /dev/sda        (Fedora/openSUSE, Forts.)
root# exit

```

## Manuelle Installation und Erste Hilfe für EFI-PCs

Eine manuelle Installation von GRUB 2 auf einem EFI-Rechner ist denkbar einfach. Sie müssen an `grub-install` bzw. `grub2-install` keinerlei Parameter übergeben: Unter CentOS, Fedora und RHEL müssen Sie allerdings vorher das Paket `grub2-efi-modules` installieren.

Manuelle  
Installation mit  
`grub-install`

```

root# grub-install      (Debian, Ubuntu)
root# grub2-install     (CentOS, Fedora, SUSE, RHEL)

```

Durch dieses Kommando wird das Verzeichnis `/boot/efi/EFI/distributionsname` erzeugt. In dieses Verzeichnis wird eine neue Boot-Datei mit der Endung `.efi` geschrieben, die den GRUB-Code enthält. Außerdem wird die `.efi`-Datei in die Liste der EFI-Boot-Einträge aufgenommen und dort an den ersten Platz gestellt. `grub-install` greift dazu auf das Kommando `efibootmgr` zurück, das ich Ihnen im übernächsten Abschnitt näher vorstelle.

Damit `grub-install` erfolgreich ausgeführt werden kann, müssen einige Voraussetzungen erfüllt sein:

- ▶ Die GRUB-Konfiguration in der Datei `/boot/grub[2]/grub.cfg` muss vorbereitet sein.
- ▶ Die EFI-Partition muss unter dem Pfad `/boot/efi` in den Verzeichnisbaum eingebunden sein.
- ▶ Das Kommando `efibootmgr` aus dem gleichnamigen Paket muss installiert sein.
- ▶ Das Kernelmodul `efivars` muss geladen sein. `modprobe efivars` gelingt allerdings nur, wenn die Distribution im EFI-Modus gestartet wurde, nicht im BIOS-Modus. Deswegen ist es ohne ein EFI-boot-fähiges Live-System schwierig, eine zuerst im BIOS-Modus installierte Linux-Distribution auf EFI umzustellen.

Im Prinzip erfolgt die GRUB-Reparatur für ein EFI-System ganz ähnlich wie bei einem BIOS-System. Entscheidend ist aber, dass Sie das Live-System im EFI-Modus starten, nicht im BIOS-Modus.

GRUB in einem  
Live-System  
reparieren (EFI)

Anschließend wechseln Sie in den `root`-Modus, binden die Systempartition und die aktiven `/dev-`, `/proc-` und `/sys-`Verzeichnisse in das Dateisystem ein und führen dann `chroot` aus. Gegebenenfalls binden Sie nun auch die Boot-Partition in das neue `root`-Dateisystem ein. Nun aktualisieren Sie die GRUB-Konfiguration und schreiben sie mit

grub-install in die EFI-Partition. Vergessen Sie nicht, in den folgenden Kommandos /dev/sda<n> durch Ihre eigenen Device-Namen zu ersetzen!

```

root# mkdir /syspart
root# mount /dev/sda2 /syspart                (Systempartition)
root# mount -o bind /dev /syspart/dev
root# mount -o bind /proc /syspart/proc
root# mount -o bind /sys /syspart/sys
root# chroot /syspart
root# mount /dev/sda1 /boot/efi              (EFI-Partition)

root# update-grub                            (Debian und Ubuntu)
root# grub2-mkconfig -o /boot/grub2/grub.cfg  (Fedora)
root# grub2-mkconfig -o /boot/grub2-efi/grub.cfg (openSUSE)

root# grub[2]-install
root# exit

```

### GRUB-Kommandos zum Linux-Start manuell eingeben

Wenn Sie zwar GRUB starten können, aber nach der Auswahl des Linux-Menüeintrags der Linux-Start scheitert, können Sie innerhalb des GRUB-Menüs mit **[c]** in den interaktiven Modus wechseln und dann die folgenden Kommandos ausführen:

```

grub> set root=(hd0,1)
grub> linux /vmlinuz root=/dev/sda1
grub> initrd /initrd.img
grub> boot

```

Statt (hd0,1) und /dev/sda1 geben Sie den Namen Ihrer Linux-Systempartition an. Die Dateien /vmlinuz und /initrd.img verweisen bei den meisten Distributionen auf die aktuellste Kernel- und Initrd-Datei im Verzeichnis /boot. Sollte das bei Ihrer Distribution nicht der Fall sein, müssen Sie den Ort der Kernel- und der Initrd-Datei exakt angeben. GRUB unterstützt Sie bei der Eingabe mit der Eingabevervollständigung durch **[Tab]**.

### EFI-Boot-Einträge und -Einstellungen ändern (efibootmgr)

Woher weiß EFI eigentlich, welche Betriebssysteme installiert sind bzw. welche Boot-Einträge es anzeigen soll? Bei EFI-Mainboards werden diese Informationen in einem nichtflüchtigen Speicher (NVRAM) festgehalten. Jedes Mal, wenn ein neues Betriebssystem installiert wird, wird nach dem Einrichten der \*.efi-Datei in der EFI-Partition ein entsprechender Eintrag im NVRAM gespeichert.

**efibootmgr** Unter Linux können Sie die so gespeicherten EFI-Daten mit dem Kommando efibootmgr auslesen oder verändern. Das Kommando setzt voraus, dass das Kernelmodul

efivars geladen ist. Sollte das nicht der Fall sein, führen Sie `modprobe efivars` aus. Das gelingt nur, wenn Linux im EFI-Modus gebootet wurde. Verwenden Sie für Reparaturarbeiten gegebenenfalls ein Linux-Live-System, das sich im EFI-Modus starten lässt!

Wenn `efibootmgr` ohne weitere Optionen ausgeführt wird, listet es die EFI-Boot-Einträge sowie einige weitere Parameter des EFI-Bootloaders auf. Das folgende Ergebnis bedeutet, dass auf dem Rechner Ubuntu, Fedora und Windows im EFI-Modus installiert sind. Beim nächsten Neustart wird nach einer Wartezeit von einer Sekunde automatisch Ubuntu gestartet (`BootCurrent`). Während dieser Wartezeit kann mit einer mainboard-spezifischen Tastenkombination (bei meinem Testrechner: `[F8]`) das EFI-Menü angezeigt werden.

```
root# efibootmgr
BootCurrent: 0000
Timeout: 1 seconds
BootOrder: 0000,0005,0003,0001,0002
Boot0000* ubuntu
Boot0001* Hard Drive
Boot0002* CD/DVD-Laufwerk
Boot0003* Windows Boot Manager
Boot0005* Fedora
```

Wenn Sie möchten, dass beim nächsten Neustart einmalig Fedora gestartet werden soll, legen Sie dessen Boot-Eintrag mit `-n` fest:

```
root# efibootmgr -n 5
```

Soll die Boot-Reihenfolge hingegen bleibend geändert werden, geben Sie den gewünschten Eintrag mit der Option `-o` an:

```
root# efibootmgr -o 5
```

Das folgende Kommando erzeugt einen neuen EFI-Boot-Eintrag. Die Pfadangabe ist relativ zur EFI-Partition (also `/boot/efi`), und als Verzeichnistrenner muss `\` verwendet werden. Die Verdoppelung der `\`-Zeichen ist erforderlich, weil die Shell ein einfaches `\`-Zeichen als Kennzeichnung von Sonderzeichen interpretiert. Mit der Option `-L` geben Sie den Namen an, der im EFI-Menü angezeigt werden soll.

```
root# efibootmgr -c -l \\EFI\\test\\abc.efi -L abc
```

Natürlich können Sie Boot-Einträge auch wieder entfernen. Dazu geben Sie mit `-b` die Nummer des Eintrags an:

```
root# efibootmgr -b 6 -B
```

Weitere Optionen des Kommandos `efibootmgr` sind in dessen `man`-Seite dokumentiert.





# Kapitel 24

## Das Init-System

Dieses Kapitel beschreibt die Vorgänge, die vom Kernelstart bis hin zum Login stattfinden. Der Kernel startet als ersten Prozess das Init-System. Es kümmert sich um die Basiskonfiguration des Systems, um das Einbinden von Dateisystemen und um den Start zahlloser Netzwerkdienste und -dämonen. Wie so oft in der Linux-Welt gibt es nicht *ein* Init-System, sondern mehrere:

- ▶ **Beim traditionellen Init-V-System**, dessen Konzepte mehrere Jahrzehnte alt sind, kümmern sich eine Menge durch Links verbundener Scripts um die Initialisierung des Rechners. Sein Name stammt vom Unix-Betriebssystem System V.

Auch wenn das Init-V-System bei immer weniger Linux-Distributionen zum Einsatz kommt, bleibt es relevant: Modernere Init-Systeme sind in der Regel Init-V-kompatibel. Bei vielen Distributionen, die bereits neuere Init-Systeme einsetzen, gibt es weiterhin Init-V-Scripts.

- ▶ **Upstart** war ein ereignisorientiertes Init-System. Seine Entwicklung wurde 2014 eingestellt. Es kam in Ubuntu in den Versionen von 6.10 bis 14.10 zum Einsatz, in Fedora von Version 9 bis 13 sowie in RHEL 6.
- ▶ **systemd** ist das Init-System der Linux-Gegenwart. Es kam erstmalig in Fedora 15 zum Einsatz. Fast alle gängigen Linux-Distributionen sind seither auf systemd umgestiegen. Zu den Hauptvorteilen zählt die höhere Geschwindigkeit.

Tabelle 24.1 fasst zusammen, welches Init-System in welcher Distribution zum Einsatz kommt. Die Spalte RHEL gilt für Red Hat Enterprise Linux sowie für alle dazu kompatiblen Distributionen, also z. B. CentOS und Scientific Linux. Analog gilt die Ubuntu-Spalte auch für Ubuntu-Derivate wie Kubuntu oder Xubuntu.

| Init-System | Debian | Fedora   | openSUSE | RHEL  | Ubuntu         |
|-------------|--------|----------|----------|-------|----------------|
| Init-V      | bis 7  | bis 8    | bis 11.4 | bis 5 | bis 6.04       |
| Upstart     |        | 9 bis 13 |          | 6     | 6.10 bis 14.10 |
| systemd     | ab 8   | ab 14    | ab 12.1  | ab 7  | ab 15.04       |

**Tabelle 24.1** Init-Systeme je nach Distribution und Version

In diesem Kapitel erläutere ich Ihnen zuerst die Grundlagen von `systemd` und des Init-V-Systems. Anschließend lernen Sie, wie Sie eigene Init-Scripts bzw. -Konfigurationsdateien einrichten. Nach einigen distributionsspezifischen Details zum Init-Prozess stelle ich Ihnen zuletzt kurz den *Internet Service Daemon* vor: Dieses nur noch selten eingesetzte Programm überwacht Netzwerk-Ports. Wenn dort Anfragen eintreffen, kann es ein Programm starten, das in der Lage ist, auf diese Anfragen zu reagieren.

## 24.1 systemd

`systemd` ist ein relativ neues Init-System, das vom Red-Hat-Mitarbeiter Lennart Poettering entwickelt wurde und erstmalig 2011 in Fedora 15 zum Einsatz kam. Mittlerweile sind die meisten aktuellen Linux-Distributionen diesem Vorbild gefolgt.

### Verzeichnisorte

Dieser Abschnitt beschreibt `systemd`, wie es momentan unter Fedora implementiert ist. Einzelne Distributionen verwenden andere Verzeichnisorte, z. B. nutzt openSUSE `/usr/lib/systemd` anstelle von `/lib/systemd`. Außerdem kommen je nach Distributionen mitunter ältere `systemd`-Versionen als unter Fedora zum Einsatz.

Der vielleicht wichtigste Unterschied zwischen dem herkömmlichen Init-System und `systemd` besteht darin, dass die Konfiguration nicht durch Shell-Scripts erfolgt, sondern durch einfache Textdateien. `systemd` selbst ist ein kompiliertes Programm, wodurch sich ein spürbarer Geschwindigkeitsvorteil ergibt. `systemd` versucht, die Dienste möglichst parallel zu starten, was bei Multi-Core-Systemen von Vorteil ist.

`systemd` verwendet `Cgroups` zur Ausführung und Überwachung von Prozessen. `Cgroups` steht für *Control Groups*. Dabei handelt es sich um eine Kernelfunktion, um die Ressourcen eines Prozesses zu limitieren (CPU, Speicher, I/O). `systemd` startet jeden Prozess in einer eigenen `Cgroup`. Wenn die Anzahl der Prozesse in dieser Gruppe auf 0 sinkt, weiß `systemd`, dass der Prozess beendet wurde oder abgestürzt ist, und kann ihn gegebenenfalls neu starten.

Auf immer mehr Distributionen ist `systemd` nicht nur für die Verwaltung von Systemdiensten verantwortlich, sondern kümmert sich nach einem Login auch um den Start von benutzerspezifischen Prozessen, beispielsweise bei der Initialisierung einer Gnome-Session.

Die folgenden Punkte geben einen kurzen Überblick über den Systemstart:

Ablauf

- ▶ GRUB lädt und startet den Kernel.
- ▶ Der Kernel startet den Prozess `systemd`. Unter CentOS, Fedora und RHEL befindet sich diese Datei im Verzeichnis `/usr/lib/systemd`. Unter Debian und Fedora wird hingegen `/sbin/init` ausgeführt. Dabei handelt es sich aber in Wirklichkeit um einen Link auf die Datei `/lib/systemd/systemd`.
- ▶ Alle an den Kernel übergebenen Optionen (die sogenannten Boot-Optionen), die der Kernel nicht kennt, gibt der Kernel an `systemd` weiter.
- ▶ `systemd` wertet die Konfigurationsdateien in `/lib/systemd` und `/etc/systemd` aus, führt die dort genannten Initialisierungsarbeiten aus, startet Netzwerkdienste und den Display-Manager samt dem Grafiksystem.
- ▶ Der Benutzer sieht nach einigen Sekunden die Login-Aufforderung. Bei manchen Distributionen wird nach dem Login eine weitere `systemd`-Instanz auf Benutzerebene gestartet, um diverse für das Desktop-System erforderliche Prozesse auszuführen.

`systemd` ist also der erste laufende Prozess. Alle weiteren Prozesse werden entweder direkt von `systemd` oder indirekt durch seine Subprozesse gestartet. Führen Sie in einem Terminal `ps tree` aus, dann erkennen Sie sofort die dominierende Rolle von `systemd`! Beim Herunterfahren des Rechners ist `systemd` der letzte noch laufende Prozess, der sich um das korrekte Beenden aller anderen Prozesse kümmert.

Während der Systeminitialisierung werden die Dinge erledigt, die während des Rechnerstarts nur einmal getan werden müssen:

System-  
initialisierung

- ▶ diverse Systemvariablen initialisieren (inklusive Host- und Domainname)
- ▶ das `/proc`-Dateisystem sowie diverse andere temporäre Dateisysteme aktivieren
- ▶ Datum und Uhrzeit einstellen
- ▶ Tastaturlayout für die Textkonsole einstellen
- ▶ `udev`-System starten
- ▶ eventuell RAID und LVM aktivieren
- ▶ Dateisysteme überprüfen
- ▶ Root-Partition im Read-Write-Modus neu einbinden
- ▶ Dateisystem der weiteren Partition überprüfen, Partitionen einbinden
- ▶ Netzwerkgrundfunktionen teilweise oder ganz initialisieren

## Administration

### Dienste starten und beenden

Das zentrale Kommando zur Administration von systemd lautet `systemctl`. Damit können Sie durch eine `*.service`-Datei beschriebene Dienste manuell starten, stoppen etc.

```
root# systemctl start  sshd  (SSH-Dämon starten)
root# systemctl stop   sshd  (SSH-Dämon stoppen)
root# systemctl restart sshd  (SSH-Dämon neu starten)
root# systemctl reload  sshd  (Konfiguration des SSH-Dämons neu einlesen)
root# systemctl status  sshd  (Status des SSH-Dämons ermitteln)
```

`systemctl` kann auch dazu verwendet werden, um einen Dienst dauerhaft zu aktivieren bzw. zu deaktivieren (so wie `chkconfig xxx on/off`):

```
root# systemctl enable  sshd
ln -s '/lib/systemd/system/ssh.service' \
    '/etc/systemd/system/multi-user.target.wants/ssh.service'
root# systemctl disable sshd
rm '/etc/systemd/system/multi-user.target.wants/ssh.service'
```

Bei der Aktivierung von Diensten wird also ein neuer Link eingerichtet, und bei der Deaktivierung wird dieser Link wieder entfernt.

### Start und Enable zugleich

Oft ist es erforderlich, einen Dienst zu starten und ihn dauerhaft zu aktivieren. Anstelle von `systemctl start` und `systemctl enable` können Sie in solchen Fällen bei aktuellen systemd-Versionen einfach `systemctl enable --now` ausführen.

Analog entspricht `systemctl disable --now` den Einzelkommandos `systemctl stop` und `systemctl disable`. Beachten Sie, dass die Option `--now` bei älteren systemd-Versionen nicht zur Verfügung steht.

### Liste aller systemd-Dienste

Wenn `systemctl` ohne weitere Parameter aufgerufen wird, liefert es eine Liste aller Prozesse, die durch systemd verwaltet werden:

```
user$ systemctl
UNIT                                LOAD    ACTIVE SUB    JOB DESCRIPTION
sys-devices-xxx                    loaded active plugged /sys/devices/virtual/block/dm-0
sys-devices-yyy                    loaded active plugged /sys/devices/virtual/block/dm-1
...
-.mount                             loaded active mounted /
boot.mount                         loaded active mounted /boot
dev-hugepages.mount                loaded active mounted Huge Pages File System
...
chronyd.service                    loaded active running NTP client/server
colord.service                     loaded active running Manage Color Profiles
```

```

crond.service          loaded active running  Command Scheduler
cups.service           loaded active running  CUPS Scheduler
...
146 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.

```

## Targets

»Targets« definieren Betriebszustände, die ein Linux-System erreichen soll. Das gerade gültige Default-Target ermitteln Sie mit `systemctl get-default`:

```

root# systemctl get-default
multi-user.target

```

Übliche Default-Targets sind das `multi-user.target` für Server (volle Funktionalität, aber ohne Grafiksystem) sowie das `graphical.target` für den Desktop-Betrieb. Targets können voneinander abhängen, mehrere Targets können zugleich aktiv sein. Eine Liste aller Targets ermitteln Sie mit dem folgenden Kommando. Das Ergebnis zeigt die aktiven Targets eines Fedora-Systems mit aktiver grafischer Benutzeroberfläche:

```

user$ systemctl list-units --type=target
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
basic.target                        loaded active active Basic System
cryptsetup.target                  loaded active active Encrypted Volumes
getty.target                        loaded active active Login Prompts
graphical.target                   loaded active active Graphical Interface
local-fs-pre.target                 loaded active active Local File Systems (Pre)
local-fs.target                     loaded active active Local File Systems
multi-user.target                   loaded active active Multi-User System
network-online.target               loaded active active Network is Online
network-pre.target                  loaded active active Network (Pre)
network.target                      loaded active active Network
nfs-client.target                   loaded active active NFS client services
nss-user-lookup.target              loaded active active User and Group Name Lookups
paths.target                        loaded active active Paths
remote-fs-pre.target                loaded active active Remote File Systems (Pre)
remote-fs.target                    loaded active active Remote File Systems
rpc_pipefs.target                   loaded active active rpc_pipefs.target
slices.target                       loaded active active Slices
sockets.target                      loaded active active Sockets
sound.target                        loaded active active Sound Card
sshd-keygen.target                  loaded active active sshd-keygen.target
swap.target                         loaded active active Swap
sysinit.target                      loaded active active System Initialization
timers.target                       loaded active active Timers

```

LOAD = Reflects whether the unit definition was properly loaded.  
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.  
SUB = The low-level unit activation state, values depend on unit type.  
23 loaded units listed. Pass --all to see loaded but inactive units, too.

Um den aktuellen Betriebszustand zu ändern, führen Sie das folgende Kommando aus:

```
root# systemctl isolate reboot.target    (Neustart des Rechners)
```

#### Default-Target einstellen

Das Default-Target wird durch den Link `/etc/systemd/system/default.target` definiert. Die Veränderung des Default-Targets und damit die Neueinstellung dieses Links gelingt am einfachsten mit `systemctl set-default`:

```
root# In systemctl set-default multi-user.target
Removed symlink /etc/systemd/system/default.target.
Created symlink from /etc/systemd/system/default.target
to /usr/lib/systemd/system/multi-user.target
```

#### Emergency-Target

Anders als das Init-V-System kennt `systemd` keinen Single-User-Modus. Dessen Rolle übernimmt das Emergency-Target. Dieses Target kann aktiviert werden, wenn Sie beim Booten die `linux`-Zeile des GRUB-Boot-Eintrags um das Schlüsselwort `single` ergänzen. Die Systeminitialisierung wird dann auf ein Minimum beschränkt; insbesondere gibt es keinen Netzwerkzugang. Ein Login ist nur in der ersten Textkonsole mit dem `root`-Passwort möglich.

Das Emergency-Target ist für Reparaturarbeiten gedacht. Wenn diese erfolgreich abgeschlossen sind, können Sie mit `systemctl isolate multi-user` oder `graphical` den normalen Betriebszustand aktivieren.

#### Verhalten bei Strg-Alt-Entf

Wenn Sie in einer Textkonsole `[Strg]+[Alt]+[Entf]` drücken, wird der Rechner neu gestartet. Verantwortlich dafür ist die spezielle Target-Datei `/lib/systemd/system/ctrl-alt-del.target`.

## Konfiguration

Die Konfigurationsdateien für `systemd` befinden sich in den Verzeichnissen `/etc/systemd/` sowie `/lib/systemd/` (Debian, Fedora, RHEL, Ubuntu) bzw. `/usr/lib/systemd` (openSUSE). Insgesamt handelt es sich dabei um rund 500 Dateien – die Konfiguration ist also ziemlich komplex und zumindest ebenso unübersichtlich wie beim Init-V-System.

Eine zentrale Rolle im `systemd`-Konzept spielen Units: Sie beschreiben Objekte, die durch das Init-System gesteuert werden sollen. Dazu zählen nicht nur Dienste, die gestartet oder gestoppt werden müssen, sondern auch Netzwerkschnittstellen, `mount`-Verzeichnisse, `Swap`-Partitionen etc.

#### Target-Dateien

Mit `*.target`-Dateien können mehrere Units zu einer Gruppe verbunden werden. Targets sind mit Init-V-Runleveln vergleichbar, es gibt aber in der Regel wesentlich mehr Targets, die sich aufeinander beziehen können. Die folgenden Zeilen zeigen die `graphical.target`-Datei:

```
# Datei /lib/systemd/system/graphical.target
[Unit]
Description=Graphical Interface
Documentation=man:systemd.special(7)
Requires=multi-user.target
Wants=display-manager.service
Conflicts=rescue.service rescue.target
After=multi-user.target rescue.service rescue.target display-manager.service
AllowIsolate=yes
```

Das `graphical.target` setzt also das `multi-user.target` voraus und erfordert außerdem den `display-manager.service`. Jedes Target kann mit einem zusätzlichen `<name>.target.wants`-Verzeichnis verknüpft werden, in dem durch Dateien bzw. durch Links auf Dateien weitere Units aufgezählt werden, die ebenfalls aktiviert werden sollen. Beim `graphical.target` enthält dieses Listing nur einen Eintrag:

```
root# cd /lib/systemd/system/graphical.target.wants
root# ls -l
systemd-update-utmp-runlevel.service -> ../systemd-update-utmp-runlevel.service
```

Wesentlich interessanter ist das Verzeichnis `/lib/systemd/system/sysinit.target.wants` mit diversen Links auf `*.service`-Dateien zur Systeminitialisierung:

```
root# ls /lib/systemd/system/sysinit.target.wants/
cryptsetup.target
dev-hugepages.mount
dev-mqueue.mount
dracut-shutdown.service
kmod-static-nodes.service
ldconfig.service
...
systemd-udev.service
systemd-udev-trigger.service
systemd-update-done.service
systemd-update-utmp.service
systemd-vconsole-setup.service
```

`*.service`-Dateien beschreiben, welche Voraussetzungen für den Start eines Dienstes erfüllt sein müssen und welches Kommando gestartet werden soll. Diese Dateien befinden sich in den Verzeichnissen `/etc/systemd/system` und `/lib/systemd/system`. Als Beispiel zeigt das folgende Listing die Datei `httpd.service`, die unter Fedora für den Start des Webservers Apache verantwortlich ist:

**Service-Dateien**

```
# Datei /lib/systemd/system/httpd.service
[Unit]
Description=The Apache HTTP Server
After = network.target remote-fs.target nss-lookup.target
```

```
[Service]
Type=notify
Environment=LANG=C
ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND
ExecReload=/usr/sbin/httpd $OPTIONS -k graceful
KillSignal=SIGWINCH
KillMode=mixed
PrivateTmp=true

[Install]
WantedBy=multi-user.target
```

### systemd auf Benutzerebene

Auf einigen Distributionen, z. B. bei Fedora, ist systemd auch auf Benutzerebene aktiv und kümmert sich dort um die Ausführung einiger Prozesse nach einem Login in Gnome. Dazu wird durch das PAM-Modul pam\_systemd ein eigener systemd-Prozess gestartet, der nur die Rechte eines gewöhnlichen Benutzers hat:

```
root# ps axu | grep 'systemd '
USER      PID ... COMMAND
root         1    /usr/lib/systemd/systemd --switched-root --system
gdm        937    /usr/lib/systemd/systemd --user
kofler    1616    /usr/lib/systemd/systemd --user
```

Das Kommando `systemctl` bezieht sich standardmäßig immer auf die Systemebene, auch wenn das Kommando ohne `root`-Rechte ausgeführt wird. Um die von systemd auf Benutzerebene gestarteten Prozesse zu ermitteln, müssen Sie die zusätzliche Option `--user` übergeben:

```
user$ systemctl --user list-units --type=service
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
at-spi-dbus-bus                     loaded active running Accessibility ...
dbus                                 loaded active running D-Bus User Message Bus
evolution-addressbook-factory        loaded active running Evolution address ...
evolution-calendar-factory           loaded active running Evolution calendar ...
evolution-source-registry            loaded active running Evolution source ...
gnome-terminal-server                loaded active running GNOME Terminal Server
gvfs-afc-volume-monitor               loaded active running Virtual filesystem ...
gvfs-daemon                          loaded active running Virtual filesystem ...
gvfs-metadata                        loaded active running metadata service
gvfs-goa-volume-monitor               loaded active running GNOME Online ...
gvfs-gphoto2-volume-monitor          loaded active running digital camera ...
gvfs-mtp-volume-monitor              loaded active running Media Transfer ...
gvfs-udisks2-volume-monitor          loaded active running disk device monitor
tracker-store                        loaded active running Tracker metadata ...
xdg-permission-store                 loaded active running sandboxed app ...
```



systemd durchsucht die folgenden Verzeichnisse nach Service-Dateien:

```
/lib/systemd/user
/etc/systemd/user
~/.local/share/systemd/user
~/.config/share/systemd/user
```

Beachten Sie, dass systemd auf User-, aber nicht auf Session-Ebene läuft. Der systemd-Prozess wird beim ersten Login des betreffenden Benutzers gestartet und mit der letzten aktiven Session des Benutzers beendet. Das gilt auch dann, wenn ein Benutzer mehrfach eingeloggt ist und daher mehrere Sessions aktiv sind. **loginctl**

Einen Überblick über alle Sessions in der Kontrolle von systemd gibt das Kommando loginctl:

```
user$ loginctl
SESSION  UID  USER   SEAT  TTY
      c1   42   gdm     seat0 /dev/tty1
      2   1000 kofler  seat0 /dev/tty2
      4   1000 kofler
```

Mit loginctl show-session können Sie Details zu einer Session ermitteln und so unter anderem feststellen, ob die Session Xorg, Wayland oder nur eine TTY-Schnittstelle verwendet, wie dies bei SSH-Logins der Fall ist:

```
root# loginctl show-session -p Type 2
Type=wayland
```

### Zusatzfunktionen

Im Zuge der Umstellung auf systemd hat dessen Entwickler Lennart Poettering vorgeschlagen, auch diverse Konfigurationsdateien zu vereinheitlichen, die in der Vergangenheit von nahezu jeder Distribution an einem anderen Ort gespeichert wurden und für die zum Teil auch unterschiedliche Syntaxregeln galten.

**Vereinheitlichung von Konfigurationsdateien**

Als erste Distribution hat Fedora 18 diese Ideen umgesetzt. Mittlerweile verwenden viele, wenn auch nicht alle auf systemd basierende Distributionen die in [Tabelle 24.2](#) aufgezählten Konfigurationsdateien. Details zu einigen dieser Dateien und zu den dazugehörigen Konfigurationswerkzeugen finden Sie in [Kapitel 18](#), »Basiskonfiguration«.

systemd beinhaltet mit dem »Journal« ein Syslog-kompatibles Logging-System. Die Logging-Dateien werden dabei in einem binären Format gespeichert und sind daher gegen nachträgliche Veränderungen abgesichert (siehe [Abschnitt 18.10](#), »Logging (Journal)«). **Journal**

| Funktion     | bisheriger Ort           | neuer Ort             | Konfiguration |
|--------------|--------------------------|-----------------------|---------------|
| Hostname     | /etc/sysconfig/network   | /etc/hostname         | hostnamectl   |
| Server-ID    | /var/lib/dbus/machine-id | /etc/machine-id       | –             |
| Sprache      | /etc/sysconfig/i18n      | /etc/locale.conf      | localectl     |
| Tastatur     | /etc/sysconfig/keyboard  | /etc/vconsole.conf    | localectl     |
| Zeitzone     | /etc/sysconfig/clock     | /etc/localtime        | timedatectl   |
| Module       | /etc/modules             | /etc/modules-load.d/* | –             |
| Distribution | /etc/xxx-release         | /etc/os-release       | –             |

**Tabelle 24.2** Geänderte Konfigurationsdateien

**Timer** Die systemd-Timer-Funktion kann sich ähnlich wie Cron um die regelmäßige Ausführung von Prozessen kümmern. Diese momentan noch spärlich genutzte Cron-Variante ist in [Abschnitt 12.7](#), »Prozesse automatisch starten (systemd-Timer)«, näher beschrieben.

## Kompatibilität

systemd ist zum herkömmlichen Init-V-System kompatibel. Init-Scripts, die sich im Verzeichnis `/etc/init.d/` befinden, werden also wie bisher gestartet bzw. beendet. Allerdings gilt: Je kleiner die Anzahl herkömmlicher Init-V-Scripts in `/etc/init.d` ist, desto besser wurde die betreffende Distribution für systemd optimiert. In dieser Hinsicht glänzen z. B. CentOS, Fedora, openSUSE und RHEL, wo das `init.d`-Verzeichnis nahezu leer ist.

Weniger gut sah es zuletzt bei Debian und Ubuntu aus, die zwar prinzipiell den systemd-Wechsel vollzogen haben, aber noch stärker auf die Init-V-Kompatibilität von systemd angewiesen waren. Freilich sind auch dort nicht alle `init.d`-Einträge tatsächlich aktiv. Teilweise gibt es sowohl systemd- als auch Init-V-Konfigurationsdateien; in diesem Fall gilt die systemd-Konfiguration.

Dienste, die systemd über Init-V-Scripts startet, kennzeichnet `systemctl` mit dem Label LSB (Linux Standard Base). Sie können die betreffenden Dienste daher rasch mit `systemctl | grep LSB` ermitteln.

### systemd-sysv-generator

Hinter den Kulissen richtet systemd zu Beginn des Bootprozesses sowie durch `systemctl daemon-reload` für alle Init-V-Dienste Wrapper-Service-Dateien ein. Verantwortlich dafür ist das Kommando `systemd-sysv-generator`. Die Service-Dateien landen in Unterverzeichnissen von `/run/systemd`. Die folgende Ausgabe ist auf einem Ubuntu-Server entstanden (Version 16.04):

```
root# find /run/systemd -name '*.service'
/run/systemd/generator.late/ondemand.service
/run/systemd/generator.late/apport.service
/run/systemd/generator.late/postfix.service
/run/systemd/generator.late/mdadm.service
/run/systemd/generator.late/apache2.service
/run/systemd/generator.late/apache-htcacheclean.service
/run/systemd/generator.late/irqbalance.service
/run/systemd/generator.late/grub-common.service
...
```

Das von beinahe allen Distributionen unterstützte Kommando `service` funktioniert auch unter `systemd`. Wenn Sie beispielsweise `service httpd stop` ausführen, erkennt `service`, dass der Webserver durch `systemd` gesteuert wird, und führt `systemctl stop httpd.service` aus.

**service-  
Kommando**

## Dokumentation

`systemd` ist ausgezeichnet dokumentiert: Alle Funktionen, Strategien und Vorzüge von `systemd` sind beinahe schon überkomplett in diversen Manual-Seiten (insbesondere `man systemd`) sowie auf der Webseite des `systemd`-Entwicklers Lennart Poettering beschrieben:

<https://www.freedesktop.org/wiki/Software/systemd>  
<http://Opointer.de/blog/projects/systemd-docs.html>

Auch die Website *heise open* hat sich ausführlich mit `systemd` befasst:

<https://heise.de/-1563259> und <https://heise.de/-1563461>

Wenn Sie kurze und prägnante Texte bevorzugen, sind die beiden folgenden Seiten hilfreich. Sie fassen die wichtigsten Änderungen im Vergleich zu `Init-V` zusammen und beantworten einige FAQs:

[https://fedoraproject.org/wiki/SysVinit\\_to\\_Systemd\\_Cheatsheet](https://fedoraproject.org/wiki/SysVinit_to_Systemd_Cheatsheet)  
<https://www.freedesktop.org/wiki/Software/systemd/FrequentlyAskedQuestions>

## 24.2 Das Init-V-System

Das traditionelle `Init-V`-System wird nur noch bei wenigen bzw. älteren Distributionen dazu verwendet, die Systeminitialisierung durchzuführen und Netzwerkdienste zu starten.

### Init-V-Grundlagen bleiben relevant!

Auch wenn die meisten aktuellen Distributionen systemd verwenden, bieten Init-V-Scripts oft die einfachste Möglichkeit, einen eigenen Systemdienst plattformübergreifend zu starten. Ein Beispiel für ein derartiges Script folgt in [Abschnitt 24.3](#), »Eigene Init-Scripts bzw. Init-Konfigurationsdateien«.

**Ablauf** Die folgenden Punkte geben einen kurzen Überblick über einen Systemstart durch das Init-V-System:

- ▶ GRUB lädt und startet den Kernel.
- ▶ Der Kernel startet das Programm `/sbin/init`.
- ▶ `init` wertet die Konfigurationsdatei `/etc/inittab` aus.
- ▶ `init` führt ein Script zur Systeminitialisierung aus.
- ▶ `init` führt das Script `/etc/rc.d/rc` oder `/etc/init.d/rc` aus. Das Script `rc` variiert von Distribution zu Distribution erheblich. Es ist für den Start der Script-Dateien verantwortlich, die sich im Verzeichnis `/etc/rc<n>.d` oder `/etc/init.d/rc<n>.d` befinden. (`n` ist der Runlevel. Details dazu folgen gleich.)
- ▶ Die Script-Dateien aus `/etc/rc<n>.d` bzw. `/etc/init.d/rc<n>.d` starten verschiedene Systemdienste, insbesondere für die Netzwerkfunktionen.

### Runlevel

Für das Verständnis der System-V-Mechanismen ist der Begriff des Runlevels von zentraler Bedeutung. Die zulässigen Runlevel werden in der Datei `/etc/inittab` definiert und beschreiben verschiedene Zustände, die das Betriebssystem einnehmen kann. In der Vergangenheit gab es je nach Distribution unterschiedliche Ziffern bzw. Buchstaben zur Bezeichnung der Runlevel. Die folgende Aufzählung galt bis vor wenigen Jahren noch für Debian- und Raspbian-Systeme. Der Default-Runlevel war 2 und wurde durch die `initdefault`-Zeile in `/etc/inittab` bestimmt.

- ▶ Runlevel S: Initialisierung des Rechners unmittelbar nach dem Start
- ▶ Runlevel 0: Shutdown mit Halt
- ▶ Runlevel 1: Single-User mit Netzwerk
- ▶ Runlevel 2–5: gleichwertig, Multi-User-Betrieb mit Netzwerk und Grafiksystem
- ▶ Runlevel 6: Shutdown mit Reboot

**Runlevel wechseln** `root` kann den Runlevel im laufenden Betrieb durch das Kommando `init x` verändern. `x` ist dabei eine Runlevel-Ziffer oder ein Runlevel-Buchstabe. Beispielsweise ist es für manche Wartungsarbeiten sinnvoll, in den Single-User-Modus zu wechseln. Auch

shutdown, halt, reboot bzw. `Strg`+`Alt`+`Entf` in einer Textkonsole ändern den Runlevel und führen auf diese Weise zu einem Rechnerneustart.

## Init-V-Scripts

Das Init-V-System führt zur Durchführung von Initialisierungsarbeiten sowie zum Start oder Stopp die sogenannten Init-V-Scripts aus. Diese befinden sich je nach Distribution im Verzeichnis `/etc/init.d` oder in `/etc/rc.d/init.d`. Um eine höhere Kompatibilität zwischen den Distributionen zu erreichen, stellen meist Links sicher, dass beide Pfade gültig sind.

Die Grundidee des Init-V-Systems besteht darin, dass zur Aktivierung eines Runlevels alle dafür erforderlichen Scripts mit dem Parameter `start` ausgeführt werden. Beim Herunterfahren des Rechners bzw. beim Wechsel in einen Runlevel mit weniger Funktionen werden die entsprechenden Scripts neuerlich ausgeführt, jetzt aber mit dem Parameter `stop`.

Die meisten Init-V-Scripts akzeptieren die folgenden Parameter:

Init-V-Script-  
Parameter

- ▶ `start` startet die betreffende Funktion.
- ▶ `stop` beendet die Funktion.
- ▶ `status` zeigt eine kurze Information an, ob die Funktion aktiv ist oder nicht.
- ▶ `reload` bietet sich dann an, wenn geänderte Konfigurationsdateien neu eingelesen werden sollen, ohne den Dämon dabei ganz zu stoppen. Allerdings sehen nicht alle Dienste diese Möglichkeit auch vor. Sie müssen gegebenenfalls neu gestartet werden.
- ▶ `restart` bewirkt, dass der Dämon vollkommen gestoppt und anschließend neu gestartet wird. Eventuell vorhandene Verbindungen zu Clients gehen dabei verloren. Bei Datenbank-Servern verlieren Sie auch den Cache-Inhalt, was dazu führt, dass die Datenbank eine Weile Abfragen deutlich langsamer ausführt.

Init-V-Scripts können nicht nur durch das Init-V-System, sondern auch manuell ausgeführt werden – beispielsweise um einen Dienst neu zu starten:

Dienste manuell  
starten/stoppen

```
root# /etc/init.d/sshd restart
```

Anstelle des `systemd`-Kommandos `systemctl` steht bei Distributionen auf Init-V-Basis oft das Kommando `service` zur Verfügung. Es hat den Vorteil, dass es auch auf den meisten Distributionen mit `systemd` funktioniert und insofern ein gemeinsamer Nenner vieler Linux-Distributionen ist:

```
root# service sshd restart
```

## Runlevel-Scripts

Welche Dienste in welchem Runlevel gestartet bzw. beendet werden sollen, wird durch Links in Runlevel-Verzeichnissen ausgedrückt. Soll beispielsweise der SSH-Server im Runlevel 2 gestartet werden, dann enthält das Verzeichnis `/etc/rc2.d` einen Link der Form `S<nn>sshd`, der auf das Script `/etc/init.d/sshd` zeigt. Dabei steht `S` für *Start*. Bei der Auswertung aller `S`-Links übergibt das Init-V-System den Parameter `start` an das Script. Die Nummer `nn` bestimmt die Reihenfolge, in der die Scripts ausgeführt werden.

Um den SSH-Server beim Herunterfahren zu stoppen, enthält `/etc/rc6.d` einen analogen Link, der jetzt aber den Namen `K<nn>sshd` hat. In diesem Fall steht `K` für *Kill*, als Parameter wird `stop` übergeben.

Der Vorteil dieses Systems besteht darin, dass es sehr einfach ist, neue Systemprozesse in den Init-V-Prozess einzubauen: Es müssen lediglich Links in den richtigen Verzeichnissen eingerichtet werden.

**insserv** Je nach Distribution gibt es verschiedene Kommandos, die beim Einrichten der Links helfen. Das folgende Beispiel zeigt den Einsatz von `insserv` unter Debian und SUSE:

```
root# insserv samba      (Start- und Stopp-Links einrichten)
root# insserv -r samba   (Start- und Stopp-Links entfernen)
```

**chkconfig** Unter RHEL hilft das Kommando `chkconfig` bei der Verwaltung der Links auf Init-V-Scripts. Mit der Option `--list` gibt das Kommando eine Übersicht über alle Scripts und zeigt an, in welchem Runlevel sie gestartet werden. Unter RHEL 7 bzw. CentOS 7 ist die Ergebnisliste kurz, weil die meisten Dienste nun ja durch `systemd` verwaltet werden. `chkconfig` ist nur für die Dienste gedacht, für die es noch herkömmliche Init-V-Scripts gibt. Das folgende Ergebnis ist unter einer CentOS-7-Installation innerhalb von VirtualBox entstanden:

```
root# chkconfig --list
netconsole    0:Aus  1:Aus  2:Aus  3:Aus  4:Aus  5:Aus  6:Aus
network       0:Aus  1:Aus  2:Ein  3:Ein  4:Ein  5:Ein  6:Aus
vboxadd       0:Aus  1:Aus  2:Ein  3:Ein  4:Ein  5:Ein  6:Aus
vboxadd-service 0:Aus  1:Aus  2:Ein  3:Ein  4:Ein  5:Ein  6:Aus
vboxadd-x11   0:Aus  1:Aus  2:Aus  3:Ein  4:Aus  5:Ein  6:Aus
```

Mit `--del` kann der Start eines Runlevel-Scripts generell verhindert werden:

```
root# chkconfig --del vboxadd-x11
```

`chkconfig --add` fügt in allen vorgesehenen Runleveln Start- und Stopp-Links für einen neuen Service ein. Die Option `--add` funktioniert allerdings nur, wenn die Init-V-Script-Datei Informationen darüber enthält, in welchem Runlevel das Script standardmäßig gestartet werden soll.

Bei vielen Scripts fehlen diese Informationen. Damit ein derartiges Script in Zukunft automatisch gestartet wird, müssen Sie `chkconfig --level n name on/off` verwenden. Im folgenden Beispiel soll ein eigenes Script zum Start eines VNC-Servers in den Run-leveln 3 und 5 gestartet werden. `chkconfig --list` zeigt das Ergebnis an:

```
root# chkconfig --level 35 myvnc on
root# chkconfig --list myvnc
myvnc    0:Aus  1:Aus  2:Aus  3:Ein  4:Aus  5:Ein  6:Aus
```

## 24.3 Eigene Init-Scripts bzw. Init-Konfigurationsdateien

Bei Netzwerkdiensten, die in fertigen Paketen zur Verfügung stehen, müssen Sie sich nicht um Init-Scripts oder -Konfigurationsdateien kümmern: Die erforderlichen Dateien werden von allen Distributionen gleich mitgeliefert. Unter CentOS, Fedora, RHEL und SUSE müssen Sie den Dienst nur noch mit `systemctl start` und `systemctl enable` starten und dauerhaft aktivieren – unter Debian, Raspbian und Ubuntu entfällt selbst dieser Schritt.

Was aber müssen Sie tun, um einen eigenen Dienst im Init-System Ihrer Distribution zu verankern? Wenn Sie nur mit Distributionen zu tun haben, die `systemd` verwenden, ist es empfehlenswert, eine neue `systemd`-Konfigurationsdatei zu erstellen und diese durch `systemctl` zu aktivieren.

Wenn Sie hingegen nach einer Lösung suchen, die auch bei alten Distributionen noch funktioniert, ist ein herkömmliches Init-V-Script die bessere Wahl. Dank der Init-V-Kompatibilität werden korrekt eingerichtete Init-V-Scripts auch von Distributionen mit `systemd` gestartet. Dieser Abschnitt gibt für beide Varianten ein Beispiel.

### Eigene `systemd`-Konfigurationsdatei

Um einen `systemd`-Dienst einzurichten, müssen Sie eine `*.service`-Datei im Verzeichnis `/etc/systemd/system` erstellen. Abermals ist es zweckmäßig, wenn Sie sich dabei an einer Service-Datei eines anderen, vergleichbaren Diensts orientieren. Die vorgegebenen Service-Dateien befinden sich je nach Distribution in den Verzeichnissen `/lib/systemd/system` oder `/usr/lib/systemd/system`.

Eine minimale Service-Datei sieht wie folgt aus:

Syntaxregeln

```
[Unit]
Description=Foo
[Service]
ExecStart=/usr/sbin/foo-daemon
[Install]
WantedBy=multi-user.target
```

Das bedeutet, dass der Dienst »Foo« in Form des Hintergrundprozesses `foo-daemon` gestartet werden soll. Ein explizites Stop-Kommando fehlt; deswegen wird `systemd` zum Beenden zuerst das Signal `SIGTERM` und, wenn das nicht hilft, auch `SIGKILL` an den Prozess senden.

An `ExecStart` muss ein Kommandoname mit vollständigem Pfad übergeben werden. In der gleichen Weise können Sie mit `ExecStop` ein zweites Kommando angeben, das ausgeführt wird, wenn der Hintergrundprozess beendet werden soll.

#### Unterschiedliche Service-Typen

Normalerweise ist es nicht zulässig, mit `ExecStart` oder `ExecStop` mehrere Kommandos anzugeben. Wenn Sie das möchten, müssen Sie im `[Service]`-Block `Type=oneshot` angeben. Nun sind beliebig viele `ExecStart`- bzw. `ExecStop`-Anweisungen erlaubt, die der Reihe nach ausgeführt werden.

Wenn Sie bei einer derartigen Konfiguration explizit zwischen zwei Zuständen hin- und herschalten möchten (`start/stop`), dann müssen Sie wie im folgenden Beispiel auch das Schlüsselwort `RemainAfterExit` verwenden. Die Option bewirkt, dass `systemd` sich den gerade aktivierten Zustand merkt. Ohne diese Option glaubt `systemd` nach `systemctl start name`, dass die Aktion mit dem Ende des letzten `StartExec`-Kommando beendet ist, und setzt den Status sofort wieder auf `stop`. Ein explizites Ausführen von `systemctl stop` bleibt dann wirkungslos.

Für das obige Beispiel galt implizit `Type=simple`. `systemd` nimmt in diesem Fall an, dass das mit `ExecStart` angegebene Kommando der zu startende Dienst ist. Sobald dieses Programm endet, betrachtet `systemd` den Dienst als regulär beendet.

Der dritte wichtige Typ für Services ist `Type=fork`: Diese Variante wählen Sie dann, wenn das mit `ExecStart` angegebene Kommando ein anderes Hintergrundprogramm startet (also einen Dämon). In diesem Fall sollten Sie mit `PIDFile` eine Datei angeben, in der die Prozessnummer gespeichert wird. Das gibt `systemd` die Möglichkeit, den Hintergrundprozess zu überwachen bzw. bei Bedarf zu stoppen.

**Links** Weitere Informationen zum Verfassen eigener `systemd`-Service-Dateien geben `man systemd.service` und `man systemd.exec` sowie die folgenden Seiten:

<https://fedoraproject.org/wiki/Packaging:Systemd>

<https://wiki.archlinux.org/index.php/Systemd>

<http://Opointer.de/blog/projects/systemd-for-admins-3.html>

**Beispiel** Das folgende Beispiel zeigt den Start von Masquerading-Funktionen. Diese sind notwendig, um den Rechner als Internet-Gateway einzurichten (siehe [Abschnitt 27.3](#), »Masquerading (NAT)«). Zur einfacheren Administration gibt es in diesem Fall gleich zwei Konfigurationsdateien:



- ▶ `/etc/sysconfig/masquerading` enthält die Definition der Variablen ADSL, die den Namen der Netzwerkschnittstelle angibt.
- ▶ `/etc/systemd/system/masquerading.service` ist die eigentliche Service-Datei für `systemd`.

Die Datei zur Festlegung der ADSL-Schnittstelle kann z. B. so aussehen:

```
# Datei /etc/sysconfig/masquerading
# Masquerading-Schnittstelle des Internet-Gateways
ADSL=enp4s0
```

Die Service-Datei verwendet das Schlüsselwort `EnvironmentFile`, um diese Konfigurationsdatei einzulesen. Alle dort definierten Variablen können nun in der Service-Datei in der Form `$varname` angesprochen werden.

```
[Unit]
Description=Masquerading
After=syslog.target network.target

[Service]
Type=oneshot
RemainAfterExit=true
EnvironmentFile=/etc/sysconfig/masquerading

ExecStart=/sbin/iptables -t nat -A POSTROUTING -o $ADSL -j MASQUERADE
ExecStart=/bin/sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"

ExecStop=/sbin/iptables -t nat -D POSTROUTING -o $ADSL -j MASQUERADE
ExecStop=/bin/sh -c "echo 0 > /proc/sys/net/ipv4/ip_forward"

[Install]
WantedBy=multi-user.target
```

Aus Effizienzgründen hält `systemd` den Inhalt der wichtigsten Konfigurationsdateien in einem Zwischenspeicher (Cache). Nach Konfigurationsänderungen müssen Sie `systemd` deswegen auffordern, die Konfigurationsdateien neu einzulesen:

**Dienst starten  
und dauerhaft  
aktivieren**

```
root# systemctl daemon-reload
```

Um das Masquerading unmittelbar und in Zukunft jedes Mal zu starten, wenn das Multi-User-Target erreicht wird, sind die folgenden Kommandos erforderlich:

```
root# systemctl start masquerading
root# systemctl enable masquerading
```

## Eigene Init-V-Scripts

**Syntax** Zum Verfassen eines eigenen Init-V-Scripts verwenden Sie am besten ein vorhandenes Script als Vorlage und modifizieren dieses. Vorweg einige Grundregeln: Da es sich um ein Script handelt, muss die Datei mit `#!/bin/sh` oder `#!/bin/bash` beginnen und ausführbar sein (`chmod a+x`). Einige Kommentarzeilen, die mit `BEGIN INIT INFO` eingeleitet werden und mit `END INIT INFO` enden, geben Auskunft darüber, welche Aufgabe das Script erfüllt, von welchen anderen Diensten es abhängig ist und in welchen Run-leveln es standardmäßig aktiviert werden soll.

Nach eventuell noch erforderlichen Initialisierungsarbeiten folgt dann die Auswertung des Parameters `$1`. Sie sollten zumindest die Fälle `start`, `stop` und `restart` berücksichtigen, möglichst auch `reload` und `status`.

**Beispiel** Das folgende Script greift nochmals das Masquerading-Beispiel aus [Abschnitt 27.3](#) auf. Das Script kümmert sich um den Start bzw. Stopp dieser Netzwerkfunktion.

```
#!/bin/sh

### BEGIN INIT INFO
# Provides:          masquerading
# Required-Start:    $network $local_fs $remote_fs
# Required-Stop:     $network $local_fs $remote_fs
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: start masquerading
### END INIT INFO

DESC="masquerading"      # Bezeichnung des Scripts
ADSL=eth1                 # Schnittstelle, über die der Internetzugang erfolgt
. /lib/lsb/init-functions # Grundfunktionen lesen
IPT=$(which iptables)    # iptables-Kommando suchen
if [ -z $IPT ]; then
    [ -x /sbin/iptables ]    && IPT=/sbin/iptables
    [ -x /usr/sbin/iptables ] && IPT=/usr/sbin/iptables
fi
[ -z $IPT ] && (echo "iptables cannot be found!"; exit 0)

# Funktionen für start, stop und restart
case "$1" in
    start)
        log_begin_msg "Starting masquerading ..."
        ERROR=0
        $IPT -t nat -A POSTROUTING -o $ADSL -j MASQUERADE
        echo 1 > /proc/sys/net/ipv4/ip_forward
        log_end_msg $ERROR
        ;;
```

```

stop)
    log_begin_msg "Stopping masquerading ..."
    ERROR=0
    echo 0 > /proc/sys/net/ipv4/ip_forward
    $IPT -t nat -D POSTROUTING -o $ADSL -j MASQUERADE
    log_end_msg $ERROR
    ;;

restart)
    $0 stop
    $0 start
    ;;

*)
    log_success_msg "Usage: masquerading {start|stop|restart}"
    exit 1
    ;;
esac
exit 0

```

Mit service starten Sie das Script erstmalig:

```
root# service masquerading start
```

**Init-Script starten  
und dauerhaft  
aktivieren**

Damit das Script in Zukunft beim Rechnerstart ausgeführt wird, führen Sie die folgenden Kommandos aus:

```

root# inserv masquerading      (Debian, openSUSE, Ubuntu)
root# chkconfig --on masquerading (CentOS, Fedora, RHEL)

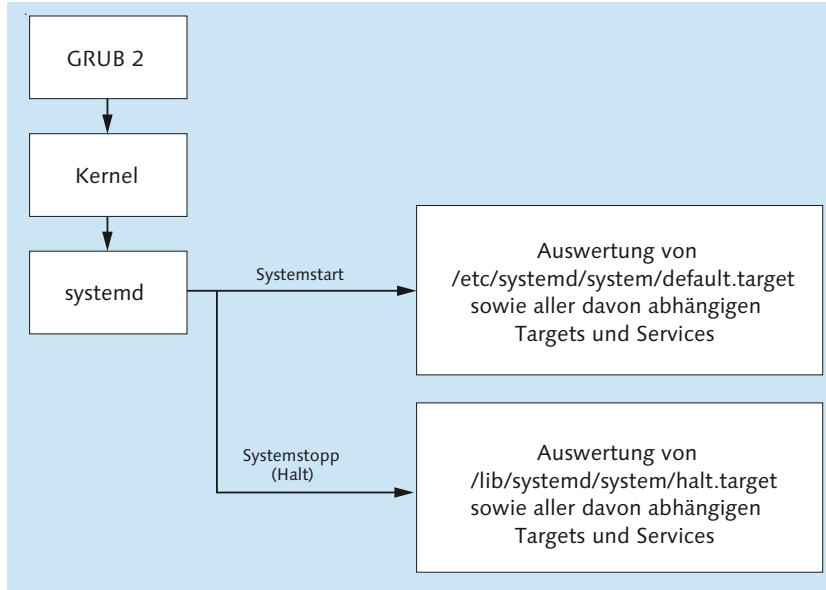
```

## 24.4 Systemstart bei CentOS, Fedora und RHEL

Abbildung 24.1 fasst zusammen, wie Fedora, CentOS 7 bzw. RHEL 7 in das Default-Target startet bzw. wie es wieder heruntergefahren wird. Tabelle 24.3 gibt einen Überblick über die Konfigurationsdateien.

Abbildung 24.1 gilt auch für Debian, openSUSE und Ubuntu. Allerdings befinden sich bei openSUSE die systemd-Dateien im Verzeichnis `/usr/lib/systemd`. Weitere Debian-, openSUSE- und Ubuntu-spezifische Besonderheiten beschreibe ich in den folgenden Abschnitten.

Fedora, CentOS 7 und RHEL 7 haben den Umstieg auf systemd sehr konsequent vollzogen: Das Verzeichnis `/etc/init.d/` ist fast leer, und das ehemals für die System- und Hardware-Initialisierung verantwortliche Script `/etc/rc.d/rcsysconfig` gibt es gar nicht mehr.



**Abbildung 24.1** CentOS 7, Debian, Fedora, openSUSE, RHEL 7 und Ubuntu starten und beenden

| Funktion                      | Konfigurationsdateien               |
|-------------------------------|-------------------------------------|
| Binary                        | /usr/lib/systemd/systemd            |
| systemd                       | /etc/systemd/*, /lib/systemd/*      |
| systemd-Default-Target (Link) | /etc/systemd/system/default.target  |
| Systeminitialisierung         | /lib/systemd/system/sysinit.target* |
| Herkömmliche Init-V-Scripts   | /etc/rc.d/init.d/*                  |
| Init-V-Runlevel-Links         | /etc/rc.d/rc<n>.d/*                 |
| Konfigurationsdateien         | /etc/sysconfig/*                    |

**Tabelle 24.3** Konfiguration des Systemstarts bei CentOS 7, Fedora und RHEL 7

**Start des Grafiksystems** Für den Start des Grafiksystems ist die Konfigurationsdatei `/lib/systemd/system/gdm.service` verantwortlich.

**Logging** Ein Protokoll aller von `systemd` gestarteten Dienste finden Sie in der Logging-Datei `/var/log/boot.log`. `systemd`-Nachrichten werden parallel auch mit dem neuen Logging-Dienst *Journal* aufgezeichnet und können mit dem Kommando `journalctl` gelesen werden.

Neben `systemctl` können Sie `systemd`-Dienste auch mit der grafischen Benutzeroberfläche `system-config-services` aus dem gleichnamigen Paket starten, stoppen oder neu starten. In der Vergangenheit bot das Programm weitere Funktionen, etwa die Aktivierung oder Deaktivierung von Diensten. Diese Funktionen sind aber dem `systemd`-Umstieg zum Opfer gefallen.

Konfiguration

Die wenigen herkömmlichen Init-V-Skripts können mit dem Fedora- bzw. Red-Hat-spezifischen Kommando `chkconfig` verwaltet werden. Das Werkzeug hilft bei der Zuordnung der Links zu den Init-V-Skripts.

Die zur individuellen Anpassung des Systemstarts ehemals vorgesehene Datei `/etc/rc.d/rc.local` existiert standardmäßig nicht mehr. Wenn Sie am Ende des Systemstarts eigene Skripts ausführen möchten, können Sie dazu aber weiterhin `/etc/rc.d/rc.local` verwenden: Dazu erzeugen Sie diese Datei und stellen mit `chmod ug+x` sicher, dass die Datei auch ausführbar ist.

Was den eigentlichen Systemstart betrifft, verhalten sich Fedora und CentOS/RHEL 7 nahezu identisch. Davon abgesehen gibt es aber durchaus Unterschiede: Fedora verwendet stets die allerneueste `systemd`-Version und nutzt auch dessen Zusatzfunktionen wie Journal und Timer.

Fedora versus CentOS 7 und RHEL 7

Die Enterprise-Distributionen CentOS und RHEL agieren da deutlich konservativer: Dort kümmert sich noch der traditionelle Syslog-Dämon `rsyslogd` um die Protokolldateien, die Timer-Funktionen stehen noch gar nicht zur Verfügung etc.

## 24.5 Systemstart bei Debian, Raspbian und Ubuntu

Debian verwendet seit Version 8 `systemd` zum Systemstart, Ubuntu seit Version 15.04 (siehe [Abbildung 24.1](#)). Allerdings spielt die Init-V-Kompatibilität unter Debian bzw. unter Ubuntu bis heute eine etwas größere Rolle als z. B. unter Fedora. Das folgende Listing ist auf einem Ubuntu-Root-Server (Version 16.04) entstanden und zeigt, wie viele Dienste durch herkömmliche Init-V-Skripts gestartet wurden:

```
root# systemctl | grep LSB
UNIT                                ACTIVE SUB    DESCRIPTION
apache2.service                    active running  LSB: Apache2 web server
apparmor.service                   active exited  LSB: AppArmor initialization
apport.service                      active exited  LSB: automatic crash report ...
glances.service                    active exited  LSB: Starts and daemonize ...
grub-common.service                active exited  LSB: Record successful boot ...
hddtemp.service                    active exited  LSB: disk temperature ...
irqbalance.service                 active exited  LSB: daemon to balance ...
mdadm.service                       active running  LSB: MD monitoring daemon
```

```

ondemand.service      active exited  LSB: Set the CPU Frequency ...
postfix.service       active running  LSB: Postfix Mail Transport Agent
spamass-milter.service active running  LSB: milter for spamassassin

```

| Funktion                      | Konfigurationsdateien   |
|-------------------------------|---|
| Binary                        | /sbin/init (Link auf /lib/systemd/systemd)                            |
| systemd                       | /etc/systemd/*, /lib/systemd/*  |
| systemd-Default-Target (Link) | /etc/systemd/system/default.target                                    |
| Systeminitialisierung         | /lib/systemd/system/sysinit.target*,<br>/etc/init.d/rcS, /etc/rcS.d/* |
| Herkömmliche Init-Scripts     | /etc/init.d/*   |
| Runlevel-Links                | /etc/rc<n>.d/rc<n>.d/*  |
| Konfigurationsdateien         | /etc/default/*  |

**Tabelle 24.4** Konfiguration des Systemstarts bei Debian und Ubuntu

#### Automatische Aktivierung neuer Dienste

Debian und Ubuntu unterscheiden sich in noch einem Punkt fundamental von der Welt von Fedora und RHEL: Neu installierte (Netzwerk-)Dienste werden sofort gestartet. Damit entfallen die sonst üblichen Kommandos `systemctl start` und `systemctl enable`. Das ist einerseits bequem, andererseits aber auch riskant. Achten Sie darauf, dass die Defaultkonfiguration Ihren Vorstellungen entspricht!

#### Grafiksystem

Bei Debian ist für den Start des Grafiksystems die Konfigurationsdatei `/lib/systemd/system/gdm.service` verantwortlich. Beim Start wird die Datei `/etc/X11/default-display-manager` ausgewertet, die den Dateinamen des gewünschten Display-Managers enthält.

Auch bei aktuellen Ubuntu-Versionen startet `systemd` das Grafiksystem. Je nach Ubuntu-Variante ist dafür die Datei `gdm.service` oder `lightdm.conf` verantwortlich.

#### Individuelle Anpassung des Init-Prozesses

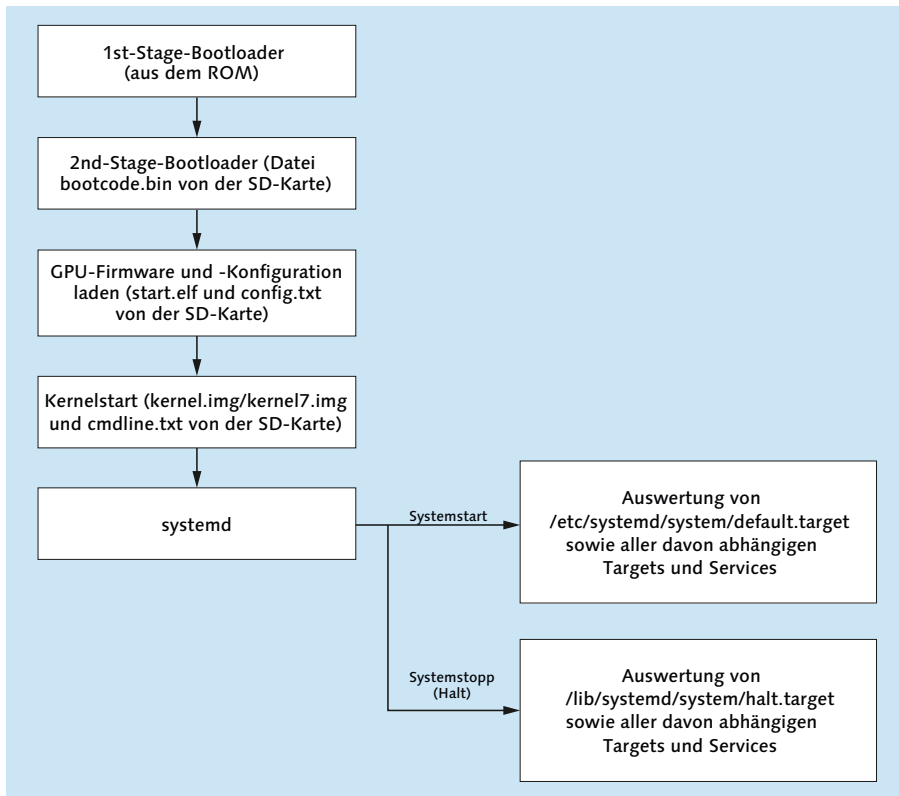
Das Script `/etc/rc.local` wird ausgeführt, nachdem das `multi-user-` oder das `graphical.target` erreicht wurde. Das Script eignet sich damit gut, um einmalig eigene Konfigurationsarbeiten durchzuführen.

## Raspbian

Raspbian basiert seit Oktober 2015 auf Debian 8 »Jessie« und somit auch auf `systemd`. Aktuelle Raspbian-Installationen verwenden daher ebenfalls `systemd` als Init-System. Wie Debian 8 verlässt sich auch Raspbian Jessie auf die Init-V-Kompatibilität von `systemd`. Für den Start nahezu aller Netzwerkprozesse sind unverändert Init-V-Scripts zuständig, die nun aber nicht durch das Init-V-System, sondern durch `systemd` aus-

geführt werden. Somit gelten die in [Tabelle 24.4](#) zusammengestellten Konfigurationsverzeichnisse auch für Raspbian.

Große Unterschiede im Vergleich zu Debian und Ubuntu gibt es aber beim Boot-Prozess. Für diesen ist nicht GRUB zuständig. Vielmehr ist der Boot-Prozess durch den Grafikeil (die GPU) des Chips BCM2835/-36/-37 vorgegeben. Die GPU lädt die Bootloader-Dateien von der ersten Partition der SD-Karte des Raspberry Pi (siehe [Abbildung 24.2](#) sowie [Abschnitt 8.5](#), »Interna und Backups«).



**Abbildung 24.2** Raspbian starten und beenden

## 24.6 Systemstart bei SUSE/openSUSE

Aktuelle SUSE- bzw. openSUSE-Distributionen verwenden ebenfalls systemd für den Startprozess (siehe [Abbildung 24.1](#)). Beachten Sie aber, dass sich die systemd-Dateien bei openSUSE im Verzeichnis `/usr/lib/systemd` befinden und dass es auch sonst Unterschiede zur systemd-Konfiguration von Fedora gibt. [Tabelle 24.5](#) zählt die für den Systemstart relevanten Konfigurationsdateien auf.

| Funktion                      | Konfigurationsdateien                   |
|-------------------------------|---|
| systemd                       | /etc/systemd/*, /usr/lib/systemd/*      |
| systemd-Default-Target (Link) | /etc/systemd/system/default.target      |
| Systeminitialisierung         | /usr/lib/systemd/system/sysinit.target* |
| Herkömmliche Init-V-Scripts   | /etc/init.d/*                           |
| Runlevel-Links                | /etc/init.d/rc<n>.d/*                   |
| Konfigurationsdateien         | /etc/sysconfig/*                        |

Tabelle 24.5 Konfiguration des SUSE-Systemstarts

- Start des Grafiksystems** Für den Start des Grafiksystems X ist die systemd-Datei `/usr/lib/systemd/system/display-manager.service` verantwortlich. Welcher Display-Manager tatsächlich gestartet wird (z. B. `sddm`, also der *Simple Desktop Display Manager*), bestimmt die Variable `DISPLAYMANAGER`, die in `/etc/sysconfig/displaymanager` eingestellt wird.
- Konfigurationswerkzeuge** Außer den aus anderen Distributionen bekannten Kommandos `systemctl` und `service` können unter SUSE Systemdienste auch mit `rc<name> start|stop` gestartet und beendet werden. Bei der Administration herkömmlicher Init-V-Scripts hilft das Kommando `inssrv`. Wohl aufgrund von Kompatibilitätsüberlegungen gibt es auch unter SUSE das Kommando `chkconfig`. Die Optionen `-add`, `-del` und `-list` funktionieren wie bei Red Hat, bei anderen Optionen gibt es aber Abweichungen. Intern greift `chkconfig` auf `insserv` zurück.
- boot.local** In der Datei `/etc/rc.d/boot.local` können Sie lokale Anpassungen durchführen. Das Script sollte ausschließlich Kommandos enthalten, die nur ein einziges Mal beim Systemstart ausgeführt werden sollen. Ein typisches Beispiel sind `modprobe`-Anweisungen, um ein ganz bestimmtes Kernelmodul zu laden. `boot.local` wird vor den `rc`-Scripts ausgeführt.

## 24.7 Internet Service Daemon

Programme, die Netzwerk- oder Internetdienste zur Verfügung stellen, können in zwei Gruppen eingeteilt werden:

- Die eine Gruppe besteht aus Programmen, die als sogenannte Dämonen ständig laufen. Nahezu alle in diesem Buch vorgestellten Netzwerkdienste – der Webserver Apache, der Datei-Server Samba, der SSH-Server etc. – zählen zu dieser Gruppe. Diese Programme werden durch das Init-System gestartet und überwachen dann einen IP-Port. Sobald ein IP-Paket eintrifft, das an diesen IP-Port adressiert ist, wird es ausgewertet und beantwortet.



- Die zweite Gruppe besteht aus selten benötigten Programmen, die erst bei Bedarf gestartet werden. Anstatt auch diese Programme alle zu starten, wird ein sogenannter *Internet Service Daemon* ausgeführt. Dieses Programm überwacht mehrere IP-Ports gleichzeitig und aktiviert erst bei Bedarf den entsprechenden Server-Dienst. Der Internet Service Daemon selbst wird übrigens ganz normal durch das Init-System gestartet.

Als *Internet Service Daemon* war früher *inetd* weit verbreitet. Heute gilt dieses Programm aber als veraltet. Deswegen kommt je nach Distribution *openbsd-inetd* oder *xinetd* zum Einsatz. Standardmäßig ist zumeist keines dieser Pakete installiert: Eine Installation ist nur erforderlich, wenn ein anderes Programm die *inetd*-Funktionalität verlangt und im Paket eine entsprechende Abhängigkeit formuliert ist.

Unabhängig davon, welcher Internet Service Daemon bei Ihrer Distribution zum Einsatz kommt, stellt die Datei `/etc/services` die Zuordnung zwischen den Namen verschiedener Internetdienste (z. B. `ftp`, `telnet` etc.) und deren Protokolltypen und Port-Nummern her. Beispielsweise verwenden E-Mail-Server (MTAs) den Port 25 und die Protokolle `tcp` und `udp`. Die folgenden Zeilen zeigen einen Ausschnitt aus dieser Datei:

`/etc/services`

```
# /etc/services (auszugsweise)
# name      port/proto  alias  comment
ftp-data    20/tcp      # File Transfer [Default Data]
ftp-data    20/udp      # File Transfer [Default Data]
ftp         21/tcp      # File Transfer [Control]
ssh         22/tcp      # SSH Remote Login Protocol
ssh         22/udp      # SSH Remote Login Protocol
smtp        25/tcp      mail   # Simple Mail Transfer
...
```

Die Dateien `/etc/hosts.allow` und `/etc/hosts.deny` steuern, von welchem Rechner aus welche Dienste verwendet werden dürfen. Die Einstellungen gelten für alle Programme, die auf die TCP-Wrapper-Bibliothek zurückgreifen. Dazu zählen neben *xinetd* auch der SSH-Server, NFS und bei SUSE auch CUPS. Details zum Aufbau der Konfigurationsdateien `hosts.allow` und `hosts.deny` finden Sie in [Abschnitt 37.2](#), »Basisabsicherung von Netzwerkdiensten«.

`/etc/hosts.allow`  
und `hosts.deny`

Ein Sonderfall ist *openbsd-inetd*: Dieses Programm ist zwar mit der TCP-Wrapper-Bibliothek verlinkt, standardmäßig sind seine Funktionen aber deaktiviert! Um TCP-Wrapper-Funktionen zu aktivieren, müssen Sie das Programm mit der Option `-l` starten. Bei Debian und Ubuntu erzeugen Sie dazu die Datei `/etc/default/openbsd-inetd` und fügen die folgende Zeile ein:

```
# Datei /etc/default/openbsd-inetd (Debian, Ubuntu)
# TCP-Wrapper-Funktionen aktivieren
OPTIONS="-l"
```

Mit dieser Option unterbleibt der in `inetd.conf` vorgesehene Aufruf von `tcpd`, um zu vermeiden, dass die TCP-Wrapper-Regeln zweimal überprüft werden.

### openbsd-inetd

`/etc/inetd.conf` Wenn auf Ihrer Distribution `openbsd-inetd` läuft, erfolgt dessen Konfiguration durch die Datei `/etc/inetd.conf`. Das Programm wird nur dann gestartet, wenn die Konfigurationsdatei `inetd.conf` zumindest einen aktiven Eintrag enthält. Jeder Eintrag in dieser Datei besteht aus einer Zeile mit sechs Spalten:

- ▶ Die erste Spalte gibt den Namen des Diensts an, der in `/etc/services` definiert sein muss.
- ▶ Die zweite und die dritte Spalte beschreiben, wie der Dienst kommuniziert (Socket-Typ und Protokoll).
- ▶ Die vierte Spalte beschreibt, ob der gleiche Dienst bei mehreren Anfragen mehrfach gestartet werden soll (`nowait`) oder ob weitere Anfragen erst verarbeitet werden sollen, nachdem der bereits gestartete Dienst fertig ist (`wait`). Optional kann eine Timeout-Zeit in Sekunden angegeben werden.
- ▶ Die fünfte Spalte gibt an, mit welchen Rechten der Prozess gestartet werden soll.
- ▶ Der Rest der Zeile gibt das Kommando an, das ausgeführt werden soll. Dabei führt `tcpd` zuerst einen Test durch, um festzustellen, ob eine Ausführung gemäß den TCP-Wrapper-Regeln erlaubt ist.

```
# Datei /etc/inetd.conf
swat stream tcp nowait.400 root /usr/sbin/tcpd /usr/sbin/swat
...
```

Normalerweise müssen Sie sich nicht selbst um die Konfiguration von `inetd.conf` kümmern. Bei der Installation eines Pakets, das auf `inted` angewiesen ist, wird `inetd.conf` automatisch entsprechend erweitert. Beachten Sie, dass mit `#` eingeleitete Zeilen wie üblich als Kommentare gelten! Änderungen an `inetd.conf` werden erst wirksam, wenn Sie `/etc/init.d/openbsd-inetd reload` ausführen.

### xinetd

`/etc/xinetd.conf` Die Datei `/etc/xinetd.conf` enthält einige Grundeinstellungen für `xinetd`, die beispielsweise das Logging oder die Standard-IP-Adresse betreffen. Im Regelfall können die Einstellungen unverändert bleiben. Entscheidend ist die Anweisung `includedir`, die das Verzeichnis mit den weiteren Konfigurationsdateien angibt, üblicherweise `/etc/xinetd.d`.

Das Verzeichnis `/etc/xinetd.d` enthält für jeden von `xinetd` gesteuerten Dienst eine eigene Konfigurationsdatei. Die Namen dieser Dateien in `/etc/xinetd.d` spielen keine Rolle: `xinetd` liest einfach alle Dateien aus diesem Verzeichnis und wertet sie aus. (Nicht berücksichtigt werden Dateien, deren Name mit `~` endet oder einen Punkt enthält.) `xinetd.d/*`

Der Aufbau der einzelnen Konfigurationsdateien ist einheitlich. Das folgende Beispiel zeigt die Datei für einen RSYNC-Server:

```
# /etc/xinetd.d/rsync
service rsync
{
    disable = yes
    socket_type      = stream
    wait            = no
    user            = root
    server          = /usr/bin/rsync
    server_args     = --daemon
    log_on_failure += USERID
}
```

Die folgende Liste ist eine kurze Erläuterung der wichtigsten Schlüsselwörter, die in `xinetd`-Konfigurationsdateien auftreten können. Eine ausführlichere Beschreibung gibt man `xinetd.conf`.

- ▶ `service` bezeichnet den Dienst (entsprechend `/etc/services`).
- ▶ `socket_type` und `protocol` geben an, wie die Daten zwischen Client und Server übertragen werden.
- ▶ `type = INTERNAL` gibt an, dass es sich um einen Dienst handelt, der direkt von `xinetd` zur Verfügung gestellt wird.
- ▶ `server` gibt den Programmnamen an (sofern es sich nicht um einen internen `xinetd`-Dienst handelt).
- ▶ `server_args` gibt optionale Parameter an, die beim Start an den Dienst übergeben werden sollen.
- ▶ `user` gibt an, unter welchem Account das Programm ausgeführt wird (oft `root`, es ist aber auch `news`, `mail` etc. möglich).
- ▶ `disable = Yes / No` gibt an, ob der Dienst aktiv oder blockiert ist. Bei blockierten Diensten enthält die Konfigurationsdatei `disable=Yes`. Bei Fedora und Red Hat können Sie `xinetd`-Dienste auch durch `chkconfig --del name` deaktivieren bzw. durch `chkconfig --add name` wieder aktivieren. Diese Kommandos verändern nur die `disable`-Zeile und rühren die restliche Konfiguration nicht an.
- ▶ `log_*` gibt an, ob die Nutzung des Dienstes protokolliert werden soll.



# Kapitel 25

## Kernel und Module

Dieses Kapitel beschäftigt sich mit dem Linux-Kernel und seinen Modulen. Module sind Teile des Kernels, die bei Bedarf geladen werden – etwa wenn eine bestimmte Hardware-Komponente zum ersten Mal angesprochen wird. Vorweg ein Überblick:

- ▶ **Kernelmodule:** [Abschnitt 25.1](#) erklärt, warum Kernelmodule automatisch geladen werden und was Sie tun müssen, wenn dieser Automatismus versagt.
- ▶ **Device Trees:** Auf Smartphones und in Embedded Devices gelten für die Verwaltung der Kernelmodule ganz andere Voraussetzungen als auf PCs. Dort haben sich Device Trees zur Verwaltung von Kernelmodulen durchgesetzt. Eine Einführung in diese auch auf dem Raspberry Pi übliche Technik gibt [Abschnitt 25.2](#).
- ▶ **Kernelmodule selbst kompilieren:** Wenn Sie spezielle Hardware einsetzen, müssen Sie eventuell ein eigenes Kernelmodul kompilieren. Wie das gelingt, verrät [Abschnitt 25.3](#).
- ▶ **Den ganzen Kernel kompilieren:** Auch wenn eher selten die Notwendigkeit besteht, den ganzen Kernel neu zu kompilieren, beweist [Abschnitt 25.4](#), dass dies durchaus keine Hexerei ist.
- ▶ **Kernel-Update ohne Reboot:** kexec ermöglicht es, einen anderen Kernel zu aktivieren, ohne den Rechner neu zu starten (siehe [Abschnitt 25.5](#)).
- ▶ **Kernel-Live-Patches:** Noch eleganter ist es, Kernel-Updates im laufenden Betrieb durchzuführen. [Abschnitt 25.6](#) stellt Mechanismen zur Aktivierung derartiger Live-Patches vor.
- ▶ **/proc- und /sys-Dateisystem:** [Abschnitt 25.7](#) zeigt, wie Sie aus dem /proc- bzw. /sys-Dateisystem aktuelle Informationen über den Kernel ermitteln.
- ▶ **Kerneloptionen:** [Abschnitt 25.8](#) erklärt, wie Sie während des Rechnerstarts Optionen an den Kernel übergeben können.
- ▶ **Kernelparameter ändern:** [Abschnitt 25.9](#) beschreibt schließlich, wie Sie Kernelparameter im laufenden Betrieb verändern.

Dieses Kapitel richtet sich explizit an fortgeschrittene Linux-Anwender. Einsteiger sind gut beraten, nur den für ihre Distribution vorgesehenen Kernel zu verwenden! Alle Informationen in diesem Kapitel gelten für die Kernelversionen 3.n und 4.n.

## 25.1 Kernelmodule

Der Kernel ist jener Teil von Linux, der für elementare Funktionen wie Speicherverwaltung, Prozessverwaltung, Zugriff auf Festplatten und Netzwerkkarten etc. zuständig ist. Der Kernel verfolgt dabei ein modularisiertes Konzept: Anfänglich – also beim Hochfahren des Rechners – wird ein Basiskernel geladen, der nur jene Funktionen enthält, die zum Rechnerstart erforderlich sind.

Wenn im laufenden Betrieb Zusatzfunktionen benötigt werden, z. B. für spezielle Hardware, wird der erforderliche Code als Modul mit dem Kernel verbunden. Werden diese Zusatzfunktionen eine Weile nicht mehr benötigt, kann das Modul wieder aus dem Kernel entfernt werden. Dieses modularisierte Konzept hat viele Vorteile:

- ▶ Kernelmodule können nach Bedarf eingebunden werden. Wenn ein bestimmtes Modul nur selten benötigt wird, kann so Speicher gespart werden, d. h., der Kernel ist nicht größer als unbedingt notwendig und optimal an die Hardware des Nutzers angepasst.
- ▶ Bei einer Änderung der Hardware (z. B. einer neuen Netzwerkkarte) muss kein neuer Kernel kompiliert, sondern nur das entsprechende Modul eingebunden werden. Alle gängigen Distributionen basieren auf diesem Konzept.
- ▶ Bei der Entwicklung eines Kernelmoduls muss nicht ständig der Rechner neu gestartet werden. Es reicht, ein Modul neu zu kompilieren. Anschließend kann es bei laufendem Betrieb getestet werden.

Eine Menge Hintergrundinformationen zum Umgang mit Kernelmodulen finden Sie im Module-HOWTO-Dokument. Es ist zwar mehr als 10 Jahre alt, an den Prinzipien der Modulverwaltung hat sich seither aber wenig geändert.

*<http://www.tldp.org/HOWTO/Module-HOWTO>*

### Module automatisch laden

Dafür, dass Kernelmodule tatsächlich automatisch geladen werden, sobald sie benötigt werden, ist die in den Kernel integrierte Komponente `kmod` verantwortlich. `kmod` wird durch die Datei `/etc/modprobe.conf` gesteuert. Diese Konfigurationsdatei wird etwas weiter unten genauer beschrieben.

### Kernel und Module müssen zusammenpassen

In der Vergangenheit mussten der Kernel und seine Module exakt zusammenpassen: Es war nicht möglich, ein Modul zu laden, das für eine andere, vielleicht nur geringfügig veränderte Kernelversion kompiliert wurde. Aus diesem Grund gab und gibt es bis heute für jede Kernelversion ein eigenes Modulverzeichnis `/lib/modules/n.n`.

Gerade bei Modulen, die nicht unmittelbar mit der Distribution mitgeliefert werden, stellte die strikte Versionsabhängigkeit oft ein Problem dar. Das betraf z. B. die Kernelmodule für die Grafiktreiber von AMD und NVIDIA.

Seit der Kernelversion 2.6.16 bringt der Mechanismus des *Module Versioning* eine gewisse Besserung: Zusammen mit dem Modul werden Zusatzinformationen gespeichert, die Aufschluss darüber geben, ob eine Zusammenarbeit zwischen dem Modul und dem Kernel auch bei unterschiedlicher Versionsnummer möglich ist. Damit können oft auch nicht zur Kernelversion passende Module genutzt werden. Dieser Mechanismus funktioniert allerdings nur, wenn das Module Versioning beim Kompilieren aktiviert wurde und wenn es zwischen der Kernel- und der Modulversion keine Änderungen an den Schnittstellen gegeben hat.

Module  
Versioning

Wenn Sie unter SUSE eigene Kernelmodule kompilieren möchten, die Module Versioning unterstützen, müssen Sie vorher das Paket `kernel-syms` installieren. Module Versioning ist auch unter dem Namen *Kernel Symbol Versions* oder *Modversions* bekannt. Im Detail ist der Mechanismus hier beschrieben:

<https://www.oreilly.de/german/freebooks/linuxdrive2ger/kerver.html>

### Kommandos zur Modulverwaltung

Alle gängigen Distributionen sind so eingerichtet, dass Module bei Bedarf automatisch geladen werden. Ein Beispiel: Sie binden mit `mount` das Dateisystem eines USB-Sticks in den Verzeichnisbaum ein. Daraufhin wird automatisch das `vfat`-Modul aktiviert, das zum Lesen des Dateisystems erforderlich ist.

Im Regelfall erfolgt die Modulverwaltung also automatisch und transparent, ohne dass Sie mit den im Folgenden beschriebenen Kommandos zur manuellen Modulverwaltung eingreifen müssen. Dennoch sollten Sie die Modulkommandos kennen, um Module zur Not auch manuell laden zu können.

Alle Module befinden sich im Verzeichnis `/lib/modules/n.n`. Dabei ist `n.n` die Version des laufenden Kernels. Moduldateien haben die Dateierdung `*.ko` bzw. `.ko.xz`, wenn sie mit `xz` komprimiert sind.

Das Kommando `uname -r` liefert die Versionsnummer des laufenden Kernels. Das folgende Kommando wurde unter Fedora 25 ausgeführt. Dort lief ein von den Fedora-Entwicklern modifizierter Kernel der Version 4.10.

Kernelversion  
ermitteln

```
user$ uname -r
4.10.17-200.fc25.x86_64
```

`insmod` integriert das angegebene Modul in den Kernel. Dabei muss der vollständige Dateiname übergeben werden. Zusätzlich können Parameter (Optionen) an das Modul übergeben werden. Falls Sie hexadezimale Werte angeben möchten, müssen Sie `0x` voranstellen, also etwa `option=0xff`. Die folgenden drei Kommandos laden zuerst die Module `dns_resolver` und `fscache`, die als Voraussetzung für das Modul `cifs` gelten.

Moduldatei laden

```
root# cd /lib/modules/4.10.17-200.fc25.x86_64
root# insmod kernel/net/dns_resolver/dns_resolver.ko.xz
root# insmod kernel/fs/fscache/fscache.ko.xz
root# insmod kernel/fs/cifs/cifs.ko.xz
```

`insmod -f` versucht das Modul selbst dann zu laden, wenn es nicht zur laufenden Kernelversion passt. Ob das tatsächlich funktioniert, hängt davon ab, ob es zwischen der Kernel- und der Modulversion irgendwelche Inkompatibilitäten gibt.

**modprobe** Normalerweise werden Sie Kernelmodule nicht mit `insmod` laden, sondern mit `modprobe`. Dieses Kommando bietet im Vergleich zu `insmod` gleich mehrere Vorteile:

- ▶ `modprobe` sucht die Moduldatei selbst, d. h., Sie müssen nur den Modulnamen angeben. Bei Modulen, die schon beim Kompilieren in den Kernel integriert wurden, endet `modprobe` ohne Fehlermeldung.
- ▶ `modprobe` lädt gegebenenfalls auch alle Module, die als Voraussetzung für das gewünschte Modul benötigt werden.
- ▶ `modprobe` berücksichtigt alle in `/etc/modprobe.conf` angegebenen Moduloptionen.

`modprobe` setzt allerdings eine korrekte Modulkonfiguration durch die Dateien `modprobe.conf` und `modules.dep` voraus.

```
root# modprobe cifs
```

#### Liste der geladenen Module

`lsmod` liefert eine normalerweise recht lange Liste aller momentan geladenen Kernelmodule. Beachten Sie, dass `lsmod` in den Kernel einkompilierte Module nicht anzeigt, sondern nur solche Module, die nachträglich geladen wurden!

```
user$ lsmod | sort
Module                Size Used by
ac                    4933  0
autofs4              19013  1
battery              9285  0
bluetooth            44069  5 hidp,rfcomm,l2cap
button               6609  0
...
cifs                 557056  0
dns_resolver         16384  1 cifs
fscache              65536  1 cifs
...
```

Sehr hilfreich bei der Zuordnung von Kernelmodulen ist das Kommando `lspci` mit der Option `-k`. Es listet alle über den PCI-Bus angeschlossenen Hardware-Komponenten auf und zeigt an, welche Treiber geeignet wären, um das Gerät zu steuern, und welcher Treiber tatsächlich verwendet wird (*Kernel driver in use*):



```

user$ lspci -k
00:00.0 Host bridge: Intel Corporation 2nd Generation
  Core Processor Family DRAM Controller (rev 09)
  Subsystem: ASUSTeK Computer Inc. P8P67/P8H67 Series Motherboard
  Kernel driver in use: snb_uncore
00:01.0 PCI bridge: Intel Corporation Xeon E3-1200/2nd Generation Core
  Processor Family PCI Express Root Port (rev 09)
  Kernel driver in use: pcieport
  Kernel modules: shpchp
02:00.0 VGA compatible controller: NVIDIA Corporation GK208
  [GeForce GT 710B] (rev a1)
  Subsystem: Micro-Star International Co., Ltd. [MSI] GK208 [GeForce GT 710B]
  Kernel driver in use: nvidia
  Kernel modules: nvidiafb, nouveau, nvidia_375_drm, nvidia_375
04:00.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL8111/8168/8411
  PCI Express Gigabit Ethernet Controller (rev 06)
  Subsystem: ASUSTeK Computer Inc. P8P67 and other motherboards
  Kernel driver in use: r8169
  !Kernel modules: r8169

```

`rmmod` entfernt das angegebene Modul wieder aus dem Kernel und gibt den belegten Speicher frei. Das Kommando kann nur erfolgreich ausgeführt werden, wenn das Modul gerade nicht verwendet wird.

**Module  
entfernen**

```
root# rmmod cifs
```

`modinfo` liefert eine Menge Informationen über ein Modul. Das Modul muss sich nicht im Kernel befinden. Das folgende Beispiel zeigt die Daten für das Modul `e1000`. Dabei handelt es sich um den Treiber für Intel-Netzwerkadapter.

**Modul-  
informationen**

```

root# modinfo e1000
filename:      /lib/modules/4.10.17-200.fc25.x86_64/kernel/drivers/
              net/ethernet/intel/e1000/e1000.ko.xz
version:      7.3.21-k8-NAPI
license:      GPL
description:   Intel(R) PRO/1000 Network Driver
author:       Intel Corporation, <linux.nics@intel.com>
...
depends:
vermagic:     4.10.17-200.fc25.x86_64 SMP mod_unload
...
parm:         TxDescriptors:Number of transmit descriptors (array of int)
parm:         RxDescriptors:Number of receive descriptors (array of int)
parm:         Speed:Speed setting (array of int)
...

```

## Modulkonfiguration

Die Modulverwaltung funktioniert scheinbar wie von Zauberhand:

- ▶ Wenn Sie eine zusätzliche Partition in das Dateisystem einbinden und dabei ein bisher nicht genutztes Dateisystemformat zum Einsatz kommt, wird automatisch das Modul für dieses Dateisystem geladen.
- ▶ Wenn sich die Partition auf einer SATA-Festplatte befindet, werden auch die SATA-Module aktiviert, sofern diese nicht ohnedies schon geladen sind.
- ▶ Während der Initialisierung der Netzwerkfunktionen wird automatisch der erforderliche Treiber für Ihre Netzwerkkarte geladen etc.

Für das automatische Laden von Kernelmodulen ist die in den Kernel integrierte Komponente `kmod` verantwortlich. Dafür, dass all das funktioniert, sorgen unterschiedliche Konfigurationsmechanismen:

- ▶ **Für den Rechnerstart erforderliche Module:** Manche Kernelmodule werden sofort beim Start des Rechners benötigt – etwa Module zum Zugriff auf das Dateisystem. Sofern diese Module nicht integrale Bestandteile des Kernels sind, müssen sie in einer `initrd`-Datei durch GRUB beim Rechnerstart an den Kernel übergeben werden (siehe [Abschnitt 23.1](#), »GRUB-Grundlagen«).
- ▶ **Module für Grundfunktionen:** Die Module für die Basisverwaltung von Hardware-Komponenten (z. B. für das USB-System) werden von verschiedenen Scripts des Init-Prozesses direkt durch `modprobe`-Anweisungen geladen.
- ▶ **Module für Schnittstellen:** Eine Reihe weiterer Module wird dann geladen, wenn eine Schnittstelle zum ersten Mal benutzt wird. Hier tritt allerdings das Problem auf, dass es für manche Schnittstellen je nach der eingesetzten Hardware unterschiedliche Module gibt. Wenn Sie also die Schnittstelle `eth0` für die erste Netzwerkkarte im Rechner ansprechen, muss das zu dieser Karte passende Modul geladen werden.

Da der Kernel nicht hellsehen kann, benötigt er eine Information darüber, welches Modul das richtige ist. Diese Information befindet sich in `/etc/modprobe.conf` sowie in den Dateien der Verzeichnisse `/etc/modprobe.d` und `/etc/modules-load.d`. Dort befinden sich installations- oder distributionsspezifische Optionen sowie Anweisungen, welche Module *nicht* automatisch zu laden sind (`blacklist`-Datei). Bei `systemd`-Distributionen werden diese Informationen durch die Service-Datei `systemd-modules-load.service` ausgewertet.

Auch die automatische Device-Verwaltung durch das `udev`-System lädt bei Bedarf die notwendigen Module. Die entsprechenden Regeln finden Sie in `/etc/udev/rules.d`.

- ▶ **Module für USB- und Firewire-Geräte etc.:** Derartige Hardware-Komponenten nehmen eine Sonderrolle ein. Mehrere `*.map`-Dateien in `/lib/modules/kernelversion/` entscheiden anhand der Identifikationscodes der Komponenten darüber, welches Modul geladen wird.

- **Modulabhängigkeiten:** Eine Menge Module sind voneinander abhängig. Beispielsweise funktioniert das Modul `nfs` für das NFS-Dateisystem nur, wenn auch die Module `lockd`, `nfs_acl` und `sunrpc` geladen sind. Derartige Modulabhängigkeiten sind zentral in der Datei `/lib/modules/n.n/modules.dep` verzeichnet.

Manchmal wollen Sie unabhängig von den hier zusammengefassten Konfigurationswegen erreichen, dass beim Rechnerstart ein bestimmtes Kernelmodul geladen wird – und das, ohne sich auf irgendwelche Automatismen zu verlassen. Die optimale Vorgehensweise hängt von Ihrer Distribution ab.

Module beim  
Rechnerstart  
laden

Besonders einfach ist es bei Debian und Ubuntu: Dort kümmert sich das Init-Script `/etc/init.d/kmod` darum, alle in `/etc/modules` zeilenweise aufgelisteten Module zu laden. Das gilt auch für aktuelle Debian- und Ubuntu-Versionen mit `systemd` als Init-System. Sie müssen also lediglich das gewünschte Modul in einer neuen Zeile in `/etc/modules` angeben.

Bei den meisten anderen Distributionen fügen Sie `modprobe modulname` in ein für lokale Anpassungen vorgesehenes Init-Script ein. Beachten Sie aber, dass die Module damit je nach Distribution erst zum Ende des Init-Prozesses geladen werden – also zu einem Zeitpunkt, zu dem es für manche System- und Netzwerkprozesse schon zu spät ist.

Red Hat, Fedora: `/etc/rc.d/rc.local`

SUSE: `/etc/init.d/boot.local`

## modprobe-Syntax

Die folgenden Absätze beschreiben die wichtigsten Schlüsselwörter für `modprobe.conf` bzw. die Dateien in `modprobe.d/`. Weitere Details liefert man `modprobe.conf`.

`alias`-Anweisungen geben an, welche Kernelmodule für welche Devices eingesetzt werden. Dazu ein Beispiel: Für das Device `/dev/eth0` soll das Modul `8139too` verwendet werden.

alias

```
alias eth0 8139too
```

Der Zugriff auf viele Hardware-Komponenten erfolgt durch block- und zeichenorientierte Device-Dateien im `/dev`-Verzeichnis. Aus der Sicht des Kernels werden diese Device-Dateien nicht durch ihren Namen, sondern durch die Major- und Minor-Device-Nummer charakterisiert (siehe auch [Abschnitt 11.9](#), »Device-Dateien«). Zahlreiche `alias`-Anweisungen stellen den Zusammenhang zwischen Device-Nummern und Modulen her.

Analog sieht auch die Definition von Netzwerkprotokollen aus: Um ein bestimmtes Protokoll zu nutzen, sucht der Kernel nach einer Protokollfamilie mit dem Namen `net-pf-N`, wobei `N` die ID-Nummer des Protokolls ist. Das folgende Beispiel bewirkt, dass für die Protokollfamilie `5` das `AppleTalk`-Modul geladen wird:

```
alias net-pf-5 appletalk
```

Wenn Sie dieses Protokoll nicht brauchen und womöglich das entsprechende Modul gar nicht installiert ist, dann erspart Ihnen die folgende Anweisung lästige Fehlermeldungen:

```
alias net-pf-5 off
```

**options** options-Anweisungen geben an, mit welchen Optionen ein bestimmtes Modul geladen werden soll. Die folgende Anweisung bewirkt, dass das Modul `ne` (für NE-2000-kompatible Ethernet-Karten) mit der Option `io=0x300` geladen wird:

```
options ne io=0x300
```

**include** include-Anweisungen laden weitere Konfigurationsdateien.

**install** Mit install-Anweisungen geben Sie Kommandos an, die ausgeführt werden, anstatt das betreffende Modul einfach zu laden. Auch hierzu sehen Sie ein Beispiel, das aus Platzgründen auf zwei Zeilen verteilt wurde. Wenn das ALSA-Modul `snd` benötigt wird, sollen die folgenden Kommandos ausgeführt werden:

```
install snd modprobe --ignore-install snd $CMDLINE_OPTS && \
  { modprobe -Qb snd-ioctl32 ; : ; }
```

**remove** Mit `remove` geben Sie Kommandos an, die beim Entfernen eines Moduls ausgeführt werden sollen.

**blacklist** `blacklist` bewirkt, dass modulinterne Alias-Definitionen nicht berücksichtigt werden. `blacklist`-Anweisungen befinden sich üblicherweise in der Datei `/etc/modprobe.d/blacklist`. Sie enthält Module, die beispielsweise wegen Kompatibilitätsproblemen oder aufgrund von besseren Alternativen *nicht* geladen werden sollen. Beispielsweise verhindert die folgende Zeile, dass das Modul `usbmouse` geladen wird:

```
blacklist usbmouse
```

## 25.2 Device Trees

Der Begriff »Device Tree« bezeichnet die hierarchische Darstellung von Hardware-Komponenten. Der Device Tree wird während des Boot-Vorgangs vom Kernel geladen und teilt diesem mit, welche Hardware-Komponenten zur Verfügung stehen und über welche Anschlüsse diese Komponenten genutzt werden.

Device Trees wurden von den Linux-Kernelentwicklern erfunden, um der Vielfalt von Chips und Geräten (sprich: Smartphones) auf Basis von ARM-CPU's Herr zu werden. Dank Device Trees ist es möglich, dass ein für ARM-Geräte kompilierter Kernel auf

unterschiedlichen Geräten mit unterschiedlichen Zusatzkomponenten laufen kann. Bei PCs ist das selbstverständlich; in der ARM-Welt war dies aufgrund der Vielfalt von CPU- und Hardware-Varianten aber bisher unmöglich.

Sofern Sie nicht gerade ein Kernelentwickler für Smartphones sind, kommen Sie am ehesten bei der Arbeit mit Minicomputern wie dem Raspberry Pi mit Device Trees in Kontakt. Device Trees werden beispielsweise in Raspbian standardmäßig eingesetzt.

Beim Raspberry Pi beschreibt der Device Tree den Chip BCM2835/-36/-37 mit all seinen vielen Steuerungsmöglichkeiten, Bus-Systemen, GPIOs etc. Die Beschreibung erfolgt in einem Textformat, das dann aus Platz- und Effizienzgründen in ein binäres Format umgewandelt wird. Die Details dieses Formats sind aus Anwendersicht nicht relevant. Wenn Sie sich dennoch dafür interessieren, finden Sie auf den folgenden Seiten eine umfassende technische Referenz:

<https://devicetree.org>

[http://elinux.org/Device\\_Tree](http://elinux.org/Device_Tree)

<http://linux-magazin.de/Ausgaben/2013/06/Kern-Technik>

### Device-Tree-Konfiguration beim Raspberry Pi

Die Device-Tree-Konfiguration des Raspberry Pi ist über mehrere Orte verteilt. Das Verzeichnis `/boot` enthält Device-Tree-Beschreibungen für alle momentan gängigen Raspberry-Pi-Modelle. Die Dateikennung `*.dtb` steht dabei für *Device Tree Blob*, wobei ein »Blob« einfach ein binäres Objekt ist. Die Nummern 2708 bis 2710 sind Broadcom-interne Nummern zur Beschreibung der Chip-Familie. Die auf dem Raspberry Pi eingesetzten Modelle BCM2835 bis BCM2837 sind konkrete Implementierungen (»Familienmitglieder«, wenn Sie so wollen).

boot-Verzeichnis

- ▶ `bcm2708-rpi-0-w.dtb`: Raspberry Pi Zero W
- ▶ `bcm2709-rpi-2-b.dtb`: Raspberry Pi 2, Modell B
- ▶ `bcm2710-rpi-3-b.dtb`: Raspberry Pi 3, Modell B
- ▶ `bcm2710-rpi-cm3.dtb`: Raspberry Pi 3, Compute Module

Während des Boot-Prozesses übergibt das in der Boot-Datei `start.elf` enthaltene Programm den für das jeweilige Raspberry-Pi-Modell geeigneten Device Tree an den Kernel.

Das Verzeichnis `/boot/overlays` enthält über 100 Device-Tree-Blob-Overlays (DTBOs), von denen ich hier nur einige wenige exemplarisch nenne:

overlays-Verzeichnis

- ▶ `hifiberry-dacplus-overlay.dtb`: für die HiFiBerry-Erweiterung DAC+
- ▶ `lirc-rpi-overlay.dtb`: für Infrarot-Fernbedienungen

- ▶ `i2c-rtc-overlay.dtb`: für I<sup>2</sup>C-Komponenten mit Real Time Clock
- ▶ `w1-gpio-overlay.dtb`: für 1-Wire-Temperatursensoren

»Overlays« sind also Ergänzungen zum Haupt-Device-Tree `bcmxxx`. Diese DTBs werden *nicht* automatisch geladen, sondern nur, wenn die Konfigurationsdatei `config.txt` entsprechende Hinweise enthält. Sie ergänzen den Device Tree des BCM2835/-36/-37.

**config.txt** Die Raspberry-Pi-spezifische Datei `/boot/config.txt` kennt drei Schlüsselwörter zum Umgang mit Device Trees:

- ▶ `dtparam=i2c_arm=on/off,i2s=on/off,spi=on/off` aktiviert oder deaktiviert die Bussysteme I<sup>2</sup>C, I<sup>2</sup>S und SPI. Die Beschreibung dieser Bussysteme ist im Haupt-Device-Tree bereits enthalten. Standardmäßig sind alle drei Bussysteme inaktiv (entspricht `dtparam=i2c_arm=off,i2s=off,spi=off`). Wenn ein Bussystem oder mehrere genutzt werden sollen, müssen sie explizit aktiviert werden.
- ▶ `dtoverlay=name,key1=val1,key2=val2...` bewirkt, dass die genannte Overlay-Datei geladen wird, wobei die übergebenen Parameter berücksichtigt werden.
- ▶ `device_tree=`, also ohne die Zuweisung eines konkreten Werts, deaktiviert das gesamte Device-Tree-System.

Die Schlüsselwörter `dtparam` und `dtoverlay` können mehrfach verwendet werden.

#### Manuelle Konfiguration

Anhand der Device-Tree-Konfiguration entscheidet der Linux-Kernel, welche Module er lädt. Daher ersetzt die Device-Tree-Konfiguration die bei älteren Raspbian-Installationen erforderlichen Einstellungen in `/etc/modules` bzw. in `/etc/modprobe.d/*`.

In einfachen Fällen können Sie `config.txt` mit dem Programm `raspi-config` im Untermenü `ADVANCED OPTIPONS` korrekt einrichten. Das gilt insbesondere, wenn Sie die Bussysteme I<sup>2</sup>C und SPI verwenden möchten.

In allen anderen Fällen müssen Sie die entsprechenden Zeilen hingegen selbst in `config.txt` eintragen. Das folgende Listing illustriert die Syntax anhand einiger Beispiele. Beachten Sie, dass mehrere Optionen nur durch Kommata, aber nicht durch Leerzeichen voneinander getrennt werden dürfen (siehe z. B. `dtoverlay=...`).

```
# Datei /boot/config.txt
# Beispiele zur Steuerung des Device-Tree-Systems
# (weitere Details siehe /boot/overlays/README in einer Raspbian-Installation)

# Audio-System aktivieren
dtparam=audio=on

# SPI-Bus aktivieren
dtparam=spi=on
```

```
# I2C-Bus aktivieren
dtparam=i2c_arm=on

# HiFiBerry DAC+ verwenden
dtoverlay=hifiberry-dacplus

# 1-Wire-Temperatursensor mit Standardeinstellungen verwenden
dtoverlay=w1-gpio-pullup

# 1-Wire-Temperatursensor verwenden, der mit
# GPIO X verbunden ist (per Default GPIO 4),
# und dabei den internen Pull-up-Widerstand aktivieren
dtoverlay=w1-gpio-pullup,gpiopin=X,pullup=y

# Echtzeituhr-Modell ds1307 verwenden
dtoverlay=rtc-i2c, ds1307

# IR-Empfänger verwenden
dtoverlay=lirc-rpi
```

## 25.3 Kernelmodule selbst kompilieren

Wenn Sie Linux in Kombination mit VirtualBox einsetzen, die binären Grafiktreiber von AMD oder NVIDIA nutzen möchten oder ein anderes hardware-spezifisches Kernelmodul brauchen, das im Kernel Ihrer Distribution fehlt, müssen Sie das Modul passend zum laufenden Kernel kompilieren.

Zum Kompilieren eines Moduls sind neben dem C-Compiler gcc und make auch weitere grundlegende Entwicklungswerkzeuge erforderlich. Die meisten Distributionen erleichtern die Sache durch fertige Paketselektionen oder Meta-Pakete, die auf alle relevanten Pakete verweisen (siehe [Tabelle 25.1](#)).

Entwicklungs-  
werkzeuge

| Distribution   | Kommando                              |
|----------------|---------------------------------------|
| CentOS, RHEL   | yum groupinstall development-tools    |
| Debian, Ubuntu | apt-get install build-essential       |
| Fedora         | dnf groupinstall development-tools    |
| SUSE           | zypper install -t pattern devel_basis |

**Tabelle 25.1** Kommandos zur Installation grundlegender Entwicklungswerkzeuge

Außerdem brauchen Sie zumindest die Include-Dateien (Header-Dateien) zum aktuellen Kernel. Diese Dateien sind Teil des Kernelcodes. Bei vielen Distributionen (aber nicht bei SUSE) befinden sich die Include-Dateien und der Rest des Codes in zwei

Kernel-Include-  
Dateien

getrennten Paketen. Das hat den Vorteil, dass Sie nicht gleich den riesigen Kernelcode installieren müssen, wenn Sie nur die vergleichsweise kleinen Include-Dateien brauchen.

Tabelle 25.2 gibt an, in welchen Paketen sich die Include-Dateien des Kernels bei den gängigen Distributionen befinden und wohin diese Dateien installiert werden. `n.n` ist dabei ein Platzhalter für die installierte Kernelversion. Diese Information ermitteln Sie mit dem Kommando `uname -a`.

| Distribution | Paket                  | Pfad                           |
|--------------|------------------------|--------------------------------|
| Debian       | linux-headers-arch     | /usr/include/linux             |
| Fedora, RHEL | kernel-[PAE-]devel-n.n | /lib/modules/n.n/build/include |
| SUSE         | kernel-devel           | /usr/src/linux-n.n/include     |
| Ubuntu       | linux-headers-generic  | /usr/include/linux             |

**Tabelle 25.2** Pakete mit den Kernel-Header-Dateien

Wenn Sie den Kernel selbst kompilieren (siehe Abschnitt 25.4), landen die zum Kernel passenden Include-Dateien automatisch im Verzeichnis `/lib/modules/n.n/build/include`.

**PAE** PAE steht für *Physical Address Extension* und ist ein Mechanismus, um mit 32-Bit-CPU's mehr als 4 GiB RAM zu nutzen. Die Aktivierung von PAE hat unabhängig von der Bit-Anzahl der CPU und dem verfügbaren RAM einen zweiten Vorteil: Nur mit PAE kann das Schutzsystem *No Execute* (NX) genutzt werden. NX verhindert, dass bei einem Pufferüberlauf Code aus dem Datenbereich eines Programms ausgeführt werden kann. Aus diesem Grund läuft bei den meisten 32-Bit-Distributionen ein Kernel mit PAE-Unterstützung.

**Modul kompilieren** Die meisten Programme, die eigene Kernelmodule benötigen, enthalten ein Installations-Script, das sich um das Kompilieren und Einrichten des Moduls kümmert. Das gilt beispielsweise für VMware, VirtualBox, die Grafiktreiber von AMD und NVIDIA etc. Bei manchen Distributionen ist der Prozess sogar dahingehend automatisiert, dass nach jedem Kernel-Update automatisch das Modul neu kompiliert wird (siehe *DKMS* etwas weiter unten).

Wenn Sie dagegen den Quellcode für eine noch nicht offiziell unterstützte Hardware-Komponente heruntergeladen haben, müssen Sie sich um den Kompilierprozess selbst kümmern. Dazu führen Sie in der Regel die folgenden Kommandos aus. Nur das letzte `make`-Kommando erfordert `root`-Rechte.



```

user$ cd quellcodeverzeichnis
user$ make clean
user$ make
root# make install

```

### Modul-Updates automatisieren

DKMS steht für *Dynamic Kernel Module Support* und hilft dabei, nach einem Kernel-Update selbst kompilierte Kernelmodule automatisch zu aktualisieren. DKMS besteht aus einigen Shell-Skripts und wurde von Dell entwickelt. Die entsprechenden `dkms`-Pakete stehen gegenwärtig für die Distributionen Debian, Fedora und Ubuntu zur Verfügung.

Um DKMS zu nutzen, muss der Quellcode des Moduls in einem Verzeichnis der Form `/usr/src/NAME-VERSION` installiert werden. Das Verzeichnis muss die Datei `dkms.conf` enthalten, die DKMS erklärt, wie es mit dem Code umgehen soll. Die folgenden Zeilen stammen vom NVIDIA-Treiber für Ubuntu, wobei ich die Formatierung des Listings ein wenig geändert habe, um die Lesbarkeit zu verbessern.

```

# Datei /usr/src/nvidia-375-375.66/dkms.conf
PACKAGE_NAME           = "nvidia-375"
PACKAGE_VERSION        = "375.66"
CLEAN                  = "make clean"
BUILT_MODULE_NAME[0]   = "nvidia"
DEST_MODULE_NAME[0]    = "nvidia_375"
DEST_MODULE_LOCATION[0] = "/kernel/drivers/char/drm"
PROCS_NUM              = `nproc`
[ $PROCS_NUM -gt 16 ] && PROCS_NUM=16

MAKE[0]                = "'make' -j$PROCS_NUM NV_EXCLUDE_BUILD_MODULES=' '
                        KERNEL_UNAME=${kernelver} IGNORE_XEN_PRESENCE=1
                        IGNORE_CC_MISMATCH=1 SYSSRC=$kernel_source_dir
                        LD=/usr/bin/ld.bfd modules"
BUILT_MODULE_NAME[1]   = "nvidia-modeset"
DEST_MODULE_NAME[1]    = "nvidia_375_modeset"
DEST_MODULE_LOCATION[1] = "/kernel/drivers/char/drm"
BUILT_MODULE_NAME[2]   = "nvidia-drm"
DEST_MODULE_NAME[2]    = "nvidia_375_drm"
DEST_MODULE_LOCATION[2] = "/kernel/drivers/char/drm"
AUTOINSTALL="yes"
...

```

Sind diese Voraussetzungen erfüllt, übergeben Sie das Kernelmodul mit `dkms add` der Kontrolle von DKMS, kompilieren es mit `dkms build` für den aktuellen Kernel und installieren es mit `dkms install`. In Zukunft geschieht dies bei Kernel-Updates automatisch. Die folgenden Beispiele beziehen sich wieder auf den NVIDIA-Kerneltreiber.

Die Kommandos werden bei der Installation des Ubuntu-Pakets des Treibers automatisch ausgeführt. Nach meinen Erfahrungen funktioniert dieser Automatismus oft, aber leider nicht immer. Insbesondere die automatische Aktualisierung von Gasttreibern in virtuellen Maschinen scheitert auf meinen Testsystemen häufig.

```
root# dkms add -m nvidia-current -v 375-375.66
root# dkms build -m nvidia-current -v 375-375.66
root# dkms install -m nvidia-current -v 375-375.66
```

`dkms status` bzw. ein Blick in das Verzeichnis `/var/lib/dkms` verrät, welche Kernelmodule sich momentan unter der Kontrolle von DKMS befinden.

#### DKMS unter Debian und Ubuntu

Bei Debian und Ubuntu gibt es eine ganze Reihe von `xxx-name-dkms`-Paketen, mit denen Kernelmodule für Hardware-Treiber kompiliert werden können:

```
root# apt-cache --names-only search '.*-dkms' | sort
```

Bei Debian befinden sich die Pakete zum Teil in den Paketquellen *contrib* und *non-free*, die Sie vorher aktivieren müssen. Die Installation eines Kernelmoduls sieht dann wie folgt aus, wobei Sie einfach `nvidia` durch den Namen des gewünschten Treibers und `amd64` durch Ihre CPU-Architektur ersetzen:

```
root# apt-get update
root# apt-get install linux-headers-amd64 nvidia-kernel-dkms
```

Im Rahmen der Installation werden alle abhängigen Pakete installiert sowie das Kernelmodul kompiliert und als DKMS-Modul eingerichtet. Bei zukünftigen Kernel-Updates wird somit automatisch eine neue Version des Moduls erzeugt.

#### DKMS oder module-assistant?

Vor allem in Debian kamen zum Kompilieren von Kernelmodulen in der Vergangenheit häufig die Werkzeuge aus dem Paket `module-assistant` zum Einsatz. Das gleichnamige Kommando bzw. dessen Kurzform `m-a` existiert weiterhin; nach Möglichkeit sollten Sie aber DKMS-Pakete vorziehen!

**akmods** Die RPMFusion-Paketquelle für Fedora verwendet anstelle von DKMS einen eigenen Mechanismus zum Kompilieren von Kernelmodulen: Das Kommando `akms` wird nach Kernel-Updates aufgerufen. Es kompiliert zu allen in `/usr/src/akmods` befindlichen RPM-Source-Paketen die entsprechenden Kernelmodule und installiert diese.

<https://rpmfusion.org/Packaging/KernelModules/Akmods>

## 25.4 Kernel selbst konfigurieren und kompilieren

Der durchschnittliche Linux-Anwender muss seinen Kernel nicht selbst kompilieren. Bei allen aktuellen Distributionen werden ein brauchbarer Standardkernel und eine umfangreiche Sammlung von Modulen mitgeliefert. Dennoch kann es Gründe geben, den Kernel neu zu kompilieren:

- ▶ Sie wollen Ihr System besser kennenlernen. Das Motto dieses Buchs ist es ja, Ihnen auch einen Blick hinter die Linux-Kulissen zu ermöglichen.
- ▶ Sie brauchen besondere Funktionen, die weder in den mitgelieferten Kernel integriert sind noch als Modul vorliegen.
- ▶ Sie möchten eine aktuellere Version des Kernels verwenden als die, die mit Ihrer Distribution mitgeliefert wurde.
- ▶ Sie möchten selbst an der Kernelentwicklung teilnehmen und daher mit dem neuesten Entwicklerkernel experimentieren.
- ▶ Sie wollen in Ihrem Bekanntenkreis mit Insider-Wissen auftrumpfen: »Ich habe den neuesten Linux-Kernel selbst kompiliert!«

Es gibt allerdings gewichtige Gründe, die gegen das Kompilieren eines eigenen Kernels sprechen: Hürden

- ▶ Die meisten Distributionen verwenden nicht den Originalkernel, wie er von Linus Torvalds freigegeben wird, sondern eine gepatchte Version mit diversen Zusatzfunktionen, wobei natürlich jede Distribution andere Patches verwendet. An sich ist das eine feine Sache für den Anwender: Er bekommt auf diese Weise Zusatzfunktionen, von denen der Distributor glaubt, dass sie schon ausreichend stabil funktionieren. Wenn Sie sich nun aber selbst den Quellcode des Originalkernels herunterladen, fehlen diese Patches. Einzelne Funktionen Ihrer Distribution, die bisher einwandfrei gearbeitet haben, machen plötzlich Probleme oder funktionieren gar nicht mehr.
- ▶ Das Kompilieren eines eigenen Kernels ist nicht schwierig. Schwierig ist aber die vorherige Konfiguration des Kompilationsprozesses. Dabei stehen Tausende von Optionen zur Auswahl. Sie können mit diesen Optionen beeinflussen, welche Funktionen direkt in den Kernel integriert werden, welche als Module und welche gar nicht zur Verfügung stehen sollen.

Wenn Sie sich – mangels Detailwissen – für die falschen Optionen entscheiden, ist das Ergebnis wie oben: Einzelne Funktionen verweigern den Dienst, und es ist relativ schwierig, die Ursache herauszufinden. Gerade für Linux-Einsteiger ist es praktisch unmöglich, die richtigen Einstellungen für alle Optionen richtig zu erraten.

Aus diesen Gründen verweigern die meisten Distributoren jeden Support, wenn Sie nicht den mit der Distribution mitgelieferten Kernel verwenden. Lassen Sie sich von diesen Warnungen aber nicht abschrecken, es einmal selbst zu versuchen. Wenn Sie nach der in diesem Abschnitt präsentierten Anleitung vorgehen, können Sie Ihren Rechner anschließend sowohl mit dem alten als auch mit dem neuen Kernel hochfahren – es kann also nichts passieren!

**Entwicklungswerkzeuge** Zur Kompilierung des Kernels sind dieselben Entwicklungswerkzeuge wie zum Kompilieren eines einzelnen Moduls erforderlich (siehe [Abschnitt 25.3](#)).

## Grundlagen

**Kernelversionen** In der Vergangenheit gab es »stabile« Kernelversionen mit einer geraden Endnummer, also 2.0.*n*, 2.2.*n*, 2.4.*n*) etc. Dazwischen gab es die sogenannten Entwickler- bzw. Hacker-Kernel 2.3.*n*, 2.5.*n* etc. Die meisten Linux-Distributionen verwendeten stabile Kernelversionen, während die Entwicklerkernel für Programmierer gedacht waren, die sich an der Kernelentwicklung beteiligen. Neue Funktionen wurden zuerst im Hacker-Kernel getestet, bevor sie später in die nächste stabile Kernelgeneration Einzug hielten.

2003 hat sich mit Kernel-Version 2.6 das Entwicklungsmodell geändert: Es gibt seither keine Hacker-Kernel mehr. Stattdessen erfolgt die Weiterentwicklung direkt in jeder einzelnen Version. Man könnte sagen, dass jede neue Kernelversion vorerst als Hacker-Kernel gilt; erst wenn Linus Torvalds entscheidet, dass die Version zuverlässig läuft, wird sie als stabile Version freigegeben. Dieser Zyklus dauert normalerweise rund zwei Monate. Der Hauptvorteil besteht darin, dass Neuerungen von wesentlich mehr Entwicklern getestet werden und viel schneller allgemein verfügbar werden.

**Versionsnummern** 2011 hat Linus Torvalds die Nummerierung der Kernelversionen geändert: Während sich eine Versionsnummer vorher aus vier Teilen zusammensetzte (z. B. 2.6.39.7, also das siebte Update zur Versionsnummer 2.6.39), gibt es seit Version 3.0 nur noch drei Teile.

Als ich dieses Kapitel Ende Mai 2017 überarbeitete, war die Kernelversion 4.11 aktuell. Diese hat Linus Torvalds am 1. Mai freigegeben. Bis zum 29. Mai gab es drei kleinere Updates zur Behebung von Bugs und Sicherheitsproblemen. Die vollständige Versionsnummer lautete zu diesem Zeitpunkt also 4.11.3.

**Statistik** Der Kernel besteht zurzeit (Version 4.11) aus ca. 23 Millionen Zeilen Code. Der Großteil davon ist in C geschrieben, ein kleiner Teil in Assembler. Wenn Sie wissen möchten, wer bzw. welche Firmen zur Kernelentwicklung beitragen, verfolgen Sie einfach die Linux-News-Site *lwn.net*. Dort finden Sie zu jedem Kernel-Release eine statistische Aufarbeitung, wer die meisten Änderungen durchgeführt hat. Auch auf

<https://heise.de> gibt es regelmäßig exzellente Artikel über alle wichtigen Neuerungen im jeweils aktuellsten Kernel:

<https://lwn.net/Articles/720336> (beide Links für Version 4.11)  
<https://heise.de/-3641334>

Tipps zur Kompilierung des Kernels finden Sie auf der folgende Seite:

Links

<https://kernelnewbies.org/FAQ>

Wenn Sie sich für technische Interna interessieren, sind die Dokumentationsdateien des Kernelcodes sehr aufschlussreich. Gerade neue Funktionen des Kernels werden zuerst an dieser Stelle beschrieben, noch bevor die entsprechenden man-Seiten aktualisiert werden:

<https://www.kernel.org/doc/Documentation>

### Kernelcode installieren

Der Quellcode für den Kernel befindet sich üblicherweise im Verzeichnis `/usr/src/linux`; nur bei Red Hat und Fedora gibt es abweichende Gepflogenheiten, die weiter unten behandelt werden. Falls dieses Verzeichnis leer ist, haben Sie den Kernelcode nicht installiert. Sie können nun wahlweise den Kernelquellcode Ihrer Distribution installieren oder den gerade aktuellen offiziellen Kernelcode herunterladen. Weniger Probleme bereitet zumeist die erste Variante, insbesondere für Einsteiger.

#### Ist genug Platz auf der Festplatte/SSD?

Beachten Sie, dass der Platzbedarf für den Kernelcode beachtlich ist: Die komprimierten Quellcodepakete sind mehr als 90 MiB groß. Nach dem Entpacken beträgt der Platzbedarf in etwa weitere 700 MiB, und nach dem Kompilieren mit den dadurch resultierenden Binärdateien rund 15 GiB! Zuletzt können Sie mit `make clean` zahllose Objektdateien wieder löschen, aber zwischenzeitlich brauchen Sie genug freien Speicherplatz.

Bei den meisten Distributionen gibt es ein eigenes Paket, das den Kernelquellcode enthält. [Tabelle 25.3](#) gibt für einige gängige Distributionen an, in welchen Paketen sich der Kernelcode befindet. Dabei ist `n.n` ein Platzhalter für die installierte Kernelversion.

Kernelcode der Distribution installieren

Bei Debian und Ubuntu wird der Kernelcode als tar-Archiv in das Verzeichnis `/usr/src` installiert. Sie müssen das Archiv selbst mit `tar xJf linux-n.n.tar.xz` auspacken. Die Kennung `.xz` deutet darauf hin, dass der Quellcode mit dem besonders effizienten XZ-Verfahren komprimiert wurde.

| Distribution    | Paket                        |
|-----------------|------------------------------|
| Debian, Ubuntu  | linux-source-n.n             |
| Fedora, Red Hat | kernel-n.n (Quellcodepaket!) |
| SUSE            | kernel-source                |

Tabelle 25.3 Pakete mit dem Kernelquellcode

**Fedora** Fedora ist unter Kernel-Entwicklern eine besonders beliebte Distribution. Bevor Sie loslegen, müssen Sie diverse Entwicklerpakete installieren und den aktuellen User-Account für `pesign` freischalten. (`pesign` ist ein Kommando zum Signieren von UEFI-Programmen.)

```
user$ sudo dnf install fedpkg fedora-packager rpmdevtools \
ncurses-devel pesign
user$ sudo /usr/libexec/pesign/pesign-authorize-users
```

Die Fedora-Entwickler empfehlen, den Quellcode des Kernels sowie aller Fedora-Patches mit `fedpkg clone -a kernel` aus einem Git-Repository herunterzuladen. Details können Sie hier nachlesen:

<https://fedoraproject.org/wiki/Docs/CustomKernel>

**Offiziellen  
Kernelcode  
installieren**

Der mit der Distribution mitgelieferte Kernel ist oft schon veraltet. Den aktuellen Kernelcode in Form von komprimierten tar-Archiven finden Sie hier:

<https://www.kernel.org>

Beachten Sie, dass es sich dabei um den originalen Kernelcode handelt, wie ihn Linus Torvalds freigegeben hat – also ohne distributionsspezifische Patches. Dieser Kernel wird oft »Vanilla Kernel« genannt.

Ein typischer Dateiname für das Kernelarchiv ist etwa `linux-4.11.3.tar.xz` (Größe ca. 90 MiB). Zur Installation wechseln Sie in ein Arbeitsverzeichnis mit genügend Platz und führen das folgende Kommando aus:

```
user$ cd /usr/src (oder ein beliebiges anderes Verzeichnis)
user$ sudo chgrp ihr-account .
user$ sudo chmod g+rxw .
user$ tar xJf linux-4.11.3.tar.xz
```

### Kernelcode aktualisieren (patchen)

Mit sogenannten Patch-Dateien können Sie einen Versionswechsel von einer Version zu einer anderen durchführen. Patches sind komprimierte Textdateien, die angeben, in welchen Dateien welche Änderungen durchgeführt werden sollen. Patches sparen

insbesondere bei kleinen Versionswechslern eine Menge Download-Volumen. Patches funktionieren allerdings nur dann, wenn sie auf die dafür gedachte (unveränderte!) Codebasis angewendet werden.

Nehmen wir an, Sie wollen den Code von 4.11.5 auf 4.11.6 aktualisieren: Naheliegender wäre es, einfach den Patch 4.11.6 anzuwenden. Das funktioniert so aber nicht, weil der Patch 4.11.6 als Basis den unveränderten Code 4.11 erwartet (nicht 4.11.5!). Daher müssen Sie auch den Patch 4.11.5 herunterladen und diesen invers anwenden (Option `-R`), um von 4.11.5 zurück zur Codebasis 4.11 zu gelangen. Erst jetzt funktioniert der Patch 4.11.6!

Die richtige Patch-Reihenfolge

Das `patch`-Kommando wird normalerweise in Kombination mit `bunzip2` eingesetzt. `bunzip2` dekomprimiert den Patch, `patch` führt die Änderungen aus. Falls die Patch-Datei unkomprimiert vorliegt, lautet das Patch-Kommando `patch -p1 < patchdatei`.

Generell sollten Sie vor der Anwendung jedes Patches mit der Option `--dry-run` sicherstellen, dass dabei keine Probleme auftreten. Nichts ist ärgerlicher als ein fehlerhaft oder nur teilweise angewendeter Patch!

Die Patches verändern nur den Code, nicht aber den Namen des Verzeichnisses, in dem sich der Code befindet. Um Verwirrung zu vermeiden, sollten Sie anschließend auch das Codeverzeichnis umbenennen. Die tatsächliche Versionsnummer können Sie der Datei `Makefile` direkt im Quellcodeverzeichnis entnehmen.

```
user$ cd /usr/src/linux-4.11.5
user$ bunzip2 -c patch-4.11.5.xz | patch -R -p1 --dry-run (Invers-Patch testen)
... keine Fehlermeldungen
user$ bunzip2 -c patch-4.11.5.xz | patch -R -p1          (4.11.5 --> 4.11)
user$ bunzip2 -c patch-4.11.6.xz | patch -p1 --dry-run (Patch testen)
... keine Fehlermeldungen
user$ bunzip2 -c patch-4.11.6.xz | patch -p1          (4.11 --> 4.11.6)
user$ cd /usr/src
user$ mv linux-4.11.5 linux-4.11.6
```

Neben den gerade beschriebenen Update-Patches gibt es auch Patches mit inoffiziellen Zusatzfunktionen, die aus den verschiedensten Gründen noch nicht in den Standardkernel integriert sind (Funktions-Patches).

Funktions-Patches

Grundsätzlich werden auch Funktions-Patches mit `patch` auf den Kernelcode angewendet. Allerdings müssen Sie darauf achten, dass Sie dieselbe Codebasis haben wie der Entwickler, der den Patch zur Verfügung gestellt hat. In der Regel ist als Codebasis nur der offizielle Kernelcode in der gerade aktuellen Version geeignet, nicht der oft schon gepatchte Kernelcode Ihrer Distribution.

## Mitgelieferte Kernelkonfigurationsdateien verwenden

Der Kernel besteht aus Tausenden von Einzelfunktionen bzw. Komponenten. Bei nahezu allen Funktionen können Sie vor dem Kompilieren angeben, ob sie direkt in den Kernel integriert werden, als Modul kompiliert werden oder gar nicht verfügbar sein sollen. Dieser Vorgang heißt den »Kernel konfigurieren«.

**.config-Datei** Die Kernelkonfiguration wird durch die Datei `.config` im Verzeichnis `/usr/src/linux-n.n` bestimmt. Dabei handelt es sich um eine fast 8000 Zeilen lange Textdatei, die angibt, ob eine Funktion direkt in den Kernel integriert (`name=y`) oder als Modul kompiliert werden soll (`name=m`). Nicht benötigte Funktionen erscheinen in der Konfigurationsdatei nicht bzw. nur in Kommentarzeilen. Die Datei kann auch zusätzliche Einstellungen enthalten (`name=wert`). Die folgenden Zeilen zeigen den Beginn einer `.config`-Datei:

```
CONFIG_64BIT=y
CONFIG_X86_64=y
CONFIG_X86=y
CONFIG_INSTRUCTION_DECODER=y
CONFIG_PERF_EVENTS_INTEL_UNCORE=y
CONFIG_OUTPUT_FORMAT="elf64-x86-64"
```

Wenn Sie bei der manuellen Kernelkonfiguration (siehe den folgenden Abschnitt) keinen Ausgangspunkt haben, müssen Sie sich wirklich um alle Kerneloptionen kümmern. Gerade beim ersten Mal ist es so gut wie sicher, dass Sie irgendetwas übersehen werden. Sie sparen eine Menge Zeit und Mühe, wenn Sie die mit Ihrer Distribution mitgelieferte Kernelkonfigurationsdatei als Ausgangspunkt verwenden:

```
user$ cp old-config /pfad/zum/code/linux-n.n/.config
```

Dieses Verfahren hat leider einen Nachteil: Wenn der ursprüngliche Kernelcode andere Patches enthält als der neu zu kompilierende Code, enthält auch die ursprüngliche Konfigurationsdatei Optionen, die im neuen Code nicht vorgesehen sind. Das kann zu Problemen führen. Wie ich schon erwähnt habe, bauen viele Distributoren diverse Patches in ihren Kernel ein, die im Standardkernel nicht enthalten sind. Deswegen müssen Sie anschließend in das Quellcodeverzeichnis wechseln und dort das folgende Kommando ausführen:

```
user$ cd /pfad/zum/code/linux-n.n
user$ make oldconfig
```

`make oldconfig` wertet die vorhandene `.config`-Datei aus. Fehlen dort Optionen, die der aktuelle Kernelcode vorsieht, dann werden entsprechende Rückfragen angezeigt.



Jetzt bleibt noch die Frage offen, woher Sie die aktuelle Kernelkonfigurationsdatei für das Kommando `cp old-config` hernehmen. Bei nahezu allen Distributionen befindet sich im Verzeichnis `/boot` die zum laufenden Kernel passende Konfigurationsdatei, also z. B. `/boot/config-n.n`. Somit wird aus `cp old-config` beispielsweise:

```
user$ cd /pfad/zum/code/linux-n.n
user$ /boot/config-4.11.0-300.fc23.x86_64 .config
```

**Aktuelle  
Konfiguration  
feststellen**

Der mit SUSE mitgelieferte Kernel verwendet die `cloneconfig`-Option (Gruppe *General setup*). Das bedeutet, dass `/proc/config.gz` den komprimierten Inhalt der `.config`-Datei enthält, mit der der gerade laufende Kernel kompiliert wurde. Mit `make cloneconfig` kopieren Sie die zuletzt verwendete Konfiguration in die Datei `.config`.

**cloneconfig**

### Kernel manuell konfigurieren

Prinzipiell müssen Sie sich zwischen zwei Kerneleypen entscheiden: monolithischen Kernen oder modularisierten Kernen. Monolithische Kernel enthalten alle benötigten Treiber direkt im Kernel und unterstützen keine Module. Modularisierte Kernel sind über die integrierten Treiber hinaus in der Lage, im laufenden Betrieb zusätzliche Module aufzunehmen. Ein modularisierter Kernel ist in fast allen Fällen die bessere Entscheidung.

**Monolithischer  
oder  
modularisierter  
Kernel?**

Bei den meisten Komponenten haben Sie die Wahl zwischen drei Optionen: YES, MODULE und NO. YES bedeutet, dass diese Komponente direkt in den Kernel integriert wird. MODULE bedeutet, dass diese Komponente als Modul kompiliert wird (nur sinnvoll bei einem modularisierten Kernel). NO bedeutet, dass die Komponente überhaupt nicht kompiliert wird. Es gibt auch eine Reihe von Funktionen, die nicht als Module zur Verfügung gestellt werden können – dort reduziert sich die Auswahl auf YES oder NO.

**Komponenten-  
auswahl**

Die übliche Vorgehensweise besteht darin, in den modularisierten Kernel nur relativ wenige elementare Funktionen zu integrieren und alle anderen Funktionen als Module verfügbar zu machen. Der Vorteil: Der Kernel an sich ist relativ klein, Module werden nur nach Bedarf nachgeladen.

**Konfigurations-  
strategien**

Eine alternative Strategie besteht darin, einen monolithischen Kernel möglichst exakt für die eigenen Hard- und Software-Ansprüche zu optimieren. Alle Funktionen, die genutzt werden sollen, integrieren Sie direkt in den Kernel. Bei allen anderen Komponenten entscheiden Sie sich für NO.

Generell wird ein monolithischer Kernel immer etwas größer als ein modularisierter Kernel. Dafür funktioniert er ohne die dynamische Modulverwaltung, und der Rechnerstart gelingt ohne `initrd`-Datei. Der Nachteil ist offensichtlich: Wenn Sie eine

bestimmte Funktion später brauchen, müssen Sie den Kernel neu kompilieren. Und nur echte Linux-Profis können abschätzen, welche Funktionen sie nutzen werden.

### Werkzeuge zur manuellen Kernelkonfiguration

Um abweichend von der aktuellen Konfiguration einzelne Einstellungen zu verändern, können Sie `.config` manuell editieren. Das ist aber fehleranfällig und erfordert eine gute Kenntnis der Namen der diversen Optionen. Besser ist es daher, mit `make xxxconfig` ein spezielles Konfigurationsprogramm zu starten. Dabei stehen unterschiedliche Varianten zur Verfügung, die Sie mit einem der aufgelisteten `make`-Kommandos starten:

```
user$ cd /usr/src/linux-n.n
user$ make config           (Konfiguration im Textmodus)
user$ make menuconfig      (dialoggeführte Konfiguration im Textmodus)
user$ make nconfig         (dialoggeführte Konfiguration im Textmodus)
user$ make localmodconfig  (automatische Konfiguration für die aktuelle Hardware)
```

**make config** `make config` funktioniert immer, ist aber umständlich zu bedienen und nicht zu empfehlen. Sie müssen immer *alle* Optionen durchlaufen, auch wenn Sie nur eine einzige Option verändern möchten.

**make menuconfig** `make menuconfig` setzt voraus, dass Sie vorher das Paket `ncurses-devel` bzw. `libncurses5-dev` installiert haben. Die Konfiguration erfolgt ebenfalls im Textmodus. Der große Vorteil im Vergleich zu `make config` besteht darin, dass die Einstellung der unzähligen Optionen durch verschachtelte Dialoge strukturiert ist.

**make nconfig** Auch mit `make nconfig` erfolgt die Konfiguration im Textmodus, und wie bei `make menuconfig` müssen Sie vorher das `ncurses`-Entwicklerpaket installieren. Der wesentliche Unterschied besteht in der Navigation: Während `menuconfig` verschachtelte Dialoge verwendet, navigieren Sie bei `nconfig` durch einen Baum, dessen Äste Sie ein- und ausklappen (siehe [Abbildung 25.1](#)).

**make localmodconfig** `make localmodconfig` ist eine interessante Kompilervariante für alle, die es eilig haben. Dabei werden nur die Module kompiliert, die im gerade laufenden Kernel tatsächlich genutzt werden. Das hat Vor- und Nachteile: Der offensichtliche Vorteil besteht darin, dass wirklich nur der Teil des Kernelcodes übersetzt wird, der tatsächlich benötigt wird. Das kann die Übersetzungszeit auf ein Drittel senken! Allerdings läuft der so kompilierte Kernel auf einem anderen Rechner unter Umständen nicht, wenn für seine Hardware-Komponenten relevante Treiber fehlen. Auch das Nachladen eines Moduls, das zur Kompilierzeit nicht aktiv war, wird scheitern. Der Kernel ist also nur zu Testzwecken geeignet, nicht aber für eine längerfristige Nutzung. Detailinformationen zu dieser `make`-Variante können Sie hier nachlesen:

<http://heise.de/-1402386>

```

kofler@fedora25:virt/vanilla-kernel/linux-4.11.3
Datei Bearbeiten Ansicht Suchen Terminal Hilfe

.config - Linux/x86 4.11.3 Kernel Configuration

File systems
[ ] OCFS2 logging support
[ ] OCFS2 expensive checks
<M> Btrfs filesystem support
[*] Btrfs POSIX Access Control Lists
[ ] Btrfs with integrity check tool compiled in (DANGEROUS)
[ ] Btrfs will run sanity tests upon loading
[ ] Btrfs debugging support
[ ] Btrfs assert support
<M> NILFS2 file system support
<M> F2FS filesystem support
[*] F2FS Status Information
[*] F2FS extended attributes
[*] F2FS Access Control Lists
[*] F2FS Security Labels
[ ] F2FS consistency checking feature
[ ] F2FS Encryption
[ ] F2FS IO tracer

F1Help F2SymInfo F3Help 2 F4ShowAll F5Back F6Save F7Load F8SymSearch F9Exit

```

Abbildung 25.1 Kernelkonfiguration mit make nconfig

Wenn Sie nur einzelne Optionen an einer vorgegebenen Kernelkonfiguration ändern möchten, ist es am besten, auf `make xxxconfig` ganz zu verzichten. Stattdessen geben Sie die gewünschten Einstellungen in der getrennten Datei `config-local` an. Die hier gewählten Optionen haben Vorrang gegenüber `.config`.

Nur wenige  
Änderungen  
durchführen

## Kernel kompilieren und installieren

Nachdem Sie mit der Konfiguration des Kernels vermutlich einige Zeit verbracht haben, muss jetzt der Rechner arbeiten. Die folgenden Kommandos beschäftigen einen zeitgemäßen Rechner (SSD, 4 CPU-Cores) circa eine halbe Stunde. Die Option `-j N` gibt an, wie viele Prozesse `make` parallel starten soll.

```

user$ cd /pfad/zum/code/linux-n.n
user$ make -j 4 all           (alles kompilieren, 4 CPU-Cores)
user$ sudo make modules_install (Module installieren)

```

Das Ergebnis am Ende dieses Prozesses ist die Datei `bzImage` im Verzeichnis `/pfad/zum/code/linux-n.n/arch/x86/boot`. Die Größe der Datei liegt meist in der Größenordnung zwischen 4 und 7 MiB und hängt davon ab, wie viele Funktionen direkt in den Kernel inkludiert sind und wie viele als Module bzw. überhaupt nicht kompiliert wurden.

`make modules_install` kopiert die Moduldateien dorthin, wo die Kommandos zur Modulverwaltung (etwa `insmod`) diese erwarten: in das Verzeichnis `/lib/modules/n.n`. Dabei ist `n.n` die genaue Versionsnummer des soeben kompilierten Kernels.

### Fehler beim Kompilieren

Wenn beim Kompilieren ein Fehler auftritt, sollten Sie naturgemäß versuchen, ihm auf den Grund zu gehen. Wenn das Problem bei einer Funktion auftritt, die für Sie nicht wichtig ist, können Sie die Konfiguration so ändern, dass die betroffene Funktion eben nicht kompiliert wird.

Hartgesottene Linux-Freaks können `make` einfach mit der zusätzlichen Option `-k` aufrufen, also z. B. `make -k all`. Diese Option bewirkt, dass Fehler ignoriert werden. `make` fährt also einfach mit der Kompilation der nächsten Datei fort. Wenn Sie Glück haben, betrifft das Kompilationsproblem ein für Sie unwichtiges Modul, das dann eben nicht zur Verfügung steht.

### Kernel installieren

Der frisch erzeugte neue Kernel ist natürlich noch nicht aktiv. Bisher wurden nur ein paar neue Dateien erstellt, sonst nichts! Der neue Kernel kann erst beim nächsten Start von Linux aktiviert werden und auch dann nur, wenn Sie Ihren Bootloader GRUB so konfigurieren, dass der neue Kernel berücksichtigt wird.

Dazu kopieren Sie als Erstes die neue Kerneldatei in das Verzeichnis `/boot`. Es ist üblich, der Datei den Namen `vmlinuz-n.n` zu geben. Gleichzeitig sollten Sie auch eine Kopie der Konfigurationsdatei erstellen:

```
root# cp /pfad/zum/code/linux-n.n/arch/x86/boot/bzImage /boot/vmlinuz-n.n
root# cp /pfad/zum/code/linux-n.n/.config /boot/config-n.n
```

### Systemstart vorbereiten

Nun müssen Sie eine neue, zum Kernel passende `Initrd`-Datei erzeugen. Dazu verwenden Sie je nach Distribution die Kommandos `dracut`, `mkinitrd`, `mkinitramfs` oder `update-initramfs`. Außerdem müssen Sie die GRUB-Konfiguration aktualisieren, damit es im GRUB-Menü einen Eintrag für den neuen Kernel gibt. Das erforderliche Kommando lautet je nach Distribution `update-grub` oder `grub2-mkconfig`. Details zur Erzeugung von `Initrd`-Dateien und zur Aktualisierung der GRUB-Konfiguration können Sie in [Kapitel 23](#), »GRUB«, nachlesen. Die folgenden Kommandos gelten für ein Fedora-System auf einem EFI-Rechner:

```
root# dracut /boot/initramfs-n.n.img n.n
root# grub2-mkconfig -o /etc/grub2-efi.cfg
```

Ob alles funktioniert hat, merken Sie beim Neustart. Sollte der neue Kernel aus irgendeinem Grund nicht funktionieren, starten Sie den Rechner einfach mit dem bisherigen Kernel und unternehmen einen weiteren Versuch, den Kernel richtig zu

konfigurieren und neu zu kompilieren. Läuft der neue Kernel dagegen zufriedenstellend, sollten Sie die nun nicht mehr benötigten Objekt-Dateien des Compilers aufräumen. Sie gewinnen auf diese Weise rund 13 GiB Platz auf der Festplatte!

```
root# cd /pfad/zum/code/linux-n.n
root# make clean
```

## 25.5 Kernel-Neustart mit kexec

Nach einem Update oder nach dem Kompilieren eines eigenen Kernels ist es üblich, den neuen Kernel durch einen Neustart des Rechners zu aktivieren. Alternativ können Sie den Kernel aber auch über die `kexec`-Funktion neu starten. Dann übergibt der laufende Kernel die Kontrolle an eine andere Kernel-Version.

Im Vergleich zu einem echten Reboot ist ein Kernel-Neustart mit `kexec` um ein paar Sekunden schneller. Insbesondere entfallen die BIOS/EFI-Initialisierung und das Durchlaufen des GRUB-Prozesses. Die Downtime ist entsprechend um ein paar Sekunden kürzer, was vor allem Server-Administratoren freut. Dem steht allerdings der Nachteil entgegen, dass es unter Umständen Probleme bei der Neuinitialisierung diverser Hardware-Komponenten geben kann. Das klappt mitunter nur bei einem richtigen Neustart zuverlässig.

Bei vielen Linux-Distributionen ist das erforderliche `kexec`-Kommando standardmäßig installiert. Unter Debian und Ubuntu müssen Sie wie folgt nachhelfen:

**Installation  
(Debian/Ubuntu)**

```
root# apt install kexec-tools (Debian/Ubuntu)
```

Nach der Installation erscheint ein Konfigurationsdialog mit der Frage, ob `kexec` Neustarts verwalten soll. Wenn Sie die Frage mit JA beantworten, verwendet Ubuntu in Zukunft `kexec` bei allen Reboots. Nach meinen Erfahrungen funktioniert das durchaus nicht immer zuverlässig! Antworten Sie lieber mit NEIN, und probieren Sie `kexec` zuerst manuell aus. Wenn `kexec` auf Ihrer Hardware zuverlässig funktioniert, können Sie den Mechanismus später mit `dpkg-reconfigure kexec-tools` immer noch aktivieren. Beachten Sie, dass der Reboot-Mechanismus der `kexec-tools` auf einem Init-V-Script basiert. Es wird von `systemd` nur dank dessen Init-V-Kompatibilität ausgeführt.

### Das kexec-Kommando

Der Kernel-Neustart wird nun in zwei Schritten initiiert. Zuerst übergeben Sie an `kexec` den Ort der neuen Kerneldatei, den Ort der `initrd`-Datei sowie die gewünschten Kernel-Parameter. Dabei können Sie wahlweise eine Zeichenkette in der Form `--commandline="xxx"` übergeben oder einfach die bisher verwendeten Parameter mit

--reuse-cmdline wiederverwerten. Welche Parameter zuletzt an den Kernel übergeben wurden, können Sie der Datei `/proc/cmdline` oder der GRUB-Konfigurationsdatei (üblicherweise `/boot/grub/grub.cfg`) entnehmen.

```
root# kexec -l /boot/vmlinuz-4.4.0-72-generic \
          --initrd=/boot/initrd.img-4.4.0-72-generic --reuse-cmdline
```

Im zweiten Schritt aktivieren Sie den neuen Kernel – entweder unmittelbar mit `kexec -e` oder nachdem Sie via `systemd` alle laufenden Dienste heruntergefahren haben. Auf einem Server ist die zweite Variante unbedingt vorzuziehen! Nur sie stellt sicher, dass z. B. ein Datenbank-Server alle offenen Transaktionen zu Ende führen und alle Dateien ordnungsgemäß schließen kann.

```
root# kexec -e                (unmittelbarer Neustart)
root# systemctl isolate kexec (zuerst Dienste herunterfahren, dann Neustart)
```

Bei meinen Tests hat `kexec` auf realer Hardware sowohl bei Desktop- als auch bei Server-Installationen überraschend gut funktioniert. Gescheitert ist der Kernel-Start allerdings bei einer Ubuntu-Server-Installation in VirtualBox. In der gleichen VirtualBox-Version ließ sich CentOS aber problemlos neu starten – es kann sich also nicht um ein generelles Problem handeln.

## 25.6 Kernel-Live-Patches

Die meisten Updates eines Linux-Systems können im laufenden Betrieb erfolgen. Aktualisierte Netzwerkdienste müssen zwar anschließend neu gestartet werden, aber es besteht keine Notwendigkeit, den ganzen Rechner neu zu starten. Die Ausnahme von dieser Regel ist der Kernel: Damit Sicherheits-Updates im Kernel wirksam werden, müssen Sie einen neuen Kernel und neue Module installieren und anschließend den ganzen Rechner oder via `kexec` den Kernel und alle Dienste neu starten.

Auf Desktop-Rechnern, die üblicherweise jeden Tag ein- und ausgeschaltet werden, ist das egal. Aber bei Servern, die möglichst ohne Unterbrechung ständig verfügbar sein sollen, ist ein Neustart zumeist unerwünscht. Und selbst Administratoren, die Updates automatisieren, scheuen in der Regel davor zurück, auch die erforderlichen Neustarts zu automatisieren. Zu groß ist die Gefahr, dass etwas schiefgeht und dann (zumeist mitten in der Nacht) keiner Zeit hat, das Problem zu beheben.

**Ksplice** Die erste Lösung für dieses Problem bot die Funktion *Ksplice*: Bei vielen Updates ist es möglich, die betreffende Kernelfunktion im laufenden Betrieb zu deaktivieren und durch neuen Code zu ersetzen.

Die nicht eben trivialen technischen Hintergründe des Verfahrens sind auf den folgenden Seiten beschrieben:

<https://kssplice.com>  
<https://lwn.net/Articles/340477>  
<https://en.wikipedia.org/wiki/Ksplice>

Mitte 2011 übernahm Oracle die Firma Ksplice. Kernel-Updates für Oracle Linux erfolgen seither zumeist durch Ksplice. Das war über mehrere Jahre ein durchaus gewichtiges Unterscheidungsmerkmal zu anderen Enterprise-Linux-Versionen, die keine vergleichbare Funktion anbieten konnten. Oracle bietet die Ksplice allerdings nur zahlenden Kunden an. Ksplice steht zwar als Open-Source-Code zur Verfügung, ist aber nicht Bestandteil des offiziellen Linux-Kernels.

Red Hat und SUSE wollten in dieser Hinsicht natürlich nicht zurückstecken und entwickelten unter den Namen *kPatch* und *kGraft* vergleichbare Update-Mechanismen. Die beiden Mechanismen stehen seit 2014 in den Enterprise-Versionen von Red Hat und SUSE zur Verfügung. Die für kPatch und kGraft erforderlichen Funktionen (gewissermaßen der größte gemeinsame Nenner) wurden von Linus Torvalds mit Version 4.0 (also Anfang 2015) in den offiziellen Linux-Kernel aufgenommen. Ob Ihr Kernel die Funktionen enthält, erkennen Sie am Vorhandensein des Verzeichnisses `/sys/kernel/livepatch`. Allerdings stellen auch Red Hat und SUSE geeignete Live-Kernel-Patches nur zahlenden Kunden zur Verfügung.

**kPatch und kGraft**

<https://github.com/dynup/kpatch>  
<https://en.wikipedia.org/wiki/KGraft>

Canonical ist gewissermaßen der Spätstarter im Live-Patching-Geschäft. Seit Ende 2016 bietet Canonical für kritische Sicherheitsprobleme in Ubuntu-LTS-Versionen Live-Patches im Rahmen des Landscape-Dienstes für kommerzielle Kunden an. Das Programm `canonical-livepatch` greift dabei auf die oben erwähnte Live-Patch-Infrastruktur im Kernel zurück. Details darüber, ob auf dieser Basis kPatch, kGraft oder ein eigenes Verfahren zum Einsatz kommt, hat Canonical nicht verraten.

**Ubuntu**

Erfreulicherweise stellt Canonical die Live Patches in beschränktem Umfang auch nichtkommerziellen Anwendern zur Verfügung: Sofern Sie über ein kostenloses Ubuntu-One-Konto verfügen, können Sie drei Rechner für den Live-Patch-Dienst anmelden. Das Einrichten des Live-Patch-Dienstes ist erfreulich einfach: Zuerst melden Sie sich bei Ubuntu One an und fordern dort ein Livepatch-Token an:

<https://auth.livepatch.canonical.com>

Anschließend führen Sie die folgenden Kommandos aus, um das Snap-Paket `canonical-livepatch` zu installieren (siehe auch [Abschnitt 19.12](#), »Flatpak und Snap«):

```
root# apt install snapd
root# snap install canonical-livepatch
root# canonical-livepatch enable <TOKEN>
```

Den Live-Patch-Status können Sie mit `canonical-livepatch status` ermitteln:

```
root# canonical-livepatch status
kernel: 4.4.0-72.93-generic
fully-patched: true
version: "22.4"
```

`dmesg | grep livepatch` liefert Meldungen über zuletzt durchgeführte Live-Patches:

```
root# dmesg | grep livepatch
...
livepatch: enabling patch 'kpatch_livepatch_Ubuntu_4_4_0_72_93_generic_22'
```

Sollten Sie zu einem späteren Zeitpunkt doch einen Neustart des Rechners durchführen, dann wird der aktuellste zuletzt installierte Kernel verwendet. Da es hierfür noch keine Patches gibt, sieht die `status`-Ausgabe dann wie folgt aus:

```
root# canonical-livepatch status
kernel: 4.4.0-78.99-generic
fully-patched: true
version: ""
```

#### Live-Patches nur für Notfälle

Den Code des laufenden Kernels zu verändern, ist ein diffiziler Vorgang (wenn Sie ein Freund dramatischer Vergleiche sind: wie die Operation am offenen Herzen).

Deswegen wird der Mechanismus aktuell von allen Distributoren nur für gefährliche Kernel-Sicherheitslücken verwendet. Bei sonstigen Korrekturen wird wie bisher ein neuer Kernel installiert, auf die Aktivierung der dort durchgeführten Änderungen via Live-Patches aber verzichtet.

Somit kann es trotz aktivierter Live-Patches sein, dass Ihre Distribution nach der Installation von (Kernel-)Updates meldet, dass ein Neustart erforderlich ist. Lassen Sie sich davon nicht verunsichern.

## 25.7 Die Verzeichnisse `/proc` und `/sys`

Die Verzeichnisse `/proc` und `/sys` werden während des Systemstarts in das Dateisystem eingebunden. Sie dienen dazu, Informationen über den Kernel, laufende Prozesse, geladene Module und viele andere Parameter auf eine transparente Art und Weise sichtbar zu machen.



Intern sind die Verzeichnisse /proc und /sys als virtuelle Dateisysteme realisiert. Sie enthalten also keine echten Dateien und beanspruchen daher auch keinen Platz auf der Festplatte. Das gilt auch für die scheinbar sehr große Datei /proc/kcore, die den Arbeitsspeicher abbildet.

Die meisten der /proc- und /sys-Dateien liegen im Textformat vor. Um die Dateien zu lesen, müssen Sie unter Umständen `cat` statt `less` verwenden, weil manche `less`-Versionen mit virtuellen Dateien nicht zurechtkommen.

Das /proc-Verzeichnis liefert eine Menge interner Kernelinformationen sowie Daten zu allen gerade laufenden Prozessen (siehe [Tabelle 25.4](#)). Unter anderem ist dort jedem Prozess ein eigenes Unterverzeichnis zugeordnet. Innerhalb des Prozessverzeichnisses befinden sich dann einige Dateien mit diversen Verwaltungsdaten (z. B. die zum Start verwendete Kommandozeile). Diese Daten werden von diversen Kommandos zur Prozessverwaltung (z. B. `top`, `ps` etc.) ausgewertet.

| Datei              | Bedeutung                                |
|--------------------|--|
| /proc/N/*          | Informationen zum Prozess mit der PID=N  |
| /proc/asound       | ALSA (Advanced Linux Sound Architecture) |
| /proc/bus/usb/*    | USB-Informationen                        |
| /proc/bus/pccard/* | PCMCIA-Informationen                     |
| /proc/bus/pci/*    | PCI-Informationen                        |
| /proc/cmdline      | an den Kernel übergebene Boot-Parameter  |
| /proc/config.gz    | Kernelkonfigurationsdatei (SUSE)         |
| /proc/cpuinfo      | CPU-Informationen                        |
| /proc/devices      | Nummern von aktiven Devices              |
| /proc/fb           | Informationen zum Frame-Buffer           |
| /proc/filesystems  | im Kernel enthaltene Dateisystemtreiber  |
| /proc/interrupts   | Nutzung der Interrupts                   |
| /proc/lvm/*        | Nutzung des Logical Volume Managers      |
| /proc/mdstat       | RAID-Zustand                             |
| /proc/modules      | aktive Module                            |
| /proc/mounts       | aktive Dateisysteme                      |
| /proc/net/*        | Netzwerkzustand und -nutzung             |
| /proc/partitions   | Partitionen der Festplatten              |

**Tabelle 25.4** Wichtige /proc-Dateien

| Datei         | Bedeutung  |
|---------------|--|
| /proc/scsi/*  | SCSI/SATA-Laufwerke und -Controller                |
| /proc/splash  | steuert das VGA-Hintergrundbild für Textkonsole 1. |
| /proc/sys/*   | System- und Kernelinformationen                    |
| /proc/uptime  | Zeit in Sekunden seit dem Rechnerstart             |
| /proc/version | Kernelversion                                      |

**Tabelle 25.4** Wichtige /proc-Dateien (Forts.)

Das /sys-Verzeichnis ist seit Kernelversion 2.6 verfügbar. Es enthält teilweise dieselben Informationen wie /proc, allerdings sind die Daten systematischer organisiert (siehe [Tabelle 25.5](#)). Das Ziel des /sys-Verzeichnisses ist es, den Zusammenhang zwischen dem Kernel und der Hardware abzubilden.

| Datei           | Bedeutung  |
|-----------------|--|
| /sys/block/*    | Informationen über alle Block-Devices (Festplatten etc.)             |
| /sys/bus/*      | Informationen über alle Bus-Systeme (PCI, SCSI, USB etc.)            |
| /sys/class/*    | Informationen über Device-Klassen (Bluetooth, Grafik, Speicher etc.) |
| /sys/devices/*  | Informationen über angeschlossene Hardware-Komponenten               |
| /sys/firmware/* | Informationen über Hardware-Treiber und -Firmware (speziell ACPI)    |
| /sys/kernel/*   | Informationen über den Kernel  |
| /sys/module/*   | Informationen über geladene Module                                   |
| /sys/power/*    | Informationen über die Energieverwaltung                             |

**Tabelle 25.5** Wichtige /sys-Dateien

## 25.8 Kernel-Boot-Optionen

Nicht immer, wenn ein Detail im Kernel geändert werden soll, muss der Kernel gleich neu kompiliert werden! Es gibt zwei Möglichkeiten, ohne ein Neukompilieren auf den Kernel Einfluss zu nehmen:

- ▶ Zum einen können Sie mit dem Bootloader während des Systemstarts Parameter an den Kernel übergeben. Dieser Mechanismus ist Thema dieses Abschnitts.
- ▶ Zum anderen können Sie eine Reihe von Kernelfunktionen dynamisch – also im laufenden Betrieb – verändern. Diese Art des Eingriffs ist insbesondere zur Steuerung von Netzwerkfunktionen gebräuchlich und wird in [Abschnitt 25.9](#), »Kernelparameter verändern«, beschrieben.

Bei der Konfiguration von GRUB können Sie in der Zeile `linux` bzw. in der Datei `/etc/default/grub` Kernel-Boot-Optionen angeben. Derartige Optionen können Sie auch interaktiv beim Start eines Linux-Installationsprogramms oder beim Start von GRUB über die Tastatur eintippen (siehe [Abschnitt 23.2](#), »GRUB-Bedienung (Anwendersicht)«). Die Syntax für die Angabe von Optionen sieht so aus:

```
linux /boot/vmlinuz-n.n optionA=parameter optionB=parameter1,parameter2
```

Die Parameter zu einer Option müssen ohne Leerzeichen angegeben werden. Mehrere Optionen müssen durch Leerzeichen voneinander getrennt werden, nicht durch Kommata. Hexadezimale Adressen werden in der Form `0x1234` angegeben. Ohne vorangestelltes `0x` wird die Zahl dezimal interpretiert.

Kernel-Boot-Optionen helfen oft dabei, Hardware-Probleme zu umgehen. Wenn der Linux-Kernel beispielsweise aufgrund eines fehlerhaften BIOS nicht erkennt, wie viel RAM Ihr Rechner hat, geben Sie den korrekten Wert mit dem Parameter `mem=` an.

Beachten Sie, dass die beim Linux-Start angegebenen Parameter nur Einfluss auf die in den Kernel integrierten Treiber haben! Parameter für Kernelmodule müssen dagegen in der Datei `/etc/modprobe.conf` bzw. in den Verzeichnissen `/etc/modprobe.d` oder `/etc/modules-load.d` angegeben werden.

Dieser Abschnitt beschreibt nur die wichtigsten Kernel-Boot-Optionen. Weitere Informationen erhalten Sie mit `man bootparam` sowie auf den folgenden Seiten:

<http://tldp.org/HOWTO/BootPrompt-HOWTO.html>

<https://www.kernel.org/doc/Documentation/admin-guide/kernel-parameters.txt>

## Wichtige Kernel-Boot-Optionen

- ▶ `root=/dev/sdb3`: Die `root`-Option gibt an, dass nach dem Laden des Kernels die dritte primäre Partition des zweiten SCSI/SATA-Laufwerks als Systempartition für das Root-Dateisystem verwendet werden soll. Analog können natürlich auch andere Laufwerke und Partitionen angegeben werden.

Wenn die Partition mit einem Label bezeichnet ist, kann die Systempartition auch in der Form `root=LABEL=xxx` angegeben werden. Insbesondere Fedora und Red Hat machen von dieser Möglichkeit Gebrauch. Als Name für die Systempartition wird üblicherweise das Zeichen `/` verwendet. Bei `ext`-Partitionen ermitteln Sie den Partitionsnamen mit `e2label` bzw. verändern ihn mit `tune2fs`.

Eine weitere Variante ist die Angabe der Systempartition durch `root=UUID=nnn`, wobei `nnn` die UUID der Festplattenpartition ist. Diese Identifikationsnummer ermitteln Sie mit `/lib/udev/vol_id` partition.

- ▶ `ro`: Die Option `ro` gibt an, dass das Dateisystem vorerst *read-only* gemountet werden soll. Das ist (in Kombination mit einer der beiden folgenden Optionen) praktisch, wenn ein defektes Dateisystem manuell repariert werden muss.
- ▶ `init`: Nach dem Kernelstart wird ein Programm gestartet, das je nach Distribution den Init-V-Prozess, Upstart oder `systemd` steuert (siehe [Kapitel 24](#), »Das Init-System«). Wenn Sie dies nicht wollen, können Sie mit der Option `init` ein anderes Programm angeben.

Mit `init=/bin/sh` erreichen Sie beispielsweise, dass eine Shell gestartet wird. Die Option kann Linux-Profis helfen, ein Linux-System wieder zum Laufen zu bringen, wenn bei der Init-Konfiguration etwas schiefgegangen ist. Beachten Sie, dass das `root`-Dateisystem nur *read-only* zur Verfügung steht. (Das können Sie mit `mount -o remount` ändern, siehe [Abschnitt 22.8](#), »Verwaltung des Dateisystems (mount und `/etc/fstab`)«.) Beachten Sie außerdem, dass in der Konsole das US-Tastaturlayout gilt und dass die `PATH`-Variable noch leer ist.

- ▶ `single` oder `emergency`: Wenn Sie eine der zwei obigen Optionen verwenden, startet der Rechner im Single-User-Modus. Genau genommen werden diese Optionen nicht vom Kernel ausgewertet, sondern so wie alle unbekanntenen Optionen an das erste vom Kernel gestartete Programm weitergegeben (siehe [Kapitel 24](#), »Das Init-System«).
- ▶ `initrd=name`: `initrd` gibt den Namen der zu ladenden Initial-RAM-Disk-Datei an. Wenn Sie *keine* Initrd-Datei verwenden möchten, geben Sie `initrd=` oder `noinitrd` an.
- ▶ `ipv6.disable=1`: Diese Option deaktiviert alle IPv6-Funktionen des Kernels.
- ▶ `reserve=0x300,0x20`: Diese Option gibt an, dass die 32 Bytes (hexadezimal `0x20`) zwischen `0x300` und `0x31F` von keinem Hardware-Treiber angesprochen werden dürfen, um darin nach irgendwelchen Komponenten zu suchen. Die Option ist bei manchen Komponenten notwendig, die auf solche Tests allergisch reagieren. Sie tritt im Regelfall in Kombination mit einer zweiten Option auf, die die exakte Adresse der Komponente angibt, die diesen Speicherbereich für sich beansprucht.
- ▶ `pci=bios|nobios`: Diese Option steuert, ob das BIOS zur Hardware-Erkennung der PCI-Komponenten verwendet werden soll oder nicht. (PCI ist das Bussystem zur Erweiterung von PCs durch Steckkarten.) Wenn die automatische Hardware-Erkennung durch den Kernel nicht funktioniert, hilft manchmal `pci=bios`.
- ▶ `pci=nommcconf`: Diese Option deaktiviert `MMCONFIG` für die PCI-Konfiguration. Das vermeidet Probleme bei manchen PCI-Express-Systemen.
- ▶ `quiet`: Diese Option bewirkt, dass während des Kernelstarts keine Meldungen auf dem Bildschirm dargestellt werden.

- ▶ `video=1024x768`: Mit dieser Option kann per *Kernel Mode Setting* (KMS) die gewünschte Grafikauflösung eingestellt werden, falls der Kernel nicht selbst die optimale Auflösung wählt, z. B. wenn das Video-Signal über einen KVM-Switch geleitet wird. Wenn Sie auch die Farbtiefe (z. B. 24 Bit) und die Bildfrequenz angeben möchten, sieht die Syntax so aus: `video=1280x800-24@60`

Die Einstellung der Grafikdaten funktioniert nur bei KMS-kompatiblen Treibern. Die `video`-Einstellung gilt normalerweise für alle angeschlossenen Monitore. Wenn Sie die Auflösung nur für einen einzelnen Monitor ändern möchten, geben Sie den entsprechenden Signalausgang an, z. B. `video=VGA-1:1024x768`.

- ▶ `nomodeset`: Diese Option deaktiviert das Kernel Mode Setting (KMS).

### SMP-Optionen

SMP steht für *Symmetric Multiprocessing* und bezeichnet die Fähigkeit des Kernels, mehrere CPUs bzw. CPU-Cores gleichzeitig zu nutzen. Sollten dabei Probleme auftreten, können die folgenden Optionen hilfreich sein:

- ▶ `maxcpus=1`: Wenn Sie bei einem Multiprozessorsystem Boot-Probleme haben, können Sie mit dieser Option die Anzahl der genutzten Prozessoren auf 1 reduzieren. Der Wert 0 entspricht der Option `nosmp`.
- ▶ `nosmp`: Diese Option deaktiviert die SMP-Funktionen. Der Kernel nutzt nur eine CPU.
- ▶ `noht`: Diese Option deaktiviert die Hyper-Threading-Funktion. (Dank Hyper-Threading verhalten sich manche CPUs so, als stünden mehrere Cores zur Verfügung. Daraus ergibt sich eine etwas höhere Rechenleistung, wenngleich die Steigerung nicht so hoch ist wie bei echtem SMP.)
- ▶ `noapic`: APIC steht für *Advanced Programmable Interrupt Controller* und bezeichnet ein Schema, um Hardware-Interrupts an die CPUs weiterzuleiten. Bei aktuellen Kernelversionen wird APIC immer aktiviert. Wenn Sie Probleme mit APIC vermuten, verhindern Sie durch `noapic`, dass der Kernel den lokalen APIC aktiviert bzw. nutzt.
- ▶ `noapic`: Diese Option reicht etwas weniger weit als `noapic` und deaktiviert nur den IO-Teil von APIC.
- ▶ `lapic`: Diese Option aktiviert APIC explizit. Das ist dann notwendig, wenn APIC durch das BIOS deaktiviert ist, aber dennoch genutzt werden soll.

## ACPI-Optionen

Das Energieverwaltungssystem ACPI (*Advanced Configuration and Power Interface*) ist nicht nur für das Ein- und Ausschalten verantwortlich, sondern auch für den sparsamen Umgang mit Energie, für die Verwaltung verschiedener Hibernate-Modi etc. Im Folgenden sind die wichtigsten Optionen zur Steuerung der ACPI-Funktionen des Kernels zusammengefasst:

- ▶ `acpi=on/off`: Diese Option (de)aktiviert die ACPI-Funktionen im Kernel.
- ▶ `acpi=oldboot`: Damit werden die ACPI-Funktionen nur während des Boot-Vorgangs genutzt. Sobald der Rechner läuft, werden die ACPI-Funktionen aber nicht mehr verwendet.
- ▶ `pci=noacpi`: Diese Option deaktiviert die Interrupt-Zuweisungen durch ACPI.
- ▶ `noresume`: Diese Option bewirkt, dass vorhandene Hibernate-Daten in der Swap-Partition ignoriert werden. Sie ist also dann sinnvoll, wenn der Rechner nicht mehr richtig aufwacht, z. B., weil die Hibernate-Daten defekt sind.

## 25.9 Kernelparameter verändern

Eine Menge Parameter des Kernels können im laufenden Betrieb über das `/proc`-Dateisystem verändert werden. Das folgende Beispiel zeigt, wie Sie die Masquerading-Funktion aktivieren, um den Rechner als Internet-Gateway für andere Rechner einzusetzen:

```
root# echo 1 > /proc/sys/net/ipv4/ip_forward
```

`sysctl` Einen eleganteren Weg bietet das Kommando `sysctl`, das mit den meisten aktuellen Distributionen mitgeliefert wird. Das analoge Kommando, um das Masquerading wieder abzuschalten, würde so aussehen:

```
root# sysctl -w net.ipv4.ip_forward=1
```

`sysctl -a` liefert eine Liste aller Kernelparameter zusammen mit ihren aktuellen Einstellungen. Mit `sysctl -p` können die in einer Datei gespeicherten `sysctl`-Einstellungen aktiviert werden. Als Dateiname wird üblicherweise `/etc/sysctl.conf` verwendet. Die Syntax ist in der Manual-Seite zu `sysctl.conf` beschrieben. Viele Distributionen sehen vor, dass diese Datei während des Init-Prozesses automatisch ausgewertet und ausgeführt wird.

TEIL VI  
**LAN-Server**





# Kapitel 26

## Netzwerkconfiguration

Dieses Kapitel beschreibt, wie Sie Ihren Linux-Rechner mit dem Internet bzw. mit einem lokalen Netzwerk verbinden. Bei Desktop-Installationen gelingt dies zumeist vollkommen mühelos mit dem NetworkManager. Schwieriger ist die Netzwerkconfiguration aber bei Server-Installationen, wo häufig kein DHCP-Server genutzt und daher eine manuelle Konfiguration durchgeführt werden muss. Deswegen geht dieses Kapitel auf die Grundlagen der Netzwerkconfiguration ein (inklusive IPv6) und erklärt, wie Sie eine statische Netzwerkconfiguration ohne Konfigurationswerkzeuge durchführen.

### 26.1 Der NetworkManager

Der NetworkManager ist das populärste Werkzeug zur LAN-, WLAN-, ADSL-, UMTS- und VPN-Konfiguration auf Desktop-Systemen. Für die Grundfunktionen ist ein Hintergrundprozess verantwortlich, der beim Hochfahren des Rechners gestartet wird. Die Benutzeroberfläche des NetworkManagers sieht je nach Desktop unterschiedlich aus. Ich konzentriere mich in diesem Abschnitt auf die Gnome-Variante. Unter KDE oder bei älteren Ubuntu-Versionen mit Unity sehen die Dialoge ein wenig anders aus, die Bedienung erfolgt aber nach denselben Mustern wie unter Gnome.

Der NetworkManager funktioniert nur, wenn das Programm die Kontrolle über die Schnittstellen hat. Alle gängigen Distributionen sollten eine entsprechende Konfiguration standardmäßig durchführen.

Voraussetzungen

Bei Debian und Ubuntu stellen Sie sicher, dass `/etc/network/interfaces` nur Einstellungen für die Loopback-Schnittstelle enthält. Schnittstellen, die vom NetworkManager gesteuert werden sollen (typischerweise `ethN`, `enpXXX`, `wlanN` bzw. `wlpXXX`), dürfen nicht durch diese Datei konfiguriert werden!

```
# Datei /etc/network/interfaces (Debian, Ubuntu)
auto lo
iface lo inet loopback
```

Bei Fedora und Red Hat darf die Datei `/etc/sysconfig/network-scripts/ifcfg-name` *nicht* den Eintrag `NM_CONTROLLED=no` enthalten. Dabei ist `name` der Device-Name der Netzwerkschnittstelle, z. B. `enpNsM` oder `wlpNsM`. Die Defaulteinstellung dieser Option lautet `yes`, d. h., wenn der Parameter ganz fehlt, ist auch alles in Ordnung.

```
# Datei /etc/sysconfig/network-scripts/ifcfg-xxx (Fedora, Red Hat)
...
NM_CONTROLLED="yes"
```

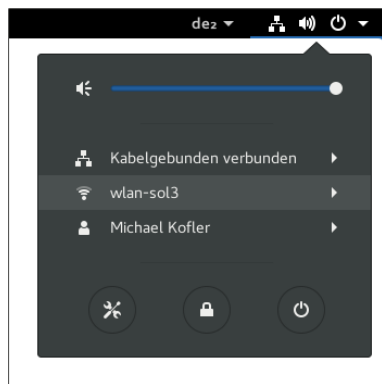
Bei SUSE kontrolliert das YaST-Modul `HARDWARE • NETZWERKEINSTELLUNGEN`, wie die Netzwerkkonfiguration erfolgt. Damit der NetworkManager verwendet werden kann, muss im Dialogblatt `GLOBALE OPTIONEN` die Option `BENUTZERGESTEUERT MIT HILFE VON NETWORKMANAGER` aktiviert sein. Das ist nur bei Notebooks standardmäßig der Fall. Die Alternativen sind `WICKED SERVICE` (ein SUSE-spezifischer Dienst zur Netzwerksteuerung) oder `NETWORKMANAGER DEAKTIVIERT` (manuelle Konfiguration).

#### NetworkManager deaktivieren

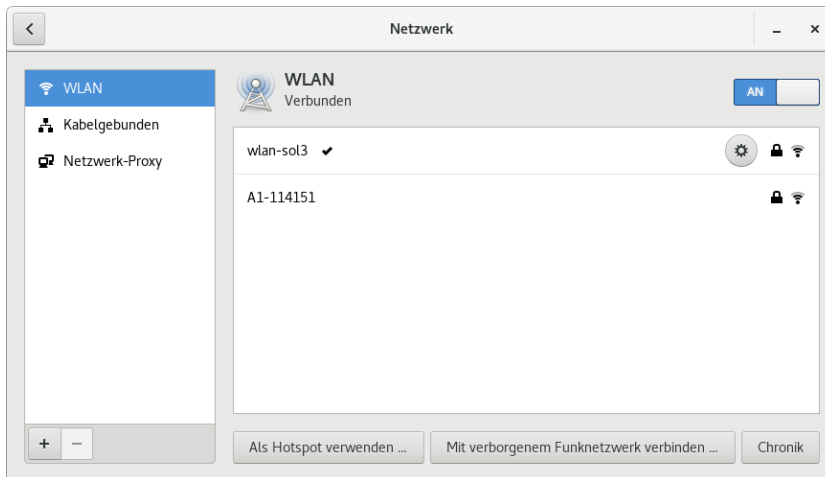
Wenn Sie Ihren Rechner als Server oder Router konfigurieren, sollten Sie den NetworkManager deaktivieren und eine statische Netzwerkkonfiguration durchführen. Tipps dazu finden Sie am Beginn von [Abschnitt 26.6](#), »Distributionsspezifische Konfigurationsdateien«.

### Konfiguration

Bei den meisten gängigen Distributionen zeigt ein Icon in der Menüleiste oder im Panel den aktuellen Netzwerkzustand an. Dieses Icon führt in ein Menü, das die aktive Verbindung und alle erreichbaren bzw. vorkonfigurierten Netzwerke auflistet (siehe [Abbildung 26.1](#)). Über dieses Menü oder im Modul `NETZWERK` der Systemeinstellungen können Sie bei Bedarf die Konfiguration verändern (siehe [Abbildung 26.2](#)).



**Abbildung 26.1** Im GNOME-Menü werden die LAN- und WLAN-Verbindungen des NetworkManagers angezeigt.



**Abbildung 26.2** Der Einstellungsdialog des NetworkManagers befindet sich unter Gnome im Modul »Netzwerk« der Systemeinstellungen.

Der einfachste Anwendungsfall für den NetworkManager liegt dann vor, wenn Ihr Rechner über ein Netzwerkkabel mit einem lokalen Netzwerk oder einem ADSL-Router verbunden ist. Der NetworkManager überprüft standardmäßig für alle LAN-Schnittstellen, ob via DHCP Konfigurationsinformationen bezogen werden können. Gelingt dies, erfolgt die Netzwerkkonfiguration bereits während des Rechnerstarts vollautomatisch.

Lokale Netzwerke  
mit DHCP

### Netzwerkschnittstellen explizit aktivieren

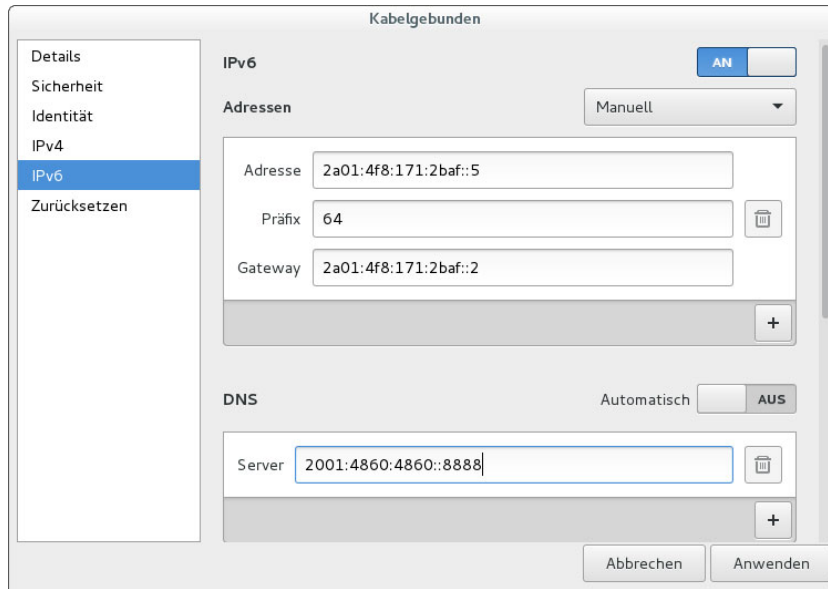
In der Vergangenheit hatte ich es immer wieder mit Distributionen zu tun, bei denen der automatische Verbindungsaufbau über ein Ethernet-Kabel scheiterte. Recht leicht passiert dies bei CentOS/RHEL, dessen Installationsprogramm eine leicht zu übersehende Option vorsieht, ob automatisch eine Netzwerkverbindung hergestellt werden soll. Die übervorsichtige Defaulteinstellung lautet KEINE AUTOMATISCHE VERBINDUNG.

Dieser Konfigurationsmangel kann zum Glück später leicht korrigiert werden: Dazu führen Sie im Menü des NetworkManagers VERBINDUNGEN BEARBEITEN aus und suchen im Konfigurationsdialog nach der Verbindung SYSTEM ETH0. Bei dieser Verbindung öffnen Sie mit BEARBEITEN den Detaildialog und aktivieren die Option AUTOMATISCH VERBINDEN.

Eine statische Konfiguration der LAN-Verbindung ist dann erforderlich, wenn Ihr Rechner nicht mit einem Router bzw. DHCP-Server verbunden ist und Sie die IP-Adresse, Netzmaske, Gateway-Adresse und die Nameserver-Adresse selbst ange-

Statische LAN-  
Konfiguration

ben müssen (siehe [Abbildung 26.3](#)). Alle hier aufgezählten Begriffe werden in [Abschnitt 26.3](#), »Netzwerkgrundlagen und Glossar«, erläutert.



**Abbildung 26.3** Statische IPv6-Konfiguration einer Linux-Installation in einer virtuellen Maschine

Zur Konfiguration führen Sie im Menü des NetworkManagers das Kommando NETZWERKEINSTELLUNGEN aus. Im Einstellungsdialog wählen Sie das Dialogblatt KABELGEBUNDEN, wählen dort die Schnittstelle aus und verändern über den Zahnrad-Button deren IPv4-Einstellungen. Für eine statische Konfiguration müssen Sie das Feld ADRESSEN auf MANUELL stellen.

#### WLAN-Zugang einrichten

Der NetworkManager erkennt selbstständig alle in Reichweite befindlichen Funknetze. Wenn Sie den Namen des WLANs zum ersten Mal im Menü des NetworkManagers auswählen, müssen Sie das WLAN-Passwort angeben.

Beim erstmaligen Verbindungsaufbau in Unternehmensnetzen mit dem Verschlüsselungssystem WPA & WPA2 ENTERPRISE müssen Sie im Einstellungsdialog des NetworkManagers ein CA-Zertifikat auswählen. Geeignete Zertifikate befinden sich im Verzeichnis `/etc/ssl/certs`. Welches Zertifikat erforderlich ist, hängt von der Konfiguration des Unternehmensnetzwerks ab – fragen Sie gegebenenfalls den Administrator der Firma.

In Zukunft stellt der NetworkManager die Verbindung dann selbstständig her. Dazu werden alle WLAN-Passwörter zentral gespeichert, wobei der Speicherort je nach Distribution unterschiedlich ist. Unter Fedora richtet der NetworkManager für

jede WLAN-Verbindung eine Textdatei in `/etc/sysconfig/network-scripts` ein. Unter Ubuntu werden die WLAN-Passwörter hingegen im Verzeichnis `/etc/NetworkManager/system-connections/*` gespeichert.

Ein WLAN kann so konfiguriert sein, dass es seinen Namen nicht sendet. In diesem Fall wird es im Menü des NetworkManagers nicht angezeigt. Um dennoch eine Verbindung herzustellen, klicken Sie im Einstellungsdialog auf `MIT EINEM VERBORGENEN FUNKNETZWERK VERBINDEN`. Anschließend können Sie den Netzwerknamen (ESSID = Extended Service Set Identification) und die Verschlüsselungstechnik selbst angeben.

Wenn Ihr Rechner den Internetzugang über eine LAN-Schnittstelle bezieht und außerdem einen WLAN-Controller besitzt, können Sie diesen mit dem NetworkManager so konfigurieren, dass Ihr Rechner jetzt als Hotspot fungiert und anderen WLAN-Geräten in Funkreichweite Internetzugang verschafft. Bei Smartphones wird dieses Verfahren oft als »Tethering« bezeichnet, in der Netzwerktechnik ist eher von einem WLAN-Access-Point die Rede.

Einen eigenen Hot-Spot einrichten



Abbildung 26.4 Der WLAN-Adapter wurde als Hotspot konfiguriert.

Die Konfiguration ist einfach: Im WLAN-Dialogblatt der Netzwerkeinstellungen klicken Sie auf den Button `ALS HOTSLOT VERWENDEN`. Nach einer Rückfrage zeigt der NetworkManager den Namen des Netzwerks und ein zufälliges Passwort an (siehe [Abbildung 26.4](#)). Weitere Konfigurationsmöglichkeiten bestehen nicht. Die Hotspot-Funktion funktioniert nur mit WLAN-Controllern, deren Linux-Treiber den sogenannten Ad-hoc-Modus unterstützt. Bei in Notebooks üblichen WLAN-Adapttern ist das leider nicht immer der Fall.

### Virtual Private Networks

Sie können mit dem NetworkManager auf einer bereits bestehenden Netzwerkverbindung aufbauend eine VPN-Konfiguration durchführen. VPN steht für *Virtual Private Network* und ist eine Technik, um Netzwerkpakete verschlüsselt in ein anderes Netzwerk zu übertragen. Das ermöglicht z. B. einen sicheren Zugang ins Firmennetz auch aus einem unsicheren Hotel-WLAN heraus.

Eine neue VPN-Verbindung richten Sie ein, indem Sie im Dialogblatt NETZWERK der Gnome-Systemeinstellungen auf den Plus-Button klicken und dann eines der unterstützten Verfahren auswählen (Cisco AnyConnect, OpenVPN, PPTP etc.). Damit gelangen Sie in einen für das jeweilige Verfahren spezifischen Dialog (siehe [Abbildung 26.5](#)). Lassen Sie sich von der Fülle der Optionen nicht irritieren – oft reicht es aus, ganz wenige Felder auszufüllen. Nicht im Dialog enthalten sind mitunter Felder für den Login-Namen und das Passwort. Nach diesen Informationen fragt der NetworkManager erst, wenn die Verbindung tatsächlich hergestellt wird.

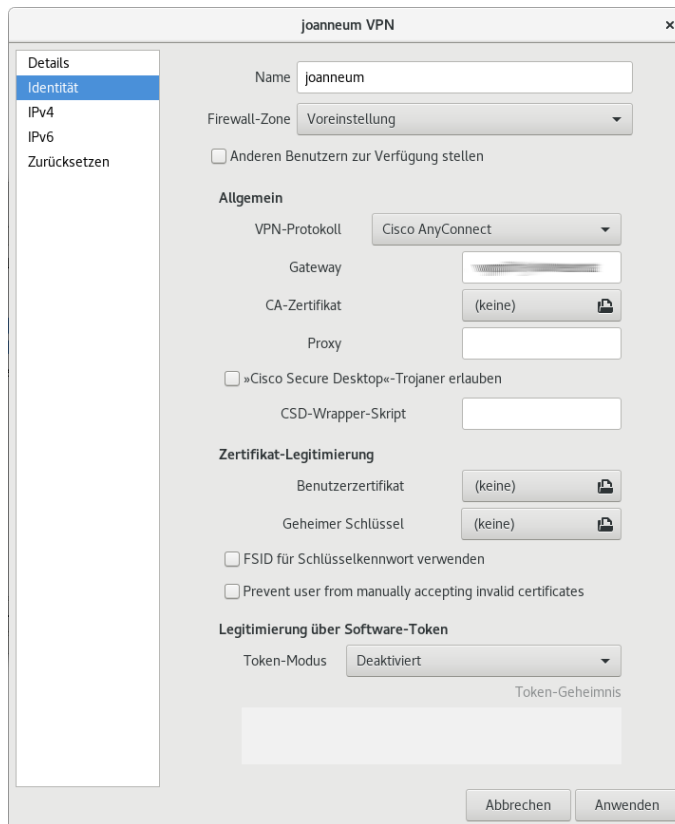


Abbildung 26.5 VPN-Konfiguration für Cisco AnyConnect

## Interna

### NetworkManager auf Kommandoebene

Sie können alle Einstellungen des NetworkManagers auch über das Kommando `nmccli` verwalten. Das ermöglicht die Steuerung des NetworkManagers durch Scripts oder bei Server-Installationen ohne grafische Benutzeroberfläche. Die folgenden Kommandos listen zuerst alle Schnittstellen auf, die der NetworkManager kennt, und deaktiviert dann die WLAN-Schnittstelle `wlp2s0`:

```

root# nmcli dev
GERÄT      TYP      STATUS      VERBINDUNG
virbr0     bridge  verbunden   virbr0
enp4s0     ethernet verbunden   enp4s0
wlp0s29u1u8 wifi     verbunden   Hotspot
lo         loopback nicht verwaltet --
virbr0-nic tun      nicht verwaltet --
root# nmcli dev disconnect wlp2s0

```

Ausführliche (d. h. seitenlange) Informationen zu allen Verbindungsparametern aller Schnittstellen gibt das Kommando `nmcli dev show`.

Es ist möglich, beim Herstellen bzw. Auflösen einer Verbindung automatisch Scripts auszuführen. Diese Scripts müssen im Verzeichnis `/etc/NetworkManager/dispatcher.d/` eingerichtet werden. Details und Beispiele zu diesem Mechanismus finden Sie auf der folgenden Seite:

Dispatcher

<https://wiki.ubuntuusers.de/NetworkManager/Dispatcher>

Unter Ubuntu ist der NetworkManager so konfiguriert, dass automatisch das Programm `Dnsmasq` als Nameserver-Cache startet. `Dnsmasq` merkt sich die IP-Adressen von Hostnamen und ermöglicht so eine raschere Adressauflösung. Eine detaillierte Beschreibung von `Dnsmasq` folgt in [Abschnitt 27.6](#), »`Dnsmasq` (DHCP- und Nameserver)«.

Nameserver  
(Ubuntu)

Der NetworkManager erzeugt selbst eine Konfigurationsdatei für `Dnsmasq` und speichert diese unter dem Namen `/var/run/NetworkManager/dnsmasq.conf`. In vielen Fällen ist die Datei allerdings einfach leer – dann läuft `dnsmasq` mit den Defaulteinstellungen. Um herauszufinden, auf welche Nameserver `dnsmasq` seinerseits zurückgreift, führen Sie am besten das folgende Kommando aus:

```

user$ nmcli dev show | grep DNS
nmcli dev show | grep DNS
IP4.DNS[1]: 10.0.0.138
IP4.DNS[2]: 8.8.8.8

```

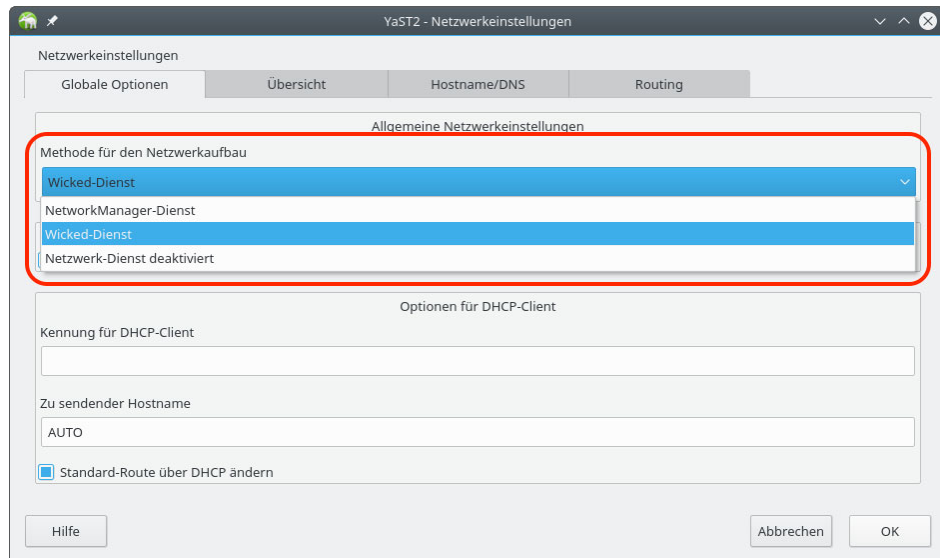
Falls Sie Sicherheitsbedenken wegen der Ausführung des lokalen Nameservers haben oder wenn es ohnedies einen Nameserver im lokalen Netzwerk gibt, können Sie den automatischen Start von `Dnsmasq` verhindern. Dazu entfernen Sie aus der Datei `/etc/NetworkManager/NetworkManager.conf` die Anweisung `dns=dnsmasq`. Hintergrundinformationen zur Ubuntu-spezifischen Konfiguration können Sie hier nachlesen:

<https://stgraber.org/2012/02/24/dns-in-ubuntu-12-04>

## Alternativen zum NetworkManager

**YaST (SUSE)** Bei SUSE entscheidet YaST während der Installation über das Standardverfahren zur Netzwerkkonfiguration. Auf Notebooks kommt der NetworkManager zum Einsatz, während auf Desktop-PCs und Servern standardmäßig die SUSE-Eigenentwicklung `wicked` läuft. Dieser Dämon ist speziell für Server optimiert, bei denen sich (oft virtuelle) Netzwerkverbindungen häufig ändern.

Selbstverständlich können Sie das Verfahren später ändern – und zwar im YaST-Modul `SYSTEM • NETZWERKEINSTELLUNGEN` im Dialogblatt `GLOBALE OPTIONEN` (siehe [Abbildung 26.6](#)). Wenn Sie hier `WICKED` einstellen, listet YaST im Dialogblatt `ÜBERSICHT` alle gefundenen Netzwerkschnittstellen auf. Sie können die einzelnen Schnittstellen nun bearbeiten. Standardmäßig konfiguriert YaST alle Ethernet-Schnittstellen so, dass diese ihre IP-Adresse und alle anderen Konfigurationseinstellungen per DHCP vom Router des lokalen Netzwerks beziehen.



**Abbildung 26.6** Einstellung des Verfahrens zur Netzwerkkonfiguration mit YaST

Unabhängig vom gewählten Konfigurationsverfahren können Sie in YaST mit `SYSTEM • NETZWERKEINSTELLUNGEN • HOSTNAME/DNS` den Hostnamen einstellen. Das ist häufig erforderlich, weil die Angabe des Hostnamens während der SUSE-Installation nicht vorgesehen ist. Stattdessen verwendet SUSE einen zufällig generierten Namen, der sich entsprechend schwer merken lässt.



## 26.2 Proxy-Konfiguration

In manchen Firmen bzw. Organisationen muss der HTTP-Verkehr über Proxy-Server fließen, auch wenn dies angesichts immer häufigerer HTTPS-Verbindungen eigentlich nicht mehr zeitgemäß ist. Der NetworkManager sieht aktuell keine Proxy-Konfiguration vor. Das Dialogblatt NETZWERK-PROXY in den Gnome-Systemeinstellungen (siehe [Abbildung 26.7](#)) speichert die Proxy-Einstellungen vielmehr in der Gnome-spezifischen dconf-Datenbank. Sie haben die Wahl zwischen einer automatischen oder einer manuellen Konfiguration. Standardmäßig ist der Proxy nicht aktiv.

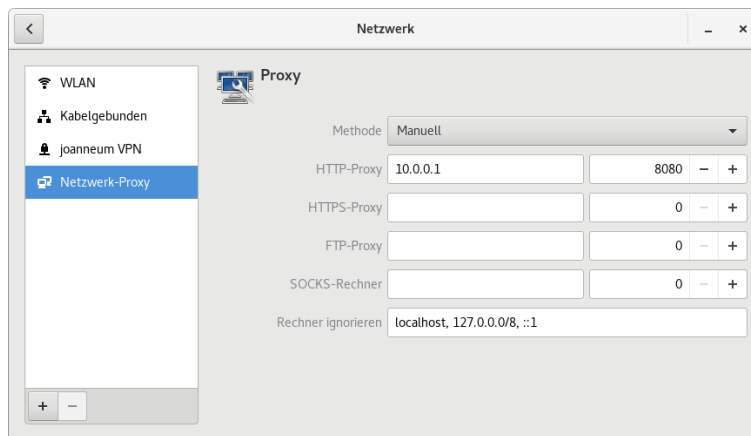


Abbildung 26.7 Proxy-Konfiguration in den Gnome-Systemeinstellungen

Nach dem Login in eine Gnome-Session werden dann die entsprechenden Umgebungsvariablen wie `http_proxy` und `https_proxy` gesetzt. Damit die geänderte Proxy-Konfiguration wirksam wird, müssen Sie sich ab- und neu anmelden.

Die Gnome-Proxy-Einstellungen können Sie auch mit dem folgenden Kommando auslesen:

```
user$ gsettings list-recursively org.gnome.system.proxy
...
org.gnome.system.proxy.http host '10.0.0.1'
org.gnome.system.proxy.http port 8080
```

KDE, Unity sowie einige andere Desktop-Systeme bieten vergleichbare Konfigurationsmöglichkeiten wie Gnome. Um die Proxy-Einstellungen losgelöst von einer Desktop-Umgebung zu konfigurieren, tragen Sie am einfachsten die folgenden Zeilen in die Datei `/etc/environment` ein:

```
# Datei /etc/environment
http_proxy=http://10.0.0.1:8080
https_proxy=http://10.0.0.1:8443
```

**Manuelle  
Konfiguration**

Weitere Details zur manuellen Proxy-Konfiguration finden Sie hier:

[https://wiki.archlinux.org/index.php/Proxy\\_settings](https://wiki.archlinux.org/index.php/Proxy_settings)

**Webbrowser** Alle gängigen Webbrowser sollten die Proxy-Konfiguration automatisch übernehmen. Gegebenenfalls können Sie die Proxy-Konfiguration aber auch direkt in den Einstellungsdialogen von Firefox oder Chrome ändern.

**Paketverwaltung** Damit auch die Paketverwaltungswerkzeuge den Proxy berücksichtigen, müssen unter Umständen auch deren Konfigurationsdateien geändert werden. Unter Debian und Ubuntu müssen Sie den Parameter `Acquire::http::Proxy` in einer `apt`-Konfigurationsdatei einstellen:

```
# beispielsweise in /etc/apt/apt.conf.d/01-proxy.conf
Acquire::http::Proxy "http://hostname-or-ipaddress:8080";
```

Unter CentOS, RHEL bzw. Fedora müssen Sie in `/etc/yum.conf` bzw. in `/etc/dnf/dnf.conf` die folgenden Zeilen einfügen:

```
# in /etc/dnf/dnf.conf bzw. /etc/yum.conf
proxy=http://hostname-or-ipaddress:8080
# nur falls erforderlich
proxy_username=name
proxy_password=pw
```

## 26.3 Netzwergrundlagen und Glossar

Dieser Abschnitt fasst die Grundlagen der Netzwerkkonfiguration zusammen. Die Informationen gelten nicht nur für den Netzwerkanschluss an ein lokales Netz, sondern auch für die WLAN-, ADSL- und Modemkonfiguration.

### Glossar

#### IP-Adressen und -Ports

Für den Großteil des Datenverkehrs in lokalen Netzen und im Internet ist das Protokoll TCP/IP verantwortlich (siehe auch [Tabelle 26.1](#)). Dabei werden Netzwerkdaten in Form von relativ kleinen Paketen transportiert. Zusammen mit jedem Paket werden mehrere Metadaten gespeichert, darunter die IP-Adresse und der IP-Port. Die IP-Adresse bestimmt den Empfänger des Pakets. Eine typische IP-Adresse für einen Rechner in einem lokalen Netz lautet 192.168.0.75. Die Port-Nummer gibt die Kategorie des Dienstes an. Vielen Internetdiensten (wie WWW oder E-Mail) sind jeweils eigene Port-Nummern zugeordnet. Eine Referenz aller Port-Nummern finden Sie in der Datei `/etc/services`, eine Zusammenstellung der wichtigsten Ports in [Tabelle 37.2](#) in [Kapitel 37](#), »Firewalls«.

| Abkürzung                                | Funktion   |
|--|--|
| IP = Internet Protocol                   | verbindungsloses Protokoll, Basis für TCP und UDP                |
| TCP = Transmission Control Protocol      | Ende-zu-Ende-Verbindung zwischen zwei Computern/Geräten          |
| UDP = User Datagram Protocol             | minimales, verbindungsloses Protokoll                            |
| ICMP = Internet Control Message Protocol | Austausch von IP-Status- und Fehlermeldungen                     |
| PPP = Point-to-Point Protocol            | IP-Verbindungsaufbau über Wählleitungen, z. B. für ADSL und UMTS |

**Tabelle 26.1** Wichtige Netzwerkprotokolle

IP-Adressen mögen für Computer praktisch sein, Menschen können sich IP-Adressen aber nur schwer merken. Aus diesem Grund werden Rechner im Netzwerk durch eine Kombination aus Host- und Domainnamen identifiziert. Beim Hostnamen handelt es sich um den eigentlichen Rechnernamen. Der Domainname bezeichnet das Teilnetz, innerhalb dessen der Rechner angesprochen werden kann. In lokalen Netzen können Sie den Domainnamen frei wählen; üblich ist z. B. `local`. Der Domainname kann auch mehrteilig sein.

Host- und  
Domainname

Wenn Ihr Linux-Rechner als öffentlich im Internet sichtbarer Server agieren soll, müssen Sie den gewünschten Domainnamen bei einem Internet Service Provider bzw. einem Network Information Center (kurz NIC) registrieren – z. B. bei <http://www.denic.de> für die `.de`-Domainnamen oder bei <http://www.corenic.org> für `.com`-, `.net`- und `.org`-Domainnamen.

### Benennung von Rechnern in lokalen Netzen

Als Hostname sollten Sie nicht den Namen des Rechnerherstellers, des Besitzers oder des gerade anstehenden Projekts verwenden – all das kann Verwirrung stiften. Verwenden Sie kurze und einprägsame Namen von Tieren, Pflanzen, Planeten, Flüssen oder was immer Ihnen einfällt. Deutsche Sonderzeichen sind nicht erlaubt.

Meine Testrechner zu Hause sind beispielsweise nach den Planeten unseres Sonnensystems benannt, z. B. `jupiter`. Als Domainname dient `sol`. Daraus ergibt sich dann der vollständige Name `jupiter.sol`.

Verwenden Sie niemals `localhost` als Hostnamen! Dieser Name hat insofern eine Sonderstellung, als dass er als vollständiger Netzwerkname gilt (*fully qualified*). Dem Namen ist immer die Adresse `127.0.0.1` der Loopback-Schnittstelle zugeordnet, unabhängig von den restlichen Parametern der Netzwerkkonfiguration.

- Schnittstelle (Interface)** Eine Schnittstelle kann wahlweise einen Hardware-Netzwerkadapter bezeichnen oder einen durch Software implementierten Verbindungspunkt zwischen verschiedenen Netzen.
- Ein Rechner hat oft mehrere Schnittstellen mit unterschiedlichen IP-Adressen. Typische Schnittstellen sind die Loopback-Schnittstelle, Ethernet- und WLAN-Schnittstellen sowie eventuell eine PPP-Schnittstelle, falls Sie ein UMTS-Modem verwenden oder sich in einem *Virtual Private Network* befinden.
- MAC-Adresse** Bei der MAC-Adresse (Media Access Control) handelt es sich um eine eindeutige ID-Nummer, mit der jeder Ethernet-Controller ausgestattet ist. Die MAC-Nummer ermöglicht eine Identifizierung des Netzwerk-Controllers, noch bevor ihm eine IP-Adresse zugewiesen wird. Die MAC-Adresse wird insbesondere vom Protokoll DHCP genutzt, das Sie im nächsten Kapitel kennenlernen.
- Schnittstellen-namen** Linux-intern bekommen alle Netzwerkschnittstellen einen Namen zugewiesen. Typische Namen sind `lo` für die Loopback-Schnittstelle, `eth0`, `eth1` etc. für die Ethernet-Schnittstelle, `wlanN` für WLAN-Schnittstellen und `pppN` für die PPP-Schnittstellen.
- Seit einigen Jahren verwenden immer mehr Distributionen für LAN- und WLAN-Schnittstellen eine andere Nomenklatur, z. B. `enp8s0` oder `wlp2s0`. Die zugrunde liegenden Namensregeln stellen sicher, dass ein bestimmter Netzwerkadapter immer wieder denselben Device-Namen erhält, selbst dann, wenn ein Computer nachträglich mit weiteren Netzwerkschnittstellen ausgestattet wird.
- Loopback-Interface** Die Loopback-Schnittstelle spielt eine besondere Rolle: Sie ermöglicht die Verwendung des Netzwerkprotokolls für lokale Dienste, also zur Kommunikation innerhalb des Rechners. Das klingt vielleicht widersinnig, ist aber für viele elementare Linux-Kommandos erforderlich. Der Grund: Manche Kommandos bauen ihre Kommunikation auf dem Netzwerkprotokoll auf, ganz egal, ob die Daten lokal auf dem Rechner bleiben oder über ein Netz auf einem fremden Rechner weiterverarbeitet werden. Ein Beispiel dafür ist das Druckersystem (CUPS), das Druckjobs sowohl lokal als auch von anderen Rechnern im Netzwerk entgegennimmt. Als IP-Adresse für das Loopback-Interface ist `127.0.0.1` vorgesehen.
- Netzwerkmaske, Netzwerk- und Broadcast-Adresse** Die Ausdehnung eines lokalen Netzes wird durch eine sogenannte Maske ausgedrückt. Dabei handelt es sich um vierteilige Zifferngruppen, die intern als Bitmuster für IP-Adressen verwendet werden. Wenn das lokale Netz z. B. alle Nummern `192.168.0.n` umfasst, lautet die dazugehörige Netzwerkmaske `255.255.255.0`, die Netzwerkadresse `192.168.0.0` und die Broadcast-Adresse `192.168.0.255`. Bei vielen Konfigurationsprogrammen brauchen Sie weder die Netzwerk- noch die Broadcast-Adresse anzugeben, da sich diese aus der IP-Adresse und der Maske ergeben.

Das resultierende Netzwerk wird jetzt mit 192.168.0.0/255.255.255.0 oder kurz mit 192.168.0.0/24 bezeichnet. Diese Kurzschreibweise heißt auch Präfix-Notation. Die Zahl hinter dem Schrägstrich gibt die Anzahl der binären Einser der Netzwerkmaske an. Zwei Rechner mit den IP-Adressen 192.168.0.71 und 192.168.0.72 können sich in diesem Netzwerk also direkt miteinander verständigen, weil die IP-Adressen im Bereich der Netzwerkmaske übereinstimmen. Die maximale Anzahl von Rechnern, die gleichzeitig in diesem Netz kommunizieren können, beträgt 254 (.1 bis .254) – die Nummern .0 und .255 sind reserviert.

Ein Gateway ist ein Rechner, der an der Schnittstelle zwischen zwei Netzen steht, oft zwischen dem lokalen Netz und dem Internet. Damit Ihr Linux-Rechner in einem lokalen Netz auf das Internet zugreifen kann, müssen Sie bei der Konfiguration die Gateway-Adresse angeben.

Gateway

Die Gateway-Adresse bezeichnet also einen besonderen Rechner im lokalen Netz. In Privathaushalten ist das Gateway oft ein ADSL-Modem oder ein damit verbundener WLAN-Router. Auch ein als Hotspot eingesetztes Smartphone (Tethering) kann als lokales Gateway dienen. In diesem Fall kommunizieren andere Rechner über das WLAN mit dem Smartphone, und dieses leitet die IP-Pakete dann über das Mobilfunknetz in das Internet weiter.

Ein Nameserver ist ein Programm, das Rechnernamen bzw. Internetadressen wie *www.yahoo.com* in IP-Adressen übersetzt. Bei kleinen Netzen erfolgt die Zuordnung zwischen Namen und Nummern manchmal durch eine statische Tabelle in der Datei */etc/hosts*. Im Internet übernehmen Rechner mit entsprechenden Datenbanken diese Aufgabe. Statt des Begriffs *Nameserver* ist auch die Abkürzung DNS für *Domain Name Server* oder *Services* üblich.

Nameserver

Wenn Sie in einem Webbrowser die Seite *www.yahoo.com* ansehen möchten, wird daher als Erstes der Nameserver kontaktiert, um die IP-Adresse des Webservers von *www.yahoo.com* herauszufinden. Erst nachdem das gelungen ist, wird eine Verbindung mit dieser IP-Adresse hergestellt.

Das *Dynamic Host Configuration Protocol* (DHCP) wird oft in lokalen Netzwerken verwendet, um die Administration des Netzwerks zu zentralisieren. Anstatt bei jedem Rechner manuell die IP-Adresse, die Maske, das Gateway, den Nameserver etc. einzustellen, übernimmt ein zentraler DHCP-Server bzw. ein ADSL-Router diese Aufgabe. Alle Rechner im lokalen Netzwerk nehmen beim Hochfahren Kontakt mit dem DHCP-Server auf und fragen ihn, welche Einstellungen sie verwenden sollen. Das reduziert die Client-Konfiguration auf ein Minimum.

DHCP

## IP-Adressen

IP-Adressen werden zur Identifizierung von Rechnern innerhalb eines Netzwerks verwendet. Das gilt sowohl in lokalen Netzen als auch im Internet. Dieser Abschnitt vermittelt Hintergrundinformationen über die Verwendung von IP-Adressen, wobei ich mich vorerst auf das »alte« Protokoll IPv4 beziehe. Informationen zu IPv6 folgen dann im nächsten Abschnitt.

Theoretisch sieht das Protokoll IPv4  $256^4$ , also rund 4 Milliarden IP-Adressen vor. Tatsächlich sind aber weit weniger IP-Adressen verfügbar: Zum einen ist ein Teil der Nummern für Spezialfunktionen reserviert, unter anderem alle IP-Adressen, die mit .0 bzw. .255 enden; zum anderen wurden IP-Adressen früher in recht großzügigen Paketen an Staaten bzw. Firmen vergeben. Mittlerweile ist der Vorrat an freien IPv4-Adressen erschöpft. Internet-Provider sowie Betreiber von Mobilfunknetzen sind deswegen als Erste auf IPv6 umgestiegen.

**IP-Adressen im Internet** Wenn Sie einen eigenen Webserver mit dem Internet verbinden möchten, benötigen Sie nicht nur einen weltweit gültigen Domainnamen (z. B. »meinefirma.de«), sondern auch eine eigene, feste IP-Adresse. Diese bekommen Sie von Ihrer Hosting-Firma zugewiesen.

**IP-Adressen in lokalen Netzen** Firmen oder Organisationen verwenden in lokalen Netzwerken zumeist IP-Adressen des privaten Adressraums. Nur das Gateway der Firma bzw. bei Privathaushalten der ADSL-Router haben eine öffentliche IP-Adresse. Alle Rechner des lokalen Netzwerks nutzen diese IP-Adresse dann gemeinsam. Das zugrunde liegende Verfahren wird *Masquerading* genannt. Es ist Thema des nächsten Kapitels.

Im IPv4-Zahlenraum wurden drei Bereiche für lokale Netzwerke reserviert (siehe [Tabelle 26.2](#)): Ganz egal, in welchem Teilnetz Sie Ihr lokales Netz bilden – es ist sichergestellt, dass es zu keinen Adresskonflikten mit »richtigen« IP-Internetadressen kommt.

| Adressbereich               | Netzwerk/Teilnetze   |
|-----------------------------|--|
| 10.0.0.0–10.255.255.255     | ein Netzwerk für ca. 16 Millionen Adressen                 |
| 172.16.0.0–172.31.255.255   | 16 Teilnetze (z. B. 172.23.*.*) für je ca. 65.000 Adressen |
| 192.168.0.0–192.168.255.255 | 256 Teilnetze (z. B. 192.168.54.*) für je 254 Adressen     |

**Tabelle 26.2** Reservierte IP-Bereiche für private Netzwerke

**Ein Rechner, mehrere IP-Adressen** Eine IP-Adresse bezeichnet nicht einen Rechner, sondern eine IP-Schnittstelle. Da die meisten Rechner über mehrere Schnittstellen verfügen, z. B. für den Ethernet-Controller, für den WLAN-Controller und für die Localhost-Schnittstelle, sind einem Rechner oft mehrere IP-Adressen zugewiesen! Wenn von *der* IP-Adresse die Rede ist,

als gäbe es nur eine einzige, dann ist zumeist diejenige Adresse gemeint, über die der Rechner im lokalen Netz oder im Internet angesprochen wird.

## IPv6

Bis jetzt habe ich mich immer auf IP-Version 4 bezogen (IPv4). Das gesamte Internet in seiner jetzigen Form basiert auf dieser IP-Version. Allerdings gibt es bereits seit 2011 keine freien IPv4-Adressblöcke mehr. Außerdem weist das Protokoll einige funktionelle Mängel auf, weswegen IP für manche Anwendungen nicht optimal geeignet ist, z. B. für das Audio- und Video-Streaming.

Die schon seit 1998 (!) standardisierte IP-Version 6 behebt diese Mängel. Die wohl markanteste und für Administratoren offensichtlichste Änderung besteht darin, dass für IP-Adressen nun statt 32 gleich 128 Bit vorgesehen sind. In der herkömmlichen Schreibweise würde eine IPv6-Adresse dann so aussehen:

IPv6-Adressen

```
121.57.242.17.122.58.243.18.19.123.59.20.244.124.60.245
```

Es ist offensichtlich, dass das nicht praktikabel ist. Um etwas Platz zu sparen, werden IPv6-Adressen in bis zu acht durch das Zeichen `:` getrennte Gruppen hexadezimaler Zahlen gegliedert, wobei in jeder Gruppe führende Nullen entfallen dürfen:

```
abcd:0017:02ff:12aa:2222:0783:00dd:1234 → abcd:17:2ff:12aa:2222:783:dd:1234
```

Um den Schreibaufwand weiter zu minimieren, gilt `::` als Kurzform für mehrere 0-Gruppen:

```
abcd:17:0:0:0:0:dd:1234 → abcd:17::dd:1234
0:0:0:0:0:783:dd:1234 → ::783:dd:1234
```

Für localhost gibt es mit `::1` eine noch kompaktere Kurzschreibweise:

```
0:0:0:0:0:0:0:1 → ::1
```

Wenn IPv4-Adressen in IPv6 abgebildet werden, sind die ersten fünf Gruppen 0, die sechste ffff. Die abschließenden 32 Bit dürfen statt in hexadezimaler Schreibweise auch in der vertrauten dezimalen Schreibweise angegeben werden:

```
110.111.112.113 (IPv4) → ::ffff:110.111.112.113 (IPv6)
```

IPv6 unterscheidet zwischen verschiedenen Typen von Adressen, von denen ich hier nur die wichtigsten vorstelle. Beachten Sie, dass Netzwerkschnittstellen zugleich *mehrere* IPv6-Adressen besitzen dürfen, z. B. eine private lokale Adresse (Link-Local) und eine öffentliche Adresse (Global Unicast):

IPv6-Adresstypen

- ▶ **Global Unicast:** Das sind »gewöhnliche«, weltweit gültige IP-Adressen. Sie beginnen zumeist mit der Ziffer 2 und eignen sich für eine Punkt-zu-Punkt-Kommunikation.
- ▶ **Link-Local:** Das sind private Adressen innerhalb eines lokalen Netzes im Adressbereich fe80::/64. Die Adressen werden automatisch generiert und ermöglichen eine konfigurationslose Kommunikation innerhalb eines lokalen Netzwerks, vergleichbar mit dem Zeroconf-Verfahren für IPv4, das in [Abschnitt 26.7](#), »Zeroconf und Avahi«, behandelt wird.
- ▶ **Site-Local:** Das sind private Adressen innerhalb der Adressbereiche fec0:: bis feff::. Sie können ähnlich wie der IPv4-Bereich 10.\*.\* zur Definition privater lokaler Netzwerke verwendet werden.
- ▶ **Multicast:** Adressen im Bereich ff00::/8 sind Multicast-Adressen. An solche Adressen gesendete Pakete werden an alle Geräte im betreffenden Netzwerk geleitet. Der Aufbau von Multicast-Adressen ist durch den Standard RFC 4291 definiert. Multicast-Adressen ersetzen die von IPv4 bekannten Broadcast-Adressen, bieten darüber hinaus aber zusätzliche Möglichkeiten.

Eine detailliertere Beschreibung der IPv6-Grundlagen sowie weiterer Sonderfälle von IPv6-Adressen finden Sie in der Wikipedia und auf der alten, aber weiterhin hilfreichen Linux-IPv6-HowTo-Seite:

<https://de.wikipedia.org/wiki/IPv6>

<http://tldp.org/HOWTO/Linux+IPv6-HOWTO>

#### IPv6-Masken und -Subnetze

Auch in IPv6 gilt das Konzept, dass Teilnetze durch Adressmasken gebildet werden. Die Maske wird ausschließlich in der Kurzschreibweise /n formuliert. Wie bei IPv4 gibt n die Anzahl der Bits am Beginn der Adresse an, die unveränderlich sind. Je kleiner n ist, desto größer ist die Anzahl der möglichen Adressen im Teilnetz.

Die Adresse 2001:78b:f2f:417::2/64 bedeutet, dass der Rechner ohne Router mit allen anderen Geräten kommunizieren kann, die eine Adresse der Form 2001:78b:f2f:417:\* haben. /64 bedeutet, dass die ersten 64 Bit der IP-Adresse das Teilnetz bestimmen und somit vorgegeben sind. Die restlichen 64 Bit dienen zur Identifizierung der Teilnehmer innerhalb des Teilnetzes. Ein /64-Netz bietet Platz für  $2^{64} = 18 \times 10^{18}$  Adressen, also ca. 4 Milliarden mal mehr Adressen als im gesamten IPv4-Netz!

Da es an IPv6-Adressen keinen Mangel gibt, sehen die IPv6-Richtlinien einen *äußerst* großzügigen Umgang mit Teilnetzen vor: Kleinere Teilnetze als /64 sind gar nicht vorgesehen! Vielmehr ist geplant, dass Internet-Provider jedem Kunden grundsätzlich ein /64-Netz zur Verfügung stellen. Damit gibt es beim Aufbau lokaler Firmen- oder Heimnetze keine Limits: Wenn Sie nicht gerade jedem Atom Ihres Haushalts seine eigene IPv6-Adresse zuordnen möchten, besteht keine Gefahr, dass Sie den Vorrat Ihrer IPv6-Adressen erschöpfen.



Der Linux-Kernel kommt mit IPv6 prinzipiell schon seit Anfang 2000 gut zurecht. Auch die meisten Netzwerkanwendungen und Distributionen sind längst IPv6-kompatibel. Wenn Sie beispielsweise einen Blick in die Datei `/etc/hosts` auf Ihrem Rechner werfen, werden Sie feststellen, dass es dort Einträge für zwei Localhost-Adressen gibt, einmal für IPv4 und einmal für IPv6:

IPv6 und Linux

```
# Datei /etc/hosts
127.0.0.1 localhost
::1      ip6-localhost
```

Mit dem Testkommando `ping6` können Sie IPv6-Pakete an `ip6-localhost` oder an die Adresse `::1` senden:

```
root# ping6 ip6-localhost
ping6 ip6-localhost
PING ip6-localhost(ip6-localhost) 56 data bytes
64 bytes from ip6-localhost: icmp_seq=1 ttl=64 time=0.016 ms
64 bytes from ip6-localhost: icmp_seq=2 ttl=64 time=0.023 ms
<Strg>+<C>
```

Grundsätzlich sind IPv4- und IPv6-Netze komplett voneinander getrennt! In einem IPv4-Netz können Sie keine IPv6-Rechner ansprechen und umgekehrt. Immerhin ist ein Parallelbetrieb möglich: Sie können also einen Rechner bzw. eine Netzwerkschnittstelle so einrichten, dass sie IPv4 *und* IPv6 spricht.

Kompatibilität  
und Tunnel

IPv4 und IPv6 werden noch über viele Jahre parallel existieren. Um den Mischbetrieb zu vereinfachen, existieren verschiedene Verfahren, um IPv6-Pakete auch über IPv4-Netze zu transportieren – und umgekehrt. Am populärsten sind hierfür sogenannte Tunnel, also spezielle Programme, die z. B. IPv6-Pakete nochmals verpacken und in einem IPv4-Netz zum nächsten IPv6-Router transportieren. Die erforderlichen Tunnelprotokolle sind aber nicht Teil des IPv6-Standards.

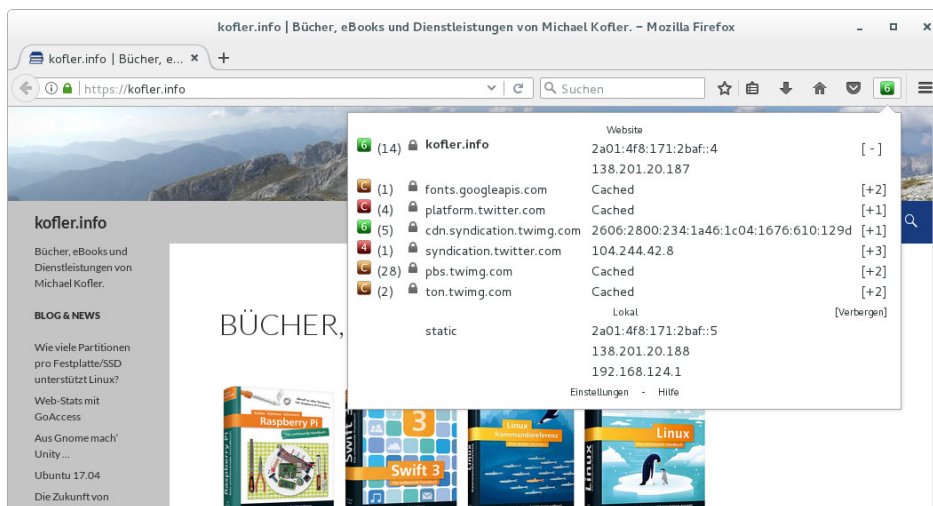
Mehrere Firmen boten in der Vergangenheit kostenlose Tunnel-Services an. Leider wurden die populärsten derartigen Dienste Ende 2015 eingestellt, angeblich, weil kein Bedarf mehr dafür besteht.

### IPv6 in der Praxis

Wenn Sie unsicher sind, ob Ihr Rechner IPv6-Zugang hat, besuchen Sie eine der vielen IPv6-Testseiten, z. B. <http://test-ipv6.com> oder <http://ipv6-test.com>. Alternativ können Sie mit `ping6 google.com` ausprobieren, ob Sie eine IPv6-Verbindung zu Google herstellen können. Zu guter Letzt können Sie die beiden Kommandos `ip -6 addr` und `ip -6 route` ausführen, wobei die Interpretation der Ergebnisse aber nicht ganz einfach ist. Details zum Kommando `ip` folgen in [Abschnitt 26.4](#), »Manuelle LAN- und WLAN-Konfiguration«.

IPv6-Test

Wenn Sie im Webbrowser feststellen möchten, ob Sie eine Seite via IPv4 oder IPv6 besuchen, installieren Sie eine Erweiterung wie IPvFoo (Chrome) oder SixOrNot (Firefox, siehe [Abbildung 26.8](#)). Diese Erweiterungen zeigen in der Adressleiste oder in einem Icon das genutzte Netz an. Bei SixOrNot zeigt das C-Icon an, dass ein Element einer Seite aus dem Firefox-Cache geladen wurde. In diesem Fall ist dann nicht mehr rekonstruierbar, ob die Daten ursprünglich via IPv4 oder IPv6 heruntergeladen wurden.



**Abbildung 26.8** Die Firefox-Erweiterung »SixOrNot« verrät, welche Elemente einer Webseite in welchem Netz verfügbar sind.

#### Tunnel für Clients

In der Praxis haben Sie als Desktop-Anwender in der Regel nichts mit IPv6 zu tun. Selbst wenn Ihr Mobilfunk- oder Internet-Provider intern IPv6 verwendet (das ist ziemlich wahrscheinlich), erfolgt die Kommunikation zu Ihnen meist in einem Tunnel und es sieht so aus, als befänden Sie sich in einem IPv4-Netzwerk. Das ist auch gut so, denn mit einem reinen IPv6-Zugang ohne IPv4-Kompatibilität könnten Sie nur wenige Websites besuchen.

Ich habe auf einer virtuellen Linux-Maschine, die ausschließlich IPv6-Zugang hatte, im Juni 2017 versuchsweise nach *hotel berlin* gesucht und die ersten 20 Treffer angeklickt. Ergebnis: Im IPv6-Netz sieht die Berlin-Touristik mager aus: Bei meinem Versuch war nur eine einzige Seite IPv6-tauglich. Selbst Branchen-Größen wie *booking.com* halten es nicht der Mühe wert, ihre Webauftritte auch in IPv6 anzubieten. Ein Trauerspiel!

Auch für die Administration lokaler Netzwerke ist IPv6 normalerweise nicht relevant: Es spricht nichts dagegen, ein LAN weiterhin als privates IPv4-Netzwerk einzurichten, z. B. im Adressbereich 192.168.0.\*.

Bleibt die Frage, wie wichtig IPv6 für Root-Server ist. Selbst hier können Sie IPv6 aus dem Weg gehen, wenn der Server überwiegend für europäische und amerikanische Benutzer gedacht ist. Anders sieht es in manchen afrikanischen oder asiatischen Ländern aus. Dort gibt es eine zunehmende Anzahl von Internetanwendern, die ausschließlich über eine IPv6-Verbindung ohne Tunnel oder andere Kompatibilitätsschichten verfügen.

Dual-Stack am Server

Langfristig geht deswegen für Root-Server kein Weg am Parallelbetrieb von IPv4 und IPv6 vorbei, also an einer sogenannten »Dual-Stack«-Konfiguration. Google, Facebook oder heise.de haben diesen Schritt bereits vor einigen Jahren vollzogen. Auch meine eigene Webseite <https://kofler.info> ist seit Jahren via IPv6 erreichbar. Auf die IPv6-Konfiguration gehe ich daher in allen Kapitel in Teil VII, »Root-Server«, ein.

### WLAN-Standards, Glossar

Zur Beschreibung drahtloser Netze haben sich mehrere Abkürzungen eingebürgert. Am gängigsten ist WLAN (*Wireless Local Area Network*) oder dessen deutsche Variante *Funk-LAN*. In englischen Texten ist sehr oft von WiFi (*Wireless Fidelity*) die Rede, wobei der Begriff oft synonym mit WLAN verwendet wird. Manchmal ist aber auch die *WiFi-Alliance* gemeint, ein Herstellerkonsortium, das sich um die Kompatibilität von WLAN-Produkten kümmert.

Dieser Abschnitt fasst kurz die WLAN-Terminologie zusammen. Die wichtigsten WLAN-Standards sind durch das IEEE (*Institute of Electrical and Electronics Engineers*) definiert und beginnen alle mit der Nummer 802.11. Die ergänzenden Buchstaben beziehen sich in chronologischer Reihenfolge auf Versionen bzw. Varianten des Standards.

802.11x

Die Bruttoübertragungsrate diverser WLAN-Standards sieht auf Datenblättern oft vielversprechend aus. Nach Abzug des großen Protokoll-Overheads bleibt davon netto meist weniger als die Hälfte übrig – und auch das nur, solange nicht mehr als zwei WLAN-Teilnehmer miteinander kommunizieren.

Brutto/netto

Es gibt verschiedene Arten von WLAN-Hardware:

WLAN-Hardware

- ▶ In alle modernen Notebooks ist ein **WLAN-Controller** eingebaut. Bei Desktop-Rechnern oder älteren Geräten kann auch ein WLAN-USB-Stick diese Funktion übernehmen.
- ▶ Eine **WLAN-Bridge** verbindet einen einzelnen Computer mit einem WLAN. Die Bridge erfüllt also dieselbe Funktion wie der eingebaute WLAN-Controller eines Notebooks, nur der Anschluss an den Rechner ist anders: Die Bridge wird durch ein Ethernet-Kabel mit dem Computer verbunden.

- ▶ Ein **Access-Point** stellt in einem lokalen Netzwerk einen Zugangspunkt für mehrere WLAN-Clients her. Der Access-Point wird mit einem Ethernet-Kabel an das lokale Netzwerk angeschlossen. Im Unterschied zur Bridge kann ein Access-Point mit mehreren Clients gleichzeitig kommunizieren.
- ▶ Ein **WLAN-Router** agiert ähnlich wie ein Access-Point, ist aber »intelligenter«: Das Gerät enthält unter anderem einen eigenen DHCP-Server, um den WLAN-Clients Ihre IP-Daten zuzuweisen. In viele moderne ADSL-Modems ist ein WLAN-Router integriert. Derartige Geräte bezeichnet man dann oft als *Gateways*.

Die Konfiguration von WLAN-Geräten erfolgt zumeist durch einen Webbrowser. Dazu stellen die Geräte unter einer bestimmten IP-Adresse (z. B. <http://192.168.0.1>) eigene Webseiten zur Konfiguration zur Verfügung.

### WLAN-Verbindungsparameter

Wenn Sie eine Verbindung zwischen zwei WLAN-Geräten herstellen, müssen Sie diverse Parameter einstellen.

**Network-Modus** WLAN-Komponenten können auf unterschiedliche Arten miteinander kommunizieren. Im Folgenden sind nur die drei wichtigsten Modi kurz beschrieben:

- ▶ Der *Infrastructure Mode* (manchmal auch *Managed Mode* genannt) erlaubt die Kommunikation mit einem zentralen Zugangspunkt. Die Netzwerkstruktur ist also sternförmig. Meist ist der Zugangspunkt ein Access-Point oder ein WLAN-Router, es kann aber auch ein entsprechend konfigurierter Rechner sein.
- ▶ Das WLAN-Gerät des Access Points läuft im *Master Mode*. Der *Infrastructure Mode* gilt also gewissermaßen für die Clients, während der *Master Mode* für den Server eines WLANs zur Anwendung kommt.
- ▶ Beim *Ad-hoc Mode* kommuniziert jedes WLAN-Gerät direkt mit jedem anderen WLAN-Gerät, das in Funkreichweite ist.

**SSID bzw. ESSID** Die Abkürzungen SSID (*Service Set Identification*) bzw. ESSID (*Extended SSID*) bezeichnen eine Zeichenkette, die einem WLAN-Netz einen Namen gibt. WLAN-Geräte können nur dann miteinander kommunizieren, wenn ihre SSIDs übereinstimmen.

Als SSID ist oft der Herstellername voreingestellt. Deswegen können Geräte desselben Herstellers oft auf Anhieb miteinander kommunizieren, während bei Geräten unterschiedlicher Herkunft zuerst eine gemeinsame SSID-Zeichenkette eingestellt werden muss.

Manche WLAN-Karten sehen für die SSID eine Auto-Konfiguration vor (Einstellung ANY). Beachten Sie, dass bei der SSID-Zeichenkette zwischen Groß- und Kleinschreibung unterschieden wird!

Innerhalb eines WLAN-Netzes mit einheitlicher SSID kann es mehrere Teilnetze (Cells) geben, zwischen denen mit der NWID (*Network ID*) differenziert wird. In der Praxis kommt das aber nur selten vor, weswegen manche Konfigurationsprogramme auf die NWID gleich ganz verzichten.

NWID

Manchmal wird statt NWID der verwirrende Begriff *Domain* verwendet. Mit dem herkömmlichen Domainnamen von IP-Adressen hat die NWID aber nichts zu tun.

Innerhalb des Frequenzbandes, das durch den jeweiligen 802.11x-Standard vorgesehen ist, gibt es mehrere Teilbereiche (Kanäle, Channels), auf denen parallel gesendet werden kann. Im Infrastructure Mode erkennen WLAN-Adapter selbstständig den Kanal, der vom Access Point vorgegeben ist. Eine explizite Einstellung des Kanals ist nur notwendig, wenn es zu Interferenzen mehrerer WLANs kommt.

Channel

### WLAN-Sicherheit

Grundsätzlich ist es möglich, ein WLAN unverschlüsselt zu betreiben. Dann kann aber jeder, der sich in Reichweite des Funknetzes befindet, dieses nutzen und die gesamte Kommunikation abhören. Ein unverschlüsselter Betrieb ist daher grob fahrlässig!

Zur Verschlüsselung des Datenverkehrs kam bei den ersten WLAN-Generationen das Verfahren *Wired Equivalent Privacy* (WEP) zum Einsatz. Dabei werden die Daten wahlweise mit einem 40- oder 104-Bit-Schlüssel verschlüsselt. Oft ist auch von 64- bzw. 128-Bit-Verschlüsselung die Rede. Die restlichen 24 Bit dienen aber nicht zur eigentlichen Verschlüsselung.

WEP

Der WEP-Schlüssel wird in der Regel als hexadezimale Zahl angegeben (10 bzw. 26 Stellen, je nach der Bitanzahl des Schlüssels). Manche Konfigurationswerkzeuge bieten Ihnen auch die Möglichkeit, den Schlüssel aus einer »Passphrase« zu erzeugen, also aus einem Text, der auch aus mehreren Wörtern bestehen darf.

#### Vermeiden Sie WEP!

WEP hat sich aufgrund von gravierenden Konzeptmängeln als unsicher herausgestellt. Selbst ein 104-Bit-Schlüssel kann durch simples Abhören des WLAN-Verkehrs innerhalb weniger Minuten ermittelt werden.

Die Nachfolge von WEP haben die Verfahren *WiFi Protected Access* (kurz WPA) sowie dessen verbesserte Version WPA2 angetreten. Die genaue Spezifikation von WPA2 ist im Standard 802.11i niedergeschrieben. Der wichtigste Unterschied zwischen WPA und WPA2 ist der Verschlüsselungsalgorithmus: RC4 bei WPA, AES bei WPA2.

WPA, WPA2

WPA war als Übergangslösung bis zur Fertigstellung des 802.11i-Standards gedacht. Da es aber WLAN-Hardware gibt, die nur WPA, aber noch nicht WPA2 unterstützt, werden auf absehbare Zeit beide Varianten im Einsatz bleiben.

Der wesentliche Vorteil von WPA besteht darin, dass der Schlüssel nur zur Initialisierung der Verbindung eingesetzt wird. Sobald die Verbindung steht, werden die Schlüssel nach einem ausgeklügelten Verfahren ständig gewechselt. WPA und WPA2 gelten nach aktuellem Stand der Technik als sicher, sofern eine ausreichend lange Passphrase eingesetzt wird, also ein aus mehreren Wörtern und Zeichen bestehender Schlüssel.

Dieses Buch behandelt nur die WPA/WPA2-Variante *Pre-Shared Key* (kurz PSK, oft auch *WPA-Personal* genannt): Hier melden sich alle WLAN-Nutzer mit demselben Schlüssel im Netz an. Bei der noch sichereren Variante *Managed Key* hat jeder Nutzer einen eigenen Schlüssel. Allerdings müssen die Schlüssel nun auf einem zentralen Server verwaltet werden.

#### Sichern Sie auch das WLAN-Gerät an sich ab!

Die Einstellungen des WLAN-Routers oder Access Points werden normalerweise per Webbrowser verändert. Der Webzugang ist durch ein firmenspezifisches Passwort abgesichert, das Sie unbedingt ändern sollten! Generell sollten Sie die Fernwartung so weit wie möglich einschränken und nicht per WLAN, sondern nur über eine LAN-Verbindung durchführen.

**Firewall und VPN** Durch eine Firewall können Sie den WLAN-Datenverkehr gezielt auf bestimmte Protokolle, Netzwerksegmente etc. beschränken. Ein anderer Ansatzpunkt besteht darin, das WLAN trotz aller anderen Schutzmaßnahmen als unsicher zu betrachten. Um dennoch sicher zu kommunizieren, verschlüsseln Sie Ihren Datenverkehr selbst. Am populärsten ist für diesen Zweck die Verwendung eines VPN (Virtual Private Network).

### Linux-Unterstützung für WLANs

**Wireless-Tools und iw** In der Vergangenheit spielten die Linux-Wireless-Tools eine große Rolle bei der Nutzung von WLAN-Komponenten. Die Wireless-Tools bestehen aus mehreren Kommandos (*iwconfig*, *iwlist* etc.) zur Konfiguration der WLAN-Adapter. Die Wireless-Tools gelten aber als konzeptionell veraltet.

Als vor einigen Jahren die Generalüberholung der Linux-WLAN-Treiber begann, wurde das neue Steuerungskommando *iw* entwickelt. *iw* kann allerdings nur für WLAN-Controller verwendet werden, deren Treiber die nl80211-Schnittstelle unterstützen. Aus diesem Grund installieren viele Linux-Distributionen die Kommandos *iwconfig*

und `iw` parallel. Hintergrundinformationen zu beiden Linux-Wireless-Systemen finden Sie hier:

<https://hewlettpackard.github.io/wireless-tools/Tools.html>

<https://wireless.wiki.kernel.org>

Die eigentlichen WLAN-Hardware-Treiber befinden sich in Kernelmodulen. Aktuelle Treiber basieren auf dem mac80211-Framework und sind zur nl80211-Schnittstelle und damit zum Kommando `iw` kompatibel. Linux enthält Treiber zu nahezu allen marktüblichen WLAN-Adaptoren – aber wie immer gibt es Ausnahmen. Besonders problematisch sind ganz neue WLAN-Adapter: Selbst bei einer guten Kooperation zwischen dem Hardware-Hersteller und der Linux-Entwicklergemeinschaft dauert es oft ein ganzes Jahr, bis neue Treiber den Weg in aktuelle Distributionen finden. Es lohnt sich also, vor dem Kauf eines Notebooks ein wenig im Internet zu recherchieren!

Hardware-Treiber

Die meisten WLAN-Controller können Sie selbst programmieren. Damit sie funktionieren, muss während der Initialisierung die sogenannte Firmware, also controller-interner Programmcode, in den Controller übertragen werden. Die Firmware stammt von den Controller-Herstellern und darf unter Einhaltung der jeweiligen Lizenzbedingungen frei weitergegeben werden. Um die Übertragung des Codes in den Controller kümmert sich in der Regel das Kernelmodul oder das `udev`-System. Der Controller-Code befindet sich in Binärdateien (Blobs), zumeist im Verzeichnis `/lib/firmware`.

Firmware

Die Chip-Hersteller stellen die Firmware nur in binärer Form zur Verfügung, nicht als Quellcode. Das ist aus Open-Source-Sicht betrüblich und wird vor allem von der Debian-Entwicklergemeinschaft kritisiert. Bei der Installation von Debian müssen Sie Firmware-Dateien zuerst aus dem Internet herunterladen und dann auf einem eigenen Datenträger zur Verfügung stellen (siehe [Abschnitt 3.2](#), »Debian«).

Persönlich sehe ich das Firmware-Problem entspannter als die Debian-Entwickler: Früher wäre dem WLAN-Controller ein EPROM hinzugefügt worden, und kein Hahn hätte danach gekräht, dass dieses keinen Open-Source-Code enthält. Die jetzige Lösung ist billiger und erlaubt Updates. Natürlich wäre es wünschenswert, wenn auch für die im Controller ausgeführten Programme der Quellcode verfügbar wäre, aber diese Hoffnung ist unrealistisch.

## 26.4 Manuelle LAN- und WLAN-Konfiguration

Normalerweise wird Ihr LAN- bzw. WLAN-Controller während des Rechnerstarts automatisch erkannt und initialisiert. Dieser Abschnitt zeigt, wie dieser Prozess hinter den Kulissen abläuft bzw. welche Schritte erforderlich sind, um die Initialisierung von Hand durchzuführen. Das hilft, die Netzwerkgrundlagen besser zu verstehen und die Benutzeroberflächen gängiger Konfigurationswerkzeuge sicherer zu verwenden.

## LAN-Controller manuell aktivieren

Der Netzwerk- bzw. LAN-Controller ist in der Regel ein Chip auf dem Mainboard Ihres Rechners, der Ethernet-Netzwerkfunktionen zur Verfügung stellt. Der Controller kann aber auch in die CPU integriert sein oder extern in Form eines USB-Steckers realisiert sein. Unabhängig davon, wie die Netzwerkfunktionen physikalisch realisiert sind, spreche ich im Folgenden nur noch vom Netzwerk-Controller.

### Hardware-Erkennung

Im ersten Schritt stellen Sie sicher, dass das richtige Kernelmodul für Ihren Netzwerk-Controller geladen wird. Oft gelingt dies dem Kernel automatisch. In diesem Fall wird das Kommando `ip link set enp4s0 up` ohne Fehlermeldung ausgeführt. Treten an dieser Stelle Probleme auf, müssen Sie herausfinden, welcher Netzwerk-Controller in Ihrem Rechner steckt und welches Kernelmodul dafür verantwortlich ist. Erste Informationen liefert in solchen Fällen `lspci`:

```
root# lspci | grep -i net
02:01.0 Ethernet controller: Intel Corporation 82540EP Gigabit Ethernet
Controller (Mobile) (rev 03)
```

Das Notebook verwendet also den Gigabit-Ethernet-Controller 82540EP von Intel. Der zweite Schritt besteht nun darin, dem Controller einen passenden Treiber zuzuordnen (also ein Kernelmodul aus dem Verzeichnis `/lib/modules/n.n/net/*`). Eine Internetsuche nach *linux kernel module 82540EP* führt rasch zum richtigen Kernelmodul `e1000`:

```
root# modinfo e1000
filename:      /lib/modules/n.n/kernel/drivers/net/ethernet/intel/e1000/e1000.ko
description:  Intel(R) PRO/1000 Network Driver
...
```

Mit `lsmod` können Sie nun überprüfen, ob das Modul bereits geladen ist. In der Regel wird das der Fall sein, d. h., Linux hat den Controller während des Systemstarts bereits richtig erkannt. Nur wenn das nicht der Fall ist, müssen Sie mit `modprobe` das passende Modul laden:

```
root# modprobe e1000
```

`dmesg` zeigt, ob beim Laden des Moduls Fehler auftreten (was hier nicht der Fall ist). Die Warnung *link is not ready* besagt nur, dass die Schnittstelle momentan mangels Konfiguration noch nicht aktiv ist.

```
root# dmesg -c
...
Intel(R) PRO/1000 Network Driver - version 7.3.21-k8-NAPI
e1000 0000:02:01.0: PCI INT A -> Link[LNKA] -> GSI 11 (level, low) -> IRQ 11
e1000: 0000:02:01.0: e1000_probe: (PCI:33MHz:32-bit) 00:11:25:32:4f:5d
e1000: enp4s0: e1000_probe: Intel(R) PRO/1000 Network Connection
ADDRCONF(NETDEV_UP): enp4s0: link is not ready
...
```



Im Regelfall wird das Modul automatisch während der Initialisierung des Rechners geladen. Sollte das nicht funktionieren, tragen Sie die Zuordnung zwischen der Schnittstelle und dem Kernelmodul in die Modulkonfigurationsdatei ein:

```
# Modulkonfigurationsdatei /etc/modprobe.d/config.conf
alias enp4s0 e1000
```

Netzwerkschnittstellen haben je nach Distribution unterschiedliche Namen. Bei einigen Distributionen sind noch die Schnittstellennamen eth0, eth1 etc. üblich, aber immer mehr Distributionen verwenden Device-Namen wie enp0s5 oder p5p1. Eine Liste aller auf Ihrem Rechner verfügbaren Netzwerkschnittstellen liefert das Kommando `ip link show`:

Liste der  
Netzwerk-  
schnittstellen

```
root# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 ...
2: enp4s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 ...
```

Anschließend aktivieren Sie die gewünschte Netzwerkschnittstelle mit `ip link set <adapter> up`:

Aktivierung der  
Schnittstelle

```
root# ip link set enp4s0 up
```

Um die Netzwerkschnittstelle zu konfigurieren, führen Sie das Kommando `ip addr add` aus. `ip addr show enp4s0` zeigt anschließend alle bekannten Informationen zur Netzwerkschnittstelle an:

Konfiguration der  
Schnittstelle

```
root# ip addr add 192.168.0.2/24 dev enp4s0
root# ip addr show enp4s0
2: enp4s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
    pfifo_fast state UP qlen 1000
    link/ether 00:1c:42:85:09:a1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.2/24 scope global enp4s0
        valid_lft forever preferred_lft forever
```

Nun können Sie mit `ping` überprüfen, ob Sie Kontakt zu anderen Rechnern im lokalen Netzwerk aufnehmen können. Die Option `-c 2` bewirkt, dass genau zwei `ping`-Pakete versendet werden:

```
root# ping -c 2 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=2.95 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=0.169 ms
```

```
--- 192.168.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.169/1.560/2.952/1.392 ms
```

## »ip« versus »ifconfig« und »route«

In vielen Büchern zur Linux-Netzwerkkonfiguration sind die Kommandos `ifconfig` und `route` beschrieben. Das galt auch für die ersten 12 Auflagen dieses Buchs. `ifconfig` und `route` gelten aber schon seit einiger Zeit als veraltet.

`ip` bietet diverse Zusatzfunktionen, die in `ifconfig` und `route` fehlen, und ist speziell für die Netzwerkfunktionen des Linux-Kernels optimiert. Erfahrene Administratoren sollten sich jetzt von alten Gewohnheiten trennen und auf das neue Kommando `ip` umsteigen! Und Linux-Einsteiger machen um `ifconfig` und `route` am besten von Anfang an einen weiten Bogen.

## Nameserver-Konfiguration

`ping` funktioniert momentan nur, wenn Sie die richtige IP-Adresse angeben. Damit Sie stattdessen auch einen Rechnernamen angeben können, muss `/etc/resolv.conf` die IP-Adresse eines Nameservers enthalten. Das folgende Beispiel geht davon aus, dass es im lokalen Netz einen eigenen Nameserver mit der IP-Adresse 192.168.0.1 gibt. Der Nameserver kann aber auch außerhalb sein und vom Internet-Provider zur Verfügung gestellt werden. (Details zu dieser Konfigurationsdatei folgen in [Abschnitt 26.5, »LAN-Konfigurationsdateien«](#).)

```
# /etc/resolv.conf
nameserver 192.168.0.1
```

## Default-Gateway

Momentan können Pakete nur innerhalb des lokalen Netzwerks versandt werden. Damit auch ein Kontakt in das Internet möglich wird, muss der Rechner wissen, wohin er derartige Pakete leiten soll. Sie müssen dazu die Adresse des Internet-Gateways Ihres Netzwerks mit `ip route` angeben. Das folgende Beispiel geht davon aus, dass die IP-Adresse des Gateways 192.168.0.1 ist:

```
root# ip route add default via 192.168.0.1
```

`ip route` ohne weitere Parameter liefert die Routing-Tabelle. Die Gateway-Adresse ist in der Zeile enthalten, die mit `default` beginnt:

```
root# ip route
default via 10.211.55.1 dev enp4s0
192.168.0.0/24 dev enp4s0 proto kernel scope link src 192.168.0.2
```

Jetzt sollte es möglich sein, Pakete an beliebige Adressen im Internet zu senden:

```
root# ping -c 2 yahoo.com
PING yahoo.com (98.139.183.24) 56(84) bytes of data.
64 bytes from ir2.fp.vip.bf1.yahoo.com (98.139.183.24):
  icmp_seq=1 ttl=50 time=151 ms
64 bytes from ir2.fp.vip.bf1.yahoo.com (98.139.183.24):
  icmp_seq=2 ttl=50 time=153 ms
--- yahoo.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 151.853/152.839/153.826/1.061 ms
```

Eine manuelle Konfiguration der Loopback-Schnittstelle ist selten erforderlich. Falls doch, führen Sie die beiden folgenden Kommandos aus:

Loopback-Schnittstelle aktivieren

```
root# ip link set lo up
root# ip addr add 127.0.0.1 dev lo
```

Grundsätzlich können Sie mit `ip addr del` bzw. `ip route del` zuvor durchgeführte Adresszuweisungen oder Routendefinitionen wieder rückgängig machen. Dabei müssen Sie exakt dieselben Parameter angeben wie beim entsprechenden Kommando `ip addr add`:

Deaktivierung der Schnittstelle

```
root# ip addr add 192.168.0.2/24 dev enp4s0
root# ip addr del 192.168.0.2/24 dev enp4s0
```

Um eine Netzwerkschnittstelle ganz zu deaktivieren, also alle Adress- und Routendaten zu löschen, führen Sie `ip addr flush` aus:

```
root# ip addr flush dev enp4s0
```

### DHCP-Informationen abrufen

Falls es im Netzwerk einen DHCP-Server gibt, können Sie diesen zur Konfiguration zu Hilfe nehmen. Nach der Aktivierung der Schnittstelle durch `ip link set enp4s0 up` führen Sie bei den meisten Distributionen das Kommando `dhclient` aus:

DHCP-Informationen abrufen

```
root# dhclient -v enp4s0
...
Listening on LPF/enp4s0/00:11:25:32:4f:5d
Sending on LPF/enp4s0/00:11:25:32:4f:5d
Sending on Socket/fallback
DHCPDISCOVER on enp4s0 to 255.255.255.255 port 67 interval 3
DHCPOFFER from 192.168.0.1
DHCPREQUEST on enp4s0 to 255.255.255.255 port 67
DHCPACK from 192.168.0.1
bound to 192.168.0.15 -- renewal in 36624 seconds.
```

Bei älteren SUSE-Distributionen kommen Sie mit `dhcpcd` zum Ziel:

```
root# dhcpcd enp4s0
```

#### Ad-hoc-Netzwerkconfiguration

Das Kommando `dhclient` bietet oft den schnellsten Weg, um auf einem noch nicht konfigurierten Rechner ad hoc einen Netzwerkzugang herzustellen. Sie ersparen sich die `ip`-Kommandos und die manuelle Einstellung von `/etc/resolv.conf`. Die einzige Voraussetzung ist ein DHCP-Server im lokalen Netzwerk.

## IPv6-Konfiguration

Alle bisherigen Beispiele haben sich auf IPv4 bezogen. Selbstverständlich können Sie mit dem `ip`-Kommando auch eine IPv6-Konfiguration durchführen. Die folgenden Kommandos setzen voraus, dass Sie über das IPv6-Subnetz `2a01:4f8:161:107::/64` verfügen und die Gateway-Adresse `fe80::1` verwenden können. Der Schnittstelle `enp4s0` wird die Adresse `2a01:4f8:161:107::2` zugewiesen. Zum Ausprobieren senden Sie mit `ping6` Kommandos an einen Server, von dem Sie wissen, dass er IPv6-tauglich konfiguriert ist.

```
root# ip -6 addr add 2a01:4f8:161:107::2/64 dev enp4s0
root# ip -6 route add default via fe80::1 dev enp4s0
root# ping6 -n google.com
PING google.com(2a00:1450:4009:805::1002) 56 data bytes
64 bytes from 2a00:1450:4009:805::1002: icmp_seq=1 ttl=54 time=19.7 ms
64 bytes from 2a00:1450:4009:805::1002: icmp_seq=2 ttl=54 time=19.1 ms
<Strg>+<C>
```

### IPv6-Konfiguration testen

Das gerade erwähnte Kommando `ping6` eignet sich für erste IPv6-Tests am besten. Sollte `ping6` die Fehlermeldung `connect: Das Netzwerk ist nicht erreichbar` liefern, können Sie sich die IPv6-Konfiguration mit dem Kommando `ip` genauer ansehen.

`ip addr show <interface>` liefert detaillierte Daten zur Netzwerkschnittstelle `<interface>`. Die mit `link/ether` beginnende Zeile gibt die MAC-Adresse der Schnittstelle an. Die mit `inet` beginnende Zeile enthält die IPv4-Adresse samt Maske in der Kurzschreibweise `/n` sowie die Broadcast-Adresse. Die Zeilen, die mit `inet6` beginnen, geben die IPv6-Adressen an. Das können mehrere sein. Eine »richtige« IPv6-Konfiguration setzt voraus, dass die Schnittstelle mit einer globalen Unicast-Adresse verbunden ist, die üblicherweise mit der Ziffer 2 beginnt.

Eine IPv6-Adresse allein reicht nicht aus. Linux muss auch wissen, wohin es IPv6-Pakete leiten soll. Mit `ip -6 route | grep default` ermitteln Sie das Default-Gateway für IPv6.

Im ersten Beispiel besteht nur eine IPv4-Konfiguration. Die in der Zeile `inet6` angegebene Adresse ist nur eine sogenannte Link-Local-Adresse, die automatisch eingerichtet wird und eine Kommunikation innerhalb eines lokalen Netzwerks erlaubt (vergleichbar mit Zeroconf für IPv4, siehe auch [Abschnitt 26.7](#), »Zeroconf und Avahi«). Es ist kein IPv6-Gateway eingerichtet.

```
root# ip addr show enp4s0
2: enp4s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 ...
    link/ether 00:1c:42:d0:29:34 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.13/24 brd 10.0.0.255 scope global enp4s0
    inet6 fe80::21c:42ff:fed0:2934/64 scope link
        valid_lft forever preferred_lft forever
```

```
root# ip -6 route | grep default
-- kein Ergebnis --
```

Das zweite Beispiel zeigt die Ergebnisse auf einem Root-Server mit IPv4- und IPv6-Konfiguration. Der Schnittstelle enp4s0 ist die Unicast-Adresse 2a01:xxx::2 zugewiesen (scope global im ip-Ergebnis), als IPv6-Gateway ist fe80::1 eingerichtet.

```
root# ip addr show enp4s0
3: enp4s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    link/ether 54:04:a6:f1:74:1f brd ff:ff:ff:ff:ff:ff
    inet 5.9.22.18 peer 5.9.22.1/32 brd 5.9.22.31 scope global enp4s0
    inet6 2a01:4f8:161:107::2/64 scope global
        valid_lft forever preferred_lft forever
    inet6 fe80::5604:a6ff:fef1:741f/64 scope link
        valid_lft forever preferred_lft forever
root# ip -6 route | grep default
default via fe80::1 dev enp4s0 metric 1024
```

### WLAN-Controller manuell steuern

Das Kernelmodul für den WLAN-Controller wird im Regelfall automatisch geladen. Bei den meisten Distributionen hat die WLAN-Schnittstelle den Namen wlan0 oder wlpxxx. Wenn Sie sich unsicher sind, ermitteln Sie mit `ip link` eine Liste aller Schnittstellen. `dmesg | egrep -i "wlan|wifi"` liefert alle relevanten Kernelmeldungen.

Hardware-  
Erkennung

Mit `iw dev NAME info` und `iw dev NAME link` können Sie den Status der Schnittstelle herausfinden. Im folgenden Beispiel existiert eine aktive Verbindung:

Aktuellen Status  
ermitteln

```
root# iw dev wlp2s0 info
Interface wlp2s0
    ifindex 3
    type managed
    wiphy 2
root# iw dev wlp2s0 link
Connected to 00:16:b6:9d:ff:4b (on wlp2s0)
    SSID: wlan-sol2
    freq: 2462
    RX: 102262 bytes (1784 packets)
    TX: 12815 bytes (86 packets)
    signal: -59 dBm
    tx bitrate: 54.0 MBit/s
    bss flags:      short-preamble short-slot-time
    dtim period:   0
    beacon int:    100
```

Die Pseudodatei `/proc/net/wireless` fasst ebenfalls Informationen zur aktuellen Qualität der WLAN-Verbindung zusammen:

```

root# cat /proc/net/wireless
Inter-| sta-|   Quality   |   Discarded packets   | Missed | WE
face | tus | link level noise | nwid crypt frag retry misc | beacon | 22
wlp2s0: 0020 91. 189. 0. 0 0 0 0 4 0

```

#### WLAN-Schnittstelle einrichten und löschen

Wenn `iw` die Schnittstelle `wlp2s0` nicht kennt (Fehlermeldung *no such device*), müssen Sie die Schnittstelle zuerst einrichten. Dazu dient das `iw`-Kommando `interface add`:

```
root# iw phy phy0 interface add wlp2s0 type managed
```

`phy0` adressiert den ersten (und bei den meisten Rechnern auch einzigen) WLAN-Controller. Anstelle von `managed` können Sie auch die Schnittstellentypen `monitor`, `wds`, `mesh` bzw. `mp` sowie `ibss` bzw. `adhoc` angeben. Die neue Schnittstelle muss anschließend aktiviert werden.

```
root# ip link set wlp2s0 up
```

Sollten Sie die Schnittstelle nicht mehr benötigen, können Sie sie mit `iw del` wieder löschen:

```
root# iw dev wlp2s0 del
```

#### WLAN-Verbindung herstellen

`iw scan` liefert detaillierte Informationen zu allen in Reichweite befindlichen Funknetzen. `grep` macht daraus eine simple Liste aller Netzwerknamen (SSIDs):

```

root# iw dev wlp2s0 scan | grep SSID
      SSID: myhome
      SSID: wlan-sol2
      ...

```

Einen manuellen Verbindungsaufbau können Sie bei nicht abgesicherten WLAN-Netzen mit `iw ... connect` herstellen. Dabei müssen Sie den Namen des Netzwerks (also die SSID) angeben. `iw` kümmert sich nur um die WLAN-Verbindung. Die Ethernet-Konfiguration führen Sie anschließend mit `dhclient` durch (`dhcpcd` unter SUSE).

```

root# iw dev wlp2s0 connect hotel-wlan-1
root# dhclient wlp2s0

```

`iw disconnect` beendet die Verbindung wieder:

```
root# iw dev wlp2s0 disconnect
```

### WLAN-Verschlüsselung

#### WLAN-Verbindung mit WEP

Wenn das Funknetz durch WEP abgesichert ist, können Sie den Schlüssel mit einem zusätzlichen Parameter in der Form `n:xxx` übergeben. Dabei gibt `n` die Schlüsselnummer an (0 bis 3). Der eigentliche Schlüssel wird wahlweise in Form von 5 oder 13 ASCII-Zeichen bzw. durch 10 oder 26 hexadezimale Ziffern angegeben.

```
root# iw dev wlp2s0 connect hotel-wlan-1 keys 0:1a790bcc31
root# dhclient wlp2s0
```

Etwas komplizierter ist die Vorgehensweise, wenn WPA oder WPA2 im Spiel sind. In diesem Fall ist für die Initialisierung der Verbindung und für den weiteren Austausch von sich immer wieder ändernden Schlüsseln das Hintergrundprogramm `wpa_supplicant` aus dem gleichnamigen Paket zuständig. Nach dessen Installation richten Sie eine Konfigurationsdatei ein, wobei Sie als Dateiname z. B. `/etc/wpa_supplicant.conf` wählen.

WLAN-Verbindung mit WPA oder WPA2

Die Datei kann einige globale Einstellungen enthalten. Anschließend folgen spezifische Parameter für verschiedene WLAN-Netze. Das folgende Beispiel zeigt eine Minimumvariante, die für den Verbindungsaufbau zu einem WLAN-Router oder WLAN-Access-Point mit WPA- oder WPA2-Personal-Verschlüsselung ausreicht. Die beiden entscheidenden Parameter sind `ssid` zur Identifizierung des Netzwerks und `psk` mit dem aus Sicherheitsgründen nochmals verschlüsselten Schlüssel. (Es ist auch zulässig, den WPA-Schlüssel in Anführungszeichen als Klartext anzugeben.)

```
# /etc/wpa_supplicant.conf
network={
    ssid="sol"
    psk=00a38f42e6681596e1a5a4c5ede9a15250fb2a01c21028c6d490bb3458b8ea00
}
network={
    ssid="wlan-sol2"
    psk=053633deb59038da9e9168e015fef97d3d54ae3794d4a12d31ee75a830cccec2
}
```

Bei der Verschlüsselung Ihres WPA-Passworts hilft `wpa_passphrase`. Wenn Sie das Passwort nicht als Parameter übergeben, erwartet das Kommando das Passwort an der Standardeingabe. Das Ergebnis dieses Kommandos können Sie direkt in `wpa_supplicant.conf` kopieren, wobei Sie die Zeile mit dem Passwort im Klartext tunlichst entfernen.

```
root# wpa_passphrase wlan-sol2 'Mein ganz geheimes Passwort!'
network={
    ssid="wlan-sol2"
    #psk="Mein ganz geheimes Passwort!"
    psk=020d93e2ddb2cdee51e800b977ff7d58fde47d0913cd394f2133648a147f513f
}
```

Jetzt können Sie `wpa_supplicant` starten. Das Kommando läuft, bis Sie es mit `Strg+C` beenden. Es kümmert sich um die Initialisierung der WLAN-Verbindung und in der Folge um die regelmäßige Erneuerung der Schlüssel für die Verbindung. Mit anderen Worten: Das Programm muss laufen, solange Sie die WLAN-Verbindung nutzen. Arbeiten Sie also in einer anderen Konsole weiter.

Kurz noch einige Anmerkungen zu den Optionen des Kommandos: `-i` gibt die Netzwerkschnittstelle an, `-c` die Konfigurationsdatei, deren Namen Sie frei wählen dürfen. `-D` gibt den oder die von Ihnen eingesetzten WLAN-Treiber an. Versuchen Sie es mit `nl80211,wext`: Damit werden sowohl die alten Linux-WLAN-Treiber als auch die neuen Implementierungen unterstützt. `man wpa_supplicant` liefert eine Liste aller unterstützten Treiber.

```
root# wpa_supplicant -iwlp2s0 -Dnl80211,wext -c /etc/wpa_supplicant.conf
Trying to associate with 00:13:46:b5:25:6e (SSID='sol' freq=0 MHz)
Associated with 00:13:46:b5:25:6e
WPA: Key negotiation completed with 00:13:46:b5:25:6e [PTK=TKIP GTK=TKIP]
CTRL-EVENT-CONNECTED - Connection to 00:13:46:b5:25:6e completed (auth)
[id=0 id_str=]
...
```

**Links** Weitere Tipps zum Umgang mit `wpa_supplicant` finden Sie auf den folgenden Seiten:

[https://w1.fi/wpa\\_supplicant](https://w1.fi/wpa_supplicant)

[https://wiki.ubuntuusers.de/WLAN/wpa\\_supplicant](https://wiki.ubuntuusers.de/WLAN/wpa_supplicant)

## 26.5 LAN-Konfigurationsdateien

Dieser Abschnitt stellt die wichtigsten Konfigurationsdateien für die Anbindung des Rechners an ein lokales Netzwerk vor. Leider gibt es nur für einen Teil dieser Dateien Regeln, die für alle wichtigen Distributionen einheitlich sind. Bei den restlichen Dateien beziehen sich die in diesem Abschnitt zusammengestellten Informationen auf Debian, Fedora, openSUSE, RHEL und Ubuntu (Stand: Sommer 2017). In der Regel werden Sie eine direkte Veränderung der Konfigurationsdateien vermeiden und stattdessen die Konfigurationwerkzeuge Ihrer Distribution einsetzen. Die in diesem Abschnitt vorgestellten distributionsspezifischen Konfigurationsdateien sind nur relevant, wenn Sie *nicht* den NetworkManager verwenden!

Für die meisten Beispiele in diesem Abschnitt gilt: Der zu konfigurierende Rechner heißt `uranus`. Er befindet sich in einem lokalen Netzwerk mit der Domain `sol`. Andere Rechner im lokalen Netz heißen `jupiter`, `saturn` etc. Das lokale Netz verwendet `192.168.0.*`-Adressen. Der lokale Rechner hat die IP-Adresse `192.168.0.2`. Der Gateway-Rechner im lokalen Netz hat die IP-Adresse `192.168.0.1`. Auf dem Gateway-Rechner läuft ein eigener Nameserver. Namen und Nummern haben natürlich nur Beispielcharakter.



## Basiskonfiguration

`/etc/hosts` enthält eine Liste bekannter IP-Adressen und der ihnen zugeordneten Namen. Die Datei muss auf jeden Fall die Daten der Loopback-Schnittstelle enthalten. Die Minimalvariante sieht so aus: `/etc/hosts`

```
# /etc/hosts (Minimalvariante)
127.0.0.1 localhost
```

Bei den meisten Linux-Distributionen ist `localhost` auch für IPv6 definiert. Die folgenden Zeilen zeigen die Defaulteinstellungen unter Fedora und RHEL:

```
# /etc/hosts (Defaultkonfiguration unter Red Hat und Fedora)
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
```

Bei einer statischen Netzwerkkonfiguration, z. B. auf einem Root-Server, kann `hosts` auch einen Eintrag mit der IP-Adresse und dem Hostnamen des Rechners enthalten:

```
# /etc/hosts (Fortsetzung, statische Konfiguration des lokalen Rechners)
...
192.168.0.2 uranus.sol uranus
```

Wenn Sie die anderen Rechner im lokalen Netz namentlich ansprechen möchten und es keinen lokalen Nameserver gibt, müssen Sie auch deren Namen in `/etc/hosts` angeben. Statt `ping 192.168.0.3` können Sie dann also einfach `ping saturn` ausführen, um die Verbindung zum Rechner `saturn` zu testen.

```
# /etc/hosts (Fortsetzung, statische Konfiguration anderer Rechner)
...
192.168.0.1 mars.sol mars
192.168.0.2 uranus.sol uranus
192.168.0.3 saturn.sol saturn
```

Analoge Einträge sind in den `/etc/hosts`-Dateien aller Rechner im lokalen Netz erforderlich. Je mehr Rechner es im lokalen Netzwerk gibt, desto mühsamer wird die Administration der `/etc/hosts`-Dateien. Aus diesem Grund empfiehlt es sich bei größeren Netzwerken, auf einem Rechner einen Nameserver einzurichten (siehe [Kapitel 27](#), »Internet-Gateway«). Der Nameserver weiß, wie alle anderen Rechner im Netzwerk heißen. Die Rechner im lokalen Netz können den Nameserver kontaktieren, um diese Information zu ermitteln. `/etc/hosts` benötigt dann nur die `localhost`-Zeilen.

`/etc/host.conf` gibt an, wie TCP/IP unbekannte IP-Adressen ermitteln soll. Die folgende Beispieldatei bestimmt, dass zuerst die Datei `/etc/hosts` ausgewertet (Schlüsselwort `hosts`) und danach der in `/etc/resolv.conf` angegebene Nameserver befragt werden soll (`bind`). Die zweite Zeile erlaubt, dass einem in `/etc/hosts` angegebenen Hostnamen mehrere IP-Adressen zugeordnet werden dürfen. `/etc/host.conf`

Diese Datei liegt bei fast allen Distributionen in der hier angegebenen Form vor und muss nicht verändert werden:

```
# /etc/host.conf
order hosts, bind
multi on
```

**Gateway** Es gibt keinen einheitlichen Standard, wie bzw. in welcher Datei die Gateway-Konfiguration erfolgt. In lokalen Netzen wird die Adresse des Gateways meist per DHCP übermittelt. Bei einer statischen Konfiguration sind je nach Distribution unterschiedliche Dateien verantwortlich.

Bei Debian und Ubuntu beschreibt `/etc/network/interfaces` alle Netzwerkschnittstellen. Bei statisch konfigurierten Schnittstellen wird das Gateway durch das Schlüsselwort `gateway` eingestellt:

```
# in /etc/network/interfaces (Debian, Ubuntu)
...
iface enp4s0 inet static
    address 192.168.0.2
    netmask 255.255.255.0
    gateway 192.168.0.1
```

Bei Red Hat bzw. Fedora enthält die Konfigurationsdatei für die Netzwerkschnittstelle die Variable `GATEWAY`:

```
# /etc/sysconfig/network-scripts/ifcfg-xxxx (Red Hat, Fedora)
GATEWAY=192.168.0.1
```

Bei SUSE erfolgt die Konfiguration zentral durch die folgende Datei:

```
# in /etc/sysconfig/network/routes (SUSE)
default 192.168.0.1 - -
```

## DNS-Konfiguration (`resolv.conf`)

`/etc/resolv.conf` `/etc/resolv.conf` steuert, wie die IP-Adressen für unbekannte Netzwerknamen (Hostnamen) ermittelt werden. »Unbekannt« bedeutet, dass die Namen nicht in `/etc/hosts` definiert sind.

Mit den Schlüsselwörtern `domain` und `search` wird erreicht, dass unvollständige Namen mit dem Domainnamen erweitert werden, also beispielsweise `jupiter` zu `jupiter.sol`. Das erhöht in erster Linie die Bequemlichkeit, weil lokale Hostnamen in verkürzter Form angegeben werden können. Bei `search` dürfen mehrere Domainnamen angegeben werden, bei `domain` aber nur einer. Dafür hat der `domain`-Name Vorrang vor den `search`-Namen, wird also zuerst getestet. Wenn wie hier nur ein einziger Domainname angegeben wird, kann auf die `domain`-Zeile verzichtet werden.

Die wichtigsten Einträge in der Datei `/etc/resolv.conf` werden mit dem Schlüsselwort `nameserver` eingeleitet: Damit können bis zu drei IP-Adressen von Nameservern angegeben werden. Diese Server werden immer dann angesprochen, wenn die IP-Adresse eines unbekanntem Rechnernamens ermittelt werden soll. Die Angabe eines Nameservers ist unbedingt erforderlich, damit Internetadressen in IP-Adressen aufgelöst werden können. Auf den meisten ADSL-Routern läuft ein lokaler DNS, der wiederum auf den DNS des Providers zurückgreift. In größeren lokalen Netzen gibt es zumeist eigene Nameserver.

```
# /etc/resolv.conf
domain sol           # Hostnamen gelten für .sol
search sol           # Hostnamen gelten für .sol
nameserver 192.92.138.35 # erster DNS
nameserver 195.3.96.67  # zweiter DNS (falls der erste ausfällt)
```

Wenn Sie IPv6 verwenden, müssen Sie auch die Adresse eines IPv6-Nameservers angeben. Falls Ihr Internet-Provider nur IPv4 unterstützt und Sie einen IPv6-Tunnel-Service verwenden, können Sie die Adresse eines öffentlichen IPv6-Nameservers angeben, z. B. den von Google:

```
# /etc/resolv.conf
...
nameserver 2001:4860:4860::8888
```

Bei vielen Distributionen wird `resolv.conf` dynamisch erzeugt:

- ▶ Wenn Ihre lokale Netzwerkverbindung (LAN, WLAN) mit DHCP konfiguriert ist, trägt das Script für den Verbindungsaufbau bzw. der NetworkManager die vom DHCP-Server übertragenen Nameserver-Adressen ein.
- ▶ Wenn eine Internetverbindung per PPP (ADSL, UMTS, VPN) hergestellt wird, trägt das Script für den Verbindungsaufbau automatisch die `nameserver`-Adressen Ihres Internet-Providers in `/etc/resolv.conf` ein.
- ▶ Ubuntu richtet ab Version 12.04 standardmäßig einen lokalen Nameserver ein. Dabei kommt das Programm `Dnsmasq` zum Einsatz, das Sie im nächsten Kapitel kennenlernen werden. Die Nameserver-Adresse lautet deswegen `127.0.0.1`, verweist also auf `localhost`. Um festzustellen, welche IP-Adresse der Nameserver tatsächlich hat, führen Sie `nmcli dev show | grep DNS` aus.

Wenn Sie den Nameserver manuell einrichten möchten, müssen Sie seine Adresse `/etc/network/interfaces` mit dem Schlüsselwort `dns-nameservers` angeben:

```
# Datei /etc/network/interfaces (ab Ubuntu 12.04)
...
auto enp4s0
  iface enp4s0 inet static
  ...
  dns-nameservers 10.0.17.1
```

Für die Verwaltung von `/etc/resolv.conf` ist das Paket `resolvconf` zuständig. Eine direkte Veränderung von `/etc/resolv.conf` ist nicht vorgesehen!

**resolv.conf vor  
Änderungen  
schützen**

Die automatische Anpassung von `resolv.conf` ist in den meisten Fällen zweckmäßig. Wenn Sie dies aber nicht wünschen, können Sie die automatische Veränderung in den meisten Fällen verhindern. Bei Netzwerkschnittstellen mit DHCP müssen Sie die Konfigurationsdatei `dhclient.conf` verändern:

```
# /etc/dhcp/dhclient.conf
...
supersede domain-name "sol";
prepend domain-name-servers 192.168.0.1;
```

Unter Ubuntu müssen Sie außerdem das Paket `resolvconf` deinstallieren, unter Raspbian `openresolv`.

Bei Red Hat bzw. Fedora müssen Sie dabei die `ifcfg-xxx`-Datei für die jeweilige Schnittstelle verändern:

```
# /etc/sysconfig/network-scripts/ifcfg-xxxx (Red Hat, Fedora)
PEERDNS=no
```

Bei SUSE ändern Sie die folgende Konfigurationsdatei:

```
# /etc/sysconfig/network/config (SUSE)
NETCONFIG_DNS_POLICY=""
```

## Hostname

Der aktuelle Hostname kann mit dem Kommando `hostname` ermittelt werden. Sofern der Hostname nicht durch DHCP eingestellt wird, erfolgt die Konfiguration bei fast allen aktuellen Distributionen in der Datei `/etc/hostname`. Zur Veränderung des Hostnamens verwenden Sie am besten das Kommando `hostnamectl set-hostname mein-neuer-hostname`. Beachten Sie, dass Hostnamen keine Leer- und Sonderzeichen enthalten dürfen. (Das Zeichen `-` ist erlaubt, aber nicht als erstes oder letztes Zeichen.)

**IDN** Umlaute und andere Sonderzeichen sind im Hostnamen eigentlich nicht vorgesehen. Bei internationalisierten Domainnamen (IDN) wird der Hostname gemäß dem Punycode-Verfahren in ASCII-Zeichen umgewandelt. Unter Linux hilft dabei das Kommando `idn`. Es befindet sich je nach Distribution im Paket `idn` oder `libidn`. Als Hostname gilt dann die resultierende Zeichenkette, die mit `xn-` beginnt. Erst im Webbrowser der Clients wird daraus dann die »richtige« Darstellung.

```
user$ idn hellö-world.com
xn--hell-world-hcb.com
root# hostnamectl set-hostname xn--hell-world-hcb.com
```

## Zuordnung zwischen Controllern und Netzwerkschnittstellen

Bei mehreren Netzwerkschnittstellen ist es oft schwierig, die Zuordnung zwischen den `ethN`- bzw. `enpNsM`-Devices und der physikalischen Hardware zu ermitteln. Je nach Hardware ist es möglich, die in die Buchse integrierte Leuchtdiode mit `ethtool -p enp4s0 10` für 10 Sekunden zum Blinken anzuregen. Wenn der Netzwerktreiber diese Operation nicht unterstützt, erhalten Sie die Fehlermeldung *operation not supported*.

Immer mehr Linux-Distributionen benennen Netzwerkschnittstellen so, dass sich die Device-Namen vorhandener Schnittstellen auch beim Hinzufügen weiterer Netzwerk-Hardware nicht ändern.

**Aktuelles  
Verfahren zur  
Device-  
Benennung**

Für die Benennung der Devices ist zumeist `systemd` in Kombination mit `udev`-Regeln verantwortlich. On-Board-Devices erhalten den Namen `enoN`, PCI-Express-Adapter den Namen `ensN`, externe Geräte den Namen `enpNsM` und WLAN-Adapter den Namen `wlpNsM`. Dabei beziehen sich `N` und `M` jeweils auf Hardware-Eigenschaften, z. B. auf den PCI-Slot.

Einigermaßen absurd ist die aktuelle Nomenklatur bei USB-WLAN-Adaptoren unter Fedora: Beispielsweise bezeichnet `wlp0s29u1u7` einen USB-WLAN-Adapter, der in eine ganz bestimmte USB-Buchse gesteckt wurde. Stecken Sie den Adapter in eine andere Buchse, erhält er auch einen anderen Device-Namen. Über die Sinnhaftigkeit dieses Verhaltens kann man geteilter Meinung sein.

Ubuntu verhält sich in dieser Hinsicht etwas zweckmäßiger und verbindet den Schnittstellennamen mit einer internen ID-Nummer des USB-Adapters. Der resultierende Name `wlx7cdd904ca740` ist aber noch unhandlicher. Unverständlich ist auch, warum nun offensichtlich jede Distribution eigene Verfahren zur Benennung der Schnittstellen verwendet.

Hintergrundinformationen zur Benennung der Schnittstellen können Sie hier nachlesen:

<https://freedesktop.org/wiki/Software/systemd/PredictableNetworkInterfaceNames>

Es gibt noch einige Distributionen, die traditionelle Device-Namen für Netzwerkadapter verwenden, also `eth0`, `eth1`, `wlan0` etc. Dazu zählen unter anderem Debian bis Version 8 und Raspbian.

**Bisheriges  
Verfahren zur  
Device-  
Benennung**

Natürlich waren die Entwickler der Distributionen auch bisher bestrebt, die Namen einmal genutzter Netzwerkschnittstellen später nicht mehr zu ändern – auch dann nicht, wenn der Rechner zwischenzeitlich mit einem neuen Netzwerkadapter ausgestattet wurde. Das alte Konzept bestand dahin, bei jedem neu angeschlossenen Adapter die MAC-Adresse auszulesen. Anschließend wurden durchnummerierte Device-Namen fix mit diesen MAC-Adressen verbunden. Die während der Nutzungsdauer

einer Distribution zuerst erkannte Ethernet-Schnittstelle bekam den Namen eth0, die nächste eth1 usw. Die Zuordnung zwischen den MAC-Adressen und den Device-Namen wurde in die Datei `net_persistent_names.rules` eingetragen. Diese Datei kann beispielsweise so aussehen:

```
# Datei /etc/udev/rules.d/70-persistent-net.rules
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="
  "00:16:17:cd:c3:81", ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="
  "00:14:6c:8e:d9:71", ATTR{type}=="1", KERNEL=="eth*", NAME="eth1"
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="
  "00:4f:4e:0f:8e:a0", ATTR{type}=="1", KERNEL=="eth*", NAME="eth2"
```

## 26.6 Distributionspezifische Konfigurationsdateien

In diesem Abschnitt geht es darum, eine bleibende Netzwerkkonfiguration durchzuführen, ohne Konfigurationswerkzeuge zu Hilfe zu nehmen. Die hier beschriebenen Vorgehensweisen sind insbesondere dann zweckmäßig, wenn Sie einen Server einrichten oder eine virtuelle Maschine konfigurieren. In beiden Fällen stehen häufig weder eine grafische Benutzeroberfläche noch die dazugehörigen Administrationswerkzeuge zur Verfügung. Auch auf die Hilfe eines DHCP-Servers müssen Sie oft verzichten. Daher müssen Sie die distributionspezifischen Dateien zur Netzwerkkonfiguration selbst editieren.

### Wenn sich zwei streiten ...

Auf vielen Distributionen gibt es standardmäßig *zwei* Verfahren zur Netzwerkkonfiguration: den NetworkManager und ein distributionspezifisches Verfahren. Problematisch wird es, wenn beide Verfahren versuchen, eine Schnittstelle zu steuern. Deswegen fasst der folgende Abschnitt Ratschläge zusammen, wie Sie den NetworkManager und gegebenenfalls andere Tools deaktivieren oder deinstallieren, die einer manuellen Konfiguration im Wege stehen.

Grundsätzlich wäre es denkbar, dass sich in Zukunft einzelne Distributionen von ihren distributionspezifischen Scripts ganz verabschieden und ausschließlich auf den NetworkManager setzen. Bisher gibt es aber keine Anzeichen in diese Richtung.

### NetworkManager deaktivieren

Vor dem Start einer manuellen Netzwerkkonfiguration sollten Sie sicherstellen, dass der NetworkManager nicht läuft:

```
root# systemctl stop NetworkManager
root# systemctl disable NetworkManager
```

Sie können auch einen Schritt weiter gehen und das Programm komplett deinstallieren. Dazu führen Sie je nach Distribution eines der folgenden Kommandos aus:

```
root# apt remove network-manager
root# dnf remove NetworkManager
root# yum remove NetworkManager
```

Unter Ubuntu sollten Sie gleich auch das Paket `resolvconf` deinstallieren. Das ist insbesondere dann zweckmäßig, wenn Sie `/etc/resolv.conf` direkt verändern oder das Programm `Dnsmasq` selbst verwenden möchten. **Ubuntu**

```
user$ sudo apt remove NetworkManager resolvconf
user$ sudo reboot
```

Bei Fedora müssen Sie außerdem explizit das Init-Script für die traditionelle Netzwerkconfiguration aktivieren. Interessanterweise handelt es sich dabei um ein Init-V-Script, obwohl Fedora ja schon lange `systemd` als Init-System verwendet. **Fedora**

```
root# systemctl start network
root# systemctl enable network
```

Das Init-V-Script `network` liegt auch in CentOS 7 und RHEL 7 vor. Dort ist es standardmäßig aktiv, weswegen die obigen Kommandos nicht erforderlich sind. Die RHEL-Dokumentation führt aus, dass die in diesem Abschnitt beschriebenen Red-Hat-spezifischen Konfigurationsdateien prinzipiell kompatibel zum `NetworkManager` sind, dass die hier empfohlene Deinstallation des `NetworkManagers` also nicht zwingend erforderlich ist: **CentOS und RHEL**

[https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/7/html/Networking\\_Guide/sec-NetworkManager\\_and\\_the\\_Network\\_Scripts.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Networking_Guide/sec-NetworkManager_and_the_Network_Scripts.html)

Ich bin diesbezüglich, also in Hinblick auf die Verwendung des `NetworkManagers` auf Servern, allerdings ein gebranntes Kind. Nur was nicht da ist, kann definitiv keine Konflikte verursachen.

Unter Raspbian ist der `NetworkManager` nicht installiert. Allerdings verwaltet `openresolv` ähnlich wie unter Ubuntu `resolvconf` die Datei `/etc/resolv.conf` und startet selbstständig eine lokale Nameserver-Instanz (`Dnsmasq`). Außerdem ist der DHCP-Client `dhcpcd` installiert, der durch `systemd` automatisch gestartet wird. Abhilfe schaffen diese Kommandos: **Raspbian**

```
pi$ sudo apt remove openresolv
pi$ sudo systemctl stop dhcpcd
```

## Fedora, CentOS und Red Hat

Unter Fedora und Red Hat erfolgt die Konfiguration jeder Netzwerkschnittstelle durch die Datei `/etc/sysconfig/network-scripts/ifcfg-xxx`, wobei `xxx` der Name der Netzwerkschnittstelle ist. Ich verwende in den folgenden Beispielen `enp4s0`, aber Sie müssen den Schnittstellennamen natürlich an Ihre Gegebenheiten anpassen! Im Regelfall existiert die Konfigurationsdatei bereits, und Sie müssen lediglich einige Einstellungen verändern. Die MAC-Adresse des Netzwerkadapters ermitteln Sie bei Bedarf mit `ip addr`.

**DHCP-Konfiguration** Wenn der Rechner die Netzwerkparameter via DHCP bezieht, richten Sie einfach die Datei `ifcfg-enp4s0` wie folgt ein:

```
# Datei /etc/sysconfig/network-scripts/ifcfg-enp4s0
DEVICE=enp4s0
HWADDR=xx:xx:xx:xx:xx:xx
NM_CONTROLLED=no
ONBOOT=yes
BOOTPROTO=dhcp
TYPE=Ethernet
```

**Statische Konfiguration** Bei einer statischen Konfiguration muss die Datei dem folgenden Muster entsprechen, wobei Sie die IP-Adressen durch eigene Werte ersetzen müssen. `PREFIX` gibt die Anzahl der gesetzten Bits der Netzmaske an. 24 entspricht also der Maske 255.255.255.0. Die folgende Beispielkonfiguration gilt somit für ein Netz mit IP-Adressen der Form 10.0.17.\*:

```
# Datei /etc/sysconfig/network-scripts/ifcfg-enp4s0
DEVICE=enp4s0
HWADDR=xx:xx:xx:xx:xx:xx
NM_CONTROLLED=no
ONBOOT=yes
BOOTPROTO=none
TYPE=Ethernet
IPADDR=10.0.17.33
PREFIX=24
GATEWAY=10.0.17.1
```

Die Gateway-Adresse können Sie statt in `ifcfg-xxx` auch in `/etc/sysconfig/network` einstellen. Das ist dann zweckmäßig, wenn es ein zentrales Gateway für alle Netzwerkschnittstellen gibt.

**Nameserver** Die Nameserver-Konfiguration führen Sie direkt in der Datei `/etc/resolv.conf` durch:

```
# /etc/resolv.conf
nameserver 10.0.17.1          # erster DNS
nameserver 10.0.17.2        # zweiter DNS
nameserver 2001:4860:4860::8888 # öffentlicher IPv6-Nameserver von Google
```



Wenn Sie IPv6 nutzen möchten, müssen Sie in `ifcfg-xxx` auch IPv6-Variablen initialisieren. Das folgende Listing zeigt die Einstellungen für eine automatische Konfiguration, bei der die Schnittstelle das *Router Advertisement* des IPv6-Routers auswertet. Dieses Verfahren ermöglicht eine Art Selbstkonfiguration von IPv6-Netzen ohne einen eigenen DHCP-Server.

IPv6-Konfiguration

```
# Datei /etc/sysconfig/network-scripts/ifcfg-enp4s0 (automatische Konfiguration)
...
IPV6INIT=yes
IPV6_AUTOCONF=yes
```

Wenn auf dem IPv6-Router ein DHCP-Server läuft, sieht die Konfiguration so aus:

```
# Datei /etc/sysconfig/network-scripts/ifcfg-enp4s0 (DHCPv6-Konfiguration)
...
IPV6INIT=yes
DHCPV6C=yes
```

Zu guter Letzt folgt hier noch die Variante für eine statische Konfiguration:

```
# Datei /etc/sysconfig/network-scripts/ifcfg-enp4s0
...
IPV6INIT=yes
IPV6ADDR=2a01:4f8:161:107::2/64
IPV6_DEFAULTGW=fe80::1
```

In Fedora, CentOS 7 und RHEL 7 können Sie in `ifcfg-xxx` auch die gewünschte Firewall-Zone für die Schnittstelle angeben:

Firewall-Zonen

```
# Datei /etc/sysconfig/network-scripts/ifcfg-enp4s0
...
ZONE="trusted"
```

Ohne diese Angabe verwendet das Firewall-System von Fedora automatisch die Defaultzone `public`. Hintergründe zur Firewall-Konfiguration von Fedora und RHEL folgen in [Kapitel 37](#), »Firewalls«.

In Fedora, CentOS 7 und RHEL 7 wird der Hostname durch die Datei `/etc/hostname` eingestellt. Zur Veränderung verwenden Sie das `systemd`-Kommando `hostnamectl set-hostname neuename`.

Hostname

```
# Datei /etc/hostname (Fedora, CentOS 7 und RHEL 7)
uranus.sol
```

Im Regelfall ist es zweckmäßig, die Zuordnung der IP-Adresse des Rechners zu seinem Hostnamen darüber hinaus auch in `/etc/hosts` einzutragen.

**Konfiguration aktivieren** Zuletzt starten Sie das Init-Script, das für die manuelle Netzwerkkonfiguration verantwortlich ist:

```
root# systemctl start network      (gilt sofort)
root# systemctl disable network    (gilt ab dem nächsten Neustart)
```

Wenn Sie nur eine einzelne Schnittstelle aktivieren bzw. wieder deaktivieren möchten, verwenden Sie die `ifup`- und `ifdown`-Scripts:

```
root# ifup enp4s0
root# ifdown enp4s0
```

**Dokumentation** Die Syntax der Datei `/etc/sysconfig/network-scripts/ifcfg-xxx` sowie vieler weiterer Dateien im Verzeichnis `/etc/sysconfig` ist in dieser Datei dokumentiert:

```
/usr/share/doc/initscripts/sysconfig.txt
```

## Debian und Ubuntu

**/etc/network/interfaces** Für die Konfiguration aller Schnittstellen ist die Datei `/etc/network/interfaces` zuständig. Die Syntax ist einfach: Jede Schnittstelle, die beim Rechnerstart aktiviert werden soll, muss durch `auto` `name` genannt werden. `iface` `name` `optionen` beschreibt die Basiskonfiguration der Schnittstelle. Bei einer statischen Konfiguration folgen in den weiteren Zeilen die Parameter `address`, `netmask` und `gateway`.

### auto versus allow-hotplug

Bei den meisten PC-Distributionen werden immer zu aktivierende Schnittstellen mit `auto` `name` gekennzeichnet. Unter Raspbian ist stattdessen `allow-hotplug` `name` üblich. Laut `man interfaces` ist `auto` für Schnittstellen gedacht, die schon während des Systemstarts aktiviert werden, während `allow-hotplug` Schnittstellen kennzeichnet, die im laufenden Betrieb bei Bedarf durch das Hotplug-System konfiguriert werden (also durch `udev`-Regeln). `allow-hotplug` ist deswegen insbesondere für USB-(WLAN-)Adapter geeignet, die während des Systemstarts erst später initialisiert werden als eingebaute WLAN-Adapter.

**DHCP-Konfiguration** Wenn der Rechner die Netzwerkparameter via DHCP beziehen soll, umfasst die gesamte Konfiguration nur vier Zeilen! Die ersten beiden Zeilen aktivieren die Loopback-Schnittstelle, die immer erforderlich ist. Sie dient zur rechnerinternen Netzwerkkommunikation. Die zwei weiteren Zeilen aktivieren die Schnittstelle `enp4s0`.

```
# /etc/network/interfaces
auto lo
iface lo inet loopback
```

```
# dynamische Verbindung zu einem DHCP-Server,
# der die Eckdaten des Internetzugangs vermittelt
auto enp4s0
iface enp4s0 inet dhcp
```

Wenn die Verbindung in das Internet statisch konfiguriert wird, enthält die `iface`-Zeile das Schlüsselwort `static`. Die Netzwerkparameter werden in der Folge durch mehrere Schlüsselwörter angegeben, deren Bedeutung selbsterklärend ist.

**Statische  
Konfiguration**

```
# /etc/network/interfaces
auto lo
iface lo inet loopback

# statische Netzwerkkonfiguration
auto enp4s0
iface enp4s0 inet static
    address      211.212.213.37
    netmask      255.255.255.224
    gateway      211.212.213.1
```

Unter Ubuntu können Sie in `interfaces` auch die Adressen von einem oder mehreren DNS-Servern angeben. Für die Auswertung des Schlüsselworts `dns-nameservers` ist das Paket `resolvconf` verantwortlich. `dns-nameservers` funktioniert nur, wenn dieses Paket installiert ist.

**Nameserver**

```
# /etc/network/interfaces (Ubuntu ab Version 12.04, sofern resolvconf
#                               installiert ist)
...
auto enp4s0
iface enp4s0 inet static
    ...
    dns-nameservers 211.222.233.244 212.223.234.245
```

In Debian bzw. auch unter Ubuntu, wenn Sie `resolvconf` deinstalliert haben, müssen Sie die Nameserver-Adressen direkt in `/etc/resolv.conf` angeben:

```
# /etc/resolv.conf (Debian)
nameserver 211.222.233.244 # erster DNS
nameserver 212.223.234.245 # zweiter DNS
```

Wenn Sie auch IPv6 nutzen möchten, definieren Sie die betreffende Schnittstelle in `/etc/network/interfaces` einfach ein zweites Mal mit dem Schlüsselwort `inet6`. Das Schlüsselwort `auto` gibt an, dass die IPv6-Konfiguration das sogenannte *Router Advertisement* des Gateways bzw. IPv6-Routers berücksichtigt. Dieses Verfahren ermöglicht eine Art Selbstkonfiguration von IPv6-Netzen.

**IPv6-  
Konfiguration**

```
# /etc/network/interfaces (automatische IPv6-Konfiguration)
...
auto enp4s0
iface enp4s0 inet dhcp
iface enp4s0 inet6 auto
```

Wenn das IPv6-Gateway einen DHCPv6-Server verwendet, lautet die korrekte Methode `dhcp`. Wenn außerdem die Router-Adresse per *Router Advertisement* konfiguriert werden soll, ist die Zusatzoption `accept_ra 1` erforderlich. Das ist beispielsweise der Fall, wenn Sie als DHCP-Server `dnsmasq` mit der Option `enable-ra` einsetzen.

```
# /etc/network/interfaces (DHCPv6-Konfiguration)
...
auto enp4s0
iface enp4s0 inet dhcp
iface enp4s0 inet6 dhcp
    accept_ra 1
```

Bei einer statischen Konfiguration muss `interfaces` so aussehen:

```
# /etc/network/interfaces (statische IPv6-Konfiguration)
...
auto enp4s0
iface enp4s0 inet static
    ... (IPv4-Konfiguration wie bisher)
iface enp4s0 inet6 static
    address 2a01:4f8:161:107::2
    netmask 64
    gateway fe80::1
```

**WLAN und WPA** Die Konfiguration von WLAN-Schnittstellen erfolgt grundsätzlich wie die von Ethernet-Schnittstellen. Dabei steuern Sie das Zusammenspiel mit `wpa_supplicant` mit `wpa`-Parametern. Im einfachsten Fall kann die Konfiguration so aussehen:

```
# /etc/network/interfaces (manuelle WLAN-Konfiguration mit WPA)
auto wlan0
iface wlan0 inet dhcp
    wpa-ssid "wlan-name"
    wpa-psk "strenge-geheim"
```

Wenn Sie mehr WPA-Parameter einstellen möchten, ist es zumeist besser, dies in einer eigenen Datei zu tun, deren Ort Sie mit `wpa-conf` angeben:

```
# /etc/network/interfaces (manuelle WLAN-Konfiguration mit WPA)
auto wlan0
iface wlan0 inet dhcp
    wpa-conf /etc/wpa.conf
```

Das folgende Beispiel zeigt die Syntax von `wpa_supplicant`. Anstelle des Passworts im Klartext wurde dabei mit `wpa_passphrase` ein äquivalenter, aber nicht so leicht lesbarer hexadezimaler Code angegeben (siehe auch [Abschnitt 26.5](#), »LAN-Konfigurationsdateien«):

```
# Datei /etc/wpa.conf
network={
    ssid="wlan-name"
    psk=113df295cd91605ce30c00d16b82bf5eb334b0e47adc9aafc153459a731fa42
    proto=RSN
    key_mgmt=WPA-PSK
    pairwise=CCMP
    auth_alg=OPEN
}
```

Wenn es wie unter Raspbian eine grafische Benutzeroberfläche zur WLAN-Konfiguration gibt, um die Verbindungsdaten zu *mehreren* WLAN-Netzen zu verwalten, müssen Sie das Schlüsselwort `wpa-roam` verwenden:

```
# /etc/network/interfaces (WLAN-Defaultkonfiguration unter Raspbian)
allow-hotplug wlan0
iface wlan0 inet manual
    wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

Die Zeile `allow-hotplug` erlaubt es anderen Programmen, die WLAN-Schnittstelle zu steuern. Die mit `wpa-roam` beginnende Zeile gibt an, in welcher Datei die WPA-Schlüssel verschiedener WLAN-Netze gespeichert sind. Sobald ein passendes Netz in Reichweite ist, wird automatisch eine Verbindung hergestellt, wobei die IP-Daten dann via DHCP bezogen werden. Die Anweisung `iface default inet dhcp` besagt, dass jedes in `wpa_supplicant.conf` definierte Netzwerk standardmäßig DHCP zur IP-Konfiguration verwenden soll.

Eine Menge weiterer `wpa`-Parameter für die `interfaces`-Datei sind in dieser Datei dokumentiert:

```
/usr/share/doc/wpasupplicant/README.modes.gz
```

Mit den Schlüsselwörtern `up` und `down` können Sie in die `interfaces`-Datei Kommandos einbauen, die automatisch ausgeführt werden, wenn eine Schnittstelle aktiviert bzw. deaktiviert wird. Ein Beispiel für eine derartige Konfiguration finden Sie in [Abschnitt 27.3](#), »Masquerading (NAT)«, wo es um die Konfiguration eines Internet-Gateways geht.

Scripts ausführen

Falls Sie den Hostnamen neu einstellen möchten, führen Sie die Änderungen in `/etc/hostname` und eventuell auch in `/etc/hosts` durch.

Hostname

**Änderungen aktivieren** Um eine neue Konfiguration für eine Schnittstelle zu aktivieren, führen Sie für diese Schnittstelle `ifdown` und `ifup` durch:

```
root# ifdown enp4s0
root# ifup enp4s0
```

Diese Debian/Ubuntu-spezifischen Kommandos sind Teil des Pakets `ifupdown`. Das Kommando `ifup -a` wertet `/etc/network/interfaces` aus und aktiviert alle auto-Schnittstellen. Sofern Schnittstellen via DHCP konfiguriert werden, greift `ifup` auf das Kommando `dhclient` zur Übertragung und Auswertung der DHCP-Daten zurück. Für die Konfiguration ist `/etc/dhcp3/dhclient.conf` zuständig.

## SUSE

Bei SUSE-Distributionen empfiehlt es sich, die Konfiguration mit YaST durchzuführen. YaST steht bei Minimalinstallationen auch im Textmodus zur Verfügung. Im Modul **NETZWERKGERÄTE • NETZWERKEINSTELLUNGEN** aktivieren Sie die globale Option **WICKED** und deaktivieren so den **NetworkManager**. Anschließend wechseln Sie in das Dialogblatt **ÜBERSICHT** und bearbeiten dort die Einstellungen der Netzwerkadapter.

## networkd (systemd)

Die `systemd`-Entwickler haben sich nicht nur die Aufgabe gestellt, den `Init`-Prozess zu modernisieren, sie möchten auch die Konfiguration unterschiedlicher Linux-Systeme auf einen gemeinsamen Nenner bringen. Dabei haben sie auch vor der Netzwerkkonfiguration nicht Halt gemacht.

Als Ort für die Konfigurationsdateien ist das Verzeichnis `/etc/systemd/network` vorgesehen, um die Auswertung kümmert sich der Hintergrundprozess `systemd-networkd`. Die Syntax der Konfigurationsdateien wirkt gut durchdacht und dokumentiert; Sie können sie mit `man systemd.network` nachlesen. Sehr hilfreich ist auch die Wiki-Dokumentation von Arch Linux:

<https://wiki.archlinux.org/index.php/systemd-networkd>

Die folgenden Zeilen zeigen beispielhaft die Einrichtung einer Ethernet-Schnittstelle mit statischer IP-Adresse:

```
# Datei /etc/systemd/network/static.network
[Match]
Name=enp2s0
```

```
[Network]
Address=10.0.0.15/24
Gateway=10.0.0.138
DNS=10.0.0.138
```

Obwohl die Infrastruktur zur systemd-Netzwerkconfiguration bereits in vielen Distributionen standardmäßig zur Verfügung steht, nutzen aktuell nur wenige Distributionen diese Möglichkeiten. Das könnte sich aber mit der Einführung von `netplan` in Ubuntu 17.10 ändern (siehe den folgenden Abschnitt).

## netplan

Ubuntu plant mit Version 17.10 auf das Konfigurationssystem `netplan` (Paketname `nplan`) umzusteigen. Das `ifupdown`-Paket mit der Konfigurationsdatei `/etc/network/interfaces` und den Kommandos `ifup` und `ifdown` wäre damit obsolet. Obwohl diese Entscheidung in den Mailing-Listen der Ubuntu-Entwickler auf geringe Begeisterung gestoßen ist, scheint sie endgültig zu sein.

Die Ubuntu-Eigenentwicklung `netplan` ist genau genommen ein Netzwerk-Backend, das auf Desktop-Systemen mit dem `NetworkManager` kommuniziert und auf Server-Systemen mit dem gerade vorgestellten Programm `networkd` aus der `systemd`-Welt. Die Konfiguration erfolgt durch `*.yaml`-Dateien in `/etc/netplan`. (YAML ist eine einfache Syntax für Konfigurationsdateien, siehe <https://de.wikipedia.org/wiki/YAML>.)

Auf Desktop-Systemen reicht die folgende Konfigurationsdatei aus, damit alle Netzwerkschnittstellen durch den `NetworkManager` gesteuert werden:

```
# Datei /etc/netplan/01-network-manager-all.yaml
# der NetworkManager soll sich um alle Netzwerkschnittstellen
# kümmern
network:
  version: 2
  renderer: NetworkManager
```

Damit eine Ethernet-Schnittstelle via DHCP konfiguriert wird, ist die folgende `*.yaml`-Datei erforderlich:

```
# Datei /etc/netplan/my-config1.yaml
# Schnittstelle enp0s3 via DHCP konfigurieren
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: true
```

Weitere Konfigurationsbeispiele sowie Hintergrundinformationen zu netplan liefern man netplan sowie die folgenden Webseiten:

<https://wiki.ubuntu.com/Netplan>

<https://wiki.ubuntu.com/MigratingToNetplan>

<https://git.launchpad.net/netplan/tree/doc/netplan.md>

## 26.7 Zeroconf und Avahi

Ich gehe in diesem Buch in der Regel davon aus, dass Sie die Rechner in Ihrem Netzwerk entweder selbst konfigurieren oder die IP-Konfiguration von einem zentralen Router oder DHCP-Server beziehen. Daneben gibt es aber noch einen dritten Weg: die automatische Konfiguration durch Zeroconf.

Bei diesem Verfahren tauschen alle im Netzwerk verbundenen Rechner ihre Konfigurationsdaten aus. Neu an das Netzwerk angeschlossene Rechner bzw. Geräte konfigurieren sich anhand dieser Informationen selbst so, dass sie ohne Konflikte mit den anderen Geräten kommunizieren können. Die automatisch konfigurierten Rechner verwenden Adressen aus dem IP-Bereich 169.254.\*.\* sowie Hostnamen, die auf .local enden. Die Zeroconf-Kommunikation erfolgt über den UDP-Port 5454. Damit Zeroconf funktioniert, darf dieser Port innerhalb des LANs nicht durch eine Firewall blockiert werden!

Zeroconf wurde zuerst von Apple unter dem Namen *Rendezvous* implementiert. Dieses Projekt wurde später in *Bonjour* umgetauft und steht auch für Windows zur Verfügung. Diese Implementierung liegt zwar als Open-Source-Code vor, die Lizenz ist aber nicht GPL-kompatibel. Aus diesem Grund entstand für Linux ein eigenes Zeroconf-Projekt unter dem Namen Avahi, dessen Code unabhängig von Bonjour ist. Als Lizenz kommt die LGPL zum Einsatz. (Die Entstehungsgeschichte des merkwürdigen Namens Avahi ist mir nicht bekannt.)

Zeroconf-kompatible Programme können nun alle anderen im Netzwerk sichtbaren Zeroconf-Rechner und deren Ressourcen anzeigen, z. B. Netzwerkverzeichnisse, SSH-, HTTP- und FTP-Server. Damit ist es ohne explizite Konfiguration möglich, zwei oder mehr Rechner in ein Netzwerk zu integrieren und Daten auszutauschen.

Allgegenwärtige ADSL- und WLAN-Router machen Zeroconf eigentlich weitgehend überflüssig. Dass Avahi-Pakete dennoch standardmäßig von vielen Linux-Distributionen installiert werden, liegt eher an den Browsing-Funktionen: Dass sich die Rechner gegenseitig sehen und namentlich kennen, ist ganz losgelöst von der Art der Netzwerkkonfiguration ein großer Vorteil. Außerdem nutzt Apple in all seinen Geräten Bonjour. Wenn Sie möchten, dass Ihr Linux-Rechner für Apple-Geräte sichtbar ist (z. B.



für die AirPrint-Funktion), muss auf dem Rechner der Avahi-Dämon laufen. Weitere Informationen und Tipps finden Sie auf den folgenden Websites:

<https://avahi.org>

<https://wiki.ubuntuusers.de/Avahi>

#### Zeroconf ist nur für IPv4 erforderlich

Zeroconf ist ein IPv4-spezifisches Verfahren. In IPv6 besteht keine Notwendigkeit für Zeroconf, weil jeder Netzwerkschnittstelle automatisch eine Link-Local-IPv6-Adresse zugewiesen wird. Diese Adresse wird für IPv6-interne Konfigurationaufgaben benötigt, kann aber auch für Anwendungen in der Art von Zeroconf verwendet werden.

Für die Kommunikation zwischen den Avahi-Rechnern ist der Dienst `avahi-daemon` zuständig. Die Konfiguration erfolgt durch `/etc/avahi/avahi-daemon.conf`, wobei Sie die Grundeinstellungen zumeist beibehalten können. Die einzige Ausnahme ist oft die Variable `enable-dbus`: Sie steuert, ob Avahi den Kommunikationsmechanismus zulässt. Einige Avahi-kompatible Programme setzen DBUS voraus. Um DBUS zu aktivieren, ändern Sie `avahi-daemon.conf` wie folgt:

**avahi-daemon**

```
# /etc/avahi/avahi-daemon.conf
[server]
...
enable-dbus=yes
```

Anschließend starten Sie den Dienst neu:

```
root# systemctl restart avahi-daemon
```

Wenn Sie externe Rechner mit gewöhnlichen, nicht Avahi-kompatiblen Netzwerkprogrammen über deren `.local`-Namen ansprechen möchten (z.B. mit `ping merkur.local`), müssen Sie das Paket `avahi-dnscconfd` installieren und den gleichnamigen Netzwerkdämon starten:

**Namensauflösung**

```
root# apt/dnf/yum install avahi-dnscconfd
root# systemctl start avahi-dnscconfd
root# systemctl enable avahi-dnscconfd
```

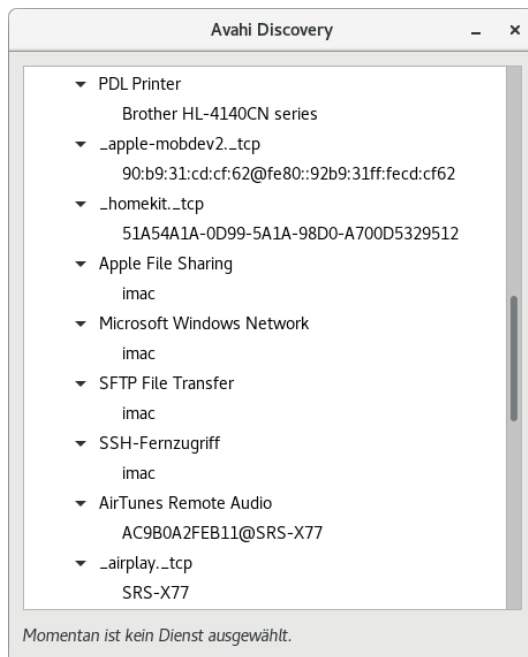
Dabei handelt es sich um eine Art Nameserver für Avahi-Hostnamen. Sie brauchen diesen Dämon nicht, wenn in Ihrem Netzwerk ohnedies ein Nameserver läuft.

Damit alle Programme bei der Namensauflösung auf `avahi-dnscconfd` zurückgreifen, müssen Sie dafür sorgen, dass die Bibliothek `libnss-mdns` installiert ist und dass die `hosts:-`Zeile in `/etc/nsswitch.conf` das Schlüsselwort `mdns4` enthält. Bei einigen Distributionen ist dies standardmäßig der Fall.

```
# in /etc/nsswitch.conf
...
hosts: files dns mdns4
...
```

**Browsing** Nach diesen Vorbereitungsarbeiten können Sie ausprobieren, welche Rechner bzw. Dienste Avahi in Ihrem Netz kennt. Dabei helfen das Konsolenkommando `avahi-browse -a -t` bzw. dessen grafische Entsprechung *Avahi Discovery* (Programmname `avahi-discover`, siehe [Abbildung 26.9](#)). Die Kommandos müssen bei vielen Distributionen extra installiert werden und befinden sich unter Fedora beispielsweise in den Paketen `avahi-tools` und `avahi-ui-tools`.

Der Gnome-Dateimanager zeigt in der Netzwerkansicht standardmäßig alle Avahi-Rechner an. Der KDE-Dateimanager bietet unter der Adresse `zeroconf:/` eine ähnliche Funktion, sofern die entsprechende Erweiterung installiert ist (je nach Distribution z. B. aus dem Paket `kde-zeroconf`). Auch diverse Messaging- und Multimedia-Anwendungen unterstützen Zeroconf.



**Abbildung 26.9** Netzwerkressourcen mit Avahi-Discovery entdecken

# Kapitel 27

## Internet-Gateway

An dieser Stelle beginnen mehrere Kapitel, die sich mit der Konfiguration eines Linux-Servers für das lokale Netzwerk (LAN) beschäftigen. Es wäre vermessen zu versuchen, die gesamte Bandbreite der Linux-Server-Konfiguration darin abzuhandeln. Beinahe jeder Abschnitt dieser Kapitel würde ein eigenes Buch rechtfertigen.

In den Kapiteln zur Server-Konfiguration versuche ich primär, Ihnen einen ersten Einstieg in dieses für fortgeschrittene Linux-Anwender so wichtige Thema zu geben. Ich richte mich dabei vor allem an Personen, die verhältnismäßig kleine lokale Netze verwalten, wobei sich in diesen Netzen durchaus auch Windows-Clients befinden dürfen.

In diesem Kapitel geht es darum, ein Internet-Gateway bzw. einen WLAN-Router für das lokale Netzwerk zusammenzustellen. Im Privatbereich erfüllt oft ein ADSL-Router diese Aufgabe. Das hat durchaus Vorteile: Die Konfiguration ist unkompliziert, das Gerät läuft lautlos und verbraucht wenig Strom. Kurz und gut: Wenn Sie mit Ihrem ADSL-Router zufrieden sind, behalten Sie ihn, und überspringen Sie dieses Kapitel! Sie ersparen sich damit eine Menge Zeit und Mühe.

ADSL-Router ...

Es gibt aber Fälle, in denen die Konfigurationsmöglichkeiten eines ADSL-Routers unzureichend sind. Das führt zum Inhalt dieses Kapitels, also zur Konfiguration eines eigenen Rechners, der die folgenden Funktionen des ADSL-Routers übernimmt: WLAN-Authenticator, Masquerading, DHCP-Server und Nameserver.

... versus eigenes Gateway

Jedes Internet-Gateway sollte unbedingt durch eine Firewall abgesichert werden. Diesbezügliche Informationen folgen in [Kapitel 37](#), »Firewalls«.

Firewall

### 27.1 Einführung

Dieses Kapitel beschreibt die Installation der folgenden Komponenten bzw. Dienste:

- ▶ **WLAN-Authenticator:** Sofern Ihr Internet-Gateway seine Dienste über ein WLAN anbieten soll, benötigen Sie ein Programm, das beim Verbindungsaufbau mit den Clients das Passwort überprüft. Diese Aufgabe übernimmt das Programm `hostapd`.

- ▶ **Masquerading/NAT:** Mit Masquerading können alle Clients in einem lokalen Netz mit dem Internet verbunden werden. Als Ausgangspunkt wird *ein* Rechner benötigt, der bereits über einen Internet-Anschluss verfügt. Alle anderen Rechner sollen nun über Ethernet-Kabel oder via WLAN mit diesem Rechner verbunden werden; sie können dann ebenfalls das Internet nutzen.
- ▶ **DHCP und Nameserver:** DHCP ermöglicht eine zentrale und einfache Verwaltung der IP-Adressen und der anderen Netzwerkparameter aller Clients. Ein lokaler Nameserver stellt sicher, dass die Clients gegenseitig ihre Namen kennen (Auflösung lokaler Namen in IP-Adressen). Außerdem fungiert das Programm als IP-Nummern-Cache, sodass wiederholte Internetzugriffe ein wenig beschleunigt werden.

Zur Realisierung der DHCP- und DNS-Funktionen stelle ich Ihnen das Programm `Dnsmasq` vor, das beide Funktionen in sich vereint und einfach zu konfigurieren ist. Nur für gehobene Anforderungen bzw. große oder komplexe lokale Netzwerke empfiehlt sich die Aufteilung der Funktionen auf die beiden Programme `dhcpd` und `bind`. Das erhöht nicht nur die Konfigurationsmöglichkeiten, sondern auch den dafür erforderlichen Aufwand.

**Hardware** Soweit das Internet-Gateway keine weiteren Aufgaben erfüllen soll, reicht ein minimal ausgestatteter PC vollkommen aus. Im Heimbereich äußerst beliebt sind Mini-computer wie der Raspberry Pi.

Eine Grundvoraussetzung sind zwei Netzwerkschnittstellen – eine, um den Rechner an den ADSL-Router anzuschließen, und eine zweite für das lokale Netzwerk:

- ▶ Im Privatbereich wird die zweite Schnittstelle zumeist ein WLAN-Adapter sein.
- ▶ In Firmennetzen ist hingegen eine zweite Ethernet-Schnittstelle sinnvoller, die mit einem Netzwerk-Switch verbunden wird. Dort beginnt die Verkabelung der Client-Rechner. WLAN-Funktionen können Sie in diesem Fall einfach durch einen WLAN-Access-Point anbieten, den Sie ebenfalls mit dem Switch verbinden.

Aus Sicherheitsgründen wird das Funknetz aber häufig vollkommen vom LAN getrennt. Das WLAN bietet dann nur einen Internetzugang an, während das LAN außerdem Zugriff auf die Dienste des Firmennetzes gibt, also auf Datei-Server, Drucker etc.

Auf der Suche nach einem leisen, kostengünstigen und energiesparenden Rechner werden Sie rasch feststellen, dass die zweite Netzwerkschnittstelle die größte Hürde ist: Mini-PCs haben oft nur eine Netzwerkschnittstelle und lassen sich selten durch PCI-Steckkarten erweitern. Ein möglicher Ausweg ist ein USB-Ethernet-Adapter. Vergewissern Sie sich aber vor dem Kauf, dass das Gerät Linux-kompatibel und ausreichend schnell ist!

Eine interessante Alternative sind OpenWrt-kompatible WLAN-Router. Das besondere Merkmal dieser Geräte besteht darin, dass die Firmware durch eine Linux-Version ersetzt werden kann und darf. Im Internet gibt es gleich eine ganze Palette von geeigneten Distributionen (Tomato, DD-WRT, OpenWrt etc.). Natürlich erfordert die Inbetriebnahme etwas Bastelarbeit. Dafür erhalten Sie ein billiges, vollkommen lautloses Gerät, dessen Stromverbrauch wesentlich geringer ist als bei marktüblichen Mini-PCs.

OpenWrt

<https://openwrt.org>

Bei den meisten Distributionen sind die Pakete der in diesem Kapitel beschriebenen Server-Dienste standardmäßig *nicht* installiert. Sie müssen die erforderlichen Pakete selbst installieren, z. B. mit `apt-get`, `dnf`, `yum` oder `zypper`.

Installation von Paketen

Dieses Kapitel richtet sich an fortgeschrittene Linux-Anwender. Es werden ausschließlich die Einstellungen der diversen Konfigurationsdateien beschrieben. Damit lernen Sie, wo sich alle relevanten Konfigurationsdateien befinden, und können eine fertige Konfiguration bei Bedarf relativ rasch auf einen anderen Server übertragen.

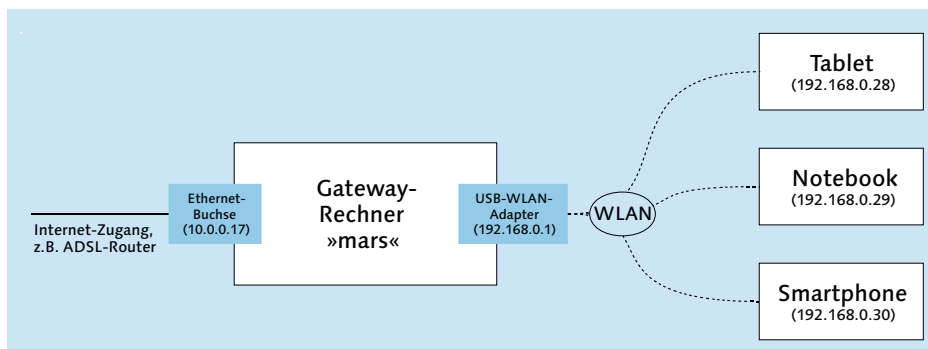
Konfiguration

Ein Router sollte das lokale Netzwerk durch eine Firewall gegen unerwünschte Zugriffe von außen absichern. Diesbezügliche Tipps gibt [Kapitel 37](#), »Firewalls«.

Sicherheit

## Topologie des Beispielnetzwerks

Um Ihnen die Orientierung in diesem Kapitel zu erleichtern, fasst [Abbildung 27.1](#) die Topologie des Beispielnetzwerks zusammen.



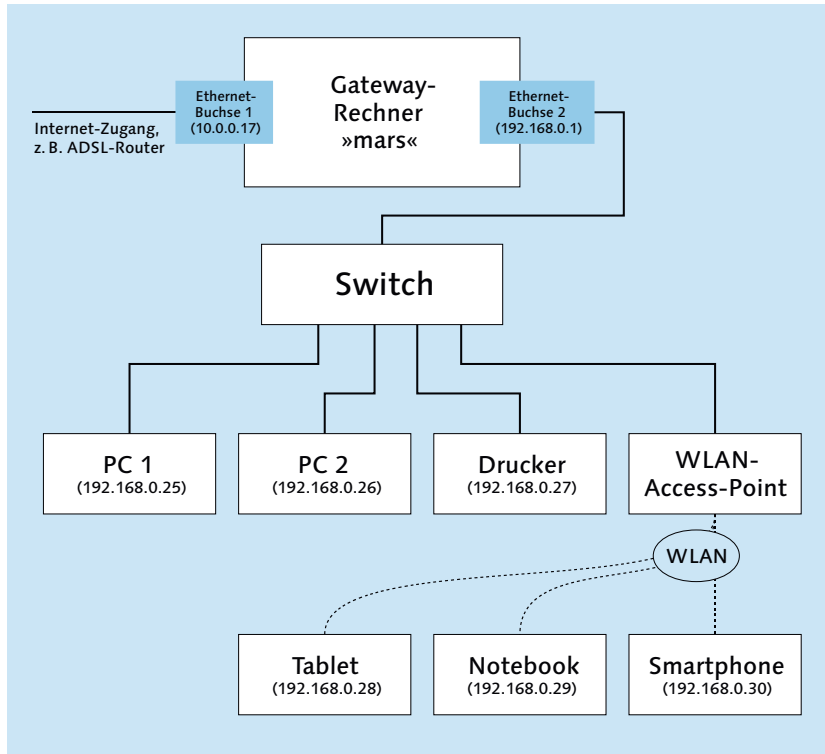
**Abbildung 27.1** Topologie des Beispielnetzwerks

Der Gateway-Rechner mit dem Hostnamen `mars` ist mit einem vorhandenen ADSL-Router verbunden, der z. B. die IP-Adresse `10.0.0.138` hat. Das Gateway mit dem Hostnamen `mars` bildet das lokale Netzwerk für den Adressraum `192.168.0.*`. `mars` ver-

wendet die IP-Adresse 192.168.0.1. Den WLAN-Clients werden dynamische Adressen zugewiesen (192.168.0.2 bis 192.168.0.253).

**Variante mit  
Netzwerk-Switch  
und WLAN-  
Access-Point**

Wenn das lokale Netz gleichermaßen über eine Ethernet-Verkabelung und WLAN verfügen soll, ist es zweckmäßiger, einen Gateway-Rechner mit zwei Ethernet-Anschlüssen zu verwenden. Das lokale Netzwerk wird über einen Switch erschlossen. Der WLAN-Zugang erfolgt über einen Access-Point, der ebenfalls mit dem Switch verbunden ist (siehe Abbildung 27.2).



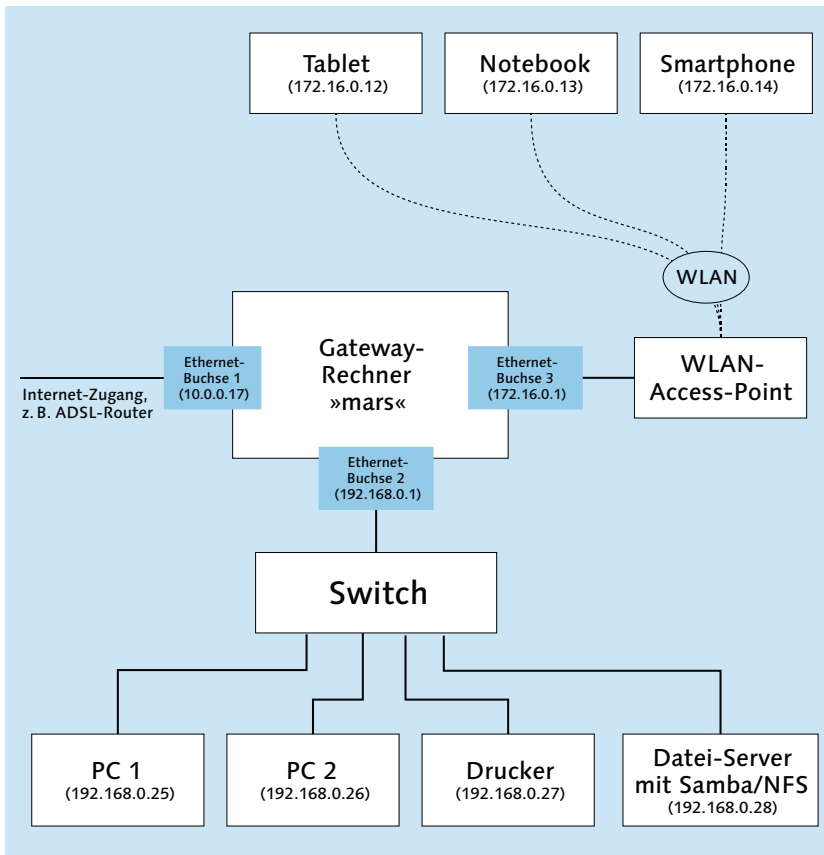
**Abbildung 27.2** Alternative Topologie mit Netzwerk-Switch und WLAN-Access-Point

Die Konfiguration von *mars* wäre sogar einfacher als die Anleitung in diesem Kapitel, weil *mars* nun keinen WLAN-Authenticator implementieren muss. Diese Aufgabe übernimmt der WLAN-Access-Point. Damit entfällt auch die nicht immer einfache Suche nach einem geeigneten USB-WLAN-Adapter.

**Trennung von  
LAN und WLAN**

In Firmennetzen werden das kabelgebundene Netz und das Funknetz zumeist getrennt. Zur tatsächlichen Realisierung gibt es verschiedene Möglichkeiten. Naheliegender wäre es, den Gateway-Rechner einfach mit *drei* Netzwerkadaptern auszustatten: einen für den Internet-Zugang, einen für das LAN und einen für das WLAN.

Bei der in [Abbildung 27.3](#) dargestellten Konfiguration sind die beiden lokalen Netze voneinander getrennt und verwenden unterschiedliche IP-Adressbereiche (192.168.0.\* und 172.16.0.\*). Sicherheitstechnisch ist das oft wünschenswert: Das WLAN-Netz ist für Gäste gedacht, die keinen Zugriff auf die Dateien des Firmen-Servers (Samba/NFS) oder auf andere firmeninterne Netzwerkdienste haben sollen.



**Abbildung 27.3** Alternative Topologie mit Trennung zwischen Firmen-LAN und Gäste-WLAN

Gleichzeitig ist diese Trennung natürlich auch unpraktisch: Per WLAN mit dem Netz verbundene Notebooks von Firmen-Mitarbeitern können nun keine Daten mit anderen Firmen-PCs oder mit dem Datei-Server der Firma austauschen.

Die übliche Lösung für solche Fälle besteht darin, auf dem Gateway-Rechner einen weiteren Dienst für ein Virtual Private Network (VPN) zu konfigurieren. Damit dringen wir aber in Regionen vor, die dieses Buch nicht mehr abdeckt.

## Voraussetzungen

**Schnittstellen-namen** Für den Rest des Kapitels beziehe ich mich auf die Topologie aus [Abbildung 27.1](#). Ich gehe davon aus, dass Ihr Rechner zwei Netzwerkschnittstellen hat:

- ▶ **Ethernet-Schnittstelle:** Über die Ethernet-Schnittstelle `enp4s0` wird die Verbindung zum Internet hergestellt, also z. B. zu einem vorhandenen ADSL-Router.
- ▶ **USB-WLAN-Adapter:** Der WLAN-Adapter mit dem Device-Namen `wlp0s29u1u7` soll die Internet-Verbindung an die WLAN-Clients weitergeben. (Dieser irrwitzige Device-Name entstand bei meinen Tests unter Fedora. Er beinhaltet auch die USB-Buchse, an die der WLAN-Adapter angesteckt ist. Wenn Sie den WLAN-Adapter später in eine andere Buchse stecken, müssen Sie Ihre Konfigurationsdateien entsprechend ändern! Ubuntu verhält sich in dieser Hinsicht anders: Dort kommt als Device-Name `wlx7cdd904ca740` zustande, was auf den ersten Blick auch nicht besser aussieht. In den Device-Namen fließt diesmal aber nur ein ID-Code des Adapters ein. Der Name ändert sich nicht mit der verwendeten USB-Buchse.)

Auf Ihrem eigenen Rechner werden natürlich andere Device-Namen gelten. Sie müssen also in den folgenden Listings jeweils `enp4s0` durch den Device-Namen Ihres Ethernet-Adapters ersetzen und `wlp0s29u1u7` durch den Namen Ihres WLAN-Adapters.

**IP-Adressen** Das Beispiel geht davon aus, dass der ADSL-Router über die Adresse `10.0.0.138` erreichbar ist. Für den Gateway-Rechner habe ich willkürlich die Adresse `10.0.0.17` gewählt. Sie müssen diese Adressen ebenso wie die Device-Namen jeweils durch eigene Werte ersetzen.

**Host- und Domainname** Für dieses Beispiel gehe ich davon aus, dass der Hostname des Gateway-Rechners `mars` lautet, seine Domain `sol`. Auch diese Zeichenketten müssen Sie in den Konfigurationsdateien entsprechend anpassen.

**WLAN-Adapter** Die meisten handelsüblichen WLAN-Adapter sind Linux-kompatibel, können also unter Linux ohne größere Probleme genutzt werden. Für dieses Kapitel reicht das aber nicht aus! Sie benötigen einen WLAN-Adapter, der vom Programm `hostapd` als Access Point angesprochen werden kann. Dieses Detail müssen Sie unbedingt vor dem Kauf recherchieren! Es gibt eine Menge WLAN-Adapter, die unter Linux grundsätzlich laufen, aber eben nur für die »gewöhnliche« Nutzung, nicht als Access Point.

Ich habe für dieses Kapitel mit dem LogiLink-Adapter WL0145 mit abnehmbarer Antenne und 150 Mbps gearbeitet (Ralink-RT5370-Chipset, erhältlich z. B. bei Pollin). `lsusb` liefert bei diesem Modell den folgenden ID-Code:

```
user$ lsusb
... ID 148f:5370 Ralink Technology, Corp. RT5370 Wireless Adapter
```



Das Kommando `iw list` muss bei den *supported interface modes* die Zeile `AP` ausgeben:

```
user$ iw list
...
Supported interface modes:
 * IBSS
 * managed
 * AP
 * AP/VLAN
...
```

Wenn Sie Ihr Gateway wie in [Abbildung 27.2](#) konfigurieren möchten, dann ersetzen Sie in den folgenden Anleitungen `wlp0s29u1u7` durch den Namen des Ethernet-Adapters, der mit dem Switch verbunden ist. Außerdem entfällt die gesamte Konfiguration des Programms `hostapd`.

Variante mit zwei  
Ethernet-  
Schnittstellen

Die für das Gateway erforderliche Netzwerkkonfiguration erfolgt manuell. Um eventuelle Kollisionen mit dem NetworkManager auszuschließen, empfiehlt es sich, dieses Programm zu deinstallieren (siehe [Abschnitt 26.6](#), »Distributionsspezifische Konfigurationsdateien«).

NetworkManager

Bei Fedora müssen Sie außerdem daran denken, das für die Auswertung der manuellen Konfigurationsdateien erforderliche Script `network` explizit zu aktivieren:

```
root# systemctl start network
root# systemctl enable network
```

## 27.2 Netzwerkkonfiguration

Die Netzwerkkonfiguration eines Internet-Gateways werden Sie zumeist manuell so durchführen, wie ich es in [Abschnitt 26.6](#), »Distributionsspezifische Konfigurationsdateien«, beschrieben habe. Davon abweichend hat ein Internet-Gateway *zwei* Netzwerkschnittstellen: Eine stellt die Verbindung zum Internet her, die andere die Verbindung zum lokalen Netzwerk.

Unter Debian oder Ubuntu muss die Konfigurationsdatei `/etc/network/interfaces` so ähnlich wie das folgende Beispiel aussehen:

Debian, Ubuntu

```
# /etc/network/interfaces
auto lo
iface lo inet loopback

# dynamische Verbindung zum ADSL-Router, der die Eckdaten des
# Netzwerkzugangs via DHCP vermittelt
auto enp4s0
iface enp4s0 inet dhcp
```

```
# statische Konfiguration für die Verbindung zum WLAN
auto wlp0s29u1u7
iface wlp0s29u1u7 inet static
    address 192.168.0.1
    netmask 255.255.255.0
```

Wenn die Verbindung in das Internet statisch konfiguriert wird, muss der enp4s0-Block in der interfaces-Datei wie folgt angepasst werden. Das Beispiel geht davon aus, dass der Internetzugang über einen ADSL-Router erfolgt, der die IP-Adresse 10.0.0.138 hat, so wie das SpeedTouch-Gerät, das bei mir zu Hause läuft. Diese Adresse ist gleichzeitig die Gateway-Adresse zum Internet (Schlüsselwort gateway).

```
# /etc/network/interfaces
...
# statische Verbindung zum ADSL-Router mit der IP-Adresse 10.0.0.138
auto enp4s0
iface enp4s0 inet static
    address 10.0.0.17
    netmask 255.255.255.0
    gateway 10.0.0.138
```

**Fedora, Red Hat** Bei Fedora bzw. Red Hat benötigen Sie je eine Konfigurationsdatei ifcfg-xxx im Verzeichnis /etc/sysconfig/network-scripts, wobei xxx jeweils der Name der Netzwerkschnittstelle ist. Die Datei für die Schnittstelle zum ADSL-Router kann z. B. so aussehen:

```
# Datei /etc/sysconfig/network-scripts/ifcfg-enp4s0
DEVICE=enp4s0
TYPE=Ethernet
BOOTPROTO=static
ONBOOT=yes
NM_CONTROLLED=no
# statische IP-Konfiguration
IPADDR=10.0.0.17
PREFIX=24
# ADSL-Router mit 10.0.0.138 ist das Gateway nach außen
GATEWAY=10.0.0.138
DNS1=10.0.0.138
# öffentlicher Nameserver von Google
DNS2=8.8.8.8
# Firewall: aktiviert Masquerading und Forwarding
ZONE=external
```

Die folgende Datei beschreibt die Schnittstelle zum lokalen Netzwerk (WLAN):

```
# Datei /etc/sysconfig/network-scripts/ifcfg-wlp0s29u1u7
DEVICE=wlp0s29u1u7
NM_CONTROLLED=no
```

```

ONBOOT=yes
BOOTPROTO=none
TYPE=Ethernet
# Lokales Netz: 192.168.0.*
IPADDR=192.168.0.1
PREFIX=24
# Firewall: die meisten Ports sperren
ZONE=internal

```

Unabhängig von der Distribution müssen Sie außerdem sicherstellen, dass `/etc/resolv.conf` die IP-Adresse des Nameservers enthält:

```

# /etc/resolv.conf
# Nameserver des ADSL-Routers
nameserver 10.0.0.138
# öffentlicher Nameserver von Google
nameserver 8.8.8.8

```

### Ubuntu-Eigenheiten

Unter Ubuntu ist `/etc/resolv.conf` ein Link, der vom Paket `resolvconf` verwaltet wird. Bevor Sie `resolv.conf` selbst konfigurieren können, müssen Sie dieses Paket deinstallieren und den Link `/etc/resolv.conf` löschen!

Die bisher durchgeführte Konfiguration sollten Sie testen: Starten Sie den Rechner neu. Anschließend überprüfen Sie mit `ip addr`, ob die Schnittstellen `enp4s0` und `wlp0s29u1u7` richtig konfiguriert sind und ob ein `ping` zu einem beliebigen Server im Internet funktioniert. Test

```

user$ ip addr
1: lo: ...
2: enp4s0: ...
   link/ether 14:da:e9:e1:5c:92 brd ff:ff:ff:ff:ff:ff
   inet 10.0.0.17/24 brd 10.0.0.255 scope global enp4s0
      valid_lft forever preferred_lft forever
   ...
3: wlp0s29u1u7: ...
   link/ether 7c:dd:90:4c:a7:40 brd ff:ff:ff:ff:ff:ff
   inet 192.168.0.1/24 brd 192.168.0.255 scope global wlp0s29u1u7
      valid_lft forever preferred_lft forever
   ...
user$ ping heise.de
PING heise.de (193.99.144.80) 56(84) bytes of data.
64 bytes from redirector.heise.de (193.99.144.80):
icmp_seq=1 ttl=247 time=42.7 ms
...

```

## 27.3 Masquerading (NAT)

Nachdem sichergestellt ist, dass der Gateway-Rechner selbst über die Schnittstelle `enp0s4` Internetzugang hat, geht es nun darum, diesen Internetzugang über die WLAN-Schnittstelle mit WLAN-Clients zu teilen. Ein Puzzlestein in der folgenden Konfiguration ist das Masquerading.

Das Prinzip des Masqueradings besteht darin, dass der Gateway-Rechner an das Internet adressierte Datenpakete der Clients annimmt und deren Absenderadresse so verändert, als würden sie vom ihm selbst stammen. Diese Adressänderung wird auch als *Network Address Translation* (NAT) bezeichnet.

Jetzt kann das Datenpaket in das Internet weitergeleitet werden. Im Regelfall kommt aus dem Internet nach einer Weile eine Antwort – beispielsweise die angeforderte Webseite. Das Gateway muss die Antwort an den richtigen Client weiterleiten. Dazu muss es die korrekte Zieladresse erraten. Das Datenpaket wurde ja (nach der Adressänderung) von ihm selbst abgesandt, daher ist auch die Antwort an das Gateway adressiert.

Um eine Adresszuordnung der Antwortpakete zu ermöglichen, verändert das Gateway nicht nur die Absenderadresse, sondern auch den Absender-Port. Für jede IP-Adresse innerhalb des lokalen Netzes wird eine bestimmte Port-Nummer verwendet. Linux-intern ist für das Masquerading `iptables` zuständig. Das ist ein in den Kernel integriertes System zur Verarbeitung von IP-Paketen.

Wenn der Gateway-Rechner über einen ADSL-Router mit dem Internet verbunden ist, führt dieses Gerät nochmals Masquerading oder eine andere Form der Adressmanipulation durch. Daraus ergeben sich glücklicherweise keine Probleme.

Masquerading- und Firewall-Funktionen sind eng miteinander verwandt. Wenn auf Ihrem Gateway-Rechner bereits eine Firewall läuft, wie dies standardmäßig bei einigen Distributionen der Fall ist (CentOS, Fedora, Red Hat, SUSE), dann müssen Sie die Masquerading-Funktionen mit dem Firewall-Konfigurationswerkzeug Ihrer Distribution einrichten. Läuft hingegen noch keine Firewall, dann reichen zur Aktivierung des Masqueradings zwei `iptables`-Kommandos, die ich Ihnen gleich erläutern werde. Hintergrundinformationen darüber, was Firewalls sind und wie sie funktionieren, finden Sie in [Kapitel 37](#), »Firewalls«.

**Forwarding** Die NAT-Firewall-Regel allein reicht nicht aus. Standardmäßig leitet Linux Netzwerkpakete nämlich nicht von einem Netzwerk zum nächsten weiter. Aus Sicherheitsgründen ist die sogenannte IP-Forwarding-Funktion deaktiviert. Zum Glück ist es nicht schwierig, diese Funktion im laufenden Betrieb zu aktivieren – es muss lediglich mit `sysctl` ein Kernelparameter verändert werden.

Um das Masquerading manuell einzuschalten, führen Sie zwei kurze Kommandos aus:

```
root# sysctl -w net.ipv4.ip_forward=1
root# iptables -A POSTROUTING -t nat -o enp4s0 -j MASQUERADE
```

Masquerading  
und Forwarding  
manuell  
aktivieren

Das `sysctl`-Kommando aktiviert die IP-Forwarding-Funktion des Kernels, die aus Sicherheitsgründen in der Standardeinstellung deaktiviert ist. Sollte `sysctl` bei Ihrer Distribution nicht zur Verfügung stehen, können Sie stattdessen das folgende Kommando ausführen:

```
root# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Das `iptables`-Kommando definiert eine Regel, gemäß der IP-Pakete, die das lokale Netzwerk verlassen sollen, über das Interface `enp4s0` geleitet und dabei entsprechend den NAT-Regeln manipuliert werden. (`enp4s0` ist in diesem Kapitel die Schnittstelle, über die der Gateway-Rechner mit dem Internet verbunden ist.)

Um die Masquerading-Funktionen zu deaktivieren, führen Sie folgende Kommandos aus:

```
root# iptables -t nat -D POSTROUTING -o enp4s0 -j MASQUERADE
root# sysctl -w net.ipv4.ip_forward=0
```

Masquerading  
und Forwarding  
deaktivieren

Statt `sysctl` funktioniert auch dieses `echo`-Kommando:

```
root# echo 0 > /proc/sys/net/ipv4/ip_forward
```

### Client/Server-Begriff

Im Folgenden wird der Gateway-Rechner auch als Server bezeichnet, alle anderen Rechner als Clients – ganz unabhängig davon, welche Funktionen diese Rechner sonst erfüllen. Im Fall von Masquerading wird der Server oft auch als *Internet-Gateway* (korrekt) oder als *Internet-Router* bezeichnet.

Diese Unterscheidung zwischen Client und Server gilt für das gesamte Buch – aber immer nur für eine bestimmte Funktion! Ein Rechner, der bezüglich seines Internetzugangs ein Client ist, kann durchaus für eine andere Funktion (etwa NFS) ein Server sein. In der Praxis werden sehr oft mehrere Server-Funktionen auf einem einzigen Rechner konzentriert.

### Masquerading und Forwarding dauerhaft aktivieren

Das manuelle Aktivieren der Masquerading- und Forwarding-Funktion mag für erste Tests angehen. Für den dauerhaften Betrieb eines Gateway-Rechners muss das Aktivieren aber automatisch erfolgen. Im Folgenden skizziere ich für einige Distributionen die beste Vorgehensweise.

**Debian, Ubuntu** Unter Debian und Ubuntu läuft standardmäßig keine Firewall. Um das Masquerading zu aktivieren, bauen Sie die beiden Zeilen in `/etc/network/interfaces` ein, die mit `up` bzw. `down` beginnen:

```
# Datei /etc/network/interfaces
...
iface wlp0s29u1u7 inet static
...
up /sbin/iptables -A POSTROUTING -t nat -o enp4s0 -j MASQUERADE
down /sbin/iptables -D POSTROUTING -t nat -o enp4s0 -j MASQUERADE
```

Die Schlüsselwörter `up` und `down` geben Kommandos an, die bei der Aktivierung bzw. bei der Deaktivierung der WLAN-Schnittstelle automatisch ausgeführt werden sollen.

Um außerdem das IP-Forwarding zu aktivieren, ändern Sie die Datei `/etc/sysctl.conf`. Sie enthält bereits die Zeile `net.ipv4.ip_forward=1`. Sie müssen lediglich das Kommentarzeichen `#` am Beginn dieser Zeile entfernen. (Diese Konfigurationsdatei gibt es nicht nur unter Debian und Ubuntu, sondern auch bei den meisten anderen Distributionen.)

```
# Änderung in /etc/sysctl.conf
net.ipv4.ip_forward=1
```

Die Änderung wird mit dem nächsten Neustart wirksam. Diesen Neustart ersparen Sie sich, wenn Sie das entsprechende `sysctl`-Kommando einmal manuell ausführen:

```
root# sysctl -w net.ipv4.ip_forward=1
```

**Fedora, CentOS 7 und RHEL 7** Bei Fedora sowie CentOS 7 und RHEL 7 ordnen Sie die Schnittstelle, über die der Rechner mit dem Internet verbunden ist, der Firewall-Zone `external` zu. Damit wird für diese Schnittstelle das Masquerading aktiviert. Gleichzeitig aktiviert Fedora auch das IP-Forwarding.

```
# Datei /etc/sysconfig/network-scripts/ifcfg-enp4s0 (Verbindung zum ADSL-Router)
...
ZONE=external
```

**SUSE** Bei SUSE ordnen Sie im YaST-Modul SICHERHEIT • FIREWALL die LAN-Schnittstelle dem internen Netzwerk zu. Anschließend aktivieren Sie im Dialogblatt MASQUERADING die gleichnamige Option.

**Aktivierung durch das Init-System** Distributionsunabhängig können Sie die Masquerading- und Forwarding-Funktionen auch über ein Init-V-Script oder eine `systemd`-Service-Datei aktivieren. Eine entsprechende Anleitung finden Sie in [Abschnitt 24.3](#), »Eigene Init-Scripts bzw. Init-Konfigurationsdateien«.

## Probleme

Das Masquerading ist zwar eine elegante Lösung, um im lokalen Netzwerk einen Internetzugang gemeinsam zu nutzen. Es kann aber auch Probleme geben:

- ▶ Bei einer ganzen Reihe von Internetprotokollen sind Schutzmechanismen vorgesehen, in denen die Zuordnung von IP-Adressen überprüft wird. Der durch das Masquerading nicht mehr eindeutige Zusammenhang zwischen IP-Adresse und einem Rechner kann Schwierigkeiten verursachen.
- ▶ Manche Protokolle sehen vor, dass IP-Adressen nicht nur in den IP-Paketen, sondern auch innerhalb der Datenpakete übertragen werden (als ASCII-Text oder auch verschlüsselt). Ein bekanntes Beispiel dafür ist FTP. Damit FTP trotz des Masqueradings funktioniert, muss der Masquerading-Server also nicht nur die Adressierung von IP-Paketen ändern, sondern in manchen Fällen auch deren Inhalt. Zu diesem Zweck sieht Linux für eine ganze Reihe von Internetdiensten entsprechende Masquerading-Module vor (z. B. `nf_nat_ftp` für FTP). Die Module werden automatisch geladen:

```
root# lsmod | grep nat
nf_nat_masquerade_ipv4 16384 1 ipt_MASQUERADE
ebtable_nat           16384 1
ebtables              32768 3 ebtable_broute,ebtable_nat,ebtable_filter
ip6table_nat          16384 1
nf_nat_ipv6           16384 1 ip6table_nat
ip6_tables            28672 5 ip6table_filter,ip6table_mangle,...
...
```

Wenn in FTP-Clients dennoch Verbindungsprobleme auftreten, lassen sich diese zumeist durch einen Betrieb im passiven Modus beheben. Die meisten FTP-Clients aktivieren diesen Modus automatisch oder bieten zumindest eine manuelle Einstellmöglichkeit.

## 27.4 Der WLAN-Authenticator hostapd

Die WLAN-Schnittstelle `wlp0s29u1u7` wäre jetzt prinzipiell funktionsbereit. Ein Detail fehlt aber noch: Notebooks, Smartphones und andere WLAN-Clients »sehen« diese Schnittstelle nicht und können keine Verbindung zu ihr herstellen.

Abhilfe schafft das Programm `hostapd` aus dem gleichnamigen Paket, das Sie installieren müssen. `hostapd` ist ein WLAN-Authenticator – also ein Programm, das sich darum kümmert, WLAN-Funktionen verschlüsselt anzubieten und die Authentifizierung durchzuführen. `hostapd` kann auf unterschiedlichste Weisen verwendet werden, unter anderem auch in Kombination mit dem zentralen Login-Server *Radius*.

### Router ohne hostapd

Wenn das lokale Netzwerk nicht über einen WLAN-Adapter des Computers verbunden ist, sondern über eine Ethernet-Schnittstelle plus Switch oder über einen externen WLAN-Access-Point, dann brauchen Sie keinen WLAN-Authenticator und können diesen Abschnitt überspringen.

**hostapd.conf** Unsere Anforderungen sind recht simpel: Wir wollen dem Funknetzwerk einen Namen geben – im Folgenden `mywlan` –, die Kommunikation durch den Standard WPA2 verschlüsseln und den WLAN-Zugang durch ein Passwort absichern. Dazu richten Sie die neue Konfigurationsdatei `/etc/hostapd/hostapd.conf` gemäß dem folgenden Muster ein. Beachten Sie, dass die Syntax dieser Datei keine Leerzeichen vor oder nach dem Zeichen `=` erlaubt:

```
# Datei /etc/hostapd/hostapd.conf
interface=wlp0s29u1u7
driver=nl80211
ssid=mywlan
wpa=2
wpa_passphrase=geheim1234
wpa_key_mgmt=WPA-PSK
hw_mode=g
channel=6
country_code=DE
# nur CentOS, Fedora, RHEL:
ctrl_interface=/var/run/hostapd
ctrl_interface_group=wheel
```

Kurz eine Erklärung der Schlüsselwörter: `interface` gibt an, welche WLAN-Schnittstelle `hostapd` verwalten soll. `driver` bestimmt, welcher Linux-WLAN-Treiber zum Einsatz kommen soll. `nl80211` ist der zurzeit modernste Treiber im Linux-Kernel; er ist für die meisten WLAN-Adapter geeignet, die derzeit auf dem Markt sind. `hostapd` kommt aber auch mit einigen älteren WLAN-Treibern zurecht: `hostap`, `madwifi` und `prism54`. Sollte also `driver=nl80211` bei Ihrem WLAN-Adapter zu einer Fehlermeldung beim Start führen, müssen Sie im Internet recherchieren, welcher Treiber zu Ihrem WLAN-Adapter passt.

### Kompatibilitätsprobleme

Leider ist nicht jeder WLAN-Adapter zu `hostapd` kompatibel. Auf dieses Problem habe ich ja schon in der Einleitung des Kapitels hingewiesen. Recherchieren Sie vor dem Kauf des WLAN-Adapters im Internet, ob andere Linux-Benutzer diesen Adapter mit `hostapd` nutzen konnten!



ssid steht für *Service Set Identifier* und bezeichnet den Namen des WLAN-Netzwerks. Das ist also der Name, den Sie in einem Notebook oder Smartphone sehen, wenn Sie das WLAN-Netz auswählen.

wpa=2 bewirkt, dass die Verschlüsselungstechnik WPA2 verwendet wird. wpa\_key\_mgmt gibt an, wie die Authentifizierung erfolgt. Zur Auswahl stehen die Verfahren *Pre-Shared Key* (WPA\_PSK) oder *Extensible Authentication Protocol* (WPA\_EAP). wpa\_passphrase ist das Passwort für den WLAN-Zugang. Es muss zumindest acht Zeichen umfassen.

hw\_mode gibt an, welcher WLAN-Standard verwendet werden soll, z.B. g für IEEE 802.11g. channel bestimmt, welchen Kanal bzw. welches Frequenzband der WLAN-Adapter verwenden soll. Die country\_code-Einstellung stellt sicher, dass die regulatorischen Funkbestimmungen eingehalten werden. Weitere Informationen zur hostapd-Konfiguration können Sie auf diesen Webseiten nachlesen:

<https://w1.fi/hostapd>  
<https://w1.fi/cgiit/hostap/plain/hostapd/hostapd.conf>  
<https://wireless.wiki.kernel.org/en/users/Documentation/hostapd>  
[https://en.wikipedia.org/wiki/List\\_of\\_WLAN\\_channels](https://en.wikipedia.org/wiki/List_of_WLAN_channels)

Da die hostapd-Konfigurationsdatei das WPA-Passwort im Klartext enthält, sollte die Datei nur für root lesbar sein:

```
root# chmod 600 /etc/hostapd/hostapd.conf
```

Damit hostapd die Konfigurationsdatei überhaupt berücksichtigt, muss unter Debian, Raspbian und Ubuntu auch die Datei /etc/default/hostapd verändert werden. Dort müssen Sie beim Parameter DAEMON\_CONF den Namen der Konfigurationsdatei eintragen:

```
# Datei /etc/default/hostapd (nur Debian, Raspbian und Ubuntu)
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Erfahrungsgemäß ist die hostapd-Konfiguration fehleranfällig. Deswegen empfiehlt es sich, den Dienst vorerst manuell zu starten. Sämtliche Statusmeldungen sind dann direkt im Terminalfenster zu sehen. Nun können Sie mit einem Smartphone, Tablet oder Notebook ausprobieren, ob das WLAN sichtbar ist und ob ein Login möglich ist. (Als Router funktioniert Ihr Gateway noch nicht, es geht hier nur um die Authentifizierung.)

**Manueller Start**

```
root# hostapd /etc/hostapd/hostapd.conf
... (unzählige Statusmeldungen)
wlp0s29u1u7: interface state COUNTRY_UPDATE->ENABLED
wlp0s29u1u7: AP-ENABLED
...
wlp0s29u1u7: STA 90:b9:31:cd:cf:62 WPA: pairwise key handshake completed (RSN)
```

**Start durch das Init-System** Wenn `hostapd` keine offensichtlichen Fehler anzeigt und weiterläuft, dann beenden Sie das Programm mit `[Strg]+[C]` wieder. Anschließend starten Sie den Dämon regulär als Systemdienst:

```
root# systemctl start hostapd
```

Unter Debian und Ubuntu ist nichts weiter zu tun. Unter Fedora, CentOS und RHEL müssen Sie sicherstellen, dass `hostapd` auch in Zukunft gestartet wird:

```
root# systemctl enable hostapd
```

### Erste Tests

Ihr eigener WLAN-Router ist zwar erst halb fertig, aber Sie können ihn schon ausprobieren. Dazu stellen Sie mit einem WLAN-Gerät, z.B. mit einem Tablet, eine Verbindung zum neuen WLAN `mywlan` her und geben dabei das in `hostapd.conf` enthaltene Passwort an. Der Verbindungsaufbau sollte klappen, aber die automatische IP-Konfiguration scheitert, weil Sie auf dem Gateway-Rechner noch keinen DHCP-Server installiert haben. Deswegen müssen Sie die IP-Parameter manuell eingeben:

```
IP-Adresse:          192.168.0.2
Netzmaske:           255.255.255.0
Gateway (Router):    192.168.0.1
Nameserver (DNS):    8.8.8.8
```

Damit bezieht das Gerät den Internetzugang über Ihren erst teilweise konfigurierten Gateway-Rechner. Als Nameserver kommt ein öffentlicher Nameserver von Google zum Einsatz.

## 27.5 DHCP- und Nameserver-Grundlagen

**DHCP** Natürlich können Sie in einem lokalen Netzwerk bei jedem Gerät die Netzwerkparameter separat einstellen. Das ist aber ebenso mühsam wie fehleranfällig. Außerdem handeln Sie sich jede Menge Zusatzarbeit ein, wenn Sie sich irgendwann dazu entschließen, die Topologie Ihres Netzwerks zu ändern.

Wesentlich intelligenter ist es, wenn *ein* Rechner sich darum kümmert, allen anderen Rechnern ihre IP-Adresse und andere Netzwerkparameter zuzuweisen. Dazu wird das *Dynamic Host Configuration Protocol* (DHCP) eingesetzt. Der Steuerungsrechner wird DHCP-Server genannt, die anderen Rechner DHCP-Clients. Es gibt zwei grundsätzliche Konfigurationsvarianten:

- ▶ **Dynamische Konfiguration:** Bei den Clients wird nur der Hostname eingestellt. Der DHCP-Server ist für alle anderen Konfigurationsparameter zuständig, weist den Clients also die IP-Adresse, die Gateway-Adresse, die Nameserver-Adresse etc. zu. Für die IP-Adressen gibt es einen Adresspool, aus dem der DHCP-Server für jeden Client dynamisch eine gerade freie Adresse wählt. Diese Konfigurationsvariante ist der Normalfall – nicht umsonst steht der erste Buchstabe von DHCP für *Dynamic*.
- ▶ **Statische Konfiguration:** Bei dieser Konfigurationsvariante identifiziert der DHCP-Server die Clients anhand der ID-Nummer der Netzwerkkarte. Damit kann er ihnen jedes Mal dieselbe IP-Adresse und optional auch den Hostnamen zuweisen. Diese Konfigurationsvariante ist mit etwas mehr Aufwand verbunden, ermöglicht dafür aber immer gleich bleibende IP-Adressen sowie eine zentrale Verwaltung der Hostnamen.

Einfacher ist in der Regel die erste Variante. Sie können beide Verfahren auch kombinieren, beispielsweise um sicherzustellen, dass ein Drucker immer dieselbe fixe IP-Adresse hat.

Die Funktionsweise von DHCP sieht in etwa so aus: Wenn ein Rechner (also ein DHCP-Client) neu gestartet wird, schickt er eine Rundsendung an die Adresse 255.255.255.255. Durch diese Adressierung erreicht die Anfrage *alle* Rechner im lokalen Netz. Der DHCP-Server reagiert auf diese Anfrage und sendet als Antwort eine IP-Adresse aus der Liste der verfügbaren IP-Adressen.

DHCP-Interna

Vielleicht fragen Sie sich, wohin der Server die Antwort sendet, denn der Client hat ja noch gar keine IP-Adresse. Zur Adressierung reicht die MAC-Adresse aus – und die ist aus der Anfrage bereits bekannt.

Der DHCP-Server vergibt IP-Adressen für eine bestimmte Zeitspanne (Lease Time). Diese Zeitspanne beträgt normalerweise einen Tag, kann aber beliebig eingestellt werden. Bevor diese Zeitspanne vergeht, muss der Client die Adresse beim DHCP-Server erneuern oder eine neue Adresse anfordern.

Ein *Domain-Nameserver* (kurz Nameserver, noch kürzer DNS) stellt den Zusammenhang zwischen Rechnernamen und IP-Adressen her. Jeder Internet Service Provider (ISP) stellt einen DNS zur Verfügung, der zu Rechnernamen die passende IP-Nummer ermittelt. Statt auf diesen DNS zurückzugreifen, können Sie für Ihr lokales Netzwerk einen eigenen Nameserver einrichten. Das hat zwei Vorteile:

Nameserver

- ▶ **Höhere Geschwindigkeit:** Der DNS verwaltet einen Cache der zuletzt benutzten Internetadressen. Wenn Sie also zum zweiten Mal an einem Tag zu `www.yahoo.com` surfen, muss nicht wieder der DNS Ihres Internet-Providers gefragt werden, welches nun die IP-Adresse von Yahoo ist. Der lokale DNS hat sich die Adresse schon gemerkt.

- ▶ **Lokale Namensauflösung:** Der DNS verwaltet die Namen und IP-Adressen der Rechner des lokalen Netzes. Damit kennen sich alle Rechner im lokalen Netzwerk namentlich, und Sie können beispielsweise am Rechner `merkur` das Kommando `ping saturn` ausführen. `merkur` kontaktiert nun den lokalen Nameserver, der die IP-Adresse von `saturn` zurückgibt.

Die lokale Namensauflösung ist eine Grundvoraussetzung dafür, dass im lokalen Netzwerk Dienste wie NFS, FTP, SSH etc. komfortabel genutzt und konfiguriert werden können.

Weltweit sind unzählige DNS im Einsatz, die alle gegenseitig in Kontakt stehen. Wenn also ein DNS einen Namen nicht selbst kennt, gibt er die Anfrage an einen anderen DNS weiter. Die DNS sind hierarchisch organisiert.

## 27.6 Dnsmasq (DHCP- und Nameserver)

Dieser Abschnitt beschreibt das Programm `Dnsmasq`, das einen DHCP-Server und einen Nameserver in sich integriert und relativ einfach zu konfigurieren ist. `Dnsmasq` ist selbst für große lokale Netze ausreichend leistungsfähig.

Die populärste Alternative zu `Dnsmasq` sind die Programme `dhcpcd` und `bind`, die jeweils eine dieser Aufgaben übernehmen. Insbesondere `bind` ist *das* dominierende Nameserver-Programm und damit ein zentraler Bestandteil der Infrastruktur des Internets. Es wird vor allem in den Knotenpunkten des Internets sowie von großen Internet-Providern und -Hostern eingesetzt. Als Nameserver für lokale Netze ist das Programm wegen seiner sehr unübersichtlichen Konfiguration aber weniger gut geeignet.

**Voraussetzungen** Ich gehe im Folgenden davon aus, dass Sie das Paket `dnsmasq` installiert haben und dass `dhcpcd` und `bind` *nicht* installiert sind – andernfalls kommen sich die Programme in die Quere.

Unter Ubuntu sollten Sie außerdem das Paket `resolvconf` deinstallieren. Es verursacht sonst Probleme, wenn Sie `Dnsmasq` konfigurieren. Außerdem müssen Sie in `/etc/default/dnsmasq` die Variable `DNSMASQ_OPTS` so einstellen, dass das Programm die Konfigurationsdatei `/etc/dnsmasq.conf` berücksichtigt:

```
# Datei /etc/default/dnsmasq (Ubuntu)
DNSMASQ_OPTS="--conf-file=/etc/dnsmasq.conf"
```

Eine weitere wichtige Voraussetzung ist die korrekte Konfiguration der Datei `/etc/hosts` auf dem Gateway-Rechner. Ich habe die Erfahrung gemacht, dass diese Datei oft nicht den Anforderungen von `Dnsmasq` entspricht. Werfen Sie einen Blick auf diese Datei, wenn Sie die lokale Netzwerkkonfiguration Ihres Gateways ändern!

/etc/hosts muss zumindest die beiden folgenden Zeilen enthalten. Entscheidend ist die Zeile für die Zuordnung zwischen dem lokalen Rechnernamen (hier mars bzw. mars.sol) und der IP-Adresse im LAN (hier 192.168.0.1).

```
# /etc/hosts auf dem Gateway-Rechner
127.0.0.1    localhost
192.168.0.1  mars        mars.sol
...
```

DNS-Anfragen werden über die TCP- und UDP-Ports 53 abgewickelt. DHCP benutzt die UDP-Ports 67 und 68. Diese Ports dürfen auf dem Gateway-Rechner nicht durch eine Firewall blockiert werden! Firewall

Bei der Netzwerkkonfiguration für Fedora bzw. CentOS 7 oder RHEL 7 haben wir die Schnittstelle wlp0s29u1u7 der Zone internal zugeordnet. In dieser Zone sind die Ports für den Nameserver und für DHCP blockiert. Also müssen wir Ausnahmeregeln einrichten:

```
root# firewall-cmd --permanent --zone=internal --add-service=dhcp
root# firewall-cmd --permanent --zone=internal --add-service=dns
root# firewall-cmd --reload
```

Die Konfiguration von Dnsmasq erfolgt durch die Datei /etc/dnsmasq.conf. Die standardmäßig mitgelieferte Datei dient gleichzeitig als Dokumentation und besteht aus rund 600 Kommentarzeilen. Sie können bei einzelnen schon vorgesehenen Anweisungen das Kommentarzeichen entfernen und die betreffende Zeile aktivieren. Übersichtlicher ist es aber zumeist, eine Sicherheitskopie von dnsmasq.conf zu erstellen und mit einer neuen, leeren Konfigurationsdatei zu beginnen: dnsmasq.conf

```
root# mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
root# touch /etc/dnsmasq.conf
```

Für die Konfiguration von Dnsmasq sind neben /etc/dnsmasq.conf auch /etc/hosts und /etc/resolv.conf entscheidend. Bei vielen Distributionen werden außerdem alle in /etc/dnsmasq.d enthaltenen Konfigurationsdateien berücksichtigt.

Bei Debian, Raspbian und Ubuntu enthält die Datei /etc/default/dnsmasq einige Grundeinstellungen für Dnsmasq, unter anderem ENABLED=1, sodass Dnsmasq standardmäßig durch das Init-System gestartet wird.

Wie die meisten Programme, die in diesem und den folgenden Kapiteln vorgestellt werden, arbeitet Dnsmasq als sogenannter Dämon (Systemdienst). Bei einigen Distributionen wird das Programm sofort nach der Installation gestartet. Wenn das nicht der Fall ist, müssen Sie das Programm manuell starten. Auch Konfigurationsänderungen werden erst nach einem Neustart wirksam: Start/Neustart

```
root# systemctl restart dnsmasq
```

Ebenfalls von der Distribution abhängig ist, ob das Programm in Zukunft beim Hochfahren des Rechners automatisch gestartet wird. Bei Fedora, Red Hat und SUSE ist das standardmäßig nicht der Fall. Abhilfe schaffen diese Kommandos:

```
root# systemctl enable dnsmasq.service (systemd-Distributionen)
```

#### Minimal- konfiguration

Bereits in der folgenden Minimalkonfiguration funktioniert `dnsmasq.conf` zufriedenstellend. Das Programm arbeitet in dieser Konfiguration als Nameserver-Cache für das Internet und stellt den Clients IP-Adressen aus dem Bereich zwischen 192.168.0.2 und 192.168.0.250 zur Verfügung. Die Clients behalten ihren eigenen Hostnamen.

```
# /etc/dnsmasq.conf (Minimalkonfiguration)
domain-needed
bogus-priv
interface=wlp0s29u1u7
dhcp-range=192.168.0.2,192.168.0.250,24h
```

Kurz eine Erläuterung der Schlüsselwörter: `domain-needed` und `bogus-priv` verhindern, dass Dnsmasq lokale Hostnamen bzw. lokale IP-Adressen an den Nameserver Ihres Internet-Providers (ISP) weitergibt. Der Nameserver des ISP ist nur für Internetnamen/-adressen zuständig, nicht für lokale Namen/Adressen.

`interface` gibt an, dass Dnsmasq in seiner Funktion als DHCP-Server nur auf Anfragen antworten soll, die von der Schnittstelle `wlp0s29u1u7` kommen, die in der Beispieltopologie für das LAN zuständig ist. Wenn Dnsmasq auf die Anfragen von mehreren Schnittstellen reagieren soll, bauen Sie für jede Schnittstelle eine eigene Anweisung `interface=xxx` in die Konfigurationsdatei ein. Alternativ besteht mit `except-interfaces=...` die Möglichkeit, die Schnittstellen anzugeben, die Dnsmasq *nicht* beachten soll.

`dhcp-range` gibt an, welchen Adressbereich der DHCP-Server zur Beantwortung von DHCP-Anfragen nutzen soll. Vergebene Adressen bleiben 24 Stunden lang gültig und müssen dann vom Client erneuert werden.

Nicht extra konfiguriert werden müssen die Nameserver- und Gateway-Adressen. Dnsmasq wertet selbstständig `/etc/resolv.conf` aus und greift auf den dort angegebenen Nameserver zurück. An DHCP-Clients wird als Nameserver- und Gateway-Adresse jeweils die lokale IP-Adresse übertragen.

#### Einsatz als lokaler Nameserver

In der obigen Minimalkonfiguration kann Dnsmasq lokale Adressen nur auflösen, wenn `/etc/hosts` entsprechende Informationen enthält. Dynamisch per DHCP zugewiesene Adressen kennt der Nameserver dagegen nicht. Damit Dnsmasq auch als Nameserver für die Clients im LAN funktioniert, fügen Sie die folgenden Zeilen zu `dnsmasq.conf` hinzu und weisen Dnsmasq an, die Konfigurationsdatei neu einzulesen. `sol` ist dabei der Domainname des Beispielnetzes.

```
# /etc/dnsmasq.conf (Einsatz als Nameserver für lokale Adressen)
...
local=/sol/
domain=sol
expand-hosts
```

Das Schlüsselwort `local` gibt an, dass Adressanfragen aus dieser Domain direkt von Dnsmasq beantwortet werden sollen (nicht vom Nameserver des ISP).

`domain` gibt an, dass Dnsmasq den DHCP-Clients den angegebenen Domainnamen zuweisen soll. Dieser Name muss mit dem in `local` angegebenen Namen übereinstimmen.

`expand-hosts` bewirkt schließlich, dass bei Nameserver-Anfragen ohne Domain automatisch die in `domain` angegebene Domain hinzugefügt wird. Wenn Sie also `ping uranus` ausführen, liefert Dnsmasq die Adresse von `uranus.sol` zurück.

Dnsmasq kennt seine Clients – also die Rechner, die eine IP-Konfiguration via DHCP angefordert haben – nur dann namentlich, wenn diese im Rahmen der DHCP-Kommunikation ihren eigenen Hostnamen an Dnsmasq übermittelt haben. Das ist bei den meisten Distributionen bzw. LAN/WLAN-Konfigurationswerkzeugen standardmäßig der Fall, auch beim NetworkManager.

Dnsmasq kann auch so konfiguriert werden, dass es die Hostnamen der Clients einstellt. Die statische Zuordnung des Hostnamens und der IP-Adresse erfolgt auf Basis der MAC-Adresse der Clients. Das ist vor allem für solche Geräte praktisch, bei denen sich nicht ohne Weiteres ein Hostname einstellen lässt – beispielsweise bei Netzwerkdruckern. Die Konfiguration erfolgt mit dem Schlüsselwort `dhcp-host`. Das folgende Listing zeigt den Eintrag für den Netzwerkdrucker `pluto`:

Statische  
Adressen und  
Hostnamen

```
# /etc/dnsmasq.conf (statische Adresszuordnung)
...
dhcp-host=00:c0:ee:51:39:9f,pluto,192.168.0.254
```

Die mühsamste Aufgabe bei dieser Konfigurationsvariante ist es naturgemäß, die MAC-Adresse (*Media Access Control*) des Clients herauszufinden. Dabei handelt es sich um eine eindeutige ID-Nummer, mit der jeder Ethernet-Controller ausgestattet ist. Unter Linux zeigt `ifconfig` die MAC-Adresse an. Ansonsten verbinden Sie das Gerät einfach mit dem LAN und lassen Dnsmasq eine dynamische DHCP-Konfiguration durchführen. Das Programm protokolliert alle dynamisch zugewiesenen IP-Adressen samt Hostname und MAC-Adresse in der Datei `/var/lib/misc/dnsmasq.leases`. Sie können die MAC-Adresse also dieser Datei entnehmen.

Standardmäßig sind via DHCP zugewiesene IP-Adressen nur 24 Stunden lang gültig. Bei Netzwerkdruckern oder anderen Geräten mit automatischem Stand-by-Modus ist

das oft zu kurz. Wenn ein Drucker mehr als 24 Stunden nicht genutzt wird, glaubt Dnsmasq, das Gerät sei ausgeschaltet, und »vergisst« es gewissermaßen. Um das zu verhindern, können Sie die Gültigkeit der Adresse durch eine zusätzliche Zeitangabe verlängern. `infinite` bewirkt, dass die Adresse nie verfällt.

```
# /etc/dnsmasq.conf (statische Adresszuordnung ohne Ablaufzeit)
...
dhcp-host=00:c0:ee:51:39:9f,pluto,192.168.0.254,infinite
```

`man dnsmasq` bzw. `/etc/dnsmasq.conf.orig` beschreibt eine Menge weiterer Syntaxvarianten für `dhcp-host`. Sie können damit beispielsweise eine Zuordnung zwischen dem Hostnamen und der IP-Adresse herstellen, bestimmte MAC-Adressen komplett blockieren etc.

#### DNS für den lokalen Rechner

Standardmäßig funktioniert Dnsmasq zwar für alle anderen Rechner im Netzwerk als Nameserver, nicht aber für den Gateway-Rechner selbst! Der Grund besteht darin, dass auf dem lokalen Rechner der in `/etc/resolv.conf` angegebene Nameserver verwendet wird. Diese Datei verweist in der Regel auf den Nameserver Ihres Internet-Providers oder Routers.

Falls auf dem Gateway weitere Server-Programme laufen sollen (ein Datei-Server, Kerberos etc.), ist es erforderlich, dass das Gateway die Clients namentlich kennt, also ebenfalls Dnsmasq als Nameserver verwendet. Damit das funktioniert, müssen Sie auf die folgenden Punkte achten:

- ▶ `/etc/resolv.conf` muss auf Dnsmasq (also auf `localhost`, Adresse `127.0.0.1`) verweisen, nicht auf einen externen Nameserver. Vorsicht: Wenn Sie die Verbindung zum Internet dynamisch konfigurieren (LAN plus DHCP oder über ein Modem plus PPP), wird `resolv.conf` bei jedem Verbindungsaufbau überschrieben. Das müssen Sie verhindern: Konfigurieren Sie die Verbindung zum ADSL-Router statisch, bzw. verändern Sie die PPP-Konfiguration so, dass `resolv.conf` nicht angeührt wird.
- ▶ Dnsmasq kann nun nicht mehr `resolv.conf` auswerten, um die Adresse des externen Nameservers zu ermitteln. Deswegen müssen Sie dessen Adresse in `dnsmasq.conf` explizit mit dem Schlüsselwort `server` angeben.

Wie so oft macht ein Beispiel alles klarer. Nehmen wir an, der Gateway-Rechner `mars` mit der IP-Adresse `192.168.0.1` im LAN ist über die Ethernet-Schnittstelle `enp4s0` mit einem ADSL-Router verbunden. Der Router hat die IP-Adresse `10.0.0.138`. Damit `mars` mit dem ADSL-Router kommunizieren kann, wird die Schnittstelle `enp4s0` statisch konfiguriert. (Auch wenn der ADSL-Router DHCP unterstützt, ist eine dynamische Konfiguration via DHCP nicht zweckmäßig, weil sonst `resolv.conf` bei jedem Rechnerstart mit den DHCP-Daten des ADSL-Routers überschrieben wird!)



```
# in /etc/network/interfaces
...
# statische Verbindung zum ADSL-Router bzw. in das Internet
auto enp4s0
iface enp4s0 inet static
    address 10.0.0.1
    netmask 255.255.255.0
    gateway 10.0.0.138
...
```

Die Datei `/etc/resolv.conf` gibt den Namen der lokalen Domäne (`sol`) und die IP-Adresse des lokalen Nameservers an (also Dnsmasq):

```
# /etc/resolv.conf
search sol
nameserver 192.168.0.1
```

Bleibt noch die Konfiguration von Dnsmasq: Dnsmasq kann die Adresse des externen Nameservers (für Internetzugriffe) nun nicht mehr als `resolv.conf` lesen. Vielmehr soll das Programm `resolv.conf` nun ignorieren (Option `no-resolv`) und als externen Nameserver die mit `server` angegebene Adresse kontaktieren:

```
# /etc/dnsmasq.conf
...
no-resolv
server=10.0.0.138
...
```

Falls Sie während dieser stückweisen Präsentation diverser Dnsmasq-Einstellungen den Überblick verloren haben, finden Sie hier die Zusammenfassung der endgültigen `dnsmasq.conf`-Datei:

**Alles zusammen**

```
# /etc/dnsmasq.conf

# Schnittstelle zum LAN
interface=wlp0s29u1u7

# lokale Hosts nicht dem Upstream-Nameserver melden
domain-needed
bogus-priv

# Domainname sol im LAN
local=/sol/
domain=sol
expand-hosts

# Dnsmasq auch für das Gateway (Upstream-Nameserver = 10.0.0.138)
server=10.0.0.138
no-resolv
```

```
# dynamische Adressen
dhcp-range=192.168.0.2,192.168.0.250,24h

# statische Adressen
dhcp-host=00:c0:ee:51:39:9f,pluto,192.168.0.254,infinite
```

#### Konfiguration für mehrere Schnittstellen

Mitunter kommt es vor, dass Dnsmasq mehrere Netzwerkschnittstellen bedienen soll und diese mit IP-Adressen aus unterschiedlichen Netzwerksegmenten versorgen soll. Das wäre z. B. der Fall, wenn Sie eine Konfiguration wie in [Abbildung 27.3](#) wünschen. Dann benötigen Sie mehrere `interface`-Anweisungen. Außerdem ist für jede Schnittstelle eine eigene `dhcp-range`-Anweisung erforderlich:

```
# /etc/dnsmasq.conf
interface=enp5s0
interface=enp6s0
dhcp-range=192.168.0.2,192.168.0.250,24h
dhcp-range=172.16.0.2,172.16.0.250,12h
...
```

Das obige Listing besagt, dass sich Dnsmasq um die Schnittstellen `enp5s0` und `enp6s0` kümmern soll. Diesen Schnittstellen müssen im Rahmen der Netzwerkkonfiguration statische IP-Adressen zugeordnet werden, z. B. `enp5s0 = 192.168.0.1` und `enp6s0 = 172.16.0.1`. Dnsmasq verwendet für jede Schnittstelle nur solche IP-Adressen, die im richtigen Netzwerksegment liegen. DHCP-Anfragen, die über die Schnittstelle `enp6s0` eintreffen, beantwortet Dnsmasq daher mit einer IP-Adresse aus dem Bereich zwischen `172.16.0.2` und `172.16.0.250`.

Nur in Ausnahmefällen ist es notwendig, die Schnittstelle in der `dhcp-range`-Anweisung explizit anzugeben – beispielsweise dann, wenn beide Schnittstellen das gleiche Netzwerksegment verwenden. Das folgende Beispiel verdeutlicht dabei nur die Syntax:

```
# /etc/dnsmasq.conf
interface=enp5s0
interface=enp6s0
dhcp-range=interface:enp5s0,192.168.0.2,192.168.0.128,24h
dhcp-range=interface:enp6s0,192.168.0.129,192.168.0.250,12h
...
```

**Logging** Dnsmasq trägt automatisch alle dynamisch zugewiesenen IP-Adressen in die Datei `/var/lib/misc/dnsmasq.leases` oder `/var/lib/dnsmasq/dnsmasq.leases` ein. Statische Adressen werden dabei nicht berücksichtigt. Jeder Eintrag in dieser Datei enthält auch die MAC-Adresse und (soweit bekannt) den Hostnamen des Clients. Die Datei bietet somit eine gute Möglichkeit, um die MAC-Adresse neuer Clients herauszufinden.

Wenn Dnsmasq nicht wunschgemäß funktioniert, fügen Sie das Schlüsselwort `log-queries` in `dnsmasq.conf` ein. Das Programm protokolliert nun sämtliche Nameserver-Anfragen in `/var/log/syslog` oder `/var/log/messages`.

### Client-Konfiguration

Die richtige Konfiguration eines Rechners, der seine IP-Konfiguration via DHCP beziehen soll, ist grundsätzlich ganz einfach. Jede Linux-Distribution und alle aktuellen Windows-Versionen bieten eine entsprechende Option im Netzwerkkonfigurationsdialog an – im Prinzip war's das schon!

Zum Ausprobieren des DHCP-Servers müssen Sie die Clients natürlich nicht jedes Mal neu starten. Ein Neustart der Netzwerkfunktionen reicht vollkommen aus. Wenn Sie den NetworkManager verwenden, wählen Sie in dessen Menü einfach die betreffende WLAN-Verbindung nochmals aus. Das Programm unterbricht dann die vorhandene Verbindung und stellt sie neu her.

Anschließend vergewissern Sie sich mit `ip addr`, `ip route` sowie durch einen Blick in die Datei `/etc/resolv.conf`, ob alles funktioniert hat. Die folgenden Ausgaben sind auf einem Notebook entstanden, das mit dem NetworkManager via WLAN eine Verbindung zum Gateway-Rechner `mars` hergestellt hat:

```
user$ ip addr show wlp2s0
3: wlp2s0: ...
    link/ether 60:d8:19:d2:85:89 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.8/24 brd 192.168.0.255 scope global dynamic wlp2s0
user$ ip route
default via 192.168.0.1 dev wlp2s0 proto static metric 600
192.168.0.0/24 dev wlp2s0 proto kernel scope link src 192.168.0.8 ...
user$ cat /etc/resolv.conf
# Generated by NetworkManager
search sol lan
nameserver 192.168.0.1
```



# Kapitel 28

## Samba

Mit Samba stellen Sie Dateien bzw. Verzeichnisse ins lokale Netzwerk. Anwender mit Windows-, Linux- und Apple-Rechnern können darauf zugreifen. Ein fein differenziertes Authentifizierungs- und Rechtesystem steuert, wer welche Dateien lesen bzw. verändern darf. Samba ist somit der zentrale Knotenpunkt zum Datenaustausch in einem lokalen Netzwerk – sei es in einer Firma, einer Organisation oder zu Hause.

Der Name Samba ist von der Abkürzung SMB abgeleitet, die wiederum für das Protokoll *Server Message Block* steht. Die ersten SMB-Konzepte stammen von IBM, später wurde das Protokoll von mehreren Firmen erweitert, vor allem von Microsoft.

Samba wird oft als Verknüpfungselement zwischen der Linux- und der Windows-Welt betrachtet. Das greift aber zu kurz: SMB kommt selbst in reinen Linux-Umgebungen häufig zum Einsatz, weil es derart einfach zu nutzen ist: Die Dateimanager von Gnome und KDE unterstützen SMB standardmäßig, SMB-Verzeichnisse können dank CIFS direkt in den Linux-Verzeichnisbaum eingebunden werden etc.

Wenn Sie nach einer Unix-typischen Alternative ohne Microsoft-Hintergrund suchen, bietet sich am ehesten NFS an (siehe [Kapitel 29](#), »NFS und AFP«). Der Einsatz von NFS erfordert aber einheitliche Login-Namen auf allen Client-Rechnern oder den Einsatz von LDAP und ist zudem inkompatibel zur Windows-Welt.

Dieses Kapitel beschreibt die Grundfunktionen von Samba 4 aus Server-Sicht. Wenn Sie die zahlreichen fortgeschrittenen Funktionen nutzen möchten, beispielsweise die Authentifizierung via LDAP oder die Verwendung von Samba als PDC, müssen Sie auf weiterführende Literatur zurückgreifen. Empfehlenswert ist das im Rheinwerk Verlag erschienene Buch *Samba 4: Das Praxishandbuch für Administratoren* von Stefan Kania. Eine Menge Informationen zu Samba finden Sie auch auf diesen Websites:

Links

<https://www.samba.org>

<https://help.ubuntu.com/community/Samba>

## 28.1 Grundlagen und Glossar

Bevor Sie mit der Konfiguration eines Samba-Servers beginnen, sollten Sie die zugrunde liegenden Konzepte und das dazugehörige Vokabular verstehen. Dieser Abschnitt soll Ihnen dabei helfen.

**NetBIOS** SMB basiert in seiner ursprünglichen Form auf dem NetBIOS-Protokoll. NetBIOS steht für das ursprünglich von IBM entwickelte *Network Basic Input/Output System*. Mittlerweile bezieht sich NetBIOS allerdings auf ein mehrfach renoviertes Protokoll. NetBIOS besteht primär aus drei Diensten:

- ▶ **Name Service:** Dieses Verfahren zum Austausch der Rechnernamen ist mit DNS unter Unix/Linux vergleichbar. Die Verwaltung der Namen kann wahlweise zentral durch einen *NetBIOS-Nameserver* (NBNS) oder dezentral erfolgen. In diesem Fall sendet jeder Client beim Rechnerstart eine Meldung an alle anderen Clients im Netzwerk und teilt ihnen mit, unter welchem Namen er präsent ist.
- ▶ **Session Service:** Dieser Kommunikationsmechanismus ermöglicht ähnlich wie TCP einen geordneten Datenaustausch zwischen zwei Rechnern in Form von Paketen. Dabei wird die Integrität überprüft, und fehlerhafte oder verlorene Pakete werden neu angefordert.
- ▶ **Datagram Service:** Bei dieser Variante zum Session Service gibt es keine Überprüfung, ob die Daten ordnungsgemäß ankommen. Dafür hat der Datagram Service den Vorteil, dass Daten an mehrere Rechner gleichzeitig versandt werden können.

**WINS** Unter Windows wird der NBNS durch den *Windows Internet Name Service* (WINS) realisiert. WINS-Server können mit Samba oder mit aktuellen Windows-Versionen eingerichtet werden.

Bei aktuellen Windows- und Samba-Versionen ist WINS optional, insbesondere, wenn es im lokalen Netz ohnedies einen Nameserver gibt. Wenn Samba in Kombination mit IPv6 zum Einsatz kommt, ist ein Nameserver anstelle von WINS sogar zwingend erforderlich. Unter Linux können Sie dazu das im vorigen Kapitel beschriebene Programm `dnsmasq` einsetzen.

**Browsing** Woher weiß ein Windows-Client, welche anderen Rechner sich im Netz befinden? Die Antwort lautet: durch Browsing. Mit diesem Begriff wird die Verwaltung der im Netz befindlichen Rechner bezeichnet. Damit sich nicht jeder Rechner selbst darum kümmern muss, übernimmt ein sogenannter Master-Browser diese Verwaltungsaufgabe. In größeren Netzen wird er von einem oder mehreren Backup-Browser(n) unterstützt. Samba ist bei Bedarf ebenfalls in der Lage, als Browser aufzutreten.

Wer die Rolle als Browser übernimmt, ist nicht fest vorgeschrieben, sondern wird unter den im Netz befindlichen Rechnern dynamisch ausgehandelt – je nachdem, welche Rechner sich gerade im Netz befinden und welcher dieser Rechner am besten dazu geeignet ist. Oft kommt einfach der Rechner mit der neuesten Betriebssystemversion zum Zuge. Dabei muss es sich keineswegs automatisch um den WINS-Server handeln, sofern es im Netz überhaupt einen gibt. Der Browsing-Ansatz wird also sehr dezentral gehandhabt.

Leider gelingt es vor allem Windows-PCs nicht immer, die Browsing-Liste mit dem tatsächlichen Zustand des Netzwerks synchron zu halten. Das liegt auch daran, dass die Clients einen Cache mit dem zuletzt gültigen Zustand verwalten, um die Netzbelastung zu minimieren. Die Aktualisierung dieses Caches dauert oft geraume Zeit. Die schnellste Lösung ist übrigens – ganz Windows-typisch! – ein Neustart des betroffenen Rechners.

Es gibt etliche Versionen des Protokolls Server Message Block (SMB). [Tabelle 28.1](#) fasst zusammen, welche Protokollversion in welcher Windows- und in welcher Samba-Version unterstützt wird. Die Zuordnung zwischen SMB- und Samba-Versionen ist insofern nicht ganz einfach, als neue SMB-Versionen von Samba anfänglich nur teilweise unterstützt werden. So wurden beispielsweise einige SMB-2.1-Funktionen bereits in Samba 3.5 realisiert.

SMB-Versionen

| SMB     | Windows                  | Samba                  |
|---------|--------------------------|------------------------|
| SMB 1.0 | ältere Windows-Versionen | ältere Samba-Versionen |
| SMB 2.0 | Windows Vista            | ab Samba 3.6           |
| SMB 2.1 | Windows 7                | ab Samba 3.6           |
| SMB 3   | Windows 8                | ab Samba 4.1           |
| SMB 3.1 | Windows 10               | ab Samba 4.3           |

**Tabelle 28.1** SMB-Versionen

Grundsätzlich sind sowohl Windows als auch Samba abwärtskompatibel, sodass das Zusammenspiel unterschiedlicher Windows- und Samba-Versionen in der Regel problemlos gelingt. Für dieses Kapitel spielen die SMB-Versionen ohnedies eine untergeordnete Rolle: Die hier beschriebenen Grundfunktionen existieren zum größten Teil schon seit vielen Jahren. Fortgeschrittene Samba-Konfigurationen mit Active Directories beschreibe ich hier nicht. Sie gehen über den Rahmen dieses Buchs hinaus.

## Zugriffsrechte und Sicherheitssysteme

Der Begriff *Shares* bezeichnet gleichermaßen Verzeichnisse oder Drucker, die anderen Rechnern zur Verfügung gestellt werden. Die deutsche Übersetzung lautet wenig elegant *Freigaben*. Es gibt verschiedene Formen der Zugriffssteuerung, die regeln, wer auf welche Freigaben zugreifen darf:

- ▶ **Share-Level-Sicherheit:** Bei der einfachsten Form der Zugriffssteuerung bekommt jedes Verzeichnis und jeder Drucker ein eigenes Passwort. Dieses Verfahren kommt aber kaum noch zum Einsatz. Der naheliegende Nachteil ist die große Anzahl erforderlicher Passwörter: Wenn auf zehn Rechnern jeweils drei Objekte freigegeben werden, ergeben sich daraus bereits 30 Passwörter. In größeren Netzen führt das naturgemäß zu chaotischen Zuständen.
- ▶ **Workgroup-Sicherheit:** Diese Erweiterung der Share-Level-Sicherheit erlaubt einen gegenseitigen Zugriff auf Objekte nur dann, wenn die Rechner derselben Arbeitsgruppe angehören. Das verbessert die Sicherheit noch nicht nennenswert: Jeder Rechner kann sich ohne zentrale Administration einer beliebigen Arbeitsgruppe zugehörig erklären. Share-Level-Sicherheit funktioniert also nach einem dezentralen Peer-to-Peer-Verfahren, im Gegensatz zu den zunehmend zentralistischen Client/Server-Verfahren für User-Level und Domain-Level.
- ▶ **User-Level-Sicherheit:** Die User-Level-Sicherheit setzt auf der Client-Seite voraus, dass sich der Anwender mit Name und Passwort anmeldet. Wenn der Anwender nun irgendwelche Daten eines Samba-Servers bzw. eines anderen Windows-PCs nutzen möchte, gelten sein aktueller Name und sein Passwort als Zugangsberechtigung. Außerdem müssen beide Rechner zur selben Arbeitsgruppe gehören.

Jedem Netzwerkobjekt ist also nicht einfach ein Passwort zugeordnet. Stattdessen ist es mit einem Benutzer verbunden oder mit einer Liste namentlich aufgezählter Benutzer oder mit allen Benutzern einer Gruppe. Wenn User-Level-Sicherheit mit Samba implementiert wird, ist eine eigene Datenbank mit Benutzernamen, Gruppenzugehörigkeiten und Passwörtern erforderlich.

Dazu ein Beispiel: Anwender X arbeitet auf Rechner A. Damit X Daten vom Samba-Server S abrufen kann, muss X sowohl auf A als auch auf S als Benutzer registriert sein, und zwar jeweils mit dem gleichen Namen und Passwort. Nun geht Rechner A kaputt. X weicht auf Rechner B aus. Damit er auf seine Daten auf S zugreifen kann, muss auch auf B der Benutzer X (wieder mit Passwort) geschaffen werden.

Wenn X sich entschließt, sein Passwort zu ändern, muss diese Änderung auf dem Server S und in der Folge auf jedem Client (A, B ...) durchgeführt werden. Die dezentrale Passwortverwaltung und Authentifizierung ist also ein immanentes Problem bei diesem Konzept.



- ▶ **Domain-Level-Sicherheit:** Mit Windows NT 4 hat Microsoft sogenannte *Domänen* in seine Netzwerkwelt eingeführt. Das Konzept der Zugriffsverwaltung ist ganz ähnlich wie bei der User-Level-Sicherheit. Die Unterschiede betreffen die Art und Weise, wie die Benutzerdatenbank verwaltet wird und wie die Authentifizierung erfolgt.

Die Clients greifen beim Login auf die zentral vom Server verwaltete Benutzerdatenbank zurück. Die Zugriffsrechte werden durch eine Art Login-Token verwaltet. Der Client erhält beim Login eine Zugriffsinformation, die bis zum Logout im gesamten Netzwerk gilt. Dieser Unterschied ist für den Anwender zwar nicht sichtbar, stellt aber einen fundamentalen Unterschied bei der internen Verwaltung dar und ist für den Server deutlich effizienter zu handhaben.

Domain-Level-Sicherheit setzt voraus, dass es im Netzwerk einen *Primary Domain Controller* (PDC) gibt. In größeren Netzen können dem PDC einige *Backup Domain Controller* (BDC) zur Seite gestellt werden, damit nicht alles stillsteht, nur weil der PDC gerade ausgefallen ist.

- ▶ **Active Directories:** Um die Verwaltung großer Netzwerke zu vereinfachen, hat Microsoft die Domain-Level-Sicherheit um sogenannte Active Directories erweitert. Zur Authentifizierung wird das *Lightweight Directory Access Protocol* (LDAP) eingesetzt. Mithilfe von Active Directories können das Netzwerk und seine Domänen hierarchisch organisiert werden. Außerdem erfolgt die Verwaltung der Rechnernamen durch einen Nameserver, also wie unter Linux.

Samba versteht clientseitig schon lange alle fünf aufgezählten Sicherheitssysteme. Die Rolle eines Active-Directory-Domain-Servers kann Samba aber erst seit Version 4 übernehmen. Dieses Buch behandelt allerdings nur das User-Level-Sicherheitssystem. Man spricht in diesem Zusammenhang auch von einer Samba-Konfiguration als Standalone-Server. Wenn Sie Samba als PDC oder als Active-Directory-Server einsetzen möchten, benötigen Sie in jedem Fall weiterführende Literatur.

### Zentrale oder dezentrale Server-Topologie?

Losgelöst vom Sicherheitssystem gibt es zwei fundamental unterschiedliche Strategien, wie Rechner in einem lokalen Netzwerk via Samba Daten austauschen:

- ▶ **Zentrale Topologie:** Ein zentraler Samba-Server stellt allen Benutzern Netzwerkverzeichnisse zur Verfügung. Es ist die Aufgabe des Administrators, die Zugriffsrechte der einzelnen Verzeichnisse so einzustellen, dass es sowohl private Verzeichnisse für individuelle Benutzer als auch mehr oder weniger öffentliche Verzeichnisse zum Datenaustausch in Benutzergruppen gibt. Anstatt um einen selbst konfigurierten Windows- oder Linux-Rechner kann es sich bei dem Samba-Server auch ganz einfach um ein NAS-Gerät handeln.

Die Vorteile dieser Konfiguration bestehen darin, dass alle Daten auf dem Server zentral gesichert werden können und dass sich die einzelnen Benutzer nicht selbst um das Einrichten von Netzwerkverzeichnissen kümmern müssen. Natürlich gibt es auch Nachteile: Das System ist relativ unflexibel, und jede Änderung muss von einem Administrator durchgeführt werden. Außerdem sind die Folgen eines Server-Ausfalls fatal für das ganze Netzwerk.

- ▶ **Dezentrale Topologie:** In diesem Fall stellt jeder Rechner, der im Netzwerk Daten für andere Benutzer zur Verfügung stellen will, diese selbst zur Verfügung. Sowohl Windows als auch Linux (genau genommen: Gnome bzw. KDE) unterstützen den Anwender dabei durch relativ einfach zu nutzende Freigabedialoge.

Der Vorteil dieser Konfigurationsvariante ist der dezentrale Ansatz, bei dem jeder für sich selbst verantwortlich ist und kein Administrator erforderlich ist. Mit zunehmender Netzwerkgröße wird die Konfiguration naturgemäß unübersichtlich und zentrale Backups sind nahezu unmöglich.

Eine zentrale Topologie ist im Unternehmenseinsatz zweckmäßiger. Wenn es aber nur darum geht, im privaten Umfeld rasch ein paar Dateien von einem Rechner zum nächsten zu kopieren, ist eine Ad-hoc-Konfiguration durch den entsprechenden KDE- oder Gnome-Dialog natürlich ausreichend. Das Hauptproblem besteht darin, dass die von KDE bzw. Gnome gebotenen Konfigurationsdialoge heute beinahe ebenso unausgegoren sind wie vor zehn Jahren.

## 28.2 Basiskonfiguration und Inbetriebnahme

**Installation** Bei vielen Distributionen gibt es getrennte Pakete für die Client- und Server-Anwendung. Die Client-Pakete sind zumeist standardmäßig installiert, sodass ein Zugriff auf Netzwerkverzeichnisse auf Anhieb funktionieren sollte. Wenn Sie selbst Netzwerkverzeichnisse freigeben möchten, brauchen Sie auch die Server-Funktionen, die bei den meisten Distributionen im Paket `samba` verpackt sind.

**Start** Samba stellt seine Dienste durch zwei Hintergrundprozesse zur Verfügung:

- ▶ `nmbd` dient zur internen Verwaltung und als Nameserver. Der Dämon kümmert sich auch um die Browsing-Funktionen. `nmbd` kann auch als Master-Browser oder als WINS-Server fungieren.
- ▶ `smbd` stellt die Schnittstelle für die Clients dar und gibt diesen Zugang zu Verzeichnissen, Druckern und zur aktuellen Browsing-Liste.

Bei aktuellen Distributionen werden die Prozesse durch `systemd` gestartet. Die Service-Namen variieren je nach Distribution: Debian und Ubuntu verwenden beispielsweise `smbd` und `nmbd`, CentOS, Fedora, RHEL und SUSE dagegen `smb` und `nmb`. Wenn

Sie Samba als Domain Controller für Active Directories konfigurieren, müssen Sie außerdem den Dienst `samba-ad-dc` starten. Darauf gehe ich in diesem Kapitel aber nicht ein.

Vergessen Sie nicht, dass Systemdienste unter CentOS, Fedora, RHEL und SUSE nicht automatisch nach der Installation des Pakets gestartet werden. Sie müssen also für jeden Dienst `systemctl start name` und `systemctl enable name` ausführen (siehe [Abschnitt 12.5, »Systemprozesse \(Dämonen\)«](#)).

Mit `smbstatus` testen Sie unkompliziert, welche Samba-Version auf Ihrem Rechner **Samba-Version** läuft:

```
root# smbstatus
Samba version 4.6.4 ...
```

Sollte dieses Kommando nicht zur Verfügung stehen, können Sie die Version auch mit `smbd -V` ermitteln.

Als zentrale Konfigurationsdatei für Samba dient `/etc/samba/smb.conf`. Die Datei setzt sich aus einem `[global]`-Abschnitt für die Grundeinstellungen sowie beliebig vielen weiteren Abschnitten für die Freigabe von Verzeichnissen, Druckern etc. zusammen. Jeder Abschnitt wird durch `[ressourcename]` eingeleitet. Kommentare beginnen wahlweise mit den Zeichen `;` oder `#`. Es ist aber nicht zulässig, im Anschluss an eine Parametereinstellung einen Kommentar hinzuzufügen. Kommentare beanspruchen also immer eine ganze Zeile. **smb.conf**

Die folgenden Zeilen zeigen leicht gekürzt den globalen Abschnitt der Samba-Standardkonfiguration unter Ubuntu. Bei anderen Distributionen ist die Datei mitunter noch kürzer, weil darauf verzichtet wird, Defaulteinstellungen explizit zu wiederholen. Nicht abgedruckt sind hier die Abschnitte `[printers]` und `[print$]`, die den Zugriff auf Drucker und Druckertreiber erlauben.

```
# Datei /etc/samba/smb.conf bei Ubuntu
[global]
workgroup          = WORKGROUP
server string      = %h server (Samba, Ubuntu)
dns proxy          = no
log file           = /var/log/samba/log.%m
max log size       = 1000
syslog             = 0
panic action       = /usr/share/samba/panic-action %d
server role        = standalone server
passdb backend     = tdbsam
obey pam restrictions = yes
unix password sync = yes
passwd program     = /usr/bin/passwd %u
passwd chat        = ...
```

```
pam password change = yes
map to guest         = bad user
usershare allow guests = yes
```

Unter CentOS, Fedora und RHEL sieht `smb.conf` recht ähnlich aus. Dort sind allerdings noch einige Zeilen enthalten, um alle Heimatverzeichnisse als Netzwerkverzeichnisse freizugeben. Außerdem lautet der `workgroup`-Name `SAMBA` anstelle von `WORKGROUP`.

**Kommentare aus `smb.conf` entfernen** Bei einigen Distributionen wird `smb.conf` gleichzeitig auch zur Dokumentation verwendet. Die resultierende Konfigurationsdatei ist dann endlos lang und unübersichtlich. Abhilfe schaffen die drei folgenden Kommandos:

```
root# cd /etc/samba
root# cp smb.conf smb.conf.orig
root# egrep -v '^#|^;' smb.conf.orig > smb.conf
```

**Server-Identifizierung** Mit `workgroup` stellen Sie den Namen der Arbeitsgruppe ein. Das ist wahrscheinlich die erste Einstellung, die Sie ändern werden – um dort den Namen Ihrer eigenen Arbeitsgruppe einzustellen, innerhalb derer Samba agieren soll.

`server string` gibt an, unter welchem Namen sich der Server identifiziert. `%h` wird dabei durch den Hostnamen ersetzt.

**WINS** `dns proxy = no` bewirkt, dass Samba, wenn es als WINS-Server agiert, nicht auf DNS zurückgreift, um Windows-Hostnamen aufzulösen. Sofern es in Ihrem LAN einen lokalen Nameserver gibt, sollten Sie diesen Parameter auf `yes` einstellen. Die Standardeinstellung `no` ist nur zweckmäßig, wenn es keine lokalen bzw. schnell erreichbaren Nameserver gibt.

**Logging** Die Parameter `log file`, `max log size` und `syslog` steuern, welche Daten Samba wo protokolliert.

Bei einem Absturz von Samba wird das Script `panic-action` ausgeführt. Es sendet eine E-Mail an `root`, die Informationen zum aufgetretenen Fehler enthält. `panic-action` bleibt wirkungslos, wenn auf dem Server kein E-Mail-System installiert ist.

**Sicherheitsmodell** Aufgrund der Einstellung `server role = standalone` gilt das User-Level-Sicherheitsmodell. Alternativ würde auch die Einstellung `security = user` dieses Sicherheitsmodell aktivieren. Selbst wenn beide Einstellungen fehlen, gilt in aktuellen Samba-Versionen User-Level-Sicherheit als Defaulteinstellung.

**Passwörter** `passdb backend` gibt an, wie die Samba-Passwörter verwaltet werden sollen. Zur Auswahl stehen `smbpasswd` (eine einfache Textdatei), `tdbsam` (TDB, ein relativ einfaches Datenbanksystem) oder `ldapsam` (LDAP). `tdbsam` ist zumeist die richtige Einstellung für kleine bis mittelgroße Netzwerke bis ca. 250 Clients. Das früher populäre System `smbpasswd` sollte nicht mehr verwendet werden, weil damit keine erweiterten Attribute gespeichert werden können (*SAM Extended Controls*).

Die Schlüsselwörter `unix password sync`, `passwd chat` und `pam password change` beschreiben, ob und wie Samba seine Passwörter mit den Linux-Passwörtern abgleichen soll. Details zur Verwaltung der Samba-Passwörter folgen in [Abschnitt 28.3](#), »Passwortverwaltung«.

`map to guest` und `usershare allow guests` regeln, wie Samba mit nicht authentifizierten Benutzern umgeht, also mit Benutzern, die sich mit einem ungültigen Namen oder Passwort anmelden. Die Bedeutung dieser und einiger weiterer `guest`-Parameter ist in [Abschnitt 28.4](#), »Netzwerkverzeichnisse«, beschrieben. Gäste

Unter Debian und Ubuntu enthält `smb.conf` einige Anweisungen, die eigentlich überflüssig sind: Beispielsweise hat die Einstellung `obey pam restrictions = yes` nur dann Einfluss auf die Passwortverwaltung, wenn Passwörter nicht verschlüsselt werden. Da dies standardmäßig der Fall ist, wird die Einstellung ignoriert. Samba-Standards

Im weiteren Verlauf dieses Kapitels werden Sie noch eine Menge weiterer Samba-Parameter kennenlernen, aber natürlich bei Weitem nicht alle. Detaillierte Informationen zu allen Einstellmöglichkeiten gibt man `smb.conf`.

## Konfigurationsänderungen und Status

Damit Änderungen an `smb.conf` wirksam werden, müssen Sie Samba auffordern, die Konfigurationsdateien neu einzulesen:

```
root# systemctl reload smb[d]
root# systemctl reload nmb[d]
```

Wenn Sie größere Änderungen an `smb.conf` durchführen möchten, sollten Sie die Datei zuerst mit `testparm` auf syntaktische Fehler überprüfen: testparam

```
root# testparm
Load smb config files from /etc/samba/smb.conf
Processing section "[printers]"
Processing section "[print$]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions <Return>
[global]
  server string = %h server (Samba, Ubuntu)
  ...
```

Wenn Sie `testparm` mit der Option `-v` ausführen, liefert das Kommando eine schier endlose Liste mit den Einstellungen aller möglichen `smb.conf`-Optionen. Das ist manchmal praktisch, wenn Sie sich nicht sicher sind, welche Einstellungen standardmäßig gelten – also bei Optionen, die Sie nicht selbst explizit eingestellt haben.

Den aktuellen Zustand des Samba-Servers ermitteln Sie mit `smbstatus`. Das Kommando liefert auch eine Liste aller zurzeit aktiven Verbindungen.

## Samba absichern

**Firewall** Unter CentOS, Fedora, SUSE und RHEL blockiert die standardmäßig aktive Firewall Samba. Damit die Netzwerkdienste genutzt werden können, müssen Sie die Samba- bzw. Windows-spezifischen TCP-Ports 135, 139 und 445 sowie die UDP-Ports 137, 138 und 445 für die Schnittstelle zum lokalen Netzwerk freigeben – und das sowohl auf dem Server als auch auf den Client-Rechnern.

Bei SUSE-Distributionen verwenden Sie zur Firewall-Konfiguration am besten YaST. Bei Fedora, CentOS und RHEL ist `firewall-cmd` das Kommando der Wahl. Sie können damit entweder die Zone der Netzwerkschnittstelle ändern, über die die Samba-Daten fließen, oder Sie behalten die Zone bei und definieren eine Ausnahmeregel für Samba. Dieser Weg ist hier skizziert:

```
root# firewall-cmd --get-zone-of-interface=enp0s3 (aktive Zone herausfinden)
FedoraWorkstation
root# firewall-cmd --permanent --zone=FedoraWorkstation --add-service=samba
root# firewall-cmd --reload
```

Die Öffnung der SMB-Ports innerhalb des lokalen Netzwerks ist unumgänglich, um Samba nutzen zu können. Eine generelle Deaktivierung der Firewall ist aber nicht empfehlenswert! Zudem sollte das gesamte lokale Netzwerk unbedingt zum Internet hin durch eine weitere Firewall abgesichert sein, die beispielsweise auf dem ADSL-Router oder Gateway-Rechner läuft. Ist dies nicht der Fall, sind die freigegebenen Verzeichnisse für alle Welt zugänglich bzw. nur noch durch die Passwörter der Samba-Benutzerverwaltung abgesichert. Hintergrundinformationen zum Thema Firewalls können Sie in [Kapitel 37](#) nachlesen.

**Schnittstellen** Unabhängig von der Firewall schadet es nicht, auch innerhalb der Samba-Konfiguration Vorsicht walten zu lassen: Dazu geben Sie mit `interfaces` explizit an, über welche Netzwerkschnittstellen Samba kommunizieren soll. Die Angabe der Schnittstellen erfolgt nicht über die Namen der Schnittstellen, sondern über den von diesen Schnittstellen genutzten Adressbereich. Sie können auch mehrere Bereiche angeben, wobei Sie diese einfach durch Leerzeichen trennen. Vergessen Sie `localhost` nicht – sonst funktionieren auf dem Server Administrationswerkzeuge wie `smbclient` nicht!

Die `interfaces`-Option ist dann relevant, wenn es auf Ihrem Rechner mehrere Netzwerkschnittstellen gibt. Auf vielen Rechnern gibt es nicht nur Schnittstellen zu physikalischen Netzwerkadaptern, sondern auch zu den virtuellen Netzwerkadaptern diverser Virtualisierungsprogramme! Standardmäßig bedient Samba *alle* Netzwerkschnittstellen.

Die Einstellungen durch `interfaces` werden nur wirksam, wenn Sie außerdem wie im folgenden Listing die Option `bind interfaces only` aktivieren.

Des Weiteren können Sie mit `hosts allow` explizit aufzählen, welche Rechner mit Samba kommunizieren dürfen. Die Hostnamen, IP-Adressen oder IP-Adressbereiche müssen durch Leerzeichen voneinander getrennt werden. `hosts allow` erlaubt eine noch genauere Selektion als `interfaces`. Ergänzend können Sie mit `hosts deny` einzelnen Hosts oder Adressen die Nutzung von Samba verbieten. **Hosts**

Grundsätzlich kommuniziert Samba sowohl über IPv4 als auch über IPv6. Bei einer Dual-Stack-Konfiguration im LAN ist es mitunter aus Sicherheitsgründen wünschenswert, dass Samba nur IPv4 spricht. Dazu geben Sie einfach bei `interfaces` und `host` ausschließlich IPv4-Adressen an. Sie sollten in solchen Fällen Hostnamen vermeiden und lieber die IP-Adressen angeben, weil bei der Auflösung der Hostnamen nicht immer vorhersehbar ist, ob das Ergebnis IPv4- oder IPv6-Adressen sind. **IPv4 und IPv6**

Diese Art der Konfiguration funktioniert auch umgekehrt: Wenn Sie mit `interfaces` ausschließlich IPv6-Adressen angeben, dann ist IPv4 gesperrt.

Samba ist grundsätzlich zu allen SMB-Protokollversionen seit 1.0 kompatibel. Allerdings gilt Version 1 des SMB-Protokolls mittlerweile als veraltet und stark anfällig für Sicherheitsprobleme. Wie fatal das ausgehen kann, hat die Schadsoftware WannaCry im Mai 2017 gezeigt. Deswegen gibt es aktuell das Bestreben, sowohl aufseiten von Microsoft als auch bei den Samba-Entwicklern, das SMB-Protokoll in der Version 1 zu deaktivieren. Das gelingt ganz einfach, indem Sie die beiden folgenden Zeilen in `smb.conf` einbauen: **SMB-Protokoll**

```
# /etc/samba/smb.conf
[global]
...
client min protocol = SMB2
server min protocol = SMB2
```

Die Client-Einstellung bewirkt, dass Sie auf Ihrem Linux-Rechner keine Netzwerkverzeichnisse von Uralt-PCs bzw. anderen Geräten sehen. Das betrifft einerseits Windows XP, andererseits aber auch (sehr) alte NAS-Geräte, Audio- und Video-Player, Spielkonsolen etc.

Die Server-Einstellung hat zur Folge, dass Ihr Linux-Rechner selbst keine Verzeichnisse im alten Protokoll anbietet. Im Zusammenspiel mit aktuellen Linux- und Windows-Installationen sind keine Probleme zu erwarten. Wiederum werden höchstens einige ältere Geräte Ihre im Netzwerk angebotenen Verzeichnisse nicht mehr sehen.

Es ist zu erwarten, dass die obigen Einstellungen in naher Zukunft von vielen Distributionen und vermutlich auch von Samba selbst standardmäßig verwendet werden.

- Gäste** Die Einstellung `map to guest` verbietet schließlich allen Benutzern, die sich nicht richtig beim Server authentifizieren können, jeglichen Zugriff. Je nach Anwendung kann es zwar durchaus sinnvoll sein, auch für Gäste Verzeichnisse einzurichten, die ohne Authentifizierung gelesen oder sogar verändert werden dürfen. Wenn dies aber nicht erforderlich ist, sollten Sie Gäste von vornherein aussperren.

```
# /etc/samba/smb.conf
[global]
...
bind interfaces only = yes
interfaces           = 192.168.0.0/24 127.0.0.1
hosts allow          = clientA clientB clientC
map to guest         = never
```

- SELinux** Unter Fedora und RHEL überwacht SELinux standardmäßig alle Samba-Aktivitäten. Damit die Freigabe von Heimatverzeichnissen unkompliziert funktioniert, müssen Sie den SELinux-Parameter `samba_enable_home_dirs` aktivieren:

```
root# setsebool -P samba_enable_home_dirs on
```

Wenn Sie ein anderes Verzeichnis freigeben möchten, müssen Sie hierfür das Attribut `samba_share_t` setzen. Dazu führen Sie die beiden folgenden Kommandos aus, im folgenden Beispiel für das Verzeichnis `/samba/shares`:

```
root# semanage fcontext -a -t samba_share_t '/samba/shares(/.*)?'
root# restorecon -R /samba/shares
```

### SELinux-Konfiguration

Es gibt noch diverse weitere SELinux-Parameter, die steuern, was Samba alles (nicht) darf. Einen Überblick über all diese Parameter samt der Möglichkeit, deren Einstellung mit einem Mausklick zu verändern, gibt das Programm `system-config-selinux` aus dem Paket `polycoreutils-gui`. Suchen Sie nach dem Start im Dialogblatt **BOOLEAN** nach *samba*! Weitere Tipps zu SELinux finden Sie in [Kapitel 38](#), »SELinux und AppArmor«, sowie nach der Installation des Paketes `selinux-policy-doc` mit `man samba_selinux`.

### Logging

Die beiden Samba-Dienste `smbd` und `nmbd` protokollieren globale Ereignisse in die Dateien `/var/log/samba/log.smbd` und `log.nmbd`. Weder der Name noch der Ort dieser beiden Logging-Dateien kann durch `smb.conf` verändert werden.

Der Parameter `log file` im globalen Abschnitt von `smb.conf` gibt an, wohin client-spezifische Nachrichten protokolliert werden sollen. Die Voreinstellung `/var/log/`



samba/log.%m bewirkt, dass für jeden Client, der auf Samba zugreift, eine eigene Logging-Datei mit dem Namen log.hostname erzeugt wird.

max log size = 1000 limitiert die maximale Größe auf 1000 KiB. Wenn eine Logging-Datei größer wird, benennt Samba sie in name.old um. syslog = 0 bedeutet nicht etwa, dass Syslog nicht verwendet wird, sondern dass in /var/log/syslog nur Fehlermeldungen protokolliert werden sollen. Alternative Einstellungen sind 1, 2, 3 etc., wenn Sie auch Warnungen, Notizen sowie Debugging-Nachrichten protokollieren möchten.

Vorsicht ist beim Einsatz von logrotate geboten (siehe [Abschnitt 18.9](#)): Dieses Programm archiviert in der bei Debian, SUSE und Ubuntu üblichen Standardeinstellung einmal pro Woche log.smbd und log.nmbd und löscht gleichzeitig alle Archivversionen, die älter als zwei Monate sind. logrotate ignoriert aber die viel schneller wachsenden clientspezifischen Logging-Dateien log.hostname. **logrotate**

Besser durchdacht ist hier die entsprechende Konfigurationsdatei bei Fedora und Red Hat, die sich um *alle* Logging-Dateien in /var/log/samba kümmert:

```
# /etc/logrotate.d/samba bei CentOS, Fedora und Red Hat
/var/log/samba/*
{
    notifempty
    olddir /var/log/samba/old
    missingok
    sharedscripts
    copytruncate
}
```

Eine andere Lösung besteht darin, dass Sie in smb.conf die Einstellung log file = /var/log/samba/log.smbd verwenden. Damit erreichen Sie, dass smbd globale wie clientspezifische Nachrichten in derselben Datei protokolliert. Solange Sie nicht Fehler in der Samba-Konfiguration suchen müssen, ist das am praktischsten.

## 28.3 Passwortverwaltung

Standardmäßig gilt für Samba security = user, also User-Level-Sicherheit. Damit der Beispielnutzer peter ein Netzwerkverzeichnis nutzen kann, müssen die folgenden Voraussetzungen erfüllt sein:

- ▶ Auf dem Server muss es einen Linux-Account mit dem Namen peter geben. Dieser Account ist für die Verwaltung der Zugriffsrechte erforderlich. Samba greift also auf die Linux-Zugriffsrechte zu, um zu entscheiden, welcher Nutzer welche Datei lesen bzw. verändern darf.

Der Linux-Account muss nicht aktiv sein. Oft ist es aus Sicherheitsgründen zweckmäßig, den Account explizit mit `passwd -l peter` zu sperren. Sie verhindern damit, dass sich Samba-Benutzer mittels SSH auf dem Server einloggen können.

- ▶ `peter` und sein Passwort müssen in der Samba-Benutzerdatenbank enthalten sein. Aus technischen Gründen werden die Passwörter und andere Metadaten eines Samba-Accounts unabhängig von denen des zugeordneten Linux-Accounts verwaltet. Es muss also parallel zum Samba-Account auch einen Linux-Account geben; das Passwort des Linux-Accounts spielt für Samba aber keine Rolle.
- ▶ `smb.conf` muss Einträge für Netzwerkverzeichnisse enthalten, die `peter` benutzen darf. Wie diese Einträge aussehen, erkläre ich in [Abschnitt 28.4](#), »Netzwerkverzeichnisse«.

### Samba-Passwörter

Bevor ein Benutzer im lokalen Netzwerk auf ein Verzeichnis zugreifen kann, muss er sich beim Samba-Server identifizieren. Beim Verbindungsaufbau von einem Windows-Client aus werden dazu der Windows-Login-Name und eine verschlüsselte Zeichenkette für das Passwort übertragen. Bei Linux-Clients können diese Daten nicht aus dem Linux-Login übernommen werden, weswegen zumeist ein eigener Dialog zur Eingabe des Samba-Benutzernamens und -Passworts erscheint. Anschließend geht es bei Windows-Clients weiter: Auch in diesem Fall wird nicht das Passwort selbst, sondern ein verschlüsselter Code an den Samba-Server übertragen.

Aus Sicherheitsgründen kann das Passwort nicht aus der verschlüsselten Passwortzeichenkette rekonstruiert werden. Die Zeichenkette wird vielmehr vom Samba-Server mit einer gleichermaßen verschlüsselten Zeichenkette verglichen. Wenn beide Zeichenketten bzw. in der Fachsprache beide »Hash-Codes« übereinstimmen, stimmen auch die Passwörter überein.

**smbpasswd** Damit `peter` also ein Samba-Verzeichnis nutzen kann, müssen Sie als Linux-Systemadministrator den Samba-Account `peter` anlegen. Dabei hilft Ihnen das Kommando `smbpasswd` mit der Option `-a` (*add*). Als Passwort geben Sie dieselbe Zeichenkette an, die der Benutzer `peter` auch unter Windows hat. Um ein bereits vorhandenes Samba-Passwort zu verändern, verwenden Sie `smbpasswd` ohne die Option `-a`.

```
root# smbpasswd -a peter
New SMB password: *****
Retype new SMB password: *****
Added user peter.
Password changed for user peter.
```

Beachten Sie, dass `smbpasswd` nur funktioniert, wenn der Linux-Benutzer `peter` auch auf dem lokalen System existiert (Datei `/etc/passwd`)! Gegebenenfalls müssen Sie neue Linux-Benutzer vorher mit `useradd` oder `adduser` anlegen.

Wo werden die Samba-Account-Daten für `peter` gespeichert? Der Samba-Server hat dazu eine Benutzer- und Passwort-Datenbank, um die Authentifizierung durchzuführen. In der Vergangenheit kam hierfür oft eine simple Textdatei zum Einsatz (`passdb backend = smbpasswd`).

**smbpasswd-Backend**

Mittlerweile verwenden aber alle Distributionen das TDB-Backend (`passdb backend = tdbsam`). Bei wirklich großen Installationen für sehr viele Benutzer, z. B. in Firmen, ist es zweckmäßig, die Login-Daten via LDAP zu verwalten, worauf ich in diesem Buch aber nicht eingehe.

**TDB-Backend**

TDB steht für *Trivial Database* und ist ein binäres Format zur Speicherung von Datensätzen. Der wesentliche Vorteil im Vergleich zu den herkömmlichen `smbpasswd`-Dateien besteht darin, dass neben dem Login-Namen und dem Passwort weitere Daten und Attribute gespeichert werden. Dazu zählen insbesondere die sogenannten *SAM Extended Controls*, die die Kompatibilität mit aktuellen Windows-Versionen erhöhen. Wo die Passwörter physikalisch gespeichert werden, ist distributionsabhängig:

Die meisten aktuellen Distributionen: `/var/lib/samba/private/passdb.tdb`  
 Alte Debian- und Ubuntu-Versionen: `/var/lib/samba/passdb.tdb`  
 Alte SUSE-Versionen: `/etc/samba/passdb.tdb`

Bei Bedarf können Sie in `smb.conf` durch `passdb backend = tdbsam:dateiname` eine andere Datei angeben. Mit dem Kommando `smbpasswd` legen Sie einen neuen Samba-Account an bzw. verändern dessen Passwort.

`pdbedit` gibt Zugriff auf alle Account-Informationen: `root`-Benutzer können damit eine Liste aller Samba-Benutzer erstellen (`pdbedit -L -v`), für jeden Account diverse Attribute einstellen etc. Beide Kommandos werten `smb.conf` aus und funktionieren für alle Passwort-Systeme, also `smbpasswd`, `tdbsam` und `ldapsam`. Die folgende Ausgabe ist aus Platzgründen stark gekürzt:

```
root# pdbedit -L -v
Unix username:      peter
NT username:
Account Flags:      [U          ]
User SID:           S-1-5-21-3702490679-2182029830-1044092605-1000
Primary Group SID:  S-1-5-21-3702490679-2182029830-1044092605-513
Home Directory:     \\fedora\peter
```

```

Profile Path:          \\fedora\peter\profile
Domain:               FEDORA
Password last set:    Sam, 10 Jun 2017 06:36:45 CEST
Bad password count   : 0
...

```

## Synchronisation der Samba- und Linux-Passwörter

Bei manchen Installationen ist es wünschenswert, dass sich die Benutzer von Samba-Verzeichnissen direkt am Linux-Server anmelden können, beispielsweise um via SSH zu arbeiten und ihre Dateien mit Linux-Kommandos zu editieren. In solchen Fällen wäre es natürlich zweckmäßig, wenn das Samba- und das Linux-Passwort immer übereinstimmen würden. Schließlich ist es ausgesprochen lästig, jede Passwortänderung mehrfach durchzuführen (im Extremfall gleich dreimal: auf dem Windows-Client, für den Samba-Server und für den Linux-Login).

Die Synchronisierung der Passwörter ist leider nicht ganz einfach zu bewerkstelligen, weil zur Verschlüsselung der Samba-Passwörter ein anderer Algorithmus als für Linux-Passwörter zum Einsatz kommt. Aus diesem Grund erfolgt die Verwaltung der Samba- und Linux-Passwörter getrennt. (Die Algorithmen sind zwar unterschiedlich, es gibt aber eine Gemeinsamkeit: Die gespeicherten Zeichenketten ermöglichen nur die Kontrolle der Passwörter, aber keine Rekonstruktion. Deswegen ist eine Umwandlung oder Konvertierung der gespeicherten Passwörter von einem System in ein anderes unmöglich.)

Die populärste Lösung dieses Problems besteht darin, bei jedem Client-Aufruf von `smbpasswd` zur Veränderung des eigenen Passworts parallel auch das Linux-Passwort zu verändern. Unter Ubuntu enthält `smb.conf` dafür bereits alle erforderlichen Einstellungen:

```

# /etc/samba/smb.conf
[global]
...
unix password sync = yes
pam password change = yes
passwd chat = *Enter\snew\s*\spassword:* %n\n
              *Retype\snew\s*\spassword:* %n\n
              *password\supdated\ssuccessfully* .

```

`unix password sync = yes` aktiviert die Synchronisierung. Dabei wird wegen `pam password change` PAM eingesetzt. PAM steht für *Pluggable Authentication Modules* und bezeichnet eine Sammlung von Bibliotheken zur Administration von Passwörtern. Der früher erforderliche Parameter `passwd program`, der den Pfad des `passwd`-Programms angab, ist für PAM nicht relevant und wird ignoriert. `passwd chat` beschreibt

die Kommunikation zwischen Samba und PAM. Die Zeichenkette wurde im obigen Listing über drei Zeilen verteilt, sie muss in `smb.conf` aber in einer Zeile angegeben werden.

Leider ist die Synchronisation mit vielen Einschränkungen verbunden:

- ▶ `smbpasswd` muss vom jeweiligen Benutzer ausgeführt werden, nicht von `root`! Der Grund: Wenn `smbpasswd` von `root` aufgerufen wird, manipuliert es direkt die Samba-Benutzerdatenbank. Das funktioniert auch, wenn der Samba-Server gar nicht läuft. Wenn das Kommando dagegen von einem gewöhnlichen Benutzer verwendet wird, kommuniziert es mit dem Samba-Server, der die eigentliche Arbeit inklusive der Passwort-Synchronisation erledigt.
- ▶ `smbpasswd` akzeptiert beliebig schlechte Passwörter, z. B. leere oder aus nur einem Buchstaben bestehende Passwörter. Das Linux-Passwortsystem bzw. PAM erzwingt dagegen eine minimale Passwortqualität und verweigert allzu einfache Passwörter. Das kann dazu führen, dass zwar das Samba-Passwort verändert wird, das Linux-Passwort aber nicht. Ab diesem Zeitpunkt sind die Passwörter nicht mehr synchron, weswegen nun jeder weitere Versuch scheitert, das Linux-Passwort neu einzustellen. Um die Passwörter wieder zu synchronisieren, muss `root` das Linux- und Samba-Passwort des betroffenen Benutzers neu einstellen.
- ▶ Die Synchronisation durch `unix password sync` funktioniert nur in eine Richtung: von Samba zu Linux. Wenn ein Benutzer dagegen sein Linux-Passwort durch `passwd` verändert, bleibt das Samba-Passwort unverändert.

Aus meiner persönlichen Erfahrung rate ich von der hier beschriebenen Passwort-Synchronisation ab. Sie ist fehleranfällig und verursacht vielfach mehr Probleme, als sie löst. Verwenden Sie die Einstellung `unix password sync = no`.

### Zuordnung zwischen Windows- und Linux-Benutzernamen

Unter Windows ist als Benutzername beinahe jede Zeichenkette mit bis zu 128 Zeichen möglich, unter Linux dagegen nur eine Zeichenkette mit maximal 32 Zeichen ohne Sonderzeichen oder Leerzeichen. Wenn ein Windows-Benutzer einen Benutzernamen verwendet, der nicht unmittelbar einem Linux-Benutzernamen entspricht, muss die Zuordnung über eine Datei hergestellt werden. Der Name dieser Datei wird in `smb.conf` durch die Option `username map` angegeben:

```
# /etc/samba/smb.conf
[global]
...
username map = /etc/samba/smbusers
```

Jede Zeile der Datei `smbusers` enthält zuerst einen Linux-Benutzernamen, dann das Zeichen `=` und schließlich einen oder mehrere Windows-Benutzernamen. Namen mit Leerzeichen stellen Sie in Anführungszeichen. Sie können die Datei auch benutzen, um mehreren Windows-Benutzern einen Linux-Benutzer zuzuordnen.

```
# /etc/samba/smbusers
peter = "Peter Mayer"
...
```

### Eine falsche Samba-Konfiguration kann das gesamte Linux-System gefährden

`/etc/samba/smbusers` kann das Sicherheitssystem von Samba bzw. Linux aushebeln, wenn `root` einem anderen Benutzer zugeordnet wird! Achten Sie darauf, dass nur `root` die Datei ändern darf:

```
root# chown root /etc/samba/smbusers
root# chmod 644 /etc/samba/smbusers
```

## Und jetzt alles zusammen

Nehmen wir an, in Ihrem lokalen Netzwerk gibt es einen Windows-PC für Peter Mayer, wobei der Login-Name auf diesem Rechner `Peter Mayer` lautet. Auf einem Linux-Server mit Samba gibt es den Account `peter`. Die Datei `smbusers` stellt die Zuordnung zwischen `Peter Mayer` und `peter` her. Unter diesen Voraussetzungen gibt es nun folgende Kombinationen aus Login-Name und Passwort:

- ▶ **Login unter Windows:** `Peter Mayer` und das Windows-Passwort  
Das Windows-Passwort gilt für das lokale Arbeiten am Windows-Rechner. Peter kann sein Windows-Passwort unter Windows ändern.
- ▶ **Login auf dem Server (Linux):** `peter` und das Linux-Passwort  
Das Linux-Passwort gilt für einen direkten Login auf dem Server, sofern der Linux-Account aktiv und nicht gesperrt ist. Peter kann sein Linux-Passwort z. B. nach einem SSH-Login auf dem Server mit dem Kommando `passwd` ändern.
- ▶ **Zugriff auf Netzwerkverzeichnisse:** `Peter Mayer` oder `peter` und das Samba-Passwort  
Das Samba-Passwort gilt für die Nutzung der Netzwerkverzeichnisse. Es sollte mit dem Windows-Passwort übereinstimmen. Ist das nicht der Fall, erscheint unter Windows eine Login-Box, sobald Peter auf ein Netzwerkverzeichnis zugreifen möchte. Peter kann sein Samba-Passwort nach einem SSH-Login auf dem Server mit `smbpasswd` ändern.

Kurz und gut: Nur technisch versierte Benutzer sind in der Lage, ihre drei Passwörter selbst zu ändern – und das auch nur, wenn der Linux-Account aktiv ist. Für alle anderen gilt: Einmal definierte Passwörter werden nie wieder geändert. Vom Sicherheitsstandpunkt aus betrachtet, ist das natürlich alles andere als optimal. Und genau hier liegt der Grund, warum es in Firmen zumeist einen zentralen Server gibt, der für alle Logins und ihre Passwörter zuständig ist (Primary Domain Controller, Active Directories).

## Arbeitstechniken

In der Praxis treten oft zwei Fälle auf:

- ▶ **Eigenes Verzeichnis freigeben:** Sie wollen ein eigenes Verzeichnis freigeben, in der Regel ein Unterverzeichnis Ihres Heimatverzeichnisses, beispielsweise `/home/<loginname>/Bilder`. Die Dateien in diesem Verzeichnis gehören Ihnen, es ist also naheliegend, das Verzeichnis für Ihren Account freizugeben.

Wenn Sie den Zugriff nur nach einem Login zu erlauben, müssten Sie dem- oder derjenigen, der/die auf die Bilder zugreifen soll, Ihr reguläres Linux-Login-Passwort verraten. Das ist selten wünschenswert! Besser ist es zumeist, das Verzeichnis nur für den Lesezugriff freizugeben, dafür aber gleich ohne Passwort (also aus Samba-Sicht: für beliebige Gäste). Diese Vorgehensweise eignet sich natürlich nicht für sensible Daten.

- ▶ **Zentrales Verzeichnis freigeben:** Sie wollen für mehrere Benutzer ein Verzeichnis mit Dateien freigeben – passwortgeschützt, aber ohne Ihr eigenes Passwort preiszugeben. Dann richten Sie dazu einen neuen Benutzer ein, für den kein Linux-Login möglich ist. Nennen wir diesen Benutzer/Account `shareuser`. Als freizugebendes Verzeichnis verwenden Sie im einfachsten Fall `/home/shareuser`. Sie können aber natürlich auch ein beliebiges anderes Verzeichnis wählen, dürfen aber nicht vergessen, die Zugriffsrechte und gegebenenfalls auch den SELinux-Kontext korrekt einzustellen.

Mit `smbpasswd` weisen Sie `shareuser` ein Samba-Passwort zu (natürlich ein anderes als Ihr eigenes). Nachdem Sie das Verzeichnis mit Daten gefüllt und in `smb.conf` eingerichtet haben, teilen Sie Ihren Mitarbeitern die folgenden Informationen mit: den Verzeichnisnamen in Windows-Schreibweise (also `\\hostname\verzeichnisname`), den Account-Namen `shareuser` und das dazugehörige Passwort.

## 28.4 Netzwerkverzeichnisse

Im vorigen Abschnitt habe ich erklärt, welche Voraussetzungen erfüllt sein müssen, damit sich ein Benutzer bei Samba anmelden kann – kurz zusammengefasst: Ein Linux-Account und ein Samba-Passwort müssen vorhanden sein. Jetzt stellt sich nur noch die Frage, welche Ressourcen ein angemeldeter Benutzer sieht und verwenden kann. Entscheidend hierfür sind die `[resourcenname]`-Abschnitte in `smb.conf`.

### Benutzer- verzeichnisse

Die Definition eines Verzeichnisses, auf das ein bestimmter Benutzer zugreifen kann, sieht so aus:

```
# in /etc/samba/smb.conf
...
[verzeichnis1]
  valid users = peter
  path        = /data/verz1
  writeable   = yes
```

Mit dieser Einstellung können der Benutzer `peter` sowie alle Benutzer, die diesem Linux-Account durch `smbusers` zugeordnet sind, das Verzeichnis `/data/verz1` lesen und verändern. Im Dateimanager hat diese Ressource den Namen `verzeichnis1`, also die in eckigen Klammern angegebene Zeichenkette.

Die Bedeutung der Schlüsselwörter ist leicht verständlich: `valid users` gibt den Benutzernamen an. Es ist zulässig, mehrere durch Komma getrennte Benutzernamen anzugeben.

`path` gibt an, welches Verzeichnis des Servers freigegeben wird. Wenn `path` nicht explizit angegeben wird, gibt Samba standardmäßig das Heimatverzeichnis des angegebenen Benutzers frei. `writeable = yes` erlaubt Veränderungen im Verzeichnis. Ohne diese Option hat der Benutzer nur Lesezugriff.

Grundsätzlich gelten für alle Zugriffe die Linux-Zugriffsrechte. Wenn es in `/data/verz1` also eine Datei gibt, die `root` gehört, dann darf der Linux-Benutzer `peter` diese Datei normalerweise nur lesen, aber nicht verändern. Diese Einschränkung gilt ebenso für alle Benutzer des Netzwerkverzeichnisses.

### SELinux kann den Zugriff auf Verzeichnisse verhindern

Wenn Sie Ihren Samba-Server unter CentOS, Fedora oder RHEL einrichten, sollten Sie auch an das schon erwähnte Sicherheitssystem SELinux denken! Sie müssen dem freizugebenden Verzeichnis das Attribut `samba_share_t` zuordnen, damit der Zugriff klappt:

```
root# semanage fcontext -a -t samba_share_t "/data/verz1(/.*)?"
root# restorecon -R /data/verz1
```



Wenn ein Datei-Server für viele Benutzer eingerichtet wird, ist es das Einfachste, dass jeder angemeldete Samba-Benutzer direkt sein Linux-Heimatverzeichnis sieht und bearbeiten darf. Anstatt `smb.conf` nun durch unzählige Einträge der Form

Home-  
Verzeichnisse

```
[benutzername]
  path          = /home/benutzername
  valid users   = benutzername
  writeable     = yes
```

aufzublähen, sieht `smb.conf` die folgende Kurzschreibweise vor, die bei manchen Distributionen (z. B. Fedora) auch gleich standardmäßig in `smb.conf` enthalten ist:

```
[homes]
  writeable     = yes
  browseable    = no
```

Damit wird das Heimatverzeichnis des gerade aktiven Benutzers unter dessen Namen sichtbar. Die Option `browseable = no` bewirkt nicht, wie man vielleicht glauben könnte, dass der Benutzer sein Verzeichnis nicht sieht. Sie verhindert nur, dass das Verzeichnis doppelt sichtbar ist: einmal unter dem jeweiligen Benutzernamen (etwa `peter`) und einmal als `homes`.

### Nochmals SELinux!

Die Standardkonfiguration von SELinux unter CentOS, Fedora und RHEL erlaubt die Freigabe der Heimatverzeichnisse nur, wenn der boolesche Parameter `samba_enable_home_dirs` gesetzt ist:

```
root# setsebool -P samba_enable_home_dirs on
```

Benutzer- und Heimatverzeichnisse ermöglichen es dem Benutzer, seine Dateien zentral auf dem Server zu speichern, bieten aber keine Möglichkeit zum Datenaustausch. Die Verzeichnisse sind für andere Benutzer unsichtbar und unerreichbar. Abhilfe schaffen Gruppenverzeichnisse, die alle Mitglieder einer Gruppe verwenden dürfen. Die Gruppenzuordnung erfolgt durch die Linux-Benutzerverwaltung. Die Gruppe wird mit dem Schlüsselwort `user` in der Schreibweise `@gruppenname` angegeben.

Gruppen-  
verzeichnisse

```
# in /etc/samba/smb.conf
...
[salesdata]
  user          = @sales
  path          = /data/sales
  writeable     = yes
  force group   = +sales
  create mask   = 0660
  directory mask = 0770
```

Beim Zugriff auf Gruppenverzeichnisse ist die richtige Einstellung der Zugriffsrechte von Dateien und Verzeichnissen besonders wichtig. Das gilt auch für Dateien und Verzeichnisse, die neu erstellt werden. `force group = +sales` bewirkt, dass neu erzeugte Dateien oder Verzeichnisse der Gruppe `sales` zugeordnet werden und nicht wie sonst üblich der Standardgruppe des Benutzers. Wenn ein Benutzer nicht Mitglied der Gruppe `sales` ist, darf er nicht auf das Verzeichnis zugreifen.

#### Vorsicht vor »force group«

Verwenden Sie im obigen Fall auf keinen Fall die Einstellung `force group = sales`, also ohne vorangestelltes Plus! Das hätte zur Folge, dass Samba jeden Zugriff auf das Verzeichnis so durchführt, als wäre der gerade aktive Benutzer Mitglied der Gruppe `sales` – und zwar selbst dann, wenn der Benutzer auf Linux-Ebene gar kein Mitglied dieser Gruppe ist! Mit anderen Worten: Mit `force group = sales` geben Sie Benutzern, die der Gruppe `sales` gar nicht angehören, Lese- und Schreibrechte für das Verzeichnis. Bei Gruppenverzeichnissen ist das selten beabsichtigt und kann ein großes Sicherheitsproblem sein!

Die Parameter `create mask` und `directory mask` stellen sicher, dass von Gruppenmitgliedern neu erstellte Dateien und Verzeichnisse von allen anderen Gruppenmitgliedern gelesen und verändert werden können. Die oktale Zahl entspricht dem `chmod`-Wert – siehe `man chmod`. Wenn neue Dateien bzw. Verzeichnisse von anderen Gruppenmitgliedern nur gelesen, aber nicht verändert werden dürfen, verwenden Sie die Werte `0440` und `0550`.

#### Frei zugängliche Verzeichnisse

Noch liberaler ist der Zugriff auf das `share`-Verzeichnis: Jeder Benutzer, der sich bei Samba authentifizieren kann, kann Dateien aus diesem Verzeichnis lesen. Der Schreibzugriff ist in diesem Beispiel deaktiviert:

```
# in /etc/samba/smb.conf
...
[share]
    path      = /data/share
    read only = yes
```

Sie können selbstverständlich auch frei zugängliche Verzeichnisse mit Schreibzugriff einrichten (`writable = yes`). Standardmäßig können alle Benutzer die von anderen Benutzern erzeugten Dateien lesen, aber nicht verändern. Abhilfe schafft die Einstellung von `force group` und der beiden `mask`-Parameter. Uneingeschränkte gegenseitige Schreib- und Leserechte erzielen Sie mit `create mask = 0666` und `directory mask = 0777`.

Alle vorangegangenen Beispiele setzten voraus, dass sich der Benutzer bei Samba authentifizieren kann. Bei entsprechender Konfiguration sieht Samba auch einen Verzeichniszugriff für nicht authentifizierte Benutzer vor. Derartige Benutzer werden im Samba-Jargon als *Gäste* (guest-Benutzer) bezeichnet. Für den Umgang mit Gästen sind die im folgenden Listing zusammengefassten globalen Einstellungen verantwortlich:

Zugriff für nicht  
authentifizierte  
Benutzer

```
# in /etc/samba/smb.conf
[global]
...
map to guest          = bad user
guest account        = nobody
```

map to guest = bad user bewirkt, dass Login-Versuche mit einem nicht existenten Benutzernamen automatisch dem virtuellen Samba-Benutzer `guest` zugeordnet werden. Standardmäßig gibt es allerdings keine Netzwerkverzeichnisse oder andere Ressourcen, die `guest` nutzen darf.

guest account gibt an, welchem Linux-Benutzer Gäste zugeordnet werden. Bei den meisten Linux-Distributionen inklusive Debian und Ubuntu ist hierfür der Benutzer `nobody` vorgesehen.

Verzeichnisse, die für Gäste benutzbar sein sollen, kennzeichnen Sie durch `guest ok = ok`. In aller Regel werden Sie solche Verzeichnisse mit dem Attribut `read only = yes` vor Schreibzugriffen schützen. Denken Sie daran, dass Gäste generell nur solche Dateien lesen bzw. verändern dürfen, die auch der Linux-Benutzer `nobody` lesen bzw. verändern darf.

```
[guest]
path          = /data/guest
guest ok     = yes
read only    = yes
```

Eine Variante zu `guest ok = yes` ist `guest only = yes`: Mit dieser Einstellung kann das Verzeichnis nur von Gästen, nicht aber von authentifizierten Benutzern verwendet werden. Wenn Sie Gästen generell keinen Zugang zu Samba-Ressourcen gewähren möchten, verwenden Sie im `[global]`-Abschnitt die Einstellung `map to guest = never`.

Samba bietet gewöhnlichen Benutzern ohne `root`-Rechte die Möglichkeit, selbst Verzeichnisse, sogenannte User Shares, freizugeben. Die entsprechenden Konfigurationsdateien werden üblicherweise im Verzeichnis `/var/lib/samba/usershares` gespeichert, wobei jedes freigegebene Verzeichnis seine eigene Datei erhält. Die folgenden Zeilen geben dafür ein Beispiel:

User Shares

```
# Datei /var/lib/samba/usershares/bilder
path      = /home/kofler/Bilder
comment   =
guest_ok  = n
sharename = Bilder
```

Details der User-Share-Konfiguration werden im globalen Abschnitt von `smb.conf` durch diverse `usershare`-Anweisungen gesteuert. `usershare allow guests` erlaubt die Freigabe von User Shares zur Benutzung durch den `guest`-Account, also ohne Passwortschutz. Das erfordert die Angabe von `guest ok` oder `guest only` bei der Definition des Verzeichnisses. `usershare max shares` limitiert die Anzahl der User Shares.

```
# in /etc/samba/smb.conf
[global]
...
usershare allow guests = yes
usershare max shares   = 100
```

#### Papierkorb für Samba- Verzeichnisse

Wenn Dateien in Netzwerkverzeichnissen gelöscht werden, sind sie in der Regel für immer verloren. Die Papierkorb-Funktionen, die Linux, Windows oder macOS anbieten, gelten jeweils nur für lokale Dateien. Um den ungewollten Verlust von Dateien zu vermeiden, können Sie aber auch auf Samba-Ebene einen Papierkorb einrichten. In der einfachsten Konfiguration reicht dazu eine einzige Zeile in `smb.conf`:

```
[global]
vfs objects = recycle
```

Damit wird das Samba-VFS-Erweiterungsmodul `recycle` aktiviert, wobei VFS für *Virtual File System* steht. Gelöschte Dateien landen nun standardmäßig im Verzeichnis `.recycle` (relativ zum Share-Verzeichnis). Dieses Verzeichnis ist allerdings in den meisten Dateimanagern unsichtbar. Um den Vorgang transparenter zu machen, können Sie mit `recycle:repository` einen anderen Verzeichnisnamen für den Papierkorb angeben, z. B. so:

```
recycle:repository = Papierkorb
```

Wirklich gelöscht werden Dateien nun erst, wenn sie auch im Papierkorb gelöscht werden. Unter Umständen kann es zweckmäßig sein, auf dem Samba-Server ein Script auszuführen, das alte Dateien nach einer gewissen Zeit automatisch aus dem Papierkorb löscht. Damit das funktioniert, müssen Sie aber das Änderungsdatum beim Verschieben in den Papierkorb aktualisieren (Option `recycle:touch = Yes`).

Diverse weitere `recycle:xxx`-Optionen können Sie mit `man vfs_recycle` nachlesen.

## Netzwerkverzeichnisse in Gnome und KDE freigeben

Wenn Desktop-Anwender rasch und unkompliziert ein Verzeichnis per Samba freigeben möchten, haben sie in der Regel keine Lust, manuell Änderungen an `smb.conf` durchzuführen. Deswegen sehen die Dateimanager von Gnome und KDE Dialoge vor, um Verzeichnisse freizugeben. Hinter den Kulissen nutzen Nautilus (Gnome) und Dolphin (KDE) den User-Share-Mechanismus von Samba: Die Parameter für jedes freigegebene Verzeichnis werden dazu jeweils in einer eigenen Konfigurationsdatei im Verzeichnis `/var/lib/samba/usershares` gespeichert. Die Zugriffsrechte dieses Verzeichnisses sind so eingestellt, dass alle Benutzer, die einer bestimmten Gruppe angehören (bei Ubuntu `sambashare`), darin neue Dateien anlegen dürfen.

So viel gleich vorweg: Machen Sie sich keine allzu großen Hoffnungen, dass das Freigeben von Verzeichnissen unkompliziert funktioniert. Soweit überhaupt vorhanden, setzen die Gnome- und KDE-Tools eine Basiskonfiguration des Samba-Servers voraus, wobei Sie je nach Distribution die Firewall- und SELinux-Konfiguration nicht vergessen dürfen. Und auch dann bleiben die Freigabewerkzeuge Stückwerk, weil sie sich nicht um Samba-Passwörter kümmern.

Im Folgenden setze ich voraus, dass der Samba-Server installiert ist, also zumeist das Paket `samba`. Falls eine Firewall aktiv ist, müssen Sie dort die Samba-Ports öffnen.

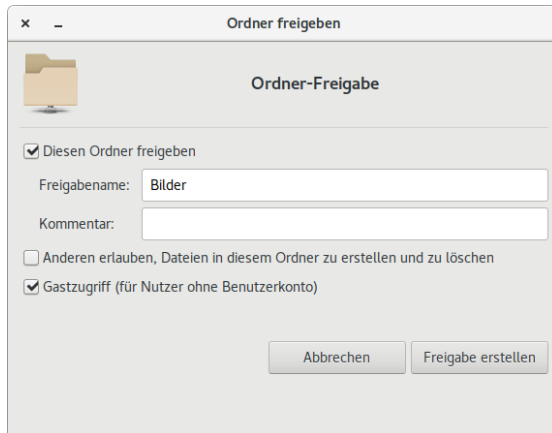
Vorbereitungen

Wenn Sie SUSE oder openSUSE verwenden, müssen Sie vor der Freigabe des ersten Verzeichnisses unter KDE oder Gnome eine Samba-Basiskonfiguration durchführen. Dazu können Sie das YaST-Modul NETZWERKDIENTSTE • SAMBA-SERVER verwenden. Als Server-Typ wählen Sie KEIN DOMAIN CONTROLLER. Die anderen Varianten sind nur dann von Interesse, wenn Samba auch Login-Daten für Windows-Accounts verwalten soll. Im Dialogblatt START wählen Sie nun die Option BEIM SYSTEMSTART, um den Samba-Server in Zukunft automatisch zu starten. Damit andere Rechner im lokalen Netzwerk auf die Netzwerkverzeichnisse zugreifen können, müssen Sie außerdem die Option FIREWALL-PORT ÖFFNEN aktivieren.

Sofern das Paket `nautilus-share` installiert sind, können Sie im Gnome-Dateimanager Verzeichnisse freigeben. Dazu klicken Sie das Verzeichnis mit der rechten Maustaste an und führen das Kontextmenükommando FREIGABEOPTIONEN aus. Damit gelangen Sie in den Dialog aus [Abbildung 28.1](#).

Nautilus/Gnome

CentOS-, Fedora- und RHEL-Anwender müssen auf `nautilus-share` leider verzichten, vermutlich weil diese Nautilus-Erweiterung nicht zur Red-Hat-spezifischen SELinux-Konfiguration kompatibel ist. In der Vergangenheit stand als Alternative das Tool `system-config-samba` aus dem gleichnamigen Paket zur Verfügung. Dieses relativ alte Programm wird aber nicht mehr gewartet und fehlt deswegen in aktuellen Versionen von CentOS, Fedora oder RHEL in den Paketquellen. Es führt also kein Weg an einer manuellen Bearbeitung der Konfigurationsdateien vorbei.



**Abbildung 28.1** Verzeichnisfreigabe unter Gnome (Debian/Ubuntu)

**Dolphin/KDE** Unter KDE können Sie in den Dateimanagern Dolphin und Konqueror im FREIGABE-Blatt des Eigenschaftsdialogs ein Netzwerkverzeichnis einrichten. Dazu muss das Paket `kdenetwork4-filesharing` installiert sein.

### Samba-Passwörter müssen manuell definiert werden!

Zuletzt bleibt noch die Frage zu klären, wer die freigegebenen Verzeichnisse benutzen darf. Wenn Sie die Option `GASTZUGRIFF` aktiviert haben, hat jeder ohne Anmeldung Zugriff auf Ihr Netzwerkverzeichnis. Wenn Sie die Option dagegen nicht aktivieren, muss sich der Benutzer mit Name und Passwort anmelden. Das funktioniert aber nur für Benutzer, für die es auf dem aktuellen Rechner einen Account gibt, und nur dann, wenn für diese Benutzer mit  `smbpasswd` ein Samba-Passwort definiert wurde. Weder KDE noch Gnome kümmert sich darum. Gegebenenfalls müssen Sie Ihr Samba-Passwort mit  `smbpasswd` in einem Terminal-Fenster festlegen.

### Freigaben in den Gnome-Einstellungen

In aktuellen Gnome-Versionen gibt es in den Einstellungen das Modul `FREIGABEN`. Damit können Sie die Verzeichnisse `Bilder`, `Musik`, `Öffentlich` und `Videos` innerhalb Ihres Home-Verzeichnisses freigeben. Diese Freigaben haben allerdings nichts mit Samba zu tun!

Vielmehr startet Gnome für Ihr öffentliches Verzeichnis eine lokale Instanz des Webservers Apache, der das betreffende Verzeichnis mithilfe des Protokolls WebDAV über die gewünschte Netzwerkschnittstelle anbietet. Für die restlichen Verzeichnisse wird eine Instanz des DLNA-Servers `rygel` gestartet. Der Zugriff auf die Dateien gelingt nur geeigneten Programmen, z. B. Linux-Dateimanagern (WebDAV) bzw. Media-Playern (DLNA).

## 28.5 Beispiel – Home- und Medien-Server

Dieser Abschnitt gibt ein einfaches Beispiel für die Konfiguration eines zentralen Samba-Servers. Ausgangspunkt ist ein computeraffiner Haushalt, in dem sich auf den drei Computern der Eltern bzw. der beiden Kinder immer mehr Daten anhäufen: Fotos, Audio- und Video-Dateien, Schuldokumente, die Buchhaltung etc. Die dezentrale Datenhaltung wirft einige Probleme auf:

- ▶ Es gibt keine ordentlichen Backups. Was passiert, wenn ein Notebook auf dem Weg zur Schule verloren geht oder das Zeitliche segnet?
- ▶ Es ist schwierig, auf gemeinsame Daten zuzugreifen. Oma soll zum nächsten Geburtstag ein Album der besten Familienfotos der letzten Jahre bekommen. Die digitalen Fotos sind aber über drei Rechner verteilt und in keiner Weise geordnet. Ähnliche Probleme gibt es auch bei Partys, wenn sich beim Abspielen von MP3-Dateien herausstellt, dass sich das gerade gewünschte Album auf einem anderen Rechner befindet.
- ▶ Der Datenaustausch zwischen den Rechnern ist umständlich und erfolgt zumeist mithilfe eines USB-Sticks.

Der Linux-begeisterte Sohn schlägt schließlich vor, diese Probleme durch einen zentralen Home- oder Medien-Server zu lösen. Der Server kann via WLAN in das Heimnetz integriert werden und bei Bedarf auch weitere Funktionen übernehmen.

An dieser Stelle ist nur die Samba-Konfiguration von Interesse: Jedes Familienmitglied bekommt ein eigenes Netzwerkverzeichnis, in dem es allein Daten schreiben und lesen darf. Die dort gespeicherten Daten sind also privat, wobei natürlich allen Familienmitgliedern klar sein muss, dass der Sohn als Administrator letztlich jede Datei lesen und verändern kann ...

Zum gemeinsamen Datenaustausch gibt es außerdem noch fünf weitere Verzeichnisse. Die Eltern dürfen auf `eltern` zugreifen, die Kinder auf `kinder` und alle Familienmitglieder auf die Verzeichnisse `familie`, `audio` und `fotos`. Natürlich wäre es möglich gewesen, die Verzeichnisse `audio` und `fotos` einfach als Unterverzeichnisse von `familie` einzurichten, die Definition eigener Netzwerkverzeichnisse macht die Anwendung aber ein wenig intuitiver. Bei Bedarf können natürlich beliebige weitere Benutzer und Verzeichnisse eingerichtet werden.

Als Benutzernamen verwende ich im Folgenden `mutter`, `vater`, `tochter`, `sohn`. In der Praxis werden Sie hier natürlich richtige Namen verwenden – aber darauf habe ich hier verzichtet, damit Sie nicht auch noch die Namen einer fiktiven Familie lernen müssen.

```
root# groupadd eltern
root# groupadd kinder
```

Linux-Benutzer  
und Gruppen  
einrichten

```

root# groupadd familie
root# useradd --create-home --groups eltern,familie vater
root# useradd --create-home --groups eltern,familie mutter
root# useradd --create-home --groups kinder,familie sohn
root# useradd --create-home --groups kinder,familie tochter

```

Da `useradd` ohne Passwort ausgeführt wurde, werden die neuen Benutzer automatisch gesperrt, d. h., es ist kein Login möglich. Das ist beabsichtigt: Es ist weder erforderlich noch zweckmäßig, dass sich die Familienmitglieder direkt auf dem Server anmelden.

Zusammen mit jedem Benutzer wird automatisch auch eine neue, gleichnamige Gruppe erzeugt, die als Standardgruppe für den Benutzer gilt. Außerdem werden den neuen Benutzern auch die Gruppen `familie` und `eltern` oder `kinder` zugeordnet. `vater` gehört somit den Gruppen `vater`, `eltern` und `familie` an, `mutter` den Gruppen `mutter`, `eltern` und `familie` etc. Wenn Sie einem Benutzer später eine weitere Gruppe zuordnen möchten, verwenden Sie am einfachsten das folgende Kommando:

```

root# usermod -a -G neuegruppe benutzer

```

#### Samba-Benutzer einrichten

Als Nächstes werden die Samba-Benutzer eingerichtet, diesmal jeweils mit einem Passwort:

```

root# smbpasswd -a vater
root# smbpasswd -a mutter
...

```

#### Verzeichnisse einrichten

Beim Einrichten der Verzeichnisse für die gemeinsamen Dateien ist es wichtig, Besitzer und Zugriffsrechte richtig einzustellen – sonst funktioniert später der Datenzugriff nicht. Das erste Kommando `chmod 770` bewirkt, dass nur Gruppenmitglieder das Verzeichnis lesen und verändern dürfen. Das zweite Kommando verbietet den Zugriff auf die Home-Verzeichnisse durch andere Benutzer.

```

root# mkdir /shared-data
root# mkdir /shared-data/{eltern,kinder,familie,audio,fotos}
root# cd /shared-data
root# chown :eltern eltern/
root# chown :kinder kinder/
root# chown :familie familie/ audio/ fotos/
root# chmod 770 *
root# chmod 770 /home/{vater,mutter,sohn,tochter}

```

Ein Vorteil des gemeinsamen Datei-Servers ist die Möglichkeit, zentrale Backups zu machen. Dabei müssen Sie lediglich die Verzeichnisse `/home` und `shared-data` sichern.

#### Samba-Konfiguration

Die folgenden Zeilen zeigen die Konfigurationsdatei `smb.conf`. Die Passwort-Synchronisierung und jeglicher Samba-Zugriff durch Gäste sind deaktiviert. Die Einstellungen für die diversen Verzeichnisse sollten nach der Lektüre von [Abschnitt 28.4](#), »Netzwerkverzeichnisse«, ohne weitere Erklärung verständlich sein.



```

# /etc/samba/smb.conf für einen Home-Server
[global]
    workgroup          = home
    server string      = %h server (Samba, Ubuntu)
    security           = user
    passdb backend     = tdbsam
    unix password sync = no
    invalid users      = root
    map to guest       = never
    log file           = /var/log/samba/log.%m
    max log size       = 1000

    syslog            = 0
    dns proxy         = no
    panic action       = /usr/share/samba/panic-action %d

[homes]
    browseable        = no
    writeable         = yes

[eltern]
    valid users       = @eltern
    path              = /shared-data/eltern
    writeable         = yes
    force group       = +eltern
    create mask       = 0660
    directory mask    = 0770

[kinder]
    valid users       = @kinder
    path              = /shared-data/kinder
    writeable         = yes
    force group       = +kinder
    create mask       = 0660
    directory mask    = 0770

[familie]
    valid users       = @familie
    path              = /shared-data/familie
    writeable         = yes
    force group       = +familie
    create mask       = 0660
    directory mask    = 0770

[fotos]
    valid users       = @familie
    path              = /shared-data/fotos
    ... wie bei [familie]

[audio]
    valid users       = @familie
    path              = /shared-data/audio
    ... wie bei [familie]

```

Die Konfiguration hat einen kleinen Schönheitsfehler: Alle Benutzer sehen *alle* Freigaben, auch die, die nicht für sie bestimmt sind und die sie nicht nutzen dürfen. Zum Beispiel sehen die Kinder das Verzeichnis `eltern`, die Eltern das Verzeichnis `kinder`. Eine tatsächliche Nutzung dieser Verzeichnisse scheitert wie geplant an den Zugriffsrechten, aber noch eleganter wäre es natürlich, wenn diese Verzeichnisse gar nicht erst sichtbar wären.

Samba bietet hierfür leider keine Konfigurationsmöglichkeiten. Sie können zwar einzelne Verzeichnisse durch `browseable = no` verstecken, aber dann sieht niemand die Verzeichnisse mehr, auch nicht die rechtmäßigen Nutzer. Die Verzeichnisse bleiben weiter benutzbar, allerdings muss der richtige Pfad manuell angegeben werden. Auch die Optionen `hide unreadable = yes` und `hide unwriteable = yes` helfen nicht weiter: Damit werden *innerhalb* eines Netzwerkverzeichnisses alle Dateien versteckt, die ein Benutzer nicht lesen bzw. nicht verändern kann. Das Netzwerkverzeichnis an sich bleibt aber weiter sichtbar.

## 28.6 Beispiel – Firmen-Server

Ein mittelständisches Unternehmen stellt Messgeräte her. Vor allem im Hinblick auf zentrale Backups, aber auch zur Vereinfachung des Datenaustauschs wird ein Samba-Server eingerichtet. Alle Mitarbeiter bekommen dort ein eigenes Verzeichnis. Außerdem gibt es mehrere gemeinsame Verzeichnisse, die [Tabelle 28.2](#) zusammenfasst. [Tabelle 28.3](#) zeigt, welchen Gruppen die Mitarbeiter je nach ihrer Position in der Firma angehören.

| Netzwerkverzeichnis | Zugriff (Gruppen)              |
|---------------------|--------------------------------|
| strategie           | leitung                        |
| buchhaltung         | buchhaltung, leitung           |
| entwicklung         | entwicklung, leitung           |
| vertrieb            | vertrieb, buchhaltung, leitung |

Tabelle 28.2 Datenaustausch im Firmen-Server

| Mitarbeiter              | Gruppenzugehörigkeit                        |
|--------------------------|---|
| Unternehmensleitung      | leitung, entwicklung, vertrieb, buchhaltung |
| Buchhaltung/Controlling  | buchhaltung, vertrieb                       |
| Entwickler               | entwicklung                                 |
| Vertriebsteam, Marketing | vertrieb                                    |

Tabelle 28.3 Gruppenzugehörigkeit der Mitarbeiter je nach Position

**Weniger ist mehr!**

Wenn Sie selbst ein Gruppen- und Verzeichnisschema erstellen müssen, sollte Einfachheit das oberste Gebot sein. Je mehr Gruppen und Verzeichnisse es gibt, desto unübersichtlicher wird das System, desto unhandlicher seine Anwendung und desto mühsamer die Wartung.

Auch wenn Sie den Samba-Server mit hochwertiger Hardware realisieren und Gigabit-Netzwerkverbindungen verwenden, ist der Zugriff auf Netzwerkverzeichnisse langsamer als auf lokale Dateien. Deswegen werden manche Benutzer große Dateien aus Performance-Gründen weiterhin lokal speichern und bearbeiten. In diesem Fall sollte es unbedingt ein automatisches Script geben, das alle lokalen Dateien einmal täglich mit dem privaten Netzwerkverzeichnis des Benutzers synchronisiert. Alle Vorteile eines zentralen Backup-Systems gehen verloren, wenn einzelne Benutzer ihre Dateien auf der lokalen Festplatte speichern!

Als Benutzernamen verwende ich im Folgenden der Einfachheit halber `chefN`, `entwicklerN` etc. In der Praxis werden Sie natürlich richtige Namen verwenden. Bis auf die Namen ähneln die Kommandos zum Einrichten der Gruppen und Benutzer sehr stark denen des vorigen Beispiels (Home-Server). Ein direkter Login der Mitarbeiter auf dem Server ist nicht vorgesehen, deswegen entfällt an dieser Stelle die Passwortangabe.

**Linux-Benutzer  
und Gruppen  
einrichten**

```
root# groupadd leitung
root# groupadd vertrieb
root# groupadd buchhaltung
root# groupadd entwicklung

root# useradd --create-home \
    --groups leitung,vertrieb,buchhaltung,entwicklung chefin1
root# useradd --create-home --groups vertrieb vertrieb1
root# useradd --create-home --groups vertrieb vertrieb2
root# useradd --create-home --groups entwicklung entwickler1
root# useradd --create-home --groups entwicklung entwicklerin2
root# useradd --create-home --groups buchhaltung,vertrieb buchhalter1
root# useradd --create-home --groups buchhaltung,vertrieb controller1
root# ...
```

Als Nächstes werden die Samba-Benutzer eingerichtet, diesmal jeweils mit einem Passwort:

**Samba-Benutzer  
einrichten**

```
root# smbpasswd -a chefin1
root# smbpasswd -a vertrieb1
...
```

**Verzeichnisse einrichten** Beim Einrichten der Verzeichnisse für die gemeinsamen Dateien ist es wichtig, Besitzer und Zugriffsrechte richtig einzustellen – sonst klappt später der Datenzugriff nicht. Das erste Kommando `chmod 770` bewirkt, dass nur Gruppenmitglieder das Verzeichnis lesen und verändern dürfen, und verhindert, dass andere Benutzer auf die Home-Verzeichnisse zugreifen dürfen.

```
root# mkdir /firmendaten
root# mkdir /firmendaten/{entwicklung,vertrieb,buchhaltung,strategie}
root# cd /firmendaten
root# chown :entwicklung entwicklung/
root# chown :vertrieb vertrieb/
root# chown :buchhaltung buchhaltung/
root# chown :leitung strategie/
root# chmod 770 /firmendaten/* /home/*
```

**Samba-Konfiguration** Der globale Abschnitt von `smb.conf` sieht mit Ausnahme der `workgroup`-Zeichenkette exakt genauso aus wie beim Home-Server-Beispiel (siehe [Abschnitt 28.5](#)). Auch die Definition der Netzwerkverzeichnisse ähnelt dem vorigen Beispiel stark. Der Unterschied besteht darin, dass Mitglieder mehrerer Gruppen auf die Verzeichnisse zugreifen dürfen. Damit der Datenaustausch reibungslos funktioniert, ist insbesondere `force group` entscheidend. Alle Benutzer, die auf ein Verzeichnis zugreifen dürfen, müssen Mitglied dieser Gruppe sein. Vergessen Sie in `smb.conf` das Plus-Zeichen vor dem Gruppennamen nicht!

```
# /etc/samba/smb.conf für einen Firmen-Server (Standalone-Konfiguration)
[global]
    ... wie im vorigen Beispiel (Home-Server)
[homes]
    browseable = no
    writeable = yes
[strategie]
    valid users = @leitung
    path = /firmendaten/strategie
    writeable = yes
    force group = +leitung
    create mask = 0660
    directory mask = 0770
[buchhaltung]
    valid users = @buchhaltung, @leitung
    path = /firmendaten/buchhaltung
    writeable = yes
    force group = +buchhaltung
    create mask = 0660
    directory mask = 0770
```

```
[entwicklung]
  valid users    = @entwicklung, @leitung
  path          = /firmendaten/entwicklung
  writeable     = yes
  force group   = +entwicklung
  create mask   = 0660
  directory mask = 0770
[vertrieb]
  valid users    = @vertrieb, @buchhaltung, @leitung
  path          = /firmendaten/vertrieb
  writeable     = yes
  force group   = +vertrieb
  create mask   = 0660
  directory mask = 0770
```

## 28.7 Client-Zugriff

Dieser Abschnitt beschäftigt sich mit der Frage, wie ein Client-PC auf die von Samba zur Verfügung gestellten Verzeichnisse zugreift. Eine wichtige Voraussetzung besteht darin, dass die TCP-Ports 135, 139 und 445 sowie die UDP-Ports 137 und 138 nicht durch eine Firewall blockiert werden.

### Linux-Clients

Bevor Sie unter Linux auf Windows- bzw. Samba-Netzwerkverzeichnisse zugreifen können, müssen Sie die Samba-Client-Tools installieren. Bei Ubuntu sind die erforderlichen Programme in `samba-common`, `smbclient` und `libsmbclient` verpackt, bei Fedora in `samba-common`, `samba-client` und `samba-libs`.

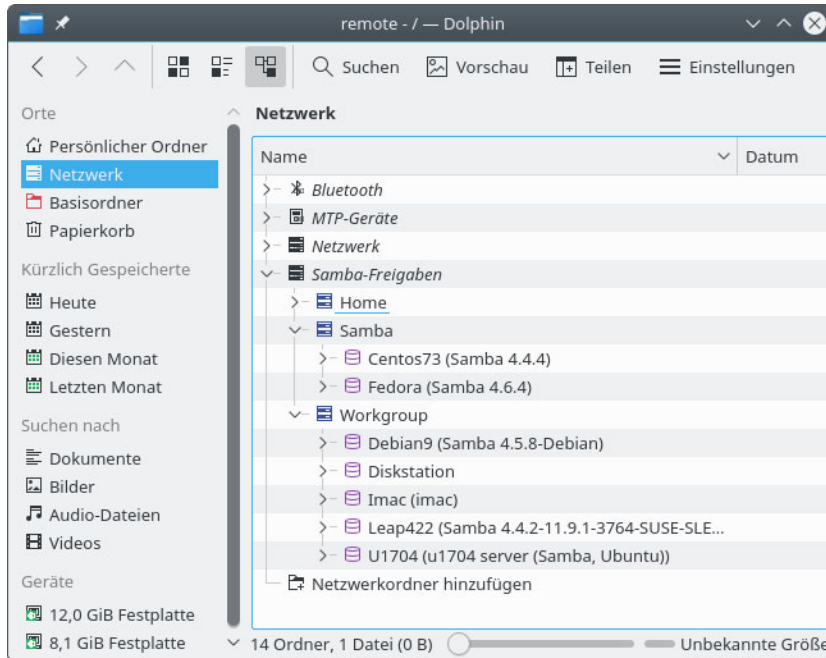
Am einfachsten verwenden Sie zum Zugriff auf ein Netzwerkverzeichnis den Dateimanager von Gnome oder KDE. In beiden Programmen gibt es einen Netzwerk-Browser, der im ersten Schritt alle verfügbaren Windows-Netzwerke anzeigt (siehe [Abbildung 28.2](#)). Ein paar Mausklicks und gegebenenfalls die Eingabe der Login-Daten (Benutzername und Passwort) führen in das gewünschte Verzeichnis.

KDE, Gnome

Bisweilen ist der Dateimanager nicht in der Lage, die Netzwerkverzeichnisse selbstständig zu finden. In diesem Fall müssen Sie deren Ort explizit mit `[Strg]+[L]` in der Adresszeile des Dateimanagers angeben. Dabei gilt die Schreibweise `smb://servername/verzeichnisname`.

KDE- und Gnome-Verweigerer, die dennoch grafische Unterstützung beim Zugriff auf Windows-Verzeichnisse suchen, sollten sich das Programm `pyNeighborhood` ansehen.

pyNeighborhood



**Abbildung 28.2** Besonders übersichtlich zeigt der KDE-Dateimanager Dolphin die im lokalen Netzwerk verfügbaren Samba-Server an.

## CIFS

Eine weitere Vorgehensweise besteht darin, Netzwerkverzeichnisse mit dem *Common Internet File System* (CIFS) direkt in den lokalen Verzeichnisbaum einzubinden. Das ist freilich nur sinnvoll, wenn anzunehmen ist, dass das Verzeichnis über längere Zeit verfügbar bleibt, also auf einem stabilen Server läuft.

CIFS ist der Nachfolger von SMBFS (*SMB File System*). CIFS-kompatible Samba-Server geben Unix/Linux-kompatible Informationen über die Zugriffsrechte von Dateien weiter, was bei SMBFS nicht möglich ist.

CIFS setzt voraus, dass die Samba-Client-Werkzeuge und die CIFS-Unterstützung für `mount` und insbesondere das Kommando `mount.cifs` zur Verfügung stehen. Bei Debian und Ubuntu muss dazu das Paket `cifs-utils` installiert werden. Bei Fedora und openSUSE ist das Paket standardmäßig installiert.

Um ein externes Verzeichnis einzubinden, geben Sie eines der beiden folgenden Kommandos an, je nachdem, ob die Windows-Freigabe auf der Basis von Benutzernamen erfolgt oder nicht:

```
root# mount -t cifs //jupiter/myshare /media/winshare
root# mount -t cifs -o username=name //jupiter/myshare /media/winshare
```

Damit wird das auf dem Rechner `jupiter` unter dem Namen `myshare` freigegebene Verzeichnis in das Linux-Dateisystem eingebunden. Die Daten stehen jetzt unter dem Linux-Verzeichnis `/media/winshare` dem Benutzer `root` zur Verfügung. Dieses Verzeichnis muss vor dem Ausführen von `mount` natürlich schon existieren. Bei der Ausführung des Kommandos werden Sie nach dem Passwort gefragt. Sie können das Passwort aber auch direkt angeben, was aber unsicher ist:

```
root# mount -t cifs -o username=name,password=xxxxxxx \
//jupiter/myshare /media/winshare
```

In der Praxis sind leider noch wesentlich mehr Optionen erforderlich. Ein typisches `mount`-Kommando sieht etwa so aus:

Weitere  
mount-Optionen

```
root# mount -t cifs -o username=name,uid=1000,gid=1000,icharset=utf8,nounix \
//jupiter/myshare /media/winshare
```

| Option                         | Bedeutung   |
|--------------------------------|---|
| <code>credentials=datei</code> | liest den Login-Namen und das Passwort aus einer Datei. |
| <code>dir_mod=n</code>         | setzt die Zugriffsbits für Verzeichnisse.               |
| <code>domain=name</code>       | bestimmt die Arbeitsgruppe oder Domäne.                 |
| <code>file_mod=n</code>        | setzt die Zugriffsbits für Dateien.                     |
| <code>gid=n</code>             | gibt an, welche Gruppe Zugriff auf die Dateien erhält.  |
| <code>icharset=utf8</code>     | verwendet den UTF8-Zeichensatz für Dateinamen.          |
| <code>nounix</code>            | deaktiviert die Unix-Extensions des CIFS-Protokolls.    |
| <code>password=xxx</code>      | gibt das Passwort an (unsicher!).                       |
| <code>uid=n</code>             | gibt an, wer Zugriff auf die Dateien erhält.            |
| <code>username=name</code>     | gibt den Benutzernamen an.                              |
| <code>sec=ntlm</code>          | stellt den Authentifizierungsmodus ein.                 |

**Tabelle 28.4** CIFS-mount-Optionen

Eine Zusammenfassung der wichtigsten `mount`-Optionen finden Sie in [Tabelle 28.4](#). Zum besseren Verständnis sind hier aber noch ein paar Erläuterungen erforderlich: Damit Sie das Netzwerkverzeichnis nicht nur als `root`, sondern auch als gewöhnlicher Benutzer lesen und schreiben können, geben Sie beim `mount`-Kommando mit `uid=n` und `gid=n` Ihre persönlichen Benutzer- und Gruppen-Identifikationsnummern an, die Sie mit dem Kommando `id` schnell ermitteln können.

Ergänzend dazu geben `dir_mode` und `file_mode` an, mit welchen Zugriffsrechten die Netzwerkverzeichnisse und Dateien sichtbar sein sollen. Die Zugriffsrechte werden in einer Oktalschreibweise angegeben, die in [Abschnitt 11.5](#), »Zugriffsrechte, Benutzer und Gruppenzugehörigkeit«, näher erläutert ist. Zweckmäßige Einstellungen sind

zumeist 0700 und 0600. Damit kann der mit `uid` genannte Benutzer alle Verzeichnisse bearbeiten und alle Dateien lesen und verändern. Andere Benutzer des Computers haben hingegen keinen Zugriff.

Ein wenig absurd mutet die Option `nounix` an: Der CIFS-Standard wurde nämlich um die sogenannten Unix Extensions erweitert, um eine bessere Kompatibilität mit Unix/Linux-Systemen zu erzielen. Bei manchen NAS-Geräten sind diese Extensions auch aktiv. In diesem Fall übernimmt `mount` die Benutzer- und Gruppeninformationen sowie die Zugriffsrechte vom NAS-Gerät und ignoriert die Optionen `uid`, `gid`, `dir_mode` und `file_mode`.

Zweckmäßig ist das aber nur, wenn diese Daten zwischen dem NAS-Gerät und den Clients abgestimmt sind. Weil das aber selten der Fall ist, muss der Unix-Kompatibilitätsmodus deaktiviert werden, damit `uid`, `gid`, `dir_mode` und `file_mode` wirksam bleiben. Mitunter können Sie mit `nounix` auch Inkompatibilitäten aus dem Weg gehen, die sich in der vagen Fehlermeldung *operation not supported* äußern.

`iocharset=utf8` vermeidet schließlich die falsche Darstellung von Dateinamen mit internationalen Zeichen.

**/etc/fstab** Um das Netzwerkverzeichnis immer automatisch in den Verzeichnisbaum einzubinden, fügen Sie `/etc/fstab` einen entsprechenden Eintrag hinzu, beispielsweise so:

```
# in /etc/fstab
//jupiter/myshare /media/winshare cifs username=u,password=p,... 0 0
```

Bei den meisten Distributionen werden alle in `/etc/fstab` genannten CIFS-Verzeichnisse während des Init-Prozesses eingebunden. Eine Ausnahme stellt SUSE dar: Hier ist für diesen Vorgang das Init-V-Skript `cifs` zuständig (auch bei SUSE-Versionen, die schon `systemd` verwenden!). Es muss explizit aktiviert werden:

```
root# inserv cifs
```

**Login-Daten auslagern** Aus Sicherheitsgründen ist die direkte Angabe des Passworts in `/etc/fstab` nicht empfehlenswert. Ein wenig besser ist es, diese Information in eine eigene Datei auszulagern, die nur `root` lesen kann. Richten Sie also die Datei `/etc/.winshare-pw` ein, die den Login-Namen, das Passwort und optional die Arbeitsgruppe (Domäne) für das Netzwerkverzeichnis enthält. Der Aufbau der Datei sieht so aus:

```
username=name
password=xxxx
domain=workgroup
```

Das folgende Kommando schränkt den Zugriff auf die Datei ein. Jetzt kann nur noch `root` die Datei lesen und verändern:

```
root# chmod 600 /etc/.winshare-pw
```



Anschließend fügen Sie dem `fstab`-Eintrag die Option `credentials` hinzu. Beim Einbinden des Verzeichnisses werden die Authentifizierungsdateien aus `.winshare-pw` gelesen:

```
# Ergänzung in /etc/fstab
//jupiter/myshare /media/winshare cifs credentials=/etc/.winshare-pw 0 0
```

### smbclient und smbtree

Freunde textorientierter Kommandos können Netzwerkverzeichnisse auch mit `smbclient` durchforschen. Das Kommando bietet zwar wenig Komfort, ist aber oft praktisch, um Samba-Problemen auf die Spur zu kommen.

`smbclient -L localhost` zeigt alle freigegebenen Ressourcen des lokalen Rechners an, listet alle sichtbaren Arbeitsgruppen des lokalen Netzwerks auf und gibt an, welcher Rechner in der jeweiligen Gruppe als Master fungiert. Die Passwortabfrage beantworten Sie bei passwortfreien Ressourcen einfach mit `[↵]`. Falls auf dem lokalen Rechner kein Samba-Server läuft, geben Sie statt `localhost` den Rechnernamen an.

Wenn `smbclient` eine Login-Fehlermeldung liefert (*access denied*), stimmen die Benutzer- oder Workgroup-Namen Ihres Linux-Rechners zumeist nicht mit denen des Windows-Rechners oder Samba-Servers überein. Die einfachste Lösung besteht darin, diese Informationen als zusätzliche Parameter an `smbclient` zu übergeben:

```
user$ smbclient -U benutzername -W workgroupname -L jupiter
```

Sie können `smbclient` auch interaktiv zur Übertragung von Dateien einsetzen. Dazu stellen Sie zuerst eine Verbindung zum Windows-Rechner oder Samba-Server für das freigegebene Verzeichnis her. Das Verzeichnis müssen Sie in der Windows-typischen Schreibweise `\\servername\verzeichnisname` angeben. Damit die `\`-Zeichen nicht von der Shell verarbeitet werden, müssen sie verdoppelt werden.

Anschließend können Sie wie beim Kommando `ftp` Verzeichnisse mit `ls` ansehen, mit `cd` wechseln, mit `get` Dateien auf den lokalen Rechner übertragen (*download*) und mit `put` Dateien auf dem externen Rechner speichern (*upload*). Einen Überblick über die wichtigsten Kommandos bekommen Sie mit `help`. Eine ausführliche Beschreibung der Kommandos gibt man `smbclient`.

```
user$ smbclient -U name -W wgname \\jupiter\myshare
Password: xxxxxx
smb: > ls
.                D          0 Sep  7 17:38:02 2017
..               D          0 Sep  7 17:38:02 2017
data             D          0 Apr  5 18:17:11 2017
file.xy         AR          226 Dec 14  8:33:38 2016
```

### Debugging mit »smbclient«

Zur Fehlersuche können Sie `smbtree` mit der Option `-d10` ausführen. Sie erhalten dann alle möglichen Debugging-Ausgaben. Mitunter scheitert der Samba-Zugriff daran, dass es im Netzwerk keinen Computer bzw. kein Gerät gibt, das als Master-Browser agiert. Abhilfe kann dann entweder ein echter Windows-Rechner schaffen oder natürlich ein Linux-Rechner, auf dem Samba läuft.

**smbtree** Das Kommando `smbtree` liefert eine baumförmige Liste aller im Netzwerk zu findenden Windows- und Samba-Server inklusive aller Objekte, die von diesen Servern freigegeben wurden. Normalerweise verwendet `smbtree` den aktuellen Benutzernamen und fragt nach einem dazugehörigen Passwort. Mit `-user=name%password` können Sie diese Daten beim Aufruf des Kommandos einstellen. Um Ressourcen zu finden, die ohne Passwort zugänglich sind, verwenden Sie die Option `-N`.

```
root# smbtree
Enter kofler's password: *****
WORKGROUP
  \\U1704                                u1704 server (Samba, Ubuntu)
    \\U1704\Brother-HL-4140CN-series      AirPrint Brother-HL-4140CN-series @ pi3
    \\U1704\IPC$                          IPC Service (u1704 server (Samba, Ubuntu))
  \\LEAP422                               Samba 4.4.2-11.9.1-3764-SUSE-SLE_12-x86_64
    \\LEAP422\groups                       All groups
    \\LEAP422\users                         All users
  \\IMAC                                   imac
    \\IMAC\kofler
  \\DISKSTATION
    \\DISKSTATION\photo                   System default shared folder
    \\DISKSTATION\music                   iTunes und Co
  \\DEBIAN9                               Samba 4.5.8-Debian
SAMBA
  \\FEDORA                                Samba 4.6.4
  \\CENTOS73                              Samba 4.4.4
```

### Windows und macOS

**Windows mit LLTD** Windows-Rechner haben mitunter Probleme, Samba-Server oder auch ältere Windows-Server im lokalen Netzwerk zu erkennen. In solchen Fällen sollten Sie in der Adresszeile des Windows Explorers den Rechner- und Verzeichnisnamen in der Form `\\hostname\sharename` eingeben.

**macOS** macOS ist grundsätzlich SMB-kompatibel und tut sich mit Samba-Freigaben erstaunlicherweise oft leichter als Windows. Jeder Samba-Server wird nach kurzer Zeit automatisch in der Seitenleiste des Finders angezeigt. macOS versucht einen Verbindungsaufbau als Gast. Gelingt dieser nicht, geben Sie im Login-Dialog den entsprechenden Samba-Benutzernamen und das Passwort an.

# Kapitel 29

## NFS und AFP

Das im vorigen Kapitel vorgestellte Server-Message-Block-Protokoll bzw. das darauf basierende Programm Samba hat sich zum großen gemeinsamen Nenner beim Dateiaustausch zwischen allen Betriebssystemen entwickelt. Daneben gibt es aber noch andere Verfahren. Im Mittelpunkt dieses Kapitels stehen das »klassische« Unix-Protokoll NFS (Network File System) sowie das unter macOS bevorzugte AFP (Apple Filing Protocol).

### 29.1 NFS

Das *Network File System* (NFS) ermöglicht es, lokale Verzeichnisse anderen Rechnern im lokalen Netzwerk zur Verfügung zu stellen. Anders als bei SMB (Samba) wird das Netzwerkverzeichnis auf dem Client-Rechner durch `mount` bzw. durch eine entsprechende Zeile in `/etc/fstab` direkt in das Dateisystem eingebunden. Sie können ein NFS-Verzeichnis also nicht komfortabel im Dateimanager auswählen und mit ein paar Mausklicks darauf zugreifen. NFS ist vielmehr für eine selten wechselnde Netzwerkkonfiguration gedacht, wo die Client-Rechner ständigen Zugriff auf den NFS-Server benötigen.

Die Basisfunktionen für NFS werden direkt vom Kernel zur Verfügung gestellt, um auf diese Weise eine optimale Geschwindigkeit zu erzielen. Alternativ gibt es auch einen User-Space-NFS-Server, der aber kaum im Einsatz ist und auf den ich in diesem Kapitel nicht eingehe.

Die in den Kernel integrierten NFS-Funktionen unterstützen die NFS-Versionen 3 und 4. NFS 4 gilt mittlerweile als vollständig ausgereift und ist im Regelfall vorzuziehen. Nur in Sonderfällen, etwa wenn Sie es mit sehr alten Client-Rechnern zu tun haben, die NFS 4 nicht unterstützen, sollten Sie den Einsatz von NFS 3 in Betracht ziehen. In diesem Buch gehe ich allerdings ausschließlich auf NFS 4 ein.

## Server-Konfiguration

Der Linux-Kernel unterstützt NFS 4 standardmäßig. Gegebenenfalls müssen Sie zur Nutzung von NFS aber noch einige Pakete installieren:

```
root# apt install nfs-common nfs-kernel-server    (Debian, Ubuntu)
root# dnf/yum install nfs-utils                  (CentOS, Fedora, RHEL)
root# zypper install nfs-kernel-server           (SUSE)
```

Die darin enthaltenen Programme und Scripts kümmern sich um den automatischen Start der erforderlichen Netzwerkdienste. Eine entscheidende Voraussetzung für den Betrieb eines NFS-4-Servers besteht darin, dass auch der Dienst `rpc.idmapd` läuft. Dieses Programm stellt die Zuordnung zwischen NFS-Benutzernamen und UIDs/GIDs her.

### Netzwerk- verzeichnisse vorbereiten

Alle Verzeichnisse, die per NFS freigegeben bzw. »exportiert« werden, müssen einem Wurzelverzeichnis untergeordnet werden, das als Pseudo-Dateisystem dient. Am einfachsten ist das anhand eines Beispiels zu verstehen. Nehmen wir an, Sie wollen die bereits vorhandenen Verzeichnisse `/data/audio`, `/data/fotos` und `/iso-images` exportieren. Dazu erzeugen Sie vier neue Verzeichnisse, wobei `/nfsexport` als Wurzelverzeichnis dient. Der Name dieses Verzeichnisses ist willkürlich. Die Verzeichnisse `/nfsexport/audio`, `/data/fotos` und `/iso` sind leer. Sie dienen nur als Einhängepunkt (*mount point*).

```
root# mkdir /nfsexport
root# mkdir /nfsexport/audio
root# mkdir /nfsexport/fotos
root# mkdir /nfsexport/iso
```

Nun binden Sie `/data/` und `/data/fotos` als neue `nfsexports`-Unterverzeichnisse ein. Damit ist der Inhalt von `/data/audio` nun auch unter `/nfsexport/audio` sichtbar; analog sehen Sie `/data/fotos` unter `/nfsexport/fotos`:

```
root# mount -t none -o bind /data/audio/ /nfsexport/audio/
root# mount -t none -o bind /data/fotos/ /nfsexport/fotos/
root# mount -t none -o bind /iso-images/ /nfsexport/iso/
```

Damit das Einbinden des NFS-Verzeichnisses in Zukunft automatisch erfolgt, fügen Sie in `/etc/fstab` auf dem Server die folgenden drei Zeilen hinzu:

```
# /etc/fstab
...
/data/audio    /nfsexport/audio none bind 0 0
/data/fotos    /nfsexport/fotos none bind 0 0
/iso-images    /nfsexport/iso    none bind 0 0
```

`/etc/exports` ist die zentrale Konfigurationsdatei für den NFS-Server. Diese Datei steuert, welcher Rechner auf welche Verzeichnisse wie zugreifen darf. Die Rechner können wahlweise durch IP-Nummern oder durch Namen angegeben werden. Subnetze können in der Prefix-Notation angegeben werden, z. B. `10.0.0.0/24`. Auch IPv6-Adressen sind zulässig. Rechnernamen dürfen außerdem das Jokerzeichen `*` enthalten (z. B. `*.sol`), IP-Adressen aber nicht!

Achten Sie darauf, dass Sie zwischen den IP-Adressen bzw. Hostnamen und den Optionen kein Leerzeichen eingeben! Wenn Sie ein Verzeichnis ohne Einschränkungen für alle Rechner freigeben möchten, die eine Verbindung zum NFS-Server herstellen können, geben Sie einfach das Zeichen `*` an.

Die folgende Beispieldatei gibt an, dass alle Clients mit IP-Nummern im Netz `10.0.0.*` oder mit dem Namen `*.lan` auf die Verzeichnisse `/nfsexport/audio` und `/nfsexport/fotos` zugreifen dürfen. Außerdem dürfen alle Rechner im Netzwerk ohne Einschränkungen bezüglich IP-Adresse oder Hostname auf `/nfsexport/iso` zugreifen. In den Verzeichnissen `audio` und `iso` sind nur Lesezugriffe erlaubt. Die langen Exportdefinitionen wurden hier durch `\` über zwei Zeilen verteilt.

```
# Datei /etc/exports für NFS 4
/nfsexport      10.0.0.0/24(rw,async,no_subtree_check,fsid=0,crossmnt) \
                *.lan(rw,async,no_subtree_check,fsid=0,crossmnt)
/nfsexport/audio 10.0.0.0/24(ro,async,no_subtree_check) \
                *.lan(ro,async,no_subtree_check)
/nfsexport/fotos 10.0.0.0/24(rw,async,no_subtree_check) \
                *.lan(rw,async,no_subtree_check)
/nfsexport/iso   *(ro,async,no_subtree_check)
```

Wenn Sie den NFS-Server in einem IPv6-Netz verwenden möchten, geben Sie in `/etc/exports` einfach das entsprechende IPv6-Netz an, z. B. `2001:7b8:2ff:8471::/64`. Die IPv6-Adressen dürfen nicht in eckige Klammern gesetzt werden.

Die Syntax von `/etc/exports` geht aus dem obigen Listing hervor. Dem Verzeichnis und den Hostnamen bzw. IP-Adressen folgen in Klammern diverse NFS-Optionen, von denen die wichtigsten im Folgenden kurz erläutert werden. Einige weitere Optionen beschreibt man `exports`.

- ▶ `ro` (read-only) bzw. `rw` (read-write) geben an, ob nur ein Lese- oder auch ein Schreibzugriff erlaubt ist.
- ▶ `sync` bzw. `async` bestimmen den Zeitpunkt, zu dem der NFS-Server die Änderungen von Dateien bestätigt. Standardmäßig gilt `sync`. Das bewirkt, dass eine Bestätigung erst erfolgt, wenn die Datei tatsächlich gespeichert wurde. Viel effizienter, aber weniger sicher ist `async`. Der Geschwindigkeitsunterschied zwischen `sync` und `async` ist bei Schreibzugriffen dramatisch (bis zu Faktor 10), weswegen `async` in der Praxis häufig zum Einsatz kommt.

- ▶ `no_subtree_check` bzw. `subtree_check` geben an, ob der NFS-Server den Subtree-Test durchführen soll. Dazu kurz einige Hintergrundinformationen: Wenn ein Verzeichnis eines Dateisystems (nicht aber ein gesamtes Dateisystem) per NFS exportiert wird, stellt der NFS-Server durch den Subtree-Test fest, ob sich die Datei innerhalb des exportierten Verzeichnisses befindet. Der NFS-Server gibt dann Informationen über den tatsächlichen Ort der Datei an den Client weiter. Wird die Datei später auf dem Server umbenannt, führt das oft zu Problemen auf dem Client.

Aus diesem Grund ist der Subtree-Test in aktuellen NFS-Server-Versionen standardmäßig deaktiviert. Die Option `no_subtree_check` sollte aber dennoch angegeben werden, um eine diesbezügliche Warnung des Servers beim Start zu verhindern.

Wenn Sie möchten, können Sie den Subtree-Test durch `subtree_check` explizit aktivieren. `man exports` empfiehlt dies vor allem für Verzeichnisse, in denen selten Dateien umbenannt werden und die im Read-Only-Modus exportiert werden.

- ▶ `root` darf zwar wie jeder andere Benutzer NFS nutzen, hat aber aus Sicherheitsgründen in den importierten Verzeichnissen nur die Rechte des Benutzers `nobody` (UID=65534 und GID=65534). Wenn Sie `root` die üblichen Rechte geben möchten, müssen Sie in `/etc/exports` die Option `no_root_squash` angeben.
- ▶ Das NFS-4-Wurzelverzeichnis muss durch die Option `fsid=0` gekennzeichnet werden. Es darf nur ein Wurzelverzeichnis geben! Es ist mit NFS 4 nicht möglich, Verzeichnisse zu exportieren, die sich außerhalb des Wurzelverzeichnisses befinden.
- ▶ Die `crossmnt`-Option wird ebenfalls nur beim Wurzelverzeichnis angegeben. Sie bewirkt, dass beim Einbinden von Unterverzeichnissen deren Inhalt bei den Clients auch dann sichtbar ist, wenn das Wurzelverzeichnis auf dem Client nicht eingebunden ist. Statt der `crossmnt`-Option beim Wurzelverzeichnis können Sie auch die `nohide`-Option bei allen Unterverzeichnissen angeben – Sie erzielen damit denselben Effekt.

**exportfs -a** Wenn der NFS-Server bereits läuft, müssen Sie nach jeder Änderung in `/etc/exports` das Kommando `exportfs -a` ausführen. Es stellt sicher, dass der NFS-Server die geänderten neuen Einträge berücksichtigt.

```
root# exportfs -a
```

**Distributions-spezifische Einstellungen** Außer in standardisierten Konfigurationsdateien können Sie je nach Distribution zusätzlich individuelle Einstellungen vornehmen:

Debian, Ubuntu: `/etc/defaults/nfs-common`, `/etc/defaults/nfs-kernel-server`  
 SUSE: `/etc/sysconfig/nfs`

Debian und Ubuntu starten den NFS-Server nach der Installation standardmäßig. Bei Fedora, Red Hat und SUSE müssen Sie wie bei anderen Systemdiensten durch die folgenden Kommandos nachhelfen (siehe auch [Abschnitt 12.5](#), »Systemprozesse (Dämonen)«):

```
root# systemctl start nfs-server      (CentOS, Fedora, RHEL)
root# systemctl enable nfs-server
root# systemctl start rpcbind        (SUSE)
root# systemctl enable rpcbind
root# systemctl start nfsserver
root# systemctl enable nfsserver
```

Start

NFS 4 verwendet standardmäßig Benutzer- und Gruppennamen zum ID-Mapping. Eine Datei, die auf dem NFS-Server dem Benutzer hofer gehört, darf auf dem NFS-Client-Rechner ebenfalls vom Benutzer hofer gelesen und verändert werden. Server-Benutzer und -Gruppen, die auf dem Client nicht existieren, werden dort dem Benutzer und der Gruppe nobody zugeordnet.

UID- und  
GID-Mapping

Grundsätzlich ist das ID-Mapping von NFS 4 zwar wesentlich intelligenter als jenes von NFS 3, wo einzig die UID- und GID-Nummern als Grundlage verwendet werden. Dennoch ist das Mapping auch unter NFS 4 nicht frei von Tücken: Sie müssen unbedingt sicherstellen, dass Benutzer im gesamten Netzwerk auf allen Rechnern exakt dieselben Account-Namen haben!

Für das UID- und GID-Mapping ist der bereits erwähnte Dämon `rpc.idmapd` verantwortlich. Dessen Konfiguration erfolgt durch `/etc/idmapd.conf`. Für die in diesem Kapitel beschriebene Minimalkonfiguration von NFS 4 können Sie die Datei so lassen, wie sie von Ihrer Distribution vorgegeben ist.

Bei NFS 4 erfolgt die gesamte Kommunikation über den TCP-Port 2049. Wenn Ihr Server durch eine Firewall abgesichert ist, müssen Sie diesen Port im lokalen Netzwerk freigeben. Unter SUSE erledigen Sie das am besten in YaST. Auf CentOS-, Fedora- und RHEL-Systemen stellen Sie zuerst fest, welche Firewall-Zone der Netzwerkschnittstelle für das LAN zugeordnet ist, und öffnen für diese Zone dann den NFS-Dienst:

Firewall

```
root# firewall-cmd --get-zone-of-interface=enp0s3  (aktive Zone herausfinden)
FedoraWorkstation
root# firewall-cmd --permanent --zone=FedoraWorkstation --add-service=nfs
root# firewall-cmd --reload
```

Hintergrundwissen sowie weitere Strategien zur Firewall-Konfiguration sind in [Kapitel 37](#), »Firewalls«, zusammengefasst.

In diesem Abschnitt habe ich die Server-Konfiguration ohne ein Authentifikationssystem beschrieben – also den einfachsten Weg, um NFS 4 in Betrieb zu nehmen. In großen Netzwerken mit vielen Benutzern werden Sie NFS 4 zumeist mit LDAP und

NFS 4 mit LDAP  
und Kerberos

Kerberos verbinden wollen. Auf diese ziemlich komplexe Konfiguration kann ich hier aus Platzgründen nicht eingehen. Sie finden entsprechende Anleitungen im Internet, beispielsweise unter:

<https://wiki.debian.org/nfs4-kerberos-ldap>

<https://help.ubuntu.com/community/NFSv4Howto>

## Client-Konfiguration

Damit NFS 4 auf einem Client-Rechner genutzt werden kann, muss je nach Distribution das Paket `nfs-common` (Debian, Ubuntu), `nfs-utils` (CentOS, Fedora, RHEL) oder `nfs-client` (SUSE) installiert sein. Außerdem muss der Dämon `rpc.idmapd` laufen.

**mount und  
umount**

Damit Sie das Netzwerkverzeichnis nutzen können, müssen Sie es mit `mount` in den Verzeichnisbaum integrieren. Die folgenden Kommandos integrieren den gesamten `nfsexport`-Verzeichnisbaum an der Stelle `/media/nfsdata` in das lokale Dateisystem. Dabei müssen Sie `jupiter` durch den Hostnamen des NFS-Servers ersetzen. Beachten Sie, dass Sie das NFS-Wurzelverzeichnis einfach mit `/` adressieren müssen, nicht mit `/nfsexport`!

```
root# mkdir /media/nfsdata
root# mount -t nfs4 jupiter:/ /media/nfsdata
root# ls /media/nfsdata
audio  fotos  iso
```

Alternativ können Sie auch nur ein Teilverzeichnis importieren:

```
root# mkdir /media/fotos
root# mount -t nfs4 jupiter:/fotos /media/fotos
```

Mit `umount` wird das NFS-Verzeichnis wieder aus dem lokalen Dateisystem entfernt. Wenn die Netzwerkverbindung gerade unterbrochen ist, sollten Sie `umount` mit der Option `-f` ausführen. Sonst müssen Sie sehr lange warten, bis `umount` ausgeführt wird!

```
root# umount /media/fotos
root# umount /media/nfsdata
```

### Tipp

Unter openSUSE verwenden Sie zum Einrichten von NFS-Verzeichnissen am besten das YaST-Modul `NETZWERKDIENTE • NFS-CLIENT`.

**/etc/fstab**

Um NFS-Verzeichnisse beim Rechnerstart automatisch in das Dateisystem zu integrieren, ergänzen Sie `/etc/fstab` um eine Zeile nach dem folgenden Muster. In der vierten Spalte können Sie die NFS-spezifische Option `bg` verwenden. Sie erreichen damit, dass `mount` im Hintergrund versucht, das Netzwerkverzeichnis einzubinden, wenn dieses nicht sofort zur Verfügung steht. Das ist vor allem beim Einbinden von Netzwerkverzeichnissen während des Rechnerstarts praktisch.



```
# Ergänzung in /etc/fstab
jupiter:/fotos /media/fotos nfs4 bg 0 0
```

Wenn es in einem großen Netzwerk viele NFS-Verzeichnisse gibt, ist es selten zweckmäßig, einfach alle per `/etc/fstab` zu aktivieren. Das kostet Zeit und Ressourcen, auch wenn die meisten Verzeichnisse dann gar nicht verwendet werden. Viel besser ist es, die Verzeichnisse erst bei der ersten Benutzung automatisch in den Verzeichnisbaum einzubinden. Diese Aufgabe übernimmt bei vielen Distributionen das Paket `autofs` bzw. `autofs4`.

**automount/  
autofs**

## Fehlersuche

Wenn das `mount`-Kommando auf dem Client scheitert, sollten Sie zur Fehlersuche die folgenden Punkte abarbeiten:

- ▶ Die Verbindung zwischen Server und Client darf nicht durch eine Firewall blockiert werden. Relevant für NFS 4 ist der TCP-Port 2049.
- ▶ Der Dämon `rpc.idmapd` muss sowohl auf dem Server als auch auf dem Client laufen. Überzeugen Sie sich davon mit `ps ax | grep idmapd`.
- ▶ Auf dem Server muss der NFS-Server laufen. Das stellen Sie mit `rpcinfo -p` fest. Die folgenden Zeilen beweisen, dass der NFS-Server für die NFS-Versionen 2, 3 und 4 läuft:

```
root# rpcinfo -p | grep nfs
  program vers proto  port  service
    100003   3   tcp   2049  nfs
    100003   4   tcp   2049  nfs
    100227   3   tcp   2049  nfs_acl
    100003   3   udp   2049  nfs
    100003   4   udp   2049  nfs
    100227   3   udp   2049  nfs_acl
    ...
```

- ▶ Überprüfen Sie mit `showmount -e`, welche Verzeichnisse für NFS freigegeben sind:

```
user@nfserver$ showmount -e
/nfsexport/iso *
/nfsexport/fotos *.lan,10.0.0.0/24
/nfsexport/audio *.lan,10.0.0.0/24
/nfsexport      *.lan,10.0.0.0/24
```

Sie können `showmount` auch auf dem Client-Rechner ausführen, müssen dann aber den Hostnamen des NFS-Servers angeben:

```
user$client# showmount -e <nfserver>
```

- ▶ Stellen Sie sicher, dass sich Client und Server im selben Netzwerksegment befinden.

## 29.2 Apple Filing Protocol

Das *Apple Filing Protocol* (kurz AFP) ist ein Netzwerkprotokoll zum Zugriff auf Dateien und Verzeichnisse über das Netzwerk. AFP kommt standardmäßig zum Einsatz, wenn mehrere Apple-Computer in einem lokalen Netzwerk Dateien austauschen möchten.

Vielleicht fragen Sie sich: »Wozu AFP, macOS unterstützt doch auch SMB?« Prinzipiell stimmt das, und seit Herbst 2013 ist SMB sogar zum bevorzugten Protokoll für Netzwerkverzeichnisse geworden. Allerdings ist AFP nach wie vor das einzige Protokoll, um Backups mit der *Time Machine* in Netzwerken durchzuführen.

**Netatalk** Für die Nachbildung des Apple Filing Protocols ist das Open-Source-Programm Netatalk verantwortlich. Entsprechende Pakete stehen in allen gängigen Distributionen zur Verfügung. Netatalk kommt auch in nahezu allen NAS-Geräten (Network Attached Storage) zum Einsatz.

In diesem Abschnitt erfahren Sie, wie Sie Netatalk so einrichten, dass macOS-Rechner im lokalen Netzwerk Dateien lesen und speichern und Ihren Linux-Server als Backup-Medium verwenden können. Netatalk bietet darüber hinaus unzählige Optionen und Konfigurationsvarianten, die nur in Sonderfällen relevant sind und auf die ich deswegen nicht eingehe. Werfen Sie gegebenenfalls einen Blick in das eher dürftige Online-Handbuch auf der Netatalk-Website:

<http://netatalk.sourceforge.net>

**Versionen** Dieses Kapitel bezieht sich auf die Netatalk-Versionen 3.n. Version 3.0 wurde bereits im Juli 2012 vorgestellt, hat in den Linux-Distributionen aber leider keine große Verbreitung gefunden: Im Frühjahr 2017 konnte lediglich openSUSE 42.2 (Leap) mit einigermaßen aktuellen Netatalk-Paketen aufwarten.

Ubuntu und Debian enthalten nur die vollkommen veraltete Netatalk-Version 2.2.5. In Fedora, CentOS bzw. RHEL 7 fehlt Netatalk gleich ganz, weil es niemanden mehr gibt, der die Pakete wartet.

Wie Sie Netatalk 3 unter CentOS, Debian, Fedora, RHEL und Ubuntu installieren, können Sie auf den folgenden Seiten nachlesen:

- ▶ [http://netatalk.sourceforge.net/wiki/index.php/Install\\_Netatalk\\_3.1.11\\_on\\_Debian\\_8\\_Jessie](http://netatalk.sourceforge.net/wiki/index.php/Install_Netatalk_3.1.11_on_Debian_8_Jessie)
- ▶ [http://netatalk.sourceforge.net/wiki/index.php/Netatalk\\_3.1.11\\_SRPM\\_for\\_Fedora\\_and\\_CentOS](http://netatalk.sourceforge.net/wiki/index.php/Netatalk_3.1.11_SRPM_for_Fedora_and_CentOS)
- ▶ [http://netatalk.sourceforge.net/wiki/index.php/Install\\_Netatalk\\_3.1.11\\_on\\_Ubuntu\\_16.04\\_Xenial](http://netatalk.sourceforge.net/wiki/index.php/Install_Netatalk_3.1.11_on_Ubuntu_16.04_Xenial)

- ▶ <https://daniel-lange.com/archives/102-Apple-Timemachine-backups-on-Debian-8-Jessie.html>

Vielleicht fragen Sie sich, warum ich mich hier derart ausführlich mit Versionsnummern aufhalte. Das hat damit zu tun, dass sich die Netatalk-Konfiguration mit Version 3.0 grundlegend verändert (und vereinfacht) hat. Die hier präsentierte Anleitung ist daher für die alten Netatalk-Versionen aus den Debian- und Ubuntu-Paketquellen nicht geeignet!

Zur Installation unter Fedora laden Sie zuerst von der oben erwähnten Netatalk-Webseite für Fedora das `*.src.rpm`-Paket mit dem Quellcode herunter. Daraus kompilieren Sie das Programm, verpacken es in ein RPM-Paket und installieren dieses. Diverse Warnungen beim Kommando `rpm -ivh` können Sie ignorieren.

Installation unter  
Fedora

```
user$ cd
user$ rpm -ivh netatalk-n.n.src.rpm
Warnung: Gruppe hat existiert nicht - Root wird benutzt
...
user$ sudo dnf install rpm-build gcc make avahi-devel bison cracklib-devel \
    dbus-devel dbus-glib-devel docbook-style-xsl flex krb5-devel \
    libacl-devel libattr-devel libevent-devel libgcrypt-devel \
    libtdb-devel mysql-devel openldap-devel openssl-devel \
    pam-devel quota-devel tcp_wrappers-devel tracker-devel
user$ cd rpmbuild/SPECS
user$ rpmbuild -bb netatalk.spec
user$ cd ../RPMS/x86_64
user$ sudo dnf install netatalk-n.n.x86_64.rpm
```

Die Konfiguration für Netatalk befindet sich in der Datei `/etc/netatalk/afp.conf`. Die Syntax ist ähnlich wie bei Samba: Die Datei wird durch `[name]` in Abschnitte unterteilt, Einstellungen erfolgen in der Form `parameter = wert`. Es gibt zwei reservierte Abschnittsnamen: `[Global]` für Grundeinstellungen und `[Homes]` für Heimatverzeichnisse. Alle weiteren Abschnitte bezeichnen »Volumes«, also Netzwerkverzeichnisse. Die folgenden Zeilen veranschaulichen die Syntax:

Konfiguration

```
[Global]
hostname = fedora

[Homes]
basedir regex = /home

[test]
path = /data/for-macos

[tm]
path = /data/tm
time machine = yes
```

```
valid users    = kofler
vol size limit = 500000
# in MB, 500.000 MB = 500 GB
```

Die folgenden Absätze erläutern die wichtigen Schlüsselwörter. `hostname` gibt an, unter welchem Namen der Rechner für macOS sichtbar sein soll. Eine Einstellung ist nur notwendig, wenn dieser Name vom Linux-Hostnamen abweichen soll.

Die Einstellung `basedir regex = /home` besagt, dass alle `/home/name`-Verzeichnisse als Heimatverzeichnisse freigegeben werden sollen. Der Zugriff auf diese Verzeichnisse erfordert einen Login, wobei Sie den Linux-Login-Namen und das dazugehörige Passwort angeben müssen. Anders als Samba verwendet Netatalk also dieselben Passwörter wie Linux.

[`test`] leitet die Definition des Netzwerkverzeichnisses (»Volume«) mit dem Namen `test` ein. Sein Inhalt ergibt sich aus dem Inhalt des lokalen Verzeichnisses `/data/for-macos`. Dabei werden die Zugriffsrechte unverändert von Linux nach macOS weitergegeben. Wenn der Login zum AFP-Volume als `kofler` erfolgt, dann können von macOS aus nur die Dateien gelesen bzw. verändert werden, für die auch der Linux-Benutzer `kofler` entsprechende Zugriffsrechte hat.

[`tm`] definiert ein weiteres Netzwerkverzeichnis, das aufgrund der Einstellung `time machine = yes` für Time-Machine-Backups verwendet werden darf. Dabei ist der Login nur für `kofler` zulässig. `vol size limit` limitiert die Größe des Backup-Volumens circa auf den angegebenen Wert in MiB. Die Einstellung verhindert, dass wiederholte Backups die gesamte Festplatte vollschreiben.

Netatalk unterstützt unzählige weitere Möglichkeiten, die Sie mit `man afp.conf` bzw. auf der Webseite <http://netatalk.sourceforge.net> nachlesen können.

**Firewall** Netatalk kommuniziert über den Port 548. Wenn dieser Port durch eine Firewall blockiert ist, was unter CentOS, Fedora, RHEL und SUSE der Fall ist, müssen Sie hierfür eine Ausnahmeregel definieren. Damit macOS überhaupt weiß, dass es einen neuen AFP-Server im Netz gibt, muss außerdem der Avahi-Dämon via MDNS kommunizieren dürfen.

In den folgenden Kommandos für Fedora/CentOS/RHEL findet das erste `firewall-cmd`-Kommando heraus, welche Firewall-Zone für die LAN-Schnittstelle `enp0s3` gilt. Für diese Zone werden dann Ausnahmeregeln definiert und aktiviert:

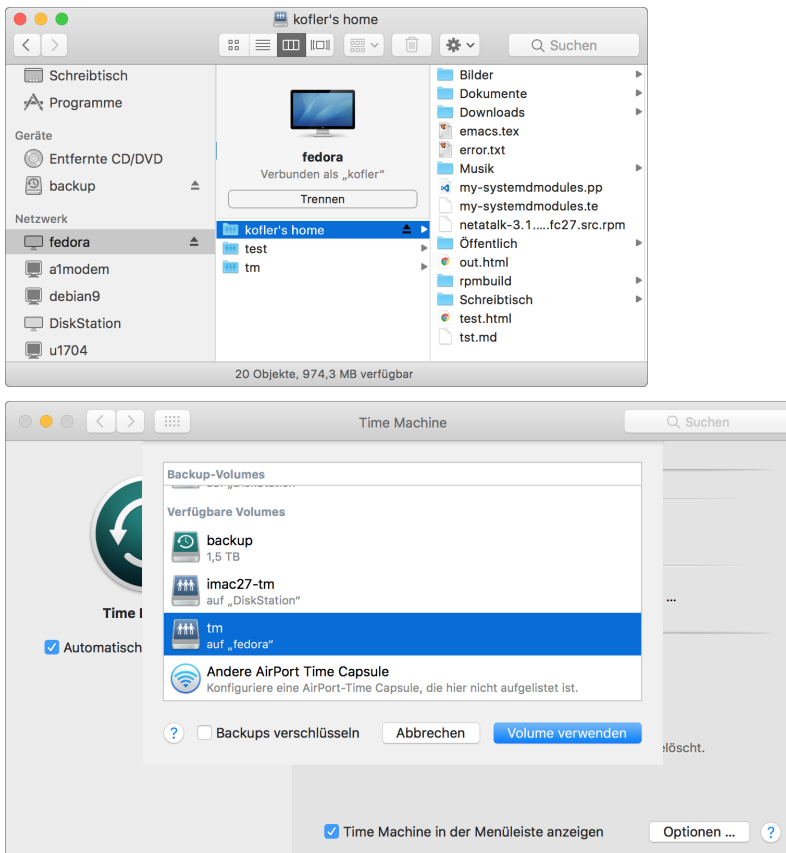
```
root# firewall-cmd --get-zone-of-interface=enp0s3 (aktive Zone herausfinden)
FedoraWorkstation
root# firewall-cmd --permanent --zone=FedoraWorkstation --add-service=mdns
root# firewall-cmd --permanent --zone=FedoraWorkstation --add-port=548/tcp
root# firewall-cmd --permanent --zone=FedoraWorkstation --add-port=548/udp
root# firewall-cmd --reload
```

Der Start von Netatalk erfolgt durch das Init-System der Distribution. Dabei sollten Sie sicherstellen, dass auch der avahi-daemon läuft, was in der Regel der Fall ist. Start

```
root# systemctl status avahi-daemon
Active: active (running) ...
root# systemctl start netatalk
root# systemctl enable netatalk
```

Unter macOS tauchen AFP-Server nahezu verzögerungsfrei in der Seitenleiste des Finders auf (siehe [Abbildung 29.1](#)). Beim ersten Login müssen Sie den passenden Linux-Benutzernamen und das Passwort angeben. In den Systemeinstellungen der Time Machine sind nur die Volumes sichtbar, die in `afp.conf` mit `time machine = yes` markiert sind. Im Konfigurationsdialog der Time Machine können Sie das bei der netatalk-Konfiguration für dafür vorgesehene Verzeichnis als »Backup-Volume« auswählen.

Zugriff auf  
AFP-Verzeichnisse  
unter macOS



**Abbildung 29.1** AFP-Verzeichnisse im Finder und in den Time-Machine-Einstellungen nutzen



# Kapitel 30

## CUPS

Das *Common UNIX Printing System* (kurz CUPS) ist unter Linux und macOS für die Verarbeitung und Zwischenspeicherung von Druckjobs und für die Umwandlung der Druckdaten in das Format des Druckers zuständig. Deswegen ist auf jedem Desktop-Rechner standardmäßig CUPS installiert. Dieses Kapitel geht auf einige CUPS-Grundlagen ein und zeigt, wie Sie einen mit dem Server verbundenen Drucker für die anderen Rechner im lokalen Netzwerk zugänglich machen.

Das Drucksystem CUPS steht unter der Lizenz GPL. Der Hauptentwickler von CUPS, Michael Sweet, ist seit einigen Jahren bei Apple angestellt. Apple hat bei dieser Gelegenheit die Rechte an CUPS erworben und kümmert sich um dessen Weiterentwicklung.

<https://www.cups.org>

### 30.1 Grundlagen

Auf einem Desktop-Rechner kommen Sie mit CUPS zumeist nur über die Drucker-Konfigurationsdialoge der Systemeinstellungen in Berührung. Sofern Ihr Drucker CUPS-kompatibel ist, bereitet seine Konfiguration und Verwendung kaum Probleme. Die folgenden Seiten richten sich in erster Linie an Anwender, die die Grundlagen und Hintergründe des Linux-Drucksystems verstehen möchten.

Generell erfolgt die Druckerverwaltung unter Linux durch einen Netzwerkdienst – auch auf Desktop-Rechnern. Jeder unter Linux eingerichtete Drucker kann bei entsprechender Konfiguration auch von allen anderen Rechnern im lokalen Netzwerk genutzt werden.

#### Ablauf des Druckprozesses

Die gesamte Druckphilosophie unter Unix/Linux basiert auf PostScript-Druckern. PostScript ist eine Programmiersprache zur Beschreibung von Seiteninhalten. PostScript-Drucker erwarten Druckdaten in diesem Format. Fast alle Linux-Programme mit Druckfunktionen senden PostScript-Daten an das Drucksystem.

PostScript und  
PDF

Seit einigen Jahren vollzieht sich ein Wandel vom PostScript- zum PDF-Format. Salopp formuliert ist PDF eine komprimierte Darstellung einer PostScript-Datei mit einigen Zusatzfunktionen. Da mittlerweile immer mehr Programme PDF-Dateien erzeugen können und immer mehr Drucker PDF-Dateien direkt verarbeiten können, versucht CUPS zunehmend, den Zwischenschritt PostScript zu vermeiden.

**Spooling-Systeme** In der Vergangenheit konnte `root` drucken, indem er eine geeignete PostScript-Datei direkt in die Device-Datei eines Druckers kopierte. Nun wollen normalerweise außer `root` auch gewöhnliche Benutzer drucken – möglicherweise auch solche, die an einem anderen Rechner im Netzwerk arbeiten. Und keiner von ihnen möchte sich dabei mit Device-Namen herumärgern. Aus diesem Grund gibt es sogenannte Spooling-Systeme. Sie haben mehrere Aufgaben:

- ▶ Sie stellen einfach zu bedienende Kommandos zum Drucken zur Verfügung, dank derer beim Ausdruck kein Device-Name, sondern einfach der Druckername angegeben werden muss.
- ▶ Sie erlauben je nach Konfiguration allen Benutzern das Drucken, bei Bedarf auch in einem Netzwerk.
- ▶ Sie ermöglichen es, an einen Rechner mehrere Drucker anzuschließen und diese zu verwalten.
- ▶ Wenn mehrere Druckaufträge gleichzeitig eintreffen, werden die Aufträge in sogenannten Warteschlangen (Print Queues) zwischengespeichert, bis der Drucker frei ist.
- ▶ Außerdem können Spooling-Systeme diverse Zusatzfunktionen übernehmen, etwa eine Protokollierung, wer wie viel druckt etc.

Das modernste und populärste Spooling-System für Linux ist CUPS. In der Vergangenheit kamen statt CUPS beispielsweise BSD-LPD oder LPRng zum Einsatz. Unabhängig vom Spooling-System sieht das Kommando zum Ausdruck einer Datei immer gleich aus:

```
user$ lpr -Pname datei
```

Dabei ist `name` der Name des Druckers (genau genommen: der Name der Druckerwarteschlange). Wenn Sie auf die Option `-P` verzichten, erfolgt der Ausdruck auf dem Standarddrucker.

**Drucker-Filter (Ghostscript)** Bis jetzt habe ich vorausgesetzt, dass Sie einen Drucker einsetzen, der PostScript oder möglicherweise sogar PDF direkt unterstützt. In der Praxis kommen aber häufig Drucker zur Anwendung, die nicht PostScript- oder PDF-kompatibel sind. Damit auch solche Drucker unter Linux funktionieren, ist eine Umwandlung der PostScript- oder PDF-Daten in das jeweilige Druckerformat erforderlich. Intern kommt dabei das Programm Ghostscript zum Einsatz (Kommando `gs`).



Um den Aufruf von `gs` kümmert sich ein sogenannter Filter. Das ist ein Programm (ein Script), das Eingabedaten verarbeitet und Ausgabedaten liefert. Der Filter für den Druckprozess muss insbesondere die richtigen Parameter an `gs` weitergeben, also den Namen des Druckermodells, die gewünschte Auflösung, die gewünschte Seitengröße etc. Er wandelt die PostScript-Daten seitenweise in Bitmaps um und gibt diese – zusammen mit den Druckbefehlen des jeweiligen Druckers – weiter.

Ghostscript greift bei seiner Arbeit auch auf externe Druckertreiber zurück. Das wichtigste Treiberprojekt für Linux ist Gutenprint (ehemals GIMP-Print):

<http://gimp-print.sourceforge.net>

Nun sind PostScript und PDF zwar die bevorzugten Formate aller CUPS-Druckdateien – aber manchmal soll einfach nur eine Text- oder Grafikdatei gedruckt werden. Natürlich können Sie die Textdatei in einen Editor laden, der die Datei dann im PostScript- oder PDF-Format an das Drucksystem weitergibt. Ebenso können Sie die Grafikdatei mit einem Grafikprogramm oder -konverter in die Formate PostScript oder PDF umwandeln.

Dokument-Filter

Noch bequemer ist es aber, auch für derartige Dateien einfach nur `lpr` datei auszuführen. Damit das funktioniert, versucht das Spooling-System, den Typ der zu druckenden Datei zu erkennen. Wenn das gelingt, wird die Datei mit geeigneten Programmen in das PostScript- oder PDF-Format umgewandelt.

Sie haben auf Ihrem Rechner einen Tintenstrahldrucker richtig konfiguriert. Der Druckername sei `pluto`. Nun möchten Sie die Grafikdatei `mypicture.png` ausdrucken und führen das folgende Kommando aus:

Alles zusammen

```
user$ lpr -Ppluto mypicture.png
```

Jetzt laufen die folgenden Operationen ab:

- ▶ `lpr` gibt die Datei an das Spooling-System CUPS weiter.
- ▶ Dieses gibt die Datei an das Filtersystem weiter.
- ▶ Der Filter erkennt den Dateityp (PNG) und wandelt die Bitmap je nach CUPS-Version in eine PostScript- oder PDF-Datei um.
- ▶ Diese Datei wird an Ghostscript weitergegeben, das sie in das herstellerspezifische Format des Druckers `pluto` umwandelt.
- ▶ Nachdem der Drucker `pluto` alle zuvor gestarteten Druckjobs verarbeitet hat, druckt er `mypicture.png` aus.

## 30.2 CUPS-Interna

### Konfigurationsdateien

Wie die meisten anderen Netzwerkfunktionen ist CUPS als Hintergrundprozess (Dämon) realisiert. Der Drucker-Dämon `cupsd` wird durch das Init-System gestartet. Bei älteren Drucksystemen erfolgte beinahe die gesamte Druckerkonfiguration durch die Datei `/etc/printcap`. Bei CUPS spielt diese Datei dagegen so gut wie keine Rolle mehr. Sie steht zwar aus Kompatibilitätsgründen noch immer zur Verfügung, enthält aber nur eine Liste aller bekannten Warteschlangen (ohne irgendwelche weiteren Parameter). Die eigentliche CUPS-Konfiguration erfolgt durch die Dateien des Verzeichnisses `/etc/cups`. [Tabelle 30.1](#) gibt einen Überblick über die wichtigsten Dateien.

| Datei                      | Inhalt  |
|----------------------------|---|
| <code>classes.conf</code>  | Definition aller Klassen                              |
| <code>cupsd.conf</code>    | zentrale CUPS-Konfigurationsdatei                     |
| <code>lptions</code>       | Veränderungen gegenüber der Grundkonfiguration        |
| <code>printers.conf</code> | Definition aller Drucker                              |
| <code>ppd/name.ppd</code>  | Konfiguration für die Warteschlange <code>name</code> |
| <code>.cups/lptions</code> | persönliche Einstellungen (KDE)                       |

**Tabelle 30.1** Konfigurationsdateien in `/etc/cups`

In `cupsd.conf` werden diverse Installationsverzeichnisse eingestellt, der Port des CUPS-Dämons für das *Internet Printing Protocol* (IPP), die Optionen für das Printerbrowsing, Sicherheitsparameter, Zugriffsrechte für Clients im Netzwerk (*allow/deny*) etc.

Das Verzeichnis `/etc/cups/ppd` enthält für jeden in `printers.conf` angeführten Druckernamen die dazugehörige PPD-Datei. Darin sind alle Druckparameter gespeichert, also Druckermodell und -treiber, Einstellungen wie Papiergröße und Auflösung etc.

Wenn der Systemadministrator `root` Druckeroptionen oder Einstellungen verändert, also die Blattgröße, die Druckauflösung, Längs- oder Querformat etc., dann werden diese Veränderungen in der Datei `lptions` gespeichert. Die Veränderungen gelten für alle Benutzer, die nicht schon selbst Veränderungen durchgeführt haben. Diese benutzerspezifischen Veränderungen werden in `.cups/lptions` gespeichert.

»If it ain't broke, don't fix it.«

CUPS ist ein sehr komplexes System. Verwenden Sie zur Konfiguration nach Möglichkeit die dazu vorgesehenen Werkzeuge. Manuelle Änderungen an der Konfiguration sind nur für CUPS-Profis empfehlenswert. Die in diesem Abschnitt zusammengefassten Informationen sind keinesfalls ausreichend! Mehr Details zur CUPS-Konfiguration finden Sie hier:

<https://www.cups.org/documentation.php>

`/usr/share/cups/mime/mime.types` enthält eine Liste aller Dokumenttypen, die von CUPS automatisch erkannt und in PostScript- oder PDF-Dateien umgewandelt werden. Die im gleichen Verzeichnis gespeicherte Datei `mime.convs` gibt an, welcher Filter verwendet werden soll. Die angegebenen Filter müssen sich als ausführbare Dateien in `/usr/lib/cups/filter` befinden. MIME

Wenn ein Drucker vorübergehend nicht erreichbar ist, z. B. weil er gerade ausgeschaltet ist, wird er von CUPS *angehalten*. CUPS merkt sich also, dass der Drucker nicht verwendet werden kann. Das Problem: Wird der Drucker später wieder eingeschaltet bzw. verbunden, erkennt CUPS das unter Umständen nicht. Sie müssen den Drucker dann selbst explizit reaktivieren. Diese Möglichkeit bieten alle CUPS-Benutzeroberflächen sowie die Kommandos `cupsenable druckername` bzw. `lpadmin -E -p <druckername>`. Nicht erreichbare Drucker

Um dem Problem ganz aus dem Weg zu gehen, empfiehlt es sich, in `/etc/cups/printers.conf` bei der Beschreibung des Druckers die Zeile `ErrorPolicy retry-job` hinzuzufügen. Manche CUPS-Konfigurationswerkzeuge verwenden diese Einstellung standardmäßig.

Für CUPS sieht jeder Drucker wie ein PostScript- oder PDF-Drucker aus. Drucker-spezifische Details wie die Größe des nicht bedruckbaren Seitenrands, die Druckerauflösung, Kommandos für bestimmte Zusatzfunktionen wie den Papiereinzug, Besonderheiten wie der Duplex-Druck etc. werden in PPD-Dateien gespeichert (*PostScript Printer Definition*). Das PPD-Format wurde von Adobe definiert und kommt auch unter Windows und macOS zum Einsatz. PPD-Dateien (PostScript Printer Definition)

Da natürlich nicht jeder Drucker tatsächlich ein PostScript- oder PDF-Drucker ist, enthalten CUPS-PPD-Dateien in Form von Kommentaren auch das erforderliche Ghostscript-Kommando inklusive aller Optionen, damit `gs` die Druckdaten in das Format des Druckers umwandeln kann. Die folgenden Zeilen zeigen einige Auszüge aus einer PPD-Datei für den Tintenstrahldrucker HP DeskJet 6980:

```
*PPD-Adobe: "4.3"
...
*Manufacturer: "HP"
*ModelName: "HP Deskjet 6980 Series hpijs"
*FoomaticIDs: "HP-DeskJet_6980 hpijs"
*FoomaticRIPCommandLine: "gs -q -dBATC -dPARANOIDSFAFER -dQUIET -dNOPAUSE
  -sDEVICE=ijs -sIjsServer=hpijs%A%B%C -dIjsUseOutputFD%Z -sOutputFile=- -"
...
```

Diese Informationen stammen aus der Datenbank `ppds.dat`, die PPD-Einträge für alle Drucker enthält, die CUPS bekannt sind. Die binäre Datei `ppds.dat` befindet sich je nach Distribution z. B. im Verzeichnis `/var/cache/cups`. Wenn Ihr Drucker in dieser Datenbank fehlt und Sie auch kein kompatibles Modell finden, hilft vielleicht eine passende `*.ppd`-Datei aus dem Internet weiter.

Beim Ausdruck extrahiert CUPS aus der `*.ppd`-Datei die Ghostscript-Parameter für den gewünschten Drucker, ruft damit `gs` auf und wandelt so die PostScript-Daten in das Format des jeweiligen Druckers um. Die resultierenden Daten werden dann an das Drucker-Device gesendet.

- HPLIP** HP entwickelt im Rahmen des Projekts *HP Linux Imaging and Printing* (kurz HPLIP) selbst freie Druckertreiber für viele seiner Drucker, Scanner und Multifunktionsgeräte. Als Lizenz kommt überwiegend die GPL zum Einsatz, teilweise auch die MIT- oder BSD-Lizenz. HP ist mit dieser aktiven Open-Source-Unterstützung ein leuchtendes Vorbild in der Computer-Industrie. Da viele HP-Drucker auch ohne HPLIP direkt von CUPS unterstützt werden, ist der Einsatz der HPLIP-Funktionen zumeist optional. Weitere Informationen zu HPLIP finden Sie hier:

<http://hplipopensource.com>

Zu HPLIP gibt es die grafische Benutzeroberfläche `hplip-toolbox`, die sich bei vielen Distributionen in einem eigenen Paket befindet (z. B. `hplip-gui` bei Ubuntu) und extra installiert werden kann. Das Programm erkennt selbstständig angeschlossene HP-Geräte und hilft bei deren Konfiguration und Anwendung. `hp-toolbox` kann unter anderem den Füllstand der Tintenpatronen vieler HP-Tintenstrahldrucker anzeigen – eine Funktion, die CUPS von sich aus nicht bietet.

- Klassen** Klassen helfen dabei, in großen Netzwerken einen Drucker-Pool einzurichten. Ein an eine Klasse geleiteter Ausdruck erfolgt dann auf dem ersten freien Drucker dieses Pools.
- IPP** CUPS unterstützt das *Internet Printing Protocol* (IPP). Dieses Protokoll vereinfacht die Nutzung von Druckern im Netzwerk über die Grenzen von Linux hinweg ganz erheblich (siehe auch [Abschnitt 30.4](#), »Drucken in lokalen Netzwerken«). IPP wird von allen

gängigen Betriebssystemen unterstützt. Detaillierte Informationen zu IPP finden Sie unter:

<https://pwg.org/ipp>

Alle an den Drucker gesandten Daten werden im Verzeichnis `/var/spool/cups/*` zwischengespeichert, bis der Ausdruck abgeschlossen ist. Beachten Sie, dass Spool-Daten auch bei einem Neustart von Linux nicht verloren gehen. `cupsd` stellt nach dem Neustart fest, dass es noch nicht ausgedruckte Dateien gibt, und wird weiterhin versuchen, die Daten an den Drucker zu übertragen.

Spooling

Der Zugriff auf CUPS wird normalerweise durch `/etc/cups/cupsd.conf` gesteuert. Auch wenn es mittlerweile unüblich ist, kann CUPS auch so kompiliert sein, dass zusätzlich die TCP-Wrapper-Bibliothek zum Einsatz kommt. Das überprüfen Sie mit `ldd`:

TCP-Wrapper

```
user$ ldd /usr/sbin/cupsd | grep wrap
...
libwrap.so.0 => /lib64/libwrap.so.0 (0x00007fa6e5c6a000)
```

Aus dem obigen Test geht hervor, dass die TCP-Wrapper-Bibliothek verwendet wird. CUPS kann somit nur genutzt werden, wenn dies in `/etc/hosts.deny` nicht verboten oder in `/etc/hosts.allow` explizit erlaubt ist. Standardmäßig sind beide Dateien leer, das Drucken ist also erlaubt. Details zur Konfiguration von `/etc/hosts.allow` und `hosts.deny` finden Sie in [Abschnitt 37.2](#), »Basisabsicherung von Netzwerkdiensten«.

## CUPS-Webschnittstelle

Grundsätzlich ist es möglich, die CUPS-Konfigurationsdateien mit einem Texteditor zu verändern. Für manche Basiseinstellungen mag das praktikabel sein, ansonsten rate ich davon wegen der großen Komplexität aber ab. Vernünftiger ist es zumeist, lokale Konfigurationswerkzeuge zu nutzen, also die Systemeinstellungen von KDE oder Gnome oder das SUSE-spezifische Programm YaST. Auf Server-Installationen ohne grafische Benutzeroberfläche können Sie zur Konfiguration auch die CUPS-Webschnittstelle verwenden (siehe [Abbildung 30.1](#)). Aus Sicherheitsgründen steht diese Schnittstelle nur auf dem lokalen Rechner zur Verfügung. Die folgende Adresse führt zur Startseite:

<http://localhost:631>

Im Frühjahr 2017 ist den Gnome-Tools zur Druckerverwaltung die Möglichkeit abhanden gekommen, Testseiten auszudrucken. Abhilfe schafft auch in diesem Fall die CUPS-Weboberfläche. Auf der Seite PRINTERS wählen Sie zuerst den gewünschten Drucker aus und führen dann MAINTENANCE • PRINT TEST PAGE aus.

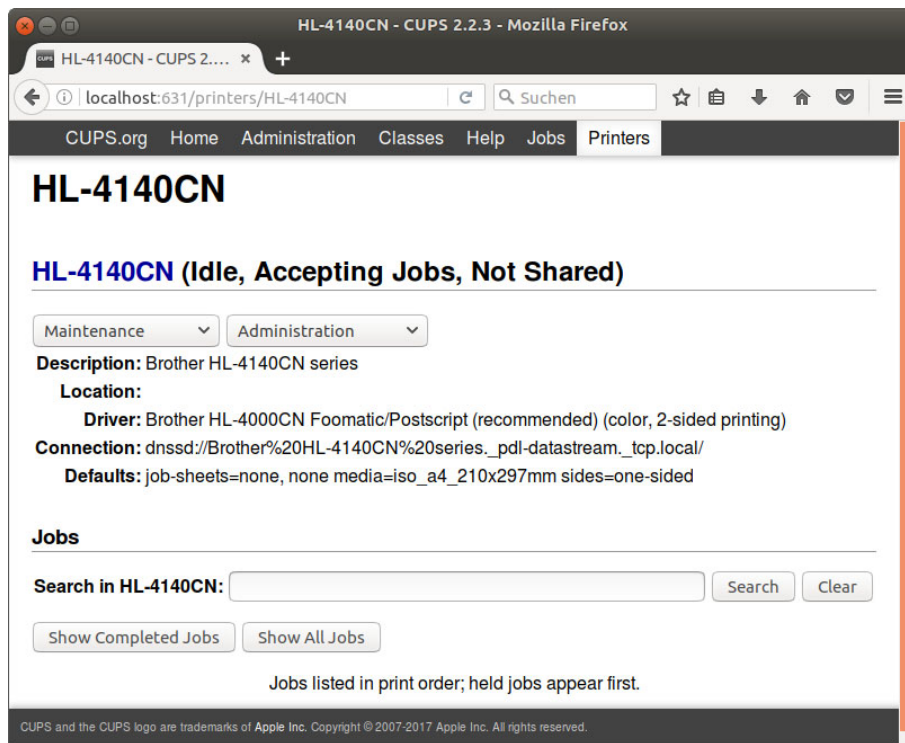


Abbildung 30.1 CUPS-Konfiguration im Webbrowser

Etwas schwieriger ist es, die Webschnittstelle auf einem anderen Rechner zu nutzen. Es ist möglich, die Konfigurationsdatei `/etc/cups/cupsd.conf` entsprechend anzupassen, was in der Praxis aber oft zu Problemen und Sicherheitslücken führt.

Einen wesentlich einfacheren Weg bietet das Kommando `ssh`, das Sie in einem Terminalfenster auf dem lokalen Rechner ausführen: Mit der Option `-L` können Sie den Port 631 des Rechners, auf dem CUPS läuft, auf einen beliebigen Port des lokalen Rechners umleiten.

Solange die SSH-Verbindung besteht, können Sie auf dem lokalen Rechner in einem Webbrowser die CUPS-Webschnittstelle nutzen, wobei Sie die im SSH-Kommando angegebene Portnummer verwenden. Im folgenden Beispiel wird also der Port 631 des CUPS-Rechners auf den Port 6310 des lokalen Rechners (`localhost`) umgeleitet:

```
user@local-machine# ssh -L 6310:localhost:631 user@cups-hostname
```

Im Webbrowser auf Ihrem lokalen Rechner geben Sie nun die folgende Adresse ein:

*`http://localhost:6310`*

Um die administrativen Teile der Webschnittstelle zu nutzen, müssen Sie sich mit einem Benutzernamen des CUPS-Rechners und dem dazugehörigen Passwort einloggen. Sie können nun neue Drucker einrichten, Druckjobs verwalten etc. Wenn Sie damit fertig sind, beenden Sie die SSH-Verbindung mit `[Strg]+[D]`.

### CUPS-Administration per Kommando

In der Regel werden Sie zum Drucken die Dialoge des jeweiligen Programms verwenden und zur Verwaltung der Druckjobs die entsprechenden Werkzeuge von Gnome oder KDE. Für Freunde der Kommandozeile stehen alternativ diverse Kommandos zur Auswahl, um Dateien zu drucken bzw. Druckjobs zu verwalten. Diese Kommandos sind vor allem dann hilfreich, wenn Sie Druckaufgaben durch Script-Dateien automatisieren möchten.

Die Kommandos `lpr`, `lpq`, `lprm` und `lpc` stehen nicht nur bei CUPS, sondern auch bei BSD-LPD und LPRng zur Verfügung. Das ist gewissermaßen ein gemeinsamer Nenner aller Spooling-Systeme. Beachten Sie aber, dass es Unterschiede bei den unterstützten Optionen gibt.

Mit `lpr` drucken Sie eine Datei aus. Falls Sie mehrere Drucker eingerichtet haben, geben Sie mit der Option `-P` ohne Leerzeichen den Namen der Warteschlange an. Für den Standarddrucker können Sie auf `-P` verzichten. `lpr`

```
user$ lpr -P<name> datei
```

Falls eine Druckdatei bereits im druckerspezifischen Format vorliegt, übergeben Sie an `lpr` die zusätzliche Option `-l`. Das Kommando umgeht nun das sonst übliche Filtersystem und sendet die Druckerdaten unverändert an den Drucker. Wenn Sie z. B. eine PostScript-Datei auf einem PostScript-Drucker ausdrucken wollen, kann das eine Menge Zeit sparen.

Durch eine Pipe kann `lpr` auch dazu verwendet werden, die Ausgabe eines anderen Kommandos auszudrucken. Das folgende Kommando druckt die mit `ls` ermittelte Dateiliste auf dem Standarddrucker aus:

```
user$ ls -l *.png | lpr
```

Statt `lpr` können Sie auch das Kommando `lp` verwenden (Syntax siehe man `1 lp`). Dieses Kommando soll Umsteigern von herkömmlichen Unix-Drucksystemen das Leben erleichtern.

Alle Druckaufträge, die nicht sofort ausgeführt werden können, werden zwischengespeichert, wobei es für jeden eingerichteten Drucker eine eigene Warteschlange gibt. Den Inhalt der Warteschlange sehen Sie sich mit `lpq -P<name>` an. `lpq, lprm`

Druckjobs, die Sie selbst initiiert haben, können Sie mit `lprm -P<name> <id>` wieder löschen, wobei Sie den Namen der Warteschlange und die ID-Nummer des Jobs angeben müssen. Die richtige Nummer ermitteln Sie vorher mit `lpq`:

```
user$ lpq
FS-1800+ ist nicht bereit
Rang   Besitz  Auftrag Datei(en)                Gesamtgröße
1st    kofler  20     evince-print             17408 Byte
2nd    kofler  21     evince-print             16384 Byte
user$ lprm 20
user$ lprm 21
```

**lpc** `lpc` gestattet eine feinere Kontrolle des Druckvorgangs. Nach dem Start dieses Kommandos befinden Sie sich in einer interaktiven Arbeitsumgebung, in der Sie Kommandos wie `status`, `help` etc. ausführen. `topq` verändert die Position eines Druckjobs in der Warteliste. Als Parameter geben Sie den Druckernamen und die Jobnummer an. Ein Teil der Kommandos in `lpc` (so auch `topq`) darf nur von `root` ausgeführt werden. `exit`, `bye` oder `quit` beenden `lpc`.

**lpstat, lpinfo, lpadmin, lpoptions** `lpstat` liefert Informationen über alle für CUPS verfügbaren Drucker. `lpinfo` ermittelt eine Liste aller verfügbaren Druck-Devices und Druckertreiber. Mit `lpadmin` richten Sie einen neuen Drucker ein bzw. löschen eine vorhandene Druckerkonfiguration. `lpoptions` zeigt die Optionen von CUPS-Druckern an bzw. verändert sie:

```
user$ lpoptions -o PageSize=A4
```

CUPS deaktiviert Drucker, die nicht erreichbar sind. Um den Drucker wieder zu aktivieren, führen Sie eines der beiden folgenden Kommandos aus:

```
user$ lpadmin -E <druckername>
user$ accept <druckername>
```

Um einen Drucker explizit zu deaktivieren, nutzen Sie das Kommando `reject`:

```
user$ reject <druckername>
```

### 30.3 Druckerkonfiguration

Bei der Konfiguration des Druckers helfen wahlweise die in Gnome bzw. KDE integrierten Werkzeuge, die CUPS-eigene Webkonfiguration oder spezielle Konfigurationsprogramme Ihrer Distribution.

**Druckertreiber** Die entscheidende Frage bei der Druckerkonfiguration ist natürlich, ob Ihr Drucker kompatibel zu Linux bzw. zur Kombination aus CUPS, Ghostscript und dessen Druckertreibern ist. Die folgende Aufzählung fasst zusammen, wie gut verschiedene Druckerkategorien unterstützt werden:



- ▶ **Laser-Drucker:** Die meisten Laser-Drucker sind PostScript- oder HP-kompatibel (Druckersprache PCL). Sie sind optimal für den Betrieb unter Linux geeignet. Das gilt auch für die meisten Netzwerkmodelle.
- ▶ **GDI-Drucker/Windows-Drucker:** Diese kaum mehr verbreiteten, zumeist billigen Laser-Drucker wurden speziell für den Einsatz unter Windows entwickelt. Die Grundidee besteht darin, dass ein Windows-Programm die gesamte zu druckende Seite zuerst vorbereitet und dann an den Drucker überträgt. Beim Seitenaufbau kommt die von Microsoft entwickelte Grafikschnittstelle GDI zum Einsatz. Leider ist das Format zur Datenübertragung zum Drucker zumeist nicht öffentlich dokumentiert. Daher werden viele derartige Drucker unter Linux nicht unterstützt.
- ▶ **Tintenstrahl- und Fotodrucker:** Bei Tintenstrahldruckern ist das Ausmaß der Linux-Unterstützung sehr stark vom jeweiligen Modell abhängig. Relativ gut klappt es bei vielen HP-Modellen. Neue Drucker fehlen aber mitunter in der CUPS-Druckerdatenbank noch. Mehr Probleme bereiten in der Regel Tintenstrahl-drucker anderer Hersteller.

Wenn Sie vor dem Kauf eines neuen Druckers stehen, lohnt auf jeden Fall ein Blick auf die folgende Website, die zahllose Informationen zum Thema Linux und Drucken enthält. Dazu zählt auch eine umfassende Datenbank der von Linux unterstützten Modelle:

<https://openprinting.org/printers>

Manche Drucker, zu denen es keinen Open-Source-Treiber gibt, werden vom kommerziellen Druckertreiber der Firma TurboPrint unterstützt. Außerdem können Sie mit TurboPrint bei manchen Fotodruckern bessere Ergebnisse erzielen als mit den Standardtreibern von CUPS. Sie finden den relativ preisgünstigen Treiber auf der folgenden Website:

**TurboPrint**

<http://www.turboprint.de>

Unabhängig davon, welches Konfigurationsprogramm Sie einsetzen, sollten die folgenden Tipps weiterhelfen:

**Konfigurations-tipps**

- ▶ Die automatische Druckererkennung kann nur funktionieren, wenn der Drucker eingeschaltet ist.
- ▶ Die Druckerkonfiguration in den Gnome-Systemeinstellungen wurde in den vergangenen Jahren immer wieder überarbeitet. Einige Versionen funktionierten akzeptabel, andere schlecht. Bei manchen Distributionen können Sie alternativ auch das alte und meiner Ansicht nach bessere Programm `system-config-printer` installieren. Alternativ können Sie die Konfiguration auch im Webbrowser auf der Seite <http://localhost:631> vornehmen.

- ▶ Zur manuellen Konfiguration müssen Sie zumindest die Schnittstelle (USB, Netzwerk etc.) und das Druckermodell angeben. Das Druckermodell wählen Sie aus einer riesigen Datenbank aus.

Falls Ihr Gerät nicht enthalten ist, versuchen Sie ein kompatibles Modell zu finden. Bei PostScript- und HP-kompatiblen Laserdruckern können Sie als Hersteller `GENERIC` wählen und dann den Standard angeben, z. B. PostScript oder PCL.

Zu manchen neuen Druckern, die in der CUPS-Modelldatenbank noch nicht enthalten sind, finden Sie im Internet passende `*.ppd`-Dateien. Sie können eine derartige Datei während der Konfiguration laden. Beachten Sie aber, dass nicht jede `*.ppd`-Datei CUPS-kompatibel ist bzw. unter Umständen eine ganz bestimmte CUPS-Version voraussetzt.

- ▶ Bei vielen Druckermodellen stehen mehrere Treiber zur Auswahl. Dafür kann es zwei Gründe geben: Erstens unterstützen viele Drucker verschiedene Standards, sind also z. B. gleichermaßen PostScript- und HP-kompatibel. Zweitens enthält CUPS Druckertreiber aus verschiedenen Projekten (Ghostscript, Gutenprint etc.). Wenn Sie unsicher sind, welcher Treiber die besten Ergebnisse erzielt, richten Sie den Drucker mehrfach unter verschiedenen Namen ein. Anschließend können Sie die verschiedenen Treiber bequem ausprobieren. Die Qualität des Ausdrucks hängt auch davon ab, was Sie drucken möchten – Text, technische Zeichnungen, Fotos etc. Die Druckqualität wird zudem durch die Einstellung der Treiberparameter beeinflusst, z. B. der DPI-Auflösung.
- ▶ Fast alle PostScript-Laserdrucker können auch in einem Kompatibilitätsmodus betrieben werden, sodass sie sich wie ein HP-Laserjet-Drucker verhalten. Sie können also Ihren PostScript-Drucker zumeist auch als HP-Laserjet-kompatiblen Drucker konfigurieren. Das bewirkt, dass die Druckdaten von Ihrem Computer in das HP-Laserjet-Format umgewandelt und dann an den Drucker gesandt werden. Das wirkt umständlich, kann aber bei älteren Druckern zu einer deutlichen Geschwindigkeitssteigerung führen.
- ▶ CUPS kann in den Grundeinstellungen wenig leistungsstarke Minirechner wie den Raspberry Pi überfordern. Der Ausdruck von Seiten dauert dann oft minutenlang. Eine erhebliche Beschleunigung des Druckprozesses erreichen Sie, wenn Sie die Druckauflösung reduzieren, z. B. auf 150 DPI (Dots Per Inch). Naturgemäß sinkt damit aber auch die Schärfe des Ausdrucks.

Sofern Sie einen Drucker verwenden, der PostScript- und womöglich auch PDF-kompatibel ist, geht es noch schneller – und das sogar bei optimaler Druckqualität! Dazu richten Sie den Drucker als sogenannten *Raw*-Drucker ein. Das bedeutet, dass die Druckdaten unverändert an den Drucker weitergeleitet werden. Ein Ausdruck kommt nun freilich nur dann zustande, wenn die Druckdaten bereits so formatiert sind, dass der Drucker sie auch versteht!

## 30.4 Drucken in lokalen Netzwerken

Dieser Abschnitt gibt einige Tipps zur Konfiguration eines Druckers, der über das Netzwerk mit dem Rechner verbunden ist. Dabei gibt es ziemlich viele Varianten, je nachdem, welche Protokolle der Netzwerkdrucker selbst versteht bzw. wie ein selbst nicht netzwerkfähiger Drucker mit einem Rechner im LAN verbunden ist:

- ▶ IPP-Drucker: Verwaltung z. B. durch Linux/Unix/macOS mit CUPS, Internet Printing Protocol
- ▶ Unix-Drucker: Verwaltung durch Linux/Unix, LPD-Protokoll
- ▶ Windows-Drucker: Verwaltung durch einen Windows-Rechner oder Samba-Server
- ▶ Socket-API: z. B. JetDirect von HP am IP-Port 9100
- ▶ herstellerspezifische Protokolle

### Firewall

Damit die Netzwerkfunktionen von CUPS bzw. des Protokolls IPP genutzt werden können, darf der Port 631 nicht blockiert sein. Unter CentOS, Fedora, RHEL und SUSE ist dies aber standardmäßig der Fall. Abhilfe schafft unter SUSE das Firewall-Modul von YaST. Unter CentOS, Fedora oder RHEL stellen Sie zuerst fest, welche Firewall-Zone der Netzwerkschnittstelle zum LAN zugeordnet ist, und erlauben dann den Dienst `ipp-client` (fremde Drucker nutzen) bzw. `ipp` (fremde Drucker nutzen und eigene Drucker freigeben):

```
root# firewall-cmd --get-zone-of-interface=enp4s0 (aktive Zone herausfinden)
FedoraWorkstation
root# firewall-cmd --permanent --zone=FedoraWorkstation --add-service=ipp
root# firewall-cmd --reload
```

### Konfiguration eines Netzwerkdruckers (Client-Sicht)

Die Details der Konfiguration hängen davon ab, über welches Protokoll die Kommunikation erfolgt. Am einfachsten funktioniert das Drucken im Netzwerk, wenn auf beiden Seiten CUPS bzw. das Protokoll IPP zum Einsatz kommt. Derartige Drucker sind ohne weitere Konfigurationsarbeiten auf der Client-Seite sichtbar und können sofort verwendet werden.

**IPP-Drucker  
verwenden**

`lpstat -v` liefert eine Liste aller konfigurierten Drucker. Das folgende Kommando zeigt, dass lokal ein Drucker mit dem Namen HL-4140 konfiguriert ist. Außerdem wurde auf einem iMac im lokalen Netz ein Epson-Drucker konfiguriert, der via IPP zugänglich ist.

```
user@uranus$ lpstat -v
device for HL-4140CN: dnssd://Brother-nnn._pdl-datastream._tcp.local
device for Epson:      ipp://imac.local:631/printers/Epson-nnn
```

Sie können beide Drucker sofort mit `lpr -P<name>` benutzen. Falls mehrere Drucker im Netzwerk denselben Namen haben, müssen deren Namen in der Form `druckername@hostname` angegeben werden, also z. B. als `lpr -PEpson-nnn@imac`.

### Drucker im Netzwerk teilen

Damit die CUPS-Drucker anderer Rechner auf dem lokalen Rechner sichtbar sind, muss CUPS auf den externen Rechnern so konfiguriert sein, wie dies im nächsten Abschnitt beschrieben wird. Port 631 darf nicht durch eine Firewall blockiert sein!

Ein IPP-Drucker kann so konfiguriert sein, dass er zwar im Netz genutzt werden kann, aber nicht automatisch sichtbar ist und von `lpstat -v` nicht angezeigt wird. In diesem Fall müssen Sie den Drucker auf dem lokalen Rechner zuerst konfigurieren. Sofern der externe Drucker via CUPS verwaltet wird, geben Sie als Hersteller und Modell RAW und QUEUE an. Das bedeutet, dass die zu druckenden Daten ohne Veränderung an den externen Rechner weitergeleitet werden. Dieser kümmert sich dann um die Aufbereitung der Daten für den Drucker.

#### Andere Netzwerkdrucker konfigurieren

Wenn der externe Netzwerkdrucker nicht IPP-kompatibel ist, muss er vor der ersten Verwendung clientseitig konfiguriert werden. Dazu setzen Sie dieselben Programme wie bei der Konfiguration eines lokalen Druckers ein, wählen als Druckertyp aber NETZWERKDRUCKER. Die weitere Konfiguration hängt vom Protokoll ab:

- ▶ LPD (Unix-LPD): Hier geben Sie den Hostnamen des Rechners/Druckers sowie den Namen der Warteschlange an, im Zweifelsfall einfach `lp` oder `lp0`.
- ▶ SMB (Windows/Samba): Sie müssen den Hostnamen des Rechners, den Druckername sowie eventuell Benutzernamen und Passwort angeben. Bevor Sie einen Windows-Drucker verwenden können, müssen Sie das Samba-Client-Paket installieren.
- ▶ SOCKET-PROTOKOLL bzw. HP JETDIRECT: Hier geben Sie den Hostnamen oder die IP-Adresse des Druckers sowie die Port-Nummer an, in der Regel 9100.

### Konfiguration eines CUPS-Netzwerkdruckers (Server-Sicht)

Immer mehr Drucker sind selbst netzwerkfähig. Sie verbinden derartige Drucker einfach mit dem lokalen Netzwerk, und schon kann jeder Rechner im LAN auf den Drucker zugreifen. Das Thema dieses Abschnitts ist aber ein anderes: Wie können Sie einen lokalen Drucker *ohne* Netzwerkschnittstelle im Netzwerk nutzen?

Sicherlich haben Sie es schon erraten – via CUPS. Auf dem Rechner, der mit dem Drucker verbunden ist, konfigurieren Sie CUPS so, dass der Drucker-Server allen anderen Rechnern im Netzwerk Zugang zum lokalen Drucker gibt. Anschließend können Sie den Drucker von allen gängigen Betriebssystemen aus über IPP ansprechen.

Normalerweise kann ein via CUPS eingerichteter Drucker nur vom lokalen Rechner aus genutzt werden. Wenn das Druckerkonfigurationsprogramm Ihres Desktop-Systems keine Möglichkeit bietet, die Druckerfreigabe zu aktivieren, öffnen Sie mit einem Webbrowser die Seite `http://localhost:631` und aktivieren auf der Seite VERWALTUNG die Option DRUCKER DIESES SYSTEMS IM NETZ FREIGEBEN (bzw. ohne deutsche Lokalisierung: ADMINISTRATION • SHARE PRINTERS CONNECTED TO THIS SYSTEM). Beim Speichern müssen Sie sich authentifizieren – in der Regel als root, nur unter Ubuntu als gewöhnlicher Benutzer.

Server-  
Konfiguration

Wenn Sie die Konfigurationsdateien lieber selbst verändern, müssen Sie sicherstellen, dass die folgenden Einstellungen in `cupsd.conf` enthalten sind. Anschließend starten Sie CUPS neu (`systemctl restart cups`):

```
# Änderungen in /etc/cups/cupsd.conf
Port 631
Browseing On
BrowseOrder allow,deny
BrowseAddress @LOCAL

<Location />
  ...
  Order allow,deny
  Allow @LOCAL
</Location>
```

Port 631 bedeutet, dass CUPS über den Netzwerk-Port 631 kommuniziert. `BrowseAddress @LOCAL` bewirkt, dass die CUPS-Informationen an alle lokalen Netzwerkschnittstellen gesendet werden (Broadcast), nicht aber an Internetschnittstellen. Alternativ kann mit `BrowseAddress @IF(enp0s4)` auch eine bestimmte Netzwerkschnittstelle angegeben werden.

`cupsd.conf` sieht eine Reihe weiterer `Browse`-Schlüsselwörter vor. Beispielsweise steuern `BrowseAllow` und `BrowseDeny`, von welchen Rechnern CUPS-Informationen *empfangen* werden. Standardmäßig gibt es keine Empfangseinschränkungen, und es ist selten notwendig oder sinnvoll, diese oder die anderen `Browse`-Einstellungen zu ändern. `Allow @LOCAL` bewirkt, dass andere Rechner im lokalen Netzwerk die von CUPS angebotenen Drucker tatsächlich nutzen dürfen.

Damit andere Linux-Rechner im Netz den externen CUPS-Drucker automatisch erkennen, müssen Sie die Option FREIGEgebenEN DRUCKER ANDERER SYSTEME ANZEIGEN oder eine ähnlich lautende Einstellung aktivieren.

Client-  
Konfiguration

Alternativ ist es auch möglich, den Drucker manuell einzurichten. Dazu starten Sie den Dialog zur Konfiguration eines neuen Druckers, wählen als Gerätetyp INTERNET PRINTING PROTOCOL und geben den Hostnamen des Servers an. Der Konfigurationsdialog zeigt dann eine Liste aller auf dem Server verfügbaren Drucker an.

**Windows** Auch unter Windows können CUPS-Drucker genutzt werden. Dazu wählen Sie im Druckerkonfigurationsdialog die Option VERBINDUNG MIT EINEM DRUCKER IM INTERNET ODER NETZWERK HERSTELLEN und geben die folgende Adresse an:

```
http://mars.sol:631/printers/pluto
```

Dabei müssen Sie natürlich `mars.sol` durch den Hostnamen des CUPS-Servers ersetzen und `pluto` durch den Namen des Druckers. Als Treiber geben Sie nach Möglichkeit den tatsächlichen Druckertreiber an. Wenn es unter Windows keinen Treiber für Ihren Drucker gibt (was unwahrscheinlich ist), können Sie auch einen beliebigen PostScript-Druckertreiber verwenden.

Wenn der Ausdruck statt des erwarteten Ergebnisses nur wirren Text bzw. undefinierbare Grafikmuster enthält, ist zumeist eine doppelte Verarbeitung der Druckdaten schuld: Zuerst wandelt der Windows-Treiber den Ausdruck in das Format des Druckers um. Diese Daten kommen dann bei CUPS an und werden dort ein zweites Mal formatiert (in der CUPS-Nomenklatur: »gefiltert«). Das kann natürlich nicht gut gehen.

Abhilfe: Richten Sie nur unter Windows den für den Drucker erforderlichen Treiber ein! Auf dem CUPS-Server konfigurieren Sie den Drucker in Form einer sogenannten Raw-Warteschlange, die die empfangenen Daten ohne Veränderung einfach an den Drucker weiterleitet. Dazu wählen Sie bei der Druckerkonfiguration als Gerätetyp RAW und als Modell RAW QUEUE.

## 30.5 AirPrint

Grundsätzlich unterstützt CUPS *AirPrint*. Das ist eine von Apple entwickelte Funktion, damit iPhones und iPads über das WLAN drucken können. Damit AirPrint funktioniert, müssen sich Ihr Linux-Rechner und das iOS-Gerät im selben Funknetz befinden. Außerdem müssen CUPS und Avahi richtig konfiguriert sein. Avahi ist ein Hintergrundprogramm, das eine automatische Nutzung des Netzwerks mittels Zeroconf ermöglicht (siehe [Abschnitt 26.7](#), »Zeroconf und Avahi«).

Leider funktioniert AirPrint bei den wenigsten Distributionen auf Anhieb. Die folgende Anleitung zeigt, wie Sie AirPrint unter Fedora und Ubuntu zum Laufen bringen. Bei anderen Distributionen ist die Vorgehensweise analog.

Der erste Schritt besteht darin, das Drucken im Netzwerk zu erlauben. Dazu öffnen Sie im Webbrowser die Seite <http://localhost:631/admin> und aktivieren die Option DRUCKER DIESES SYSTEMS IM NETZ FREIGEBEN.

Netzwerkdruck  
erlauben

Als Nächstes laden Sie das winzige Python-Script `airprint-generate` von der folgenden Website herunter:

AirPrint-  
Beschreibung des  
Druckers

<https://github.com/tjfontaine/airprint-generate>

Dieses Script führen Sie nun aus:

```
user$ python ~/Downloads/airprint-generate.py
```

Das Script speichert im lokalen Verzeichnis Dateien mit der Avahi-Beschreibung aller konfigurierten Drucker. Für den designierten AirPrint-Drucker verschieben Sie diese Datei in das Verzeichnis `/etc/avahi/services`:

```
root# mv druckername.service /etc/avahi/services
```

Unter CentOS/Fedora/RHEL müssen Sie auf die korrekten SELinux-Attribute achten:

SELinux und  
Firewall

```
root# restorecon /etc/avahi/services/*
```

Außerdem muss die Firewall für den Port 631 durchlässig sein (siehe [Abschnitt 30.4](#), »Drucken in lokalen Netzwerken«).

Führen Sie nun den erforderlichen Neustart von Avahi und CUPS durch:

Neustart von  
CUPS und Avahi

```
root# systemctl restart avahi-daemon
```

```
root# systemctl restart cups
```

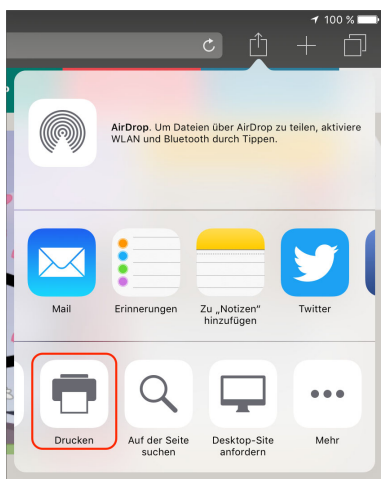


Abbildung 30.2 AirPrint-Drucker auswählen

**Praxis** iOS-Geräte sollten den Drucker nun erkennen. In den Drucker-Dialog gelangen Sie, wenn Sie nach dem Drücken des TEILEN-Buttons die Icons in der untersten Leiste nach links verschieben (siehe Abbildung 30.2). In Safari ist das Drucker-Icon hinter den Icons LESEZEICHEN, ZUR LESELISTE HINZUFÜGEN, ZUM HOME-BILDSCHIRM und ZU FREIGEgebenEN LINKS HINZUFÜGEN versteckt.



TEIL VII

# Root-Server



# Kapitel 31

## Secure Shell (SSH)

Server, die nicht physikalisch vor Ihnen stehen, können Sie nur über eine Netzwerkverbindung administrieren. Das bevorzugte Werkzeug hierfür ist SSH (*Secure Shell*). Es ermöglicht sowohl die einfache Ausführung von Kommandos als auch die Bedienung grafischer Programme über eine sichere Netzwerkverbindung. Die einzige Voraussetzung besteht darin, dass auf dem Server ein SSH-Server installiert ist.

Das Kommando `ssh` ist der Nachfolger von `telnet` und `rlogin` und ist erheblich sicherer. Der Client-Einsatz dieses Kommandos wurde bereits in [Abschnitt 14.2](#), »Auf anderen Rechnern arbeiten (SSH)«, beschrieben. Dieses kurze Kapitel gibt einige Tipps, wie Sie einen SSH-Server so sicher wie möglich einrichten. Außerdem geht das Kapitel auf die Verwendung von Schlüsseln zur Authentifizierung ein.

Dieses SSH-Kapitel ist das erste von mehreren Kapiteln, in denen Netzwerkdienste beschrieben werden, die üblicherweise auf einem Root-Server laufen. Als »Root-Server« wird ein externer Rechner in einem Rechenzentrum bezeichnet, den Sie ohne Einschränkungen selbst administrieren dürfen, eben mit `root`-Rechten. Für erfahrene Linux-Administratoren stellen Root-Server eine preisgünstige Möglichkeit dar, eigene Webauftritte einzurichten, einen eigenen E-Mail-Server zu betreiben etc. Das ist vor allem für kleine und mittelgroße Unternehmen interessant, deren Internetzugang auf ADSL basiert und damit ungeeignet für Server-Dienste mit einem eigenen Domainnamen ist.

Root-Server

Bei der Auswahl eines Root-Servers sollten Sie vor allem auf die Seriosität des Hosting-Unternehmens achten. Nichts ist ärgerlicher als ein nicht funktionierender Support bei einem Rechner, zu dem Sie selbst keinen physischen Zugang haben. Wichtig sind auch administrative Hilfen, z. B. eine Webschnittstelle, um nach einem Absturz einen Server-Reboot auszulösen oder ein Rettungssystem zu starten.

Achten Sie schließlich darauf, welche Linux-Distributionen installiert werden können – nicht jeder Provider unterstützt alle gängigen Distributionen. Gut geeignet für den Root-Server-Einsatz sind neben den teuren Enterprise-Distributionen vor allem Debian, Ubuntu LTS und CentOS. Aufgrund des zu kleinen Wartungszeitraums sind Fedora und openSUSE nicht empfehlenswert.

## 31.1 Installation

Bei vielen Distributionen wandert der SSH-Server bereits während der Erstinstallation auf die Festplatte. Nur wenn der SSH-Server noch nicht installiert ist, müssen Sie selbst Hand anlegen.

**Debian, Ubuntu** Unter Debian und Ubuntu installieren Sie das Paket `openssh-server`. Der Dämon `sshd` wird sofort automatisch gestartet.

```
root# apt install openssh-server
```

**Fedora, CentOS, RHEL** Unter Fedora und RHEL führen Sie die Installation analog mit `yum` durch:

```
root# dnf install openssh-server (Fedora)
root# yum install openssh-server (CentOS, RHEL)
```

Unter Fedora wird der SSH-Server sofort automatisch ausgeführt. Unter CentOS/RHEL muss hingegen sowohl der erstmalige Start als auch die Aktivierung des automatischen Starts manuell durchgeführt werden.

```
root# systemctl start sshd      (SSH-Server erstmalig starten)
root# systemctl enable sshd     (SSH-Server zukünftig automatisch starten)
```

**SUSE** Unter SUSE lautet der Paketname `openssh`. Das Paket ist standardmäßig installiert. Während der Installation gibt es aber zwei Optionen, die steuern, ob der SSH-Server auch tatsächlich gestartet wird und ob die Firewall den Port 22 blockieren soll. Die Defaulteinstellungen lauten: nicht starten und blockieren.

Wenn Sie diese Optionen also bei der Installation übersehen haben, dann erlauben Sie den Dienst SSH im YaST-Modul `FIREWALL` und führen die folgenden Kommandos aus:

```
root# systemctl start sshd      (SSH-Server erstmalig starten)
root# systemctl enable sshd     (SSH-Server zukünftig automatisch starten)
```

**Firewall** Der Zugriff auf den SSH-Server scheitert, wenn eine Firewall den Port 22 blockiert. Unter CentOS, Fedora, openSUSE und RHEL müssen Sie gegebenenfalls die Firewall-Konfiguration entsprechend ändern (siehe auch [Kapitel 37](#), »Firewalls«).

## 31.2 Konfiguration und Absicherung

Die Konfigurationsdateien zu `sshd` befinden sich im Verzeichnis `/etc/ssh`. Für die Server-Konfiguration ist `sshd_config` zuständig. Normalerweise kann diese Datei unverändert bleiben, d. h., der SSH-Server sollte auf Anhieb funktionieren. Die Kommunikation erfolgt standardmäßig über den IP-Port 22. Wenn Sie besondere Anforderungen stellen, finden Sie in den `man`-Seiten zahlreiche weitere Informationen.

Ein Bestandteil des SSH-Servers ist der sftp-Server. Dabei handelt es sich um eine sichere Alternative zu einem normalen FTP-Server. Die sftp-Funktionen stehen normalerweise automatisch zur Verfügung, sobald der SSH-Server läuft. Werfen Sie gegebenenfalls einen Blick auf die Zeile `Subsystem sftp` in `sshd_config`. Die sftp-Funktionen können mit sftp-kompatiblen Clients genutzt werden, z. B. mit dem Programm `sftp`. `sftp` sieht lediglich User-Logins vor, aber kein Anonymous FTP.

Secure FTP

Grundsätzlich läuft der SSH-Server auf Anhieb ohne Konfigurationsarbeit. Das ist allerdings ein nicht zu unterschätzendes Sicherheitsrisiko: Jeder, der eine gültige Kombination aus Benutzername und Passwort errät, kann sich auf Ihrem Rechner anmelden! Cracker verwenden automatisierte Tools, die im Internet nach Servern suchen und sich dort einzuloggen versuchen. Alle derartigen Aktivitäten werden in der Datei `/var/log/auth.log` vermerkt. Auf öffentlich erreichbaren Servern finden Sie darin täglich Tausende von Einlog-Versuchen.

Absicherung

Sie tun also gut daran, alle Benutzer durch nichttriviale Passwörter abzusichern! Unter Debian und Ubuntu können Sie beispielsweise mit dem Kommando `makepasswd` sichere Passwörter erzeugen. Unter CentOS, Fedora und RHEL installieren Sie stattdessen das Paket `expect` mit dem vergleichbaren Kommando `mkpasswd`.

Die Datei `/etc/shadow`, in der in verschlüsselter Form alle Benutzerpasswörter gespeichert sind, darf auf keinen Fall Einträge ohne Passwort enthalten! Sie erkennen derartige Einträge daran, dass in einer Zeile zwischen dem ersten und dem zweiten Doppelpunkt kein Text enthalten ist. Üblicherweise befindet sich dort entweder ein verschlüsseltes Passwort oder bei System-Accounts ein Sonderzeichen (zumeist `*` oder `!`), das Logins vollständig unmöglich macht. Sollten Sie in dieser Datei tatsächlich einen Eintrag ohne Passwort finden, beheben Sie den Missstand mit `password name`.

Die folgenden Maßnahmen reduzieren jeweils die Wahrscheinlichkeit eines Crack-Angriffs auf Ihren SSH-Server. Sie können einzeln oder in Kombination angewendet werden.

Ein Angreifer möchte `root`-Rechte erzielen – und am einfachsten gelingt das natürlich durch einen `root`-Login. Dabei muss nur ein Parameter (das `root`-Passwort) erraten werden. Wesentlich sicherer ist es, einen direkten `root`-Login via SSH zu verbieten. Sie müssen sich also unter einem anderen Benutzernamen einloggen und dann mit `su` oder `sudo` in den `root`-Modus wechseln. Wenn Sie einen Root-Server administrieren, sollten Sie das unbedingt testen, bevor Sie die folgende Änderung durchführen – sonst sperren Sie sich womöglich selbst aus!

Kein root-Login

```
# Änderung in /etc/ssh/sshd_config
...
PermitRootLogin = no
```

Damit die Änderung wirksam wird, müssen Sie `sshd` dazu auffordern, die Konfigurationsdateien neu einzulesen:

```
root# systemctl reload sshd
```

Für den Angreifer hat das die Konsequenz, dass nun zwei Parameter unbekannt sind: der Login-Name *und* das Passwort!

Eine sinnvolle Alternative zu `PermitRootLogin = no` ist die Einstellung `prohibit-password` bzw. das gleichwertige Schlüsselwort `without-password`:

```
# Änderung in /etc/ssh/sshd_config
...
PermitRootLogin = without-password
```

Keine Angst, diese Einstellung erlaubt keineswegs einen `root`-Login mit leerem Passwort! `without-password` bedeutet vielmehr, dass ein Login weiterhin möglich ist, dass aber die Authentifizierung durch ein sichereres Verfahren als durch die simple Passwordeingabe erfolgen muss – in aller Regel durch den Austausch von Schlüsseln (siehe den [Abschnitt 31.4](#)).

#### Passwort-Authentifizierung sperren

`PermitRootLogin` gilt nur für `root`. Wenn Sie generell jede Authentifizierung per Passwort ausschließen möchten, bauen Sie in `sshd_config` die Anweisung `PasswordAuthentication no` ein. Eine Authentifizierung ist dann ausschließlich über Schlüssel möglich. Vergessen Sie nicht, das vorher zu testen – sonst sperren Sie womöglich den einzigen Netzwerkzugang zu Ihrem Server.

#### IPv6-Zugang sperren

Wenn Rechner im lokalen Netzwerk über IPv4 kommunizieren, außerdem aber eine IPv6-Konfiguration oder ein IPv6-Tunnel vorliegt, ist es zweckmäßig, den IPv6-Zugang auf den SSH-Server zu blockieren. Damit können Sie im lokalen Netzwerk via SSH auf Ihren Rechner zugreifen, nicht aber von außen. Genau das erreichen Sie durch die Einstellung `AddressFamily inet`. Andere zulässige Einstellungen sind `inet6` (nur IPv6) und `any` (sowohl IPv4 als auch IPv6, gilt per Default).

```
# Änderung in /etc/ssh/sshd_config, nur IPv4 akzeptieren
...
AddressFamily inet
```

#### SSH-Port ändern

Der SSH-Server kommuniziert standardmäßig über den Port 22. Mit der `Port`-Zeile können Sie mühelos einen anderen, momentan unbenutzten Port einstellen. Da viele automatisierte Crack-Tools nur den Port 22 berücksichtigen, vermeiden Sie auf einen Schlag viele Sicherheitsprobleme.

Auf Red-Hat-, CentOS- und Fedora-Systemen müssen Sie die Änderung der Port-Nummer auch SELinux mitteilen:

```
root# semanage port -a -t ssh_port_t -p tcp <newportnr>
```

Bei der Verwendung von `ssh` müssen Sie nun jedes Mal mit `-p` den Port Ihres SSH-Servers explizit angeben. Beachten Sie, dass Sie beim Kommando `scp` die Option `-P` verwenden müssen, weil `-p` dort die Bedeutung *preserve* hat und bewirkt, dass Zeit- und Zugriffsinformationen der zu kopierenden Datei erhalten bleiben!

Sie sollten sich freilich im Klaren darüber sein, dass der Schutz durch den Port-Wechsel nur begrenzte Wirkung hat: Wer Ihren Server ernsthaft angreifen will und nicht nur auf der Suche nach dem nächstbesten schlecht konfigurierten Server zur Installation eines Root-Kits ist, der wird einen Port-Scan durchführen. Damit bleibt Ihr SSH-Server nicht lange unentdeckt, egal auf welchem Port er läuft.

Die Veränderung des SSH-Ports hat zudem einen nicht unerheblichen Nachteil: Während die meisten Firewalls so konfiguriert sind, dass sie Verkehr über den Port 22 zulassen, wird dies für Ihren neuen Port wahrscheinlich nicht zutreffen. Wenn Sie von einem Unternehmen, wo Sie gerade ein paar Tage arbeiten, schnell via SSH auf Ihren Server zugreifen möchten, scheitern Sie womöglich bereits an der Firmen-Firewall.

Der SSH-Server verwendet die sogenannte TCP-Wrapper-Bibliothek. Deswegen können Sie auch durch die Konfigurationsdateien `/etc/hosts.allow` und `/etc/hosts.deny` steuern, von welchen Netzwerkadressen der SSH-Server genutzt werden kann.

TCP-Wrapper

Bei einem SSH-Server auf einem Root-Server ist diese Art der Absicherung selten zweckmäßig – SSH soll ja gerade aus dem ganzen Internet verwendbar sein. Wenn Sie dagegen einen SSH-Server auf einem LAN-Server installiert haben und die Administration nur innerhalb des LANs erfolgen soll, ist es durchaus sinnvoll, die Zugriffsmöglichkeiten entsprechend einzuschränken. Details zur TCP-Wrapper-Bibliothek und den dazugehörigen Konfigurationsdateien finden Sie in [Abschnitt 37.2](#), »Basisabsicherung von Netzwerkdiensten«.

### 31.3 DenyHosts und Fail2Ban

SSH ist insofern ein inhärentes Sicherheitsrisiko, als ein entsprechend hartnäckiger Angreifer durch endloses Ausprobieren früher oder später ein gültiges Login-Passwort erraten kann. Hacker können dazu auf automatisierte Passwort-Cracking-Tools wie `hydra`, `ncrack` oder `medusa` zurückgreifen.

Solchen Angriffen lässt sich zum Glück relativ leicht ein Riegel verschieben, wenn man nach mehreren erfolglosen Versuchen, die von einem bestimmten Host ausgehen, diesen einfach eine Weile blockiert. Genau diese Aufgabe erfüllen die Programme `DenyHosts` und `Fail2Ban`.

## DenyHosts

Das Python-Script DenyHosts überwacht SSH-Login-Versuche. Sobald es feststellt, dass es von einer IPv4-Adresse aus mehrere vergebliche Logins gibt, wird diese IP-Adresse automatisch der Datei `/etc/hosts.deny` hinzugefügt. Je nach Konfiguration bleibt die IP-Adresse nun für immer in `hosts.deny` oder wird nach einer bestimmten Zeit wieder entfernt, z. B. nach einem Tag oder einer Woche.

Der Ansatz von DenyHosts ist ebenso simpel wie effizient. DenyHosts verhindert automatisierte Login-Versuche wirksam – es sei denn, es handelt sich um einen Angriff, der gleichzeitig von sehr vielen unterschiedlichen Rechnern aus erfolgt. Leider ist die Weiterentwicklung des Scripts 2008 eingestellt worden; deswegen stellen immer weniger Distributionen fertige DenyHosts-Pakete zur Verfügung. Eine mögliche Alternative ist das im nächsten Abschnitt beschriebene Programm Fail2Ban. Weitere Informationen und die gerade aktuelle Version von DenyHosts finden Sie unter:

<http://denyhosts.sourceforge.net>

**Konfiguration** DenyHosts wird durch das Init-System gestartet und wertet die Konfigurationsdatei `/etc/denyhosts.conf` aus. Entscheidend ist, dass `denyhost` die richtige Logging-Datei überwacht (Parameter `SECURE_LOG`). Das folgende Listing zeigt eine Beispielkonfiguration:

```
# Datei /etc/denyhosts.conf
SECURE_LOG = /var/log/auth.log
HOSTS_DENY = /etc/hosts.deny

# blockierte IP-Adressen nach 24h wieder freigeben
PURGE_DENY = 1d

# blockieren nach drei vergeblichen Versuchen für einen falschen Login-Namen
DENY_THRESHOLD_INVALID = 3

# blockieren nach fünf vergeblichen Versuchen für einen richtigen Login-Namen
DENY_THRESHOLD_VALID = 5

# blockieren nach einem vergeblichen Versuch für root
DENY_THRESHOLD_ROOT = 1
```

**IPv6** DenyHosts bietet keinen Schutz vor Angriffen aus dem IPv6-Netz! Sofern Sie den SSH-Zugriff via IPv6 nicht unbedingt benötigen, sollten Sie ihn einfach komplett sperren oder nur einen Login mit Schlüsseln zulassen.



## Fail2Ban

Fail2Ban ist ein moderner Nachfolger von DenyHosts. Fail2Ban unterscheidet sich in zwei Punkten von DenyHosts: Zum einen kann das Programm neben SSH auch diverse andere Netzwerkdienste überwachen und bei Bedarf für einzelne Hosts blockieren; zum anderen erfolgt die Blockade nicht durch Einträge in `/etc/denyhosts.conf`, sondern durch Firewall-Regeln.

Nach der Installation des `fail2ban`-Pakets erstellen Sie im Verzeichnis `/etc/fail2ban` eine Kopie der vorgegebenen Konfigurationsdatei `jail.conf` unter dem Namen `jail.local`. Dort führen Sie alle weiteren Änderungen durch. Diese Änderungen haben nun Vorrang gegenüber den Grundeinstellungen in `jail.conf`.

Konfiguration

```
root# cd /etc/fail2ban
root# cp jail.conf jail.local
```

Als *Jail* bezeichnet Fail2Ban einen zu überwachenden Dienst. Die mitgelieferten Konfigurationsdateien in `/etc/fail2ban` enthalten Regeln für alle möglichen Dienste (Jails); standardmäßig sind diese aber alle deaktiviert. Um die SSH-Regeln zu aktivieren, suchen Sie in `jail.local` nach dem Abschnitt `[sshd]`. In diesem Abschnitt fügen Sie die Zeile `enabled=true` ein:

```
# Datei /etc/fail2ban/jail.local
...
[sshd]
port    = ssh
logpath = %(sshd_log)s
enabled = true
```

Die Defaultkonfiguration von Fail2Ban sieht vor, dass ein Host nach fünf vergeblichen SSH-Login-Versuchen innerhalb von zehn Minuten für zehn Minuten gesperrt wird. Die drei relevanten Parameter `maxretry`, `bantime`, `findtime` sind gemeinsam für alle Dienste in `jail.local` vordefiniert. Sie können diese Defaultwerte bei Bedarf unkompliziert ändern oder im `[sshd]`-Block nur für SSH andere Werte festlegen. Details zur Konfiguration von Fail2Ban finden Sie in `man jail.conf`.

Jetzt müssen Sie sich nur noch darum kümmern, dass Fail2Ban auch gestartet wird: **Start**

```
root# systemctl restart fail2ban
root# systemctl enable fail2ban
```

Den aktuellen Status von Fail2Ban können Sie mit dem folgenden Kommando feststellen: **Status**

```

root# fail2ban-client status sshd
Status for the jail: sshd
  filter
    File list: /var/log/auth.log
    Currently failed: 0
    Total failed:    6456
  action
    Currently banned: 2
    - IP list:        177.n.n.n
                      43.n.n.n
    Total banned:    830

```

Ein Protokoll mit allen IP-Adressen, die aktuell oder in der Vergangenheit blockiert worden sind, liefert die Datei `/var/log/fail2ban.log`.

### 31.4 Authentifizierung mit Schlüsseln

#### Schlüssel erzeugen

Am sichersten ist die Verwendung des SSH-Servers, wenn Sie sich nicht mit einem Passwort authentifizieren, sondern mit einem Schlüssel. Dazu erzeugen Sie auf dem lokalen Rechner mit `ssh-keygen` ein Schlüsselpaar. Diesen Schlüssel sollten Sie durch eine Passphrase selbst verschlüsseln. Als *Passphrase* wird ein besonders langes, oft aus mehreren Wörtern bestehendes Passwort bezeichnet. Anschließend fügen Sie – noch per Passwort-Authentifizierung – den öffentlichen Schlüssel mit dem Kommando `ssh-copy-id` in die Datei `.ssh/authorized_keys` auf dem Server ein:

```

user@client$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa): <Return>
Enter passphrase (empty for no passphrase): ******
Enter same passphrase again: ******
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
user@client$ ssh-copy-id -i user@server
user@server's password: ******

```

Wenn Sie die Passphrase-Frage einfach mit  oder mit der Eingabe `empty` beantworten, verzichtet `ssh-keygen` auf die Verschlüsselung. Das ist bequem, weil es eine SSH-Nutzung ohne Passwort-Rückfrage ermöglicht. Sie gehen damit aber ein Sicherheitsrisiko ein: Jeder, dem Ihr Schlüssel auf dem Client-Rechner in die Hände gerät, kann sich ohne Weiteres auf allen Rechnern anmelden, auf denen Sie den öffentlichen Teil des Schlüssels installiert haben!

Das Kommando `ssh-copy-id` scheitert, wenn der Server nur eine Authentifizierung per Schlüssel zulässt. Zur Lösung dieses Henne-Ei-Problems müssen Sie die Schlüsselübertragung von einem anderen Client aus durchführen, der sich beim Server

authentifizieren kann. Sie müssen in diesem Fall Ihre öffentliche Schlüsseldatei `.ssh/id_rsa.pub` über einen dritten Rechner zum Server übertragen und dort manuell am Ende der Datei `.ssh/authorized_keys` einfügen.

Wenn Sie nun eine Verbindung zum Zielrechner erstellen, tauscht `ssh` die Schlüsselinformationen aus. Ein Login ist nicht mehr erforderlich, Sie müssen aber die Passphrase der privaten Schlüsseldatei eingeben.

Sofern Ihre Schlüssel durch ein Passwort bzw. eine Passphrase abgesichert sind, haben Sie durch die Verwendung von Schlüsseln zwar an Sicherheit, aber nicht an Komfort gewonnen. Eine sichere und doch einigermaßen bequeme Lösung bietet `ssh-agent`. Dieses Programm verwaltet alle privaten Schlüssel des Anwenders. Das Programm wird folgendermaßen gestartet:

```
user$ eval $(ssh-agent)
```

Dadurch werden einige Umgebungsvariablen der aktuellen Konsole geändert. `ssh-agent` läuft als Hintergrundprozess weiter. Durch `ssh-add` können Sie nun Ihre privaten Schlüssel hinzufügen:

```
user$ ssh-add ~/.ssh/id_rsa
Enter passphrase for /home/user/.ssh/id_rsa: *****
```

Von nun an verwendet `ssh` die von `ssh-agent` verwalteten Schlüsseldateien. Das heißt, Sie werden nie mehr nach dem Passwort für die Schlüsseldatei gefragt. Anstatt das Passwort bei jedem `ssh`-Kommando einzugeben, ist die Eingabe nur noch einmal erforderlich. Der große Nachteil von `ssh-agent` besteht darin, dass die Wirkung der Umgebungsvariablen auf eine einzige Konsole beschränkt ist.

Unter Gnome kümmert sich standardmäßig der `gnome-keyring-daemon` um Passwörter und SSH-Schlüssel. Der erstmalige Zugriff auf die Daten erfordert zumeist einen richtigen Login (vermeiden Sie Auto-Logins durch den Display-Manager!) oder die Angabe des Master-Passworts. Zur Administration der gespeicherten Schlüssel und Passwörter verwenden Sie das Programm `seahorse`.

Nicht immer funktioniert der passwortfreie Login beim SSH-Server auf Anhieb. Schuld sind diverse Sicherheitseinstellungen. Falls `sshd_config` die Einstellung `StrictModes yes` enthält, wird die Datei `authorized_keys` nur berücksichtigt, wenn deren Zugriffsrechte sehr restriktiv eingestellt sind. Wurde die Schlüsseldatei von `ssh-copy-id` erstellt, ist das zumeist nicht der Fall. Abhilfe schaffen die beiden folgenden Kommandos:

```
user@server$ chmod 700 ~/.ssh
user@server$ chmod 600 ~/.ssh/authorized_keys
```

Bei Fedora-, CentOS- und RHEL-Servern tritt häufig ein weiteres Problem auf: Der von `ssh-copy-id` erzeugten Schlüsseldatei fehlen die erforderlichen SELinux-Kontextinformationen. Abhilfe:

```
root@server# /sbin/restorecon -r /root/.ssh      (für root)
root@server# /sbin/restorecon -r /home/user/.ssh (für andere Benutzer)
```

### Fehlersuche

Wenn der SSH-Login nicht wie erwartet funktioniert, führen Sie das `ssh`-Kommando mit der zusätzlichen Option `-v` aus. `ssh` liefert dann eine Menge Debugging-Meldungen. Die Option kann bis zu dreimal angegeben werden, also `-v -v -v`. `ssh` protokolliert dann entsprechend detaillierter.

#### Passwort-Login ganz abstellen

Sobald der Aufbau einer SSH-Verbindung mit dem Schlüssel funktioniert und keine Login-Aufforderung mehr erscheint, können Sie sich überlegen, den Passwort-Login ganz zu deaktivieren. Dazu verändern Sie auf dem Server die Konfigurationsdatei `sshd_config`. Entscheidend sind zwei Zeilen:

```
# in /etc/ssh/sshd_config
...
PasswordAuthentication no
UsePAM                  no
```

Damit ist von nun an eine SSH-Authentifizierung *nur* noch mit Schlüsseln möglich. Passen Sie aber auf, dass Sie sich nicht selbst aus Ihrem System aussperren! Wenn Sie den Schlüssel auf Ihrem Client-Rechner verlieren, beispielsweise weil Ihnen Ihr Notebook gestohlen wird und Sie kein Backup haben, können Sie sich auf dem Server nicht mehr einloggen! Es ist wie so oft: Jede zusätzliche Sicherheit bezahlen Sie durch geringere Flexibilität ...

#### Mehrere Schlüssel

Bis jetzt bin ich davon ausgegangen, dass Sie nur einen einzigen, selbst erzeugten Schlüssel verwenden. Aber vielleicht haben Sie mehrere Schlüssel, die Sie in der Vergangenheit auf unterschiedlichen Rechnern eingesetzt haben, oder Sie wollen unterschiedliche Schlüssel für unterschiedliche externe Server verwenden oder es übergibt Ihnen jemand eine Schlüsseldatei, damit Sie einen weiteren Rechner administrieren können.

Die einfachste Lösung besteht darin, auf dem Client-Rechner im Verzeichnis `.ssh` die Datei `config` einzurichten. Vergessen Sie `chmod 600` nicht! Diese Datei enthält mit dem Schlüsselwort `IdentityFile` Referenzen auf alle Schlüsseldateien, beispielsweise so:

```
# Datei .ssh/config
IdentityFile ~/.ssh/id_rsa
IdentityFile ~/.ssh/id_rsa.my-other-server
IdentityFile ~/.ssh/id_rsa.fh-xen-server
...
```

In dieser Konfiguration testet das `ssh`-Kommando einfach alle Schlüssel, bis einer passt. Sie können der `config`-Datei noch mehr »Intelligenz« verleihen, wenn Sie die Einträge nach Hosts gruppieren. Damit können Sie für jeden Host zusätzlich angeben, welcher Port, welcher Benutzer etc. verwendet werden soll. Die folgenden Zeilen veranschaulichen die Syntax; mehr Details finden Sie in `man ssh_config`.

```
# Datei .ssh/config
Host kofler.info
  IdentityFile ~/.ssh/id_rsa
  User strenggeheim
Host my-other-server.com
  IdentityFile ~/.ssh/id_rsa.my-other-server
  User otheradmin
  Port 22022
```

## 31.5 Zusatzwerkzeuge

Für intensive SSH-Nutzer existieren diverse Zusatzwerkzeuge, um die Nutzung von SSH effizienter oder sicherer zu gestalten. Einige dieser Werkzeuge möchte ich Ihnen hier ganz kurz vorstellen.

### Cluster SSH

Cluster SSH – oder kurz CSSH – ist ein Perl-Script, mit dem Sie mehrere SSH-Verbindungen initiieren können. Für jede Verbindung wird ein eigenes Xterm-Fenster geöffnet, wobei die Textfarbe je nach Server variiert. Die Farbe ist nicht einstellbar, bleibt aber für einen Server immer gleich. Wenn Sie CSSH eine Weile benutzen, können Sie allein anhand der Farben zuordnen, auf welchem Server Sie gerade arbeiten.

Der eigentliche Clou von CSSH besteht aber darin, dass Sie Kommandos nicht nur direkt in den Xterm-Fenstern eintippen können, sondern auch in einer Texteingabebox des winzigen CSSH-Fensters. Diese Eingabe wird dann auf alle offenen Fenster vervielfacht. Wenn Sie also auf drei Ubuntu- oder Debian-Servern ein Update durchführen möchten, loggen Sie sich zuerst mittels CSSH auf allen drei Servern ein und geben dann im CSSH-Steuerungsfenster einmal `apt dist-upgrade` ein. Das Kommando wird damit auf allen drei Servern ausgeführt.

Cluster SSH kann auf vielen Distributionen als Paket `clusterssh` installiert werden. Sollte das Paket in Ihrer Distribution fehlen, finden Sie Cluster SSH hier zum Download:

<https://sourceforge.net/projects/clusterssh>

## Parallel SSH

Die Python-Script-Sammlung Parallel SSH (Paketname `pssh`) hat gewisse Ähnlichkeiten zu `CSSH`. Das in Parallel SSH enthaltene Kommando `pssh` eignet sich ebenfalls dazu, ein Kommando auf vielen via SSH erreichbaren Servern auszuführen – parallel eben. Der entscheidende Unterschied zu `CSSH` besteht darin, dass Parallel SSH nicht zur interaktiven Nutzung, sondern zur automatisierten Script-Verarbeitung gedacht ist. Dazu speichern Sie zuerst die Hostnamen oder IP-Adressen in eine Textdatei und führen die gewünschten Kommandos dann via Parallel SSH auf allen Hosts aus:

```
user$ pssh -h hosts.txt apt-get update
```

Wenn Sie die Ausgaben von Kommandos speichern möchten, geben Sie mit `-o` ein Ausgabeverzeichnis an. Dort speichert `pssh` für jeden Host eine Datei mit dem entsprechenden Namen.

```
user% mkdir allfstabs
user$ pssh -h hosts.txt -o allfstabs cat /etc/fstab
```

Administrative Arbeiten erfordern bei `pssh` einen `root`-Login auf den Servern. Unter Ubuntu gelingt das standardmäßig nicht. Sie können aber mit `passwd` ein `root`-Passwort festlegen und danach mit `ssh-copy-id` den eigenen Schlüssel kopieren. `pssh` setzt voraus, dass eine Authentifizierung mit Schlüsseln möglich ist oder dass sich das Hintergrundprogramm `ssh-agent` um die Authentifizierung kümmert.

Das Paket `pssh` stellt auch die Kommandos `pscp` und `pnuke` zur Verfügung. Mit `pscp` können Sie eine Datei über viele Rechner verteilen. Das Zielverzeichnis muss dabei absolut angegeben werden. `pnuke` beendet auf allen Hosts einen Prozess durch `kill -9`. Anwendungsbeispiele zu `pssh`, `pscp` und `pnuke` finden Sie hier:

<http://linux-magazin.de/Ausgaben/2008/11/Und-wenn-ja-so-viele>

## Mosh

Es ist nahezu unmöglich, SSH während einer Zugfahrt zu benutzen: Ständige Verbindungsabbrüche zwingen Sie, sich immer wieder neu einzuloggen. Abhilfe schafft Mosh (*Mobile Shell*), eine SSH-Alternative zur Nutzung eigens für instabile Netzwerkverbindungen.

Bevor Sie `mosh` verwenden können, müssen Sie das gleichnamige Paket sowohl auf dem Server als auch auf dem Client installieren. Anschließend stellen Sie die Verbindung analog zu Verbindungen mit `ssh` her:

```
localhost$ mosh user@externalhost
```

mosh verwendet SSH für den Verbindungsaufbau und startet dann auf dem externen Host das Programm `mosh-server` mit den Rechten des Benutzers, zu dem Sie die Verbindung hergestellt haben. `mosh-server` läuft also anders als `sshd` nicht ständig als Dienst, sondern nur bei Bedarf. Die weitere Kommunikation erfolgt dann über das Protokoll UDP (nicht TCP) und über einen Port zwischen 60000 und 61000. Diese Ports dürfen also nicht durch eine Firewall blockiert sein.

Weitere Details zu `mosh` können Sie auf der Projektwebseite nachlesen:

<https://mosh.org>

### Login mit dem Google-Authenticator

Bei der Arbeit mit SSH sind zwei Authentifizierungsverfahren üblich: ein simples Passwort (flexibel, aber sicherheitstechnisch nicht perfekt) oder die Verwendung von Schlüsseln (sicherer, erfordert aber, dass Sie immer Ihr Notebook mit dem Schlüssel bei sich haben). Eine dritte Variante besteht darin, eine Zwei-Faktor-Authentifizierung einzurichten. Bei jedem Login müssen Sie zusätzlich zum regulären Passwort einen nur für kurze Zeit gültigen Einmal-Code eingeben. Dieser Code wird von einer Smartphone-App generiert.

Idee

Gut geeignet für derartige Zwecke ist die App *Google Authenticator*, die es kostenlos für iOS und Android gibt. Diese App arbeitet rein lokal. Weder die Schlüssel noch die Einmal-Codes werden auf einen Google-Server übertragen.

Auf dem abzusichernden Linux-Server installieren Sie je nach Distribution das Paket `google-authenticator` oder `libpam-google-authenticator`. Es enthält ein PAM-Modul zur Kontrolle der Einmal-Codes sowie das Kommando `google-authenticator` zum Einrichten der Authentifizierung. Dieses Kommando führen Sie für den Account aus, in dem Sie sich anmelden möchten. Es zeigt einen QR-Code an (siehe [Abbildung 31.1](#)), den Sie mit der Google-Authenticator-App Ihres Smartphones einscannen. Auf diese Weise richten Sie in der Smartphone-App ein neues Konto ein. Damit ist der App Ihr Konto bekannt, und sie beginnt sofort damit, sechsstellige Einmal-Codes zu erzeugen. Jeder Code ist 30 Sekunden lang gültig. Danach wird automatisch der nächste Code erzeugt.

Konfiguration

Das Kommando `google-authenticator` gibt außerdem fünf Emergency Scratch Codes aus. Auch dabei handelt es sich um Einmal-Codes, allerdings ohne zeitliche Begrenzung. Sollten Sie Ihr Smartphone nicht bei sich haben, können Sie sich zur Not mit einem dieser Codes einloggen (aber nur einmal!). Bevor die fünf Codes aufgebraucht sind, fügen Sie der Datei `.google_authenticator` mit einem Editor neue Codes hinzu.



Abbildung 31.1 Einrichtung von Google Authenticator

Damit das PAM-Modul verwendet wird, verändern Sie die folgenden beiden Konfigurationsdateien:

```

# Datei /etc/pam.d/sshd
# diese Zeile am Beginn der Datei hinzufügen
auth required pam_google_authenticator.so

# Datei /etc/ssh/sshd_config
# die vorhandene Einstellung ändern
ChallengeResponseAuthentication yes

```

Mit `systemctl restart sshd` starten Sie schließlich den SSH-Server neu.

**Anwendung** Ein Login beim Server sieht nun so aus:

```

user$ ssh user@hostname
Verification code: ***** (wird auf dem Smartphone angezeigt)
Password: ***** (das reguläre Login-Passwort)

```



### Achtung

Auf dem Server können sich nun nur noch Benutzer anmelden, die zuvor für ihr Konto `google-authenticator` ausgeführt haben. Beachten Sie, dass `google-authenticator` immer nur für *einen* Account gilt! Wenn Sie das Programm für `root` ausgeführt haben, gilt es nur für den `root`-SSH-Login, nicht für andere Benutzer.

Probieren Sie das Verfahren zuerst in einer virtuellen Maschine aus, nicht auf einem realen Server! Wenn Sie Google Authenticator für einen Server einrichten, auf den Sie keinen physischen Zugriff haben, sollten Sie während der gesamten Arbeiten bis zum Abschluss eines erfolgreichen Tests unbedingt in einer getrennten SSH-Session mit `root`-Rechten eingeloggt bleiben. So können Sie zur Not über diese Verbindung Reparaturarbeiten durchführen. Andernfalls kann es Ihnen passieren, dass Sie sich auf Ihrem eigenen Server nicht mehr anmelden können, falls bei der Konfiguration irgendetwas schiefgeht!

Die zusätzliche Sicherheit erkaufen Sie sich mit einer neuen Abhängigkeit: Wenn Sie Ihr Smartphone irgendwo liegen gelassen haben oder der Akku leer ist, dann gelingt der SSH-Login nur mit einem der Notfall-Codes.



# Kapitel 32

## Apache

In diesem Kapitel beschreibe ich, wie Sie Ihren eigenen Webserver aufsetzen. Im Mittelpunkt des Kapitels steht das Programm Apache und seine Basiskonfiguration inklusive der Verwendung der kostenlosen HTTPS-Schlüssel von *Let's Encrypt*. Darüber hinaus gehe ich auch auf einige beliebte Erweiterungen ein, unter anderem auf die Programmiersprache PHP und auf das Programm GoAccess zur Erstellung von Zugriffsstatistiken. Das Kapitel endet mit Informationen zu FTP – und der Empfehlung, auf einen FTP-Server möglichst zu verzichten.

Ich gehe in diesem Kapitel davon aus, dass Sie einen öffentlichen Webserver im Internet betreiben möchten. Grundsätzlich ist es natürlich auch möglich, Apache nur innerhalb eines LANs einzusetzen, beispielsweise als firmeninternes Kommunikationszentrum mit einem Wiki und Seiten zur Projektplanung, Zeiterfassung etc. In diesem Fall müssen Sie aber unbedingt sicherstellen, dass die hier gesammelten Daten tatsächlich intern bleiben und dass kein ungeschützter Webzugriff aus dem Internet möglich ist (siehe auch [Abschnitt 32.2](#), »Webverzeichnisse einrichten und absichern«).

Generell gilt: Dieses Kapitel ist lediglich eine Einführung in die Konfiguration von Apache und beschreibt bestenfalls ein Prozent der Schlüsselwörter zur Apache-Konfiguration. Der professionelle Einsatz von Apache setzt das Studium weiterführender Dokumentation voraus, sei es in Buchform oder aus dem Internet.

### 32.1 Apache

Apache ist der beliebteste Webserver der Open-Source-Welt. Im März 2017 liefen laut <https://netcraft.com> ca. 46 Prozent aller aktiven Websites unter Apache. Wird nur die Million der am meisten besuchten Websites betrachtet, beträgt der Marktanteil ca. 41 Prozent. Aktuelle Informationen sowie eine umfassende Dokumentation zu Apache finden Sie auf der Apache-Website:

<https://httpd.apache.org>

**Versionen und Alternativen** Die aktuelle Apache-Version ist 2.4.n. Version 2.2.n wird zwar ebenfalls noch gewartet, kommt aber nur noch auf Langzeit-Server-Installationen zum Einsatz. Aktuelle Linux-Distributionen verwenden durchwegs Apache 2.4, weswegen ich in diesem Buch auf Version 2.2 nicht mehr eingehe.

Zunehmend beliebt mit einem Marktanteil von rund 20 Prozent ist der Webserver `nginx`: Auch dabei handelt es sich um ein Open-Source-Programm, das speziell im Hinblick auf hohe Geschwindigkeit und Skalierbarkeit optimiert wurde.

**Installation** Eine typische Apache-Installation besteht aus zahlreichen zusammengehörenden Paketen: dem Server an sich, diversen Bibliotheken, Plugins, Programmiersprachen etc. Um Ihnen die Installation zu erleichtern, können Sie bei einigen Distributionen jeweils eine ganze Gruppe von Paketen zur Installation auswählen. Damit werden neben Apache auch die wichtigsten MySQL- und PHP-Pakete installiert.

```
root# taskel install web-server           (Debian)
root# yum groupinstall 'Web-Server'      (CentOS, RHEL)
root# dnf groupinstall 'Web-Server'      (Fedora)
root# zypper intall -t pattern lamp_server (SUSE)
user$ sudo apt install taskel           (Ubuntu)
user$ sudo taskel install lamp-server
```

### Das richtige Multi-Processing-Modul (MPM)

Apache stellt vier unterschiedliche Multi-Processing-Module zur Auswahl, nämlich `perchild`, `prefork`, `worker` und `event`. Diese Multi-Threading-Verfahren haben Einfluss darauf, wie effizient Apache mehrere Anfragen gleichzeitig verarbeiten kann. Beim Einrichten von Apache müssen Sie sich für eine dieser Varianten entscheiden, indem Sie das entsprechende `apache2-mpm-xxx`-Paket installieren.

Wenn Sie zusammen mit Apache die Programmiersprache PHP einsetzen möchten, ist das Verfahren `prefork` die sicherste Wahl. Bei den anderen Varianten sind Fehler aufgrund von nicht threadsicheren PHP-Bibliotheken möglich:

<https://php.net/manual/en/faq.installation.php>

**Start/Stop** Apache ist ein Dämon, der je nach Distribution explizit gestartet werden muss. Eine Zusammenfassung der erforderlichen Kommandos finden Sie in [Abschnitt 12.5](#), »Systemprozesse (Dämonen)«. Der Name des Init-Scripts variiert je nach Distribution: Er lautet `apache2` bei Debian, SUSE und Ubuntu bzw. `httpd` bei CentOS, Fedora und RHEL.

**Firewall** Unter CentOS, Fedora, RHEL und (open)SUSE blockiert die standardmäßig aktive Firewall den Zugriff auf den Webserver von außen. Sie können Apache also vorerst nur direkt auf dem Rechner ausprobieren, auf dem der Webserver läuft (`http://localhost`). Damit der Webserver auch von außen erreichbar ist, müssen Sie in der Firewall

Ausnahmeregeln für die Protokolle HTTP und HTTPS definieren, also für die Port-Nummern 80 und 443.

Unter SUSE verwenden Sie zur Firewall-Konfiguration am besten YaST. Unter CentOS/Fedora/RHEL können Sie wie folgt vorgehen: Sie stellen zuerst fest, welche Firewall-Zone für die Netzwerkschnittstelle zum Internet gilt (häufig `public`, hier aber `FedoraWorkstation`), und aktivieren dann für diese Zone Ausnahmeregeln:

```
root# firewall-cmd --get-zone-of-interface=enp0s3 (aktive Zone herausfinden)
FedoraWorkstation
root# firewall-cmd --permanent --zone=FedoraWorkstation --add-service=http
root# firewall-cmd --permanent --zone=FedoraWorkstation --add-service=https
root# firewall-cmd --reload
```

Alternative Verfahren zur Firewall-Konfiguration sowie eine Menge Hintergrundinformationen zu diesem Thema folgen in [Kapitel 37](#), »Firewalls«.

Der Service-Name für das Init-System variiert je nach Distribution. Aus Sicherheitsgründen wird der Webserver wie die meisten anderen Netzwerkdämonen nicht unter dem Account `root` ausgeführt, sondern unter einem anderen Account. Dessen Namen stellen Sie am einfachsten mit `ps axu` fest. [Tabelle 32.1](#) fasst zusammen, unter welchem Namen Apache dem Init-System bekannt ist, unter welchem Account das Programm läuft und wo sich standardmäßig die HTML-Dateien befinden.

Name und Account

| Distribution         | Prozessname           | Account  | DocumentRoot    |
|----------------------|-----------------------|----------|-----------------|
| Debian, Ubuntu       | apache2               | www-data | /var/www/html   |
| CentOS, Fedora, RHEL | httpd                 | apache   | /var/www/html   |
| SUSE                 | httpd-threadverfahren | wwwrun   | /srv/www/htdocs |

**Tabelle 32.1** Programmname, Account und DocumentRoot-Verzeichnis von Apache

Um zu testen, ob alles funktioniert, starten Sie auf dem lokalen Rechner einen Webbrowser und geben als Adresse `http://localhost/` oder `http://servername/` ein. Sie sollten nun eine Testseite des Webservers sehen (siehe [Abbildung 32.1](#)).

Test

Damit statt der Testseite die Startseite Ihres eigenen Webauftritts erscheint, müssen Sie Ihre HTML-Dateien in das Dokumentverzeichnis von Apache speichern. Auch dieses Verzeichnis ist distributionsabhängig (Schlüsselwort `DocumentRoot` in den Konfigurationsdateien, siehe [Tabelle 32.1](#)). Ihre HTML-Dateien müssen für den Account des Apache-Webservers lesbar sein!

Eigene HTML-Seiten

Wenn Sie unter Fedora oder RHEL arbeiten, müssen Sie außerdem darauf achten, dass alle HTML-Dateien mit dem SELinux-Attribut `httpd_sys_content_t` ausgestattet sind. Für Dateien innerhalb von `/var/www/html` erreichen Sie das am einfachsten durch das folgende Kommando:

SELinux

```
root# restorecon -R -v /var/www/html/*
```

Persönliches Exemplar für Uwe Hametner

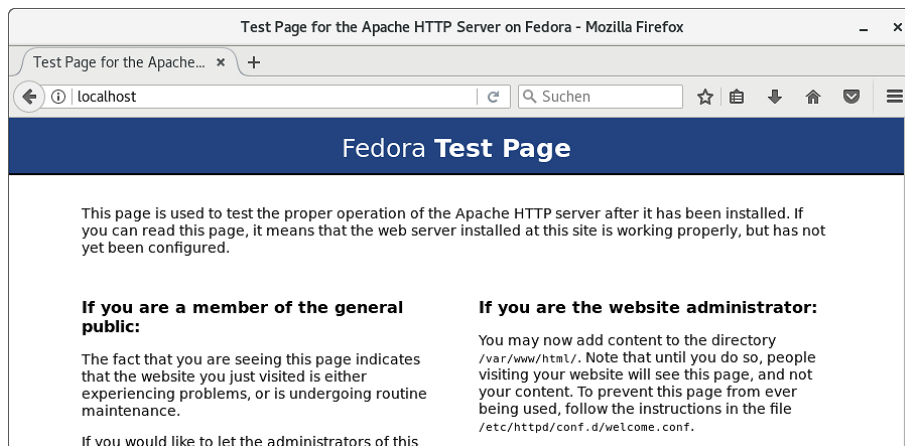


Abbildung 32.1 Apache-Testseite eines Fedora-Rechners

Wenn Sie Ihre HTML-Dateien in einem anderen Verzeichnis ablegen, ist hingegen das folgende Kommando erforderlich:

```
root# chcon -R system_u:object_r:httpd_sys_content_t:s0 /mein-web-verzeichnis
```

Beachten Sie, dass für CGI-, Webalizer- und Konfigurationsdateien andere Attribute vorgesehen sind. Details zum SELinux-Modul für Apache können Sie mit `man httpd_selinux` nachlesen, wenn Sie vorher das Paket `selinux-policy-doc` installieren.

## Konfiguration

In diesem Buch fehlt der Platz, um detailliert auf alle Konfigurationsoptionen und -varianten einzugehen. Ich möchte Ihnen an dieser Stelle aber zumindest einen Überblick darüber geben, wo sich die Konfigurationsdateien je nach Distribution befinden und wie ganz elementare Einstellungen durchgeführt werden.

Früher erfolgte die Konfiguration von Apache durch eine einzige Datei `httpd.conf`, wobei deren genauer Ort distributionsabhängig war. Diese Konfigurationsdatei wurde im Laufe der Zeit immer unübersichtlicher.

Aus diesem Grund sind die meisten Distributionen dazu übergegangen, die Einstellungen auf diverse Dateien zu verteilen, die durch `Include`-Anweisungen aus verschiedenen Verzeichnissen gelesen werden (siehe [Tabelle 32.2](#) bis [Tabelle 32.4](#)). Das macht jede einzelne Datei übersichtlicher und ermöglicht eine automatisierte Wartung – also beispielsweise das Aktivieren oder Deaktivieren von Plugins durch Kommandos oder Scripts.

| Dateien                           | Inhalt                                  |
|-----------------------------------|---|
| /etc/apache2/apache2.conf         | Startpunkt                              |
| /etc/apache2/httpd.conf           | benutzerspezifische Konfiguration       |
| /etc/apache2/ports.conf           | überwachte Ports, normalerweise Port 80 |
| /etc/apache2/conf.d/*             | weitere Konfigurationsdateien           |
| /etc/apache2/mods-available/      | verfügbare Erweiterungsmodule           |
| /etc/apache2/mods-enabled/*.conf  | Links auf aktive Erweiterungsmodule     |
| /etc/apache2/conf-available/      | verfügbare Konfigurationsdateien        |
| /etc/apache2/conf-enabled/*.conf  | Links auf aktive Konfigurationsdateien  |
| /etc/apache2/sites-available/     | verfügbare Websites (virtuelle Hosts)   |
| /etc/apache2/sites-enabled/*.conf | Links auf aktive Websites               |
| /etc/apache2/envvars              | Umgebungsvariablen für das Init-Script  |

**Tabelle 32.2** Apache-Konfiguration bei Debian und Ubuntu

| Dateien                    | Inhalt                            |
|----------------------------|-----------------------------------|
| /etc/httpd/conf/httpd.conf | Startpunkt                        |
| /etc/httpd/conf/magic      | MIME-Konfiguration (für mod_mime) |
| /etc/httpd/conf.d/*.conf   | sonstige Konfigurationsdateien    |

**Tabelle 32.3** Apache-Konfiguration bei CentOS, Fedora und Red Hat

| Dateien                       | Inhalt   |
|-------------------------------|--|
| /etc/apache2/httpd.conf       | Startpunkt                                       |
| /etc/apache2/*.conf           | globale Konfigurationsdateien                    |
| /etc/apache2/conf.d/*.conf    | sonstige Konfigurationsdateien                   |
| /etc/apache2/sysconf.d/*.conf | automatisch erzeugte Systemkonfigurationsdateien |
| /etc/apache2/vhosts.d/*.conf  | Websites (virtuelle Hosts)                       |
| /etc/sysconfig/apache2        | Grundeinstellungen                               |

**Tabelle 32.4** Apache-Konfiguration bei SUSE

Wenn Sie ein bestimmtes Schlüsselwort in den Konfigurationsdateien suchen, gehen Sie am besten so vor:

```
user$ cd /etc/httpd (bzw.) cd /etc/apache2
user$ grep -i -r Schlüsselwort
```

**Debian/Ubuntu** Bei Debian/Ubuntu enthält das Verzeichnis `mods-available` eine Kollektion von `*.load`- und `*.conf`-Dateien für diverse Apache-Module. Um weitere Module zu aktivieren, richten Sie in `mods-enabled` Links auf diese Dateien ein. Bei der Verwaltung der Links helfen die Debian-spezifischen Kommandos `a2enmod` und `a2dismod`.

Mit `a2ensite` und `a2dissite` aktivieren bzw. deaktivieren Sie virtuelle Hosts. Standardmäßig enthält `sites-available` nur die Dateien `000-default.conf` und `default-ssl.conf`: Dort befinden sich diverse Grundeinstellungen für das Verzeichnis `/var/www`. Der Mechanismus funktioniert wie bei den Modulen: Das Verzeichnis `sites-available` enthält die Konfigurationsdateien für alle Hosts, in `sites-enabled` befinden sich die entsprechenden Links.

Derselbe Mechanismus kümmert sich auch um sonstige Konfigurationsdateien. Die Dateien befinden sich im Verzeichnis `conf-available`. Mit den Kommandos `a2enconf` bzw. `a2disconf` werden im Verzeichnis `conf-enabled` entsprechende Links darauf eingerichtet bzw. wieder entfernt. Bei älteren Ubuntu-Versionen sowie bei Debian werden derartige Konfigurationsdateien in `conf.d` gespeichert. Die Verwaltung erfolgt manuell ohne Kommandos.

**SUSE** Bei SUSE werden sämtliche `*.conf`-Dateien im Verzeichnis `sysconf.d` bei jedem Apache-Start durch das Init-System neu erstellt. Es ist daher zwecklos, Änderungen an diesen Dateien vorzunehmen. Vielmehr müssen Sie die Variablen in `/etc/sysconfig/apache2` ändern. In dieser Datei ist auch festgelegt, welche Module beim Apache-Start geladen werden (Variable `APACHE_MODULES`). Wenn Sie den SUSE-Konfigurationsdateien eine eigene Datei hinzufügen möchten, geben Sie deren Dateinamen in der Variablen `APACHE_CONF_INCLUDE_FILES` an.

**Konfiguration testen** Nach Änderungen an der Syntax können Sie mit `httpd -t`, `httpd2 -t` bzw. `apache2 -t` testen, ob die Konfiguration frei von Syntaxfehlern ist. Bei Debian und Ubuntu müssen Sie vorher einige Umgebungsvariablen aus `envvars` einlesen:

```
root# . /etc/apache2/envvars
root# apache2 -t
Syntax OK
```

Anschließend fordern Sie Apache dazu auf, die Konfigurationsdateien neu einzulesen:

```
root# systemctl restart apache2|httpd
```



Der Webserver Apache funktioniert zwar im Regelfall auf Anhieb. Je nach Netzwerk-konfiguration müssen Sie aber oft eine Zeile in den Konfigurationsdateien ändern bzw. zu ihnen hinzufügen: `ServerName` sollte den Namen Ihres Rechners enthalten. Falls diese Einstellung nicht wirksam wird, müssen Sie außerdem die Einstellung `UseCanonicalName Off` verwenden.

ServerName

```
# in /etc/apache2/httpd.conf (Debian/Ubuntu)
# bzw. /etc/httpd/conf/httpd.conf (Fedora/Red Hat)
ServerName mars.sol # geben Sie hier den Namen Ihres Rechners an
```

Bei SUSE stellen Sie den Rechnernamen in `/etc/sysconfig/apache2` mit der Variablen `APACHE_SERVERNAME` ein.

Sofern der Root-Server über eine IPv6-Adresse verfügt, beantwortet Apache auch IPv6-Webanfragen. Dafür verantwortlich ist die Standardeinstellung `Listen 80`, mit der Apache den Port 80 überwacht, unabhängig von der IP-Version. Wenn Sie IPv6 deaktivieren möchten, fügen Sie die Anweisung `Listen 0.0.0.0:80` in die passende Konfigurationsdatei ein. Falls Apache auch HTTPS-Seiten liefern soll, benötigen Sie eine weitere `Listen`-Anweisung für den Port 443:

IPv6 blockieren

```
# Datei /etc/apache2/ports.conf (Debian, Ubuntu)
# Dateien /etc/httpd/conf/httpd.conf und conf.d/ssl.conf (CentOS, Fedora, RHEL)
# Datei /etc/apache2/listen.conf (SUSE)
Listen 0.0.0.0:80
Listen 0.0.0.0:443 https
```

## Standardzeichensatz

Bei allen gängigen Linux-Distributionen gilt automatisch der Unicode-Zeichensatz UTF-8. Wenn Sie also mit einem Texteditor eine Textdatei erstellen, die die deutschen Buchstaben ä, ö, ü oder ß enthält, werden diese in der UTF-8-Codierung gespeichert.

Apache ist die Codierung der HTML-Dateien grundsätzlich egal. Das Programm überträgt die Dateien einfach Byte für Byte an den Webbrowser, der die Seite angefordert hat. Allerdings sendet Apache zusätzlich einen sogenannten Header mit, der unter anderem Informationen darüber enthält, in welchem Zeichensatz die Seite codiert ist. Der Webbrowser wertet diese Information aus und verwendet den angegebenen Zeichensatz zur Darstellung der Seite.

Der springende Punkt ist nun, dass Apache den richtigen Zeichensatz angibt: Wenn das schiefgeht, sieht der Benutzer in seinem Webbrowser statt ä oder ü irgendwelche merkwürdigen Zeichenkombinationen. Aus diesem Grund bietet Apache diverse Möglichkeiten zur Zeichensatzkonfiguration:

Zeichensatz  
einstellen

- ▶ **AddDefaultCharset off:** Bei dieser Einstellung wertet Apache das `<meta>`-Tag in der zu übertragenden HTML-Datei aus und sendet den dort angegebenen Zeichensatz an den Browser. Wenn die HTML-Datei wie folgt beginnt, kommt der Zeichensatz Unicode UTF-8 zur Anwendung:

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    ...
```

- ▶ **AddDefaultCharset Zeichensatz:** Apache überträgt den hier angegebenen Zeichensatz für alle Seiten an den Browser. Die Einstellung gilt sowohl für HTML- als auch für PHP-Dateien. Das `<meta>`-Tag im HTML-Code wird ignoriert.
- ▶ **AddCharset Zeichensatz kennung:** Damit wird ein Zeichensatz für Dateien mit einer bestimmten Kennung eingestellt. `AddCharset utf-8 .utf8` bewirkt also, dass für alle Dateien, deren Name auf `.utf8` endet, als Zeichensatz Unicode UTF-8 an den Browser gesendet wird. `AddCharset` setzt das Apache-Modul `mod_mime` voraus.

**Debian, Ubuntu** Natürlich gilt je nach Distribution eine unterschiedliche Standardkonfiguration. Für die globale Voreinstellung des Zeichensatzes ist unter Debian und Ubuntu die Konfigurationsdatei `/etc/apache2/conf-enabled/charset.conf` vorgesehen. Normalerweise ist diese Datei leer, d. h., es gilt `AddDefaultCharset off`.

Sie können `AddDefaultCharset` und `AddCharset` auch in den Konfigurationsdateien für virtuelle Hosts (Verzeichnis `sites-available`) sowie in `.htaccess`-Dateien einsetzen, wenn Sie eine `host-` bzw. `verzeichnisspezifische` Konfiguration wünschen. Beachten Sie aber, dass die Zeichensatzeinstellungen in `.htaccess` nur berücksichtigt werden, wenn für das Webverzeichnis `AllowOverride All` oder `FileInfo` gilt.

**Fedora, Red Hat** Bei Fedora und Red Hat gilt `AddDefaultCharset UTF-8`. Die Einstellung befindet sich in `/etc/httpd/conf/httpd.conf`. In derselben Datei ist auch `AllowOverride None` für das Verzeichnis `/var/www/html` eingestellt.

**SUSE** Bei SUSE fehlt in den Konfigurationsdateien eine explizite Zeichensatzeinstellung. Damit gilt `AddDefaultCharset off`, d. h., die `<meta>`-Informationen in den HTML-Dateien sind für die richtige Zeichensatzerkennung entscheidend. Ein geeigneter Ort zur Einstellung von `AddDefaultCharset` ist die Datei `/etc/apache2/mod_mime-defaults.conf`. Auch bei SUSE gilt `AllowOverride None` für das Verzeichnis `/srv/www/htdocs`. Sie können die Einstellung in `/etc/apache2/default-server.conf` verändern.

## Logrotate

Die Logging-Dateien von Apache zählen bei vielen Servern zu den Dateien, die am schnellsten wachsen. Deswegen müssen Sie sich darum kümmern, dass die Logging-Dateien regelmäßig umbenannt, komprimiert und schließlich gelöscht werden. Genau diese Aufgabe erledigt das Programm Logrotate (siehe [Abschnitt 18.9](#), »Logging (Syslog)«), das auf Linux-Servern in der Regel standardmäßig installiert ist.

Das Programm wird üblicherweise einmal täglich durch `/etc/cron.daily/logrotate` gestartet. In der Standardkonfiguration verarbeitet es die Apache-Logging-Dateien `/var/log/httpd/*.log` (Fedora/RHEL) bzw. `/var/log/apache2/*.log` (Debian/Ubuntu) einmal pro Woche, benennt sie in `name.nn` um und komprimiert sie. Die komprimierten Dateien werden für 52 Wochen archiviert und dann gelöscht.

Falls Sie bei der Konfiguration virtueller Hosts eigene Logging-Verzeichnisse definieren, müssen Sie in der Konfigurationsdatei `/etc/logrotate.d/apache2` die erste Zeile anpassen und dort die Orte der zusätzlichen Logging-Dateien angeben. Dabei sind auch Muster wie `/home/*/www-log/*.log` erlaubt:

```
# Datei /etc/logrotate.d/apache2
/var/log/apache2/*.log /home/meinefirma/www-log/*.log {
    weekly
    missingok
    rotate 52
    ...
}
```

## 32.2 Webverzeichnisse einrichten und absichern

Nach der Grundkonfiguration von Apache werden Sie in der Regel verschiedene Webverzeichnisse einrichten, die jene HTML- und PHP-Dateien enthalten, aus denen sich Ihre Webseite zusammensetzt. Wenn Sie also beispielsweise WordPress als CMS für Ihre Webseite einrichten möchten, laden Sie die Installationsdateien herunter, richten ein für Apache erreichbares Verzeichnis ein und packen die Dateien dort aus. Dieser Abschnitt beschäftigt sich natürlich nicht mit den Details der WordPress-Installation, erläutert aber, welche Einstellungen Sie in Apache für das Verzeichnis vornehmen müssen, in dem Sie WordPress, phpMyAdmin, ownCloud/Nextcloud oder irgendeine andere Webapplikation einrichten möchten.

Auf den folgenden Seiten gehe ich dabei von der Apache-Standardkonfiguration aus, wie Sie sie unter Ubuntu bzw. Debian vorfinden. Wenn Sie mit einer anderen Distribution arbeiten, gibt es bei der Standardkonfiguration kleine Variationen. Die hier präsentierten Schlüsselwörter und Arbeitstechniken gelten aber auch dort.

**Ubuntu-Standard-  
konfiguration**

Unter Ubuntu ist Apache so vorkonfiguriert, dass für die Standard-Website Dateien aus dem Verzeichnis `/var/www` verwendet werden. Die erforderlichen Einstellungen befinden sich in der Datei `/etc/apache2/sites-available/000-default.conf`:

```
# Datei /etc/apache2/sites-available/000-default.conf (Ubuntu)
<VirtualHost *:80>
    ServerAdmin    webmaster@localhost
    DocumentRoot  /var/www/html
    ErrorLog       ${APACHE_LOG_DIR}/error.log
    CustomLog      ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Eine Debian- bzw. Ubuntu-spezifische Besonderheit der Defaultkonfiguration besteht darin, dass alle Einstellungen in einer `<VirtualHost>`-Gruppe gebündelt sind. `<VirtualHost>`-Gruppen dienen dazu, Einstellungen für mehrere eigenständige Hosts (Websites) voneinander zu trennen (siehe [Abschnitt 32.3](#), »Virtuelle Hosts«). Der Host in der Datei `default` ist allerdings weder an eine IP-Adresse noch an einen Hostnamen gekoppelt und gilt aus diesem Grund für alle Webzugriffe, die nicht einem speziellen virtuellen Host zugeordnet werden können.

### Host-Konfiguration

Mit den im Folgenden beschriebenen Schlüsselwörtern zur Konfiguration einer `<VirtualHost>`-Gruppe werden die Details des Hosts festgelegt – also die Herkunft der Daten, die E-Mail-Adresse des Administrators, der Ort der Logging-Dateien etc.:

- ▶ **DocumentRoot** gibt an, in welchem Verzeichnis sich die HTML-Dateien befinden.
- ▶ **ServerAdmin** gibt die E-Mail-Adresse des Administrators des virtuellen Hosts an. Die Adresse wird z. B. bei Fehlermeldungen angezeigt. Sie sollten hier eine E-Mail-Adresse angeben, die tatsächlich aktiv ist. Üblich ist `webmaster@hostname`.
- ▶ **ServerSignature** steuert, ob Apache bei selbst generierten Dokumenten (Fehlermeldungen, Verzeichnislisten etc.) am Ende eine Signatur hinzufügen soll. Die Signatur besteht aus der Apache-Version und dem Hostnamen. Mit `ServerSignature=EMail` wird auch die E-Mail-Adresse des Administrators hinzugefügt.
- ▶ **LogLevel** bestimmt, in welchem Ausmaß Webserver-Probleme protokolliert werden sollen. Mögliche Werte reichen von `emerg` (nur kritische Fehler protokollieren, die zum Ende von Apache führen) bis `debug` (alles protokollieren, selbst Debugging-Texte). Sinnvolle Einstellungen sind in der Regel `error` oder `warn`. Letztere Einstellung gilt per Default.
- ▶ **ErrorLog** gibt den Dateinamen der Protokolldatei für Fehlermeldungen an.

- **CustomLog** gibt den Dateinamen des Zugriffsprotokolls an. In dieser Datei protokolliert Apache jede erfolgreiche Übertragung einer Datei. An den zweiten Parameter übergeben Sie entweder den Namen eines vordefinierten Loggingformats oder eine Zeichenkette mit eigenen Formatanweisungen. Die erlaubten Formatcodes sind hier beschrieben:

[http://httpd.apache.org/docs/2.4/de/mod/mod\\_log\\_config.html](http://httpd.apache.org/docs/2.4/de/mod/mod_log_config.html)

Unter Ubuntu sind in `apache2.conf` einige Formate vorkonfiguriert, z. B. `combined` oder `common`.

- **ErrorDocument** gibt an, wie Apache auf Fehler reagieren soll. Als ersten Parameter geben Sie die Fehlernummer an (z. B. 404 für *not found*), im zweiten Parameter den Namen einer lokalen Datei bzw. die Adresse einer externen Seite, die in diesem Fall angezeigt werden soll. Der Dateiname muss relativ zu `DocumentRoot` angegeben werden. Die wichtigsten Fehlercodes sind:

```
400    Bad Request
401    Authorization Required
403    Forbidden
404    Not Found
500    Internal Server Error
```

Eine Liste aller HTTP-Statuscodes finden Sie hier:

<https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

Standardmäßig ist `ErrorDocument` nicht konfiguriert. Um unschöne Fehlermeldungen zu vermeiden, sollten Sie sich die Mühe machen, eine Fehlerseite einzurichten und deren Ort mit `ErrorDocument` anzugeben.

- **Alias** stellt eine Zuordnung zwischen einem Webverzeichnis und einem Verzeichnis der Festplatte (auch außerhalb von `DocumentRoot`) her. Beispielsweise bewirkt `Alias /mytool /usr/local/mytool`, dass bei Zugriffen auf `http://meinserver.de/mytool` die Dateien aus dem Verzeichnis `/usr/local/mytool` gelesen werden.

In der Regel müssen Sie für jedes `alias`-Verzeichnis in einer `<Directory>`-Gruppe die Zugriffsrechte einstellen (siehe den folgenden Abschnitt). Zu `Alias` gibt es die Variante `ScriptAlias`, die zur Definition von Verzeichnissen mit CGI-Scripts dient.

## Verzeichniskonfiguration

Im Anschluss an diese Einstellungen, die für den gesamten virtuellen Host gelten, können Sie in einer oder mehreren `<Directory "/verzeichnis/">`-Gruppen die Eigenschaften für einzelne Verzeichnisse Ihres Hosts einstellen. Die folgende Liste nennt hierfür nur die wichtigen Schlüsselwörter:

- ▶ **DirectoryIndex** gibt an, welche Datei Apache senden soll, wenn eine Adresse mit / endet und somit ein ganzes Verzeichnis betrifft (standardmäßig `index.html`). Es dürfen auch mehrere Dateien angegeben werden. In diesem Fall arbeitet Apache alle Angaben der Reihe nach bis zum ersten Treffer ab (z. B. `DirectoryIndex index.php index.html`).
- ▶ **Options** ermöglicht die Angabe diverser Optionen, die für das Verzeichnis gelten. Dazu zählen:

|                |  |
|----------------|--|
| ExecCGI        | CGI-Scripts ausführen  |
| FollowSymLinks | symbolische Links verfolgen                                      |
| Includes       | Include-Dateien hinzufügen (Modul <code>mod_include</code> )     |
| Indexes        | Dateiliste anzeigen, wenn <code>index.html</code> fehlt          |
| MultiViews     | automatische Sprachauswahl (Modul <code>mod_negotiation</code> ) |

Standardmäßig gilt in Apache die Einstellung `All`. Damit sind alle Optionen mit der Ausnahme von `MultiViews` aktiv. Die Ubuntu-Konfiguration ist etwas restriktiver: Für das gesamte Dateisystem gilt `Options FollowSymLinks`, für das `/var/www`-Verzeichnis gilt `Options Indexes FollowSymLinks MultiViews`.

Um einzelne Optionen gegenüber den Voreinstellungen eines übergeordneten Verzeichnisses wieder zu deaktivieren, muss ein Minuszeichen vorangestellt werden. Ein vorangestelltes Pluszeichen ist ebenfalls erlaubt, hat aber keine Wirkung: Die Option ist genau so aktiviert, als wäre sie ohne Pluszeichen angegeben.

Aus Sicherheitsgründen sollte für `Options` die Devise »Weniger ist mehr« gelten: Die Option `Indexes` verrät neugierigen Websurfern die Namen aller Dateien, die sich in einem Verzeichnis befinden, sofern Sie einmal `index.html` vergessen. Das ist ein potenzielles Sicherheitsrisiko. `MultiView` brauchen Sie nur für mehrsprachige Websites mit automatischer Sprachauswahl. Bietet Ihre Seite so etwas nicht, können Sie auch auf diese Option verzichten.

- ▶ **AllowOverride** gibt an, welche Einstellungen verzeichnisspezifisch durch eine `.htaccess`-Datei verändert werden dürfen. Zur Auswahl stehen:

|            |   |
|------------|---|
| AuthConfig | Authentifizierungsverfahren einstellen  |
| FileInfo   | Datei- und Dokumenttypen einstellen   |
| Indexes    | Verzeichnisindex modifizieren   |
| Limit      | Zugriffsrechte ändern ( <code>Allow</code> , <code>Deny</code> , <code>Order</code> ) |
| Options    | Verzeichnisoptionen ändern  |

Standardmäßig sind in Apache alle Möglichkeiten aktiv, d. h., jede Option kann verändert werden. Bei den meisten Distributionen ist die Standardkonfiguration aus Sicherheitsgründen aber restriktiver. So ist unter Debian und Ubuntu für alle relevanten Verzeichnisse `None` voreingestellt (siehe `/etc/apache2/apache2.conf`).

## Verzeichnisse absichern

Auf einen zentralen Punkt bin ich bisher noch nicht eingegangen: auf die Steuerung der Zugriffsrechte für Verzeichnisse. Die Apache-Versionen 2.2 und 2.4 unterscheiden sich hierbei deutlich. Da die Version-2.2-Syntax auf vielen Apache-2.4-Installationen weiterhin funktioniert und immer noch weitverbreitet ist, gehe ich hier auf beide Varianten ein. Für Neuinstallationen sollten Sie aber unbedingt die Schlüsselwörter von Apache 2.4 verwenden!

In Apache 2.2 können Sie innerhalb der `<Directory>`-Gruppe mit `Order`, `Allow` und `Deny` einstellen, unter welchen Umständen Apache Dateien aus dem jeweiligen Verzeichnis lesen und weitergeben darf.

Zugriffsrechte für Verzeichnisse (Apache 2.2)

Zugriffsregeln gelten auch für alle Unterverzeichnisse, sofern nicht explizit in einer weiteren `<Directory>`-Gruppe andere Regeln definiert werden. Die Zugriffsregeln für das Verzeichnis / geben daher Standardregeln für das gesamte Dateisystem vor!

- ▶ **Order Allow,Deny** bedeutet, dass zuerst alle `Allow`- und dann alle `Deny`-Regeln ausgewertet werden. Wenn auf einen Seitenzugriff keine Regel angewendet werden kann, wird der Zugriff blockiert.
- ▶ **Order Deny,Allow** dreht die Reihenfolge der Regeln um. Beachten Sie aber: Wenn bei einem Seitenzugriff keine Regel passt, ist der Zugriff erlaubt! Diese Regel gilt in Apache standardmäßig.
- ▶ **Allow from** gibt an, von welchen Hostnamen bzw. IP-Adressen Zugriffe erlaubt sind – also beispielsweise `Allow from 213.214.215.216` `bekannteseite.de`. IP-Adressbereiche können Sie in der Form `213.214` oder `213.214.0.0/255.255.0.0` oder `213.214.0.0/16` (für `213.214.*.*`) angeben. Bei Hostnamen gilt `site.de` auch für `www.site.de`, `sub.site.de` etc. Die Regel `Allow from all` erlaubt jeden Zugriff.
- ▶ **Deny from** funktioniert gerade umgekehrt und blockiert den Zugriff für die angegebenen Hosts bzw. Adressen.

Per Default gilt `Order Deny,Allow`, und mangels anderer Regeln ist somit der gesamte Zugriff auf alle Verzeichnisse blockiert! Wenn Sie Webdateien in anderen Verzeichnissen unterbringen, vergessen Sie nicht, den Zugriff darauf zu erlauben.

Unter Apache 2.4 gelten die drei Schlüsselwörter `Order`, `Allow` und `Deny` als obsolet. Die Schlüsselwörter werden aber weiterhin vom Modul `mod_access_compat` verarbeitet. Dieses Modul steht bei den meisten Apache-2.4-Installationen zur Verfügung und stellt sicher, dass ein Apache-2.4-Update nicht die gesamte bisherige Konfiguration über den Haufen wirft.

Zugriffsrechte für Verzeichnisse (Apache 2.4)

Bei einer Neukonfiguration wird der Einsatz des neuen Schlüsselworts `Require` empfohlen. Die folgenden Beispiele zeigen verschiedene Anwendungsformen:

```

# erlaubt den Zugriff vom Rechner mit der IP-Adresse 192.168.0.2
Require ip 192.168.0.2

# erlaubt den Zugriff aus dem Adressbereich 10.0.*.*
Require ip 10.0

# erlaubt den Zugriff aus einem IPv6-Adressbereich
Require ip 2001:1234:789a:0471::/64

# erlaubt den Zugriff für einen bestimmten Hostnamen
Require host intern.meine-firma.de

# erlaubt den Zugriff für *.meine-firma.de
Require host meine-firma.de

# erlaubt den Zugriff von localhost (IPv4 und IPv6)
Require local

# erlaubt den Zugriff für authentifizierte Benutzer
Require valid-user

# erlaubt den Zugriff von überall
Require all granted

# blockiert jeden Zugriff
Require all denied

```

Wenn Sie für ein `<Directory>` mehrere Bedingungen formulieren, dann reicht es, wenn *eine* dieser Bedingungen erfüllt ist:

```

<Directory /var/www/cms>
  Require local
  Require ip 192.168
  Require host meine-firma.de
</Directory>

```

Mit `<RequireAll>` können Sie mehrere Bedingungen durch ein logisches Und kombinieren. Apache liefert die angeforderte Seite nur, wenn *alle* Bedingungen gleichzeitig zutreffen.

```

<Directory /var/www/internal-wiki>
  <RequireAll>
    Require valid-user
    Require ip 192.168.17
  </RequireAll>
</Directory>

```



Wenn Sie Apache zur firmeninternen Kommunikation einrichten, können Sie den Webzugriff auf das lokale Netzwerk beschränken. Wenn das lokale Netzwerk den Adressbereich 192.168.1.\* und die lokale Domain .sol nutzt, sieht die richtige Konfiguration für `/var/www` so aus:

Internet-Zugriff  
blockieren

```
# für Apache 2.4
<Directory /var/www/>
    Require local
</Directory>
```

Eine alternative Vorgehensweise besteht darin, mit `Listen` die IP-Adresse der lokalen Netzwerkschnittstelle anzugeben. Das setzt voraus, dass die IP-Adresse statisch ist. Nehmen wir an, der Server hat zwei Netzwerkschnittstellen: eine für die Verbindung in das Internet und eine zweite für das LAN mit der IP-Adresse 192.168.1.17. Dann bewirkt `Listen 192.168.1.17`, dass Apache nur noch auf Anfragen aus dem lokalen Netzwerk reagiert. `Listen` gilt allerdings für die gesamte Apache-Konfiguration, nicht nur für einzelne Verzeichnisse oder virtuelle Hosts. `Listen` funktioniert gleichermaßen unter Apache 2.2 und Apache 2.4.

Zugriffsschutz  
durch »Listen«

Eine dritte Variante ist die Verwendung einer Firewall: Die Firewall muss den Empfang von Paketen verweigern, die von außen (also aus dem Internet) kommen und an die Ports 80 und 443 (https) gerichtet sind. Die Verwendung einer Firewall ist generell eine gute Idee, weil sie vollkommen unabhängig von Apache funktioniert.

Zugriffsschutz  
durch eine  
Firewall

## Passwortschutz für Webverzeichnisse

Häufig sollen Webverzeichnisse nur nach einer Authentifizierung durch einen Benutzernamen und das dazugehörige Passwort freigegeben werden. Apache sieht hierfür ein einfaches Verfahren vor, das gleichermaßen in den Versionen 2.2 und 2.4 funktioniert.

Der erste Schritt hin zum Passwortschutz ist eine Passwortdatei. Die Datei sollte aus Sicherheitsgründen außerhalb aller Webverzeichnisse angelegt werden, um einen Zugriff per Webadresse auszuschließen. Das folgende Beispiel geht davon aus, dass die Passwortdatei im Verzeichnis `/var/www-private` gespeichert wird. Wenn Sie ein neues Verzeichnis einrichten, achten Sie darauf, dass Apache hierfür Leserechte hat. Unter CentOS/Fedora/RHEL müssen Sie auch SELinux im Auge haben und das Passwortverzeichnis entweder innerhalb von `/var/www` einrichten oder nach dem Erzeugen des Verzeichnisses den SELinux-Kontext korrekt einstellen.

Passwortdatei

Um eine neue Passwortdatei anzulegen, verwenden Sie das Kommando `htpasswd` mit der Option `-c` (*create*). Das Passwort wird selbstverständlich verschlüsselt.

```

root# cd /var/www-private
root# htpasswd -c passwords.pwd username
New password: *****
Re-type new password: *****
Adding password for user username

```

Weitere Benutzernamen/Passwort-Paare werden mit `htpasswd` ohne die Option `-c` hinzugefügt:

```

root# cd /var/www-private
root# htpasswd passwords.pwd name2
New password: *****
Re-type new password: *****
Adding password for user username

```

**Konfiguration** Es gibt nun zwei Varianten, um Apache so zu konfigurieren, dass die Passwortdatei tatsächlich berücksichtigt wird. Die erste Variante setzt voraus, dass Sie die Konfiguration direkt in einer Apache-Konfigurationsdatei durchführen, unter Debian oder Ubuntu also in `/etc/apache2/sites-available/default` für die Standard-Website des Servers bzw. in `.../sitename` für einen virtuellen Host. Bei der zweiten Variante erfolgt die Konfiguration in der Datei `.htaccess`, die sich innerhalb des Webverzeichnisses befindet.

Damit die Passwortdatei von Apache berücksichtigt wird, müssen Sie in die `<Directory>`-Gruppe diverse Authentifizierungsoptionen einfügen. Wenn es für das zu schützende Verzeichnis noch keine eigene `<Directory>`-Gruppe gibt, legen Sie eine neue Gruppe an. Dabei werden automatisch alle Optionen vom übergeordneten Verzeichnis übernommen. Sie müssen also nur die Authentifizierungsoptionen hinzufügen. Die folgenden Zeilen geben hierfür ein Muster:

```

# in /etc/apache2/sites-available/xxx (Debian/Ubuntu)
...
# passwortgeschütztes Verzeichnis
<Directory "/var/www/admin/">
    AuthType      Basic
    AuthUserFile  /var/www-private/passwords.pwd
    AuthName      "admin"
    Require       valid-user
</Directory>

```

Kurz eine Erklärung der Schlüsselwörter:

- ▶ **AuthType** gibt den Authentifizierungstyp an. Ich gehe hier nur auf den `Basic`-Typ ein.
- ▶ **AuthUserFile** gibt den Ort der Passwortdatei an.
- ▶ **AuthName** bezeichnet den Bereich (Realm), für den der Zugriff gültig ist. Der Sinn besteht darin, dass Sie nicht jedes Mal einen Login durchführen müssen, wenn

Sie auf unterschiedliche Verzeichnisse zugreifen möchten, die durch dieselbe Passwortdatei geschützt sind. Sobald Sie sich mit einer bestimmten AuthName-Bezeichnung eingeloggt haben, gilt dieser Login auch für alle anderen Verzeichnisse mit diesem AuthName.

- **Require valid-user** bedeutet, dass als Login jede gültige Kombination aus Benutzername und Passwort erlaubt ist. Alternativ können Sie hier auch angeben, dass ein Login nur für ganz bestimmte Benutzer erlaubt ist:

```
Require user name1 name2
```

Die oben skizzierte Vorgehensweise ist nur möglich, wenn Sie Zugang zu den Apache-Konfigurationsdateien haben, d. h., wenn Sie selbst der Webadministrator sind. Ist das nicht der Fall, kann eine gleichwertige Absicherung auch durch die Datei `.htaccess` erfolgen, die sich im zu schützenden Verzeichnis befindet. In dieser Datei müssen sich dieselben Anweisungen befinden, die vorhin innerhalb der `<Directory>`-Gruppe angegeben wurden, also `AuthType`, `AuthUserFile`, `AuthName` und `Require`.

#### **.htaccess erfordert AllowOverride AuthConfig**

`.htaccess`-Dateien werden nur beachtet, wenn innerhalb des Webverzeichnisses eine Veränderung der Authentifizierungsinformationen zulässig ist. Die (übergeordnete) `<Directory>`-Gruppe muss `AllowOverride AuthConfig` oder `AllowOverride All` enthalten.

## 32.3 Virtuelle Hosts

Für jede Website (für jeden Host) ein eigener Webserver – das wäre angesichts der Leistungsfähigkeit aktueller Rechner eine Verschwendung von Ressourcen! Mit Apache können Sie dank sogenannter virtueller Hosts viele Websites parallel einrichten. Solange die Gesamtzugriffszahlen nicht an die Limits des Rechners gehen, bemerkt kein Anwender, dass die Websites in Wirklichkeit alle auf demselben Rechner laufen.

Aus technischer Sicht gibt es drei Verfahren, wie Apache entscheidet, an welchen virtuellen Host eine Webanfrage gerichtet ist. Als Ausgangspunkt dient in jedem Fall der vom Browser an den Server übertragene HTTP-Header.

- **Namensbasierte virtuelle Hosts:** Apache erkennt die gewünschte Website anhand des im HTTP-Header enthaltenen Hostnamens. Diese Variante ist am einfachsten zu realisieren und am weitesten verbreitet.
- **IP-basierte virtuelle Hosts:** Apache erkennt die gewünschte Website anhand der IP-Adresse im Header. Diese Vorgehensweise ist mit einem gravierenden Nachteil verbunden: Jeder virtuelle Host erfordert eine eigene IP-Adresse, und IP-Adressen sind für IPv4 Mangelware.

- **Port-basierte virtuelle Hosts:** Apache erkennt aufgrund der Port-Nummer die gewünschte Website. Diese Variante ist in der Praxis unüblich, weil die Port-Nummer als Teil der Webadresse angegeben werden muss. Das sieht unübersichtlich aus und eignet sich bestenfalls für eine technisch versierte Zielgruppe, z. B. für Administratoren.

Ich beziehe mich in diesem Abschnitt wiederum auf die Defaultkonfiguration von Debian bzw. Ubuntu. Dort ist es üblich, für jeden virtuellen Host eine eigene Konfigurationsdatei zu verwenden.

Wenn Sie Ihren Webserver unter Fedora oder RHEL einrichten, kommen naturgemäß dieselben Apache-Schlüsselwörter zum Einsatz. Allerdings erfolgen sämtliche Einstellungen wahlweise in `/etc/httpd/conf/httpd.conf` oder in `/etc/httpd/conf.d/sitename.conf`.

### Virtuelle Hosts einrichten

**NameVirtualHost (Apache 2.2)** Apache 2.4 erkennt bei der Analyse der Konfigurationsdateien automatisch, dass namensbasierte virtuelle Hosts verwendet werden. Das aus Apache 2.2 vertraute Schlüsselwort `NameVirtualHost` ist nicht mehr erforderlich.

**Host-Dateien (Debian, Ubuntu)** Unter Debian und Ubuntu ist die Standard-Website in `/etc/apache2/sites-available/default` als virtueller Host definiert. Um einen neuen virtuellen Host zu definieren, legen Sie unter Debian oder Ubuntu eine neue Datei im Verzeichnis `/etc/apache2/sites-available/` an. Diese Datei sollte genau eine `<VirtualHost>`-Gruppe enthalten. Die drei folgenden Listings geben je ein Beispiel für einen namens-, IP- und port-basierten Host.

**Namensbasierte virtuelle Hosts** `ServerName` gibt den Namen des Hosts an. Dieser Hostname muss in den Header-Informationen einer Webanfrage enthalten sein, damit Apache darauf reagiert. Optional können Sie mit `ServerAlias` weitere Namen nennen. Beispielsweise empfiehlt sich zur Einstellung `ServerName www.meinserver.de` die Ergänzung `ServerAlias meinserver.de`, damit ein virtueller Host mit oder ohne die vorangestellten Buchstaben `www.` verwendet werden kann.

```
# /etc/apache2/sites-available/beispiel-named-host (Debian/Ubuntu)
<VirtualHost *:80>
    DocumentRoot /var/www/verzeichnis1/
    ServerName    www.firma-1.de
    ServerAlias   firma-1.de
    ...
</VirtualHost>
```

Bei IP- und port-basierten Hosts muss die IP-Adresse mit einer der IP-Adressen des Servers übereinstimmen:

IP- und  
port-basierte  
virtuelle Hosts

```
# /etc/apache2/sites-available/beispiel-IP-host (Debian/Ubuntu)
<VirtualHost 213.214.215.216:80>
  DocumentRoot /var/www/verzeichnis2/
  ServerName www.firma-2.com
  ...
</VirtualHost>

# /etc/apache2/sites-available/beispiel-port-host (Debian/Ubuntu)
<VirtualHost 213.214.215.216:12001>
  DocumentRoot /var/www/verzeichnis3/
  ServerName www.admin-firma3.de
  ...
</VirtualHost>
```

#### Vergessen Sie nicht, Zusatzports mit »Listen« anzugeben!

Die Adress- und Port-Angaben in `<VirtualHost>` haben keinen Einfluss darauf, welche IP-Adressen und Ports Apache überwacht. Bei den meisten Distributionen sind nur die Ports 80 und 443 (https) vorgesehen (also `Listen 80` und `Listen 443`).

Wenn Apache weitere Ports überwachen soll, müssen Sie die Apache-Konfiguration entsprechend erweitern. Weitere Informationen zur Konfiguration für virtuelle Hosts finden Sie hier:

<https://httpd.apache.org/docs/2.4/de/vhosts>

Um einen virtuellen Host zu aktivieren bzw. später wieder zu deaktivieren, führen Sie nun unter Debian/Ubuntu `a2ensite name` bzw. `a2dissite name` aus und fordern Apache dann zum Neuladen der Konfigurationsdateien auf:

Virtuelle Hosts  
aktivieren

```
root# a2ensite beispiel-named-host (Debian/Ubuntu)
root# systemctl reload apache2
```

Theoretisch ist es möglich, mit `a2dissite` auch die Standard-Website des Servers zu deaktivieren. Das sollten Sie aber nicht tun, weil die Datei `/etc/apache2/sites-available/000-default.conf` diverse Standardeinstellungen für Apache enthält!

Sobald Sie virtuelle Hosts eingerichtet haben, wird die in `sites-available/default` definierte Standard-Website nur noch angezeigt, wenn Webanfragen für keine der virtuellen Hosts zutreffen.

Unter CentOS, Fedora, RHEL und SUSE entfallen die Kommandos `a2ensite/a2dissite`, weil sich alle Angaben zu den virtuellen Hosts in der zentralen Konfigurationsdatei `httpd.conf` oder in eigenen Dateien im Verzeichnis `conf.d` befinden. Dort durchgeführte Änderungen aktivieren Sie mit dem folgenden Kommando:

```
root# systemctl reload httpd          (CentOS, Fedora, RHEL)
root# systemctl reload apache2       (SUSE)
```

## Beispiel

Dieser Abschnitt beschreibt, wie Sie auf einem Debian- oder Ubuntu-Server den neuen virtuellen Host `firma-123.de` einrichten – zusammen mit einem neuen Login `firma123`, sodass Ihr Kunde, Freund etc. den virtuellen Host selbst administrieren kann. Dabei gehe ich davon aus, dass die Web- und Logdateien des virtuellen Hosts innerhalb des Heimatverzeichnisses des neuen Benutzers `firma123` angeordnet werden. Ebenso gut ist es möglich, zu diesem Zweck ein neues Verzeichnis `/var/www-firma123` einzurichten und dem Benutzer hierfür Schreibrechte zu geben.

Der erste Schritt besteht darin, einen neuen Account einzurichten, ein Passwort zuzuweisen und die erforderlichen Verzeichnisse zu erzeugen. In den folgenden Kommandos müssen Sie natürlich `firma123` durch den tatsächlichen Benutzernamen ersetzen!

```
root# adduser firma123
root# passwd firma123
Enter new UNIX password: *******
Retype new UNIX password: *******
passwd: password updated successfully
root# mkdir ~firma123/www
root# chown firma123:firma123 ~firma123/www
root# mkdir ~firma123/www-log
root# chown root:root ~firma123/www-log
root# chmod go-w ~firma123/www-log
```

Im zweiten Schritt erzeugen Sie eine neue Datei im Verzeichnis `sites-available`, die so ähnlich wie das folgende Muster aufgebaut ist. Abermals müssen Sie `firma123` durch den tatsächlichen Benutzernamen ersetzen und außerdem statt `firma-123.de` den tatsächlichen Hostnamen angeben. Mit `AllowOverride AuthConfig File` geben Sie Ihrem Kunden relativ weitreichende Möglichkeiten, die Konfiguration der Website durch eine `.htaccess`-Datei anzupassen. Wenn Sie das nicht möchten, müssen Sie diverse Konfigurationsdetails absprechen und `fix` einstellen.

```
# /etc/apache2/sites-available/firma-123.de
<VirtualHost * >
    DocumentRoot /home/firma123/www/
    ServerName firma-123.de
```

```

ServerAlias    www.firma-123.de
ErrorLog       /home/firma123/www-log/error.log
CustomLog      /home/firma123/www-log/access.log combined
ServerAdmin    webmaster@firma-123.de
ErrorDocument  404 /not-found.html
<Directory    "/home/firma123/www/" >
    AllowOverride AuthConfig File
</Directory>
</VirtualHost>

```

Zur Aktivierung der Website führen Sie die folgenden Kommandos aus:

```

root# a2ensite firma-123.de
root# systemctl reload apache2

```

Ihr Kunde muss nun nur noch die DNS-Konfiguration seiner Domain anpassen: Die zugeordnete IP-Adresse muss mit der Ihres Servers übereinstimmen. Sobald das der Fall ist, beantwortet Ihr Webserver alle Anfragen, die an *www.firma-123.de* gerichtet sind.

Wie ich bereits erwähnt habe, läuft Apache aus Sicherheitsgründen nicht mit `root`-Rechten, sondern unter einem Account mit eingeschränkten Rechten (`www-data` bei Debian/Ubuntu, `apache` bei Fedora/RHEL bzw. `wwwrun` bei SUSE). Stellen Sie die Zugriffsrechte der Webdateien so ein, dass Apache sie lesen kann!

**Zugriffsrechte**

Wenn Apache einzelne Dateien auch verändern soll (z. B. über ein PHP-Script), ordnen Sie den Verzeichnissen und Dateien die Gruppe `www-data/apache/wwwrun` zu und geben den Gruppenmitgliedern Schreibrechte (`chmod g+w`). Unter Fedora und RHEL müssen Sie außerdem den SELinux-Kontext korrekt einstellen.

Im obigen Beispiel werden alle Fehler- und Zugriffsmeldungen in eigenen Dateien für den virtuellen Host gespeichert. Diese Vorgehensweise erleichtert die Auswertung der Logging-Dateien. Allerdings ist die Anzahl der offenen Datei-Handles für Apache (wie für jeden anderen Linux-Prozess) beschränkt. Wenn Sie sehr viele virtuelle Hosts einrichten, müssen Sie alle Zugriffe in einer zentralen Datei protokollieren und diese Datei dann durch ein anderes Programm in kleinere Dateien je nach Host zerlegen. Weitere Informationen zu diesem Thema finden Sie hier:

**Logging**

<https://httpd.apache.org/docs/2.4/vhosts/fd-limits.html>

Virtuelle Hosts setzen voraus, dass die DNS-Konfiguration stimmt! Um die im vorigen Abschnitt beschriebene Website *firma-123.de* zu testen, muss der DNS-Eintrag der Domain *firma-123.de* auf die IP-Adresse Ihres Webserver zeigen. Änderungen am DNS-Eintrag kann nur der Eigentümer der Domain durchführen. Die meisten Domain-Händler bieten dazu entsprechende Werkzeuge an. Beachten Sie, dass DNS-Änderungen nicht sofort gelten. Die Synchronisation der vielen, weltweit verteilten Nameserver kann etliche Stunden dauern, auch wenn es oft schneller geht.

**Test**

Bei Serverumbauten oder -umzügen besteht oft der Wunsch, den neuen Server zuerst in Ruhe zu testen, bevor die DNS-Änderung tatsächlich durchgeführt wird. Der einfachste Weg besteht darin, den neuen virtuellen Host anfänglich nicht namensbasiert, sondern port-basiert zu konfigurieren. Dazu entfernen Sie die `ServerName`- und `ServerAlias`-Anweisungen und geben im `<VirtualHost>`-Tag statt des Sterns die IP-Adresse des Servers sowie eine freie Port-Nummer an, beispielsweise so:

```
<VirtualHost 213.214.215.216:12001>
```

Standardmäßig verarbeitet Apache nur Anfragen, die an die Ports 80 und 443 (für https) gerichtet sind. Damit Apache auch den hier eingesetzten Port 12001 berücksichtigt, müssen Sie in `/etc/apache2/ports.conf` eine weitere Zeile mit `Listen 12001` einfügen. Nun ist noch das Kommando `systemctl reload apache2` erforderlich, damit Apache die veränderte Konfiguration berücksichtigt. Jetzt können Sie den neuen Webauftritt mit Ihrem Webbrowser testen, indem Sie die IP-Adresse des Servers samt der Port-Nummer 12001 angeben, also beispielsweise `http://213.214.215.216:12001`.

**/etc/hosts zum Testen ändern**

Ganz anders können Sie vorgehen, wenn sich bei einem Server-Umzug inklusive Wechsel auch die IP-Adresse ändert: In diesem Fall können Sie in Ruhe den neuen Server einrichten. Für Ihre Kunden ist der neue Server noch nicht sichtbar, weil Ihr DNS-Eintrag ja noch auf den alten Server verweist. Um den neuen Server aber schon jetzt selbst unter dem richtigen Hostnamen zu testen, können Sie auf Ihrem lokalen Linux-Rechner zu Hause (also nicht auf dem Server mit Apache!) vorübergehend den folgenden Eintrag in `/etc/hosts` vornehmen.

Nehmen wir an, für `firma-123.de` soll ein neuer Webauftritt erstellt werden. Momentan zeigt der DNS-Eintrag der Firma noch auf den alten Server, z. B. auf die IP-Adresse `234.234.236.237`. Als Administrator haben Sie mittlerweile den neuen Server eingerichtet, der die IP-Adresse `123.124.125.126` hat. Jetzt möchten Sie testen, ob alles funktioniert. Also verändern Sie vorübergehend auf Ihrem lokalen Rechner `/etc/hosts`. Die dort durchgeführte Einstellung hat Vorrang vor allen Nameservern!

```
# /etc/hosts auf einem lokalen Rechner (NICHT auf dem Server)
...
# neue IP-Adr.      # vorhandener Name
123.124.125.126   firma-123.de www.firma-123.de
```

Der entscheidende Vorteil dieser Variante im Vergleich zur vorhin skizzierten Vorgehensweise mit einem eigenen Port besteht darin, dass beim Test auch alle Links funktionieren. Wenn Sie im Testbetrieb den Link `http://www.firma-123.de/cms/seite-xy.html` anklicken, wird auch die neue Seite wieder korrekt in Ihrem Webbrowser angezeigt.



## 32.4 Verschlüsselte Verbindungen (HTTPS)

In der Standardkonfiguration verwendet Apache das Protokoll HTTP. Es überträgt alle Daten unverschlüsselt. Für lokale Testinstallationen ist das in Ordnung, aber für im Internet erreichbare Webseiten ist HTTP nicht mehr zeitgemäß. Selbst für Webseiten, die keine Benutzerdaten entgegennehmen, wird heute HTTPS empfohlen – und sei es nur, um das Ranking in den Suchergebnissen zu verbessern. Sobald Ihre Webseite in Formularen oder auf anderen Wegen persönliche Daten (Passwörter, Kreditkartennummern etc.) entgegennimmt, sollte eine HTTPS-Konfiguration selbstverständlich sein. Immer mehr Webbrowser zeigen andernfalls unmissverständliche Warnungen an, die darauf hinweisen, dass die unverschlüsselte Übertragung der Daten unsicher ist.

HTTPS vereint die Protokolle *Hypertext Transfer Protocol* (HTTP) mit *Secure Sockets Layer* (SSL) und fügt HTTP so Verschlüsselungsfunktionen hinzu. In diesem Abschnitt erkläre ich Ihnen, wie Sie Apache für HTTPS-Verbindungen konfigurieren. Details zur Verwendung der kostenlosen Zertifikate der Organisation *Let's Encrypt* folgen in [Abschnitt 32.5](#).

In der Vergangenheit war es unmöglich, HTTPS mit virtuellen Hosts zu kombinieren, die sich nur durch den Hostnamen unterscheiden. Das Problem bestand darin, dass auch der Hostname selbst verschlüsselt übermittelt wurde. Für den Webserver war es damit unmöglich, das richtige Zertifikat zu »erraten«.

Namensbasierte  
virtuelle Hosts  
und HTTPS

Mittlerweile senden Webbrowser den Hostnamen beim Verbindungsaufbau vorweg einmal unverschlüsselt. Damit weiß Apache, welches Zertifikat er für die Verschlüsselung verwenden soll. Damit das funktioniert, muss

```
SSLStrictSNIVHostCheck off
```

gelten, was bei aktuellen Apache-Versionen standardmäßig der Fall ist. Weitere Details zur korrekten Konfiguration mehrerer Hosts, die jeweils ihr eigenes HTTPS-Zertifikat verwenden, finden Sie hier:

<https://wiki.apache.org/httpd/NameBasedSSLVHostsWithSNI>

### Zertifikate

Bevor Sie nun mit den Konfigurationsarbeiten beginnen, brauchen Sie noch ein Server-Zertifikat. Und an dieser Stelle muss ich etwas ausholen ...

Die Verschlüsselung der Daten erfolgt auf der Basis asymmetrischer Verschlüsselungsverfahren. Die Grundidee besteht darin, dass es ein Schlüsselpaar gibt, das aus einem privaten (geheimen) und einem öffentlichen Schlüssel besteht. Der öffentliche

Grundlagen

Schlüssel eignet sich nur zum *Verschlüsseln* von Daten. Zum *Entschlüsseln* ist der private Schlüssel erforderlich. Auf die Details dieser Verfahren gehe ich hier nicht ein – sie wurden schon unzählige Male beschrieben und erklärt, unter anderem auch in der Wikipedia.

Beim Verbindungsaufbau zwischen dem Client (also einem Webbrowser) und dem Server (Apache) wird zuerst auf der Basis einer Zufallszahl vom Client und des öffentlichen Schlüssels vom Server ein gemeinsamer Schlüssel ausgehandelt (Handshake-Verfahren). Dieser *Session Key* wird dann zur Verschlüsselung der gesamten weiteren Kommunikation eingesetzt. Der Webbrowser ist damit in der Lage, die zu sendenden Daten so zu verschlüsseln, dass nur der Webserver diese mit seinem privaten Schlüssel auswerten kann.

Der nach heutigem Wissensstand nahezu abhörsichere Datenaustausch ist aber nur *ein* Punkt zur Verbesserung der Sicherheit. Ein zweiter Punkt besteht darin, dass der Anwender Gewissheit haben muss, dass er mit dem richtigen Partner kommuniziert. Was nützt es, wenn die Daten für das Online-Banking zwar abhörsicher übertragen werden, aber statt zur Bank direkt in die Hände eines Betrügers gelangen?

Aus diesem Grund können Zertifikate auch Daten über die Website sowie eine Art Unterschrift einer Zertifizierungseinrichtung enthalten. Deren Aufgabe ist es, die Identität des Zertifikatbewerbers anhand einer Passkopie, eines Gewerbescheins etc. zu überprüfen. Dieser Kontrollprozess macht derartige Zertifikate leider relativ teuer.

Wie vertrauenswürdig ein Zertifikat ist, hängt von der Vertrauenswürdigkeit der Authentifizierungsstelle ab. Bekannte Webbrowser wie Firefox oder Internet Explorer akzeptieren nur Zertifikate, die von etablierten Authentifizierungseinrichtungen ausgestellt wurden (z. B. Verisign oder Thawte). Bei anderen Zertifikaten werden unübersehbare Warnungen angezeigt. Mit etwas Hartnäckigkeit kann man den Webbrowser zwar dennoch dazu bringen, auch unsichere Zertifikate zu akzeptieren, ein florierendes Online-Geschäft ist auf dieser Basis aber unmöglich. Mit anderen Worten: Für ernsthafte Geschäftsanwendungen ist ein autorisiertes Zertifikat unabdingbar.

**Dateikennungen** Tabelle 32.5 fasst die üblichen Dateikennungen für Schlüssel- und Zertifikatsdateien zusammen. Dabei steht *pem* für *Privacy Enhanced Mail*. Das ist ein Verfahren für verschlüsselte E-Mails, das sich leider nicht durchgesetzt hat. Das dort definierte PEM-Format ist aber bis heute üblich. \*.*crt*-Dateien enthalten zumeist ebenfalls *pem*-Dateien. Dank der abweichenden Dateikennung wird die Datei vom Internet Explorer als Zertifikat erkannt.

| Kennung | Bedeutung  |
|---------|--|
| *.key   | Private-Key-Datei  |
| *.csr   | unsigniertes Zertifikat (Certificate Signing Request)                          |
| *.pem   | Container-Datei für ein signiertes Zertifikat oder eine ganze Zertifikatskette |
| *.crt   | *.pem-Datei mit anderer Kennung (Windows)                                      |

**Tabelle 32.5** Dateikennungen für Schlüssel- und Zertifikatsdateien

### Selbst signierte Zertifikate erstellen

Nachdem ich Sie gerade zu überzeugen versucht habe, ein »richtiges« Zertifikat zu erwerben, erkläre ich Ihnen jetzt, wie Sie ein Zertifikat selbst erstellen können. Es gibt gute Gründe für diesen scheinbaren Sinneswandel: Für erste Experimente reicht ein selbst erstelltes Zertifikat vollkommen aus. Außerdem lässt sich ein eigenes Zertifikat in wenigen Minuten erstellen, während die Erteilung eines autorisierten Zertifikats erfahrungsgemäß tage-, wenn nicht wochenlang dauert. Diese Wartezeit nutzen Sie am besten, indem Sie sich mit den wichtigsten Stolperfallen vertraut machen. Wenn grundsätzlich alles funktioniert, können Sie Ihr eigenes Zertifikat mühelos durch ein »richtiges« ersetzen.

Bei vielen Distributionen sind selbst signierte Zertifikate sogar standardmäßig vorhanden. Unter Debian und Ubuntu werden sie »Snakeoil-Zertifikate« genannt (`/etc/ssl/certs/ssl-cert-snakeoil.pem`). CentOS, Fedora und RHEL haben stattdessen ein selbst signiertes Zertifikat für den Hostnamen des Rechners (`/etc/pki/tls/certs/localhost.crt`).

Als Erstes installieren Sie das Paket `openssl`. Es enthält das gleichnamige Kommando zum Erzeugen, Verwalten und Manipulieren von Schlüsseln und Zertifikaten. `openssl`

```
root# apt/dnf/yum/zypper install openssl
```

Das folgende Kommando erzeugt eine Datei mit einem privaten 2048-Bit-RSA-Schlüssel. Diesen Schlüssel werden Sie zweimal benötigen: einmal zum Erzeugen einer Zertifikatsanfrage (*Certificate Signing Request*) und dann nochmals zur Signierung. Zuerst der Schlüssel

```
root# openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 \
-out server.key
```

Achten Sie darauf, dass nur `root` diese Datei lesen kann! Wenn diese Datei in fremde Hände gerät, ist Ihr Serverzertifikat wertlos, und Sie müssen es widerrufen!

```
root# chmod 400 server.key
```

### Verschlüsselte Schlüssel

Schlüssel sind wertvoll – deswegen werden sie normalerweise selbst mit einer *Passphrase*, also mit einem besonders langen Passwort verschlüsselt. Im Englischen spricht man hier deutlich klarer von einem *encrypted key*. Wenn Sie das möchten, ergänzen Sie das obige `openssl`-Kommando um die Option `-aes-256-cbc`. Bei den weiteren Kommandos müssen Sie dann jedes Mal, wenn Sie den Schlüssel nutzen, wiederum dieses Passwort angeben.

Das gilt auch für Apache: Der Webserver braucht den Schlüssel, um das Zertifikat auszulesen. Deswegen fragt Apache nun bei jedem (Neu-)Start nach dem Passwort des Schlüssels. Ein automatisierter Neustart, z. B. bei einem Update, wird so unmöglich.

Um diesem Problem zu entgehen, wird für Apache eine unverschlüsselte Kopie des Schlüssels erzeugt (`openssl rsa -in server.key -out server-unencrypted.key`). Apache wird nun so konfiguriert, dass er anstelle des verschlüsselten Schlüssels die unsichere Schlüsseldatei verwendet. Diese muss also im Dateisystem des Servers gespeichert werden. Und spätestens jetzt wird klar, dass ein verschlüsselter Schlüssel in unserem Fall nur ein unnötiger Mehraufwand ist, der die Sicherheit nicht steigert. Das heißt aber natürlich nicht, dass verschlüsselte Schlüssel generell nicht zu empfehlen wären – ganz im Gegenteil!

... dann das  
Zertifikat

Schon etwas mehr Arbeit macht es, das Zertifikat zu erstellen. Genau genommen erzeugt das folgende Kommando nicht das endgültige Zertifikat, sondern einen *Certificate Signing Request*, also eine Anfrage zur Signierung des Zertifikats. In das Zertifikat fließt auch der Schlüssel `server.key` ein (Option `-key`).

Bei der Ausführung des Kommandos müssen Sie angeben, in welchem Land und in welchem Ort Sie wohnen, wie Sie heißen etc. Entscheidend ist die Frage nach dem *Common Name*: Hier ist nicht Ihr Name gefragt, sondern der exakte Name Ihrer Website in der Form, in der er für verschlüsselte Verbindungen verwendet wird. Manche Websites verwenden für verschlüsselte Verbindungen eine eigene Subdomain (z. B. `banking.ing-diba.de`), andere nicht (z. B. `www.amazon.de`). Wie auch immer, das Zertifikat gilt nur für eine bestimmte Schreibweise. Sie können also beispielsweise ein Zertifikat für `www.firma-abc.de` nicht auch für `firma-abc.de` verwenden (oder umgekehrt)! Achten Sie darauf, dass Sie das Challenge-Passwort leer lassen, die entsprechende Frage also mit der Eingabe eines Punktes beantworten.

```
root# openssl req -new -sha256 -key server.key -out server.csr
Enter pass phrase for server.key: *****
You are about to be asked to enter information that will be
incorporated into your certificate request.
What you are about to enter is what is called a Distinguished
Name or a DN. There are quite a few fields but you can leave
some blank. For some fields there will be a default value,
```

If you enter '.', the field will be left blank.

```
-----
Country Name (2 letter code) [AU]:          DE
State or Province Name (full name) [...]:  .
Locality Name (eg, city) []:               Berlin
Organization Name (eg, company) [Sample Ltd]: Max Muster
Organizational Unit Name (eg, section) []:  .
Common Name (eg server FQDN or YOUR name) []: www.firma-abc.de
Email Address []:                           webmaster@firma-abc.de
```

Please enter the following 'extra' attributes to be sent with your certificate request

```
A challenge password []: .
An optional company name []: .
```

Mit dem nächsten Kommando unterschreiben Sie Ihr Zertifikat selbst. Bei einem »richtigen« Zertifikat erfolgt dieser Vorgang – natürlich nach einer Kontrolle der von Ihnen vorgelegten Dokumente – durch die Authentifizierungseinrichtung. Zur Unterschrift wird dann der Schlüssel der Authentifizierungsstelle verwendet.

... und zuletzt die  
Signatur

Standardmäßig gilt das fertige Zertifikat nur für 30 Tage. Die Option `-days 1900` verlängert den Gültigkeitszeitraum auf circa fünf Jahre:

```
root# openssl x509 -req -days 1900 -in server.csr \
      -signkey server.key -sha256 -out server.pem
Signature ok
subject=/C=DE/L=Berlin/O=Max Muster/CN=www.firma-abc.de/
      emailAddress=webmaster@firma-abc.de
Getting Private key
```

#### unable to write random state

Mitunter tritt beim Ausführen des obigen Kommandos der Fehler *unable to write random state* auf. Der Grund dafür besteht zumeist darin, dass `openssl` das Heimatverzeichnis nicht findet – oft deswegen, weil Sie zuvor `sudo -s` ausgeführt haben. Abhilfe schafft `export PATH=/root`.

Physikalisch gesehen, handelt es sich bei den erzeugten Schlüsseln und Zertifikaten um relativ kleine Textdateien. Um die in einem Zertifikat enthaltenen Daten im Klartext anzuzeigen, verwenden Sie das Kommando `openssl x509 -text`. Die folgenden Ausgaben sind aus Platzgründen gekürzt:

Kontrolle

```
root# ls
server.key  (unverschlüsselter privater Schlüssel)
server.csr  (Zertifikat ohne Unterschrift)
server.crt  (Zertifikat mit Unterschrift)
```

```

root# cat server.pem
-----BEGIN CERTIFICATE-----
MIICWTCCAeICQCCL6ExhrQiELDANBgkqhkiG9wOBAQUFADBxMQswCQYDVQQGEWJB ...
-----END CERTIFICATE-----

root# openssl x509 -text -in server.pem
Certificate:
  Data:
    Version: 1 (0x0)
    Serial Number: 12669601459972319941 (0xafd37766c36baac5)
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: C=DE, L=Berlin, O=Max Muster,
           CN=www.firma-abc.de/emailAddress=webmaster@firma-abc.de
    Validity
      Not Before: Sep 28 14:48:03 2015 GMT
      Not After : Dec 10 14:48:03 2020 GMT
    Subject: C=DE, L=Berlin, O=Max Muster,
            CN=www.firma-abc.de/emailAddress=webmaster@firma-abc.de
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      Public-Key: (2048 bit)
      Modulus:
        00:be:37:21:23:b6:13:e4:92:74:36:9f:de:4e:9a:
      Signature Algorithm: sha256WithRSAEncryption
        43:b8:92:82:0d:f6:e2:ef:ff:07:eb:3a:1f:da:3d:d9:ba:53:
        d7:1f:4a:49:ec:5a:c1:fb:0f:95:e8:94:89:ab:7b:05:95:62:

```

## Apache-Konfiguration für den HTTPS-Betrieb

**mod\_ssl** Die für das Protokoll HTTPS erforderlichen Apache-Funktionen befinden sich im Modul `mod_ssl`. Unter Debian oder Ubuntu ist dieses Modul standardmäßig installiert und muss nur aktiviert werden:

```

root# a2enmod ssl
root# systemctl restart apache2

```

Unter Fedora bzw. CentOS/RHEL müssen Sie das SSL-Modul zuerst installieren:

```

root# dnf/yum install mod_ssl
root# systemctl restart httpd

```

**SSL-Konfiguration** Apache muss die Schlüssel- und Zertifikatsdatei lesen – daher liegt es nahe, die beiden Dateien in das Apache-Konfigurationsverzeichnis zu kopieren:

```

root# cp server.pem server.crt /etc/apache2 (Debian, SUSE, Ubuntu)
root# cp server.pem server.crt /etc/httpd (CentOS, Fedora, RHEL)

```

Als Nächstes müssen Sie `httpd.conf` (Fedora, RHEL) um einen `VirtualHost`-Eintrag erweitern bzw. die entsprechenden Zeilen in eine neue Datei in `/etc/apache2/sites-available` einfügen. Bei Debian und Ubuntu wird eine entsprechende Musterdatei gleich mitgeliefert (`default-ssl`). Sie können diese Datei als Ausgangsbasis für eine eigene Site-Datei verwenden, der Sie zur besseren Unterscheidbarkeit von anderen Site-Dateien `ssl` oder `https` voranstellen, also beispielsweise `ssl.firma-abc.de`.

Die folgenden Zeilen zeigen eine minimale Konfiguration, bei der parallel zur Default-website (HTTP) eine HTTPS-Seite eingerichtet wird. Für beide Seiten kommt dieselbe IP-Adresse zum Einsatz. Die Unterscheidung erfolgt durch die in der `VirtualHost`-Zeile eingestellte Port-Nummer 443.

`SSLEngine on` aktiviert die Verschlüsselungsfunktionen. `SSLxxxFile` gibt an, wo sich die Dateien mit dem Zertifikat und dem privaten Schlüssel befinden. `SSLProtocol` und `SSLCipherSuite` bestimmen, welche Version des SSL-Protokolls bzw. welcher Mechanismus zur Erzeugung des gemeinsamen Session Keys eingesetzt werden soll. In der Regel tauschen Apache und der Webbrowser Informationen darüber aus, welche Protokolle sie jeweils unterstützen, und verwenden dann das sicherste Verfahren, das beide beherrschen. Nur wenn es gute Gründe dafür gibt – etwa, weil Sie bestimmte ältere Protokolle/Verfahren nicht akzeptieren möchten –, sollten Sie hier explizite Vorgaben machen.

```
# z.B. in /etc/httpd/conf.d/ssl.conf (CentOS, Fedora, RHEL)
# oder in /etc/apache2/sites-available/ssl-firma-abc.conf (Debian, Ubuntu)
<VirtualHost _default_:443>
    ServerName          www.firma-abc.de
    DocumentRoot        /var/www/
    SSLEngine           on
    SSLCertificateFile  /etc/apache2/server.pem
    SSLCertificateKeyFile /etc/apache2/server.key
    <Directory /var/www/>
        AllowOverride None
        Require all granted
    </Directory>
</VirtualHost>
```

Zur Aktivierung der HTTPS-Site müssen Sie Apache dazu auffordern, die Konfiguration neu einzulesen. Falls Sie die HTTPS-Datei unter Debian/Ubuntu in einer eigenen Konfigurationsdatei in `/etc/apache2/sites-available` durchgeführt haben, müssen Sie diese Datei aktivieren:

```
root# systemctl restart httpd      (CentOS/Fedora/RHEL)
root# a2ensite ssl-firma-abc        (Debian/Ubuntu, Teil 1)
root# systemctl restart apache2    (Debian/Ubuntu, Teil 2)
```

Wenn Sie erstmals von einem Betrieb ohne SSL auf einen Betrieb mit SSL umstellen, reicht ein Neueinlesen der Konfigurationsdateien nicht aus. Damit das SSL-Modul geladen wird, müssen Sie `restart` angeben, nicht `reload`!

#### Sicherheitswarnung

Wenn ein Webbrowser auf ein selbst signiertes Zertifikat oder ein Snakeoil-Zertifikat stößt, zeigt der Browser eine Sicherheitswarnung wie in [Abbildung 32.2](#) an. Die Warnung bezieht sich nicht darauf, dass die Verschlüsselung nicht sicher wäre – das ist sie! Vielmehr warnt der Webbrowser, weil die Identität dessen, der das Zertifikat unterzeichnet hat, unbekannt ist. Für technisch unbedarfte Anwender ist das eine Feinheit; sie werden die Webseite als unsicher betrachten. Wenn es Ihnen aber nur darum geht, dass Sie selbst phpMyAdmin sicher verwenden können, um die MySQL-Installation auf dem Server zu administrieren, dann reicht dieses Zertifikat vollkommen aus!

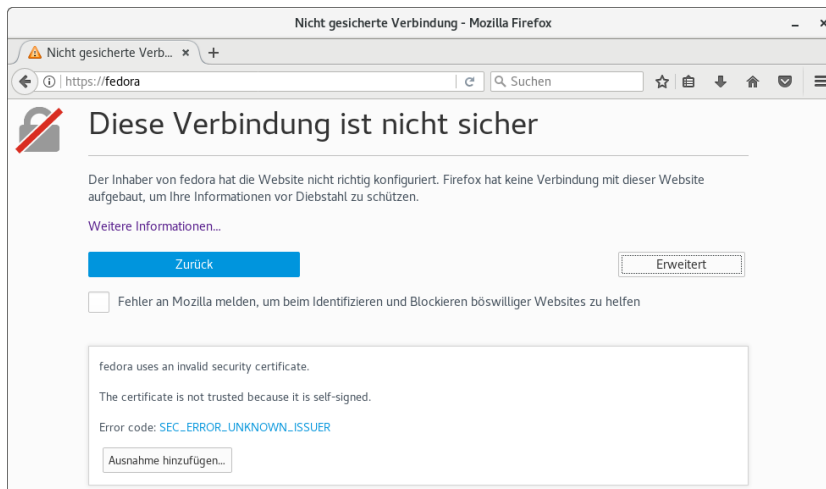


Abbildung 32.2 Warnung vor einem selbst signierten Zertifikat

### Snakeoil-Zertifikate

#### Debian und Ubuntu

Bei Debian und Ubuntu werden bei der Installation von Apache automatisch ein Snakeoil-Schlüssel und ein zehn Jahre lang gültiges Snakeoil-Zertifikat erzeugt:

```
/etc/ssl/certs/ssl-cert-snakeoil.pem (Zertifikat)
/etc/ssl/private/ssl-cert-snakeoil.key (Schlüssel)
```

»Snakeoil« wird im Englischen als Bezeichnung für vorgebliche Wunder- oder Allheilmittel verwendet. Das Zertifikat wird erzeugt, damit Web- und Mail-Server ohne das umständliche Erzeugen eigener Zertifikate sofort verschlüsselt verwendet werden können. Die Apache-Konfigurationsdatei `default-ssl.conf` enthält dementsprechend die folgenden Zeilen:



```
# Datei /etc/apache2/sites-available/default-ssl.conf
...
SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
```

Natürlich gelten für das Snakeoil-Zertifikat dieselben Einschränkungen wie bei Zertifikaten, die Sie mit openssl selbst erzeugt und signiert haben – daher auch der Name. Das Snakeoil-Zertifikat berücksichtigt den bei der Erstellung gültigen Rechnernamen (/etc/hostname). Wenn Sie nach der Änderung eines Hostnames ein neues Zertifikat benötigen, rufen Sie das folgende Kommando auf:

```
root# make-ssl-cert generate-default-snakeoil --force-overwrite
```

make-ssl-cert ist ein relativ simples Script, das auf das vorhin beschriebene openssl-Kommando zurückgreift.

Bei Distributionen aus der Red-Hat-Familie gibt es ähnliche, automatisch erstellte und selbst signierte Defaultzertifikate:

CentOS, Fedora  
und RHEL

```
# Datei /etc/httpd/conf.d/ssl.conf
...
SSLCertificateFile /etc/pki/tls/certs/localhost.crt
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
```

Die Zertifikate sind ein Jahr für den Hostnamen gültig, der während der Installation des httpd-Pakets eingestellt war.

### Zertifikate etablierter Zertifizierungsanbieter verwenden

Wenn Sie sich entscheiden, ein Zertifikat bei Thwate, Verisign etc. zu erwerben, ersparen Sie sich naturgemäß das Erzeugen der Schlüssel- und Zertifikatsdateien. Sie erhalten diese Dateien vom Zertifikatsanbieter und kopieren Sie in ein Verzeichnis, auf das Apache Zugriff hat, z. B. /etc/apache2, /etc/httpd, /etc/pki oder /etc/ssl. Verwenden Sie aber auf keinen Fall ein Verzeichnis mit Webdateien, damit der Webserver den privaten Schlüssel nicht unbeabsichtigt öffentlich macht!

Die Pfade zum Zertifikat und zum Schlüssel geben Sie wiederum durch die Schlüsselwörter SSLCertificateFile und SSLCertificateKeyFile an. Außerdem müssen Sie in vielen Fällen an den Browser Zusatzinformationen dazu übergeben, wie er Ihre Zertifikate überprüfen kann. Genau genommen geht es hier um Informationen darüber, welche anerkannte Zertifizierungsstelle wiederum Ihrer Zertifizierungsstelle vertraut. Der Browser muss in der Lage sein, eine Vertrauenskette bis zu einer Zertifizierungsstelle herzustellen, die ihm bekannt ist.

SSLCertificate-  
ChainFile

Vergessen Sie diese Zusatzfunktionen, beklagt sich der Webbrowser beim Besuch Ihrer Seite, dass der Verbindung nicht vertraut wird, weil keine Zertifikatsaussteller-

kette angegeben wurde. Abhilfe schaffen die Schlüsselwörter `SSLCertificateChainFile` und `SSLCACertificateFile`, mit denen Sie an Apache Zertifikate Ihrer Zertifizierungsstelle übergeben (Certification Authority, daher die Abkürzung CA). Die erforderlichen Zertifikatsdateien stellt Ihnen Ihre Zertifizierungsstelle zum Download zur Verfügung. Nachdem Sie die Dateien so auf Ihrem Rechner eingerichtet haben, dass Apache sie lesen kann, ändern Sie die Konfiguration wie folgt und führen dann `systemctl reload apache2/httpd` aus.

```
...
SSLCertificateFile      /etc/apache2/server.pem
SSLCertificateKeyFile   /etc/apache2/server.pem
SSLCertificateChainFile /etc/apache2/sub.class1.server.ca.pem
SSLCACertificateFile   /etc/apache2/ca.pem
...
```

Diese Zusatzinformationen ermöglichen es dem Webbrowser, die Korrektheit der von Ihnen benutzten Zertifikate zu überprüfen.

## SSL-Einstellungen

Mit dem Einrichten von Zertifikaten ist es leider nicht getan. Es gibt unzählige Verschlüsselungsverfahren und -versionen, die alle unter dem Begriff HTTPS zusammengefasst werden. Manche von ihnen sind jedoch nicht mehr sicher. Deswegen müssen Sie mit `SSLCipherSuite` und `SSLProtocol` explizit angeben, welche Verfahren Ihre Webseite unterstützt und welche sie ablehnt. Eine große Hilfe bei der optimalen HTTPS-Konfiguration ist die folgende Webseite (siehe [Abbildung 32.3](#)):

<https://www.ssllabs.com/ssltest>

Dort können Sie die Adresse Ihrer Webseite angeben. Ein Script überprüft dann die Sicherheit Ihrer Webseite und gibt Optimierungstipps. Dazu gleich ein konkretes Beispiel: Meine Webseite <https://kofler.info> läuft auf einem Ubuntu-Server mit Apache 2.4. Für HTTPS verwende ich ein Zertifikat von Let's Encrypt (siehe den folgenden Abschnitt). Die relevanten Apache-Konfigurationszeilen sehen wie folgt aus:

```
SSLEngine on

# Zertifikate von Let's Encrypt
SSLCertificateFile      /etc/letsencrypt/live/kofler.info/cert.pem
SSLCertificateKeyFile   /etc/letsencrypt/live/kofler.info/privkey.pem
SSLCertificateChainFile /etc/letsencrypt/live/kofler.info/chain.pem

# »CRIME«-Attacke, siehe
# https://raymii.org/s/tutorials/Strong_SSL_Security_On_Apache2.html
SSLCompression off
SSLCipherSuite AES128+EECDH:AES128+EDH
```

```
# »Poodle«-Problem, siehe https://access.redhat.com/solutions/1232413
SSLProtocol All -SSLv2 -SSLv3

# HTTP Strict Transport Security (HSTS) aktivieren
# siehe https://de.wikipedia.org/wiki/HTTP_Strict_Transport_Security
Header always set Strict-Transport-Security \
    "max-age=63072000; includeSubdomains; preload"
```

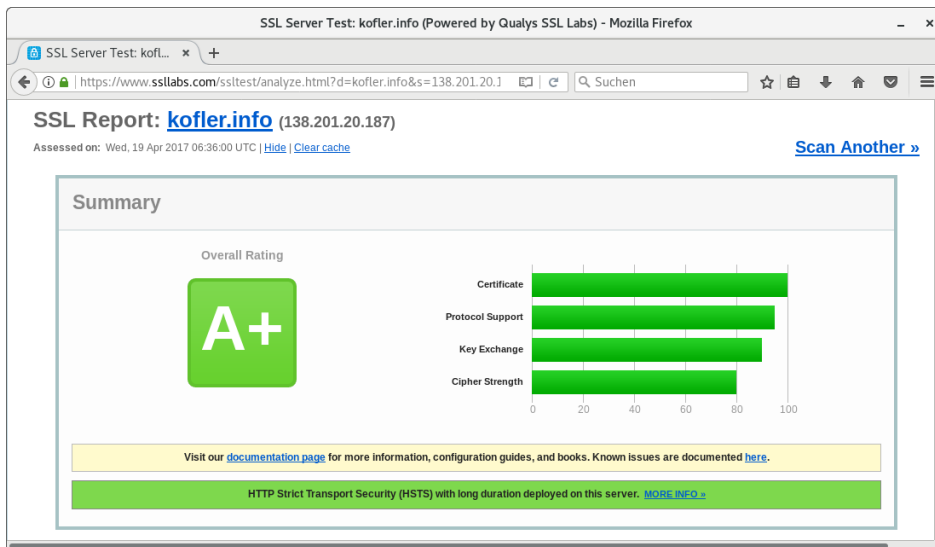


Abbildung 32.3 Online-Überprüfung der HTTPS-Konfiguration

## 32.5 Let's Encrypt

Seit Anfang 2016 bietet die Organisation *Let's Encrypt* (<https://letsencrypt.org>) kostenlose Zertifikate an. Mittlerweile sind diese Zertifikate sehr weit verbreitet: Anfang 2017 vertrauten bereits mehr als 20 Millionen Websites auf Let's Encrypt. Die naheliegende Frage lautet daher: »Warum noch Geld für Zertifikate anderer Anbieter ausgeben?«

Let's Encrypt ist mit zwei wesentlichen Einschränkungen verbunden: Zum einen sind die Zertifikate nur 90 Tage gültig. Um diesen Nachteil zu umgehen, muss das System so eingerichtet werden, dass die Zertifikate regelmäßig automatisch aktualisiert werden. (Das ist nicht schwierig.)

Zum anderen handelt es sich bei den Zertifikaten von Let's Encrypt um reine Domänenzertifikate: Sier ermöglichen die Kontrolle, dass eine Webseite tatsächlich von der Domäne *firma-abc.de* kommt und nicht von einer anderen Seite. Let's Encrypt führt aber keinerlei Kontrolle durch, wem diese Domäne gehört (*Organization Validation*),

geschweige denn, wer diese Person bzw. Organisation ist (*Extended Validation* durch Überprüfung einer Passkopie, eines Firmenbuchauszugs etc.). Mit anderen Worten: Auch Betrüger können kostenlose Zertifikate von Let's Encrypt verwenden.

Zertifikate von Let's Encrypt sind mit der billigsten und gleichzeitig am weitesten verbreiteten Zertifikatsvariante der etablierten Zertifizierungsstellen vergleichbar. Die Zertifikate reichen aus, um ein eigenes Blog oder ähnliche Seiten so zu verschlüsseln, dass der Webbrowser ein grünes Schloss oder ein anderes Sicherheitssymbol anzeigt. Firmen bzw. Organisationen, die auf ihren Webseiten einen Shop realisieren, Online-Banking anbieten, Gesundheits- oder Versicherungsdaten verwalten etc., sind aber weiterhin auf höherwertige und damit leider auch recht teure Zertifikate von anderen Anbietern angewiesen.

**Software** In den ersten Monaten hat Let's Encrypt selbst Software für die Zertifikatsverwaltung angeboten, unter anderem das inzwischen veraltete Kommando `letsencrypt`. Mittlerweile kümmert sich die Electronic Frontier Foundation (EFF) um die Entwicklung und Wartung der Software. Der offizielle Let's-Encrypt-Client hat seither den Namen `certbot` (<https://certbot.eff.org>). Darüber hinaus gibt es eine Reihe anderer Clients in allen erdenklichen Programmiersprachen. Eine Referenz finden Sie hier:

<https://letsencrypt.org/docs/client-options>

### certbot installieren

Einige Distributionen stellen `certbot` bereits in den eigenen Paketquellen zur Verfügung, z. B. Fedora. Bei anderen Distributionen müssen Sie externe Repositories aktivieren, z. B. EPEL für CentOS/RHEL oder `jessie-backports` für Debian 8.

```
root# yum install python-certbot-apache           (CentOS/RHEL)
root# apt install python-certbot-apache -t jessie-backports (Debian 8)
root# apt install python-certbot-apache           (Debian 9)
root# dnf install python-certbot-apache           (Fedora)
```

Ubuntu-Pakete befinden sich im Private Package Archive `ppa:certbot/certbot`:

```
root# add-apt-repository ppa:certbot/certbot       (Ubuntu)
root# apt update
root# apt install python-certbot-apache
```

Für openSUSE gab es im Frühjahr 2017 noch kein passendes Paket. Zur Installation führen Sie die folgenden Kommandos durch:

```
root# wget https://dl.eff.org/certbot-auto       (openSUSE)
root# mv certbot-auto /usr/local/bin
root# chmod a+x /usr/local/bin/certbot-auto
root# certbot-auto -h
```

certbot-auto aktualisiert sich bei der ersten Ausführung selbst (und in der Folge immer wieder, wenn es neue Versionen gibt). Beim ersten Mal werden dabei diverse Pakete installiert. Beachten Sie, dass das Kommando bei einer manuellen Installation certbot-auto heißt, bei der Installation aus einem Paket dagegen einfach certbot!

Weitere Installationsanleitungen für alle erdenklichen Kombinationen aus verschiedenen Webservern und Distributionen finden Sie hier:

<https://certbot.eff.org>

### Zertifikate einrichten

Das Kommando certbot erfüllt verschiedene Aufgaben, je nachdem, welche Optionen übergeben werden:

- ▶ Es kontaktiert den Server des Let's-Encrypt-Projekts, fordert dort ein Zertifikat an, lädt es herunter und installiert es im Verzeichnis `/etc/letsencrypt/domainname`.
- ▶ Es passt die Konfigurationsdateien von Apache oder nginx so an, dass das neue Zertifikat verwendet wird.
- ▶ Es testet, welche der aktuell installierten Zertifikate in den nächsten 30 Tagen ablaufen, und erneuert diese.

#### Verwenden Sie certbot zuerst im Testmodus!

Um Missbrauch zu vermeiden, gibt es strikte Limits, wie viele Zertifikate für eine Domain in einer bestimmten Zeit erzeugt werden dürfen:

<https://letsencrypt.org/docs/rate-limits>

Die wichtigste Regel lautet, dass Sie pro Domain maximal 20 Zertifikate pro Woche erzeugen können, wobei ein Zertifikat mehrere Sub-Domänen umfassen darf. Um zu vermeiden, dass Sie diese Limits überschreiten, sollten Sie certbot für erste Tests immer mit der Option `--staging` bzw. `--test-cert` (veraltet) aufrufen. Damit erhalten Sie Zertifikate von einem Test-System. Erst wenn Sie sicher sind, dass alles funktioniert, erstellen Sie die richtigen Zertifikate ohne diese Option.

Um Let's-Encrypt-Zertifikate anzufordern und für den Webserver Apache zu installieren, führen Sie das folgende Kommando aus. Dabei ersetzen Sie `meine-domain.de` durch Ihren Domainnamen. Die Zertifikate für `smtp.*`, `mail.*` und `imap.*` können später zur Konfiguration des Mail-Servers verwendet werden (siehe [Abschnitt 34.3](#), »Postfix-Verschlüsselung (TLS/STARTTLS«). Wenn Sie nicht vorhaben, hierfür Let's-Encrypt-Zertifikate zu verwenden, lassen Sie diese Hostnamen weg.

**Zertifikate  
anfordern und  
installieren**

Auf keinen Fall verzichten sollten Sie hingegen auf die `www`-Variante, selbst dann, wenn Sie `http://domainname` gegenüber `http://www.domainname` vorziehen. Das `www`-Subdomain-Zertifikat ist erforderlich, damit Rewrite-Regeln von `www.domainname` auf `domainname` später auch für HTTPS funktionieren.

```
root# certbot --apache --staging -d meine-domain.de \
      -d www.meine-domain.de -d imap.meine-domain.de \
      -d smtp.meine-domain.de -d mail.meine-domain.de
```

`certbot` fordert Sie bei der ersten Installation eines Zertifikats dazu auf, eine E-Mail-Adresse anzugeben. An diese E-Mail-Adresse wird vor dem Ablauf eines Zertifikats eine Warnung gesendet (z. B. falls die automatische Aktualisierung aus irgendeinem Grund versagt hat). Oft fragt `certbot` auch, welche Apache-Konfigurationsdatei es verändern soll. Wenn die automatische Konfiguration der Konfigurationsdateien zu Fehlern führt, rufen Sie das Kommando in der Form `certbot --certonly` auf, und führen die Konfiguration selbst durch.

Erst wenn Sie sicher sind, dass alles klappt, entfernen Sie die Option `--staging` und wiederholen das Kommando nochmals zur Installation der endgültigen Zertifikate.

#### Wo sind die Zertifikate?

`certbot` installiert die Zertifikate und Schlüssel in Verzeichnisse der Form `/etc/letsencrypt/archive/domainname`. In `/etc/letsencrypt/live/domainname` befinden sich Links auf die gerade gültigen Zertifikate. `/etc/letsencrypt` enthält darüber hinaus diverse Metadaten, die für den Zertifikatserneuerungsprozess benötigt werden.

#### Apache-Konfiguration

Damit Let's-Encrypt-Zertifikate für einen bestimmten virtuellen Host verwendet werden, müssen nur drei `SSLxxxFile`-Anweisungen eingefügt bzw. geändert werden:

```
<VirtualHost *:443>
  ServerName    meine-domain.de
  ServerAlias   www.meine-domain.de
  ServerAlias   imap.meine-domain.de
  ServerAlias   smtp.meine-domain.de
  DocumentRoot  ...

  SSLCertificateFile    /etc/letsencrypt/live/meine-domain.de/cert.pem
  SSLCertificateKeyFile /etc/letsencrypt/live/meine-domain.de/privkey.pem
  SSLCertificateChainFile /etc/letsencrypt/live/meine-domain.de/chain.pem
</VirtualHost>
```

#### Für jede Subdomain einen ServerAlias!

Aus nicht ganz einsichtigen Gründen funktioniert die automatische Zertifikatserneuerung nur dann ohne Fehler, wenn die Apache-Konfigurationsdateien jede Subdomain mit `ServerName` oder `ServerAlias` enthält!

Wenn Sie Let's Encrypt wie beschrieben auch dazu verwenden, um Zertifikate für den Mail-Server zu erzeugen, dann müssen Sie entsprechende `ServerAlias`-Zeilen hinzufügen, obwohl diese für den Webserver-Betrieb vollkommen sinnlos sind.

Anfang Juli 2017 wurde bekannt, dass Let's Encrypt ab Januar 2018 auch Wildcard-Zertifikate anbieten wird, also Zertifikate, die für alle Subdomains gelten (\*.meine-domain.de). Das wird die Sache voraussichtlich vereinfachen. Allerdings wird dazu eine neue API-Version zum Einsatz kommen (ACME v2, aktuell ist ACME v1). Es ist aktuell nicht abzusehen, ob sich dadurch der Umgang mit dem `certbot`-Kommando ändert.

Die README-Datei in `/etc/letsencrypt/live/meine-domain.de` empfiehlt eigentlich die Verwendung von `fullchain.pem` anstelle von `cert.pem` und `chain.pem`, aber die Webseite <https://www.ssllabs.com/ssltest> kritisiert dann, dass die Zertifikatskette unvollständig sei.

Unter Debian und Ubuntu richtet `certbot` außerdem eine Datei mit diversen SSL-Optionen ein:

```
# /etc/letsencrypt/options-ssl-apache.conf
SSLEngine on
SSLProtocol          all -SSLv2 -SSLv3
SSLCipherSuite       ECDHE-RSA-AES128-GCM-SHA256:...:!KRB5-DES-CBC3-SHA
SSLHonorCipherOrder on
SSLCompression       off
SSLOptions           +StrictRequire
...
```

Auf CentOS/RHEL-Rechnern verzichtet `certbot` auf eine vergleichbare Konfiguration. Sie sollten sich selbst darum kümmern, weil die Defaulteinstellungen in `ssl.conf` unter CentOS/RHEL mangelhaft sind.

Als weitere Optimierung können Sie noch *HTTP Strict Transport Security* (HTST) aktivieren. Damit teilen Sie dem Browser mit, dass er für eine bestimmte Zeit ausschließlich verschlüsselte Verbindungen zu Ihrer Seite herstellen darf:

[https://de.wikipedia.org/wiki/HTTP\\_Strict\\_Transport\\_Security](https://de.wikipedia.org/wiki/HTTP_Strict_Transport_Security)

```
Header always set Strict-Transport-Security \
    "max-age=63072000; includeSubdomains; preload"
```

Häufig ist erwünscht, dass alle HTTP-Seitenzugriffe automatisch auf HTTPS umgeleitet werden. `certbot` fragt, ob es entsprechende Anweisungen in die Apache-Konfigurationsdateien einbauen soll – aber das können Sie natürlich später auch selbst erledigen. Die folgenden Zeilen geben dafür ein Muster:

Umleitung von  
HTTP auf HTTPS

```

<VirtualHost _default_:80>
  ServerName meine-domain.de
  ServerAlias www. meine-domain.de
  ServerSignature Off

  RewriteEngine On
  RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,QSA,R=permanent]

  ErrorLog /var/log/httpd/redirect.error.log
  LogLevel warn
</VirtualHost>

```

Falls Sie außerdem eine Umleitung von *www.meine-domain.de* auf *meine-domain.de* wünschen, fügen Sie die folgenden Zeilen hinzu:

```

# Quelle: http://stackoverflow.com/questions/21467329
RewriteCond %{HTTP_HOST} ^(www\.)(.*) [NC]
RewriteRule (.*) https://%2%{REQUEST_URI} [L,R=301]

```

### Zertifikate erneuern

Das Kommando `certbot renew` kontrolliert die Let's-Encrypt-Zertifikate Ihres Rechners und erneuert alle, die in den nächsten 30 Tagen auslaufen. Es geht also nur noch darum, den Aufruf von `certbot renew` automatisch einmal pro Woche durchzuführen. Dazu erstellen Sie mit einem Editor die Datei `/etc/cron.weekly/letsencrypt` mit dem folgenden Inhalt:

```

#!/bin/bash
# Datei /etc/cron.weekly/letsencrypt
certbot renew

```

Diese Datei machen Sie mit `chmod a+x` ausführbar – fertig! Zu Testzwecken können Sie eine Zertifikatserneuerung mit `certbot --force-renewal renew` ausnahmsweise erzwingen. (Diese Option darf auf keinen Fall in einem Script stehen, das regelmäßig durch Cron ausgeführt wird.)

Um den nach einem Zertifikatswechsel erforderlichen Neustart von Apache kümmert sich `certbot` selbst. Dabei laufen vorhandene Apache-Prozesse mit der alten Konfiguration weiter, neue Prozesse verwenden die neue Konfiguration. Es kommt also zu keinem Verbindungsabbruch oder Session-Verlust. Informationen dazu finden Sie unter:

<https://community.letsencrypt.org/t/is-server-restart-needed-when-obtaining-certs-using-certbot-and-apache-module/17267>



Falls Sie Let's-Encrypt-Zertifikate auch für Postfix und Dovecot verwenden, müssen Sie diese Programme explizit neu starten. Die folgende Variante des obigen letsencrypt-Scripts testet, ob es in `/etc/letsencrypt/live` Zertifikate gibt, die sich innerhalb der letzten 24 Stunden geändert haben:

Postfix und Dovecot neu starten

```
#!/bin/bash
# Datei /etc/cron.weekly/letsencrypt
certbot renew
result=$(find /etc/letsencrypt/live/ -type l -mtime -1 )
if [ -n "$result" ]; then
    systemctl restart postfix
    systemctl restart dovecot
fi
```

### Einschränkungen und Sicherheitsvorkehrungen

Bei der Nutzung von Let's Encrypt sollten Ihnen die folgenden Einschränkungen bewusst sein:

- ▶ Sie können ein Zertifikat später nicht um eine Sub-Domäne erweitern. Wenn Sie das möchten, müssen Sie vielmehr das gesamte Zertifikat neu erzeugen. Das ist an sich kein Problem, es gelten dafür aber die Let's-Encrypt-Rate-Limits (aktuell maximal 20 Zertifikate je Domäne pro Woche).
- ▶ Sie können mit `certbot` nur Zertifikate für Domänen erstellen, deren DNS-Einträge auf Ihren Server zeigen. Diese naheliegende Einschränkung verhindert die missbräuchliche Erstellung von Zertifikaten für fremde Domänen.
- ▶ Der Zertifikats-Update-Prozess setzt voraus, dass die bisherigen Zertifikate und Schlüssel in `/etc/letsencrypt` zur Verfügung stehen. Das klingt auf den ersten Blick wie eine Selbstverständlichkeit. Pech haben Sie freilich, wenn diese Dateien bei einem Server-Crash oder -Umzug verloren gehen. Eine Aktualisierung der Zertifikate ist unmöglich, und eine Neueinstellung ist erst erlaubt, nachdem die bisherigen Zertifikate ausgelaufen sind – in der Regel also erst nach drei Monaten! Kümmern Sie sich rechtzeitig um Backups des gesamten Verzeichnisses `/etc/letsencrypt`!

Ich habe im vergangenen Jahr mehrere Websites auf Zertifikate von Let's Encrypt umgestellt. Dabei hatte ich mehrfach Probleme mit der durch das `certbot`-Kommando durchgeführten Veränderung der Apache-Konfiguration. Dieser heikle Prozess funktioniert offensichtlich nur einfachen Fällen fehlerfrei. Rufen Sie `certbot` gegebenenfalls in der Form `certbot certonly ...` auf und führen Sie die Apache-Konfiguration anschließend selbst durch. Davon abgesehen funktionierten die Zertifikatsausstellung, deren Installation in das Verzeichnis `/etc/letsencrypt` sowie deren Updates absolut problemlos.

## 32.6 Webzugriffsstatistiken

Wer einen eigenen Webserver betreibt oder für jemand anderen administriert, will in der Regel auch wissen, wie viele Personen die Website pro Tag besuchen, welche Webbrowser dabei zum Einsatz kommen etc. Ich stelle Ihnen hier vorweg einige Tools kurz vor und gehe dann etwas ausführlicher auf den Einsatz von GoAccess ein.

**Webalizer und AWStats** In der Vergangenheit wurden zur Erstellung derartiger Statistiken häufig die Programme Webalizer oder AWStats verwendet. Diese Programme aus den Anfangszeiten des Internets werden zwar noch gewartet, sind aber nicht mehr zeitgemäß. Die Konfiguration ist relativ aufwendig, weil ihr Aufruf mit der Rotation der Logging-Dateien synchronisiert werden muss. Genaueres finden Sie unter:

<http://awstats.org>

<http://webalizer.org>

**Google Analytics** Gewissermaßen der Star unter den modernen Web-Analyzer-Tools ist *Google Analytics*. Dazu müssen Sie auf den Seiten Ihrer Website JavaScript-Code einbauen, der jeden Seitenzugriff an einen zentralen Server weiterleitet. Diese Vorgehensweise erleichtert die Unterscheidung zwischen »echten« Besuchern und Suchrobotern und führt zu genaueren Ergebnissen, die in Echtzeit beobachtet werden können. Da Google Analytics nicht die Logging-Dateien auswertet, sondern auf Code basiert, der direkt in der Website integriert ist, kommt Google Analytics wesentlich besser mit dynamischen Websites zurecht, bei denen die Startseite nie verlassen wird (Single-Page-Webanwendungen).

Beachten Sie aber, dass der Einsatz von Google Analytics unter Einhaltung der deutschen Datenschutzgesetze problematisch ist! Auf jeden Fall sollten Sie in Ihrer Website einen Bestätigungsdialo einbauen, der die Besucher auf die Verwendung von Google Analytics hinweist. Dieser Dialog wird oft mit der in vielen Ländern ebenfalls vorgeschriebenen (wenngleich vollkommen sinnlosen) Cookie-Einverständniserklärung kombiniert.

<https://www.google.com/analytics>

**Piwik** Piwik ist eine Open-Source-Alternative zu Google Analytics. Sein Einsatz vermeidet, dass Google mit noch mehr Daten gefüttert wird – die erfassten Daten bleiben unter Ihrer Kontrolle. Aus der Perspektive des Datenschutzes agiert Piwik aber ähnlich wie Google Analytics; daher sollten Sie auch bei der Verwendung von Piwik einen unübersehbaren Hinweis in Ihre Website einbauen.

<https://piwik.org>

<https://www.datenschutzzentrum.de/uploads/projekte/verbraucherdatenschutz/20110315-webanalyse-piwik.pdf>

## GoAccess

GoAccess ist aus meiner Sicht ein guter Kompromiss zwischen den veralteten Programmen Webalizer und AWStats auf der einen Seite und datenschutztechnisch problematischen Tools wie Google Analytics oder Piwik auf der anderen Seite. GoAccess wertet die Logging-Dateien des Webservers aus und agiert insofern ähnlich wie Webalizer oder AWStats. Im Gegensatz zu diesen Programmen eignet sich das Programm aber auch zur Echtzeit-Analyse des Web-Traffics. Außerdem kann es bei Bedarf in einem Terminalfenster via SSH verwendet werden, also ohne den sonst üblichen Zwischenschritt der Generierung von Reports in Form von Webseiten.

<https://goaccess.io>

GoAccess steht bei vielen Distributionen als Paket zur Verfügung und kann mit `apt/dnf/yum/zypper` installiert werden. Bei meinen Tests waren die so installierten Versionen aber durchwegs veraltet. Besser ist es daher, wenn Sie eine manuelle Installation gemäß den Anweisungen auf der Download-Seite des Projekts durchführen. Im Frühjahr 2017 waren dazu die folgenden Anweisungen erforderlich:

Installation

```
root# apt install libncursesw5-dev libgeoip-dev libssl-dev      (Ubuntu)
root# yum groupinstall development                            (CentOS)
root# yum install ncurses-devel geoip-devel openssl-devel wget (CentOS Forts.)

user$ wget http://tar.goaccess.io/goaccess-1.2.tar.gz        (alle Distrib.)
user$ tar -xvzf goaccess-1.2.tar.gz
user$ cd goaccess-1.2/
user$ ./configure --enable-utf8 --enable-geoip=legacy --with-openssl
user$ make
root# make install

root# ln -s /usr/local/bin/goaccess /usr/bin                  (CentOS)
```

In Zukunft wird sich die Versionsnummer sicher ändern.

Am einfachsten rufen Sie `goaccess` in einer SSH-Session auf und übergeben als Parameter den Dateinamen der Apache-Logging-Datei. Unter Ubuntu gelingt das einem nicht privilegierten Benutzer:

Anwendung im Terminal

```
user$ goaccess /var/log/apache2/access.log
```

Unter CentOS/Fedora/RHEL befinden sich die Logging-Dateien in `/var/log/httpd` und sind nur für `root` zugänglich. Der Aufruf von `goaccess` muss dann mit `root`-Rechten erfolgen.

```
root# goaccess /var/log/httpd/access_.log
```

Beim Start müssen Sie angeben, in welchem Format die Logging-Datei vorliegt. Normalerweise reicht es aus, den vorgegebenen Eintrag `NCSA COMBINED LOG FORMAT`

zu bestätigen. goaccess zeigt dann eine Auswertung der Logging-Datei an und aktualisiert diese regelmäßig, bis Sie das Programm mit `Q` beenden (siehe [Abbildung 32.4](#)).

```

kofler@host1: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
Dashboard - Overall Analyzed Requests (24/Apr/2017 - 26/Apr/2017) [Active Panel: Visitors]

Total Requests 69081 Unique Visitors 5931 Unique Files 3228 Referrers 0
Valid Requests 69081 Init. Proc. Time 1s Static Files 504 Log Size 17.18 MiB
Failed Requests 0 Excl. IP Hits 0 Unique 404 64 Bandwidth 2.12 GiB
Log Source /var/log/apache2/access.log

> 1 - Unique visitors per day - Including spiders Total: 3/3

Hits h% Vis. v% Bandwidth Data
-----
6606 9.56% 678 11.43% 169.26 MiB 26/Apr/2017 |||||
32399 46.90% 2787 46.99% 1015.07 MiB 25/Apr/2017 |||||
30076 43.54% 2466 41.58% 983.30 MiB 24/Apr/2017 |||||

2 - Requested Files (URLs) Total: 366/3228

Hits h% Vis. v% Bandwidth Mtd Proto Data
-----
5699 10.98% 302 0.76% 157.35 MiB GET HTTP/1.1 /feed/
3625 6.98% 2584 6.47% 6.19 MiB GET HTTP/1.1 /wp-content/plugins/wp-spamshield/js/jscripts.php
2801 5.39% 2658 6.66% 4.46 MiB GET HTTP/1.1 /wp-content/plugins/cookie-notice/css/front.css?ver=
2753 5.30% 2560 6.41% 5.00 MiB GET HTTP/1.1 /wp-content/themes/twentyfourteen-child/style.css?ve
2721 5.24% 2566 6.43% 39.25 MiB GET HTTP/1.1 /wp-content/themes/twentyfourteen/style.css?ver=505d
2689 5.18% 2549 6.39% 52.15 MiB GET HTTP/1.1 /wp-content/themes/twentyfourteen/genericons/generic
[?] Help [Enter] Exp. Panel 162 - Wed Apr 26 08:44:45 2017 [Q]uit GoAccess 1.2

```

Abbildung 32.4 Zugriffsstatistiken im Terminalfenster

Mit den Cursortasten oder mit `↵` können Sie durch weitere Rubriken scrollen (statische Zugriffe, 404-Fehler, Hostnamen und IP-Adressen etc.). `←` erweitert die gerade aktive Rubrik um weitere Details. `S` sortiert die Zeilen nach einem anderen Kriterium. Eine Zusammenfassung weiterer Tastenkürzel liefert `H`.

Die meisten Distributionen richten täglich oder wöchentlich neue Access-Log-Dateien ein und komprimieren die älteren Dateien (siehe auch die Beschreibung von `logrotate` in [Abschnitt 18.9](#), »Logging (Syslog)«). Das folgende Kommando verarbeitet alle komprimierten sowie die beiden nicht komprimierten Dateien `access.log.1` und `access.log.2`. Beachten Sie, dass ein derartiger Aufruf von `goaccess` anfänglich ziemlich lange dauert (alle Logging-Dateien müssen dekomprimiert und eingelesen werden) und dass der Speicherbedarf von `goaccess` in diesem Fall erheblich ist – je nach Größe der Log-Dateien im GiB-Bereich! Wenn Sie das Zeichen `*` durch `?` ersetzen, werden nur die letzten zehn Logging-Dateien berücksichtigt, was oft auch ausreicht.

```

user$ cd /var/log/apache2
user$ zcat access.log.*.gz | \
      goaccess --log-format=COMBINED access.log access.log.1 -

```

## Format-Fehlermeldung

Wenn `goaccess` eine Fehlermeldung der Art *No time format was found on your conf file* liefert, haben Sie zwei Möglichkeiten: Entweder geben Sie das Logformat Ihres Webserver mit einer Option an (wie im obigen Beispiel mit `--log-format`) oder Sie schreiben das Format in der Konfigurationsdatei fest, deren Ort im Rahmen der Fehlermeldung angezeigt wird (bei einer manuellen Konfiguration `/usr/local/etc/goaccess.conf`).

Indem Sie `goaccess` zusätzlich die Option `-o out.html` übergeben, erzeugen Sie eine HTML-Datei mit der Zugriffsstatistik. Diese Seite können Sie dann mit einem Webbrowser ansehen (siehe [Abbildung 32.5](#)). Die Erzeugung derartiger Zugriffsstatistiken können Sie natürlich in einem Cron-Job automatisieren und einmal täglich oder wöchentlich durchführen.

HTML-Reports erzeugen

```
user$ cd /var/log/apache2
user$ zcat access.log.?.gz | goaccess -o /var/www/html/myreports/out.html \
    --log-format=COMBINED access.log access.log.1
```

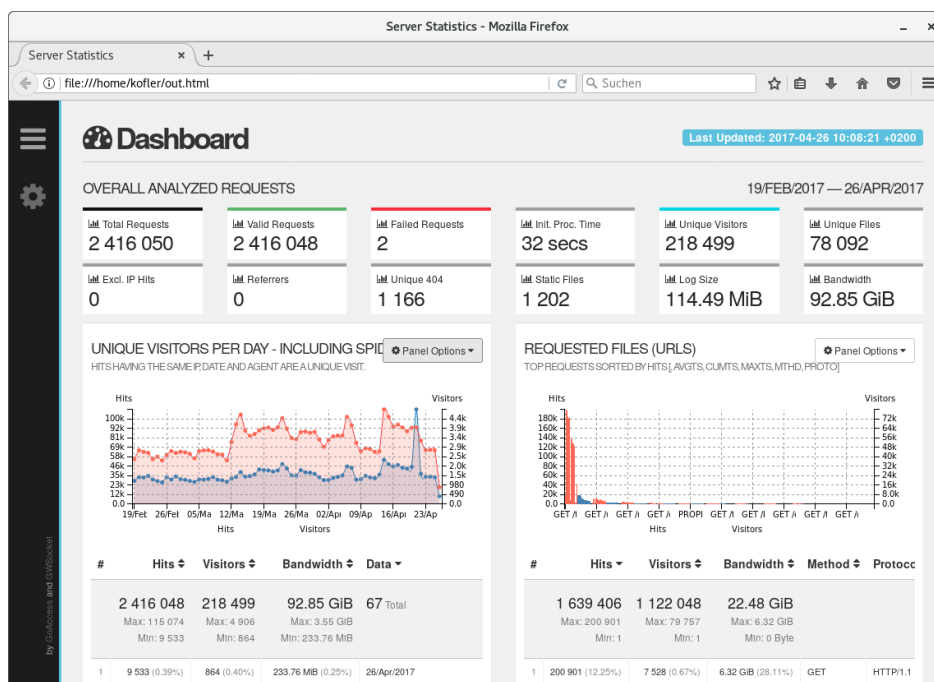


Abbildung 32.5 Zugriffsstatistiken im Webbrowser auswerten

`goaccess` kann sogar Echtzeit-Updates über die HTML-Seite liefern. Dazu übergeben Sie an `goaccess` zusätzlich die Option `--real-time-html`. Wenn die resultierende Seite

über eine HTTPS-Verbindung übertragen werden soll, müssen Sie außerdem das Zertifikat und dessen Schlüssel in Parametern übergeben. Der Zugriff auf diese Dateien erfordert die Ausführung des Kommandos mit `root`-Rechten.

```
root# cd /var/log/apache2
root# goaccess --log-format=COMBINED access.log access.log.1 \
  -o /var/www/html/myreports/uploads/report.html --real-time-html \
  --ssl-cert=/etc/letsencrypt/live/mysite/fullchain.pem \
  --ssl-key=/etc/letsencrypt/live/mysite/privkey.pem
```

## 32.7 PHP

**Dynamische Webseiten** Apache an sich kann nur statische Webseiten übertragen. Alle modernen Websites nutzen aber dynamische Seiten. Jedes Mal, wenn eine derartige Seite angefordert wird, startet Apache ein externes Programm, verarbeitet den Code der Seite und liefert als Ergebnis eine Seite, die individuell angepasst ist. Damit kann die Seite beispielsweise die aktuelle Uhrzeit enthalten oder das Ergebnis einer Datenbankabfrage oder eine ständig wechselnde Werbeeinblendung etc.

**PHP** Zur Programmierung dynamischer Webseiten eignen sich zahllose Programmiersprachen – z. B. Perl, PHP oder Java. Die Grundidee einer PHP-Webseite besteht darin, dass die Datei mit der Kennung `*.php` sowohl HTML- als auch PHP-Code enthält. PHP-Code wird mit dem Tag `<?php` eingeleitet und endet mit `?>`.

Wenn ein Webnutzer eine PHP-Seite anfordert, übergibt Apache die Seite an den PHP-Interpreter. Dort wird der PHP-Code ausgeführt. Das Ergebnis des Codes wird direkt in die HTML-Datei eingebettet. Der PHP-Interpreter übergibt die resultierende Seite zurück an Apache, und dieser sendet sie dem Webnutzer. Der Webbrowser des Nutzers sieht also nie den PHP-Code, sondern immer nur die resultierende HTML-Seite.

**Hello World!** Der Platz reicht hier nicht für eine Einführung in die Programmiersprache PHP. Stattdessen soll das folgende Minibeispiel das Konzept von PHP veranschaulichen. Die folgende Datei liefert nach der Verarbeitung durch den PHP-Interpreter eine HTML-Seite mit der aktuellen Uhrzeit:

```
<!DOCTYPE html>
<html><head>
  <meta http-equiv="Content-Type"
    content="text/html; charset=utf-8" />
  <title>PHP-Beispiel</title>
</head><body>

<p>Die aktuelle Uhrzeit auf diesem Server:
```

```
<?php
    date_default_timezone_set("Europe/Berlin");
    echo strftime("%k:%M:%S") . "</p>";
    echo "<p>Sonderzeichentest: äöü</p>";
?>
</body></html>
```

Sofern PHP nicht bereits mit Apache mitinstalliert wurde, installieren Sie mit Ihrem Paketverwaltungsprogramm die erforderlichen php-Pakete. Was »erforderlich« ist, ist allerdings gar nicht so einfach festzustellen: Ähnlich wie bei Apache ist auch PHP über zahlreiche Pakete verteilt, die die Sprache an sich sowie diverse Erweiterungen enthalten. Für erste Experimente reichen üblicherweise `php<n>`, `php<n>-common` sowie `libapache2-mod-php<n>`. Soweit sich nicht die Paketverwaltung darum kümmert, müssen Sie Apache nach der Installation neu starten, damit der Webserver neu hinzugekommene PHP-Module berücksichtigt.

Installation

Zahllose Optionen des PHP-Interpreters werden durch die Datei `php.ini` gesteuert. Im Regelfall können Sie die Grundeinstellungen einfach beibehalten. Der Ort dieser Datei sowie weiterer PHP-Konfigurationsdateien ist wieder einmal distributionsabhängig:

Konfiguration

|                              |  |
|------------------------------|--|
| Debian und Ubuntu mit PHP 5: | <code>/etc/php5/apache2/php.ini,</code><br><code>/etc/php5/apache2/conf.d/*.ini</code>                       |
| Debian und Ubuntu mit PHP 7: | <code>/etc/php/7.&lt;n&gt;/apache2/php.ini,</code><br><code>/etc/php/7.&lt;n&gt;/apache2/conf.d/*.ini</code> |
| CentOS, Fedora, RHEL:        | <code>/etc/php.ini, /etc/php.d/*.ini</code>  |
| SUSE:                        | <code>/etc/php5/apache2/php.ini</code>   |

Um zu testen, ob die PHP-Installation funktioniert, erstellen Sie die Datei `phptest.php`, die aus nur einer einzigen Zeile Code besteht:

Test

```
<?php phpinfo(); ?>
```

Kopieren Sie diese Datei in das `DocumentRoot`-Verzeichnis (siehe [Tabelle 32.1](#)), und stellen Sie sicher, dass Apache die Datei lesen darf. Obwohl es sich bei PHP-Dateien eigentlich um Script-Dateien handelt, reichen Leserechte. Zugriffsrechte zum Ausführen (x-Zugriffsbits) sind nicht erforderlich.

Mit einem Webbrowser sehen Sie sich nun die Seite `http://localhost/php-test.php` an. Das Ergebnis ist eine sehr umfangreiche Seite, die alle möglichen Optionen und Einstellungen von Apache und PHP enthält (siehe [Abbildung 32.6](#)). Aus Sicherheitsgründen ist es nicht empfehlenswert, eine derartige Seite frei zugänglich ins Internet zu stellen. Sie enthält eine Menge Informationen über die Konfiguration Ihres Webservers.

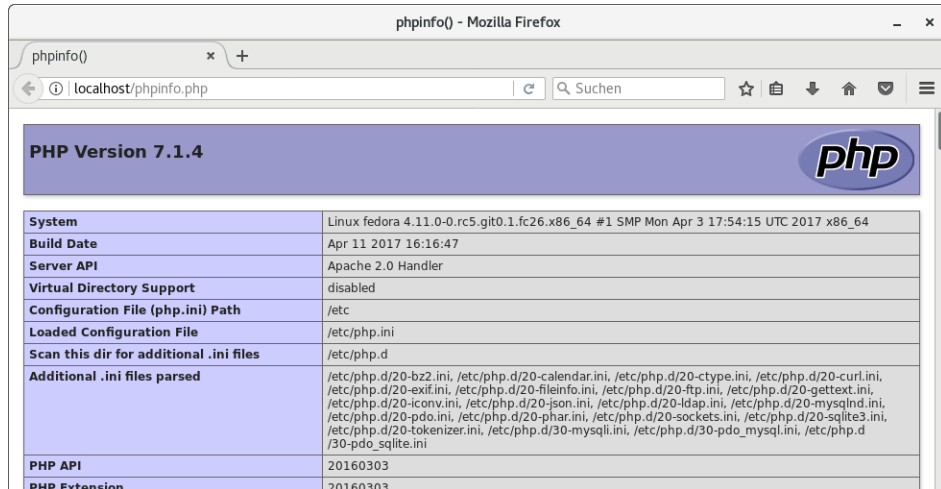


Abbildung 32.6 PHP-Testseite

Wenn es nicht funktioniert

Wenn Sie statt der Testseite den PHP-Code sehen oder die PHP-Datei zum Download angeboten bekommen, ist die wahrscheinlichste Fehlerursache die, dass Sie als Webadresse den Dateinamen (z. B. `/srv/www/htdocs/phpinfo.php`) angegeben haben. In diesem Fall wird die Datei direkt aus dem lokalen Dateisystem gelesen, anstatt von Apache und PHP verarbeitet zu werden. Die Webadresse muss mit `http://` beginnen!

Eine weitere Fehlerursache ist die Apache-Konfiguration: Haben Sie Apache nach der Installation von PHP bzw. nach der Veränderung von Konfigurationsdateien neu gestartet?

Wenn es einmal nicht geklappt hat, kann Ihnen in der Folge der Cache Ihres Webbrowsers einen Strich durch die Rechnung machen. Anstatt die Seite neu von Apache anzufordern, was nun vielleicht funktionieren würde, liest der Browser die Seite aus dem internen Cache. Starten Sie den Browser sicherheitshalber neu bzw. löschen Sie den Cache!

## 32.8 FTP-Server (vsftpd)

Vielen Webservern gesellt sich ein FTP-Server hinzu, der je nach Website zwei Aufgaben erfüllt: Einerseits ermöglicht er den Download großer Dateien, die auf der Website zur Verfügung gestellt werden, andererseits hilft er bei der Wartung bzw. Aktualisierung der Website, indem er eine einfache Möglichkeit zum Upload von Dateien zulässt.



FTP ist ein sehr altes Programm. Sein Protokoll führt in Kombination mit Firewalls bzw. mit Masquerading oft zu Problemen. Noch problematischer ist der Umstand, dass beim Verbindungsaufbau zwischen einem FTP-Client und dem -Server der Benutzername und das Passwort unverschlüsselt übertragen werden. Da stehen jedem sicherheitsbewussten Anwender die Haare zu Berge!

Sicherheit

Natürlich gibt es schon längst sichere Alternativen zu FTP. Unter anderem stellt der in [Kapitel 31](#) beschriebene SSH-Server mit SFTP (*Secure FTP*) auch Dienste zur Dateiübertragung zur Verfügung. Das Problem liegt hier mehr auf der Client-Seite: Es gibt nur relativ wenige benutzerfreundliche Programme, die SFTP beherrschen. Aus diesem Grund wird FTP trotz aller Sicherheitsmängel noch immer recht häufig eingesetzt.

Eine andere Alternative ist der WebDAV-Standard, der das HTTP-Protokoll erweitert und die Datenübertragung in beide Richtungen erleichtert. Beispielsweise unterstützt Apache in Kombination mit dem Modul `mod_dav` WebDAV:

[http://httpd.apache.org/docs/2.4/mod/mod\\_dav.html](http://httpd.apache.org/docs/2.4/mod/mod_dav.html)

<https://wiki.ubuntuusers.de/Webdav>

Wenn Sie auf einen traditionellen FTP-Server nicht verzichten möchten, können Sie diesen auch als reinen Anonymous-FTP-Server konfigurieren. Dabei werden beim Login keine kritischen Daten übertragen. Allerdings schränkt das auch die Anwendung von FTP stark ein. Zur einfachen Wartung einer Website lässt sich FTP dann nicht mehr verwenden.

Es gibt unzählige verschiedene FTP-Server. Das populärste Programm ist momentan `vsftpd`. Alle gängigen Distributionen stellen hierfür ein Paket zur Verfügung. `vsftpd` steht für *Very Secure FTP Daemon*. Das Attribut *Very Secure* ist aber unter dem Vorbehalt zu sehen, dass auch der beste FTP-Server die Sicherheitsmängel des FTP-Protokolls aufweist.

vsftpd

`vsftpd` kann auf zwei Arten gestartet werden: entweder als eigenständiger Dämon durch das Init-System oder über `xinetd`. Bei den meisten Distributionen ist die Dämon-Variante vorkonfiguriert. Die Konfigurationsdatei `vsftpd.conf` muss dazu die Anweisung `listen=YES` enthalten.

Start als Dämon

Um den FTP-Server zu starten bzw. zu stoppen, verwenden Sie je nach Distribution die üblichen Kommandos (siehe [Abschnitt 12.5](#), »Systemprozesse (Dämonen)«). Unter CentOS, Fedora und RHEL gehen Sie z. B. so vor:

```
root# systemctl start vsftpd
root# systemctl enable vsftpd
```

Wie üblich müssen Sie auch sicherstellen, dass die Firewall die FTP-Ports 20 und 21 nicht blockiert. Unter SUSE verwenden Sie zur Firewall-Konfiguration am besten YaST.

Unter CentOS/Fedora/RHEL stellen Sie zuerst fest, welche Firewall-Zone für die Netzwerkschnittstelle zum Internet gilt (hier FedoraWorkstation), und aktivieren dann für diese Zone Ausnahmeregeln:

```
root# firewall-cmd --get-zone-of-interface=enp0s3 (aktive Zone herausfinden)
FedoraWorkstation
root# firewall-cmd --permanent --zone=FedoraWorkstation --add-service=ftp
root# firewall-cmd --reload
```

**Konfiguration** Die Konfiguration von vsftpd erfolgt durch die Datei `/etc/vsftpd.conf` bzw. `/etc/vsftpd/vsftpd.conf`. Standardmäßig ist oft nur ein Read-Only-Zugang per Anonymous FTP zugelassen. FTP-Clients können also nur einen Download, aber keinen Upload durchführen. Wenn Sie neben Anonymous FTP auch Benutzer-Logins benötigen, müssen Sie `local_enable` auf YES stellen. Wenn Sie bei dieser FTP-Form auch einen Daten-Upload zulassen möchten, müssen Sie zusätzlich `write_enable` auf YES stellen. Wenn `vsftpd.conf` die Zeile `tcp_wrappers=Yes` enthält, wertet vsftpd wie xinetd die Dateien `/etc/hosts.allow` und `/etc/hosts.deny` aus (siehe [Abschnitt 37.2](#), »Basisabsicherung von Netzwerkdiensten«). Die folgenden Zeilen fassen die wichtigsten Einstellungen in `vsftpd.conf` zusammen:

```
# /etc/vsftpd.conf bzw. /etc/vsftpd/vsftpd.conf
...
local_enable=YES / NO      # FTP-Login zulassen
write_enable=YES / NO     # Daten-Upload grundsätzlich zulassen
...
anonymous_enable=YES / NO # Anonymous FTP zulassen
anon_upload_enable=YES / NO # Daten-Upload auch bei Anonymous FTP
...
listen=YES / NO          # Start als Init-Dämon (YES) oder durch xinetd (NO)
tcp_wrapper=YES / NO     # hosts.allow und hosts.deny auswerten
```

**FTP ausprobieren** FTP müsste jetzt eigentlich auf Anhieb funktionieren. Führen Sie auf dem Server-Rechner `ftp localhost` aus, um zu testen, ob der FTP-Server ordnungsgemäß gestartet wird. Beachten Sie dabei, dass `root` grundsätzlich keinen FTP-Login durchführen darf.

**Anonymous FTP** Wenn Anonymous FTP in `vsftpd.conf` zugelassen ist, akzeptiert vsftpd als Login die Namen `anonymous` und `ftp` in Kombination mit einem beliebigen Passwort. Es ist üblich, als Passwort die E-Mail-Adresse anzugeben. vsftpd kontrolliert das aber nicht.

Nach dem Login kann der FTP-Client auf die Dateien des Home-Verzeichnisses des Linux-Benutzers `ftp` zugreifen. Der Ort dieses Verzeichnisses wird in `/etc/passwd` angegeben:

```
Debian, Ubuntu:    /srv/ftp
Fedora, Red Hat:   /var/ftp/
SUSE:              /srv/ftp/
```

Wenn Sie den Upload von Dateien per Anonymous FTP zulassen, sollten Sie darauf achten, dass es nur ein einziges Verzeichnis innerhalb des FTP-Datenverzeichnisses gibt, das Schreibrechte hat – z.B. `/var/ftp/upload` bei Fedora oder Red Hat. Dieses Verzeichnis sollte dem Benutzer `ftp` gehören und aus Sicherheitsgründen keine Lese-rechte haben:

Upload per  
Anonymous FTP

```
root# mkdir /var/ftp/upload
root# chown ftp upload
root# chmod 730 upload
```

Somit kann jeder einen Upload durchführen und dem FTP-Administrator anschließend eine E-Mail mit Instruktionen senden, wofür die Datei dient. Andere FTP-Nutzer können die Datei aber im `upload`-Verzeichnis weder sehen noch herunterladen. Wenn Sie auf derartige Sicherheitsmaßnahmen verzichten, kann es passieren, dass das FTP-Upload-Verzeichnis zum Austausch illegaler Dateien missbraucht wird.

Aus Sicherheitsgründen sind `root` und einige andere Spezialbenutzer (wie `daemon`, `lp` oder `nobody`) von der FTP-Benutzung ausgeschlossen. Die dazu erforderliche Konfiguration variiert von Distribution zu Distribution.

FTP für root und  
andere  
Spezialbenutzer

Bei Fedora und Red Hat erfolgt der Login-Schutz doppelgleisig. Einerseits greift `vsftpd` für die Login-Kontrolle auf PAM zurück (*Pluggable Authentication Modules*). PAM wertet die Datei `/etc/pam.d/vsftpd` aus, die auf die Datei `etc/vsftpd/ftpusers` verweist. Diese Datei enthält eine Liste aller Login-Namen, die FTP *nicht* benutzen dürfen.

Andererseits wendet `vsftpd` auch eine interne Login-Kontrolle an und sperrt alle Benutzer, die in `/etc/vsftpd.user_list` genannt sind. Diese Login-Kontrolle wird in `vsftpd.conf` durch `userlist_enable=YES` und `userlist_deny=YES` (gilt standardmäßig) aktiviert.

Bei Debian, SUSE und Ubuntu greift `vsftpd` für den Login ebenfalls auf PAM zurück. `/etc/pam.d/vsftpd` verweist hier allerdings auf `/etc/ftpusers`. Diese Datei enthält eine Liste aller Login-Namen, die FTP nicht benutzen dürfen.



# Kapitel 33

## MySQL und MariaDB

MySQL ist seit vielen Jahren das populärste Datenbanksystem der Open-Source-Welt. Auch wenn Sie nicht vorhaben, selbst irgendwelche Datenbanken zu verwalten, gibt es unzählige Webanwendungen, die den MySQL-Server voraussetzen – beispielsweise Wikis, Diskussionsforen wie phpBB, Content-Management-Systeme wie WordPress oder Fehlerverwaltungssysteme wie Bugzilla.

In diesem Kapitel zeige ich Ihnen, wie Sie den MySQL- oder MariaDB-Server auf Ihrem Rechner installieren und wie Sie grundlegende Administrationsaufgaben durchführen. Auf das Design von Datenbanken, auf den Einsatz von SQL zur Abfrage bzw. zur Manipulation von Daten sowie auf die Entwicklung von Datenbankanwendungen gehe ich hingegen nicht ein – das würde hier den Rahmen sprengen.

MySQL wurde ursprünglich von der MySQL AB entwickelt, einer eigenständigen schwedischen Firma. Diese wurde von Sun aufgekauft. Später kaufte Oracle Sun, und so ist nun Oracle der Eigentümer von MySQL. Einige der ursprünglichen Gründer von MySQL initiierten später ein neues Projekt, dessen Ergebnis ein weitgehend zu MySQL kompatibler Datenbank-Server mit dem Namen MariaDB ist.

MySQL versus  
MariaDB

Die Zusammenarbeit zwischen Oracle und diversen Linux-Distributoren war in der Vergangenheit nicht friktionsfrei. Insbesondere beklagen viele Distributoren, dass sie Informationen über Sicherheitsprobleme nicht bzw. zu spät erhalten und es daher unmöglich sei, Sicherheitslücken rasch zu beheben.

Dieser Ärger führte dazu, dass viele Distributionen jetzt standardmäßig MariaDB anstelle von MySQL installieren. Von den großen Distributionen ist aktuell nur noch Ubuntu im MySQL-Lager, und auch dort stehen MariaDB-Pakete als Alternative zu MySQL zur Verfügung.

Ältere Versionen von MySQL und MariaDB waren nahezu vollständig kompatibel zueinander, sodass Sie bei einem Wechsel die Datenbankdateien unverändert belassen konnten. Für aktuelle Versionen gilt dies aber nicht mehr: Beide Datenbank-Server gehen bei Zusatzfunktionen, Erweiterungen und speziellen Konfigurationseinstellungen getrennte Wege. Der Wechsel von einem System zum anderen ist nur mit zunehmend aufwendiger Handarbeit möglich.

In der Praxis werden die meisten Linux-Anwender nicht bemerken, ob das CMS, Wiki etc. hinter den Kulissen mit einem originalen MySQL-Server oder mit einem MariaDB-Server kommuniziert. Auch die in diesem Kapitel beschriebenen Konfigurationsdateien sowie die Administrationswerkzeuge und -techniken gelten gleichermaßen für MySQL und für MariaDB. Insbesondere heißt auch bei MariaDB der Dienstname `mysqld` und das Steuerungskommando `mysql`. Wenn ich in diesem Kapitel ohne weitere Erläuterungen »MySQL« schreibe, gelten diese Informationen gleichermaßen für MySQL und für MariaDB.

**Lizenz** MySQL und die dazugehörenden Treiber für diverse Programmiersprachen unterstehen der GPL. Die firmeninterne Nutzung von MySQL, der Einsatz auf einem Webserver sowie die Nutzung in GPL-Projekten sind grundsätzlich kostenlos. Beachten Sie aber, dass die Weitergabe kommerzieller Projekte (Closed Source, keine GPL), die auf MySQL aufbauen, eine kommerzielle Lizenz des MySQL-Servers erfordert! Details zu den Lizenzbedingungen von MySQL finden Sie hier:

<https://mysql.com/about/legal>

MariaDB kann ausschließlich gemäß den Regeln der GPL verwendet werden.

### 33.1 Installation und Inbetriebnahme

**Installation** Bei allen gängigen Distributionen stehen MySQL- oder MariaDB-Pakete zur Verfügung. Einige Distributionen stellen sogar Pakete für beide Varianten zur Wahl, etwa Ubuntu. Der Datenbank-Server selbst, seine Bibliotheken und Administrationswerkzeuge befinden sich in unterschiedlichen Paketen. [Tabelle 33.1](#) fasst zusammen, welche Pakete Sie üblicherweise benötigen (Stand: Frühjahr 2017). Diese Pakete installieren Sie je nach Distribution mit `apt`, `dnf`, `yum` oder `zypper`. Vielfach ist der MySQL- oder MariaDB-Server ohnedies bereits installiert, weil Paketabhängigkeiten ihn erforderlich machen.

```
root# yum groupinstall mysql
```

**Start/Stop** MySQL bzw. MariaDB ist ein Dämon, der bei Fedora, RHEL und openSUSE explizit gestartet werden muss (siehe auch [Abschnitt 12.5](#), »Systemprozesse (Dämonen)«):

```
root# systemctl start mariadb
root# systemctl enable mariadb
```

Beachten Sie, dass openSUSE auch für MariaDB den Dienstnamen `mysql` verwendet. Sie müssen also `systemctl start mysql` ausführen, um den MariaDB-Server zu starten.

| Distribution                   | Datenbanksystem | Pakete  |
|--------------------------------|-----------------|---|
| CentOS/RHEL 7                  | MariaDB 5.5     | mariadb, mariadb-server, mariadb-libs                                 |
| Debian 8                       | MySQL 5.5       | mysql-server-5.5, mysql-common, libmysqlclient18                      |
| Debian 9                       | MariaDB 10.1    | mariadb-server-10.1, mariadb-common, libmariadbclient18               |
| Fedora 26                      | MariaDB 10.1    | mariadb, mariadb-common, mariadb-config, mariadb-libs, mariadb-server |
| openSUSE 42.2                  | MariaDB 10.0    | mariadb, mariadb-client, libmysqlclient18                             |
| Ubuntu 16.04 LTS, Ubuntu 17.04 | MySQL 5.7       | mysql-server-5.7, mysql-client-5.7, mysql-common                      |

**Tabelle 33.1** MySQL- und MariaDB-Pakete je nach Distribution

Egal, ob MySQL oder MariaDB: Die Datenbankdateien werden immer im Verzeichnis `/var/lib/mysql` gespeichert.

**Datenbank- und Logging-Dateien**

Bei einem Datenbank-Server gibt es diverse Logging-Dateien für unterschiedliche Aufgaben: Fehler-Logging, Transaktions-Logging, Update-Logging etc. Wenn beim Start Probleme auftreten, z. B. aufgrund einer falschen Konfiguration, werden die Fehlermeldungen im Fehlerprotokoll festgehalten. Der Ort dieser Datei variiert wieder stark je nach Distribution (siehe [Tabelle 33.2](#)).

| Distribution   | Ort der Logging-Dateien                   |
|----------------|---|
| CentOS/RHEL 7  | <code>/var/log/mariadb/mariadb.log</code> |
| Debian 8 und 9 | <code>/var/log/mysql/error.log</code>     |
| Fedora 26      | <code>/var/log/mariadb/mariadb.log</code> |
| openSUSE 42.2  | systemd-Journal                           |
| Ubuntu         | <code>/var/log/mysql/error.log</code>     |

**Tabelle 33.2** Speicherort für das Fehler-Logging bei verschiedenen Distributionen

Bei aktuellen openSUSE-Versionen extrahieren Sie die Meldungen aus dem systemd-Journal:

```
root# journalctl -u mysql
[Note] /usr/sbin/mysqld (mysqld 10.0.29-MariaDB) starting as process 13293 ...
[Note] InnoDB: Using mutexes to ref count buffer pool pages
[Note] InnoDB: The InnoDB memory heap is disabled
[Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
...
```

**Konfiguration** Die Konfiguration des MySQL- bzw. MariaDB-Servers erfolgt durch die Datei `/etc/my.cnf` bzw. `/etc/mysql/my.cnf` (Debian, Ubuntu). Unter CentOS, Fedora und RHEL werden außerdem die Dateien `/etc/my.conf.d/*.conf` berücksichtigt.

`my.cnf` ist für den gewöhnlichen Betrieb vorkonfiguriert. Aus Platzgründen kann ich nicht auf alle Schlüsselwörter für diese Datei eingehen. Einige Details möchte ich aber doch herausgreifen.

Grundsätzlich ist die Konfigurationsdatei durch `[abname]` in mehrere Abschnitte gegliedert. Im Folgenden beziehe ich mich ausschließlich auf den Abschnitt `[mysqld]`, der den MySQL- bzw. MariaDB-Server an sich betrifft. Unter Fedora befinden sich diese Einstellungen in der getrennten Datei `/etc/my.cnf.d/mariadb-server.cnf`.

Änderungen am `[mysqld]`-Abschnitt werden nur wirksam, wenn Sie den Datenbank-Server neu starten. Die anderen Abschnitte in `my.cnf` bzw. in den restlichen Konfigurationsdateien dienen zur Konfiguration diverser Client-Programme.

- ▶ `bind-address = 127.0.0.1`: Diese Einstellung bewirkt, dass Netzwerkverbindungen zum Datenbank-Server ausschließlich vom lokalen Rechner aus möglich sind, nicht aber von anderen Rechnern im lokalen Netzwerk oder aus dem Internet. Wenn MySQL/MariaDB ohnedies nur von lokalen Programmen genutzt werden soll, z. B. von PHP-Skripts des auf dem gleichen Rechner installierten Webservers, vergrößert diese Einstellung die Sicherheit. Bei Debian und Ubuntu gilt diese Einstellung standardmäßig, bei anderen Distributionen sollte sie nach Möglichkeit hinzugefügt werden.

Wenn Sie Netzwerkverbindungen zulassen möchten, den Datenbank-Server also ohne `bind-address`-Einschränkungen ausführen, müssen Sie unter CentOS, Fedora, RHEL und SUSE in der Firewall den Port 3306 freigeben.

- ▶ `skip-networking`: Diese Einstellung verhindert jeglichen Netzwerkzugang zum MySQL- bzw. MariaDB-Server. Selbst lokale Netzwerkverbindungen sind damit verboten. Die Einstellung ist noch restriktiver als `bind-address = 127.0.0.1`. Ein Verbindungsaufbau ist nur noch für lokale Programme möglich, die über eine sogenannte Socket-Datei mit dem Datenbank-Server kommunizieren. Das trifft z. B. für PHP-Skripts und C-Programme zu. Programme, die via TCP/IP mit dem Datenbank-Server kommunizieren, können den Datenbank-Server nicht nutzen. Diese Einschränkung betrifft insbesondere alle Java-Programme. Aus diesem Grund ist `bind-address = 127.0.0.1` zumeist zweckmäßiger.
- ▶ `character-set-server` und `collation-server`: Aus Kompatibilitätsgründen zu sehr alten Versionen verwenden MySQL und MariaDB-Server standardmäßig noch immer den Zeichensatz Latin-1 und die schwedische Sortierordnung. Nur Debian sieht in `/etc/mysql/mariadb.conf.d/50-server.cnf` sinnvollere, UTF-8-taugliche Einstellungen vor.



Bei allen anderen Distributionen können Sie zwar auch UTF8-Datenbanken und internationale Sortierordnungen verwenden, aber Sie müssen beim Erzeugen neuer Datenbanken und Tabellen explizit das gewünschte Format angeben. Das wird oft vergessen und führt z. B. dazu, dass Österreich auf unzähligen Webseiten in der Auswahlliste für den Staat an letzter Stelle angeführt wird (nach den Staaten, die mit Z beginnen). Das entspricht der schwedischen Sortierordnung – aber nicht dem, was man in deutschsprachigen Ländern erwarten würde.

Abhilfe für neue Datenbanken schaffen die folgenden beiden Zeilen im `[mysqld]`-Abschnitt der Konfigurationsdateien:

```
character_set_server=utf8mb4
collation_server=utf8mb4_general_ci
```

MySQL und MariaDB unterstützen IPv6. Allerdings ist der Datenbank-Server standardmäßig so vorkonfiguriert, dass IPv6-Verbindungen abgelehnt werden. Wenn Sie sowohl IPv4- als auch IPv6-Verbindungen zulassen möchten, müssen Sie in `my.cnf` die Einstellung `bind-address=::` verwenden. `bind-address>:::1` lässt ausschließlich IPv6-Verbindungen durch `localhost` zu. Beachten Sie, dass ein IPv6-Verbindungsaufbau nur gelingt, wenn Sie die betreffende IP-Adresse vorher mit dem SQL-Kommando `GRANT` oder durch eine direkte Änderung an der `host`-Spalte der Tabelle `mysql.user` zugelassen haben. IPv6

## Zugriffsabsicherung

In aktuellen MySQL- und MariaDB-Versionen gibt es zwei unterschiedliche Verfahren zur Authentifizierung beim Datenbank-Login:

- ▶ **Kombination aus Name, Host und Passwort:** Beim Verbindungsaufbau werden drei Informationen ausgewertet: der Login-Name, der Name des Hosts, von dem aus die Verbindung hergestellt wird (oft `localhost`) und schließlich ein Passwort. Stimmen alle drei Daten mit einer in MySQL gespeicherten Kombination überein, wird der Zugriff grundsätzlich erlaubt.

Dabei ist zu beachten, dass die MySQL-Login-Namen vollkommen losgelöst von denen des Linux-Systems verwaltet werden. Es gibt allerdings einen Standardbenutzer, dessen Name Ihnen vertraut ist: Der MySQL-Benutzer `root` hat so wie der Linux-Benutzer `root` uneingeschränkte Rechte.

- ▶ **Verbindung zu einem Linux-Account:** Bei dieser Variante besteht eine 1:1-Zuordnung zwischen dem Namen eines Linux-Accounts und dem eines MySQL-Anwenders. Besonders oft wird diese Zuordnung für `root` hergestellt. In diesem Fall darf, wer auf dem Linux-Rechner als `root` angemeldet ist, ohne weiteren Login auch den MySQL-Server als `root` administrieren.

Hinter den Kulissen ist für diese Art der Authentifizierung unter MySQL das Plugin `auth_socket` zuständig, unter MariaDB aber das Plugin `unix_socket`.

Welches Authentifizierungsverfahren zur Anwendung kommt, hängt vom Inhalt der Tabelle *user* der Datenbank *mysql* ab. Die Datenbank *mysql* wird bei jeder Installation eines MySQL- oder MariaDB-Servers standardmäßig eingerichtet, um dort diverse Einstellungen des Servers zu speichern.

Die *user*-Tabelle dieser Datenbank legt fest, wer sich wie beim Datenbank-Server anmelden kann. Grundsätzlich können Sie diese Tabelle nach Ihren eigenen Vorstellungen frei erweitern – die Frage ist aber, welche Einstellungen nach der Installation des Datenbank-Servers anfänglich gelten. Und das ist wieder einmal stark von der Linux-Distribution abhängig, weswegen ich auf dieses Detail hier relativ ausführlich eingehen muss.

Erst wenn ein Verbindungsaufbau zur Datenbank überhaupt möglich ist, entscheiden weitere Regeln (sogenannte Privilegien) darüber, welche Operationen Sie ausführen dürfen, welche Datenbanken Sie verändern dürfen etc.

#### MySQL versus *mysql* versus `mysql`

Der Begriff »MySQL« hat je nach Kontext unterschiedliche Bedeutungen, weswegen ich die Begriffe unterschiedlich formatiere. MySQL in dieser Schreibweise bezeichnet den Datenbank-Server.

*mysql* in kursiver Auszeichnung bezeichnet eine Datenbank, in der der MySQL- bzw. MariaDB-Server diverse Metadaten speichert – unter anderem die vorhin erwähnten Zugriffsrechte.

`mysql` ist ein Kommando bzw. ein Client, das bzw. der eine Verbindung zum Datenbank-Server herstellen kann. Sobald eine Verbindung besteht, können mit `mysql` SQL-Kommandos ausgeführt werden. `mysql` wird oft für administrative Aufgaben verwendet.

### MySQL-Absicherung je nach Distribution

#### Debian 8 und Ubuntu

Unter Ubuntu sowie unter Debian 8 kommt standardmäßig ein MySQL-Server in der Version 5.n zum Einsatz. Während der Installation erscheint ein Dialog, in dem Sie das Passwort für den MySQL-Benutzer *root* zweimal angeben müssen. Dieses Passwort gilt nur für Logins vom lokalen Rechner aus (Authentifizierungsverfahren 1). Um die Authentifizierung auszuprobieren, führen Sie als beliebiger Linux-Benutzer das folgende Kommando aus:

```
user$ mysql -u root -p
Enter password: ***** (das Passwort, das Sie bei der Installation
des MySQL-Servers angegeben haben)
Server version: 5.7.17-0ubuntu1 (Ubuntu)
...
mysql> exit
```

Debian 8 sowie aktuelle Ubuntu-Versionen bis einschließlich 17.04 richten außerdem den MySQL-Benutzer `debian-sys-main` ein und versehen ihn mit einem zufälligen Passwort. Dieses befindet sich im Klartext in der Datei `/etc/mysql/debian.cnf`, die nur `Linux-root` lesen kann. Der MySQL-Benutzer `debian-sys-main` wird in einigen Scripts verwendet, die z. B. nach einem Update ausgeführt werden, und darf nicht deaktiviert werden! Nach einer Passwortänderung müssen Sie auch `debian.cnf` entsprechend aktualisieren.

Sofern Sie bei der Installation des MySQL-Servers ein vernünftiges Passwort angegeben haben, ist diese Defaultkonfiguration sicher und kompatibel mit den Installationsverfahren anderer Pakete bzw. Programme, die den MySQL-Server nutzen.

Unter Debian 9 wird MariaDB so konfiguriert, dass der lokale Linux-Benutzer `root` gleichzeitig auch Datenbankadministrator ist. `root` kann ohne Passwort einen Verbindungsaufbau zum MariaDB-Server herstellen (Authentifizierungsverfahren 2). Um die Authentifizierung auszuprobieren, melden Sie sich zuerst in Debian als `root` an und führen das Kommando `mysql` dann ohne weitere Optionen aus:

Debian 9

```
root# mysql
Server version: 10.1.22-MariaDB- Debian 9.0
...
MariaDB> exit
```

Auch diese Konfiguration ist sicher. Sie erlaubt aber nur dem Linux-Benutzer `root` einen Zugriff auf den MySQL-Server. Das ist insbesondere für Webanwendungen zu wenig, die im Account `www-data` ausgeführt werden. Bevor Sie den MySQL-Server auf irgendeine Weise nutzen oder durch `phpMyAdmin` administrieren können, müssen Sie weitere MySQL-Benutzer einrichten. Zweckmäßig ist es in der Regel, einen zweiten Administrator-Login für `root2` vorzusehen, der mit einem Passwort abgesichert ist (Authentifizierungsverfahren 1). Dazu stellen Sie nochmals mit `mysql` eine Verbindung zum Server her und führen dann das folgenden SQL-Kommando aus:

```
root# mysql
MariaDB> GRANT ALL ON *.* TO root2@localhost
        IDENTIFIED BY 'geheim' WITH GRANT OPTION;
MariaDB> exit
```

Beachten Sie, dass Sie `root2` und nicht `root` angeben; andernfalls überschreiben Sie die Standardkonfiguration für `root`. Den `root2`-Login können Sie nun unter einem beliebigen Debian-Account ausprobieren:

```
user$ mysql -u root -p
Enter password: ***** (Passwort wie beim GRANT-Kommando)
```

### Verwenden Sie unterschiedliche Passwörter für Linux-root und MySQL-root!

Die Passwörter des Linux-Systems und die Passwörter des MySQL- oder MariaDB-Servers werden getrennt voneinander verwaltet. Egal, welche Distribution und welcher Datenbank-Server bei Ihnen läuft: Verwenden Sie aus Sicherheitsgründen in den beiden Systemen nie dieselben Passwörter!

MySQL bzw. MariaDB verwendet ausgesprochen schlechte Hash-Algorithmen zur Verschlüsselung der Passwörter. Bei synchronen Passwörtern könnte ein Sicherheitsproblem im Datenbank-Server dem Angreifer auch das root-Passwort für den Linux-Account verraten.

CentOS 7,  
Fedora 26,  
openSUSE 42.2  
und RHEL 7

Schon seit einem Jahrzehnt ist die Defaultkonfiguration des MySQL- bzw. MariaDB-Servers in CentOS, Fedora, openSUSE und RHEL sicherheitstechnisch haarsträubend (zuletzt getestet mit den nebenstehenden Versionen): Der MySQL-Benutzer root ist standardmäßig eingerichtet, aber mit keinem Passwort ausgestattet!

Darüber hinaus lässt die Konfiguration auch noch sogenannte anonyme Benutzer zu: Damit kann sich – losgelöst vom root-Login – jeder ohne Passwort anmelden. Anonyme Benutzer haben zwar in der Folge keine Zugriffsrechte auf eine Datenbank. Dennoch ist es nicht zweckmäßig, diese MySQL-Accounts so zu belassen.

Um zum einen root mit einem Passwort abzusichern und zum anderen die anonymen MySQL-Benutzer zu löschen, stellen Sie mit `mysql -u root` eine Verbindung zum Datenbank-Server her und führen dann die folgenden Kommandos aus. Dabei ersetzen Sie `geheim` natürlich durch ein eigenes Passwort, wobei die obige Warnbox selbstverständlich weiter gilt.

```
user$ mysql -u root
mysql> UPDATE mysql.user SET password=PASSWORD('geheim')
      WHERE user='root' AND plugin='';
mysql> DELETE FROM mysql.user
      WHERE user='' AND plugin='';
mysql> FLUSH PRIVILEGES;
mysql> exit
```

Von nun an müssen Sie sich beim Start von `mysql` anmelden:

```
user$ mysql -u root -p
Enter password: ***** (das Passwort, das Sie mit UPDATE eingestellt haben)
```

### Weitere MySQL-Benutzer einrichten und die Absicherung kontrollieren

Benutzer für eine  
Datenbank  
einrichten

Wie unter Linux sollten Sie auch innerhalb von MySQL bzw. MariaDB nur als root arbeiten, um Administratorarbeiten zu erledigen. Für den Zugriff auf einzelne Datenbanken richten Sie am besten jeweils eigene MySQL-Accounts ein. Dazu melden Sie

sich als MySQL-root an und führen dann ein SQL-Kommando nach dem folgenden Muster aus. Dabei müssen Sie natürlich den Datenbanknamen, den Account-Namen und das Passwort anpassen!

```
root# mysql [-u root -p]
mysql/MariaDB> GRANT ALL ON datenbankname.*
                TO accountname@localhost IDENTIFIED BY 'ganzgeheim';
```

Wenn Sie sich rasch einen Überblick verschaffen möchten, welche MySQL-Benutzer wie abgesichert sind, werfen Sie mit dem folgenden SELECT-Kommando einen Blick in die Tabelle *mysql.user*. Die Konfiguration ist nur sicher, wenn es für jede Kombination aus user und host entweder einen Passwort-Hashcode oder einen Eintrag in der Plugin-Spalte gibt. Wenn Sie z. B. MariaDB unter CentOS 7 wie oben beschrieben abgesichert haben, sollte das Ergebnis so aussehen:

```
user$ mysql -u root -p
Enter password: ***** (das Passwort von MySQL/MariaDB-root)
MariaDB> SELECT user, host, password, plugin FROM mysql.user;
```

| user        | host             | password     | plugin |
|-------------|------------------|--------------|--------|
| root        | localhost        | *64236691... |        |
| root        | centos-buch.info | *64236691... |        |
| root        | 127.0.0.1        | *64236691... |        |
| root        | :::1             | *64236691... |        |
| accountname | localhost        | *325D19F9... |        |

```
4 rows in set (0.00 sec)
```

Wenn Sie unter Debian 9 root2 eingerichtet haben, sollte das SELECT-Kommando dieses Ergebnis liefern:

```
root# mysql
mysql> SELECT user, host, password, plugin FROM mysql.user;
```

| user        | host      | password     | plugin      |
|-------------|-----------|--------------|-------------|
| root        | localhost |              | unix_socket |
| root2       | localhost | *AFFCO...    |             |
| accountname | localhost | *325D19F9... |             |

```
4 rows in set (0.00 sec)
```

In MySQL-Versionen ab 5.7 (nicht in MariaDB!) wurde die Spalte *password* in *authentication\_string* umbenannt. Außerdem enthält die *plugin*-Spalte den Eintrag *mysql\_native\_password*, wenn das Authentifizierungsverfahren 1 mit Passwort zum Einsatz kommt. Bei aktuellen Ubuntu-Versionen müssen Sie daher das SELECT-Kommando ein wenig anpassen:

```

user$ mysql -u root -p
Enter password: ***** (das Passwort von MySQL/MariaDB-root)
MariaDB> SELECT user, host, authentication_string, plugin FROM mysql.user;
+-----+-----+-----+-----+
| user          | host      | authentication_string | plugin          |
+-----+-----+-----+-----+
| root          | localhost | *AFFC054603403E683... | mysql_native_password |
| mysql.sys     | localhost | *THISISNOTAVALIDPA... | mysql_native_password |
| debian-sys-maint | localhost | *25389D1A4B4D3D2AC... | mysql_native_password |
| accountname   | localhost | *325D19FC0542341F9... | mysql_native_password |
+-----+-----+-----+-----+

```

### Erste Tests

#### Neue Datenbank, neuer Benutzer

Um MySQL zu testen, müssen Sie die Datenbanksprache SQL kennen, was ich hier voraussetze. Das Ziel der folgenden Kommandos besteht darin, die neue Datenbank `mydatabase` und einen neuen Nutzer `newuser` zu schaffen, der auf diese Datenbank zugreifen darf. Für derartige Arbeiten setzen Sie am einfachsten das Programm `mysql` ein. Es hat eine vergleichbare Aufgabe wie eine Linux-Shell: Es nimmt SQL-Kommandos entgegen, leitet diese an den MySQL- oder MariaDB-Server weiter und zeigt schließlich das Ergebnis an. Dabei müssen alle SQL-Kommandos mit einem Strichpunkt enden.

```

user$ mysql -u root -p
Enter password: *****
...
mysql> CREATE DATABASE mydatabase;
mysql> GRANT ALL ON mydatabase.* TO newuser@localhost
IDENTIFIED BY 'xxxxxxxxx';
mysql> exit

```

Alle weiteren Tests mit der neuen Datenbank kann nun der MySQL-Benutzer `newuser` durchführen. Wie unter Linux ist es auch in MySQL zweckmäßig, so wenig wie möglich als `root` zu arbeiten.

#### Tabelle erzeugen und mit Daten füllen

Mit den folgenden Kommandos erzeugt `newuser` eine neue Tabelle (`CREATE TABLE`), fügt darin einige Datensätze ein (`INSERT`) und sieht sich schließlich alle Datensätze an (`SELECT`). Bei der Tabelle spielt die Spalte `id` eine besondere Rolle: Der MySQL-Server fügt dort für jeden neuen Datensatz selbstständig eine eindeutige Zahl ein. Diese Zahl dient zur Identifizierung des Datensatzes.

```

user$ mysql -u newuser -p
Enter password: *****
mysql> USE mydatabase;
mysql> CREATE TABLE mytable (
        id INT NOT NULL AUTO_INCREMENT,

```

```

        txt VARCHAR(100),
        n INT,
        PRIMARY KEY(id));
mysql> INSERT INTO mytable (txt, n) VALUES('abc', 123);
mysql> INSERT INTO mytable (txt, n) VALUES('efgsd', -4);
mysql> INSERT INTO mytable (txt, n) VALUES(NULL, 0);
mysql> SELECT * FROM mytable;
+----+-----+-----+
| id | txt  | n    |
+----+-----+-----+
|  1 | abc  | 123  |
|  2 | efgsd | -4   |
|  3 | NULL | 0    |
+----+-----+-----+
mysql> exit

```

## 33.2 Administrationswerkzeuge

Für MySQL und MariaDB gibt es unzählige Administrationswerkzeuge. Standardmäßig stehen die Kommandos `mysql`, `mysqldump` sowie einige weitere, textbasierte Werkzeuge zur Verfügung. Optional können Sie diverse weitere Programme installieren, die aber teilweise einen grafischen Desktop voraussetzen. Dieser Abschnitt gibt einen kurzen Überblick über die wichtigsten Werkzeuge. Nicht berücksichtigt ist das Backup-Kommando `mysqldump`, zu dem in [Abschnitt 33.3](#), »Backups«, weitere Informationen folgen.

### mysql

Hinter dem Programm `mysql` verbirgt sich nicht etwa der Datenbank-Server (der hat den Programmnamen `mysqld`), sondern ein Kommandozeilen-Client. Damit können Sie eine Verbindung zum MySQL- oder MariaDB-Server herstellen und dann SQL-Kommandos ausführen. Soweit es sich dabei um `SELECT`-Kommandos handelt, zeigt das Programm die Abfrageergebnisse im Textmodus an.

Beim Start von `mysql` geben Sie normalerweise mit `-u` den MySQL-Benutzernamen an. Die Option `-p` bewirkt, dass Sie nach dem Start das dazugehörige Passwort angeben können. Beim Authentifizierungsverfahren 2 (Zuordnung zwischen MySQL- und Linux-Accounts) entfällt diese Option.

Wenn der MySQL-Server nicht auf dem lokalen Rechner läuft, geben Sie den Hostnamen mit `-h name` an. Optional können Sie auch eine Standarddatenbank angeben, die dann per Default für alle weiteren SQL-Kommandos gilt:

```
user$ mysql -u root -p meinedatenbank
Enter password: *****
```

Anschließend können Sie interaktiv SQL-Kommandos eingeben, die Sie durch einen Strichpunkt abschließen. Das `mysql`-spezifische Kommando `status`, das auch ohne Strichpunkt ausgeführt werden kann, zeigt Informationen zur aktuellen Verbindung sowie Eckdaten des MySQL-Servers an. `[Strg]+[D]` beendet das Programm.

```
mysql> status
mysql Ver 15.1 Distrib 10.1.21-MariaDB, for Linux (x86_64) using readline 5.1
Connection id:          319042
...
Server:                 MariaDB
Server version:        10.1.21-MariaDB MariaDB Server
Protocol version:      10
Connection:            Localhost via UNIX socket
Server characterset:   utf8
Db characterset:       utf8
Client characterset:   utf8
Conn. characterset:    utf8
UNIX socket:          /var/run/mysqld/mysqld.sock
Uptime:               101 days 10 hours 26 min 55 sec
Threads: 1            Questions: 9479450  Slow queries: 1067  Opens: 88575
Flush tables: 4       Open tables: 400    Queries per second avg: 1.081
```

#### mysql im Batch-Modus

Für administrative Zwecke und insbesondere zum Einspielen von Backups kann `mysql` auch SQL-Kommandos aus einer `*.sql`-Datei verarbeiten:

```
user$ mysql -u root -p meinedatenbank < kommandos.sql
Enter password: *****
```

### mysqldadmin

`mysqldadmin` ermöglicht es, verschiedene administrative Aufgaben in Form eines Kommandos auszuführen. Zu allen `mysqldadmin`-Kommandos gibt es auch gleichwertige SQL-Kommandos (z. B. `CREATE DATABASE`). Der Vorteil von `mysqldadmin` besteht darin, dass es dabei hilft, wiederkehrende Aufgaben durch Scripts zu automatisieren.

Wie bei `mysql` geben Sie mit der Option `-u` den Benutzernamen und mit `-h` den Hostnamen (standardmäßig `localhost`) an. `-p` ohne weitere Angaben führt zur einer interaktiven Passwortabfrage. Die können Sie durch `-ppassword` ohne ein Leerzeichen nach `-p` vermeiden. Diese Bequemlichkeit hat aber den Nachteil, dass das Passwort im Klartext in der Prozessliste zu sehen ist.

Eine alternative Vorgehensweise besteht darin, die Login-Daten in einer Passwortdatei zu speichern bzw. auf den unter Debian 8 und Ubuntu bereits vordefinierten MySQL-Benutzer `debian-sys-maint` zurückzugreifen. Dessen Passwortdatei `/etc/`



mysql/debian.cnf ist allerdings nur für den Linux-root lesbar. Der Aufruf von mysql-admin sieht dann so aus:

```
root# mysqladmin --defaults-file=/etc/mysql/debian.cnf kommando ...
```

Einen Überblick über die für mysqladmin verfügbaren Kommandos gibt mysqladmin --help. Die folgenden Zeilen geben einige Beispiele: Das erste Kommando erzeugt eine neue Datenbank, das zweite liefert eine Liste aller MySQL-Statusvariablen und das dritte gibt eine Liste aller aktiven MySQL-Threads (Verbindungen) zurück.

```
user$ mysqladmin -u root -p create neuedatenbank
Enter password: *****
user$ mysqladmin -u root -p extended-status
...
user$ mysqladmin -u root -p processlist
...
```

## MySQL Workbench

MySQL stellt mit der MySQL Workbench (siehe [Abbildung 33.1](#)) ein hochwertiges Administrationsprogramm mit grafischer Benutzeroberfläche zur Verfügung.

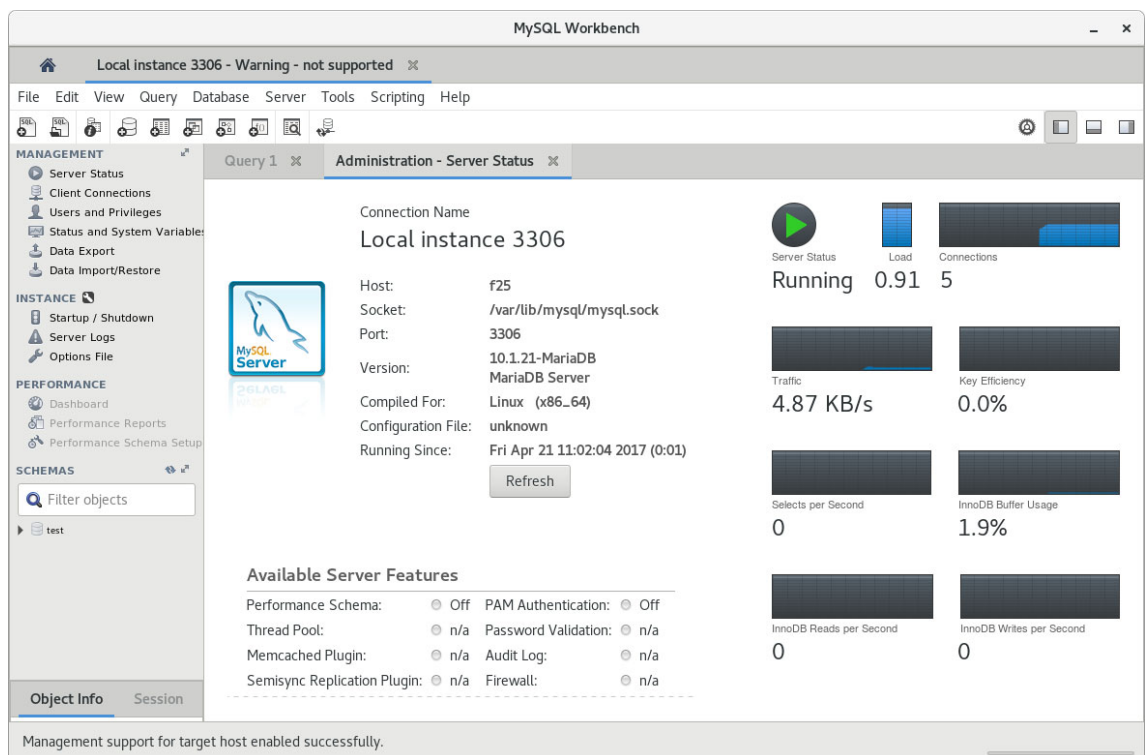


Abbildung 33.1 Die MySQL Workbench

Sie können mit der MySQL Workbench den Datenbank-Server überwachen, seine Einstellungen ändern, Benutzerrechte einstellen, neue Datenbanken einrichten, vorhandene Datenbanken auslesen, verändern und sichern, Datenbankschemas entwickeln etc. Die meisten Funktionen des Programms stehen gleichermaßen für MySQL und MariaDB zur Verfügung; einzig die Konfigurationsfunktionen zur Veränderung von `my.cnf` sind MySQL-spezifisch und bei MariaDB-Installationen gesperrt.

Die meisten Distributionen stellen für die MySQL Workbench eigene Pakete zur Verfügung. Zu den prominenten Ausnahmen zählen CentOS, Fedora und RHEL. Glücklicherweise gibt es auf der MySQL-Webseite geeignete Pakete, die sich mühelos installieren lassen:

<https://dev.mysql.com/downloads/tools/workbench>

### phpMyAdmin

Das bekannteste MySQL-Administrationsprogramm ist phpMyAdmin. Sein größter Vorteil besteht darin, dass das Programm über einen Webbrowser bedient wird. Der eigentliche phpMyAdmin-Code läuft direkt auf dem Webserver. Deswegen funktioniert phpMyAdmin auch dann, wenn der MySQL-Server aus Sicherheitsgründen so konfiguriert ist, dass keine Verbindungen über das Netzwerk zulässig sind.

Bei vielen Distributionen gibt es fertige phpMyAdmin-Pakete, wobei Sie unter Umständen nach der Installation `systemctl reload apache2/httpd` ausführen müssen.

Alternativ können Sie phpMyAdmin von der Website <https://www.phpmyadmin.net> herunterladen. Anschließend installieren Sie alle Dateien in ein Verzeichnis, auf das der Webserver Apache zugreifen kann. Bei dieser Installationsvariante müssen Sie sich allerdings selbst regelmäßig um Updates kümmern.

Nach der Installation erreichen Sie die Administrationsoberfläche in der Regel über die Adresse <http://hostname/phpMyAdmin> (siehe [Abbildung 33.2](#)).

#### Debian, Ubuntu

Unter Debian und Ubuntu gelten einige Besonderheiten: Alle Konfigurationsdateien befinden sich in `/etc/phpmyadmin`. Sofern Sie Apache als Webserver verwenden, kümmert sich `apache.conf` darum, das tatsächliche Installationsverzeichnis `/usr/share/phpmyadmin` durch einen Alias auf das Webverzeichnis `phpmyadmin` abzubilden. Die Sicherheitseinstellungen in `apache.conf` betreffen standardmäßig nur das Setup-Script von phpMyAdmin, das Sie normalerweise aber gar nicht brauchen.

`apache.conf` wird durch einen symbolischen Link in `/etc/apache2/conf.d` in die Apache-Konfiguration integriert. Wenn Sie über das Protokoll HTTPS mit phpMyAdmin kommunizieren möchten, löschen Sie diesen Link und fügen dann eine Include-Anweisung für `apache.conf` in Ihre HTTPS-Konfigurationsdatei ein:

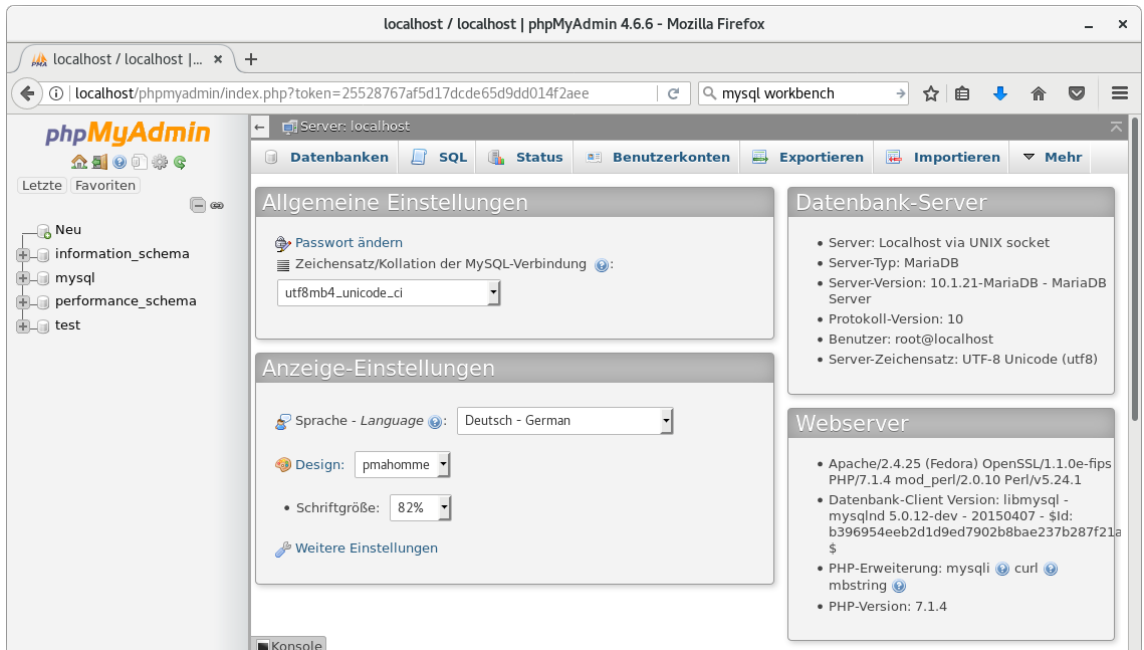


Abbildung 33.2 MySQL mit phpMyAdmin im Webbrowser administrieren

```
<VirtualHost _default_:443>
...
    Include /etc/phpmyadmin/apache.conf
</VirtualHost>
```

phpMyAdmin ist ein beliebtes Einfallstor für Cracker. Es gibt im Internet zahllose automatisierte Tools, die Webserver nach schlecht abgesicherten oder veralteten phpMyAdmin-Installationen absuchen. Ein fehlendes oder leicht erratbares MySQL-root-Passwort gibt dem Angreifer MySQL-Administratorrechte! Ergreifen Sie daher die folgenden Vorsichtsmaßnahmen:

**phpMyAdmin  
absichern**

- ▶ Sichern Sie die MySQL-Installation ab, bevor Sie phpMyAdmin installieren (siehe [Abschnitt 33.1](#), »Installation und Inbetriebnahme«).
- ▶ Geben Sie dem phpMyAdmin-Verzeichnis einen Alias, der nicht ganz leicht erraten werden kann. <http://mysite.de/pMa1> ist sicherer als <http://mysite.de/phpmyadmin>.
- ▶ Akzeptieren Sie für das phpMyAdmin-Verzeichnis nach Möglichkeit nur das sichere https-Protokoll, nicht http.
- ▶ Sichern Sie das phpMyAdmin-Verzeichnis auch auf Apache-Ebene durch ein Passwort ab, z. B. durch eine `.htaccess`-Datei (siehe [Abschnitt 32.2](#), »Webverzeichnisse einrichten und absichern«).

- ▶ Achten Sie darauf, dass Ihre phpMyAdmin-Version aktuell ist. Bei einigen Distributionen werden die phpMyAdmin-Pakete leider nur mangelhaft gewartet. Dann ist es sicherer, anstelle der Distributionspakete phpMyAdmin manuell zu installieren und regelmäßig zu aktualisieren.

### 33.3 Backups

Auch wenn Sie nicht vorhaben, sich auf die Datenbankadministration zu spezialisieren, sollten Sie wissen, wie Sie ein fehlerfreies Backup einer MySQL- oder MariaDB-Datenbank durchführen. Dabei gibt es mehr Varianten und Spielarten, als man es für möglich halten möchte. Dieser Abschnitt stellt das Kommando `mysqldump` vor. Wenn Sie außer den zu bestimmten Zeiten durchgeführten Backups auch kontinuierliche Backups benötigen, aktivieren Sie das (binäre) Logging. Damit wird jede Änderung an der Datenbank in einer Logging-Datei festgehalten. Die Logging-Dateien können auch als Basis für die Replikation der Datenbank auf einen zweiten Server verwendet werden.

**Tabellentypen** Die Vielfalt der MySQL-Backup-Verfahren hat unter anderem damit zu tun, dass MySQL verschiedene Tabellentypen kennt: Beim Anlegen einer Tabelle kann der Datenbankentwickler bzw. das Programm zwischen verschiedenen Tabellentypen wählen. Es ist sogar möglich, innerhalb einer Datenbank Tabellen mit unterschiedlichen Typen zu verwenden. Die beiden wichtigsten Typen sind momentan MyISAM und InnoDB, außerdem Aria bei MariaDB. Je nach Tabellentyp stehen nicht nur unterschiedliche Zusatzfunktionen zur Verfügung (Transaktionen, Foreign-Key-Regeln, Volltextsuche), auch das zugrunde liegende Dateiformat und das optimale Backup-Verfahren variieren.

Wenn Sie nicht wissen, welche Datenbanken Ihr MySQL-Server verwaltet und welchen Typ die darin enthaltenen Tabellen aufweisen, führen Sie im Kommandozeilen-Client `mysql` das folgende SQL-Kommando aus. Das resultierende Ergebnis listet alle Tabellen auf. Die Spalte `table_schema` gibt dabei den Datenbanknamen an, `engine` den Tabellentyp.

```
mysql> SELECT table_schema, table_name, engine
        FROM information_schema.tables
        WHERE table_type='BASE TABLE' AND table_schema='dbname'
        ORDER BY table_schema, table_name;
```

| table_schema | table_name | engine |
|--------------|------------|--------|
| mylibrary    | authors    | InnoDB |
| mylibrary    | categories | InnoDB |
| mylibrary    | counters   | MyISAM |

```
mylibrary    fulltitles   MyISAM
mylibrary    languages   InnoDB
...
```

## mysqldump

Das zum Lieferumfang von MySQL zählende Kommando `mysqldump` erstellt ein Backup einer MySQL-Datenbank in Form von SQL-Anweisungen. Die resultierende Datei kann später mit `mysql` wieder in eine bereits vorhandene Datenbank eingespielt werden. Die prinzipielle Syntax sieht so aus:

```
user$ mysqldump -u root -p [optionen] datenbankname > backup.sql
```

Details des Backups können Sie durch zahllose Optionen steuern (siehe auch `mysqldump --help`). Die optimale Kombination von Optionen hängt unter anderem davon ab, in welchem Format die Tabellen Ihrer MySQL-Datenbank vorliegen, ob sie also im MyISAM- oder im InnoDB-Format gespeichert sind.

Bei MyISAM-Tabellen reicht die Option `--lock-all-tables`. Sie bewirkt, dass `mysqldump` am Beginn des Backups *alle* Tabellen durch ein `LOCK`-Kommando blockiert und erst am Ende des Backups wieder freigibt. Standardmäßig führt `mysqldump` das Locking hingegen Tabelle für Tabelle aus, also immer nur für die Tabelle, die gerade bearbeitet wird. Das kann aber zur Folge haben, dass sich einzelne Tabellen während des Backups verändern und dass die Verknüpfungen zwischen den Tabellen letztlich nicht mehr stimmen.

**Backup von  
MyISAM-Tabellen**

```
user$ mysqldump -u root -p --lock-all-tables dbname > backup.sql
```

Wesentlich mehr Optionen brauchen Sie für ein Backup einer Datenbank mit InnoDB-Tabellen: Mit `--skip-opt` deaktivieren Sie einige, speziell für MyISAM gedachte Standardoptionen. `--single-transaction` bewirkt, dass das gesamte Backup im Rahmen einer Transaktion durchgeführt wird. Damit ist ausgeschlossen, dass sich während des Backups Daten ändern, was zu inkonsistenten Verknüpfungen zwischen den Tabellen führen kann. `--disable-keys` bewirkt, dass beim späteren Einlesen der Daten vorübergehend die Indexaktualisierung deaktiviert wird. Der Index wird erst zum Schluss vollständig neu erzeugt, was wesentlich schneller ist. Dank `--create-options` verwendet `mysqldump` bei der Ausgabe alle MySQL-spezifischen Optionen des `CREATE-TABLE`-Kommandos.

**Backup von  
InnoDB-Tabellen**

`--quick` bewirkt, dass `mysqldump` die Tabellen Datensatz für Datensatz vom Server abholt, anstatt sie alle auf einmal zu lesen. Das ist bei großen Tabellen effizienter. Aufgrund von `--extended-insert` erzeugt `mysqldump` `INSERT`-Kommandos, die mehrere Datensätze auf einmal einfügen, was einerseits die Größe der Backup-Datei ein wenig reduziert und später das Wiedereinspielen der Daten beschleunigt. `--add-drop-table` bewirkt, dass `mysqldump` jedem `CREATE-TABLE`-Kommando ein `DROP-TABLE`-Kommando

voranstellt. Das vermeidet Fehler, wenn – z. B. aufgrund eines unvollständig eingespielten Backups – einzelne Tabellen bereits in der Datenbank existieren.

```
user$ mysqldump -u root -p --skip-opt --single-transaction \
    --disable-keys --create-options --quick \
    --extended-insert --add-drop-table dbname > backup.sql
```

Bleibt noch die naheliegende Frage, wie Sie am besten vorgehen, wenn eine Datenbank sowohl MyISAM- als auch InnoDB-Tabellen enthält. In diesem Fall verwenden Sie `mysqldump` einfach ohne weitere Optionen. Falls die Datenbank während des Backups genutzt wird, kann es nun aber passieren, dass sich während des Backups Tabellen verändern. Wenn Sie Pech haben, ist das Backup dann inkonsistent, d. h., miteinander verknüpfte Tabellen enthalten Verweise auf nicht mehr existente Datensätze. Dieses Problem lässt sich mit `mysqldump` nicht umgehen. Am besten verwenden Sie stattdessen ein anderes Backup-Verfahren (siehe den folgenden Abschnitt zu `mylvmbackup`).

Wenn Sie ein Backup *aller* Datenbanken erstellen (nicht ein Backup einer bestimmten Datenbank), geben Sie die Option `--all-databases` an.

**SPs und Trigger** Beachten Sie, dass `mysqldump` standardmäßig weder Stored Procedures (SPs) noch Trigger speichert. Wenn Sie das wünschen, geben Sie zusätzlich die Optionen `--routines` und `--triggers` an.

**Datenbank wiederherstellen** Um eine Datenbank aus einem Backup wiederherzustellen, erzeugen Sie zuerst die betreffende Datenbank (falls es sie noch nicht gibt). Anschließend übergeben Sie die Backup-Datei an `mysql`. Dabei stellt die Option `-default-character-set` sicher, dass die im UTF8-Format gespeicherten Zeichenketten auch tatsächlich in diesem Format gelesen werden.

```
user$ mysqladmin create dbname
user$ mysql -u root -p --default-character-set=utf8 dbname < backup.sql
```

**Komprimierte Backups** Mit `mysqldump` erzeugte Backup-Dateien sind aufgrund des Textformats sehr groß. Dieses Problem umgehen Sie am einfachsten dadurch, dass Sie die Backups sofort komprimieren bzw. beim Wiedereinspielen direkt dekomprimieren. Die entsprechenden Kommandos sehen so aus:

```
user$ mysqldump [optionen] dbname | gzip -c > backup.sql.gz
user$ gunzip -c backup.sql.gz | mysql [optionen] dbname
```

Die Komprimierung durch `gzip` kostet allerdings eine Menge CPU-Ressourcen. Wenn Sie Rechenzeit sparen möchten und sich dafür mit etwas größeren Backup-Dateien abfinden können, empfiehlt sich der Einsatz des Komprimierkommandos `lzop` aus dem gleichnamigen Paket:

```
user$ mysqldump [optionen] dbname | lzop -c > backup.sql.lzo
user$ lzop -c -d backup.sql.lzo | mysql [optionen] dbname
```

## Backup-Tools und -Varianten

Das Kommando `mysqldump` funktioniert für kleine Datenbanken ausgezeichnet. Allerdings sind die betroffenen Tabellen während des Backups nur eingeschränkt verwendbar – bei MyISAM-Tabellen aufgrund des Lockings, bei InnoDB-Tabellen aufgrund einer lange andauernden Transaktion. Für Datenbanksysteme, die im 24-Stunden-Betrieb unterbrechungsfrei laufen sollen, wird das mit zunehmender Datenbankgröße zu einem echten Problem.

Die optimale Lösung für dieses Problem wäre ein Hot-Backup-Verfahren, das störungsfrei im laufenden Betrieb funktioniert. MySQL bietet ein entsprechendes Backup-Programm an, dieses ist aber nur im Rahmen einer kostenpflichtigen MySQL-Enterprise-Lizenz verfügbar. Eine gute Alternative ist das Open-Source-Programm *XtraBackup*:

Hot-Backup

<https://www.percona.com/software/mysql-database/percona-xtrabackup>

Eine andere Möglichkeit bietet das hier vorgestellte Kommando `mylvmbackup`: Es minimiert die Zeit, während der die Datenbank blockiert ist, auf wenige Sekunden. Es setzt allerdings voraus, dass sich sämtliche Datenbankdateien in einem Logical Volume befinden. Das Script erstellt einen Snapshot dieses Logical Volumes und verwendet ihn, um alle Datenbankdateien in eine komprimierte Backup-Datei zu übertragen.

LVM-Backup

Das Perl-Script `mylvmbackup` kann bei vielen Distributionen als Paket installiert werden. Wenn Ihre Distribution kein entsprechendes Paket anbietet, laden Sie das Script von dieser Seite herunter:

<https://launchpad.net/mylvmbackup>

Das so erstellte Backup unterscheidet sich in zwei Punkten von `mysqldump`-Resultaten: Erstens erfasst es grundsätzlich *alle* Datenbanken inklusive aller Zusatzdaten wie Stored Procedures, Trigger, Zugriffsrechte etc. Und zweitens liegt es in binärer Form vor. Das hat zur Folge, dass nur alle Datenbanken zusammen wiederhergestellt werden können und dass zum Wiedereinspielen des Backups ein MySQL- oder MariaDB-Server in derselben Konfiguration und möglichst auch mit derselben Versionsnummer erforderlich ist.

Über ein Script können Sie Backups mit `mysqldump` oder mit anderen Werkzeugen automatisieren und so täglich oder wöchentlich ein Backup erstellen. Um einen möglichen Datenverlust im Katastrophenfall weiter zu minimieren, können Sie außerdem inkrementelle Backups aktivieren. Dazu fügen Sie in `/etc/mysql/my.cnf` die folgende Zeile ein und starten den Datenbank-Server dann neu:

Inkrementelle Backups durch binäres Logging

```
# Änderung in /etc/mysql/my.cnf
log_bin = /var/log/mysql/mysql-bin.log
```

Der Datenbank-Server protokolliert nun alle SQL-Kommandos, die Daten verändern. Die Logging-Dateien werden automatisch durchnummeriert (`mysql-bin.000001`, `.000002` etc.). Da sich immer nur die letzte Datei ändert, ist es relativ einfach, diese Dateien in kurzen Abständen in ein Backup-Verzeichnis zu übertragen (z. B. mit `rsync`).

Wenn Sie Ihre Datenbanken wiederherstellen müssen, führen Sie zuerst die oben beschriebenen Restore-Schritte für das letzte Komplett-Backup aus. Anschließend spielen Sie mit `mysqlbinlog` alle Änderungen ein, die seither aufgetreten sind. Die Kommandoabfolge sieht so aus:

```
root# mysqlbinlog --start-position=<p> mysql-bin.<n> | mysql -u root -p
root# mysqlbinlog mysql-bin.<n+1> | mysql -u root -p
root# mysqlbinlog mysql-bin.<n+2> | mysql -u root -p
..
```

Bevor Sie loslegen können, brauchen Sie noch zwei Informationen: Welche Nummer hat die erste Logging-Datei, die Sie berücksichtigen müssen (`<n>`)? Und mit welcher Position innerhalb der ersten Datei beginnen Sie (`<p>`)? `mylvmbackup` hat diese Informationen zum Glück im Backup-Archiv gespeichert. Wenn Sie die Daten wie oben beschrieben extrahiert haben, finden Sie die erforderlichen Informationen in den ersten zwei Zeilen der Datei `/etc/mysql/*_mysql.pos`:

```
root# less /etc/mysql/backup-20170628_145023_mysql.pos
Master:File=mysql-bin.000016
Master:Position=98
...
```

**Replikation** Ist das binäre Logging einmal aktiviert, ist es nur noch ein kleiner Schritt zur Replikation: Damit synchronisieren Sie einen zweiten MySQL-Server mit dem ersten und haben so jederzeit ein aktives zweites Datenbanksystem, das im Notfall das Hauptsystem ersetzen kann. Replikation ersetzt aber keine Backups, weil natürlich auch Kommandos wie `DROP TABLE` sofort synchronisiert werden! Eine Einführung in die Replikation von MySQL-Datenbanken gibt das MySQL-Handbuch:

<https://dev.mysql.com/doc/refman/5.7/en/replication.html>



# Kapitel 34

## Postfix und Dovecot

Dieses Kapitel beschreibt, wie Sie auf einem Root-Server einen E-Mail-Server einrichten. Damit können Sie alle Mitarbeiter einer Firma oder Organisation mit eigenen E-Mail-Adressen ausstatten. Jeder Mitarbeiter kann E-Mails per SMTP versenden und per POP/IMAP abholen bzw. lesen. Als SMTP-Server kommt dabei das Programm Postfix zum Einsatz, als POP/IMAP-Server und zur SMTP-Authentifizierung das Programm Dovecot. Diese Programme stehen unter allen gängigen Distributionen zur Verfügung. Selbst Red Hat hat mit RHEL 7 endlich dem unübersichtlichen und schwer zu konfigurierenden Programm Sendmail Ade gesagt und sieht nun als Default-E-Mail-Server Postfix vor.

Der Spam-Flut können Sie versuchen mit SpamAssassin Herr zu werden. (So viel gleich vorweg: Auch SpamAssassin bewirkt keine Wunder.) Wenn Ihre Mitarbeiter bzw. die Ihrer Auftraggeber mit Windows-PCs arbeiten, lohnt sich eventuell auch die Installation des Virenschutzprogramms ClamAV. Bevor Sie an die Arbeit gehen können, brauchen Sie einen Server mit einem international gültigen Hostnamen. Sie müssen in der Lage sein, dessen DNS-Einträge selbst zu konfigurieren (MX-Eintrag, Reverse DNS).

E-Mail ist ein wesentlich komplexeres Thema, als viele Einsteiger in diese Materie vermuten. Für jede Teilaufgabe stehen unterschiedliche Programme und Kommandos zur Auswahl, und es existieren schier unendlich viele Konfigurationsmöglichkeiten. Dieses Kapitel beschreibt deswegen zuerst die wesentlichen Grundlagen der E-Mail-Kommunikation und gibt Ihnen einen ersten Überblick über die zur Auswahl stehenden Werkzeuge.

### 34.1 Einführung und Grundlagen

E-Mail ist ganz einfach, oder? Aus der Sicht des Endanwenders stimmt das – zumindest, solange alles funktioniert. Hinter den Kulissen ist das E-Mail-System wesentlich komplexer, als es den Anschein hat. Es gibt viele Konfigurationsmöglichkeiten, die alle unter bestimmten Umständen ihre Berechtigung haben. Gute Bücher über E-Mail-Server wie Sendmail, Postfix oder Exim umfassen oft mehr als 1000 Seiten! Es liegt auf der Hand, dass ich hier nur auf die Grundkonfiguration eingehen kann.

## Komponenten eines E-Mail-Servers

Ein vollständiger E-Mail-Server besteht aus drei Komponenten:

- ▶ **MTA:** Der *Mail Transfer Agent* ist das, was umgangssprachlich als E-Mail-Server bezeichnet wird. Der MTA kümmert sich darum, E-Mails über das Internet zu versenden bzw. zu empfangen, wobei das Protokoll SMTP eingesetzt wird.

Die meisten Einsteiger in die Interna der E-Mail-Welt sind sich nicht darüber im Klaren, dass sich die Zuständigkeit des MTAs auf den Netzwerkverkehr beschränkt und am Server endet. Empfangene E-Mails werden an den MDA weitergegeben, der sich um die lokale Speicherung kümmert. Es ist *nicht* Aufgabe des MTAs, E-Mails zu einem Benutzer zu bringen, der in der Regel auf einem anderen Rechner arbeitet!

Beispiele: Courier, Cyrus, Exim, Postfix, Qmail, Sendmail

- ▶ **MDA:** Der *Mail Delivery Agent* kümmert sich um die lokale Zustellung von E-Mails, also um die Speicherung der beim MTA eintreffenden E-Mails in lokalen Postfächern. Ein »Postfach« meint in diesem Zusammenhang einfach ein Verzeichnis bzw. eine Datei auf dem Server.

Beispiele: Maildrop, Procmail

In einige MTAs ist ein MDA integriert bzw. wird mitgeliefert, z. B. das Kommando `local` bei Postfix. Die Programme Maildrop bzw. Procmail sind dennoch sehr populär, weil sie in der Regel noch mehr Konfigurationsmöglichkeiten bieten.

- ▶ **POP/IMAP-Server:** E-Mails werden selten direkt auf dem Server gelesen. Damit ein extern arbeitender Benutzer die E-Mails auf seinen lokalen Rechner übertragen bzw. von dort aus verwalten kann, haben sich die Protokolle POP und IMAP durchgesetzt. Zur Unterstützung dieser Protokolle muss auf dem Server ein POP- und/oder IMAP-Server eingerichtet werden. Bisweilen werden POP- und IMAP-Server zu den MDAs hinzugerechnet, was der ursprünglichen Bedeutung eines MDAs aber widerspricht und somit falsch ist.

Beispiel: Courier IMAP, Dovecot

In diesem Zusammenhang werden Sie häufig auf eine weitere Abkürzung stoßen: E-Mail-Programme (Clients), wie der Benutzer sie sieht und verwendet, heißen in der Nomenklatur der E-Mail-Welt MUAs (*Mail User Agents*). Diese Programme holen E-Mails beim E-Mail-Server ab (Protokoll POP) bzw. helfen beim Lesen und bei der Verwaltung der externen E-Mails (IMAP). Zum Versenden kommuniziert der MUA direkt mit dem MTA (Protokoll SMTP). Populäre Vertreter dieser Gattung sind Thunderbird, Evolution, KMail, Microsoft Outlook, Apple Mail sowie das textbasierte Programm

mutt. Anstelle eines lokalen Mail-Clients werden immer häufiger Webanwendungen eingesetzt, z. B. Google Mail.

Damit Sie den Überblick über die vielen Abkürzungen nicht verlieren, fasst [Tabelle 34.1](#) die wichtigsten Abkürzungen aus dem E-Mail-Umfeld zusammen. Einige Abkürzungen sind im Text noch nicht vorgekommen, tauchen aber auf den nächsten Seiten auf.

| Abkürzung | Bedeutung                                |
|-----------|--|
| IMAP      | Internet Message Access Protocol         |
| MDA       | Mail Delivery Agent                      |
| MTA       | Mail Transfer Agent                      |
| MUA       | Mail User Agent                          |
| POP       | Post Office Protocol                     |
| SASL      | Simple Authentication and Security Layer |
| SMTP      | Simple Mail Transfer Protocol            |

**Tabelle 34.1** Wichtige E-Mail-Abkürzungen

### E-Mail einst und jetzt

In der Anfangszeit des Internets war *jeder* Rechner direkt mit dem Internet verbunden und hatte – bei Bedarf – seinen eigenen E-Mail-Server. POP oder IMAP waren überflüssig, weil die Anwender die E-Mails direkt auf ihren Rechner = Server serviert bekamen. Die zu diesem Zeitpunkt üblichen textbasierten Mail-Clients (`elm`, `mail`, `pine`) konnten die lokalen Postfächer direkt auslesen – eine Fähigkeit, die den meisten modernen Mail-Clients mit grafischer Benutzeroberfläche abhandengekommen ist. Herkömmliche Mail-Clients kommunizierten mit dem MTA auch nicht via SMTP, sondern übergaben die zu sendende E-Mail ganz einfach an das Kommando `sendmail`.

Microsoft kocht bei E-Mails mit dem Exchange-Server sein eigenes Süppchen. Der Exchange-Server verwendet standardmäßig eigene Protokolle zur Kommunikation mit dem E-Mail-Client, weswegen die meisten Benutzer wohl oder übel Outlook, eine Weboberfläche oder eine Exchange-Server-kompatible App verwenden. POP, IMAP und SMTP werden nur bei spezieller Konfiguration bzw. mit Zusatz-Software unterstützt. Der einzige Linux-Mail-Client, der gut mit dem Exchange-Server kooperiert, ist Evolution.

Exchange-Server

## Protokolle und Ports

Die drei Abkürzungen POP, SMTP und IMAP bezeichnen verschiedene Protokolle zur Übertragung von E-Mails:

- ▶ **SMTP:** Zum Versenden von E-Mails wird das *Simple Mail Transfer Protocol* (SMTP) verwendet. Lokale Programme können mit einem auf dem gleichen Rechner laufenden SMTP-Server normalerweise ohne Authentifizierung kommunizieren. Wenn dagegen ein externer Mail-Client eine E-Mail versendet, übergibt er die Nachricht an den SMTP-Server und muss sich dabei authentifizieren.
- ▶ **POP:** Zur Übertragung von E-Mails vom Mail-Server auf einen lokalen Rechner (Client) kam in der Vergangenheit das *Post Office Protocol* (POP) zum Einsatz. Dieses Protokoll ist perfekt, wenn Anwender nur *ein* E-Mail-Programm verwenden und die heruntergeladenen E-Mails dann auf ihren eigenen Rechnern speichern. POP ist hingegen ungeeignet, wenn parallel mehrere Mail-Clients verwendet werden.
- ▶ **IMAP:** Eine Alternative zu POP ist das *Internet Message Access Protocol* (IMAP). Der Hauptunterschied zu POP besteht darin, dass bei IMAP die E-Mails üblicherweise auf dem IMAP-Server bleiben und dort in mehreren Verzeichnissen (»Postfächern«) organisiert werden. Der E-Mail-Client dient in diesem Fall also nur zur Kommunikation mit dem Server. IMAP ist dann optimal, wenn Sie Ihre E-Mails von unterschiedlichen Rechnern, Smartphones, Tablets etc. aus bearbeiten möchten.

**Ports** Tabelle 34.2 fasst zusammen, welche Ports für die verschiedenen Mail-Protokolle vorgesehen sind. Besonders erwähnenswert ist Port 587: Über diesen Port sollen Mail-Clients mit dem SMTP-Server kommunizieren, während die Kommunikation zweier Mail-Server miteinander in der Regel über Port 25 erfolgt. Port 25 war ursprünglich für die unverschlüsselte Kommunikation konzipiert, wird mittlerweile aber häufig auch mit Verschlüsselung verwendet.

**Mailbox-Formate** Der Mail-Server speichert eintreffende E-Mails in lokalen Dateien oder Verzeichnissen. Oft kommt dabei das mbox-Format zur Anwendung: Alle E-Mails eines Kontos werden einfach zu einer langen Textdatei verbunden. Als Dateiname ist `/var/spool/mail/accountname` üblich. Zur Trennung zwischen den E-Mails dienen Zeilen, die mit `From` beginnen. Das Format ist im Internet dokumentiert, z. B. unter:

<http://www.qmail.org/qmail-manual-html/man5/mbox.html>

Eine Alternative zum mbox-Format ist das maildir-Format. Dabei wird jede einzelne E-Mail in einer eigenen Datei gespeichert, oft in einem Verzeichnis innerhalb des Heimatverzeichnisses des Benutzers.

| Port | Verwendung  |
|------|---|
| 25   | SMTP unverschlüsselt/verschlüsselt, Kommunikation zwischen zwei Servern |
| 110  | POP unverschlüsselt   |
| 143  | IMAP unverschlüsselt  |
| 465  | SMTPS verschlüsselt (veraltet)  |
| 587  | SMTP verschlüsselt, speziell gedacht für Mail-Clients                   |
| 993  | IMAP verschlüsselt  |
| 995  | POP verschlüsselt   |

**Tabelle 34.2** Ports für die E-Mail-Kommunikation

Eine Mailbox besteht aus allen Dateien innerhalb eines Verzeichnisses. Der offensichtliche Vorteil besteht darin, dass einzelne Nachrichten einfacher gelöscht werden können. Das maildir-Format ist insbesondere dann zu empfehlen, wenn der Mail-Server mit einem IMAP-Server kombiniert ist.

Traditionell verwenden Linux-Rechner häufig E-Mails als lokales Kommunikationsmedium. Bei vielen Distributionen ist dazu standardmäßig Postfix installiert. E-Mails mit Fehler- oder Statusmeldungen werden an `root@localhost` gesendet und landen dann im Postfach von `root`, häufig also in der Datei `/var/spool/mail/root`. Die Gefahr ist groß, dass Sie derartige E-Mails nie zu sehen bekommen.

Lokale E-Mails  
lesen

Die einfachste Möglichkeit, derartige Nachrichten zu lesen, bietet das Programm `mutt` (siehe [Abschnitt 14.5](#)): Wenn Sie dieses Programm installieren und in einem Terminalfenster als `root` ausführen, können Sie die lokalen Mails lesen.

Eleganter ist es, alle an `root` adressierten E-Mails mit `/etc/aliases` in die Inbox des Benutzers umzuleiten, der normalerweise für die Administration des Rechners verantwortlich ist:

```
# am Ende von /etc/aliases
...
root:  kofler
```

Die geänderte Einstellung wird erst wirksam, wenn Sie das Kommando `newaliases` ausführen. Sie brauchen nun aber weiterhin einen Mail-Client, der die lokale Mailbox auswertet. Die meisten grafischen Mail-Clients wie Thunderbird, KMail oder Evolution können entsprechend konfiguriert werden.

## Der Nachrichtenfluss im Detail

Anhand von [Abbildung 34.1](#) können Sie verfolgen, wie eine E-Mail von Herrn Huber von Firma-Abc an schmiedt@ziel.de gesendet wird bzw. wie eine E-Mail von irgendwer@absender.de zurück zu huber@firma-abc.de kommt.

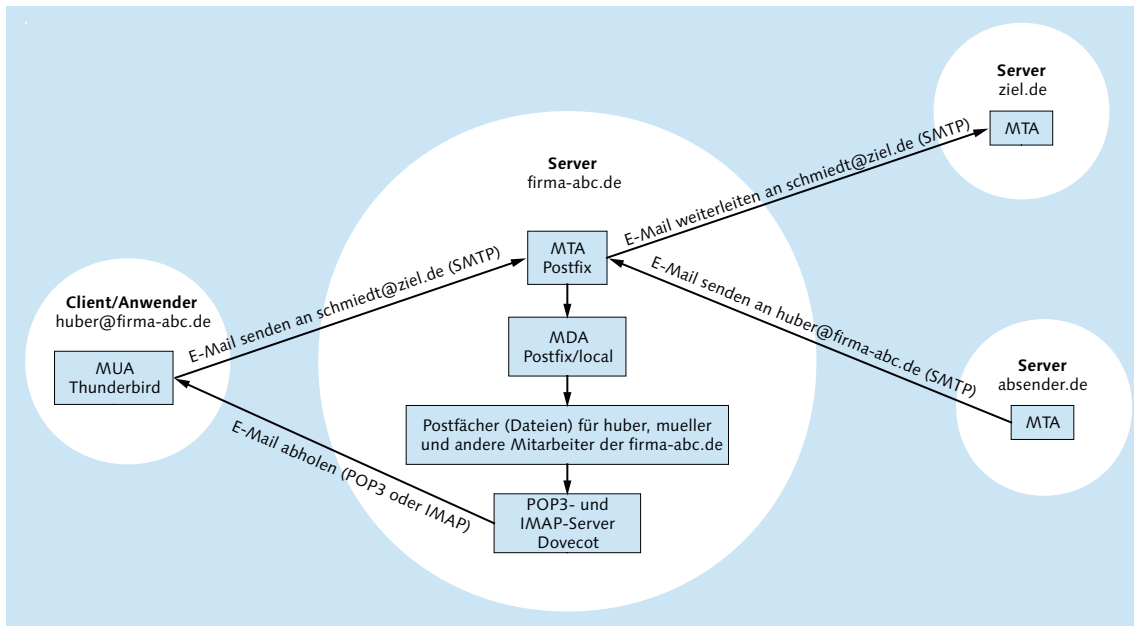


Abbildung 34.1 E-Mail-Kommunikationsfluss

Beginnen wir mit dem ersten Fall: Ein Mitarbeiter der Firma Abc verfasst in seinem E-Mail-Client Thunderbird eine E-Mail an schmiedt@ziel.de und sendet diese ab. Thunderbird nimmt nun über das Protokoll SMTP Kontakt mit dem Mail-Server der eigenen Firma auf, also mit dem Programm Postfix, das auf dem Server firma-abc.de läuft. Postfix übernimmt die E-Mail, stellt fest, an welche Domain sie adressiert ist, und nimmt seinerseits Kontakt mit dem Mail-Server von ziel.de auf. Der MTA von ziel.de vergewissert sich, dass der angegebene Benutzer schmiedt tatsächlich einen E-Mail-Account auf ziel.de hat, und nimmt die E-Mail entgegen.

Währenddessen hat irgendwer@absender.de eine E-Mail für Herrn Huber verfasst und diese versendet. Der MTA von absender.de tritt nun mit dem MTA von firma-abc.de in Verbindung und leitet die E-Mail weiter. Auf dem Server firma-abc.de stellt Postfix fest, dass es einen E-Mail-Account für huber gibt. Postfix akzeptiert die E-Mail und übergibt sie an das zu Postfix gehörende Kommando local, das die E-Mail im Postfach von Herrn Huber speichert. Das »Postfach« ist eine Datei oder ein Verzeichnis, das sich auf dem Server firma-abc.de befindet.

Dort bleibt die E-Mail liegen, bis der E-Mail-Client von Herrn Huber über das Protokoll POP oder IMAP mit dem Programm Dovecot auf dem Rechner `firma-abc.de` in Kontakt tritt. Die meisten E-Mail-Programme sind so eingestellt, dass sie das alle paar Minuten automatisch tun. Wenn Herr Huber gerade offline ist, kann es aber natürlich Stunden oder Tage dauern, bis es so weit ist. Nun muss die E-Mail nur noch in das E-Mail-Programm übertragen werden und kann dann dort gelesen werden.

### Varianten und Optionen

Um Sie nicht zu sehr mit Details zu erschlagen, habe ich die obige Beschreibung der E-Mail-Kommunikation ein wenig verkürzt: Beispielsweise bin ich nicht auf den Umgang mit vorübergehend unzustellbaren E-Mails eingegangen. Ein E-Mail-Server unternimmt über mehrere Stunden Zustellversuche, bevor er den Versand aufgibt. Wenn die Zustellung definitiv scheitert, bekommt der Absender die E-Mail mit einer kurzen Beschreibung der Fehlerursache zurück.

Auch das Thema Authentifizierung ist außen vor geblieben: Nicht jeder darf einfach mit einem beliebigen MTA in Kontakt treten und E-Mails versenden. Um einen derart unkontrollierten E-Mail-Versand zu verhindern, dürfen E-Mails in der Basiskonfiguration nur lokal losgesandt werden. Damit auch ein externer E-Mail-Client E-Mails versenden darf, bedarf es einer Authentifizierung. Das in diesem Kapitel präsentierte Programm Postfix unterstützt zwar das Protokoll SASL (*Simple Authentication and Security Layer*), kann die Authentifizierung aber nicht selbst durchführen. In der Beispielkonfiguration dieses Kapitels übernimmt das Programm Dovecot diese Aufgabe.

Authentifizierung

Damit der Inhalt der E-Mails nicht im Klartext übertragen wird, muss die Verbindung zwischen Client und Mail-Server bzw. die Verbindung zwischen zwei Mail-Servern (MTAs) verschlüsselt werden. Die meisten Mail-Server unterstützen dazu das Protokoll *Transport Layer Security* (TLS, ehemals SSL). Beim Verbindungsaufbau wird die Verschlüsselung zumeist mit STARTTLS eingeleitet: Die Kommunikation beginnt unverschlüsselt; Client und Server vereinbaren dann das bestmögliche Verschlüsselungsverfahren und setzen die Kommunikation verschlüsselt fort. Das funktioniert mit vielen Mail-Clients und -Servern ausgezeichnet.

Verschlüsselung  
und STARTTLS

Die hier skizzierte Verschlüsselung betrifft nur die *Übertragung* von Mails von einem Server zum anderen. Sie hat nichts damit zu tun, ob der *Inhalt* der Nachricht selbst verschlüsselt oder zumindest signiert ist. Die Verschlüsselung bzw. Signierung des Inhalts beim Sender bzw. die korrekte Anzeige der Nachricht beim Empfänger hat nichts mit der Mail-Server-Konfiguration zu tun, sondern ist eine Aufgabe des Mail-Clients. Trotz der unbestrittenen Vorteile signierter bzw. verschlüsselter E-Mails werden Sie in der Praxis nur selten auf derartige E-Mails stoßen. Bequemlichkeit, die

relativ komplexe Schlüsselverwaltung und zwei zueinander inkompatible Standards (PGP und S/MIME) stehen einer weiten Verbreitung im Wege.

**Relaying** Ein weiterer Punkt ist die Nachrichtenübertragung von einem MTA zum nächsten: Nicht immer ist der Weg so direkt wie in Abbildung 34.1. Bisweilen erfolgt der Versand über mehrere Stationen. Die dazwischenliegenden MTAs geben die Nachricht nur weiter (Relaying). Das ist vor allem dann zweckmäßig, wenn es für eine Mail-Domäne einen Haupt- und einen oder mehrere Backup-Server gibt. Wenn der Haupt-Server gerade nicht erreichbar ist, nehmen die Backup-Server die E-Mails entgegen und leiten sie später an den Haupt-Server weiter. Diese Art der Konfiguration mindert das Risiko, dass das E-Mail-System während Wartungsarbeiten nicht erreichbar ist.

#### Sichern Sie Ihren Mail-Server ab, sonst landet er auf einer Blacklist!

Das Schlimmste, was bei der Konfiguration eines E-Mail-Servers passieren kann, ist die mangelnde oder fehlerhafte Absicherung des Relaying: Dann kann jeder ohne Authentifizierung Nachrichten zur Weiterleitung an Ihren E-Mail-Server übergeben.

Spam-Versender durchsuchen das Internet beständig nach solchen Servern und missbrauchen sie für die allgegenwärtige Werbeflut. Das zieht unangenehme Konsequenzen nach sich: Innerhalb weniger Tage landet Ihr Server auf Blacklists, die gefährliche bzw. falsch konfigurierte E-Mail-Server auflisten. Viele E-Mail-Server akzeptieren zur Vermeidung von Spam keine E-Mails von derartigen Servern. Es ist wesentlich schwieriger, aus solchen Blacklists wieder gelöscht zu werden, als auf ihnen zu landen.

Seien Sie bei der Konfiguration also vorsichtig! Wenn Sie den Verdacht haben, dass Ihr Server (genau genommen: dessen IP-Adresse) auf einer Blacklist gelandet ist, können Sie das sehr einfach auf der folgenden Seite verifizieren:

<https://mxtoolbox.com/blacklists.aspx>

**Spam- und  
Virenschutz,  
Weboberfläche**

Schließlich lässt sich das in Abbildung 34.1 dargestellte Szenario noch durch einen serverseitigen Spam- und Virenschutz erweitern. Wenn Sie möchten, dass Ihre Anwender E-Mails auch ohne Mail-Client direkt auf einer Webseite lesen können, brauchen Sie außerdem eine Weboberfläche, z. B. das Programm RoundCube, Horde oder Squirrel-Mail.

## DNS-Konfiguration

Um nochmals auf Abbildung 34.1 zurückzukommen: Woher kennt der MTA auf dem Rechner `firma-abc.de` die IP-Adresse des Mail-Servers für `ziel.de`? Dank DNS natürlich, werden Sie antworten. Grundsätzlich ist das richtig, allerdings sind für den E-Mail-Verkehr nicht gewöhnliche DNS-Einträge (sogenannte A-Records) zuständig, sondern spezielle MX-Einträge. Ein MX-Eintrag gibt den Hostnamen und nicht die



IP-Adresse des Rechners an, der für die E-Mail einer Domain zuständig ist. Das ermöglicht es, die E-Mail-Dienste auf einem anderen Rechner zu realisieren als die restlichen Internetdienste wie Web, SSH, FTP etc. Wenn Ihr Mail-Server auch IPv6 unterstützt, benötigen Sie einen AAAA-Eintrag mit der IPv6-Adresse des Mail-Hosts. Am MX-Eintrag ändert sich nichts.

Außerdem enthält jeder MX-Eintrag eine Prioritätsnummer. Wenn mehrere E-Mail-Server mit unterschiedlicher Priorität eingerichtet werden, erhält normalerweise der Server mit der höchsten Priorität alle E-Mails. Ist dieser Server vorübergehend nicht erreichbar, kommen die niedriger priorisierten Server zum Zuge. Diese Server dienen normalerweise nur als Backup-System und leiten die E-Mails an den Haupt-Server weiter (Relaying), sobald dieser wieder online ist.

Tabelle 34.3 fasst eine typische DNS-Konfiguration für einen einfachen Server zusammen. Alle Internetdienste inklusive Mail laufen auf demselben Rechner, es gibt keinen Backup-E-Mail-Server. Da beim MX-Eintrag ein Domainname (keine IP-Adresse) angegeben werden muss, muss der dort angegebene Domainname – üblicherweise `mail.domain` – ebenfalls durch einen A-Eintrag definiert werden. Der TXT-Eintrag enthält Informationen zum Spam-Schutzsystem SPF, das ich auf den nächsten Seiten erläutere. Neben den in Tabelle 34.3 angegebenen Einträgen hat jede Domain diverse weitere DNS-Einträge, die unter anderem auf die zugrunde liegenden Nameserver verweisen.

| Typ  | Name              | Wert                      | Priorität |
|------|-------------------|---------------------------|-----------|
| A    | firma-abc.de      | 213.214.215.216           |           |
| A    | www.firma-abc.de  | 213.214.215.216           |           |
| A    | mail.firma-abc.de | 213.214.215.216           |           |
| A    | smtp.firma-abc.de | 213.214.215.216           |           |
| A    | imap.firma-abc.de | 213.214.215.216           |           |
| AAAA | firma-abc.de      | 2001:1234:789a:0471::1234 |           |
| AAAA | www.firma-abc.de  | 2001:1234:789a:0471::1234 |           |
| AAAA | mail.firma-abc.de | 2001:1234:789a:0471::1234 |           |
| AAAA | smtp.firma-abc.de | 2001:1234:789a:0471::1234 |           |
| AAAA | imap.firma-abc.de | 2001:1234:789a:0471::1234 |           |
| MX   | –                 | mail.firma-abc.de         | 10        |
| TXT  | firma-abc.de      | "v=spf1 ..."              |           |

**Tabelle 34.3** DNS-Konfiguration eines einfachen Web- und Mail-Servers mit IPv4 und IPv6

Häufig werden bei der DNS-Konfiguration zusätzlich A- und AAAA-Einträge für smtp.firma-abc.de und imap.firma-abc.de eingerichtet, jeweils mit derselben IP-Adresse wie bei mail.firma-abc.de. Diese Hostnamen sind für den Mail-Server-Betrieb nicht erforderlich, erleichtern aber oft die Client-Konfiguration: Manche Mail-Clients gehen standardmäßig davon aus, dass es mit smtp, imap und eventuell auch pop oder pop3 beginnende Hostnamen für die entsprechenden Protokolle gibt.

DNS-Einträge sind öffentlich. Mit dem Kommando `host -a` können Sie problemlos alle Einträge für einen bestimmten Hostnamen auslesen. Die folgende Ausgabe ist gekürzt und sortiert:

```
user$ host -a kofler.info
...
kofler.info. 7200    IN      A       138.201.20.187
kofler.info. 7200    IN      AAAA    2a01:4f8:171:2baf::4
kofler.info. 7200    IN      MX      10 mail.kofler.info.
kofler.info. 7200    IN      TXT     "v=spf1 a mx ~all"
```

#### DNS-Konfiguration

Bleibt zuletzt noch die Frage, wie bzw. wo Sie die DNS-Konfiguration durchführen. In der Regel verwenden Sie dazu eine Weboberfläche, die Ihnen der Service-Provider zur Verfügung stellt, bei dem Sie Ihren Domainnamen registriert haben (siehe z. B. [Abbildung 34.2](#)).

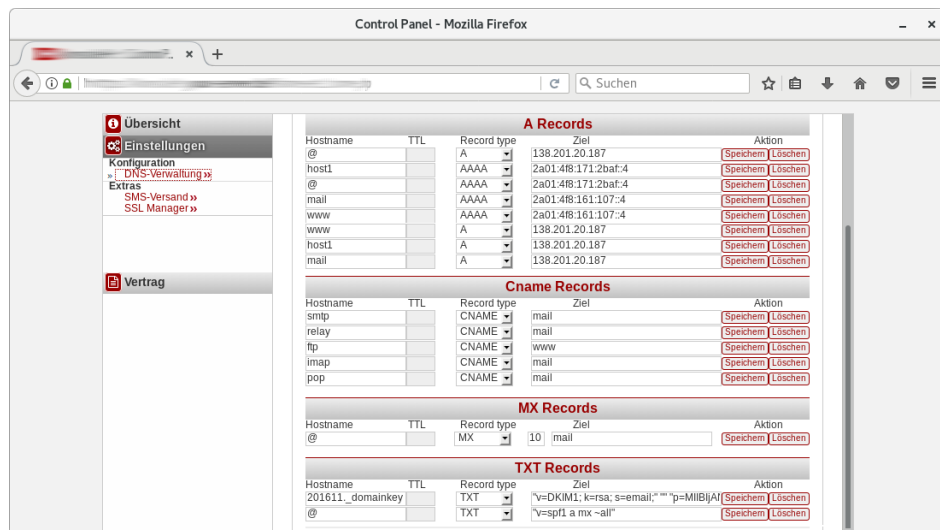


Abbildung 34.2 DNS-Konfiguration bei einem Domainnamen-Registral

**host-Kommando** Wenn Sie die DNS-Mail-Konfiguration überprüfen möchten, ist das Kommando `host` hilfreich, das je nach Distribution im Paket `bind9-host` versteckt ist. Die folgenden Zeilen zeigen, wie Sie zuerst den oder die Hostnamen der Mail-Server ermitteln und dann deren IP-Adresse abfragen:

```

user$ host -t MX firma-abc.de
firma-abc.de mail is handled by 10 mail.firma-abc.de
user$ host mail.firma-abc.de
mail.firma-abc.de has address 213.214.215.216

```

Führen Sie `host` nicht direkt auf dem Root-Server, sondern auf einem externen Rechner aus! Andernfalls können ein eigener Nameserver sowie der Nameserver Ihres Providers das Ergebnis beeinflussen. Das erschwert die Suche nach eventuell vorhandenen eigenen Konfigurationsfehlern.

### Reverse-DNS-Eintrag

Normalerweise liefern Domain Name Server (DNS) die IP-Adresse, die zu einem Hostnamen gehört. Wenn ein fremder Rechner mit `firma-abc.de` in Kontakt treten möchte – sei es mit einem Webbrowser, via SSH oder per E-Mail –, kontaktiert er zuerst den nächsten DNS. Dieser liefert die IP-Nummer der Servers von `firma-abc.de`.

*Reverse DNS* funktioniert gerade umgekehrt: Die IP-Nummer ist bekannt, dafür wird nun der Hostname gesucht. Damit das funktioniert, muss ein Reverse-DNS-Eintrag vorhanden sein. Wenn Sie einen Root-Server gemietet haben, muss Ihr Provider diesen Reverse-DNS-Eintrag durchführen. Viele Provider stellen ihren Kunden ein entsprechendes Konfigurationswerkzeug zur Verfügung. Der Reverse-DNS-Eintrag hat nichts mit den übrigen DNS-Einträgen zu tun! Die Einstellung erfolgt deswegen nicht auf der DNS-Konfigurationsseite Ihres Domainnamens, sondern im Rahmen der Root-Server-Konfiguration.

Auch Reverse DNS testen Sie am einfachsten mit dem Kommando `host`. Beachten Sie, dass zwar mehrere Hostnamen zur selben IP-Adresse führen können (etwa, wenn dank *Virtual Hosting* mehrere Websites auf einem Server laufen), dass die umgekehrte IP-Auflösung aber immer nur einen eindeutigen Hostnamen liefert:

```

user$ host firma-abc.de
firma-abc.de has address 213.214.215.216
firma-abc.de mail is handled by 10 mail.firma-abc.de.
user$ host 213.214.215.216
216.215.214.213.in-addr.arpa domain name pointer firma-abc.de.

```

An sich sind Reverse-DNS-Einträge für den Internetverkehr nicht zwingend erforderlich. Es gibt keinen Internetstandard, der derartige Einträge vorschreibt. Bei E-Mail-Servern haben sich Reverse-DNS-Einträge dennoch durchgesetzt: Viele MTAs akzeptieren nämlich den Empfang von E-Mails von externen MTAs nur, wenn für den Hostnamen des Servers, auf dem der MTA läuft, ein Reverse-DNS-Eintrag existiert und dieser vermutlich nicht dynamisch generiert ist. Diese Schutzmaßnahme richtet sich gegen durch Schadsoftware infizierte (Windows-)PCs, die – oft ohne das Wissen des Eigentümers – zum Spam-Versand missbraucht werden.

Auch wenn der Reverse-DNS-Test keinen zuverlässigen Schutz gegen Spam bietet, wird er oft eingesetzt. Und das bedeutet: Damit andere MTAs Mails von Ihrem Server nicht als Spam klassifizieren, braucht der Server einen Reverse-DNS-Eintrag und gegebenenfalls für die IPv6-Adresse einen zweiten.

**SPF und DKIM** Grundsätzlich ist die DNS-Konfiguration jetzt ausreichend, um einen Mail-Server einzurichten und auszuprobieren. Wenn alles funktioniert, sollten Sie darüber hinaus eine SPF- und DKIM-Konfiguration durchführen. SPF steht für *Sender Policy Framework*, DKIM für *DomainKeys Identified Mail*. Beide Verfahren helfen bei der Überprüfung, ob eine Mail vom richtigen Server versandt wurde. Der praktische Nutzen der beiden Verfahren ist nicht unumstritten. Die Implementierung kann sich dennoch lohnen, weil damit die Chancen steigen, dass von Ihrem Server versendete Mails vom Empfänger nicht als Spam klassifiziert werden. Details zu diesem Thema folgen in Abschnitt 34.9, »SPF, DKIM und DMARC«.

## 34.2 Postfix (MTA)

Dieser Abschnitt beschreibt die Installation und Konfiguration des MTAs Postfix unter CentOS und Ubuntu. Postfix zählt momentan zu den populärsten MTAs und ist sehr gut dokumentiert – sowohl auf <http://www.postfix.org> als auch in unzähligen Artikeln und einigen Büchern.

Dennoch ist die Wahl von Postfix keineswegs selbstverständlich: Während es für manche Aufgaben nur ein weitverbreitetes Programm gibt, stellt die Open-Source-Welt gleich mehrere ausgezeichnete MTAs zur Auswahl. Ob Postfix oder Exim, Qmail oder Sendmail – alle genannten Programme sind weitverbreitet und erfüllen ihren Zweck. Wenn Sie mit Administratoren sprechen, wird vermutlich jeder den MTA empfehlen, den er selbst einsetzt und gut kennt. Einen zehn Jahre alten, aber dafür neutralen und gut fundierten Vergleich verschiedener MTAs finden Sie hier:

[http://shearer.org/MTA\\_Comparison](http://shearer.org/MTA_Comparison)

**Voraussetzungen** Bevor Sie mir auf den nächsten Seiten durch diverse Konfigurationsdetails folgen, sollten einige Voraussetzungen erfüllt sein:

- ▶ Sie brauchen einen von Ihrem Server unabhängigen E-Mail-Account (beispielsweise bei GMX, Google oder mailbox.org), um Ihr neues E-Mail-System zu testen.
- ▶ Die DNS-Konfiguration Ihrer Domain muss korrekt sein.
- ▶ Der Hostname Ihres Rechners sollte korrekt eingestellt sein. Die Kommandos `hostname` und `cat /etc/hostname` sollten jeweils den richtigen Namen Ihres Servers liefern (also z. B. `firma-abc.de` oder `hostxy.firma-abc.de`).
- ▶ Für Ihren Root-Server sollte es einen Reverse-DNS-Eintrag geben.

- ▶ Zu guter Letzt ist die Installation eines kleinen textbasierten E-Mail-Clients empfehlenswert, um den MTA direkt zu testen (siehe [Abschnitt 14.5](#), »Mutt«).

### Installation unter Debian und Ubuntu

Bei Debian und Ubuntu erscheint nach der Installation des `postfix`-Pakets ein Konfigurationsprogramm. Dort müssen Sie angeben, welche Art von Grundinstallation Sie wünschen. Auf einem Root-Server ist `INTERNET SITE` die richtige Wahl: Sie wollen Postfix einsetzen, um auf dem Server E-Mails per SMTP zu versenden und zu empfangen.

Im nächsten Punkt müssen Sie den Namen des E-Mail-Servers angeben (standardmäßig einfach den Hostnamen des Rechners, also beispielsweise `firma-abc.de`). Dieser Name wird dazu verwendet, um E-Mail-Adressen ohne Domainnamen zu vervollständigen. Aus einer E-Mail an `name` wird also eine an `name@firma-abc.de`.

Nach dieser Minimalkonfiguration wird Postfix sofort gestartet. Das Programm erfüllt in der Grundkonfiguration die folgenden Funktionen:

Grund-  
konfiguration

- ▶ Postfix empfängt via SMTP E-Mails an `name@firma-abc.de`. Sofern es auf dem Server den Login `name` gibt, wird die E-Mail akzeptiert und gespeichert. Das gilt für alle Accounts, die in `/etc/passwd` definiert sind: Sofern Apache installiert ist, ist beispielsweise auch `www-data@firma-abc.de` eine gültige E-Mail-Adresse. Wenn `name` nicht bekannt ist, wird die E-Mail zurückgewiesen (*user unknown*).
- ▶ Akzeptierte E-Mails werden vom Postfix-eigenen MDA im Mbox-Format in der Datei `/var/mail/name` gespeichert. Die Dateien in `/var/mail` sind also die »Postfächer« (Mailboxes) der verschiedenen E-Mail-Benutzer des Rechners. Das Mbox-Format bedeutet vereinfacht gesagt, dass die E-Mails in einer immer größer werdenden Datei aneinandergefügt werden. Da E-Mails in der Regel in einem Textformat codiert sind, können Sie die Mbox-Datei zur Not sogar mit `cat` oder `less` ansehen. Eine Alternative zum Mbox-Format ist das Maildir-Format, in dem es für jedes Postfach ein eigenes Verzeichnis und für jede E-Mail eine eigene Datei gibt.
- ▶ Lokale Benutzer können E-Mails versenden – sowohl intern an alle Accounts auf dem Server als auch extern an beliebige andere E-Mail-Adressen.

Als ersten Test senden Sie von einem externen Account eine E-Mail an `name@firma-abc.de`, wobei `name` ein aktiver Linux-Account auf dem Root-Server ist. Die E-Mail sollte nach kurzer Zeit in `/var/mail/name` auftauchen. Wenn Sie sich als `name` anmelden, können Sie die E-Mail mit `mutt` lesen. Ebenfalls mit `mutt` testen Sie als Nächstes das Versenden einer E-Mail an Ihre externe E-Mail-Adresse. Sollten bei den beiden Tests Probleme auftreten, ist die wahrscheinlichste Fehlerursache eine falsche bzw. fehlende DNS-Konfiguration.

Test

## Installation unter CentOS

Unter CentOS und RHEL ist Postfix in der Regel standardmäßig installiert und aktiv. Sie können sich davon mit `systemctl status postfix` überzeugen:

```
root# systemctl status postfix
Loaded: loaded (/usr/lib/systemd/system/postfix.service;
        enabled; vendor preset: disabled)
Active: active (running) since ...
```

### Grund- konfiguration

Anders als bei Debian und Ubuntu haben Sie nicht die Wahl zwischen vordefinierten Konfigurationsvarianten. In der Standardkonfiguration kann Postfix nur E-Mails von lokalen Benutzern nach außen versenden und E-Mails von lokalen Benutzern auch empfangen. Der Mail-Server kann aber keine E-Mails von außen entgegennehmen.

Alle weiteren Funktionen müssen Sie durch entsprechende Einstellungen in `/etc/postfix/main.cf` explizit freischalten. Dabei können Sie sich an dem auf der übernächsten Seite abgedruckten Listing orientieren (Überschrift »main.cf«). Insbesondere muss `main.cf` die Einstellung `inet_interfaces=all` enthalten. Außerdem müssen Sie für `myorigin` und `myhostname` passende Namen angeben. Beachten Sie, dass die Veränderung von `inet_interfaces` erst nach einem Neustart von Postfix wirksam wird!

In der Defaultkonfiguration ist `main.cf` aufgrund zahlloser Kommentare recht unübersichtlich. Abhilfe: Erstellen Sie ein Backup der Datei und dann eine aufgeräumte Version für die weitere Bearbeitung:

```
root# cd /etc/postfix
root# cp main.cf main.cf.orig
root# grep -Ev '^#' main.cf.orig > main.cf
```

### Firewall

Damit der Mail-Server aus dem Internet erreichbar ist, dürfen die Ports 25 und 587 nicht durch eine Firewall blockiert werden. Genau das ist unter CentOS, aber auch unter Fedora, RHEL und SUSE der Fall. Für die Defaultkonfiguration von Postfix spielt das keine Rolle, aber wenn Ihr Mail-Server von außen zugänglich sein soll, müssen Sie einige Ports der Firewall öffnen.

Unter CentOS stellen Sie zuerst fest, wie der Name Ihrer Netzwerkschnittstelle zum Internet lautet (hier `eth0`) und welche Firewall-Zone dieser Netzwerkschnittstelle zugeordnet ist (hier `public`). Anschließend öffnen Sie die Ports mit `firewall-cmd`. Das Kommando für den Port 465 ist optional. Port 465 ist für das mittlerweile als veraltet geltende Protokoll SMTPS vorgesehen.

```
root# firewall-cmd --get-zone-of-interface=eth0  (aktive Zone herausfinden)
public
root# firewall-cmd --permanent --zone=public --add-service=smtp
root# firewall-cmd --permanent --zone=public --add-port=465/tcp  (optional)
root# firewall-cmd --permanent --zone=public --add-port=587/tcp
root# firewall-cmd --reload
```

Hintergrundwissen und andere Strategien zur Konfiguration einer Firewall sind in [Kapitel 37](#), »Firewalls«, zusammengefasst.

## Konfiguration

Die grundlegenden Postfix-Konfigurationsdateien befinden sich in `/etc/postfix`. Innerhalb der Konfigurationsdateien können Sie bereits eingestellte Optionen wie Variablen verwenden – also `option1 = wert1` und dann `option2 = $option1`. Anweisungen in der Konfigurationsdatei dürfen über mehrere Zeilen reichen, wobei der Text ab der zweiten Zeile eingerückt sein muss.

In den Konfigurationsdateien wird oft auf Tabellen oder Listen verwiesen, die in der englischen Dokumentation *Lookup Tables* oder *Mappings* heißen. Dabei gilt die Syntax `option=type:name`. Der gebräuchlichste Dateityp ist `hash`. In diesem Fall wertet Postfix die Datei `name.db` aus, d. h., es fügt dem angegebenen Dateinamen die Endung `.db` hinzu. `*.db`-Dateien sind Tabellen in einem binären Format (*Berkeley Database*, kurz BDB). Zur Manipulation solcher Dateien verwenden Sie das Kommando `postmap`.

Lookup Tables

Tabellen im Textformat sind aus Effizienzgründen nicht vorgesehen. Eine Ausnahme ist lediglich die Textdatei `/etc/aliases`, deren Format kompatibel zu Sendmail ist. Aber auch in diesem Fall greift Postfix auf die dazugehörige BDB-Datei `aliases.db` zurück. Deswegen muss `aliases.db` nach jeder Änderung an `aliases` durch das Kommando `newaliases` synchronisiert werden. Was Mail-Aliase sind und wie sie konfiguriert werden, erfahren Sie in [Abschnitt 34.4](#), »Postfix-Konten«.

Postfix kann aber auch mit externen Datenbanken (MySQL, PostgreSQL, LDAP) kommunizieren, sofern die entsprechenden Postfix-Erweiterungspakete installiert sind. Die in der Konfigurationsdatei genannte Datei enthält nun nicht die eigentlichen Daten, sondern die Verbindungsinformationen und eine (SQL-)Abfrage. Der Einsatz externer Datenbanken bietet sich vor allem dann an, wenn Sie sehr viele, also Hunderte oder Tausende von E-Mail-Accounts verwalten müssen.

Externe Datenbanken

```
virtual_mailbox_domains=mysql:/etc/postfix/mysql-virt-domains.cf
```

## main.cf

Die wichtigste Konfigurationsdatei für Postfix ist `/etc/postfix/main.cf`. Das folgende Listing gibt die wichtigsten Zeilen dieser Datei in der Grundeinstellung (Typ `INTERNET SITE`) wieder:

```
# Datei /etc/postfix/main.cf (auszugsweise)
# so meldet sich Postfix bei anderen MTAs
smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
```

```

# keine automatische Adressvervollständigung durch .firma-abc.de
append_dot_mydomain = no

# Hostname
myhostname = firma-abc.de

# Domain für lokale E-Mails ohne explizite Domain-Angabe
myorigin = firma-abc.de

# Debian/Ubuntu:
# myorigin = /etc/mailname
# /etc/mailname enthält in der Beispielkonfiguration firma-abc.de

# Ort der Alias-Datei
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases

# Versand neuer E-Mails nur vom lokalen Rechner zulassen
mydestination = firma-abc.de, localhost
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128

# keine E-Mail-Weitergabe an andere Hosts (kein Relaying)
relayhost =

# E-Mail-Empfang über alle Netzwerkschnittstellen
inet_interfaces = all

# keine Beschränkung der E-Mail- und Postfach-Größe
mailbox_size_limit = 0

```

Neben den im obigen Listing enthaltenen Schlüsselwörtern gibt es unzählige weitere, die in man 5 `postconf` dokumentiert sind. Für alle nicht explizit eingestellten Optionen gelten Defaulteinstellungen. Wie diese aussehen, verrät das Kommando `postconf -d`. Im Folgenden stelle ich Ihnen einige besonders wichtige Einstellungen kurz vor. Eine Beschreibung diverser weiterer Optionen finden Sie im weiteren Verlauf dieses Kapitels.

- ▶ **myhostname** sollte den Hostnamen des Servers enthalten. `myhostname` gilt als Standardeinstellung für viele andere Optionen. In der hier vorgestellten Beispielkonfiguration wird als Hostname einfach der Domainname verwendet (also `firma-abc.de`). Größere Firmen werden aber sicher mehrere Server betreiben, die unterschiedliche Hostnamen haben – beispielsweise `host1.firma-abc.de`, `host2.firma-abc.de`. `hostname` muss den tatsächlichen Hostnamen enthalten!
- ▶ **myorigin** gibt an, welcher Domain lokal versandte E-Mails zugeordnet werden sollen – im Beispiel dieses Kapitels also `firma-abc.de`. Bei der mit Ubuntu bzw. Debian mitgelieferten Konfigurationsdatei wird `myorigin` aus der Datei `/etc/mailname` gele-



sen. Stellen Sie sicher, dass dort der richtige Name enthalten ist, oder ändern Sie die Einstellung in der Datei `main.cf`!

- ▶ `mydestination` listet Domänen auf, für die empfangene E-Mails lokal in ein Postfach zugestellt (also gespeichert) werden sollen. Nach der Basiskonfiguration hat diese Zeile bei mir auch `localhost.de` enthalten; diesen Eintrag habe ich wieder entfernt.

Vorsicht: Auch wenn Postfix für mehrere Domänen zuständig ist, darf `mydestination` nur Einträge für die Hauptdomäne enthalten. Virtuelle Domänen geben Sie mit der Option `virtual_alias_domains` an.

- ▶ `mynetworks` gibt an, von welchen Adressen Postfix E-Mails ohne Authentifizierung via SMTP entgegennimmt. Die hier angegebenen Adressen bzw. Adressbereiche bezeichnen also die Rechner, denen Postfix »vertraut« (*Trusted SMTP Clients*).

Bei der hier präsentierten Konfiguration (also für einen eigenständigen E-Mail-Server auf einem Root-Server) darf `mynetworks` nur `localhost` (oben in IP4- und IP6-Schreibweise) enthalten. Wenn Sie `mynetworks` falsch (zu liberal) konfigurieren, können fremde Benutzer Ihren Mail-Server dazu verwenden, ohne Authentifizierung E-Mails zu versenden. Vorsicht, Spam-Versender lieben solche Rechner!

- ▶ `relayhost` gibt an, an welchen MTA E-Mails weitergeleitet werden sollen, die *nicht* für die lokale Zustellung gedacht sind. Bei der hier vorgestellten Konfiguration muss `relayhost` leer bleiben. Wenn Postfix dagegen auf einem Rechner im LAN läuft und zu versendende E-Mails an einen externen MTA im Internet weitergeben soll, ist `relayhost` der entscheidende Parameter.
- ▶ `inet_interfaces` steuert, an welchen Netzwerkschnittstellen Postfix auf einkommende E-Mails wartet. Mit der Defaulteinstellung `all` sind alle Schnittstellen aktiv. `loopback_only` oder die Angabe der Adressen `127.0.0.1` und `::1` bedeutet, dass nur server-interner Netzwerkverkehr verarbeitet wird. Änderungen an dieser Option erfordern einen Neustart von Postfix (`system postfix restart`)!

## Änderungen an der Konfiguration


Postfix besteht aus einer Menge von Einzelprogrammen, von denen viele bei Bedarf jedes Mal neu gestartet und wenig später gleich wieder beendet werden. Diese Programme lesen die für sie relevanten Konfigurationsdateien jedes Mal neu ein.

Allerdings gibt es auch Postfix-Komponenten, die Konfigurationsänderungen nicht selbstständig bemerken. Postfix-Einsteiger, denen oft unklar ist, welche Änderungen Postfix selbstständig bemerkt, sollten nach Konfigurationsänderungen grundsätzlich `systemctl reload postfix` ausführen. Das gilt insbesondere für Änderungen an `master.cf` und `main.cf`.

Nach einer Neueinstellung von `inet_interfaces` ist sogar `systemctl restart postfix` erforderlich. Postfix-Prozesse werden `reload` und `restart` wegen des damit verbundenen Geschwindigkeitsverlusts möglichst vermeiden, besonders bei großen aktiven Systemen.

**postconf** Anstatt `main.cf` mit einem Editor zu ändern und anschließend ein `reload`-Kommando auszuführen, können Sie die Änderung auch mit `postconf -e option=wert` durchführen. Das Kommando `postconf` benachrichtigt dann auch gleich Postfix über die Änderung. Das Kommando verändert vorhandene Einstellungen in `main.cf` an ihrem bisherigen Ort, neue Parameter werden am Ende der Datei hinzugefügt. Achten Sie darauf, dass Sie vor und nach dem `=`-Zeichen kein Leerzeichen angeben:

```
root# postconf disable_vrfy_command=yes
```

Das Kommando `postconf` zeigt ohne weitere Optionen alle aktuell gültigen Einstellungen von Postfix an. `postconf -d` liefert die Defaulteinstellungen Ihrer Postfix-Version. Wenn Sie nur an einem bestimmten Parameter interessiert sind, geben Sie diesen explizit an. Den Parameternamen können Sie sogar mit  vervollständigen:

```
root# postconf -d local_recipient_maps
local_recipient_maps = proxy:unix:passwd.byname $alias_maps
```

Besonders praktisch ist `postconf -n`: Dieses Kommando zeigt alle Einstellungen, die *nicht* den Defaultwerten entsprechen. Die zusätzliche Option `-f (fold)` verbessert die Lesbarkeit der Ausgabe durch Zeilenumbrüche.

**soft\_bounce** Sobald ein Mail-Server einmal in Betrieb gegangen ist, sind Änderungen im laufenden Betrieb natürlich heikel. Wenn Sie etwas falsch machen, kann es passieren, dass eintreffende E-Mails abgewiesen werden. Diese sind dann endgültig verloren.

Deswegen ist es besser, vor Beginn der Umbauten `soft_bounce=yes` in `main.cf` einzubauen (und natürlich ständig die Logging-Dateien scharf zu beobachten). Mit `soft_bounce=yes` wandelt Postfix alle 5xx-Fehler (*Permanent Failures*) in 4xx-Fehler (*Persistent Transient Failures*) um. Der Sender wird daher nach einiger Zeit einen neuerlichen Zustellversuch unternehmen. Die E-Mail ist nicht verloren.

Vergessen Sie nicht, nach Fertigstellung und Test der geänderten Konfiguration wieder `soft_bounce=no` einzustellen!

## IPv6 (de)aktivieren

Postfix unterstützt zwar IPv6, bei älteren Installationen/Versionen ist standardmäßig aber nur IPv4 aktiv. Wenn Sie IPv6 nutzen möchten, müssen Sie sicherstellen, dass die beiden folgenden Einstellungen in `main.cf` enthalten sind:

```
# in /etc/postfix/main.cf
...
inet_protocols = all
smtp_address_preference = any
```

Wenn Sie IPv6 explizit deaktivieren wollen, verwenden Sie `inet_protocols = ipv4`. Diese Änderungen werden erst nach einem Neustart von Postfix wirksam. Weitere Informationen zu den IPv6-Funktionen von Postfix können Sie hier nachlesen:

[http://www.postfix.org/IPV6\\_README.html](http://www.postfix.org/IPV6_README.html)

### Port 587 öffnen

Standardmäßig nimmt Postfix Nachrichten nur am Port 25 zur weiteren Bearbeitung entgegen. Dieser Port ist für die Kommunikation zwischen Mail-Servern gedacht. Die meisten Mail-Clients können zwar ebenfalls so konfiguriert werden, dass sie Port 25 verwenden, aber eigentlich ist für den Versand derartiger E-Mails der Port 587 vorgesehen.

Postfix enthält in der Datei `master.cf` (nicht wie sonst `main.cf`) bereits die entsprechenden Anweisungen – diese müssen nur auskommentiert werden. Suchen Sie in `master.cf` nach der Zeile, die mit `submission inet` beginnt, und entfernen Sie dann bei dieser Zeile und den darauffolgenden Zeilen, die mit `-o` beginnen und diverse Parameter enthalten, das Kommentarzeichen `#`. Nur die drei Zeilen, in denen auf `$mua`-Variablen verwiesen wird, belassen Sie unverändert (es sei denn, Sie haben in `main.cf` entsprechende `mua_`-Einstellungen vorgenommen).

```
# Port 587 in /etc/postfix/master.cf aktivieren
...
submission inet n      -      n      -      -      smtpd
  -o syslog_name=postfix/submission
  -o smtpd_tls_security_level=encrypt
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_reject_unlisted_recipient=no
# -o smtpd_client_restrictions=$mua_client_restrictions
# -o smtpd_helo_restrictions=$mua_helo_restrictions
# -o smtpd_sender_restrictions=$mua_sender_restrictions
  -o smtpd_recipient_restrictions=permit_sasl_authenticated,reject
  -o milter_macro_daemon_name=ORIGINATING
```

Kurz gefasst bewirken die obigen Einstellungen, dass Postfix am Port 587 Nachrichten entgegennimmt, allerdings nur, wenn sich der Client authentifizieren kann und die Kommunikation verschlüsselt erfolgt. Damit die veränderte Konfiguration wirksam wird, führen Sie `systemctl reload postfix` aus.

## Logging und Administration

- Logging** Postfix protokolliert seine Tätigkeiten via Syslog. Wo die zahlreichen Meldungen landen, ist wie so oft distributionsabhängig:
- ▶ Bei manchen Distributionen gibt es mehrere Logging-Dateien in `/var/log`, die z. B. die Namen `mail.log`, `mail.info`, `mail.warn` und `mail.err` haben. Das erleichtert es, zwischen mehr oder weniger wichtigen Nachrichten zu differenzieren.
  - ▶ Gebräuchlicher ist es, dass einfach alle mail-relevanten Nachrichten in `/var/log/mail.log` oder `/var/log/maillog` landen. Mit `grep` können Sie dann die Nachrichten herausfiltern, die ein bestimmtes Programm betreffen (`grep postfix mail.log`), die einen Fehler beschreiben (`grep -i error mail.log`) etc.
  - ▶ Immer mehr Distributionen verwenden das `systemd`-Journal für Logging-Aufgaben. `journalctl -u postfix` liefert dann alle Postfix-Meldungen, `journalctl -u dovecot` die von Dovecot etc.
- mailq** Wenn Sie wissen möchten, welche E-Mails auf den Versand warten, führen Sie `postqueue -p` oder `mailq` aus. Die beiden Kommandos liefern eine Liste aller E-Mails, die – aus welchen Gründen auch immer – bisher nicht versandt werden konnten.

### 34.3 Postfix-Verschlüsselung (TLS/STARTTLS)

Postfix unterstützt das Protokoll *Transport Layer Security* (TLS) und das Verfahren STARTTLS zum Aufbau einer verschlüsselten Verbindung zwischen sich und dem Mail-Client, der eine E-Mail versenden möchte. Das setzt aber voraus, dass `main.cf` entsprechend konfiguriert ist und dass Sie für Ihren Rechner ein geeignetes Zertifikat haben. Grundlagenwissen zu »richtigen« und zu selbst erzeugten Zertifikaten können Sie in [Abschnitt 32.4](#), »Verschlüsselte Verbindungen (HTTPS)«, nachlesen. Weitere Informationen zur TLS-Unterstützung von Postfix finden Sie hier:

[http://www.postfix.org/TLS\\_README.html](http://www.postfix.org/TLS_README.html)

Beachten Sie, dass Änderungen an der TLS-Konfiguration erst nach einem Neustart von Postfix wirksam werden!

#### smtpd = Incoming, smtp = Outgoing

Bei der TLS-Konfiguration in `main.cf` gibt es zwei Gruppen von Parametern, die Sie nicht durcheinanderbringen dürfen: Bei der einen Gruppe beginnen die Parameternamen mit `smtpd_`; diese Parameter steuern das Verhalten von Postfix beim Empfang von E-Mails. Das betrifft insbesondere auch die Kommunikation zwischen Ihrem Mail-Client (bzw. den Mail-Clients der Mitarbeiter Ihrer Firma/Organisation) mit dem Mail-Server.

Bei der anderen Gruppe beginnen die Parameternamen mit `smtp_`; diese Parameter betreffen das Versenden von E-Mails, also die Kommunikation des lokalen MTAs mit anderen MTAs.

Eigene Zertifikate sind nur für eintreffende E-Mails (`smtpd_*`) erforderlich! Der Mail-Client oder ein externer MTA fragt Postfix nach dem öffentlichen Teil des Schlüssels und verschlüsselt die Nachricht. Postfix hat den privaten Schlüssel und kann die ankommende Nachricht damit wieder entschlüsseln.

Beim Versenden ist es gerade umgekehrt: Da fragt Postfix den Ziel-MTA, ob dieser einen öffentlichen Schlüssel zum Verschlüsseln der Nachricht zur Verfügung stellen kann. Postfix ist also auf den öffentlichen Teil eines externen Schlüssels angewiesen.

### Beispielkonfiguration und Schlüsselwörter

Die folgenden Zeilen zeigen eine Beispielkonfiguration:

```
# in/etc/postfix/main.cf
...
# Mails nach Möglichkeit verschlüsselt empfangen;
# Zertifikat, Schlüssel und, falls erforderlich,
# Zertifikatskette zur Certification Authority
smtpd_tls_security_level = may
smtpd_tls_cert_file      = /etc/ssl/my-cert.pem
smtpd_tls_key_file       = /etc/ssl/my-key.key
smtpd_tls_CAfile        = /etc/ssl/ca-bundle.pem

# Mails nach Möglichkeit verschlüsselt senden
smtp_tls_security_level = may
# Debian/Ubuntu
smtp_tls_CAfile         = /etc/ssl/certs/ca-certificates.crt
# CentOS/RHEL
smtp_tls_CAfile         = /etc/ssl/certs/ca-bundle.crt

# optional: höhere Effizienz durch Cache
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

# optional: Logging zur Fehlersuche
smtpd_tls_loglevel = 1
smtp_tls_loglevel  = 1
```

Die eingesetzten Schlüsselwörter bedürfen einer etwas ausführlicheren Erklärung:

- ▶ `smtpd_tls_security_level = may` bewirkt, dass Postfix STARTTLS anbietet und eintreffende E-Mails nach Möglichkeit TLS-verschlüsselt entgegennimmt. Bei einer falschen Konfiguration des Mail-Clients bzw. für alte Mail-Clients ist allerdings weiterhin eine Authentifizierung im Klartext möglich. Wenn Sie die Verschlüsselung erzwingen möchten, geben Sie anstelle von `may` die Zeichenkette `encrypt` an. Das garantiert die Verschlüsselung, macht den E-Mail-Empfang von schlecht konfigurierten Clients oder MTAs aber unmöglich.
- ▶ `smtpd_tls_cert_file` und `smtpd_tls_key_file` geben an, welches Zertifikat mit welchem Schlüssel Postfix verwenden soll. Bei Zertifikaten von offiziellen Zertifizierungsstellen erhalten Sie in der Regel eine weitere Datei, die die Zertifikatskette schließt. Diese geben Sie mit `smtpd_tls_CAfile` an. Bei selbst erstellten Zertifikaten bzw. bei Zertifikaten von Let's Encrypt können Sie darauf verzichten.
- ▶ `smtp_tls_security_level = may` bedeutet, dass Postfix E-Mails nach Möglichkeit verschlüsselt versendet, sofern der Empfänger (also der Mail-Server auf der anderen Seite) dazu in der Lage ist. Zur Not wird die Nachricht aber auch im Klartext übertragen. Sicherer wäre `smtp_tls_security_level = encrypt` – aber dann kann Ihr Mail-Server keine Nachrichten mehr an Mail-Server senden, die keine Verschlüsselung unterstützen. Solche Server sollten mittlerweile die Ausnahme sein.
- ▶ `smtp_tls_CAfile` verweist auf eine Sammlung von CAs, denen Postfix vertrauen soll. Unter Debian und Ubuntu wird eine derartige Datei durch das Kommando `update-ca-certificates` automatisch aggregiert und dann in `/etc/ssl/certs/ca-certificates.crt` gespeichert. Die zugrunde liegenden Einzeldateien befinden sich ebenfalls im Verzeichnis `/etc/ssl/certs`.

Postfix läuft allerdings in einer `chroot`-Umgebung und hat gar keinen Zugriff auf `/etc/ssl/certs`. Deswegen wird `ca-certificates.crt` beim Start von Postfix durch das Init-System in das `chroot`-Verzeichnis `/var/spool/postfix/etc/ssl/certs` kopiert. Der Dateiname in `smtp_tls_CAfile` gilt relativ zum `chroot`-Verzeichnis.

Deutlich einfacher ist die Sache unter CentOS/RHEL: Dort verweisen Sie einfach auf die fertige Bundle-Datei `/etc/ssl/certs/ca-bundle.crt`.

Wenn `smtp_tls_CAfile` nicht eingestellt wird, dann wird beim Senden von Mails die Nachricht *Untrusted TLS connection* protokolliert. Das bedeutet, dass die Nachricht zwar verschlüsselt übertragen wurde, dass Postfix aber das Zertifikat des anderen Mail-Servers nicht verifizieren konnte. Mit der korrekten Einstellung von `smtp_tls_CAfile` und vorausgesetzt, dass das Zertifikat des Empfänger-Servers durch eine CA ausgestellt wurde, die in der CA-Sammlung enthalten ist, ändert sich der Logging-Eintrag zu *Trusted TLS connection*. Hilfe bei der Interpretation von *Untrusted*, *Trusted* etc. gibt die folgende Seite der Postfix-Dokumentation:

[http://www.postfix.org/FORWARD\\_SECRECY\\_README.html#status](http://www.postfix.org/FORWARD_SECRECY_README.html#status)

- ▶ Mit `smtp[d]_tls_session_cache_database` verwaltet Postfix einen Cache, in dem TLS-Session-Informationen zwischengespeichert werden. Die Datenbank wird von mehreren Postfix-Instanzen gemeinsam genutzt. Sie beschleunigt die TLS-Kommunikation erheblich. Die Cache-Dateien werden im Postfix-Data-Directory `/var/lib/postfix` gespeichert.

Unter Debian und Ubuntu verwendet Postfix ein sogenanntes Snakeoil-Zertifikat. Dieses selbst signierte Zertifikat wird bei der Installation von Apache oder Postfix (je nachdem, welches Paket zuerst installiert wurde) für den zu diesem Zeitpunkt gültigen Hostnamen erzeugt und ist 10 Jahre lang gültig (siehe auch den [Unterabschnitt »Snakeoil-Zertifikate«](#) in [Abschnitt 32.4](#)). Die Parameter `smtpd_tls_cert_file` und `-key_file` in `main.cf` geben die Orte der Zertifikats- und Schlüsseldateien an:

Defaultkonfiguration in Debian und Ubuntu

```
# Datei /etc/postfix/main.cf
...
smtpd_tls_cert_file = /etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file  = /etc/ssl/private/ssl-cert-snakeoil.key
```

Der Mail-Client wird bei der Konfiguration natürlich melden, dass das Zertifikat nicht vertrauenswürdig sei. In den meisten Mail-Clients können Sie das Zertifikat dennoch akzeptieren; damit werden die Mails sicher verschlüsselt vom Mail-Programm zu Postfix übertragen, und Sie haben Ihr Ziel erreicht.

Während die Verwendung selbst signierter Zertifikate für öffentliche Webserver ein absolutes No-go ist (außer für den Testbetrieb), ist die Situation bei Mail-Servern eine ganz andere. Höchste Priorität hat aktuell, dass der Datenfluss überhaupt verschlüsselt wird. Deswegen akzeptieren die meisten Mail-Server beim Empfang einer Mail die Verschlüsselung auch dann, wenn der sendende Server ein selbst signiertes Zertifikat verwendet, wenn dieses schon abgelaufen ist etc.

Beachten Sie, dass Debian und Ubuntu standardmäßig zwar E-Mails verschlüsselt entgegennehmen, diese aber ohne Verschlüsselung versenden! Abhilfe schaffen `smtp_tls_security_level = may` sowie die Einstellung von `smtp_tls_CAfile`.

Unter CentOS/RHEL enthält `main.cf` anfänglich keinerlei Verschlüsselungsoptionen, weder für eintreffende noch für ausgehende Mails. Sie müssen die gesamte Konfiguration selbst vornehmen.

Defaultkonfiguration in CentOS/RHEL

## Zertifikate einrichten

Sofern es sich bei Ihrem Zertifikat nicht um ein Wildcard-Zertifikat handelt, das für alle Subdomains `*.firma-abc.de` gilt, stellt sich die Frage: Für welchen Domainnamen soll das von Postfix genutzte Zertifikat eigentlich gelten? Für `firma-abc.de` oder für

Für welche Domain soll das Zertifikat gelten?

mail.firma-abc.de (laut MX-Eintrag) oder vielleicht für smtp.firma-abc.de? Laut

<http://serverfault.com/questions/389413>

ist ausschließlich der Client (z. B. Thunderbird) für die Verifizierung verantwortlich. Der Client wiederum vergleicht den Domainnamen des Zertifikats mit dem Hostnamen, den Sie bei der Client-Konfiguration des Mail-Kontos angeben. Der MX-Eintrag ist hierfür nicht relevant.

Standardmäßig stellt Thunderbird für die SMTP-Konfiguration dem Hostnamen smtp voran. Daraus ergibt sich beispielsweise smtp.firma-abc.de. Dann müsste auch das Zertifikat auf diesen Namen lauten. Wenn Sie das konsequent weiterführen, brauchen Sie für jedes Mail-Protokoll ein eigenes Zertifikat (pop.firma-abc.de, imap.firma-abc.de etc.).

#### Snakeoil-Zertifikate (Debian/Ubuntu)

Unter Debian und Ubuntu wird bei der Postfix-Installation standardmäßig ein sogenanntes Snakeoil-Zertifikat samt Schlüssel eingerichtet, also ein selbst signiertes, für zehn Jahre gültiges Zertifikat. Die Parameter smtpd\_tls\_cert\_file und -key\_file in main.cf geben die Orte der Zertifikats- und Schlüsseldateien an:

```
# Datei /etc/postfix/main.cf
...
smtpd_tls_cert_file = /etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file  = /etc/ssl/private/ssl-cert-snakeoil.key
```

Wenn Sie nach einer Änderung des Hostnamens ein neues Zertifikat benötigen, rufen Sie das folgende Kommando auf:

```
root# make-ssl-cert generate-default-snakeoil --force-overwrite
```

Das Snakeoil-Zertifikat hat allerdings gleich zwei Nachteile: Zum einen ist es selbst signiert, zum anderen gilt es nur für den Hostnamen, aber nicht für mail.hostname.

#### Selbst ein Wildcard-Zertifikat erstellen

Unter CentOS/RHEL können Sie anstelle des Snakeoil-Zertifikats mit dem Kommando openssl rasch ein selbst signiertes Wildcard-Zertifikat erzeugen. Entscheidend ist, dass Sie bei der Ausführung von openssl bei der Eingabe des *Common Names* Ihrem Hostnamen »\*.« voranstellen, also z. B. \*.firma-abc.de. Vergessen Sie chmod 600 für die Key-Datei nicht!

```
root# mkdir /etc/ssl/private    (nur CentOS/RHEL)

root# openssl req -new -x509 -days 3650 -nodes \
    -out /etc/ssl/certs/postfix.pem \
    -keyout /etc/ssl/private/postfix.key
...
Common Name (eg server FQDN or YOUR name) []: *.firma-abc.de
...
root# chmod 600 /etc/ssl/private/postfix.key
```



Anschließend verändern Sie `main.cf` wie folgt:

```
# Datei /etc/postfix/main.cf, selbst erzeugtes Zertifikat verwenden
...
smtpd_tls_cert_file = /etc/ssl/certs/postfix.pem
smtpd_tls_key_file  = /etc/ssl/private/postfix.key
```

Das Projekt Let's Encrypt wurde zwar gegründet, um die Verschlüsselung von Webseiten voranzubringen (siehe [Abschnitt 32.5](#)), es spricht aber nichts dagegen, Let's-Encrypt-Zertifikate auch für den Mail-Server zu verwenden. Let's Encrypt bietet aber keine Option für Wildcard-Zertifikate. Deswegen müssen Sie darauf achten, dass das Zertifikat alle üblicherweise für Mail-Server gängigen Hostnamen umfasst, also `mail.xxx`, `smtp.xxx`, `imap.xxx` und `pop.xxx`. Falls Sie ursprünglich nur ein Zertifikat für `xxx` und `www.xxx` erstellt haben, müssen Sie dieses um weitere Sub-Domänen erweitern und dazu neu erzeugen:

Let's-Encrypt-  
Zertifikate  
verwenden

```
root# certbot certonly --apache -d firma-abc.de -d www.firma-abc.de \
      -d mail.firma-abc.de -d smtp.firma-abc.de -d imap.firma-abc.de
```

Anschließend verändern Sie `main.cf` wie im folgenden Muster-Listing:

```
smtpd_tls_cert_file = /etc/letsencrypt/live/kofler.info/fullchain.pem
smtpd_tls_key_file  = /etc/letsencrypt/live/kofler.info/privkey.pem
```

Achten Sie darauf, dass Sie als `smtpd_tls_cert_file` die Datei `fullchain.pem` einstellen, nicht `cert.pem`. (Auch `cert.pem` funktioniert, aber dann müssen Sie als dritten Parameter `smtpd_tls_CAfile` mit `chain.pem` einstellen.)

Änderungen an der TLS-Konfiguration werden erst mit einem Neustart von Postfix wirksam:

Neustart  
erforderlich

```
root# systemctl restart postfix
```

## Forward Secrecy

*Forward Secrecy* bzw. *Perfect Forward Secrecy* bezeichnet ein kryptografisches Protokoll zum Schlüsselaustausch, das sicherstellt, dass eine früher aufgezeichnete Kommunikation später nicht rekonstruiert werden kann, selbst dann nicht, wenn der damals verwendete Schlüssel mittlerweile kompromittiert oder geknackt ist. Dazu wird jede Kommunikation mit einem nur kurzzeitig gültigen Schlüssel verschlüsselt.

Postfix unterstützt Forward Secrecy schon seit einigen Jahren. Bei aktuellen Versionen ist dazu nicht einmal eine besondere Konfiguration erforderlich – Forward Secrecy wird automatisch verwendet, sofern die Gegenseite mitspielt. In den Logging-Dateien lässt sich das anhand der eingesetzten Verschlüsselungsverfahren (Cipher) erkennen. Eine lange Liste aller Forward-Secrecy-tauglichen Verschlüsselungsverfahren liefert das folgende, nicht ganz unkomplizierte `openssl`-Kommando:

```

root# openssl ciphers -v \
      'aNULL:-aNULL:kEECDH:kEDH:+RC4:!eNULL:!EXPORT:!LOW:@STRENGTH' | \
      awk '{printf "%-32s %s\n", $1, $3}'
AECDH-AES256-SHA                Kx=ECDH
ECDHE-RSA-AES256-GCM-SHA384    Kx=ECDH
ECDHE-ECDSA-AES256-GCM-SHA384  Kx=ECDH
...

```

#### Forward Secrecy in den Logging-Dateien

Bei einer Mail, die ich von `kofler.info` an `ubuntu-buch.info` gesendet habe, wurde auf `ubuntu-buch.info` die folgende Nachricht in `mail.log` protokolliert:

```

... postfix/smtpd[11753]: Anonymous TLS connection established from
      host1.kofler.info[138.201.20.187]: TLSv1.2 with cipher
      AECDH-AES256-SHA (256/256 bits)

```

Es wurde also das Verschlüsselungsverfahren `AECDH-AES256-SHA` verwendet, das erste aus der obigen Liste der Forward-Secrecy-tauglichen Verschlüsselungsverfahren (*Cipher*). Dabei gilt:

`AECDH` = *Anonymous Elliptic Curve Diffie Hellman*

`AES` = *Advanced Encryption Standard*

`SHA` = *Secure Hash Algorithm*

#### Eigenen Schlüssel für Forward Secrecy einrichten

Postfix nutzt Forward Secrecy also ohne weitere Konfiguration. Für den Schlüsselaustausch, der für Forward Secrecy erforderlich ist, werden allerdings standardmäßig beim Kompilieren von Postfix vorgegebene Schlüssel verwendet. Sicherheitstechnisch noch besser ist es, stattdessen eigene Schlüssel zu generieren und `main.cf` um einige Zeilen zu erweitern, damit diese Schlüssel auch verwendet werden. Um eigene Schlüssel zu erzeugen, führen Sie diese Kommandos aus:

```

root# cd /etc/postfix
root# umask 022
root# openssl dhparam -out dh512.tmp 512 && mv dh512.tmp dh512.pem
root# openssl dhparam -out dh1024.tmp 1024 && mv dh1024.tmp dh1024.pem
root# openssl dhparam -out dh2048.tmp 2048 && mv dh2048.tmp dh2048.pem
root# chmod 644 dh512.pem dh1024.pem dh2048.pem

```

Anschließend fügen Sie die folgenden zwei Zeilen in `main.cf` ein:

```

# Anpassung in /etc/postfix/main.cf für noch bessere Forward Secrecy
smtpd_tls_dh1024_param_file = ${config_directory}/dh2048.pem
smtpd_tls_dh512_param_file = ${config_directory}/dh512.pem

```

Weitere Informationen zur Forward Secrecy können Sie hier nachlesen:

[http://www.postfix.org/FORWARD\\_SECRECY\\_README.html#quick-start](http://www.postfix.org/FORWARD_SECRECY_README.html#quick-start)

<https://www.heinlein-support.de/blog/security/perfect-forward-secrecy-pfs-fur-postfix-und-dovecot>

## 34.4 Postfix-Konten

Grundsätzlich gilt für Postfix: Jeder reguläre Benutzer in `/etc/passwd` ist auch ein Mail-User. Für diese Benutzer bedarf es keiner zusätzlichen Konfiguration. Jeder dieser Linux-Benutzer kann E-Mails senden und empfangen.

Externe Mail-Clients wie Thunderbird müssen sich bei Postfix bzw. Dovecot authentifizieren. Für diese Authentifizierung kommen ebenfalls die regulären Linux-Passwörter zum Einsatz. (Das ist ein großer Unterschied zu Samba: Dieser Datei-Server verwaltet für jeden Samba-User getrennte Passwörter.)

### Sicherheitsrisiko Mail-Passwort

So bequem es ist, dass Linux- und Mail-User standardmäßig parallel laufen, so riskant kann dies auch sein. Nun wird vermutlich niemand so unvernünftig sein, `root` als regulären E-Mail-Account zu verwenden (also `root@firma-abc.de`) – das würde nämlich erfordern, bei der Konfiguration des Mail-Clients das `root`-Passwort anzugeben.

Viel eher wird das Risiko des folgenden Szenarios übersehen: Als Administrator des Servers `firma-abc.de` haben Sie dort einen Account mit `sudo`-Rechten – z. B. `huber`. Für Administrationsarbeiten loggen Sie sich mit `ssh huber@firma-abc.de` ein, führen dann `sudo -s` aus und erledigen, was eben gerade zu tun ist. So weit, so gut.

Nach der Konfiguration des Mail-Servers ist es natürlich naheliegend, den Mail-Account `huber@firma-abc.de` zu verwenden. Wie vorhin erläutert, sind dafür keine weiteren Konfigurationsarbeiten notwendig. Das Problem besteht darin, dass Sie nun Ihr `huber`-Passwort einmal bei der Konfiguration von Thunderbird auf Ihrem Linux-Notebook eingeben, noch einmal bei der Konfiguration Ihres Mail-Programms auf dem Smartphone, noch einmal für das Tablet etc. Geht eines dieser Geräte verloren, ist es mit etwas IT-Wissen nicht allzu schwer, das gespeicherte Passwort zu `huber` zu finden. Und dieses Passwort reicht nicht nur aus, um Ihre E-Mails zu lesen bzw. in Ihrem Namen zu versenden – der unehrliche Finder kann sich nun ebenfalls mit `ssh` auf `firma.abc-de` einloggen und hat dort dank `sudo` uneingeschränkte Rechte. Sicherheitstechnisch ist das das Schlimmste, was passieren kann.

Abhilfe: Verwenden Sie nie Mail-Accounts von Benutzern mit `sudo`-Rechten! Verwenden Sie getrennte Accounts für E-Mails und für Administrationsarbeiten, auch wenn dies unbequem ist.

Die oben skizzierte Parallelschaltung von Linux- und Mail-Accounts entspricht der Defaultkonfiguration, aber dabei wird es selten bleiben. In den folgenden Abschnitten erläutere ich Ihnen daher einige Möglichkeiten für eine davon abweichende Konfiguration:

Von den  
Linux-Usern  
abweichende  
Mail-Accounts

- ▶ Mail-Aliase ermöglichen es, E-Mails auf vorhandene Accounts umzuleiten.
- ▶ Mit der Konfiguration einer expliziten Empfängerliste können Sie die Liste der E-Mail-Accounts steuern und z. B. verhindern, dass System-Accounts wie `adm` oder `lp` E-Mails senden oder empfangen können.
- ▶ Wenn Sie virtuelle Domänen nutzen, können Sie mit einem Postfix-Server E-Mails verschiedener Hostnamen verarbeiten (nicht nur für `firma-abc.de`, sondern auch für `firma-xy.info`).
- ▶ Mit virtuellen Postfächern können Sie Mail-Accounts und deren Passwörter vollkommen von den Linux-Accounts trennen. Die Login-Daten werden dann oft in einer Datenbank gespeichert.

Zunächst beschäftigt sich der folgende Abschnitt aber mit der Frage, wo E-Mails eigentlich gespeichert werden.

### mbox- oder Maildir-Format

Standardmäßig speichert Postfix eintreffende E-Mails im mbox-Format in der Datei `/var/mail/name`. Wenn E-Mails stattdessen in einer lokalen Datei im Benutzerverzeichnis gespeichert werden sollen (weiterhin im mbox-Format), geben Sie den gewünschten Dateinamen relativ zum Homeverzeichnis mit `home_mailbox` an:

```
# in /etc/postfix/main.cf
...
# E-Mails im mbox-Format in der Datei /home/name/Mailbox speichern
home_mailbox = Mailbox
```

Wenn Sie vorhaben, die Mails überwiegend via IMAP abzurufen, sollten Sie das Maildir-Format vorziehen. Die korrekte Einstellung von `home_mailbox` sieht nun so aus:

```
# in /etc/postfix/main.cf
...
# E-Mails im maildir-Format im Verzeichnis /home/name/Maildir speichern
home_mailbox = Maildir/
```

Der Verzeichnisname muss mit `/` enden, damit Postfix das Maildir-Format verwendet. Falls `main.cf` eine `mailbox_command`-Anweisung enthält, müssen Sie diese auskommentieren. Neu eintreffende E-Mails werden nun in eigenen Dateien im Verzeichnis `/home/name/Maildir` gespeichert.

Vergessen Sie nicht, auch Ihrem lokalen Mail-Client bzw. Dovecot den Ort und das Format Ihres Postfachs mitzuteilen (siehe [Abschnitt 34.5](#), »Dovecot (POP- und IMAP-Server)«)! Wenn Sie `mutt` einsetzen, müssen Sie das Programm entsprechend konfigurieren, damit es die E-Mails aus dem Maildir-Verzeichnis liest (siehe [Abschnitt 14.5](#), »Mutt«).

## Mail-Aliase

Ein Mail-Alias ist ein zusätzlicher Mail-Name zum Empfang von E-Mail. Die E-Mail wird aber tatsächlich an einen bereits vorhandenen Account weitergeleitet. Aliase werden in der Datei `/etc/aliases` definiert. Diese Datei sieht üblicherweise so ähnlich wie das folgende Muster aus:

```
# Datei /etc/aliases
postmaster:    root
webmaster:     huber
Bernhard.Huber: huber
...
```

Die erste Spalte gibt den Alias-Namen ohne Domäne an. Der Name gilt für die in `myhostname` definierte Domäne, also beispielsweise für `postmaster@firma-abc.de`. Die zweite Spalte enthält den lokalen Empfänger. Im obigen Beispiel werden an `postmaster` adressierte E-Mails an `root` weitergeleitet, E-Mails an `webmaster` und an `Bernhard.Huber` an `huber`. Es ist zulässig, in `/etc/aliases` für jeden Alias mehrere, durch Kommas getrennte Empfänger anzugeben.

Als Empfänger können Sie statt eines lokalen E-Mail-Account-Namens auch eine externe E-Mail-Adresse angeben oder eine Datei, an die die E-Mail angefügt wird, oder ein Programm in der Schreibweise `|kommando`, an das die E-Mail weitergegeben wird. Das Weiterleiten an externe E-Mail-Adressen funktioniert zwar problemlos, scheitert in der Praxis aber oft am Spam-Schutz des Ziel-MTAs. Wenn Sie also beispielsweise E-Mails an `webmaster` an `name@gmx.de` weiterleiten, erkennt der Spam-Filter von GMX, dass die E-Mail nicht direkt an `gmx.de` übertragen wurde, sondern indirekt über `firma-abc.de`. Das reicht für einen misstrauischen Spam-Filter, um die E-Mail als Spam zu klassifizieren.

Damit geänderte Aliase wirksam werden, müssen Sie das Kommando `newaliases` ausführen. Dieses Kommando synchronisiert die Textdatei `/etc/aliases` mit der BDB-Datei `/etc/aliases.db`.

In `/etc/postfix/main.cf` stoßen Sie in der Regel auf zwei `alias`-Optionen, was bisweilen Verwirrung stiftet:

`alias_database`  
und `alias_maps`

```
# in /etc/postfix/main.cf
...
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
```

`alias_database` gibt an, welche Datenbankdatei durch das Kommando `newaliases` aktualisiert werden soll. Die hier angegebene Datei (deren Name durch `.db` ergänzt wird) enthält die in `/etc/aliases` angegebenen Aliase in einem binären Format.

`alias_maps` gibt an, welche Alias-Datenbanken Postfix berücksichtigen soll. Normalerweise geben Sie hier dieselbe Datei an wie bei `alias_database`. Es ist aber zulässig, darüber hinaus weitere Quellen für Alias-Definitionen anzugeben. Der entscheidende Unterschied zwischen den beiden Parametern besteht darin, dass `alias_database` für `newaliases` gilt, `alias_maps` dagegen für Postfix!

#### aliases aufräumen bei CentOS/RHEL

In der Defaultkonfiguration enthält `/etc/aliases` unter CentOS/RHEL eine schier endlose Liste von Aliases, die in der Praxis so selten zweckmäßig sind. Erstellen Sie ein Backup der Datei, und löschen Sie dann alle Zeilen, die Sie aktuell nicht benötigen!

**.forward** Auch als lokaler Benutzer, der keinen Zugriff auf `/etc/aliases` hat, können Sie Ihre E-Mail an eine andere Adresse umleiten: Dazu erzeugen Sie die Datei `.forward` und speichern darin die neue Zieladresse. Fertig!

Wie ich oben bereits erwähnt habe, funktioniert diese simple Form der E-Mail-Umleitung zumeist nur bei lokalen Adressen wunschgemäß. Bei externen Zieladressen kann es dagegen passieren, dass sich die umgeleiteten E-Mails im Spam-Schutz des Ziel-MTAs verfangen.

### Explizite Empfängerliste

Standardmäßig kann jeder Linux-Account E-Mails empfangen. Es ist aber selten wünschenswert, dass System-Accounts wie `daemon`, `sys` oder `man` E-Mails erhalten. Um diesen Missstand zu beheben, müssen Sie am Parameter `local_recipient_maps` drehen. Die Standardeinstellung lautet:

```
local_recipient_maps = proxy:unix:passwd.byname $alias_maps
```

Das bedeutet, dass alle in der Unix-Datei `/etc/passwd` aufgezählten Benutzer sowie alle in den Alias-Datenbanken genannten Benutzer E-Mails empfangen können. Wenn Sie möchten, dass nur `fischer`, `huber`, `schmidt` sowie `root` (für System-Benachrichtigungen) E-Mails empfangen sollen, gehen Sie wie folgt vor: Zuerst erzeugen Sie eine Textdatei, die zeilenweise die Account-Namen enthält. Die Datei muss in einer zweiten Spalte einen beliebigen Wert enthalten, weil Postfix und das Kommando `postmap` generell Schlüssel/Wert-Paare (*Key/Value Pairs*) erwarten – auch bei Listen, bei denen eigentlich nur die Existenz eines Schlüssels relevant ist, der dazugehörige Wert aber gar nicht berücksichtigt wird.

```
# Datei /etc/postfix/local-recips
fischer    x
huber      x
schmidt    x
root       x
```

Aus dieser Datei erstellen Sie nun mit `postmap` eine für Postfix lesbare Datenbankdatei `local-recips.db`:

```
root# cd /etc/postfix
root# postmap local-recips
```

Mit `postmap -s` können Sie überprüfen, ob die Datei korrekt erstellt wurde:

```
root# postmap -s hash:local-recips
huber    x
root     x
schmidt  x
fischer  x
```

Anschließend fügen Sie in `/etc/postfix/main.cf` eine Zeile zur Einstellung von `local_recipient_maps` ein:

```
# in /etc/postfix/main.cf
local_recipient_maps = hash:/etc/postfix/local-recips $alias_maps
```

### Mapping-Dateien ganz sicher aktualisieren

Nach jeder Änderung an `local-recips` müssen Sie `postmap` ausführen, um die für Postfix relevante Datenbankdatei `local-recips.db` zu aktualisieren. Diese Aktualisierung ist allerdings ein kritischer Vorgang: `postmap` löscht dabei `local-recips.db` und schreibt die Datei anschließend neu. Wenn Postfix gerade während dieses Zeitpunkts auf `local-recips.db` zugreift, erhält es falsche bzw. unvollständige Daten.

Eine sichere Vorgehensweise sieht deswegen so aus: Sie geben der zugrunde liegenden Textdatei einen anderen Namen (z. B. `local-recips1`), wenden `postmap` auf diese Datei an und führen dann `mv local-recips1.db local-recips.db` aus. Somit enthält `local-recips.db` zu jedem Zeitpunkt konsistente Daten – entweder in der alten oder in der neuen Version. Diesen Vorgang können Sie durch eine Script- oder `make`-Datei automatisieren, wie es hier beschrieben ist:

[http://www.postfix.org/DATABASE\\_README.html#safe\\_db](http://www.postfix.org/DATABASE_README.html#safe_db)

Dieser Hinweis gilt für *alle* Mapping-Dateien, die in diesem Kapitel vorkommen.

Von nun an akzeptiert Postfix nur noch E-Mails an die in `local-recips` genannten Empfänger. Es gibt allerdings noch eine Einschränkung: Es werden nur dann E-Mails zugestellt (also lokal gespeichert), wenn der Benutzer einen Account auf dem Rechner hat. Ist das nicht der Fall, müssen Sie mit `adduser` einen neuen Account erstellen.

Linux-Accounts  
einrichten

Im folgenden Beispiel bewirkt die Option `--shell /bin/false`, dass dem Account statt einer Shell das Programm `/bin/false` zugeordnet ist. Das macht ein interaktives Arbeiten unmöglich. `--gecos , ,` unterdrückt die Fragen nach dem vollständigen Namen und weiteren überflüssigen Kontaktinformationen. Wichtig ist hingegen das Passwort, und das, obwohl sich der Benutzer gar nicht einloggen kann. Der Grund: Das

Passwort gilt für die in [Abschnitt 34.5](#), »Dovecot (POP- und IMAP-Server)«, beschriebene POP- und SMTP-Authentifizierung.

```
root# adduser --shell /bin/false --gecos ,, , huber
Geben Sie ein neues UNIX-Passwort ein:      *****
Geben Sie das neue UNIX-Passwort erneut ein: *****
```

### Tipp

Sichere Passwörter können Sie komfortabel mit `pwgen`, `makepasswd` (Debian, Ubuntu) oder `mkpasswd` (Fedora, RHEL, Paket `expect`) erzeugen.

Wenn Sie mit virtuellen Domänen arbeiten (siehe den übernächsten Abschnitt), können Sie darauf verzichten, für jeden Benutzer einen eigenen Linux-Account einzurichten. Postfix unterstützt dann sogenannte virtuelle Postfächer, die ganz real sind, aber keinem gültigen Benutzernamen entsprechen.

### Vom Linux-Account abweichende E-Mail-Adressen

In vielen Firmen sind E-Mail-Adressen der Form `Vorname.Nachname@firma.de` üblich. Linux sieht jedoch derart lange Benutzernamen – noch dazu mit einem Punkt – nicht vor. Das hindert Sie aber nicht daran, dennoch lange E-Mail-Namen zu verwenden:

- ▶ Legen Sie einen Linux-Account mit einem Linux-typischen, kurzen Benutzernamen an, z. B. `huber`.
- ▶ Definieren Sie eine Alias-Regel, die E-Mails an `Bernhard.Huber` an `huber` weiterleitet.
- ▶ Verwenden Sie bei der Konfiguration des E-Mail-Clients als Absenderadresse die Langform, also `Bernhard.Huber@firma-abc.de`. Beachten Sie aber, dass Sie zur POP-, IMAP- und SMTP-Authentifizierung den Linux-Account-Namen angeben müssen!
- ▶ Damit auch lokal (z. B. durch `mutt`) versandte E-Mails die richtige Absenderadresse in der Langform enthalten, richten Sie eine neue Tabelle in der Textdatei `/etc/postfix/canonical` ein. Diese Tabelle gibt an, wie Postfix E-Mail-Adressen verändern soll:

```
# /etc/postfix/canonical
huber      Bernhard.Huber@firma-abc.de
...
```

Diese Tabelle wandeln Sie mit `postmap` in eine für Postfix lesbare Tabelle um.

- ▶ Anschließend stellen Sie in `main.cf` den Parameter `canonical_maps` ein und führen dann `systemctl reload postfix` aus:

```
# in /etc/postfix/main.cf
...
canonical_maps = hash:/etc/postfix/canonical
```



Neben den Canonical- und Alias-Tabellen bietet Postfix diverse weitere Möglichkeiten, um E-Mail-Adressen in verschiedenen Phasen des E-Mail-Verkehrs zu manipulieren, also beim Empfang, vor dem Versenden etc. Einen guten Überblick gibt die folgende Seite:

[http://www.postfix.org/ADDRESS\\_REWRITING\\_README.html](http://www.postfix.org/ADDRESS_REWRITING_README.html)

### Virtuelle Domänen mit gemeinsamen E-Mail-Benutzern

Im einfachsten Fall ist Postfix nur für E-Mails an den Hostnamen des Rechners zuständig, z. B. xxx@firma-abc.de. Oft ist es aber wünschenswert, dass *ein* MTA für mehrere E-Mail-Domänen zuständig ist, also xxx@noch-eine-firma.de. Alle Domänen, die nicht mit dem Hostnamen des Rechners übereinstimmen, werden in der Postfix-Nomenklatur »virtuell« genannt (auf Englisch oft auch *Hosted Domains*).

Postfix sieht mehrere Möglichkeiten zur Realisierung virtueller Domänen vor. Der einfachste Weg besteht darin, beim Parameter `mydestination` einfach mehrere Domänen einzustellen, etwa so:

```
# in /etc/postfix/main.cf
...
mydestination = firma-abc.de, localhost, noch-eine-firma.de
```

Selbstverständlich müssen Sie auch die DNS-Konfiguration von `noch-eine-firma.de` entsprechend anpassen. Dem MX-Hostnamen muss also die IP-Adresse Ihres Servers zugeordnet sein (siehe auch [Abschnitt 34.1](#), »Einführung und Grundlagen«).

Sie erreichen damit, dass Mails an `noch-eine-firma.de` genauso behandelt werden wie Mails an `firma-abc.de`. Es spielt also keine Rolle, ob eine E-Mail an `huber@firma-abc.de` oder an `huber@noch-eine-firma.de` adressiert wird: Postfix stellt die E-Mail auf jeden Fall dem lokalen Benutzer `huber` zu. Für manche Fälle ist das ausreichend – insbesondere dann, wenn eine Firma oder Organisation mehrere Webauftritte hat, aber in Wirklichkeit immer dieselben Personen dafür verantwortlich sind.

### Virtuelle Domänen mit getrennten E-Mail-Benutzern

Wenn Sie zwischen gleichnamigen Benutzern je nach Domäne differenzieren möchten, geben Sie die betroffenen Domänen nicht in `mydestination` an, sondern mit dem Schlüsselwort `virtual_alias_domains`.

Außerdem brauchen Sie nun eine Tabelle, die die Zuordnung zwischen den E-Mail-Adressen der virtuellen Domänen und den realen Linux-Accounts herstellt. Weiterhin ist also für jede E-Mail-Adresse ein Linux-Account erforderlich. Um beim Beispiel der `huber`-Adressen zu bleiben: Der Account für `huber@firma-abc.de` ist weiterhin `huber`. Für

huber@noch-eine-firma.de müssen Sie einen neuen Account anlegen, z. B. huberNEF. Der Aufbau der Tabelle für die virtuellen E-Mail-Benutzer sieht so aus:

```
# Textdatei /etc/postfix/virtual
huber@noch-eine-firma.de      huberNEF
mueller@noch-eine-firma.de   muellerNEF
...
```

Die Tabelle kann wie die Alias-Tabelle mehreren E-Mail-Adressen denselben Benutzer zuordnen, also etwa:

```
# in /etc/postfix/virtual
...
webmaster@noch-eine-firma.de  huberNEF
```

Mit `postmap` machen Sie aus dieser Datei eine für Postfix lesbare Datenbankdatei:

```
root# postmap /etc/postfix/virtual
```

Jetzt müssen Sie noch `main.cf` anpassen. `virtual_alias_domains` zählt alle virtuellen Domänen auf (aber nicht die Hauptdomäne, die geben Sie weiterhin mit `mydestination` an!). `virtual_alias_maps` gibt den Dateinamen der virtuellen Alias-Tabelle an.

```
# in /etc/postfix/main.cf
...
mydestination      = firma-abc.de, localhost
virtual_alias_domains = noch-eine-firma.de, firma-xyz.de, ...
virtual_alias_maps = hash:/etc/postfix/virtual
```

### Virtuelle Domänen mit virtuellen Postfächern

Bis jetzt war es immer erforderlich, dass jeder E-Mail-Adresse ein Linux-Account gegenüberstand. Postfix weigert sich, E-Mails in einem Postfach zu speichern, wenn es nicht einen gleichnamigen Linux-Account gibt. Bei sehr vielen E-Mail-Adressen wird es aber zunehmend unpraktisch, jedes Mal auch einen neuen Account anzulegen. Postfix sieht zur Lösung dieses Problems virtuelle Postfächer vor. Das sind ganz gewöhnliche Postfachdateien; die Bezeichnung »virtuell« bezieht sich nur darauf, dass es keine dazugehörenden Linux-Accounts gibt.

Freilich müssen auch die virtuellen Postfächer jemandem gehören. Dazu erzeugen Sie eine neue Gruppe und einen neuen Benutzer mit jeweils noch unbenutzten GIDs und UIDs, im folgenden Beispiel lauten sie jeweils 5000:

```
root# groupadd -g 5000 vmail
root# useradd -g vmail -u 5000 vmail -d /home/vmail -m
```

Nun ändern Sie `main.cf` wie im folgenden Beispiel-Listing. `virtual_mailbox_domains` gibt die virtuellen Domänen an, deren E-Mails in virtuellen Postfächern gespeichert werden sollen. `virtual_mailbox_base` gibt das Verzeichnis an, in dem die virtuellen Postfächer angelegt werden sollen. Die Tabelle `virtual_mailbox_maps` stellt die Zuordnung zwischen den E-Mail-Adressen und den Postfächern her. `virtual_uid_maps` und `virtual_gid_maps` geben die UID und GID der Postfachdateien an. Theoretisch ist es hier möglich, eigene UIDs und GIDs für jedes Postfach anzugeben, aber das ist selten zweckmäßig.

```
# in /etc/postfix/main.cf
...
mydestination          = firma-abc.de, localhost
virtual_mailbox_domains = noch-eine-firma.de, firma-xyz.de, ...
virtual_mailbox_base   = /var/mail
virtual_mailbox_maps    = hash:/etc/postfix/virtual-mboxes
virtual_uid_maps       = static:5000
virtual_gid_maps       = static:5000
```

Die Datei `virtual-mboxes` gibt für jede E-Mail-Adresse die dazugehörige Postfachdatei relativ zum Pfad `virtual_mailbox_base` an. Dieses Beispiel verwendet für jede Domäne ein eigenes Verzeichnis und innerhalb dieses Verzeichnisses dann einfach den Benutzernamen. Grundsätzlich können Sie hier aber nach Belieben verfahren. Für die eigentliche Zustellung ist das Postfix-Kommando `virtual` zuständig. Es speichert die E-Mails standardmäßig im `mbox`-Format. Wenn Sie das `Maildir`-Format vorziehen, geben Sie in `virtual-mboxes` einfach im Anschluss an den Dateinamen einen Schrägstrich an, also beispielsweise `noch-eine-firma.de/huber/`:

```
# Datei /etc/postfix/virtual-mboxes
huber@noch-eine-firma.de    noch-eine-firma.de/huber
mueller@noch-eine-firma.de  noch-eine-firma.de/mueller
webmaster@firma-xyz.de     firma-xyz.de/webmaster
...
```

`postmap` macht aus `virtual-mboxes` eine Datenbankdatei:

```
root# postmap /etc/postfix/virtual-mboxes
```

Ein letzter Schritt besteht nun darin, dass Sie für jede Domäne das Postfachverzeichnis erzeugen müssen. Entscheidend sind dabei die Zugriffsrechte.

```
root# mkdir /var/mail/noch-eine-firma.de
root# chown mail:mail /var/mail/noch-eine-firma.de
root# chmod g+w /var/mail/noch-eine-firma.de
```

systemctl postfix reload aktiviert die Konfiguration. Nun senden Sie eine Testnachricht an `huber@noch-eine-firma.de` und werfen danach einen Blick in das Verzeichnis `/var/mail/noch-eine-firma.de/`. Dort sollte nun die Datei `huber` mit der neuen E-Mail auftauchen.

Wenn Sie *alle* Domänen virtuell verwalten möchten, also auch die Domäne des Hostnamens Ihres Rechners, entfernen Sie den Hostnamen aus der `mydestination`-Zeile und fügen ihn der `virtual_mailbox_domains`-Zeile hinzu:

```
# in /etc/postfix/main.cf
...
mydestination          = localhost
virtual_mailbox_domains = firma-abc.de, noch-eine-firma.de, ...
```

### Virtuelle Postfächer mit MySQL-Tabellen verwalten

Virtuelle Postfächer ersparen Ihnen es zwar, für jede E-Mail-Adresse einen Account einzurichten, machen dafür aber die Konfiguration von Dovecot zur Abholung der E-Mails (POP) sowie zur SMTP-Authentifizierung komplizierter: Sie müssen nun eine weitere Tabelle administrieren, die für jeden Benutzer einen Login-Namen und ein Passwort enthält.

Virtuelle Postfächer reduzieren den Verwaltungsaufwand nur dann, wenn Sie gleichzeitig sämtliche Daten der E-Mail-Accounts in einer Datenbank oder in einem LDAP-System speichern und Postfix und Dovecot gleichermaßen auf diese Datenbank zugreifen können. Eine ausführliche Anleitung, wie Sie dies mit MySQL bewerkstelligen, finden Sie auf den folgenden Seiten:

<https://workaround.org/ispmail/jessie>  
<https://www.linode.com/docs/email/postfix/email-with-postfix-dovecot-and-mariadb-on-centos-7>

### Adressüberprüfung abstellen (VRFY)

Das Protokoll SMTP sieht unter anderem das Kommando `VRFY` vor. Damit kann die Existenz einer Mail-Adresse überprüft werden:

```
user$ telnet localhost 25
Trying 127.0.0.1...
220 firma-abc.de ESMTP Postfix
VRFY huber@firma-abc.de
252 2.0.0 huber@firma-abc.de
VRFY bla@firma-abc.de
550 5.1.1 <huber@firma-abc.de>: Recipient address rejected:
User unknown in local recipient table
```

An sich ist das eine praktische Sache. Leider wird diese Funktion von Spammern gern missbraucht, um gültige E-Mail-Adressen zu ermitteln. Deswegen wird vielfach empfohlen, das `Verify`-Kommando zu deaktivieren. Dazu fügen Sie eine Zeile zu `main.cf` hinzu:

```
# in /etc/postfix/main.cf
disable_vrfy_command=yes
```

## 34.5 Dovecot (POP- und IMAP-Server)

Bis jetzt haben wir nur Postfix installiert. Wenn Sie `mutt` oder einen vergleichbaren E-Mail-Client direkt auf dem Server ausführen, können Sie E-Mails senden und empfangen. Wollen Sie aber mit externen Clients arbeiten, z. B. mit Thunderbird auf einem Linux-Notebook oder mit Mail-Apps auf einem Smartphone, dann müssen diese in der Lage sein, auf die Postfächer des Mail-Servers zuzugreifen.

Postfix bietet dafür keine Funktionen; als Mail-Server beherrscht er nur das Protokoll SMTP. Zum Abholen von E-Mails gibt es ergänzend dazu das Protokoll POP. Wenn Ihre E-Mails auf dem Server bleiben, aber extern verwaltet werden sollen, bietet sich alternativ das Protokoll IMAP an. Für diese beiden Protokolle ist das Programm Dovecot zuständig.

Bei Red-Hat-Distributionen (CentOS, Fedora, RHEL) befindet sich Dovecot mit all seinen Funktionen im Paket `dovecot`. Bei Debian und Ubuntu ist Dovecot hingegen über mehrere Pakete verteilt. Je nachdem, welches Protokoll Sie verwenden möchten, installieren Sie `dovecot-imapd` und/oder `dovecot-pop3d`.

Installation

Die Protokolle POP3 und IMAP verwenden die in [Tabelle 34.4](#) zusammengestellten Ports.

Firewall

| Port     | Protokoll   |
|----------|---|
| Port 110 | POP3 mit STARTTLS-Verschlüsselung bzw. ohne Verschlüsselung |
| Port 995 | POP3S mit Verschlüsselung                                   |
| Port 143 | IMAP mit STARTTLS-Verschlüsselung bzw. ohne Verschlüsselung |
| Port 993 | IMAPS mit Verschlüsselung                                   |

**Tabelle 34.4** Firewall-Ports für POP und IMAP

Bei den Protokollen POP3S und IMAPS ist die Verschlüsselung zwingend vorgesehen. Demgegenüber wurden die Protokolle POP3 und IMAP nachträglich um die STARTTLS-Funktionen erweitert, bei denen die Kommunikation unverschlüsselt beginnt, aber nach Möglichkeit verschlüsselt fortgesetzt wird.

Unter CentOS, Fedora, RHEL und SUSE sind alle POP- und IMAP-Ports standardmäßig blockiert. Unter SUSE öffnen Sie die Ports am einfachsten mit YaST. Unter Fedora sowie CentOS/RHEL 7 gehen Sie hingegen wie folgt vor, wobei Sie natürlich nur die Ports freischalten, die Sie tatsächlich nutzen möchten:

```
root# firewall-cmd --get-zone-of-interface=eth0 (aktive Zone herausfinden)
public
root# firewall-cmd --permanent --zone=public --add-service=pop3s
root# firewall-cmd --permanent --zone=public --add-service=imaps
root# firewall-cmd --permanent --zone=public --add-port=110/tcp (POP3)
root# firewall-cmd --permanent --zone=public --add-port=143/tcp (IMAP)
root# firewall-cmd --reload
```

**Konfiguration** Im einfachsten Fall funktioniert Dovecot ohne Konfigurationsarbeiten – man würde es nicht für möglich halten, dass es so etwas überhaupt noch gibt! In der Praxis werden sich ein Blick in die Konfigurationsdateien und diverse kleine Änderungen aber nicht ganz vermeiden lassen.

Für die Konfiguration von Dovecot 2.n sind eine Menge Dateien vorgesehen: `/etc/dovecot/dovecot.conf` und `/etc/dovecot/conf.d/*.conf`. Die Datei `10-auth.conf` enthält zudem einige `include`-Anweisungen für `auth-*.ext`-Dateien mit Ergänzungen. Die meisten `include`-Anweisungen sind aber standardmäßig auskommentiert und daher nicht aktiv.

```
user$ cd /etc/dovecot/conf.d
user$ ls
10-auth.conf      10-ssl.conf      90-plugin.conf      auth-static.conf.ext
10-director.conf 15-lda.conf      90-quota.conf        auth-system.conf.ext
10-logging.conf  20-imap.conf     auth-deny.conf.ext   auth-vpopmail.conf.ext
10-mail.conf      20-pop3.conf     auth-master.conf.ext
10-master.conf    90-acl.conf      auth-passwdfile.conf.ext
```

Die `*.conf`-Dateien dienen auch zur Dokumentation von Dovecot. Das ist einerseits praktisch, andererseits aber auch sehr unübersichtlich: Standardmäßig umfassen die `*.conf`-Dateien rund 1100 Zeilen Code! Die tatsächlich relevanten Anweisungen befinden sich in ganz wenigen Zeilen, der Rest sind Kommentare.

Mit den folgenden Kommandos können Sie sich einen Überblick verschaffen. Die beiden verschachtelten `grep`-Kommandos eliminieren alle Kommentare und leeren Zeilen:

```
user$ cd /etc/dovecot
user$ grep -h -v '^[[:space:]]*#' dovecot.conf conf.d/*.conf | \
    grep -v '^[[:space:]]*#' | less
```

Die folgenden Zeilen präsentieren die Einstellungen der wichtigsten Konfigurationsdateien. Auf den Abdruck der Dateien, die nur Kommentare, aber keinen aktiven Code enthalten, habe ich aus Platzgründen verzichtet.

```

# Datei dovecot.conf
!include_try /usr/share/dovecot/protocols.d/*.protocol
dict {
}
!include conf.d/*.conf
!include_try local.conf
# Datei conf.d/10-auth.conf
auth_mechanisms = plain
!include auth-system.conf.ext

# Datei conf.d/10-director.conf
service director {
    unix_listener login/director {
    }
    fifo_listener login/proxy-notify {
    }
    unix_listener director-userdb {
    }
    inet_listener {
    }
}
service imap-login {
}
service pop3-login {
}
protocol lmtp {
}

# Datei conf.d/10-master.conf
service imap-login {
    inet_listener imap {
    }
    inet_listener imaps {
    }
}
service pop3-login {
    inet_listener pop3 {
    }
    inet_listener pop3s {
    }
}
service lmtp {
    unix_listener lmtp {
    }
}
service imap {
}

```

```

service pop3 {
}
service auth {
    unix_listener auth-userdb {
    }
}
service auth-worker {
}
service dict {
    unix_listener dict {
    }
}

# Datei conf.d/10-ssl.conf
ssl_cert = </etc/ssl/certs/dovecot.pem
ssl_key = </etc/ssl/private/dovecot.pem

# Datei conf.d/15-lda.conf
protocol lda {
}

# Datei conf.d/20-imap.conf
protocol imap {
}

# Datei conf.d/20-pop3.conf
protocol pop3 {
}

```

Beachten Sie, dass Sie die geschwungenen Klammern in einer eigenen Zeile schließen müssen. Die Anweisung `protocol imap { }` in nur einer Zeile ist syntaktisch nicht erlaubt.

Dovecot braucht deswegen so wenige Einstellungen, weil es für alle Optionen Defaultwerte gibt. `dovecot -a` liefert eine Liste aller Einstellungen, `dovecot -n` eine Liste mit allen Optionen, die von der Defaulteinstellung abweichen.

- IPv6** Dovecot ist zwar IPv6-tauglich, kommuniziert aber standardmäßig nur über das IPv4-Protokoll. Um auch IPv6 zuzulassen, müssen Sie in `dovecot.conf` den Parameter `listen` einstellen. `*` bedeutet, dass alle IPv4-Schnittstellen berücksichtigt werden, `::` gilt analog für IPv6-Schnittstellen. Alternativ können Sie auch explizit die IPv4- und IPv6-Adressen der Schnittstellen angeben, die für den Mail-Verkehr vorgesehen sind.



```
# Datei dovecot.conf, IPv6 aktivieren
...
listen = *, ::
```

Dovecot kommt sowohl mit dem Maildir- als auch mit dem mbox-Format zurecht, ganz egal, ob Sie das Programm als POP- oder IMAP-Server verwenden (oder beides). Wenn Sie Dovecot allerdings überwiegend als IMAP-Server einsetzen, sollten Sie das Maildir-Format aus Effizienzgründen vorziehen.

Ort der  
Mailbox-Dateien

Dovecot versucht die Postfächer automatisch zu entdecken, was bei meinen Tests auch problemlos funktioniert hat. Es durchsucht dabei in dieser Reihenfolge die folgenden Verzeichnisse:

```
/home/username/Maildir           (Maildir-Format)
/home/username/mail und /var/mail/username (mbox-Format)
/home/username/Mail und /var/mail/username (mbox-Format)
```

Die automatische Mailbox-Suche kann allerdings scheitern, wenn ein Benutzer noch keine Mail erhalten hat und sein Postfach somit leer ist, die Postfachdatei also noch gar nicht existiert. Deswegen empfiehlt es sich, den Mailbox-Ort in `10-mail.conf` explizit einzustellen. Für Postfächer in `/var/mail/name` lautet die richtige Einstellung:

```
# in /etc/dovecot/conf.d/10-mail.conf
...
# mbox-Postfächer in /var/mail
mail_location = mbox:~/Mail:INBOX=/var/mail/%u
```

Damit weiß Dovecot, dass Ihr Server das mbox-Format verwendet und dass sich neue E-Mails in `/var/mail/username` befinden. Die zusätzliche Angabe des Verzeichnisses `Mail` ist für diverse Zusatzfunktionen von Dovecot erforderlich – auch dann, wenn sich in diesem Verzeichnis keine E-Mails befinden! Falls Dovecot als IMAP-Server verwendet wird, werden dort alle anderen IMAP-Postfächer gespeichert. Sie können natürlich ein beliebiges anderes Benutzerverzeichnis angeben, `mail` oder `Mail` sind aber die übliche Wahl. Das Benutzerverzeichnis muss vor der `INBOX`, also dem Postfach für neue E-Mails, angegeben werden.

Falls Sie Postfix so konfiguriert haben, dass der MTA die E-Mails im Maildir-Format an das lokale Verzeichnis `Maildir` zustellt, sieht die korrekte Einstellung von `mail_location` so aus:

```
# in /etc/dovecot/conf.d/10-mail.conf
...
# mbox-Postfächer in /var/mail
mail_location = maildir:~/Maildir
```

Wenn Postfix so konfiguriert ist, dass es E-Mail-Adressen aus diversen Domänen in virtuellen Postfächern speichert, wird die Konfiguration komplizierter: Zum einen müssen Sie Dovecot verraten, wo sich die Postfächer befinden, und zum anderen brauchen Sie nun eine eigene Tabelle, die für alle E-Mail-Adressen Login-Namen und Passwörter für die POP- und SMTP-Authentifizierung enthält. Konfigurationstipps und ein konkretes Beispiel finden Sie auf den folgenden Webseiten:

<https://wiki2.dovecot.org/VirtualUsers>

<https://workaround.org/ispmail/jessie>

### Betrieb als POP- bzw. IMAP-Server

Dovecot funktioniert auf Anhieb als POP- oder IMAP-Server. Um Ihre E-Mails von einem externen Rechner mit einem E-Mail-Client herunterzuladen, richten Sie im Client das Mail-Konto ein und geben als Verschlüsselungsverfahren STARTTLS an. Der Benutzername entspricht dem Namen Ihres Linux-Accounts auf dem Server. Auch das Passwort ist dasselbe wie auf dem Server.

#### Verschlüsselung (TLS/STARTTLS)

Dovecot unterstützt das Protokoll *Transport Layer Security* (TLS) und das Verfahren STARTTLS zum Aufbau einer verschlüsselten Verbindung. Standardmäßig verwendet Dovecot das selbst generierte Zertifikat `/etc/ssl/certs/dovecot.pem`, das Sie beim ersten Verbindungsaufbau im Mail-Client akzeptieren müssen. Wenn Sie ein eigenes Zertifikat und einen eigenen Schlüssel verwenden möchten, müssen Sie die Orte der Zertifikats- und Schlüsseldateien in `10-ssl.conf` anpassen.

Im folgenden Beispiel kommen Dateien zum Einsatz, die auch für die HTTPS-Konfiguration von Apache verwendet wurden. Das zusätzliche CA-Zertifikat ist nur erforderlich, wenn Ihr Zertifikat von einer Zertifizierungsstelle stammt, die dem E-Mail-Client nicht bekannt ist. Der E-Mail-Client kann damit überprüfen, ob Ihre Zertifizierungsstelle befugt ist, Zertifikate auszustellen.

```
# in /etc/dovecot/conf.d/10-ssl.conf
...
ssl_cert = </etc/apache2/firma-abc.de.crt
ssl_key = </etc/apache2/firma-abc.de.key
ssl_ca = </etc/apache2/ca.pem
```

Die folgenden Kommandos zeigen, wie Sie bei Bedarf selbst ein neues, für zehn Jahre gültiges Zertifikat für Dovecot erzeugen (siehe auch [Abschnitt 32.4](#), »Verschlüsselte Verbindungen (HTTPS)«):

```
root# openssl req -new -x509 -days 3650 -nodes \
      -out /etc/ssl/certs/dovecot.pem \
      -keyout /etc/ssl/private/dovecot.pem
root# chmod 600 /etc/ssl/private/dovecot.key
```

Die dazu passende Dovecot-Konfiguration sieht wie folgt aus:

```
# in /etc/dovecot/conf.d/10-ssl.conf
...
ssl_cert = </etc/ssl/certs/dovecot.pem
ssl_key = </etc/ssl/private/dovecot.key
```

Natürlich können Sie auch Let's-Encrypt-Zertifikate verwenden:

```
# in /etc/dovecot/conf.d/10-ssl.conf
...
ssl_cert = </etc/letsencrypt/live/firma-abc.de/fullchain.pem
ssl_key = </etc/letsencrypt/live/firma-abc.de/key.pem
```

### SMTP-Authentifizierung für Postfix

Im Postfix-Abschnitt habe ich bereits erwähnt, dass Postfix zwar das Protokoll SASL (*Simple Authentication and Security Layer*) unterstützt, die Authentifizierung aber nicht selbst durchführen kann. Dovecot kann Postfix in dieser Angelegenheit unter die Arme greifen.

#### Wozu SMTP-Authentifizierung?

Möglicherweise ist Ihnen nicht ganz klar, worum es bei der SMTP-Authentifizierung eigentlich geht. SMTP-Server nehmen von lokalen Programmen ohne Weiteres E-Mails zum Versand entgegen. Das gilt auch für Postfix: Wenn Sie `mutt` auf demselben Rechner wie Postfix ausführen, können Sie damit eine E-Mail verfassen und versenden, wobei Postfix sich um den eigentlichen Versand kümmert.

Wenn der Mail-Client aber extern läuft – zu Hause auf Ihrem Notebook, unterwegs auf Ihrem Smartphone –, dann müssen Sie sich beim Mail-Server authentifizieren, bevor Sie eine E-Mail versenden können. Um diese Authentifizierung geht es hier.

Der erforderliche Konfigurationsaufwand ist minimal. Zum Ersten müssen Sie in der Dovecot-Konfigurationsdatei `10-master.conf` den im Abschnitt `service auth` bereits vorgesehenen Code zur Authentifizierung über die Socket-Datei `/var/spool/postfix/private/auth` aktivieren:

```
# Ergänzungen in /etc/dovecot/conf.d/10-master.conf
...
service auth {
    unix_listener auth-userdb {
        mode = 0600
        user = postfix
        group = postfix
    }
}
```

Dovecot-  
Konfiguration

```

# Postfix smtp-auth
unix_listener /var/spool/postfix/private/auth {
    mode = 0666
}
# Auth process is run as this user.
user = $default_internal_user
}

```

Zum Zweiten fügen Sie in 10-auth.conf den Authentifizierungsmechanismus login hinzu. Diese Ergänzung ist erforderlich, damit die Authentifizierung auch mit Outlook Express bzw. Windows Mail funktioniert.

```

# Ergänzungen in /etc/dovecot/conf.d/10-auth.conf
...
auth_mechanisms = plain login

```

#### Postfix-Konfiguration

Zuletzt müssen Sie am Ende der Postfix-Konfigurationsdatei /etc/postfix/main.cf einige Zeilen einfügen. Beachten Sie, dass die Pfadangabe für smtpd\_sasl\_path relativ zum Verzeichnis /var/spool/postfix erfolgen muss. Der Grund: Postfix läuft aus Sicherheitsgründen in einer chroot-Umgebung und interpretiert *alle* Pfadangaben in main.cf relativ zum Postfix-Queue-Verzeichnis.

```

# Ergänzung in /etc/postfix/main.cf
...
smtpd_sasl_auth_enable      = yes
smtpd_sasl_type             = dovecot
smtpd_sasl_path             = private/auth
smtpd_recipient_restrictions = permit_mynetworks,
                             permit_sasl_authenticated,
                             reject_unauth_destination

```

#### Konfiguration aktivieren

Anschließend fordern Sie beide Dienste dazu auf, ihre Konfigurationsdateien neu einzulesen:

```

root# systemctl restart dovecot
root# systemctl reload postfix

```

## 34.6 Client-Konfiguration

»Richtige« Tests mit einem Mail-Client können erst nach Abschluss der Postfix- und Dovecot-Konfiguration beginnen. Bei der Konfiguration sind die folgenden Punkte wichtig:

- ▶ Verwenden Sie nach Möglichkeit POP, nicht IMAP.
- ▶ Aktivieren Sie sowohl bei SMTP (Postausgang) als auch bei IMAP (oft nicht ganz korrekt als »Posteingang« bezeichnet) die Option STARTTLS.

- Falls Sie ein selbst signiertes Zertifikat verwenden, müssen Sie dieses explizit akzeptieren, oft nicht bei der Konfiguration, sondern beim ersten Verbindungsaufbau.

Lassen Sie sich bei der Konfiguration nicht irritieren, wenn dort von Klartextpasswörtern die Rede ist. Sofern Sie die Verschlüsselungsmethode STARTTLS verwenden, sind Klartextpasswörter sicher, weil die Passwortübertragung innerhalb einer TLS-Sitzung erfolgt, also bereits auf einer höheren Ebene verschlüsselt wird.

The screenshot shows the 'Konto einrichten' dialog box in Thunderbird. It contains the following fields and options:

- Ihr Name:** Michael Kofler (with a sub-label: Ihr Name, wie er anderen Personen gezeigt wird)
- E-Mail-Adresse:** kofler@firma-abc.de
- Passwort:** [Redacted with dots]
- Passwort speichern
- Posteingangs-Server:** IMAP (dropdown)
- Server-Adresse:** imap.firma-abc.de
- Port:** 143
- SSL:** STARTTLS (dropdown)
- Authentifizierung:** Passwort, normal (dropdown)
- Postausgangs-Server:** SMTP
- Server-Adresse:** smtp.firma-abc.de
- Port:** 587
- SSL:** STARTTLS (dropdown)
- Authentifizierung:** Passwort, normal (dropdown)
- Benutzername: Posteingangs-Server:** kofler
- Postausgangs-Server:** kofler

Buttons at the bottom include: 'Neue E-Mail-Adresse erhalten...', 'Erweiterte Einstellungen', 'Abbrechen', 'Erneut testen', and 'Fertig'.

Abbildung 34.3 Beispielhafte Konfiguration eines Mail-Kontos in Thunderbird

## 34.7 Spam-Abwehr

Die Abwehr von Spam ist ein Kampf gegen Windmühlen, der wohl nie mehr gewonnen werden kann. In diesem Abschnitt zeige ich Ihnen zwei Möglichkeiten, die Spam-Flut zumindest ein wenig zu mindern. Dazu können Sie zum einen das Programm SpamAssassin verwenden. Es versucht, Spams aufgrund diverser Merkmale zu klassifizieren. Wesentlich simpler ist ein anderer Ansatz, das sogenannte Greylisting: Dabei werden E-Mails von unbekanntem Absendern erstmalig abgelehnt. Trifft die E-Mail ein paar Minuten später nochmals ein, wird sie akzeptiert. Dieses Verfahren funktioniert, weil sich viele Spammer nicht die Mühe machen, SMTP-konform mehrere Zustellversuche durchzuführen.

### SpamAssassin

Das bekannteste Open-Source-Programm zur Spam-Bekämpfung ist SpamAssassin. Es versucht aufgrund diverser Kriterien zu entscheiden, ob eine E-Mail Spam enthält

oder nicht. Gründe, die zu einer Spam-Klassifizierung führen, sind fehlende Reverse-DNS-Einträge, offensichtlich falsche DNS-Angaben oder die Herkunft der E-Mails von bekannten »Spam-Schleudern«, also von Rechnern, die aufgrund von massenhaftem Spam-Versand auf sogenannten Blacklists gelandet sind.

SpamAssassin berücksichtigt auch den *Inhalt* der E-Mail. Letzteres erfordert eine relativ aufwendige Analyse, die auf stark frequentierten E-Mail-Servern eine hohe CPU-Last verursacht. Die Ergebnisse der verschiedenen Spam-Erkennungsregeln werden addiert. Überschreitet die Summe einen Grenzwert, wird die E-Mail als Spam gekennzeichnet.

Machen Sie sich aber keine zu großen Hoffnungen: Auf der einen Seite scheitert SpamAssassin häufig daran, offensichtliche Spams zu erkennen. Dafür passiert es immer wieder, dass ordnungsgemäße E-Mails als Spam klassifiziert werden. Vermutlich ist SpamAssassin immer noch besser als gar kein Spam-Schutz – wenngleich mich in den letzten Jahren selbst daran Zweifel beschlichen.

#### Umgang mit spam-verdächtigen E-Mails

Standardmäßig verpackt SpamAssassin als Spam erkannte E-Mails neu. Die geänderte E-Mail enthält dann einen Hinweis auf den Spam-Verdacht. Die originale Nachricht wird im Anhang mitgeliefert, sodass der Anwender eine irrtümlich als Spam klassifizierte E-Mail problemlos lesen kann.

Jetzt bleibt noch die Frage offen, wie der Endanwender am besten mit spam-verdächtigen E-Mails verfährt: SpamAssassin fügt in jede E-Mail eine Zeile der Form X-Spam-Flag: YES ein. Außerdem bekommt jede von SpamAssassin kontrollierte E-Mail eine Zeile X-Spam-Level: \*\*\*\*\*, wobei die Anzahl der Sterne die Spam-Bewertungssumme wiedergibt: Je mehr Sterne, desto höher ist die Spam-Wahrscheinlichkeit. Standardmäßig betrachtet SpamAssassin E-Mails ab fünf Sternen als Spam.

Aufgrund der X-Spam-Zeilen im E-Mail-Header können Sie bei den meisten E-Mail-Clients eine Filterregel aufstellen, sodass derart gekennzeichnete E-Mails automatisch in einen Junk- oder Spam-Ordner verschoben werden.

#### Installation und Konfiguration

Es gibt verschiedene Möglichkeiten, SpamAssassin mit Postfix zu kombinieren. Dieser Abschnitt beschreibt, wie Sie SpamAssassin über die Milter-Schnittstelle von Postfix aufrufen. *Milter* steht für *Mail Filter* und ist eigentlich eine von Sendmail definierte Schnittstelle, um E-Mail-Filterprogramme einzubinden. Postfix unterstützt diese Schnittstelle in aktuellen Versionen ebenfalls und vereinfacht so die Integration von Spam- und Virenfiltern im Vergleich zu anderen Verfahren ganz erheblich.

Alle gängigen Distributionen stellen Pakete für SpamAssassin zur Verfügung. Unter Debian und Ubuntu führen Sie zur Installation das folgende Kommando aus:

```
root# apt install spamassassin spamass-milter
```

Um SpamAssassin als Dämon zu aktivieren, führen Sie zwei Änderungen in `/etc/default/spamassassin` durch:

```
# Änderungen in /etc/default/spamassassin
...
# den SpamAssassin-Dämon spamd starten
ENABLED=1
...
# regelmäßige Updates der SpamAssassin-Regeln durchführen
CRON=1
```

Die Basiskonfiguration von SpamAssassin erfolgt durch diverse `*.cf`-Dateien im Verzeichnis `/usr/share/spamassassin`. Davon abweichende Einstellungen führen Sie am besten in der Datei `/etc/spamassassin/local.cf` durch. Die wahrscheinlich interessanteste Einstellung ist `required_score` (Defaultwert 5.0): Sie gibt an, ab welcher Punktzahl eine E-Mail als Spam klassifiziert wird. Ebenfalls oft praktisch ist die Einstellung `rewrite_header Subject *****SPAM*****`: Damit wird die Subject-Zeile aller spam-verdächtigen E-Mails verändert. Das erleichtert die Spam-Erkennung für E-Mail-Anwender, die mit der Definition von Filterregeln in ihrem E-Mail-Client überfordert sind.

Nach diesen Vorbereitungsarbeiten starten Sie SpamAssassin:

```
root# systemctl start spamassassin
```

Die Milter-Erweiterung zu SpamAssassin, also das Programm `spamass-milter`, ist bereits aktiv. Es wurde unmittelbar nach der Installation gestartet und kommuniziert direkt mit dem SpamAssassin-Dämon `spamd`. Die Konfiguration erfolgt durch die Datei `/etc/default/spamass-milter`, Änderungen sind in der Regel aber nicht erforderlich. Die Kommunikation zwischen `spamass-milter` und Postfix erfolgt über die Socket-Datei `/var/spool/postfix/spamass/spamass.sock`.

Jetzt müssen Sie Postfix noch dazu bringen, dass es alle eintreffenden E-Mails durch den SpamAssassin-Filter leitet. Dazu fügen Sie die folgende Zeile in `main.cf` ein. Beachten Sie, dass der Pfad zur Socket-Datei relativ zum Postfix-Queue-Verzeichnis `/var/spool/postfix` angegeben werden muss!

**Postfix-Konfiguration**

```
# Ergänzung in /etc/postfix/main.cf
...
smtpd_milters = unix:spamass/spamass.sock
```

Anschließend laden Sie die Postfix-Konfiguration neu:

```
root# systemctl reload postfix
```

**Test** Um SpamAssassin auszuprobieren, senden Sie von einem externen E-Mail-Account eine speziell für SpamAssassin konzipierte Testnachricht an Ihren Server. Diese Nachricht muss die folgende Zeichenkette enthalten. Anstatt die Zeichenkette abzutippen, können Sie in der Wikipedia nach GTUBE (*Generic Test for Unsolicited Bulk Email*) suchen und die Zeichenkette von dort kopieren.

```
XJS*C4JDBQADN1.NSBN3*2IDNEN*GTUBE-STANDARD-ANTI-UBE-TEST-EMAIL*C.34X
```

Wenn alles funktioniert, wird die Nachricht als Spam erkannt. Der Adressat erhält die als Spam markierte E-Mail zusammen mit einer Information, warum es sich vermutlich um Spam handelt. Der Text der Nachricht sollte in etwa so aussehen:

Software zur Erkennung von "Spam" auf dem Rechner kofler.info hat die eingegangene E-mail als mögliche "Spam"-Nachricht identifiziert. Die ursprüngliche Nachricht wurde an diesen Bericht angehängt, sodass Sie sie anschauen können (falls es doch eine legitime E-Mail ist) oder ähnliche unerwünschte Nachrichten in Zukunft markieren können. Bei Fragen zu diesem Vorgang wenden Sie sich bitte an

```
@@CONTACT_ADDRESS@@
```

```
Vorschau: XJS*C4JDBQADN1.NSBN3*2IDNEN*GTUBE-STANDARD-ANTI-UBE-TEST-EMAIL*C.34X
test [...]
```

Inhaltsanalyse im Detail: (1000.0 Punkte, 5.0 benötigt)

| Pkte | Regelname              | Beschreibung   |
|------|------------------------|--|
| -0.0 | RCVD_IN_DNSWL_NONE     | RBL: Sender listed at <a href="http://www.dnswl.org/">http://www.dnswl.org/</a> , no trust<br>[213.165.64.42 listed in list.dnswl.org] |
| 0.0  | FREEMAIL_FROM          | Sender email is commonly abused enduser mail provider  |
| -0.0 | T_RP_MATCHES_RCVD      | Envelope sender domain matches handover relay domain   |
| -0.0 | SPF_PASS               | SPF: Senderechner entspricht SPF-Datensatz   |
| 0.0  | UNPARSEABLE_RELAY      | Informational: message has unparseable relay lines   |
| 1000 | GTUBE                  | BODY: Test zur Prüfung von Anti-Spam-Software  |
| 0.0  | T_TO_NO_BRKTS_FREEMAIL | T_TO_NO_BRKTS_FREEMAIL   |

**Whitelist** Um E-Mails von einer Domäne generell zu akzeptieren, können Sie die Domäne mit der folgenden Anweisung in eine Whitelist aufnehmen:

```
# Datei /etc/spamassassin/local.cf
...
whitelist_from *@befreundete-firma.de
```



Weitere Möglichkeiten zur Formulierung bzw. zur automatischen Generierung von Whitelists (z. B. auf der Basis aller Adressen, an die Mails versendet wurden) sind auf der folgenden Seite beschrieben:

<https://wiki.apache.org/spamassassin/ManualWhitelist>

## Greylisting

Eine Alternative oder Ergänzung zu SpamAssassin kann das Greylisting sein. Dabei verwaltet das Mail-System eine Datenbank aller bekannten E-Mail-Adressen. Trifft eine E-Mail von einem noch unbekanntem Absender ein, wird sie vorübergehend abgewiesen. Der Absender sollte davon gar nichts bemerken. Sein Mail-Server wird, wie dies der SMTP-Standard vorsieht, nach ein paar Minuten einen weiteren Zustellversuch unternehmen. Das Programm `greylist` wird die E-Mail dann akzeptieren und ein Datentriplett speichern, das aus Absender-Host, Absender-IP-Adresse und Absender-Mail-Adresse besteht. In Zukunft werden E-Mails, bei denen alle drei Parameter übereinstimmen, ohne Verzögerung akzeptiert.

Greylisting funktioniert trotz des minimalistischen Ansatzes überraschend gut, weil sich viele Spammer nicht die Mühe machen, E-Mails wiederholt zuzustellen. Weitere Informationen zu Greylisting finden Sie hier:

<https://de.wikipedia.org/wiki/Greylisting>

<https://wiki.centos.org/HowTos/postgrey>

[http://www.postfix.org/SMTPE\\_POLICY\\_README.html#greylist](http://www.postfix.org/SMTPE_POLICY_README.html#greylist)

Greylisting hat aber natürlich den Nachteil, dass E-Mails von neuen Geschäftskontakten beim ersten Mal erst mit mehreren Minuten Verzögerung eintreffen. Das ist einigermassen lästig, wenn Sie auf den Bestätigungs-Code bei der Anmeldung auf einer Webseite oder nach der Buchung eines Flugs warten. Auch wenn Sie einen Pizza-Zustell-Service mit E-Mail-Bestellmöglichkeit betreiben, ist Greylisting nicht der richtige Ansatz.

**Nachteile**

Besonders problematisch ist, dass nicht jeder Mail-Server gleich auf Greylisting reagiert. Manche Mail-Server führen den zweiten Zustellversuch erst Stunden später durch. Große Firmen verwenden wiederum ganze Cluster von Mail-Servern. Das kann dazu führen, dass der zweite Zustellversuch von einem anderen Mail-Server innerhalb des Clusters durchgeführt wird – und von Ihrem Mail-Server deswegen neuerlich abgelehnt wird.

Bei den meisten Distributionen können Sie das Perl-Script `postgrey` in Form des gleichnamigen Pakets installieren. Das Programm wird unter Debian/Ubuntu sofort als Hintergrundprogramm gestartet. Unter CentOS/RHEL müssen Sie sich wie üblich

**Installation und Konfiguration von postgrey**

selbst um den Start kümmern und `systemctl start postgrey` sowie `systemctl enable postgrey` ausführen.

Jetzt geht es noch darum, `postgrey` in `main.cf` einzubinden. Dazu müssen Sie die vorgegebene `smtpd_recipient_restrictions`-Einstellung um einen weiteren Eintrag ergänzen. Unter CentOS/RHEL kommuniziert `postgrey` standardmäßig über eine Socket-Datei (siehe `ps ax | grep postgrey`):

```
# /etc/postfix/main.cf (CentOS/RHEL)
smtpd_recipient_restrictions = permit_mynetworks,
                               permit_sasl_authenticated,
                               reject_unauth_destination,
                               check_policy_service unix:postgrey/socket
```

Unter Debian/Ubuntu kommuniziert `postgrey` dagegen standardmäßig über den lokalen Port 10023.

```
# /etc/postfix/main.cf (Debian/Ubuntu)
smtpd_recipient_restrictions = permit_mynetworks,
                               permit_sasl_authenticated,
                               reject_unauth_destination,
                               check_policy_service unix:postgrey/socket
```

Während der Timeout für weitere Zustellversuche unter CentOS/RHEL mit 50 Sekunden angenehm kurz gewählt ist (`/etc/sysconfig/postgrey`), liegt die Defaultzeitspanne bei Debian/Ubuntu bei schon recht langen fünf Minuten. Wenn Sie möchten, können Sie diese Zeit in `/etc/default/postgrey` verkürzen:

```
POSTGREY_OPTS="--inet=10023 --delay=50"
```

Anschließend müssen Sie `postfix` auffordern, seine Konfiguration neu einzulesen:

```
root# systemctl restart postfix
```

**Kontrolle** Um sich zu vergewissern, dass das Greylisting funktioniert, senden Sie von einem anderen Account eine E-Mail an Ihren Mail-Server. Die Mail sollte nicht sofort eintreffen, sondern erst mit einer Verzögerung von einigen Minuten. Dabei sollten in `mail.log` bzw. `maillog` Einträge nach dem folgenden Muster auftauchen:

```
root# grep greylist /var/log/maillog
Apr 28 12:15:55 postgrey: action=greylist, reason=new, client_name=mout.gmx.net,
  client_address=212.227..., sender=...@gmx.com, recipient=...
...
Apr 28 12:20:56 postgrey[2290]: action=pass, reason=triplet found, delay=301,
  client_name=mout.gmx.net, client_address=212.227..., sender=...@gmx.com, ...
```

Die Mail wurde also beim ersten Versuch abgewiesen (*Temporary Error*, `action=greylist` in der Logging-Datei). Einige Minuten später versucht der externe Mail-Server es erneut. Jetzt wird die Mail akzeptiert (`action=pass`).

## 34.8 ClamAV (Virenabwehr)

Mit Spam ist es leider noch nicht getan: Wenn sich in Ihrem Netzwerk Windows-Rechner befinden, müssen Sie auch dafür sorgen, dass die E-Mails frei von Viren eintreffen. Nun sind per E-Mail verbreitete Viren für Windows-PCs heute nicht mehr das ganz große Thema, das sie vor ein paar Jahren waren. E-Mail-Viren sind aber noch immer ein sporadisches Risiko, das es zu minimieren gilt. Deswegen empfiehlt es sich, auf dem E-Mail-Server ein weiteres Programm zu installieren, das E-Mails bzw. deren Anhänge auf bekannte Viren untersucht und befallene E-Mails löscht. Technisch funktioniert ein E-Mail-Filter ähnlich wie ein Spam-Filter, wobei aber vor allem Anhänge auf Muster bekannter E-Mails durchsucht werden müssen. Das kostet eine Menge CPU-Zeit und ist nur dann zielführend, wenn eine aktuelle Virendatenbank zur Verfügung steht.

ClamAV ist das populärste Open-Source-Programm zur Erkennung von Viren in Dateien oder E-Mails. Dabei geht es primär um Schadsoftware für Windows-Rechner. E-Mail-Viren für Linux und macOS gibt es ja glücklicherweise noch nicht. Im Vergleich zu kommerziellen Virenschutzprogrammen war ClamAV in der Vergangenheit selten Testsieger, hat sich aber in der Regel einigermaßen gut geschlagen. Eine Garantie, die allerneuesten Viren von der ersten Stunde an korrekt zu erkennen, gibt es aber nicht.

Ähnlich wie bei SpamAssassin gibt es auch bei ClamAV verschiedene Möglichkeiten zur Integration in Postfix. Ich stelle Ihnen hier wieder die Militer-Variante vor, die am einfachsten zu konfigurieren ist. Entsprechende Pakete finden Sie in allen gängigen Distributionen. Unter Debian und Ubuntu heißen sie `clamav`, `clamav-daemon` und `clamav-milter`. Zusammen mit ClamAV wird in der Regel auch das Programm `freshclam` installiert. Es kümmert sich darum, die initiale Virendatenbank herunterzuladen und in der Folge regelmäßig zu aktualisieren. Werfen Sie einen Blick in das Verzeichnis `/var/lib/clamav` (es darf nicht leer sein!) bzw. lesen Sie die `man`-Seite zu `freshclam`.

**Installation**

Damit Sie ClamAV über die Postfix-Milter-Schnittstelle benutzen können, sind einige Vorbereitungsarbeiten erforderlich: Als Erstes entfernen Sie in `/etc/default/clamav-milter` das Kommentarzeichen für die bereits vorgesehene Zeile zur Postfix-Konfiguration. Die Variable `SOCKET_RWGROUP` gibt an, welcher Gruppe die ClamAV-Socket-Datei zugeordnet werden soll.

**Konfiguration**

```
# in /etc/default/clamav-milter
...
SOCKET_RWGROUP=postfix
```

In `/etc/clamav/clamav-milter.conf` geben Sie an, an welchem Ort die ClamAV-Socket-Datei erzeugt werden soll:

```
# in /etc/clamav/clamav-milter.conf
...
MilterSocket /var/spool/postfix/clamav/clamav-milter.ctl
```

Anschließend erzeugen Sie das Verzeichnis für die Socket-Datei so, dass sowohl Postfix als auch ClamAV darin lesen und schreiben dürfen:

```
root# mkdir -p /var/spool/postfix/clamav/
root# chown clamav:postfix /var/spool/postfix/clamav/
root# chmod g+s /var/spool/postfix/clamav/
```

Ein Neustart von `clamd` und `clamav-milter` stellt sicher, dass ClamAV diese Änderungen übernimmt:

```
root# systemctl restart clamav-daemon
root# systemctl restart clamav-milter
```

Nun müssen Sie noch `/etc/postfix/main.cf` so anpassen, dass Postfix alle eintreffenden E-Mails zur Kontrolle an ClamAV weiterleitet. Wenn Sie nur ClamAV, nicht aber SpamAssassin verwenden, lassen Sie die Socket-Datei für SpamAssassin einfach weg. Die Pfadangaben der Socket-Dateien sind relativ zum Postfix-Queue-Verzeichnis `/var/spool/postfix`.

```
# Ergänzung in /etc/postfix/main.cf
...
smtpd_milters = unix:spamass/spamass.sock unix:clamav/clamav-milter.ctl
```

Dank `postfix reload` übernimmt Postfix die Konfigurationsänderung sofort:

```
root# systemctl reload postfix
```

**Test** ClamAV fügt nun in den Header jeder überprüften E-Mail den folgenden Text ein:

```
X-Virus-Scanned: clamav-milter 0.96.1 at firma-abc.de
X-Virus-Status: Clean
```

Wenn ClamAV tatsächlich einen Virus feststellt, wird die E-Mail nicht weitergeleitet. Weder der Absender noch der Empfänger wird davon informiert. Diese Vorgehensweise kann in `/etc/clamav/clamav-milter.conf` verändert werden. Die dort eingesetzten Schlüsselwörter sind in `/usr/share/doc/clamav-milter/examples/clamav-milter.conf` dokumentiert.

Um die korrekte Funktion von ClamAV zu testen, senden Sie von einem externen E-Mail-Account eine Testnachricht mit dem folgenden Text an Ihren Server:

```
X50!P%@AP[4\ZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

Sie müssen den Text nicht abtippen, sondern können ihn auch von der folgenden Wikipedia-Seite kopieren:

[http://en.wikipedia.org/wiki/EICAR\\_test\\_file](http://en.wikipedia.org/wiki/EICAR_test_file)

## 34.9 SPF, DKIM und DMARC

Eine Überschrift nur aus Abkürzungen, entsetzlich! Die drei Abkürzungen beschreiben Verfahren, die zusätzliche Informationen zum Verhalten Ihres Mail-Servers in DNS-Einträgen speichern:

- ▶ **Sender Policy Framework (SPF):** Bei diesem Verfahren wird in einem DNS-Eintrag gespeichert, welche Hosts berechtigt sind, E-Mails für die Domänen zu versenden.
- ▶ **DomainKeys Identified Mail (DKIM):** Bei diesem Verfahren wird in einem DNS-Eintrag der öffentliche Teil eines Schlüssels gespeichert. Der Mail-Server signiert die E-Mail mit einem privaten Schlüssel. Der Empfänger kann anhand des öffentlichen Schlüssels die Signatur testen und so zweifelsfrei feststellen, dass die E-Mail wirklich vom vorgeblichen Host versendet wurde.
- ▶ **Domain-based Message Authentication, Reporting and Conformance (DMARC):** Wenn bei Ihrem Mail-Server sowohl SPF als auch DKIM funktioniert, können Sie zusätzlich auch noch DMARC aktivieren: In einem weiteren DNS-TXT-Eintrag geben Sie so öffentlich bekannt, dass Ihr Mail-Server SPF und DKIM unterstützt und wie mit Mails verfahren werden soll, die den SPF-Regeln widersprechen bzw. die eine falsche oder gar keine Signatur aufweisen.

Wozu das alles? Während SpamAssassin und Greylisting vor unerwünschtem Spam schützen soll, geht es jetzt darum, den eigenen Mail-Server so regelkonform wie möglich zu konfigurieren. Nichts ist ärgerlicher, als wenn die Mails vom eigenen Server von Google, Apple oder Microsoft als Spam klassifiziert werden.

Nun kann niemand Spammer daran hindern, ebenfalls SPF, DKIM und DMARC zu verwenden. Insofern sind DKIM und SPF kein eindeutiges Kriterium zur Spam-Erkennung. Aber die Implementierung dieser Verfahren verbessert die Chance, dass E-Mails vom eigenen Server den Empfänger nicht im Spam-Ordner erreichen – zumindest ein ganz klein wenig: Meiner Erfahrung nach ist der Nutzen nämlich nur geringfügig höher als der von SpamAssassin bei der Spam-Abwehr.

### Sender Policy Framework (SPF)

Im *Sender Policy Framework* geben Sie mit einem DNS-Eintrag vom Typ TXT bekannt, von welchem Host bzw. von welcher IP-Adresse Mails der eigenen Domäne versendet werden dürfen. Außer dem Einrichten des korrekten DNS-TXT-Eintrags ist keine weitere Konfigurationsarbeit notwendig. Ähnlich minimal ist aber auch der Nutzen. Besonders ungnädig ist der Postfix-Experte Peer Heinlein, der SPF als »Bullshit und Broken by Design bezeichnet«, insbesondere im Kontext von Mailing-Listen:

<https://www.heinlein-support.de/blog/news/gmx-de-und-web-de-haben-mail-rejects-durch-spf>

**SPF-Syntax** Die Syntax ist einfach: Die Zeichenkette muss mit `v=spf1` beginnen. Danach werden mit `ip4:1.2.3.4` bzw. `ip6:1234:...:6789` die IP-Adressen der Server aufgezählt, die E-Mails für den betreffenden Mail-Server versenden dürfen. Bei einer einfachen Mail-Server-Konfiguration ohne Backup- und Relaying-Server handelt es sich hierbei einfach um die IP-Adressen des Servers.

Der Eintrag endet üblicherweise mit `~all`. Das bedeutet, dass alle anderen IP-Adressen keine E-Mails für die eigene Domäne versenden *sollen*. Noch strenger ist der Eintrag `-all`, der dies explizit verbietet. Die Einstellung `-all` wird aber nicht empfohlen, weil sie beim Weiterleiten von E-Mails oft Probleme verursacht.

Der folgende TXT-Eintrag besagt also, dass nur E-Mails, die von den IP-Adressen `5.9.22.29` bzw. `2a01:4f8:161:107::4` versendet wurden, als zulässige E-Mails des Mail-Servers `kofler.info` zu betrachten sind:

```
user$ host -t TXT kofler.info
kofler.info descriptive text
      "v=spf1 ip4:5.9.22.29 ip6:2a01:4f8:161:107::4 ~all"
```

Die gleiche Information hätte auch in der Form `"v=spf1 mx ~all"` verpackt werden können. Das bedeutet, dass alle IP-Adressen der durch MX-Einträge angegebenen Hostnamen gültig sind. Wie Sie sehen, stimmt das mit der obigen Angabe überein:

```
user$ host -t MX kofler.info
kofler.info mail is handled by 10 mail.kofler.info.
user$ host -t A mail.kofler.info
mail.kofler.info has address 5.9.22.29
user$ host -t AAAA mail.kofler.info
mail.kofler.info has IPv6 address 2a01:4f8:161:107::4
```

Besonders einfach können Sie den für Ihren Mail-Server passenden SPF-Eintrag über ein Formular auf dieser Seite ermitteln:

<https://www.spf-record.de>

**TXT- versus SPF-Eintrag** Ursprünglich war geplant, für SPF-Einträge einen eigenen Typ von DNS-Einträgen zu definieren. Es hat sich dann aber als einfacher herausgestellt, die Informationen einfach in den ohnedies schon vorgesehenen Texteinträgen unterzubringen. Weitere Informationen zu SPF können Sie hier nachlesen:

[https://de.wikipedia.org/wiki/Sender\\_Policy\\_Framework](https://de.wikipedia.org/wiki/Sender_Policy_Framework)  
<https://spfwizard.net/what-is-spf-record.html>

Damit bleibt nur noch die Frage: Wie verändern Sie die DNS-Konfiguration? Kompetente Provider bzw. Domänenregistrare geben Ihnen die Möglichkeit, bei der DNS-Konfiguration über eine Weboberfläche auch die TXT-Einträge zu verändern (siehe [Abbildung 34.2](#)).

## DomainKeys Identified Mail (DKIM)

Schon deutlich komplexer und aufwendiger zu realisieren ist DomainKeys Identified Mail: Wenn ein DKIM-konfigurierter Mail-Server eine Mail versendet, verwendet er einen auf dem Mail-Server gespeicherten privaten Schlüssel und fügt der E-Mail eine Signatur hinzu. Der Empfänger-Mail-Server kann nun zur Kontrolle den DNS-TXT-Eintrag des Senders auslesen. Dieser enthält den öffentlichen Teil des Schlüssels. Damit kann überprüft werden, ob die E-Mail und die Signatur zusammenpassen (siehe [Abbildung 34.4](#)). Mit DKIM kann also überprüft werden, ob die Mail tatsächlich von der in ihr angegebenen Absenderadresse stammt – und das ist schon eine Menge wert. (Nur der legitime Mail-Server verfügt über den privaten Schlüssel und kann damit eine korrekte Signatur durchführen.)

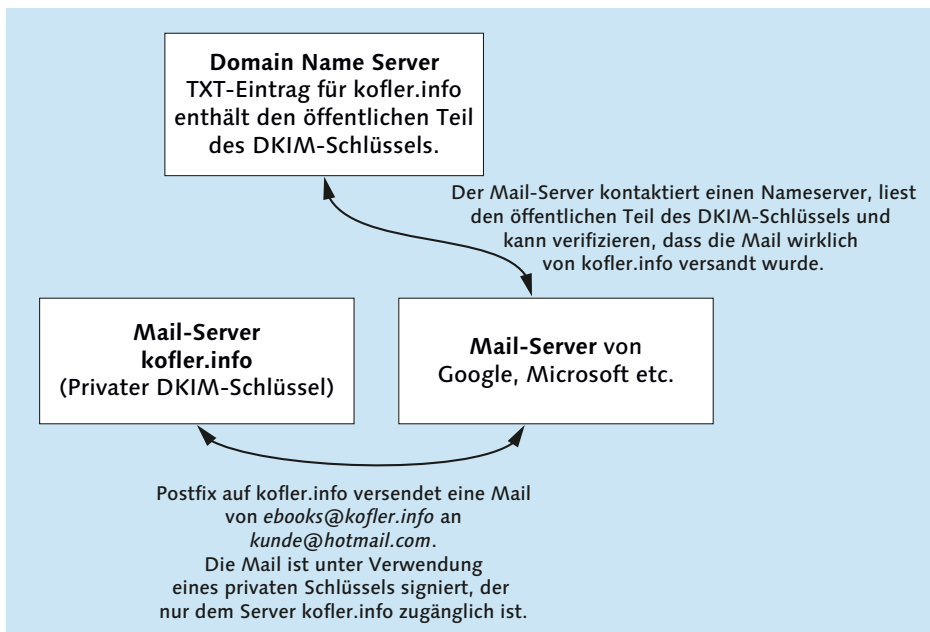


Abbildung 34.4 Das Konzept von DomainKeys Identified Mail (DKIM)

## OpenDKIM

OpenDKIM ist eine Open-Source-Implementierung von DKIM. Das Programm kann gleichermaßen ausgehende Mails mit einem eigenen Schlüssel signieren als auch die Signatur eintreffender Mails überprüfen. OpenDKIM ist mit diversen MTAs kompatibel. Diese Anleitung zeigt das Zusammenspiel von DKIM mit Postfix, wobei das Hauptaugenmerk des Artikels beim korrekten Signieren liegt.

OpenDKIM  
installieren und  
einrichten

Unter Debian/Ubuntu installieren Sie die Pakete `opendkim` und `opendkim-tools`, unter CentOS/RHEL reicht nur `opendkim`. Dann richten Sie ein Verzeichnis für die DKIM-Schlüssel ein:

```
root# mkdir -p /etc/opendkim/keys
root# chown -R opendkim:opendkim /etc/opendkim
root# chmod go-rw /etc/opendkim/keys
```

**opendkim.conf** Die zentrale Konfigurationsdatei von OpenDKIM ist `/etc/opendkim.conf`. Diese Datei richten Sie wie das folgende Muster ein. Details zu den Dateien `trusted`, `key.table` und `signing.table`, auf die die Konfiguration verweist, folgen gleich. Einzelheiten zu den diversen `opendkim.conf`-Optionen können Sie mit `man opendkim.conf` nachlesen.

```
# Datei /etc/opendkim.conf
# OpenDKIM agiert als Mail Filter (= Milter) in den
# Modi signer (s) und verifier (v) und verwendet
# einen Port oder eine Socket-Datei zur Kommunikation
Mode                sv
# Socket             local:/var/run/opendkim/opendkim.sock
# Socket             inet:12345@localhost
Socket              inet:8891@localhost

# OpenDKIM verwendet diesen Benutzer bzw. diese Gruppe
UserID              opendkim:opendkim
UMask               002
PidFile             /var/run/opendkim/opendkim.pid

# OpenDKIM bei Problemen neu starten, aber max. 10-mal pro Stunde
AutoRestart         yes
AutoRestartRate     10/1h

# Logging (wenn alles funktioniert eventuell reduzieren)
Syslog              yes
SyslogSuccess       yes
LogWhy              yes

# Verfahren, wie Header und Body durch OpenDKIM verarbeitet
# werden sollen
Canonicalization    relaxed/simple

# interne Mails nicht mit OpenDKIM verarbeiten
ExternalIgnoreList  refile:/etc/opendkim/trusted
InternalHosts       refile:/etc/opendkim/trusted

# welche Verschlüsselungs-Keys sollen für welche
# Domänen verwendet werden
# (refile: für Dateien mit regulären Ausdrücken)
SigningTable        refile:/etc/opendkim/signing.table
KeyTable            /etc/opendkim/key.table
```



```
# diesen Signatur-Algorithmus verwenden
SignatureAlgorithm    rsa-sha256

# Always oversign From (sign using actual From and a null From to prevent
# malicious signatures header fields (From and/or others) between the signer
# and the verifier. From is oversigned by default in the Debian package
# because it is often the identity key used by reputation systems and thus
# somewhat security sensitive.
OversignHeaders      From
```

Beachten Sie, dass unter Debian und Ubuntu beim Start von OpenDKIM auch die Datei `/etc/default/opendkim` ausgewertet wird. Dort durchgeführte Einstellungen haben Vorrang gegenüber `opendkim.conf`. Standardmäßig enthält `/etc/default/opendkim` nur Kommentare, und ich gehe in dieser Anleitung davon aus, dass Sie es dabei belassen.

Auch unter CentOS/RHEL gibt es Eigenheiten: Dessen Musterdatei für `opendkim.conf` verwendet den Port 8891. Nun könnte man meinen, jeder andere freie Port würde ebenso funktionieren. Dem ist aber nicht so: Nur für Port 8891 gibt es eine SELinux-Ausnahmeregel, die `opendkim` die Verwendung erlaubt. Wenn Sie einen anderen Port verwenden möchten, führen Sie das folgende Kommando aus, wobei Sie natürlich anstelle von 17654 die gewünschte Port-Nummer angeben:

```
root# semanage port -a -t milter_port_t -p tcp 17654
```

Die Datei `trusted` gibt an, welchen Hosts der Mail-Server vertraut, d.h., für welche Hosts auf die DKIM-Signatur verzichtet wird. Ersetzen Sie die drei `kofler.info`-Einträge durch entsprechende Einträge mit dem Hostnamen Ihres Mail-Servers!

**Keine Signatur für interne Mails**

```
# Datei /etc/opendkim/trusted
127.0.0.1
::1
localhost
kofler
kofler.info
host1.kofler.info
```

Als Nächstes richten Sie mit einem Editor die Dateien `signing.table` und `key.table` gemäß dem folgenden Muster ein. `signing.table` gibt an, für welche `From`-Adressen welcher Schlüssel verwendet werden soll. Die zweite Spalte in `signing.table` bezieht sich dabei auf die erste Spalte in `key.table`. Natürlich müssen Sie `kofler.info` und `kofler` durch eigene Namen ersetzen!

**Welcher Schlüssel für welchen Hostnamen?**

```
# Datei /etc/opendkim/signing.table
# für E-Mails von xxx@kofler.info den Schlüssel 'kofler'
# zum Signieren verwenden
*@kofler.info kofler
```

key.table gibt dann den tatsächlichen Ort der Schlüsseldateien an. Die zweite Spalte besteht dabei aus drei Teilen: dem Hostnamen (hier kofler.info), dem Selektor (hier 201705) und dem eigentlichen Dateinamen. Die Verwendung einer Zeitangabe für den Selektor hilft dabei, einen Überblick zu behalten, wann die Schlüssel erzeugt wurden – hier also im Mai 2017.

```
# Datei /etc/opendkim/key.table
# Der Schlüssel 'kofler' befindet sich in
# der Datei /etc/opendkim/keys/kofler.private
kofler kofler.info:201705:/etc/opendkim/keys/kofler.private
```

Wenn Ihr Mail-Server für mehrere Hosts zuständig ist, ergänzen Sie key.table und signing.table um entsprechende Einträge für alle weiteren Hosts.

#### Schlüssel generieren

Das Kommando `opendkim-genkey` erzeugt einen privaten Schlüssel (`name.private`) und einen öffentlichen Schlüssel (`name.txt`). Der private Schlüssel verbleibt am Server und darf nicht in falsche Hände geraten! Der öffentliche Schlüssel wird später im DNS-TXT-Record des Mail-Servers gespeichert. Die wichtigsten Optionen des Kommandos lauten:

- ▶ `-d domain` gibt an, für welche Domäne die Schlüssel gelten sollen. Die Option wird aktuell nur dazu verwendet, um einen Kommentar in `name.txt` einzubauen. Ohne die Option verwendet `opendkim-genkey` die Zeichenkette `example.com`.
- ▶ `-b 2048` gibt an, dass ein Schlüssel in der Länge von 2048 Bits generiert werden soll. Größere Schlüssel sind momentan mit der gängigen Mail-Server-Infrastruktur inkompatibel. Standardmäßig verwendet `opendkim-genkey` 1024 Bits, was aber vielfach als nicht mehr als ausreichend sicher erachtet wird.
- ▶ `-r` gibt an, dass der Schlüssel ausschließlich zur Signatur von E-Mails verwendet werden darf (*restricted*).
- ▶ `-s selector` gibt eine Selektor-Zeichenkette an, um das Schlüsselpaar zu identifizieren (standardmäßig: `default`). Die Selektor-Zeichenkette bestimmt die Namen der Schlüsseldateien. `-s 201705` bewirkt also, dass die Dateien `201705.txt` und `201705.private` erzeugt werden. Vorsicht: Eventuell schon vorhandene Schlüsseldateien werden ohne Rückfrage überschrieben.

Um also Schlüssel zu erzeugen, die zum obigen Beispiel passen, führen Sie das folgende Kommando aus:

```
root# cd /etc/opendkim
root# opendkim-genkey -d kofler.info -b 2048 -r -s 201705
```

Die Dateien müssen nun noch entsprechend der Angaben in `key.table` platziert werden. Zuletzt stellen Sie sicher, dass nur der Benutzer `opendkim` auf die Dateien und Verzeichnisse zugreifen darf:

```

root# cd /etc/opendkim
root# mv 201705.private keys/kofler.private
root# mv 201705.txt keys/kofler.txt
root# chown -R opendkim:opendkim /etc/opendkim
root# chmod -R go-rwx /etc/opendkim/keys

```

Der nächste Schritt besteht darin, den öffentlichen Teil des DKIM-Schlüssels, also in der Logik dieses Beispiels den Inhalt von `kofler.txt` in einem TXT-Eintrag in der DNS-Konfiguration Ihres Hosts zu speichern:

DNS-TXT-Eintrag  
hinzufügen

```

201705._domainkey IN TXT (
    "v=DKIM1; k=rsa; s=email; "
    "p=MIIBIjANB..."
    "LLV1AmTutta..." ) ; ----- DKIM key 201705 for kofler.info

```

Dabei verwenden Sie die Zeichenkette vor dem Schlüsselwort `IN` für das Feld `HOSTNAME` des DNS-Eintrags und den gesamten Text zwischen den runden Klammern für das Feld `DESTINATION` (siehe [Abbildung 34.5](#)).

**Abbildung 34.5** Definition eines neuen TXT-Eintrags in der Weboberfläche eines Hosting-Unternehmens

Bis die Änderung der DNS-Daten im Internet allgemein verfügbar ist, dauert es möglicherweise mehrere Stunden. Die TXT-Einträge eines Hosts können Sie mit dem Kommando `host -t TXT` auslesen, wobei Sie Ihrem Domainnamen die Bezeichnung `<selector>._domainkey` voranstellen müssen, hier also `201705._domainkey`:

```

root# host -t TXT 201705._domainkey.kofler.info
201705._domainkey.kofler.info descriptive text
    "v=DKIM1; k=rsa; s=email; " "p=MIIBIjANB..." "LLV1AmTutta..."

```

Die geänderten Einstellungen erfordern einen Neustart von OpenDKIM:

OpenDKIM testen

```

root# systemctl restart opendkim

```

Sobald der auf dem Nameserver gespeicherte öffentliche Schlüssel verfügbar ist (das testen Sie mit dem Kommando `host -t TXT`, siehe oben), können Sie mit dem Kommando `opendkim-testkey` lokal überprüfen, ob alles korrekt eingerichtet ist. Sofern das Kommando keine Fehlermeldungen liefert, ist alles in Ordnung. Die Option `-vvv` bewirkt, dass das Kommando zumindest einige Debugging-Daten liefert.

```

root# opendkim-testkey -d kofler.info -s 201705 -vvv
opendkim-testkey: using default configfile /etc/opendkim.conf
opendkim-testkey: checking key '201705._domainkey.kofler.info'
opendkim-testkey: key not secure
opendkim-testkey: key OK

```

*Key not secure* weist darauf hin, dass kein DNSSEC verwendet wird. Das hat aber nichts mit DKIM zu tun. Sollte hingegen die Meldung *record not found* erscheinen, kann OpenDKIM den erforderlichen Schlüssel nicht finden – dann liegt wirklich ein Konfigurationsproblem vor.

### Postfix-Konfiguration für OpenDKIM

Damit Postfix ausgehende E-Mails mit OpenDKIM signiert, muss OpenDKIM als Mail-Filter (= Milter) eingerichtet werden. Dazu sind in der Postfix-Konfiguration die Parameter `smtpd_milters` und `non_smtpd_milters` vorgesehen. Wenn OpenDKIM der einzige Milter ist, sieht die korrekte Konfiguration so aus:

```

# in /etc/postfix/main.cf
milter_default_action = accept
milter_protocol      = 6
smtpd_milters        = inet:localhost:8891
non_smtpd_milters    = inet:localhost:8891

```

Vielleicht gab es bisher schon Milter – dann kommt der OpenDKIM-Milter (getrennt durch ein Komma) eben dazu. Das Beispiel zeigt eine Kombination mit SpamAssassin:

```

# in /etc/postfix/main.cf
... (restliche Zeilen wie oben)
smtpd_milters      = unix:spamass/spamass.sock, inet:localhost:8891
non_smtpd_milters = inet:localhost:8891

```

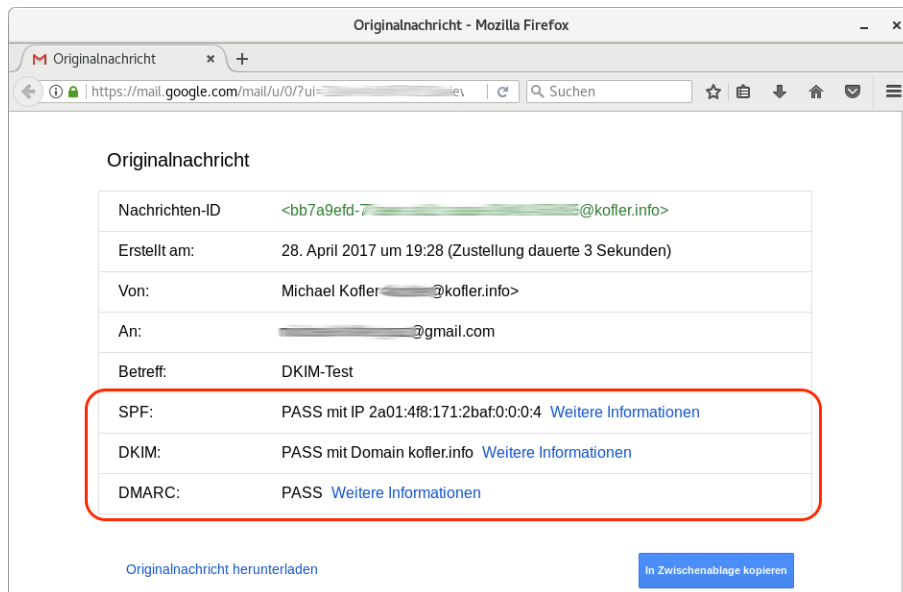
Einige Anmerkungen zu den Parametern:

- ▶ `milter_default_action = accept` bedeutet, dass Postfix auch dann Mails akzeptieren soll, wenn ein Milter defekt ist, nicht reagiert, falsch konfiguriert ist etc.
- ▶ In vielen Anleitungen ist von `milter_protocol = 2` die Rede. Das gilt aber nur für alte Postfix-Versionen (älter als Version 2.6). Bei aktuellen Postfix-Versionen gilt standardmäßig `milter_protocol = 6`. Die Konfiguration funktioniert also auch, wenn Sie die Einstellung ganz weglassen.
- ▶ Vielleicht fragen Sie sich, was der Unterschied zwischen `smtpd_milters` und `non_smtpd_milters` ist: Die `smtpd_milters` gelten für E-Mails, die über das SMTP-Protokoll geliefert werden. `non_smtpd_milters` gilt dagegen für lokale E-Mails, die z. B. durch `sendmail` oder durch ein PHP-Skript erzeugt und versendet werden. In der obigen Konfiguration wird für solche Mails auf den Spam-Test verzichtet.

Um DKIM im Zusammenspiel von Postfix und OpenDKIM zu testen, senden Sie sich **Test** zuerst selbst eine Mail und werfen dann einen Blick in den Header. Dieser sollte eine Zeile enthalten, die mit der DKIM-Signatur beginnt, z. B. wie hier:

```
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/simple; d=kofler.info;
s=201705; t=1479644038;
bh=g3zLYH4xKxcPrHOD18z9YfpQcnk/GaJedfustWU5uGs=;
h=To:From:Subject:Date:From;
b=VI6TIDLrzG8nAUyWwt5QasKJkkGU+Sv8sPGC1fynSEGoGSULGgCjVN6KXPfx1rgm
1uX2sWET/oMcpXjBFBVUbM7yHGd1lhbADa2SarzYhkoEuNhmo+yxGpXkuh0ttn4z7n
```

Senden Sie nun eine E-Mail von Ihrem eigenen Mail-Server an einen Mail-Server, von dem Sie wissen, dass er DKIM unterstützt und verwendet. Eine gute Wahl ist Google (also Gmail). In der Gmail-Weboberfläche öffnen Sie die E-Mail und führen ORIGINAL ANZEIGEN aus (siehe [Abbildung 34.6](#)).



**Abbildung 34.6** Google Mail eignet sich gut zur Kontrolle der DKIM-Konfiguration.

### Domain-based Message Authentication, Reporting and Conformance (DMARC)

Wenn bei Ihrem Mail-Server sowohl SPF als auch DKIM funktioniert, können Sie sich noch überlegen, *Domain-based Message Authentication, Reporting and Conformance* (DMARC) zu aktivieren. Dazu geben Sie mit einem weiteren DNS-TXT-Eintrag öffentlich bekannt, dass Ihr Mail-Server SPF und DKIM unterstützt und wie mit Mails verfahren werden soll, die Ihren Hostnamen in der Absenderadresse enthalten, die aber von anderen Mail-Servern versandt wurden bzw. die nicht korrekt mit DKIM signiert sind.

Das Einrichten von DMARC ist mit wenig Arbeit verbunden. Sie müssen lediglich einen weiteren DNS-TXT-Eintrag definieren, der als Hostname `_dmarc` verwendet und einen Eintrag gemäß der DMARC-Syntax aufweist. Dabei können Sie sich an den DMARC-Einstellungen anderer Mail-Provider orientieren:

```
user$ host -t TXT _dmarc.yahoo.com
_dmarc.yahoo.com descriptive text
"v=DMARC1; p=reject; pct=100; rua=mailto:dmarc_y_rua@yahoo.com;"
```

```
user$ host -t TXT _dmarc.gmail.com
_dmarc.gmail.com descriptive text
"v=DMARC1; p=none; rua=mailto:mailauth-reports@google.com"
```

```
user$ host -t TXT _dmarc.hotmail.com
_dmarc.hotmail.com descriptive text
"v=DMARC1; p=none; pct=100; rua=mailto:d@rua.agari.com;
ruf=mailto:d@ruf.agari.com; fo=1"
```

Am radikalsten ist Yahoo. (Diese Firma hat DMARC ursprünglich entwickelt.) Mails, die SPF und DKIM nicht einhalten, sollen ganz einfach verworfen werden. Fehlerberichte können an `dmarc_y_rua@yahoo.com` gesendet werden.

Deutlich entspannter sehen Gmail und Hotmail die Sache: `p=none` schlägt vor, nicht korrekt signierte Mails dennoch zu akzeptieren (»Monitor-Modus«). Aber auch diese Mail-Provider sind an Fehlermeldungen interessiert und geben entsprechende Mail-Adressen an.

Wie alle anderen Techniken ist auch DMARC umstritten. Wird DMARC wie bei Yahoo streng ausgeführt, dann verursachen durch Mailing-Listen weitergeleitete Mails oft Probleme. Die meisten Programme für Mailing-Listen verändern die From-Zeile nämlich nicht, wodurch es beim Weiterleiten zu einer Diskrepanz zwischen dem angegebenen Absender und dem Mail-Server der Mailing-Liste kommt.

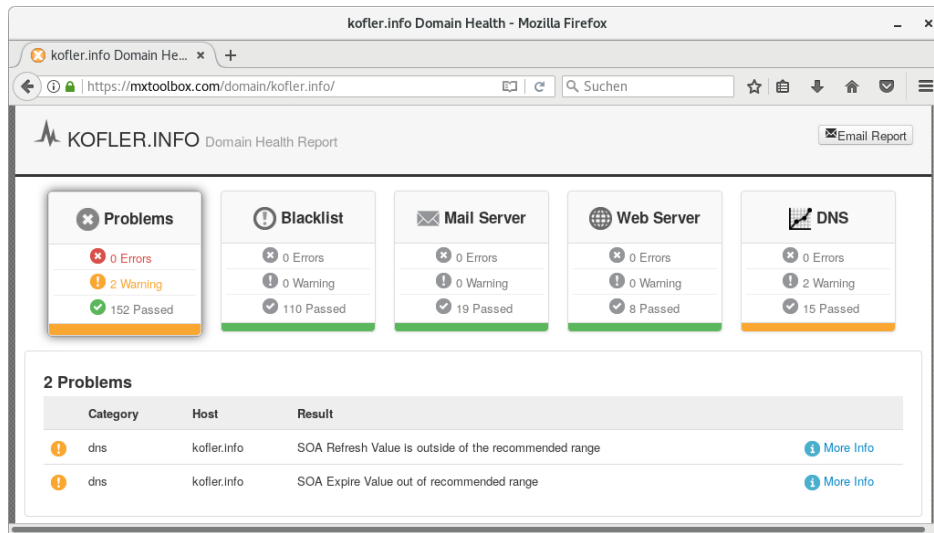
Wenn Sie also einen DMARC-Eintrag einrichten möchten, sollten Sie sich meiner Ansicht nach Gmail als Vorbild nehmen und DMARC ohne Reject-Regel nur im Monitoring-Modus aktivieren. Auf `rua` können Sie verzichten, wenn Sie nicht an automatisierten Feedback-Mails anderer Server-Betreiber interessiert sind.

An die mit dem `rua`-Schlüsselwort angegebene Feedback-Adresse erhalten Sie nun von einigen Mail-Providern, die DKIM nutzen und an die Ihr Server Mails sendet, regelmäßig (zumeist täglich) Feedback in Form einer komprimierten XML-Datei. Aus der Datei geht hervor, ob DKIM korrekt verwendet wurde und ob (vermutlich ohne Ihr Wissen) andere Mail-Server versucht haben, in Ihrem Namen Mails zu versenden und so DKIM-Fehler ausgelöst haben.

## 34.10 Konfigurationstest und Fehlersuche

Es gibt eine Reihe von Seiten, die Ihnen dabei helfen, die korrekte Konfiguration Ihres Mail-Servers zu überprüfen.

Auf <https://mxtoolbox.com> können Sie den Hostnamen Ihres Mail-Servers angeben. Die Seite überprüft, ob die DNS-Konfiguration korrekt ist und ob Ihr Server auf einer Blacklist steht (siehe [Abbildung 34.7](#)). mxtoolbox.com



**Abbildung 34.7** MXToolbox hat nur zwei geringfügige Probleme festgestellt.

Auf <https://www.checktls.com> können Sie eine gültige E-Mail-Adresse Ihres Servers angeben. Scripts der Seite überprüfen, ob die TLS-Konfiguration korrekt ist. checktls.com

Wenn Sie die Seiten <http://isnotspam.com> und <https://www.mail-tester.com> besuchen, erhalten Sie eine zufällige Mail-Adresse, an die Sie eine Test-Mail senden können. Die Mail sollte allerdings einen richtigen Inhalt enthalten, nicht nur »Test« – sonst fällt sie beim Spam-Test durch. Die E-Mail wird dann überprüft, und Sie erhalten einen Report, der verschiedene Aspekte der Konfiguration überprüft (siehe [Abbildung 34.8](#)). Spam-Test

Sollte es bei der SMTP-Authentifizierung Probleme geben, ist erstaunlicherweise das veraltete Programm `telnet` ein wertvolles Hilfsmittel zur Fehlersuche. Sie können dieses Kommando auf einem beliebigen Rechner ausführen. Es muss sich dabei nicht um den Mail-Server handeln. Fehlersuche mit telnet

Der Verbindungsaufbau über Port 25 beginnt mit Textnachrichten. Die Willkommensnachricht der Servers müssen Sie mit `EHL0 <hostname>` beantworten. Postfix gibt

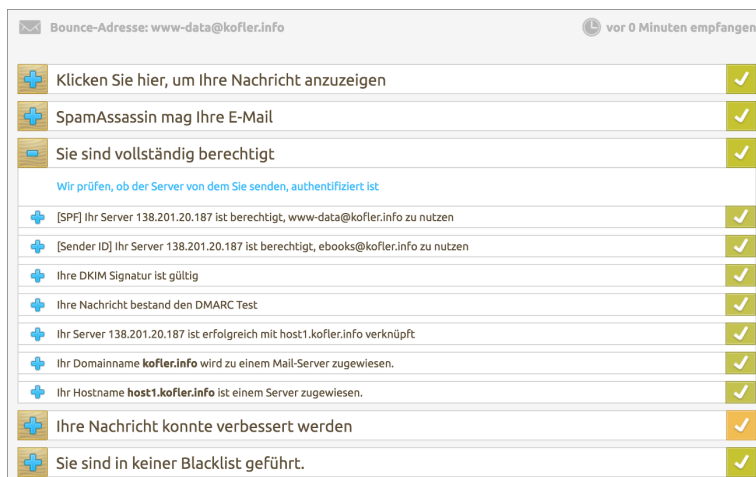


Abbildung 34.8 Beurteilung einer E-Mail, die an mail-tester.com gesendet wurde

nun bekannt, dass die weitere Kommunikation gemäß dem STARTTLS-Verfahren verschlüsselt werden kann und dass die Authentifizierung durch ein Klartextpasswort erfolgen darf. Mit quit beenden Sie nun das »Gespräch« mit dem Mail-Server.

```
user telnet mail.firma-abc.de 25
Trying 211.212.213.214...
Connected to mail.firma-abc.de
Escape character is '^]'.
220 firma-abc.de ESMTP Postfix (Ubuntu)
EHLO firma-abc.de
250-firma-abc.de
250-PIPELINING
250-SIZE 50480000
250-ETRN
250-STARTTLS
250-AUTH PLAIN LOGIN
...
quit
Connection closed by foreign host.
```

Weitere Tipps, wie Sie mit telnet, nc (Netcat), openssl und anderen Kommandos die SMTP-Kommunikation inklusive der Verschlüsselung testen können, finden Sie auf der folgenden Seite:

<https://www.heise.de/security/artikel/StartTLS-785453.html>



# Kapitel 35

## Nextcloud

Nextcloud ermöglicht es, Dateien, Termine und Kontakte auf einem eigenen Server im Internet abzulegen und diese Daten über alle erdenklichen Client-Geräte hinweg zu synchronisieren. Unterstützt werden unter anderem Rechner unter Linux, Windows und macOS sowie Smartphones und Tablets unter Android und iOS. Damit bietet Nextcloud auf den ersten Blick ähnliche Funktionen wie Dropbox.

Demgegenüber hat Nextcloud einige wesentliche Vorteile:

- ▶ Sie geben die Daten nicht aus der Hand, sondern speichern sie auf Ihrem eigenen Server. Wer die Datensammelwut von Google & Co. sowie der Geheimdienste kennt, wird das zu schätzen wissen.
- ▶ Sofern Sie ohnedies einen Root-Server betreiben, fallen unabhängig von der Datenmenge (fast) keine Kosten an – auf jeden Fall viel weniger als bei Dropbox oder iCloud.

Nextcloud bietet sich damit als Komplettlösung zur Speicherung und Synchronisation nahezu aller gängigen Daten zwischen mehreren Computern und iOS- oder Android-Geräten an – sowohl für Einzelpersonen als auch für ganze Unternehmen, Schulen oder andere Organisationen.

Im Vergleich zu kommerziellen Cloud-Diensten ist die Client-Integration nicht so perfekt. Das gilt insbesondere auf Smartphones: Zwar erleichtert eine App den Zugriff auf Dateien, aber das ersetzt nicht die viel tiefergehende Integration der iCloud in unzählige iOS-Apps bzw. den direkten Zugang zu Google-Drive-Dateien in vielen Android-Apps.

**Nachteile**

Auf dem Server hat Nextcloud im Vergleich zu kommerziellen Angeboten den Nachteil, dass Sie sich selbst um Updates kümmern müssen. Anders als bei grundlegenden Server-Programmen wie Apache oder Postfix, die als Teil der Linux-Distribution weitgehend automatisch aktualisiert werden, ist hier Handarbeit notwendig. Die vergangenen Jahre haben gezeigt, dass die Wartung von ownCloud- bzw. Nextcloud-Installationen vergleichsweise aufwendig ist und entsprechend oft vernachlässigt wurde.

**Varianten** Ich konzentriere mich in diesem Kapitel auf die frei verfügbare Nextcloud-Variante zur Installation auf eigenen Servern. Die Firma Nextcloud bietet ihre Software auch in kommerziellen Paketen an, wahlweise mit oder ohne Hosting.

### Nextcloud versus ownCloud

Es gehört fast zur Open-Source-Kultur dazu: Sobald ein Projekt populär wird und versucht, durch kommerzielle Zusatzfunktionen Geld zu verdienen, beginnen Konflikte zwischen den Entwicklern und dem Management. Genau das ist Mitte 2016 bei ownCloud passiert. Es kam zu einem »Fork«, also zu einer Aufspaltung des Quellcodes in zwei Zweige, und ein Teil der ownCloud-Gründungsmitglieder wechselte in das Nextcloud-Lager. Nextcloud schreibt auf seine Fahnen, Open-Source-freundlicher zu agieren und einige Funktionen kostenlos anzubieten, die ownCloud nur zahlenden Kunden gibt.

Tatsächlich sind die funktionellen Unterschiede zwischen ownCloud und Nextcloud aktuell nicht groß. Die Entscheidung für eines der beiden Programme ist mir insofern schwer gefallen – und sie ist nicht als Wertung zu sehen: Aktuell machen Sie mit keinem der beiden Programme etwas verkehrt. Welches Programm sich längerfristig durchsetzen wird, welche Firma genug Geld für den anhaltenden Betrieb verdienen kann, steht in den Sternen.

## 35.1 Installation

**Voraussetzungen** Bevor Sie Nextcloud installieren können, müssen Sie einen Webserver samt HTTPS einrichten. (HTTPS ist keine zwingende Voraussetzung für den Nextcloud-Betrieb, aus Sicherheitsgründen aber unbedingt zu empfehlen.) Ich gehe in diesem Kapitel davon aus, dass Sie als Webserver Apache verwenden. Alternativ ist auch Nginx geeignet.

**PHP** Große Teile von Nextcloud sind in PHP programmiert. Neben PHP selbst (idealerweise in Version 7 oder höher) müssen eine Menge PHP-Erweiterungsmodule installiert sein, unter anderem `php-gd`, `php-json`, `php-mysql`, `php-curl`, `php-mbstring`, `php-intl`, `php-mcrypt`, `php-imagick`, `php-xml` und `php-zip`. Bei den meisten Distributionen befinden sich die Module in gleichnamigen Paketen oder in Paketen mit Namen wie `php7.0-xxx`. Mitunter sind einige Module direkt in das PHP-Basispaket integriert und deswegen nicht extra verfügbar.

**MySQL/MariaDB** Die Nextcloud-Entwickler empfehlen MySQL oder MariaDB als Datenbanksystem. (Alternativ wird PostgreSQL und SQLite unterstützt, für Unternehmenskunden auch Oracle.) Bevor Sie die Nextcloud-Installation beginnen, müssen Sie sowohl eine Datenbank als auch einen Datenbank-Benutzer mit Zugriffsrechten auf die neue

Datenbank einrichten. Sofern MySQL oder MariaDB bereits installiert ist, sieht die Vorgehensweise so aus:

```
root# mysql -u root -p
mysql> CREATE DATABASE nextdb;
mysql> GRANT ALL ON nextdb.* TO nextuser@localhost IDENTIFIED BY 'geheim';
```

## Installationsvarianten

Viele Wege führen nach Rom, beinahe ebenso viele zu einer Nextcloud-Installation. Na ja, vier Varianten stehen auf jeden Fall zur Auswahl:

- ▶ **Paketinstallation aus inoffiziellen Quellen:** Aktuell bieten die großen Linux-Distributionen keine fertigen Nextcloud-Pakete an. Es gibt aber eine ganze Reihe inoffizieller Paketquellen. An sich wären solche Pakete eine feine Sache – man installiert sie einmal, und die Paketverwaltung kümmert sich in der Folge um alle Updates. Leider weiß man nie, wie lange solche Paketquellen gepflegt werden. Eine gute Übersicht über Nextcloud-Paketquellen für diverse Distributionen finden Sie hier:

<https://help.nextcloud.com/t/linux-packages-status/10216>

- ▶ **Snap-Paket:** Für Ubuntu und andere Distributionen, auf denen die Snap-Infrastruktur installiert ist, gibt es ein fertiges Snap-Paket. Als ich diese Variante Ende Juni 2017 ausprobiert habe, enthielt das Snap-Paket allerdings noch Version 11.0.3, während Nextcloud bereits seit sechs Wochen in Version 12 verfügbar war – so viel zum Thema *ohne Wartungsaufwand immer aktuell* ...

Das Snap-Paket umfasst sowohl Apache als auch MySQL, die oben skizzierten Anforderungen sind also hinfällig. Allerdings darf auf dem Rechner keine normale Instanz des Webservers laufen, sonst gibt es einen Konflikt der beiden Instanzen. Nextcloud ist über die Adresse <http://hostname> erreichbar, verwendet also kein Unterverzeichnis. Mit `sudo nextcloud.enable-https` kann die Seite mit Let's-Encrypt-Zertifikaten auch verschlüsselt eingerichtet werden. Viel mehr Konfigurationsmöglichkeiten gibt es allerdings nicht. Weitere Details können Sie hier nachlesen:

<https://github.com/nextcloud/nextcloud-snap>

- ▶ **Docker:** Wenn Sie sich mit Docker angefreundet haben (siehe [Kapitel 41](#)), können Sie Nextcloud als Docker-Container ausführen. Eine gute Beschreibung der Eckdaten des Containers sowie seiner Konfigurationsmöglichkeiten finden Sie hier:

<https://github.com/nextcloud/docker>

- ▶ **Manuelle Installation:** Die größte Flexibilität bietet immer noch eine manuelle Konfiguration. Sie ist vor allem dann sinnvoll, wenn Sie aktuell nur *eine* Nextcloud-Instanz brauchen und nicht vorhaben, Dutzende Installationen durchzuführen.

## Manuelle Installation

Die Testinstallationen für dieses Buch habe ich unter Ubuntu 16.04 durchgeführt. Die erforderlichen Voraussetzungen, soweit nicht schon vorhanden, installieren Sie so:

```
root# apt install apache2 php libapache2-mod-php mysql-server \
      php-zip php-mbstring php-gd php-mysql php-mbstring php-intl \
      php-xml php-curl php-imagick
```

Unter Ubuntu sind die Apache-Module `env`, `dir` und `mime` standardmäßig aktiv. Zwei weitere werden benötigt:

```
root# a2enmod rewrite
root# a2enmod headers
root# systemctl restart apache2
```

Die Nextcloud-Dateien laden Sie mit `wget` von der Nextcloud-Seite herunter und packen sie aus. Wenn Sie eine andere Distribution als Ubuntu oder Debian verwenden, müssen Sie sowohl den Installationspfad als auch den Besitzer der Dateien anpassen (siehe [Tabelle 32.1](#) auf Seite 1075).

```
root# cd /var/www/html
root# wget https://download.nextcloud.com/server/releases/nextcloud-n.n.tar.bz2
root# tar xjf nextcloud-n.n.tar.bz2
root# chown -R www-data:www-data nextcloud/
```

Die folgenden beiden Kommandos sind optional. Sie stellen die Zugriffsbits aller Dateien korrekt ein:

```
root# cd /var/www/html
root# find nextcloud/ -type d -exec chmod 750 {} \;
root# find nextcloud/ -type f -exec chmod 640 {} \;
```

### Apache-Konfiguration

Grundsätzlich sollte es nun möglich sein, in einem Webbrowser Nextcloud unter der Adresse `https://firma-abc.de/nextcloud` in Betrieb zu nehmen. Wenn Sie HTTPS nicht konfiguriert haben, funktioniert natürlich auch eine entsprechende `http`-Adresse.

Wenn Sie Nextcloud unter einer anderen Adresse ansprechen möchten oder andere verzeichnisspezifische Einstellungen vornehmen wollen, fügen Sie einen Directory-Block gemäß dem folgenden Muster aus der Nextcloud-Dokumentation in die entsprechende Apache-Konfigurationsdatei ein (unter Ubuntu z. B. in `/etc/apache2/sites-available/default-ssl.conf`).

```
# z.B. in der Datei /etc/apache2/sites-available/default-sss.conf
<IfModule mod_ssl.c>
  <VirtualHost _default_:443>
    ...
    Alias /mycloud /var/www/html/nextcloud
```

```

<Directory /var/www/html/nextcloud/>
  Options          +FollowSymLinks
  AllowOverride    All
  Require all granted
  <IfModule mod_dav.c>
    Dav off
  </IfModule>
  SetEnv HOME /var/www/html/nextcloud
  SetEnv HTTP_HOME /var/www/html/nextcloud
</Directory>
...
# für iOS
Redirect 301 /.well-known/carddav /mycloud/remote.php/dav
Redirect 301 /.well-known/caldav /mycloud/remote.php/dav
...

```

Falls Sie vorhaben, Nextcloud auch zum Speichern von Kontakten und Terminen einzusetzen, sollten Sie in der Apache-Konfiguration die obigen beiden `Redirect`-Zeilen einbauen, wobei Sie `mycloud` durch Ihren Alias-Namen ersetzen (oder einfach `nextcloud`, wenn Sie keinen Alias verwenden). Manche Programme, insbesondere das Programm *Einstellungen* unter iOS, tun sich dann bei der CalDAV- oder CardDAV-Konfiguration leichter. Beachten Sie, dass die `Redirect`-Zeilen außerhalb des `Directory`-Blocks stehen müssen! Eine Erklärung zur Funktionsweise dieser `.well-known`-Umleitungen finden Sie hier:

<http://sabre.io/dav/service-discovery>

Beim ersten Aufruf der Nextcloud-Startseite sollte ein Formular zur Inbetriebnahme erscheinen (siehe [Abbildung 35.1](#)). Häufig kommt es vor, dass stattdessen ein Hinweis auf noch fehlende PHP-Module angezeigt wird – dann installieren Sie diese Module, starten Apache neu und versuchen es nochmals.

**Inbetriebnahme**

Zur Inbetriebnahme ist nicht viel zu tun: Sie geben den Account-Namen des Cloud-Administrators und ein möglichst sicheres Passwort ein. Empfehlenswert ist es, den Admin-Account von Ihrem privaten Daten-Account zu trennen und daher *nicht* den eigenen Namen zu verwenden.

Außerdem müssen Sie sich für ein Datenbank-Backend entscheiden. Je nachdem, welche PHP-Module installiert sind, stehen zumeist SQLite und MySQL/MariaDB zur Wahl. MySQL/MariaDB ist schneller und kommt auch mit vielen gleichzeitigen Zugriffen zurecht. Sie müssen in diesem Fall drei Parameter angeben: den Namen der leeren Datenbank, die Sie zuvor eingerichtet haben, den MySQL-Benutzer, der Zugriff auf diese Datenbank hat, und das zugehörige Passwort. Fertig!

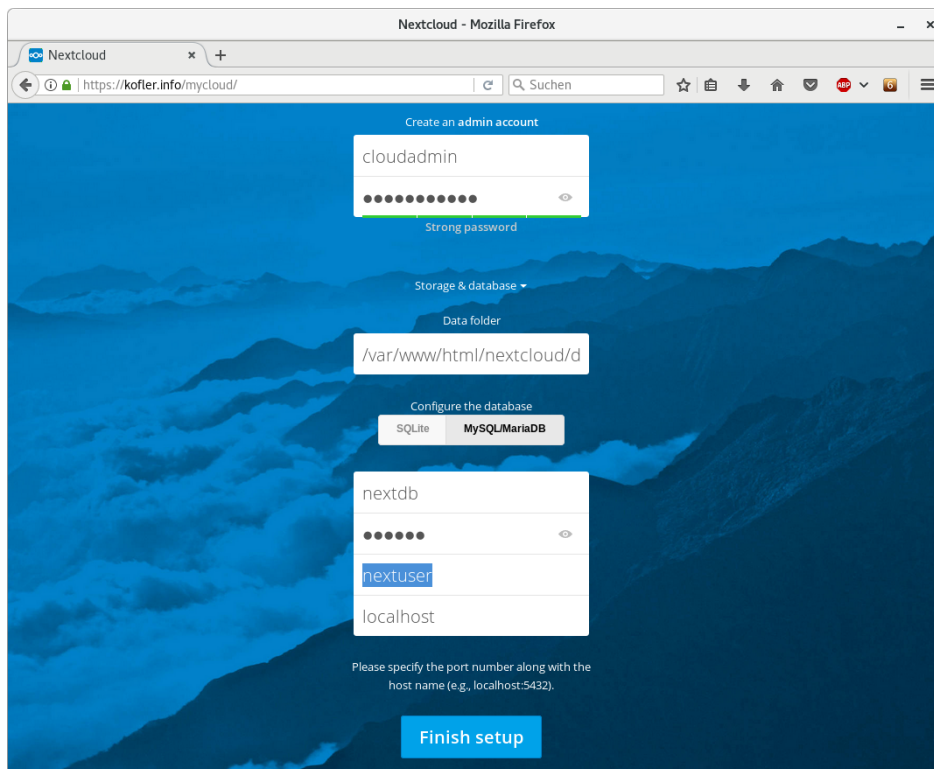


Abbildung 35.1 Nextcloud-Inbetriebnahme im Webbrowser

## Benutzerverwaltung

So wie Sie unter Linux nur in Ausnahmefällen als `root` arbeiten, werden Sie auch bei ownCloud das Admin-Konto nicht zur Speicherung gewöhnlicher Daten verwenden. Sie sollten deswegen nach dem ersten Administrator-Login im Einstellungspunkt **BENUTZER** im Menü rechts oben (siehe [Abbildung 35.2](#)) einen gewöhnlichen Benutzer einrichten. An dieser Stelle können Sie außer Ihrem eigenen Konto auch die Konten für Familienmitglieder, Firmenmitarbeiter etc. anlegen.

Im Dialog zur Benutzerverwaltung können Sie ein Standard-Quota festlegen, z. B. 5 GiB. Damit geben Sie den maximalen Speicherplatz vor, den ein Benutzer beanspruchen darf. Davon abweichend können Sie die maximale Speichermenge individuell für jeden Benutzer einstellen. Leider verrät ownCloud nicht, wie viel Speicherplatz jeder Benutzer tatsächlich beansprucht. Eine Liste aller Datenverzeichnisse mit dem Platzbedarf in MiB liefert das folgende Kommando:

```
root# du -m --max 1 /var/www/html/nextcloud/data | sort -rn
```

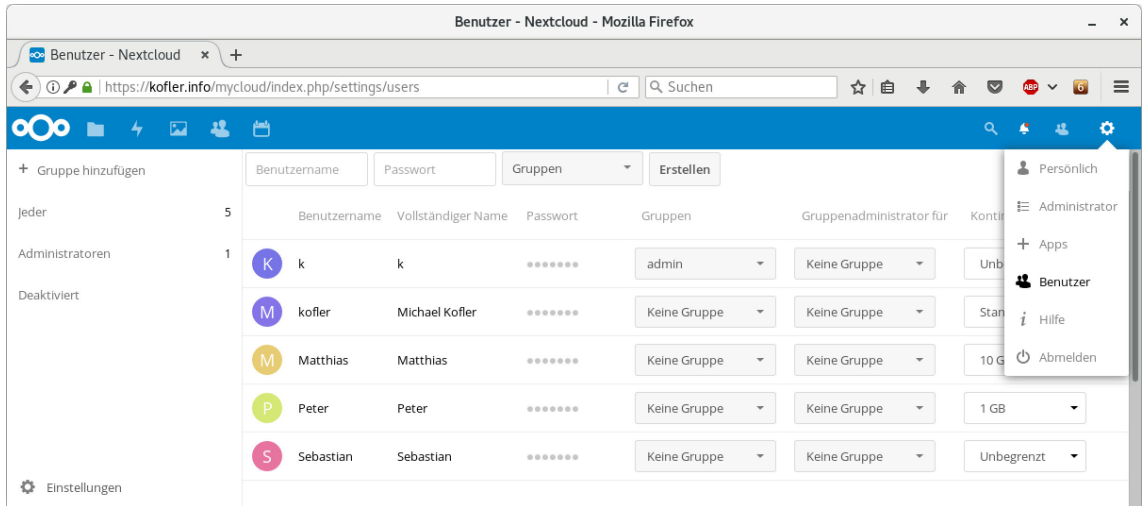


Abbildung 35.2 Nextcloud-Benutzerverwaltung in der Weboberfläche

Nextcloud gibt Ihnen auch die Möglichkeit, den Benutzern Gruppen zuzuordnen. Standardmäßig gibt es nur die Gruppe *Administratoren* bzw. keine Gruppenzugehörigkeit (*Jeder*).

Um einen Benutzer vorübergehend zu deaktivieren oder inklusive aller Daten zu löschen, führen Sie das entsprechende Kommando über einen Menübutton aus, der ganz rechts in der Zeile des Benutzers erscheint, sobald Sie den Mauszeiger dorthin bewegen. Nach dem Löschen haben Sie für einige Sekunden eine Undo-Möglichkeit, danach ist der Vorgang endgültig.

Jeder Benutzer kann nach dem Login in der Nextcloud-Weboberfläche seine persönlichen Einstellungen inklusive des Passworts ändern. Dazu wählen Sie im Login-Menü den Eintrag **PERSÖNLICH**. Hier können Sie auch die gewünschte Sprache einstellen. Empfehlenswert ist schließlich die Angabe einer E-Mail-Adresse, an die gegebenenfalls ein Login-Link versandt wird, falls Sie Ihr Passwort vergessen sollten. Diese Funktion setzt voraus, dass auf dem Nextcloud-Server ein Mail-Server zur Verfügung steht.

## Einstellungen

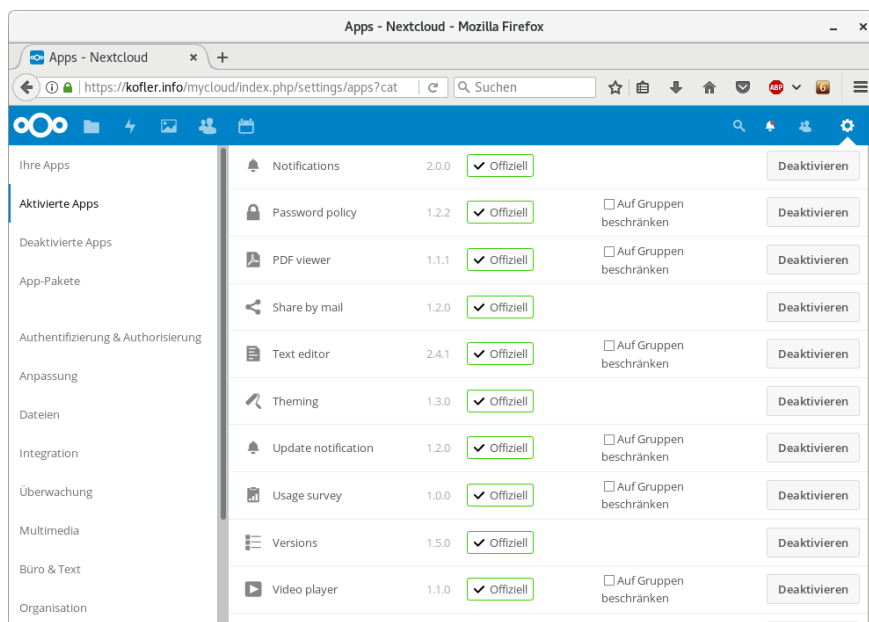
Bei Benutzern mit Administratorrechten führt der Eintrag **ADMINISTRATOR** im Menü rechts oben auf eine Seite mit diversen Grundeinstellungen. Hier können Sie z. B. die Aktivitäten Ihrer Nextcloud-Instanz verfolgen, die maximale Dateigröße verändern, das optische Erscheinungsbild modifizieren (**THEMING**) etc.

Benutzer löschen

Benutzer-einstellungen

Grund-einstellungen

**Apps** Im Menü führt der Eintrag APPS in eine Liste von Zusatzprogrammen (Plugins), die in der Nextcloud-Instanz zur Verfügung stehen (siehe [Abbildung 35.3](#)). Diese können bei Bedarf aktiviert oder deaktiviert werden. Standardmäßig aktiviert sind lediglich zwei Viewer für PDF-Dateien und Bilder und ein simpler Texteditor. Die Apps zur Kontakt- und Terminverwaltung müssen Sie nun explizit installieren (CALENDAR, CONTACTS).



**Abbildung 35.3** App-Verwaltung in Nextcloud

**Verschlüsselung** Aus Sicherheitsgründen ist es empfehlenswert, alle Daten ausschließlich verschlüsselt zwischen Ihren Clients und Nextcloud zu übertragen. Dazu müssen Sie Nextcloud auf einer HTTPS-Webseite einrichten.

Wünschen Sie darüber hinaus auch, dass alle Dateien innerhalb der Nextcloud-Verzeichnisse verschlüsselt gespeichert werden, müssen Sie die Option `SERVERSEITIGE VERSCHLÜSSELUNG` aktivieren. Das ist insbesondere dann zu empfehlen, wenn Nextcloud so konfiguriert ist, dass es auf einen externen Cloud-Speicher zurückgreift (z. B. auf Amazon S3). Intern zur Verschlüsselungsstrategie lesen Sie hier nach:

[https://docs.nextcloud.com/server/12/user\\_manual/files/encrypting\\_files.html](https://docs.nextcloud.com/server/12/user_manual/files/encrypting_files.html)

**Versionen** Standardmäßig erstellt Nextcloud Backups aller Dateien, die geändert oder gelöscht werden. Verantwortlich ist dafür die App `VERSIONS`, die standardmäßig aktiviert ist. Nextcloud-Anwender können also wie unter Dropbox bei Bedarf auch auf ältere Versionen von synchronisierten Dateien zurückgreifen. Der Zugriff auf die Backup-



Versionen von Dateien erfolgt in der Weboberfläche bei der jeweiligen Datei mit dem Link `VERSIONEN`.

Nextcloud speichert regelmäßig alte Versionen von geänderten Dateien, wobei alte Sicherheitskopien nach und nach gelöscht werden, sodass nie mehr als ca. 50 Prozent des freien Platzes für Backups verwendet werden:

[https://docs.nextcloud.com/server/12/user\\_manual/files/version\\_control.html](https://docs.nextcloud.com/server/12/user_manual/files/version_control.html)

## 35.2 Wartung

Tabelle 35.1 fasst die Speicherorte von Nextcloud-Daten relativ zum Nextcloud-Installationsverzeichnis zusammen. Wenn Sie der obigen Anleitung gefolgt sind, hat dieses Verzeichnis den Pfad `/var/www/html/nextcloud`. Interna

| Pfad und Dateiname                           | Inhalt                                 |
|--|--|
| <code>data/</code>                           | alle Daten außer der Datenbank         |
| <code>data/nextcloud.log</code>              | Logging (Warnungen, Fehler)            |
| <code>data/loginname/files/*</code>          | Dateien eines Nextcloud-Benutzers      |
| <code>data/loginname/files_versions/*</code> | Backups alter Dateien (Versioning-App) |
| <code>data/loginname/files_trash/*</code>    | Mülleimer                              |

**Tabelle 35.1** Nextcloud-Speicherorte relativ zum Installationsverzeichnis

Alle Metadaten sowie Kontakte, Termine und sonstige Zusatzdaten von Apps werden in der MySQL- oder MariaDB-Datenbank gespeichert, die Sie bei der Inbetriebnahme angegeben haben. Der Aufbau der Datenbank ist einigermaßen komplex. Sie besteht standardmäßig aus über 40 Tabellen. Je nachdem, welche Apps Sie installieren, kommen weitere Tabellen hinzu. Manuelle Änderungen sind nicht zu empfehlen.

### Backups

Grundsätzlich sollten Sie regelmäßig und insbesondere vor jedem Update ein Backup durchführen. Das ist nicht weiter schwierig. Sie benötigen eine vollständige Sicherung aller Dateien im Nextcloud-Installationsverzeichnis sowie in der dazugehörigen MySQL/MariaDB-Datenbank. Die mit Nextcloud gesicherten Dateien befinden sich relativ zum Installationsverzeichnis im Unterverzeichnis `data`. Für die in dieser Anleitung verwendeten Pfade und Datenbanknamen könnten Sie ein Backup z. B. so durchführen:

```

root# mkdir /mybackup
root# sudo -u www-data php occ maintenance:mode --on
root# cp -a /var/www/html/nextcloud /mybackup/ncbackup-$(date +"%Y-%m-%d")
root# mysqldump -u root -p --single-transaction nextdb | \
    gzip -c > /mybackup/ncdb-$(date +"%Y-%m-%d").sql.gz
root# sudo -u www-data php occ maintenance:mode --off

```

Die Aktivierung des Wartungsmodus ist nicht zwingend erforderlich. Sie stellt sicher, dass sich die Nextcloud-Daten während des Backups nicht ändern und somit Dateien und der Inhalt der Datenbank synchron zueinander sind. Eleganter, d. h. ohne Downtime, können Sie dieses Ziel mit LVM-Snapshots erreichen (siehe auch [Abschnitt 33.3](#), »Backups«, und [Abschnitt 36.7](#), »Backup-Scripts«). Bei großen Nextcloud-Installationen sollten Sie zudem versuchen, das Backup von `/var/www/html/nextcloud` inkrementell durchzuführen (z. B. mit `rsync`). Weitere Backup-Empfehlungen und -Beispiele finden Sie wiederum im Handbuch:

[https://docs.nextcloud.com/server/12/admin\\_manual/maintenance/backup.html](https://docs.nextcloud.com/server/12/admin_manual/maintenance/backup.html)

## Updates

Die Nextcloud-Weboberfläche weist Sie unübersehbar auf mögliche Updates hin. Wenn Sie die Nextcloud allerdings über diverse Client-Tools nutzen, besteht die Gefahr, dass Sie diese Hinweise nie sehen.

Zur Durchführung eines Updates gibt es zwei Möglichkeiten: Die einfachere Variante besteht darin, dass Sie sich als Administrator in der Weboberfläche anmelden und dann die Seite `ADMINISTRATOR • GRUNDEINSTELLUNGEN` besuchen. Dort können Sie das Update direkt in der Weboberfläche initiieren. Beachten Sie, dass Nextcloud während des Updates in einen Wartungsmodus versetzt wird und in dieser Zeit für alle Benutzer nicht erreichbar ist.

Die andere Variante erfordert einen SSH-Login auf Ihren Server. Danach wechseln Sie in das Nextcloud-Installationsverzeichnis und führen das Update mit dem PHP-Script `occ` durch. Beachten Sie, dass das folgende `sudo`-Kommando *nicht* dazu dient, das Update im `root`-Modus auszuführen. Vielmehr muss das Script `occ` mit den Rechten des Accounts ausgeführt werden, in dem auch der Webserver läuft – unter Debian und Ubuntu also `www-data`. Würden Sie `occ` mit `root`-Rechten ausführen, könnte Apache später auf die nun ebenfalls mit `root`-Rechten vorliegenden Dateien nicht mehr zugreifen.

```

root# cd /var/www/html/nextcloud
root# sudo -u www-data php occ upgrade

```

Mehr Details zu den verschiedenen Update-Verfahren sind im Nextcloud-Handbuch dokumentiert. Dort finden Sie auch Tipps, wie Sie vorübergehend den Wartungsmodus aktivieren, wie Sie bei Bedarf die Zugriffsrechte der Dateien korrigieren etc.

[https://docs.nextcloud.com/server/12/admin\\_manual/maintenance/upgrade.html](https://docs.nextcloud.com/server/12/admin_manual/maintenance/upgrade.html)

## 35.3 Betrieb

Sie können Nextcloud ohne die Installation irgendwelcher Client-Werkzeuge sofort nutzen. Dazu öffnen Sie in einem Webbrowser die Nextcloud-Seite und loggen sich mit Ihrem Benutzernamen und Passwort ein. Anschließend können Sie Dateien direkt per Drag & Drop hochladen. Für ganze Verzeichnisse funktioniert dies aber nicht. Neue Verzeichnisse richten Sie mit dem Plus-Button ein. In der Detailansicht zu einer Datei (siehe [Abbildung 35.4](#)) können Sie die letzten Aktivitäten verfolgen, ältere Versionen der Datei wiederherstellen, die Datei mit anderen Benutzern teilen bzw. einen Freigabe-Link einrichten, der auch ohne Nextcloud-Login zugänglich ist.

Weboberfläche

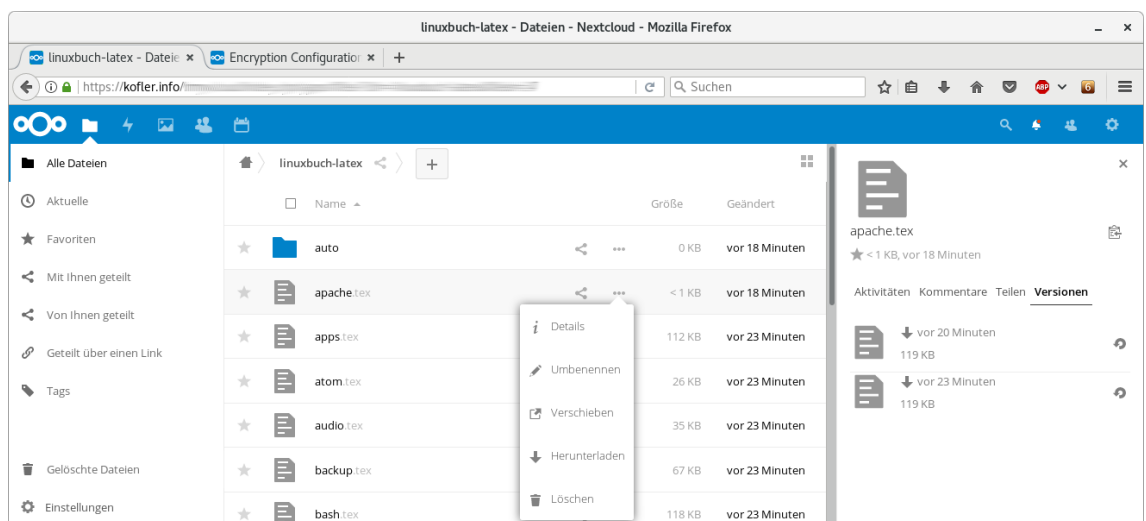


Abbildung 35.4 Dateien in der Nextcloud-Benutzeroberfläche verwalten

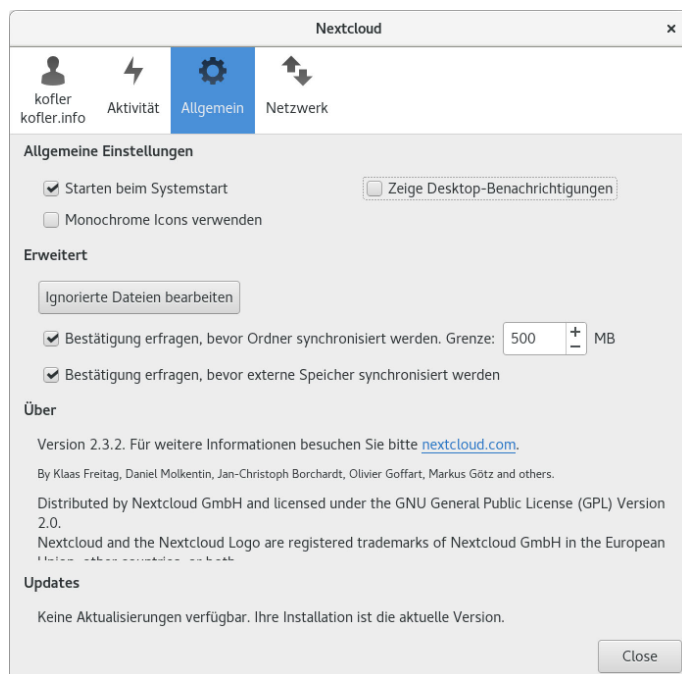
Manche Distributionen stellen den Nextcloud-Client im Paket `nextcloud-client` zur Verfügung. Fallweise befindet sich die entsprechende Erweiterung für den Gnome-Dateimanager im eigenen Paket `nextcloud-client-nautilus`. Debian- und Ubuntu-Nutzer müssen aktuell eine zusätzliche Paketquelle bzw. ein PPA einrichten. Die unkomplizierte Vorgehensweise ist hier dokumentiert:

Nextcloud-Client

[https://github.com/nextcloud/client\\_theming](https://github.com/nextcloud/client_theming)

Die Installation erfolgt anschließend unkompliziert mit `apt`, `dnf`, `yum` oder `zypper`, der Start über das Startmenü Ihrer Distribution unter dem sperrigen Namen *Nextcloud Desktop-Synchronisationsclient*.

Beim ersten Start geben Sie die HTTPS-Adresse Ihres Nextcloud-Servers sowie den Login-Namen und das Passwort an. Der Installationsassistent richtet automatisch das Verzeichnis Nextcloud ein und synchronisiert alle darin enthaltenen Dateien. Im Einstellungsmenü können Sie einzelne Verzeichnisse oder Dateikennungen von der Synchronisation ausnehmen und den Status der Synchronisation verfolgen (siehe [Abbildung 35.5](#)).



**Abbildung 35.5** Der Nextcloud-Client für Linux

**Android, iOS,  
macOS, Windows**

Der Nextcloud-Client kann mit den gleichen Funktionen auch unter Windows und macOS installiert werden. Downloads finden Sie auf der Nextcloud-Website. Für iOS und Android gibt es im App Store bzw. Play Store Apps, die den Zugriff auf die Dateien unterwegs ermöglichen.

**WebDAV**

Anstelle der Client-Apps können Sie zur Verwaltung der Dateien auch jedes Programm verwenden, das das WebDAV-Protokoll unterstützt. Unter Linux zählen dazu z. B. die Dateimanager von Gnome und KDE, also Nautilus und Dolphin. In Nautilus klicken Sie in der Seitenleiste auf den Eintrag MIT SERVER VERBINDEN oder drücken

`Strg` + `L` und geben dann eine Adresse nach dem folgenden Muster an, wobei Sie nextcloud durch den Pfad ersetzen, der für Ihre Installation gilt:

davs://loginname@hostname/nextcloud/remote.php/webdav (HTTPS)

dav://loginname@hostname/nextcloud/remote.php/webdav (HTTP, nicht empfehlenswert)

## 35.4 Kontakte und Termine

Mit Apps können Sie die Funktionalität von Nextcloud beinahe grenzenlos erweitern. Ich konzentriere mich hier auf zwei Apps, die besonders praktisch sind: *Calendar* zur Verwaltung von Terminen und *Contacts* zur Verwaltung von Adressen, Telefonnummern und E-Mail-Adressen.

### Wirklich empfehlenswert?

Solange Sie die Nextcloud-Weboberfläche verwenden (siehe [Abbildung 35.6](#)) bzw. unter Gnome die Programme *Kontakte* und *Kalender* ausführen, funktioniert das Speichern und Synchronisieren der Daten wunderbar. In der Regel wollen Sie auf diese Daten aber auch auf Ihrem Smartphone, Tablet oder in Ihrem E-Mail-Client zugreifen – und da wird die Sache hakelig. Ich habe Stunden verbracht, bis mir die Synchronisation auf meinem Fuhrpark von Geräten und Testinstallationen gelungen ist. Ähnliche Erfahrungen habe ich in der Vergangenheit auch schon im Zusammenspiel mit ownCloud gemacht.

Soweit ich es beurteilen kann, sind nicht die Next- bzw. ownCloud-Entwickler an dieser Misere schuld. Vielmehr haben Apple und Google größtes Interesse an Ihren Daten. Die Motivation von Apple oder Google, externe Anbieter zu unterstützen, ist hingegen nicht übermäßig ausgeprägt. Irreführende Fehlermeldungen (iOS), keine Unterstützung der offenen Standards CardDAV und CalDAV (Android), unübersichtliche Konfigurationsdialoge (iOS und Android) machen die Konfiguration je nach Blickwinkel zu einem Abenteuerspiel oder zu einer Geduldsprobe für IT-Profis.

Das Internet ist voll von Berichten frustrierter Next-/ownCloud-Anwender und von mehr oder weniger zielführenden Lösungsvorschlägen. Auch die hier präsentierten Lösungswege funktionieren womöglich nach dem nächsten Update nicht mehr. Voraussichtlich machen Sie sich nicht beliebt, wenn Sie diese Lösung Ihren Familienmitgliedern bzw. den Mitarbeitern Ihrer Firma/Organisation aufzwingen ...

Der erste Schritt ist der einfachste: Sie melden sich als Administrator bei Nextcloud an und aktivieren die beiden offiziellen Apps *Calendar* und *Contacts*. Ab sofort erscheinen in der Weboberfläche für alle Benutzer zwei neue Icons, die in die Kontakt- oder Terminverwaltung führen. Die Bedienung dieser Module ist weitgehend selbst-erklärend – vor allem, wenn Sie schon die vergleichbaren Oberflächen von Google kennen.

**Apps einrichten**

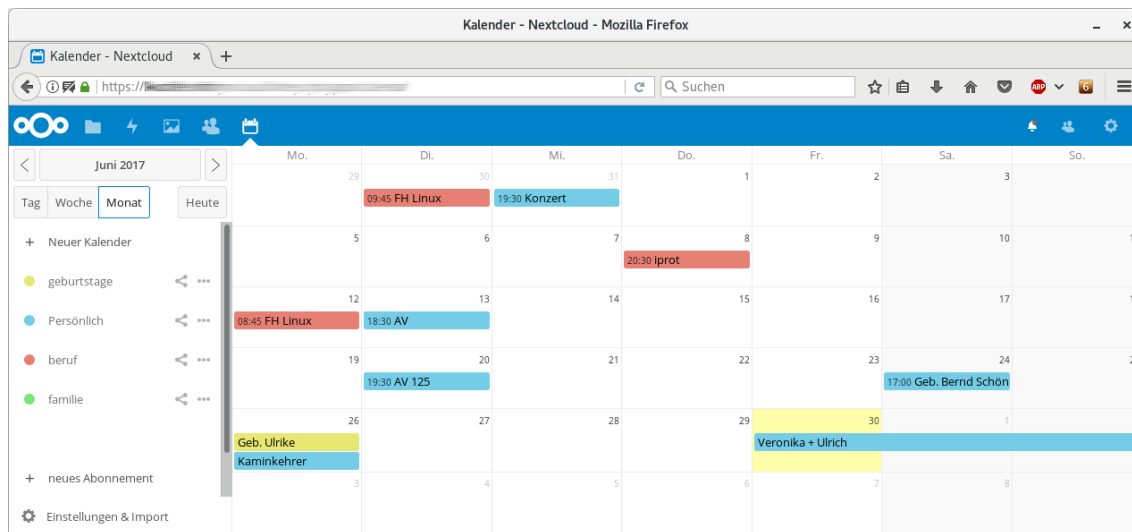


Abbildung 35.6 Terminverwaltung in der Nextcloud-Weboberfläche

Fantastisch ist auch die Integration unter Gnome: Dort richten Sie einfach im Modul ONLINE-KONTEN Ihr Nextcloud-Konto ein. Ab sofort werden Sie in der Benachrichtigungsansicht an fällige Termine erinnert. Außerdem können Sie in den Programmen *Kontakte* und *Kalender* Ihre Daten bearbeiten (siehe [Abbildung 5.8](#) auf Seite 165).

**Externe Clients** Wenn Sie Ihr Adressbuchprogramm, Ihren E-Mail-Client oder andere Programme mit Nextcloud-Kontakten und -Terminen synchronisieren möchten, müssen Sie zuerst einmal feststellen, ob das Programm die Protokolle CardDAV oder CalDAV unterstützt. Bei vielen Programmen ist das der Fall, die Konfiguration kann aber dennoch mühsam sein. Die folgenden Abschnitte gehen auf einige Sonderfälle ein.

**macOS** Um unter macOS auf die Nextcloud-Kontakte zuzugreifen, starten Sie in den System-einstellungen das Modul INTERNET-ACCOUNTS und fügen einen neuen Account hinzu, wobei Sie den Typ ANDEREN ACCOUNT • CARDDAV-ACCOUNT wählen. Anschließend müssen Sie sich für den Account-Typ ERWEITERT entscheiden und als Server-Adresse den folgenden Pfad angeben:

```
/mycloud/remote.php/dav/principals/users/kofler
```

Dabei ersetzen Sie `mycloud` durch den Pfad, über den die Nextcloud-Installation auf Ihrem Server zugänglich ist (oft `nextcloud`), und `kofler` durch den Benutzernamen (siehe [Abbildung 35.7](#)). Auch die explizite Angabe der Portnummer ist erforderlich; macOS ist nicht in der Lage zu erraten, dass HTTPS üblicherweise den Port 443 verwendet.

Abbildung 35.7 Einrichten eines CardDAV-Kontos unter macOS

Absurderweise funktionierten exakt dieselben Angaben beim Einrichten eines CalDAV-Kontos nicht mehr. Dort führte etwas überraschend die simple Konfiguration des Accounttyps AUTOMATISCH und `username@hostname` zum Ziel. Das setzt voraus, dass Sie in der Webserver-Konfiguration die `Redirect`-Zeilen für die `.well-known`-Adressen eingebaut haben (siehe [Abschnitt 35.1](#), »Installation«).

Sofern auf dem Webserver die vorhin erwähnten `.well-known`-Umleitungen eingerichtet sind, gelingt die Konfiguration unter iOS erstaunlich einfach. (Ohne die Umleitungen ist mir die Konfiguration dagegen gar nicht gelungen. iOS behauptete, keine Verbindung herstellen zu können und bot an, HTTP statt HTTPS zu verwenden – womit ich aber nicht einverstanden war. Diese vermeintliche Hilfe ist insofern irreführend, weil sie nahelegt, dass SSL-Probleme vorlagen – was gar nicht der Fall war.)

Zur Konfiguration starten Sie die App EINSTELLUNGEN, suchen nach den Punkten KONTAKTE bzw. KALENDER und führen ACCOUNT HINZUFÜGEN AUS. Im folgenden Dialog geben Sie die folgenden Daten an:

- ▶ Server: nur den Hostnamen (ohne `https://`, ohne den Pfad `mycloud`, `nextcloud` etc.)
- ▶ Benutzername: den Nextcloud-Benutzernamen
- ▶ Passwort: das Passwort
- ▶ Beschreibung: eine beliebige Zeichenkette zur Beschreibung der Datenquelle, z. B. `nextcloud`

Während die CalDAV/CardDAV-Konfiguration in iOS und macOS extrem hakelig ist, hat sich Google gleich gar nicht die Mühe gemacht, diese Protokolle überhaupt zu unterstützen. Sie sind daher auf Apps angewiesen, die diese Protokolle hinzufügen und dann als Schnittstelle zu den anderen Apps wie *Kalender* oder *Kontakte* dienen.

Gut bewährt haben sich bei mir die Apps *CalDAV-Sync* und *CardDAV-Sync*. Neuere Alternativen, die ich aber nicht getestet habe, sind *OpenSync* und *DavDroid*.

#### Thunderbird-Termine

In aktuellen Versionen des E-Mail-Programms Thunderbird ist die Terminverwaltungskomponente *Lightning* integriert. Es kommt grundsätzlich mit NextCloud-Kalendern zurecht, allerdings müssen Sie – anders als bei den meisten anderen Programmen – jeden Kalender einzeln einrichten: also einen für Ihre persönlichen Termine, einen für die beruflichen Termine usw. Dazu führen Sie DATEI • NEU • KALENDER aus, wählen die Optionen IM NETZWERK und CALDAV und geben dann eine Adresse nach dem folgenden Muster an:

<https://hostname/mycloud/remote.php/dav/calendars/kofler/beruf>

Dabei ersetzen Sie den Hostnamen, *mycloud*, *kofler* und *beruf* durch den tatsächlichen Hostnamen, den Pfad der Nextcloud-Installation, Ihren Nextcloud-Account-Namen und den Namen des Kalenders. Anstatt diese Adresse fehleranfällig einzutippen, führen Sie in der Nextcloud-Weboberfläche im Minimenü neben dem gewünschten Kalender das Kommando LINK aus. Nextcloud zeigt dann die korrekte Adresse an.

#### Thunderbird-Kontakte

Noch umständlicher ist die Verwendung von Nextcloud-Kontakten in Thunderbird. Dazu müssen Sie zuerst den *SOGGo-Connector* installieren. Dieses Add-on ist in Thunderbird in der Add-on-Suche nicht zu finden. Deswegen müssen Sie es in einem Webbrowser herunterladen:

<https://sogo.nu/download.html#/frontends>

Wenn Sie Firefox als Webbrowser verwenden, müssen Sie das Add-on explizit mit ZIEL SPEICHERN UNTER herunterladen. Simples Anklicken führt dazu, dass Firefox glaubt, sie wollten die XPI-Datei in Firefox installieren.

Zur Installation führen Sie nun in Thunderbird EXTRAS • ADD-ONS • ERWEITERUNGEN aus und klicken dann auf das Werkzeug-Icon neben dem Suchfeld. Es enthält das Kommando ADD-ON AUS DATEI INSTALLIEREN. Nach einem Thunderbird-Neustart öffnen Sie mit EXTRAS • ADRESSBUCH das Adressbuchfenster. In dessen Menü führen Sie DATEI • NEU • REMOTE-ADRESSBUCH aus. Im Konfigurationsdialog müssen Sie die Adresse Ihres Adressbuchs angeben. Diese ermitteln Sie am einfachsten in der Web-oberfläche von ownCloud, indem Sie auf den Zahnrad-Button klicken und dann auf das neben dem Adressbuch angezeigte Link-Icon. Der Link hat die folgende Form:

<https://hostname/mycloud/remote.php/dav/addressbooks/users/kofler/contacts>

Abermals müssen Sie den Hostnamen, *mycloud* und *kofler* durch Ihre Daten ersetzen. Den fertigen Link finden Sie in der Nextcloud-Weboberfläche in den Einstellungen des Adressbuchs. Die Synchronisation müssen Sie per Kontextmenü explizit starten. Beim ersten Mal werden Sie aufgefordert, den Nextcloud-Login-Namen und das dazugehörige Passwort anzugeben.



TEIL VIII

# Sicherheit



# Kapitel 36

## Backups

Ich mache hier gar nicht erst den Versuch, Sie von der Notwendigkeit von Backups zu überzeugen. Vielmehr konzentriere ich mich darauf, Ihnen einige Backup-Tools vorzustellen. Welche Werkzeuge Sie in welcher Kombination einsetzen, bleibt Ihnen überlassen – dafür gibt es kein allgemeingültiges Rezept. Zu sehr hängt die optimale Backup-Strategie von der Natur der Daten, von der Art des Rechners (Desktop-PC/Notebook/Server), vom Backup-Medium (externe Festplatte, Netzwerkverzeichnis, Cloud) und vielen anderen Faktoren ab.

### 36.1 Backup-Benutzeroberflächen

Apple hat mit seiner Time Machine bewiesen, dass selbst ein Backup-Programm Begeisterung hervorrufen kann. Linux kann in dieser Hinsicht leider nicht mithalten: Das Angebot an Backup-Kommandos und -Tools ist zwar groß, aber auf das geniale, einfach zu nutzende Backup-Werkzeug warten Desktop-Anwender noch immer. In diesem Abschnitt stelle ich Ihnen die folgenden drei Programme kurz vor:

Déjà Dup: <https://launchpad.net/deja-dup>  
Grsync: <http://opbyte.it/grsync>  
Back in Time: <http://backintime.readthedocs.io>

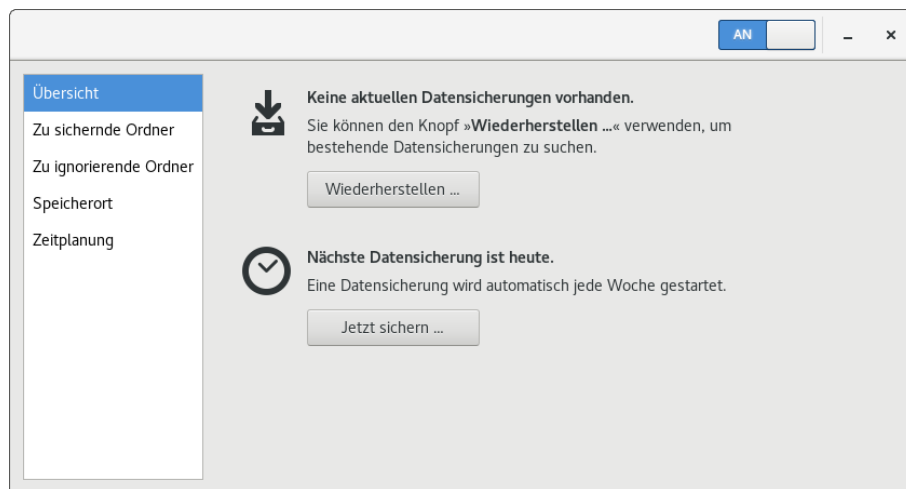
Das populärste dieser Programme ist momentan Déjà Dup: Es wird von vielen Distributionen als Default-Backup-Programm eingerichtet. Auch mit Grsync können Sie nichts falsch machen – es synchronisiert Ihre Dateien einfach in ein anderes Verzeichnis, sodass das Backup ohne Zusatzwerkzeuge gelesen werden kann.

#### Déjà Dup

Déjà Dup ist dahingehend konzipiert, das Heimatverzeichnis möglichst unkompliziert in einem lokalen oder via SSH erreichbaren Backup-Verzeichnis zu sichern. Unter Ubuntu ist Déjà Dup in die Systemeinstellungen eingebettet und kann aus diesen heraus gestartet werden. Bei anderen Distributionen mit Gnome als Desktop-

System starten Sie das Programm DATENSICHERUNG. Gegebenenfalls müssen Sie vorher die Pakete `deja-dup` und `deja-dup-nautilus` installieren.

**Konfiguration** Déjà Dup wird in mehreren Dialogblättern konfiguriert (siehe [Abbildung 36.1](#)). Im Dialogblatt ZU SICHERNDE ORDNER geben Sie an, welche Verzeichnisse vom Backup-Programm gesichert werden sollen – üblicherweise das Heimatverzeichnis. In ZU IGNORIERENDE ORDNER geben Sie die Ausnahmen an. Standardmäßig sind dort bereits der Mülleimer und das Verzeichnis Downloads vorgesehen. Oft ist es sinnvoll, außerdem Verzeichnisse mit sehr großen Dateien von den regelmäßigen Backups auszunehmen, z. B. Videos oder den Speicherort virtueller Maschinen, falls Sie VirtualBox einsetzen.



**Abbildung 36.1** Backup-Konfiguration in Déjà Dup

Im Dialogblatt SPEICHERORT geben Sie an, wo die Backups abgelegt werden sollen. Im Regelfall wählen Sie als Ort der Datensicherung den Eintrag LOKALER ORDNER und können dann ein beliebiges Verzeichnis auswählen, in dem Sie Dateien lesen und schreiben dürfen. Wenn Sie eine externe Festplatte oder einen USB-Stick nicht ausschließlich für Backups nutzen möchten, ist es zweckmäßig, dort ein eigenes Backup-Verzeichnis einzurichten. Als Backup-Ziel kann auch ein FTP-Verzeichnis, ein via SSH erreichbarer Server, ein WebDAV-Verzeichnis, ein S3-Server oder ein Netzwerkverzeichnis dienen. Wenn die Option S3 wie unter Ubuntu fehlt, müssen Sie noch das Paket `python-boto` installieren.

Sofern Sie den zentralen Ein/Aus-Schalter auf AN stellen, können Sie im Dialogblatt ZEITPLANUNG bestimmen, ob die Backups täglich oder wöchentlich ausgeführt werden sollen. Déjà Dup wird in Zukunft immer automatisch gestartet, sobald Sie sich

einloggen, und kümmert sich dann um die Backups. Falls das Backup-Medium gerade nicht erreichbar ist, verschiebt Déjà Dup das Backup auf später und startet die Sicherung automatisch, sobald die externe Festplatte erreichbar ist. Das ist ausgesprochen bequem: Sie müssen sich nur darum kümmern, dass regelmäßig ein Backup-Medium zur Verfügung steht – um alles andere kümmert sich das Backup-Programm selbst.

Außerdem können Sie hier festlegen, über welchen Zeitraum das Backup-Programm Dateien, die sich ändern, sichern soll. Die Standardeinstellung FÜR IMMER ist am sichersten, führt aber unweigerlich dazu, dass selbst das größte Backup-Medium irgendwann voll sein wird. Mit den Einstellungen MINDESTENS EIN JAHR oder MINDESTENS SECHS MONATE erreichen Sie, dass *alle* Dateien im Backup gesichert werden, wobei aber bei sich ändernden Dateien nur die Versionen der letzten sechs oder zwölf Monate aufbewahrt werden.

Das Dialogblatt ÜBERSICHT fasst die wichtigsten Einstellungen zusammen. Dort können Sie nun jederzeit eine Sicherung manuell starten (Button JETZT SICHERN). Beim ersten Backup müssen Sie außerdem angeben, ob die Backups verschlüsselt werden sollen, und falls ja, mit welchem Passwort. Beachten Sie aber, dass das Verschlüsseln eine Menge zusätzlicher CPU-Leistung erfordert!

Das erste Backup

Das erste mit Déjà Dup durchgeführte Backup dauert unverhältnismäßig lange. Das liegt daran, dass das Backup komprimiert wird. Bei weiteren Backups werden nur noch die Änderungen gespeichert, was den Vorgang stark beschleunigt.

Mit dem Button WIEDERHERSTELLUNG stellen Sie ein vollständiges Backup wieder her. Dabei können Sie die gewünschte Backup-Version auswählen und angeben, wohin die Backup-Dateien kopiert werden sollen.

Daten wiederherstellen

Wenn Sie eine ältere Version einer einzelnen Datei wiederherstellen möchten, ist es gar nicht notwendig, das Modul DATENSICHERUNG der Systemeinstellungen zu starten. Stattdessen klicken Sie die Datei bzw. das Verzeichnis im Dateimanager Nautilus an und führen das Kontextmenükommando AUF FRÜHERE VERSION ZURÜCKSETZEN aus. Im Dateimanager können Sie auch gelöschte Dateien wiederherstellen: Das Kommando VERSCHWUNDENE DATEIEN WIEDERHERSTELLEN öffnet einen Dialog, der alle im Backup gesicherten Dateien anzeigt, die es im aktuellen Verzeichnis *nicht* mehr gibt.

Déjà Dup basiert auf dem in Python entwickelten Backup-Script Duplicity. Das hat den Nachteil, dass die Backup-Dateien in einem sehr speziellen Format vorliegen, sodass die Daten nur mit Déjà Dup selbst oder mit Duplicity wiederhergestellt werden können:

<http://duplicity.nongnu.org>

## Grsync

Eigentlich ist es übertrieben, Grsync als Backup-Werkzeug zu bezeichnen. In Wirklichkeit handelt es sich um eine simple Benutzeroberfläche zum Kommando `rsync` (siehe [Abbildung 36.2](#)). Nach der Installation verbinden Sie eine externe Festplatte oder einen USB-Stick mit Ihrem Computer, starten Grsync und kopieren den Inhalt Ihres Heimatverzeichnisses in ein Verzeichnis der externen Festplatte. Beim ersten Mal müssen dabei alle Dateien kopiert werden, in der Folge nur noch geänderte oder neue Dateien.

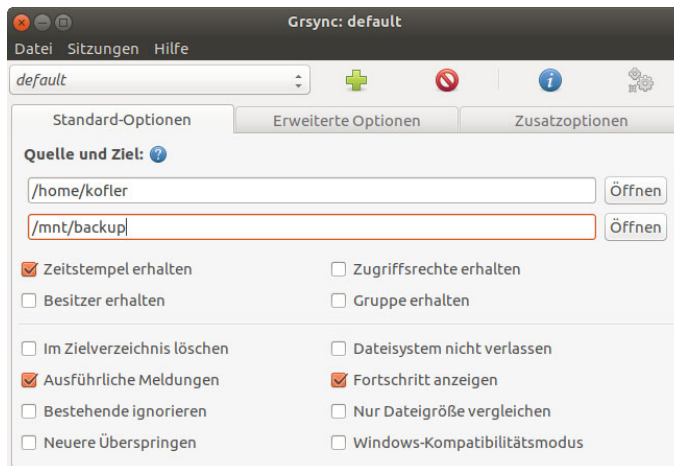


Abbildung 36.2 Verzeichnisse synchronisieren mit Grsync

Die zahlreichen Optionen können Sie im Wesentlichen so lassen, wie sie voreingestellt sind. Zwei Optionen bedürfen aber einer genaueren Erläuterung:

- ▶ **IM ZIELVERZEICHNIS LÖSCHEN** gibt an, ob Grsync auch Löschvorgänge synchronisieren soll. Wenn Sie nach dem ersten Backup in Ihrem Heimatverzeichnis eine Datei löschen, wird diese Datei beim nächsten Mal auch im Backup-Verzeichnis gelöscht. Wenn Ihr Backup vor versehentlichen Löschvorgängen geschützt sein soll, dürfen Sie diese Option nicht aktivieren. Das ist auch die Grundeinstellung. Wenn es Ihnen hingegen wichtig ist, dass das Backup exakt den gleichen Inhalt hat wie das zu sichernde Verzeichnis, sollten Sie die Option aktivieren.
- ▶ **DATEISYSTEM NICHT VERLASSEN** bedeutet, dass Grsync nur solche Dateien synchronisiert, die sich im Dateisystem des Quellverzeichnisses befinden. Zumeist ist es zweckmäßig, diese Option zu aktivieren.

Die größten Vorteile von Grsync sind die einfache Bedienung und der Umstand, dass Ihre Dateien 1:1 in ein zweites Verzeichnis kopiert werden. Sollten Sie je auf Ihr Backup zurückgreifen müssen, brauchen Sie dazu keine speziellen Werkzeuge.

## Back In Time

*Back In Time* ist ein Backup-Programm für persönliche Daten, wobei es sowohl für KDE als auch für Gnome eine eigene Benutzeroberfläche gibt. Die meisten aktuellen Distributionen stellen fertige Back-in-Time-Pakete zur Verfügung. Nach deren Installation führen Sie zur erstmaligen Konfiguration `[Alt]+[F2]` `backintime-qt4` aus.

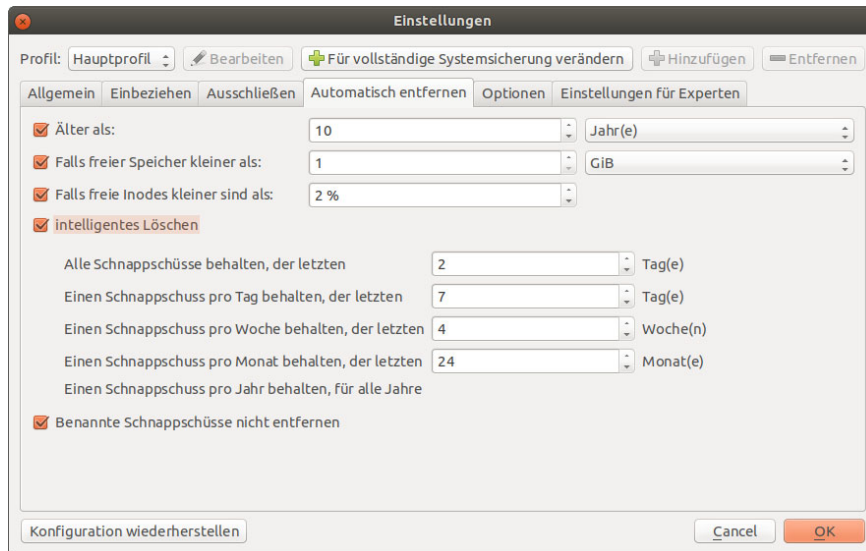


Abbildung 36.3 Konfiguration von Back In Time

Die Konfiguration erfolgt in sechs Dialogblättern (siehe [Abbildung 36.3](#)):

- ▶ **ALLGEMEIN:** Hier geben Sie an, in welchem Verzeichnis die Backups gespeichert werden sollen. Es kann sich dabei auch um einen externen Datenträger handeln, wenn dieser ständig mit Ihrem Rechner verbunden ist. Sie müssen für das Backup-Verzeichnis Schreibrechte haben.

Außerdem stellen Sie in diesem Dialogblatt ein, wie oft die Backups durchgeführt werden sollen. Sinnvolle Einstellungen sind in der Regel STÜNDLICH oder TÄGLICH. Sie können auch den Listeneintrag DEAKTIVIERT angeben – dann müssen Sie jedes Backup manuell starten. Das ist z. B. dann sinnvoll, wenn Sie eine externe Festplatte für Ihre Backups verwenden, die Festplatte aber nicht immer an den Rechner angeschlossen ist.

- ▶ **EINBEZIEHEN:** Hier wählen Sie aus, welche Verzeichnisse gesichert werden sollen. Üblicherweise werden Sie hier einfach Ihr Heimatverzeichnis angeben. Sie können aber auch eine differenzierte Auswahl treffen und beispielsweise nur Ihre Verzeichnisse Dokumente und Bilder sichern.

- ▶ **AUSSCHLIESSEN:** Hier geben Sie an, welche Verzeichnisse und Dateimuster vom Backup ausgenommen sind, z. B. Downloads. Standardmäßig sieht Back In Time vor, dass versteckte Dateien und Verzeichnisse *nicht* gesichert werden. Das ist eine gefährliche Voreinstellung, weil sich gerade in versteckten Verzeichnissen oft wichtige Anwendungsdaten befinden – in `.thunderbird` z. B. Ihre E-Mails, wenn Sie Thunderbird als E-Mail-Client verwenden.
- ▶ **AUTOMATISCH ENTFERNEN:** Damit das Backup-Volumen nicht grenzenlos wächst, können Sie angeben, welche Backup-Daten automatisch gelöscht werden sollen. Zumeist ist SMART REMOVE eine sinnvolle Option: Damit werden Backups von gestern und heute nie angerührt. Ältere Backups werden größtenteils gelöscht, wobei aber sichergestellt wird, dass es je ein Backup für die letzten zwei Wochen, für jeden Monat des laufenden Jahres sowie für jedes vergangene Jahr gibt.
- ▶ **OPTIONEN UND EINSTELLUNGEN FÜR EXPERTEN:** Hier finden Sie einige Optionen für fortgeschrittene Benutzer, die in der Regel nicht verändert werden müssen.

Nach Abschluss der Konfiguration erscheint die Benutzeroberfläche von Back In Time. Hier können Sie jederzeit manuell ein Backup starten, also außerhalb der eingestellten Backup-Periode. Außerdem können Sie vorhandenen Backups Namen geben.

#### Daten wiederherstellen

Die wichtigste Funktion der Benutzeroberfläche besteht aber darin, dass Sie in einem einfachen Verzeichnis-Browser zu jedem gewünschten Zeitpunkt Zugriff auf alle gesicherten Dateien haben. Mit dem Button WIEDERHERSTELLEN können Sie eine irrtümlich gelöschte oder veränderte Datei wiederherstellen. KOPIEREN kopiert die ausgewählten Dateien. Sie können die Dateien nun in einem Dateimanager an einer beliebigen Stelle einfügen.

Standardmäßig kann Back In Time nur persönliche Dateien sichern. Wenn Sie Back In Time zur Sicherung von Systemdateien einsetzen möchten, müssen Sie es im root-Modus starten. Sowohl KDE als auch Gnome sehen entsprechende Starteinträge vor. Leider ist diese Art des Starts nicht Wayland-kompatibel. Der Start funktioniert also nur, wenn Sie X als Grafiksystem verwenden.

#### Interna

Hinter den Kulissen kümmert sich Cron um die automatische Durchführung der Backups durch das Kommando `backintime`. Dabei wird die Konfigurationsdatei `.config/backintime/config` ausgewertet. Die Benutzeroberfläche von Back In Time muss für die automatischen Backups nicht laufen! Die Cron-Steuerung erfolgt durch die Datei `/var/spool/cron/tabs/loginname`.

Im Backup-Verzeichnis werden alle Dateien unkomprimiert gespeichert. (Es gibt leider keine Komprimieroption.) Dateien, die sich von einem Backup zum nächsten nicht ändern, werden nur durch sogenannte *Hardlinks* miteinander verbunden, was eine Menge Platz auf der Festplatte spart. Das funktioniert allerdings nur auf Datenträgern, die derartige Links unterstützen, also nicht auf FAT-formatierten Festplatten oder Memory-Sticks!



## 36.2 Backups auf NAS-Geräten

Wenn Sie nur einen einzelnen Computer besitzen, ist das ideale Backup-Medium in der Regel eine externe Festplatte oder ein USB-Stick. Sobald aber mehrere Rechner im Einsatz sind, besteht oft der Wunsch nach einem zentralen Platz für Backups und nach dem Austausch gemeinsamer Daten. Ideal geeignet für diesen Zweck sind NAS-Geräte. Die Abkürzung NAS steht für *Network Attached Storage* und bezeichnet Datenspeicher, die über das Netzwerk zugänglich sind.

Auf fast allen marktüblichen NAS-Geräten laufen ein Linux-System, der Datei-Server Samba und ein Webserver, der über eine Webschnittstelle bei der Konfiguration des Geräts hilft. Viele Modelle stellen außerdem einen NFS-Server, einen FTP-Server, einen AFP-Server (für Apple), einen SSH-Server, Multimedia-Streaming-Clients, Download-Werkzeuge etc. zur Verfügung.

Auch wenn NAS-Systeme intern in aller Regel Linux-Rechner sind – nach außen verhalten sie sich dank Samba wie ein Windows-Rechner, der Netzwerkverzeichnisse freigibt. Und genau hier liegt das Problem: Linux-Backup-Werkzeuge setzen in der Regel voraus, dass das Ziel des Backups ein Linux- bzw. Unix-kompatibles Dateisystem ist. In einem Windows-Netzwerkverzeichnis ist es aber unmöglich, die Unix-typischen Zugriffsrechte zu speichern. Außerdem können Linux-Backup-Werkzeuge in der Regel nicht direkt auf ein Windows-Netzwerkverzeichnis zugreifen.

Backups in  
Netzwerkver-  
zeichnissen

Es gibt verschiedene Möglichkeiten, diese Probleme zu umgehen:

- ▶ Damit Backup-Werkzeuge in ein Windows-Netzwerkverzeichnis schreiben können, muss dieses vorher in das Dateisystem des Linux-Rechners eingebunden werden. Das kann wahlweise über einen Dialog des Dateimanagers oder durch die Veränderung der Systemdatei `/etc/fstab` erfolgen.
- ▶ Um zu vermeiden, dass Linux-Zugriffsrechte beim Backup verloren gehen, können die zu sichernden Dateien verpackt werden, z. B. in ein komprimiertes tar-Archiv. Das auf den vorangegangenen Seiten vorgestellte Programm Déjà Dup geht auf diese Weise vor.
- ▶ Sie können versuchen, die zu sichernden Dateien direkt im Windows-Netzwerkverzeichnis zu speichern, z. B. mit dem Kommando `rsync` oder mit `Grsync`. Nach meinen Erfahrungen funktioniert das aber leider häufig nicht zufriedenstellend.
- ▶ Zu guter Letzt bieten manche NAS-Geräte die Möglichkeit, auch NFS-Verzeichnisse oder einen `Rsync`-Server als Backup-Medium zu verwenden. Je nach Gerät ist die korrekte Konfiguration aber oft schwierig.

Der meiner Ansicht nach zuverlässigste Weg zu einem Backup auf einem NAS-Gerät führt über ein `mount`-Kommando bzw. einen Eintrag in `/etc/fstab`. Das Ziel besteht darin, ein Verzeichnis des NAS-Geräts in den Verzeichnisbaum des lokalen Rechners

mount bzw.  
`/etc/fstab`

einzubinden. Die Vorgehensweise ist im Detail in [Abschnitt 28.7](#), »Client-Zugriff«, beschrieben. Die Anleitung setzt allerdings voraus, dass das NAS-Gerät ständig läuft, das Netzwerkverzeichnis also immer zur Verfügung steht.

#### Netzwerkverzeichnis unter Gnome nutzen

Anstatt manuell `mount` auszuführen oder `/etc/fstab` zu ändern, können Sie das Netzwerkverzeichnis auch unter Gnome in das lokale Dateisystem einbinden. Dazu führen Sie `MIT SERVER VERBINDEN` aus und geben das gewünschte Verzeichnis wie im folgenden Beispiel an:

```
smb://benutzername@nas-hostname/verzeichnisname
```

Damit Sie diese Angaben in Zukunft nicht jedes Mal wiederholen müssen, setzen Sie nach dem Verbindungsaufbau mit `Strg+D` ein Lesezeichen. Hinter den Kulissen verwendet Gnome das GVFS (*Gnome Virtual File System*) für den Zugriff auf Netzwerkverzeichnisse. Die Netzwerkverzeichnisse werden im unsichtbaren Verzeichnis `.gvfs` in den Verzeichnisbaum integriert. In Nautilus sind die eingebundenen Verzeichnisse leicht zu finden und zu nutzen, das gilt aber leider nicht für alle Backup-Programme: Während Déjà Dup die Netzwerkverzeichnisse direkt im Dateiauswahldialog anzeigt, müssen Sie das Netzwerkverzeichnis bei Grsync explizit als Unterverzeichnis von `.gvfs` öffnen.

### 36.3 Dateien komprimieren und archivieren

In den folgenden Abschnitten stelle ich eine ganze Palette von Kommandos vor, die in unterschiedlicher Form bei der Archivierung und Sicherung von Dateien helfen: `tar`, `zip`, `rsync`, `rdiff-backup`, `rsnapshot` etc. Dieser Abschnitt beginnt mit Kommandos zum Komprimieren und Archivieren. Einen ersten Überblick gibt [Tabelle 36.1](#).

| Kommando             | Bedeutung  |
|----------------------|--|
| <code>gzip</code>    | komprimiert eine Datei.  |
| <code>gunzip</code>  | dekomprimiert die Datei wieder.  |
| <code>bzip2</code>   | komprimiert eine Datei (höhere Kompression als <code>gzip</code> , aber langsamer).  |
| <code>bunzip2</code> | dekomprimiert die Datei wieder.  |
| <code>xz</code>      | komprimiert eine Datei (höhere Kompression als <code>bzip2</code> , noch langsamer). |
| <code>unxz</code>    | dekomprimiert die Datei wieder.  |
| <code>lzop</code>    | komprimiert/dekomprimiert deutlich schneller als <code>gzip</code> .                 |
| <code>tar</code>     | erstellt bzw. extrahiert ein Dateiarchiv.  |

**Tabelle 36.1** Werkzeuge zum Komprimieren und Archivieren von Dateien

| Kommando | Bedeutung                                   |
|----------|---|
| zip      | erzeugt ein Windows-kompatibles ZIP-Archiv. |
| unzip    | extrahiert ein ZIP-Archiv.                  |
| zipinfo  | zeigt Informationen über ein ZIP-Archiv an. |

Tabelle 36.1 Werkzeuge zum Komprimieren und Archivieren von Dateien (Forts.)

### Dateien komprimieren (gzip, bzip2, xz, lzop)

gzip komprimiert die als Parameter angegebenen Dateien und benennt sie in `name.gz` um. `gunzip` funktioniert in die umgekehrte Richtung. Die beiden Kommandos verwenden den sogenannten LZ77-Lempel-Ziv-Algorithmus, der sich besonders für Textdateien eignet, nicht aber für Audio- oder Video-Dateien. Die Komprimierung ist selbstverständlich verlustlos, d. h., nach dem Dekomprimieren steht die ursprüngliche Datei wieder unverändert zur Verfügung. Die folgenden Kommandos demonstrieren die Anwendung:

gzip und gunzip

```
user$ ls -l filesystem.tex
... 178794 1. Aug 17:43 filesystem.tex
user$ gzip filesystem.tex
user$ ls -l filesystem.tex.gz
... 57937 1. Aug 17:43 filesystem.tex.gz
user$ gunzip filesystem.tex.gz
```

bzip2 und bunzip2 sind Alternativen zu gzip/gunzip. Der Vorteil dieser Kommandos besteht in der etwas besseren Komprimierung, der Nachteil in der etwas langsameren Ausführung. Die Dateierweiterung derart komprimierter Dateien ist `.bz2`.

bzip2 und bunzip2

```
user$ bzip2 filesystem.tex
user$ ls -l filesystem.tex.bz2
... 47105 1. Aug 17:43 filesystem.tex.bz2
user$ bunzip2 filesystem.tex.bz2
```

Anstelle von gzip und bzip2 können Sie zum Komprimieren auch xz verwenden. Das Ergebnis sind in den meisten Fällen noch kleinere Dateien, das Komprimieren erfordert dafür noch mehr Zeit bzw. CPU-Ressourcen. Wenn die kleinstmögliche Komprimierung das vorrangige Ziel ist, können Sie auch das Kommando `7zr` aus dem Paket `p7zip` ausprobieren.

xz und unxz

Ganz anders ist die Zielsetzung von lzop: Dieses Komprimierkommando arbeitet *viel* schneller als alle bisher genannten Kommandos. Dafür sind die resultierenden Dateien aber vergleichsweise groß (bei meinen Tests ca. 50 Prozent größer als bei gzip). Der Einsatz von lzop ist vor allem dann empfehlenswert, wenn Sie *on the fly* mit möglichst

lzop

geringer CPU-Belastung komprimieren möchten, z. B. zur Übertragung einer großen Datei über eine Netzwerkverbindung.

Im folgenden Beispielkommando wird ein Logical Volume mit `cat` ausgelesen und mit `lzop` komprimiert. Das dauert nur unwesentlich länger als das direkte Kopieren des Logical Volumes in eine Image-Datei.

```
root# cat /dev/vg1/lv3 | lzop -c > lv3.img.lzo (55 Sekunden)
root# cat /dev/vg1/lv3 > lv3.img (50 Sekunden)
```

### Komprimierte Archive erstellen (tar, zip)

**tar** tar ist das bevorzugte Kommando, um unter Linux mehrere Dateien in einem Archiv zusammenzufassen, wobei das Archiv üblicherweise mit `gzip` oder `bzip2` komprimiert wird. `tar` war ursprünglich dazu konzipiert, Dateien auf einen Streamer zu schreiben bzw. von dort zu lesen. Da derartige Streamer nur noch relativ selten eingesetzt werden, beschreibe ich an dieser Stelle nur die Anwendung für Dateiarhive.

Das folgende Kommando fügt sämtliche Dateien aus dem Verzeichnis `buch` in die komprimierte Archivdatei `buch.tgz` ein. Kurz eine Erklärung zu den Optionsbuchstaben: `c` steht für *create*, d. h., `tar` soll ein Archiv erzeugen. `z` steht *zip*, d. h., das Archiv soll mit `gzip` komprimiert werden. `f` steht für *file*, d. h., `tar` soll eine Archivdatei erzeugen, anstatt das Archiv auf eine Streamer-Kassette zu schreiben. Den gewünschten Dateinamen geben Sie im Anschluss an die Option an. Die übliche Dateikennung für derartige Archive lautet `.tar.gz` oder kurz `.tgz`.

```
user$ tar -czf meinarchiv.tgz buch/
```

`tar -tzf` liefert ein Inhaltsverzeichnis des Archivs. Die Dateien innerhalb des Archivs sind willkürlich geordnet. Bei den meisten Distributionen ist `less` so konfiguriert, dass Sie den Archivinhalt einfach mit `less name.tgz` ansehen können.

```
user$ tar -tzf meinarchiv.tgz
linuxbuch/
linuxbuch/lanserver.tex
linuxbuch/security.tex~
linuxbuch/buch.tex
linuxbuch/u4.txt~
...
```

`tar -xzf` packt das Archiv aus und extrahiert alle enthaltenen Dateien:

```
user$ cd anderes-verzeichnis/
user$ tar -xzf meinarchiv.tgz
```

Beim folgenden Beispiel extrahiert `tar` nur `*.tex`-Dateien aus dem Archiv. Achten Sie auf die Apostrophe für das Dateimuster, um eine sofortige Auswertung durch die Shell zu vermeiden!

```
user$ tar -xzf meinarchiv.tgz '*.tex'
```

Wenn Sie Archive mit `bzip2` statt mit `gzip` komprimieren möchten, ersetzen Sie die Option `z` durch `j`.

In der Unix/Linux-Welt sind `tar`-Dateien das bevorzugte Format zur Weitergabe von Dateiarchiven. Wenn Sie mit Windows-Anwendern kommunizieren, sind ZIP-Archive aber die bessere Wahl. Das folgende Kommando fügt alle als Parameter übergebenen HTML-Dateien in `meinarchiv.zip` ein:

```
user$ zip meinarchiv.zip *.html
```

Wenn Sie den Inhalt ganzer Verzeichnisse archivieren möchten, geben Sie die Option `-r` an:

```
user$ zip -r meinarchiv.zip mywebsite/
```

Den Inhalt einer ZIP-Datei sehen Sie sich mit `zipinfo` an:

```
user$ zipinfo meinarchiv.zip
Archive:  test.zip   143677915 bytes   1899 files
-rw-r--r--  2.3 unx    78039 tx defN 10-Jul-16 11:27 linuxbuch/lanserver.tex
-rw-r--r--  2.3 unx   115618 tx defN  7-Apr-15 15:58 linuxbuch/security.tex
-rw-r--r--  2.3 unx    3899 tx defN 28-Jul-16 16:38 linuxbuch/buch.tex
-rw-r--r--  2.3 unx     752 tx defN 11-Feb-14 12:06 linuxbuch/u4.txt
...
```

Zum Extrahieren des Archivs verwenden Sie `unzip`:

```
user$ cd anderes-verzeichnis/
user$ unzip meinarchiv.zip
```

## 36.4 Verzeichnisse synchronisieren (rsync)

Das Kommando `rsync` synchronisiert Verzeichnisbäume. Sie können damit im ersten Durchlauf alle Dateien von einem Verzeichnis in ein neues Verzeichnis kopieren. Bei den weiteren Durchläufen werden nur noch geänderte Dateien kopiert und (auf Wunsch) auch Löschvorgänge repliziert. `rsync` eignet sich damit ausgezeichnet, um z. B. täglich oder wöchentlich eine vollständige Kopie eines Verzeichnisses auf einer externen Festplatte zu synchronisieren.

`rsync` kann auch über Netzwerkverbindungen eingesetzt werden. Standardmäßig erfolgt die Kommunikation via `ssh`. Damit ist die Datenübertragung auch gleich verschlüsselt. Alternativ können Sie auch ein anderes externes Shell-Programm einsetzen oder konfigurieren auf der Gegenseite einen `rsync`-Server.

Tabelle 36.2 zeigt die Syntax zur Angabe der Quell- und Zielverzeichnisse. Tabelle 36.3 fasst die wichtigsten Optionen zur Steuerung von `rsync` zusammen.

| Schreibweise                             | Bedeutung  |
|--|--|
| <code>datei1 datei2</code>               | lokale Dateien   |
| <code>verzeichnis</code>                 | lokales Verzeichnis                                      |
| <code>host:verz</code>                   | Verzeichnis auf dem Rechner <code>host</code>            |
| <code>user@host:verz</code>              | wie oben mit SSH-Login unter dem Namen <code>user</code> |
| <code>rsync://user@host/verz</code>      | Kommunikation mit <code>rsync</code> -Server             |
| <code>rsync://user@host:port/verz</code> | <code>rsync</code> -Server am angegebenen Port           |

**Tabelle 36.2** Angabe von Quell- und Zielverzeichnissen in `rsync`

| Option   | Wirkung  |
|--|--|
| <code>-a</code> bzw. <code>--archive</code>    | kopiert rekursiv und erhält alle Dateiinformationen.   |
| <code>--delete</code>                          | löscht im Zielverzeichnis Dateien bzw. Verzeichnisse, die im Quellverzeichnis nicht mehr existieren. |
| <code>-D</code>                                | berücksichtigt auch Device- und Spezialdateien.  |
| <code>--exclude=muster</code>                  | überspringt die angegebenen Dateien.   |
| <code>-g</code> bzw. <code>--group</code>      | erhält die Gruppenzugehörigkeit.   |
| <code>-l</code> bzw. <code>--links</code>      | dupliziert symbolische Links.  |
| <code>-o</code> bzw. <code>--owner</code>      | erhält die Besitzerinformationen.  |
| <code>-p</code> bzw. <code>--perms</code>      | erhält die Zugriffsrechte.   |
| <code>-r</code> bzw. <code>--recursive</code>  | kopiert rekursiv auch alle Unterverzeichnisse.   |
| <code>-t</code> bzw. <code>--times</code>      | erhält die Änderungszeit.  |
| <code>-u</code> bzw. <code>--update</code>     | ignoriert bereits vorhandene, ältere Dateien.  |
| <code>-v</code> bzw. <code>--verbose</code>    | zeigt an, was gerade passiert.   |
| <code>-W</code> bzw. <code>--whole-file</code> | kopiert bei Änderungen die gesamte Datei.  |
| <code>-z</code>                                | komprimiert die via SSH übertragenen Daten.  |

**Tabelle 36.3** `rsync`-Optionen

#### Lokale Anwendung

Um ein ganzes Verzeichnis inklusive aller Unterverzeichnisse zu synchronisieren, verwenden Sie die Option `-a`, die als Kurzschreibweise für eine ganze Reihe anderer Optionen gilt (`-r1ptgoD`). Die Option bewirkt eine rekursive Verarbeitung aller Unterverzeichnisse und stellt sicher, dass möglichst alle Dateiinformationen erhal-

ten bleiben, also Besitzer, Gruppenzugehörigkeit, Zeitpunkt der letzten Änderung etc. Falls `verz2` noch nicht existiert, wird das Verzeichnis erzeugt. Anders als bei `cp` werden bereits vorhandene Dateien, die seit dem letzten Kopieren unverändert geblieben sind, nicht neuerlich kopiert.

```
user$ rsync -a verz1/ verz2/
```

Standardmäßig kopiert bzw. aktualisiert `rsync` alle neuen bzw. geänderten Dateien, löscht aber nichts. Wenn Sie möchten, dass aus `verz1` gelöschte Dateien oder Verzeichnisse auch in `verz2` gelöscht werden, geben Sie zusätzlich die Option `--delete` an. Es sollte klar sein, dass diese Option gefährlich ist: Wenn Sie versehentlich ein Verzeichnis löschen, wird genau dieses Verzeichnis beim nächsten Backup-Vorgang auch auf der Backup-Festplatte gelöscht!

Bei der Anwendung von `rsync` zur Synchronisation von Verzeichnissen auf unterschiedlichen Rechnern müssen Sie das Quell- und das Zielverzeichnis in der Schreibweise `hostname:verzeichnis` bzw. `username@hostname:verzeichnis` angeben. Im ersten Fall verwendet `rsync` den aktuellen Benutzernamen.

Anwendung im  
Netzwerk

Durch das folgende Kommando wird das Verzeichnis `verz1` des lokalen Benutzers `username` auf dem Rechner `saturn.sol` mit dem Verzeichnis `verz2` auf dem Rechner `mars.sol` synchronisiert. Für die Passwordeingabe ist `ssh` verantwortlich. Sie müssen also das Login-Passwort des Benutzers `username` auf dem Rechner `mars.sol` eingeben.

```
username@saturn.sol$ rsync -e ssh -az verz1/ mars.sol:verz2/
username@mars.sol's password: *****
```

`rsync` kann Dateien auch von einem entfernten Rechner auf den lokalen übertragen. Das folgende Kommando synchronisiert also in die umgekehrte Richtung:

```
username@saturn.sol$ rsync -e ssh -az mars.sol:verz2/ verz3/
username@mars.sol's password: *****
```

Wenn `rsync` durch ein automatisches Backup-Script aufgerufen werden soll, stört natürlich die interaktive Passwordeingabe. Die Lösung besteht darin, auf dem lokalen Rechner eine private Schlüsseldatei einzurichten und auf dem Partnerrechner den dazu passenden öffentlichen Schlüssel (siehe [Abschnitt 31.4](#), »Authentifizierung mit Schlüsseln«). Wenn Sie bei der Erzeugung der Schlüssel auf die sogenannte *Passphrase* verzichten, ist nun ein SSH-Login ohne Passwort möglich. Aus Sicherheitsgründen sollten Sie für das Backup-Script einen eigenen Account vorsehen.

Neben der hier vorgestellten Anwendung von `rsync` in Kombination mit SSH besteht auch die Möglichkeit, dass das lokale `rsync`-Kommando mit einem `rsync`-Server auf dem entfernten Rechner kommuniziert. Das setzt voraus, dass auf dem Partnerrechner ein `rsync`-Server eingerichtet wurde. Dessen Konfiguration erfolgt durch die Datei

rsync-Server

/etc/rsyncd.conf. Die Kommunikation zwischen dem rsync-Client und dem rsync-Server erfolgt dann über den Port 873. Dieser Port darf daher nicht durch eine Firewall blockiert sein.

In der Praxis ist die Konfiguration eines rsync-Servers zumeist nur empfehlenswert, wenn der Server regelmäßig von verschiedenen Clients gespiegelt wird, also als Mirror-Server dient:

<https://unix.stackexchange.com/questions/26182>

<https://serverfault.com/questions/100707>

### 36.5 Inkrementelle Backups (rdiff-backup)

Eine interessante Alternative zu rsync ist das Kommando rdiff-backup. Der wichtigste Unterschied zu rsync besteht darin, dass rdiff-backup bei veränderten Dateien auch die alte Version im Backup-Verzeichnis archiviert. Um Platz zu sparen, können statt einer Kopie der betreffenden Datei auch nur die Änderungen gespeichert werden, optional in komprimierter Form. rdiff-backup liefert also ohne viel Mühe ein inkrementelles Backup, aus dem Sie auch ältere Versionen einer Datei wiederherstellen können. Im Prinzip bietet rdiff-backup dieselben Funktionen wie die »Time Machine« von Apples macOS – nur ohne spektakuläre Benutzeroberfläche.

**Backups durchführen**

In der einfachsten Form wenden Sie rdiff-backup auf zwei lokale Verzeichnisse an. Wenn das Zielverzeichnis noch nicht existiert, wird es erzeugt.

```
root# rdiff-backup /home /home-backup
```

rdiff-backup erzeugt im Backup-Verzeichnis das Unterverzeichnis rdiff-backup-data. Darin speichert es diverse statistische Daten und Statusinformationen. Außerdem enthält das Verzeichnis increments alte Versionen von Dateien, die sich mittlerweile geändert haben oder die gelöscht wurden. Dabei werden nur die Änderungen gespeichert (.diff) und zusätzlich komprimiert. Außerdem wird in den Dateinamen das Datum der letzten Version integriert. Daraus ergeben sich dann unübersichtliche Dateinamen in der Form dateiname.2017-11-03T08:37:58+02:00.diff.gz.

**Zugriff auf Backups**

Wenn Sie auf das Backup zurückgreifen möchten, enthält /home-backup den Zustand des /home-Verzeichnisses zum Zeitpunkt des letzten Backups mit Ausnahme des rdiff-backup-data-Verzeichnisses genau so, als hätten Sie das Backup mit `cp -a` oder `rsync -a --delete` ausgeführt. Der Zugriff auf das letzte Backup ist also ganz einfach. Natürlich können Sie das Backup auch mit rdiff-backup wiederherstellen. Dazu verwenden Sie die Option `-r` und die Zeitangabe `now`. Das folgende Kommando stellt das Backup probeweise in einem temporären Verzeichnis wieder her:

```
root# rdiff-backup -r now /home-backup /tmp/home-aktuell
```



Wenn Sie auf eine ältere Version einer Datei bzw. auf eine mittlerweile gelöschte Datei zugreifen möchten, wird es komplizierter: Sie müssen der Reihe nach alle `.diff`-Dateien anwenden (die neueste zuerst), bis Sie den gewünschten Zeitpunkt in der Vergangenheit wiederhergestellt haben. Natürlich müssen Sie das nicht manuell tun – `rdiff-backup` hilft Ihnen dabei. Das folgende Kommando stellt den Zustand des `/home`-Verzeichnisses so wieder her, wie er vor zehn Tagen war:

```
root# rdiff-backup -r 10D /home-backup/ /tmp/home-historisch
```

Den Backup-Zeitpunkt können Sie wahlweise absolut (z. B. 2017-12-31) oder relativ in Stunden (h), Tagen (D), Wochen (w) etc. angeben – siehe auch `man rdiff-backup` im Abschnitt `TIME FORMATS`. Beachten Sie, dass die Wiederherstellung alter Dateien mit zunehmender Versionsanzahl einen erheblichen CPU-Aufwand verursacht und entsprechend langsam ist!

Oft wollen Sie nur eine einzelne Datei oder ein Unterverzeichnis in einer alten Version wiederherstellen. Dabei können Sie auch eine gar nicht mehr existierende Datei bzw. ein mittlerweile gelöscht Verzeichnis angeben:

```
root# rdiff-backup -r 10D /home-backup/datei datei-historisch
root# rdiff-backup -r 10D /home-backup/verz/ verz-historisch
```

Wenn Sie `rdiff-backup` regelmäßig ausführen, wächst das Backup-Verzeichnis im Laufe der Zeit immer stärker an. Um alle Backup-Dateien zu löschen, die älter als vier Monate sind, gehen Sie so vor:

**Alte Backups  
löschen**

```
root# rdiff-backup --remove-older-than 4M --force /home-backup/
```

Statt eines konkreten Zeitpunkts können Sie auch angeben, wie viele Backup-Versionen maximal archiviert bleiben sollen. Das folgende Kommando reduziert die Backup-Versionen auf drei:

```
root# rdiff-backup --remove-older-than 3B --force /home-backup/
```

In allen bisherigen Beispielen bin ich davon ausgegangen, dass sich das Quell- und das Zielverzeichnis im lokalen Dateisystem befinden. `rdiff-backup` kann aber über das Netzwerk auch auf externe Verzeichnisse zugreifen. Anders als bei `rsync` muss dazu `rdiff-backup` auch auf dem externen Rechner installiert sein! Die Kommunikation erfolgt über SSH. Eine spezielle Konfiguration von `rdiff-backup` ist nicht erforderlich.

**Netzwerk-Backup**

Bei der Angabe externer Verzeichnisse gilt nahezu dieselbe Syntax wie bei `rsync`. Der einzige Unterschied besteht darin, dass nach dem Hostnamen *zwei* Doppelpunkte angegeben werden müssen:

```
root# rdiff-backup user@firma-abc.de::/home /home-backup
```

Noch mehr Details und Beispiele zum Umgang mit `rdiff-backup` bietet die folgende Webseite:

<http://www.nongnu.org/rdiff-backup>

**Duplicity** Wenn Ihnen die Idee von `rdiff-backup` zusagt, Sie sich aber außerdem noch die Verschlüsselung des Backups sowie ein Upload via SSH oder FTP auf einen externen Server wünschen, lohnt sich vielleicht ein Blick auf das Python-Programm `Duplicity`. Es ist ähnlich wie `rdiff-backup` zu bedienen, erzeugt allerdings `tar`-Archive. `Duplicity` befindet sich allerdings nun schon seit mehreren Jahren im Beta-Stadium. Nichtsdestotrotz dient es als Basis für die Backup-Benutzeroberfläche `Déjà Dup`, die ich Ihnen am Beginn dieses Kapitels vorgestellt habe.

<http://duplicity.nongnu.org>

### 36.6 Inkrementelle Backups (`rsnapshot`)

Auch das Perl-Script `rsnapshot` aus dem gleichnamigen Paket baut auf `rsync` auf. Im Unterschied zum gerade beschriebenen Kommando `rdiff-backup` verwendet es Hardlinks, um auf bereits gesicherte Dateien früherer Backups zurückzugreifen. Das macht den Zugriff auf ältere Backup-Versionen (»Snapshots«) einfacher als bei `rdiff-backup`. Eine Komprimierung der Backups ist hingegen nicht vorgesehen.

`rsnapshot` ist so konzipiert, dass das Script regelmäßig automatisch ausgeführt wird. `rsnapshot` sichert bei entsprechender Konfiguration sowohl lokale Verzeichnisse als auch via SSH Verzeichnisse von anderen Rechnern im lokalen Netzwerk. Alle Backups werden auf dem lokalen Rechner gespeichert, auf dem `rsnapshot` ausgeführt wird. Dieser Ansatz ist genau umgekehrt als bei vielen anderen Backup-Tools: `rsnapshot` ist nicht dazu gedacht, ein Backup der lokalen Dateien auf einem anderen Rechner zu speichern. Vielmehr sichert das Kommando die Daten mehrerer via `ssh` oder `rsync` erreichbarer Rechner im lokalen Dateisystem.

**Konfiguration** Anstatt an `rsnapshot` zahlreiche Optionen zu übergeben, steuern Sie das Kommando durch die Konfigurationsdatei `/etc/rsnapshot.conf`. Für diese Datei gelten zwei wichtige Syntaxregeln: Verzeichnisse müssen mit einem Slash enden (also `/verzeichnis/`, nicht `/verzeichnis`), und die Elemente der Konfigurationsdatei müssen durch Tabulatorzeichen (nicht Leerzeichen) voneinander getrennt werden!

Für erste Experimente können Sie die zusammen mit `rsnapshot` mitgelieferte Konfigurationsdatei bis auf wenige Details unverändert lassen. Die drei wichtigsten Parameter, die Sie kennen und gegebenenfalls selbst einstellen müssen, sind `snapshot_root`, `backup` und `interval`.

`snapshot_root` gibt an, in welchem Verzeichnis die Backups gespeichert werden sollen. Standardmäßig kommt das Verzeichnis `/var/cache/rsnapshot` zum Einsatz, das bei der Installation von `rsnapshot` automatisch eingerichtet wurde. `snapshot_root`

`backup` gibt an, welches Verzeichnis wo gesichert werden soll. `backup` kann mehrfach jeweils in einer eigenen Zeile angegeben werden, um mehrere Verzeichnisse an unterschiedlichen Orten zu sichern. Für Backups innerhalb des lokalen Dateisystems sehen die `backup`-Einstellungen wie folgt aus: `backup`

```
# in /etc/rsnapshot.conf
...
backup /home/ localhost/
backup /etc/ localhost/
```

Um ein Verzeichnis `/home/user/Mail` des externen, via SSH erreichbaren Rechners `mars.sol` zu sichern, müssen Sie in der bereits vorhandenen Zeile `cmd_ssh` das Kommentarzeichen `#` entfernen. In der `backup`-Zeile geben Sie den Login-Namen, den Hostnamen sowie das gesamte zu sichernde Verzeichnis an. Damit das funktioniert, müssen Sie vor dem ersten Backup einen mit `root`-Rechten eingerichteten SSH-Schlüssel ohne Passphrase in die Datei `/home/user/.ssh/authorized_keys` des Rechners `mars.sol` kopieren (siehe [Abschnitt 31.4](#), »Authentifizierung mit Schlüsseln«). Vorsicht: Die auf dem lokalen Rechner im Verzeichnis `/root/.ssh` gespeicherte Schlüsseldatei darf nicht in falsche Hände geraten!

```
# in /etc/rsnapshot.conf
...
cmd_ssh /usr/bin/ssh
...
backup user@mars.sol:/home/user/Mail/ mars.sol/
```

Optional besteht die Möglichkeit, während des Backups einen LVM-Snapshot durchzuführen oder Scripts auszuführen, bestimmte Dateimuster vom Backup auszuschließen etc. Details können Sie in der mitgelieferten Konfigurationsdatei sowie in der `man`-Seite zu `rsnapshot` nachlesen.

`interval` definiert, wie viele Backup-Versionen für ein bestimmtes Zeitintervall gespeichert werden sollen. Die Standardeinstellungen sehen so aus: `interval`

```
# in /etc/rsnapshot.conf
...
interval hourly 6
interval daily 7
interval weekly 4
#interval monthly 3
```

Das bedeutet, dass von Backups, die durch das Kommando `rsnapshot hourly` ausgeführt werden, sechs Versionen gespeichert werden. Des Weiteren werden sieben

Backups von `rsnapshot daily` archiviert sowie vier Backups von `rsnapshot weekly`. Das Intervall `monthly` ist standardmäßig nicht definiert.

Die Einstellung `interval hourly 6` ist dann zweckmäßig, wenn das Kommando `rsnapshot hourly` nicht stündlich, sondern nur alle vier Stunden ausgeführt wird, wie dies in `/etc/cron.d/rsnapshot` vorgesehen ist. Wenn Sie `rsnapshot hourly` hingegen wirklich jede Stunde ausführen möchten, wäre die Einstellung `interval hourly 24` zweckmäßiger.

Sie können nach Bedarf beliebige weitere Intervalle definieren. `rsnapshot` speichert die Backups für jedes Intervall in einem jeweils eigenen Verzeichnis und verlinkt nicht geänderte Backups nur innerhalb des Verzeichnisses eines Intervalls. Das bedeutet, dass der Speicherbedarf mit jedem zusätzlichen Intervall stark steigt.

**Manueller Aufruf** `rsnapshot` kann manuell aufgerufen werden. Das erfordert `root`-Rechte. Als Parameter müssen Sie dabei ein in der Konfigurationsdatei mit `interval` definiertes Zeitintervall angeben:

```
root# rsnapshot daily
```

Nach dem Backup finden Sie die gesicherten Daten im folgenden Verzeichnis:

```
/var/cache/rsnapshot/<interval.n>/<hostname>/<verzeichnis>
```

Dabei gibt `interval` das Intervall an, standardmäßig `hourly`, `daily`, `weekly` oder `monthly`. `n` gibt die Backup-Version an: 0 für das aktuellste Backup, 1 für die letzte Version, 2 für die vorletzte Version etc. `hostname` gibt an, von welchem Rechner die gesicherten Daten stammen, wobei `localhost` den lokalen Rechner bezeichnet.

Das aktuellste stündliche Backup des lokalen Verzeichnisses `/etc` befindet sich also im folgenden Verzeichnis:

```
/var/cache/rsnapshot/hourly.0/localhost/etc
```

Das aktuellste monatliche Backup des Verzeichnisses `/home/user/Mail` des Servers `mars.sol` befindet sich in diesem Verzeichnis:

```
/var/cache/rsnapshot/monthly.0/mars.sol/home/user/Mail
```

**Automatischer Aufruf** Um die Backups zu automatisieren, ist ein regelmäßiger Aufruf von `rsnapshot` durch die Cron-Konfigurationsdatei `/etc/cron.d/rsnapshot` vorgesehen. Sie müssen dazu nur die Kommentarzeilen vor den vier bereits vorgesehenen Cron-Zeilen entfernen:

```
# /etc/cron.d/rsnapshot
0 */4 * * * root /usr/bin/rsnapshot hourly
30 3 * * * root /usr/bin/rsnapshot daily
0 3 * * 1 root /usr/bin/rsnapshot weekly
30 2 1 * * root /usr/bin/rsnapshot monthly
```

Damit wird `rsnapshot` alle vier Stunden, also um 0:00 Uhr, 4:00 Uhr, 8:00 Uhr etc., täglich um 3:30 Uhr, wöchentlich montags um 3:00 Uhr sowie monatlich am ersten Tag des Monats um 2:30 Uhr mit den Parametern `hourly`, `daily`, `weekly` und `monthly` ausgeführt. Natürlich können Sie die Zeiten nach eigenem Gutdünken variieren. Beachten Sie, dass alle in der Cron-Datei verwendeten Zeitintervalle auch in `/etc/rsnapshot.conf` definiert sein müssen! Für das Intervall `monthly` ist das standardmäßig nicht der Fall.

## 36.7 Backup-Scripts

Im letzten Abschnitt dieses Kapitels stehen praktische Beispiele im Vordergrund.

Das folgende Script synchronisiert den Inhalt des lokalen Verzeichnisses `data` mit dem gleichnamigen Verzeichnis auf einer externen Festplatte mit dem Namen `backup`. Das Script testet, ob die externe Festplatte zur Verfügung steht. Sollte das nicht der Fall sein, wird auf die Durchführung des Backups verzichtet.

**rsync mit Cron automatisieren**

```
#!/bin/bash
if [ -d /media/backup/ ]; then
    rsync -avW --delete /home/kofler/data /media/backup/
fi
```

In aller Regel werden Sie Backup-Scripts regelmäßig automatisch ausführen. Dazu richten Sie am einfachsten einen Cron-Job ein. Am einfachsten ist es, das Script direkt in einem Cron-Verzeichnis zu speichern, z. B. unter dem Dateinamen `/etc/cron.daily/mybackup`. Achten Sie darauf, dass der Dateiname keinen Punkt enthalten darf und dass die Datei ausführbar sein muss (`chmod a+x`).

Die andere Variante besteht darin, dass Sie das Backup-Script in einem beliebigen Verzeichnis speichern, z. B. in `/etc/myscripts`. Zum automatischen Aufruf des Scripts richten Sie eine neue Datei in `/etc/cron.d` ein, z. B. nach diesem Muster:

```
# Datei /etc/cron.d/mybackup
15 2 * * * root /etc/myscripts/mybackup
```

Das Backup-Script `mybackup` wird damit täglich um 2:15 ausgeführt.

Das folgende Script erzeugt ein komprimiertes `tar`-Archiv des Verzeichnisses `data`. Das Archiv wird unter zwei Dateinamen gespeichert: `mydata-day-DD.tar.gz` und `mydata-month-MM.tar.gz`. Dabei gibt `DD` den Tag (01 bis 31) und `MM` den Monat an (01 bis 12). Wenn das Script täglich ausgeführt wird, haben Sie mit der Zeit 43 Backup-Versionen, die den Zustand des Backup-Verzeichnisses für die letzten 28 bis 31 Tage sowie für die letzten 12 Monate widerspiegeln.

**Tägliche und monatliche Sicherheitskopien**

```
#!/bin/bash
fname1=/backup/mydata-day-$(date +%d).tar.gz
fname2=/backup/mydata-month-$(date +%m).tar.gz
tar czf $fname1 /home/kofler/data
cp $fname1 $fname2
chmod 600 $fname1 $fname2
```

**Backups mit LVM** Wenn sich Dateien während der Durchführung eines Backups ändern, ist die resultierende Sicherheitskopie inkonsistent. Nicht immer ist es aber möglich, für das Backup die betreffenden Programme oder Server-Dienste zu stoppen. Eine mögliche Lösung für dieses Problem besteht darin, zu Beginn des Backups einen LVM-Snapshot zu erstellen und diesen als Basis für das Backup zu verwenden. Das setzt natürlich voraus, dass sich das Verzeichnis mit den zu sichernden Daten in einem Dateisystem befindet, das in einem Logical Volume gespeichert wird (und nicht direkt auf einer Festplattenpartition).

Zum Erzeugen eines Snapshots verwenden Sie das Kommando `lvcreate` mit der Option `-s`. Mit `-L` geben Sie an, wie viele Daten sich während der Lebensdauer des Snapshots – also während der Zeit, in der das Backup durchgeführt wird – maximal verändern dürfen. Im LVM-Speicherpool (also im PV) muss dafür ausreichend freier Platz sein. Die erforderliche Größe des Pufferspeichers ist anfänglich schwer abzuschätzen. Wenn Sie mit dem Backup fertig sind, können Sie mit `lvdisplay` feststellen, wie viel Prozent des Puffers bis jetzt beansprucht wurden.

Beim folgenden Beispiel-Skript gehe ich davon aus, dass sich das Verzeichnis `/home` im LV `/dev/vg1/myhome` befindet. Um vom `/home`-Verzeichnis ein konsistentes Backup durchzuführen, während dessen sich keine Dateien ändern, erstellen Sie mit `lvcreate` den Snapshot `homesnap`. Als Pufferspeicher sehen Sie 2 GiB vor.

Diesen Snapshot binden Sie beim Verzeichnis `/var/homesnap` in den Verzeichnisbaum ein und verwenden ihn anschließend als Datenquelle für Ihr Backup-Kommando oder -Skript. Zuletzt lösen Sie `homesnap` wieder aus dem Dateisystem und entfernen den Snapshot mit `lvremove`. Die Option `-f` unterbindet die Rückfrage, ob Sie das wirklich wollen.

```
#!/bin/bash
mkdir -p /var/homesnap
lvcreate -s -L 2G -n homesnap /dev/vg1/home
mount -t ext4 /dev/vg1/homesnap /var/homesnap
tar czf /backup/mybackup.tgz /var/homesnap
lvdisplay /dev/vg1/homesnap > /tmp/backup.log
umount /var/homesnap
lvremove -f /dev/vg1/homesnap
```

Das Kommando `lvdisplay` dient zur Kontrolle, ob Sie den Pufferspeicher für den LVM-Snapshot richtig dimensioniert haben:

```
root# cat /var/homesnap
...
COW-table size      2.00 GB
Allocated to snapshot 5.23%
```

Im obigen Beispiel wurden also nur fünf Prozent des Pufferspeichers beansprucht. Das ist natürlich keine Garantie dafür, dass das beim nächsten Mal auch so sein wird. Vielleicht verursacht dann gerade irgendein Benutzer oder ein Prozess große Änderungen im /home-Verzeichnis.

Auch wenn Sie Logical Volumes verwenden, um darin den Datenträger einer virtuellen Maschine zu speichern (siehe [Kapitel 40](#), »KVM«), können Sie LVM-Snapshots verwenden, um das Logical Volume sicher in eine komprimierte Image-Datei zu kopieren. Das folgende Script benennt zuerst das eventuell schon vorhandene Backup `image.lzo` in `old-image.lzo` um. Anschließend erstellt es den Snapshot `snap` des Logical Volumes `/dev/vg1/lv1`, das den virtuellen Datenträger enthält.

Logical Volume  
als Image  
auslesen

Das Image wird nun mit `cat` ausgelesen und mit `lzop` komprimiert. Dank `ionice -c 3` wird dieses IO- und CPU-intensive Kommando ausgeführt, ohne alle anderen laufenden Prozesse allzu stark zu beeinträchtigen.

Das komprimierte Image wird anschließend mit `curl` auf einen Backup-Server hochgeladen. Die Option `--limit-rate` limitiert dabei die Übertragungsgeschwindigkeit. Das stellt sicher, dass die Netzwerkkapazitäten des Servers nicht vollständig durch das Backup-Script blockiert werden.

```
#!/bin/bash
mv /backup/image.lzo /backup/old-image.lzo
lvcreate -s -L 2G -n snap /dev/vg1/lv1
ionice -c 3 cat /dev/vg1/snap | lzop -c > /backup/image.lzo
lvremove -f /dev/vg1/snap

# Image per FTP auf einen Backup-Server hochladen
pw=u12345:2n34jkkj546wqdsr
ftp=u12345.backup-server.de
curl --limit-rate 8m -T /backup/image.lzo -u $pw ftp://$ftp/image.lzo
```

Wozu das Rad neu erfinden, wenn es im Internet fertige Backup-Scripts gibt? Eine Internetsuche nach *linux backup script* liefert unzählige Ergebnisse. Das Problem besteht allerdings darin, die Spreu vom Weizen zu trennen. Viele Scripts sind für eine ganz spezifische Aufgabenstellung konzipiert und kaum allgemeingültig verwendbar. Aber zum Glück gibt es auch positive Ausnahmen: Bei meiner Arbeit bewährt hat sich beispielsweise das `bash`-Script `Tartarus`. Es verpackt in ca. 600 Zeilen Code die folgenden Funktionen:

Tartarus

- ▶ tar-Archivformat
- ▶ Speicherung der Backups wahlweise im lokalen Dateisystem, auf einem FTP-Server oder über ein eigenes Kommando (z. B. mit ssh)
- ▶ inkrementelle Backups (optional)
- ▶ LVM-Snapshots (optional)
- ▶ Verschlüsselung mit GPG (optional)

Tartarus wird unter anderem vom Webhosting- und Root-Server-Provider Hetzner empfohlen. Sie finden das Script sowie eine brauchbare Dokumentation hier:

<http://wertarbyte.de/tartarus.shtml>

## 36.8 Backups auf S3-Speicher

S3 steht für *Simple Storage Service* und ist ein Puzzlestück der Amazon Web Services (AWS). S3 ist ein vergleichsweise kostengünstiger Cloud-Dienst von Amazon zur Speicherung von Dateien. S3 ist vor allem als externer Backup-Speicher sehr beliebt, besonders wenn es darum geht, zusätzlich zu einem lokalen Backup auch ein geografisch vom Server getrenntes Backup durchzuführen. Das gibt Ihnen die Gewissheit, auf Ihre Daten selbst dann noch zugreifen zu können, wenn Ihre Hosting-Firma pleitegegangen ist oder ihr Rechenzentrum durch eine Naturkatastrophe zerstört wurde.

Das Preismodell von Amazon S3 ist außergewöhnlich unübersichtlich. Sie bezahlen nicht nur für die gespeicherten Datenmengen an sich, sondern auch für Transfers, wobei zwischen GET- und PUT-Anforderungen unterschieden wird, also Down- und Uploads. Lassen Sie sich von der verwirrenden Darstellung des Preismodells aber nicht irritieren: Im Vergleich zu Diensten wie Dropbox ist S3 bei einer vernünftigen Nutzung ausgesprochen billig. Sie sollten aber bei der Programmierung von Backup-Scripts darauf achten, unnötige Transfers zu minimieren.

### S3-Speicher einrichten

**Buckets** Sie können Amazon S3 mit Datenmengen bis zu 5 GiB für ein Jahr kostenlos testen. Sie müssen sich allerdings auch für den Test mit Name, Adresse, einer funktionierenden Telefonnummer und einer Kreditkarte anmelden. Die Kontoeinrichtung dauert nur wenige Minuten. Anschließend müssen Sie in der S3-Weboberfläche zumindest einen sogenannten »Bucket« (wörtlich »Eimer«) anlegen, gewissermaßen ein Grundverzeichnis innerhalb von S3. Der Bucket-Name muss innerhalb der gesamten S3-Cloud eindeutig sein. Es empfiehlt sich deswegen, den Firmen- oder Domainnamen miteinzubeziehen, z. B. `meinefirma.erster.test` oder `info.kofler.backup`.



Nach diesen ein wenig langwierigen Vorbereitungen können Sie nun endlich im Terminal loslegen. Dort brauchen Sie das Python-Kommando `aws`. Zu diesem kommen Sie, indem Sie zuerst das Python-Paketverwaltungswerkzeug `pip` installieren und damit dann das Paket `awscli`:

awscli-Paket  
installieren

```
root# apt-get/dnf/yum/zypper install python-pip
root# pip install awscli
```

Wenn Sie bei der Script-Programmierung Python 3 vorziehen, sieht die Vorgehensweise so aus:

```
root# apt-get/dnf/yum/zypper install python3-pip
root# pip3 install awscli
```

Die Verbindung zu S3 erfolgt verschlüsselt. Bevor Sie im Terminal das Kommando `aws` für die weitere Nutzung konfigurieren können, müssen Sie sich auf der AWS-Seite *Identity & Access Management* (IAM) einen Benutzer und eine Gruppe einrichten. Die Gruppe verbinden Sie mit der *Policy* `AmazonS3FullAccess` und mit dem neuen Benutzer. Sie finden die AWS-Seite unter:

aws-  
Konfiguration

<https://console.aws.amazon.com/iam>

Im Terminal führen Sie nun `aws configure` aus und geben dabei die *Access Key ID* und den *Secret Access Key* an, den Sie beim Einrichten des Benutzers erhalten haben. `aws` speichert diese Daten und die weiteren Optionen in `.aws/credentials` und `.aws/config`.

```
root# aws configure
AWS Access Key ID [None]: AKxxxxxxx
AWS Secret Access Key [None]: ZRxxxxxxxxxxxxxxxxxx
Default region name [None]: eu-central-1
Default output format [None]:
```

Nicht ganz einfach ist die korrekte Angabe der Region. Beim Anlegen neuer Buckets können Sie wählen, wo diese physikalisch gespeichert werden sollen. Für S3-Kunden in Deutschland ist Frankfurt der beste Ort. In welcher Form erwartet S3 nun aber die Angabe der Region? Am besten ist es, die Region vorerst leer zu lassen. Anschließend ermitteln Sie mit `aws s3api get-bucket-location` die Region und wiederholen dann `aws configure`:

```
root# aws s3api get-bucket-location --bucket meinefirma.erster.test
"LocationConstraint": "eu-central-1"
```

## Das aws-Kommando

**aws s3 ls** `aws s3 ls` liefert eine Liste aller eigenen Buckets:

```
root# aws s3 ls
2017-10-05 21:53:08 meinefirma.erster.test
2017-10-06 07:58:11 linux.buch.test
```

**aws s3 cp** Mit `aws s3 cp` laden Sie eine lokale Datei hoch. Die folgenden Zeilen erstellen zuerst ein komprimiertes Backup des `/etc`-Verzeichnisses und übertragen dieses dann in einen Bucket, der in der Form `s3://bucketname` angegeben wird. Natürlich ist auch ein Transfer in die umgekehrte Richtung möglich (drittes Kommando).

```
root# tar czf etc.tgz /etc
root# aws s3 cp etc.tgz s3://linux.buch.test
upload: ./etc.tgz to s3://linux.buch.test/etc.tgz
root# aws s3 cp s3://linux.buch.test/etc.tgz etc-copy.tgz
download: s3://linux.buch.test/etc.tgz to ./etc-copy.tgz
```

Ein wenig verwirrend ist der Umgang mit Verzeichnissen. Innerhalb eines S3-Buckets gibt es offiziell nämlich gar keine Verzeichnisse. Vielmehr befinden sich alle Dateien ohne eine hierarchische Ordnung direkt im Bucket. Daher fehlen Kommandos wie `mkdir`, `rmdir` oder `cd`.

Bucket-Dateien dürfen aber Namen in der Form von `v1/v2/name` haben. `ls` verarbeitet derartige Dateinamen so, als würde es sich dabei um Verzeichnisse handeln. Auch die S3-Weboberfläche zerlegt derartige Dateinamen in virtuelle Verzeichnisse.

```
root# touch tst.txt
root# aws s3 cp tst.txt s3://linux.buch.test/v1/v2/tst.txt
root# aws s3 ls s3://linux.buch.test --recursive
2017-10-06 08:12:47 2595826 etc.tgz
2017-10-06 08:30:23 0 v1/v2/tst.txt
root# aws s3 ls s3://linux.buch.test/v1/v2/
2017-10-06 08:30:23 0 tst.txt
```

**aws s3 sync** Anstatt einzelne Dateien hochzuladen, können Sie mit `aws s3 sync` ganze Verzeichnisbäume synchronisieren:

```
root# aws s3 sync ein_lokales_verzeichnis/ s3://linux.buch.test
```

Wenn Sie das Kommando zum ersten Mal ausführen, werden alle Dateien des lokalen Verzeichnisses hochgeladen. Wiederholen Sie das Kommando später, werden nur noch die Änderungen durchgeführt, wobei lokal gelöschte Dateien im S3-Bucket standardmäßig erhalten bleiben (es sei denn, Sie verwenden die Option `--delete`).

Neben `ls`, `cp` und `sync` gibt es nur fünf weitere Kommandos: `mv` verschiebt eine Datei, `rm` löscht eine Datei, `mb` erzeugt einen Bucket und `rb` löscht ihn wieder. Mit `website` können Sie schließlich aus einem Bucket eine statische Webseite machen. Das ist dann zweckmäßig, wenn Sie Dateien öffentlich zum Download anbieten möchten. Eine detaillierte Beschreibung der `aws-s3`-Kommandos mit all ihren Optionen finden Sie hier:

Weitere `aws-s3`-Kommandos

<https://docs.aws.amazon.com/cli/latest/reference/s3/index.html>

### Verschlüsselung und Beispiel

Nach den Enthüllungen von Edward Snowden müssen Sie leider davon ausgehen, dass die NSA und andere Geheimdienste uneingeschränkten Zugriff auf Ihre Daten in der Cloud haben – ob das den Cloud-Betreibern nun passt oder nicht. Dass Ihre Daten beim Upload verschlüsselt übertragen werden, ist zwar an sich erfreulich, aber letzten Endes irrelevant. Wenn Sie Wert darauf legen, dass Ihre Dateien nur von Ihnen gelesen werden können, müssen Sie sie *vor* dem Upload verschlüsseln.

Ich gehe dabei oft so vor, dass ich auf einem Server zuerst ein lokales Backup-Verzeichnis erstelle. Dort führe ich die Backups durch, die mir zweckmäßig erscheinen, wobei ich bei dieser Gelegenheit bereits alle Backup-Dateien verschlüssele. In einem zweiten Schritt synchronisiere ich dann das lokale Backup-Verzeichnis in einen S3-Bucket, auf einen FTP-Server des Hosting-Providers etc.

Im folgenden Beispiel gehe ich davon aus, dass es das lokale Backup-Verzeichnis `/local-backup` gibt. Es ist zweckmäßig, dieses in einem eigenen Logical Volume mit einem eigenen Dateisystem einzurichten. Alle Backup-Skripts und Schlüssel befinden sich in `/etc/myscripts`. Zum Ver- und Entschlüsseln wird eine *Passphrase* aus einer Datei verwendet.

Ver- und Entschlüsseln

```
root# makepasswd --chars 100 > /etc/myscripts/key
root# chmod 400 /etc/myscripts/key
root# cat /etc/myscripts/key
ISMLX...AaxdI
```

Zum Verschlüsseln verwende ich das Kommando `gpg`, das in die Skripts `mycrypt` und `myuncrypt` verpackt ist. Die Verschlüsselung erfolgt der Einfachheit halber symmetrisch.

```
#!/bin/sh -e
# Datei /etc/myscript/mycrypt
# Verwendung: mycrypt < plain > crypted
gpg -c --batch --no-use-agent --cipher-algo AES256 --compress-algo none \
  --passphrase-file /etc/myscripts/key --passphrase-repeat 0
```

```
#!/bin/sh -e
# Datei /etc/mycrypt/myuncrypt
# Verwendung: myuncrypt < crypted > plain
gpg -d --batch --no-tty -q --no-use-agent --cipher-algo AES256 \
  --compress-algo none --passphrase-file /etc/myscripts/key \
  --passphrase-repeat 0
```

### Die hohe Kunst des Verschlüsseln

Das hier präsentierte Verschlüsselungskonzept ist einfach, aber sicherheitstechnisch alles andere als perfekt. Solange die Datei `/etc/myscripts/key` nicht in falsche Hände gerät, ist die so errichtete Hürde ein ausreichender Schutz vor Gelegenheits-Hackern – mehr nicht! Wenn Sie höhere Sicherheitsansprüche stellen, müssen Sie entsprechende Fachliteratur konsultieren.

**Lokales Backup** Das täglich um 2:00 Uhr ausgeführte Cron-Script `backup-cms` erstellt ein Backup der MySQL- oder MariaDB-Datenbank `cms`, komprimiert und verschlüsselt dieses und speichert es in `/local-backup/mysql` unter dem Namen `cms-nn.sql.gz.crypt`. Dabei ist `nn` der Tag im Monat (1 bis 31). Damit gibt es immer individuelle tägliche Backups, die einen Monat zurückreichen.

```
#!/bin/sh -e
# Datei /etc/myscripts/backup-cms
db=cms
opts='--skip-opt --single-transaction --create-options --quick \
  --extended-insert --disable-keys --add-drop-table'
day=$(date +%d)
fname="/local-backup/$db-$day.sql.gz.crypt"
mysqldump --defaults-file=/root/.my.cnf $mysqlopt $db \
  | gzip -c \
  | /etc/myscripts/mycrypt > $fname
```

**S3-Sync** Ein weiteres Cron-Script, das täglich um 3:00 Uhr ausgeführt wird, muss nun nur noch die Backup-Dateien in `/local-backup` in einen S3-Bucket synchronisieren:

```
#!/bin/sh -e
aws s3 sync /local-backup/ s3://meine.firma.backups
```

**Test** Jetzt müssen Sie die Prozedur natürlich testen: Funktionieren die automatischen Backups? Sind Sie auch auf einem anderen Server in der Lage, die Backup-Datei aus dem S3-Speicher herunterzuladen, zu entschlüsseln, zu entkomprimieren und dann in einen MySQL-Server hochzuladen? Wichtig ist natürlich, dass Sie über sichere Kopien der S3-Zugangsdaten sowie der Schlüsseldatei `/etc/myscripts/key` verfügen!

# Kapitel 37

## Firewalls

Die Überschrift für dieses Kapitel ist ein wenig plakativ – ganz einfach deswegen, weil fast jeder etwas mit dem Begriff »Firewall« anfangen kann. Tatsächlich geht es in diesem Kapitel aber nicht nur darum, einen Filter für Netzwerkpakete einzurichten, sondern auch um Netzwerkgrundlagen und um andere Techniken zur Absicherung. Daher lernen Sie in diesem Kapitel auch einige elementare Werkzeuge kennen, um den aktuellen Zustand des Netzwerks zu analysieren, z. B., um offene Ports zu finden. Ein Abschnitt zur TCP-Wrapper-Bibliothek zeigt zudem, wie der Zugriff auf bestimmte Netzwerkdienste durch einfache Regeln eingeschränkt werden kann.

### 37.1 Netzwerkgrundlagen und -analyse

Bevor Sie Ihren Rechner absichern können, müssen Sie eine Vorstellung davon gewinnen, wie die Netzwerkdienste funktionieren, welche Dienste gerade laufen, welche Ports offen sind etc. Dieser Abschnitt beschäftigt sich daher mit TCP/IP-Grundlagen und beschreibt einige Programme, um den aktuellen Netzwerkstatus zu analysieren und beispielsweise alle gerade aktiven Netzwerkverbindungen aufzulisten. Vorweg fasst Tabelle 37.1 die wichtigsten Abkürzungen zusammen.

| Abkürzung | Bedeutung                         |
|-----------|-----------------------------------|
| DNS       | Domain Name Service               |
| HTTP      | Hypertext Transfer Protocol       |
| ICMP      | Internet Control Message Protocol |
| IP        | Internet Protocol                 |
| NFS       | Network File System               |
| TCP       | Transmission Control Protocol     |
| UDP       | User Datagram Protocol            |

**Tabelle 37.1** Netzwerk-Glossar

**Internet Protocol** Praktisch alle gängigen Netzwerkdienste basieren auf IP-Paketen. Wenn beispielsweise ein Internetbenutzer per FTP auf Ihren Rechner zugreifen möchte, startet er dazu auf seinem Rechner einen FTP-Client. Dieser Client sendet ganz spezielle IP-Pakete an Ihren Rechner. Wenn auf Ihrem Rechner ein FTP-Server installiert ist, erhält dieser die IP-Pakete und reagiert auf die Anfrage, indem er selbst IP-Pakete an den Client zurücksendet.

Neben den eigentlichen Daten enthalten IP-Pakete unter anderem vier wesentliche Informationen: die Absender-IP-Adresse, den Absender-Port, die Zieladresse und den Ziel-Port. Diese Daten geben an, woher das Paket kommt und wohin es gehen soll.

**IP-Adressen und -Ports** Die Bedeutung der IP-Adresse sollte klar sein (siehe auch [Kapitel 26](#), »Netzwerkkonfiguration«). IP-Ports werden dazu verwendet, um verschiedene Dienste zu identifizieren. Beispielsweise wird zur Anforderung eines WWW-Dokuments üblicherweise der Port 80 verwendet, bzw. Port 443, wenn das HTTPS-Protokoll zum Einsatz kommt. Bei Port-Nummern handelt es sich um 16-Bit-Zahlen. Die Ports bis 1024 gelten als privilegiert und sind für Server-Dienste reserviert, z. B. für den HTTP-Server. Die verbleibenden Ports können an sich von Clients eingesetzt werden, allerdings gibt es auch hier eine Reihe von Nummern, die nicht verwendet werden sollten, weil sie oft schon für bestimmte Zwecke reserviert sind.

Zu vielen IP-Port-Nummern sind in `/etc/services` Alias-Namen definiert. [Tabelle 37.2](#) führt die wichtigsten Port-Nummern mit den üblicherweise gültigen Namen und einer kurzen Erklärung auf.

| Name              | Port   | Funktion                           |
|-------------------|--------|------------------------------------|
| ftp               | 20, 21 | FTP                                |
| ssh               | 22     | SSH                                |
| telnet            | 23     | Telnet                             |
| smtp              | 25     | E-Mail                             |
| domain            | 53     | DNS                                |
| bootps und bootpc | 67, 68 | DHCP                               |
| http              | 80     | Web                                |
| pop3              | 110    | E-Mail                             |
| portmap           | 111    | Portmap (für NFS)                  |
| ntp               | 123    | Zeit (Network Time Protocol)       |
| netbios-ns        | 137    | Microsoft/NetBIOS Name Service     |
| netbios-dgm       | 138    | Microsoft/NetBIOS Datagram Service |

**Tabelle 37.2** Wichtige IP-Ports

| Name         | Port      | Funktion                                 |
|--------------|-----------|--|
| netbios-ssn  | 139       | Microsoft File Sharing (SMB, Samba)      |
| imap         | 143       | E-Mail                                   |
| ldap         | 389       | LDAP                                     |
| –            | 427       | Apple Filing Protocol (AFP)              |
| https        | 443       | Web (verschlüsselt)                      |
| microsoft-ds | 445       | CIFS-Dateisystem (SMB, Samba)            |
| ssmtp        | 465       | E-Mail verschlüsselt (veraltet)          |
| printer      | 515       | Drucken mit LPD/LPR                      |
| –            | 548       | Apple Filing Protocol (AFP)              |
| –            | 587       | E-Mail verschlüsselt                     |
| ipp          | 631       | Drucken mit IPP/CUPS                     |
| rmi          | 1099      | Remote Method Incovation (Java)          |
| pptp         | 1723      | PPTP/VPN                                 |
| nfs          | 2049      | NFS                                      |
| –            | 3128      | Squid (Web-Proxy)                        |
| mysql        | 3306      | MySQL- oder MariaDB-Datenbank-Server     |
| –            | 5353      | Netzkonfiguration durch Zeroconf/Bonjour |
| –            | 5999–6003 | X-Display                                |
| –            | 9100      | HP-JetDirect-Netzwerkdrucker             |

**Tabelle 37.2** Wichtige IP-Ports (Forts.)

Es gibt unterschiedliche Protokolle für IP-Pakete: Die meisten Internetdienste verwenden TCP. Dieses Protokoll verlangt eine Bestätigung des Empfangs. Es gibt aber auch Protokolle, die keine derartige Bestätigung erwarten, nämlich ICMP (wird z. B. von ping verwendet) und UDP (wird z. B. von DNS und NFS verwendet).

IP-Protokolle

IP-Pakete können durch lokale Programme erzeugt werden oder von außen – also über Netzwerkschnittstellen – in den Rechner kommen. Der Kernel muss nun entscheiden, was mit den Paketen passieren soll. Er kann die Pakete verwerfen oder an laufende Programme bzw. an andere Schnittstellen weiterleiten. Dabei können alle oben beschriebenen Paketmerkmale als mögliche Entscheidungskriterien dienen. Um einen Paketfilter zu realisieren, brauchen Sie also eine Möglichkeit, dem Kernel Regeln mitzuteilen, wie er mit bestimmten IP-Paketen verfahren soll. Dazu dient das Kommando `iptables`, das in [Abschnitt 37.5](#), »Firewall mit iptables selbst gebaut«, näher erläutert wird.

IP-Paketfilter

### Aktive Netzwerk-Ports ermitteln

Das Funktionsprinzip der meisten Netzwerkdienste sieht so aus, dass diese einen bestimmten Port überwachen. Treffen für diesen Port IP-Pakete ein, kümmert sich der Dienst um deren Verarbeitung und Beantwortung. Pakete, die an nicht überwachte Ports adressiert sind, werden einfach ignoriert und stellen insofern auch keine Gefahr dar. Um die Gefährdung eines Rechners abzuschätzen, ist es daher zweckmäßig, eine Liste der überwachten Ports zu ermitteln. Umgekehrt wird auch ein Angreifer als Erstes versuchen, die aktiven Ports herauszufinden.

**netstat** Um festzustellen, welche Netzwerkaktivitäten auf dem lokalen Rechner stattfinden, ist das Kommando `netstat` aus dem Paket `net-tools` ein praktisches Hilfsmittel. Je nachdem, mit welchen Optionen es aufgerufen wird, liefert es eine Fülle unterschiedlicher Informationen. `netstat` kann von gewöhnlichen Benutzern ausgeführt werden, einige Optionen erfordern aber `root`-Rechte. Ich beschränke mich hier auf eine beispielhafte Vorstellung des Kommandos. Eine Referenz der zahlreichen Optionen gibt bei Bedarf `man netstat`.

Eine leicht zu merkende Kombination von Optionen ergibt `-tulpen`: `netstat` berücksichtigt dann TCP- und UDP-Verbindungen (`-t` und `-u`), zeigt aktive Sockets (`-l` für *listening*) und die Prozessnummer des Programms an (`-p`), reichert das Ergebnis mit diversen Zusatzspalten an (`-e`, *extended*) und zeigt IP-Adressen numerisch an (`-n`). Das folgende, aus Platzgründen gekürzte Ergebnis ist auf einem Server entstanden, auf dem Apache, Postfix, Dovecot und ein MySQL-Server laufen. Aus dem Ergebnis geht hervor, an welchen Ports Programme auf Anfragen warten (Status LISTEN).

```
root# netstat -tulpen
Aktive Internetverbindungen (Nur Server)
Proto Recv Send Local Address Foreign Addr. State User PID/Program name
tcp 0 0 127.0.0.1:10023 0.0.0.0:* LISTEN 0 1000/postgrey.pid
tcp 0 0 127.0.0.1:3306 0.0.0.0:* LISTEN 105 963/mysqld
tcp 0 0 0.0.0.0:587 0.0.0.0:* LISTEN 0 1158/master
tcp 0 0 127.0.0.1:783 0.0.0.0:* LISTEN 0 1019/spamd.pid
tcp 0 0 0.0.0.0:143 0.0.0.0:* LISTEN 0 896/dovecot
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN 0 1354/sshd
tcp 0 0 0.0.0.0:25 0.0.0.0:* LISTEN 0 1158/master
tcp 0 0 127.0.0.1:12345 0.0.0.0:* LISTEN 111 1053/opendkim
tcp 0 0 0.0.0.0:993 0.0.0.0:* LISTEN 0 896/dovecot
tcp6 0 0 :::1:10023 :::* LISTEN 0 1000/postgrey.pid
tcp6 0 0 :::587 :::* LISTEN 0 1158/master
...
tcp6 0 0 :::993 :::* LISTEN 0 896/dovecot
```

Wenn Sie nur an aktuell aktiven TPC- und UDP-Verbindungen interessiert sind, lassen Sie die Option `-l` weg. Die lokale Adresse habe ich im folgenden Listing durch `n.n.n.n` ersetzt, und ich habe einige Spalten aus Platzgründen entfernt:



```

root# netstat -tuep
Active Internet connections (w/o servers)
Proto Local Addr. Foreign Address      State      User      PID/Program
tcp    n.n.n.n:143 194.118.35.145:53013 ESTABLISHED 108      2676/imap-login
tcp    n.n.n.n:22  61.216.145.154:36084 ESTABLISHED 0        2884/sshd: [acc
tcp    n.n.n.n:22  61.216.145.154:9224  FIN_WAIT1  0        -
tcp    n.n.n.n:22  194.118.35.145:53355 ESTABLISHED 0        2764/sshd: kofl
tcp    n.n.n.n:143 194.118.35.145:53003 ESTABLISHED 108      2674/imap-login
tcp    n.n.n.n:143 194.118.35.145:52989 ESTABLISHED 108      2664/imap-login
tcp    n.n.n.n:143 194.118.35.145:52991 ESTABLISHED 108      2666/imap-login
tcp    n.n.n.n:143 194.118.35.145:52988 ESTABLISHED 108      2662/imap-login
tcp6   n.n.n.n:443 95.143.172.218:56266 TIME_WAIT  0        -

```

Wenn Sie herausfinden möchten, welche Programme TCP- bzw. UDP-Ports nutzen, hilft auch das Kommando `lsof`. In der Form `lsof -i [protokoll]@[hostname][:port]` liefert es eine Liste von Prozessen, die die angegebenen Netzwerkressourcen nutzen. Die beiden folgenden Kommandos zeigen alle Prozesse, die das Protokoll UDP bzw. den Port 22 nutzen:

```

root# lsof -i udp
ntpd      3696      ntp      16u IPv4  9026      UDP *:ntp
ntpd      3696      ntp      17u IPv6  9028      UDP *:ntp
ntpd      3696      ntp      18u IPv6  9031      UDP ip6-localhost:ntp

portmap   4745  daemon   3u IPv4  12931     UDP *:sunrpc
rpc.statd 4764  statd    5u IPv4  12962     UDP *:700
rpc.statd 4764  statd    7u IPv4  12970     UDP *:39146
...
root# lsof -i :22
COMMAND  PID USER  FD  TYPE DEVICE SIZE NODE NAME
sshd     5559 root   3u  IPv6  14097     TCP *:ssh (LISTEN)
sshd     7729 root   3r  IPv6  33146     TCP mars.sol:ssh->merkur.sol:45368
                                           (ESTABLISHED)

```

`netstat` und `lsof` können nur auf dem lokalen Rechner ausgeführt werden und stehen einem Angreifer normalerweise nicht zur Verfügung. Dieser greift stattdessen auf sogenannte Port-Scanner zurück. Solche Programme senden Pakete an die wichtigsten Ports eines Rechners und finden anhand der Antwort heraus, welche Ports aktiv sind. Das hier vorgestellte Kommando `nmap` ist das bekannteste, aber keineswegs das einzige derartige Programm. Bei den meisten Distributionen muss es vor der ersten Verwendung installiert werden.

Mit der Option `-sV` versucht `nmap` herauszufinden, welches Programm in welcher Version für die aktiven Ports zuständig ist. Die Option `-O` (Oh, nicht Null) bewirkt, dass `nmap` auch das laufende Betriebssystem zu erkennen versucht. Die folgenden Zeilen zeigen die Ergebnisse für einen Web- und Mail-Server, dessen IP-Adresse ich durch `n.n.n.n` ersetzt habe. Auch sonst habe ich die Ausgabe aus Platzgründen gekürzt.

```

root# nmap -sV -O ein-hostname
Starting Nmap 6.40 ( http://nmap.org ) at 2017-05-26 15:42 CEST
Nmap scan report for ein-hostname (n.n.n.n)
Host is up (0.00055s latency).
rDNS record for 138.201.20.187: host1.ein-hostname
Not shown: 993 closed ports

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      (protocol 2.0)
25/tcp    open  smtp     Postfix smtpd
80/tcp    open  http     Apache httpd 2.4.7 ((Ubuntu))
143/tcp   open  imap     Dovecot imapd
443/tcp   open  ssl/http Apache httpd 2.4.7 ((Ubuntu))
587/tcp   open  smtp     Postfix smtpd
993/tcp   open  ssl/imap Dovecot imapd
1 service unrecognized despite returning data.
...
MAC Address: 52:54:00:B9:3D:5F (QEMU Virtual NIC)
No exact OS matches for host
Nmap done: 1 IP address (1 host up) scanned in 28.95 seconds

```

Zu `nmap` existiert auch eine grafische Benutzeroberfläche, die sich je nach Distribution im Paket `zenmap` oder `nmap-frontend` befindet.

Bitte beachten Sie, dass das obige Beispiel die Möglichkeiten von `nmap` nicht einmal andeutungsweise repräsentieren kann. Ein guten Überblick über die unzähligen `nmap`-Optionen gibt die rund 20-seitige `man`-Seite. Und wenn Ihnen das noch nicht reicht, beschreibt das Buch *Nmap Network Scanning* (2009 im Eigenverlag des `nmap`-Entwicklers erschienen) auf beinahe 500 Seiten alle erdenklichen Grundlagen und Details des Network-Scannings. Rund die Hälfte dieses Buchs können Sie auf der `nmap`-Webseite sogar kostenlos lesen:

<https://nmap.org/book>

### Führen Sie nie ungefragt Port-Scans für fremde Server durch!

Ein Port-Scan wird von vielen Administratoren als Einbruchversuch gewertet. Adressieren Sie mit Programmen wie `nmap` nie ungefragt einen fremden Rechner! `nmap` ist aber ein praktisches Hilfsmittel, um Schwächen im eigenen Netzwerk zu erkennen.

## 37.2 Basisabsicherung von Netzwerkdiensten

Der vorige Abschnitt hat gezeigt, wie Sie sich rasch einen Überblick über die laufenden Netzwerkdienste verschaffen. Der nächste Schritt besteht nun darin, die Dienste möglichst gut abzusichern:

- ▶ Deinstallieren Sie alle Netzwerkdienste, die Sie nicht brauchen. Nicht installierte Programme laufen nicht und können daher keine Gefahr darstellen.
- ▶ Bei den erforderlichen Netzwerkdiensten reicht es vielfach aus, ihren Zugriff auf bestimmte Clients einzuschränken, z. B. aus dem lokalen Netzwerk. Es ist beispielsweise selten notwendig, dass ein Drucker-Server seine Dienste im Internet anbietet!

Bei Apache, Samba, MySQL und zahlreichen weiteren »großen« Diensten muss die Absicherung in der jeweiligen Konfigurationsdatei erfolgen. Erfreulicherweise gibt es aber auch eine Reihe von Netzwerkdiensten, die für die Zugriffskontrolle auf die TCP-Wrapper-Bibliothek zurückgreifen. Das ermöglicht eine zentrale Konfiguration (siehe den folgenden Abschnitt).

- ▶ Notwendige Netzwerkdienste sollten mit minimalen Rechten ausgeführt werden. Darum kümmert sich das Init-System Ihrer Distribution. Soweit es möglich und sinnvoll ist, startet das Init-System die Dienste ohne `root`-Rechte in einem für den Dienst konzipierten Account oder in einer `chroot`-Umgebung, die den Zugriff auf Dateien außerhalb des `chroot`-Verzeichnisses verhindert.
- ▶ Als zusätzliche Schutzebene empfiehlt sich eine Paketfilter-Firewall, die durch Regeln Pakete, die aus dem Internet kommen, für diverse Dienste von vornherein blockiert (siehe [Abschnitt 37.3](#), »Firewalls – eine Einführung«).
- ▶ Kein Programm ist fehlerfrei. Programmfehler können es einem Angreifer ermöglichen, durch die gezielte Übertragung manipulierter Netzwerkpakete das Programm zum Absturz zu bringen oder gar eigene Befehle auszuführen. Um das daraus entstehende Risiko zu minimieren, kann der Kernel die Ausführung von Programmen anhand von Regeln überwachen. Diese Vorgehensweise wird als *Mandatory Access Control* bezeichnet, kurz *MAC*. Unter Linux sind zu diesem Zweck die Verfahren *SELinux* und *AppArmor* populär, die ich Ihnen in [Kapitel 38](#) vorstelle.

Die sichere Konfiguration eines Rechners ist keine einmalige Arbeit, sondern ein stetiger Prozess. Nur regelmäßige Software-Updates halten die Software auf Ihrem Rechner auf dem aktuellen Stand. Empfehlenswert ist auch ein regelmäßiger Blick in die Logging-Dateien Ihres Rechners.

Updates, Logging

## TCP-Wrapper-Bibliothek

Gerade auf einem LAN-Server ist es selten zweckmäßig, alle Netzwerkdienste global verfügbar zu machen. Es reicht aus, wenn die Dienste im lokalen Netzwerk verwendet werden können. Eine Reihe von Netzwerkdiensten greift für diese Basisabsicherung auf die sogenannte TCP-Wrapper-Bibliothek zurück. Dazu zählen insbesondere der SSH- und der NFS-Server. Auch Dienste, die bei Bedarf durch `xinetd` gestartet werden, profitieren von der TCP-Wrapper-Bibliothek (siehe [Abschnitt 24.7](#), »Internet Service Daemon«).

### `/etc/hosts.allow` und `hosts.deny`

Die Dateien `/etc/hosts.allow` und `/etc/hosts.deny` steuern, von welchem Rechner aus welche Dienste verwendet werden dürfen. Die Einstellungen gelten nur für Netzwerkdienste, die die TCP-Wrapper-Bibliothek bzw. das Kommando `tcpd` für die Zugriffskontrolle verwenden. Standardmäßig sind beide Dateien leer, d. h., es gelten keinerlei Einschränkungen.

Die TCP-Wrapper-Bibliothek wertet zuerst `hosts.allow` aus: Wenn der Zugriff dort explizit gestattet wird, ist die Kontrolle erledigt. Andernfalls wird auch `hosts.deny` ausgewertet: Ist der Zugriff dort verwehrt, erhält der Client eine Fehlermeldung. Vorsicht: In allen Fällen, die weder durch `allow`- noch durch `deny`-Regeln erfasst sind, wird der Zugang gewährt!

Eine möglichst sichere Konfiguration erreichen Sie dadurch, dass Sie als Erstes in `/etc/hosts.deny` durch `all:all` generell den Start jedes Netzwerkdienstes verbieten. Die `spawn`-Anweisung bewirkt darüber hinaus, dass jeder Versuch, irgendeinen Dienst zu starten, in der Datei `/var/log/deny.log` protokolliert wird.

`/var/log/deny.log` verrät Ihnen von nun an, wer wann versucht, einen Netzwerkdienst des Rechners zu nutzen. Nicht jeder Versuch muss zwangsläufig einen Angriff darstellen. Es kann auch sein, dass sich jemand bei der Eingabe des Hostnamens oder der IP-Adresse vertippt hat.

```
# /etc/hosts.deny
# standardmäßig alles verbieten, jeden Verbindungsversuch
# protokollieren
ALL : ALL : spawn (echo Attempt from %h %a to %d at $(date) \
    >> /var/log/deny.log)
```

Im zweiten Schritt lassen Sie nun in `/etc/hosts.allow` die Nutzung bestimmter Dienste zu. Die unten gezeigte Beispielkonfiguration erlaubt die folgenden IPv4-Zugriffe:

- ▶ vom lokalen Rechner aus (`localhost`) den Zugriff auf alle Dienste
- ▶ von jedem Rechner aus den `ssh`-Zugriff, also auch aus dem Internet
- ▶ innerhalb des lokalen Netzes die Nutzung von NFS (`portmap` und `mountd`) und SWAT

Das Beispiel geht davon aus, dass der Server unter den Namen `mars` und `mars.sol` erreichbar ist, dass das lokale Netz im Adressraum `192.168.0.*` betrieben wird und dass alle Rechner die Domain `*.sol` nutzen. Nach demselben Muster können Sie natürlich auch andere Netzwerkdienste, die Sie zuerst global abgeschaltet haben, für das lokale Netz oder für einen beliebigen Adressbereich aktivieren:

```
# /etc/hosts.allow
# einzelne Dienste erlauben
ALL      : localhost mars mars.sol : ALLOW
sshd     : 0.0.0.0                  : ALLOW
portmap  : 192.168.0.0/24 *.sol   : ALLOW
mountd   : 192.168.0.0/24 *.sol   : ALLOW
swat     : 192.168.0.0/24 *.sol   : ALLOW
```

Die Syntax innerhalb von `hosts.allow` bzw. `hosts.deny` sollte aus den Beispielen klar werden. Jeder Eintrag besteht aus drei Teilen, die durch Doppelpunkte getrennt sind. Der erste Teil gibt den Dienst an, der zweite Teil die IP-Adresse bzw. den Netzwerknamen, der dritte Teil die resultierende Aktion. Eine genauere Syntaxbeschreibung erhalten Sie mit `man 5 hosts_access`.

Aktuelle Versionen der TCP-Wrapper-Bibliothek kommen auch mit IPv6 zurecht. Die IPv6-Adressen müssen in `hosts.allow` und `hosts.deny` in eckige Klammern gestellt werden. Die folgenden Zeilen erlauben jede IPv6-Verbindung von `localhost` sowie SSH-Verbindungen aus dem lokalen IPv6-Netz:

IPv6

```
# /etc/hosts.allow
ALL      : [::1]                  : ALLOW
sshd     : [2001:1234:789a:0471::1/64] : ALLOW
```

Mit `ldd` können Sie leicht selbst feststellen, ob ein bestimmtes Programm die TCP-Wrapper-Bibliothek (`libwrap`) nutzt. Die Ergebnisse für `cupsd` und `sshd` unter Ubuntu sehen wie folgt aus:

TCP-Wrapper-Unterstützung feststellen

```
user$ ldd /usr/sbin/cupsd | grep wrap
user$ ldd /usr/sbin/sshd | grep wrap
      libwrap.so.0 => /lib/libwrap.so.0 (0x00007f1a5f7f0000)
```

`cupsd` verwendet die Wrapper-Bibliothek also nicht, `ssh` schon.

## Start von Netzwerkdiensten ohne root-Rechte

Damit Programme wie Apache oder MySQL ihre Arbeit erledigen können, ist es nicht erforderlich, dass die Programme mit `root`-Rechten laufen. Deswegen sehen die meisten Distributionen für derartige Dienste eigene Accounts vor, deren Namen von Distribution zu Distribution variieren. Unter Ubuntu wird beispielsweise Apache im

Account `www-data` ausgeführt und darf daher nur auf Dateien zugreifen, die für diesen Account lesbar sind. Sie können sich davon mit `ps axu` überzeugen. Eine Instanz von Apache läuft übrigens doch mit `root`-Rechten. Sie ist aber nur für den Start der anderen Instanzen verantwortlich und erfüllt sonst keine Aufgaben.

```
root# ps axu | grep apache2
root    ... /usr/sbin/apache2 -k start
www-data ... /usr/sbin/apache2 -k start
www-data ... /usr/sbin/apache2 -k start
...
```

Da die Scripts des Init-Systems grundsätzlich mit `root`-Rechten ausgeführt werden, ist ein spezieller Mechanismus erforderlich, um den Netzwerkdämon in einem anderen Account zu starten. Im einfachsten Fall wird der Prozess dazu in der Form `su account-name -c daemon` gestartet.

Die meisten Netzwerkprozesse sehen allerdings ausgefeiltere Mechanismen vor, bei denen das Programm die Initialisierung mit `root`-Rechten durchführt und erst dann in einen Account mit weniger Rechten wechselt. Bei manchen Programmen wie `syslogd` gibt es eine eigene Option, um den gewünschten Account anzugeben. Bei Apache, MySQL und einigen weiteren Server-Diensten, die mehrere Instanzen starten, läuft außerdem ein Steuerungsprozess mit `root`-Rechten. Dieser Prozess erfüllt zumeist nur ganz wenige Aufgaben, in der Regel den Start bzw. das Beenden von Instanzen.

### Start von Netzwerkdiensten in einer chroot-Umgebung

Das Kommando `chroot rootdir kommando` startet das angegebene Kommando, wobei es `rootdir` als Wurzelverzeichnis verwendet. Das Kommando kann nun nur auf Dateien zugreifen, die sich innerhalb dieses Verzeichnisses befinden. Um sicherzustellen, dass das Programm aus seinem »chroot-Gefängnis« nicht ausbrechen kann, muss es zudem in einem Account mit eingeschränkten Rechten ausgeführt werden, also nicht als `root`.

In der Praxis werden Netzwerkdienste allerdings selten mit `chroot` gestartet. Vielmehr sehen viele Dienste eine spezielle Option vor, um das `chroot`-Verzeichnis direkt anzugeben. Dieses Verzeichnis muss dann alle erforderlichen Bibliotheken, Konfigurationsdateien etc. enthalten. Gegebenenfalls kopiert ein Init-Script alle erforderlichen Dateien vor dem Start dorthin.

Wenn ein Netzwerkdienst durch SELinux oder AppArmor überwacht wird und die Regeln korrekt formuliert sind, ist die Verwendung einer `chroot`-Umgebung überflüssig bzw. bietet keine zusätzliche Sicherheit. Fedora und Red Hat verzichten deswegen standardmäßig auf `chroot`-Verzeichnisse und verlassen sich stattdessen auf die SELinux-Regeln.

## 37.3 Firewall-Grundlagen

Der Begriff »Firewall« ist zwar in aller Munde, es gibt aber keine allgemein akzeptierte Definition dafür. Eine »Firewall« kann sich auf die Hardware beziehen: Dann ist damit meist ein Rechner gemeint, der zwischen dem lokalen Netz und dem Internet steht. Viele ADSL-Router enthalten elementare Firewall-Funktionen.

Oft wird mit »Firewall« aber auch ein Programm bezeichnet, das auf dem Rechner installiert wird und das bei korrekter Konfiguration die Sicherheit verbessern soll. Manche Distributionen enthalten Werkzeuge zur Konfiguration der Firewall.

In diesem Buch bezeichne ich mit dem Begriff »Firewall« die Absicherung des TCP/IP-Verkehrs durch einen Paketfilter. Ein derartiger Filter analysiert alle Netzwerkpakete, die in den Rechner kommen bzw. diesen wieder verlassen. Je nachdem, ob dabei alle Regeln eingehalten werden, dürfen die Pakete passieren oder werden blockiert. Details zur Konfiguration eines solchen Paketfilters folgen im nächsten Abschnitt. Hier geht es vorerst darum, die Terminologie zu klären.

Die Notwendigkeit von Firewalls auf privaten Rechnern ist umstritten. Die Ubuntu-Entwickler sind beispielsweise der Ansicht, dass bei einer Desktop-Installation von Ubuntu ohnedies keine Netzwerkdienste laufen, die zu schützen sind. Bei einer Minimalinstallation ist das an sich korrekt, aber dabei bleibt es selten: Sobald der Benutzer Samba installiert, um eigene Dateien über ein Netzwerkverzeichnis anderen Benutzern zur Verfügung zu stellen, gibt es den ersten von außen angreifbaren Dienst.

Firewalls für  
private Rechner

Solange der Computer zu Hause betrieben wird und den Internetzugang durch einen ADSL-Router bezieht, sorgt der Router für einen gewissen Schutz von außen – einerseits durch das üblicherweise eingesetzte NAT-Verfahren, andererseits vielleicht auch durch eine Firewall, die auf dem Router läuft. Ganz anders sieht die Sache aus, wenn Sie mit Ihrem Notebook im ungesicherten WLAN eines Hotels E-Mails abrufen. Spätestens dann sollte eine Firewall auch auf Privat-PCs selbstverständlich sein.

Bei einem Firmen-LAN ist der Wunsch nach einer guten Absicherung noch stärker ausgeprägt. Gleichzeitig bestehen auch bessere Voraussetzungen, was die Infrastruktur betrifft. In der Praxis kümmert sich oft ein eigener Rechner um den Internetzugang für die Firma und um dessen Absicherung. Alle weiteren Netzwerkdienste laufen auf anderen Rechnern. [Abbildung 37.1](#) veranschaulicht das Konzept.

Firewalls  
für lokale  
Netzwerke

Bei sehr kleinen Netzen dient manchmal *ein* Rechner gleichzeitig als Firewall und als Netzwerk-Server. Sicherheitstechnisch ist das nicht optimal, weil auf diesem Rechner zwangsläufig eine ganze Menge Netzwerkdienste laufen, die alle ein gewisses Sicherheitsrisiko darstellen.

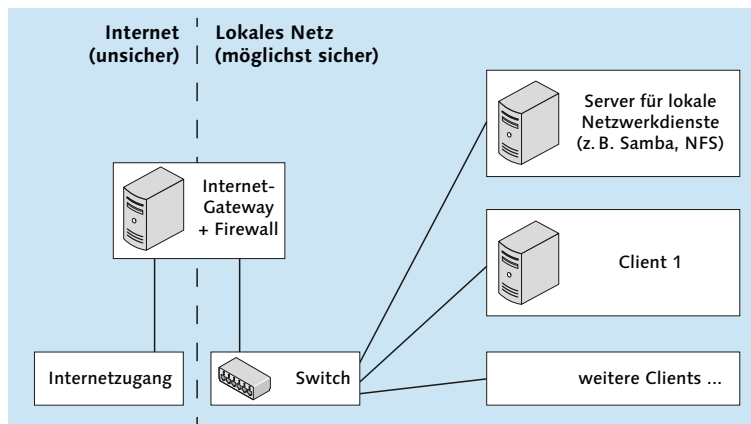


Abbildung 37.1 Firewall für lokale Netzwerke

Bei sehr großen Netzwerken gibt es oft sogar zwei Firewalls: Die erste Firewall dient nur zur Basisabsicherung, ist aber durchlässig für gewöhnliche Internetprotokolle wie HTTP oder FTP. Der Netzwerkbereich dahinter wird als *Demilitarized Zone* (DMZ) bezeichnet. Das soll zum Ausdruck bringen, dass in diesem Bereich nur eine eingeschränkte Sicherheit gegeben ist. In dieser Zone befinden sich in der Regel der Webserver sowie andere Netzwerk-Server, die öffentlich (also über das Internet) zugänglich sein müssen.

Die DMZ wird vom eigentlichen lokalen Netzwerk durch eine zweite Firewall getrennt. Erst dahinter befinden sich alle weiteren Server-Dienste, die nur für das lokale Netzwerk zuständig sind und die von außen absolut unzugänglich sein sollen. Die Konfiguration einer mehrstufigen Firewall geht allerdings weit über die Bandbreite dieses Buchs hinaus. Konfigurationsanleitungen finden Sie in speziellen Firewall-Büchern.

## Netfilter

Kernelintern kümmert sich das Netfilter-System um die Verarbeitung von Firewall-Regeln. [Abbildung 37.2](#) veranschaulicht stark vereinfacht, welche Wege IP-Pakete innerhalb des Paketfiltersystems gehen können. Eine detailliertere Abbildung finden Sie unter:

[http://open-source.arkoon.net/kernel/kernel\\_net.png](http://open-source.arkoon.net/kernel/kernel_net.png)

Die folgende Liste beschreibt ganz kurz die Stationen eines IP-Pakets im Kernel:

- **Routing:** Anhand der IP- und Port-Adresse entscheidet der Kernel, ob das Paket lokal bearbeitet werden soll oder ob es an eine Netzwerkschnittstelle und damit an einen anderen Rechner weitergeleitet werden soll. Dieser Mechanismus ist



unabhängig davon, ob sich der andere Rechner im lokalen Netz oder im Internet befindet.

- ▶ **Filter Input:** Anhand einer Reihe von Regeln wird getestet, ob das Paket zur weiteren Verarbeitung durch lokale Programme akzeptiert wird oder nicht.
- ▶ **Local Processes:** Diese Box symbolisiert alle Programme, die IP-Pakete auf dem lokalen Rechner verarbeiten bzw. die selbst neue IP-Pakete erzeugen. Dazu zählen Programme wie Apache, MySQL, Samba oder der NFS-Server.
- ▶ **Filter Output:** Anhand einer Reihe von Regeln wird getestet, ob das Paket den Kernel wieder verlassen darf.
- ▶ **Filter Forward:** Dieser Filter entscheidet, welche der Pakete, die nur weitergeleitet, aber nicht bearbeitet werden sollen, den Kernel passieren dürfen.
- ▶ **NAT Postrouting:** Falls der lokale Rechner via Masquerading anderen Rechnern einen Zugang ins Internet gewähren soll, kümmert sich diese Station um die erforderliche Manipulation der IP-Pakete.

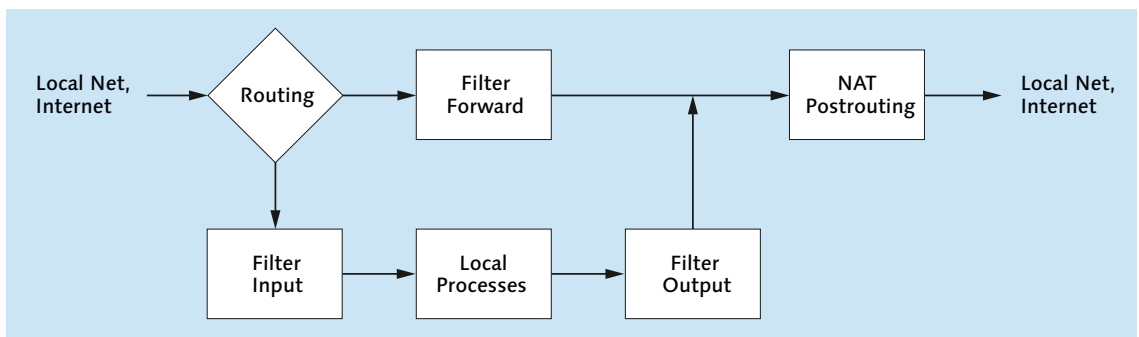


Abbildung 37.2 Vereinfachte Darstellung des iptables/netfilter-Systems

Einem Paketfilter sind in [Abbildung 37.2](#) nur die Boxen *Filter Input*, *Filter Output*, *Filter Forward* und eventuell auch *NAT Postrouting* zuzuordnen. Alle anderen Teile der Abbildung beschreiben die Netzwerkfunktionen des Kernels bzw. gewöhnliche Netzwerkdienste, die auf dem lokalen System laufen und mit dem Paketfilter nichts zu tun haben.

### Firewall-Abbildungen

Bei vielen Firewall-Abbildungen sehen Sie links das gefährliche Internet, dann die Firewall und rechts das vergleichsweise sichere lokale Netz. [Abbildung 37.2](#) entspricht nicht diesem Schema! Die Pakete, die links in den Rechner kommen, stammen sowohl aus dem lokalen Netz als auch aus dem Internet. Dasselbe gilt auch für die Pakete, die die Firewall rechts verlassen.

**Aktionen** Für die Weiterleitung von Paketen – egal, ob diese von einer Netzwerkschnittstelle kommen oder von einem lokalen Programm erzeugt wurden – ist der Kernel zuständig. Dieser hat dabei in den unterschiedlichen Stufen des Filtersystems jeweils drei Alternativen:

- ▶ **Deny:** Die Weiterleitung des Pakets wird ohne Rückmeldung abgelehnt. Das Paket wird damit gewissermaßen gelöscht. Es existiert nicht mehr weiter. Der Sender erfährt nie, was mit seinem Paket passiert ist.
- ▶ **Reject:** Die Weiterleitung wird mit einer Rückmeldung abgelehnt. Die Folgen für das Paket sind dieselben, allerdings bekommt der Sender durch ein anderes ICMP-Paket die Nachricht, dass sein Paket abgelehnt wurde.
- ▶ **Accept:** Das Paket wird weitergeleitet.

**Tabellen** Die Grundidee eines Netfilter-Systems sieht so aus: Jedes IP-Paket durchläuft verschiedene Orte im Kernel, an denen anhand von Regeln überprüft wird, ob das Paket zulässig ist. Wenn das der Fall ist, wird es weitergeleitet, sonst wird es gelöscht oder zurückgesandt. Mehrere Tabellen steuern den Netfilter:

- ▶ **Filter-Tabelle:** Diese Tabelle enthält üblicherweise das gesamte Regelsystem für den eigentlichen Paketfilter, also die »Firewall«.
- ▶ **NAT-Tabelle:** Diese Tabelle ist nur aktiv, wenn die Masquerading-Funktion des Kernels aktiviert ist. Sie ermöglicht verschiedene Formen der Adressveränderung (Network Address Translation) bei Paketen, die von außen in den Kernel eintreten bzw. diesen wieder verlassen.
- ▶ **Mangle-Tabelle:** Auch mit dieser Tabelle können IP-Pakete manipuliert werden. Die Tabelle wird für Spezialaufgaben verwendet, z. B. zur Realisierung eines transparenten Web-Proxys.

Daneben gibt es noch die Raw- und die Security-Tabelle, auf die ich in diesem Buch aber nicht eingehe.

**Regelketten (Chains)** Jede dieser Tabellen sieht wiederum mehrere Regelketten (Chains) vor:

- ▶ **Filter-Tabelle:** Input, Forward und Output
- ▶ **NAT-Tabelle:** Prerouting, Input, Output und Postrouting
- ▶ **Mangle-Tabelle:** Prerouting, Input, Forward, Output und Postrouting

Von diesen insgesamt zwölf Regelketten sind in [Abbildung 37.2](#) nur die wichtigsten dargestellt.

### Unterschiedliche Regelketten gleichen Namens

Alle Regelketten sind voneinander unabhängig! Es gibt also zwei verschiedene Prerouting- und sogar drei Output-Regelketten.

Dennoch ist in der `iptables`-Dokumentation oft einfach von der Output-Regelkette die Rede, ohne genaue Angabe, auf welche Tabelle sich diese Regelkette eigentlich bezieht. Gemeint sind in derartigen Fällen immer die Regelketten der Filter-Tabelle, die bei Weitem am wichtigsten ist.

Diese Sprachregelung gilt auch für das Kommando `iptables`: Dort kann die gewünschte Tabelle mit der Option `-t` angegeben werden. Entfällt diese Option, gilt das Kommando automatisch für die Filter-Tabelle.

Wenn ein IP-Paket bei seiner Wanderung durch den Kernel auf eine Regelkette stößt, überprüft der Kernel der Reihe nach sämtliche Regeln. Sobald eine Regel auf das Paket zutrifft, wird die in der Regel vorgesehene Aktion durchgeführt und das Paket weitergeleitet, gelöscht oder zurückgesendet. Nur wenn keine einzige der Regeln auf das Paket zutrifft, kommt das Standardverhalten des Filters zur Anwendung. Dieses lautet je nach Konfiguration abermals: weiterleiten, löschen oder zurücksenden.

Im Grundzustand des Kernels ist nur die Filter-Tabelle mit ihren drei Regelketten Input, Forward und Output aktiv. Keine dieser drei Regelketten enthält eine Regel, und das Standardverhalten lautet für alle drei Regelketten: weiterleiten.

**Grundzustand**

Die Regelsysteme für IPv4 und IPv6 sind vollkommen voneinander getrennt. Es gibt also gewissermaßen eine Firewall für IPv4 und eine zweite für IPv6. Im Regelfall ist es zweckmäßig, das Firewall-System parallel für IPv4 und IPv6 aufzubauen. Wenn Sie also einkommende Pakete für den Port 23456 sperren wollen, sind dazu zwei nahezu gleichlautende Regeln für IPv4 und IPv6 erforderlich.

**IPv6**

Nicht jede IPv4-Firewall-Regel ist auch für IPv6 geeignet: Beispielsweise ist NAT unter IPv4 ein wichtiges Thema, während NAT für IPv6 gänzlich unüblich ist und erst seit der Kernelversion 3.7 überhaupt unterstützt wird.

Die Kunst, einen Paketfilter zu erstellen, besteht darin, für jede relevante Filterkette das Standardverhalten sowie eine Reihe von Regeln zu definieren. Auf Kommandoebene verwalten Sie diese Regeln mit `iptables` für IPv4 und `ip6tables` für IPv6.

**iptables und ip6tables**

Das manuelle Einrichten einer Firewall erfordert eine Menge Fachwissen. Besser ist es zumeist, eine Konfigurationshilfe in Anspruch zu nehmen. Im folgenden Abschnitt stelle ich Ihnen dazu einige distributionsspezifische Programme vor. Wenn Sie die Firewall doch lieber manuell konfigurieren möchten, gibt [Abschnitt 37.5](#), »Firewall mit `iptables` selbst gebaut«, ein Einführungsbeispiel.

Weitere Informationen und Details gibt es wie üblich im Internet:

<https://netfilter.org>

<https://people.netfilter.org/rusty/unreliable-guides>

## nftables

Das Netfilter-System ist seit 2001 Bestandteil des Kernels. Mittlerweile haben sich einige grundlegende Mängel herausgestellt, weswegen bereits 2008 mit der Arbeit am Nachfolgesystem nftables begonnen wurde. nftables gilt mittlerweile als weitgehend ausgereift, aktuelle Distributionen enthielten im Sommer 2017 standardmäßig die erforderlichen Kernelfunktionen. Das zur Steuerung erforderliche nft-Kommando muss in der Regel extra installiert werden (Paketname nftables). Die folgenden Kommandos zum Test des nftables-Status wurden unter Debian 9 ausgeführt:

```
root# nft -v
nftables v0.7 (Scrooge McDuck)
root# systemctl start nftables
root# nft list tables
table inet filter
root# nft list chains
table inet filter {
    chain input {
        type filter hook input priority 0; policy accept;
    }
    chain forward {
        type filter hook forward priority 0; policy accept;
    }
    chain output {
        type filter hook output priority 0; policy accept;
    }
}
```

Aktiv genutzt wird nftables aber aktuell in keiner mir bekannten Distribution. Alle vorhandenen Firewall-Tools basieren noch auf dem Netfilter-System und dessen iptables-Kommando. Weitere Informationen finden Sie unter:

<https://wiki.nftables.org>

<https://lwn.net/Articles/324989>

<https://developers.redhat.com/blog/2016/10/28/what-comes-after-iptables-its-successor-of-course-nftables>

## 37.4 Firewall-Konfigurationshilfen

Viele Distributionen helfen ihren Anwendern bei der Firewall-Konfiguration mit komfortablen Benutzeroberflächen. Damit können Sie quasi per Mausklick eine einfache Firewall einrichten oder verändern. Hinter der recht einfachen Konfiguration verbirgt sich in der Regel eine Menge Paketfilter-Know-how. Das Resultat ist also oft ein besserer Schutz als eine selbst gebastelte Lösung.

Der Nachteil vorgegebener Firewalls ist deren Blackbox-Verhalten: Wenn die Wirkung des Filters überhaupt dokumentiert ist, dann zumeist nur dürftig. Sie wissen weder, wogegen der Filter Sie schützt, noch, welche Nebenwirkungen der Filter hat. Da kann es schon passieren, dass Linux-Einsteiger nach tagelangem Suchen, warum das Drucken im Netzwerk unmöglich ist, schließlich die Firewall als Schuldigen ausmachen. Der Versuch, den Paketfilter nun entsprechend anzupassen, wird vermutlich scheitern, vor allem dann, wenn ein Firewall-Grundwissen fehlt. Das vorauszusehende Ergebnis: Die Firewall wird einfach ausgeschaltet!

Alle im Folgenden beschriebenen Programme richten auch Regeln für IPv6 ein. Debian sieht standardmäßig keine Firewall vor und kennt auch keine distributions-spezifischen Konfigurationswerkzeuge. IPv6

Mit dem Kommando `iptables -L | wc -l` können Sie abschätzen, aus wie vielen Regeln die aktuelle Firewall besteht. Die resultierende Zahl ist ein Maß für die Komplexität der Firewall, aber nicht für ihre Sicherheit! Am sichersten wäre es, den Netzwerkverkehr ganz lahmzulegen – und das gelingt mit einer oder mit zwei Regeln. Anzahl der Filterregeln

Neben den hier beschriebenen Programmen finden Sie im Internet unzählige weitere Konfigurationshilfen. Populär und seit vielen Jahren erprobt sind: Links

<http://www.fwbuilder.org>

<http://www.shorewall.net>

### CentOS, Fedora und RHEL

Fedora sowie RHEL/CentOS seit Version 7 verwenden das FirewallD-System. Es wird durch `systemd` gestartet und durch Dateien in `/etc/firewalld` sowie `/usr/lib/firewalld` konfiguriert. FirewallD

Der Hauptvorteil von FirewallD gegenüber herkömmlichen Firewall-Systemen besteht darin, dass Änderungen dynamisch, also im laufenden Betrieb, durchgeführt werden. Im Gegensatz dazu war es beim bisherigen statischen System erforderlich, die Firewall bei Änderungen vorübergehend abzuschalten und dann neu zu errichten. Das war nicht nur ein Sicherheitsrisiko, sondern führte auch dazu, dass vorhandene Netzwerkverbindungen unterbrochen wurden.

**Zonen** Für die Verwaltung der Firewall ist der Hintergrundprozess `firewalld` verantwortlich. Konfigurationswerkzeuge kommunizieren via D-BUS mit dem Dämon. Eine zentrale Grundidee von FirewallD besteht darin, dass jeder Netzwerkschnittstelle eine sogenannte Zone zugeordnet wird. Eine »Zone« im Sinne von FirewallD ist eine Sammlung von Regeln für einen bestimmten Anwendungszweck. Die folgende Liste beschreibt ganz kurz einige Zonen:

- ▶ `block`: blockiert jeden Netzwerkverkehr. Der Absender erhält eine ICMP-Fehlermeldung.
- ▶ `drop`: blockiert jeden Netzwerkverkehr. Der Absender wird nicht informiert.
- ▶ `trusted`: erlaubt jeden Netzwerkverkehr. Diese Zone ist für gut gesicherte lokale Netzwerke gedacht, aber nicht für WLAN-Verbindungen.
- ▶ `external`: blockiert die meisten Ports und aktiviert Masquerading (IPv4). Bei einem Router ist diese Zone für die Schnittstelle vorgesehen, die die Verbindung zum Internet herstellt.
- ▶ `home` und `internal`: blockiert die meisten Ports, akzeptiert aber Samba (nur als Client), CUPS und Zeroconf/Avahi/mdns. Beide Zonen sind für Rechner in einem als einigermaßen sicher geltenden lokalen Netzwerk gedacht. Wenn Sie diese Zone nutzen und selbst Windows-Netzwerkverzeichnisse freigeben möchten, müssen Sie außerdem den Dienst SAMBA freischalten.
- ▶ `public`: ähnlich wie `home`, blockiert aber auch CUPS und Samba-Client-Funktionen. Die Zone ist für die Internetnutzung in unsicheren Netzwerken gedacht, z. B. in einem öffentlichen WLAN.
- ▶ `FedoraWorkstation` und `FedoraServer`: Diese beiden Zonen kommen standardmäßig für Netzwerkschnittstellen von Fedora-Installationen zum Einsatz. `FedoraServer` blockiert alle Dienste außer SSH und der DHCP-Client-Konfiguration. `FedoraWorkstation` ist deutlich liberaler und akzeptiert auch Samba-Client-Verbindungen sowie den Verkehr über alle Ports zwischen 1025 und 65535.

Die Zonenregeln mit der Ausnahme von `block`, `drop` und `trusted` können verändert werden. Bei Bedarf können Sie auch selbst eigene Zonen definieren.

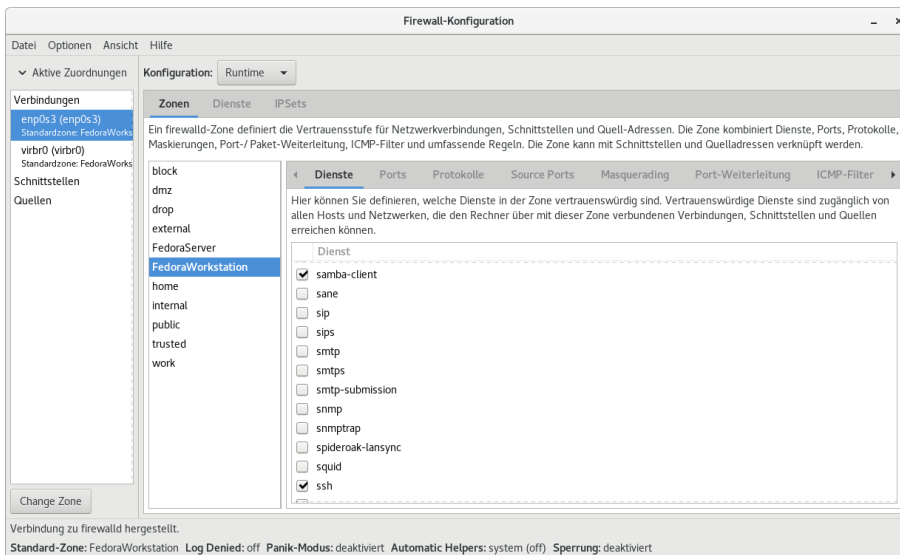
**Zonenzuordnung in `ifcfg-xxx`** Standardmäßig ordnen CentOS und RHEL alle Schnittstellen der Zone `public` zu. Bei aktuellen Fedora-Versionen kommt stattdessen die Zone `FedoraWorkstation` bzw. `FedoraServer` zum Einsatz.

Um eine Schnittstelle explizit einer Firewall-Zone zuzuordnen, tragen Sie in die Datei `/etc/sysconfig/network-scripts/ifcfg-xxx` die Zeile `ZONE=zonename` ein. Dabei ist `xxx` der Name der Schnittstelle.

```
# Datei /etc/sysconfig/network-scripts/ifcfg-xxx
...
ZONE=trusted
```

Zur Firewall-Konfiguration steht die Benutzeroberfläche `firewall-config` zur Verfügung (siehe [Abbildung 37.3](#)). Das gleichnamige Paket muss vor dem ersten Start installiert werden. Die Bedienung ist leider recht unübersichtlich. Nach dem Start sehen Sie die sogenannte `RUNTIME-KONFIGURATION` der diversen Zonen. Hier durchgeführte Änderungen gelten sofort für die betreffende Zone, sie werden aber nicht gespeichert und gehen somit beim Neustart des Rechners verloren.

Benutzeroberfläche



**Abbildung 37.3** Die grafische Benutzeroberfläche zur FirewallD-Konfiguration

Mit `OPTIONEN • STANDARDZONE ÄNDERN` können Sie einstellen, welche Zone standardmäßig gilt, also für alle Netzwerkschnittstellen, bei denen nicht explizit eine andere Zone eingestellt ist. Um die Firewall-Zone individuell für eine bestimmte Schnittstelle zu ändern, führen Sie `OPTIONEN • VERBINDUNGSZONEN ÄNDERN` aus.

Um die Definition einer Zone dauerhaft zu ändern, wechseln Sie im Konfigurationsprogramm in die Ansicht `KONFIGURATION = PERMANENT`. Ärgerlicherweise müssen Änderungen, die Sie zuvor in der `RUNTIME`-Ansicht ausprobiert haben, jetzt wiederholt werden.

### Firewall deaktivieren

Das Konfigurationsprogramm bietet keine direkte Option, um die Firewall ganz zu deaktivieren. Wenn Sie das wünschen, müssen Sie `OPTIONEN • STANDARDZONE ÄNDERN` ausführen und die Zone `trusted` einstellen. Standardmäßig sind alle Schnittstellen der Standardzone zugewiesen. Mit `trusted` ist damit jeglicher Netzverkehr erlaubt.

Beachten Sie aber, dass diese Veränderung der Standardzone keinen Einfluss auf Schnittstellen hat, die Sie explizit einer anderen Zone zugewiesen haben. Sollte es bei Ihnen derartige Schnittstellen geben, müssen Sie diese mit `OPTIONEN • VERBINDUNGSZONEN ÄNDERN` extra bearbeiten. In der Standardkonfiguration ist das nicht erforderlich.

Wenn Sie eine vollkommen eigene Firewall einrichten möchten, müssen Sie das Paket `firewalld` deinstallieren.

## firewall-cmd

Im Terminal führen Sie FirewallD-Konfigurationsänderungen mit `firewall-cmd` durch. Standardmäßig werden die Änderungen dabei nur dynamisch durchgeführt, aber nicht gespeichert. Wenn die Änderungen dauerhaft gelten sollen, müssen Sie zusätzlich die Option `--permanent` angeben und die Änderungen anschließend mit `firewall-cmd --reload` explizit aktivieren. Eigene Regeln werden im Verzeichnis `/etc/firewalld` gespeichert.

**Status** Um einen Überblick über den aktuellen Zustand der Firewall zu erhalten, führen Sie die beiden folgenden Kommandos aus:

```
root# firewall-cmd --state
running
root# firewall-cmd --get-active-zones
internal
  interfaces: enp0s8
external
  interfaces: enp0s3
```

Sie wissen jetzt, dass die Firewall aktiv ist und welche Netzwerkschnittstellen welchen Firewall-Zonen zugeordnet sind.

**Zonen** Die folgenden Kommandos ergründen, welche Zonen es gibt und welche davon aktiv sind:

```
root# firewall-cmd --get-zones                (alle Zonen auflisten)
  FedoraServer FedoraWorkstation block dmz drop
  external home internal public trusted work
root# firewall-cmd --get-default-zone         (Defaultzone feststellen)
  FedoraWorkstation
root# firewall-cmd --get-zone-of-interface=enp4s0 (Zone für eine Schnittstelle)
  FedoraServer
```

Die Defaultzone können Sie mit `firewall-cmd --set-default-zone` verändern. Die Zuordnung einer Zone zu einer Schnittstelle können Sie mit `--add-interface` bzw. `--remove-interface` verändern:



```
root# firewall-cmd --permanent --zone=FedoraServer --remove-interface=enp4s0
root# firewall-cmd --permanent --zone=FedoraWorkstation --add-interface=enp4s0
```

Beachten Sie aber, dass die Zonenzuordnung auch durch `/etc/sysconfig/network-scripts/ifcfg-xxx` erfolgen kann. Die `ifcfg`-Dateien haben Vorrang gegenüber der Veränderung der Zonen!

Es gibt kein Kommando, um alle für FirewallD bekannten Dienste aufzulisten. Stattdessen können Sie mit `ls` alle entsprechenden XML-Dateien ermitteln. Die Syntax der XML-Dateien ist leicht zu verstehen, d. h., Sie erkennen auf Anhieb, welche Ports jeweils freigeschaltet werden. Dienste

```
user$ ls /usr/lib/firewalld/services/*.xml /etc/firewalld/services/*.xml
```

Um für eine Zone einen Dienst oder einen IP-Port freizuschalten, führen Sie die folgenden Kommandos aus:

```
root# firewall-cmd --permanent --zone=FedoraWorkstation --add-service=name
root# firewall-cmd --permanent --zone=public --add-port=nnn/tcp
root# firewall-cmd --permanent --zone=public --add-port=nnn/udp
root# firewall-cmd --reload    (Änderungen aktivieren)
```

Dabei ist `name` der Name eines Dienstes (z. B. `https` oder `nfs`) bzw. `nnn` die Nummer eines Ports.

Wenn Sie zusätzlich zu den FirewallD-Regeln eigene iptables-Regeln integrieren möchten, können Sie das unkompliziert in `/etc/firewalld/direct.xml` tun. Die Syntax dieser Datei dokumentiert `man firewalld.direct`. iptables-Zusatzregeln

Eine umfassende Dokumentation zu FirewallD sowie eine Menge Anwendungsbeispiele zu `firewall-cmd` finden Sie hier: Dokumentation

<https://fedoraproject.org/wiki/FirewallD>

## SUSE

SUSE-Distributionen richten standardmäßig eine Firewall ein, wobei die Schnittstelle zum Internet automatisch der externen Zone zugeordnet wird. Die Konfiguration erfolgt durch das YaST-Modul SICHERHEIT • FIREWALL. Dabei werden – abweichend vom sonst dialogorientierten YaST-Konzept – die einzelnen Dialogblätter durch Einträge im linken Teil des YaST-Fensters ausgewählt (siehe [Abbildung 37.4](#)). Falls der Rechner als Gateway zu einem LAN dient, ordnen Sie im Dialogblatt INTERFACES die LAN-Schnittstelle der internen Zone zu und aktivieren im Dialogblatt MASQUERA-DING die gleichnamige Option. Insgesamt ist die Konfiguration sehr geradlinig und einfach durchzuführen.

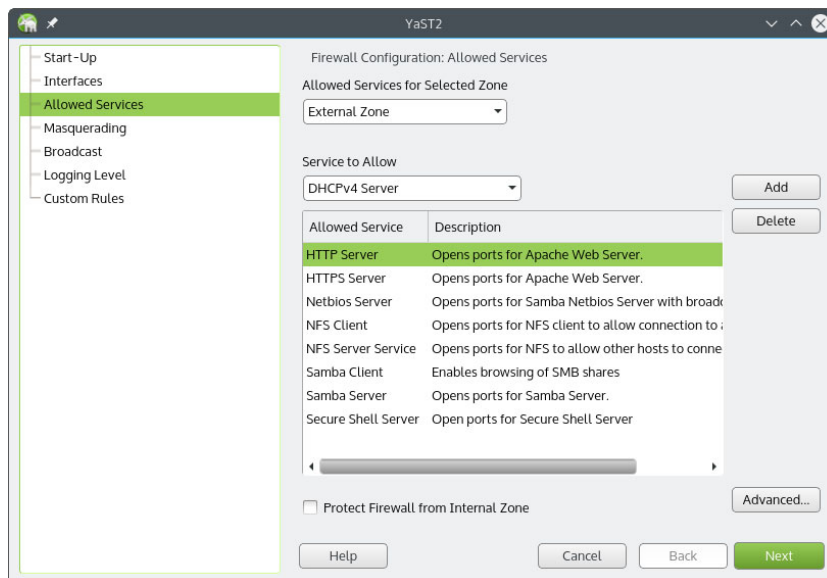


Abbildung 37.4 Firewall-Konfiguration in SUSE-Distributionen

Die Firewall-Einstellungen werden in `/etc/sysconfig/SuSEfirewall` gespeichert. Um den Start der Firewall kümmert sich `systemd`.

## Ubuntu

**ufw** Bei Ubuntu wird standardmäßig keine Firewall eingerichtet, und es gibt auch keine grafischen Konfigurationswerkzeuge. Dafür enthält Ubuntu das Kommando `ufw` (*Uncomplicated Firewall*). Es ermöglicht die Definition von Firewall-Regeln in einer wesentlich einfacheren Syntax als `iptables`. Zudem sollen in zukünftigen Ubuntu-Versionen bei der Installation von Netzwerkdiensten die entsprechenden `ufw`-Regeln zur Absicherung gleich mitinstalliert werden. Das ist freilich noch Zukunftsmusik: `ufw` hat bislang nur eine geringe Akzeptanz gefunden.

Wenn Sie schon Erfahrung mit Paketfiltern haben, werden Sie mit `ufw` rasch zum Ziel kommen: `ufw enable` aktiviert die Firewall. Die Firewall wird sofort und in Zukunft auch bei jedem Rechnerstart aktiviert. `ufw disable` deaktiviert die Firewall wieder. `ufw default allow` bzw. `ufw default deny` gibt an, ob eintreffende Pakete grundsätzlich akzeptiert oder abgewiesen werden. Normalerweise gilt `deny`, d. h., mit der Aktivierung der Firewall kann kein auf dem Rechner laufender Netzwerkdienst mehr von außen angesprochen werden!

Zusätzlich definieren Sie mit `ufw allow/deny nnn` bzw. `ufw allow/deny dienst` Regeln, die für spezielle IP-Ports bzw. Protokolle gelten. `ufw status` gibt Informationen zum

aktuellen Zustand der Firewall. `ufw` kümmert sich standardmäßig sowohl um IPv4 als auch um IPv6. Wenn Sie IPv6-Verkehr ganz blockieren möchten, verwenden Sie in `/etc/sysconfig/ufw` die Einstellung `IPV6=no`.

```
user$ sudo -s
root# ufw enable
root# ufw allow ssh
root# ufw status
Status: Aktiv
Zu      Aktion  Von
--      -
22      ALLOW   Anywhere
22      ALLOW   Anywhere (v6)
```

Bei Bedarf können Sie Firewall-Regeln für bestimmte Schnittstellen oder IP-Adressbereiche formulieren. Details dazu erläutert `man ufw`. Eigene Regeln werden in `/lib/ufw/user.rules` und `user6.rules` (für IPv6) gespeichert. Außerdem berücksichtigt `ufw` die Regeldateien aus dem Verzeichnis `/etc/ufw/`.

Darüber hinaus werden bei der Installation von Netzwerkdiensten dazu passende Regeldateien im Verzeichnis `/etc/ufw/applications.d` gespeichert. `ufw app list` liefert die Liste der so definierten Profile. Mit `ufw allow/deny` können Sie derartige Profile aktivieren (also die entsprechenden Ports in der Firewall freigeben) bzw. sperren. Profile

```
root# ufw app list
Verfügbare Anwendungen:
  Apache
  Apache Full
  Apache Secure
  CUPS
  Dovecot IMAP
  Dovecot POP3
  Dovecot Secure IMAP
  Dovecot Secure POP3
  OpenSSH
  Postfix
  ...
root# ufw app info "Apache Full"
Profil: Apache Full
Titel: Web Server (HTTP,HTTPS)
Beschreibung: Apache v2 is the next generation of the omnipresent Apache
web server.
Ports: 80,443/tcp
root# ufw allow "Apache Full"
```

Bitte beachten Sie, dass Applikationsprofile zwar standardmäßig installiert, aber nicht aktiviert werden! Die Profile sollen Ihnen lediglich eine Hilfestellung geben, wenn Sie eine Firewall einrichten möchten.

**Dokumentation** Weitere Informationen und Beispiele zu `ufw` erhalten Sie mit `man ufw` bzw. auf den folgenden Seiten:

<http://wiki.ubuntuusers.de/ufw>

<https://help.ubuntu.com/16.04/serverguide/firewall.html>

<https://wiki.ubuntu.com/UncomplicatedFirewall>

**Gufw** Zu `ufw` gibt es mit `Gufw` eine grafische Benutzeroberfläche, die bei aktuellen Ubuntu-Versionen als `universe`-Paket zur Verfügung steht, also ohne offiziellen Support. Leider ignoriert das Programm die Applikationsprofile. Die Weiterentwicklung des Programms wurde offensichtlich eingestellt; das letzte Update der Projektseite stammt von 2014:

<https://help.ubuntu.com/community/Gufw>

## 37.5 Firewall mit iptables selbst gebaut

Dieser Abschnitt zeigt anhand zweier kleiner Beispiele, dass das Zusammenstellen einfacher Firewalls keine Hexerei ist. Die Beispiele richten sich an Leser, die Firewalls besser verstehen und damit experimentieren möchten.

### Warnung

Ich möchte mit diesem Abschnitt nicht den Anschein erwecken, dass Sie mit selbstgebastelten 20-, 30-zeiligen Scripts professionelle Firewall-Tools ersetzen können! Ich rate Ihnen ganz im Gegenteil dazu, auf die Firewall-Werkzeuge Ihrer Distribution zurückzugreifen oder distributionsunabhängige Firewall-Konfigurationshilfen einzusetzen.

### Minimale Client-Absicherung (IPv4 und IPv6)

Die meisten Rechner genießen zu Hause oder in einer Firma in lokalen IPv4-Netzwerken dank NAT eine gewisse Sicherheit. Damit ist es vorbei, wenn Sie unterwegs in einem öffentlichen WLAN E-Mails abrufen: Sie wissen nicht, wer sich sonst noch im Funknetz befindet!

Ein ähnliches Sicherheitsproblem besteht, wenn Sie eine IPv6-Adresse haben. NAT spielt nun keine Rolle mehr: Weltweit kann jetzt jeder, der über eine IPv6-Verbindung verfügt, IP-Pakete zu Ihrem Rechner senden! Mit diesen Datenpaketen könnte jemand z. B. einen SSH-Login versuchen oder über Samba freigegebene Netzwerkverzeichnisse auslesen.

Die folgende Mini-Firewall verbessert die Sicherheit in beiden Fällen ganz erheblich. Sie verbietet grundsätzlich jeden Datenverkehr, der nicht von Ihrem Rechner initiiert wird. Mit anderen Worten: Sie können z.B. via SSH einen externen Rechner administrieren, umgekehrt kann aber niemand auch nur versuchen, sich via SSH bei Ihnen einzuloggen. Analog funktioniert dieser Schutz auch für alle anderen Netzwerkdienste. **Idee**

Das Beispiel habe ich unter Ubuntu entwickelt und getestet. Wenn Sie eine andere Distribution verwenden, müssen Sie unbedingt vorher die distributionspezifische Firewall deaktivieren bzw. unter Fedora das Paket `firewalld` deinstallieren!

Der erste Teil des Firewall-Scripts führt eine Art iptables-Reset durch. `iptables -P` stellt dann das Standardverhalten aller Filter auf `ACCEPT`. `iptables -F` löscht alle vorhandenen Regeln, wobei für die NAT-Tabelle ein eigenes Kommando erforderlich ist. `iptables -X` löscht alle benutzerdefinierten Regelketten. Analog werden all diese Kommandos auch für IPv6 ausgeführt. Netfilter erlaubt nun jeglichen IP-Verkehr. **Reset**

```
#!/bin/bash
# Mini-Firewall (Teil 1)
IPT4=$(which iptables)
IPT6=$(which ip6tables)

# reset iptables
for IPT in $IPT4 $IPT6; do
    $IPT -P INPUT ACCEPT
    $IPT -P OUTPUT ACCEPT
    $IPT -P FORWARD ACCEPT
    $IPT -F
    $IPT -X
done
```

Damit IPv6-Funktionen zur Autokonfiguration funktionieren, ist es erforderlich, ICMPv6-Pakete passieren zu lassen. Dazu dienen die zwei abschließenden Kommandos im ersten Teil des Listings: **ICMPv6**

```
# Mini-Firewall (Teil 2): ICMPv6 zulassen
$IPT6 -A INPUT -p ipv6-icmp -j ACCEPT
$IPT6 -A FORWARD -p ipv6-icmp -j ACCEPT
```

Die folgenden Zeilen definieren eine neue Regelkette mit dem Namen `wall`. Sie stellt einen ebenso eleganten wie wirkungsvollen Schutz vor neuen Verbindungen von außen dar. Die erste `wall`-Regel besagt, dass alle Pakete akzeptiert werden, die zu einer bereits vorhandenen Verbindung gehören. **wall-Regelkette für IPv4**

Die zweite Regel akzeptiert Pakete, die eine neue Verbindung initiieren, sofern die Verbindung *nicht* über die Internetschnittstelle hergestellt wird. Die Inversion wird syntaktisch durch das Ausrufezeichen vor der Option `-i` ausgedrückt. Im Klartext

bedeutet die Regel, dass es beispielsweise möglich ist, aus dem lokalen Netz heraus eine HTTP-Kommunikation mit dem Rechner zu starten, nicht aber aus dem Internet heraus.

Die dritte Regel lautet: Alle Pakete, die nicht den vorigen Regeln entsprechen, werden abgewiesen. Diese letzte Regel entspricht also dem Motto: Alles verbieten, was nicht explizit erlaubt ist! Einem potenziellen Angreifer aus dem Internet wird es daher nicht gelingen, eine SSH-Session auch nur zu starten. Das Gleiche gilt natürlich auch für alle anderen Netzwerkdienste – HTTP, FTP, Telnet etc.

Die zwei abschließenden Kommandos des Scripts geben an, dass für alle Pakete, die die Input- oder Forward-Filter durchlaufen, die wall-Regeln zur Anwendung kommen. Vergessen Sie nicht, die Variable INET mit dem tatsächlichen Namen Ihrer Schnittstelle zum Internet einzustellen, z. B. wlan0 oder enp0s3. Den Schnittstellennamen können Sie mit `ip addr` ermitteln.

```
# Mini-Firewall (Teil 3): wall-Regelkette für IPv4
INET=eth0
$IPT4 -N wall
$IPT4 -A wall -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT4 -A wall -m state --state NEW ! -i $INET -j ACCEPT
$IPT4 -A wall -j DROP

# die Regelkette für INPUT und FORWARD anwenden
$IPT4 -A INPUT -j wall
$IPT4 -A FORWARD -j wall
```

#### wall-Regelkette für IPv6

Der letzte Teil der Mini-Firewall definiert dieselbe wall-Regelkette für IPv6. Normalerweise kommt dabei dieselbe Schnittstelle wie für die IPv4-Verbindung zum Einsatz. Eine Ausnahme können IPv6-Verbindungen sein, die durch einen Tunnel geleitet werden – dann müssen Sie die Tunnel-Schnittstelle in INET6 angeben:

```
# Mini-Firewall (Teil 4): wall-Regelkette für IPv6
INET6=$INET
$IPT6 -N wall
$IPT6 -A wall -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT6 -A wall -m state --state NEW ! -i $INET6 -j ACCEPT
$IPT6 -A wall -j DROP
$IPT6 -A INPUT -j wall
$IPT6 -A FORWARD -j wall
```

### Internet-Gateway für lokales Netzwerk absichern (IPv4 und IPv6)

#### Aufgabenstellung

Das zweite Beispiel baut auf den obigen Ideen auf, ist aber schon etwas anspruchsvoller. Es geht darum, ein Gateway abzusichern, das anderen Rechnern im LAN Internetzugang gibt (siehe [Kapitel 27](#), »Internet-Gateway«). [Abbildung 37.1](#) zeigt die Ausgangssituation.

Die Aufgabe der Firewall besteht darin, gefährliche Ports nach außen hin ganz zu blockieren und bei den restlichen Ports eine Kommunikation nur dann zu erlauben, wenn die Kommunikation von innen initiiert wurde. Der Schutz gilt nun auch für alle IPv4- und IPv6-Clients im lokalen Netzwerk. Das Firewall-Script kümmert sich außerdem um die Aktivierung der Masquerading- und Forwarding-Funktionen.

Die Firewall besteht aus den beiden Script-Dateien `myfirewall-start` und `myfirewall-stop`. Die Grundeinstellungen der Firewall werden in der Konfigurationsdatei `myfirewall` gespeichert.

Überblick

```
/etc/myfirewall/myfirewall-start (Firewall-Start)
/etc/myfirewall/myfirewall-stop  (Firewall-Stopp)
/etc/default/myfirewall          (Grundeinstellungen)
```

### Basiskonfiguration (myfirewall)

Die Variable `MFW_ACTIVE` in `/etc/default/myfirewall` steuert, ob die Firewall während des Systemstarts aktiviert werden sollen. `MFW_MASQ` gibt an, ob Masquerading und Forwarding aktiviert werden sollen. Die restlichen Variablen geben die Schnittstellen und Adressen des lokalen Netzwerks bzw. der Internetverbindung an. `eth0` und `eth1` müssen Sie durch die Namen Ihrer Schnittstellen ersetzen, z. B. durch `enp4s0` oder `wlp2s0`.

Sofern auf Ihrem Gateway eine Tunnel-Software läuft (z. B. `aiccu` für SixXs), um auch IPv6-Verkehr vom LAN in das Internet zu leiten, gibt die Variable `MFW_INET6` die entsprechende Schnittstelle an. Wenn Ihr lokales Netz kein IPv6 nutzt, können Sie alle diesbezüglichen Einstellungen und Kommandos aus den Scripts streichen.

```
# Datei /etc/default/myfirewall
# Firewall starten: yes/no
MFW_ACTIVE=yes
# Masquerading und Forwarding aktivieren: yes/no
MFW_MASQ=yes
# Lokales Netzwerk
MFW_LAN=eth1
MFW_LAN_IP=192.168.0.0/24
# Schnittstelle zum Internet (IPv4 und IPv6)
MFW_INET=eth0
MFW_INET6=sixxs
```

### Firewall stoppen (myfirewall-stop)

Das Script `myfirewall-stop` stellt den iptables-Grundzustand her und deaktiviert die Firewall – wie in den ersten Zeilen der Mini-Firewall. Ergänzend dazu werden diesmal auch die `nat`-Regelketten zurückgesetzt.

```
#!/bin/bash
# Datei /etc/myfirewall/myfirewall-stop

# Konfigurationseinstellungen lesen
. /etc/default/myfirewall
IPT4=$(which iptables)
IPT6=$(which ip6tables)
SYS=$(which sysctl)

# Firewall-Reset (IPv4 und IPv6)
for IPT in $IPT4 $IPT6; do
    $IPT -P INPUT ACCEPT
    $IPT -P OUTPUT ACCEPT
    $IPT -P FORWARD ACCEPT
    $IPT -F
    $IPT -X
done

# NAT-Reset nur für IPv4
$IPT4 -F -t nat
$IPT4 -P POSTROUTING ACCEPT -t nat
$IPT4 -P PREROUTING ACCEPT -t nat
$IPT4 -P OUTPUT ACCEPT -t nat

# Forwarding stoppen
$SYS -q -w net.ipv4.ip_forward=0
$SYS -q -w net.ipv6.conf.all.forwarding=0
```

Vergessen Sie nicht, die Script-Datei mit `chmod u+x` als ausführbar zu kennzeichnen!

### Firewall starten (`myfirewall-start`)

Wesentlich interessanter ist naturgemäß `myfirewall-start`. Das Script beginnt damit, die Stopp-Regeln auszuführen. Das bewirkt gleichsam ein Reset des Netfilter-Systems. Alle weiteren `iptables`-Kommandos können sich somit darauf verlassen, dass vorher keine anderen Regeln definiert wurden.

**SSH öffnen,  
ICMPv6 zulassen**

Die ersten drei `iptables`-Kommandos lassen den Zugriff auf SSH-Server aus dem Internet zu. Diese Kommandos zeigen beispielhaft, wie Sie trotz Firewall einzelne Dienste nach außen zugänglich machen. Beachten Sie, dass das Absichern eines SSH-Servers im IPv6-Netz schwierig ist; sicherer ist es, SSH zumindest für IPv6 zu sperren.

```
#!/bin/bash
# Datei /etc/myfirewall/myfirewall-start (Teil 1)

# Konfigurationseinstellungen lesen
. /etc/default/myfirewall
```



```

IPT=$(which iptables)
SYS=$(which sysctl)

if [ $MFW_ACTIVE != "yes" ]; then
    echo "Firewall disabled in /etc/default/myfirewall"
    exit 0
fi

# Reset aller Firewall-Regeln
. /etc/myfirewall/myfirewall-stop

# Zugriff auf den SSH-Server (Port 22) aus dem Internet erlauben
$IPT4 -A INPUT -i $MFW_INET -p tcp --dport 22 -j ACCEPT
$IPT6 -A INPUT -i $MFW_INET6 -p tcp --dport 22 -j ACCEPT
$IPT6 -A FORWARD -i $MFW_INET6 -p tcp --dport 22 -j ACCEPT

# ICMPv6 erlauben
$IPT6 -A INPUT -p ipv6-icmp -j ACCEPT
$IPT6 -A FORWARD -p ipv6-icmp -j ACCEPT

```

In der for-Schleife werden nun einige Ports gegenüber dem Internet vollständig blockiert. Sie können die Port-Liste bei Bedarf selbst ergänzen. **Ports sperren**

```

# Datei /etc/myfirewall/myfirewall-start (Teil 2)
# einige Ports komplett sperren
# 23 (telnet)
# 69 (tftp)
# 135 (Microsoft DCOM RPC)
# 139 (NetBIOS/Samba/etc.)
# 445 (CIFS-Dateisystem für Samba/SMB)
# 631 (ipp/CUPS)
# 1433 (Microsoft SQL Server)
# 2049 (NFS)
# 3306 (MySQL)
# 5999-6003 (X-Displays)
for PORT in 23 69 135 139 445 631 1433 2049 3306 \
5999 6000 6001 6002 6003; do
    $IPT4 -A INPUT -i $MFW_INET -p tcp --dport $PORT -j DROP
    $IPT4 -A OUTPUT -o $MFW_INET -p tcp --dport $PORT -j DROP
    $IPT4 -A INPUT -i $MFW_INET -p udp --dport $PORT -j DROP
    $IPT4 -A OUTPUT -o $MFW_INET -p udp --dport $PORT -j DROP
    $IPT6 -A INPUT -i $MFW_INET6 -p tcp --dport $PORT -j DROP
    $IPT6 -A OUTPUT -o $MFW_INET6 -p tcp --dport $PORT -j DROP
    $IPT6 -A INPUT -i $MFW_INET6 -p udp --dport $PORT -j DROP
    $IPT6 -A OUTPUT -o $MFW_INET6 -p udp --dport $PORT -j DROP
done

```

**wall-Regelkette** Die Idee der wall-Regelkette habe ich schon im Rahmen der Mini-Firewall beschrieben. Bei diesem größeren Beispiel kommt sie nur zur Anwendung, wenn nicht schon vorher eine der DROP- oder ACCEPT-Regeln zutraf. Die Reihenfolge der Regeln ist also beim Aufbau einer Firewall entscheidend!

```
# Datei /etc/myfirewall/myfirewall-start (Teil 3)
# wall-Regelkette für IPv4 ..
$IPT4 -N wall
$IPT4 -A wall -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT4 -A wall -m state --state NEW ! -i $MFW_INET -j ACCEPT
$IPT4 -A wall -j DROP
$IPT4 -A INPUT -j wall
$IPT4 -A FORWARD -j wall

# ... und für IPv6
$IPT6 -N wall
$IPT6 -A wall -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT6 -A wall -m state --state NEW ! -i $MFW_INET6 -j ACCEPT
$IPT6 -A wall -j DROP
$IPT6 -A INPUT -j wall
$IPT6 -A FORWARD -j wall
```

**Masquerading und Forwarding** Zu guter Letzt müssen auf einem Gateway das Masquerading und das Forwarding aktiviert werden (siehe [Abschnitt 27.3](#), »Masquerading (NAT)«):

```
# Datei /etc/myfirewall/myfirewall-start (Teil 4)
if [ $MFW_MASQ = 'yes' ]; then
    $IPT4 -A POSTROUTING -t nat -o $MFW_INET -s $MFW_LAN_IP -j MASQUERADE
    $SYS -q -w net.ipv4.ip_forward=1
    $SYS -q -w net.ipv6.conf.all.forwarding=1
fi
```

### systemd-Integration

Zum Start der Firewall bietet sich eine systemd-Konfigurationsdatei an. Die Syntax derartiger Dateien ist in [Abschnitt 24.3](#), »Eigene Init-Scripts bzw. Init-Konfigurationsdateien«, beschrieben.

```
# Datei /etc/systemd/system/myfirewall.service
[Unit]
Description=myfirewall
After=syslog.target network.target

[Service]
Type=oneshot
RemainAfterExit=true
ExecStart=/etc/myfirewall/myfirewall-start
ExecStop=/etc/myfirewall/myfirewall-stop
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Nach dem Einrichten der Datei müssen Sie systemd auffordern, die Konfigurationsdateien neu einzulesen:

```
root# systemctl daemon-reload
```

Um die Firewall unmittelbar und in Zukunft jedes Mal zu starten, wenn das Multi-User-Target erreicht wird, sind die folgenden Kommandos erforderlich:

```
root# systemctl start myfirewall
```

```
root# systemctl enable myfirewall
```



# Kapitel 38

## SELinux und AppArmor

SELinux und AppArmor sind Kernelerweiterungen, die laufende Prozesse überwachen und sicherstellen, dass diese bestimmte Regeln einhalten. SELinux kommt standardmäßig unter CentOS, Fedora und RHEL zum Einsatz, AppArmor unter SUSE- und Ubuntu-Distributionen.

Dieses Kapitel gibt eine Einführung in die Funktionsweise und Konfiguration von SELinux und AppArmor. Es zeigt auch, wie Sie Regelverstöße feststellen und wie Sie darauf reagieren können.

### 38.1 SELinux

Unter Linux gilt normalerweise das traditionelle System zur Verwaltung von Zugriffsrechten: Jedes Programm läuft in einem Benutzer-Account. Dieser Account bestimmt, auf welche (Device-)Dateien das Programm zugreifen darf.

Gewöhnliche Programme verwenden den Account des Benutzers, der das Programm gestartet hat. Netzwerkdienste, Datenbank-Server etc. werden mit `root`-Rechten gestartet, wechseln aber aus Sicherheitsgründen zumeist unmittelbar nach dem Start in einen anderen Account mit eingeschränkten Rechten.

Das Unix-Rechtesystem ist zwar ausgesprochen einfach, bietet aber nur eingeschränkte Konfigurationsmöglichkeiten. Wenn es einem Angreifer gelingt, die Steuerung eines Programms zu übernehmen, kann er auf zahllose Dateien zugreifen, die das Programm normalerweise gar nicht benötigt. Besonders schlimm ist es, wenn der Angreifer die Kontrolle über ein Programm mit `root`-Rechten erhält bzw. wenn er über Umwege erreichen kann, dass eigener Code mit `root`-Rechten ausgeführt wird: Damit kann er den Rechner uneingeschränkt manipulieren, eigene Programme installieren und starten etc.

Vielleicht fragen Sie sich, wie ein Angreifer die Kontrolle über ein Programm erlangen kann. Fast immer werden dabei Fehler im Programmcode ausgenutzt. Beispielsweise wird durch das Übersenden manipulierter Daten ein sogenannter Pufferüberlauf ausgelöst. Dieser Fehler wird wiederum dazu genutzt, um dem Programm eigenen

Code unterzujubeln und diesen auszuführen. Natürlich gibt es auch andere Verfahren – aber immer geht es darum, Sicherheitslücken des Programms zu missbrauchen, um das Programm zweckentfremdet zu nutzen bzw. zu manipulieren.

**Sicherheitsmaßnahmen** Fehlerfreie Programme gibt es nicht und wird es wohl auch in Zukunft nie geben, wenn man einmal von winzigen Trivialprogrammen absieht. Deswegen wurden im Laufe der Zeit alle möglichen Verfahren entwickelt, um die durch Programmfehler verursachten Risiken zu minimieren: Zu den etablierten Sicherheitsmaßnahmen zählt, möglichst auf Dämonen mit root-Rechten zu verzichten, möglichst wenige Programme bzw. Scripts mit `setuid`-Bit zu installieren (siehe [Abschnitt 11.6](#), »Spezialbits und die `umask`-Einstellung«) und die Ausführung von Code im Stack durch die von Red Hat entwickelte Kernelerweiterung Exec Shield zu verbieten.

**SELinux** Noch einen Schritt weiter geht die ursprünglich von der NSA als Open-Source-Code entwickelte Kernelerweiterung SELinux. Der Zweck dieser Erweiterung ist es, dass der Kernel die Ausführung von Programmen anhand von Regeln überwacht. Diese Vorgehensweise wird als *Mandatory Access Control* bezeichnet, kurz MAC. Wird eine Regel verletzt, verhindert SELinux die Operation oder protokolliert eine Warnung. Das regelverletzende Programm wird durch SELinux nicht beendet. Es hängt vom Programm ab, wie es darauf reagiert, dass es auf eine bestimmte Datei nicht zugreifen kann oder eine Netzwerkschnittstelle nicht nutzen kann.

MAC-Regeln ermöglichen eine sehr viel engmaschigere Sicherheitskontrolle als das Unix-Zugriffssystem. Mit ihnen kann man einem Programm unabhängig von Unix-Zugriffsrechten bzw. -Accounts den Zugriff auf bestimmte Verzeichnisse oder Netzwerkfunktionen generell verbieten. Da diese Regeln auf Kernelebene überwacht werden, gelten sie selbst dann noch, wenn das Programm aufgrund eines Fehlers bzw. Sicherheitsmangels außer Kontrolle gerät.

#### SELinux ist sauber, obwohl der Code von der NSA entwickelt wurde

Die *National Security Agency* ist ein Nachrichtendienst der USA, der zuletzt aufgrund der umfassenden Überwachung des gesamten Internetverkehrs eine Menge negative Presse auf sich zog. Obwohl SELinux also aus Geheimdienstkreisen stammt, besteht keine Gefahr, dass Linux auf diese Weise um Überwachungsfunktionen erweitert wurde: Der SELinux-Code ist öffentlich, wurde von vielen unabhängigen Experten kontrolliert und verbessert und ist Bestandteil des offiziellen Kernels. Wenn die NSA Sie und andere überwacht, dann sicher nicht durch eine Hintertür in SELinux.

**SELinux-Regeln** Ohne entsprechende Regeln ist SELinux wirkungslos. Ob ein System durch SELinux sicherer wird, hängt somit vor allem von der Qualität der Regeln ab. Von den gängigen Distributoren hat bisher nur Red Hat intensiv Zeit und Mühe in die Entwicklung der-

artiger Regeln investiert. Dabei dient Fedora gewissermaßen als Testvehikel. Was sich dort bewährt, findet schließlich Eingang in die Red-Hat-Enterprise-Versionen (RHEL).

SELinux ist nicht unumstritten. Die zwei wichtigsten Kritikpunkte sind:

Kritik an SELinux

- ▶ Dateien müssen mit erweiterten Attributen (EAs, siehe [Abschnitt 11.7](#), »Access Control Lists und Extended Attributes«) gekennzeichnet werden, um ein Zusammenspiel mit SELinux zu gewährleisten. Das erfordert EA-kompatible Dateisysteme und führt oft zu Problemen bei Updates und Backups.
- ▶ Das größte Problem von SELinux ist seine riesige Komplexität. Bereits die Absicherung der wichtigsten Netzwerkdienste erfordert Tausende von Regeln. Nur wenige Experten sind in der Lage, die Wirksamkeit dieser Regeln zu beurteilen. Durchschnittliche Linux-Anwender sind nicht in der Lage, SELinux-Regeln an eigene Erfordernisse anzupassen.

SELinux wird deswegen von der Mehrheit seiner Anwender zu Recht als »Black-box« betrachtet. Die Komplexität führt beinahe zwangsläufig zu Implementierungsfehlern und verleitet dazu, das System beim ersten Problem komplett auszuschalten.

Die populärste Alternative zu SELinux ist das von SUSE und Ubuntu eingesetzte System AppArmor (siehe [Abschnitt 38.2](#)). Daneben gibt es mit »Smack« ein weiteres MAC-System im Kernel. Smack kommt primär in Embedded-Linux-Systemen zum Einsatz.

Alternativen

## SELinux-Internia und -Praxis

SELinux steht auf zwei Fundamenten: Einerseits setzt es die richtige Kennzeichnung aller Dateien und Prozesse durch einen sogenannten Sicherheitskontext voraus, andererseits beruht es auf Regeln, die von den überwachten Prozessen eingehalten werden müssen.

SELinux basiert darauf, dass jedes Objekt (z. B. Dateien) und jedes Subjekt (z. B. Prozesse) mit einem Sicherheitskontext verbunden ist. Bei Dateien wird der Dateikontext in Form von erweiterten Attributen gespeichert. Die Sicherheitsinformationen sind damit unmittelbar mit der Datei verbunden und unabhängig vom Namen der Datei. Den Sicherheitskontext einer Datei ermitteln Sie am einfachsten mit `ls -Z`. Alternativ funktioniert auch `getfattr -n security.selinux -d dateiname`.

Sicherheitskontext

```
user$ ls -Z /usr/sbin/httpd
... system_u:object_r:httpd_exec_t:s0 /usr/sbin/httpd
user$ ls -Z /etc/httpd/conf/httpd.conf
... system_u:object_r:httpd_config_t:s0 /etc/httpd/conf/httpd.conf
```

```
user$ getfattr -n security.selinux -d /etc/httpd/conf/httpd.conf
getfattr: Removing leading '/' from absolute path names
# file: etc/httpd/conf/httpd.conf
security.selinux="system_u:object_r:httpd_config_t:s0\000"
```

**chcon und restorecon**

SELinux steht und fällt damit, dass zu allen Dateien der richtige Kontext gespeichert ist. Damit dies auch funktioniert, wenn Sie nach der Installation neue Dateien erzeugen, gibt es für viele Verzeichnisse SELinux-Regeln, die den darin erzeugten neuen Dateien automatisch den passenden Kontext zuweisen. Wenn dieser Automatismus versagt, z. B. beim Verschieben von Dateien aus einem anderen Verzeichnis, können Sie den Kontext korrigieren bzw. neu einstellen.

Sofern sich Ihre Dateien in den von SELinux vorgesehenen Verzeichnissen befinden, führt `restorecon` am schnellsten zum Ziel. Durch das folgende Kommando wird der Kontext aller im `DocumentRoot`-Verzeichnis von Apache gespeicherten Dateien richtig eingestellt:

```
root# restorecon -R -v /var/www/html/*
```

Wenn Sie Dateien an einem anderen Ort gespeichert haben, z. B. HTML-Dateien im Verzeichnis `/var/myotherserver`, müssen Sie hingegen mit `chcon` den richtigen Kontext einstellen:

```
root# chcon -R system_u:object_r:httpd_sys_content_t:s0 /var/myotherserver
```

Jetzt bleibt noch eine Frage offen: Woher wissen Sie, welcher Kontext erforderlich ist? Antworten geben die `man`-Seiten aus dem Paket `selinux-policy-doc`. Zu jedem von SELinux überwachten Dienst enthält dieses Paket eine Seite, deren Name sich aus `dienst_selinux` zusammensetzt. Alle Spezialregeln, die für den Apache-Webserver gelten, können Sie daher mit `man httpd_selinux` nachlesen.

**Prozesskontext (Domäne)**

Bei Prozessen wird der Kontext oft als »Domäne« bezeichnet. Den Sicherheitskontext eines Prozesses (einer Domäne) ermitteln Sie mit `ps axZ`. Im Regelfall übernimmt ein Prozess den Kontext des Accounts, aus dem er gestartet wird. Der Kontext kann aber auch automatisch nach dem Start durch eine SELinux-Regel verändert werden. Das ist notwendig, wenn ein bestimmtes Programm (z. B. Firefox) unabhängig davon, von wem bzw. wie es gestartet wird, einen bestimmten Kontext erhalten soll.

```
user$ ps axZ | grep httpd
unconfined_u:system_r:httpd_t:s0 2373 ? Ss 0:00 /usr/sbin/httpd
unconfined_u:system_r:httpd_t:s0 2376 ? S 0:00 /usr/sbin/httpd
...
```



Der Sicherheitskontext besteht aus drei oder vier Teilen, die durch Doppelpunkte getrennt sind:

*benutzer:rolle:typ:mls-komponente*

Am wichtigsten ist der dritte Teil, der den Typ der Datei bzw. des Prozesses angibt. Die meisten SELinux-Regeln werten diese Information aus. Eine detaillierte Beschreibung aller vier Teile des Sicherheitskontexts finden Sie hier:

[https://fedoraproject.org/wiki/Security\\_context](https://fedoraproject.org/wiki/Security_context)

Die allgemeine Syntax einer typischen SELinux-Regel sieht so aus:

Regeln

```
allow type1_t type2_t:class { operations };
```

Dazu ein konkretes Beispiel: Die folgende Regel erlaubt es Prozessen, deren Kontexttyp `httpd_t` lautet, in Verzeichnissen mit dem Kontexttyp `httpd_log_t` neue Dateien zu erzeugen:

```
allow httpd_t httpd_log_t:dir create;
```

Ein typisches SELinux-Regelwerk besteht aus Zehntausenden solcher Regeln! Aus Geschwindigkeitsgründen erwartet SELinux die Regeln nicht als Text, sondern in einem binären Format. In einer Analogie zum Programmieren kann man dabei auch von einem Kompilat sprechen. Unter aktuellen Fedora- und RHEL-Versionen kommt standardmäßig das Regelwerk `targeted` zum Einsatz (Paket `selinux-policy-targeted`). Es überwacht ausgewählte Programme und Server-Dienste und ist in den unzähligen man-Seiten des Pakets `selinux-policy-devel` dokumentiert.

SELinux-Regelwerke

Alternativ kann das speziell für Server konzipierte Regelwerk `MLS (Multilevel Security)` installiert werden. Es befindet sich im Paket `selinux-policy-mls`. Das Ziel dieses Regelwerks ist es, mit RHEL eine Zertifizierung der Klasse EAL 4 zu erreichen. Diese Zertifizierung wird in den USA für bestimmte, oft militärische Anwendungen verlangt.

Mittlerweile ist Ihnen wahrscheinlich klar, dass Änderungen am Regelwerk schwierig sind. Um ein gewisses Maß der Anpassung auch ohne Regeländerungen zu ermöglichen, enthält das Regelwerk `targeted` diverse boolesche Parameter, die Sie im laufenden Betrieb verändern können. Unter CentOS/Fedora/RHEL verwenden Sie dazu am einfachsten die grafische Benutzeroberfläche `system-config-selinux` (siehe [Abbildung 38.1](#)), die im Paket `policycoreutils-gui` versteckt ist. Dieses Paket ist standardmäßig nicht installiert. Unter Fedora hatte ich in der Vergangenheit immer wieder Probleme damit, das Programm zu starten. Unter Fedora 25 gelang der Start in einer X-Session (aber nicht unter Wayland), in einer Beta-Version von Fedora 26 führte der Start zu Python-Fehlermeldungen.

SELinux-Parameter (Booleans)

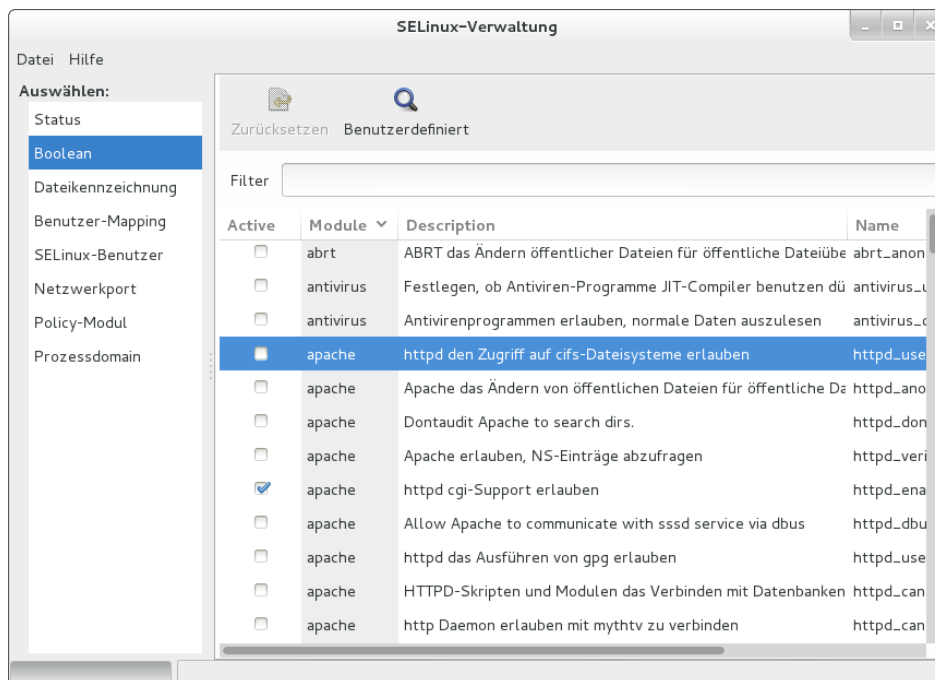


Abbildung 38.1 SELinux-Boolean-Parameter ändern

Auch mit `getsebool` können Sie den Wert boolescher Konfigurationsparameter auslesen. `setsebool` verändert derartige Parameter und erlaubt im folgenden Beispiel, dass Apache CGI-Skripts ausführen darf:

```
root# setsebool -P httpd_enable_cgi 1
```

Eine Liste aller booleschen Konfigurationsparameter ermitteln Sie mit `getsebool -a`. Als ich das zuletzt unter Fedora 26 getestet habe, waren es mehr als 300!

```
root# getsebool -a
  abrt_anon_write --> off
  abrt_handle_event --> off
  abrt_upload_watch_anon_write --> on
  antivirus_can_scan_system --> off
  ...
```

**Dokumentation** Die Regeln und booleschen Parameter für diverse Netzwerkdienste sind jeweils in eigenen `man`-Seiten dokumentiert, die aber extra installiert werden müssen:

```
root# dnf/yum install selinux-policy-doc
```

Anschließend können Sie die Dokumentation mit `man dienst_selinux` lesen, wobei Sie `dienst` durch den jeweiligen Dienstenamen ersetzen. `man httpd_selinux` liefert also

Informationen zu den SELinux-Regeln, die für den Webserver Apache gelten, man `sshd_selinux` beschreibt die Regeln für den SSH-Dämon etc. Eine gute Zusammenfassung zur Bedeutung vieler boolescher Parameter gibt diese Seite:

<https://wiki.centos.org/TipsAndTricks/SelinuxBooleans>

SELinux ist als Teil des Kernels implementiert. Ein expliziter Start durch das Init-System ist daher nicht erforderlich. Ebenso wenig gibt es einen SELinux-Dämon oder andere Hintergrundprozesse.

Start und  
Konfiguration

Die Konfiguration erfolgt durch die Dateien im Verzeichnis `/etc/selinux`. Von zentraler Bedeutung ist `/etc/selinux/config`. Die Datei gibt an, in welchem Modus SELinux läuft (*Enforcing*, *Permissive* oder *Disabled*) und welches Regelwerk gilt. Änderungen an dieser Datei werden allerdings nur durch einen Neustart wirksam.

```
# /etc/selinux/config
SELINUX=enforcing
SELINUXTYPE=targeted
```

`sestatus` ermittelt den aktuellen Status von SELinux. Auf dem Testrechner ist SELinux mit dem Regelwerk *Targeted* aktiv:

Status

```
root# sestatus
SELinux status:           enabled
SELinuxfs mount:         /sys/fs/selinux
SELinux root directory:  /etc/selinux
Loaded policy name:      targeted
Current mode:             enforcing
...
```

Grundsätzlich bestehen die folgenden Möglichkeiten, auf SELinux-Regelverstöße zu reagieren:

SELinux-Probleme  
beheben

- ▶ Sie suchen nach einem für Ihr Problem passenden Boolean-Parameter im Regelwerk und stellen diesen mit `system-config-selinux` oder `setsebool` richtig ein.
- ▶ Sie ändern die Kontextinformationen der betroffenen Dateien.
- ▶ Sie ändern bzw. erweitern das Regelwerk. Das erfordert allerdings wesentlich mehr SELinux-Kenntnisse, als in diesem Abschnitt vermittelt werden.
- ▶ Sie schalten SELinux ganz aus.

Nicht immer sind durch SELinux verursachte Probleme auf den ersten Blick zu erkennen. Wenn Sie beispielsweise einen Verzeichnisbaum mit HTML-Dateien mit `cp -a` in das Verzeichnis `/var/www/html` kopieren, können die HTML-Dateien anschließend nicht von Apache gelesen werden. Der Grund: Die `cp`-Option `-a` bewirkt, dass auch die Extended Attributes und damit die SELinux-Kontextinformationen mitkopiert werden. Dieser Umstand verhindert, dass die kopierten Dateien in `/var/www/html` durch

eine SELinux-Regel automatisch die richtigen Kontextinformationen erhalten. Diese Probleme vermeiden Sie, wenn Sie statt `cp -a` die Variante `cp -r` einsetzen.

Apache selbst weiß nichts von SELinux. Das Programm bemerkt nur, dass es nicht auf die Dateien zugreifen kann, und liefert die in die Irre führende Fehlermeldung *You don't have permission to access <filename>*. Erst ein Blick in `/var/log/messages` macht klar, dass die Zugriffsprobleme von SELinux verursacht wurden:

```
root# less /var/log/messages      (CentOS/RHEL7)
root# journalctl | grep setrouble (Fedora)
...
setroubleshoot[2685]: SELinux is preventing httpd from read access
on the file test.html. For complete SELinux messages run
sealert -l ca111078-1aba-4481-bbfc-fbd59d8ca6b9
...
```

Wenn Sie nun, wie in der Protokolldatei angegeben, `sealert` ausführen, erhalten Sie einen kunterbunt aus deutschen und englischen Begriffen zusammengewürfelten Text. Besser lesbar wird der Text, wenn Sie die Lokalisierung auf Englisch umstellen:

```
root# LANG=; sealert -l ca111078-1aba-4481-bbfc-fbd59d8ca6b9
SELinux is preventing httpd from read access on the file test.html.
```

```
** Plugin catchall_boolean (89.3 confidence) suggests
```

```
If you want to allow httpd to read user content, then you must tell
SELinux about this by enabling the 'httpd_read_user_content' boolean.
You can read 'httpd_selinux' man page for more details. Do
```

```
setsebool -P httpd_read_user_content 1
```

```
** Plugin catchall (11.6 confidence) suggests
```

```
If you believe that httpd should be allowed read access on the test.html
file by default, then you should report this as a bug ...
```

Leider sind die Lösungsvorschläge nicht wirklich zielführend. Um das Problem tatsächlich zu beheben, reicht es aus, die Kontextinformationen der betroffenen Dateien mit `restorecon` richtig einzustellen:

```
root# restorecon -R -v /var/www/html/*
```

#### SELinux deaktivieren

Um SELinux vorübergehend zu aktivieren, starten Sie `system-config-selinux` und aktivieren den Modus `PERMISSIVE`. Damit läuft SELinux weiter und protokolliert Regelverstöße in `/var/log/messages`. SELinux lässt den Regelverstoß aber zu und blockiert das betroffene Programm nicht. Dieselbe Wirkung hat auch das Kommando `setenforce 0`.

Natürlich können Sie SELinux in `system-config-selinux` auch ganz abschalten (Einstellung `DISABLED`). Das ist aber nur empfehlenswert, wenn Sie SELinux auch in Zukunft nicht mehr nutzen möchten. Der Grund: Wenn SELinux deaktiviert wird, sind auch alle Regeln außer Kraft, die neuen Dateien die SELinux-Kontextinformationen zuordnen. Wird SELinux später wieder aktiviert, verursachen die Dateien mit fehlenden Kontextinformationen Probleme. Bei der späteren Richtigstellung der Kontextdaten hilft das Kommando `restorecon`, der Prozess ist aber mühsam und fehleranfällig.

Sollte SELinux bereits während des Systemstarts Probleme verursachen, verhindert der Kernelparameter `selinux=0`, dass das SELinux-System gestartet wird. Eine Reaktivierung ist dann aber erst beim nächsten Neustart möglich. Alternativ bewirkt der Boot-Parameter `enforcing=0`, dass SELinux zwar gestartet wird, Regelübertritte aber nur protokolliert.

## 38.2 AppArmor

Anstatt das komplexe SELinux-System für die eigenen Distributionen zu adaptieren, kaufte Novell 2005 die Firma Immunix, gab deren Sicherheitslösung *Subdomain* den neuen Namen *AppArmor*, stellte sie unter die GPL und entwickelte einige YaST-Module zur Administration. Einige Jahre später erhielt AppArmor gewissermaßen den Ritterschlag der Kernelentwickler und wurde offiziell in den Kernel integriert. Danach ist bei SUSE rund um AppArmor aber Stille eingekehrt. AppArmor ist zwar weiter im Einsatz, in den letzten Jahren sind aber weder an den Regeln noch an den Administrationswerkzeugen merkbare Verbesserungen durchgeführt worden. Um die AppArmor-Weiterentwicklung kümmern sich seither vor allem von Canonical angestellte Entwickler.

AppArmor ist wie SELinux ein MAC-Sicherheitssystem (*Mandatory Access Control*). Im Unterschied zu SELinux basieren AppArmor-Regeln auf absoluten Dateinamen. Daher ist eine eigene Kennzeichnung aller Dateien durch EAs nicht erforderlich; zudem funktioniert AppArmor auch für Dateisysteme, die keine EAs unterstützen. In den AppArmor-Regeln sind Jokerzeichen erlaubt. Aus diesem Grund kommt AppArmor für typische Anwendungsfälle mit wesentlich weniger Regeln aus als SELinux.

Naturgemäß gibt es auch Argumente, die gegen AppArmor sprechen:

Kritik

- ▶ Sicherheitsexperten von Red Hat sind der Meinung, dass absolute Pfade in den Regeln ein inhärentes Sicherheitsrisiko sind. Der Schutz von AppArmor kann durch das Umbenennen von Dateien oder Verzeichnissen umgangen werden – was natürlich nur gelingt, wenn ein Angreifer dazu bereits ausreichende Rechte hat.

- ▶ Das Regelwerk für AppArmor ist nicht so umfassend wie das von SELinux. Standardmäßig werden weniger Programme geschützt. Zwar ist es einfacher als bei SELinux, selbst Regeln zu erstellen bzw. zu ändern, aber diese Art der Do-it-yourself-Sicherheit hinterlässt einen wenig professionellen Eindruck.

**Links** Dieser Abschnitt gibt nur eine Einführung zu AppArmor. Weitere Informationen finden Sie hier:

<https://www.suse.com/de-de/documentation/sles11>

<https://wiki.ubuntu.com/AppArmor>

<https://wiki.ubuntu.com/SecurityTeam/KnowledgeBase/AppArmorProfiles>

### AppArmor unter Ubuntu

Die folgenden Ausführungen beziehen sich auf AppArmor, wie es unter Ubuntu aktiv ist. SUSE-spezifische Anmerkungen folgen am Ende dieses Abschnitts.

AppArmor ist unter Ubuntu standardmäßig im Kernel integriert. Das Sicherheitssystem wird durch systemd gestartet, berücksichtigt die Grundkonfiguration aus dem Verzeichnis `/etc/apparmor` und lädt alle Regeldateien aus dem Verzeichnis `/etc/apparmor.d`.

Beim Start von AppArmor wird das Dateisystem `securityfs` in das Verzeichnis `/sys/kernel/security` eingebunden. Seine Dateien geben Auskunft über aktive Profile, die Anzahl der aufgetretenen Regelverletzungen etc.

**Status** Das Kommando `aa-status` gibt einen Überblick über den gegenwärtigen Zustand von AppArmor. Das Kommando liefert sowohl eine Liste aller Profile als auch eine Liste der tatsächlich überwachten Prozesse. Die folgende Liste zeigt an, welche Dienste eines Ubuntu-Root-Servers überwacht werden:

```
root# aa-status
apparmor module is loaded.
12 profiles are loaded.
12 profiles are in enforce mode.
  /sbin/dhclient
  /usr/bin/lxc-start
  ...
  lxc-container-default-with-nesting
0 profiles are in complain mode.
1 processes have profiles defined.
1 processes are in enforce mode.
  /usr/sbin/mysqld (1145)
0 processes are in complain mode.
0 processes are unconfined but have a profile defined.
```

Zusammengefasst bedeutet das: Es gibt nur ein einziges aktives Programm, das tatsächlich unter der Kontrolle von AppArmor steht, und das ist der MySQL-Server. Es gibt zwar weitere AppArmor-Profile, aber da die betreffenden Programme nicht laufen, bleiben diese Profile wirkungslos. Umgekehrt laufen auf dem Server natürlich diverse andere Server-Dienste, die überwacht werden sollten (Apache, Dovecot, Postfix, SpamAssassin, ein SSH-Server) – aber für die gibt es wiederum keine offiziellen AppArmor-Regelprofile. Im Vergleich zu einem SELinux-System ist die Absicherung also mehr als spärlich.

Das Bild ändert sich ein wenig, wenn Sie AppArmor und SELinux auf einem Desktop-System miteinander vergleichen. Dabei zeigt sich, dass AppArmor diverse Desktop-Programme absichert, während SELinux sich überwiegend auf System- und Server-Dienste konzentriert.

Die Wirkung von AppArmor steht und fällt mit den Überwachungsregeln. Diese Regeln werden auch »Profile« genannt und befinden sich in den Dateien des Verzeichnisses `/etc/apparmor.d/`. Beispielsweise enthält die Datei `usr.sbin.cupsd` die Profile für CUPS und den CUPS-PDF-Treiber.

Regeln (Profile)

Die offiziell gewarteten Regelprofile werden üblicherweise vom jeweiligen Paket zur Verfügung gestellt. Das Regelprofil `usr.sbin.cupsd` steht also nur dann zur Verfügung, wenn Sie das Druckersystem CUPS installiert haben. Aus diesem Grund ist `/etc/apparmor.d` nach einer Ubuntu-Server-Installation anfänglich fast leer und füllt sich erst in dem Ausmaß, in dem Sie Server-Dienste installieren.

Außerdem können Sie das Paket `apparmor-profiles` aus der `universe`-Paketquelle installieren. Es enthält zahlreiche weitere Profile, die aber nicht offiziell unterstützt und gewartet werden. Die meisten Profile laufen nur im sogenannten `complain`-Modus: In diesem Modus werden Regelverstöße zwar protokolliert, aber nicht geahndet. Mit den Kommandos `aa-enforce` und `aa-complain` aus dem Paket `apparmor-utils` können Sie den Modus eines Profils ändern. An die beiden Kommandos übergeben Sie den vollständigen Pfad des zu überwachenden Programms:

```
root# aa-enforce /usr/sbin/dnsmasq
Setting /usr/sbin/dnsmasq to enforce mode.
root# aa-complain /usr/sbin/dnsmasq
Setting /usr/sbin/dnsmasq to complain mode.
```

Alternativ können Sie an `aa-enforce` und `aa-complain` auch die Dateinamen der Profildateien übergeben. Das macht es einfach, den Modus mehrerer Profile auf einmal zu verändern:

```
root# cd /etc/apparmor.d
root# aa-enforce usr.lib.dovecot*
```

Bei Server-Diensten müssen Sie nach einer Aktivierung eines AppArmor-Profiles auch das jeweilige Programm neu starten:

```
root# systemctl restart <name>
```

Losgelöst vom AppArmor-Modus sind Profile natürlich nur dann relevant, wenn das betreffende Programm tatsächlich ausgeführt wird. Welche Programme momentan aktiv überwacht werden, verrät das oben erwähnte Kommando `aa-status`.

#### Aufbau von Regeldateien

Regeldateien, die bei AppArmor *Profile* heißen, liegen in einem einfachen Textformat vor. Die folgenden Zeilen zeigen die AppArmor-Regeln für `mysqld`:

```
# Datei /etc/apparmor.d/usr.sbin.mysqld
#include <tunables/global>

/usr/sbin/mysqld {
  #include <abstractions/base>
  #include <abstractions/nameservice>
  #include <abstractions/user-tmp>
  #include <abstractions/mysql>
  #include <abstractions/winbind>
  capability dac_override,
  capability sys_resource,
  capability setgid,
  capability setuid,
  network tcp,
  /etc/hosts.allow          r,
  /etc/hosts.deny          r,
  /etc/hosts.allow         r,
  /etc/hosts.deny         r,
  /etc/mysql/**            r,
  /usr/lib/mysql/plugin/   r,
  /usr/lib/mysql/plugin/*.so* mr,
  /usr/sbin/mysqld        mr,
  /usr/share/mysql/**      r,
  /var/log/mysql.log       rw,
  /var/log/mysql.err       rw,
  /var/lib/mysql/         r,
  /var/lib/mysql/**       rwk,
  /var/log/mysql/         r,
  /var/log/mysql/*        rw,
  /var/run/mysqld/mysqld.pid rw,
  /var/run/mysqld/mysqld.sock w,
  /run/mysqld/mysqld.pid  rw,
  /run/mysqld/mysqld.sock w,
  /sys/devices/system/cpu/ r,
  # Site-specific additions and overrides. See local/README for details.
  #include <local/usr.sbin.mysqld>
}
```



In den Regeldateien werden zuerst einige Include-Dateien gelesen und dann grundlegende Merkmale (siehe `man capabilities`) des Programms festgelegt. Die weiteren Regeln geben an, welche Dateien das Programm wie nutzen darf.

In den AppArmor-Regeldateien gilt das Jokerzeichen `*` als Platzhalter für eine beliebige Anzahl von Zeichen. `**` hat eine ähnliche Bedeutung, schließt aber das Zeichen `/` ein und umfasst damit auch Dateien in allen Unterverzeichnissen. Die Zugriffsrechte werden durch Buchstaben oder Buchstabenkombinationen ausgedrückt, deren Bedeutung im AppArmor-Administration-Manual genau beschrieben ist. [Tabelle 38.1](#) beschreibt die wichtigsten Buchstaben. Die `?x`-Kombinationen steuern die Rechte von Sub-Prozessen, die das Hauptprogramm startet.

| Kürzel | Bedeutung   |
|--------|---|
| r      | erlaubt Lesezugriffe (read).  |
| w      | erlaubt Schreibzugriffe (write).  |
| a      | erlaubt es, die Datei zu erweitern (append).                                  |
| l      | wendet auf harte Links dieselben Regeln wie für die Ursprungsdatei an (link). |
| k      | erlaubt es, die Datei zu blockieren (lock).                                   |
| m      | lässt die <code>mmap</code> -Funktion zu (allow executable mapping).          |
| ix     | Das Programm erbt die Regeln des Basisprogramms (inherent execute).           |
| px     | Das Programm hat ein eigenes AppArmor-Profil (discrete profile execute).      |
| ux     | führt das Programm ohne AppArmor-Regeln aus (unconstrained execute).          |

**Tabelle 38.1** Elementare AppArmor-Zugriffsrechte

AppArmor sieht einen Mechanismus vor, um einzelne Parameter der Regeln auf eine einfache Weise zu verändern. Diese Parameter sind in den Dateien des Verzeichnisses `/etc/apparmor.d/tunables` definiert. In der aktuellen Implementierung gibt es allerdings nur wenige Parameter, mit denen Sie beispielsweise den Ort der Heimatverzeichnisse individuell einstellen können. Wenn Ihr Server außer `/home` auch andere Orte für Heimatverzeichnisse vorsieht, müssen Sie die Variable `@{HOMEDIRS}` verändern. Im folgenden Beispiel wurden dort die beiden zusätzlichen Verzeichnisse `/home1` und `/myhome` hinzugefügt:

```
# Datei /etc/apparmor.d/tunables/home
@{HOME}=@{HOMEDIRS}/*/ /root/
@{HOMEDIRS}=/home/ /home1/ /myhome/
```

Details über Regelverletzungen, die im `complain`- oder `enforce`-Modus stattfanden, werden in Form von Kernelmeldungen weitergegeben und standardmäßig in den Dateien `/var/log/kern.log` und `/var/log/syslog` aufgezeichnet. Sie erkennen AppArmor-Meldungen am Schlüsselwort `audit`:

Regelparameter  
(`tunables`)

Logging und  
Wartung

```

root# grep audit /var/log/kern.log
[...] audit(1238580174.435:3): type=1503 operation="inode_permission"
        requested_mask="a::" denied_mask="a::" name="/dev/tty"
        pid=6345 profile="/usr/sbin/cupsd" namespace="default"
[...] audit(1238580174.435:4): type=1503 operation="inode_permission"
        requested_mask="w::" denied_mask="w::" name="/etc/krb5.conf"
        pid=6345 profile="/usr/sbin/cupsd" namespace="default"

```

Oft sind die Audit-Meldungen ein Indikator dafür, dass die AppArmor-Regeln unvollständig sind. Ein Fehlverhalten des Programms ist natürlich auch möglich, aber eher unwahrscheinlich. Mit Sicherheit kann das nur ein Experte für das jeweilige Programm beurteilen. Insofern ist eine angemessene Reaktion auf Regelübertretungen schwierig.

Wenn Sie vermuten, dass das betroffene Programm ordnungsgemäß funktioniert, sollten Sie das Profil in den `complain`-Modus umschalten und die Audit-Meldung im Ubuntu-Bug-System melden (<https://bugs.launchpad.net>). Sie können auch versuchen, das Profil um eine Regel zu erweitern, die den gemeldeten Vorgang erlaubt. Oder Sie ignorieren die Meldung einfach, sofern das betroffene Programm anstandslos weiterläuft.

### AppArmor unter SUSE

#### Konfiguration und Start

AppArmor ist in aktuellen openSUSE- und SUSE-Enterprise-Distributionen standardmäßig installiert und wird durch `systemd` gestartet. YaST enthält ein Modul zur AppArmor-Konfiguration. Dessen Funktionalität reicht aber nicht weit über die der Kommandos `aa-status`, `aa-enforce` und `aa-complain` hinaus. Einige Dialogpunkte führen sogar zum Absturz des Moduls.

TEIL IX

## **Virtualisierung & Co.**



# Kapitel 39

## VirtualBox und Vagrant

Virtualisierung macht es möglich, auf einem Rechner mehrere Betriebssysteme parallel auszuführen. Daraus ergeben sich unzählige Anwendungen: Sie können Linux unter Windows ausprobieren oder Windows unter Linux ausführen, eine neue Alpha-Version der Distribution xyz gefahrlos testen, ohne die vorhandene Linux-Installation zu gefährden, Server-Funktionen sicher voneinander trennen etc.

Das für die Plattformen Windows, Linux und macOS verfügbare Programm VirtualBox eignet sich am besten zur Desktop-Virtualisierung, also zur Ausführung von virtuellen Maschinen, die im Grafiksystem bedient werden sollen. Hinter VirtualBox stand ursprünglich die deutsche Firma InnoTek. 2008 übernahm Sun InnoTek, und 2010 kaufte Oracle Sun. Damit ist nun Oracle der Eigentümer von VirtualBox. Umfassende Dokumentation zu VirtualBox finden Sie unter:

<https://www.virtualbox.org>

Große Teile von VirtualBox bestehen aus Open-Source-Code. Die einzige Ausnahme sind einige Zusatzfunktionen, die extra installiert werden müssen. Ihre Nutzung ist für Privatanwender ebenfalls kostenlos, für kommerzielle Anwender hingegen kostenpflichtig.

Dieses Kapitel beschreibt, wie Sie VirtualBox unter Linux installieren und darin virtuelle Maschinen ausführen. Mit Einschränkungen ist VirtualBox auch zur Server-Virtualisierung geeignet. Für diesen Zweck ist das Programm KVM, das ich Ihnen in [Kapitel 40](#) näher vorstelle, wesentlich besser geeignet.

Das Einrichten neuer virtueller Maschinen ist mit Arbeit verbunden. Mit Vagrant können Sie diesen Prozess automatisieren. Das ist vor allem dann praktisch, wenn Sie reproduzierbar Testumgebungen aufsetzen möchten, z. B. für eine bestimmte Server-Konfiguration. Vagrant kann derartige Aufgaben auch für andere Virtualisierungssysteme erledigen. VirtualBox eignet sich aber besonders gut, um Vagrant kennenzulernen.

**Vagrant**

## 39.1 VirtualBox installieren

Zur Installation von VirtualBox gibt es grundsätzlich zwei Möglichkeiten. Die bequeme Variante besteht darin, einfach die VirtualBox-Pakete zu verwenden, die sich in den Paketquellen Ihrer Distribution befinden. Sollten diese Pakete fehlen oder nicht ausreichend aktuell sein, können Sie VirtualBox selbst von der Webseite <https://www.virtualbox.org> herunterladen und manuell installieren. Das ist ein wenig umständlicher.

In diesem Abschnitt gehe ich auf beide Varianten ein. Losgelöst davon sind nach Abschluss der Installation noch einige Vorbereitungsarbeiten zu erledigen, die ich am Ende dieses Abschnitts erläutere.

### Host und Gast

Bei der Beschreibung von Virtualisierungssystemen hat es sich eingebürgert, das Grundsystem als Wirt (*Host*) und die darauf laufenden virtuellen Maschinen als Gäste (*Guests*) zu bezeichnen. In diesem Kapitel gehe ich davon aus, dass der Host ein bereits funktionierendes Linux-System ist.

### VirtualBox-Pakete Ihrer Distribution

#### VirtualBox unter Linux installieren

Die meisten Distributionen bieten fertige VirtualBox-Pakete an. Bei Fedora müssen Sie vorher die `rpmfusion`-Paketquelle aktivieren. Bei openSUSE befinden sich die Kernfunktionen und die Benutzeroberfläche in getrennten Paketen; dort müssen Sie auch das Paket `virtualbox-qt` installieren.

Nicht erforderlich sind hingegen die diversen `virtualbox-guest`-Pakete! Diese Pakete enthalten Treiber, die *in* virtuellen Maschinen auszuführen sind, also wenn eine Linux-Distribution selbst innerhalb von VirtualBox ausgeführt werden soll.

#### VirtualBox-Kernelmodule

VirtualBox greift auf dem Wirtssystem auf die vier Kernelmodule `vboxdrv`, `vboxpci`, `vboxnetadp` und `vboxnetflt` zurück. Manche Distributionen stellen diese Module in binärer Form durch ein eigenes Paket zur Verfügung, das bei jedem Kernel-Update aktualisiert wird. Bei openSUSE lautet der Paketname `virtualbox-host-kmp-default`.

**DKMS** Bei anderen Distributionen wird der Quellcode der VirtualBox-Pakete installiert. Bei jedem Kernel-Update müssen die entsprechenden VirtualBox-Module neu kompiliert werden. Darum kümmert sich bei einigen Distributionen DKMS (*Dynamic Kernel Module Support*). Dies ist z. B. bei Ubuntu der Fall, wo Sie das Paket `virtualbox-dkms` installieren müssen.

Die RPMFusion-Paketquelle für Fedora sieht anstelle von DKMS das Kommando `akmods` (Fedora) vor. Zur erstmaligen Installation der VirtualBox-Kernelmodule aktivieren Sie zuerst die RPMFusion-Paketquellen und führen dann die folgenden Kommandos aus:

```
root# dnf install akmod-VirtualBox kernel-devel-$(uname -r)
root# akmods
root# systemctl restart systemd-modules-load
```

Das erste Kommando installiert die erforderlichen Pakete, das zweite kompiliert die Kernelmodule, das dritte lädt sie. In der Zukunft sollte sich `akmods` nach jedem Kernel-Update selbstständig um die Aktualisierung der VirtualBox-Treiber kümmern. Bei meinen Tests hat das häufig nicht funktioniert. Sie müssen dann die obigen drei Kommandos neuerlich ausführen.

Fedora bereitet auch sonst als Gast in VirtualBox oft Probleme. Das liegt daran, dass Fedora häufig die allerneuesten xorg-Grafiktreiber verwendet. Diese sind nicht immer mit VirtualBox kompatibel. Das kann dazu führen, dass Sie trotz installierter Gasttreiber nur mit einer Bildschirmauflösung von 1024 × 768 Pixel arbeiten können.

Steht weder DKMS noch `akmods` zur Verfügung, können Sie die Module durch ein Script manuell kompilieren: **Manuell kompilieren**

```
root# /usr/lib/virtualbox/vboxdrv.sh setup
```

Zum Kompilieren sind aber auch der C-Compiler `gcc` sowie die Kernel-Header-Dateien erforderlich. Bei vielen Distributionen müssen Sie die entsprechenden Pakete vorher installieren (siehe [Abschnitt 25.3](#), »Kernelmodule selbst kompilieren«).

Ob das Kompilieren und Laden der VirtualBox-Kernelmodule funktioniert hat, prüfen Sie mit dem folgenden Kommando: **Test**

```
root# lsmod | grep vbox
vboxpci                24576  0
vboxnetadp             28672  0
vboxnetflt             28672  0
vboxdrv                434176  3 vboxnetadp,vboxnetflt,vboxpci
```

### VirtualBox-Pakete von Oracle

Statt der mit Ihrer Distribution mitgelieferten VirtualBox-Pakete können Sie auch die von Oracle zum Download angebotene Version installieren. Das ist vor allem dann zweckmäßig, wenn Oracle eine neuere VirtualBox-Version anbietet als Ihre Distribution.

[https://www.virtualbox.org/wiki/Linux\\_Downloads](https://www.virtualbox.org/wiki/Linux_Downloads)

Auf der obigen Website finden Sie VirtualBox in verschiedenen Formaten: als RPM- und Debian-Paket für diverse Distributionen sowie als Universal-Installer, den Sie wie folgt starten:

```
root# chmod u+x VirtualBox_nnn.run install
root# ./VirtualBox_nnn.run install
```

**Kernelmodule** Nach Möglichkeit sollten Sie vor VirtualBox das `dkms`-Paket Ihrer Distribution installieren. In diesem Fall verwaltet DKMS die VirtualBox-Kernelmodule und kümmert sich bei Kernel-Updates automatisch um eine Neukompilierung. Bei meinen VirtualBox-Installationen hat das allerdings nicht immer zuverlässig funktioniert.

Wenn DKMS nicht zur Verfügung steht bzw. versagt, kompilieren Sie die Kernelmodule selbst. Wie vorhin schon erwähnt, müssen Sie gegebenenfalls vorher den C-Compiler und die Kernel-Header-Dateien oder den Kernel-Quellcode installieren.

```
root# /usr/lib/virtualbox/vboxdrv.sh setup
```

**APT-Paketquelle** Für Debian- und Ubuntu-Anwender gibt es eine eigene APT-Paketquelle. Gegenüber der manuellen Installation eines einzelnen Pakets hat die Paketquelle den Vorteil, dass Sie innerhalb der gewählten Major-Version automatisch Updates erhalten. Dazu fügen Sie zu `/etc/apt/sources.list` eine der folgenden Zeilen hinzu:

```
deb https://download.virtualbox.org/virtualbox/debian stretch contrib
deb https://download.virtualbox.org/virtualbox/debian xenial contrib
```

Anstelle von `stretch` bzw. `xenial` müssen Sie den Codenamen der von Ihnen eingesetzten Debian- bzw. Ubuntu-Distribution verwenden. Werfen Sie gegebenenfalls einen Blick in die Datei `/etc/os-release`.

Außerdem führen Sie diese beiden Kommandos aus, um den Schlüssel der Paketquelle zu installieren:

```
root# wget -q https://www.virtualbox.org/download/oracle_vbox_2016.asc
root# apt-key add oracle_vbox_2016.asc
```

Anschließend installieren Sie VirtualBox mit `apt` oder `apt-get`:

```
root# apt update
root# apt install virtualbox-5.1
```

**Yum-Paketquelle** Für Anwender von Yum-kompatiblen Distributionen (CentOS, Fedora, openSUSE, Red Hat etc.) gibt es analog eine Yum-Paketquelle. Auch in diesem Fall müssen Sie zuerst den Schlüssel importieren:

```
root# wget -q https://www.virtualbox.org/download/oracle_vbox.asc
root# rpm --import oracle_vbox.asc
```



Anschließend laden Sie die für Ihre Distribution passende \*.repo-Datei von der VirtualBox-Download-Seite herunter und kopieren sie in das Verzeichnis `/etc/yum.repos.d`. Die folgenden Zeilen zeigen die Fedora-Variante der \*.repo-Datei:

```
# Datei /etc/yum.repos.d/virtualbox.repo
[virtualbox]
name=Fedora $releasever - $basearch - VirtualBox
baseurl=http://download.virtualbox.org/virtualbox/rpm/fedora/$releasever/$basearch
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://www.virtualbox.org/download/oracle_vbox.asc
```

Die VirtualBox-Installation führen Sie nun mit `dnf install` oder `yum install` oder `zypper install` durch.

### Vorbereitungsarbeiten

VirtualBox richtet für jede virtuelle Maschine ein Unterverzeichnis innerhalb von `VirtualBox VMs` ein. In mehreren Dateien werden dort die Einstellungen der virtuellen Maschine sowie die virtuelle Festplatte gespeichert. Mit `DATEI • GLOBALE EINSTELLUNGEN` können Sie gegebenenfalls einen anderen Speicherort einstellen.

Speicherort für virtuelle Maschinen

Oracle bietet auf seiner Website ein sogenanntes Extension Pack zum Download an. Beim Download des Extension Packs schlägt der Webbrowser vor, die Datei direkt mit VirtualBox zu öffnen. Diesem Vorschlag folgen Sie einfach.

Extension Pack

Das Extension Pack ergänzt VirtualBox um einige Zusatzfunktionen: Unter anderem können Sie dann in den virtuellen Maschinen auf USB-Geräte (USB-2 und USB-3), PCI-Karten und Webcams zugreifen und die virtuellen Maschinen via RDP (Remote Display Protocol) auf einem anderen Rechner im Netzwerk steuern. Diese Erweiterungen werden nur in Binärform vertrieben, es handelt sich also nicht um Open-Source-Code. Die kommerzielle Nutzung dieser Erweiterungen erfordert eine Lizenz von Oracle!

Unabhängig davon, aus welcher Quelle Ihre VirtualBox-Installation stammt, wurde die Gruppe `vboxusers` eingerichtet. Nur Benutzer, die dieser Gruppe angehören, können in virtuellen Maschinen auf USB-Geräte zugreifen. Deswegen müssen Sie vor dem ersten Start von VirtualBox Ihren Account der Gruppe `vboxusers` hinzufügen. Ersetzen Sie beim folgenden Kommando `kofler` durch Ihren Login-Namen:

vboxusers-Gruppe

```
root# usermod -a -G vboxusers kofler
```

Damit die geänderte Gruppenzuordnung wirksam wird, müssen Sie sich aus- und neu einloggen. Anschließend starten Sie die Benutzeroberfläche von VirtualBox über das KDE- oder Gnome-Menü bzw. mit dem Kommando `VirtualBox`.

## VirtualBox unter Windows oder macOS installieren

Die Installation von VirtualBox unter Windows oder macOS ist grundsätzlich ein Kinderspiel: Sie laden das passende Setup-Programm von der VirtualBox-Seite herunter und führen es aus. Das Extension Pack muss auch in diesem Fall extra heruntergeladen und eingerichtet werden.

Unter Windows kann es allerdings passieren, dass VirtualBox nicht richtig funktioniert: Virtuelle Maschinen lassen sich dann nur im 32-Bit-Modus einrichten oder können gar nicht gestartet werden. Schuld daran ist in der Regel, dass VirtualBox die Virtualisierungstechnik VT nicht nutzen kann, die in viele Intel-CPU's integriert ist.

Dafür kann es mehrere Ursachen geben: Am wahrscheinlichsten ist es, dass Windows die Funktion durch Hyper-V blockiert. Abhilfe: Starten Sie das Programm WINDOWS-FEATURES, suchen Sie nach HYPER-V und deaktivieren Sie die Option. Danach muss der Rechner neu gestartet werden. Sollte Hyper-V nicht schuld sein, ist VT möglicherweise im BIOS/EFI deaktiviert. Und natürlich kann es auch sein, dass Ihre CPU die Funktion wirklich nicht enthält.

## 39.2 VirtualBox-Maschinen einrichten

Ist VirtualBox einmal installiert, können Sie mit dem Einrichten virtueller Maschinen beginnen. Dieser Abschnitt berücksichtigt sowohl Linux- als auch Windows-Gäste.

### Eine virtuelle Maschine mit Linux einrichten

Dieser Abschnitt beschreibt, wie Sie innerhalb von VirtualBox eine virtuelle Maschine mit Linux einrichten. Dabei spielt es keine Rolle, ob VirtualBox selbst unter Linux, Windows oder macOS läuft.

Beim Einrichten einer neuen virtuellen Maschine unterstützt Sie ein Assistent. Als Betriebssystemtyp stehen neben Windows diverse Linux-Distributionen zur Auswahl. Wenn Ihre Distribution nicht vertreten ist, wählen Sie LINUX MIT KERNEL 2.6 / 3.X / 4.X; diese Einstellung gilt für alle aktuellen Kernelversionen. Achten Sie darauf, dass es für jedes Betriebssystem *zwei* Versionen gibt: eine für 32- und eine für 64-Bit-Installationen. Wählen Sie den passenden Eintrag!

VirtualBox sieht standardmäßig 1 GiB RAM für virtuelle Linux-Maschinen vor. Viele Desktop-Distributionen laufen flüssiger, wenn Sie etwas mehr RAM spendieren.

Als Nächstes müssen Sie eine virtuelle Festplatte einrichten. Der Datenträger wird als Image-Datei im Host-Dateisystem gespeichert. Dazu stehen verschiedene Forma-

te zur Auswahl. Im Regelfall sollten Sie beim VirtualBox-eigenen Format VDI bleiben und auch die Option `DYNAMISCH ALLOZIERT` beibehalten. Damit wird der Speicherplatz für die Festplatte erst nach und nach angefordert. Die Alternative `FESTE GRÖSSE` bedeutet, dass der gesamte Speicherplatz sofort vorreserviert wird.

Die vorgeschlagenen 8 GiB sind allerdings arg knapp bemessen. Bei vielen Distributionen reicht das nicht einmal für eine Minimalinstallation aus. Stellen Sie zumindest 16 GiB ein.

Schließlich zeigt VirtualBox eine Zusammenfassung aller Hardware-Komponenten an. Mit `ÄNDERN` können Sie nun bei Bedarf weitere Einstellungen durchführen, z. B. den Netzwerkzugang verändern oder im Dialogblatt `MASSENSPEICHER` eine ISO-Datei als Datenquelle für das DVD-Laufwerk auswählen.

Wenn Sie mit der Konfiguration fertig sind, starten Sie die virtuelle Maschine. Das von der ISO-Datei geladene Linux-Installationsprogramm erscheint in einem eigenen Fenster. Dort installieren Sie Linux wie auf einem realen Rechner.

#### Mögliche Fehlermeldungen beim ersten Start einer virtuellen Maschine

VirtualBox testet erst mit dem Start einer virtuellen Maschine, ob die VirtualBox-Kernelmodule geladen sind und ob Hardware-Virtualisierungsfunktionen zur Verfügung stehen. Ist eine dieser Voraussetzungen nicht erfüllt, wird eine Fehlermeldung oder Warnung angezeigt. Bei den Kernelmodulen müssen Sie sicherstellen, dass diese installiert sind. Wenn Sie VirtualBox frisch installiert haben, hilft es oft, das Script `/usr/lib/virtualbox/vboxdrv.sh setup` zum Neukompilieren der Module auszuführen. Denken Sie auch daran, dass die Hardware-Virtualisierungsfunktionen im BIOS oder EFI aktiviert sein müssen.

Die virtuelle Maschine erhält automatisch den Tastatur- und Mausfokus, sobald Sie eine Taste drücken. Standardmäßig lösen Sie den Fokus mit der rechten `[Strg]`-Taste. Im VirtualBox-Hauptfenster können Sie mit `DATEI • EINSTELLUNGEN • EINGABE • VIRTUELLE MASCHINE` eine andere »Host«-Taste einstellen. Die gerade gültige Kombination wird rechts in der Statusleiste des VirtualBox-Fensters angezeigt. Die wichtigsten Host-Tastenkombinationen sind in [Tabelle 39.1](#) zusammengefasst.

Host-Tasten-  
kombination

Nachdem die eigentliche Installation abgeschlossen ist, sollten Sie in der virtuellen Maschine noch die sogenannten Guest Additions installieren. Sie stellen dem Gastsystem zusätzliche Treiber zur Verfügung und verbessern das Zusammenspiel mit dem Wirt: Die Maus kann nun aus der virtuellen Maschine herausbewegt werden, die virtuelle Bildschirmauflösung des Gasts passt sich automatisch an die Fenstergröße an, der Datenaustausch mit dem Wirtssystem kann über Shared Folders erfolgen, Text kann über die Zwischenablage kopiert werden etc.

Gasterweiterungen  
installieren

| Tastenkürzel                          | Bedeutung   |
|---------------------------------------|---|
| <code>Host</code>                     | Tastatur- und Mausfokus lösen   |
| <code>Host</code> + <code>F</code>    | Vollbildmodus (de)aktivieren  |
| <code>Host</code> + <code>Entf</code> | <code>Strg</code> + <code>Alt</code> + <code>Entf</code> an das Gastsystem senden |
| <code>Host</code> + <code>←</code>    | <code>Strg</code> + <code>Alt</code> + <code>←</code> an das Gastsystem senden    |
| <code>Host</code> + <code>Fn</code>   | <code>Strg</code> + <code>Alt</code> + <code>Fn</code> an das Gastsystem senden   |
| <code>Host</code> + <code>S</code>    | Snapshot der virtuellen Maschine erstellen  |
| <code>Host</code> + <code>H</code>    | virtuelle Maschine per ACPI ausschalten   |
| <code>Host</code> + <code>R</code>    | virtuelle Maschine sofort ausschalten (Reset, Vorsicht!)                          |

Tabelle 39.1 VirtualBox-Tastenkürzel

Manche Distributionen liefern fertige Pakete mit den VirtualBox-Gasterweiterungen mit. Bei openSUSE werden sie sogar gleich automatisch installiert. Allerdings sind diese Pakete selten auf dem aktuellen Stand. Sie bezahlen die Bequemlichkeit der Installation also möglicherweise mit Inkompatibilitäten zu der von Ihnen eingesetzten aktuelleren VirtualBox-Version.

Debian, Ubuntu:            `virtualbox-guest-dkms`, `virtualbox-guest-utils`,  
                                   `virtualbox-guest-x11`  
 Fedora mit RPMFusion: `VirtualBox-guest-additions`  
 openSUSE:                    `virtualbox-guest-kmp-default`, `virtualbox-guest-tools`,  
                                   `virtualbox-guest-x11`

Bei anderen Distributionen bzw. dann, wenn Sie die neueste Version der Gasterweiterungen benötigen, müssen Sie eine manuelle Installation durchführen. Dazu werfen Sie eine eventuell eingebundene CD/DVD aus und führen dann im VirtualBox-Fenster `GERÄTE • GASTERWEITERUNGEN EINLEGEN` aus. Im Regelfall erscheint nach einigen Sekunden in der virtuellen Maschine ein Dateimanagerfenster, in dem Sie `autorun.sh` starten. Sollte das nicht funktionieren, helfen die folgenden Kommandos weiter:

```
root# mkdir /media/cdrom
root# mount /dev/sr0 /media/cdrom
root# sh /media/cdrom/autorun.sh
```

Das Installationsprogramm richtet nun die drei neuen Kernelmodule `vboxadd`, `vboxvideo` und `vboxvifs` sowie einen neuen X-Treiber ein und fügt einige Init-Scripts hinzu, damit diese Gasterweiterungen beim nächsten Start der virtuellen Maschine auch verwendet werden.

Unter Ubuntu funktioniert die Installation der Gasterweiterungen auf Anhieb. Bei den meisten anderen Linux-Distributionen müssen Sie vor der Installation der Gasterweiterungen diverse Pakete installieren, die den C-Compiler und die Kernel-Header-Dateien enthalten. Führen Sie vorher ein Update aus, um sicherzustellen, dass die installierte Kernelversion und die Version der Kernel-Header-Dateien zusammenpassen!

```
root# yum      install gcc make kernel-headers kernel-devel    (CentOS)
root# apt      install gcc make linux-headers-platform        (Debian)
root# dnf      install gcc make kernel-headers kernel-devel    (Fedora)
root# zypper   install gcc make kernel-source kernel-syms      (openSUSE)
```

Fedora verwendet mitunter ganz aktuelle Versionen des Xorg-Servers, zu denen der VirtualBox-Grafiktreiber noch nicht kompatibel ist. Abhilfe schafft dann unter Umständen ein Downgrade auf eine ältere Xorg-Version vor der Installation der Gasttreiber. Das folgende Kommando zeigt, wie Sie unter Fedora 26 die Grafiktreiber von Fedora 25 installieren:

```
root# dnf --showduplicates --allowerasing --releasever=25 \
      downgrade xorg-x11-server-Xorg
```

Im Idealfall stehen innerhalb der virtuellen Maschine sogar 3D-Funktionen zur Verfügung. Dazu müssen auf jeden Fall die Gasterweiterungen aktiv sein, außerdem müssen die 3D-Funktionen in den Eigenschaften der virtuellen Maschine aktiviert sein (Dialogblatt ANZEIGE, Option 3D-BESCHLEUNIGUNG). Gleichzeitig sollten Sie den Grafikspeicher auf zumindest 64 MiB stellen (siehe [Abbildung 39.1](#)).

3D-Grafik

Das allein ist aber nicht in jedem Fall ausreichend – ob 3D-Funktionen an den Gast weitergereicht werden können, hängt auch davon ab, in welchem Host-Betriebssystem VirtualBox an sich läuft und welchen Grafiktreiber Sie im Host-System verwenden. Recht gute Erfahrungen habe ich mit Linux-Hosts in Kombination mit dem Intel-Grafiktreiber gemacht. In vielen anderen Fällen, insbesondere auch, wenn VirtualBox unter macOS läuft, funktionierte die 3D-Unterstützung gar nicht. Und selbst wenn die 3D-Funktionen prinzipiell durchgereicht werden, können fallweise Fehldarstellungen auftreten, z. B. nach der Veränderung der Fenstergröße.

Wenn Sie sich vergewissern möchten, ob alles funktioniert, installieren Sie in der virtuellen Maschine je nach Distribution das Paket `mesa-utils`, `glx-utils` oder `Mesa-demo-x` und führen dann `glxinfo` aus. Das Ergebnis sollte so wie im folgenden Listing aussehen:

```
user$ glxinfo | grep render
...
OpenGL renderer string: Chromium
```

Wenn der OpenGL renderer string hingegen `llvmpipe` enthält, dann werden die 3D-Funktionen durch die CPU emuliert, was spürbar langsamer ist.

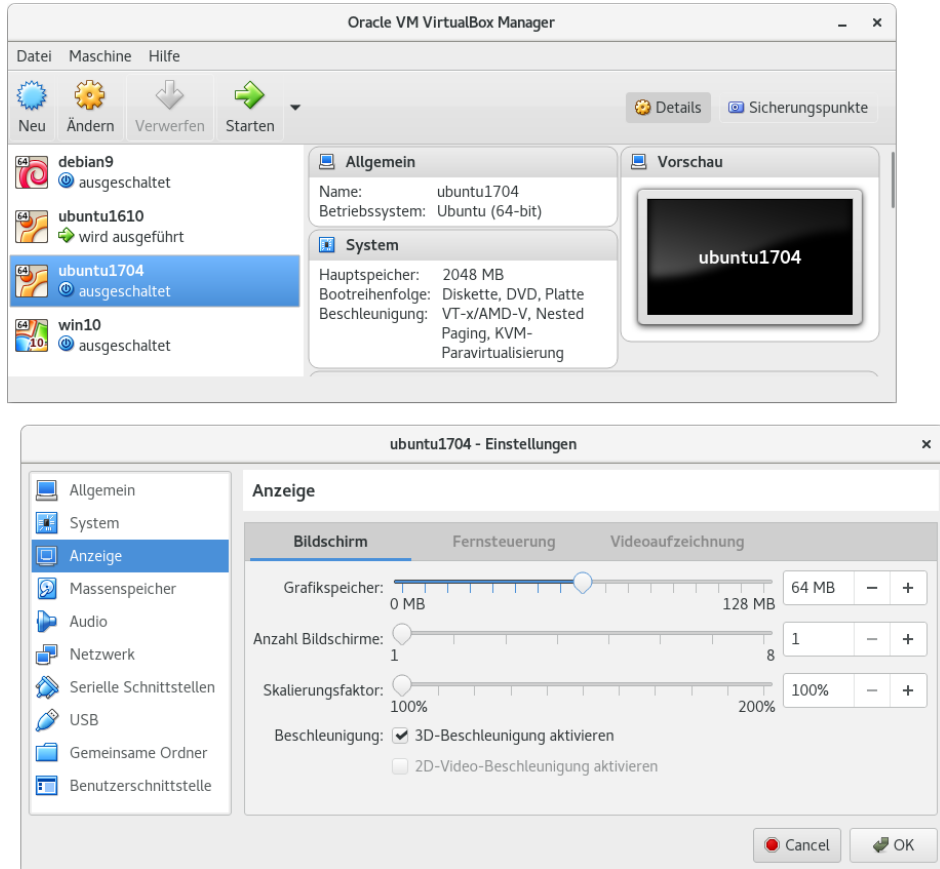


Abbildung 39.1 Überblick über alle virtuellen Maschinen (oben) und deren Einstellungen

### Eine virtuelle Maschine mit Windows einrichten

Sofern Sie über eine Installations-CD/DVD bzw. die entsprechende ISO-Datei sowie eine gültige Lizenz und den dazugehörigen Schlüssel verfügen, können Sie in VirtualBox auch Windows installieren (siehe [Abbildung 39.2](#)). Die Installation von Windows und der VirtualBox-Gasterweiterungen verlief bei meinen Tests stets problemlos.

Warten Sie mit der Online-Registrierung so lange ab, bis Sie mit der Leistung zufrieden sind. Wenn Sie später in den Einstellungen der virtuellen Maschine das RAM vergrößern oder andere virtuelle Hardware-Parameter ändern, müssen Sie unter Umständen die Registrierung wiederholen!

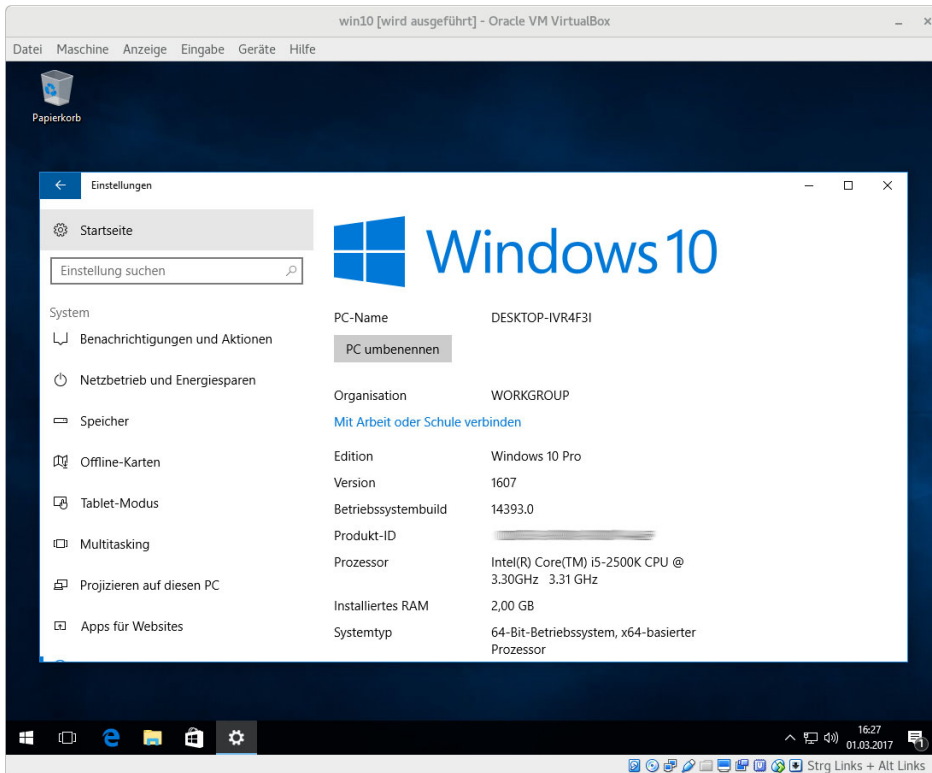


Abbildung 39.2 Windows 10 in einer virtuellen Maschine unter Linux ausführen

## 39.3 Arbeitstechniken und Konfigurationstipps

Dieser Abschnitt gibt Tipps zur Optimierung virtueller Maschinen sowie zum Datenaustausch zwischen dem Host-System bzw. dem »realen« lokalen Netzwerk und den virtuellen Maschinen.

### Netzwerkconfiguration

VirtualBox stellt seinen Gästen die Netzwerkinfrastruktur des Wirts in Form einer virtuellen Netzwerkkarte zur Verfügung. Dabei existieren unterschiedliche Verfahren, wie der Netzwerkverkehr von der virtuellen Netzwerkkarte in das reale Netzwerk geleitet wird. Die entsprechenden Parameter finden Sie im Einstellungsdialog im Dialogblatt NETZWERK (siehe [Abbildung 39.3](#)). Entscheidend ist die Einstellung des Listenfelds ANGESCHLOSSEN AN, wobei der Vorgabewert NAT lautet.

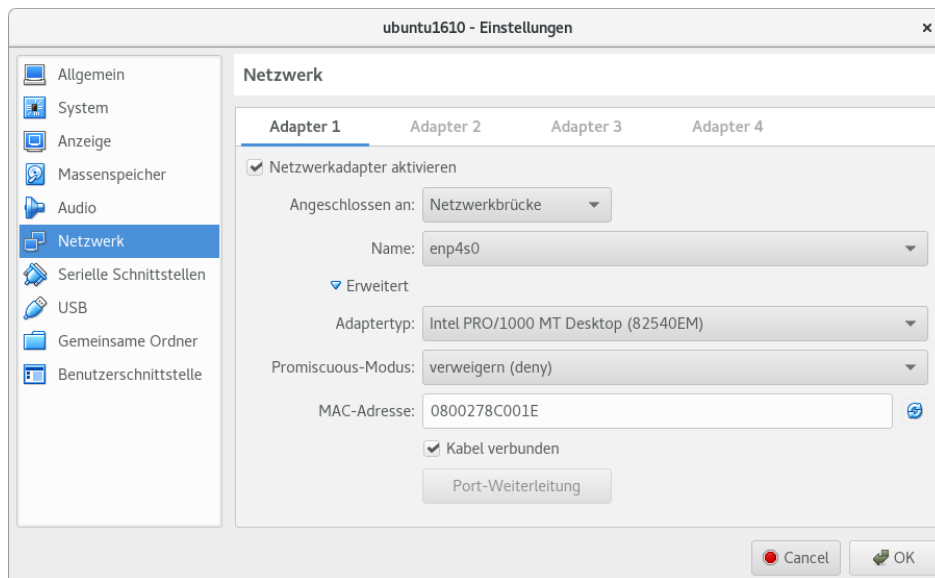


Abbildung 39.3 Netzwerkeinstellungen für die virtuelle Maschine

- ▶ **NAT:** Bei der NAT-Variante stellt VirtualBox seinen Gästen einen eigenen DHCP-Server zur Verfügung und realisiert Masquerading (NAT) (siehe auch [Kapitel 27](#), »Internet-Gateway«). Auf diese Weise können die Gäste den Internetzugang des Wirtssystems nutzen. Ein Zugang zum lokalen Netzwerk ist wegen der unterschiedlichen Adressbereiche für das lokale Netz und das virtuelle NAT-Netz des Virtualisierungssystems unmöglich. Ebenso wenig können Sie vom Host eine SSH-Verbindung zum Gast herstellen. Die virtuellen Maschinen sind vom Host wie durch eine einfache Firewall getrennt.

Bei der NAT-Variante verwendet VirtualBox auf dem Host die IP-Adresse 10.0.2.2. Die virtuellen Maschinen erhalten andere 10.0.2.\*-Adressen.

- ▶ **Netzwerkbrücke:** Bei dieser Variante erscheint der Gast als zusätzlicher Client im lokalen Netz. Diese Variante ist optimal, wenn es im lokalen Netzwerk einen DHCP-Server gibt bzw. wenn der Host-Rechner mit einem ADSL- oder WLAN-Router verbunden ist. Die virtuellen Gäste beziehen ihre Netzwerkkonfiguration dann über diesen Server/Router und können sowohl auf das lokale Netzwerk als auch auf das Internet zugreifen. Wenn Ihr Host-Rechner mehrere Netzwerkschnittstellen besitzt, müssen Sie angeben, welche Schnittstelle die Verbindung zum lokalen Netzwerk herstellt.

Im Büro ist diese Variante meine bevorzugte Konfiguration: Reale und virtuelle Maschinen sind damit im lokalen Netzwerk gleichwertige Partner, und der Datenaustausch via SSH, Samba etc. funktioniert unkompliziert. Beachten Sie aber, dass



die Netzwerkbrücke in manchen (Unternehmens-)WLANs nicht funktioniert. Die besten Erfahrungen habe ich mit dieser Konfigurationsvariante gemacht, wenn der Host-Rechner über ein Ethernet-Kabel (also nicht über WLAN) mit dem lokalen Netzwerk verbunden ist.

- ▶ **Host-only Adapter:** Bei dieser Variante kann der Gast über die Netzwerkfunktionen nur mit dem Wirt kommunizieren, nicht aber mit anderen Rechnern im lokalen Netzwerk oder mit dem Internet. Diese Variante ist dann zweckmäßig, wenn Sie ein von außen nicht zugängliches Testsystem aus mehreren virtuellen Maschinen aufbauen möchten.
- ▶ **Internes Netzwerk:** Hier bildet VirtualBox ein virtuelles Netzwerk, in dem ausschließlich virtuelle Maschinen kommunizieren können. Sie haben bei dieser Variante weder Zugriff auf das lokale Netzwerk noch auf das Internet.

Sie können virtuelle Maschinen mit bis zu vier Netzwerkadaptern ausstatten. Das gibt Ihnen die Möglichkeit, mehrere Konfigurationsvarianten parallel zu verwenden – z. B. einen NAT-Adapter, damit die virtuellen Maschinen Internetzugang erhalten, und einen Host-only-Adapter, damit Sie eine SSH-Verbindung zwischen den virtuellen Maschinen und dem Host-Rechner herstellen können.

Die Netzwerkkonfiguration kann im laufenden Betrieb geändert werden! Es ist also nicht erforderlich, die virtuelle Maschine bei jeder Änderung neu zu starten. Der schnellste Weg in den Konfigurationsdialog führt über das Icon **AKTIVITÄT DER NETZWERKADAPTER** in der Statusleiste des VirtualBox-Fensters.

### Datenaustausch über die Zwischenablage

In den Einstellungen der virtuellen Maschine können Sie im Dialogblatt **ALLGEMEIN • ERWEITERT** für die gemeinsame Zwischenablage und für die Funktion **Drag & Drop** den Modus **BIDIREKTIONAL** aktivieren. Beide Optionen setzen auf jeden Fall voraus, dass in der virtuellen Maschine die Gasterweiterungen installiert sind.

Diese Konfiguration gibt Ihnen die Möglichkeit, über die Zwischenablage Text zwischen dem Host und dem Gast zu kopieren. Außerdem können Sie nun per **Drag & Drop** Dateien zwischen einem Dateimanager im Host und einem Dateimanager im Gast hin- und herkopieren. Fallweise hat dies bei meinen Tests gut funktioniert, aber ganz ausgereift wirkt diese Funktion noch nicht.

### Datenaustausch mit einem Shared Folder

Ein zuverlässigerer Weg zum Datenaustausch zwischen Wirt und Gast sind sogenannte **Shared Folder**. Zur Konfiguration öffnen Sie mit **ÄNDERN** den Einstellungsdialog, wechseln in das Dialogblatt **GEMEINSAME ORDNER**, wählen dann ein lokales Verzeichnis auf dem Wirtssystem aus und geben dem Ordner einen Namen (z. B. *myshare*). Das

Host-  
Konfiguration

Verzeichnis gilt spezifisch für eine bestimmte virtuelle Maschine. Für Windows-Gäste aktivieren Sie auch gleich die Option `AUTOMATISCH EINBINDEN`.

**Linux-Gäste** Nach einem Neustart eines Linux-Gastsystems ist nun ein manuelles `mount`-Kommando erforderlich, um auf das gemeinsame Verzeichnis zugreifen zu können. Dabei müssen Sie `myshare` durch den Namen ersetzen, den Sie bei der Konfiguration verwendet haben.

```
root@gast# mkdir /media/vbox-share
root@gast# mount -t vboxsf myshare /media/vbox-share
```

Wenn Linux die Fehlermeldung `unknown filesystem vboxsf` liefert, sind die VirtualBox-Gasterweiterungen nicht richtig installiert. Abhilfe schafft bei den meisten Distributionen die Installation des Pakets `virtualbox-guest-utils`.

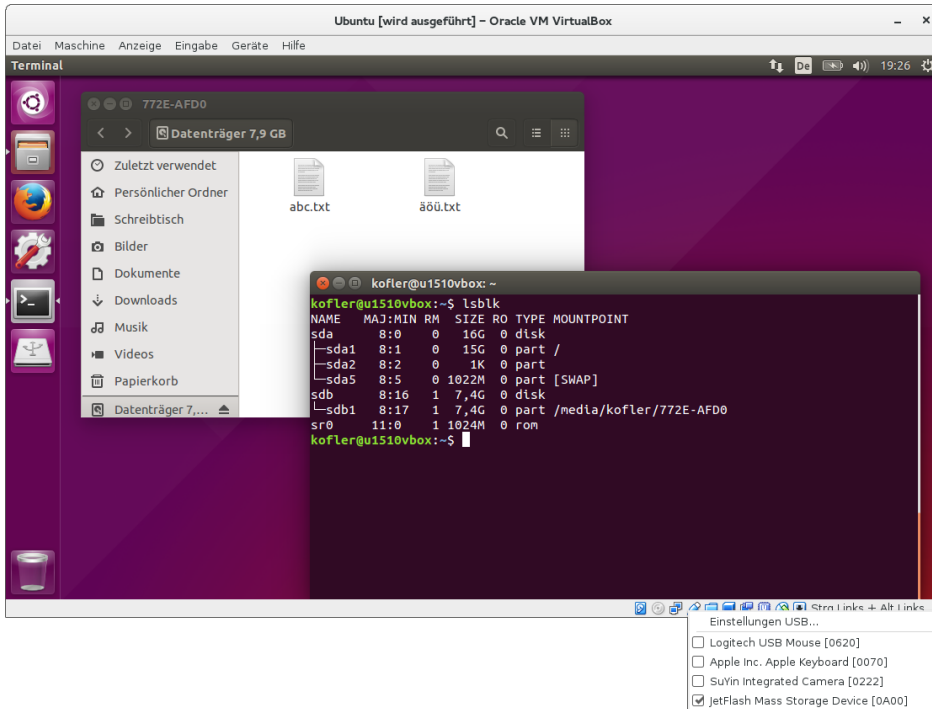
**Windows-Gäste** In Windows-Gästen finden Sie das gemeinsame Verzeichnis im Explorer als Netzwerkverzeichnis des virtuellen Rechners `vboxsrv`. Wenn Sie bei der Konfiguration die Option `AUTOMATISCH EINBINDEN` verwendet haben, dann wird dem Verzeichnis unter Windows auch gleich ein eigener Laufwerksbuchstabe zugeordnet.

### USB-Geräte in virtuellen Maschinen

Sofern Sie auf dem Host das VirtualBox Extension Pack installiert haben, können Sie USB-Geräte auch in virtuellen Maschinen nutzen. Das funktioniert nur, wenn das USB-Gerät im Wirtssystem *nicht* verwendet wird. USB-Datenträger werden im Wirtssystem normalerweise automatisch in das Dateisystem eingebunden; Sie müssen sie wieder aus ihm lösen, um sie im Gast verwenden zu können.

Eine weitere Voraussetzung besteht darin, dass der Benutzer, der VirtualBox ausführt, Mitglied der Gruppe `vboxusers` ist. Schließlich müssen Sie darauf achten, dass der `USB-CONTROLLER` bei den Einstellungen der virtuellen Maschine im Dialogblatt `USB` aktiviert ist. In diesem Dialogblatt können Sie auch einen Filter definieren, um ein USB-Gerät direkt einer virtuellen Maschine zuzuordnen. Das ist aber keine zwingende Voraussetzung. Sie können das USB-Gerät nach dem Einschalten auch dynamisch in der VirtualBox-Statusleiste beim `USB-Icon` der virtuellen Maschine zuordnen (siehe [Abbildung 39.4](#)).

Generell funktionierten die von mir getesteten USB-Geräte (ein Scanner und eine Digitalkamera) in den virtuellen Maschinen anstandslos, wenn auch langsamer als im Wirtssystem.



**Abbildung 39.4** Die virtuelle Ubuntu-Maschine kann auf einen USB-Stick zugreifen, der an das Notebook angesteckt wurde, auf dem VirtualBox ausgeführt wird.

### Export/Import virtueller Maschinen

Um eine virtuelle Maschine weiterzugeben, erzeugen Sie mit `DATEI • APPLIANCE EXPORTIEREN` eine sogenannte Virtual Appliance, also eine zur Weitergabe bestimmte virtuelle Maschine, die üblicherweise aus zwei Dateien besteht: `*.ovf` enthält eine Beschreibung der virtuellen Maschine, `*.vmdk` das Festplatten-Image in komprimierter Form. Diese virtuelle Maschine können Sie nun bei einer anderen VirtualBox-Installation mit `DATEI • APPLIANCE IMPORTIEREN` wieder einrichten.

#### Eine virtuelle Maschine auf einen anderen Host übertragen

Wenn es Ihnen nur darum geht, eine oder mehrere virtuelle Maschinen von einem Rechner auf einen anderen zu übertragen, können Sie sich die Umwandlung in eine Virtual Appliance sparen. In diesem Fall reicht es aus, das betreffende Verzeichnis `VirtualBox VMs/vm-name` zu kopieren. Anschließend führen Sie in VirtualBox das Kommando `MASCHINE • HINZUFÜGEN` aus und wählen die `*.vbox`-Datei aus.

## Geschwindigkeitsoptimierung

Mit zwei Optionen bei der Einstellung der virtuellen Hardware können Sie ein klein wenig mehr Geschwindigkeit aus Ihren virtuellen Maschinen herauskitzeln:

- ▶ **Host-Caching für die virtuelle Festplatte:** Im Dialogblatt MASSENSPEICHER der virtuellen Maschine können Sie für den SATA-Controller die Option HOST-I/O-CACHE VERWENDEN aktivieren. Sie erreichen damit, dass Schreibzugriffe zwischengespeichert werden, was die Geschwindigkeit I/O-lastiger Vorgänge stark vergrößern kann. Der Nachteil: Sollte der Host-Rechner abstürzen, riskieren Sie ein beschädigtes Dateisystem in der virtuellen Maschine.
- ▶ **Paravirtualisierte Netzwerktreiber:** Sofern es sich bei der virtuellen Maschine um eine Linux-Distribution handelt, können Sie im Dialogblatt NETZWERK bei den erweiterten Einstellungen die Option PARAVIRTUALISIERTES NETZWERK (VIRTIO-NET) aktivieren. VirtualBox spielt der virtuellen Maschine nun nicht mehr die Logik eines Netzwerkadapters vor, sondern spricht direkt mit dem virtio-net-Treiber des Linux-Kernels. Das ist deutlich effizienter.

## Virtuelle Festplatten vergrößern

**Linux-Gast** Die Benutzeroberfläche von VirtualBox gibt Ihnen leider keine Möglichkeit, eine virtuelle Festplatte nachträglich zu vergrößern. Wo die Benutzeroberfläche versagt, hilft oft auch ein Kommando weiter – so auch in diesem Fall. Bevor Sie loslegen, müssen Sie Ihre virtuelle Maschine herunterfahren. Ein vollständiges Backup ist sehr zu empfehlen!

Anschließend suchen Sie die \*.vdi-Datei der virtuellen Festplatte und wenden darauf das Kommando `vboxmanage` an. Mit der Option `--resize` geben Sie die gewünschte neue Größe in MiB an. Im Regelfall wird das Kommando blitzschnell ausgeführt.

```
root# vboxmanage modifyhd debian.vdi --resize 60000
```

Das ist aber erst die halbe Miete. Die virtuelle Maschine weiß nämlich noch nichts davon, dass ihre Festplatte größer geworden ist. Bei einer Gast-Installation ohne LVM und mit `ext4`- oder `xfs`-Dateisystemen binden Sie nun ein ISO-Image einer Linux-Live-CD in das virtuelle CD/DVD-Laufwerk ein und starten innerhalb der virtuellen Maschine ein Live-System. Dort führen Sie `parted /dev/sda` aus und können nun die Größe der letzten Partition erhöhen. Anschließend müssen Sie auch das darin enthaltene Dateisystem mit `resize2fs` oder `xfs_growfs` vergrößern.

Wenn Sie im Linux-Gast hingegen LVM oder `btrfs`-Dateisysteme verwenden, können Sie das Dateisystem im laufenden Betrieb vergrößern. Diese Eingriffe sind natürlich nicht ganz ungefährlich. Lesen Sie vorher die relevanten Abschnitte aus [Kapitel 22](#), »Administration des Dateisystems«!

Analog kann auch ein Windows-Dateisystem vergrößert werden. Auch in diesem Fall fahren Sie die virtuelle Maschine zuerst herunter und vergrößern die \*.vdi-Datei mit dem Kommando `vboxmanage`. Dann starten Sie Windows, öffnen darin ein Eingabeaufforderungsfenster mit Administratorrechten und führen die folgenden Kommandos aus:

Windows-Gast

```
> Diskpart
list disk
select disk 0
list partition
select partition 2
extend
```

`list disk` liefert eine Liste aller virtuellen Festplatten. Normalerweise muss die erste Platte mit dem Index 0 ausgewählt werden. Nun ermittelt `list partition` die Partitionen. Abermals muss mit `select` eine Partition zur weiteren Bearbeitung ausgewählt werden – im Regelfall die letzte. Mit `extend` wird diese nun auf die maximale Größe erweitert.

### Virtuelle Maschinen unsichtbar ausführen

Normalerweise wird jede laufende virtuelle Maschine in einem eigenen Fenster angezeigt. Beim Schließen des Fensters haben Sie die Wahl, den Status der virtuellen Maschine zu speichern (die virtuelle Maschine also gewissermaßen zu pausieren), sie per ACPI herunterzufahren oder sie gewaltsam zu stoppen (wie durch das Lösen eines Netzkabels).



**Abbildung 39.5** Das Menü des Start-Buttons enthält zwei versteckte Einträge zum Start der virtuellen Maschine ohne bzw. mit abkoppelbarer Oberfläche.

Mitunter wäre es aber praktisch, virtuelle Maschinen unsichtbar, also *ohne* eigenes Fenster auszuführen. Das gilt besonders für Server-Installationen, die ohnedies im Textmodus laufen. Für derartige virtuelle Maschinen können Sie beim Start-Button den Menüeintrag OHNE GUI STARTEN wählen.

Noch mehr Flexibilität gibt der Eintrag ABKOPPELBARER START (siehe [Abbildung 39.5](#)). Damit wird die virtuelle Maschine beim Start wie üblich in einem Fenster angezeigt. Mit MASCHINE • GUI ABKOPPELN können Sie das Fenster dann aber bei Bedarf schließen, ohne die virtuelle Maschine zu stoppen. Mit einem Doppelklick auf das Symbol der virtuellen Maschine im VirtualBox-Hauptfenster können Sie die Benutzeroberfläche der virtuellen Maschine sogar wiederbeleben.

## 39.4 Vagrant

Das Einrichten einer neuen virtuellen Maschine ist mit relativ viel Handarbeit verbunden. Solange es nur um eine Installation geht, ist das kein großes Problem. Wenn Sie aber regelmäßig virtuelle Maschinen einrichten müssen und dabei womöglich Wert darauf legen, dass die virtuellen Maschinen reproduzierbar exakt gleich konfiguriert sind, sollten Sie sich mit dem Programm *Vagrant* anfreunden. Vagrant ist ein Werkzeug, das beim Einrichten, Ausführen, Steuern und Stoppen von virtuellen Umgebungen hilft.

Vagrant wird zusammen mit einigen weiteren Programmen (Atlas, Packer, Vault, Nomad, Consul) von der Firma Hashicorp entwickelt. Alle Produkte verwenden Open-Source-Lizenzen und stehen kostenlos zur Verfügung. Zum Teil gibt es darüber hinaus Enterprise-Varianten mit Zusatzfunktionen für zahlende Kunden.

<https://www.hashicorp.com/#open-source-tools>

### Vagrant ist unabhängig von der Betriebssystem- und Virtualisierungsplattform!

Auch wenn ich Ihnen Vagrant hier im VirtualBox-Kapitel vorstelle, kommt das Programm auch mit anderen Virtualisierungssystemen zurecht, z.B. mit VMware und Hyper-V. Vagrant unterstützt auch Cloud-Systeme wie AWS sowie Docker. Für die libvirt-Werkzeuge, die ich in [Kapitel 40](#), »KVM«, vorgestellt habe, gibt es auf GitHub einen Provider, der allerdings Mitte 2017 noch nicht vollständig ausgereift war. Vagrant lässt sich problemlos auch unter Windows und macOS installieren.

**Nomenklatur** Die Dokumentation zu Vagrant ist leichter zu verstehen, wenn Sie sich zuerst mit einigen Begriffen vertraut machen:

- ▶ **Vagrant-Datei:** Vagrant richtet virtuelle Maschinen auf der Basis einer Vagrant-Datei und einer Box ein. Die Textdatei `Vagrantfile` gibt die Quelle der Box-Datei an und beschreibt, welche Operationen auf die Box angewendet werden müssen, um die virtuelle Maschine fertigzustellen. Dieser einmalig durchzuführende Vorgang wird »Provisioning« genannt. Die Anweisungen in der Vagrant-Datei werden in der Syntax der Programmiersprache Ruby angegeben. Die Vagrant-Datei kann beispielsweise Kommandos zum Einrichten der Netzwerkverbindung und des SSH-Servers enthalten. Sie können aber auch externe Scripts aufrufen, die zur Installation von Zusatz-Software oder für Konfigurationsarbeiten in der virtuellen Maschine auf Werkzeuge wie *Puppet* oder *Chef* zurückgreifen.
- ▶ **Boxes:** Eine Box ist eine komprimierte Datei, die eine virtuelle Maschine enthält. Box-Dateien sind wegen des inkludierten Festplatten-Images zumeist recht groß (mehrere Hundert MiB). Auf der Webseite <https://atlas.hashicorp.com/boxes> finden Sie einen Katalog kostenlos verfügbarer Vagrant-Boxes. Vagrant kommt aber auch mit Boxes zurecht, die lokal gespeichert sind oder auf anderen Webservern zugänglich sind.  
  
Beim ersten Start wird die virtuelle Maschine zuerst aus der Box geklont; anschließend führt Vagrant die in `Vagrantfile` aufgezählten Konfigurationsarbeiten durch, führt am Klon also noch Änderungen durch. Die Box selbst bleibt dabei unverändert und kann später neuerlich als Basis verwendet werden, wenn weitere Instanzen erzeugt werden sollen oder die virtuelle Maschine neu eingerichtet werden soll.
- ▶ **Vagrant-Kommando:** Die gesamte Administration von Vagrant erfolgt durch das Kommando `vagrant`. Damit starten und stoppen Sie virtuelle Maschinen, stellen SSH-Verbindungen zu ihnen her etc.
- ▶ **Provider:** Vagrant verwendet standardmäßig VirtualBox als Virtualisierungssystem. Sogenannte Provider stellen optionale Schnittstellen zu anderen Virtualisierungssystemen her. Einige Provider sind standardmäßig in Vagrant enthalten, andere können extra installiert werden.
- ▶ **Plugins:** Vagrant hat einen modularen Aufbau. Selbst etliche Grundfunktionen sind als Plugins realisiert. Zur Realisierung von Zusatzfunktionen können Sie Vagrant durch externe Plugins erweitern (`vagrant plugin install name`).

Bei vielen Distributionen installieren Sie Vagrant am einfachsten mit den Paketverwaltungskommandos. Allerdings erhalten Sie damit selten die aktuellste Version. Auf der Vagrant-Webseite <https://www.vagrantup.com> finden Sie aktuelle Pakete im Debian- und RPM-Format, deren Installation in der Regel auf Anhieb aus dem Webbrowser heraus gelingt. Nach der Installation können Sie mit `vagrant version` die Versionsnummer feststellen:

Installation

```
user$ vagrant version
Installed Version: 1.9.2
Latest Version: 1.9.2
```

**Base Boxes** Ich gehe in diesem Buch nur auf die Nutzung und Modifizierung vorgefertigter Boxes ein. Fortgeschrittene Vagrant-Anwender können aber auch vollkommen neue Boxes erzeugen. In der Regel ist es zweckmäßig, dabei eine sogenannte »Base Box« einzurichten, also eine virtuelle Maschine, die auf einer minimalen Installation der jeweiligen Distribution basiert und die speziell für Vagrant vorkonfiguriert ist. Eine typische Vagrant-Konfiguration besteht aus einem SSH-Server, einem vagrant-Benutzer mit sudo-Rechten ohne Passwort und eventuell der Installation von Gasterweiterungen für das gewünschte Virtualisierungssystem. Eine ausführliche Anleitung, wie Sie Vagrant-kompatible Base Boxes einrichten, finden Sie hier:

<https://www.vagrantup.com/docs/boxes/base.html>

## Hello World!

Um Vagrant auszuprobieren, greifen Sie am besten auf eine der vielen vorgefertigten Vagrant-Boxes zurück. Der Boxes-Katalog auf <https://atlas.hashicorp.com/boxes> enthält leider nicht viel mehr als den Namen der jeweiligen Box und eine Liste der unterstützten Provider. Unbegreiflicherweise fehlt eine Beschreibung, welche Zielsetzung die jeweilige Box hat. Nicht einmal die Größe der Box ist dokumentiert.

Für erste Experimente können Sie z. B. die Box `ubuntu/xenial64` verwenden. Sie enthält einen tagesaktuellen Build einer Minimalinstallation von Ubuntu 16.04 für den Server-Einsatz (also ohne grafische Benutzeroberfläche):

```
user$ mkdir u1604
user$ cd u1604
user$ vagrant init ubuntu/xenial64
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment!
user$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Adding box 'ubuntu/xenial64' (v20170303.1.0)
    default: Downloading: https://atlas.hashicorp.com/ubuntu/ \
        boxes/xenial64/versions/20170303.1.0/providers/virtualbox.box
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Mounting shared folders...
    default: /vagrant => /home/kofler/u1604
...
user$ vagrant ssh
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-64-generic x86_64)
```



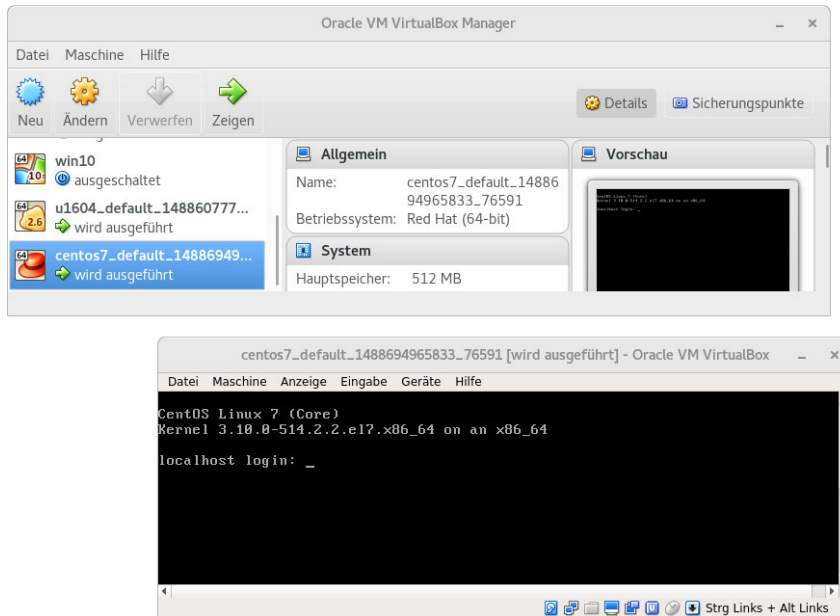
Kurz einige Erklärungen zu den obigen Kommandos, deren Ausgaben aus Platzgründen stark gekürzt abgedruckt sind. `vagrant init` lädt vom Hashicorp-Server die Datei `Vagrantfile` für die gewünschte virtuelle Maschine. Das geht schnell, da die Datei nur wenige Kilobyte groß ist. Sie wird im gerade aktuellen Verzeichnis gespeichert. `vagrant init` verwendet standardmäßig VirtualBox als Virtualisierungsplattform. Wenn Sie ein anderes System verwenden möchten, wählen Sie dieses mit `--provider name` aus.

`vagrant up` startet die virtuelle Maschine. Ab dem zweiten Mal wird auch dieses Kommando recht schnell ausgeführt, beim ersten Mal dauert es aber geraume Zeit: Zuerst muss nämlich die Box für die virtuelle Maschine heruntergeladen werden. Diese Box sowie diverse Zusatzdateien werden im Verzeichnis `./vagrant.d/boxes` gespeichert, also getrennt von dem Verzeichnis, in dem sich `Vagrantfile` befindet. Das hat den Vorteil, dass später bei Bedarf weitere virtuelle Maschinen auf Basis der bereits vorhandenen Box eingerichtet werden können. Für `ubuntu/xenial64` beträgt der Platzbedarf der Box ca. 275 MiB.

Sobald die Box heruntergeladen ist, wird die entsprechende virtuelle Maschine eingerichtet. Im obigen Beispiel verwendet Vagrant den Default-Provider für VirtualBox. Die Dateien der virtuellen Maschine landen daher in dem von VirtualBox vorgesehenen Verzeichnis. Wenn Sie die VirtualBox-Defaulteinstellungen nicht verändert haben, ist das `VirtualBox VMs` in ihrem Heimatverzeichnis. Damit gibt es nun Dateien an drei verschiedenen Orten:

- ▶ in Ihrem eigenen Vagrant-Verzeichnis: Es enthält neben `Vagrantfile` einige weitere Konfigurationsdateien und beansprucht nur wenig Speicherplatz. Alle `vagrant`-Kommandos müssen in diesem Verzeichnis oder in einem seiner Unterverzeichnisse ausgeführt werden.
- ▶ in `./vagrant.d/boxes`: Das Verzeichnis enthält je eine Box für alle irgendwann mit Vagrant eingerichteten Maschinen. Der Platzbedarf beträgt typischerweise einige Hundert MiB pro Box.
- ▶ in `VirtualBox VMs`: Dieses Verzeichnis enthält die virtuellen Maschinen inklusive der Disk-Images für jede mit Vagrant eingerichtete Maschine. Der Platzbedarf ist hoch und beträgt oft mehrere GiB pro virtueller Maschine.

Die von Vagrant eingerichtete VirtualBox-Maschine wird im VirtualBox-Hauptfenster zwischen selbst erzeugten virtuellen Maschinen aufgelistet. Ihr Name endet immer mit einer zufällig generierten Zahl (siehe [Abbildung 39.6](#)). `vagrant up` startet die virtuelle Maschine unsichtbar, also ohne ein VirtualBox-Fenster zu öffnen. Zwar ist es möglich, per Doppelklick auf die Liste der virtuellen Maschinen ein entsprechendes Fenster zu öffnen; im Normalfall ist es aber üblich, Vagrant-Maschinen per SSH zu administrieren.



**Abbildung 39.6** Die von Vagrant eingerichteten virtuellen Maschinen sind in der Liste der VirtualBox-Maschinen an der Namensweiterung »default\_nnn« zu erkennen.

### Bento-Boxes

Das Boxes-Angebot auf <https://atlas.hashicorp.com/boxes> ist leider ziemlich unübersichtlich. Wenn Sie auf der Suche nach kleinen, vernünftig vorkonfigurierten Boxes für die wichtigsten Linux-Distributionen sind, lohnt sich ein Blick auf die Webseite <http://chef.github.io/bento/>! Zur Verwendung einer derartigen Box führen Sie einfach `vagrant init bento/<name>` aus.

### Netzwerkconfiguration

Vagrant-Maschinen verwenden in VirtualBox einen NAT-Netzwerkadapter. Das ist aus Sicherheitsgründen zweckmäßig, weil in Vagrant-Maschinen üblicherweise der Account `vagrant` mit einem gleichnamigen Passwort eingerichtet ist. Wäre die virtuelle Maschine im lokalen Netz oder gar im Internet öffentlich erreichbar, würde sie unweigerlich das Ziel von Hacker-Angriffen.

Damit zwischen dem Host-Rechner und der virtuellen Maschine eine SSH-Verbindung möglich ist, richtet Vagrant standardmäßig eine Port-Umleitung zwischen dem Port 22 der virtuellen Maschine und dem Port 2222 des Hosts her. Ist dieser Port schon von einer anderen Box belegt, sucht `vagrant` `up` selbstständig einen anderen freien Port mit der Nummer `22nn`.

Um eine SSH-Verbindung zur virtuellen Maschine herzustellen, führen Sie einfach das Kommando `vagrant ssh` aus. Vagrant startet den SSH-Client dann mit den richtigen Optionen. Sie brauchen kein Passwort anzugeben. In der virtuellen Maschine werden Sie in der Regel als Benutzer `vagrant` bzw. beim hier vorgestellten Beispiel als Benutzer `ubuntu` angemeldet. Dank einer in `/etc/sudoers.d` vorgesehenen Konfigurationsdatei erlangen Sie mit `sudo -s` ohne Passwort root-Rechte:

```
user@hostsystem$ vagrant ssh
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-64-generic x86_64)
ubuntu@ubuntu-xenial64:~$ sudo -s
root@ubuntu-xenial64:~# cat /etc/sudoers.d/90-cloud-init-users
# User rules for ubuntu
ubuntu ALL=(ALL) NOPASSWD:ALL
```

Bei vielen Boxes richtet Vagrant darüber hinaus ein gemeinsames Verzeichnis zwischen Host und virtueller Maschine ein. Auf dem Host wird dazu das Verzeichnis verwendet, in dem sich `Vagrantfile` befindet. Auf dem Client ist der Shared Folder in Linux-Gästen üblicherweise unter `/vagrant` zugänglich.

Beim vorgestellten Ubuntu-Beispiel ist das zum Datenaustausch vorgesehene Verzeichnis als VirtualBox Shared Folder realisiert. In der Box sind dazu standardmäßig die VirtualBox-Gasterweiterungen installiert. Andere von Vagrant unterstützte Verfahren zur Realisierung des gemeinsamen Verzeichnisses basieren auf Rsync (Synchronisierung nur beim Start), NFS oder SSHFS (`vagrant-sshfs`-Plugin).

### Default-Login

Es ist üblich, dass Vagrant-Maschinen einen Default-Account mit dem Login-Namen `vagrant` und einem gleichnamigen Passwort haben. Dieser Benutzer wird auch für SSH-Verbindungen verwendet, wobei die Authentifizierung über eine Schlüsseldatei erfolgt.

Die in diesem Beispiel vorgestellte Ubuntu-Box widerspricht leider den Vagrant-Empfehlungen: In diesem Fall lautet der Default-Login `ubuntu`. Als Passwort wird ein zufälliger hexadezimaler Code verwendet. Um ein eigenes Passwort einzustellen, stellen Sie mit `vagrant ssh` eine Verbindung zur virtuellen Maschine her und führen dann `passwd` aus.

## Administration

Die gesamte Administration von Vagrant erfolgt mit dem gleichnamigen Kommando (siehe [Tabelle 39.2](#)). Soweit sich die gewünschte Operation auf eine Box bezieht, sucht `vagrant` zuerst im aktuellen Verzeichnis nach `Vagrantfile`, danach in allen übergeordneten Verzeichnissen.

vagrant wird in der Regel ohne root-Rechte ausgeführt. `vagrant -h` liefert eine Liste aller Kommandos. `vagrant kommando -h` zeigt weiterführende Informationen zum betreffenden Kommando an.

| Kommando                                 | Bedeutung                                      |
|--|--|
| <code>vagrant box list</code>            | heruntergeladene Vagrant-Boxes auflisten       |
| <code>vagrant box update</code>          | Vagrant-Box aktualisieren                      |
| <code>vagrant destroy</code>             | Vagrant-Maschine löschen                       |
| <code>vagrant halt</code>                | Vagrant-Maschine herunterfahren                |
| <code>vagrant init name</code>           | vorgefertige Vagrant-Datei herunterladen       |
| <code>vagrant login</code>               | Login zu eigenem Atlas-Account durchführen     |
| <code>vagrant plugin install name</code> | Plugin installieren                            |
| <code>vagrant provision</code>           | Provisioning wiederholen                       |
| <code>vagrant resume</code>              | pausierte Vagrant-Maschine wieder aktivieren   |
| <code>vagrant share</code>               | Vagrant-Maschine öffentlich zugänglich machen  |
| <code>vagrant ssh</code>                 | SSH-Verbindung zur Vagrant-Maschine herstellen |
| <code>vagrant status</code>              | Status der Vagrant-Maschine anzeigen           |
| <code>vagrant suspend</code>             | Vagrant-Maschine pausieren                     |
| <code>vagrant up</code>                  | Vagrant-Maschine starten                       |

**Tabelle 39.2** Wichtige vagrant-Kommandos

## VagrantFile

Die Datei `VagrantFile` beschreibt die Konfiguration der virtuellen Maschine, die Vagrant einrichten soll. Im einfachsten Fall sind drei Zeilen ausreichend, die einfach den Ort der zugrunde liegenden Vagrant-Box auf dem Hashicorp-Server angeben. "2" bedeutet, dass die Vagrant-Datei die Syntax von Version 2 verwendet. `config.vm.box` gibt den Namen der Box an. Vagrant sucht üblicherweise im Hashicorp-Katalog nach der Box und lädt sie von dort herunter. Wenn sich die Box auf einem anderen Server oder in einem lokalen Verzeichnis befindet, geben Sie diesen Ort zusätzlich mit `config.vm.box_url` an. Für lokale Dateien verwenden Sie dabei die Syntax `"file:///pfad/name.box"`.

```
# VagrantFile für ubuntu/xenial64
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/xenial64"
end
```

Nicht explizit in `VagrantFile` aufgeführt sind die Operationen zum Einrichten der Port-Umleitung für den SSH-Server sowie für die Synchronisation des gemeinsamen Verzeichnisses. Darum kümmert sich Vagrant automatisch, sofern dies nicht durch anderslautende Optionen verhindert wird.

Im Folgenden stelle ich Ihnen exemplarisch einige Optionen für `VagrantFile` vor. Eine vollständige Referenz finden Sie in der Vagrant-Dokumentation:

<https://www.vagrantup.com/docs/vagrantfile>

Einfache Änderungen an `VagrantFile` werden wirksam, wenn Sie die virtuelle Maschine einfach nur neu starten:

**VagrantFile-  
Änderungen  
aktivieren**

```
root# vagrant reload
```

Alle mit `config.vm.provision` definierten Konfigurationsarbeiten erfordern aber die Option `--provision`. Damit erzwingen Sie eine neuerliche Konfiguration der virtuellen Maschine. (Normalerweise wird das sogenannte Provisioning ja nur beim ersten Start durchgeführt.)

```
user$ vagrant reload --provision
```

Alternativ können Sie das Provisioning auch im laufenden Betrieb durchführen bzw. wiederholen. Dabei können Sie mit der Option `--provision-with` einschränken, welchen Typ von Provisioning-Maßnahmen (z. B. `shell` oder `file`) bzw. welche benannte Provisioning-Anweisung Sie ausführen möchten:

```
user$ vagrant provision --provision-with shell
```

Bei komplexen Änderungen, für die Vagrant in der richtigen Reihenfolge mehrere Scripts ausführen muss, kann es sogar erforderlich sein, dass Sie die virtuelle Maschine mit `vagrant destroy` löschen und dann vollständig neu einrichten müssen:

```
user$ vagrant destroy
```

```
user$ vagrant up
```

`config.vm.hostname` legt den Hostnamen der virtuellen Maschine fest.

**Hostname**

```
config.vm.hostname = "vagrant-u1604"
```

Mit `config.vm.network` können Sie diverse Parameter der Netzwerkkonfiguration verändern. Die folgende Zeile bewirkt eine Port-Umleitung vom Port 80 der virtuellen Maschine auf den Port 8080 des Hosts:

**Portumleitung**

```
config.vm.network "forwarded_port", guest: 80, host: 8080
```

Standardmäßig teilt Vagrant das Projektverzeichnis, also das Verzeichnis des Hosts, in dem sich `vagrantfile` befindet, im Gast als `/vagrant`. Bei Bedarf können Sie das verhindern:

**Gemeinsame  
Verzeichnisse**

```
config.vm.synced_folder ".", "/vagrant", disabled: true
```

Umgekehrt können Sie mit `config.vm.synced_folder` weitere gemeinsame Verzeichnisse einrichten. Dabei bezieht sich der erste Parameter auf den Host-Rechner (relativ zum Projektverzeichnis), der zweite Parameter auf den Gast.

```
config.vm.synced_folder "html/", "/var/www/html"
```

#### VirtualBox-spezifische Konfiguration

Wenn Sie Parameter der VirtualBox-Konfiguration verändern möchten, müssen Sie dazu einen eigenen `config.vm.provider`-Block definieren. Das folgende Listing gibt dafür drei Beispiele:

```
Vagrant.configure("2") do |config|
  ...
  config.vm.provider "virtualbox" do |vb|
    # RAM in MiB für die virtuelle Maschine (Default: laut Box)
    vb.memory = 1024
    # CPU-Cores (Default: laut Box)
    vb.cpus = 2
    # beim Start VirtualBox-Fenster anzeigen (Default: false)
    vb.gui = true
  end
end
```

**Scripts** Mit `vagrant.vm.provision "shell" ...` erreichen Sie, dass Vagrant im Zuge des Provisionings das angegebene Script mit `root`-Rechten in der virtuellen Maschine ausführt. Kleinere Scripts können Sie direkt als Zeichenkette mit dem Schlüsselwort `inline` angeben:

```
config.vm.provision "shell", inline: "echo $(date)"
```

Auch mehrzeilige Scripts können Sie direkt in die Vagrant-Datei einbetten:

```
$myscript = <<END
apt-get update
apt-get install -y joe
END
```

```
Vagrant.configure("2") do |config|
  ...
  config.vm.provision "shell", inline: $myscript
end
```

Längere Scripts sind besser in eigenen Dateien untergebracht. Diese können Sie z. B. direkt im Vagrant-Projektverzeichnis speichern. Mit `path` geben Sie einfach den relativen Ort der Datei an. Anders als bei lokal auszuführenden Scripts ist es übrigens nicht erforderlich, die Datei mit `chmod a+x` ausführbar zu machen.

```
config.vm.provision "shell", path: "my-long-script.sh"
```

Wenn Sie in die Vagrant-Datei mehrere Scripts einbauen, ist es zweckmäßig, diese zu benennen. Dabei gilt die folgende Syntax:

```
config.vm.provision "script1", type: "shell", inline: "echo $(date)"
config.vm.provision "script2", type: "shell", inline: $myscript
```

Das hat zwei Vorteile: Zum einen können Sie damit die Ausgaben des Vagrant-Kommandos klarer einzelnen Scripts zuordnen, zum anderen ist es so möglich, nur ein bestimmtes Script auszuführen:

```
user$ vagrant provision --provision-with script2
```

Scripts für virtuelle Windows-Maschinen müssen übrigens in der Syntax von PowerShell formuliert werden – aber die PowerShell ist in diesem Buch ohnedies kein Thema.

### Beispiel: CentOS-Webserver

Der Ausgangspunkt für das folgende Beispiel ist die Box `centos/7` aus dem Hashicorp-Katalog. Ähnlich wie mit `ubuntu/xenial64` erhalten Sie damit eine minimale Server-Installation: Die Box ist ca. 400 MiB groß und kompatibel mit vier Providern: VirtualBox, VMWare Workstation, VMWare Fusion und libvirt. Unter VirtualBox beansprucht `centos/7` anfänglich ca. 1 GiB Platz im Verzeichnis der VirtualBox-Maschinen. Die Eckdaten und einige Konfigurationsdetails sind hier dokumentiert:

<https://atlas.hashicorp.com/centos/boxes/7>

Um die Maschine im originalen Zustand einzurichten, führen Sie die folgenden Kommandos aus:

```
user$ mkdir centos7
user$ cd centos7
user$ vagrant init centos/7
user$ vagrant up
```

Im Gegensatz zu `ubuntu/xenial64` sind in der CentOS-Maschine die VirtualBox-Gasterweiterungen nicht standardmäßig installiert. Die Synchronisation des gemeinsamen Vagrant-Verzeichnisses erfolgt daher mit `rsync` beim Start der virtuellen Maschine. Beachten Sie, dass die Synchronisation einseitig ist: Es werden Dateien vom Host zum Gast übertragen, aber keine Änderungen vom Gast zurück zum Host synchronisiert.

Minimal ist auch die Vagrant-Datei von `centos/7`. Ohne Kommentare verbleiben nur drei Zeilen:

```
Vagrant.configure("2") do |config|
  config.vm.box = "centos/7"
end
```

**Webserver  
installieren und  
starten**

Das Ziel dieses Beispiels ist es, in der virtuellen Maschine automatisiert einen Webserver einzurichten. Dazu erstellen Sie im Vagrant-Projektverzeichnis die folgende Script-Datei:

```
#!/bin/bash
# Datei /home/kofler/centos7/install-webserver.sh
yum install -y httpd
systemctl enable httpd
systemctl start httpd
```

Anschließend ergänzen Sie die Vagrant-Datei um die folgenden beiden Zeilen:

```
config.vm.provision "shell", path: "install-webserver.sh"
config.vm.network "forwarded_port", guest: 80, host: 8080
```

Die erste Zeile gibt an, dass das Script `install-webserver.sh` im Rahmen des Provisioning in der virtuellen Maschine mit `root`-Rechten ausgeführt werden soll. Die zweite Zeile leitet den Port 80 der virtuellen Maschine auf den lokalen Port 8080 um, sodass der neu installierte Webserver direkt auf dem Hostrechner ausprobiert werden kann.

Um die Installation durchzuführen, starten Sie die virtuelle Maschine mit der Option `--provision neu`. Dabei werden sämtliche Ausgaben des `yum`-Kommandos angezeigt. Im folgenden Listing habe ich die Ausgaben aus Platzgründen stark gekürzt:

```
user$ vagrant reload --provision
==> default: Running provisioner: shell...
    default: Running: script
...
==> default: Install 1 Package (+4 Dependent packages)
...
==> default: Created symlink
    from /etc/systemd/system/multi-user.target.wants/httpd.service
    to /usr/lib/systemd/system/httpd.service.
```

Um den Webserver auszuprobieren, öffnen Sie auf dem Hostrechner in einem Webbrowser die Seite `http://localhost:8080`. Sie sollten darin die Testseite des Webservers sehen.

**Eigene  
HTML-Dateien**

Wenn Sie anstelle der Testseite eigene Webseiten anzeigen möchten, können Sie im Vagrant-Projektverzeichnis ein Unterverzeichnis mit den gewünschten HTML-Dateien einrichten:

```
user$ mkdir html
user$ cat > html/index.html << END
> <html>
> <body>
> <h1>Hello World!</h1>
> </body>
> </html>
> END
```



Damit alle Dateien aus dem lokalen `html`-Verzeichnis mit dem Verzeichnis `/var/www/html` in der virtuellen Maschine synchronisiert werden, ist die folgende Ergänzung in `VagrantFile` erforderlich:

```
config.vm.synced_folder "html/", "/var/www/html", type: "rsync"
```

Beachten Sie, dass dieses Beispiel nur statische Webseiten berücksichtigt. Wenn Sie eine dynamische Webseite einrichten möchten, müssen Sie in der virtuellen Maschine auch einen Datenbank-Server sowie geeignete Apache-Erweiterungen installieren, also z. B. MySQL und PHP. Dazu ist ein komplexeres Provisioning-Script erforderlich.

## Sharing

Der Begriff »Sharing« meint in der Vagrant-Nomenklatur nicht den Datenaustausch zwischen Host und Gast über ein gemeinsames Verzeichnis (das ist ein »Shared Folder«), sondern bedeutet, dass Sie eine Vagrant-Maschine öffentlich im Internet zugänglich machen. Das setzt voraus, dass Sie zuerst einen kostenlosen Account auf <https://atlas.hashicorp.com> einrichten und dann mit `vagrant login` einmal einen entsprechenden Login durchführen:

```
root# vagrant login
```

```
In a moment we will ask for your username and password to HashiCorp's
Atlas. After authenticating, we will store an access token locally on
disk. Your login details will be transmitted over a secure connection, and
are never stored on disk locally.
```

```
Atlas Username: accountname
Password (will be hidden): *****
```

Sind diese Vorbereitungsarbeiten einmal erledigt, aktivieren Sie das Sharing mit dem Kommando `vagrant share`. Mit jeder Ausführung des Kommandos wird dem Share ein zufälliger Name zugewiesen – im folgenden Beispiel `arctic-gazelle-1751.vagrantshare.com`:

```
user$ vagrant share
```

```
=> default: Detecting network information for machine...
default: Local machine address: 127.0.0.1
default:
default: Note: With the local address (127.0.0.1), Vagrant Share can only
default: share any ports you have forwarded. Assign an IP or address to your
default: machine to expose all TCP ports. Consult the documentation
default: for your provider ('virtualbox') for more information.
default:
default: Local HTTP port: 8080
default: Local HTTPS port: disabled
default: Port: 2222
```

```

default: Port: 8080
==> default: Checking authentication and authorization...
==> default: Creating Vagrant Share session...
default: Share will be at: arctic-gazelle-1751
==> default: Your Vagrant Share is running! Name: arctic-gazelle-1751
==> default: URL: http://arctic-gazelle-1751.vagrantshare.com
==> default:
==> default: You're sharing your Vagrant machine in "restricted" mode. This
==> default: means that only the ports listed above will be accessible by
==> default: other users (either via the web URL or using `vagrant connect`).

```

Solange `vagrant share` läuft, kann nun jede Person mit Internetzugang über die oben abgedruckte Adresse auf den in Webserver zugreifen, der in einer virtuellen Maschine läuft. Diese Funktion bietet eine großartige Möglichkeit, eine in Entwicklung befindliche Web-Applikation unkompliziert auszuprobieren. Dazu müssen Sie nur `vagrant share` ausführen und dem Tester die zufällig erzeugte URL senden.

Im Gegensatz zu anderen `vagrant`-Kommandos läuft `vagrant share` unlimitiert. Sie müssen explizit `Strg+C` drücken, wenn Sie das Sharing beenden möchten.

**SSH-Sharing** Standardmäßig kümmert sich `vagrant share` nur um HTTP-Sharing. Jedes Mal, wenn Sie an das Kommando zusätzlich die Option `--ssh` übergeben, erzeugt Vagrant einen neuen SSH-Schlüssel. Der Zugriff auf diesen Schlüssel wird durch ein Passwort abgesichert, das Sie zweimal angeben müssen:

```

user$ vagrant share --ssh
...
==> default: Generating new SSH key...
default: Please enter a password to encrypt the key: *******
default: Repeat the password to confirm: *******
...
==> default: You're sharing with SSH access. This means that another user
==> default: simply has to run `vagrant connect --ssh arctic-gazelle-1751`
==> default: to SSH to your Vagrant machine.

```

Andere Vagrant-Benutzer können nun eine SSH-Verbindung zu Ihrem Server herstellen, indem sie das Kommando `vagrant connect --ssh ...` ausführen. Beim Login müssen sie das Passwort des SSH-Schlüssels angeben. Aus Sicherheitsgründen ist die Ausführung von `vagrant connect --ssh ...` nur Vagrant-Benutzern erlaubt, die ebenfalls einen Atlas-Account haben und sich mit `vagrant login` eingeloggt haben.

# Kapitel 40

## KVM

KVM (*Kernel-based Virtual Machine*) ist eine Linux-spezifische Virtualisierungstechnik für das Server- und Enterprise-Segment. Red Hat und SUSE setzen in ihren Enterprise-Distributionen voll auf KVM, und auch alle anderen gängigen Linux-Distributionen enthalten KVM-Pakete.

Dieses Kapitel führt zuerst in die Grundlagen von KVM ein und konzentriert sich dann auf die Server-Virtualisierung mit KVM: Damit können auf einem Rechner mehrere virtuelle Linux-Server laufen. In der Praxis wird das häufig gemacht, um die Server-Funktionen so gut wie möglich voneinander zu trennen und so die Sicherheit zu maximieren. Aber auch praktische Gründe sprechen oft für die Server-Virtualisierung: Während der eine Anwender für seine Website spezielle Apache-Module braucht, will ein anderer die neueste MySQL-Version einsetzen. Wenn viele derartige Sonderwünsche auf *einem* System erfüllt werden, führt das rasch zu unerwünschten Nebenwirkungen und Instabilitäten.

KVM ist prinzipiell auch zur Desktop-Virtualisierung geeignet, dieser Aspekt steht hier aber im Hintergrund. Für den Desktop-Einsatz empfehle ich Ihnen VirtualBox, das einfacher zu bedienen ist. Sollten Sie dennoch KVM auf Desktop-Systemen einsetzen wollen, können Sie einen Blick auf das Gnome-Programm »Boxes« werfen: Wie bei Gnome typisch, ist die Bedienung sehr einfach, allerdings bietet das Programm selbst für einfache Anwendungen zu wenige Konfigurationsmöglichkeiten.

Naturgemäß können Sie KVM auch auf einem Root-Server einsetzen. Das gibt Ihnen die Möglichkeit, auf einem realen Server mehrere virtuelle Maschinen einzurichten, die nach außen wie eigene Server aussehen. Dabei müssen Sie zwei Dinge beachten:

**KVM auf dem  
Root-Server**

- ▶ Sie brauchen einen *echten* Root-Server. Heutzutage bieten viele Hosting- und Cloud-Provider virtuelle Umgebungen an. Das ist preisgünstig und flexibel, weil Sie später unkompliziert RAM oder CPU-Cores hinzufügen können – aber es macht eine Virtualisierung unmöglich. (Bis auf wenige Ausnahmen ist es unmöglich bzw. zu ineffizient, dass eine virtualisierte Maschine selbst wieder als Virtualisierungs-Host agiert – also gleichsam verschachtelt wie russische Matroschkapuppen.)

- Für die meisten Anwendungen benötigt sowohl der KVM-Host als auch jeder Gast eine eigene IP-Adresse. Für die meisten Einsatzzwecke benötigen Sie IPv4-Adressen, die rar sind und von den Hosting-Anbietern daher entsprechend teuer weitergegeben werden. Relativ kompliziert ist in solchen Fällen auch das Routing zwischen dem KVM-Host und seinen Gästen.

**Links** Dieses Kapitel kann nur eine Einführung in KVM geben. Weitere Informationen finden Sie hier:

<https://www.linux-kvm.org>

<https://libvirt.org>

<https://help.ubuntu.com/community/KVM>

<https://access.redhat.com/documentation/en/red-hat-enterprise-linux>

## 40.1 Grundlagen

### KVM versus QEMU

Das Programm QEMU emuliert verschiedene CPUs und elementare Hardware-Komponenten eines Rechners, also seine Netzwerkkarte, ein DVD-Laufwerk für die Installation etc. QEMU ist auch in der Lage, zur Wirts-CPU inkompatible Prozessoren zu emulieren (ARM, Sparc, PowerPC, MIPS etc.).

KVM ist ein Kernelmodul, das seine Wirkung erst in Kombination mit QEMU entfaltet. KVM setzt eine CPU mit Funktionen zur Hardware-Virtualisierung voraus und macht aus dem Emulator QEMU ein Hardware-Virtualisierungssystem. Die Eleganz von KVM besteht darin, dass es typische Hypervisor-Aufgaben nicht selbst ausführt, sondern dazu auf Speicher- und Prozessverwaltungsfunktionen des Linux-Kernels zurückgreift. Die Nutzung der KVM-Funktionen erfolgt über die Device-Datei `/dev/kvm`.

### Hardware-Voraussetzungen

KVM funktioniert nur, wenn der Prozessor des Host-Systems Virtualisierungsfunktionen unterstützt (Intel-VT bzw. AMD-V). Das ist bei den meisten aktuellen Prozessoren der Fall. Zu den Ausnahmen zählen sehr preisgünstige CPUs für Billig-PCs bzw. -Notebooks. Um festzustellen, ob Ihre CPU bei der Hardware-Virtualisierung hilft (Intel-VT oder AMD-V), führen Sie das folgende `egrep`-Kommando aus. Wenn das Ergebnis leer ist, unterstützt Ihre CPU keine Virtualisierung oder die Funktion wurde im BIOS/EFI deaktiviert.

```
user$ egrep '^flags.*(vmx|svm)' /proc/cpuinfo
flags :...  vmx ...
```

Bei Ubuntu-Systemen können Sie – noch einfacher – das Kommando `kvm-ok` aus dem Paket `cpu-checker` ausführen:

```
user$ kvm-ok
INFO: Your CPU supports KVM extensions
INFO: /dev/kvm exists
KVM acceleration can be used
```

Im weiteren Verlauf dieses Kapitels setze ich voraus, dass Ihre CPU KVM-kompatibel ist. Sollte das nicht der Fall sein, funktioniert KVM scheinbar auch. Tatsächlich werden die virtuellen Maschinen aber nur durch QEMU und somit ohne KVM-Unterstützung ausgeführt und laufen dann wesentlich langsamer.

### Virtualisierungsfunktionen im BIOS/EFI aktivieren

Erstaunlicherweise sind die vorhandenen Virtualisierungsfunktionen der CPU oft durch das BIOS oder EFI deaktiviert. Abhilfe: Öffnen Sie beim Rechnerstart die BIOS/EFI-Dialoge, und suchen Sie nach der betreffenden Einstellung.

Um virtuelle KVM-Maschinen auszuführen und mit den `libvirt`-Werkzeugen steuern zu können, müssen Sie die Pakete `qemu-kvm` und `virt-manager` installieren:

Erforderliche  
Pakete

```
root# apt/dnf/yum/zypper install qemu-kvm virt-manager
```

KVM stellt seine Funktionen in drei Kernelmodulen zur Verfügung: Die Grundfunktionen befinden sich im Modul `kvm`, die Intel-VT-spezifischen Funktionen in `kvm-intel`, die AMD-V-spezifischen Funktionen in `kvm-amd`. Damit Sie KVM nutzen können, muss das zu Ihrer Hardware passende KVM-Modul geladen werden. Das Modul `kvm` wird dabei gleich mitgeladen. Bei den meisten Distributionen kümmert sich das Init-System darum. Sollte das nicht funktionieren, greifen Sie manuell ein:

Kernelmodule

```
root# modprobe kvm-intel (für Intel-VT-Prozessoren)
root# modprobe kvm-amd (für AMD-V-Prozessoren)
```

Um eine virtuelle Maschine mit QEMU oder KVM auszuführen, können Sie direkt das KVM-Kommando ausführen. Es wird je nach Distribution unterschiedlich angesprochen (siehe [Tabelle 40.1](#)).

KVM-Kommando

| Distribution | KVM-Kommando                       |
|--------------|------------------------------------|
| CentOS/RHEL  | <code>/usr/libexec/qemu-kvm</code> |
| Debian       | <code>kvm</code>                   |
| Fedora       | <code>qemu-kvm</code>              |
| openSUSE     | <code>qemu-kvm</code>              |
| Ubuntu       | <code>kvm</code>                   |

**Tabelle 40.1** KVM-Kommandoname je nach Distribution

Nach dem Start des KVM-Kommandos wird die virtuelle Maschine in einem Fenster angezeigt oder muss durch einen VNC-Client gesteuert werden. Generell ist der direkte Einsatz des KVM-Kommandos aber selten zu empfehlen: Sie müssen die Hardware-Komponenten der virtuellen Maschine durch unzählige Optionen einstellen. Das macht den Einsatz von KVM unübersichtlich und fehleranfällig.

**libvirt-Werkzeuge** Der Einsatz der diversen libvirt-Werkzeuge vereinfacht die Administration virtueller Maschinen erheblich:

- ▶ Der Virtual Machine Manager (Programm- bzw. Paketname `virt-manager`) hilft mit einer grafischen Benutzeroberfläche beim Einrichten und Ausführen virtueller Maschinen.
- ▶ Wenn Sie lieber im Terminal arbeiten, können Sie mit der Shell `virsh` virtuelle Maschinen erzeugen, starten und wieder stoppen sowie andere Administrationsarbeiten durchführen.
- ▶ Daneben gibt es diverse Kommandos für Spezialaufgaben: `virt-clone` kopiert eine virtuelle Maschine, `virt-top` liefert ähnlich wie `top` eine Auflistung aller virtuellen Maschinen samt RAM- und CPU-Nutzung etc.

**Distributionen** Alle gängigen Distributionen stellen KVM- und libvirt-Pakete zur Verfügung. Da die Entwicklung von KVM aber in einem sehr hohen Ausmaß von Red Hat vorangetrieben wird, eignen sich RHEL oder RHEL-Clones wie CentOS besonders gut für den KVM-Einsatz. Wenn Sie die allerneuesten KVM-Features testen möchten, ist Fedora die ideale Spielwiese.

Beachten Sie, dass Red Hat KVM-Pakete nur mit der 64-Bit-Version seiner Enterprise-Distribution ausliefert. Auf einem Virtualisierungs-Host sollte der Einsatz einer 64-Bit-Installation aber ohnedies selbstverständlich sein.

### libvirt-Intern

libvirt ist eine Schnittstelle zur Verwaltung von virtuellen Maschinen und der dazugehörigen virtuellen Netzwerk- und Festplatten-Devices. Eine Voraussetzung für die Nutzung der libvirt-Werkzeuge besteht darin, dass auf dem Hostsystem der Dämon `libvirtd` läuft. Dieses Programm wird beim Hochfahren des Hostrechners durch das Init-System gestartet.

Die Steuerung der virtuellen Maschinen erfolgt wahlweise durch die Shell `virsh`, den Virtual Machine Manager oder durch andere libvirt-Kommandos. Jedes dieser Programme muss vorher eine Verbindung zum libvirt-Dämon herstellen. Der libvirt-Dämon erlaubt auch Netzwerkverbindungen, die üblicherweise durch SSH getunnelt werden.

Die libvirt-Werkzeuge können neben KVM auch das Virtualisierungssystem Xen steuern. In diesem Kapitel beziehe ich mich aber ausschließlich auf KVM.

KVM-Maschinen können via libvirt auf zwei Ebenen ausgeführt werden:

System- versus  
Benutzerebene

- ▶ **Benutzerebene** (`qemu:///session`): Diese Variante ist vor allem für die Desktop-Virtualisierung gedacht und gibt den virtuellen Maschinen weniger Zugriffsmöglichkeiten auf die Hardware des Hostrechners. Intern wird beim ersten Aufruf eines libvirt-Werkzeugs auf Benutzerebene ein eigener libvirtd-Prozess gestartet, dem nur die Rechte des aktuellen Benutzers zukommen. KVM-Maschinen auf Benutzerebene minimieren also die Sicherheitsrisiken durch die Virtualisierung.
- ▶ **Systemebene** (`qemu:///system`): Virtuelle Maschinen auf Systemebene sind besser für die Server-Virtualisierung geeignet, weil sie direkt auf Hardware-Komponenten des Hostrechners zugreifen können und weil mehr Möglichkeiten zur Integration der virtuellen Maschinen in das Netzwerk bestehen. Die libvirt-Prozesse kommunizieren dabei mit dem Dämon libvirtd, der mit root-Rechten läuft.

Bei der Kommunikation zwischen den libvirt-Werkzeugen und dem Dämon libvirtd bestehen starke Konfigurationsunterschiede zwischen den Distributionen. Ganz einfach ist es bei CentOS, Fedora und RHEL: Wenn Sie mit libvirtd auf Systemebene kommunizieren möchten, benötigen Sie root-Rechte. Der Virtual Machine Manager kann zwar mit Benutzerrechten gestartet werden, das Programm erwartet aber unmittelbar nach dem Start die Angabe des root-Passworts.

CentOS, Fedora,  
RHEL

Beachten Sie dabei, dass zwar die libvirt-Werkzeuge mit root-Rechten ausgeführt werden, nicht aber das eigentliche Virtualisierungskommando! Vielmehr starten die libvirt-Werkzeuge das Kommando `qemu-kvm` unter dem Benutzer-Account `qemu`. Auf diese Feinheit müssen Sie vor allem bei der richtigen Einstellung der Zugriffsrechte für Image- oder ISO-Dateien achten!

Auch unter Ubuntu kommunizieren libvirt-Werkzeuge, die mit root-Rechten ausgeführt werden, mit libvirtd auf Systemebene. Aber auch libvirt-Kommandos, die nur mit Benutzerrechten ausgeführt werden, dürfen mit libvirtd auf Systemebene kommunizieren, sofern der Benutzer der Gruppe libvirtd angehört! Genau genommen ist entscheidend, ob der Benutzer auf die Datei `/var/run/libvirt/libvirt-sock` zugreifen darf. Diese Datei gehört root und der Gruppe libvirtd.

Ubuntu

Die Zuordnung zur Gruppe libvirtd wird bei der Installation des Pakets `libvirt-bin` automatisch für den Benutzer hergestellt, der die Installation durchführt. Weitere Benutzer können mit dem folgenden Kommando der libvirtd-Gruppe hinzugefügt werden:

```
root# adduser loginname libvirtd    (Ubuntu)
```

**Debian** Debian verhält sich bei der libvirt-Konfiguration anders als Ubuntu: Dort haben alle Benutzer Lese- und Schreibrechte auf `/var/run/libvirt/libvirt-sock`. Eine Gruppenzuordnung ist deswegen nicht erforderlich.

**Konfiguration** Die Konfigurationsdateien des libvirt-Systems befinden sich im Verzeichnis `/etc/libvirt`. Besonders interessant ist die in diesem Verzeichnis enthaltene Datei `qemu.conf`: Sie gibt diverse Grundeinstellungen für das KVM-Kommando vor. Die Datei steuert unter anderem die Defaulteinstellungen des VNC- bzw. Spice-Servers der virtuellen Maschine. Dabei kommt standardmäßig die IP-Adresse 127.0.0.1 zum Einsatz. Somit sind nur lokale Verbindungen zulässig, wobei eine Weiterleitung via SSH durch Port Forwarding möglich ist.

Außerdem werden die Eigenschaften jeder virtuellen Maschine in einer XML-Datei im Verzeichnis `/etc/libvirt/qemu` festgehalten. Die meisten Einstellungen sind ohne weitere Erklärung verständlich und korrespondieren direkt mit entsprechenden KVM-Optionen. Im Detail ist das Format der libvirt-XML-Dateien auf folgender Seite dokumentiert:

<https://libvirt.org/format.html>

#### Ändern Sie die Beschreibung virtueller Maschinen immer durch »virsh edit«!

Sie sollten die XML-Dateien mit den Eckdaten einer virtuellen Maschine nicht direkt mit einem Editor ändern – sonst kann es passieren, dass ein anderes libvirt-Werkzeug Ihre Änderungen überschreibt. Verwenden Sie stattdessen das `virsh`-Kommando `edit`!

**Image-Dateien** Wenn Sie beim Einrichten virtueller Maschinen auf Disk Images zur Abbildung der virtuellen Datenträger zurückgreifen, werden diese standardmäßig im Verzeichnis `/var/lib/libvirt/images` gespeichert. Wenn Sie Disk Images in einem anderen Verzeichnis speichern möchten oder Logical Volumes, Festplattenpartitionen oder Netzwerkgeräte zur Speicherung der Datenträger nutzen möchten, müssen Sie vorher einen sogenannten Storage Pool einrichten. Am einfachsten gelingt das im Virtual Machine Manager. Alternativ führen Sie die entsprechenden `pool`-Kommandos innerhalb von `virsh` aus.

### Verhalten beim Neustart des Hostsystems

Was passiert mit virtuellen Maschinen, wenn Sie das Hostsystem herunterfahren?

- ▶ CentOS/RHEL sowie Debian und Ubuntu sichern mit dem `virsh`-Kommando `save` den Speicherinhalt aller durch `libvirtd` auf Systemebene ausgeführten virtuellen Maschinen. Beim nächsten Start des Rechners wird der Zustand der virtuellen



Maschinen automatisch wiederhergestellt (`restore`), d. h., die virtuellen Maschinen laufen weiter, als wäre in der Zwischenzeit nichts passiert.

Verantwortlich für diesen Mechanismus ist das Script `/usr/libexec/libvirt-guests.sh`, das vom `systemd`-Service `libvirt-guests` aufgerufen wird. Einige Konfigurationsparameter können Sie in `/etc/sysconfig/libvirt-guests` einstellen.

Bei der Sicherung bzw. Wiederherstellung mehrerer virtueller Maschinen muss jeweils deren gesamtes RAM auf der Festplatte gespeichert bzw. von dort gelesen werden. Das setzt ausreichend freien Speicherplatz im Verzeichnis `/var/lib/libvirt/qemu/save` voraus und dauert natürlich einige Zeit.

- ▶ Fedora agiert wie CentOS/RHEL, sofern Sie das Paket `libvirt-client` installieren. Anders als unter CentOS/RHEL wird das Paket nicht durch Abhängigkeiten automatisch installiert!
- ▶ Auch aktuelle SUSE-Distributionen verwenden den gleichen Mechanismus wie CentOS/RHEL. Das entsprechende Script befindet sich aber in `/usr/lib64/libvirt/libvirt-guests.sh`.

## Virtuelle Hardware

Beim Einrichten einer neuen virtuellen Maschine haben Sie eine Menge Wahlmöglichkeiten: Disk-Images im RAW- oder im QCOW2-Format, IDE- oder virtio-Festplattenadapter, das Grafiksystem auf der Basis von SDL, VNC oder Spice etc. Dieser Abschnitt fasst dazu die wichtigsten Informationen zusammen.

Grundsätzlich führt KVM eine vollständige Virtualisierung durch. Das in der virtuellen Maschine laufende Gastsystem benötigt also keine besonderen Treiber.

virtio-Treiber und  
Paravirtualisierung

Das Gastsystem kann freilich noch effizienter ausgeführt werden, wenn zur Kommunikation zwischen KVM und der virtuellen Maschine die optionalen virtio-Treiber zum Einsatz kommen. In der Fachsprache ist dann von »Paravirtualisierung« die Rede, d. h., das Gastsystem hilft gewissermaßen bei der Virtualisierung mit.

Bei Linux-Gästen stehen standardmäßig drei virtio-Treiber zur Beschleunigung von Festplatten-, Speicher- und Netzwerkzugriffen zur Verfügung. Es geht also nur darum, beim Einrichten der virtuellen Maschine die entsprechenden virtio-Komponenten auszuwählen.

Ein wenig diffiziler ist die Angelegenheit, wenn Sie Windows in einer KVM-Maschine ausführen möchten: In diesem Fall richten Sie die virtuelle Maschine zuerst mit traditionellen Hardware-Komponenten ein, also z. B. mit einer virtuellen IDE-Schnittstelle. Nach der Installation von Windows installieren Sie die virtio-Treiber, und erst dann

können Sie die virtio-Komponenten durch eine nachträgliche Veränderung der virtuellen Maschine aktivieren.

#### Virtuelle Datenträger

Um einem Gast eine virtuelle Festplatte anzubieten, wird häufig auf dem KVM-Host eine Image-Datei verwendet. Dabei unterstützen QEMU/KVM drei Image-Formate:

- ▶ **RAW-Format:** Beim RAW-Format werden die Blöcke der virtuellen Festplatte einfach 1:1 abgebildet. Sofern das Dateisystem des Hostrechners sogenannte *Sparse Files* unterstützt, werden Blöcke, die ausschließlich Nullen enthalten, nicht physikalisch gespeichert. Das funktioniert so unter anderem bei ext-, xfs- und btrfs-Dateisystemen und spart anfänglich eine Menge Platz. Das RAW-Format ist das einfachste und schnellste Image-Format für virtuelle Maschinen.
- ▶ **QCOW2-Format:** QCOW2 steht für *Qemu Copy-on-Write, Version 2*. Dieses Format bietet gegenüber RAW eine Menge Zusatzfunktionen: Die Datenblöcke werden erst bei Bedarf reserviert, ohne ein Sparse-kompatibles Dateisystem vorauszusetzen. Außerdem kann das virtuelle Dateisystem komprimiert und verschlüsselt werden. Schließlich bieten QCOW2-Images die Möglichkeit, Snapshots zu verwalten. QCOW2-Images sind langsamer als RAW-Images, der Geschwindigkeitsnachteil ist aber nicht mehr so groß wie in der Vergangenheit. QCOW2 ist seit mehreren Jahren das Defaultdateisystem des Virtual Machine Managers.
- ▶ **QED-Format:** Das *QEMU Enhanced Disk Format*, kurz QED, liegt in seiner Geschwindigkeit zwischen RAW und QCOW2. Das Format bietet aber weniger Funktionen als QCOW2. Insbesondere fehlt die praktische Snapshot-Funktion.

Anstelle von Image-Dateien können Sie auch Festplattenpartitionen, Logical Volumes oder iSCSI-Devices als virtuelle Festplatten nutzen. Diese Varianten bieten in großen Virtualisierungssystemen administrative Vorteile, aber keine nennenswert höhere Geschwindigkeit im Vergleich zu RAW-Images.

#### Netzwerk-anbindung

Um die virtuelle Maschine mit dem lokalen Netzwerk oder dem Internet verbinden zu können, müssen Sie sie mit einem Netzwerkadapter ausstatten. Bei Linux-Gästen ist der virtio-Treiber die erste Wahl. Bei Windows-Gästen haben Sie unter anderem die Wahl zwischen einem RTL-8139- oder einem Intel-E1000-Netzwerkadapter.

Die zweite Frage ist, wie Sie den Adapter mit Ihrem Netzwerk verbinden:

- ▶ **NAT:** Standardmäßig entscheiden sich das KVM-Kommando bzw. die libvirt-Werkzeuge für die NAT-Variante, also für Network Address Translation. Damit wird der Internetzugang des Hosts an den Gast weitergeleitet. Der Gast ist aber weder im lokalen Netzwerk noch im Internet sichtbar.
- ▶ **Netzwerkbrücken:** Für den Server-Einsatz müssen Sie die virtuelle Maschine durch eine Netzwerkbrücke oder durch Routing mit dem Netzwerk bzw. Internet verbinden. Das erfordert eine spezielle Netzwerkkonfiguration des KVM-Hosts.

- ▶ **MacVTap:** Aktuelle Versionen der libvirt-Werkzeuge unterstützen mit MacVTap-Devices eine dritte Variante, bei der ein virtueller Netzwerkadapter mit einem physischen verbunden wird.

Zumindest während der Installation müssen Sie die Ausgaben der virtuellen Maschine sehen. Der Gast braucht also ein eigenes Grafiksystem. Dazu wird eine VGA-kompatible Grafikkarte emuliert, deren Ausgaben dann via VNC oder Spice in einem Fenster angezeigt werden. Für den 2D-Einsatz funktioniert dies selbst in hoher Auflösung gut. KVM bietet zurzeit aber keine Unterstützung für 3D-Funktionen.

Grafik

VNC und Spice sind netzwerktauglich. Für die relativ neue Spice-Architektur sprechen die etwas höhere Geschwindigkeit und der Umstand, dass auch Audio-Ausgaben der virtuellen Maschine über das Netzwerk an den lokalen Spice-Client weitergeleitet werden können. Gegen Spice spricht die schlechte Verfügbarkeit von Spice-Clients außerhalb der Linux-Welt.

## 40.2 KVM ohne libvirt

Grundsätzlich rate ich Ihnen davon ab, KVM ohne die libvirt-Werkzeuge zu benutzen. Dass ich Ihnen in diesem Abschnitt dennoch zeige, wie Sie eine virtuelle Maschine ohne libvirt-Overhead zum Laufen bringen, hat primär didaktische Gründe: Es ist immer gut zu wissen, was hinter den Kulissen vor sich geht. Außerdem werden Sie die libvirt-Werkzeuge mehr schätzen, wenn Sie sehen, mit wie vielen Optionen Sie sich beim KVM-Kommando auseinandersetzen müssen.

Bevor Sie mit dem Kommando `qemu-kvm` oder `kvm` eine virtuelle Maschine starten können, müssen Sie mit dem Kommando `qemu-img` eine Image-Datei für die virtuelle Festplatte einrichten. Beim Aufruf von `qemu-img` geben Sie mit `-f raw` oder `-f qcow2` das Image-Format an. Die Größe der Datei geben Sie in MiB (100M) oder GiB (4G) an.

Image-Datei erstellen

```
user$ qemu-img create -f qcow2 disk.img 16G
```

KVM starten Sie je nach Distribution nun mit dem Kommando `qemu-kvm`, `kvm` oder `/usr/libexec/qemu-kvm`. Es gilt die folgende Syntax:

Das KVM-Kommando

```
user$ qemu-kvm [optionen] [image-datei]
```

Eine bereits installierte virtuelle Maschine können Sie einfach so ausführen:

```
root# qemu-kvm disk.img
```

Damit stattet das KVM-Kommando die virtuelle Maschine einfach mit einigen standardmäßig vorgesehenen Hardware-Komponenten aus: mit einem CPU-Core, mit 128 MiB RAM, mit einer Festplatte und mit einem Netzwerkadapter mit der immer

gleichen MAC-Adresse 52:54:00:12:34:56. Die virtuelle Maschine wird über NAT mit dem Netzwerk des Hosts verbunden und bezieht über eine von KVM gestartete Instanz von `dnsmasq` ihre Netzwerkkonfiguration.

Da diese Defaultkonfiguration selten praktikabel ist, reichen KVM-Kommandos in der Praxis oft über mehrere Zeilen, in denen alle erdenklichen Hardware-Details eingestellt werden. Das folgende Kommando startet die Installation einer Linux-Distribution. Die virtuelle Maschine wird mit zwei CPUs und 1024 MiB RAM ausgestattet. Die virtuelle Festplatte und der Netzwerkadapter werden über die `virtio`-Treiber des Gasts angesteuert. Als Installationsquelle dient eine ISO-Datei, die im Gast als virtuelles DVD-Laufwerk sichtbar ist und beim ersten Start als Boot-Medium gilt.

Zur Steuerung der virtuellen Maschine startet das KVM-Kommando einen nur für `localhost` erreichbaren VNC-Server (Display 0, also IP-Port 5900). Dank `-k de` gilt das deutsche Tastaturlayout. `-usbdevice tablet` ist erforderlich, damit die Mausbedienung funktioniert und die Zeigerposition des VNC-Clients mit der Zeigerposition der virtuellen Maschine synchronisiert werden kann.

```
root# qemu-kvm -machine accel=kvm -smp 2 -m 1024 -boot once=d \
    -cdrom linux.iso -drive file=disk.img,if=virtio,format=qcow2 \
    -net user -net nic,macaddr=52:54:00:12:e4:4e,model=virtio \
    -vga cirrus -vnc 127.0.0.1:0 -k de -usb -usbdevice tablet
```

Bei virtuellen Maschinen, die vom Virtual Machine Manager bzw. von den `libvirt`-Werkzeugen gestartet werden, ist die KVM-Optionenliste übrigens noch viel länger. Davon können Sie sich mit `ps ax | grep kvm` überzeugen.

Um die virtuelle Maschine zu bedienen, müssen Sie nun noch einen VNC-Client starten. Bei Gnome-basierten Distributionen ist häufig das Programm `Vinagre` vorinstalliert. Eine gute Alternative ist das Programm `TightVNC`, das Sie unter Fedora/CentOS/RHEL im Paket `tigervnc` finden, unter Ubuntu in `xtightvncviewer`. Wenn dieses Programm installiert ist, stellen Sie die Verbindung zu dem VNC-Server her, der auf dem KVM-Host läuft:

```
user@client$ vncviewer localhost:0
```

Gegebenenfalls müssen Sie den VNC-Viewer neu starten, wenn dieser nach dem Wechsel vom Text- in den Grafikmodus einen Teil des Bildschirms abschneidet.

Nachdem Sie die virtuelle Maschine gestartet haben, führen Sie im VNC-Client eine ganz gewöhnliche Installation durch. Wenn Sie die virtuelle Maschine nach Abschluss der Installation später neuerlich starten, können Sie auf die Optionen `-cdrom linux.iso -boot once=d` verzichten.

Für die Installation von Windows müssen Sie KVM mit etwas anderen Optionen starten: ohne virtio-Adapter, mit einem Standard-VGA-Adapter und mit der Option `-localtime`, weil Windows annimmt, dass die Uhr der virtuellen Maschine die lokale Zeit enthält:

Windows  
installieren

```
root# qemu-kvm -enable-kvm -m 1024 -smp 2 -boot once=d -cdrom windows.iso \
      -drive file=disk.img,format=qcow2 \
      -net user -net nic,macaddr=52:54:00:12:e4:4e \
      -vga std -vnc 127.0.0.1:0 -k de -usb -usbdevice tablet
```

## 40.3 Der Virtual Machine Manager

Der Virtual Machine Manager (Programm- und Paketname `virt-manager`) ist der beste Weg, um mit KVM vertraut zu werden. Das Programm hat auch für Profis eine Menge zu bieten und ist für viele Virtualisierungsaufgaben absolut ausreichend.

Damit Sie das Programm benutzen können, müssen Sie eine Verbindung zum `libvirt`-Dämon herstellen. Bei den meisten Distributionen erfolgt der Verbindungsaufbau zum bereits vorhandenen Eintrag `QEMU/KVM` automatisch. Unter CentOS, Fedora und RHEL müssen Sie beim Start des Virtual Machine Managers das `root`-Passwort angeben.

Unter Ubuntu funktioniert der Verbindungsaufbau nur, wenn der Benutzer zur Gruppe `libvirt` gehört. Ist das nicht der Fall, führen Sie `sudo adduser loginname libvirt` aus und loggen sich aus und neu ein.

Das Hauptfenster des Virtual Machine Managers enthält eine Liste aller `libvirt`-Verbindungen (siehe [Abbildung 40.1](#)). Standardmäßig besteht diese Liste nur aus einem Eintrag, nämlich `QEMU/KVM` für das lokale Virtualisierungssystem. Sie können aber mit `DATEI • VERBINDUNG HINZUFÜGEN` die Eckdaten weiterer KVM-Hosts angeben.

Virtuelle  
Maschinen  
ausführen

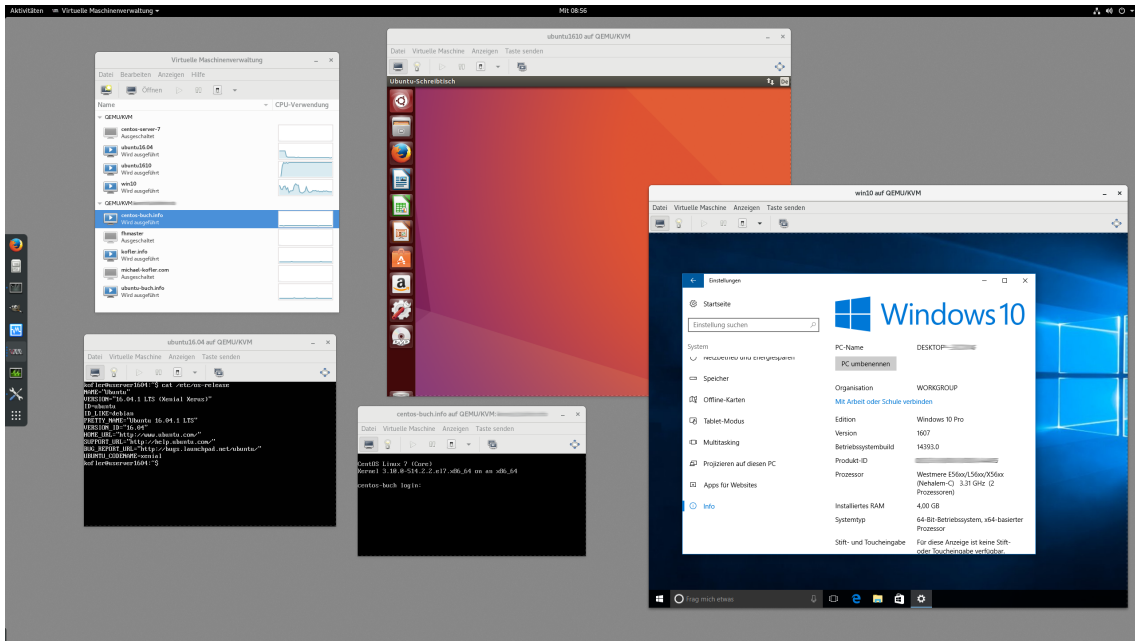
Ein Doppelklick auf einen Eintrag dieser Liste stellt die Verbindung zum KVM-Host her und zeigt dann alle virtuellen Maschinen dieses Hosts an. Bei jeder virtuellen Maschine zeigt ein Icon, ob die Maschine heruntergefahren ist, läuft oder pausiert.

Um eine virtuelle Maschine zu starten, klicken Sie deren Eintrag mit der rechten Maustaste an und führen `AUSFÜHREN` aus. Ein Doppelklick auf den Eintrag öffnet ein neues Fenster, das in zwei Ansichten den Zustand der virtuellen Maschine zeigt:

- **Die Konsolenansicht** zeigt das Grafiksystem der virtuellen Maschine. Hier sehen Sie die Ausgaben der virtuellen Maschine und können per Tastatur und Maus Eingaben durchführen. Bei virtuellen Maschinen, die im Textmodus laufen, wird der Mauscursor durch einen Klick in der virtuellen Maschine gleichsam eingefangen.

`Strg` + `Alt` löst den Cursor wieder.

- Die Detailansicht zeigt die Eckdaten der virtuellen Maschine an. Hier können Sie die Hardware-Ausstattung der virtuellen Maschine verändern. Die meisten Änderungen können allerdings nur durchgeführt werden, wenn die virtuelle Maschine vorher heruntergefahren wird.



**Abbildung 40.1** Der Virtual Machine Manager auf einem Fedora-Host mit virtuellen CentOS-, Ubuntu- und Windows-Installationen

Um zwischen den beiden Ansichten umzuschalten, führen Sie ANZEIGEN • KONSOLLE bzw. ANZEIGEN • DETAILS aus bzw. klicken in der Symbolleiste auf die entsprechenden Buttons.

#### Administration im Netzwerk via SSH

Sie können den Virtual Machine Manager auch verwenden, um einen via SSH erreichbaren externen KVM-Host zu administrieren. Dazu definieren Sie zuerst mit DATEI • VERBINDUNG HINZUFÜGEN eine Verbindung zu diesem Server, wobei Sie als Verbindungsmethode SSH auswählen (siehe [Abbildung 40.2](#)). Beim Verbindungsaufbau müssen Sie zudem das entsprechende Login-Passwort angeben.

Wenn der externe KVM-Host unter RHEL oder Fedora läuft, erfordert die libvirt-Administration root-Rechte und somit einen root-Login via SSH. Aus Sicherheitsgründen sind SSH-Server aber häufig so konfiguriert, dass ein direkter root-Login unmöglich ist. Ein Kompromiss kann so aussehen, dass Sie den SSH-Server so konfigurieren, dass ein root-Login nur bei einer Authentifizierung durch einen Schlüssel akzeptiert wird, nicht aber per Passwort. Generell funktioniert die Administration externer KVM-Hosts nur mit einem SSH-Schlüssel komfortabel.

The screenshot shows a dialog box titled "Verbindung hinzufügen" (Add Connection). It contains the following fields and options:

- Hypervisor:** A dropdown menu with "QEMU/KVM" selected.
- Mit entferntem Rechner verbinden:** A checked checkbox.
- Methode:** A dropdown menu with "SSH" selected.
- Benutzername:** A text input field containing "root".
- Rechnername:** A dropdown menu with "externer-kvm-host.com" selected.
- Automatische Verbindung:** An unchecked checkbox.
- Adresse erstellen:** A text field showing the generated address: "qemu+ssh://root@externer-...".
- Buttons:** "Abbrechen" (Cancel) and "Verbinden" (Connect).

Abbildung 40.2 Verbindungsaufbau via SSH

### Eine neue virtuelle Maschine einrichten

Das Einrichten einer neuen virtuellen Maschine beginnt mit dem Button NEUE VIRTUELLE MASCHINE ERSTELLEN. Bei der Einstellung der Eckdaten hilft ein Assistent in fünf Schritten:

- ▶ Im ersten Schritt geben Sie an, auf welchem KVM-Host die virtuelle Maschine installiert werden soll. Diese Auswahlmöglichkeit ist dann relevant, wenn Sie mit dem Virtual Machine Manager nicht nur einen lokalen KVM-Host, sondern mehrere Virtualisierungsrechner administrieren.

Außerdem geben Sie hier die Installationsquelle an. Bei einer Linux-Installation handelt es sich üblicherweise um eine ISO-Datei. Es ist aber auch möglich, die Installationsdaten vom DVD-Laufwerk des Hostrechners zu lesen. Die Option VORHANDENES FESTPLATTEN-ABBILD IMPORTIEREN erzeugt eine Kopie einer bereits vorhandenen Image-Datei mit einer virtuellen Maschine.

- ▶ Wenn Sie im ersten Schritt ein ISO-Abbild oder eine CD/DVD als Installationsquelle ausgewählt haben, können Sie im zweiten Schritt den Dateinamen einer ISO-Datei oder das CD/DVD-Laufwerk angeben. Der Button DURCHSUCHEN führt in einen Dialog, der vorerst nur die dem Virtual Machine Manager bekannten STORAGE POOLS anzeigt. Um eine ISO-Datei direkt auszuwählen, müssen Sie in diesem Dialog den Button LOKAL DURCHSUCHEN anklicken.

Manchmal gelingt es dem Assistenten, den Typ des Betriebssystems und seine Version aus dem Installationsmedium zu entnehmen – also z. B. LINUX und FEDORA 23. Gelingt dies nicht, müssen Sie diese Daten selbst angeben. Die Einstellungen sind erforderlich, damit der Assistent die für das Gastsystem optimalen virtuellen Hardware-Komponenten einrichtet.

- ▶ Im dritten Schritt geben Sie an, wie viel Speicher (RAM) und wie viele CPU-Cores Sie der virtuellen Maschine zuweisen möchten.
- ▶ Im vierten Schritt richten Sie die virtuelle Festplatte ein. Normalerweise werden Sie die bereits vorselektierte Option `PLATTENABBILD AUF FESTPLATTE DES SYSTEMS ERSTELLEN` nutzen. Die neue Image-Datei wird standardmäßig im Verzeichnis `/var/lib/libvirt/images` angelegt. Wählen Sie die Größe der virtuellen Festplatte nicht zu klein – eine nachträgliche Vergrößerung ist nur mit großem Aufwand möglich.

Neben der Größe können Sie auch auswählen, ob der virtuelle Festplattenspeicher sofort zugewiesen werden soll. Das ist effizienter und schließt aus, dass zu einem späteren Zeitpunkt vielleicht zu wenig Platz auf der Festplatte ist, um dem steigenden Platzbedarf der virtuellen Maschine gerecht zu werden. Allerdings geht damit der gesamte Platz sofort im Dateisystem verloren, während die Datei für die virtuelle Festplatte standardmäßig erst bei Bedarf wächst.

Die Alternative `VERWALTETEN ODER ANDEREN SPEICHER WÄHLEN` ist nur dann von Relevanz, wenn Sie vorher mit `BEARBEITEN • VERBINDUNGSDetails` im Dialogblatt `SPEICHER` weitere Speicherpools eingerichtet haben. Ein derartiger Pool kann z. B. ein beliebiges Verzeichnis im Dateisystem oder ein LVM-System sein.

- ▶ Im fünften Schritt können Sie der virtuellen Maschine einen Namen geben und zwischen verschiedenen Netzwerkoptionen wählen. Standardmäßig verwendet der Virtual Machine Manager das Verfahren NAT. Damit werden die virtuellen Maschinen dem privaten Netzwerk `192.168.124.0/24` zugeordnet. Die virtuelle Maschine kann so den Internetzugang des Hostrechners nutzen, aber keine Verbindungen zu anderen Rechnern in Ihrem lokalen Netzwerk herstellen.

Mit dem Button `FERTIG` wird die Konfiguration beendet und die neue virtuelle Maschine sofort gestartet. Wenn Sie das nicht wünschen, aktivieren Sie im letzten Dialogblatt des Assistenten die Option `KONFIGURATION BEARBEITEN VOR DER INSTALLATION`. Damit gelangen Sie nach dem Ende des Assistenten in einen Dialog, der die Hardware-Komponenten der virtuellen Maschine zusammenfasst (siehe [Abbildung 40.3](#)).

Die virtuelle Maschine startet, sobald Sie die Konfiguration abgeschlossen haben. Die Ausgaben der virtuellen Maschine sehen Sie in einem neuen Fenster des Virtual Machine Managers. Hinter den Kulissen agiert dieses Fenster als VNC- oder Spice-Client.



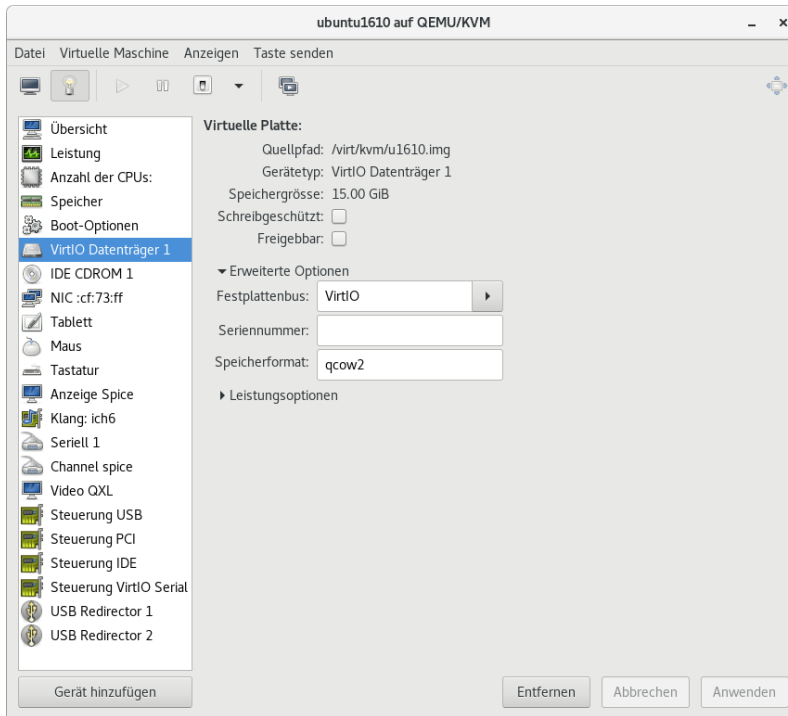


Abbildung 40.3 Hardware-Verwaltung im Virtual Machine Manager

### Virtuelle Maschinen stoppen

Die virtuellen Maschinen laufen vollkommen unabhängig vom Virtual Machine Manager! Sie können also die Fenster des Virtual Machine Managers schließen und später wieder öffnen – die virtuellen Maschinen laufen in der Zwischenzeit weiter. Sie können sich sogar aus- und neu einloggen, ohne die Ausführung von virtuellen Maschinen zu beeinträchtigen.

Es gibt viele Möglichkeiten, eine virtuelle Maschine zu stoppen:

- ▶ HERUNTERFAHREN • NEUSTART sendet ein entsprechendes ACPI-Ereignis an die virtuelle Maschine. Wenn die virtuelle Maschine ACPI-Ereignisse verarbeitet, leitet sie einen Shutdown und anschließend einen Neustart ein.
- ▶ HERUNTERFAHREN • HERUNTERFAHREN leitet via ACPI einen Shutdown ein.
- ▶ HERUNTERFAHREN • ZURÜCKSETZEN ERZWINGEN beendet die Ausführung der virtuellen Maschine sofort – so, als würden Sie bei einem realen Rechner das Stromkabel ziehen. Naturgemäß sollten Sie versuchen, diese Variante des Ausschaltens zu vermeiden, da sie mit Datenverlusten verbunden sein kann.

- ▶ HERUNTERFAHREN • SPEICHERN speichert den Inhalt des virtuellen RAMs der Maschine in einer Datei und beendet dann die Ausführung. Wird die virtuelle Maschine später wieder gestartet, befindet sie sich exakt im selben Zustand wie beim Herunterfahren.
- ▶ Natürlich können Sie auch die virtuelle Maschine an sich herunterfahren oder neu starten, z. B. durch die Kommandos `shutdown now`, `reboot` oder `halt -p`.

### Minimalinstallation von CentOS/RHEL

Für Server-Aufgaben reicht oft eine Minimalinstallation ohne grafische Benutzeroberfläche. Dazu laden Sie am besten das Minimal-Image der gewünschten Distribution herunter und installieren dieses.

Immerhin wird standardmäßig ein SSH-Server installiert und eine Firewall eingerichtet. Auch SELinux ist aktiv. Die gesamte weitere Administration muss nun mit textbasierten Werkzeugen erfolgen und setzt daher gute Fedora- oder RHEL-Grundlagenkenntnisse voraus. Das betrifft auch die Netzwerkkonfiguration – standardmäßig ist nur die Loopback-Schnittstelle aktiv.

#### Manuelle Netzwerk- konfiguration

Somit ist als Nächstes eine manuelle Netzwerkkonfiguration notwendig. Mit `ip addr` ermitteln Sie zuerst den Namen der Netzwerkschnittstelle. Ich verwende in den folgenden Beispielen `eth0`; der Name kann aber auch `enp0s4` oder so ähnlich lauten.

#### Ad-hoc-Netzwerkkonfiguration zur Installation von Paketen

Während der ersten Konfigurationsarbeiten steht Ihnen als einziger Editor `vi` zur Verfügung. Wenn Sie einen anderen Editor vorziehen, können Sie die Netzwerkschnittstelle `eth0` vorweg durch das Kommando `dhclient eth0` aktivieren – einmal vorausgesetzt, dass die virtuelle Maschine in einem Netzwerk mit DHCP-Server läuft. Anschließend können Sie mit `yum` bzw. `dnf` einen anderen Editor installieren.

Nun stellen Sie in der bereits vorhandenen Datei `/etc/sysconfig/network-scripts/ifcfg-eth0` die Option `ONBOOT` auf `yes`:

```
# Datei /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
HWADDR=52:54:00:xx:xx:xx    (eigene MAC-Adresse)
...
ONBOOT=yes
```

Bei einer statischen Konfiguration muss die Datei dem folgenden Muster entsprechen, wobei Sie die IP-Adressen und -Masken durch eigene Werte ersetzen müssen:

```
# Datei /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
HWADDR=52:54:00:xx:xx:xx      (eigene MAC-Adresse)
NM_CONTROLLED=no
ONBOOT=yes
BOOTPROTO=none
TYPE=Ethernet
USERCTL=no
IPV6INIT=no
IPADDR=10.0.17.33
PREFIX=24
GATEWAY=10.0.17.1
```

Den oder die Nameserver tragen Sie in `/etc/resolv.conf` ein:

```
# /etc/resolv.conf
nameserver 10.0.17.1    # erster DNS
nameserver 10.0.17.2    # zweiter DNS
```

Um den Hostnamen einzustellen, führen Sie `hostnamectl set-hostname name` aus. Bei einer statischen IP-Konfiguration ist es zumeist zweckmäßig, die Zuordnung der IP-Adresse des Rechners zu seinem Hostnamen auch in `/etc/hosts` einzutragen:

```
# /etc/hosts
...
10.0.17.33      myhostname.mydomainname myhostname
```

Das folgende Kommando aktiviert die Netzwerkeinstellungen:

```
root# ifup eth0
```

### Minimalinstallation von Ubuntu Server

Wenn Sie in einer virtuellen Maschine eine minimale Ubuntu-Installation durchführen möchten, sollten Sie als Installationsquelle das ISO-Image der Server-CD verwenden und nicht die ISO-Datei zur Desktop-Installation. Am Beginn einer Ubuntu-Server-Installation können Sie mit `[F4]` die Installationsvariante `EINE MINIMALE VIRTUELLE MASCHINE INSTALLIEREN` auswählen. Der Vorteil gegenüber einer herkömmlichen Server-Installation besteht darin, dass ein spezieller Kernel eingesetzt wird, der für den Einsatz in virtuellen Maschinen optimiert ist und mit wenig zusätzlichem Ballast auskommt.

Anders als bei CentOS/RHEL kümmert sich das Installationsprogramm um die Netzwerkkonfiguration. Wenn Sie die Konfiguration später ändern möchten, ist der zentrale Dreh- und Angelpunkt die Datei `/etc/network/interfaces`. Bezieht die virtuelle Maschine ihre Netzwerkparameter von einem DHCP-Server, muss diese Datei wie

Netzwerk-  
konfiguration

folgt aussehen, wobei Sie gegebenenfalls eth0 durch einen anderen Schnittstellenamen ersetzen müssen, z. B. ens3. Welcher Name tatsächlich zum Einsatz kommt, können Sie mit `ip addr` herausfinden.

```
# /etc/network/interfaces
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
```

Bei einer statischen Konfiguration können Sie sich am folgenden Muster orientieren:

```
# /etc/network/interfaces
auto lo
iface lo inet loopback

auto eth0 inet static
    address 10.0.17.33
    netmask 255.255.255.0
    gateway 10.0.17.1
```

Den oder die Nameserver tragen Sie in `/etc/resolv.conf` ein. Damit Änderungen an der Konfiguration wirksam werden, führen Sie das folgende Kommando aus:

```
root# ifup eth0
```

## Windows-Installation

KVM ist Windows-kompatibel, und prinzipiell unterscheidet sich eine Windows-Installation nur unwesentlich von einer Linux-Installation. Sie beginnen abermals damit, dass Sie eine neue virtuelle Maschine einrichten. Achten Sie darauf, dass Sie im zweiten Schritt des Assistenten den Betriebssystemtyp WINDOWS und die entsprechende Windows-Version auswählen! Nur dann verwendet der Virtual Machine Manager für Windows geeignete virtuelle Hardware-Komponenten.

Standardmäßig kommen die folgenden Hardware-Komponenten zum Einsatz:

- ▶ eine CPU mit zwei Cores
- ▶ ein RTL-8139-Netzwerkadapter im NAT-Netzwerk
- ▶ eine IDE-Festplatte
- ▶ eine Grafikkarte mit nahezu beliebiger Auflösung

### virtio-Treiber einrichten

Um Netzwerk- und Festplattenzugriffe effizienter zu gestalten, sollten Sie nach der Inbetriebnahme von Windows unbedingt virtio-Treiber installieren und anschließend die Hardware-Einstellungen der virtuellen Maschine entsprechend ändern. Auf

den folgenden Webseiten finden Sie Download-Links für eine ISO-Datei mit Treibern für gängige Windows-Versionen:

[https://www.linux-kvm.org/page/WindowsGuestDrivers/Download\\_Drivers](https://www.linux-kvm.org/page/WindowsGuestDrivers/Download_Drivers)

[https://fedoraproject.org/wiki/Windows\\_Virtio\\_Drivers](https://fedoraproject.org/wiki/Windows_Virtio_Drivers)

Diese ISO-Datei laden Sie auf das Hostsystem herunter, also nicht in der virtuellen Maschine. Anschließend fahren Sie Ihren Windows-Gast herunter. Mit ANZEIGEN • DETAILS wechseln Sie in die Hardware-Ansicht der virtuellen Maschine. Dort geben Sie die ISO-Datei als Quelle für das virtuelle CD-Laufwerk an. Außerdem fügen Sie der virtuellen Maschine zusätzlich zu den vorhandenen Netzwerk- und Festplattenadaptern eine neue virtio-Netzwerkarte und eine virtio-Festplatte hinzu. Die Image-Datei für die neue Festplatte muss nicht groß sein – es geht nur darum, dass Windows beim nächsten Start die neuen Hardware-Komponenten bemerkt.

Anschließend starten Sie den Windows-Gast neu und rufen im Windows-Menü das Programm GERÄTE-MANAGER auf (siehe [Abbildung 40.4](#)). Dort erscheinen mehrere noch unbekannte Hardware-Komponenten, unter anderem als ETHERNET- und SCSI-CONTROLLER. Bei jeder dieser Komponenten öffnen Sie nun per Doppelklick den Eigenschaftendialog, klicken auf TREIBER AKTUALISIEREN und schließlich auf AUF DEM COMPUTER NACH TREIBERSOFTWARE SUCHEN. Bei dieser Suche müssen Sie mit-  
helfen. Sie geben als Ort der Treibersoftware das DVD-Laufwerk an, also üblicherweise »D:«.

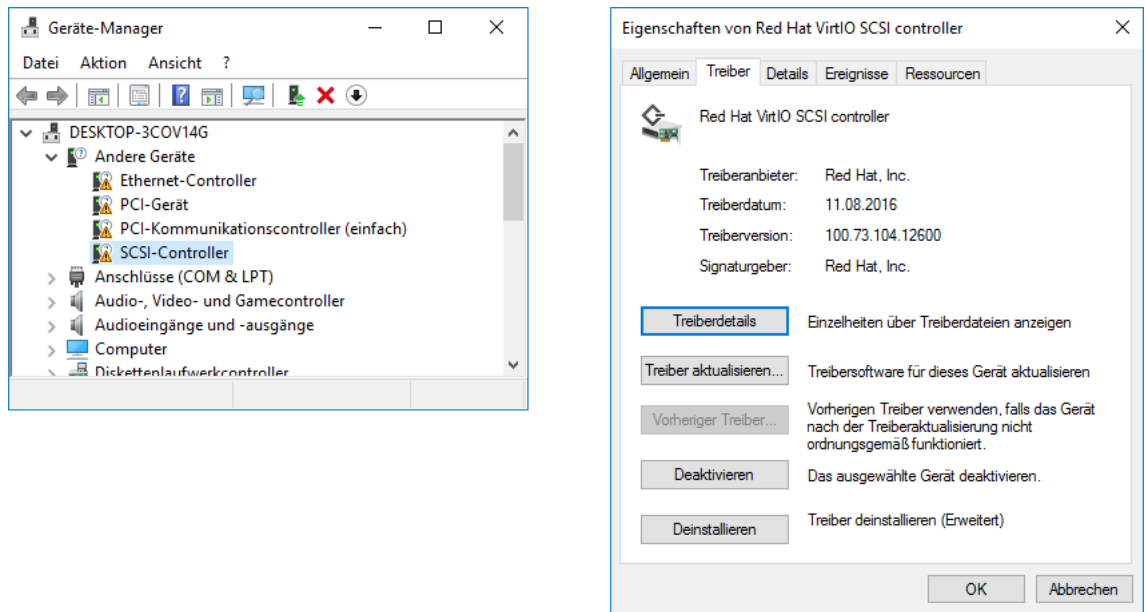


Abbildung 40.4 virtio-Treiberinstallation unter Windows 10

Nun fahren Sie Windows neuerlich herunter und entfernen in der Hardware-Übersicht des Virtual Machine Managers die vorhin eingerichtete virtio-Festplatte und den virtio-Netzwerkadapter. Außerdem stellen Sie bei der IDE-Festplatte bei den erweiterten Optionen den Festplattenbus auf VIRTIO sowie bei der Netzwerkkarte das Gerätemodell ebenfalls auf VIRTIO. Das Festplatten-Image und die restlichen Parameter der beiden Geräte bleiben unverändert. Vergewissern Sie sich nach dem Neustart von Windows mit einem Blick in den Geräte-Manager, dass das Laufwerk und der Netzwerkadapter nun als RED HAT VIRTIO DEVICES bezeichnet werden.

## 40.4 libvirt-Kommandos

Nachdem ich Ihnen im vorigen Abschnitt recht ausführlich den Virtual Machine Manager als wichtigsten Vertreter der libvirt-Werkzeuge präsentiert habe, folgen in diesem Abschnitt einige Kommandos, deren Nutzung keine grafische Benutzeroberfläche voraussetzt.

### virsh

Mit dem Kommando `virsh` starten Sie die libvirt-Shell. Darin können Sie Kommandos zur Verwaltung aller virtuellen Maschinen ausführen, die libvirt bekannt sind.

```
root# virsh
virsh# list --all
Id      Name                               Status
-----
 9      ubuntu                             laufend
13      fedora                             laufend
15      win10                              laufend
16      userver1604                        laufend
-      centos                             ausschalten
virsh# start centos
Domain centos gestartet
virsh# exit
```

Mit der `virsh` können Sie virtuelle Maschinen mit `start`, `suspend/resume`, `shutdown`, `destroy`, `save/restore` bzw. `undefine` starten, vorübergehend anhalten und wieder fortsetzen, geordnet herunterfahren (ACPI-Shutdown), sofort ausschalten, speichern und wieder fortsetzen oder aus der Liste der libvirt-Definitionen löschen.

`edit` ermöglicht es, die XML-Datei mit der Beschreibung der virtuellen Maschine direkt in einem Editor zu bearbeiten. Als Editor kommt üblicherweise `vi` zum Einsatz. Wenn Sie mit einem anderen Editor arbeiten möchten, müssen Sie die Umgebungsvariable `EDITOR` entsprechend einstellen.

Bei allen oben aufgezählten `virsh`-Kommandos müssen Sie den Namen der virtuellen Maschine, deren UUID-Nummer oder bei laufenden virtuellen Maschinen die ID-Nummer angeben. Die UUID-Nummer geht aus der XML-Definitionsdatei hervor. Auskunft über die ID-Nummern laufender virtueller Maschinen gibt das `virsh`-Kommando `list`.

Sofern `virsh` erkennt, dass der Rechner ein KVM-Host ist, stellt es nach Möglichkeit eine Verbindung zu dem auf Systemebene laufenden Dämon `libvirtd` her. Wenn `virsh` nur mit Benutzerrechten ausgeführt wird, erfordert eine Verbindung auf Systemebene unter Ubuntu die Zugehörigkeit zur Gruppe `libvirtd`. Unter CentOS, Fedora und RHEL müssen Sie das Kommando als `root` starten, wenn Sie auf Systemebene arbeiten möchten.

Verbindungs-  
aufbau

Es ist möglich, innerhalb der `virsh`-Shell die Verbindung mit dem Kommando `connect` zu verändern: `qemu:///session` bezeichnet dabei eine Verbindung auf Benutzerebene, `qemu:///system` eine Verbindung auf Systemebene.

```
virsh# connect qemu:///system
```

Via SSH können Sie auch eine Verbindung zum Dämon `libvirtd` auf einem anderen Rechner herstellen. Wenn es sich beim KVM-Host um einen RHEL- oder Fedora-Rechner handelt, müssen Sie als Benutzername `root` angeben, weil auf diesen Systemen nur `root` eine Verbindung zum `libvirt`-Systemdämon herstellen darf. Beachten Sie, dass nach `qemu+ssh`: nur zwei Schrägstriche folgen, nicht drei! Wenn auf dem KVM-Host aus Sicherheitsgründen ein `root`-Login mit Passwortangabe via SSH unmöglich ist, müssen Sie vor dem ersten Verbindungsaufbau Ihren öffentlichen SSH-Schlüssel auf dem KVM-Host einrichten.

```
virsh# connect qemu qemu+ssh://user@hostname/system
user@hostname's password: *****
```

Anstatt `virsh` interaktiv zu verwenden, können Sie ein einzelnes `virsh`-Kommando in der Form `virsh kommando` ausführen:

```
root# virsh list --all
...
```

Wenn Sie dabei nicht die von `virsh` standardmäßig vorgesehene Verbindung verwenden möchten, geben Sie die Verbindungszeichenkette mit der Option `-c` an:

```
root# virsh -c qemu:///session list --all
...
```

Zum Abschluss stelle ich Ihnen einige ausgewählte `virsh`-Kommandos vor. man `virsh` dokumentiert mindestens hundert weitere Kommandos. Innerhalb der Shell erhalten Sie mit `help name` eine ausführliche Beschreibung des jeweiligen Kommandos.

Ausgewählte  
virsh-Kom-  
mandos

- ▶ `connect qemu:///session`: stellt eine gewöhnliche Benutzerverbindung zu `libvirtd` her. Auf diese Weise können Sie eigene virtuelle Maschinen verwalten.
- ▶ `connect qemu:///system`: stellt eine `root`-Verbindung zu `libvirtd` her. Das ist nur erforderlich, wenn globale KVM-Optionen oder virtuelle Netzwerke verändert werden sollen.
- ▶ `list [--inactive oder --all]`: listet alle laufenden virtuellen Maschinen auf. Wenn Sie nur die gerade nicht aktiven oder überhaupt alle Maschinen auflisten möchten, geben Sie die Optionen `--inactive` oder `--all` an.
- ▶ `start <vmname>`: startet die angegebene virtuelle Maschine. Wenn Sie mit der Maschine im Grafikmodus kommunizieren möchten, verwenden Sie dazu entweder das Programm `virt-viewer` oder einen VNC-Client. Die Verbindungsdaten ermittelt das `virsh`-Kommando `vncdisplay`.
- ▶ `suspend/resume <vmname>`: stoppt die angegebene virtuelle Maschine vorübergehend bzw. setzt die Ausführung wieder fort. Die gestoppte virtuelle Maschine beansprucht weiterhin RAM! Es wird also nur die virtuelle CPU angehalten.
- ▶ `shutdown/reboot <vmname>`: fährt die virtuelle Maschine herunter bzw. startet sie neu. Die virtuelle Maschine erhält via ACPI ein Shutdown-Signal. Es ist allerdings der virtuellen Maschine überlassen, ob sie auch darauf reagiert. Wenn das bei älteren Distributionen nicht der Fall ist, hilft in der Regel die Installation des Pakets `acpid` in der virtuellen Maschine.
- ▶ `save <vmname> <dateiname>`: speichert den Zustand der virtuellen Maschine in einer Datei und stoppt dann die Ausführung der Maschine.
- ▶ `restore <dateiname>`: aktiviert die zuvor gespeicherte virtuelle Maschine wieder. Die Zustandsdatei kann anschließend gelöscht werden.
- ▶ `destroy <vmname>`: beendet die virtuelle Maschine sofort. Das ist so, als würden Sie bei Ihrem Rechner das Stromkabel ausstecken, und es kann dieselben Folgen haben.
- ▶ `undefine <vmname>`: löscht die XML-Datei, die die virtuelle Maschine beschreibt. Die Image-Datei mit der virtuellen Festplatte bleibt erhalten.
- ▶ `autostart [--disable] <vmname>`: gibt an, dass die virtuelle Maschine während des Boot-Prozesses des Hostrechners automatisch gestartet werden soll. Mit der Option `--disable` wird der automatische Start wieder abgestellt. Der automatische Start funktioniert nur für Maschinen, die auf Systemebene eingerichtet werden. Auf Session-Ebene werden `autostart`-Maschinen dagegen erst gestartet, wenn mit `virsh` zum ersten Mal eine Verbindung zu `libvirtd` hergestellt wird.
- ▶ `console <vmname>`: ermöglicht die Bedienung der angegebenen virtuellen Maschine direkt in der Konsole. Das setzt voraus, dass in der virtuellen Maschine ein



getty-Prozess für die serielle Schnittstelle `/dev/ttyS0` läuft. Um die Verbindung zu beenden, drücken Sie `[Strg]+[J]`.

- ▶ `ttyconsole <vmname>`: gibt an, über welches Device des Host-Computers die serielle Schnittstelle des Gastsystems zugänglich ist (z. B. `/dev/pts/5`). Sie können nun in einem Terminalfenster `socat - /dev/pts/5` ausführen und dann mit der virtuellen Maschine kommunizieren (ganz ähnlich wie beim oben beschriebenen `console-Kommando`). Vorher muss in der Regel das Paket `socat` installiert werden.
- ▶ `vncdisplay <vmname>`: liefert die IP-Adresse (leer für `localhost`) und Portnummer für die VNC-Anzeige der virtuellen Maschine. Sie können nun einen beliebigen VNC-Client (z. B. `Vinagre`) starten, um mit der virtuellen Maschine zu interagieren. Auf dem KVM-Host können Sie stattdessen auch `virt-viewer vmname` ausführen. Aus Sicherheitsgründen funktioniert der VNC-Zugang nur von `localhost`.

`vncdisplay` liefert kein Ergebnis, wenn die virtuelle Maschine ihr Grafiksystem gar nicht über VNC freigibt, sondern stattdessen das modernere Spice-System verwendet. In diesem Fall können Sie die virtuelle Maschine mit dem Programm `virt-viewer` bedienen. An dieses Programm können Sie direkt den Namen der virtuellen Maschine übergeben.

Sollte sich dennoch die Notwendigkeit ergeben, die Spice-Port-Nummer zu ermitteln, wird es schwierig. In `virsh` fehlt ein Kommando, um ähnlich wie mit `vncdisplay` den Spice-Port einer virtuellen Maschine herauszufinden. Abhilfe schafft das folgende Kommando, das ich in einem Forum von *ubuntuusers.de* gefunden habe. Es extrahiert die Port-Nummer aus der Prozessliste:

```
spice_port=$(ps aux | grep vm_name | grep -oP "(?<=-spice port=).*?(?=,)")
```

- ▶ `edit <vmname>`: lädt die XML-Datei zur Beschreibung der virtuellen Maschine in den Editor, den Sie in der Umgebungsvariablen `$EDITOR` eingestellt haben.

## virt-clone

Das Einrichten einer neuen virtuellen Maschine nimmt normalerweise geraume Zeit in Anspruch. Wenn Sie eine virtuelle Maschine wünschen, die im Wesentlichen dieselben Eckdaten wie eine bereits vorhandene virtuelle Maschine hat, ist es wesentlich schneller, diese einfach zu kopieren bzw. zu »klonen«. Dabei hilft das Kommando `virt-clone`: Standardmäßig erzeugt es eine neue XML-Definitionsdatei, kopiert die Image-Datei für die virtuelle Festplatte und gibt dem Netzwerkadapter eine neue, zufällige MAC-Adresse. Die restlichen Hardware-Komponenten bleiben unverändert. Die virtuelle Maschine muss vor dem Kopieren heruntergefahren werden.

Das folgende Kommando kopiert eine Ubuntu-Server-Installation. Die neue virtuelle Maschine erhält den Namen `server6`, die neue Image-Datei wird in der Datei

/var/lib/libvirt/images/userver6.img gespeichert. Achten Sie darauf, die neue Image-Datei in einem libvirt-Speicherpool anzulegen: Sonst verhindern die SELinux-Regeln unter RHEL/Fedora die Ausführung der virtuellen Maschine.

```
root# virt-clone --original userver5 --name userver6 \
               --file /var/lib/libvirt/images/userver6.img
```

Nach dem Kopieren müssen Sie in der virtuellen Maschine diverse Anpassungen vornehmen. Beispielsweise müssen Sie die Netzwerkkonfiguration ändern, damit es keine IP-Adresskonflikte gibt. Je nach Konfiguration ist es erforderlich, auch die Dateien /etc/hosts und /etc/hostname zu aktualisieren. Wenn es in der ursprünglichen virtuellen Maschine einen SSH-Server gab, sollten Sie in der virtuellen Maschine unbedingt einen neuen SSH-Schlüssel erzeugen:

```
root# systemctl stop sshd                (Debian/Ubuntu)
root# rm /etc/ssh/ssh_host_*
root# dpkg-reconfigure openssh-server

root# systemctl stop sshd                (CentOS/Fedora/RHEL)
root# rm /etc/ssh/ssh_host_*
root# systemctl start sshd
```

### virt-viewer

virt-viewer aus dem gleichnamigen Paket ist ein VNC- und Spice-Client zur Darstellung des Bildschirminhalts sowie zur Kommunikation mit einer virtuellen Maschine. virt-viewer vmname stellt die Verbindung zu einer laufenden virtuellen Maschine her. Das setzt voraus, dass die virtuelle Maschine VNC oder Spice nutzt.

```
root# virt-viewer <vmname>
```

Wenn Sie virt-viewer ohne root-Rechte ausführen, aber die Verbindung zu einer auf Systemebene laufenden virtuellen Maschine herstellen möchten, geben Sie die Verbindungszeichenkette mit der Option -c an:

```
user$ virt-viewer -c qemu:///system <vmname>
```

Sofern die virtuelle Maschine VNC zur Weitergabe des virtuellen Grafiksystems nutzt, können Sie statt virt-viewer jeden beliebigen anderen VNC-Client einsetzen. Der einzige Unterschied besteht darin, dass Sie zuerst mit dem virsh-Kommando vncdisplay die Verbindungsdaten ermitteln müssen.

Um virtuelle Maschinen von Windows aus zu steuern, können Sie virt-viewer von der folgenden Seite auch als Windows-Programm herunterladen:

<https://www.spice-space.org/download.html>

## virt-top

Das Kommando `virt-top` aus dem gleichnamigen Paket liefert ähnlich wie `top` eine Auflistung aller virtuellen Maschinen. Zu jeder virtuellen Maschine werden deren Speicher- und CPU-Bedarf sowie diverse andere Parameter angezeigt:

```
user$ virt-top
virt-top 10:41:40 - x86_64 4/4CPU 1600MHz 15860MB 0,6%
6 domains, 2 active, 2 running, 0 sleeping, 0 paused, 4 inactive D:0 O:0 X:0
CPU: 25,0% Mem: 2048 MB (2048 MB von Gast)
```

| ID | S | RDRQ | WRRQ | RXBY | TXBY | %CPU | %MEM | TIME    | NAME      |
|----|---|------|------|------|------|------|------|---------|-----------|
| 15 | R | 0    | 0    | 0    | 0    | 24,8 | 6,0  | 0:07.57 | ubuntu    |
| 13 | R | 0    | 0    | 104  | 0    | 0,2  | 6,0  | 1:24.07 | centos    |
| -  |   |      |      |      |      |      |      |         | (debian)  |
| -  |   |      |      |      |      |      |      |         | (windows) |

## libvirt-Provider für Vagrant

Vagrant (siehe [Abschnitt 39.4](#)) hatte Mitte 2017 noch keine integrierte Unterstützung für libvirt. Auf GitHub gibt es aber ein Projekt für ein libvirt-Provider-Plugin:

<https://github.com/vagrant-libvirt/vagrant-libvirt>

Um das Plugin zu installieren, müssen Sie zuerst diverse Entwicklungspakete installieren, unter Fedora z.B. mit diesem Kommando:

```
root# dnf -y install qemu libvirt libvirt-devel ruby-devel gcc
```

Anschließend kompilieren Sie mit `vagrant plugin install` das libvirt-Plugin und richten dieses ein:

```
root# vagrant plugin install vagrant-libvirt
Fetching: fog-xml-0.1.2.gem (100%)
Fetching: fog-json-1.0.2.gem (100%)
Fetching: fog-libvirt-0.3.0.gem (100%)
Fetching: vagrant-libvirt-0.0.37.gem (100%)
Installed the plugin 'vagrant-libvirt (0.0.37)'
```

Unter CentOS/RHEL/Fedora kann libvirt auf Systemebene nur mit root-Rechten genutzt werden. Deswegen müssen Sie auch `vagrant` als root ausführen:

```
root# mkdir centos7-libvirt
root# cd centos7-libvirt
root# vagrant init centos/7
root# vagrant up --provider libvirt
...
==> default: -- Base box:          centos/7
==> default: -- Storage pool:      default
```

```

==> default: -- Image:          /var/lib/libvirt/images/
                                centos7-libvirt_default.img (41G)
==> default: -- Volume Cache:   default
==> default: -- Graphics Type:  vnc
==> default: -- Graphics Port:  5900
==> default: -- Graphics IP:    127.0.0.1
==> default: -- Graphics Password: Not defined
==> default: -- Video Type:     cirrus
==> default: -- Video VRAM:     9216
==> default: -- Keymap:         en-us
...
root# vagrant ssh
[vagrant@localhost ~]$ cat /etc/os-release
NAME="CentOS Linux"
VERSION="7 (Core)"
...

```

## 40.5 Integration der virtuellen Maschinen in das LAN (Netzwerkbrücke)

Standardmäßig verwendet libvirt ein sogenanntes Usermode-Networking. Den virtuellen Maschinen wird dabei via DHCP eine IP-Adresse im Bereich 192.168.122.\* zugewiesen. Auf dem Hostrechner richten Sie die IP-Adresse 192.168.122.1 als Gateway ins Internet ein. Die Gäste haben also Internetzugang und können zudem mit dem Host über dessen Adresse 192.168.122.1 kommunizieren. Davon abgesehen, können die KVM-Gäste aber nicht auf Netzwerkdienste im lokalen Netzwerk zugreifen, und umgekehrt können auch die Rechner im LAN nicht mit KVM-Gästen kommunizieren.

Damit Sie auf KVM-Gästen Server-Dienste für das lokale Netzwerk anbieten können, brauchen Sie eine virtuelle Netzwerkbrücke (*Bridge*), die die virtuellen Netzwerkadapter der KVM-Maschinen mit dem physikalischen Netzwerkadapter des Hostrechners verbindet.

### Konfiguration der Netzwerkbrücke auf dem Host-Rechner

Um die Brücke zu bauen, verwenden Sie die Werkzeuge aus dem Paket *bridge-utils*. Die Konfigurationsdetails variieren aber wie üblich von Distribution zu Distribution. Ich zeige Ihnen hier die Vorgehensweise für Fedora und Ubuntu. Beachten Sie bei beiden Varianten, dass Sie *eth0* durch den bei Ihnen gültigen Schnittstellennamen ersetzen!

**Fedora** Unter Fedora müssen Sie die vorhandene Datei *ifcfg-eth0* modifizieren und eine neue Datei *ifcfg-br0* hinzufügen.

Aus `ifcfg-eth0` müssen alle Parameter zur IP-Konfiguration entfernt werden. Neu ist dafür `BRIDGE=br0`. Damit wird die Schnittstelle mit der Brücke `br0` verbunden.

```
# Datei /etc/sysconfig/network-scripts/ifcfg-eth0
# HWADDR muss die MAC-Adresse der Netzwerkschnittstelle enthalten
HWADDR=nn:nn:nn:nn:nn:nn
DEVICE=eth0
ONBOOT=yes
NM_CONTROLLED=no
BRIDGE=br0
```

Bei der Konfiguration der Brücke gehe ich davon aus, dass die IP-Konfiguration via DHCP erfolgen soll. Für eine statische Konfiguration verwenden Sie in `ifcfg-br0` die Einstellung `BOOTPROTO=static` und fügen Zeilen zur Konfiguration von `IPADDR`, `NETMASK` und `GATEWAY` hinzu. Vergessen Sie nicht, dass Sie dann auch `/etc/resolv.conf` manuell einrichten müssen!

```
# Datei /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=br0
TYPE=Bridge
ONBOOT=yes
BOOTPROTO=dhcp
NM_CONTROLLED=no
DELAY=0
```

Damit die neue Konfiguration wirksam wird, deaktivieren Sie den NetworkManager und aktivieren stattdessen die Scripts zur herkömmlichen Netzwerkkonfiguration:

```
root# systemctl disable NetworkManager
root# systemctl stop NetworkManager
root# rm /etc/resolv.conf
root# ... (gegebenenfalls resolv.conf manuell konfigurieren)
root# systemctl enable network
root# systemctl start network
```

Auch unter Ubuntu müssen Sie sicherstellen, dass der NetworkManager deaktiviert ist. Die Konfiguration erfolgt durch die Datei `/etc/network/interfaces`. Die dort vorhandenen Zeilen zur manuellen Konfiguration der Schnittstelle zum LAN (in diesem Beispiel also `eth1`) müssen dahingehend geändert werden, dass diese Schnittstelle nun manuell gesteuert werden kann. Dafür wandern die entsprechenden Konfigurationseinstellungen nun in die Beschreibung des Interfaces `br0` (oder wie auch immer Sie die Brücke nennen). In diesem Beispiel ist `10.0.0.138` das Gateway und der DHCP-Server des lokalen Netzwerks. Der Bridge selbst wird die IP-Adresse zugewiesen, die bisher der Hostrechner innehatte (`10.0.0.120`). Vergessen Sie nicht, dass `/etc/resolv.conf` die Adresse des Nameservers enthalten muss! Ubuntu

```

# Datei /etc/interfaces/network (Ubuntu)
# Loopback-Netzwerkschnittstelle (unverändert)
auto lo
iface lo inet loopback

# Schnittstelle zum LAN (manuell)
auto eth1
iface eth1 inet manual

# Brücke zu eth1
auto br0
iface br0 inet static
    address 10.0.0.120
    network 10.0.0.0
    netmask 255.255.255.0
    broadcast 10.0.0.255
    gateway 10.0.0.138
    bridge_ports eth1

```

Mit `ifdown/ifup` starten Sie die betroffenen Netzwerkschnittstellen neu. Die Brücke `br0` hat nun die IP-Adresse `10.0.0.120` und überträgt die IP-Pakete an den physikalischen Netzwerkadapter `eth1`. Falls die Brücke ihre IP-Adresse via DHCP beziehen soll, vereinfacht sich die Konfiguration der Schnittstelle `br0`:

```

# Datei /etc/interfaces/network
# Loopback-Netzwerkschnittstelle (unverändert)
auto lo
iface lo inet loopback
# Schnittstelle zum LAN (manuell)
auto eth1
iface eth1 inet manual
# Brücke zu eth1
auto br0
iface br0 inet dhcp
    bridge_ports eth1

```

### Netzwerkkonfiguration auf einem Root-Server

Deutlich komplizierter wird die Konfiguration des Netzwerks, wenn Sie auf einem Root-Server mehrere Gäste öffentlich im Internet zugänglich machen möchten. Die Aufgabenstellung lautet dann, den gesamten Verkehr für mehrere IP-Adressen durch den Host korrekt zu den Gästen zu leiten. Die konkrete Vorgehensweise ist stark von den technischen Gegebenheiten des Hosting-Providers abhängig. Zwei Anleitungen, die sich auf die von der deutschen Firma Hetzner genutzte Infrastruktur beziehen, finden Sie hier:

<https://kofler.info/ubuntu-kvm-host-konfigurieren-root-server>

[https://wiki.hetzner.de/index.php/KVM\\_mit\\_Nutzung\\_aller\\_IPs\\_-\\_the\\_easy\\_way](https://wiki.hetzner.de/index.php/KVM_mit_Nutzung_aller_IPs_-_the_easy_way)

## Konfiguration der virtuellen Maschine

Beim Einrichten der virtuellen Maschine geben Sie nun den Namen der Brücke als Netzwerkquelle an (siehe [Abbildung 40.5](#)). Virtuelle Maschinen können Netzwerkbrücken nur verwenden, wenn sie auf Systemebene ausgeführt werden. Der Grund: Die Netzwerkkommunikation zwischen dem Hostrechner und dem KVM-Gast erfolgt durch sogenannte TUN/TAP-Devices. Dabei handelt es sich um vom Kernel simulierte Netzwerkschnittstellen, die bei jedem Start der virtuellen Maschine eingerichtet werden. Die libvirt-Werkzeuge kümmern sich zum Glück um alle Details, können ihre Arbeit aber nur mit `root`-Rechten verrichten.

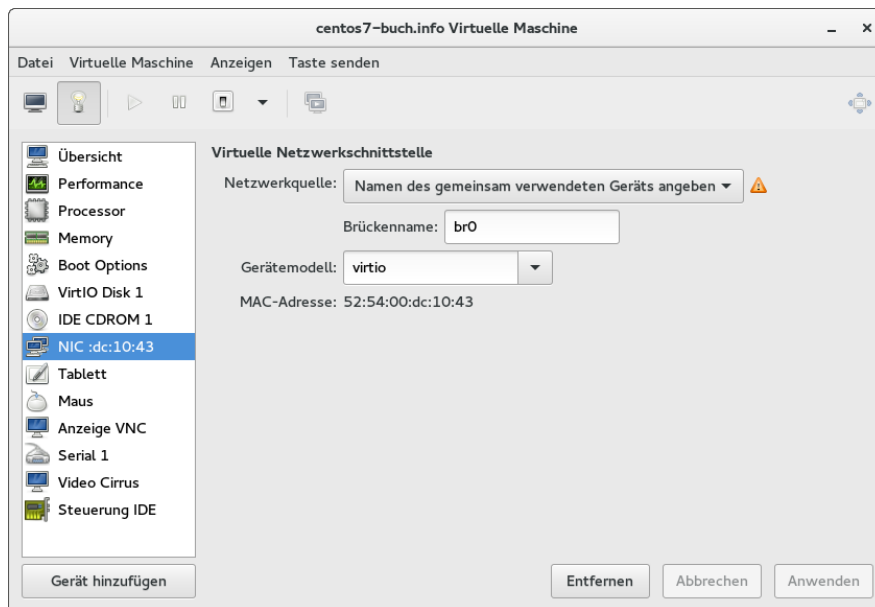


Abbildung 40.5 Konfiguration der Netzwerkschnittstelle als Brücke

## 40.6 Direkter Zugriff auf den Inhalt einer Image-Datei

In diesem Abschnitt geht es um die Frage, wie Sie den Inhalt eines virtuellen Datenträgers auslesen oder verändern können, ohne die virtuelle Maschine selbst zu starten. Das ist beispielsweise praktisch, um von außen Reparaturen durchzuführen oder um Konfigurationsdateien durch ein Script zu verändern.

Für den Zugriff auf den virtuellen Datenträger gibt es verschiedene Vorgehensweisen:

- ▶ Sie können den Inhalt der virtuellen Datenträger direkt im Hostsystem lesen oder verändern,

- ▶ Sie können auf diverse libguestfs-Werkzeuge zurückgreifen oder
- ▶ Sie können ein Linux-Live-System zu Hilfe nehmen, also die virtuelle Maschine vom ISO-Image einer Linux-Distribution starten und von dort aus auf die virtuellen Festplatten zugreifen.

In diesem Abschnitt gehe ich nur auf die beiden ersten Varianten ein. Manche der hier vorgestellten Werkzeuge können ausschließlich RAW-Images bearbeiten. Gegebenenfalls müssen Sie eine in einem anderen Format vorliegende Image-Datei in dieses Format umwandeln:

```
user$ qemu-img convert -f qcow2 image.qcow2 -O raw image.raw
```

Mit dem folgenden Kommando erzeugen Sie aus einem Logical Volume oder einer Festplattenpartition ein äquivalentes RAW-Image:

```
root# dd if=/dev/mapper/vg1-lv2 of=copy.raw bs=64M
312+1 Datensätze ein
312+1 Datensätze aus
20971520000 Bytes (21 GB) kopiert, 133,462 s, 157 MB/s
```

Ein Schreibzugriff auf einen virtuellen Datenträger ist nur zulässig, wenn die virtuelle Maschine vollkommen heruntergefahren ist! Andernfalls riskieren Sie ein kaputtes Dateisystem!

### Zugriff auf partitionierte RAW-Images im Hostsystem

- kpartx** Mit dem Kommando `kpartx` aus dem gleichnamigen Paket verbinden Sie alle in einer RAW-Datei enthaltenen Partitionen mit Loop-Devices:

```
root# kpartx -av image.raw
add map loop0p1 (252:12): 0 1024000 linear /dev/loop0 2048
add map loop0p2 (252:13): 0 19945472 linear /dev/loop0 1026048
```

Bei meinen Tests hat dies auch funktioniert, wenn die Image-Datei eine GUID Partition Table enthält (also keine traditionelle Partitionstabelle im Master Boot Record). Sofern es sich um normale Partitionen handelt, können Sie diese nun direkt mit `mount` in das Dateisystem einbinden:

```
root# mkdir /repair1
root# mount /dev/mapper/loop0p1 /repair1
```

- LVM** Wenn Sie innerhalb der virtuellen Maschine LVM konfiguriert haben, stehen die resultierenden Physical und Logical Volumes sowie Volume Groups direkt zur Verfügung. Listen der LVM-Elemente liefern `lvscan`, `pvscan` und `vgscan`. Der Zugriff auf die LVs setzt voraus, dass auf dem Hostsystem die LVM-Werkzeuge installiert sind.



```

root# lvscan
ACTIVE          '/dev/VolGroup/lv_root' [7,56 GiB] inherit
ACTIVE          '/dev/VolGroup/lv_swap' [1,94 GiB] inherit
...
root# mkdir /repair2
root# mount /dev/VolGroup/lv_root /repair2

```

Nun können Sie über die Verzeichnisse `repairN` auf die Dateisysteme der virtuellen KVM-Festplatte zugreifen. Wenn Sie damit fertig sind, müssen Sie aufräumen:

```

root# umount /repair1
root# umount /repair2
root# kpartx -dv image.raw

```

### libguestfs-Werkzeuge

Anstatt die Image-Datei selbst zu analysieren und die relevanten Partitionen in das lokale Dateisystem zu integrieren, können Sie diese Aufgaben diversen Werkzeugen überlassen, die auf der `libguestfs`-Bibliothek aufbauen.

Die Bibliothek `libguestfs` erlaubt den direkten Zugriff auf die Dateisysteme, die sich innerhalb einer Image-Datei befinden. `libguestfs` unterstützt alle erdenklichen Image-Formate, Partitionen, LVM sowie alle gängigen Dateisysteme. Die Bedienung der `libguestfs`-Werkzeuge ist auf der folgenden Website sowie auf der `man`-Seite `libguestfs` umfassend dokumentiert:

<http://libguestfs.org>

Aktuelle CentOS-, Debian-, Fedora-, RHEL- und Ubuntu-Distributionen liefern alle erforderlichen Pakete gleich mit:

**Installation**

```

root# apt/dnf/yum install libguestfs-tools libguestfs-tools-c

```

Das Paket enthält diverse Kommandos zur Manipulation von Image-Dateien. Die zu bearbeitende oder zu analysierende Image-Datei geben Sie in der Regel mit `-a image-datei` an. Alternativ können Sie mit `-d vmname` den `libvirt`-Namen der virtuellen Maschine angeben.

**virt-df**

`virt-df` gibt einen raschen Überblick über die Auslastung aller Dateisysteme aller virtuellen Maschinen, die den `libvirt`-Werkzeugen bekannt sind. Das Kommando kommt auch mit Logical Volumes innerhalb von virtuellen Datenträgern zurecht.

```

root# virt-df
Filesystem                1K-blocks      Used Available Use%
centos-mini:/dev/sda1      495844         52091   418153   11%
centos-mini:/dev/sdb1      20157836       253468  18880396   2%
centos-mini:/dev/vg_centosmini/lv_root

```

```

                                9571132   1051104   8033836   11%
centos64-vm2:/dev/sdb           4319464   4319464     0 100%
centos64-vm2:/dev/sda1         495844    32918    437326    7%
centos64-vm2:/dev/vg_vm2/lv_root 7539088   2369932   4786180   32%
...

```

Wenn Sie nur an den Ergebnissen einer einzelnen virtuellen Maschine interessiert sind, geben Sie den Dateinamen der Image-Datei mit `-a` oder den libvirt-Namen der virtuellen Maschine mit `-d` an.

**virt-filesystems** `virt-filesystems` verrät, welche Dateisysteme sich in einer Image-Datei befinden. Mit den Optionen `--all`, `--long` und `--uuid` listet das Kommando auch LVM- und Swap-Partitionen auf und gibt die UUID-Nummern der Dateisysteme an:

```

root# virt-filesystems -a vm2.img --all --long --uuid
Name          Type      VFS Label Size      Parent  UUID
/dev/sda1     filesystem ext4 -   524288000 -       7a95...
/dev/vg_vm2/lv_root filesystem ext4 -   7843348480 -       e69f...
/dev/vg_vm2/lv_swap filesystem swap -   2113929216 -       e0f4...
...

```

**virt-inspector** `virt-inspector` wirft einen Blick in die Dateisysteme einer Image-Datei und ermittelt, welche Distribution darin installiert ist, welche Partitionen es gibt, welche Kernelversion und welche Pakete installiert sind etc. `virt-inspector` kennt sowohl das RPM- als auch das Debian-Paketsystem. Das Kommando liefert als Ausgabe ein schier endloses XML-Dokument, das eigentlich zur maschinellen Weiterverarbeitung gedacht ist.

```

root# virt-inspector disk.img | less
<?xml version="1.0"?>
<operatingsystems>
  <operatingsystem>
    <root>/dev/cl/root</root>
    <name>linux</name>
    <arch>x86_64</arch>
    <distro>centos</distro>
    <product_name>CentOS Linux release 7.3.1611 (Core) </product_name>
    ...
  <mountpoints>
    <mountpoint dev="/dev/cl/root"/></mountpoint>
    <mountpoint dev="/dev/sda1"/>/boot</mountpoint>
  </mountpoints>
  <filesystems>
    <filesystem dev="/dev/cl/root">
      <type>xfs</type>
      <uuid>271158c8-8134-4082-9b74-f2413259ae4b</uuid>
    </filesystem>
  ...

```

`virt-cat` gibt eine Datei einer virtuellen Maschine aus. Dabei können Sie wahlweise mit `-a` den Namen eines Images oder mit `-d` den (Domain-)Namen der virtuellen Maschine angeben: `virt-cat`

```
root# virt-cat -d centos-server-7 /etc/fstab
/dev/mapper/cl-root          /      xfs  defaults 0 0
UUID=0112eac4-e995-4dfc-b6ca-4a5fc3b10d7b /boot  xfs  defaults 0 0
/dev/mapper/cl-swap         swap   swap defaults 0 0
...
```

Um die Datei in einem Editor zu verändern, verwenden Sie anstelle von `virt-cat` das Kommando `virt-edit`. Es darf nur für ausgeschaltete virtuelle Maschinen verwendet werden! `virt-edit` berücksichtigt bei der Wahl des Editors die Umgebungsvariable `EDITOR`. Standardmäßig wird der Editor `vi` ausgeführt.

```
root# virt-edit -d centos-server-7 /etc/fstab
```

Mit `virt-tar-out` lesen Sie ein Verzeichnis aus dem Dateisystem einer Image-Datei (`-a`) oder einer virtuellen Maschine (`-d`) aus und schreiben es in ein `tar`-Archiv: `virt-tar-in` und  
`virt-tar-out`

```
root# virt-tar-out -d centos-server-7 /etc etc.tar
```

Wenn Sie das Archiv auch gleich komprimieren möchten, müssen Sie die Ausgabe von `virt-tar-out` mit einer Pipe an `gzip` weiterleiten:

```
root# virt-tar-out -d centos-server-7 /etc - | gzip > etc.tar.gz
```

Umgekehrt packt `virt-tar-in` ein Archiv im Image aus. Das Zielverzeichnis muss bereits existieren. Die virtuelle Maschine muss dazu vorher gestoppt werden!

```
root# virt-tar-in -d centos-server-7 data.tar /home/user01
```

Bei komprimierten Archiven gehen Sie so vor:

```
root# zcat data.tar.gz | virt-tar-in -d centos-server-7 - /home/user01
```

`virt-make-fs` erzeugt eine neue Image-Datei und speichert darin den Inhalt eines Verzeichnisses oder eines `tar`-Archivs, das wahlweise auch komprimiert sein darf: `virt-make-fs`

```
root# virt-make-fs mydata.tar.gz new-disk.img
```

Das Kommando erzeugt standardmäßig ein RAW-Image ohne Partitionstabelle mit einem `ext2`-Dateisystem. Durch Optionen können Sie ein anderes Image- oder Dateisystemformat einstellen, MBR- oder GPT-Partitionierung vorsehen, die Größe der Image-Datei steuern etc.

`virt-resize` kopiert ein Disk-Image und verändert dabei die Größe der darin enthaltenen Partitionen und Dateisysteme. Das Ziel-Image muss vorher erzeugt werden. `virt-resize`

Wenn die zu verändernde Partition als Physical Volume für LVM verwendet wird, vergrößert `virt-resize` das PV mit `pvresize`. Alle weiteren LVM-Kommandos müssen Sie später im laufenden System selbst durchführen. Vor der Ausführung von `virt-resize` müssen Sie die zugrunde liegende virtuelle Maschine ausgeschaltet!

Im folgenden Beispiel analysieren die ersten beiden Kommandos das Quell-Image. Das dritte Kommando richtet das vergrößerte Ziel-Image ein. `virt-resize` überträgt dann die Daten vom Quell- in das Ziel-Image.

```
root# qemu-img info centos7.qcow2
file format: qcow2
virtual size: 10G (10737418240 bytes)
...
root# virt-filesystems --partitions --long -a centos7.qcow2
Name      Type      MBR  Size      Parent
/dev/sda1 partition 83   524288000 /dev/sda
/dev/sda2 partition 8e   10212081664 /dev/sda
root# qemu-img create -f qcow2 centos7-copy.qcow2 15G
root# virt-resize --expand /dev/sda2 centos7.qcow2 centos7-copy.qcow2
Summary of changes:
  /dev/sda1: This partition will be left alone.
  /dev/sda2: This partition will be resized from 9,5G to 14,5G.
             The LVM PV on /dev/sda2 will be expanded using
             the 'pvresize' method.
```

# Kapitel 41

## Docker

Das Open-Source-Programm Docker ist das aktuell populärste System zur Ausführung von Containern. Container sind in einer isolierten Umgebung ausgeführte Prozesse. Docker-Container basieren wiederum auf Images. Ein Image ist die Verpackungsform einer Anwendung, die häufig in eine minimale Linux-Distribution eingebettet ist. Derartige Anwendungen können von einer simplen Shell-Umgebung über einen Webserver bis hin zu einem kompletten Content-Management-System reichen. Sie können also z. B. WordPress in einem Docker-Container ausführen.

Docker hilft dabei, fertige Applikationen unkompliziert und reproduzierbar auf unterschiedlichen Rechnern, NAS-Geräten oder in der Cloud zum Laufen zu bringen. In der Praxis wird Docker sehr häufig für Test- und Entwicklungsaufgaben eingesetzt. Grundsätzlich können Sie mit Docker so wie in virtuellen Maschinen auch Server-Dienste für den Produktionseinsatz zur Verfügung stellen. Dabei kommen die Stärken von Docker aber nur bedingt zum Tragen und gleichzeitig erweisen sich die langfristige Wartung eines Containers durch Updates sowie die Einhaltung hoher Sicherheitsstandards als relativ schwierig.

Im Gegensatz zu KVM, VirtualBox oder anderen Virtualisierungssystemen verzichtet Docker darauf, für jeden Container einen kompletten PC zu simulieren. Stattdessen teilt Docker möglichst viele Ressourcen zwischen Host und Container. Das hat den Vorteil, dass Docker-Container viel leichtfüßiger sind als virtuelle Maschinen.

Andererseits sind Docker-Container weniger stark als virtuelle Maschinen voneinander getrennt. Plattformüberschreitende Container sind mit Einschränkungen verbunden (Linux-Container auf einem Windows-Host) bzw. ganz unmöglich (Windows-Container auf einem Linux-Host). Vergleichsweise gut funktioniert die Ausführung von Linux-Containern unter macOS. Docker ist zur Ausführung von Server-Applikationen gedacht, aber nicht zur Darstellung und Bedienung grafischer Benutzeroberflächen.

Dieses Kapitel gibt eine Einführung in die Funktionsweise und Anwendung der *Docker Community Edition* (Docker CE), wobei ich mich auf den gängigsten Fall konzentriere: auf die Ausführung von auf Linux basierenden Containern auf einem Linux-Host.

Links Umfassende Dokumentation zu Docker finden Sie auf der Firmenwebseite:

<https://www.docker.com>

<https://docs.docker.com>

Eine ausgezeichnete Einführung zu Docker in Form eines ca. 400-seitigen E-Books können Sie hier erwerben:

<https://dockerbook.com>

## 41.1 Grundlagen, Nomenklatur und Installation

**Client/Server-Modell** Docker ist als Client/Server-Modell realisiert. Im Hintergrund läuft der Docker-Dämon (`dockerd`). Er ist für die Ausführung der Container verantwortlich. Auf den Clients verwenden Sie das Kommando `docker`, um den Dämon zu steuern. Häufig laufen `dockerd` und `docker` auf demselben Host; es ist aber auch möglich, mit `docker` eine Netzwerkverbindung zu einem externen Docker-Dämon herzustellen.

**Images und Container** Der Ausgangspunkt für die Ausführung jedes Containers ist ein Image. Den Begriff kennen Sie ja schon von virtuellen Maschinen – dort ist ein Disk-Image aus Host-Sicht eine Datei, die dem Gast mit Treibern als Festplatte präsentiert wird. Das Gastsystem richtet dort Partitionen und Dateisysteme ein.

Bei Docker stellt ein Image hingegen als Read-only-Dateisystem die Basis für den Container zur Verfügung. Im Unterschied zu virtuellen Maschinen wird das Image durch den laufenden Container nie verändert. Alle vom Container veränderten oder hinzugefügten Dateien landen stattdessen in einem getrennten Overlay-Dateisystem, wobei dieses auf dem Host durch ein Verzeichnis innerhalb von `/var/lib/docker` abgebildet wird. Diese Trennung zwischen unveränderlichen Image-Dateien und veränderlichen Container-Dateien macht es möglich, von einem Image beliebig viele Container abzuleiten, die durchaus auch zugleich ausgeführt werden können.

**Virtuelle Maschinen versus Container** Auch wenn virtuelle Maschinen und Container ähnliche Ziele verfolgen, z. B. Testprogramme oder Server-Anwendungen auf einem physikalischen Rechner möglichst isoliert voneinander auszuführen, so unterscheidet sich die Funktionsweise doch fundamental.

Bei jeder virtuellen Maschine wird die gesamte Hardware eines PCs durch Software nachgebildet. Die virtuelle Maschine hat den Eindruck, auf einem echten Rechner zu laufen. Das erfordert aber einen riesigen Ressourcenaufwand (RAM, Speicherplatz auf dem Datenträger, CPU-Zyklen).

Container nutzen dagegen direkt die Infrastruktur des Host-Systems, also insbesondere dessen Dateisystem und Kernel. Docker-Container laufen gewissermaßen wie

isolierte Prozesse (ähnlich wie »Sandboxes«) direkt im Host-System. Der Platzbedarf von Containern ist in der Regel viel kleiner als der von virtuellen Maschinen, weil der Unterbau auf ein Minimum reduziert werden kann. Zudem starten Container deutlich schneller.

Der Vorteil von virtuellen Maschinen besteht darin, dass diese besser und sicherer voneinander getrennt sind als Docker-Anwendungen. Docker kann in dieser Hinsicht nur bedingt mithalten, da insbesondere der Kernel als zentrale gemeinsame Ressource eine vollständige Isolierung unmöglich macht.

Für Docker spricht andererseits der stark reduzierte Overhead. Zudem lassen sich Docker-Anwendungen sekundenschnell script-gesteuert erzeugen und konfigurieren. Wenn es darum geht, Anwendungen automatisiert einzurichten, sei es für Testzwecke oder zur Skalierung, bietet Docker riesige Vorteile gegenüber den vergleichsweise schwerfälligen virtuellen Maschinen. Während es bei virtuellen Maschinen meist zweckmäßig ist, mehrere zusammengehörige Dienste in einer virtuellen Maschine zusammenzufassen, kann bei Docker jede Funktion ihren eigenen Container bekommen. In der Docker-Nomenklatur spricht man in diesem Zusammenhang von der »Micro Services Architecture«.

Unter Linux schließen sich Virtualisierung und Container keineswegs aus. In der Praxis kommt es oft vor, dass die Docker-Engine selbst wieder in einer virtuellen Maschine läuft.

## Docker-Varianten

Dieses Buch behandelt nur die kostenlos verfügbare Community Edition von Docker (Docker CE). Parallel dazu bietet die Firma auch Enterprise-Varianten an (Docker EE) mit diversen Zusatzfunktionen an.

Community  
versus Enterprise  
Edition

Auf der folgenden Webseite finden Sie eine Tabelle, die angibt, für welche Distribution welche Docker-Variante offiziell unterstützt wird. Beispielsweise können Sie unter CentOS und Ubuntu sowohl Docker CE als auch Docker EE verwenden, unter RHEL aber nur Docker EE, unter Fedora und Debian hingegen nur Docker CE.

<https://docs.docker.com/engine/installation/>

Docker steht auch für Windows und macOS zur Verfügung. Verblüffend mühelos funktioniert die Ausführung von Linux-Containern unter macOS. Der Linux-Kernel wird dabei durch das Hypervisor-Framework ausgeführt, das in aktuellen macOS-Versionen enthalten ist.

Docker unter  
macOS und  
Windows

Die Windows-Version von Docker setzt Windows 10 oder Windows Server 2016 sowie Hyper-V voraus. Windows 10 Home scheidet aus, weil in dieser Windows-

Variante Hyper-V fehlt. Der Parallelbetrieb von Docker und VirtualBox ist unmöglich, weil VirtualBox nur läuft, wenn Hyper-V deaktiviert ist. Das explizite Aktivieren/Deaktivieren von Hyper-V erfordert jeweils einen Reboot.

## Installation

Bevor es losgehen kann, müssen Sie Docker installieren. Soweit die Distributionen überhaupt Docker-Pakete zur Verfügung stellen, sind diese selten aktuell. Deswegen ist es in der Regel am besten, wenn Sie den folgenden Installationsanweisungen folgen. Dabei wird jeweils eine eigene Paketquelle eingerichtet, damit Docker anschließend im Rahmen der normalen Updates aktualisiert werden kann.

**Installation unter CentOS** Mit diesen Kommandos installieren Sie Docker unter CentOS und stellen sicher, dass Docker sofort und in Zukunft nach jedem Reboot gestartet wird:

```
root# yum install -y yum-utils
root# yum-config-manager --add-repo \
    https://download.docker.com/linux/centos/docker-ce.repo
root# yum makecache fast
root# yum install docker-ce
root# systemctl enable docker
root# systemctl start docker
```

**Installation unter Fedora** Ganz ähnlich sehen die Kommandos unter Fedora aus:

```
root# dnf -y install dnf-plugins-core
root# dnf config-manager \
    --add-repo https://download.docker.com/linux/fedora/docker-ce.repo
root# dnf makecache fast
root# dnf -y install docker-ce
root# systemctl enable docker
root# systemctl start docker
```

**Installation unter Ubuntu** Unter Ubuntu ist das Einrichten der Paketquelle ein wenig umständlicher. Dafür entfallen die `systemctl`-Kommandos, weil der Docker-Dämon unter Ubuntu automatisch ausgeführt wird.

```
root# apt -y install apt-transport-https ca-certificates curl
root# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | apt-key add -
root# sudo add-apt-repository \
    "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) stable"
root# apt update && apt -y install docker-ce
```



Falls Sie unter Ubuntu den Firewall UFW verwenden, können Docker-Container keine Netzwerkbrücke nutzen. Schuld ist die Defaulteinstellung von UFW, die die Weiterleitung von IP-Paketen unterbindet. Abhilfe: Führen Sie die folgende Änderung in `/etc/default/uw` durch, und starten Sie die Firewall dann mit `uw reload neu`.

```
# Datei /etc/default/uw
...
DEFAULT_FORWARD_POLICY="ACCEPT"
```

Installationsanleitungen für andere Linux-Distributionen, für macOS und Windows sowie für die Cloud (AWS und Azure) finden Sie hier:

Andere  
Betriebssysteme

<https://store.docker.com/search?offering=community&type=edition>

## 41.2 Docker kennenlernen

Auf den nächsten Seiten stelle ich Ihnen die wichtigsten Konzepte von Docker anhand von fünf Beispielen vor. Eine systematischere Beschreibung des `docker`-Kommandos folgt dann in [Abschnitt 41.3](#).

### Beispiel 1: Hello World!

Um Docker auszuprobieren, sollten Sie den Hello-World-Container verwenden. Dazu führen Sie in einem Terminal `docker run hello-world` aus. Dieses Kommando sieht zuerst nach, ob das Image `hello-world` auf dem lokalen Rechner bereits zur Verfügung steht. Ist das nicht der Fall, wird es vom Docker Hub (<https://hub.docker.com>) heruntergeladen und lokal gespeichert. Docker bildet daraus einen Container und führt diesen schließlich aus.

```
root# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
78445dd45222: Pull complete
Digest: sha256:c5515758d4c5e1e838e9cd307f6c6a0d620b5e07e6f927b07d05f6d12a1ac8d7
Status: Downloaded newer image for hello-world:latest
```

```
Hello from Docker!
This message shows that your installation appears to be working correctly.
```

Ab dem zweiten Mal wird `docker run hello-world` blitzschnell ausgeführt, weil jetzt ein lokaler Container der Hello-World-App zur Verfügung steht.

Das offizielle Hello-World-Image ist insofern ein untypisches Docker-Beispiel, als die darin verpackte App einmal ausgeführt wird und dann sofort endet. Viele Docker-

Container enthalten dagegen Programme, die Eingaben verarbeiten, dauerhaft laufen oder interaktiv genutzt werden können.

**Statusinformationen** `docker ps -a` liefert eine Liste aller laufenden bzw. in der Vergangenheit ausgeführten Docker-Container. `docker images` listet die lokal installierten Images auf:

```
root# docker ps -a
CONTAINER ID   IMAGE          COMMAND        ...  STATUS  PORTS   NAMES
f5269c759b7d  hello-world   "/hello"      ...  Exited  ...     nervous_lamarr
root# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest   48b5124b2768  7 weeks ago   1.84 kB
```

## Beispiel 2: Base Images verwenden

Um Docker besser verstehen zu können, müssen Sie ein »richtiges« Image installieren und dessen Container starten. Gut geeignet sind dazu die offiziellen »Base Images«, die Docker für einige gängige Linux-Distributionen anbietet, z. B. für CentOS, Debian, Fedora, openSUSE und Ubuntu.

Die Bezeichnung Base Image rührt daher, dass diese Images als Startpunkt für eigene Entwicklungen gedacht sind. Wenn Sie also z. B. einen Container für einen Webserver zusammensetzen möchten, beginnen Sie mit dem Base Image Ihrer Wunschdistribution und installieren darin Apache. Aus Docker-Sicht ist dann nicht ein vollkommen neues Image erforderlich, sondern das Base Image kann um ein (vergleichsweise kleines) Image mit Ihren Erweiterungen ergänzt werden.

Für die folgenden Beispiele verwende ich das Ubuntu-Image als Ausgangspunkt. Dabei spielt es keine Rolle, welche Distribution Sie für den Docker-Host verwenden. Sie können also Container des Ubuntu-Images auch unter Fedora, CentOS, ja sogar unter macOS ausführen.

Vom Ubuntu-Image gibt es mehrere Versionen. Wenn Sie einfach `docker run -it ubuntu` ausführen, bekommen Sie die Version, die Docker mit dem Attribut `latest` ausgestattet hat. In der Regel ist das die gerade aktuelle LTS-Version. Wenn Sie davon abweichend eine andere Version wünschen, müssen Sie diese explizit angeben, also z. B. `docker run -it ubuntu:17.10`. Informationen zur Konfiguration dieses Docker-Images können Sie hier nachlesen:

[https://hub.docker.com/\\_/ubuntu](https://hub.docker.com/_/ubuntu)

Die Option `-it` beschreibt das `run`-Kommando näher: `-i` bedeutet, dass der Container interaktiv ausgeführt und mit der Standardeingabe verbunden werden soll. Wegen `-t` verwendet Docker einen Pseudo-Terminalemulator (Pseudo-TTY) und verbindet

diesen mit der Standardeingabe. Die Kombination dieser beiden Optionen ist erforderlich, wenn ein Container über eine Shell interaktiv bedient werden soll.

Mit dem folgenden Kommando wird also das aktuelle Ubuntu-LTS-Image heruntergeladen, ein darauf aufbauender Container erzeugt und schließlich ausgeführt. Sie landen dabei automatisch in einer `bash`-Instanz, in der Sie Kommandos mit `root`-Rechten ausführen können. Beachten Sie, dass im Container – ganz im Gegensatz zu einer gewöhnlichen virtuellen Maschine – keinerlei andere Prozesse laufen: kein SSH-Server, kein Cron-Dämon, kein `systemd`-Dämon, kein Logging-Dienst, nichts!

```
root# docker run -it ubuntu
root@f8fec4640176:/# cat /etc/os-release
NAME="Ubuntu"
VERSION="16.04.2 LTS (Xenial Xerus)"
root@f8fec4640176:/# df -h | grep -v tmpfs
Filesystem      Size  Used Avail Use% Mounted on
overlay          20G   11G   7.6G  59% /
/dev/sda5        20G   11G   7.6G  59% /etc/hosts
shm              64M    0    64M   0% /dev/shm
root@f8fec4640176:/# ps ax
  PID TTY          STAT       TIME COMMAND
    1 ?            Ss         0:00 /bin/bash
   11 ?            R+         0:00 ps ax
root@f8fec4640176:/# exit
```

Der Ubuntu-Container erhält beim Start einen zufälligen Hostnamen. Die Größe des Dateisystems innerhalb des Containers hängt davon ab, wie viel freier Speicherplatz auf dem Host-Rechner in dem Dateisystem ist, in dem sich das Verzeichnis `/var/lib/docker` befindet. Die Ausführung des Containers endet, wenn Sie die Shell mit `exit` oder `[Strg]+[D]` beenden.

### Image-Namen

Docker-Image-Namen setzen sich aus drei Teilen zusammen: `quelle/imagename:tag`.

Dabei ist `quelle` der Name der Person oder Organisation, die das Image zusammengestellt und in den Docker Hub hochgeladen hat. Entfällt die Angabe der Quelle, nimmt das `docker`-Kommando an, dass Sie eines der offiziellen Images der Firma Docker meinen.

`imagename` gibt erwartungsgemäß den Namen des Images an.

Sofern es von dem Image mehrere Versionen gibt, wählt die optionale Angabe von `tag` explizit das mit diesem Tag gekennzeichnete Image aus. Entfällt die Versionsangabe, entscheidet sich `docker` für das mit `latest` gekennzeichnete Image.

**Netzwerk-  
anbindung**

Innerhalb des Docker-Containers steht Ihnen eine Internetverbindung zur Verfügung. Sie können sich davon mit einem `ping`-Kommando überzeugen und mit `hostname -I` die IP-Adresse ermitteln:

```
root# hostname -I
172.17.0.2
```

`ip addr` funktioniert allerdings nicht, weil das `ip`-Kommando aus Platzgründen nicht im Image installiert ist. Das lässt sich aber schnell ändern:

```
root# apt update && apt install iproute2
root# ip addr
1: lo: ...
    inet 127.0.0.1/8 scope host lo
41: eth0@if42: ...
    inet 172.17.0.2/16 scope global eth0
```

`ip addr` zeigt, dass die Netzwerkanbindung in ein privates Netzwerk führt, das Docker standardmäßig zur Verfügung stellt. Docker weist den Containern beim Start eindeutige IP-Adressen in diesem Netzwerk zu. Anders als bei virtuellen Maschinen kommt dabei allerdings nicht DHCP zum Einsatz.

**Container versus  
Images**

Jetzt ist es an der Zeit, ein wenig mit `docker` zu experimentieren, damit Sie den Unterschied zwischen einem Image und seinen Containern verstehen lernen. Wenn Sie den Ubuntu-Container mit `[Strg]+[D]` verlassen und dann mit `docker run` neuerlich starten, wird Ihnen vielleicht auffallen, dass der Container einen neuen Hostnamen erhalten hat. Merkwürdig!

Versuchen Sie nun, mit `touch` eine neue Datei anzulegen. Beenden Sie den Container, führen Sie erneut `docker run` aus, und suchen Sie nach der Datei. Sie werden sie nicht finden. Kann Docker Veränderungen am Dateisystem nicht konsistent speichern?

```
root# docker run -it ubuntu
root@168a2ec648f0:/# touch abc
root@168a2ec648f0:/# exit
```

```
root# docker run -it ubuntu
root@26a4a5f5b281:/# ls -l abc
ls: cannot access 'abc': No such file or directory
```

Ein wenig klarer wird die Sache, wenn Sie ausprobieren, `docker run` in mehreren Terminalfenstern parallel auszuführen. Das ist kein Problem für Docker! Der Grund für dieses auf den ersten Blick merkwürdige Verhalten besteht darin, dass Docker jedes Mal, wenn Sie `docker run imagename` ausführen, einen *neuen* Container für dieses Image erzeugt! Jeder Container erhält automatisch eine zufällige UID, deren Kurzform sich im Hostnamen widerspiegelt, sowie ein eigenes Dateisystem. (Genau genommen wird über das Read-Only-Dateisystem des Images ein Read-Write-Dateisystem des Containers gelegt – siehe [Abschnitt 41.5](#), »Interna«.)

```

root# docker ps -a | grep ubuntu
CONTAINER ID   IMAGE     COMMAND                  STATUS              NAMES
26a4a5f5b281  ubuntu   "/bin/bash"             Exited (2) 6 seconds ago  wizardly_ptolemy
168a2ec648f0  ubuntu   "/bin/bash"             Exited (0) 6 minutes ago  musing_kilby
f8fec4640176  ubuntu   "/bin/bash"             Up 36 minutes           naughty_borg
...

```

Die naheliegende Lösung besteht darin, dass Sie ab dem zweiten Start explizit angeben, welchen Container Sie nun verwenden möchten. Dabei können Sie wahlweise die zufällige Container-ID oder den ebenso zufälligen, aber leichter zu merkenden Container-Namen verwenden (siehe das Ergebnis von `docker ps -a`).

**Container mit  
docker start  
erneut ausführen**

Beachten Sie, dass Sie zum neuerlichen Start eines bereits vorhandenen Containers `docker start` verwenden müssen, nicht `docker run`! Damit Sie den Container interaktiv verwenden können, müssen Sie wieder die Option `-i` angeben. `-t` ist hingegen nicht zulässig. Die meisten Optionen, die Sie beim Erzeugen eines Containers mit `docker run` angeben, werden dauerhaft im Container gespeichert. `-i` zählt zu den Ausnahmen.

Die folgenden zwei Beispiele zeigen, dass Docker die Datei `abc` durchaus gespeichert hat. Entscheidend ist nur, dass Sie beim nächsten Start die richtige Container-ID bzw. den Container-Namen angeben.

```

root# docker start -i musing_kilby
root@168a2ec648f0:/# ls -l abc
-rw-r--r--. 1 root root 0 Mar 10 15:27 abc
root@168a2ec648f0:/# exit

```

```

root# docker start -i 168a2ec648f0
root@168a2ec648f0:/# ls -l abc
-rw-r--r--. 1 root root 0 Mar 10 15:27 abc
root@168a2ec648f0:/# exit

```

### Autovervollständigung

Bei der Eingabe von `docker start` können Sie die Container-ID bzw. den Namen des Containers mit `[Tab]` vervollständigen.

Die zufälligen Container-Namen sind zwar handlicher als die IDs, noch besser ist es aber, Containern eigene Namen zuzuordnen. Dazu geben Sie, wenn Sie den Container mit dem Kommando `docker run` erzeugen, mit `--name` den gewünschten Namen an. Optional können Sie dem Container bei dieser Gelegenheit auch gleich mit `-h` einen schöneren Hostnamen zuweisen. Dieser kann, muss aber nicht mit dem Container-Namen übereinstimmen.

**Eigene  
Container-Namen**

```
root# docker run -it --name myubuntu -h myubuntu ubuntu
root@myubuntu:/# ...
root@myubuntu:/# exit
```

In der Folge starten Sie den Container so:

```
root# docker start -i myubuntu
```

Parallel weitere  
Prozesse mit  
docker exec  
ausführen

Häufig wird in einem Docker-Container nur ein Prozess ausgeführt – sei es eine interaktive Shell im Vordergrund wie in den vorigen Beispielen, sei es ein Server-Dienst im Hintergrund. `docker exec` bietet Ihnen die Möglichkeit, parallel zu einem laufenden Container beliebige weitere Prozesse zu starten. `docker exec` ist primär zum Debugging gedacht, kann aber universell eingesetzt werden. Das folgende Kommando startet parallel zum laufenden `myubuntu`-Container das Kommando `top`. Es wird ausgeführt, bis es durch `[q]` beendet wird.

```
root# docker exec -it myubuntu /usr/bin/top
```

### Beispiel 3: Swift unter Linux ausführen

Swift ist eine relativ neue Programmiersprache, die Apple primär zur App-Programmierung für seine Plattformen (iOS, macOS etc.) entwickelt hat. Apple möchte Swift aber auch auf Servern zum Durchbruch verhelfen und wartet daher eine für Ubuntu kompilierte Version. Diese stand im Frühjahr 2017 allerdings nur als TAR-Archiv für Ubuntu zur Verfügung. Es gab keine Pakete, geschweige denn richtige Paketquellen; andere Distributionen wurden offiziell gar nicht unterstützt.

Anstatt sich mit der manuellen Installation von Swift zu plagen, können Sie Swift als Docker-Image ausführen. Diese Vorgehensweise hat den Vorteil, dass sie unter macOS und auf allen Linux-Distributionen funktioniert, nicht nur unter Ubuntu.

Offizielles  
Docker-Image

Im Sommer 2017 basierte das offizielle Swift-Image auf Ubuntu 16.04 und enthielt Swift 3. Mit einer Größe von rund 1 GiB ist es vergleichsweise riesig. Aktuelle Informationen und Tipps zur Anwendung des Images können Sie hier nachlesen:

[https://hub.docker.com/\\_/swift](https://hub.docker.com/_/swift)

Um Swifts interaktiven Kommandointerpreter (REPL = Read–Eval–Print Loop) auszuprobieren, erzeugen Sie einen neuen Container wie im folgenden Beispiel. Dabei stellt die Option `--cap-add` sicher, dass der Container Prozesse mit `ptrace` verfolgen kann. Diese Funktion benötigt der Swift-Interpreter zum Debugging.

`--rm` bewirkt, dass der Container nach dem Ende der Ausführung sofort wieder gelöscht wird. Die doppelte Angabe von `swift` am Ende des Kommandos ist kein Tippfehler: Das erste `swift` bezeichnet das Image, das zweite `swift` gibt an, dass innerhalb

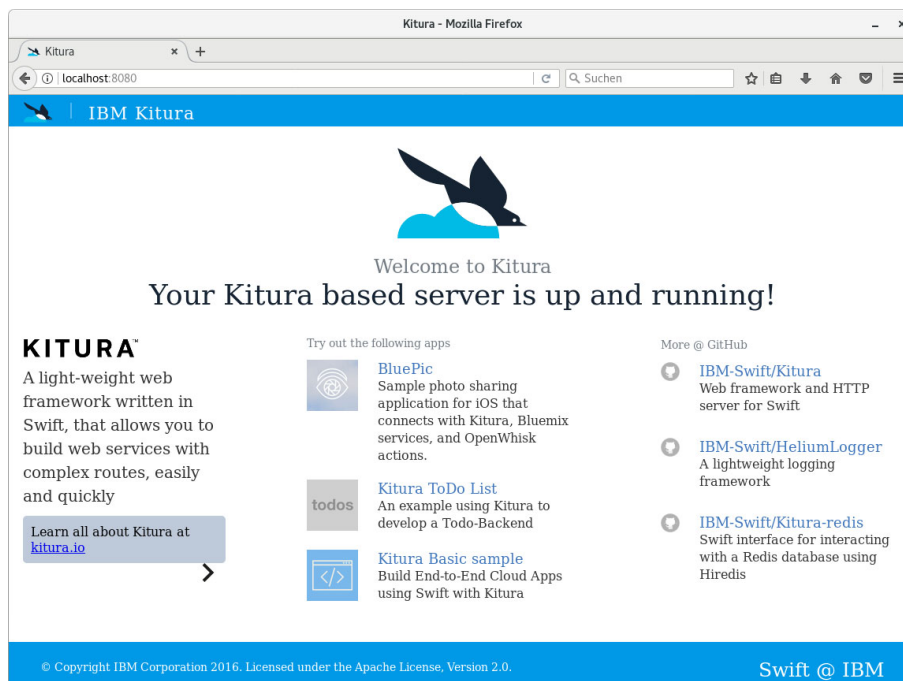
des Containers das Kommando `swift` (also der Swift-Kommandointerpreter) ausgeführt werden soll.

```
root# docker run --cap-add sys_ptrace -it --rm swift swift
Welcome to Swift version 3.0.2 (swift-3.0.2-RELEASE).
Type :help for assistance.
 1> let s="Hello World"
s: String = "Hello World"
 2> print(s)
Hello World
 3> :exit
```

Außerhalb des Apple-Universums ist Swift primär zur Entwicklung der serverseitigen Infrastruktur von Apps gedacht. Verschiedene Frameworks helfen dabei. Das aktuell bekannteste Framework Kitura (<http://kitura.io>) wurde von IBM entwickelt. Server-side-Swift

Auch für Kitura gibt es ein fertiges Docker-Image. Es basierte im Sommer 2017 unerklärlicherweise noch auf Ubuntu 14.04, was aber für den Einsatzzweck – das Ausprobieren von Swift mit Kitura – nicht wirklich eine Einschränkung darstellte.

```
root# docker run -it --cap-add sys_ptrace -p 8080:8080 -h kituratest \
      --name kituratest ibmcom/kitura-ubuntu:latest /bin/bash
```



**Abbildung 41.1** Der getestete Webserver basiert auf der Programmiersprache Swift 3 und dem Kitura-Framework von IBM.

Weitere Starts erfolgen mit `docker start`. Innerhalb des Containers ist im Verzeichnis `Kitura-Starter` schon alles vorbereitet, um die Kitura-Beispiel-App in Betrieb zu nehmen. Sie müssen den Code nur noch kompilieren und starten:

```
root@kiturestest:~# cd Kitura-Starter
root@kiturestest:~# swift build           (Beispielprojekt kompilieren ...)
root@kiturestest:~# .build/debug/Kitura-Starter (... und starten)
... Server will be started on 'http://localhost:8080'.
```

Da der Port 8080 des Containers mit Port 8080 des lokalen Rechners verbunden ist, können Sie den durch Kitura realisierten Webserver auf einem lokalen Webbrowser unter der Adresse `http://localhost:8080` ausprobieren (siehe [Abbildung 41.1](#)).

### Beispiel 4: MariaDB ausführen

Im Docker-Hub gibt es für die Datenbank-Server MySQL und MariaDB diverse Images. Für diesen Abschnitt habe ich das offizielle MariaDB-Image getestet. Es ist auf der folgenden Seite wunderbar dokumentiert:

[https://hub.docker.com/r/\\_/mariadb](https://hub.docker.com/r/_/mariadb)

Beim Einrichten des Containers wird das Root-Passwort für den Datenbank-Server als Parameter übergeben. Das ist kein optimaler Mechanismus, weil das Passwort kurz im Klartext in der Prozessliste aufscheint, die für alle sichtbar ist. Die Umgebungsvariable `MYSQL_ROOT_PASSWORD` kann später aber auch mit `docker inspect mariadb-test1` ausgelesen werden – aber nur von Anwendern, die `docker` ausführen dürfen.

```
root# docker run -d --name mariadb-test1 -e MYSQL_ROOT_PASSWORD=geheim mariadb
...
root# docker stop mariadb-test1           (Container stoppen)
```

Im Vergleich zu den bisherigen Beispielen gibt es hier einige Besonderheiten:

- ▶ Der MariaDB-Server wird als Hintergrunddienst ausgeführt, nicht interaktiv im Vordergrund. Um die Container-Ausführung zu beenden, führen Sie `docker stop` aus, für einen neuerlichen Start im Hintergrund `docker start`.
- ▶ MariaDB speichert alle Datenbankdateien in `/var/lib/mysql`. Im Dockerfile, also in der Beschreibung des Images, ist dieses Verzeichnis als »Volume« gekennzeichnet. Das bedeutet, dass das Verzeichnis außerhalb des Containers direkt im Dateisystem angelegt wird. Docker richtet dazu automatisch ein Verzeichnis in `/var/lib/docker/volumes` ein. Den genauen Verzeichnisnamen können Sie mit `docker inspect mariadb-test1` ermitteln, wenn Sie in der mehrseitigen Ausgabe dieses Kommandos nach `Mounts` suchen.



Im MariaDB-Container wird nur der MySQL-Server ausgeführt. Anders als in den beiden vorangegangenen Beispielen ist eine interaktive Nutzung nicht vorgesehen. Sie können aber mit `docker exec` parallel zum Container einen zweiten Prozess starten, um dort den MariaDB-Client `mysql` auszuführen:

**MariaDB-Client aufrufen**

```
root# docker exec -it mariadb-test1 mysql -u root -p
Enter password: ***** (MariaDB-root-Passwort)
Welcome to the MariaDB monitor.
Server version: 10.1.21-MariaDB
MariaDB [(none)]> show status;
...
MariaDB [(none)]> exit
```

Wenn Sie neugierig sind und weitere Details über den Container ermitteln möchten, starten Sie parallel zum MariaDB-Server mit `docker exec` einen `bash`-Prozess für den Container:

**Ein Blick in den Container**

```
root@host# docker exec -it mariadb-test1 /bin/bash
root@34de866bfff12:/# cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 8 (jessie)"
...
root@34de866bfff12:/# ps -ax
  PID TTY          STAT       TIME COMMAND
    1 ?           Ssl        0:00 mysql
   147 ?           Ss         0:00 /bin/bash
   218 ?           R+         0:00 ps -ax
root@34de866bfff12:/# mysql --version
mysql Ver 10.1.21-MariaDB-1~jessie for debian-linux-gnu on x86_64
      (mariadb.org binary distribution)
root@34de866bfff12:/# exit
```

Da der MariaDB-Container im Hintergrund läuft, sehen Sie im Terminal weder Fehlermeldungen noch Logging-Ausgaben. Diese können Sie mit `docker logs` lesen:

**Logging**

```
root# docker logs mariadb-test1
2017-03-18 ... [Note] mysqld (mysqld 10.1.21-MariaDB-1~jessie) starting
              as process 1 ...
2017-03-18      [Note] InnoDB: Using mutexes to ref count buffer pool pages
2017-03-18      [Note] InnoDB: The InnoDB memory heap is disabled
2017-03-18      [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
...
```

Mit `docker rm mariadb-test1` können Sie den Container löschen. Das externe Volume, das wegen des InnoDB-Masterspace mindestens 100 MiB groß ist, bleibt dabei allerdings erhalten. Das ist beabsichtigt, weil sich dort ja die Datenbankdateien befinden. Deren irrtümliches Löschen soll vermieden werden. Wenn Sie sicher sind, dass Sie das

**Container und Volume löschen**

Volume nicht mehr benötigen, führen Sie `docker volume prune` aus. (Vorsicht: Dieses Kommando löscht *alle* Volumes, die keinem Container zugeordnet sind.)

```
root# docker rm mariadb-test1
root# docker volume prune
```

#### Unkomplizierte Updates dank eigener Volume-Verzeichnisse

Wenn Sie vorhaben, den Datenbank-Server »richtig« zu nutzen, sollten Sie für das Volume mit den Datenbankdateien ein Verzeichnis Ihrer Wahl vorsehen und dieses mit der Option `-v` angeben. Das hat den Vorteil, dass Sie dieses Verzeichnis z. B. nach einem Update des Containers weiterverwenden können. Die folgenden Kommandos zeigen die Kommandoabfolge. Beachten Sie, dass Sie das MariaDB-Root-Passwort nur einmal angeben müssen!

```
root# mkdir /home/kofler/varlibmysql
root# docker run -d --name mariadb-test2 -e MYSQL_ROOT_PASSWORD=geheim \
-v /home/kofler/varlibmysql:/var/lib/mysql mariadb
... (Datenbank nutzen)
root# docker stop mariadb-test2
root# docker rm mariadb-test2 (Container löschen, Datenbank bleibt erhalten)
... (neue MariaDB-Version installieren)
root# docker run -d --name mariadb-test3 \
-v /home/kofler/varlibmysql:/var/lib/mysql mariadb
```

#### Port 3306 dem Host-Rechner zugänglich machen

Es gibt verschiedene Möglichkeiten, auf den MariaDB-Server zuzugreifen. Den Aufruf des MariaDB-Clients mit `docker exec` habe ich bereits erwähnt. Eine zweite Variante besteht darin, den Port 3306 des Containers mit der Option `-p` im Host-System zugänglich zu machen – hier unter dem lokalen Port 13306, um Konflikte mit einer eventuell laufenden lokalen Instanz zu vermeiden. Ich gehe in diesem Beispiel davon aus, dass das lokale Verzeichnis `/home/kofler/varlibmysql` bereits mit MariaDB-Datenbanken initialisiert ist.

```
root# docker run -d --name mariadb-test4 -p 13306:3306 \
-v /home/kofler/varlibmysql:/var/lib/mysql mariadb
```

Sofern auf dem Host die MariaDB-Client-Tools installiert sind (unter Fedora: `dnf install mariadb`), können Sie eine Verbindung zum MariaDB-Docker-Container herstellen:

```
user$ mysql -u root -p --port 13306 --protocol=tcp
Enter password: *****
```

#### Container-Container-Kommunikation

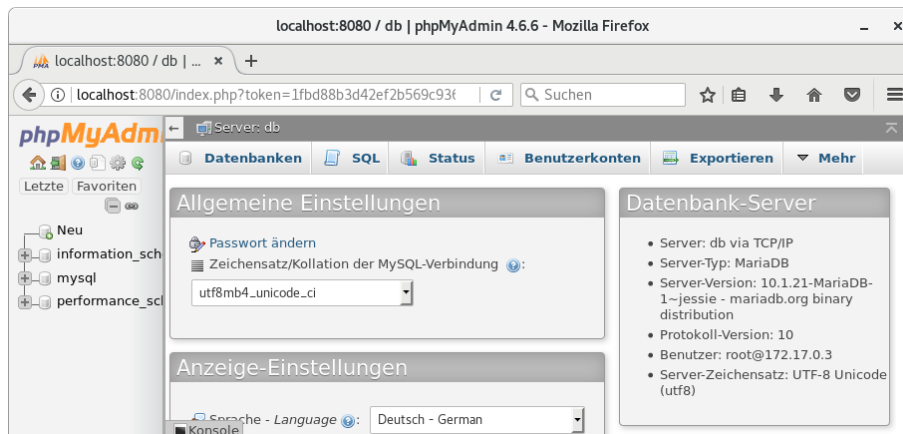
Die Kommunikationsvarianten 1 (MySQL-Client als Docker-Prozess) und 2 (über einen Netzwerkport) sind gewissermaßen Brücken zur klassischen Welt. Beide Varianten sind im Docker-Universum aber untypisch. Dieses sieht als dritte, wesentlich elegantere Form die Kommunikation direkt zwischen Docker-Containern vor. Dieser Fall liegt z. B. vor, wenn in einem Container der Datenbank-Server läuft, im zweiten ein Webserver mit einem CMS.

Um das auszuprobieren, benötigen Sie einen weiteren Container. Für derartige Tests besonders gut geeignet ist das Image `phpmyadmin`, das eine phpMyAdmin-Installation enthält. Die Inbetriebnahme sieht so aus:

```
root# docker run -d --name pma --link mariadb-test4:db \
    -p 8080:80 phpmyadmin/phpmyadmin
```

Entscheidend ist hier die Option `--link`: Der erste Teil des Link-Parameters gibt an, zu welchem laufenden Container die Verbindung hergestellt werden soll. Der zweite Teil nach dem Doppelpunkt gibt an, unter welchem Hostnamen der externe Container intern angesprochen werden soll. In der Docker-Dokumentation wird dieser Teil als Link-Alias bezeichnet, und man bekommt den Eindruck, es spiele keine große Rolle, welche Bezeichnung Sie dafür verwenden. Das ist aber irreführend. Beispielsweise erwartet das `phpmyadmin`-Image, dass der MySQL- oder MariaDB-Server auf dem Host `db` läuft.

Um phpMyAdmin auszuprobieren, öffnen Sie auf dem Docker-Host im Webbrowser die Seite `http://localhost:8080` und loggen sich als `root` mit dem bei `docker run -name mariadb-xxx` genannten Passwort ein (siehe [Abbildung 41.2](#)).



**Abbildung 41.2** phpMyAdmin und der MariaDB-Server laufen in zwei getrennten Containern.

Intern kommunizieren die beiden Container im Docker-eigenen privaten Netzwerk. Docker trägt dazu beim Start des Containers in `/etc/hosts` die entsprechenden Daten ein:

```
root# docker exec -it pma /bin/sh
/ # cat /etc/hosts
127.0.0.1      localhost
...
172.17.0.2    db 91540b356cfd mariadb-test4
```

## Beispiel 5: WordPress installieren

Wie lange haben Sie das letzte Mal gebraucht, um auf einem Server WordPress zu installieren? Wenn Sie das regelmäßig machen, vermutlich nur ein paar Minuten – aber beim ersten Mal dauert es oft viel länger, bis das Fundament, das aus Apache, PHP und MySQL oder MariaDB besteht, mit allen seinen Zusatzpaketen installiert und konfiguriert ist.

Deutlich schneller geht es mit Docker. Einmal vorausgesetzt, dass Sie zuvor wie in Beispiel 4 beschrieben einen MySQL- oder MariaDB-Container eingerichtet haben, sind zur Inbetriebnahme von WordPress nur die folgenden zwei Kommandos notwendig:

```
root# mkdir /home/kofler/wp-html
root# docker run -d --name wp-test1 --link mariadb-test4:mysql \
    -v /home/kofler/wp-html:/var/www/html -p 8081:80 \
    -e WORDPRESS_DB_PASSWORD=geheim wordpress
```

Unter der Adresse `http://localhost:8081` nehmen Sie die Webseite nun in Betrieb (siehe [Abbildung 41.3](#)). Die Anfangskonfiguration beschränkt sich auf zwei Dialoge: Auf der ersten Seite wählen Sie die gewünschte Sprache aus, auf der zweiten Seite richten Sie den WordPress-Administrator ein. Die oft heikle Konfiguration der Zugangsdaten des Datenbank-Servers entfällt: Um sie hat sich ein Script des WordPress-Containers bereits gekümmert.

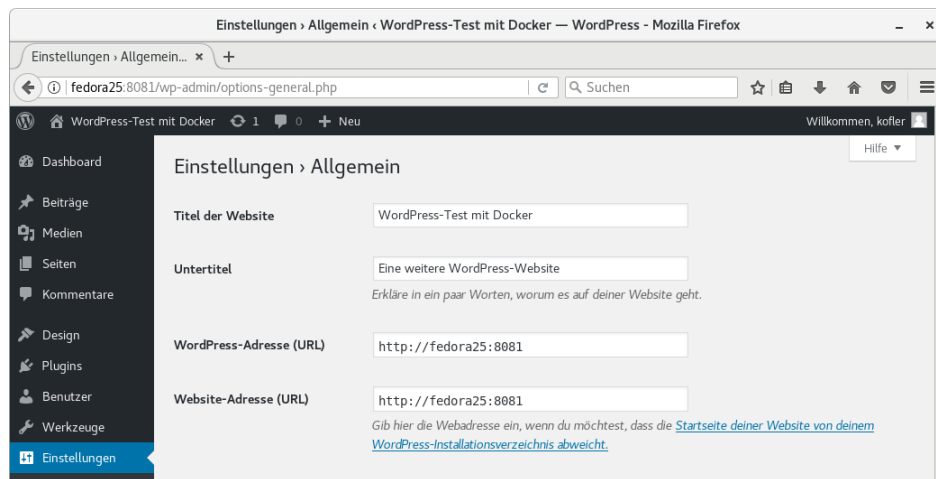


Abbildung 41.3 WordPress und der MariaDB-Server laufen in zwei getrennten Containern.

Kurz eine Erklärung der `docker-run`-Optionen:

- ▶ Bei `--link` ist es entscheidend, dass Sie vor dem Doppelpunkt den Namen des Containers angeben, in dem der MySQL- oder MariaDB-Server läuft. Dieser Container muss zu dem Zeitpunkt, zu dem Sie `docker run` ausführen, bereits aktiv sein! Nach

dem Doppelpunkt müssen Sie `mysql` angeben, ganz egal, ob Sie wirklich MySQL oder wie in diesem Beispiel MariaDB verwenden. `mysql` ist der Hostname, unter dem der WordPress-Container den Datenbank-Container im privaten Docker-Netzwerk anspricht.

- ▶ Die Option `-v` ist optional. Sie erreichen damit, dass alle HTML-, PHP-, CSS- und sonstigen Dateien der WordPress-Installation außerhalb des Containers an dem von Ihnen angegebenen Ort gespeichert werden. Wenn Sie später den WordPress-Container aktualisieren, können Sie dieses Verzeichnis weiterverwenden.
- ▶ Mit `-p` geben Sie an, unter welchem Port die WordPress-Installation auf dem Docker-Host angesprochen werden kann. Da der Port 8080 im vorigen Beispiel schon für phpMyAdmin zum Einsatz kam, habe ich jetzt eben 8081 verwendet. Wenn auf dem Docker-Host der Port 80 frei ist, können Sie natürlich auch `-p 80:80` angeben.
- ▶ Die Option `-e` ist normalerweise überflüssig, weil Docker das root-Passwort des MariaDB-Servers mit `docker inspect` aus dem Datenbank-Container ausliest. Das funktioniert aber nur, wenn beim Einrichten des MariaDB-Containers die Option `-e MYSQL_ROOT_PASSWORD=xxx` verwendet wurde. Beim ersten Mal ist das immer der Fall – nicht aber, wenn Sie später ein MariaDB-Update durchführen und das `/var/lib/mysql`-Volume weiterverwenden.

Weitere Umgebungsvariablen des offiziellen WordPress-Images finden Sie hier dokumentiert:

[https://hub.docker.com/\\_/wordpress](https://hub.docker.com/_/wordpress)

Solange Sie vom Docker-Host aus auf WordPress zugreifen (URL `http://localhost:8081`), funktioniert alles bestens. Wenn Sie aber von einem anderen Host aus auf die Webseite zugreifen, erscheint die Seite unvollständig ohne Bilder, ohne CSS-Formatierung etc. Schuld daran sind die allgemeinen WordPress-Einstellungen. In diesem Feld ist in die WordPress-URL der Hostname `localhost` eingetragen. Dort müssen Sie den richtigen Hostnamen angeben (siehe [Abbildung 41.3](#)).

**Zugriff auf die WordPress-Seite von anderen Hosts**

WordPress enthält ein eigenes Update-System. Neue Minor-Versionen werden automatisch installiert, Major-Versionen über die Admin-Seite des CMS. Im Laufe der Zeit wird es aber auch Updates für das zugrunde liegende Image geben, z. B. neue Versionen von Apache und PHP.

**Updates**

Derartige Updates führen Sie am einfachsten durch, indem Sie den Container stoppen, löschen und neu einrichten. Auf die Angabe des Passworts für MariaDB können Sie verzichten – der Datenbankzugang ist ja schon konfiguriert, die entsprechenden Daten befinden sich im `/var/lib/mysql`-Verzeichnis.

```

root# docker pull wordpress (aktuelles Image herunterladen)
root# docker stop wp-test1 (alten Container stoppen)
root# docker rm wp-test1 (alten Container löschen)
      (neuen Container einrichten und starten)
root# docker run -d --name wp-test2 --link mariadb-test4:mysql \
      -v /home/kofler/wp-html:/var/www/html -p 8081:80 wordpress

```

### 41.3 Docker administrieren

Die gesamte Administration von Docker erfolgt durch das gleichnamige Kommando. Es kommuniziert standardmäßig mit dem Docker-Dämon, der auf dem gleichen Rechner läuft. Mit der Option `-H` können Sie aber auch einen externen Docker-Host angeben und diesen steuern, bei Bedarf über eine TLS-verschlüsselte Verbindung.

Eine vollständige Beschreibung des `docker`-Kommandos ist hier aus Platzgründen unmöglich. [Tabelle 41.1](#) gibt einen Überblick über die wichtigsten Sub-Kommandos. Zahlreiche Beispiele für deren Anwendung sind über das gesamte Kapitel verteilt.

| Kommando                           | Bedeutung  |
|------------------------------------|--|
| <code>docker build</code>          | neues Image laut Dockerfile erzeugen               |
| <code>docker create</code>         | neuen Container erzeugen, aber nicht starten       |
| <code>docker exec</code>           | Kommando in einem laufenden Container ausführen    |
| <code>docker images</code>         | lokale Docker-Images auflisten                     |
| <code>docker info</code>           | Status des Docker-Systems anzeigen                 |
| <code>docker inspect cname</code>  | Konfiguration und Status eines Containers anzeigen |
| <code>docker login/logout</code>   | bei einem Docker-Account an-/abmelden              |
| <code>docker logs cname</code>     | Logging-Ausgaben des Containers zeigen             |
| <code>docker network subcmd</code> | Netzwerkconfiguration verwalten                    |
| <code>docker ps</code>             | aktuell laufende Container auflisten               |
| <code>docker ps -a</code>          | zuletzt gestartete Container auflisten             |
| <code>docker pull iname</code>     | Image herunterladen bzw. aktualisieren             |
| <code>docker push iname</code>     | eigenes Image in ein Docker-Repository hochladen   |
| <code>docker rm cname/cid</code>   | Container löschen                                  |
| <code>docker rmi iname</code>      | lokales Image löschen                              |
| <code>docker run iname</code>      | neuen Container erzeugen und starten               |

**Tabelle 41.1** Wichtige `docker`-Kommandos

| Kommando                                  | Bedeutung                                     |
|---|---|
| <code>docker start cname/cid</code>       | vorhandenen Container starten                 |
| <code>docker stop cname/cid</code>        | Container stoppen                             |
| <code>docker tag oldiname newiname</code> | Images umbenennen bzw. mit einem Tag versehen |
| <code>docker volume subcmd</code>         | Volumes verwalten                             |

**Tabelle 41.1** Wichtige docker-Kommandos (Forts.)

`man docker` führt in die allgemeine `man`-Seite des Docker-Kommandos. Diese Seite enthält einen Überblick über alle Kommandos, die innerhalb von `docker` ausgeführt werden können. Die Details dieser Sub-Kommandos sind über weitere `man`-Seiten verteilt, die Sie mit `man docker-run`, `docker-ps` etc. aufrufen.

Online-Dokumentation

Wenn Sie keine umfassenden Hilfetexte, sondern nur eine kurze Zusammenfassung wünschen, führen Sie `docker -h` oder `docker help` aus. Details zu einem Sub-Kommando erhalten Sie mit `docker help run`, `docker help ps` etc. (Achten Sie auf die richtige Reihenfolge! Zuerst `help`, dann der Kommandoname.) Alternativ funktioniert auch `docker run --help`. Die Kurzschreibweise `-h` ist in diesem Kontext aber nicht erlaubt.

Schöner formatiert als die `man`-Seiten ist die Dokumentation zum `docker`-Kommando im Internet. Sie ist über zahlreiche Seiten verteilt. Ein guter Startpunkt ist:

<https://docs.docker.com/engine/reference/run/>

## Container erzeugen

Das Docker-Kommando `run` ist etwas irreführend benannt. Die primäre Aufgabe des Kommandos ist es, einen neuen Container zu erzeugen. Die angegebenen Optionen bestimmen dauerhaft die Eigenschaften des neuen Containers. Zum Schluss wird der neue Container auch gleich ausgeführt – daher `run`. In der Folge müssen Sie zum neuerlichen Ausführen des Containers aber das Kommando `docker start` verwenden!

`docker run`

`docker run` kennt unzählige Optionen, auch wenn ich an dieser Stelle nur auf die wichtigsten eingehen kann (siehe [Tabelle 41.2](#)). (Die Manual-Seiten von `man docker-run` würden sich in der Formatierung dieses Buchs über rund 15 Seiten erstrecken!) Die allgemeine Syntax für `docker run` sieht so aus:

```
docker run [optionen] imagename [kommando [kommandoargumente]]
```

| Option             | Bedeutung   |
|--------------------|---|
| --cpus="1.25"      | Der Container darf maximal 1,25 CPU-Cores auslasten.          |
| -d                 | Container als Dämon ausführen                                 |
| -e VAR=value       | Umgebungsvariable für den Container setzen                    |
| -h hostname        | den Hostnamen für den Container festlegen                     |
| -i                 | Container diesmal interaktiv ausführen                        |
| --link cname:hname | Container cname unter dem Hostnamen hname ansprechen          |
| -m 512m            | Container-RAM auf 512 MiB limitieren                          |
| --name cname       | dem Container einen Namen zuweisen                            |
| -p localport:cport | Ports des Containers mit Ports des Hosts verbinden            |
| -P                 | alle Ports des Containers mit zufälligen Host-Ports verbinden |
| --rm               | Container löschen, sobald die Ausführung endet                |
| -t                 | Pseudo-Terminal mit Standardeingabe verbinden                 |
| -v cdir            | Container-Verzeichnis als Volume einrichten                   |
| -v localdir:cdir   | Host-Verzeichnis mit Container-Verzeichnis verbinden          |

**Tabelle 41.2** Wichtige Optionen des docker-run-Kommandos

#### Kommando im Container ausführen

Wenn Sie nach dem Image-Namen kein Kommando angeben, wird das für das Image festgelegte Defaultkommando ausgeführt (siehe auch die Beschreibung der Schlüsselwörter `CMD` und `ENTRYPOINT` in [Abschnitt 41.4](#), »Docker Images erzeugen und weitergeben«). Bei Base Images für Linux-Distributionen lautet das Defaultkommando oft `bash` und ermöglicht so die interaktive Verwendung des Containers.

Beachten Sie, dass Sie sich beim Erzeugen eines neuen Containers nicht nur mit den Optionen, sondern auch mit dem auszuführenden Kommando festlegen! Wenn Sie den mit `docker run` erzeugten Container später mit `docker start` neuerlich ausführen, wird im Container dasselbe Kommando wie beim ersten Start ausgeführt. Immerhin besteht die Möglichkeit, bei einem laufenden Container mit `docker exec` ein weiteres Kommando parallel auszuführen. Diese Möglichkeit ist aber primär zu Debugging-Zwecken gedacht.

#### docker inspect

Eine detaillierte Zusammenfassung aller Konfigurationsparameter sowie des Status eines Containers erhalten Sie mit `docker inspect`:

```
root# docker inspect mariadb1
[
  {
    "Id": "96258e673d9a651a420e0e2c45632bc83f967779952111493f18392a680f404b",
    "Created": "2017-03-09T18:34:33.887279271Z",
```



```
"Path": "docker-entrypoint.sh",
"Args": [ "mysqld" ],
"State": { "Status": "exited",
...

```

### Container-Eigenschaften sind unveränderlich!

Die meisten Eigenschaften eines Containers, die Sie mit `docker run` festlegen, können später nicht mehr verändert werden, weder im laufenden Betrieb noch wenn Sie bereit sind, die Container-Ausführung zu stoppen und den Container später mit `docker start` wieder zu starten. Diese Einschränkung ist in der Praxis sehr ärgerlich, wenn Sie nach mehreren Tagen Arbeit bemerken, dass Sie vergessen haben, einen Port oder ein Verzeichnis zwischen Host und Container zu verbinden. Virtuelle Maschinen bieten in dieser Hinsicht definitiv mehr Flexibilität.

In den Docker-Foren taucht der Wunsch nach veränderlichen Containern immer wieder auf, aber die Realisierung scheint technisch schwierig zu sein. Für das Port-Binding gibt es immerhin eine Notlösung durch die manuelle Ausführung von `iptables`:

<https://stackoverflow.com/questions/19897743>

Für Container, die im Hintergrund (also als Dämon) ausgeführt werden sollen, bietet `docker create` eine Alternative zu `docker run -d` an. `docker create` kennt größtenteils dieselben Optionen wie `docker run`. Es richtet den Container nur ein, ohne diesen gleich zu starten. Dazu führen Sie dann später `docker start` aus.

### Container und Images löschen

`docker rm id` bzw. `docker rm name` löscht den angegebenen Container. Container-IDs und -Namen können Sie gegebenenfalls mit `docker ps -a` ermitteln.

```
root# docker rm 7be570d73bd3
```

#### Vorsicht

`docker rm` hat große Ähnlichkeiten mit dem klassischen `rm`-Kommando: Es löscht ohne Rückfrage und ohne die Möglichkeit, die Operation rückgängig zu machen!

Das folgende Kommando erzeugt zuerst mit `ps` eine Liste aller IDs von Containern, die vom Image `ubuntu` abgeleitet sind. Diese Liste wird an `docker rm` weitergegeben, um alle Container zu löschen. Wenn Sie also eine Weile mit `docker run ubuntu` experimentiert haben, können Sie so alle bei dieser Gelegenheit erzeugten Container wieder löschen:

```
root# docker rm $(docker ps -a -q -f ancestor=ubuntu)
```

Noch radikaler ist das nächste Kommando: Es löscht alle existierenden Container!

```
root# docker rm $(docker ps -aq)
```

`docker rmi imagename` löscht das angegebene Image. Das ist nur möglich, wenn es keine von dem Image abgeleiteten Container gibt. (Sie können das Löschen mit `-f` erzwingen, die vom Image abgeleiteten Container sind dann aber nicht mehr verwendbar.) Die folgenden Kommandos löschen zuerst alle `hello-world`-Container und dann das `hello-world`-Image:

```
root# docker rm $(docker ps -a -q -f ancestor=hello-world)
root# docker rmi hello-world
```

Beachten Sie, dass `docker rmi` nur lokal ausgeführt wird. Wenn Sie ein eigenes Image in den Docker Hub hochgeladen haben (siehe [Abschnitt 41.4](#), »Docker Images erzeugen und weitergeben«), bleibt das Image dort erhalten. Sie können es bei Bedarf in der Weboberfläche von <https://hub.docker.com> löschen.

## Volumes

Grundsätzlich werden alle veränderlichen Daten im Overlay-Dateisystem eines Containers gespeichert. (Wie das hinter den Kulissen vor sich geht, erklärt der [Abschnitt 41.5](#), »Intern«.) Wird ein Container gelöscht, gehen damit auch alle Dateien verloren, die während der Lebenszeit des Containers erzeugt bzw. verändert wurden.

Oft ist es aber zweckmäßig, dass manche Verzeichnisse die Lebenszeit eines Containers überdauern. Besonders einfach ist das bei einem Datenbank-Server zu verstehen: Wenn ein Container für die Version  $x$  durch einen Container für die Version  $x+1$  ersetzt wird, sollen die im ersten Container eingerichteten Datenbanken auch nach dem Update noch zur Verfügung stehen.

Als Lösung für dieses Problem bietet Docker »Volumes« an. Das sind zwischen dem Container und dem Host-Rechner geteilte Verzeichnisse. Es gibt verschiedene Arten, wie Volumes definiert werden: solche, die von Docker automatisch verwaltet werden, und solche, wo Sie beim Einrichten des Containers explizit den entsprechenden Ort im Host-System angeben.

### Automatische Volume- Verwaltung

Wenn Sie `docker run` mit der Option `-v` ausführen (z. B. `-v /var/lib/mysql`) oder wenn ein Verzeichnis im Dockerfile mit dem Schlüsselwort `VOLUME` genannt ist (siehe auch [Abschnitt 41.4](#), »Docker Images erzeugen und weitergeben«), dann legt Docker beim Erzeugen des Containers auf dem Hostsystem automatisch ein Verzeichnis `/var/lib/docker/volumes/nnn` an. Es wird an der Stelle des Volume-Verzeichnisses in das Docker-Dateisystem eingebunden. Wenn dann z. B. der MySQL- oder MariaDB-Server im Container eine Datei in `/var/lib/mysql` erzeugt, wird diese tatsächlich im Verzeichnis `/var/lib/docker/volumes/nnn/` gespeichert.

Grundsätzlich funktioniert also alles automatisch. Der Nachteil dieser Vorgehensweise besteht darin, dass es mühsam ist, Volumes und Container korrekt zuzuordnen – insbesondere dann, wenn ein neu eingerichteter Container das Volume eines älteren, womöglich schon gelöschten Containers weiter nutzen soll. Die relevanten Informationen finden Sie, wenn Sie in der endlosen Ausgabe von `docker inspect containername` einen scharfen Blick auf den `Mounts`-Abschnitt werfen:

```
root# docker inspect mariadb2
...
"Mounts": [
  {
    "Type": "volume",
    "Name": "8283...",
    "Source": "/var/lib/docker/volumes/8283.../_data",
    "Destination": "/var/lib/mysql",
    "Driver": "local",
    "Mode": "",
    "RW": true,
    "Propagation": ""
  }
]
...
```

Die oben erläuterte Option `-v` von `docker run` kann auch in der Form `-v host-dir:containerdir` verwendet werden. Auf diese Weise geben Sie selbst an, an welcher Stelle auf dem Host-Rechner ein beliebiges Verzeichnis des Containers als Volume genutzt werden soll. Das lokale Verzeichnis muss bereits existieren; es wird nicht automatisch erzeugt.

**Volumes manuell zuordnen**

Beachten Sie, dass das Volume durch `mount` in den Container-Verzeichnisbaum eingebunden wird. Eventuell im Container schon vorhandene Dateien des Verzeichnisses werden für den Container damit unsichtbar. Zugänglich sind nur die Dateien des Volumes.

```
root# docker create ... -v /myvolumes/varlibmysql:/var/lib/mysql imagename
```

Der Docker-Host und die Container verwenden unterschiedliche UIDs und GIDs. (Hinter den Kulissen sind dafür sogenannte *Namespaces* verantwortlich.) Deswegen werden bei einem `ls`-Kommando im Host falsche oder fehlende Besitzer- und Gruppen-Namen angezeigt. Nur ein `ls`-Kommando im Container liefert die richtige Zuordnung.

**UID- und GID-Probleme**

```
root@host# cd /myvolumes/varlibmysql (Sichtweise des Docker-Hosts)
root@host# ls -l
-rw-rw----. 1 systemd-timesync input 16384 17. Mär 16:24 aria_log.00000001
-rw-rw----. 1 systemd-timesync input 52 17. Mär 16:24 aria_log_control
drwx-----. 2 systemd-timesync input 4096 17. Mär 16:24 db1
...
```

```

root@host# ls -ln
-rw-rw----. 1 999 999    16384 Mar 17 15:24 aria_log.00000001
-rw-rw----. 1 999 999         52 Mar 17 15:24 aria_log_control
drwx-----. 2 999 999    4096 Mar 17 15:24 db1
...

root@container# cd /var/lib/mysql      (Sichtweise im Container)
root@host# ls -l
-rw-rw----. 1 mysql mysql    16384 Mar 17 15:24 aria_log.00000001
-rw-rw----. 1 mysql mysql         52 Mar 17 15:24 aria_log_control
drwx-----. 2 mysql mysql    4096 Mar 17 15:24 db1
...

root@host# ls -ln
-rw-rw----. 1 999 999    16384 Mar 17 15:24 aria_log.00000001
-rw-rw----. 1 999 999         52 Mar 17 15:24 aria_log_control
drwx-----. 2 999 999    4096 Mar 17 15:24 db1
...

```

Dieses Problem tritt nicht auf, wenn der Container-Prozess mit `root`-Rechten läuft. `root` hat sowohl auf dem Host als auch im Container die `UID=0` und `GID=0`. Aus Sicherheitsgründen ist die Ausführung von Docker-Prozessen mit `root`-Rechten aber nicht wünschenswert.

**Volumes löschen** Von Docker automatisch eingerichtete Volumes werden aus Sicherheitsgründen nicht gelöscht, wenn Sie den zugehörigen Container oder das zugrunde liegende Image löschen. Eine Liste aller Volume-IDs liefert `docker volume ls`. Mit `docker volume rm id` können Sie das angegebene Volume löschen, und mit `docker volume prune` löschen Sie alle Volumes, die keinem Container zugeordnet sind (weil dieser gelöscht wurde).

Manuell zugeordnete Volumes können auf keinen Fall mit dem `docker`-Kommando gelöscht werden. Es handelt sich aus Host-Sicht um ganz gewöhnliche Verzeichnisse. Wenn Sie diese wirklich löschen möchten, verwenden Sie `rm -rf`.

## Container-Updates durchführen

Viele Container basieren auf Linux-Distributionen. Wenn Sie einen Container über längere Zeit in Betrieb halten, müssen Sie sich naturgemäß auch um Updates kümmern. Der für Linux-Veteranen übliche Weg bestünde nun darin, im Container einfach die passenden Kommandos des jeweiligen Paketverwaltungssystems auszuführen, in einem Ubuntu-Container also `apt update` und `apt dist-upgrade`.

Grundsätzlich funktioniert das, aber die Vorgehensweise entspricht nicht den Docker-Empfehlungen. Die sehen so aus, dass Sie bei Bedarf das jeweils neueste Image herunterladen, den aktuellen Container löschen und dann einen entsprechenden neuen Container einrichten:

```

root# docker run -d --name version1 \
    -v /host-verzeichnis:/container-datenverzeichnis img
root# docker stop version1
root# docker rm version1
root# docker pull img      (aktualisiertes Image herunterladen)
root# docker run -d --name version2 \
    -v /host-verzeichnis:/container-datenverzeichnis img

```

Entscheidend dafür, dass das ohne Datenverlust funktioniert, ist die Verwendung eigener Volumes für alle Datenverzeichnisse des Containers. Oder, drastischer formuliert: Wenn Sie Docker richtig verwenden, ist der Container ein austauschbares Wegwerfprodukt. Sie müssen sicherstellen, dass sich alle Daten, die persistent über Updates erhalten bleiben, in eigenen Volumes gespeichert werden.

### Container verbinden (Links)

In der Docker-Welt herrscht der Grundsatz, dass Container jeweils nur eine möglichst abgegrenzte Aufgabe übernehmen sollen. Erfordert ein Test das Zusammenspiel mehrerer Server-Dienste, ist es in der Regel empfehlenswert, mehrere Container einzurichten und deren Funktionen zu verbinden. `docker run` sieht dazu die Option `--link cname:hname` vor.

Die Syntax ist am besten anhand eines Beispiels verständlich: Nehmen Sie an, der Container `c2` ist auf Funktionen des Containers `c1` angewiesen. In diesem Fall muss zuerst `c1` eingerichtet werden, danach `c2` mit einem Link auf `c1`:

```

root# docker run -d --name c1 ... img1
root# docker run -d --name c2 --link c1:hname ... img2

```

Anstelle von `hname` müssen Sie den Hostnamen angeben, unter dem `c2` den Dienst des Containers `c1` anspricht. Den erforderlichen Namen finden Sie in der Dokumentation des Images `img2`.

Docker-intern erfolgt die Kommunikation zwischen den Containern einfach innerhalb eines privaten Netzwerks (z. B. `172.17.*.*`; siehe `docker network inspect bridge`), das Docker für seine Container einrichtet. Damit `c2` auf den Container `c1` zugreifen kann, trägt Docker in die Datei `/etc/hosts` von `c2` automatisch den Hostnamen `hname` und die dazugehörige IP-Adresse ein. Die Beispiele 4 und 5 in [Abschnitt 41.2](#), »Docker kennenlernen«, zeigen jeweils Anwendungen der `--link`-Option.

## Container automatisch starten

Docker enthält keine Mechanismen oder Konfigurationsmöglichkeiten, um Container beim Hochfahren des Rechners automatisch zu starten. Wenn Sie das möchten, empfiehlt die Docker-Dokumentation das Einrichten von systemd-Dateien (siehe auch [Abschnitt 24.3](#), »Eigene Init-Scripts bzw. Init-Konfigurationsdateien«):

[https://docs.docker.com/engine/admin/host\\_integration](https://docs.docker.com/engine/admin/host_integration)

## docker-compose

Das Einrichten von Docker-Containern mit `docker run` ist zwar unkompliziert, aber wenn Sie immer wieder die gleiche Art von Containern benötigen, gibt es einen noch komfortableren Weg: Das Kommando `docker-compose` wertet die Textdatei `docker-compose.yml` im aktuellen Verzeichnis aus und richtet die entsprechenden Container ein. Es fehlt hier der Platz, auf `docker-compose` im Detail einzugehen, aber ich möchte Ihnen hier zumindest eine Kurzvorstellung des Programms bieten.

**Installation** Unter Linux ist `docker-compose` nicht Teil der Standard-Docker-Pakete. Die Installation ist aber unkompliziert:

```
root# curl -L https://github.com/docker/compose/releases/download/1.11.1/\
      docker-compose-$(uname -s)-$(uname -m) > /usr/bin/docker-compose
root# chmod +x /usr/bin/docker-compose
root# docker-compose --version
docker-compose version 1.11.1, build 7c5d5e4
```

Das Installationskommando für die gerade aktuellste Version von `docker-compose` finden Sie auf den folgenden Seiten:

<https://docs.docker.com/compose/install>

<https://github.com/docker/compose/releases>

Beachten Sie, dass ich die Installation in das Verzeichnis `/usr/bin` durchgeführt habe, während die gerade erwähnten Webseiten `/usr/local/bin` empfehlen. Allerdings ist das Verzeichnis `/usr/local/bin` je nach Distribution mitunter nicht in der Umgebungsvariablen `PATH` enthalten.

**docker-  
compose.yml**

Die Beschreibung der einzurichtenden Container erfolgt in der Datei `docker-compose.yml`, die Sie in einem Projektverzeichnis einrichten. Das folgende Listing illustriert die YAML-Syntax und entspricht inhaltlich größtenteils den Beispielen 4 und 5 aus dem [Abschnitt 41.2](#), »Docker kennenlernen«:

```
# Datei docker-compose.yml
version: '2'

services:
  db:
    image: mariadb:latest
    volumes:
      - /var/dc-test-db:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: geheim

  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    volumes:
      - /var/dc-test-www:/var/www/html
    ports:
      - "8082:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_PASSWORD: geheim
```

`mkdir` richtet die Volume-Verzeichnisse ein. `docker-compose up -d` sucht im aktuellen Verzeichnis nach der Datei `docker-compose.yml`, richtet gemäß der darin enthaltenen Anweisungen die Docker-Container ein und startet diese als Hintergrundprozesse (Option `-d`):

Container  
einrichten und  
löschen

```
root# mkdir /var/dc-test-www
root# mkdir /var/dc-test-db
root# docker-compose up -d
Creating network "composetest_default" with the default driver
Creating composetest_db_1
Creating composetest_wordpress_1
```

`docker-compose up` baut für die betreffenden Container automatisch ein eigenes Netzwerk auf. Damit sind die neuen Container von den restlichen Docker-Containern getrennt. Das Einrichten der Container (speziell die Initialisierung des MariaDB-Servers) dauert ein paar Sekunden. Danach können Sie die WordPress-Installation unter der Adresse `http://localhost:8082` ausprobieren.

Wenn Sie die Container und die dazugehörigen Volumes löschen möchten, führen Sie die folgenden Kommandos aus:

```
root# docker-compose down
root# rm -rf /var/dc-test-www /var/dc-test-db
```

## 41.4 Docker Images erzeugen und weitergeben

Sie können mit Docker auch eigene Images erzeugen und, einen Docker-Login vorausgesetzt, anderen Benutzern öffentlich zur Verfügung stellen. Kurz gefasst sieht die Vorgehensweise so aus: Sie richten in einem eigenen Verzeichnis die Datei `Dockerfile` ein. Dort schreiben Sie mit Schlüsselwörtern die Eigenschaften Ihres Images fest. `docker build` erzeugt das lokale Image. Mit `docker push` laden Sie es schließlich in die öffentliche Docker-Image-Sammlung hoch.

### Automatisierte GitHub-Builds, private Images und eigene Docker-Repositories

Aus Platzgründen gehe ich hier nur auf die einfachste Variante zum Einrichten und Hochladen von Docker-Images ein. Dabei befinden sich die erforderlichen Dateien in einem lokalen Verzeichnis und Sie laden das fertige Image manuell in den Docker Hub hoch.

Alternativen dazu sind einerseits die Verwendung eines GitHub-Projekts zur Speicherung des `Dockerfile`s und gegebenenfalls anderer Dateien (das ermöglicht automatisierte Builds), andererseits die Verwendung privater Image-Repositories auf <https://docker.com> (kostenpflichtig) bzw. das Einrichten eines eigenen Docker-Repositories. Weiterführende Informationen finden Sie hier:

<https://docs.docker.com/docker-hub/builds/#create-an-automated-build>  
<https://docs.docker.com/registry>

**Docker-Account** Sofern Sie vorhaben, Ihre Images öffentlich anzubieten, sollten Sie als Erstes auf der Webseite <https://hub.docker.com> einen Account einrichten. Sie müssen dazu lediglich einen Account-Namen, eine E-Mail-Adresse und ein Passwort angeben. Nach dem Anklicken der Bestätigungs-Mail melden Sie sich auf Ihrem Docker-Host mit `docker login` an:

```
root# docker login
Login with your Docker ID to push and pull images from Docker Hub.
Username: accountname
Password: *****
Login Succeeded
```

Mit `docker login` wird ein Authentifizierungs-Token in der Datei `.docker/config.json` gespeichert. Der Login bleibt damit unbegrenzt gültig.

### Das Dockerfile

Die Datei `Dockerfile` bestimmt die Eigenschaften des Images. [Tabelle 41.3](#) fasst die wichtigsten Schlüsselwörter zusammen. Eine Menge weiterer Schlüsselwörter und Details finden Sie in der offiziellen Dokumentation:



<https://docs.docker.com/engine/reference/builder>

Eine einfache Methode, um mit der Dockerfile-Syntax vertraut zu werden, besteht darin, sich im Docker Hub die Dockerfiles von Images anzusehen, die ähnliche Aufgaben erfüllen wie Ihr eigenes Image.

| Schlüsselwort | Bedeutung   |
|---------------|---|
| CMD           | führt das angegebene Kommando beim Container-Start aus.       |
| COPY          | kopiert Dateien aus dem Projektverzeichnis in das Image.      |
| ENTRYPOINT    | führt das angegebene Kommando immer beim Container-Start aus. |
| ENV           | setzt eine Umgebungsvariable.                                 |
| EXPOSE        | gibt die aktiven Ports des Containers an.                     |
| FROM          | gibt das Basis-Image an.                                      |
| LABEL         | legt eine Zeichenkette fest.                                  |
| RUN           | führt das angegebene Kommando aus.                            |
| USER          | gibt den Account für RUN, CMD und ENTRYPOINT an.              |
| VOLUME        | gibt Volume-Verzeichnisse an.                                 |

**Tabelle 41.3** Wichtige Dockerfile-Schlüsselwörter

Ein minimales Dockerfile, das das Ubuntu-Base-Image um das Paket des Editors joe erweitert, sieht so aus:

```
FROM ubuntu:16.04
LABEL maintainer "kontakt@kofler.info"
RUN apt-get update && \
    apt-get install -y joe
CMD ["/bin/bash"]
```

Mit RUN geben Sie Kommandos an, die einmalig beim Erstellen des neuen Images ausgeführt werden. Diese Kommandos werden in dem zur Image-Erzeugung vorübergehend eingerichteten Container ausgeführt, also im Gastsystem, nicht im Hostsystem. Oft handelt es sich dabei um Kommandos zur Installation von Paketen oder zum Kompilieren/Konfigurieren von Programmen.

**RUN**

Welches Programm wird ausgeführt, wenn ein Container mit run erstmalig bzw. später mit start ausgeführt wird? Auf diese Frage gibt es verwirrend viele Antworten:

**CMD versus  
ENTRYPOINT  
versus docker  
run/start/exec**

- ▶ Sie können im Dockerfile mit den Schlüsselwörtern CMD oder ENTRYPOINT ein Defaultkommando angeben.
- ▶ Beim Einrichten des Containers können Sie bei der CMD-Variante ein alternatives Kommando angeben oder bei der ENTRYPOINT-Variante dieses um weitere Parameter ergänzen.

- ▶ Gegebenenfalls können Sie auch das ENTRYPOINT-Kommando durch ein eigenes Kommando ersetzen, wenn Sie dieses mit der Option `--entrypoint` an `docker run` übergeben.
- ▶ Während ein Container läuft, können Sie mit `docker exec` ein beliebiges weiteres Kommando ausführen.

Die empfohlene Syntax für ENTRYPOINT bzw. CMD sieht so aus, dass Sie den vollständigen Dateinamen des Kommandos sowie seine Parameter jeweils in doppelte Anführungszeichen stellen und durch Kommata getrennt in eckigen Klammern übergeben:

```
CMD      ["/bin/ls", "/var"]
```

CMD und ENTRYPOINT scheinen dieselbe Aufgabe zu erfüllen. Es gibt aber einen entscheidenden Unterschied: Bei der CMD-Variante wird das Kommando, das Sie an `docker run` nach dem Image-Namen übergeben, *anstelle* von CMD ausgeführt. Bei der ENTRYPOINT-Variante wird dagegen immer das ENTRYPOINT-Kommando ausgeführt; die an `docker run` übergebenen Parameter werden diesem Kommando lediglich hinzugefügt.

Normalerweise geben Sie in Dockerfile entweder CMD oder ENTRYPOINT an, aber nicht beides. Tun Sie dies doch, dann wird das mit ENTRYPOINT formulierte Kommando ausgeführt (siehe [Tabelle 41.4](#)). Die in CMD angegebenen Schlüsselwörter werden als zusätzliche Parameter an das ENTRYPOINT-Kommando übergeben. Wenn Sie CMD oder ENTRYPOINT mehrfach angeben, dann gilt die letzte derartige Anweisung.

| ENTRYPOINT  | CMD         | run-Parameter | ausgeführt wird       |
|-------------|-------------|---------------|-----------------------|
| ["/bin/ls"] |             |               | /bin/ls               |
| ["/bin/ls"] |             | /bin/bash     | /bin/ls /bin/bash     |
| ["/bin/ls"] | ["a", "b"]  |               | /bin/ls a b           |
| ["/bin/ls"] | ["a", "b"]  | /bin/bash     | /bin/ls a b /bin/bash |
|             | ["/bin/ls"] |               | /bin/ls               |
|             | ["/bin/ls"] | /bin/bash     | /bin/bash             |

**Tabelle 41.4** Zusammensetzung des Kommandos, das durch »docker run« ausgeführt wird

Bei Linux-Base-Images lautet CMD zumeist `/bin/sh` oder `/bin/bash`. Bei Images für eine konkrete Aufgabe ist CMD in der Regel so eingestellt, dass das gewünschte Programm (z. B. der Webserver Apache) gestartet wird.

Beachten Sie, dass das Schlüsselwort RUN nichts mit CMD und ENTRYPOINT zu tun hat! RUN gibt Kommandos an, die einmalig beim Erzeugen des Images ausgeführt werden sollen. CMD bzw. ENTRYPOINT geben dagegen das Kommando an, das später beim Start des Containers auszuführen ist.

### Shell-Variante zu CMD und ENTRYPOINT

Die Docker-Dokumentation empfiehlt, das auszuführende Kommando und seine Parameter wie in den obigen Beispielen in eckigen Klammern und in doppelten Anführungszeichen an CMD bzw. ENTRYPOINT zu übergeben. Es gibt aber eine zweite Syntaxform, gemäß der Sie das Kommando einfach ohne Klammern und Apostrophe weitergeben, also z. B. `CMD ls /etc/*`. In diesem Fall wird das Kommando über eine Shell ausgeführt. In diesem Fall erhält nicht das Kommando die Prozess-ID 1, sondern die Shell.

Die Shell-Variante hat Vor- und Nachteile. Zu den Vorteilen zählt der Umstand, dass die von der Shell bekannten Substitutionsmechanismen funktionieren. Beispielsweise wird `*` durch Dateinamen ersetzt, `$VAR` durch den Inhalt der Umgebungsvariablen etc. Außerdem können Sie darauf verzichten, den vollständigen Pfad des Kommandos anzugeben – die Shell findet das Kommando, sofern es sich in einem der PATH-Verzeichnisse befindet.

Der größte Nachteil besteht darin, dass es keine Signalweiterleitung gibt: `Strg+C` bei einem interaktiven Container bzw. `docker stop` bei einem nicht interaktiven Container stoppen zwar die Shell, geben dem eigentlich auszuführenden Kommando aber keine Gelegenheit, die Signalverarbeitung selbst durchzuführen.

VOLUME gibt an, welche Verzeichnisse als Volumes im Dateisystem des Hosts abgebildet werden sollen. Wie bei CMD geben Sie mehrere Verzeichnisse in eckigen Klammern und jeweils in doppelten Anführungszeichen an:

VOLUME

```
VOLUME ["/var/lib/mysql" "/var/log/mysql"]
```

Wo die Volumes tatsächlich im Host-Dateisystem landen, hängt davon ab, wie der Container eingerichtet wird. Wenn Sie `docker run` oder `docker create` ohne die Option `-v` ausführen, richtet Docker für jedes Volume ein Verzeichnis mit einer zufälligen UID ein (`/var/lib/docker/volumes/uid`). Der Anwender des Images kann mit der Option `-v` den gewünschten Ort im Host auch selbst angeben:

```
root# docker run ... -v /myvolumes/mysql:/var/lib/mysql \
      -v /myvolumes/log:/var/log/mysql imagename
```

### Image erzeugen, hochladen und dokumentieren

Sie führen `docker build` üblicherweise in dem Verzeichnis aus, in dem sich Dockerfile befindet. Als Parameter übergeben Sie einen Punkt, mit dem Sie auf das Verzeichnis mit dem Dockerfile verweisen. Mit `-t` geben Sie den gewünschten Image-Namen an:

build-Kommando

```
root# cd projektverzeichnis
root# docker build -t accountname/imagename .
```

Wenn bei der Ausführung von `docker build` Fehler auftreten, korrigieren Sie das Dockerfile und wiederholen den Build-Prozess. Docker geht dabei recht intelligent vor: Es erzeugt in einem Cache-Verzeichnis für jede Anweisung im Dockerfile ein Interim-Image. Wenn nach einer Änderung am Dockerfile Anweisungen unverändert bleiben, kann `docker build` die entsprechenden Images aus dem Cache weiterverwenden. (Das können Sie bei Bedarf mit der Option `--no-cache` verhindern.)

Insofern ist es zum Debugging empfehlenswert, ein endloses, fehleranfälliges `RUN`-Kommando zu vermeiden und stattdessen mehrere `RUN`-Kommandos für jeden Teilschritt vorzusehen.

Diese Vorgehensweise kann leider zu Images führen, die größer sind als unbedingt notwendig. Das ist insbesondere dann der Fall, wenn in einem `RUN`-Kommando etwas installiert oder kompiliert wird und in einem zweiten `RUN`-Kommando Aufräumarbeiten durchgeführt werden. Docker ist nicht in der Lage, in einem Delta-Image hinzugefügte und in einem weiteren Delta-Image wieder gelöschte Dateien vernünftig aufzuräumen. Wenn Ihr Dockerfile also funktioniert, sollten Sie zusammengehörende Einzelkommandos wie im Beispiel auf der nächsten Seite wieder zusammenfügen.

`docker history imagename` liefert eine Auflistung der Interim-Images und der dabei ausgeführten Kommandos:

```
root# docker history accountname/imagename .
```

Oft bedarf es etlicher Versuche, bis das neue Image so funktioniert, wie es soll. Denken Sie daran, hin und wieder alle nicht benötigten Docker-Images zu löschen:

```
root# docker rmi $(docker images accountname/imagename -f dangling=true -q)
```

**push-Kommando** Nachdem Sie das Image lokal getestet haben, können Sie es zur öffentlichen Verteilung hochladen:

```
root# docker push accountname/imagename
```

**Image-Beschreibung**

Nach dem erfolgreichen Upload finden Sie das Image im Docker-Hub. In der Weboberfläche sollten Sie nun noch die Textfelder `SHORT` und `FULL DESCRIPTION` ausfüllen (siehe [Abbildung 41.4](#)).

Leider besteht keine Möglichkeit, diese Texte im Dockerfile anzugeben. Bei automatisierten Builds, bei denen Docker das Dockerfile und alle anderen Projektdaten von GitHub übernimmt, können Sie aber im Projektverzeichnis die Datei `README.md` vorsehen. Der Inhalt dieser Datei erscheint dann im Docker Hub als `FULL DESCRIPTION` Ihres Images. (Bei lokalen Builds funktioniert dieser Mechanismus nicht!)

**Tags** Standardmäßig erhält das neueste mit `docker build` erzeugte Image das Tag `latest`. Davon abweichende Tags können Sie wahlweise direkt bei der Ausführung von `docker build` oder nachträglich mit `docker tag` setzen:

```
root# docker build -t account/image:tag
root# docker tag account/image[:oldtag] account/image:newtag
```

Wenn Sie einmal in den Docker Hub hochgeladene Images wieder löschen möchten, wechseln Sie in der Webseite des jeweiligen Images in das Dialogblatt TAGS. Dort finden Sie für jede hochgeladene Version ein Mülleimer-Icon zum Löschen. Im Dialogblatt SETTINGS besteht außerdem die Möglichkeit, das gesamte Projekt zu löschen.

Hochgeladene  
Images löschen

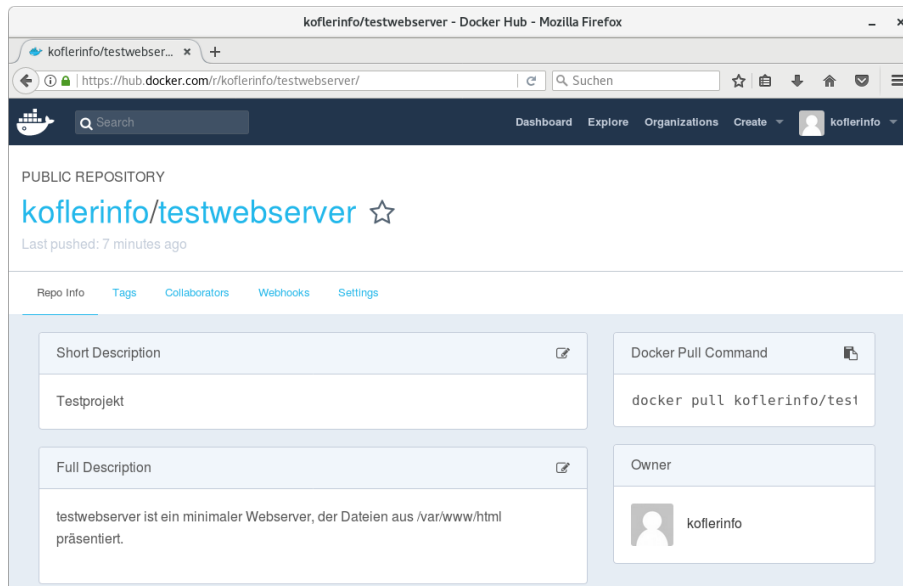


Abbildung 41.4 Präsentation eines eigenen Docker-Images auf dem Docker Hub

## Beispiel

Das Ziel dieses Beispiels ist es, ein Image für einen simplen Webserver zu erstellen. Die folgenden Zeilen zeigen das Dockerfile: Dockerfile

```
FROM ubuntu:16.04

LABEL maintainer "kontakt@kofler.info"
LABEL description "Test"

# Apache installieren, unnötige Dateien aus dem Paket-Cache
# gleich wieder entfernen
RUN apt-get update && \
    apt-get install -y apache2 && \
    apt-get -y clean && \
    rm -r /var/cache/apt /var/lib/apt/lists/*
```

```

# HTTPS-Unterstützung aktivieren
RUN a2ensite default-ssl && a2enmod ssl

ENV APACHE_RUN_USER=www-data \
    APACHE_RUN_GROUP=www-data \
    APACHE_LOG_DIR=/var/log/apache2

EXPOSE 80 443

# gesamten Inhalt des Projektverzeichnisses
# samplesite nach /var/www/html kopieren
COPY samplesite/ /var/www/html

CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]

```

Die Datei richtet ein Image ein, das auf dem Base Image von Ubuntu 16.04 basiert. Dort wird zusätzlich der Webserver Apache installiert. Dieser wird mit `a2ensite` und `a2enmod` auch für den HTTPS-Betrieb konfiguriert. Dementsprechend sollen die Ports 80 und 443 des Containers nach außen hin zugänglich sein.

**Projektdateien** Das lokale Verzeichnis `samplesite`, das sich im gleichen Verzeichnis wie das Dockerfile befindet, enthält die Dateien `index.html` und `style.css`. Der gesamte Inhalt dieses Verzeichnisses wird durch die `COPY`-Anweisung im Dockerfile in das Verzeichnis `/var/www/html` des Containers kopiert.

**Image erstellen und testen** `docker build` erzeugt nun das Image. Mit der Option `-t` können Sie dem Image gleich den gewünschten Namen und eventuell auch ein Tag zuordnen. Dabei müssen Sie natürlich `koflerinfo` durch den Namen Ihres Docker-Accounts ersetzen.

```

root# cd projektverzeichnis
root# docker build -t koflerinfo/testwebservice .
Step 1/9 : FROM ubuntu:16.04
---> 0ef2e08ed3fa
Step 2/9 : LABEL maintainer "kontakt@kofler.info"
---> Using cache
---> cd8be43178c9
Step 3/9 : LABEL description "Test"
Step 4/9 : RUN apt-get update && apt-get install -y apache2 && ...
...
Removing intermediate container 9dc0da8ffeeb
Successfully built 52df98fef2ec

```

Mit `docker run` erzeugen Sie vom lokalen Image einen Container und führen darin den Webserver aus. Wegen der Option `-d` läuft der Container im Hintergrund, bis Sie den Prozess mit `docker stop` beenden. `docker start name` startet den Container gegebenenfalls neu. `docker run` bzw. `docker create` setzen voraus, dass die Ports 80 und 443 auf dem lokalen Rechner frei sind. Sollte das nicht der Fall sein, z. B. weil dort ein loka-

ler Webserver läuft, müssen Sie andere lokale Ports angeben (z. B. durch die Optionen `-p 8080:80 -p 8443:443`).

```
root# docker run -d -p 80:80 -p 443:443 -h webtest \
    --name testwebserver koflerinfo/testwebserver
...
root# docker stop webtest    (im Hintergrund laufenden Container stoppen)
root# docker start webtest   (wieder starten)
```

In einem Webbrowser auf dem Docker-Host können Sie nun `http://localhost` besuchen und so die Testwebseite ansehen (siehe [Abbildung 41.5](#)).



**Abbildung 41.5** Der Webserver läuft in einem Docker-Container.

Die Adresse `https://localhost` führt zur HTTPS-Variante der Website. Zur HTTPS-Verschlüsselung wird ein automatisch erzeugtes, selbst signiertes »Snakeoil«-Zertifikat verwendet. Im Webbrowser wird deswegen eine entsprechende Warnung angezeigt. Gegebenenfalls müssen Sie ein richtiges Zertifikat einrichten (siehe [Abschnitt 32.4](#), »Verschlüsselte Verbindungen (HTTPS)«). Natürlich können Sie auf HTTPS auch einfach verzichten und den Port 443 des Containers nicht mit einem lokalen Port verbinden.

HTTPS

Das vorhin präsentierte Kommando eignet sich für erste Tests, aber nicht für den dauerhaften Betrieb des Webserver. Für diesen ist es nämlich zweckmäßig, den Container von den veränderlichen Daten zu trennen. Das betrifft einerseits die Webseite an sich und andererseits die Logging-Dateien des Webserver.

Volumes für das HTML- und Log-Verzeichnis

Abhilfe schaffen zwei lokale Verzeichnisse, von denen eines die HTML- und CSS-Seiten für die lokale Webseite aufnimmt und das andere die Logging-Dateien. Diese Verzeichnisse werden nun über die Option `-v` als Volumes an den Container weitergegeben und ersetzen so dessen lokale Verzeichnisse `/var/www/html` und `/var/log/apache2`. Im folgenden Beispiel gehe ich davon aus, dass sich die lokalen Verzeichnisse im Home-Verzeichnis befinden – aber jeder andere Ort ist ebenfalls denkbar.

```
root# docker run -d -p 80:80 -p 443:443 \
-v /home/kofler/webdir:/var/www/html \
-v /home/kofler/logdir:/var/log/apache2 \
-h webtest --name webtest koflerinfo/testwebserver
```

Der entscheidende Vorteil dieser Container-Konfiguration besteht darin, dass es nun jederzeit möglich ist, einen neuen Container auf Basis eines aktualisierten Images einzurichten. Wenn es also eine neue Apache-Version gibt, erzeugen Sie ein neues Image, stoppen den laufenden Container, richten einen neuen ein und starten ihn wie oben – fertig ist das Server-Update!

**Debugging** Wenn Sie dem laufenden Webserver quasi vom Inneren des Containers bei der Arbeit zusehen möchten, richten Sie mit `docker exec` parallel zum laufenden Container einen zweiten Prozess ein, in dem Sie eine interaktive Shell ausführen:

```
root# docker exec -it webtest /bin/bash
root@webtest# ps axu
USER      PID ...  COMMAND
root       1      /bin/sh /usr/sbin/apache2ctl -D FOREGROUND
root      15      /usr/sbin/apache2 -D FOREGROUND
www-data  16      /usr/sbin/apache2 -D FOREGROUND
www-data  17      /usr/sbin/apache2 -D FOREGROUND
root      72      /bin/bash
root      89      ps axu
```

**Image in den Docker-Hub hochladen** Sobald Sie mit der Funktion Ihres Images zufrieden sind, laden Sie es mit `docker push` hoch. Das setzt voraus, dass Sie sich irgendwann vorher mit `docker login` angemeldet haben und dass der Account-Name und der erste Teil des Images-Namens übereinstimmen.

```
root# docker push koflerinfo/testwebserver
```

Sobald das gelungen ist, kann nun jeder andere Docker-Benutzer das Image mit `docker run koflerinfo/testwebserver` verwenden.

## 41.5 Interna

**docker-Gruppe** Zur Steuerung von Containern muss das Kommando `docker` mit `root`-Rechten bzw. mit `sudo` ausgeführt werden. `dockerd` und `docker` kommunizieren über die Socket-Datei `/var/run/docker.sock` miteinander. Diese kann nur von `root` und von Mitgliedern der `docker`-Gruppe gelesen und beschrieben werden:

```
user$ ls -l /var/run/docker.sock
srw-rw----. 1 root docker ... /var/run/docker.sock
```

Die naheliegende Lösung besteht nun darin, einzelne Benutzer einfach der `docker`-Gruppe hinzuzufügen, beispielsweise so:



```
root# usermod -aG docker kofler (Vorsicht, Sicherheitsproblem!)
```

Damit kann kofler nun nach einem neuerlichen Login alle docker-Kommandos ohne sudo oder su ausführen.

### Vorsicht

Die Zuordnung zur docker-Gruppe ist bequem, gibt dem betreffenden Benutzer aber indirekt root-Rechte! Der Benutzer kann einen Container einrichten, der Zugriff auf das Wurzelverzeichnis / des Hosts hat, und kann so alle Sicherheitsmechanismen des Linux-Hosts aushebeln. Lesen Sie unbedingt die folgende Seite und speziell den Abschnitt *Docker daemon attack surface*:

<https://docs.docker.com/engine/security/security>

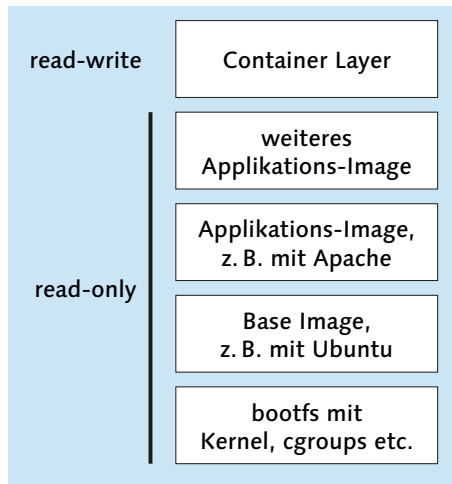
Auf einem Entwicklungsrechner kann das Hinzufügen eines Benutzers zur docker-Gruppe dennoch zweckmäßig sein; andernfalls besteht die Gefahr, dass alle Arbeiten (nicht nur die Ausführung von docker-Kommandos) als root erledigt werden. Das ist niemals wünschenswert.

## Overlay-Dateisystem

Eine zentrale Funktion von Docker ist der Umgang mit Images und der darauf aufbauenden Dateisysteme für die Container. Der Grundansatz ist einfach: Mehrere Images werden als Read-Only-Dateisysteme übereinandergelegt (siehe [Abbildung 41.6](#)). Die unterste Ebene bildet dabei das vom Docker-Host vorgegebene bootfs-Dateisystem. Es enthält den Kernel, den Device Mapper, die cgroups-Infrastruktur etc. Darauf setzt in der Regel ein sogenanntes Base Image auf, also ein Image, das ein minimales Linux-System enthält (z. B. Ubuntu). Weitere Images können dieses System nun um Applikationen ergänzen, z. B. um einen Webserver und um einen Editor.

Erst an der Spitze dieser Konstruktion befindet sich der sogenannte *Container Layer*. Alle Veränderungen an dem durch mehrere Images zusammengesetzten Dateisystem werden physikalisch im Container Layer gespeichert.

Zur tatsächlichen Implementierung des Overlay-Dateisystems stellt Linux mehrere Varianten zur Auswahl. In der Vergangenheit galt aufs (Advanced Multi-layered Unification Filesystem) als Docker-Standard, und bis heute kommt dieses Dateisystem auf manchen Distributionen zum Einsatz. Zunehmend populärer wird das overlay-Dateisystem bzw. dessen neue Variante overlay2. Wenn das Hostsystem btrfs einsetzt, kann auch dieses Dateisystem die Layering-Funktionen übernehmen.



**Abbildung 41.6** Das Dateisystem des Containers setzt sich aus Read-Only-Images und einem veränderlichen Container Layer zusammen.

Welches Overlay-Dateisystem bei Ihnen zum Einsatz kommt, können Sie mit `docker info` feststellen:

```
root# docker info | grep Storage    (CentOS)
Storage Driver: overlay
root# docker info | grep Storage    (Fedora, macOS)
Storage Driver: overlay2
root# docker info | grep Storage    (Ubuntu)
Storage Driver: aufs
```

`/var/lib/docker` Jetzt bleibt noch die Frage zu klären, wo die Docker-Images und Container Layer nun tatsächlich im Host-Dateisystem gespeichert werden. Grundsätzlich landen alle Docker-Daten im Verzeichnis `/var/lib/docker`. Dieses ist allerdings wieder in zahlreiche Unterverzeichnisse gegliedert, von denen ich hier nur zwei herausgreife:

- ▶ `/var/lib/docker/image` enthält Metadaten zu den Images.
- ▶ `/var/lib/docker/<overlay-driver>` enthält die ausgepackten Images, jeweils optimiert für das eingesetzte Overlay-Dateisystem (`aufs`, `overlay[2]`, `btrfs` etc.). Die Unterverzeichnisse `diffs` und `merged` enthalten für jeden Container die geänderten bzw. neuen Dateien des Container Layers.

Die Architektur des Overlay-Dateisystems ist sehr komplex, was sich auch in den zeilenlangen `mount`-Anweisungen widerspiegelt. (Führen Sie `mount | grep docker` aus, während ein Container läuft!) Viele Dateien und Verzeichnisse verwenden UIDs als Namen und sind entsprechend schwer zu lesen. Generell ist es das Beste, den gesamten Inhalt von `/var/lib/docker` als Blackbox zu betrachten und nicht anzurühren.

Wenn Sie aufräumen wollen, löschen Sie nicht mehr benötigte Container und Images mit `docker rm` bzw. `docker rmi`.

Wie groß ist eigentlich der Platzbedarf für Images bzw. für den Container Layer? Diese Frage beantworten die Kommandos `docker images` sowie `docker ps -a -s`.

Platzbedarf von  
Images und  
Containern

`docker images` liefert eine Liste aller heruntergeladenen bzw. selbst erzeugten Images samt Größenangabe:

```
root# docker images
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
koflerinfo/pandoc   latest     56d7c46781da 19 hours ago 1.73 GB
ibmcom/kitura-ubuntu latest     1210f73b5ed8 7 days ago   1.21 GB
mariadb             latest     56741a13bbb9 2 weeks ago  393 MB
ubuntu             16.04     0ef2e08ed3fa 2 weeks ago  130 MB
```

`docker ps` listet standardmäßig nur die laufenden Container auf. Die Option `-a` weitet die Ausgabe auf alle jemals eingerichteten Container aus. `-s` fügt die Spalte `SIZE` hinzu. Sie gibt an, wie viel Platz die gegenüber dem Image geänderten Dateien beanspruchen. Beachten Sie, dass der tatsächliche Platzbedarf im lokalen Dateisystem bei vielen kleinen Dateien aufgrund des Overheads des Dateisystems oft spürbar größer ist.

```
root# docker ps -a -s
CONTAINER ID   IMAGE                                ... NAMES          SIZE
cc522bf0ecba  ibmcom/kitura-ubuntu:latest        kituratest       14 MB (virtual 1.23 GB)
cbe544a5b7df  ubuntu                               myubuntu         21 B (virtual 130 MB)
...
```

Volumes befinden sich getrennt von den Containern und Images in `/var/lib/docker/volumes`. Beim Löschen von Containern rührt Docker Volumes generell nicht an. Sie können aber wie folgt eine Liste mit den IDs aller Volumes ermitteln, zu denen der dazugehörige Container nicht mehr existiert:

Volumes

```
root# docker volume ls -q -f dangling=true
4df85efbf1240b7429f7bf554e2ead52b90a1934875d57773c1c80c405ddd64a
6eec952744a21b55c71b8e6dc28da822bf3c8147ed54351dcb88c3009bb4eb1b
...
```

Die Größe der Volumes ermitteln Sie, indem Sie das obige Ergebnis an `du` weiterleiten:

```
root# du -h --max 0 \
    /var/lib/docker/volumes/$(docker volume ls -q -f dangling=true)
```

Alle verwaisten Volumes können Sie bei Bedarf so löschen. Die Rückfrage können Sie mit `-f` unterdrücken.

```
root# docker volume prune
WARNING! This will remove all volumes not used by at least one container.
Are you sure you want to continue? [y/N] y
```

## Prozessverwaltung

Zwischen virtuellen Maschinen und Containern gibt es gleich zwei elementare Unterschiede bei der Prozessverwaltung: Der eine betrifft die Anzahl der aktiven Prozesse, der zweite deren Kontrolle direkt durch die Host-Prozessverwaltung.

Je weniger  
Prozesse, desto  
besser

In Linux-Installationen auf physischer Hardware oder in virtuellen Maschinen laufen typischerweise Dutzende Prozesse gleichzeitig. In Docker-Containern ist das zwar nicht verboten, aber unüblich. Viele Container sind so konzipiert, dass beim Start genau ein Prozess ausgeführt wird – z. B. die `bash`, ein Web- oder ein Datenbank-Server.

Es gibt zwar keine strenge Regel *one process per container*, aber laut den Best Practices für das `Dockerfile` sollte jeder Container nur genau *eine* überschaubare Aufgabe übernehmen. Daraus ergibt sich dann fast automatisch eine sehr kleine Prozessanzahl.

[https://docs.docker.com/engine/userguide/eng-image/dockerfile\\_best-practices](https://docs.docker.com/engine/userguide/eng-image/dockerfile_best-practices)

Prozess-  
verwaltung

Docker-Prozesse laufen nicht in einer eigenen virtuellen Umgebung, sondern werden durch die Prozessverwaltung des Host-Systems verwaltet. Wenn Sie `ps -ax` auf dem Host-System ausführen, sind in der Prozessliste auch alle Docker-Prozesse enthalten. Besonders klar sehen Sie das, wenn Sie `ps tree` oder `ps axf` ausführen. Die folgende, stark gekürzte Ausgabe ist entstanden, während Docker zwei Container ausführte. Im einen Container lief Apache mit diversen Sub-Prozessen; parallel dazu wurde mit `docker exec` eine Shell ausgeführt. Im zweiten Container lief die `bash`, mit `docker exec` wurde parallel noch `top` ausgeführt.

```
root# ps axf
2601 ... /usr/bin/dockerd
2683     docker-containerd -l unix:///var/run/docker/libcontainerd/...
6109     docker-containerd-shim 853d... /var/run/docker/libcontainerd/853d...
6127     /bin/sh /usr/sbin/apache2ctl -D FOREGROUND
6163     /usr/sbin/apache2 -D FOREGROUND
6166     /usr/sbin/apache2 -D FOREGROUND
6167     /usr/sbin/apache2 -D FOREGROUND
6951     docker-containerd-shim 853d... /var/run/docker/libcontainerd/853d...
6968     /bin/bash
7136     docker-containerd-shim 3a4f... /var/run/docker/libcontainerd/3a4f...
7153     /bin/bash
7384     docker-containerd-shim 3a4f... /var/run/docker/libcontainerd/3a4f...
7401     /usr/bin/top
```

Ressourcen-  
management  
durch Control  
Groups

Grundsätzlich können Docker-Prozesse die gesamte CPU-Leistung oder den gesamten Arbeitsspeicher des Hosts in Anspruch nehmen. Das können Sie verhindern, wenn Sie beim Erzeugen eines Containers durch `docker run` Limits für den Speicher, die Anzahl der CPU-Cores etc. angeben. Die beiden wichtigsten Optionen lauten

-m 256m (der Container darf maximal 256 MiB nutzen) und --cpus="2.5" (der Container darf durchschnittlich 2,5 CPU-Cores auslasten). Zur Feinsteuerung gibt es unzählige weitere Optionen, die hier dokumentiert sind:

[https://docs.docker.com/engine/admin/resource\\_constraints](https://docs.docker.com/engine/admin/resource_constraints)

Hinter den Kulissen verwendet Docker Control Groups (*cgroups*), um die Einhaltung dieser Limits sicherzustellen.

Docker-Container sind zueinander und gegenüber dem Host viel weniger gut isoliert als virtuelle Maschinen. Das heißt aber nicht, dass es gar keine Trennung gibt:

Container-  
Isolierung durch  
Namespaces

- ▶ Docker stellt durch sogenannte Namespaces sicher, dass jeder Container eigene UIDs, GIDs und PIDs hat (also eigene User-, Gruppen- und Prozessnummern) und somit die Prozesse des Hosts oder anderer Container nicht sieht.
- ▶ Mount-Namespaces vermitteln jedem Container seine eigene Sichtweise auf das Dateisystem. Der Container kann außer seinem bereits beschriebenen Overlay-Dateisystem und eventuell gemeinsam genutzten Volumes nicht auf andere Verzeichnisse des Host-Dateisystems zugreifen. Mount-Namespaces sind auf Kernebene implementiert und sicherer als eine *chroot*-Umgebung.
- ▶ Schließlich erhält jeder Container seinen eigenen Netzwerk-Stack.

### Sicherheitsrisiko Docker

Trotz aller Maßnahmen zur Container-Isolierung muss Ihnen bewusst sein, dass Docker bei Weitem nicht mit den Sicherheitsmodellen virtueller Maschinen mithalten kann (und selbst dort gab es in der Vergangenheit immer wieder Probleme). Solange Docker nur für Entwicklungs- und Test-Aufgaben verwendet wird, spielt das eine untergeordnete Rolle. Wenn Docker aber für Server-Dienste im Produktionseinsatz verwendet wird, ist größte Vorsicht angebracht!

Besonders problematisch ist der Umstand, dass Docker-Prozesse oft mit *root*-Rechten ausgeführt werden. Das ist nicht immer zwingend erforderlich, aber es ist aus Entwicklersicht die einfachste und bequemste Lösung. Aktuell machen sich leider nur recht wenige *Dockerfile*-Autoren die Mühe, ihre Images sicherheitstechnisch zu optimieren. (Eine positive Ausnahme stellen die meisten offiziellen Images dar.)

Eine Möglichkeit zur Verbesserung der Sicherheit bietet das Schlüsselwort *USER*. Es bestimmt, in welchem Account das Docker-Kommando ausgeführt wird.

<http://blog.dscpl.com.au/2015/12/don-run-as-root-inside-of-docker.html>

<http://blog.dscpl.com.au/2016/12/what-user-should-you-use-to-run-docker.html>

Eine Menge weiterer Details zur Container-Isolation und zu anderen Sicherheitsthemen können Sie auf den folgenden Seiten nachlesen:

<https://docs.docker.com/engine/security/security>

[https://success.docker.com/KBase/Introduction\\_to\\_User\\_Namespaces\\_in\\_Docker\\_Engine](https://success.docker.com/KBase/Introduction_to_User_Namespaces_in_Docker_Engine)

<https://security.stackexchange.com/questions/107850>

<https://integratedcode.us/2015/10/13/user-namespaces-have-arrived-in-docker>

## Netzwerkverwaltung

Standardmäßig laufen Docker-Container in einem privaten Netzwerk, um dessen Verwaltung sich Docker selbstständig kümmert. Einen Überblick über die aktuelle Netzwerkkonfiguration erhalten Sie mit den Kommandos `docker network ls` und `docker network inspect`:

```
root# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
3646ca641eed       bridge             bridge              local
812c2a539b16       host               host                local
9ea2794fb2df       none              null                local
root# docker network inspect bridge (Ausgabe stark gekürzt)
"Subnet": "172.17.0.0/16",
"Gateway": "172.17.0.1"
...
"Containers":
  "Name": "pma",
  "IPv4Address": "172.17.0.4/16",
  ...
  "Name": "mariadb-test4",
  "IPv4Address": "172.17.0.2/16",
  ...
```

Wenn Sie das Docker-Netzwerk über die Defaultkonfiguration hinaus adaptieren möchten, liefert die folgende Seite ausführliche Hintergrundinformationen:

<https://docs.docker.com/engine/userguide/networking>

# Kapitel 42

## Linux on Windows

Schon seit Jahrzehnten gibt es die Möglichkeit, Linux in einer virtuellen Maschine unter Windows auszuführen. Das Angebot von Virtualisierungssystemen ist in den letzten Jahren immer größer geworden, und auch das im vorigen Kapitel vorgestellte Programm Docker bietet die Möglichkeit, einzelne Prozesse in Linux-Containern unter Windows auszuführen.

Dessen ungeachtet hat Microsoft im Frühjahr 2016 eine weitere Variante vorgestellt: das *Windows Subsystem for Linux*, kurz WSL. Bemerkenswert ist, dass Linux in diesem Fall ohne Virtualisierungstechnik läuft. Windows wurde dazu um neue Bibliotheken ergänzt, insbesondere um LXSS, LXCode und DriveFS. Damit gelingt es, ein nahezu originales Ubuntu-Server-System unter Windows auszuführen – ohne Hypervisor und ohne Neukompilierung der Programme! (Die Ausführung von Programmen im Grafikmodus ist nicht vorgesehen.)

So faszinierend WSL aus technischer Sicht ist, so stellt sich doch die Frage, was eigentlich die Zielsetzung ist. Eine schlüssige Antwort dazu ist Microsoft bisher schuldig geblieben – und auch ich bin mir darüber nicht ganz im Klaren. Vielleicht liegt es daran, dass WSL nach wie vor mit dem Attribut *Beta* gekennzeichnet ist und noch keinen vollständig ausgereiften Eindruck macht. Für alle, die primär unter Linux, aber hin und wieder auch unter Windows arbeiten, ist es natürlich ein riesiger Komfortgewinn, dass nun neben `cmd.exe` und der PowerShell auch die `bash` mit Kommandos wie `find`, `grep`, `ssh` und `scp` ohne Weiteres zur Verfügung steht.

Innerhalb der Linux-Gemeinde war natürlich auch die Wahl von Ubuntu als Basissystem umstritten. Das soll sich aber im Herbst 2017 ändern. Im Windows Store sollen dann auch WSL-Varianten für openSUSE und Fedora zum kostenlosen Download angeboten werden. Microsoft verspricht, dass mehrere WSL-Varianten parallel installiert und gleichzeitig ausgeführt werden können.

WSL ist eine der spannendsten Linux-Neuentwicklungen der letzten Jahre und insofern sicher ein kurzes Kapitel wert. Ich gebe Ihnen darin eine Einführung in den Umgang mit WSL auf Basis der im Frühjahr 2017 verfügbaren Ubuntu-Variante und erläutere einige technische Hintergründe und Interna.

Weitere Details können Sie im Internet auf den folgenden Seiten nachlesen:

<https://msdn.microsoft.com/commandline/wsl>

<https://blogs.msdn.microsoft.com/wsl>

<http://blog.dustinkirkland.com/2016/08/howdy-windows-six-part-series.html>

<https://wsl-forum.qztc.io> (inoffizielles Forum)

## 42.1 WSL ausprobieren

**Installation** WSL setzt eine 64-Bit-Version von Windows 10 voraus. Ich habe meine Tests mit Windows 10, Version 1703 durchgeführt, also nach der Installation des »Windows 10 Creator Updates«. Zur Installation müssen Sie zuerst den Entwicklermodus aktivieren. Dazu öffnen Sie das Modul UPDATES UND SICHERHEIT der Systemeinstellungen. Die relevanten Einstellungen befinden sich im Dialogblatt FÜR ENTWICKLER.

Als Nächstes suchen Sie im Startmenü das Programm WINDOWS-FEATURES. In diesem Programm aktivieren Sie die Option WINDOWS-SUBSYSTEM FÜR LINUX (BETA) (siehe [Abbildung 42.1](#)). Nach dem obligatorischen Windows-Neustart finden Sie im Startmenü den neuen Eintrag BASH. Im Terminalfenster müssen Sie die Lizenzbedingungen von Canonical akzeptieren, bevor der Download der Ubuntu-Pakete aus dem Windows Store beginnt.

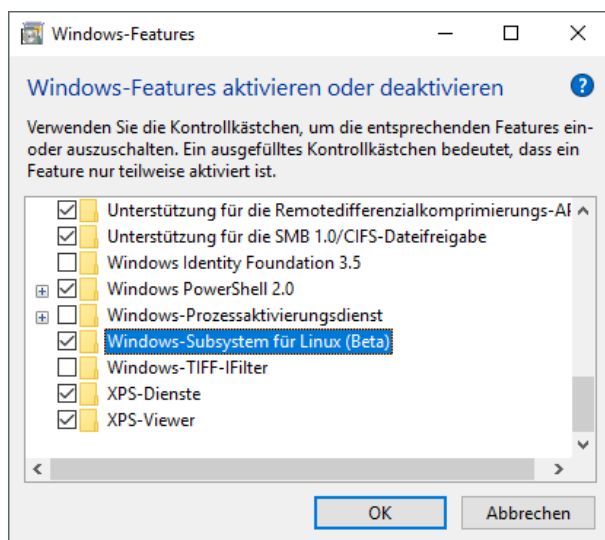


Abbildung 42.1 WSL installieren



Sobald der Download abgeschlossen ist, fragt das Installationsprogramm, ob es die deutsche Lokalisierung aktivieren soll. Es spricht nichts dagegen. Jetzt muss noch ein Ubuntu-Benutzer samt Passwort eingerichtet werden. Dieser Benutzer hat sudo-Rechte.

Benutzer  
einrichten

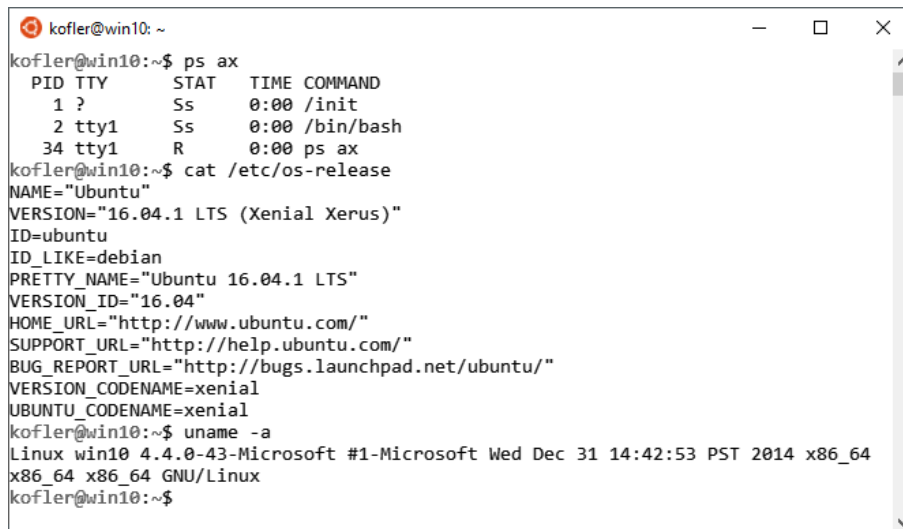
Wenn es auf Ihrem Windows-Rechner mehrere Konten gibt, deren Benutzer WSL nutzen möchten, muss die Installation für jedes Konto wiederholt werden. Jede WSL-Installation gilt nur für das betreffende Windows-Konto.

WSL-Installation  
für mehrere  
Windows-Nutzer

### Arbeiten in der bash

Nach dem Start der `bash` sind Sie standardmäßig in dem Account eingeloggt, den Sie bei der Installation angegeben haben. Erfreulicherweise funktionieren die wichtigsten Tastenkürzel wie unter Linux: `[Strg]+[D]` löscht das aktuelle Zeichen, `[Strg]+[A]` bewegt den Cursor an den Beginn der Zeile etc.

Sie können nun die neue Umgebung erkunden (siehe [Abbildung 42.2](#)). Die Datei `/etc/os-release` verrät die installierte Ubuntu-Version. `uname -a` schwindelt ein wenig: WSL kann zwar für Linux kompilierte Programme direkt ausführen, es läuft aber kein richtiger Kernel.



```

kofler@win10: ~
kofler@win10:~$ ps ax
  PID TTY          STAT       TIME COMMAND
    1 ?           Ss          0:00 /init
    2 tty1       Ss          0:00 /bin/bash
   34 tty1       R           0:00 ps ax
kofler@win10:~$ cat /etc/os-release
NAME="Ubuntu"
VERSION="16.04.1 LTS (Xenial Xerus)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 16.04.1 LTS"
VERSION_ID="16.04"
HOME_URL="http://www.ubuntu.com/"
SUPPORT_URL="http://help.ubuntu.com/"
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"
VERSION_CODENAME=xenial
UBUNTU_CODENAME=xenial
kofler@win10:~$ uname -a
Linux win10 4.4.0-43-Microsoft #1-Microsoft Wed Dec 31 14:42:53 PST 2014 x86_64
x86_64 x86_64 GNU/Linux
kofler@win10:~$

```

Abbildung 42.2 Erste Experimente im bash-Fenster

Die mit `ps ax` erstellte Prozessliste sieht ähnlich trist wie in Docker-Containern aus (siehe das vorige Kapitel): Außer der `bash` läuft lediglich ein `Init`-Prozess. Der zugrundeliegende `systemd` ist allerdings nur sehr eingeschränkt verwendbar: Insbesondere

systemd mit  
Einschränkungen


lässt sich `systemctl` nicht aufrufen, sondern liefert nur die Fehlermeldung *failed to connect to bus*.

#### Mehrere bash-Fenster

Sie können mehrere `bash`-Fenster gleichzeitig öffnen. Diese teilen sich einen gemeinsamen Prozessraum und den `Init`-Prozess. Wie im »echten« Linux sehen Sie mit `ps ax` oder `top` auch Prozesse, die in anderen Fenstern gestartet wurden, und können diese bei Bedarf mit `kill` beenden. (`bash`-Fenster sind also nicht vergleichbar mit getrennten Docker-Containern.)

#### Text kopieren und einfügen

Das `bash`-Fenster ist offensichtlich von `cmd.exe` abgeleitet. Das unter Linux vertraute Kopieren von Text mit der Maus bzw. dem Trackpad und das anschließende Einfügen mit der mittleren Maus-/Trackpad-Taste funktioniert so nicht.

`cmd.exe`-Experten wissen aber, dass es stattdessen eine andere Vorgehensweise gibt: Mit der Maustaste markieren Sie einen Textblock.  kopiert den Text in die Zwischenablage. Mit der rechten Maustaste fügen Sie diesen Text an der aktuellen Cursorposition wieder ein.

#### Licht ...

In der `bash` stehen alle elementaren Kommandos zum Umgang mit Dateien inklusive `find` und `grep` zur Verfügung. Ebenso können Sie alle gängigen Kommandos zur Paketverwaltung verwenden, also insbesondere `dpkg` und `apt`. Sie können neue Pakete installieren und Updates durchführen. Standardmäßig sind beachtliche 440 Pakete installiert. (Zum Vergleich: Eine minimale Ubuntu-Server-Installation umfasst 480 Pakete.)

#### Benutzerverwaltung

Problemlösung funktioniert die Verwaltung von Benutzern und Gruppen. Sie können mit `adduser` neue Benutzer einrichten, mit `passwd` deren Passwörter zurücksetzen, mit `visudo` die Sudo-Konfiguration ändern etc. Beim Start eines `bash`-Fensters kommt zwar immer der Default-Account zur Anwendung, Sie können mit `su -l` aber den aktiven Benutzer wechseln.

#### Editoren

Per Default stehen die Editoren `nano` und `vi` zur Auswahl. Emacs-Fans können wahlweise das Paket `joe` oder `emacs-nox` installieren.

#### Sonstige Kommandos

Im Folgenden habe ich ohne Anspruch auf Vollständigkeit einige weitere funktionierende Kommandos aufgezählt: `df`, `ifconfig`, `ip`, `less`, `locate`, `man`, `mount`, `ping`, `route`, `rsync`, `scp` und `ssh`.

## ... und Schatten

Standardmäßig sind zwar eine Menge Netzwerkkommandos installiert, viele davon sind aber nur eingeschränkt verwendbar. Das betrifft z. B. `iptables`: Das Kommando kann nicht verwendet werden, weil die normalerweise vom Kernel zur Verfügung gestellte Infrastruktur fehlt. Eine detaillierte Beschreibung des Netzwerk-Stacks von WSL finden Sie hier:

Netzwerk-  
funktionen

<https://blogs.msdn.microsoft.com/wsl/2016/11/08/225>

Ungewöhnlich leer sind die Verzeichnisse `/dev`, `/proc` und `/sys`. Das liegt daran, dass WSL ohne Kernel keine eigene Hardware-Verwaltung hat. Die Hardware wird von Windows übernommen, ein Low-Level-Zugriff auf Partitionen, USB-Geräte etc. ist nicht vorgesehen. Hardware-nahe Kommandos wie `lsblk` oder `lspci` stehen nicht zur Verfügung.

Kein Hard-  
ware-Zugriff

WSL unterstützt offiziell keine Hintergrunddienste. Grundsätzlich können nur Kommandos ausgeführt werden, die mit dem Terminalfenster verbunden sind. Natürlich kann niemand Sie daran hindern, einen Prozess im Hintergrund zu starten. Es muss Ihnen aber klar sein, dass alle Prozesse enden, sobald Sie das `bash`-Fenster schließen.

Keine Hinter-  
grunddienste

Naturgemäß schränkt das die Anwendung von WSL enorm ein: Sie können zwar alle erdenklichen Programmiersprachen installieren und damit Programme entwickeln, aber die Ausführung von Server-Diensten ist nicht vorgesehen. Ich werde Ihnen aber gleich zeigen, wie Sie das Problem mit etwas Handarbeit umgehen und z. B. einen SSH-Server dauerhaft einrichten können.

Eigentlich sollte es nun schon klar sein: Da Cron, Syslog und das Journal auf gewöhnlichen Linux-Systemen als Hintergrunddienste realisiert sind, müssen Sie darauf in WSL verzichten. Mangels Kernel funktioniert natürlich auch `dmesg` nicht.

Kein Cron,  
kein Syslog,  
kein Journal

## Dateisystem

Um von Linux aus auf das Windows-Dateisystem zuzugreifen, verwenden Sie das Verzeichnis `/mnt/c`. Das Verzeichnis wird über den WSL-spezifischen `drvfs`-Treiber in den Verzeichnisbaum eingebettet (siehe `mount`). Sie haben aus der `bash` Lese- und Schreibrechte im Windows-Verzeichnisbaum (siehe [Abbildung 42.3](#)).

Windows-  
Dateisystem

Alle Dateien des Windows-Dateisystems haben aus WSL-Sicht den Besitzer `root` und die Gruppe `root`. Das eigene Heimatverzeichnis ist mit Schreibrechten ausgestattet, viele andere Dateien zumindestens mit Leserechten. Fremde Benutzerverzeichnisse sowie manche Systemdateien und -verzeichnisse sind auch lesegeschützt:

```

kofler$ ls -l /mnt/c/Users/
dr-xr-xr-x 0 root root  ... Default
-r-xr-xr-x 1 root root  desktop.ini
drwxrwxrwx 0 root root  kofler
d--x--x--x 0 root root  otheruser
...

kofler$ ls -l /mnt/c/
Zugriff auf '/mnt/c/hiberfil.sys' nicht möglich: Keine Berechtigung
Zugriff auf '/mnt/c/pagefile.sys' nicht möglich: Keine Berechtigung
Zugriff auf '/mnt/c/swapfile.sys' nicht möglich: Keine Berechtigung
-r-xr-xr-x 1 root root  ... bootmgr
-r-xr-xr-x 1 root root  BOOTNXT
drwxrwxrwx 0 root root  OneDriveTemp
d--x--x--x 0 root root  PerfLogs
drwxrwxrwx 0 root root  ProgramData
dr-xr-xr-x 0 root root  Program Files
dr-xr-xr-x 0 root root  Program Files (x86)
d--x--x--x 0 root root  System Volume Information
dr-xr-xr-x 0 root root  Users
dr-xr-xr-x 0 root root  Windows
...

```

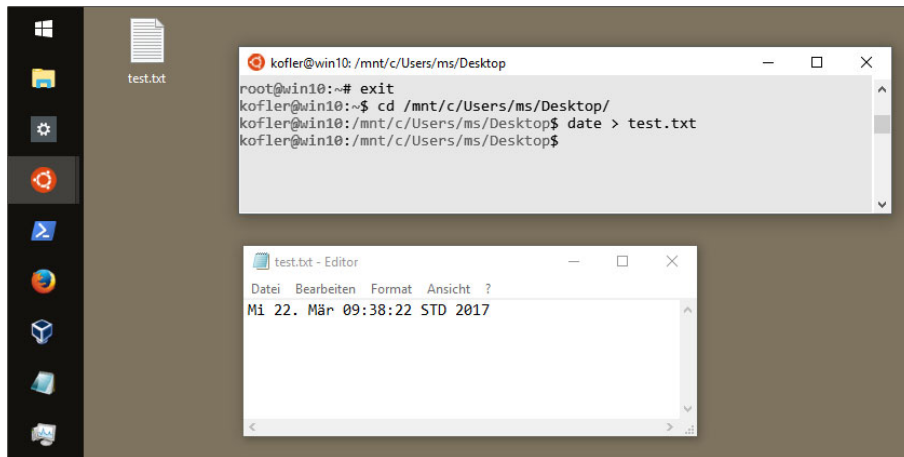


Abbildung 42.3 Die Datei test.txt auf dem Desktop wurde in der bash erzeugt.

#### Linux-Dateisystem

Wo befindet sich das Linux-Dateisystem aus Windows-Sicht? Wenn Sie im Windows Explorer die Option GESCHÜTZTE SYSTEMDATEIEN AUSBLENDEN deaktivieren (doppelte Verneinungen sind immer schön), dann finden Sie das Linux-Root-Dateisystem sowie weitere Verzeichnisse für home, root etc. am folgenden Ort (siehe [Abbildung 42.4](#)):

```
C:\Users\\AppData\Local\lxss
```

Microsoft rät dringend davon ab, Dateien innerhalb dieses Verzeichnisses von Windows aus zu verändern. Linux-Dateien dürfen nur durch bash-Prozesse angerührt werden!

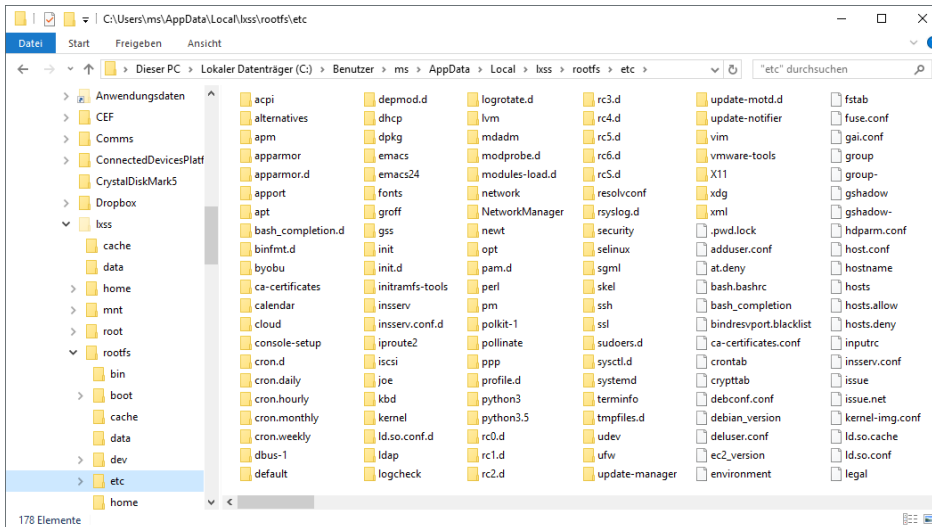


Abbildung 42.4 Das Linux-Dateisystem befindet sich im AppData-Verzeichnis des Windows-Nutzers.

## Reset

Sollten Sie das Ubuntu-System innerhalb von WSL zerstört haben (z. B. durch `rm -rf /etc`), starten Sie `cmd.exe` und führen dort das folgende Kommando aus:

```
cmd.exe> lxrund /uninstall /full
```

Nach einer Rückfrage wird das Dateisystem für Ubuntu gelöst. Damit gehen natürlich auch alle von Ihnen durchgeführten Änderungen verloren. Anschließend starten Sie im Windows-Startmenü neuerlich `bash` und initiieren so eine Neuinstallation.

## 42.2 Serverbetrieb

Ich habe es bereits gesagt: WSL ist eigentlich nicht für den Server-Betrieb gedacht. Hintergrundprozesse werden nur über Umwege unterstützt, ihr automatischer Start ist schwer möglich. Andererseits macht Linux ganz generell, aber auch im Kontext vieler Entwicklungsaufgaben ohne Server-Dienste wenig Sinn. Dieser Abschnitt zeigt anhand von zwei Beispielen (SSH und LAMP), welche Möglichkeiten WSL in dieser Hinsicht trotz aller Einschränkungen bietet.

## SSH-Server einrichten

In WSL ist standardmäßig ein SSH-Server installiert. Er läuft aber nicht, und seine Inbetriebnahme ist ein wenig umständlicher als bei »richtigen« Linux-Distributionen. Als Erstes müssen Sie die für die sichere Kommunikation erforderlichen Schlüssel manuell einrichten:

```
root# dpkg-reconfigure openssh-server
```

Anschließend müssen Sie in `/etc/ssh/sshd_config` eine Zeile ändern, um die Authentifizierung per Passwort zu erlauben:

```
# in /etc/ssh/sshd_config
...
PasswordAuthentication yes
```

Jetzt können Sie den SSH-Dienst starten. Das Kommando `systemctl` funktioniert nicht, aber zum Glück gibt es als Alternative das Kommando `service`:

```
root# service ssh start
```

### Port-Konflikt mit dem SSH-Server von Windows

Windows 10 verfügt seit Build 14352 über einen eigenen SSH-Server, der noch weitgehend unbekannt und in der Regel auch nicht aktiv ist. Wenn er aber läuft, dann kann der SSH-Server von WSL nicht gestartet werden, weil Port 22 blockiert ist. Dafür gibt es zwei Lösungen: Entweder verhindern Sie den Start des Windows-eigenen SSH-Servers im Programm DIENSTE oder Sie weisen dem WSL-SSH-Server in `/etc/ssh/sshd_config` mit `Port nn` einen anderen, freien Port zu (etwa 2222). Informationen zum Windows-eigenen SSH-Server finden Sie hier:

<https://noise.paulos.cz/post/windows10-14352-ssh-server>

Lokal, also z. B. aus einem zweiten `bash`-Fenster, können Sie nun SSH testen. Dabei sollten keine Probleme auftreten. Beachten Sie, dass Sie als SSH-Login-Name nicht den Namen eines Windows-Kontos, sondern den Namen eines Users in WSL angeben müssen. Sie stellen ja genau genommen keine SSH-Verbindung zu Windows her, sondern eine zu Ubuntu in WSL.

### Windows-Firewall

Den Verbindungsaufbau durch einen anderen Rechner verhindert die Windows-Firewall. Abhilfe schafft eine neue Regel für den Port 22. Dazu starten Sie die Systemeinstellungen und öffnen das Modul WINDOWS-FIREWALL. Der Link ERWEITERTE EINSTELLUNGEN führt in ein weiteres Programm mit den Detailsinstellungen der Firewall. Mit AKTION • NEUE REGEL gelangen Sie in einen Assistenten. Dort wählen Sie den REGELTYP = PORT aus, geben die Port-Nummer 22 (TCP) an, aktivieren die Option VERBINDUNG ZULASSEN und geben der Regel zuletzt einen Namen, z. B. SSH. Sollten Sie

unter Windows ein anderes Firewall-Programm installiert haben (z. B. als Bestandteil eines Virencanners), müssen Sie sich mit dessen Konfiguration auseinandersetzen.

Solange das `bash`-Fenster offen ist, können Sie sich nun von jedem Linux- oder macOS-Rechner im gleichen Netzwerk per SSH an Ihrem Windows-Rechner anmelden (siehe [Abbildung 42.5](#)). Beim ersten Mal ist das ein erhebendes Gefühl – »Windows gezähmt« gewissermaßen. Und wie ungemein praktisch ist es, eine Datei von einem Linux-Rechner ohne Netzwerkverzeichnisse, DropBox etc. einfach in einen Windows-Rechner kopieren zu können!

```
user@linuxhost$ scp lokaledatei <wsuser>@win10:/mnt/c/Users/winuser/Desktop
```

```

kofler@win10: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
[kofler@localhost ~]$ ssh kofler@win10
kofler@win10's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-43-Microsoft x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 Software-Pakete können aktualisiert werden.
0 Aktualisierungen sind Sicherheitsaktualisierungen.

*** Neustart des Systems erforderlich ***
Last login: Wed Mar 22 13:55:13 2017 from 10.0.0.52
kofler@win10:~$

```

Abbildung 42.5 SSH-Login an einem Windows-Rechner

### Sicherheitsrisiko SSH

Jeder SSH-Server ist ein Sicherheitsrisiko, aber in WSL gilt dies ganz besonders: Gewöhnliche Benutzer dürfen viele Dateien des Windows-Dateisystems (`/mnt/c`) lesen und teilweise auch verändern! Stellen Sie unbedingt sicher, dass alle im WSL eingerichteten Benutzer sichere Passwörter haben!

So schön der SSH-Server unter Windows ist – er läuft nur, bis Sie das `bash`-Fenster wieder schließen. Komfortabler wäre es, wenn der SSH-Server sofort beim Login automatisch gestartet würde. Mit etwas Kreativität und viel Arbeit ist auch das möglich.

SSH-Server  
automatisch  
starten

Als Erstes richten Sie in WSL, z. B. im Verzeichnis `/root`, das folgende Shell-Script ein:

```
#!/bin/bash
# Datei /root/start-sshd
/etc/init.d/ssh start
sleep 86400000
```

Das Script startet den SSH-Server über ein Init-Script. Das nachfolgende `sleep`-Kommando garantiert, dass das Script 1000 Tage läuft. Das ist deswegen notwendig, weil mit dem andernfalls sofortigen Ende des Scripts auch die WSL-bash und damit der gerade gestartete SSH-Server wieder beendet würden.

Im zweiten Schritt bauen Sie in `/etc/sudoers` die folgende Zeile ein, wobei Sie `kofler` durch den Namen Ihres Ubuntu-Accounts ersetzen:

```
# Ergänzung in /etc/sudoers
...
kofler ALL = (root) NOPASSWD: /root/start-sshd
```

Die Zeile erlaubt es dem Nutzer `kofler`, das Script `start-sshd` mit `sudo` ohne Passwortangabe zu starten.

Der dritte Schritt erfolgt nicht mehr in WSL, sondern unter Windows. Dort verfassen Sie mit einem Editor (sogar `notepad.exe` reicht aus) in Ihrem Heimatverzeichnis das zweizeilige Script `start-sshd.vbs` mit folgendem Inhalt:

```
set ws=wscript.createobject("wscript.shell")
ws.run "C:\Windows\System32\bash.exe -c 'sudo /root/start-sshd'", 0
```

Beenden Sie die WSL-bash, und probieren Sie dann das Script per Doppelklick aus: Es ist zwar am Bildschirm keine Reaktion zu sehen, der SSH-Server sollte aber laufen. Sie können sich davon mit `ps ax` in einem neuerlich gestarteten bash-Fenster oder natürlich durch einen SSH-Login überzeugen.

Nun müssen Sie nur noch erreichen, dass dieses Script beim Start von Windows automatisch ausgeführt wird. Dazu starten Sie das Programm AUFGABENPLANUNG (bei englischen Windows-Versionen den TASK SCHEDULER). Dort richten Sie mit AKTION • EINFACHE AUFGABE ERSTELLEN einen neuen Job ein. Im Assistenten geben Sie dem Job einen Namen, wählen dann die Option BEIM ANMELDEN aus (nicht BEIM START DES COMPUTERS, das funktioniert nicht), belassen im dritten Dialogblatt die schon aktivierte Option AUFGABE und wählen zuletzt das vorhin eingerichtete `*.vbs`-Script aus (siehe [Abbildung 42.6](#)).

### SSH-Server-Start beim Hochfahren

Noch eleganter wäre es natürlich, wenn der SSH-Server nicht erst nach dem Login, sondern gleich beim Hochfahren des Rechners gestartet würde. Das ist offensichtlich aktuell aus technischen Gründen unmöglich, die für mich als Windows-Laien schwer nachvollziehbar sind. Details können Sie hier nachlesen:

<https://github.com/Microsoft/BashOnWindows/issues/612>  
<https://wsl-forum.qztc.io/viewtopic.php?f=6&t=10>



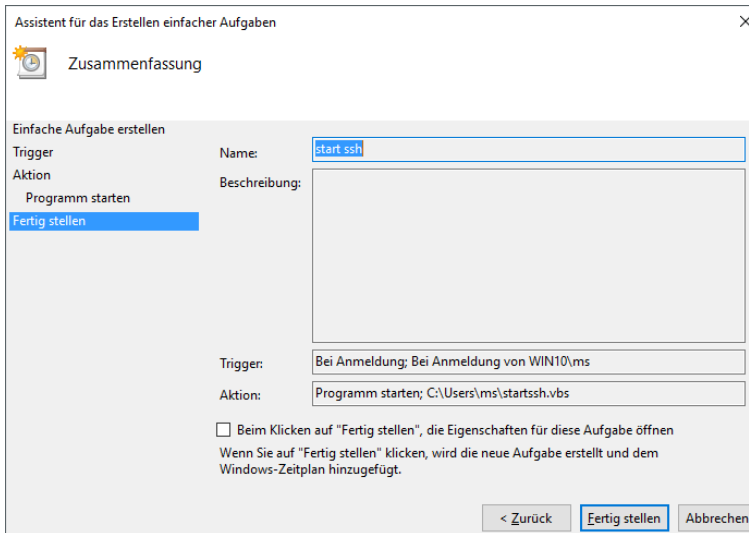


Abbildung 42.6 Das Script startssh.vbs soll beim Login automatisch ausgeführt werden.

## LAMP-Server einrichten

Das Kürzel LAMP steht für *Linux, Apache, MySQL* und *PHP*. Die Kombination dieser Programme ist ein bewährtes Fundament für PHP-affine Webentwickler. Die Installation führen Sie am schnellsten mit den beiden folgenden Kommandos aus:

```
root# apt install tasksel
root# tasksel install lamp-server
```

Während der Installation müssen Sie das root-Passwort für den MySQL-Server festlegen. Anders als in »richtigen« Ubuntu-Systemen werden aber weder der Web- noch der MySQL-Server automatisch gestartet. Abhilfe schaffen zwei service-Kommandos:

```
root# service apache2 start
Starting Apache httpd web server
[core:warn] [pid 9124] (92)Protocol not available: AH00076:
Failed to enable APR_TCP_DEFER_ACCEPT
root@win10:~# service mysql start
Starting MySQL database server
mysql: No directory, logging in with HOME=/
```

Die angezeigten Warnungen sind zwar unschön, davon abgesehen funktioniert aber alles auf Anhieb. Wenn Sie mit dem folgenden Kommando ein PHP-Script mit der Funktion `phpinfo()` einrichten, können Sie unter `localhost://testphp.php` alle Details der PHP-Konfiguration nachlesen (siehe [Abbildung 42.7](#)).

```
root# echo "<?php phpinfo(); ?>" > /var/www/html/testphp.php
```

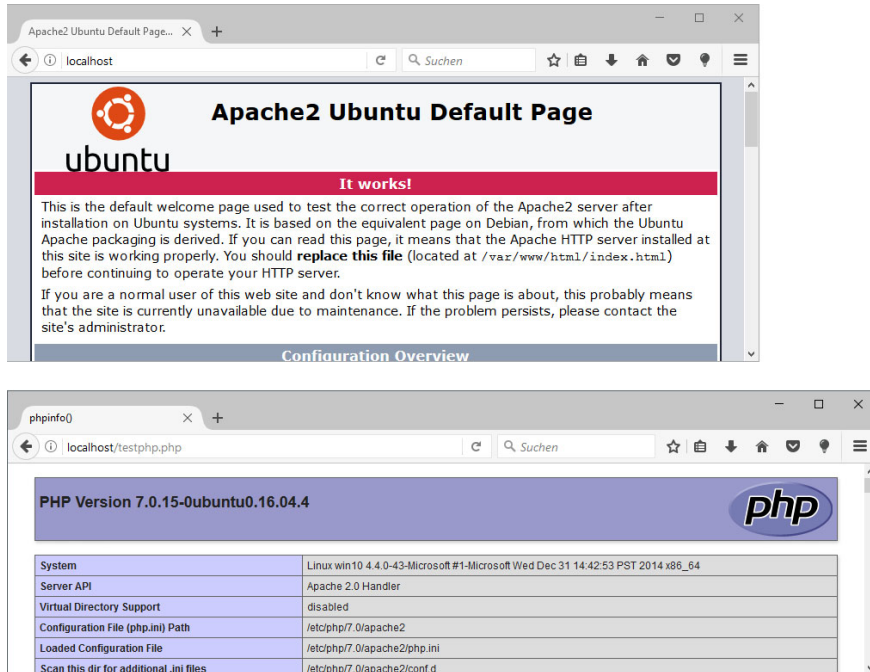


Abbildung 42.7 Die Defaultseite des Apache-Servers und die Info-Seite des PHP-Systems

### Port-Konflikte, Firewall und Auto-Start

Ähnlich wie im vorigen Abschnitt zum SSH-Server setze ich auch hier voraus, dass unter Windows nicht schon ein anderer Web-, MySQL- oder MariaDB-Server läuft. Ist das der Fall, können Sie in `/etc/apache2/ports.conf` den Default-Port von Apache ändern. Die Port-Konfiguration für den MySQL-Server befindet sich in der Datei `/etc/mysql/mysql.conf.d/mysqld.conf`.

Der Webserver ist wegen der Windows-Firewall nur lokal erreichbar. Für Netzwerkzugriffe müssen Sie für Port 80 eine Ausnahmeregel definieren.

Einen automatischen Start können Sie bei Bedarf auf die gleiche Art und Weise wie beim SSH-Server bewerkstelligen.

#### Apache ohne Warnung starten

Die Warnung beim Start von Apache können Sie eliminieren, wenn Sie eine neue Konfigurationsdatei einrichten, die eine Linux-spezifische Optimierung des Sockets deaktiviert:

```
root# echo "AcceptFilter http none" > /etc/apache2/conf-available/myown.conf
root# a2enconf myown
root# service apache2 restart
```

Auch die Warnung beim Start des MySQL-Servers lässt sich abstellen. Sie resultiert daraus, dass in `/etc/passwd` für den Account `mysql` kein richtiges Heimatverzeichnis eingetragen wurde.

MySQL-Server  
ohne Warnung  
starten

```
root# service mysql stop
root# usermod -d /var/lib/mysql/ mysql
root# service mysql start
```

## 42.3 Interna

Dem Windows Subsystem für Linux (WSL) fehlt gewissermaßen der wichtigste Bestandteil – der Linux-Kernel. Dennoch verwendet das WSL die ganz gewöhnlichen, für Linux kompilierten Pakete, keine speziell für Windows hergestellten Kompilate. Wie kann das funktionieren? In diesem Abschnitt versuche ich, etwas Licht in die Mysterien des WSL zu bringen.

Microsoft hat, in den Worten von Dustin Kirkland von Canonical, eine Art inverses Wine geschaffen, in dem Linux-Systemfunktionen durch entsprechende Windows-Funktionen ersetzt bzw. emuliert werden. Dieser Code ist ein Kernstück des Windows Subsystems für Linux. Er ist momentan nicht als Open Source verfügbar, und es gibt auch keine konkreten Informationen dazu, ob dies in Zukunft geplant ist.

Wine inverse

<http://blog.dustinkirkland.com/2016/03/ubuntu-on-windows.html>

Windows enthält schon seit mehr als zwei Jahrzehnten Subsysteme für POSIX und OS/2. Diese Subsysteme halfen, vorhandenen Quellcode für UNIX bzw. OS/2 mit vertretbarem Aufwand auf Windows zu portieren. Jede Portierung setzte allerdings Änderungen im Code und ein neuerliches Kompilieren voraus.

Windows-  
Subsysteme

Das neue Windows Subsystem for Linux ist in dieser Hinsicht radikaler: Es kann für Linux kompilierte 64-Bit-Binaries ohne jede Änderung direkt ausführen. Das ist nicht so einfach, wie es vielleicht klingt: Linux-Programme und -Bibliotheken erwarten, dass alle Grundfunktionen des Kernels zur Verfügung stehen – eine Voraussetzung, die unter Windows natürlich nicht erfüllt ist. Mehrere Bausteine, die teilweise zusammen mit Canonical entwickelt wurden, überwinden diese Hürde.

Im sogenannten User-Space dient `bash.exe` als Startpunkt. Das ist eine speziell für Windows kompilierte Version der `bash`. Rein optisch sieht `bash.exe` wie das Programm `cmd.exe` aus, darüber hinaus gibt es aber wenig Ähnlichkeiten.

bash.exe

`bash.exe` kommuniziert mit den neuen Windows-Kernel-Treibern LXSS (`lxss.sys`) und LXCore (`lxcore.sys`). LXSS fungiert dabei als »Broker« und übersetzt Linux- in Windows-Funktionen. Gleichzeitig kümmert sich LXSS um die Linux-Prozess-

lxss.sys und  
lxcore.sys

verwaltung: Aus `bash.exe` heraus können sowohl Vorder- als auch Hintergrundprozesse gestartet werden.

LXSS greift dabei auf Pico-Prozesse aus dem Sandboxing-Framework *Drawbridge* zurück. Damit sind Linux-Prozesse von Windows-Prozessen getrennt, Systemaufrufe werden direkt vom zugeordneten Treiber LXCore verarbeitet.

LXCore stellt für Windows die wichtigsten Linux-Kernelfunktionen zur Verfügung. In einfachen Fällen kann LXCore eine äquivalente Windows-Kernelfunktion aufrufen. Da es nicht immer gleichwertige Funktionen gibt, enthält LXCore die Nachbildung diverser Linux-Kernelfunktionen. Microsoft hat betont, dass LXCore von Grund auf neu entwickelt wurde, ohne Linux-Quellcode zu verwenden (Cleanroom-Implementierung). Offensichtlich versucht Microsoft hier, Lizenzproblemen (GPL) von vorneherein aus dem Weg zu gehen.

In WSL gestartete Prozesse werden direkt im Task-Manager von Windows angezeigt. Detailliertere Informationen liefert der *Process Explorer*. Dieses kostenlose Programm können Sie hier herunterladen:

<https://docs.microsoft.com/en-us/sysinternals/downloads/process-explorer>

#### VolFS und DriveFS

Die Dateisysteme von Linux und Windows sind grundverschieden und inkompatibel zueinander. Das beginnt bei der Differenzierung zwischen Groß- und Kleinbuchstaben und endet bei den Metadaten zu Dateien (Besitzer, Gruppe, Zugriffsrechte). Zwei neue Dateisystemtreiber schlagen hier die Brücke zwischen Linux und Windows:

- ▶ VolFs stellt den Linux-Prozessen ein Dateisystem zur Verfügung, das wie unter Linux funktioniert. Das gesamte Linux-Systemverzeichnis ist unter Windows auch im Explorer sichtbar, und zwar im verborgenen Verzeichnis `C:\Users\name\lxss\rootfs`.
- ▶ Um umgekehrt Linux-Programmen Zugriff auf das Windows-Dateisystem zu geben, wird dieses unter `/mnt/c` in den Linux-Verzeichnisbaum eingebunden. Verantwortlich dafür ist ein zweiter neuer Dateisystemtreiber, nämlich DriveFS.

Weitere technische Details hat Microsoft in mehreren Blog-Beiträgen erläutert, die Sie alle ausgehend von dieser Startseite finden:

<https://blogs.msdn.microsoft.com/wsl>

#### Nicht Open Source

Die Komponenten von WSL sind aktuell kein Open-Source-Projekt. Ich habe keine Informationen gefunden, ob Microsoft Pläne hat, den Quellcode vielleicht später freizugeben.

# Index

|                                     |          |
|-------------------------------------|----------|
| 1-Wire-Thermometer .....            | 313      |
| 389-Directory-Server .....          | 564      |
| 4-KiB-Sektoren .....                | 55       |
| 4k-Bildschirme .....                | 168      |
| 64-Bit-Bibliotheken .....           | 680      |
| 64-Bit-Distributionen .....         | 40       |
| 7zr .....                           | 1233     |
| 802.11x-Standards .....             | 929      |
| <br>                                |          |
| \$ (Variablen in der bash) .....    | 359      |
| \$( ) (Kommandosubstitution) .....  | 357      |
| & (Hintergrundprozesse) .....       | 434      |
| < (Ausgabeumleitung) .....          | 349      |
| > (Eingabeumleitung) .....          | 350      |
| [ ] (arithmetische Ausdrücke) ..... | 357      |
| * (Jokerzeichen) .....              | 355, 396 |
| ? (Jokerzeichen) .....              | 355, 396 |
| # (Kommandointerpreter) .....       | 364      |
| ~ (Heimatverzeichnis) .....         | 144, 390 |
| "" (Zeichenketten) .....            | 358      |
| ' ' (Zeichenketten) .....           | 358      |
| ` (Kommandosubstitution) .....      | 357      |

## A

|                                   |               |
|-----------------------------------|---------------|
| A-Eintrag (DNS) .....             | 1150          |
| a2disconf .....                   | 1078          |
| a2dismod .....                    | 1078          |
| a2enconf .....                    | 1078          |
| a2enmod .....                     | 1078          |
| a2ensite .....                    | 1078          |
| a2ps .....                        | 470           |
| aa-complain .....                 | 1293          |
| aa-enforce .....                  | 1293          |
| aa-status .....                   | 1292          |
| AAAA-Eintrag (DNS) .....          | 1150          |
| Abkürzungen .....                 | 348           |
| Access Control Lists .....        | 420           |
| Access Point (WLAN) .....         | 915, 929, 966 |
| ACL .....                         | 420           |
| ACPI .....                        | 601           |
| <i>Kernel-Boot-Optionen</i> ..... | 908           |
| Active Directories .....          | 991           |
| Ad-hoc-Modus (WLAN) .....         | 930           |
| addgroup .....                    | 577           |
| adduser .....                     | 577           |

|   |        |
|---|--------|
| Administration .....                          | 563    |
| Administrator-Account .....                   | 72     |
| Adobe Flash .....                             | 218    |
| AFP .....                                     | 1032   |
| afp.conf .....                                | 1033   |
| AirPrint .....                                | 1052   |
| airprint-generate .....                       | 1053   |
| akmods .....                                  | 1301   |
| akms .....                                    | 888    |
| Aktion (Syslog) .....                         | 614    |
| Aktivitäten                                   |        |
| <i>Gnome</i> .....                            | 149    |
| <i>KDE</i> .....                              | 195    |
| Alias (httpd.conf) .....                      | 1083   |
| Alias (E-Mail) .....                          | 1171   |
| alias .....                                   | 348    |
| alias (in modprobe.conf) .....                | 881    |
| alias_database .....                          | 1171   |
| alias_maps .....                              | 1172   |
| alien .....                                   | 656    |
| Allow (Apache) .....                          | 1085   |
| allow-hotplug (/etc/network/interfaces) ..... | 952    |
| Allowed-Origins .....                         | 651    |
| AllowOverride .....                           | 1084   |
| alsactl .....                                 | 610    |
| alsamixer .....                               | 610    |
| alternatives .....                            | 658    |
| Amarok .....                                  | 252    |
| Amazon Web Services (AWS) .....               | 1246   |
| amdgpu .....                                  | 696    |
| anacron .....                                 | 458    |
| Android .....                                 | 26, 29 |
| Antergos .....                                | 29     |
| Apache .....                                  | 1073   |
| <i>Authentifizierung</i> .....                | 1087   |
| <i>HTTPS</i> .....                            | 1095   |
| <i>IPv6</i> .....                             | 1079   |
| <i>Passwort</i> .....                         | 1087   |
| <i>SELinux</i> .....                          | 1075   |
| <i>Sicherheit</i> .....                       | 1087   |
| <i>Unicode</i> .....                          | 1079   |
| <i>Verzeichnis absichern</i> .....            | 1087   |
| <i>virtuelle Hosts</i> .....                  | 1089   |
| <i>WSL</i> .....                              | 1415   |
| <i>Zugriff sperren</i> .....                  | 1087   |
| apfs-Dateisystem (Apple) .....                | 746    |
| APIC .....                                    | 907    |
| aplay .....                                   | 610    |

- apm (Atom-Paketverwaltung) ..... 552  
 AppArmor ..... 1283, 1291  
 apparmor-utils ..... 1293  
 Apple  
     *AirPrint* ..... 1052  
     *Dateisystem* ..... 746  
     *Filing Protocol (AFP)* ..... 1032  
     *Samba* ..... 1024  
 Applets  
     *Gnome* ..... 149  
     *KDE* ..... 192  
 applydeltarpm ..... 628  
 Apps ..... 221  
 APT ..... 643  
     *automatische Updates* ..... 650  
 apt ..... 644  
 apt-cache ..... 649  
 apt-get ..... 644  
 apt-key ..... 645  
 aptitude ..... 644, 647  
 apturl ..... 650  
 arandr ..... 719  
 Arbeitsfläche  
     *Gnome* ..... 152  
     *KDE* ..... 195  
 Arch Linux ..... 29  
 Archivieren von Dateien ..... 1232  
     *Gnome* ..... 161  
 arecord ..... 610  
 Arithmetische Ausdrücke (bash) ..... 357  
 Artifex Ghostscript ..... 473  
 ASCII ..... 593  
 Asymmetrische Verschlüsselung ..... 1095  
 async (NFS) ..... 1027  
 Atlas ..... 1316  
 Atom ..... 548  
 Audacious ..... 253  
 Audio  
     *ALSA* ..... 609  
     *Dateien recodieren* ..... 259  
     *Konverter* ..... 465  
 audiofile ..... 466, 467  
 aufs-Dateisystem ..... 748, 1400  
 Ausgabeumleitung  
     *sudo* ..... 444  
 Auslagerungsdatei ..... 68  
 authconfig (PAM) ..... 590  
 AuthConfig (Apache) ..... 1084  
 Authentifizierung  
     *Apache* ..... 1087  
     *POP/IMAP* ..... 1184  
     *SMTP* ..... 1185  
 AuthName ..... 1088  
 AuthType ..... 1088  
 AuthUserFile ..... 1088  
 auto-Dateisystem ..... 747  
 Auto-Login ..... 708  
     *KDE* ..... 201  
     *SUSE* ..... 709  
 autofs ..... 747  
 autojump ..... 392  
 automatic.conf ..... 637  
 Automatische Ausführung von Jobs 454, 459  
 automount ..... 747  
 Autostart  
     *Gnome* ..... 181  
     *KDE* ..... 201  
     *Unity/Ubuntu* ..... 211  
 Avahi ..... 958  
 avahi-browse ..... 960  
 avahi-daemon ..... 959  
 avahi-discover ..... 960  
 avahi-dnsmconfd ..... 959  
 avconv ..... 248, 467  
 aws ..... 1247  
 awscli ..... 1247  
 AWStats ..... 1112
- ## B
- 
- Babe ..... 253  
 Background-Prozesse ..... 434  
 backintime ..... 1229  
 Backports (Debian) ..... 667  
 Backup Domain Controller ..... 991  
 Backups ..... 1225  
     *Emacs* ..... 521  
     *inkrementelle* ..... 1238  
     *KVM* ..... 1245  
     *LVM-Snapshots* ..... 1244  
     *MySQL* ..... 1138  
     *Scripts* ..... 367  
 bad-interpreter-Fehlermeldung ..... 370  
 baobab ..... 161  
 Base Boxes (Vagrant) ..... 1318  
 Base Images (Docker) ..... 1368  
 bash ..... 341  
     *bash\_history* ..... 344  
     *bashrc* ..... 105  
     *completion* ..... 346  
     *Programmierung* ..... 363  
     *Script-Beispiele* ..... 363, 461  
     *Tastatureinstellung* ..... 343  
     *Tastenkürzel* ..... 346

- unter Windows (WSL)* ..... 1405
  - Variablen* ..... 373
  - Batterie (Notebooks) ..... 601
  - BCM2835/-36/-37 ..... 299
  - BDC ..... 991
  - Bedingungen (bash) ..... 379
  - Benutzer ..... 578
    - einrichten* ..... 576
    - Gruppen* ..... 410
    - verwalten* ..... 576
  - Besitzer
    - neue Dateien* ..... 418
    - von Dateien* ..... 409
  - Betriebssystem ..... 25
  - bg ..... 435
  - Bibliotheken ..... 677
    - 32/64 Bit* ..... 680
    - glibc* ..... 678
    - libc* ..... 678
    - Prelinking* ..... 681
  - Bilder-Verzeichnis ..... 183
  - Bildschirmfreigabe ..... 171
  - Bildschirmschoner (Raspberry Pi) ..... 281
  - bin-Verzeichnis ..... 371, 426
  - Binärpaket ..... 628
  - bind ..... 978
  - bind interfaces only ..... 997
  - bind-address (MySQL) ..... 1126
  - BIOS ..... 41
    - GRUB-Partition* ..... 841
    - GRUB-Reparatur* ..... 841
    - RAID* ..... 57
    - Systemstart* ..... 822
  - BitTorrent ..... 237
  - blacklist (modprobe.conf) ..... 882
  - Blacklist (E-Mail) ..... 1150
  - blkid ..... 752, 762
  - bluedevil ..... 606
  - Bluetooth ..... 606
    - Raspberry Pi* ..... 283
  - bluetoothd ..... 606
  - BMP-PS-Konverter ..... 464
  - bmp2eps ..... 464
  - bmp2tiff ..... 465
  - Bonjour ..... 958
  - Bookmarks (Firefox Sync) ..... 215
  - /boot
    - /efi* ..... 43, 818
    - /grub* ..... 827
    - /initrd* ..... 820
    - /initrd selbst erzeugen* ..... 823
    - /vmlinuz* ..... 819, 898
  - Boot-Optionen ..... 904
  - Boot-Partition ..... 66, 67
    - voll wegen Kernel-Updates* ..... 652
  - Boot-Probleme ..... 76
  - Boot-Prozess ..... 817, 847
    - Bootloader* ..... 817, 823
    - boot.local* ..... 870
    - System-V-Init* ..... 857
  - BOOTREC ..... 82
  - Boxes (Vagrant) ..... 1317
  - Bridge ..... 1354
  - bridge-utils ..... 1354
  - Bridged Networking ..... 1310
  - browseable (Samba) ..... 1007
  - Browsing (Samba) ..... 988
  - BSD-Lizenz ..... 33
  - btrfs-Dateisystem ..... 745, 764
    - RAID* ..... 771
    - Snapper/opensUSE* ..... 775
    - Swap-Dateien* ..... 790
  - Buckets (S3) ..... 1246
  - Budgie ..... 31, 122
  - build-Kommando (Docker) ..... 1393
  - bunzip2 ..... 1232
  - Buster ..... 667
  - bzip2 ..... 1232, 1233
- ## C
- 
- C/C+ (Programmiersprache) ..... 686
  - .cache-Verzeichnis ..... 183
  - CalDAV ..... 1220
  - Canonical (Ubuntu) ..... 120
  - canonical-livepatch ..... 901
  - canonical\_maps ..... 1174
  - Capabilities ..... 424
  - CapsLock-Taste deaktivieren ..... 162
  - CardDAV ..... 1220
  - Carriage Return ..... 468
  - case ..... 380
  - cat ..... 334
  - CD
    - auswerfen* ..... 785
    - Devices* ..... 784
    - ins Dateisystem einbinden* ..... 784
    - Ripper* ..... 261
    - umount-Problem* ..... 785
    - wechseln* ..... 785
  - cd-da2wav ..... 465
  - cdparanoia ..... 465
  - CentOS ..... 30, 84
    - statische Netzwerkkonfiguration* ..... 950
    - sudo* ..... 446

- Systemstart* ..... 865
  - Tastatur* ..... 568
  - certbot (Let's Encrypt) ..... 1106
  - Certification Authority (CA) ..... 1103
  - cgroups ..... 747
    - Docker* ..... 1402
  - chage ..... 583
  - chainloader ..... 837
  - Channel (WLAN) ..... 931
  - character set ..... 593
  - character-set-server (MySQL/MariaDB) . 1126
  - chattr ..... 767
  - chcon ..... 1286
  - checkarray ..... 792, 798
  - checkrestart ..... 652
  - chkconfig ..... 860, 867, 873
  - Chrome ..... 220
  - Chrome OS ..... 30
  - chrome-gnome-shell ..... 176
  - Chromium ..... 220
  - Chrony ..... 573
  - chroot ..... 842, 843, 1260
  - chsh ..... 342
  - cifs-Dateisystem ..... 1020, 1022
  - cifs-utils ..... 1020
  - Cinnamon Desktop ..... 186
  - ClamAV ..... 1193
  - classes.conf ..... 1040
  - Client-Konfiguration ..... 911
  - cloneconfig ..... 895
  - Cluster SSH ..... 1067
  - Cluster-Dateisystem ..... 723, 748
  - CMD (Dockerfile) ..... 1391
  - cmus ..... 254
  - Codec ..... 248
  - collation-server (MySQL/MariaDB) ..... 1126
  - Compiler ..... 682
  - Compiz (Unity) ..... 205
  - complete ..... 346
  - .config-Verzeichnis ..... 183
  - config.txt (Device Trees) ..... 883
  - configure ..... 684
  - conky ..... 437
  - console-setup ..... 567, 569
  - Container (Docker) ..... 1363
  - Container Layer (Docker) ..... 1399
  - Contrib-Pakete ..... 666
  - Control Groups ..... 747
    - Docker* ..... 1402
  - Converseen ..... 464
  - convert ..... 463
  - convmv ..... 469
  - Copr (DNF) ..... 637
  - Copy on Write (COW) ..... 764
    - deaktivieren* ..... 766
  - cp ..... 391, 392
    - Namen beim Kopieren ändern* ..... 398
  - cPanel ..... 564
  - cpio ..... 825
  - CPU-Frequenz limitieren ..... 600
  - CPU-Temperatur ..... 600
  - cpu-checker ..... 1330
  - cpufreq ..... 600
  - cpufreq-set ..... 600
  - cpufrequtils ..... 600
  - cracklib ..... 584
  - cramfs-Dateisystem ..... 748
  - create mask ..... 1008
  - cron ..... 454
    - durch systemd ersetzen* ..... 459
  - crontab ..... 454
  - crossmnt (NFS) ..... 1028
  - cryptsetup ..... 809
  - Crypto-Dateisystem ..... 722
  - csh ..... 342
  - CSS ..... 249
  - CSSH ..... 1067
  - ctrl-alt-del.target ..... 852
  - CUPS ..... 1037
    - Browsing-Funktion* ..... 1051
    - cupsd* ..... 1040
    - cupsd.conf* ..... 1040
    - cupsenable* ..... 1041
    - Firewall* ..... 1049
    - Interna* ..... 1040
    - Netzwerkdrucker nutzen* ..... 1049
    - SUSE-Besonderheiten* ..... 1043
  - curl ..... 495, 1245
  - CustomLog ..... 1083
  - Cut&Paste ..... 143
- ## D
- 
- Dämonen ..... 450
  - dash (Shell) ..... 364
  - Dash (Gnome) ..... 150
    - Dash to Dock* ..... 178
  - Dash (Unity) ..... 205
  - data-Option (Journaling-Modus) ..... 758
  - Dateien
    - Dateinamen* ..... 389
    - drucken* ..... 1045
    - Grundlagen* ..... 389
    - Jokerzeichen* ..... 355
    - komprimieren* ..... 1232



- kopieren mit sed* ..... 398
- Nautilus* ..... 153
- suchen* ..... 404
- Dateisystem
  - ext-Dateisystem* ..... 758
  - Fragmentierung* ..... 763
  - Konfiguration* ..... 752
  - Loopback-Device* ..... 748
  - maximale Dateigröße* ..... 757
  - reparieren* ..... 756
  - Schnelleinstieg* ..... 144
  - Typen* ..... 744, 754
  - überprüfen* ..... 756
  - vergrößern (ext)* ..... 762
  - vergrößern (xfs)* ..... 780
  - verschlüsseln* ..... 722
  - verwalten* ..... 721
  - virtuelles* ..... 747
  - WSL* ..... 1409
- Dateityp
  - im ls-Kommando* ..... 393
  - Magic-Datei* ..... 404
  - MIME* ..... 402
- Datenbank-Server ..... 1123
- Datenpartition ..... 66
- Datenträger formatieren ..... 725
- dbus-daemon ..... 609
- dcfldd ..... 277, 323
- dconf ..... 181
- dconf-editor ..... 181
- dcraw ..... 247, 465
- dctrl-Format ..... 641
- dd ..... 44
- Dead Keys ..... 49, 567
- Debian ..... 30
  - debian-goodies* ..... 652
  - DKMS* ..... 887
  - Firmware* ..... 94
  - initrd-Datei* ..... 824
  - Paketverwaltung* ..... 640
  - Systemstart* ..... 867
  - Tastatur* ..... 567
  - VirtualBox* ..... 1306
- declare ..... 361
- decode\_MPG2 und \_WVC1 ..... 289
- Decoder ..... 248
- Decodierer (MPEG-2, VC-1) ..... 288
- Defragmentierung ..... 763
- Deja Dup ..... 1225
- Delayed Allocation (ext4-Dateisystem) .... 759
- delgroup ..... 577
- Delta-Updates ..... 628
- deltarpm ..... 628
- deluser ..... 577
- Deny (Apache) ..... 1085
- DenyHosts ..... 1062
- Desklets ..... 187
- .desktop-Dateien ..... 708
- Desktop
  - Gnome* ..... 147
  - KDE* ..... 190
- deutsche Sonderzeichen
  - bash* ..... 343
  - Emacs, US-Tastatur* ..... 546
- /dev ..... 426
  - /disk* ..... 732
  - Interna* ..... 429
  - Liste* ..... 431
  - /mapper* ..... 800
  - /md* ..... 791
  - /pts* ..... 747
  - /sd* ..... 729
  - /vd* ..... 730
- Device-Abschnitt (X) ..... 713
- device is busy (Fehlermeldung) ..... 785
- Device Trees ..... 314, 882
- device-tree-Parameter ..... 883
- DeviceKit ..... 608
- Devices ..... 412, 429, 728
  - CD/DVD-Laufwerke* ..... 784
  - Interna* ..... 429
  - Kernelmodule* ..... 881
  - udev-Dateisystem* ..... 430
- Devil Linux ..... 32
- devtmpfs-Dateisystem ..... 747
- df ..... 750
- dhclient ..... 937, 956
- dhclient.conf ..... 956
- DHCP ..... 923, 976
  - Client-Konfiguration* ..... 923, 985
  - Hostname* ..... 985
  - Server* ..... 976
  - Server-Konfiguration (Dnsmasq)* ..... 978
- dhcpcd ..... 937, 949
- dhcpcd ..... 978
- digikam ..... 240
- Dillo ..... 222
- directory mask ..... 1008
- Directory Server ..... 564
- DirectoryIndex ..... 1084
- disable\_vrfy\_command (Postfix) ..... 1179
- discard ..... 754, 807
- Discover ..... 655
- Disk-Images (libvirt/KVM) ..... 1336
- Disk-Quotas ..... 723
- Display-Abschnitt (X) ..... 715

- Display Manager ..... 707
- Distributionen
  - Überblick* ..... 29
  - Updates* ..... 78
- DKIM (Postfix) ..... 1197
- DKMS (Kernel) ..... 887
  - VirtualBox* ..... 1300
- DLLs ..... 678
- DLNA ..... 296
- DMARC ..... 1203
- dm\_crypt ..... 809
- dmesg ..... 615
- DNF (Paketmanager) ..... 636
  - dnf.conf* ..... 636
  - dnf-plugin-system-upgrade* ..... 668
  - downgrade* ..... 1307
  - systemd-Timer-Beispiel* ..... 460
- DNS
  - Client-Konfiguration* ..... 923, 944
  - Mail-Server* ..... 1150
  - Proxy* ..... 994
  - Reverse DNS* ..... 1153
  - Server-Konfiguration (Dnsmasq)* ..... 978
  - Ubuntu* ..... 917
- dns-nameservers ..... 953
- Dnsmasq ..... 978
  - NetworkManager* ..... 917
- do-release-update ..... 676
- Dock (Unity) ..... 205
- Docker ..... 1363
- docker-compose ..... 1388
- Dockerfile-Syntax ..... 1390
- DocumentRoot (Apache) ..... 1075, 1082
- Dokumente-Verzeichnis ..... 183
- Dokumentkonverter ..... 469
- Dolphin ..... 196
  - Verzeichnis freigeben* ..... 1012
- Domain-Level-Sicherheit ..... 991
- Domain-Nameserver siehe DNS ..... 977
- DomainKeys Identified Mail ..... 1197
- Domainname ..... 921
- Doppellizenzen ..... 33
- DOS-Dateien konvertieren ..... 468
- DOS-Dateisystem ..... 746
- dos2unix ..... 468
- dotglob ..... 356
- Dovecot
  - dovecot.conf* ..... 1180
  - IPv6* ..... 1182
  - POP/IMAP-Authentifizierung* ..... 1184
  - SMTP-Authentifizierung* ..... 1185
- Downloads-Verzeichnis ..... 183
- dpkg ..... 641
  - dpkg-reconfigure* ..... 572, 642
  - Beispiele* ..... 641
  - Multiarch* ..... 680
  - Statuscode* ..... 642
- DPMS (Raspberry Pi) ..... 281
- dracut ..... 825
- Dragon ..... 256
- Dreisritt (Kompilieren) ..... 684
- DRI ..... 693
- DriveFS-Dateisystem ..... 1418
- DRM ..... 249, 691
- Dropbox ..... 235
- Drucken ..... 1037
  - automatische Datenkonversion* ..... 1038
  - Dämon (CUPS)* ..... 1040
  - Druckjobs verwalten* ..... 1045
  - Filter* ..... 1038
  - GDI-Drucker (Windows)* ..... 1047
  - Gnome* ..... 164
  - KDE* ..... 203
  - Konfiguration* ..... 1037, 1046
  - MIME (CUPS)* ..... 1041
  - per Kommando* ..... 1045
  - PostScript* ..... 1037
  - Server-Konfiguration* ..... 1037, 1050
  - Spooling-System* ..... 1038
  - Warteschlange* ..... 1038
- DS1820 ..... 313
- DSC (PostScript) ..... 475
- DTB-Dateien ..... 883
- dtoverlay-Schlüsselwort ..... 883
- dtparam-Schlüsselwort ..... 883
- Dual-Stack (IPv6) ..... 929
- Duplicity ..... 1240
- DVD
  - auswerfen* ..... 785
  - brennen in Gnome* ..... 157
  - brennen in KDE* ..... 199
  - Dateisystem* ..... 746
  - Devices* ..... 784
  - ins Dateisystem einbinden* ..... 784
  - Ripper* ..... 262
  - umount-Problem* ..... 785
  - Videos abspielen* ..... 786
  - wechseln* ..... 785
- Dynamic Host Configuration Protocol ..... 976
- Dynamisch gelinkte Programme ..... 678

## E

- E-Mail
  - Alias ..... 1171
  - Blacklist ..... 1150
  - DNS ..... 1150
  - Evolution ..... 229
  - Grundlagen ..... 1144
  - Kontakt ..... 232
  - mutt ..... 497
  - Relaying ..... 1150
  - Server ..... 1143
  - Thunderbird ..... 222
  - Viren ..... 1193
- e2label ..... 762
- e4defrag ..... 763
- EasyTAG ..... 260
- eBox ..... 564
- EDITOR ..... 337
- Editoren ..... 335
  - Atom ..... 547
  - Emacs ..... 519
  - Joe ..... 337
  - Nano ..... 337
  - Vim ..... 501
  - VSCode ..... 555
- EFI ..... 41
  - Bootloader per GRUB starten ..... 839
  - efibootmgr ..... 844
  - GPT ..... 818
  - GRUB 2 ..... 818
  - GRUB-Reparatur ..... 843
  - Partition ..... 43, 818
  - Secure Boot ..... 43, 77, 820
  - Systemstart ..... 818
- EGL ..... 691, 693
- Eingabefokus (X) ..... 143
- Eingabeumleitung ..... 349
  - sudo ..... 444
- eject ..... 785
- Electronic Frontier Foundation ..... 1106
- Elementary OS ..... 31, 122
- ELinks ..... 471, 496
- Elvis ..... 335
- Emacs ..... 519
  - automatische Sicherheitskopie ..... 521
  - Bearbeitungsmodi ..... 523, 539
  - Cursorbewegung ..... 525
  - dynamische Abkürzungen ..... 534
  - Editierkommandos ..... 528
  - Ein- und Ausrückungen ..... 530
  - Einrückungen im Fließtext ..... 532
  - .emacs-Datei ..... 541, 542
  - Erweiterungen ..... 544
  - farbiger Text ..... 539
  - Fenster ..... 538
  - Fließtext ..... 531
  - font-lock-mode ..... 539
  - fremdsprachige Zeichen ..... 545
  - Hintergrundfarbe einstellen ..... 542
  - Konfiguration ..... 541
  - MELPA ..... 544
  - Online-Hilfe ..... 522
  - Puffer ..... 537
  - reguläre Ausdrücke ..... 535
  - Schnelleinstieg ..... 335
  - Schriftart einstellen ..... 542
  - suchen ..... 534
  - suchen und ersetzen ..... 536
  - Syntaxhervorhebung ..... 539
  - Tabulatoren ..... 529
  - Textmodus ..... 533
  - Unicode ..... 545
- emergency (Kerneloption) ..... 906
- Emergency-Target ..... 852
- Encoder ..... 248
- Encryption (Dateisystem) ..... 722
- Energiesparfunktionen ..... 601
- Enigmail ..... 228
- encrypt ..... 470
- ENTRYPOINT (Dockerfile) ..... 1391
- env ..... 597
- Environment-Variablen ..... 360
- EnvironmentFile (systemd) ..... 863
- EPEL-Paketquelle ..... 90, 665
- Epiphany ..... 222
- EPS-Konverter ..... 472
- epsffit ..... 474
- epstopdf ..... 472
- ErrorDocument ..... 1083
- ErrorLog ..... 1082
- erweiterte Partition ..... 53
- ESP Ghostscript ..... 473
- ESR-Version
  - Firefox ..... 214
  - Thunderbird ..... 223
- ESSID (WLAN) ..... 930
- /etc ..... 426, 566
  - /adduser.conf ..... 577
  - /adjtime ..... 570
  - /aliases ..... 1147, 1157, 1171
  - /alternatives ..... 657
  - /apparmor.d ..... 1292
  - /apt/apt.conf ..... 644
  - /apt/sources.list ..... 644
  - /auto.master ..... 747

|   |          |                                    |          |
|---|----------|------------------------------------|----------|
| <i>/boot/grub.cfg</i> .....               | 827      | <i>/mdadm/mdadm.conf</i> .....     | 791      |
| <i>/chrony.conf</i> .....                 | 573      | <i>/mime.types</i> .....           | 404      |
| <i>/cron.daily</i> .....                  | 457      | <i>/modprobe.conf</i> .....        | 880, 935 |
| <i>/cron.hourly</i> .....                 | 457      | <i>/modprobe.d</i> .....           | 880      |
| <i>/cron.monthly</i> .....                | 457      | <i>/modules</i> .....              | 881      |
| <i>/cron.weekly</i> .....                 | 457      | <i>/modules-load.d</i> .....       | 880      |
| <i>/crontab</i> .....                     | 454      | <i>/mtab</i> .....                 | 750      |
| <i>/crypttab</i> .....                    | 811      | <i>/my.cnf</i> .....               | 1126     |
| <i>/cups</i> .....                        | 1040     | <i>/netatalk/afp.conf</i> .....    | 1033     |
| <i>/default/language</i> .....            | 596      | <i>/network/interfaces</i> .....   | 952      |
| <i>/default/grub</i> .....                | 830      | <i>/nscd.conf</i> .....            | 592      |
| <i>/default/prelink</i> .....             | 681      | <i>/nsswitch.conf</i> .....        | 591      |
| <i>/deluser.conf</i> .....                | 577      | <i>/PackageKit/*</i> .....         | 655      |
| <i>/denyhosts.conf</i> .....              | 1062     | <i>/pam.conf</i> .....             | 588      |
| <i>/dhcp3/dhclient.conf</i> .....         | 956      | <i>/pam.d/*</i> .....              | 588      |
| <i>/dnf/dnf.conf</i> .....                | 636      | <i>/passwd</i> .....               | 579      |
| <i>/dnsmasq.conf</i> .....                | 979      | <i>/php.ini</i> .....              | 1117     |
| <i>/dovecot</i> .....                     | 1180     | <i>/polkit-1</i> .....             | 448      |
| <i>/dracut.conf</i> .....                 | 825      | <i>/postfix</i> .....              | 1157     |
| <i>/file</i> .....                        | 404      | <i>/prelink.conf</i> .....         | 681      |
| <i>/firewall.d</i> .....                  | 1268     | <i>/printcap (CUPS)</i> .....      | 1040     |
| <i>/fstab</i> .....                       | 752      | <i>/profile</i> .....              | 360, 362 |
| <i>/fstab (CIFS)</i> .....                | 1022     | <i>/rc.d/*</i> .....               | 859      |
| <i>/fstab (LABEL)</i> .....               | 752      | <i>/rc.d/rc.local</i> .....        | 867      |
| <i>/fstab (NFS)</i> .....                 | 1030     | <i>/rc.local</i> .....             | 868      |
| <i>/fstab (SSD-TRIM)</i> .....            | 807      | <i>/resolv.conf</i> .....          | 944      |
| <i>/ftputers</i> .....                    | 1121     | <i>/resolv.conf (Ubuntu)</i> ..... | 945      |
| <i>/group</i> .....                       | 580      | <i>/rsyslog.conf</i> .....         | 612      |
| <i>/gshadow</i> .....                     | 586      | <i>/samba/smb.conf</i> .....       | 993      |
| <i>/host.conf</i> .....                   | 943      | <i>/selinux</i> .....              | 1289     |
| <i>/hostapd.conf</i> .....                | 974      | <i>/services</i> .....             | 871      |
| <i>/hostname</i> .....                    | 946      | <i>/shadow</i> .....               | 582      |
| <i>/hosts</i> .....                       | 943, 979 | <i>/shells</i> .....               | 342      |
| <i>/hosts (neuen Server testen)</i> ..... | 1094     | <i>/skel</i> .....                 | 580      |
| <i>/hosts.allow</i> .....                 | 1258     | <i>/smartd.conf</i> .....          | 806      |
| <i>/hosts.deny</i> .....                  | 1258     | <i>/ssh</i> .....                  | 1058     |
| <i>/idmapd.conf</i> .....                 | 1029     | <i>/ssl/certs</i> .....            | 914      |
| <i>/inetd.conf</i> .....                  | 872      | <i>/sudoers</i> .....              | 443      |
| <i>/init.d</i> .....                      | 869      | <i>/sysconfig</i> .....            | 869      |
| <i>/init.d/boot.local</i> .....           | 870      | <i>Dokumentation</i> .....         | 952      |
| <i>/inittab</i> .....                     | 858      | <i>network-scripts</i> .....       | 950      |
| <i>/inputrc</i> .....                     | 343      | <i>/sysconfig/authconfig</i> ..... | 590      |
| <i>/ld.so.cache</i> .....                 | 679      | <i>/sysconfig/i18n</i> .....       | 596      |
| <i>/ld.so.conf</i> .....                  | 679      | <i>/sysconfig/language</i> .....   | 596      |
| <i>/letsencrypt</i> .....                 | 1107     | <i>/sysconfig/console</i> .....    | 569      |
| <i>/libvirt</i> .....                     | 1334     | <i>/sysconfig/locate</i> .....     | 406      |
| <i>/lightdm</i> .....                     | 708      | <i>/sysconfig/network</i> .....    | 946      |
| <i>/locale.conf</i> .....                 | 596      | <i>/sysconfig/prelink</i> .....    | 681      |
| <i>/localtime</i> .....                   | 571      | <i>/sysctl.conf</i> .....          | 908, 972 |
| <i>/login.defs</i> .....                  | 420, 583 | <i>/systemd/network</i> .....      | 956      |
| <i>/logrotate.conf</i> .....              | 616      | <i>/systemd/journal.conf</i> ..... | 621      |
| <i>/mailcap</i> .....                     | 404      | <i>/timezone</i> .....             | 571      |
| <i>/manpath.conf</i> .....                | 339      | <i>/udev</i> .....                 | 430      |

- /updatedb.conf* ..... 406
  - /vconsole.conf* ..... 568
  - /vsftpd.conf* ..... 1120
  - /vsftpd/ftpusers* ..... 1121
  - /vsftpd/user\_list* ..... 1121
  - /wpa\_supplicant.conf* ..... 282
  - /wpa\_supplicant* ..... 941
  - /X11/xorg.conf* ..... 710
  - /xdg/user-dirs.conf* ..... 183
  - /xinetd.d/\** ..... 873
  - /yum.conf* ..... 632
  - Ethernet-Controller
    - IP-Adresse ..... 924
    - konfigurieren ..... 934
    - MAC-Adresse ..... 922
  - ethtool ..... 947
  - evim ..... 517
  - Evolution ..... 229
  - except-interfaces ..... 980
  - Exchange-Server ..... 1145
  - \*.exe-Datei ..... 434
  - Exec Shield ..... 1284
  - ExecCGI ..... 1084
  - ExecStart-Schlüsselwort (systemd) ..... 862
  - exFAT-Dateisystem ..... 780
  - exfat-utils ..... 781
  - EXIF-Informationen ..... 465
  - exit (bash) ..... 385
  - Expansion von Dateinamen ..... 344
  - expect ..... 584, 1059
  - Experimental-Pakete ..... 667
  - export ..... 361
  - ext-Dateisystem ..... 745, 758
    - Verschlüsselung ..... 812
    - Windows-Zugriff ..... 763
  - Extended Attributes ..... 421
  - Extension Pack (VirtualBox) ..... 1303
  - externe Laufwerke ..... 786
  - extractres ..... 474
- F**
- f.lux-Programm ..... 168
  - faac ..... 466
  - faad ..... 466
  - FAI (Fully Automatic Installation) ..... 564
  - Fail2Ban ..... 1063
  - faillock ..... 585
  - faillog ..... 585
  - Fake-RAID ..... 57
  - Fan Control ..... 604
  - Farbprofile ..... 168
  - Fastest Mirror (Yum) ..... 634
  - FAT-Dateisystem ..... 780
  - fbdev-Treiber (X) ..... 714
  - fc-list ..... 174
  - Fedora ..... 30, 99
    - automount ..... 747
    - Distributions-Update ..... 668
    - DKMS ..... 887
    - dracut ..... 825
    - Firewall ..... 1267
    - Gateway-Konfigurationsdatei ..... 944
    - initrd-Datei ..... 825
    - LABEL in /etc/fstab ..... 752
    - Masquerading ..... 972
    - statische Netzwerkkonfiguration ..... 950
    - sudo ..... 446
    - Systemstart ..... 865
    - Tastatur ..... 568
    - VirtualBox ..... 1307
  - Fensterbuttons (Gnome) ..... 174
  - Fernwartung ..... 171
  - feste Links ..... 400
  - Festplatte
    - 4-KiB-Sektoren ..... 55
    - formatieren ..... 52
    - partitionieren, Linux ..... 64
    - überwachen (SMART) ..... 803
  - fetchmsttfonts ..... 173
  - FFmpeg ..... 467, 248
  - fg ..... 435
  - fglrx ..... 696
  - FHS ..... 425
  - FIFO ..... 350
  - file ..... 404
  - FileInfo ..... 1084
  - Filesystem Hierarchy Standard ..... 425
  - FileZilla ..... 237
  - Filter
    - CUPS ..... 1038
    - IP-Paketfilter ..... 1262
  - find ..... 407
  - findmnt ..... 750
  - Firefox ..... 214
    - MIME ..... 216
    - Plugins ..... 217
    - Sync ..... 215
  - Firewall ..... 1251
    - AFP-Server (Netatalk) ..... 1034
    - Beispiel ..... 1274
    - CUPS ..... 1049
    - FTP ..... 493
    - Grundzustand herstellen ..... 1277
    - IPv6 ..... 1265, 1274

- Mail-Server* ..... 1156
  - NFS 4* ..... 1029
  - openSUSE* ..... 119
  - Paketfilter* ..... 1262
  - Samba* ..... 996
  - Web-Server* ..... 1074
  - firewall-cmd ..... 1156, 1270
    - NFS* ..... 1029
  - FirewallD ..... 1267
  - Firewire-Laufwerke ..... 786
  - Firmware ..... 933
    - Debian* ..... 94, 99
  - fish (Konqueror) ..... 199
  - fixfmps ..... 474
  - fixmacps ..... 474
  - fixscribeps ..... 474
  - fixtpps ..... 474
  - fixwfwps ..... 474
  - fixwpps ..... 474
  - fixwwps ..... 474
  - flac ..... 467
  - Flash ..... 218
  - flashplugin-installer ..... 219
  - Flatpak ..... 661
  - Fokus (X) ..... 143
  - FollowSymLinks ..... 1084
  - font-lock-mode ..... 539
  - Fonts ..... 173, 595
  - for (bash) ..... 381
  - force group ..... 1008
  - forcefsck ..... 757
  - Fork-Typ (systemd) ..... 862
  - FORMAT (Windows) ..... 52
  - Formatieren
    - btfs-Dateisystem* ..... 765
    - exfat-Dateisystem* ..... 725, 781
    - ext3/ext4-Dateisystem* ..... 760
    - ntfs-Dateisystem* ..... 725, 781
    - vfat-Dateisystem* ..... 725, 781
    - xfs-Dateisystem* ..... 779
  - Forward Secrecy ..... 1167
  - Fotodrucker ..... 1047
  - Fragmentierung ..... 763
  - Framebuffer (X) ..... 714
  - free ..... 601
  - Free Software Foundation ..... 33
  - Freigaben
    - Gnome-Einstellungen* ..... 1012
    - Medien (Gnome)* ..... 159
    - Nautilus* ..... 158, 1011
    - WebDAV (Gnome)* ..... 159
  - fremdsprachige Zeichen (Emacs) ..... 545
  - freshclam ..... 1193
  - fsck ..... 757
  - fsck.xfs ..... 780
  - FSF ..... 33
  - fsid (NFS) ..... 1028
  - FSSTND ..... 425
  - fstab ..... 752
    - CIFS* ..... 1022
    - NFS* ..... 1030
  - fstrim ..... 808
  - ftp-Kommando ..... 491
  - FTP ..... 491
    - Client* ..... 237, 491
    - Masquerading* ..... 973
    - passiver Modus* ..... 493
    - Secure FTP Server* ..... 1059
    - Server* ..... 1118
    - Server (sftp)* ..... 1059
  - ftpusers (vsftpd) ..... 1121
  - function (bash) ..... 384
  - FUSE ..... 748, 783
  - fuser ..... 437
- ## G
- 
- Gateway ..... 923
    - Client-Konfiguration* ..... 936, 944
    - Server-Konfiguration* ..... 970
  - gconf ..... 181
  - gconf-editor ..... 181
  - gconftool-2 ..... 181
  - GDI-Drucker ..... 1047
  - gdm ..... 707
  - Geary ..... 234
  - getafm ..... 474
  - getcap ..... 425
  - getfacl ..... 422
  - getfattr ..... 424
  - getsebool ..... 1288
  - gfs-Dateisystem ..... 748
  - gftp ..... 492
  - Ghostscript ..... 472
  - GID ..... 580
  - gif2tiff ..... 465
  - GIMP ..... 242
    - Screenshots* ..... 264
  - gimp-dcraw ..... 247
  - GIMP-Print ..... 473
  - GL (Open GL) ..... 694
  - glibc ..... 678
    - Zeitzone* ..... 571
  - Global Filesystem ..... 723
  - Global Unicast (IPv6) ..... 926

- globstar-Option ..... 356
  - GLX ..... 693
  - glx-utils ..... 704, 1307
  - glxinfo ..... 704
  - GMT (Greenwich Mean Time) ..... 570
  - Gnome ..... 147
    - Bildschirmeinstellungen* ..... 718
    - Extensions* ..... 176
    - gdm* ..... 708
    - Proxy-Einstellungen* ..... 919
    - Screenshots* ..... 264
    - Shell Extensions* ..... 176
    - Shell Themes* ..... 179
    - Tweak Tool* ..... 174
    - Verzeichnis freigeben* ..... 1011
  - gnome-bluetooth ..... 606
  - gnome-disk-utils ..... 743
  - gnome-disks ..... 743
  - gnome-keyring-daemon ..... 1065
  - gnome-language-selector ..... 595, 676
  - gnome-nettool ..... 484
  - gnome-screenshot ..... 264
  - gnome-software ..... 655
  - gnome-software-Programm ..... 665
  - gnome-sound-recorder ..... 257
  - gnome-system-monitor ..... 437
  - gnome-terminal ..... 331
  - gnome-tweak-tool ..... 162, 175
  - gnome-vfs.keys ..... 182
  - gnome-vfs.mime ..... 182
  - GNU ..... 35
    - Emacs* ..... 519
    - General Public License* ..... 33
    - Ghostsript* ..... 473
    - GRUB* ..... 817
  - GoAccess ..... 1113
  - Google Analytics ..... 1112
  - Google Authenticator ..... 1069
  - Google Chrome ..... 220
  - GOsa ..... 564
  - GParted ..... 742
  - gpasswd ..... 586
  - gpg ..... 809, 1249
  - gpio ..... 305
  - GPL ..... 33
  - gpm ..... 569
  - GPT ..... 52, 734
    - EFI* ..... 818
    - GRUB 2* ..... 841
    - Partitionsnummern* ..... 731
  - Grafik-Konverter ..... 463
  - graphical-Target ..... 705
  - grep ..... 408
    - Beispiele* ..... 364
  - grep-dctrl ..... 641
  - Greylisting (Postfix) ..... 1191
  - groupadd ..... 577
  - Grsync ..... 1228
  - GRUB ..... 817
    - Bedienung* ..... 826
    - GPT* ..... 841
    - Festplattennamen* ..... 834
    - Kernel-Updates* ..... 825
    - Konfiguration* ..... 827
    - Partitionsnamen* ..... 834
    - Secure Boot* ..... 820
    - Version 2* ..... 827
    - Reparatur (BIOS)* ..... 841
    - Reparatur (EFI)* ..... 843
  - grub-editenv ..... 834
  - grub-install ..... 841
  - grub.cfg ..... 827
  - grub2-mkconfig ..... 828, 829
  - grubby ..... 817
  - Gruppen ..... 580
    - neue Dateien* ..... 418
    - von Dateien* ..... 409
  - gs ..... 472
  - gsettings ..... 181
  - gsox ..... 467
  - GStreamer ..... 250, 610
  - gtf ..... 712, 718
  - GTUBE-Testnachricht ..... 1190
  - gucharmap ..... 174
  - guest account ..... 1009
  - guest ok ..... 1009
  - guest only ..... 1009
  - Gufw ..... 1274
  - gunzip ..... 1232, 1233
  - Gutenprint ..... 473, 1039
  - GVFS ..... 158, 1232
  - gvim ..... 502
  - gzip ..... 1232, 1233
- ## H
- 
- Hacker-Kernel ..... 890
  - HAL ..... 608
  - HandBrake ..... 263
  - Hardware
    - Devices* ..... 429
    - Devices (udev-Dateisystem)* ..... 430
    - RAID* ..... 57
    - Referenz* ..... 599

- Hardware Enablement Stack ..... 130
  - Hashicorp ..... 1316
  - hawkey ..... 636
  - hdO,O (GRUB) ..... 834
  - HDR-Bilder ..... 247
  - Heimatverzeichnis ..... 144, 390, 579, 580
  - Hello World ..... 686
  - help ..... 340
  - Heredoc-Syntax ..... 385
    - SSH ..... 487
  - HFS-Dateisystem (Apple) ..... 746
  - Hibernate-Kerneloptionen ..... 908
  - HiDPI-Bildschirme ..... 168
  - Hintergrundprozesse ..... 434
  - History
    - APT ..... 650
    - bash ..... 344
    - Yum/DNF ..... 634
    - ZYpp ..... 640
  - hold-Status (dpkg-Kommando) ..... 643
  - Home-Partition ..... 67
  - Home-Server ..... 1013
  - Home-Verzeichnis ..... 144, 390, 426
  - host ..... 1152, 1153
  - host.conf ..... 943
  - hostapd ..... 973
  - Hostname ..... 921
    - DHCP-Client-Konfiguration ..... 985
    - DHCP-Server (Dnsmasq) ..... 981
    - einstellen ..... 946
    - HOSTNAME-Variable ..... 362
    - SUSE ..... 118
  - hostnamectl ..... 856
  - hosts ..... 943
  - hosts allow (Samba) ..... 997
  - hosts deny (Samba) ..... 997
  - hosts.allow (TCP-Wrapper) ..... 1258
  - hosts.deny (TCP-Wrapper) ..... 1258
  - Hotplug-System ..... 608
  - Hotspot einrichten (WLAN) ..... 915
  - HP-Druckertreiber ..... 1042
  - HPLIP ..... 1042
  - hplip-toolbox ..... 1042
  - .htaccess-Datei (Apache) ..... 1089
  - html2ps ..... 471
  - html2text ..... 471
  - htop ..... 436
  - htpasswd ..... 1087
  - HTST (HTTP Strict Transport Security) ... 1109
  - HTTP
    - Apache ..... 1073
    - httpd.conf ..... 1076
    - HTTPS ..... 1095
    - Proxy ..... 919
    - Webserver (Apache) ..... 1073
  - hwclock ..... 570
  - HWE-Pakete ..... 130
  - hydra ..... 1061
  - Hyper-Threading ..... 907
- 
- I
  - il8n ..... 593
  - i915 ..... 696
  - icedax ..... 465
  - IcedTea ..... 687
  - ICMP ..... 920, 1253
  - Icon Themes ..... 180
  - iconv ..... 468
  - IdentityFile ..... 1066
  - idn ..... 946
  - if (bash) ..... 378
  - ifcfg-enp4s0 ..... 950
  - ifconfig ..... 936
  - ifdown ..... 956
  - ifupdown ..... 957
  - Image Magick ..... 463
  - Images
    - Docker ..... 1364
    - Docker, Interna ..... 1399
    - erzeugen (qemu-img) ..... 1337
    - Formate ..... 1336
    - lesen/manipulieren ..... 1357
    - libvirt/KVM ..... 1334
    - VirtualBox ..... 1304
  - IMAP ..... 1144
    - Authentifizierung ..... 1184
  - Immunix ..... 1291
  - includeres (psutils-Kommando) ..... 474
  - Includes (Apache) ..... 1084
  - Indexes (Apache) ..... 1084
  - indicator-multiloader ..... 208
  - Indikatorprogramme (Ubuntu) ..... 208
  - inet6 ..... 953
  - inet\_interfaces (Postfix) ..... 1159
  - inetd.conf ..... 872
  - info ..... 340
  - Infrastructure-Modus (WLAN) ..... 930
  - init ..... 857
  - init (Kerneloption) ..... 906
  - Init-System ..... 847
    - CentOS ..... 865
    - Debian ..... 867
    - Fedora ..... 865
    - Init-V-Prozess ..... 450, 857



- Init-V-Scripts* ..... 859
  - Kernelparameter* ..... 908
  - Protokoll* ..... 616
  - Raspbian* ..... 868
  - RHEL* ..... 865
  - SUSE* ..... 869
  - Ubuntu* ..... 867
  - Initial-RAM-Disk ..... 823, 906
  - initrd (Kerneloption) ..... 906
  - initrd-Datei ..... 820
    - selbst erzeugen* ..... 823
  - initrd (GRUB) ..... 835
  - initrd16/initrdefi (GRUB) ..... 836
  - inittab ..... 858
  - inkrementelle Backups ..... 1238
  - InnoTek ..... 1299
  - inputrc ..... 343
  - insmod ..... 877
  - insserv ..... 860
  - install (in modprobe.conf) ..... 882
  - Installation ..... 47
    - Anleitungen* ..... 83
    - Benutzerverwaltung* ..... 72
    - externe Festplatten* ..... 45
    - Grundkonfiguration* ..... 72
    - Grundlagen* ..... 39
    - Linux deinstallieren* ..... 81
    - Netzwerkinstallation* ..... 45
    - Netzwerkkonfiguration* ..... 73
    - Probleme* ..... 75
    - root-Passwort* ..... 72
    - SUSE* ..... 112
    - Tastaturprobleme* ..... 75
    - Updates* ..... 78
    - Varianten* ..... 44
  - inted ..... 871
  - Interface (Netzwerkschnittstelle) ..... 922
  - interfaces ..... 996
  - interfaces (Samba-Konfiguration) ..... 952
  - Internationalisierter Domainname ..... 946
  - Internationalisierung ..... 592
  - Internet
    - Gateway* ..... 961
    - Gateway (Client-Konfiguration)* ..... 944
    - Gateway (Server-Konfiguration)* ..... 970
    - Masquerading* ..... 970
    - Netzwerkgrundlagen* ..... 920
    - Printing Protocol (IPP)* ..... 1042
    - Router* ..... 970
    - Sicherheit* ..... 1251
  - Internet Service Daemon ..... 870
  - ionice ..... 440, 1245
  - iotop ..... 436
  - IP-Adresse ..... 920, 924
  - IP-Filter ..... 1262
  - ip-Kommando ..... 481
    - addr* ..... 935
    - addr show* ..... 938
    - link* ..... 935
    - route* ..... 936
  - IP-Nummer ..... 920, 924
  - IP-Ports ..... 920
    - Liste* ..... 1252
  - ip6tables ..... 1265
  - IPng ..... 925
  - IPP ..... 1042
  - iptables ..... 1265
    - Beispiel* ..... 1274
    - Masquerading* ..... 971
  - IPv6
    - Apache* ..... 1079
    - deaktivieren* ..... 906
    - Debian, Ubuntu* ..... 953
    - DenyHosts* ..... 1062
    - Dovecot* ..... 1182
    - Fedora, Red Hat* ..... 951
    - Firewall* ..... 1265, 1274
    - Grundlagen* ..... 925
    - Mail-Server* ..... 1150
    - manuelle Konfiguration* ..... 938
    - MySQL und MariaDB* ..... 1127
    - Nameserver* ..... 945
    - NFS* ..... 1027
    - Postfix* ..... 1160
    - Samba* ..... 997
    - SSH-Server* ..... 1060
    - TCP-Wrapper* ..... 1259
  - IR-Empfänger ..... 315
  - IR-Fernbedienung ..... 291
  - irrecord ..... 291
  - irw ..... 293
  - ISO-10646-Zeichensatz ..... 593
  - ISO-8859-Zeichensätze ..... 593
  - iso9660-Dateisystem ..... 746, 754, 784
  - iw ..... 932
- ## J
- 
- j-Kommando ..... 392
  - J8-Header ..... 299
  - Java ..... 687
  - javac ..... 688
  - jed ..... 335, 519
  - Jessie ..... 93
  - JetDirect (HP-Netzwerkdrucker) ..... 1049

- jmacs ..... 335, 519
  - Jobs regelmäßig ausführen ..... 454, 459
  - joe ..... 337, 519
  - Jokerzeichen ..... 355, 396
    - Komplikationen ..... 397
  - Joliet-Extension ..... 746, 784
  - journal (Journaling-Modus) ..... 759
  - Journal (systemd) ..... 612
  - journal.conf ..... 621
  - journalctl ..... 620, 866
  - Journaling-Dateisysteme ..... 755
    - btrfs ..... 764
    - ext3/ext4 ..... 758
    - xfs ..... 778
  - jove ..... 335, 519
  - jpico ..... 337
  - jumpstats ..... 392
- K**
- 
- K3b ..... 199
  - kacpid ..... 452
  - Kaffeine ..... 256
  - Kalender
    - Evolution ..... 231
    - Gnome ..... 164
    - Lightning (Thunderbird) ..... 228
  - Kali Linux ..... 30
  - Kamera (Raspberry Pi) ..... 319
  - Kanal (WLAN) ..... 931
  - Kazam ..... 266
  - kbd ..... 568
  - kblockd ..... 452
  - KDE ..... 190
    - Bildschirmeinstellungen ..... 719
    - Screenshots ..... 264
    - Verzeichnis freigeben ..... 1012
  - KDE Wallet ..... 199
  - kdenetwork4-filessharing ..... 1012
  - kdevtmpps ..... 452
  - Kernel ..... 26, 890
    - Boot-Optionen ..... 904
    - Boot-Optionen (GRUB) ..... 826
    - Device Trees ..... 882
    - Dokumentation ..... 891
    - Einstellungen ändern ..... 908
    - Hotplug-Funktion ..... 608
    - installieren ..... 898
    - IP-Filter ..... 1262
    - kompilieren ..... 889, 897
    - Konfiguration feststellen ..... 894
    - konfigurieren ..... 894
    - Logging ..... 615
    - Module ..... 876
    - neueste Version ..... 892
    - Optionen ..... 882
    - Optionen (GRUB) ..... 826
    - Parameter ..... 908
    - Patches ..... 892
    - Prozesse ..... 452
    - Update (GRUB) ..... 825
  - Kernel Mode Setting ..... 694
  - kernel.img-Datei ..... 281
  - kexec ..... 899
  - keyboard-setup ..... 568
  - keymap.cson-Datei (Atom) ..... 553
  - keys-Dateien (Gnome MIME) ..... 182
  - kGraft ..... 901
  - khelperd ..... 452
  - kill ..... 439
  - killall ..... 439
  - KIPI ..... 241
  - Klammererweiterung ..... 356
  - kmmod ..... 876, 880
  - KMS ..... 691, 694
    - video (Kerneloption) ..... 907
  - knfsd ..... 452
  - Kodi ..... 283
  - Kommandos ..... 433
    - ausführen ..... 352
    - bedingt ausführen ..... 353
    - Eingabe ..... 344
    - im Hintergrund ausführen ..... 353
    - Kommandointerpreter ..... 341
    - regelmäßig ausführen ..... 454, 459
    - siehe auch Prozesse ..... 433
    - starten ..... 434
    - starten (bash) ..... 345
    - Substitution (bash) ..... 357
  - Konfiguration ..... 563
    - bash ..... 343
    - Benutzer einrichten ..... 576
    - Dateisystem ..... 752
    - Kernel ..... 889, 894
    - LAN ..... 911
    - Maus ..... 569
    - Netzwerk ..... 911
    - Passwort ..... 583
    - Prompt ..... 343
    - Schriftart ..... 569
    - Tastatur (Textkonsole) ..... 567
    - Textkonsole ..... 567
    - X ..... 710
    - Zeitzone ..... 570

- Konsole ..... 331  
*Schriftart* ..... 569  
*Tastatur* ..... 567  
*wechseln* ..... 330
- Kontakt ..... 232
- Kontakte  
*Evolution* ..... 231  
*Gnome* ..... 164  
*Thunderbird* ..... 227
- Konverter ..... 463  
*Audio* ..... 465
- kpartx ..... 1358
- kPatch ..... 901
- krandrrc ..... 719
- KRename ..... 198
- ksh ..... 342
- ksnapshot ..... 264
- ksoftirqd ..... 452
- Ksplice ..... 900
- kswapd ..... 452
- ksysguard ..... 437
- kthread ..... 452
- KTorrent ..... 237
- Kubuntu ..... 31, 122
- KVM ..... 1330, 1331  
*Backup* ..... 1245  
*Vagrant* ..... 1353
- kvm-ok ..... 1330
- kworker ..... 452
- L**
- 
- liOn ..... 593
- Lüftersteuerung ..... 604
- Label  
*/etc/fstab* ..... 752  
*root-Kerneloption* ..... 905
- lame ..... 466
- LAMP-Server ..... 1074  
*WSL* ..... 1415
- LAN ..... 911  
*NetworkManager* ..... 913  
*Netzwerkkonfiguration* ..... 920  
*Sicherheit* ..... 1251
- Landscape (Ubuntu) ..... 564
- LANG ..... 596, 597
- language-selector-gnome ..... 676
- laptop (Kerneloption) ..... 907
- L<sup>A</sup>T<sub>E</sub>X ..... 476
- Latin-Zeichensätze ..... 593
- Laufwerke (gnome-disks) ..... 743
- Laufwerksbuchstaben (C:, D:) ..... 724
- Lautstärke ..... 610
- LC\_ALL ..... 597
- LC\_COLLATE ..... 596
- LC\_CTYPE ..... 596
- LC\_MESSAGES ..... 596
- LC\_MONETARY ..... 596
- LC\_NUMERIC ..... 596
- LC\_PAPER ..... 596
- LC\_TIME ..... 596
- LC\_TYPE ..... 592
- ldconfig ..... 679
- ldd ..... 588, 678
- LD\_LIBRARY\_PATH ..... 679
- ld.so ..... 679
- Leap (openSUSE) ..... 110
- Lenses (Unity) ..... 206
- Lesezeichen synchronisieren (Firefox) ..... 215
- less ..... 334  
*/proc-Dateien* ..... 903
- let ..... 361
- Let's Encrypt (Zertifikate) ..... 1105
- LFS ..... 757
- lftp ..... 495
- LGPL ..... 33
- /lib  
*/firmware* ..... 933  
*/modules/\*/modules.dep* ..... 881  
*/modules* ..... 876, 877, 898
- libav-tools ..... 467
- libc ..... 678
- libcap ..... 425
- libdbus ..... 609
- libdrm ..... 695
- libgudev ..... 608
- libguestfs ..... 1358, 1359
- libinput ..... 691, 715
- libinput-list-devices ..... 715
- libpam-google-authenticator ..... 1069
- Libraries ..... 677
- LibreELEC ..... 284
- librsvg2 ..... 465
- libsolv ..... 636
- libtiff ..... 465
- libudisks2 ..... 608
- libvirt ..... 1332  
*SSH* ..... 1340  
*Vagrant* ..... 1353
- libvirtd ..... 1332
- libwrap ..... 1259
- libzyp ..... 638
- lightdm ..... 707
- Lightning ..... 1222
- Limit ..... 1084

|                                     |                    |                                   |               |
|-------------------------------------|--------------------|-----------------------------------|---------------|
| Line Feed .....                     | 468                | Logical Volume Manager .....      | 59, 798       |
| Link-Local-Adressen (IPv6) .....    | 926                | Login .....                       | 139           |
| Links (Hard und Soft Links) .....   | 400                | <i>Name</i> .....                 | 579           |
| Linus Torvalds .....                | 35                 | login.defs .....                  | 583           |
| Linux .....                         | 25                 | loginit .....                     | 855           |
| <i>deinstallieren</i> .....         | 81                 | Logische Partition .....          | 53            |
| <i>Distribution</i> .....           | 28                 | LogLevel .....                    | 1082          |
| <i>Entstehung</i> .....             | 32                 | LOGNAME-Variable .....            | 362           |
| <i>Installation</i> .....           | 39, 47             | logrotate .....                   | 616           |
| <i>Kernel kompilieren</i> .....     | 889                | <i>Apache</i> .....               | 1081          |
| <i>Kernelmodule</i> .....           | 876                | <i>Samba</i> .....                | 999           |
| <i>Konfiguration</i> .....          | 563                | logwatch .....                    | 617           |
| <i>Linux Standard Base</i> .....    | 29                 | Lokale Netze .....                | 911           |
| <i>Shutdown</i> .....               | 141                | <i>Sicherheit</i> .....           | 1251          |
| <i>Startprobleme</i> .....          | 76                 | lokale Variablen .....            | 360           |
| <i>Systemveränderungen</i> .....    | 78                 | Lokalisierung .....               | 592           |
| <i>Updates</i> .....                | 78                 | Loopback-Device .....             | 748           |
| <i>Voraussetzungen</i> .....        | 39                 | Loopback-Interface .....          | 922           |
| linux-Schlüsselwort (GRUB) .....    | 835                | lostfound .....                   | 427, 762      |
| Linux Mint .....                    | 106, 122           | lp .....                          | 1045          |
| <i>Paketverwaltung</i> .....        | 669                | lpadmin .....                     | 1041, 1046    |
| LIRC .....                          | 291                | lpc .....                         | 1046          |
| lircd .....                         | 293, 316           | lpd .....                         | 1040          |
| Listen .....                        | 1087               | lpinfo .....                      | 1046          |
| Live-System .....                   | 29, 45             | lptions .....                     | 1040, 1046    |
| <i>Ubuntu</i> .....                 | 123                | lpq .....                         | 1045          |
| Livna-Paketquelle .....             | 105                | lprm .....                        | 1046          |
| Lizenzen .....                      | 32                 | lpstat .....                      | 1046, 1049    |
| llvmpipe .....                      | 704                | ls .....                          | 393           |
| lm-sensors .....                    | 600                | LSB .....                         | 29            |
| ln .....                            | 400                | lsblk .....                       | 599, 731      |
| .local-Verzeichnis .....            | 183                | lsmod .....                       | 878           |
| locale .....                        | 597                | lsop .....                        | 1255          |
| localectl .....                     | 568, 595, 596, 856 | lspci .....                       | 599, 605, 878 |
| Locales/Internationalisierung ..... | 592                | lspci .....                       | 605, 703      |
| localhost .....                     | 921, 943           | lsusb .....                       | 599, 605      |
| localmodconfig .....                | 896                | LTS-Version (Ubuntu) .....        | 121           |
| local_recipient_maps .....          | 1172               | <i>Enablement Stack</i> .....     | 130           |
| locate .....                        | 405                | <i>Support Status</i> .....       | 674           |
| lockd .....                         | 452                | Lubuntu .....                     | 122           |
| log file .....                      | 998                | LUKS .....                        | 809           |
| Logrotate .....                     | 1081               | luksFormat .....                  | 811           |
| logger .....                        | 615                | Luminance HDR .....               | 247           |
| Logging .....                       | 611                | lvcreate .....                    | 800, 1244     |
| <i>Apache</i> .....                 | 1081               | lvextend .....                    | 801           |
| <i>Docker</i> .....                 | 1375               | LVM .....                         | 798           |
| <i>Logrotate</i> .....              | 616                | <i>Backup mit Snapshots</i> ..... | 1244          |
| <i>Logwatch</i> .....               | 617                | <i>Grundlagen</i> .....           | 59            |
| <i>MySQL</i> .....                  | 1141               | <i>RAID</i> .....                 | 798           |
| <i>Postfix</i> .....                | 1162               | <i>Snapshots</i> .....            | 802           |
| <i>Samba</i> .....                  | 998                | <i>TRIM</i> .....                 | 808           |
| <i>X</i> .....                      | 704                | lvremove .....                    | 1244          |
| Logical Volume .....                | 60                 | LXDE (Raspbian) .....             | 279           |

LXSS ..... 1417  
 Lynx ..... 471, 496  
 lzop ..... 1232, 1233, 1245

## M

m-a (module-assistent) ..... 888  
 m23 ..... 564, 625  
 MAC ..... 1257, 1284  
 MAC-Adresse ..... 922, 981  
   *feststellen* ..... 938  
 mac80211-Framework ..... 933  
 Machine Owner Keys (Secure Boot) ..... 821  
 macOS  
   *Dateisystem* ..... 746  
   *Samba* ..... 1024  
   *Time Machine* ..... 1032  
 Macromedia Flash ..... 218  
 MacVTap-Device ..... 1337  
 madplay ..... 466  
 Magic-Dateien ..... 404  
 Mail *siehe* E-Mail ..... 1143  
 MAIL (Variable) ..... 362  
 Mail-Server ..... 1143  
   *Fehlersuche* ..... 1205  
   *IPv4* ..... 1150  
 Mailbox ..... 1146, 1155  
   *Dovecot* ..... 1183  
 mailcap ..... 404  
 maildir-Format ..... 1146  
 Maildir-Postfach  
   *Dovecot* ..... 1183  
   *Mutt* ..... 498  
   *Postfix* ..... 1170  
 mailq ..... 1162  
 Main-Pakete ..... 666  
 main.cf-Datei (Postfix) ..... 1156, 1157  
 Major Device Number ..... 429  
 make ..... 684  
 make-ssl-cert ..... 1102  
 makepasswd ..... 584, 1059, 1174  
 makethumbs ..... 368  
 man ..... 339  
 Mandatory Access Control ..... 1284  
 Mangle-Tabelle (iptables) ..... 1264  
 Manjero ..... 29  
 Manuelle Netzwerkkonfiguration ..... 948  
 map to guest = bad user ..... 1009  
 mapfile ..... 375  
 MariaDB ..... 1123  
   *als Docker-Container ausführen* ..... 1374  
 Markdown ..... 477  
   *Emacs-Erweiterung* ..... 544  
 Masquerading ..... 970  
   *Fedora* ..... 972  
   *FTP* ..... 493, 973  
   *Probleme* ..... 973  
 Master Boot Record ..... 822  
   *wiederherstellen* ..... 82  
 master.cf-Datei (Postfix) ..... 1161  
 MATE ..... 185  
 Maus  
   *blockiert* ..... 439  
   *KDE* ..... 203  
   *per Tastatur steuern* ..... 143  
   *Textmodus* ..... 569  
   *X* ..... 143  
 max log size ..... 999  
 maxcpus (Kerneloption) ..... 907  
 mb ..... 1249  
 mbox-Format ..... 1146  
 Mbox-Postfach ..... 1155  
 MBR ..... 734, 822  
   *Partitionsnummern* ..... 730  
   *wiederherstellen* ..... 82  
 md ..... 452  
 MDA ..... 1144  
 mdadm ..... 790, 793  
 mdadm.conf ..... 791  
 md\_mod (LVM) ..... 798  
 md\_mod (RAID) ..... 791  
 mdnsd ..... 451  
 /media ..... 427  
 Medien-Server ..... 1013  
 Medienfreigabe ..... 159  
 medusa ..... 1061  
 Meld ..... 160  
 MELPA (Emacs-Erweiterungen) ..... 544  
 Memtest86 ..... 601  
 mencoder ..... 468  
 menu.lst (GRUB) ..... 827  
 Mesa-Bibliothek ..... 691, 694  
   *mesa-utils* ..... 704, 1307  
 mhdfs ..... 748  
 Microsoft  
   *Exchange Server* ..... 1145  
   *Joliet-Extension* ..... 746  
   *KVM-Installation* ..... 1346  
   *SMB-Protokoll* ..... 988  
   *Subsystem for Linux* ..... 1405  
   *TrueType-Fonts* ..... 173  
   *VSCode* ..... 555  
   *Windows-Partitionen* ..... 780  
   *WSL* ..... 1405

- Midori ..... 222
- migration ..... 452
- Milter
  - ClamAV* ..... 1193
  - OpenDKIM* ..... 1202
  - SpamAssassin* ..... 1188
- MIME
  - CUPS (drucken)* ..... 1041
  - Firefox* ..... 216
  - Gnome* ..... 182
  - KDE* ..... 203
  - Konfiguration* ..... 402
- mime.convs ..... 1041
- mime.types ..... 404, 1041
- Minor Device Number ..... 429
- Mint ..... 31, 106
  - mintbackup* ..... 109
  - mintdrivers* ..... 109
  - mintinstall* ..... 109, 669
  - mintnanny* ..... 109
  - mintstick* ..... 109
  - mintupdate* ..... 108, 669
  - Paketverwaltung* ..... 669
- Mirroring ..... 58
- MIT-Lizenz ..... 33
- mkconf ..... 791
- mke2fs ..... 760
- mkfs.btrfs ..... 765
- mkfs.ntfs ..... 784
- mkfs.xfs ..... 779
- mkinitramfs ..... 824
- mklablel ..... 735
- mkpasswd ..... 584, 1059
- mkswap ..... 789, 790
- mlocate ..... 406
- /mnt ..... 427
- mode2 ..... 316
- ModeLine ..... 712
- modinfo ..... 879
- modprobe ..... 878
- modprobe.conf ..... 880, 935
- Module ..... 876
  - Abhängigkeiten* ..... 881
  - automatisch laden* ..... 881
  - Device Trees* ..... 882
  - Device-Dateien* ..... 881
  - kompilieren* ..... 885, 897
  - Optionen* ..... 882
  - Parameter* ..... 879
  - Versioning* ..... 877
  - verwenden* ..... 877
- module-assistant ..... 888
- modules.dep ..... 881
- mogrify ..... 464
- MOKs (Secure Boot) ..... 821
- Monitor (X-Konfiguration) ..... 711
- monitors.xml ..... 718
- Monolithischer Kernel ..... 895
- more ..... 334
- Mosh ..... 1068
- mount ..... 750, 751
  - Beispiele* ..... 751
  - Optionen* ..... 754
  - remount für Systempartition* ..... 751
- mp3ogg ..... 466
- mpage ..... 470
- MPEG-2-Decodierer ..... 288
- mpg123 ..... 466
- mpg321 ..... 466
- MPlayer ..... 256
- msdos-Dateisystem ..... 746
- msttcorefonts ..... 173
- MTA ..... 1144
- mtab ..... 750
- MUA ..... 1144
- Muffin (Window Manager) ..... 108
- Mule (Emacs) ..... 545
- Multiarch-Verzeichnisse ..... 680
- Multicast-Adressen (IPv6) ..... 926
- multiuser-Target ..... 705
- MultiViews ..... 1084
- Munin ..... 437
- Musik (Gnome) ..... 251
- Musik-Verzeichnis ..... 183
- Musique ..... 254
- mutt ..... 497
- Mutter-Programm (Gnome Shell) ..... 691
- mv ..... 398
  - Dateien umbenennen* ..... 398
  - Sicherheitsabfragen* ..... 105
- MX-Eintrag (DNS) ..... 1150
- mydestination ..... 1159, 1175
- myhostname ..... 1158
- mylvmbackup ..... 1141
- mynetworks ..... 1159
- myorigin ..... 1158
- MySQL ..... 1123
  - Administration* ..... 1133
  - Backups* ..... 1138
  - IPv6* ..... 1127
  - mysql-Kommando* ..... 1133
  - mysqladmin* ..... 1134
  - mysqldump* ..... 1139
  - Workbench* ..... 1135
  - WSL* ..... 1415
- Mythbuntu ..... 122

## N

- Nachtmodus ..... 168
- nachträgliche Installation ..... 78
- Nagios ..... 437
- Name Service Switch ..... 590
- Nameserver
  - Client-Konfiguration* ..... 923, 944
  - Server-Konfiguration (Dnsmasq)* ..... 978
  - Ubuntu* ..... 917
- Namespaces (Docker) ..... 1385
- NameVirtualHost ..... 1090
- nano ..... 337
- NAS-Geräte (Backups) ..... 1231
- NAT ..... 970
- NAT-Tabelle (iptables) ..... 1264
- Nautilus
  - MIME* ..... 182
  - nautilus-compare* ..... 160
  - nautilus-image-converter* ..... 160
  - nautilus-image-manipulator* ..... 160
  - nautilus-open-terminal* ..... 160
  - nautilus-pastebin* ..... 160
  - nautilus-share* ..... 1011
  - Verzeichnis freigeben* ..... 1011
- ncrack ..... 1061
- negativo-Paketquelle (Fedora) ..... 700
- Nemo (Datei Manager) ..... 108
- Neon ..... 31, 190
- net-tools ..... 1254
- Netatalk ..... 1032
- NetBIOS ..... 988
- Netfilter ..... 1262
- Netpbm ..... 464
- netplan ..... 957
- netstat ..... 1254
- Network File System ..... 746
- Network Time Protocol ..... 572
- Network-Maske ..... 922
- networkd ..... 956
- NetworkManager ..... 911
- Netzwerk ..... 911
  - Aktivität überwachen* ..... 1254
  - Brücke* ..... 1354
  - Ethernet-Controller konfigurieren* ..... 934
  - Drucker* ..... 1049
  - Grundlagen* ..... 920
  - Konfiguration* ..... 948
  - Netzwerk-Controller* ..... 934
  - Schnittstelle* ..... 922
  - Server-Konfiguration* ..... 961
  - Sicherheit* ..... 1251
- Neustart des Hostsystems ..... 1334
- newaliases ..... 1147, 1171
- newgrp
  - Beispiel* ..... 418
- Nextcloud ..... 1207
  - Backups* ..... 1215
  - Dateien synchronisieren* ..... 1217
  - Interna* ..... 1215
  - Updates* ..... 1216
- nfs-Dateisystem ..... 746
- NFS ..... 1025
  - /etc/fstab* ..... 1030
  - Geschwindigkeit (Server)* ..... 1027
  - IPv6* ..... 1027
  - NFS 4* ..... 1025
  - root* ..... 1028
  - Server* ..... 1025
- nfsd ..... 452
- nft ..... 1266
- nftables ..... 1266
- nginx ..... 1074
- NIC ..... 921
- nice ..... 440
- nl80211-Schnittstelle ..... 933
- nmap ..... 1255
- nmbd ..... 992
- nmcli ..... 916
- noapic (Kerneloption) ..... 907
- noauto ..... 754
- nodeadkeys ..... 567
- nodev ..... 754
- nodev-Dateisysteme ..... 747
- noexec ..... 754
- nohide (NFS) ..... 1028
- noht (Kerneloption) ..... 907
- nolapic (Kerneloption) ..... 907
- nomodeset (Kerneloption) ..... 907
- Non-Free-Pakete ..... 666
- none-Dateisystem ..... 748
- noresume (Kerneloption) ..... 908
- no\_root\_squash (NFS) ..... 1028
- nosmp (Kerneloption) ..... 907
- no\_subtree\_check (NFS) ..... 1028
- nosuid ..... 754
- Notebook
  - Batterie* ..... 601
  - Lüftersteuerung* ..... 604
- Notfall
  - Dateisystem reparieren* ..... 756
  - Init-V-Prozess umgehen* ..... 906
  - Linux-Startprobleme* ..... 76
  - Windows-Startprobleme* ..... 77
- NPAPI-Plugins ..... 217
- nplan ..... 957

- nproc ..... 600  
 nscd ..... 591, 592  
 NSS ..... 591  
 ntfs-Dateisystem ..... 746, 780  
     *Streams* ..... 783  
 ntfsclone ..... 784  
 ntfsinfo ..... 784  
 ntfslabel ..... 784  
 ntfsprogs ..... 726, 781, 784  
 ntfsresize ..... 784  
 ntfsundelete ..... 784  
 NTP ..... 572  
 ntpd ..... 572  
 ntpdate ..... 572  
 ntpq ..... 575  
 Nuvola Player ..... 255  
 nvidia-Treiber (X)  
     *Debian* ..... 99  
     *Fedora* ..... 106  
     *nvidia-settings* ..... 702  
     *openSUSE* ..... 120  
     *Treiberinstallation* ..... 700  
     *Ubuntu* ..... 129  
 NWID (WLAN) ..... 931
- O**
- 
- OCFS ..... 723  
 ocfs-Dateisystem ..... 748  
 OCICLI ..... 671  
 Öffentlich-Verzeichnis ..... 183  
 oggdec ..... 466  
 oggenc ..... 466  
 One-Click-Install (openSUSE) ..... 671  
 Oneshot-Typ (systemd) ..... 862  
 Online-Dokumentation ..... 145  
 Online-Konten (Gnome) ..... 164  
 Open GL ..... 694  
 Open Source ..... 32  
 openbsd-inetd ..... 871  
 OpenCL ..... 694  
 OpenDKIM ..... 1197  
 OpenJDK ..... 687  
 openresolv ..... 946, 949  
 openssh ..... 1057  
 openssl ..... 1097, 1166  
 openSUSE ..... 31  
     *Samba* ..... 1011  
     *Snapper* ..... 775  
 OpenWrt ..... 963  
 /opt ..... 427  
 Optimus-Hybrid-Grafik ..... 697  
 Optionen (Kernel) ..... 894  
 options (modprobe.conf) ..... 882  
 Options (Apache) ..... 1084  
 Oracle  
     *Cluster Filesystem* ..... 723  
     *Java* ..... 687  
     *Linux* ..... 31, 85  
     *MySQL* ..... 1123  
     *VirtualBox* ..... 1299  
 Order (Apache) ..... 1085  
 ordered (Journaling-Modus) ..... 758  
 Origin-Patterns ..... 651  
 os-prober ..... 833  
 OSMX ..... 284  
 Overclocking (Raspberry Pi) ..... 320  
 Overlay-Dateisystem ..... 748  
     *Docker* ..... 1399  
 Overlays (Device Trees) ..... 883  
 ownCloud ..... 1207  
 owner ..... 754
- P**
- 
- P1-Header ..... 299  
 p7zip ..... 1233  
 PackageKit ..... 655  
 packagekitd ..... 655  
 Packer ..... 1316  
 PAE ..... 886  
 Pakete ..... 627  
     *Abhängigkeiten* ..... 628  
     *Debian* ..... 640, 666  
     *Format ändern* ..... 656  
     *Multiarch* ..... 680  
     *Paketmanager* ..... 670  
     *Red Hat* ..... 627  
     *Ubuntu* ..... 673  
     *Verwaltung* ..... 623  
 Paketfilter ..... 1262  
 PAM ..... 587  
     *Google Authenticator* ..... 1069  
     *pam-auth-update* ..... 588  
     *pam\_cracklib* ..... 584  
     *pam\_faillock* ..... 585  
     *pam\_pwquality* ..... 584  
     *pam\_unix* ..... 584  
     *systemd* ..... 854  
 Pandoc ..... 479  
     *Atom* ..... 554  
 Panel  
     *Gnome* ..... 148  
     *KDE* ..... 193  
     *Unity* ..... 208



- Papierkorb (Samba) ..... 1010
- Parallel SSH ..... 1068
- Parametersubstitution ..... 375
- Paravirtualisierung ..... 1335
- Parity Striping ..... 58
- parted ..... 737
  - EFI-Partition* ..... 818
- Partition
  - ändern, Linux* ..... 64
  - Bezeichnung unter Linux* ..... 728
  - Dateisystem* ..... 69
  - EFI* ..... 818
  - Grundlagen* ..... 51
  - ideale Partitionierung* ..... 66
  - im Verzeichnisbaum* ..... 724
  - Partitionsname* ..... 752
  - remount* ..... 751
  - Typen* ..... 53
- passdb-backend ..... 994, 1001
- PasswordAuthentication (sshd\_config) 1412
- Passwort ..... 581
  - ändern* ..... 583
  - Ablaufdatum (chage)* ..... 583
  - aging* ..... 583
  - Apache* ..... 1087
  - für Gruppen* ..... 586
  - PAM* ..... 587
  - Qualität* ..... 584
  - root* ..... 583
  - Samba* ..... 1000
  - vergessen* ..... 584
- patch-Kommando ..... 683, 893
- Patches (Kernel) ..... 892
- Patente ..... 36
- path ..... 1006
- PATH ..... 345, 362
  - Einstellung ändern* ..... 362
- Pattern (ZYpp) ..... 640
- pavucontrol ..... 258, 611
- pci (Kerneloption) ..... 906, 908
- PCI-Bus ..... 605
- PCM-Lautstärke ..... 610
- pdbedit ..... 1001
- PDC ..... 991
- PDF
  - pdf2ps* ..... 472
  - pdf90* ..... 476
  - pdfedit* ..... 476
  - pdfimages* ..... 476
  - pdfinfo* ..... 476
  - pdfjam* ..... 476
  - pdfjoin* ..... 476
  - pdfnup* ..... 476
  - pdftops* ..... 472
  - pdftotext* ..... 476
  - pdksh* ..... 342
  - PostScript-Konverter* ..... 471
  - Tools* ..... 475
- Pepper-Plugins (Flash) ..... 219
- Perfect Forward Secrecy ..... 1167
- Periodische Ausführung von Jobs ..... 454, 459
- pesign ..... 892
- PGP ..... 1150
- Phonon ..... 611
- PHP ..... 1116
  - Emacs-Erweiterung* ..... 544
  - phpMyAdmin* ..... 1136
  - Unicode* ..... 1079
- Physical Device ..... 60
- Physical Extent ..... 60
- Physical Volume ..... 60
- pico ..... 337
- PID ..... 436
- PID-Datei ..... 437
- pidof ..... 437
- pinfo ..... 340
- ping ..... 482
- pip ..... 1247
- Pipes ..... 350
- Pixel-Desktop ..... 279
- pkcon ..... 655
- pkexec ..... 442, 448
- pkmon ..... 655
- Plasma ..... 190
- Plasmoids ..... 192
- Plesk Panel ..... 564
- Plex ..... 284
- Pluggable Authentication Modules ..... 587
- Plugins
  - Firefox* ..... 217
  - Flash* ..... 218
  - Yum* ..... 633
- pnuke ..... 1068
- Policy-Dateien (X) ..... 449
- polycoreutils-gui ..... 1287
- PolicyKit ..... 447
- POP-Server ..... 1144, 1179
  - Authentifizierung* ..... 1184
- Poppler ..... 476
- Port-Nummer
  - FTP (20, 21)* ..... 1252
  - HTTP (80)* ..... 1252
  - IMAP (25, 587)* ..... 1146
  - Liste* ..... 1252
  - Referenz* ..... 1252
  - SMTP (25, 587)* ..... 1146



## R

- radeon-Treiber ..... 696
  - Firmware-Dateien* ..... 99
- RAID ..... 57
  - LVM* ..... 798
  - RAID-0* ..... 58
  - RAID-1* ..... 58
  - RAID-10* ..... 58
  - Scrubbing* ..... 797
  - TRIM* ..... 808
  - Überwachung* ..... 791
- RANDOM-Variable ..... 363
- RANDOM\_DELAY-Variable ..... 459
- RandR ..... 694, 716
- Raspberry Pi ..... 267
  - Device Trees* ..... 882
  - Kamera* ..... 319
  - Kodi* ..... 283
  - Raspbian* ..... 272
- Raspbian ..... 31, 271, 272
  - NTP* ..... 575
  - Systemstart* ..... 868
  - WLAN-Konfiguration* ..... 954
- Raspbmc ..... 284
- raspistill ..... 319
- raspivid ..... 319
- Raspplex ..... 284
- RAW-Bilddateien ..... 246
- RAW-Format ..... 246, 1336
- rb ..... 1249
- rc-Dateien ..... 859
- rc.local ..... 867
- rdiff-backup ..... 1238
- RDP-Server ..... 692
  - Wayland* ..... 692
- read ..... 377
- readline ..... 343
- Reboot des Hostsystems ..... 1334
- reboot-required-Datei ..... 652
- Rechnername siehe Hostname ..... 985
- Rechnerstart ..... 817
  - Probleme* ..... 76
- recode ..... 468
- recordMyDesktop ..... 265
- recover-file (Emacs) ..... 521
- recycle ..... 1010
- Red Hat ..... 31, 84
  - automount* ..... 747
  - Gateway-Konfigurationsdatei* ..... 944
  - initrd-Datei* ..... 825
  - LABEL in /etc/fstab* ..... 752
  - RHN (Red Hat Network)* ..... 564, 625
  - statische Netzwerkkonfiguration* ..... 950
  - sudo* ..... 446
- redshift-Programm ..... 168
- ReFS-Dateisystem ..... 780
- Regelmäßige Ausführung von Jobs ..... 454, 459
- Reguläre Ausdrücke (Emacs) ..... 535
- reject ..... 1046
- Rekonq ..... 222
- relayhost ..... 1159
- reload (Init-V-Prozess) ..... 859
- RemainAfterExit-Schlüsselwort (systemd) ..... 862
- Remmina ..... 171
- remount (Systempartition) ..... 751
- remove (modprobe.conf) ..... 882
- Remove-Unused-Dependencies (APT) ..... 652
- Rendezvous ..... 958
- renice ..... 440
- Require ..... 1085, 1089
- reserve (Kerneloption) ..... 906
- reset ..... 335
- resize2fs ..... 762
- resolv.conf-Datei ..... 944
  - Ubuntu* ..... 945
- resolvconf-Paket ..... 946, 953
- restart (Init-V-Prozess) ..... 859
- restorecon ..... 1286, 1290
- Retina-Bildschirme ..... 168
- Reverse DNS ..... 1153
- RFCs ..... 146
- RHEL ..... 84
  - Systemstart* ..... 865
  - Tastatur* ..... 568
- RHN ..... 564
- RHSM ..... 84
- Rhythmbox ..... 251
- Richard Stallman ..... 35
- Ripper (CDs/DVDs einlesen) ..... 261
- rlogin ..... 485
- rm-Sicherheitsabfragen ..... 105
- rmmod ..... 879
- ro (Kerneloption) ..... 906
- Rockridge-Extension ..... 746, 784
- Rolling Release ..... 111
  - Ubuntu-Kernel* ..... 130
- /root ..... 427
- root ..... 72, 583
  - Kerneloption* ..... 905
  - MySQL* ..... 1127
  - NFS* ..... 1028
- Root-Partition ..... 66
- root-Passwort vergessen ..... 584
- Root-Server ..... 1057

- root\_squash (NFS) ..... 1028
  - route ..... 936
  - Router (Masquerading) ..... 970
  - Routing-Tabelle ..... 922
  - rpc.idmapd ..... 1026
  - rpcinfo ..... 1031
  - rpciod ..... 452
  - rpi-update ..... 281
  - rpm ..... 627
    - Beispiele* ..... 629
    - cannot open packages database* ..... 629
    - Datenbank reparieren* ..... 629
    - Quellcodepakete installieren* ..... 683
  - RPM Fusion ..... 105
  - RPMS ..... 628
  - rsnapshot ..... 1240
  - rsvg ..... 465
  - rsvg-convert ..... 465
  - rsync ..... 1235
  - rsyslog.conf ..... 612
  - rsyslogd ..... 612
  - Ruhezustand ..... 602
  - /run ..... 427, 747
    - /log/journal* ..... 620
  - RUN (Dockerfile) ..... 1391
  - run-crons ..... 458
  - run-Kommando (Docker) ..... 1381
  - run-parts ..... 458
  - Runlevel ..... 858
  - runtime linker ..... 679
  - rygel ..... 1012
- S**
- 
- S/MIME ..... 1150
  - S3-Cloud-Dienst ..... 1246
  - Samba ..... 987, 988
    - /etc/fstab* ..... 1022
    - Fedora* ..... 1011
    - Firewall* ..... 996
    - Gäste* ..... 1009
    - Inbetriebnahme* ..... 992
    - IPv6* ..... 997
    - Nautilus* ..... 157
    - Netzwerkverzeichnisse einrichten* ..... 1006
    - Papierkorb* ..... 1010
    - Passwörter* ..... 1000
    - RHEL* ..... 1011
    - SELinux* ..... 998
    - Sicherheitsmechanismen* ..... 990
    - SUSE* ..... 1011
    - Ubuntu* ..... 1011
  - Sandbox (Flatpak/Snap) ..... 660
  - /sbin ..... 427
    - /init* ..... 857, 858
    - /init.d* ..... 869
  - Schlüssel
    - HTTPS (Apache)* ..... 1095
    - POP/SMTP (Dovecot)* ..... 1184
    - SSH* ..... 1064
  - Schlafmodus ..... 601
  - Schleifen ..... 381
  - Schnittstelle ..... 922
  - Schriftart ..... 173, 595
    - Emacs* ..... 542
    - Textkonsolen* ..... 569
  - Scientific Linux ..... 30, 85
  - scp ..... 488
  - Screen-Abschnitt (X) ..... 714
  - Screencast ..... 264
  - Screenshots ..... 264
    - Wayland* ..... 692
  - Script
    - bash* ..... 371
    - Programmierung* ..... 364
  - ScriptAlias ..... 1083
  - Scripts
    - bash* ..... 363
    - SSH* ..... 487
    - Vagrant* ..... 1324
  - Scrubbing (RAID) ..... 797
  - SCSI ..... 728
  - scsi\_eh ..... 452
  - SD-Karte formatieren ..... 725
  - seahorse ..... 1065
  - seahorse-nautilus ..... 160
  - search (GRUB) ..... 835
  - Secure Boot ..... 43, 77, 820
  - Secure Shell ..... 1057
    - WSL* ..... 1412
  - Secure Sockets Layer ..... 1095
  - security (Samba) ..... 994
  - securityfs ..... 1292
  - sed-Beispiel ..... 398
  - Selektor (Syslog) ..... 612
  - SELinux ..... 1283, 1284
    - AirPrint* ..... 1053
    - Apache* ..... 1075
    - opendkim* ..... 1199
    - Samba* ..... 998
    - selinux-policy-mls* ..... 1287
    - SSH* ..... 1066
    - SSH-Port* ..... 1060
  - Sender Policy Framework ..... 1195
  - sensors ..... 600

- Server
  - cron* ..... 454
  - Datenbank (MySQL)* ..... 1123
  - DHCP* ..... 976
  - FTP (vsftpd)* ..... 1118
  - Nameserver (DNS)* ..... 977
  - Netzwerk* ..... 961
  - NFS* ..... 1025
  - Samba* ..... 988
  - SSH* ..... 1057
  - Webserver (Apache)* ..... 1073
  - X* ..... 690
- Server Message Block ..... 988
- server role (Samba) ..... 994
- server string ..... 994
- ServerAdmin ..... 1082
- ServerAlias ..... 1090
- ServerName ..... 1090
- ServerName (Apache) ..... 1079
- ServerSignature ..... 1082
- service-Kommando ..... 857, 859
- services-Datei ..... 871
- Services (Hintergrunddienste) ..... 450
- sestatus ..... 1289
- set ..... 356
- set ..... 361
- setcap ..... 425
- setenforce ..... 1290
- setfacl ..... 422
- setfattr ..... 424
- Setgid-Bit ..... 416
- setsebool ..... 1288
- Setuid-Bit ..... 415
- setup.exe ..... 627
- sfconvert ..... 467
- sftp ..... 493
  - Server* ..... 1059
- SGI-Dateisystem ..... 745
- shadow ..... 582
- /share ..... 428
- Share-Level-Sicherheit ..... 990
- Shared Folder (VirtualBox) ..... 1311
- Shared Libraries ..... 677, 678
- Shares (Samba) ..... 990
- Sharing (Vagrant) ..... 1327
- Shebang ..... 364
- Shell ..... 341
  - Programmierung* ..... 363
  - Script-Beispiele* ..... 363, 461
  - Standard-Shell ändern* ..... 342
  - Variablen* ..... 359, 371
- Shim ..... 44, 820
- shopt ..... 356
- Shotwell ..... 238
- showmount ..... 1031
- shutdown ..... 141
  - des Hostsystems* ..... 1334
- Shutter ..... 265
- Shuttleworth, Mark (Ubuntu) ..... 120
- Sicherheit ..... 1251
  - Apache* ..... 1087
  - WLAN* ..... 931
- Sicherheitskontext ..... 1285
- Sid ..... 667
- Simple Storage Service (S3) ..... 1246
- single (Kerneloption) ..... 906
- Single-User-Modus (systemd) ..... 852
- Site-Local-Adressen (IPv6) ..... 926
- skip-networking (MySQL/MariaDB) ..... 1126
- Smack ..... 1285
- SMART ..... 803
- smartd ..... 806
- SMB-Protokoll ..... 988
  - Version 1 deaktivieren* ..... 997
- SMB-Versionen ..... 989
- smb.conf ..... 993
- smbclient ..... 1023
- smbd ..... 992
- smbfs-Dateisystem ..... 746, 1020
- smbpasswd ..... 1001
- smbstatus ..... 993, 996
- smbtree ..... 1024
- SMTP ..... 1146
  - Authentifizierung* ..... 1185
  - Fehlersuche* ..... 1205
- smtp\_tls-Parameter (Postfix) ..... 1164
- smtpd\_tls-Parameter (Postfix) ..... 1164
- Snakeoil-Zertifikat und -Schlüssel
  - Apache* ..... 1102
  - Postfix* ..... 1162
- Snap (Ubuntu) ..... 662
- snapt ..... 664
- Snapper (openSUSE) ..... 775
- Snapshots
  - btrfs* ..... 770
  - LVM* ..... 802
- socat ..... 1351
- Socket-API (Netzwerkdrucker) ..... 1049
- Socket-Dateien ..... 400
- soft\_bounce (Postfix) ..... 1160
- Software-Installation ..... 623
- Software-Patente ..... 36
- software-properties ..... 675
- software-properties-gtx ..... 645
- Software-RAID ..... 57
- Solus Desktop ..... 31

- Sonderzeichen (bash) ..... 386
- Sound Converter ..... 259
- Sound Juicer ..... 261
- Sound-System (ALSA) ..... 609
- source ..... 372
- sources.list ..... 644
- sox ..... 467
- Spam-Schutz ..... 1187
- spamass-milter ..... 1189
- SpamAssassin ..... 1187
- speaker-test ..... 610
- special bits (Zugriffsrechte) ..... 415
- SPF-Eintrag (Mail-Server) ..... 1195
- Spice ..... 1337
- Spin (Fedora) ..... 100
- splash ..... 904
- Spooling-System (drucken) ..... 1038
- Spracheinstellung ..... 592
- squashfs-Dateisystem ..... 748
  - Snap* ..... 664
- Squeeze ..... 93
- SRPM-Pakete ..... 628, 683
- /srv ..... 428
  - /ftp* ..... 1120
  - /www* ..... 1075
- SSD-TRIM ..... 807
- SSH ..... 485, 1057
  - absichern* ..... 1059
  - Dateisystem* ..... 490
  - Google Authenticator* ..... 1069
  - Konqueror* ..... 199
  - IPv6* ..... 1060
  - libvirt* ..... 1340
  - Login vermeiden* ..... 1064
  - Port ändern* ..... 1060
  - Portumleitung* ..... 1044
  - SELinux* ..... 1066
  - Server* ..... 1057
  - Tunnel* ..... 488
  - unter Windows* ..... 1412
- ssh-agent ..... 1065
- ssh-keygen ..... 1064
- sshd ..... 1057
  - WSL* ..... 1412
- sshfs-Dateisystem ..... 490, 747
- SSID ..... 974
- SSID (WLAN) ..... 930
- SSL ..... 1095
- SSL (Apache) ..... 1095
- ssl-cert-snakeoil.key ..... 1102
- ssl-cert-snakeoil.pem ..... 1102
- SSLCACertificateFile (Apache) ..... 1103
- SSLCertificateChainFile ..... 1104
- SSLCipherSuite (Apache) ..... 1101, 1104
- SSLEngine (Apache) ..... 1101
- SSLProtocol (Apache) ..... 1101, 1104
- SSLStrictSNIVHostCheck (Apache) ..... 1095
- SSLxxxFile (Apache) ..... 1101
- Stable-Pakete ..... 666
- Stallman, Richard ..... 33
- Standardausgabe ..... 349
- Standardeingabe ..... 349
- star ..... 423
- start.elf-Datei ..... 281
- Startprobleme ..... 76
- STARTTLS
  - Dovecot* ..... 1184
  - Postfix* ..... 1162
- startx ..... 709
- stat ..... 412
- Statisch gelinkte Programme ..... 678
- Statische Netzwerkkonfiguration ..... 948
- Sticky-Bit ..... 417, 419
- Streams (NTFS-Dateisystem) ..... 783
- Strict Transport Security ..... 1109
- stripcomments (bash-Beispiel) ..... 365
- Striping ..... 58
- Stromsparfunktionen ..... 601
- su ..... 442
  - grafische Variante* ..... 442
  - Wayland* ..... 692
- submission (Postfix) ..... 1161
- Substitutionsmechanismen (bash) ..... 354
- subtree\_check (NFS) ..... 1028
- Subvolumes (btrfs) ..... 768
- suchen
  - Dateien* ..... 404
  - Emacs* ..... 534
  - find und grep* ..... 407
- sudo ..... 443
  - Ein-/Ausgabeumleitung* ..... 444
  - Fedora* ..... 446
  - Raspbian* ..... 280
  - Ubuntu* ..... 445, 447
  - Wayland* ..... 692
- suid ..... 415
- SUSE
  - AppArmor* ..... 1291
  - CIFS* ..... 1022
  - Firewall* ..... 1271
  - Gateway-Konfigurationsdatei* ..... 944
  - Init-Prozess* ..... 869
  - Kernelkonfiguration* ..... 895
  - Paketverwaltung* ..... 670
  - Samba* ..... 1011
  - Snapper* ..... 775

- Updates* ..... 672
  - VirtualBox* ..... 1306
  - Suspend to Disk ..... 601
  - Kerneloptionen* ..... 908
  - SVG-Konverter ..... 465
  - Swap-Datei ..... 789
  - Swap-Partition ..... 68
    - einbinden* ..... 787
    - einrichten* ..... 789
  - swapon ..... 789, 790
  - swappiness-Parameter ..... 788
  - symbolische Links ..... 401
  - Synaptic ..... 653
    - ohne Passwort ausführen* ..... 446
  - sync (S3) ..... 1248
  - sync (NFS) ..... 1027
  - Syntaxhervorhebung ..... 539
  - /sys ..... 428, 904
    - /kernel/security* ..... 1292
  - sysctl ..... 908, 970
  - sysfs-Dateisystem ..... 747
  - syslog ..... 999
  - System Security Services Daemon ..... 592
  - system-config-lvm ..... 798
  - system-config-printer ..... 165, 1047
  - system-config-samba ..... 1011
  - system-config-selinux ..... 1287
  - system-config-services ..... 867
  - system-config-users ..... 576
  - System-V-Init-Prozess ..... 857
  - Systemadministration ..... 563
  - systemctl ..... 459, 850
  - systemd ..... 848
    - als Cron-Ersatz* ..... 459
    - CentOS* ..... 865
    - eigene Service-Datei* ..... 861
    - Fedora* ..... 865
    - Firewall-Beispiel* ..... 1280
    - Grafiksystem starten* ..... 705
    - Netzwerk-Device-Namen* ..... 947
    - Netzwerkkonfiguration* ..... 956
    - Netzwerkschnittstellen* ..... 922
    - Prozesse periodisch ausführen* ..... 459
    - RHEL* ..... 865
    - Timers* ..... 459
  - systemd-journald ..... 619
  - systemd-networkd ..... 956
  - systemd-sysv-generator ..... 856
  - systemd-timedated ..... 570
  - systemd-timesyncd ..... 573
  - systemd-udev ..... 430
  - systemd-udev ..... 430
  - systemd-vconsole-setup ..... 569
  - Systemeinstellungen (KDE) ..... 199
  - Systempartition ..... 66
    - remount* ..... 751
  - systemsettings ..... 199
  - Systemstart ..... 139
    - GRUB* ..... 817
    - Init-V* ..... 857
    - systemd* ..... 848
- ## T
- 
- Tabulatoren (Emacs) ..... 529
  - tail ..... 334
  - Taktfrequenz ..... 600
  - tar ..... 683, 1232, 1234
  - targeted ..... 1287
  - Tartarus ..... 1245
  - tasksel ..... 649
  - Tastatur ..... 141
    - bash* ..... 343
    - blockiert* ..... 439
    - Gnome* ..... 162
    - KDE* ..... 203
    - Konfiguration* ..... 567
    - US-Tastatortabelle* ..... 75
  - Tastenkürzel ..... 141
    - Linux* ..... 141
  - TCP-Wrapper-Bibliothek ..... 1258
  - TCP/IP ..... 920
  - tcsh ..... 342
  - TDB ..... 1001
  - TeamViewer ..... 172
  - tee ..... 351
  - telnet ..... 485, 490
    - SMTP-Fehlersuche* ..... 1205
  - Temperatur (CPU) ..... 600
  - Temperaturmessung (Raspberry Pi) ..... 313
  - Terminal ..... 330
  - Termine
    - Evolution* ..... 231
    - Lightning (Thunderbird)* ..... 228
  - test-Kommando (bash) ..... 379
  - Tethering ..... 915
  - Text-Konverter ..... 468
  - Textdatei
    - durchsuchen* ..... 408
    - PostScript-Konverter* ..... 469
  - Texteditoren ..... 335, 519
  - Textkonsole ..... 330
    - Konfiguration* ..... 567
    - Schriftart* ..... 569
    - Tastatur* ..... 567

- Themen (KDE) ..... 202  
 Themes (Gnome) ..... 179  
 Thumbnails ..... 368, 463  
 Thunderbird ..... 222  
   *CalDAV/CardDAV* ..... 1222  
 tiff2pdf ..... 465  
 tiff2ps ..... 465  
 tigervnc-viewer ..... 171  
 Tilde ..... 144, 390  
 time-sync ..... 572  
 timedatectl ..... 570, 572, 856  
 Timers (systemd) ..... 459  
 TinyCore ..... 32  
 TLS ..... 1149  
   *Dovecot* ..... 1184  
   *Postfix* ..... 1162  
 /tmp ..... 428  
 tmpfs-Dateisystem ..... 747  
 top ..... 436  
 Torrent ..... 237  
 Torvalds, Linus ..... 35  
 Totem ..... 256  
 Touchpad deaktivieren ..... 163  
 traceroute ..... 483  
 Transmission ..... 237  
 Transport Layer Security ..... 1149  
 Treiberinstallation (Ubuntu) ..... 675  
 TRIM (SSDs) ..... 807  
 Troll Tech ..... 190  
 Trusted TLS Connection (Postfix) ..... 1164  
 TSOP4838 ..... 315  
 Tumbleweed ..... 111, 672  
 tune2fs ..... 761  
 Tunnel (SSH) ..... 488  
 TurboPrint ..... 1047  
 tvservice ..... 319  
 type name ..... 346  
 Type-Schlüsselwort (systemd) ..... 862
- ## U
- 
- uappexplorer-cli ..... 664  
 Ubuntu ..... 31  
   *AirPrint* ..... 1052  
   *als Docker-Image* ..... 1368  
   *Bildschirmeinstellungen* ..... 719  
   *DKMS* ..... 887  
   *Dnsmasq* ..... 917  
   *initrd-Datei* ..... 824  
   *Paketverwaltung* ..... 673  
   *sudo* ..... 445, 447  
   *Systemstart* ..... 867  
   *Tastatur* ..... 567  
   *Unity* ..... 204  
   *VirtualBox* ..... 1306  
 Ubuntu Server ..... 122  
 Ubuntu Studio ..... 122  
 ubuntu-drivers ..... 129, 675  
 ubuntu-restricted-extras ..... 129  
 ubuntu-support-status ..... 674  
 udev ..... 430, 608  
   *Netzwerk-Device-Namen* ..... 947  
 udf-Dateisystem ..... 746, 784  
 udisks2 ..... 609  
 UDP ..... 920, 1253  
 UEFI ..... 41  
   *Partition* ..... 43  
   *Secure Boot* ..... 43, 77, 820  
 ufw ..... 1272  
   *Docker* ..... 1367  
 Uhrzeit ..... 570  
 UID ..... 579  
 ulimit ..... 440  
 umask ..... 419  
 Umgebungsvariablen ..... 360  
 umount ..... 785, 1030  
 unattended-upgrades ..... 650  
 Unicode ..... 593  
   *Apache* ..... 1079  
   *Dateisystem* ..... 389  
   *drucken* ..... 471  
   *Emacs* ..... 545  
   *Konsole* ..... 569  
   *PHP* ..... 1079  
   *PostScript* ..... 471  
   *UTF* ..... 593  
   *Zeichensatz* ..... 593  
 unionfs-Dateisystem ..... 748  
 Unity ..... 204, 205  
 Unity Tweak Tool ..... 212  
 Univention Corporate Server ..... 564  
 Universal Disk Format ..... 746  
 Unix ..... 25  
 Unix Pseudo TTYs ..... 747  
 unix2dos ..... 468  
 unset ..... 361  
 Unstable-Pakete ..... 666, 667  
 Untrusted TLS Connection (Postfix) ..... 1164  
 unxz ..... 1232  
 unzip ..... 1233  
 update-alternatives ..... 658  
 update-ca-certificates ..... 1164  
 update-grub ..... 828  
 update-initramfs ..... 824  
 update-manager ..... 676



- update-ms-fonts ..... 173
  - Update-Patch ..... 892
  - updatedb ..... 406
  - Updates ..... 78
    - LibreELEC* ..... 293
  - UPnP ..... 296
  - upower ..... 609
  - Upstart (Ubuntu) ..... 867
  - US-Tastaturtabelle ..... 75
  - USB ..... 605
    - Laufwerke* ..... 786
    - USB-Stick formatieren* ..... 725
  - usb-creator-gtk ..... 129
  - User einrichten ..... 576
  - User Shares (Samba) ..... 1009
  - User Themes ..... 179
  - User-Level-Sicherheit ..... 990
  - useradd ..... 577
  - username map ..... 1003
  - usershare allow guests ..... 1010
  - user\_xattr ..... 421
  - /usr ..... 428
  - UTC (Universal Time, Coordinated) ..... 570
  - UTF-16 ..... 593
  - UTF-8 ..... 593
    - MySQL/MariaDB* ..... 1126
  - UUID
    - einstellen (ext3/ext4)* ..... 762
    - einstellen (xfs)* ..... 780
    - ermitteln* ..... 752
    - in /dev/disk* ..... 732
    - in /etc/fstab* ..... 752
- ## V
- 
- Vagrant ..... 1316
    - libvirt/KVM* ..... 1353
  - VagrantFile ..... 1317, 1322
  - valid users ..... 1006
  - /var ..... 428, 747
    - /ftp* ..... 1120
    - /lib/docker* ..... 1400
    - /lib/dpkg/alternatives* ..... 659
    - /lib/rpm/alternatives* ..... 659
    - /log/Xorg.0.log* ..... 704
    - /log/journal* ..... 620
    - /run* ..... 437
    - /spool/cron/tabs* ..... 454
    - /www* ..... 1075
  - Variablen (bash) ..... 359, 371, 377
  - varlock-Dateisystem ..... 747
  - varrun-Dateisystem ..... 747
  - vboxadd ..... 1306
  - vboxdrv ..... 1300
  - vboxmanage ..... 1314
  - vboxnetadp ..... 1300
  - vboxnetflt ..... 1300
  - vboxpci ..... 1300
  - vboxsf-Dateisystem ..... 1312
  - vboxvideo ..... 1306
  - VCL-Decodierer ..... 288
  - vcgencmd ..... 289, 322
  - VDDPAU ..... 694
  - Vergleiche (bash) ..... 379
  - Verschlüsselung ..... 61
    - Dateien* ..... 809
    - Dateisysteme* ..... 722, 809
    - Mail-Server* ..... 1149
  - Verzeichnis ..... 144, 390, 425
    - Multiarch* ..... 680
    - Partitionen* ..... 724
  - Verzweigungen (bash) ..... 378
  - VESA-Modi ..... 714
  - VESA-Treiber (X) ..... 713
  - vfat-Dateisystem ..... 746, 780
  - vga-Treiber (X) ..... 714
  - vgcreate ..... 800
  - vgscan ..... 800
  - Vi ..... 335, 501
  - video (Kerneloption) ..... 907
  - Video-Codecs ..... 288
  - Videos (DVDs) abspielen ..... 786
  - Videos-Verzeichnis ..... 183
  - Vim ..... 335, 501
    - Cursorbewegung* ..... 505
    - Easy-Modus* ..... 517
    - Konfiguration* ..... 514
    - Makros* ..... 517
    - Maus* ..... 516
    - Optionen* ..... 513
    - suchen und ersetzen* ..... 510
    - Swap-Datei* ..... 515
    - Tabulatoren* ..... 516
    - Unicode* ..... 515
    - Zeichensatz* ..... 515
  - vimrc-Datei ..... 514
  - Vinagre ..... 171
  - Virenschutz ..... 1193
  - virsh ..... 1332, 1348
  - virt-cat ..... 1361
  - virt-clone ..... 1351
  - virt-df ..... 1359
  - virt-edit ..... 1361
  - virt-filesystems ..... 1360
  - virt-inspector ..... 1360

- virt-make-fs ..... 1361
  - virt-manager ..... 1332, 1339
  - virt-resize ..... 1361
  - virt-tar-in ..... 1361
  - virt-tar-out ..... 1361
  - virt-top ..... 1353
  - virt-viewer ..... 1352
  - virt-viewer vmname ..... 1352
  - virtio-Treiber ..... 730, 1335
    - Windows* ..... 1346
  - Virtual Private Networks ..... 915
  - virtual\_alias\_domains ..... 1175
  - VirtualBox ..... 1299
  - VirtualHost (Apache) ..... 1090
  - virtual\_mailbox\_domains ..... 1177
  - Virtuelle Dateisysteme ..... 747
  - Virtuelle Domänen (Postfix) ..... 1175
  - Virtuelle Hosts ..... 1089
    - mit HTTPS* ..... 1095
  - Virtuelle Postfächer ..... 1176
  - VISUAL ..... 337
  - Visual Studio Code ..... 555
  - visudo ..... 444
  - vmlinux ..... 898
  - vmlinux-Datei ..... 819
  - VNC ..... 171
    - Server* ..... 692
    - Wayland* ..... 692
  - vncviewer ..... 172
  - VolFS-Dateisystem ..... 1418
  - vol\_id ..... 752
  - VOLUME (Dockerfile) ..... 1393
  - Volume Group ..... 60
  - Volumes
    - Docker* ..... 1384
    - Docker, löschen* ..... 1401
    - LVM* ..... 60
  - vorbis-tools ..... 466
  - Vordergrundprozesse ..... 434
  - Vorlagen-Verzeichnis ..... 183
  - VPFY-Kommando (Postfix) ..... 1179
  - VSCode ..... 555
  - vsftpd ..... 1119
- W**
- 
- w3m ..... 471, 496
  - WannaCry-Schad-Software ..... 997
  - Warteschlange ..... 1038
  - watchdog ..... 452
  - Wayland ..... 689
    - Einschränkungen* ..... 692
  - Web-Apps (Ubuntu) ..... 205
  - Webalizer ..... 1112
  - Webbrowser (Textmodus) ..... 496
  - WebDAV ..... 1119
  - Webmin ..... 564
  - Webserver ..... 1073
  - website ..... 1249
  - WebUpd8-Paketquelle (Java) ..... 688
  - Webverzeichnis absichern ..... 1087
  - well-known-DAV-Umleitungen ..... 1211
  - WEP ..... 931
  - Weston ..... 692
  - wget ..... 494
  - Wheezy ..... 93
  - whereis ..... 346, 405
  - which ..... 405
  - while (bash) ..... 383
  - Whitelist (SpamAssassin) ..... 1190
  - WiFi (WLAN) ..... 929
  - Wildcard-Zertifikate ..... 1109
  - Window Manager ..... 690
  - Windows
    - CUPS-Netzwerkdrucker drucken* ..... 1052
    - Dateisystem* ..... 746, 780
    - Drucker* ..... 1047
    - Hibernate* ..... 781
    - KVM-Installation* ..... 1346
    - MBR wiederherstellen* ..... 82
    - Netzwerkverzeichnisse* ..... 988, 1020
    - Samba-Freigaben nutzen* ..... 1024
    - Startprobleme* ..... 77
    - Subsystem for Linux* ..... 1405
  - winff ..... 467
  - WINS ..... 988
  - WiringPi ..... 305
  - WLAN ..... 929
    - Access Point* ..... 929, 966
    - Access-Point* ..... 915
    - Adapter* ..... 929
    - Authenticator (hostapd)* ..... 973
    - LibreELEC* ..... 286
    - NetworkManager* ..... 914
    - Raspberry Pi* ..... 282
    - Router* ..... 929
    - Sicherheit* ..... 931
  - wmf2eps ..... 465
  - wmf2gd ..... 465
  - wmf2svg ..... 465
  - workgroup ..... 994
  - Workgroup-Sicherheit ..... 990
  - WPA ..... 931, 941, 975
  - WPA2 ..... 931, 941
  - wpa\_passphrase ..... 941

wpa\_supplicant ..... 282  
 wpasupplicant ..... 941  
   *in /etc/network/interfaces* ..... 954  
 writeable ..... 1006  
 writeback (Journaling-Modus) ..... 759  
 WSL ..... 1405

## X

X ..... 689  
   *Auflösung* ..... 714  
   *Farbanzahl* ..... 714  
   *Grafikkarte* ..... 713  
   *Konfiguration* ..... 710  
   *Logging* ..... 704  
   *Maus* ..... 143  
   *Monitor-Konfiguration* ..... 711  
   *Protokoll* ..... 704  
   *Server* ..... 690  
   *SSH* ..... 487  
   *Version feststellen* ..... 703  
   *Window Manager* ..... 690  
   *Window System* ..... 689  
 X11R6 ..... 690  
 xargs ..... 358  
 XBian ..... 284  
 XBMC ..... 283  
 xconsole ..... 615  
 XDG ..... 183  
 xdg-desktop-icon ..... 183  
 xdg-desktop-menu ..... 183  
 xdg-email ..... 184  
 xdg-icon-resource ..... 183  
 xdg-mime ..... 183  
 xdg-open ..... 184  
 xdg-screensaver ..... 184  
 xdg-user-dirs ..... 183  
 xdg-user-dirs-gtk ..... 183  
 xdm ..... 707  
 xdpinfo ..... 703  
 xdpinfo ..... 703  
 xfs-Dateisystem ..... 745, 778  
 xfs\_admin ..... 780  
 xfs\_check ..... 779  
 xfs\_growfs ..... 780  
 xfs\_repair ..... 779  
 xine ..... 256  
 xinetd ..... 871  
 xinput ..... 715  
 xkill ..... 439  
 Xorg.O.log ..... 704  
 xorg.conf ..... 710

xpdf-utils ..... 476  
 XPI ..... 217  
 xrandr ..... 716  
 XRender ..... 694  
 xsensors ..... 601  
 xterm ..... 331  
 Xubuntu ..... 31, 122  
 XWayland ..... 691  
 xz ..... 1232, 1233

## Y

YaST ..... 110  
   *Online Updates* ..... 672  
   *Paketverwaltung* ..... 670  
   *YOU* ..... 672  
 Yorba ..... 234  
 YOU (YaST Online Update) ..... 669, 672  
 yum ..... 631, 634  
   *automatische Updates* ..... 636  
 yum-cron ..... 636  
 yum-utils ..... 635  
 yumdownloader ..... 635  
 yumex ..... 636

## Z

Zahlenvergleiche (bash) ..... 379  
 Zeichenketten  
   *bash* ..... 358  
   *Parametersubstitution (bash)* ..... 375  
 Zeichensatz ..... 592, 593  
   *ändern* ..... 468  
   *Apache* ..... 1079  
   *PHP* ..... 1079  
 Zeitgesteuerte Job-Ausführung ..... 454, 459  
 Zeitzone ..... 570  
   *glibc* ..... 571  
 Zentyal ..... 122, 564  
 ZENworks ..... 564, 625  
 Zero Install ..... 627  
 Zeroconf ..... 958  
 Zertifikate  
   *HTTPS (Apache)* ..... 1095  
   *Let's Encrypt* ..... 1106  
   *POP/SMTP (Dovecot)* ..... 1184  
   *Postfix* ..... 1162

|   |      |
|---|------|
| <i>selbst erstellen und signieren</i> ..... | 1097 |
| <i>Snakeoil-Zertifikat</i> .....            | 1102 |
| ZFS-Dateisystem .....                       | 745  |
| zile .....                                  | 519  |
| zip .....                                   | 1233 |
| zipinfo .....                               | 1233 |
| Zorin OS .....                              | 31   |
| zsh .....                                   | 342  |
| Zugriffsbits .....                          | 410  |
| <i>bei neuen Dateien</i> .....              | 419  |
| <i>setuid, setgid</i> .....                 | 415  |
| <i>sticky</i> .....                         | 417  |
| Zugriffsrechte .....                        | 409  |
| Zugriffssteuerung .....                     | 576  |
| Zwei-Faktor-Authentifizierung .....         | 1069 |
| Zwischenablage (VirtualBox) .....           | 1311 |
| ZYpp .....                                  | 638  |
| zypper .....                                | 639  |



# Die Serviceseiten

Im Folgenden finden Sie Hinweise, wie Sie Kontakt zu uns aufnehmen können.

## Lob und Tadel

Wir hoffen sehr, dass Ihnen dieses Buch gefallen hat. Wenn Sie zufrieden waren, empfehlen Sie das Buch bitte weiter. Wenn Sie meinen, es gebe doch etwas zu verbessern, schreiben Sie direkt an den Lektor dieses Buches: *christoph.meister@rheinwerk-verlag.de*. Wir freuen uns über jeden Verbesserungsvorschlag, aber über ein Lob freuen wir uns natürlich auch!

Auch auf unserer Webkatalogseite zu diesem Buch haben Sie die Möglichkeit, Ihr Feedback an uns zu senden oder Ihre Leseerfahrung per Facebook, Twitter oder E-Mail mit anderen zu teilen. Folgen Sie einfach diesem Link: *http://www.rheinwerk-verlag.de/4465*.

## Zusatzmaterialien

Zusatzmaterialien (Beispielcode, Übungsmaterial, Listen usw.) finden Sie in Ihrer Online-Bibliothek sowie auf der Webkatalogseite zu diesem Buch: *http://www.rheinwerk-verlag.de/4465*. Wenn uns sinnentstellende Tippfehler oder inhaltliche Mängel bekannt werden, stellen wir Ihnen dort auch eine Liste mit Korrekturen zur Verfügung.

## Technische Probleme

Im Falle von technischen Schwierigkeiten mit dem E-Book oder Ihrem E-Book-Konto beim Rheinwerk Verlag steht Ihnen gerne unser Leserservice zur Verfügung: *ebooks@rheinwerk-verlag.de*.

## Über uns und unser Programm

Informationen zu unserem Verlag und weitere Kontaktmöglichkeiten bieten wir Ihnen auf unserer Verlagswebsite <http://www.rheinwerk-verlag.de>. Dort können Sie sich auch umfassend und aus erster Hand über unser aktuelles Verlagsprogramm informieren und alle unsere Bücher und E-Books schnell und komfortabel bestellen. Alle Buchbestellungen sind für Sie versandkostenfrei.

# Rechtliche Hinweise

In diesem Abschnitt finden Sie die ausführlichen und rechtlich verbindlichen Nutzungsbedingungen für dieses E-Book.

## Copyright-Vermerk

Das vorliegende Werk ist in all seinen Teilen urheberrechtlich geschützt. Alle Nutzungs- und Verwertungsrechte liegen beim Autor und beim Rheinwerk Verlag. Insbesondere das Recht der Vervielfältigung und Verbreitung, sei es in gedruckter oder in elektronischer Form.

© **Rheinwerk Verlag GmbH, Bonn 2017**

## Ihre Rechte als Nutzer

Sie sind berechtigt, dieses E-Book ausschließlich für persönliche Zwecke zu nutzen. Insbesondere sind Sie berechtigt, das E-Book für Ihren eigenen Gebrauch auszudrucken oder eine Kopie herzustellen, sofern Sie diese Kopie auf einem von Ihnen alleine und persönlich genutzten Endgerät speichern. Zu anderen oder weitergehenden Nutzungen und Verwertungen sind Sie nicht berechtigt.

So ist es insbesondere unzulässig, eine elektronische oder gedruckte Kopie an Dritte weiterzugeben. Unzulässig und nicht erlaubt ist des Weiteren, das E-Book im Internet, in Intranets oder auf andere Weise zu verbreiten oder Dritten zur Verfügung zu stellen. Eine öffentliche Wiedergabe oder sonstige Weiterveröffentlichung und jegliche den persönlichen Gebrauch übersteigende Vervielfältigung des E-Books ist ausdrücklich untersagt. Das vorstehend Gesagte gilt nicht nur für das E-Book insgesamt, sondern auch für seine Teile (z. B. Grafiken, Fotos, Tabellen, Textabschnitte).

Urheberrechtsvermerke, Markenzeichen und andere Rechtsvorbehalte dürfen aus dem E-Book nicht entfernt werden, auch nicht das digitale Wasserzeichen.



## Digitales Wasserzeichen

Dieses E-Book-Exemplar ist mit einem **digitalen Wasserzeichen** versehen, einem Vermerk, der kenntlich macht, welche Person dieses Exemplar nutzen darf. Wenn Sie, lieber Leser, diese Person nicht sind, liegt ein Verstoß gegen das Urheberrecht vor, und wir bitten Sie freundlich, das E-Book nicht weiter zu nutzen und uns diesen Verstoß zu melden. Eine kurze E-Mail an [service@rheinwerk-verlag.de](mailto:service@rheinwerk-verlag.de) reicht schon. Vielen Dank!

## Markenschutz

Die in diesem Werk wiedergegebenen Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. können auch ohne besondere Kennzeichnung Marken sein und als solche den gesetzlichen Bestimmungen unterliegen.

## Haftungsausschluss

Ungeachtet der Sorgfalt, die auf die Erstellung von Text, Abbildungen und Programmen verwendet wurde, können weder Verlag noch Autor, Herausgeber oder Übersetzer für mögliche Fehler und deren Folgen eine juristische Verantwortung oder irgendeine Haftung übernehmen.

## Über den Autor

**Dr. Michael Kofler** studierte Telematik an der TU Graz. Er zählt zu den erfolgreichsten und vielseitigsten Computerbuchautoren im deutschen Sprachraum. Zu seinen Themengebieten zählen neben Linux auch OS X, MySQL, KVM, Visual Basic und Excel-VBA. Viele seiner Bücher wurden übersetzt. Michael Kofler arbeitet auch als Software-Entwickler, Berater sowie als Lehrbeauftragter an zwei Fachhochschulen.