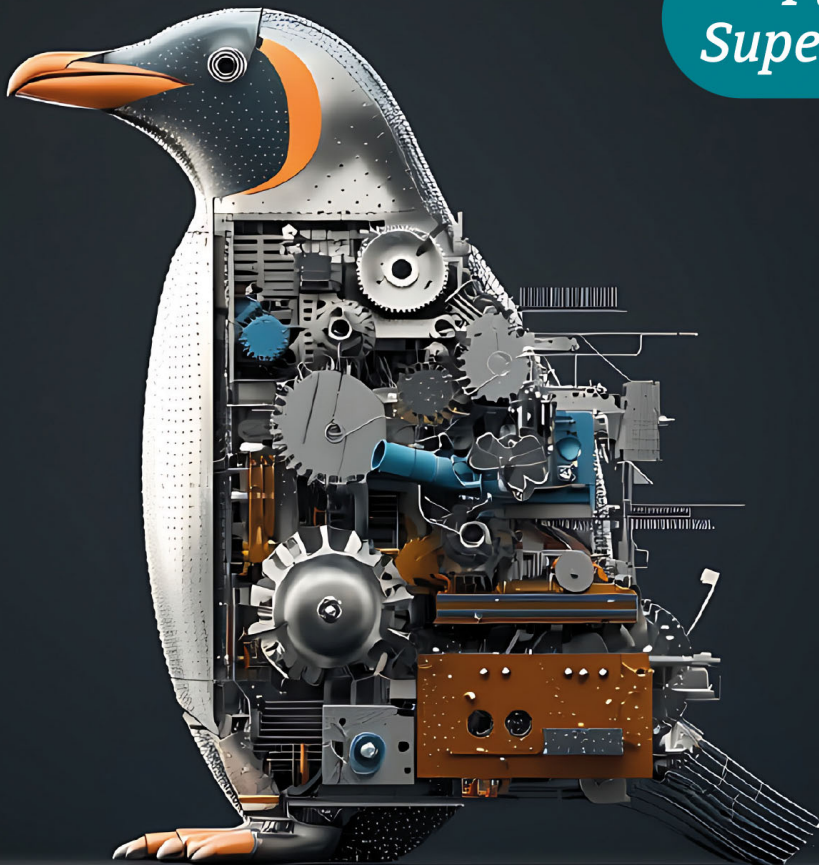


Robert Gödl

Linux intern verstehen

Ein systemnaher Einblick in Aufbau, Kernel
und Systemkomponenten

*Für
Superuser*



Inhaltsverzeichnis

	Einleitung	11
1	Linux-Grundlagen	13
1.1	Der Linux-Kernel	13
1.1.1	Prozess-Management	13
1.1.2	Hauptspeicher verwalten	14
1.1.3	Hardware verwalten	15
1.1.4	Systemaufrufe annehmen und abgeben	15
1.2	Die Verzeichnis-Hierarchie	15
1.3	Die Shell – arbeiten auf dem Terminal	17
1.3.1	Befehle als root (als Administrator) ausführen	19
1.3.2	Grundlegende Shell-Befehle	21
1.3.3	Fehlerausgaben von Befehlen verstehen	35
1.3.4	Texteditoren auf der Shell	36
1.3.5	Arbeiten mit komprimierten Archiven	39
1.3.6	Befehle werden nicht gefunden – der Pfad	41
1.4	Benutzer und Gruppen – die Rechteverteilung unter Linux	43
1.4.1	Benutzer anlegen	46
1.4.2	Benutzer löschen	47
1.4.3	Benutzer bearbeiten und Rechte vergeben	48
1.5	Rechte an Dateien und Verzeichnissen	49
1.5.1	Berechtigungen vergeben – ein Beispiel	52
1.6	Softlinks (Symlinks) und Hardlinks	54
1.6.1	Softlinks	54
1.6.2	Hardlinks	55
2	Arbeiten mit Hardware und Speichermedien	57
2.1	Die Hardware auflisten	57
2.2	Gerätedateien	59
2.2.1	Gerätedateien für Speichermedien (Festplatten, SSDs, USB-Sticks und mehr)	61
2.2.2	Weitere Gerätedateien	63
2.2.3	Geräteerkennung mit udev	64
2.3	Speichermedien, Partitionen, Dateisysteme	69

2.3.1	Mechanische Festplatten und SSDs	69
2.3.2	Partitionen – Aufteilen der Festplatte	70
2.3.3	Dateisysteme – Formatieren	73
2.3.4	Partitionen temporär einhängen/mounten	76
2.3.5	Partitionen dauerhaft einhängen/mounten	78
2.3.6	LVM – Logical Volume Manager	79
2.3.7	RAID unter Linux	91
2.3.8	Dateisystemfehler prüfen und korrigieren	96
3	Start des Betriebssystems	99
3.1	BIOS/UEFI – das Motherboard	99
3.1.1	UEFI und Linux	99
3.2	Der Linux-Bootloader	101
3.2.1	Problembehandlung beim Systemstart	102
3.2.2	GRUB-Shell – wenn GRUB nicht startet	103
3.3	Der Kernel startet	106
4	Prozesse steuern	107
4.1	Was sind Prozesse?	107
4.2	Prozesse anzeigen	107
4.2.1	Die Momentaufnahme mit ps	108
4.2.2	Der Standard-Prozessmonitor top	110
4.2.3	Der Profi-Prozessmonitor htop	111
4.2.4	iotop – Welche Prozesse greifen auf die Festplatte zu?	113
4.3	Prozesse steuern	114
4.3.1	Auf der Shell gestartete Prozesse abbrechen	114
4.3.2	Prozesse im Hintergrund starten	115
4.3.3	Bereits gestartete Prozesse in den Hintergrund verschieben	115
4.3.4	Mit im Hintergrund laufenden Prozessen arbeiten	115
4.3.5	Prozesse beenden	116
4.3.6	Die Priorität von Prozessen ändern	118
5	Init-Systeme, Services und zeitgesteuerte Aufgaben	121
5.1	Die Aufgaben des Init-Systems	121
5.2	Die verschiedenen Init-Systeme	122
5.3	Systemd – das meistgenutzte Init-System	122
5.3.1	Targets (Systemzustände)	123
5.3.2	Services (Dienste)	125
5.3.3	Timer – zeitgesteuerte Aufgaben	135
5.3.4	Sockets – Kommunikation zwischen Services und Systemd	140
5.4	SysVinit – der alte Standard	145

5.4.1	Systemzustände – Runlevel unter SysVinit	146
5.4.2	Services anzeigen und steuern	147
5.5	Aufgaben automatisch ausführen mit at, Cron und Anacron	153
5.5.1	Einmalige Aufgaben mit at	153
5.5.2	Zeitgesteuerte und wiederkehrende Aufgaben mit Cron ...	154
6	Log-Meldungen unter Linux	161
6.1	Log-Meldungen	161
6.1.1	Log-Meldungen verstehen	161
6.1.2	Der Aufbau eines Logs	162
6.2	Log-Meldungen unter Systemd	162
6.2.1	Log-Ausgabe zeitlich eingrenzen	164
6.2.2	Log-Ausgabe weiter einschränken	166
6.2.3	Log-Meldungen nach Level anzeigen	167
6.2.4	Das Log live ansehen	168
6.2.5	Start-Analyse des Systems	168
6.2.6	Logs auf der Festplatte begrenzen	171
6.2.7	Logs löschen	172
6.3	Log-Meldungen unter SysVinit und anderen Init-Systemen	173
6.3.1	Log-Ausgabe zeitlich einschränken	175
6.3.2	Ältere Logs löschen	175
6.3.3	Log-Dateien leeren	176
6.4	Wichtige Log-Dateien	176
6.4.1	dmesg – das Hardware-Log	176
6.4.2	wtmp.db (oder wtmp) – das Log der Benutzer- Anmeldungen	178
6.4.3	dpkg.log und apt-History-Log – Paket-Log unter auf Debian basierenden Linux-Distributionen	179
6.4.4	rpm_pkgs – das Paket-Log unter Fedora, Red Hat, openSUSE	181
6.4.5	Weitere Logs des Systems	182
6.4.6	Logs der Desktop-Umgebung	182
6.4.7	Logs der Benutzer-Anwendungen	183
7	Einstieg in die System-Konfiguration	185
7.1	Sicherung einer Konfiguration anlegen	186
7.2	Die Syntax der Konfigurationsdateien	187
7.3	Konfiguration des Bootloaders GRUB, /etc/default/grub	187
7.3.1	Zusätzliche Einstellungen aktivieren	188
7.3.2	Optische Einstellungen	190
7.4	Partitionen automatisch einhängen, /etc/fstab	192
7.4.1	Die Konfigurationsdatei /etc/fstab	192
7.4.2	Konfiguration bearbeiten	194

7.5	Administrator-Rechte, /etc/sudoers	196
7.6	Problematische Kernel-Module (Treiber) ausschließen, /etc/modprobe.d	198
7.7	Die Konfiguration der Debian-Paketverwaltung (Ubuntu, Linux Mint ...), /etc/apt.	201
7.7.1	Repositorys einbinden	202
7.8	Die Shell-Konfiguration.	204
7.8.1	Die Grundkonfiguration von Bash und Zsh.	204
7.8.2	Den Bash-Prompt anpassen.	207
7.8.3	Zsh – mehr Komfort auf dem Terminal.	212
8	Netzwerk-Konfiguration, SSH und Sicherheit	215
8.1	Netzwerk-Grundlagen	215
8.1.1	Netzwerk-Karten/Hardware.	215
8.1.2	TCP/IP – die Technik hinter dem Netzwerk/ IP-Adressen.	217
8.1.3	DNS – das Telefonbuch des Internets	221
8.1.4	Netzwerk-Konfiguration.	222
8.1.5	Statische IP-Adressen für Heim-Netzwerke	224
8.2	SSH – auf entfernten Linux-Computern arbeiten.	229
8.2.1	Kopieren von Dateien und Verzeichnissen über SSH.	231
8.2.2	Tmux – abgebrochene SSH-Sitzungen fortführen	232
8.2.3	SSH absichern	233
8.3	Die Firewall	234
8.3.1	Firewall-Grundlagen.	235
8.3.2	GUFW – die unkomplizierte Firewall mit grafischer Oberfläche.	236
8.3.3	UFW – die unkomplizierte Firewall auf der Shell	238
8.3.4	Iptables und Nftables – die professionelle Firewall	242
8.3.5	Welche Ports sind geöffnet?	249
8.4	Wireshark und tcpdump – wenn es im Netzwerk hängt	255
8.4.1	tcpdump – Wireshark für die Shell	258
9	Die grafische Oberfläche	261
9.1	Die grafische Oberfläche unter Linux	261
9.2	X-Server und Wayland – Kommunikation zwischen Linux und grafischer Oberfläche.	261
9.2.1	Konfiguration des X-Servers	262
9.2.2	Wayland konfigurieren.	264
9.3	Desktop-Umgebungen und Fenster-Manager.	265
9.3.1	KDE Plasma	266
9.3.2	GNOME.	267

9.3.3	XFCE	267
9.3.4	Cinnamon	268
9.3.5	Mate	269
10	Skripte und Programmieren.	271
10.1	Bash-/Shell-Skripte schreiben	271
10.1.1	Grundlagen für Shell-Skripte	272
10.1.2	Erweiterte Techniken	276
10.1.3	Ein Skript Schritt für Schritt aufbauen: Dateien und Verzeichnisse zählen	283
10.1.4	Skripte mit grafischer Oberfläche	287
10.1.5	Skripte testen und Fehler finden	298
10.2	Programmiersprachen und Bibliotheken unter Linux installieren und nutzen	300
10.2.1	Python	300
10.2.2	PHP	302
10.2.3	C/C++	303
10.2.4	Rust	303
10.2.5	Weitere Programmiersprachen	303
10.3	Anwendungen übersetzen	304
11	Software	307
11.1	Software unter Linux verwalten	307
11.1.1	Paket-Formate	307
11.1.2	Grafische Paket-Verwaltungen	308
11.1.3	Software auf der Shell verwalten	313
11.1.4	Flatpak- und Snap-Pakete verwalten	318
11.1.5	AppImages – ausführbare Dateien unter Linux	325
11.1.6	Software über das Python-Repository installieren	326
11.1.7	Rust-Software mittels Cargo verwalten	327
11.1.8	Software unter Linux kompilieren	329
11.1.9	Software über selbst extrahierende Skripte installieren	332
11.2	Treiber und Firmware	332
11.2.1	Treiber-Probleme erkennen	333
11.2.2	Installation von Treibern und Firmware	334
12	Zusätzliche Sicherheit durch LDAP/AD, automatische Entschlüsselung und Virtualisierung	343
12.1	LDAP und Active Directory	343
12.1.1	Linux an LDAP/AD anbinden	344
12.2	Verschlüsseltes Linux automatisch entschlüsseln	346
12.3	Daten von verschlüsselten Partitionen retten	349

12.4	Virtualisierung	351
	12.4.1 Hardware-Virtualisierung	351
	12.4.2 Container-Virtualisierung	353
A	Weiterführende Informationen	363
B	Glossar	365
	Stichwortverzeichnis	369

Linux-Grundlagen

Ein Betriebssystem hat im Hintergrund mehr zu tun, als nur Mausclicks entgegenzunehmen. Es startet den Computer sowie die nötigen Dienste, damit alles läuft, wie es soll. Meist stellt es eine grafische Oberfläche bereit, es steuert die Prozesse und regelt die Rechte der Benutzer.

Was im Hintergrund eines Betriebssystems passiert, ist recht komplex – in diesem Kapitel lesen Sie mehr darüber, welche Aufgaben Linux als Betriebssystem hat. Sie lernen die Verzeichnis-Hierarchie kennen, um zu verstehen, wo Sie die wichtigsten Dateien finden, und erfahren, wie Sie mit der Shell arbeiten.

1.1 Der Linux-Kernel

Der Kernel ist der Kern eines jeden Betriebssystems – unter Linux natürlich der Linux-Kernel. Der Kernel ist der erste Bestandteil, der vom Betriebssystem gestartet wird. Alles andere, was im Betriebssystem läuft, wird von diesem verwaltet bzw. gesteuert. Im Folgenden finden Sie die vier wichtigsten Aufgaben des Kernels, anschließend werden diese genauer beschrieben.

- **Prozess-Management** – Der Kernel verwaltet die Prozesse, die vom Betriebssystem oder vom Benutzer gestartet werden.
- **Hauptspeicher verwalten** – Der Kernel gibt an, welche Prozesse wie viel vom vorhandenen Hauptspeicher (auch Arbeitsspeicher oder RAM genannt) nutzen können.
- **Hardware verwalten** – Der Kernel ist die Schnittstelle zwischen Betriebssystem und Hardware. Hierfür nutzt der Kernel Treiber.
- **Systemaufrufe annehmen und abgeben** – Damit Anwendungen funktionieren, muss der Kernel von Prozessen bzw. Anwendungen Befehle – sogenannte *Systemaufrufe* – entgegennehmen und ebensolche den Prozessen/Anwendungen wieder zurücksenden.

1.1.1 Prozess-Management

In einem Betriebssystem laufen zur selben Zeit viele Prozesse, damit alles funktioniert, wie es soll. Starten Sie einen Webbrowser, ist dies mindestens ein Prozess (oft nutzen Anwendungen auch mehrere Prozesse für verschiedene Aufgaben).

Im Hintergrund laufen parallel jedoch viele weitere Prozesse – etwa einer für die Synchronisierung der Uhrzeit mit einem Server im Internet, ein anderer stellt die grafische Oberfläche bereit und andere warten darauf, dass Sie einen USB-Stick anschließen, um diesen nutzbar zu machen.

Der Kernel startet die nötigen Prozesse zur richtigen Zeit, pausiert solche, die gerade nicht benötigt werden, aber vielleicht später noch gebraucht werden. Natürlich beendet der Kernel solche Prozesse, die nicht mehr benötigt werden, auch wieder.

Er steuert auch, welche Prozesse wie viel Zeit von welchem Kern der CPU benutzen dürfen.

Man könnte meinen, alles am Computer würde zur selben Zeit geschehen, dem ist jedoch nicht so. Ein Prozessor kann nur eine gewisse Menge an Aufgaben erledigen (genauer gesagt errechnen). Wie schon beschrieben, laufen im Betriebssystem aber viele Prozesse zur selben Zeit. Der Kernel teilt ein, für wie lange ein Prozess die CPU nutzen kann. Oft wird ein Prozess kurz unterbrochen, damit ein anderer kurz die CPU nutzen kann, und wieder zurück. Dies geschieht in sehr kurzen Abständen, sodass man am Bildschirm von diesen kurzen Unterbrechungen nichts mitbekommt. Hier hat aber nicht der Kernel alleine die Kontrolle, sondern auch die CPU selbst hat einen gewissen Einfluss.

1.1.2 Hauptspeicher verwalten

Der Hauptspeicher, auch *Arbeitsspeicher* oder einfach *RAM* genannt, wird in aktuellen Computern zwar immer größer, doch auch dieser wird vom Kernel verwaltet, auch wenn moderne CPUs dem Kernel dabei helfen.

Jeder Prozess braucht nicht nur Zeit der CPU, sondern auch Platz im Hauptspeicher. Der Hauptspeicher ist im Grunde ein sehr schnelles Speichermedium, das jedoch nur begrenzt Platz hat. Prozesse können ihre Daten zwar auch auf der SSD oder der noch langsameren Festplatte speichern, können dadurch jedoch nur viel langsamer abgerufen werden, um von der CPU verarbeitet zu werden.

Damit der Hauptspeicher nicht irgendwann voll wird und zu berechnende Daten auf den langsameren Speicher ausgelagert werden, muss der Kernel den Hauptspeicher für jeden Prozess einteilen. Der Kernel regelt also, wie viel Platz jeder Prozess im Hauptspeicher bekommt sowie mit welchen anderen Prozessen diese Daten geteilt werden dürfen und wann die Daten aus diesem Speicher wieder entfernt werden.

Der Kernel selbst nutzt natürlich auch einen gewissen Bereich im Hauptspeicher – dies ist der sogenannte *Kernel-Space*.

1.1.3 Hardware verwalten

Der Kernel macht auch jegliche Hardware benutzbar – egal ob Tastatur oder Maus, Bildschirm, Webcam, USB-Stick und so weiter. Er sorgt dafür, dass Sie diese Geräte nutzen können.

Der Kernel erkennt eingebaute Hardware und wartet darauf, dass weitere Hardware – etwa über USB – angeschlossen wird, und nutzt dann die passenden Treiber, um mit der Hardware arbeiten zu können.

1.1.4 Systemaufrufe annehmen und abgeben

Innerhalb des Betriebssystems herrscht ständige Kommunikation. Starten Sie etwa eine Anwendung per Mausklick aus dem Anwendungsmenü oder per Befehl auf der Shell, muss der Kernel erfahren, was er tun soll.

Zur Kommunikation im Betriebssystem werden sogenannte *Systemaufrufe* genutzt. Systemaufrufe bestehen einfach gesagt aus Befehlen, mit denen der Kernel und die Anwendungen bzw. Prozesse umgehen können.

1.2 Die Verzeichnis-Hierarchie

Die Verzeichnis-Struktur von Linux ist anders aufgebaut als die von Windows. Unter Windows werden Partitionen oft *Laufwerke* genannt und haben als Bezeichnung einen Buchstaben. So nennt sich die Systempartition etwa »C«.

Unter Linux gibt es solche eigenen Laufwerke nicht. Stattdessen werden alle Partitionen unter dem sogenannten *Wurzelverzeichnis* – auch einfach »/*«* genannt – eingebunden. So gesehen gibt es unter Linux nur ein wichtiges Verzeichnis – eben das Wurzelverzeichnis. Abbildung 1.1 zeigt Ihnen die obersten Schichten des Wurzelverzeichnisses.

Eine genauere Beschreibung der Verzeichnisse finden Sie in folgender Liste:

- */* – Das Wurzelverzeichnis, fälschlicherweise auch gerne »root« genannt. Alle möglichen weiteren Partitionen werden in diesem Verzeichnis eingebunden. Auf die meisten Unterverzeichnisse hat der normale Benutzer zwar lesenden Zugriff, kann aber nichts anlegen oder verändern.
- *bin* – Hier liegen die ausführbaren Dateien für Anwendungen, die darin liegenden Dateien kann jeder benutzen.
- *boot* – Das Verzeichnis für den Bootloader und seiner Konfiguration.
- *dev* – Dieses Verzeichnis beinhaltet sogenannte *Gerätedateien*, über die der Kernel mit der Hardware kommuniziert.

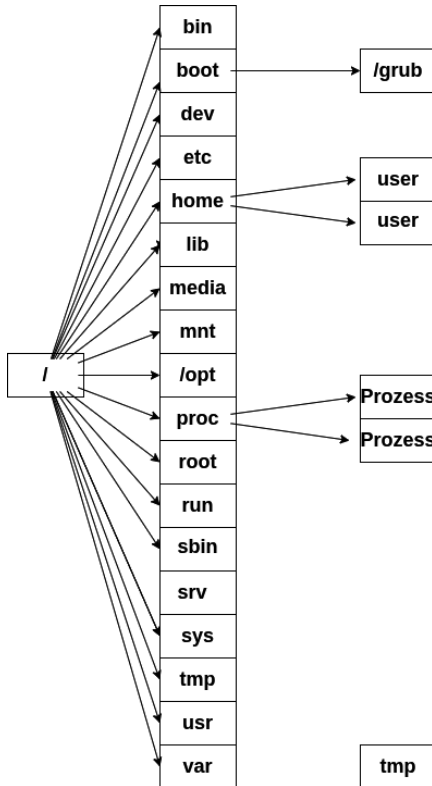


Abb. 1.1: Die Linux-Verzeichnis-Hierarchie

- **etc** – Eines der wichtigsten Verzeichnisse. Hier finden Sie in Textdateien und Unterverzeichnissen die Konfiguration des Systems sowie Dienste (Services) und die Grundkonfiguration bestimmter Anwendungen der Benutzer.
- **home** – Das Benutzer-Verzeichnis. Für jeden Benutzer gibt es hier ein Unterverzeichnis, das nach dem jeweiligen Benutzer benannt ist. Nur hier hat der normale Benutzer ohne administrative Rechte uneingeschränkten Zugriff.
- **lib**, **lib32** und **lib64** – Hier sind Bibliotheken des Betriebssystems zu finden, der Unterordner »firmware« beinhaltet für Treiber wichtige Dateien.
- **media** – In diesem Verzeichnis werden Wechselmedien eingehängt, die vom Betriebssystem automatisch eingebunden werden. Etwa USB-Sticks und externe Festplatten sowie CDs oder DVDs.
- **mnt** – Ähnlich wie »media«, hier werden jedoch keine Medien eingebunden, die automatisch vom System eingebunden werden. In diesem Verzeichnis bindet der Administrator etwa Verzeichnisse aus dem Netzwerk ein.
- **opt** – Das Verzeichnis für optionale Software, etwa für solche Software, die nicht aus den Quellen der genutzten Distribution stammt.

- **proc** – Dieses Verzeichnis liegt nicht auf der Festplatte, sondern im Hauptspeicher (Arbeitsspeicher), es handelt sich also um ein sogenanntes *virtuelles Dateisystem*. Es beinhaltet in Textdateien und Unterverzeichnissen Live-Informationen zum laufenden System und zu Prozessen.
- **root** – Das Verzeichnis des Administrators namens »root«. Hier liegt die Konfiguration der Anwendungen, die dieser benutzt. Nur root selbst hat hier Zugriff.
- **run** – Hier finden Sie in Unterverzeichnissen Dateien zu laufenden Prozessen, also solche Dateien, die von diesen Prozessen benötigt werden.
- **sbin** – In diesem Verzeichnis liegen die ausführbaren Dateien, die nur mit administrativen Rechten ausgeführt werden können, also vom Administrator root.
- **srv** – Hier legen Programmierer oft ihre Entwicklungen und deren Konfiguration ab (etwa Skripte, nicht fertige Anwendungen, Dateien für Docker und Podman).
- **sys** – Ähnlich wie unter »proc« liegen hier Informationen zum laufenden System in Textdateien und Unterverzeichnissen. Auch hierbei handelt es sich um ein virtuelles Dateisystem, da die darin liegenden Daten sich nicht auf der Festplatte, sondern im Hauptspeicher befinden.
- **tmp** – Das Verzeichnis der temporären Dateien. Alle Daten, die vom System und von Benutzern benötigt werden, werden in diesem Verzeichnis zwischengespeichert. Auf modernen Distributionen liegt dieses Verzeichnis nicht auf der Festplatte, sondern als virtuelles Dateisystem im Hauptspeicher.
- **usr** – In diesem Verzeichnis finden Sie die installierten Anwendungen und deren Dateien. Im Unterverzeichnis »bin« finden sich auch einige der ausführbaren Dateien dieser Anwendungen.
- **var** – Im Unterverzeichnis »log« finden Sie alle Log-Meldungen des Systems und installierter Anwendungen. Ist etwa ein Webserver installiert, ist hier auch »var/www/html« zu finden – dort liegen dann beispielsweise die Webseiten.

1.3 Die Shell – arbeiten auf dem Terminal

Wollen Sie sich tiefergehend mit Linux befassen, führt kein Weg an der Shell – also an der Kommandozeile – vorbei. Das Arbeiten auf der Shell zu erlernen, muss kein steiniger Weg sein. Mit einigen wichtigen Befehlen und dem Wissen, wie Sie weitere Hilfe am Terminal finden, werden Sie schnell zum Profi.

Die Shell ist ein sogenannter *Shell-Interpreter*, wir werden in diesem Buch jedoch bei der Bezeichnung »Shell« bleiben. Terminal, Konsole und ähnliche Namen sind nur die Bezeichnungen für die grafischen Fenster der jeweiligen Desktop-Umgebung. Die Shell nimmt Befehle von Ihnen entgegen und übersetzt diese in

eine für den Kernel verständliche Sprache und gibt (oft) für den Menschen verständliche Ausgaben zurück.

Unter Linux finden Sie viele Shells: die Bash, Zsh, Fish, Dash – man kann sie nicht alle aufzählen. Grundsätzlich ist die Bash (Bourne Again Shell) auf fast jeder Distribution vorinstalliert, sie wird auch am meisten genutzt. Aus diesem Grund wird in diesem Buch auch die Bash verwendet. Alle Shells haben ihre kleinen Eigenheiten. Alles, was Sie in diesem Kapitel und in diesem Buch lernen, können Sie jedoch mit jeder Shell ohne Anpassungen umsetzen.

Starten Sie die Shell, indem Sie das Terminal (die Konsole) der jeweiligen Desktop-Umgebung öffnen. Anschließend sehen Sie eine Ausgabe ähnlich dieser – den Eingabe-Prompt:

```
robertg@debian-hp:~$
```

Zu Beginn sehen Sie den Benutzernamen, mit dem Sie am System angemeldet sind, nach dem »@« finden Sie den Namen des Computers, nach dem Doppelpunkt wird Ihnen angezeigt, in welchem Verzeichnis Sie sich befinden – die Tilde »~« zeigt Ihnen, dass Sie sich in Ihrem Home-Verzeichnis aufhalten. Das »\$«-Zeichen vermittelt Ihnen, dass Sie gerade als normaler Benutzer arbeiten und das »#«-Zeichen, dass Sie root-Rechte nutzen.

Sie können auch auf die Shell wechseln, indem Sie die grafische Oberfläche verlassen. Hierzu nutzen Sie die Tastenkombination `[Strg]+[Alt]+[F1]` bis `[F7]` – es ist unter den verschiedenen Distributionen unterschiedlich. Sie haben sechs Shells (sogenannte *virtuelle Terminals*) zur Verfügung. Mit einer dieser Tastenkombinationen gelangen Sie wieder auf die grafische Oberfläche zurück. Anders als bei der Shell im Fenster müssen Sie sich hier erst anmelden – dies sehen Sie an der Ausgabe:

```
login:
```

Geben Sie Ihren Benutzernamen ein, bestätigen Sie mit `[↵]`, anschließend werden Sie nach dem Passwort gefragt – auch dieses bestätigen Sie mit `[↵]`. Mit der Tastenkombination `[Strg]+[d]` können Sie sich wieder abmelden.

Wichtig beim Arbeiten auf der Shell!

Die Shell nimmt Ihre Befehle entgegen und führt diese aus. Bei den meisten Befehlen fragt die Shell nicht nach, ob Sie den Befehl wirklich ausführen wollen! Löschen Sie eine Datei oder ein Verzeichnis auf der Shell, landet diese nicht im Papierkorb, von wo Sie diese wiederherstellen können – es wird umgehend gelöscht. Seien Sie also vorsichtig.

Die Shell unterscheidet zwischen Groß- und Kleinschreibung. Dies gilt bei Befehlen, Optionen, Dateien und Verzeichnissen. So wären Verzeichnisse mit den Namen »Dokumente« und »dokumente« zwei unterschiedliche Verzeichnisse.

1.3.1 Befehle als root (als Administrator) ausführen

Als normaler Benutzer haben Sie unter Linux nur in Ihrem eigenen Home-Verzeichnis alle Rechte – hier können Sie Dateien und Verzeichnisse erstellen, verändern und wieder löschen. Außerhalb dieses Verzeichnisses kann man als normaler Benutzer zwar viele Verzeichnisse und Dateien einsehen, aber nichts an diesen verändern oder solche löschen. Um an Systemdateien oder Systemverzeichnissen Änderungen vornehmen zu können, müssen Sie administrative Rechte erwerben – zum Administrator namens »root« werden.

Unter Linux gibt es zwei Arten, administrative Rechte zu erwerben. Hierzu können folgende zwei Befehle genutzt werden:

```
su
sudo
```

Auf aktuellen auf Ubuntu basierenden Distributionen kommt noch folgender Befehl hinzu:

```
sudo-rs
```

Welche Befehle Sie ohne weitere Umstände nutzen können, hängt von der genutzten Distribution ab und davon, welche Optionen Sie bei der Installation gewählt haben. Unter einem traditionellem Debian (nicht als Live-System) wird bei der Installation das Anlegen eines root-Passworts verlangt, bei den meisten anderen Distributionen ist dies optional – nach der Installation können Sie ein solches immer anlegen.

In Tabelle 1.1 finden Sie die grundsätzlichen Unterschiede zwischen `su` und `sudo`.

Befehl	su	sudo
Zweck	Wechselt zwischen Benutzern	Führt einen einzelnen Befehl als root aus
Passwort	Ein root-Passwort für alle Benutzer	Jeder Benutzer nutzt sein eigenes Passwort
Sitzung	root-Rechte sind aktiv, bis man sich abmeldet	root-Rechte sind nur einige Minuten aktiv

Tabelle 1.1: Die Unterschiede zwischen `su` und `sudo`


Befehl	su	sudo
Kontrolle	Jeder Benutzer, der das Passwort kennt, kann administrative Befehle ausführen	Sie können bestimmen, welcher Benutzer welche Befehle nutzen darf
Sicherheit	Weniger sicher, wenn mehrere Benutzer das Passwort kennen	Sicherer, da Sie festlegen können, welcher Benutzer welche Befehle ausführen darf

Tabelle 1.1: Die Unterschiede zwischen su und sudo (Forts.)

Grundsätzlich ist root ein eigener Benutzer mit eigenem Passwort. Jeder Benutzer, der dessen Passwort kennt, kann zu root werden und somit alle administrativen Befehle nutzen, die verfügbar sind. Um zu root zu werden, nutzen Sie den Befehl su. Wurde die Linux-Distribution ohne root-Passwort installiert, kann der erste erstellte Benutzer mit seinem Passwort administrative Befehle ausführen und weiteren Benutzern zu administrativen Befehlen verhelfen (dazu mehr in Abschnitt 7.5).

Administrative Rechte mit su

Mit dem Befehl su werden Sie tatsächlich zu einem anderen Benutzer, und zwar zum Benutzer »root« (Substitute User). Sie haben damit alle Rechte im Linux-Betriebssystem und können das Betriebssystem so gesehen auch komplett unbrauchbar machen – und jeder Benutzer, der das Passwort von root kennt, kann dies ebenso.

Um zum Benutzer root zu werden, geben Sie im Terminal den Befehl su ein und bestätigen mit , anschließend werden Sie nach dem Passwort von root gefragt:

```
su
Passwort:
```

Der Eingabe-Prompt ändert sich von:

```
benutzername@computername:~$
```

zu:

```
root@computername:/home/benutzername/#
```

Das Doppelkreuz »#« zeigt Ihnen, dass Sie jetzt administrative Rechte haben – auch root-Rechte genannt – und uneingeschränkt arbeiten können. Hierfür gilt

keine Zeitvorgabe. Erst wenn Sie sich mit der Tastenkombination `[Strg]+[d]` abmelden, arbeiten Sie wieder als normaler Benutzer auf der Shell.

Administrative Rechte mit sudo

Mit dem Befehl `sudo` führen Sie einfach gesagt nur kurzzeitig Befehle mit administrativen Rechten aus. Mittels `sudo` können Sie auch anderen Benutzern das selbe Recht geben, administrative Befehle auszuführen – aber Sie können diese Rechte auch einschränken und anderen Benutzern etwa nur das Recht geben, gewisse administrative Befehle auszuführen (dazu mehr in Abschnitt 7.5).

Der Befehl `sudo` lässt sich auf mehrere Arten nutzen, dies dient auch ein wenig der Sicherheit. Erstens werden die administrativen Rechte nach einiger Zeit wieder entzogen (nach ungefähr fünf Minuten) und zweitens lassen sich die Rechte etwa auf ein Verzeichnis beschränken.

Grundsätzlich führt man mit dem Befehl `sudo` einen einzelnen Befehl aus. Mit folgendem Befehl würden Sie etwa den Shell-Editor Nano mit administrativen Rechten öffnen:

```
sudo nano
```

Der auszuführende Befehl wird also nach dem Befehl `sudo` geschrieben. Nach der Bestätigung mit `[↵]` werden Sie nach Ihrem Passwort gefragt. Für fünf weitere Minuten können Sie Befehle mit voranstehendem `sudo` ausführen, ohne nach dem Passwort gefragt zu werden.

Fügen Sie dem Befehl `sudo` die Option `-i` hinzu, können Sie dauerhaft administrative Befehle ohne zeitliche Einschränkungen ausführen – in diesem Fall geben Sie jedoch keinen anschließenden Befehl an:

```
sudo -i
```

Jetzt können Sie administrative Befehle ohne vorangehendes `sudo` ausführen, bis Sie sich mit `[Strg]+[d]` abmelden.

1.3.2 Grundlegende Shell-Befehle

Das Wichtigste voran: Die Shell unter Linux unterscheidet zwischen Groß- und Kleinbuchstaben! Dies ist sowohl bei Befehlen der Fall als auch bei Optionen, Dateien und Verzeichnissen und gilt genauso bei Benutzernamen und Passwörtern.

Ebenfalls wichtig: Denken Sie gut darüber nach, was Sie tun möchten, und handeln Sie danach! Auch wenn sich in den letzten Jahren viel verbessert hat, verzeiht die Shell keine Fehler: Eine gelöschte Datei wird tatsächlich gelöscht.

Befehle vervollständigen lassen

Die Shell hilft Ihnen bei der Arbeit. Sie müssen nicht alle Befehle vollständig ausschreiben, dies gilt auch bei Verzeichnissen und Dateien.

Geben Sie die ersten Zeichen eines Befehls ein, eines Verzeichnisses oder einer Datei, und drücken Sie anschließend die Taste `Tabulator`, vervollständigt die Shell den Befehl, das Verzeichnis oder die Datei automatisch. Sind mehrere Möglichkeiten vorhanden, die mit denselben Zeichen beginnen, tut die Shell dies nicht – in diesem Fall drücken Sie die `Tabulator`-Taste nochmals und die Shell zeigt Ihnen die vorhandenen Möglichkeiten:

```
ls Do  
Docker/ Dokumente/ Downloads/
```

Vervollständigen Sie die Eingabe so lange, bis keine Gemeinsamkeiten mehr vorhanden sind, anschließend vervollständigt die Shell die Eingabe, dann bestätigen Sie den Befehl mit `↵`.

Navigation im Dateisystem

Wenn Sie die Shell in einem Fenster öffnen oder sich in einem virtuellen Terminal (siehe Abschnitt 1.3) anmelden, landen Sie in Ihrem Home-Verzeichnis – dies zeigt Ihnen die Tilde `»~«` am Eingabe-Prompt:

```
benutzername@computername:~$
```

Um im Dateisystem zu navigieren, nutzen Sie den Befehl `cd` und geben danach das gewünschte Unterverzeichnis an. Um beispielsweise in das Unterverzeichnis `»Dokumente«` zu wechseln:

```
cd Dokumente/
```

Der Slash (`»/«`) zeigt der Shell, dass es sich dabei um ein Verzeichnis handelt. Sie müssen ihn nicht unbedingt angeben, solange Sie keine weiteren Unterverzeichnisse angeben. Sie können natürlich auch einige Unterverzeichnisse überspringen. Ein Beispiel:

```
cd Dokumente/Firma/PDF-Dateien/
```

Mehrere Unterverzeichnisse müssen Sie natürlich durch einen Slash trennen.

Die Shell zeigt Ihnen an, in welchem Verzeichnis Sie sich gerade befinden – wie schon beschrieben, bedeutet die Tilde, dass Sie sich in Ihrem Home-Verzeichnis

befinden. Haben Sie das Verzeichnis gewechselt, wird Ihnen dies nach der Tilde angezeigt:

```
benutzername@computername:~/Dokumente/Bilder/
```

Möchten Sie in das Wurzelverzeichnis wechseln, geben Sie einfach nur den Slash an:

```
cd /
```

Befinden Sie sich in einem Unterverzeichnis und möchten ein Verzeichnis in der Hierarchie hochgehen, geben Sie nach dem Slash zwei Punkte an:

```
cd ..
```

Möchten Sie direkt in Ihr Home-Verzeichnis wechseln, egal wo Sie sich in der Verzeichnis-Hierarchie gerade befinden, nutzen Sie den Befehl alleine ohne weitere Optionen:

```
cd
```

Als Benutzer root landen Sie so natürlich im Verzeichnis »/root«, also im Verzeichnis des Administrators.

Verzeichnis-Inhalt anzeigen

Um den Inhalt des Verzeichnisses anzuzeigen, in dem Sie sich auf der Shell befinden, nutzen Sie folgenden Befehl:

```
ls
```

Als Ausgabe erhalten Sie Dateien und Verzeichnisse angezeigt, Verzeichnisse natürlich mit einem abschließenden Slash. Ein Beispiel zeigt sich in Abbildung 1.2.

```
robertg@debian-hp:~$ ls
1      derstandard.at_ips.txt  Dokumente  go          Öffentlich
Software ticket.jpg  Vorlagen
Bentopdf Dify      Downloads  KAG-Klimaticket_Vereinbarung_2025.pdf  Schreibtisch
stargate Vertrag    Zotero
Bilder  Docker    dwhelper   Musik
Test    Videos
robertg@debian-hp:~$ █
```

Abb. 1.2: Der Inhalt eines Verzeichnisses auf der Shell

Sie können sich natürlich auch den Inhalt eines Verzeichnisses anzeigen lassen, in dem Sie sich nicht befinden – hierzu geben Sie den Pfad zu diesem Verzeichnis an. Ein Beispiel:

```
ls Dokumente/Firma/PDF/
```

Dasselbe gelingt natürlich auch mit Verzeichnissen im Wurzelverzeichnis – Sie geben hierzu den Slash für das Wurzelverzeichnis zu Beginn an und anschließend das gewünschte Unterverzeichnis:

```
ls /etc/
```

Optionen für Befehle

Nachdem Sie die wichtigsten Befehle zur Navigation im Dateisystem kennen, haben Sie auch schon den Begriff »Optionen« gelesen. Optionen dienen dazu, Befehle zu verfeinern, also genauer angepasste Ausgaben zu ermöglichen.

Zuvor haben Sie beispielsweise den Befehl `ls` kennengelernt, der dazu dient, sich den Inhalt eines Verzeichnisses anzeigen zu lassen. Im Home-Verzeichnis eines Benutzers gibt es versteckte Dateien und Verzeichnisse. Diese beginnen mit einem Punkt, werden mit dem Befehl `ls` jedoch nicht angezeigt, um eine bessere Übersicht zu haben. Um sich solche versteckten Dateien anzeigen zu lassen, nutzen Sie zusätzlich die Option `-a`:

```
ls -a
```

Weitere Informationen zu Dateien und Verzeichnissen sehen Sie, indem Sie die Option `-l` (kleines »L«) nutzen:

```
ls -l
```

```
robertg@debian-hp:~$ ls -l
insgesamt 1848
drwxr-xr-x  2 robertg robertg  4096 15. Dez 2024  1
drwxrwxr-x  2 robertg robertg  4096 28. Nov 19:19 Bentopdf
drwxr-xr-x 11 robertg robertg  4096 27. Aug 02:52 Bilder
-rw-rw-r--  1 robertg robertg   256  9. Nov 08:27 derstandard.at_ips.txt
drwxr-xr-x  3 robertg robertg  4096 15. Jun 09:54 Dify
drwxr-xr-x  5 robertg robertg  4096  1. Nov 19:13 Docker
```

Abb. 1.3: Nähere Informationen zu Dateien und Verzeichnissen auf der Shell

Welche Bedeutungen diese Informationen haben, lesen Sie in Abschnitt 1.5.

Noch weiter verfeinern lassen sich Befehle, indem Sie Optionen miteinander kombinieren. Nehmen wir an, Sie möchten sich auch die versteckten Inhalte ansehen – also mit der Option `-a` – und die näheren Informationen – Option `-l`. Geben Sie hierfür die beiden Optionen hintereinander an. Dies funktioniert in folgenden beiden Varianten, wobei die erste natürlich bequemer ist:

```
ls -la
ls -l -a
```

Die meisten Befehle auf der Shell verfügen über solche Optionen, manche davon sogar über sehr viele. Der Befehl `ls` kennt über 50 solcher Optionen. Zu beinahe jedem Befehl auf der Shell gibt es eine sogenannte »Manpage« – einfach gesagt, eine Hilfeseite. Um eine solche Manpage zu öffnen, nutzen Sie den Befehl `man`, gefolgt vom Befehl, zu dem Sie Hilfe benötigen – ein Beispiel:

```
man ls
```

Nähere Informationen zur Manpage finden Sie später in diesem Abschnitt.

Dateien löschen und erstellen

Zum Löschen von Dateien nutzen Sie den Befehl `rm`, gefolgt von der zu löschenden Datei – um etwa die Datei »dateiname.txt« zu löschen:

```
rm dateiname.txt
```

Wichtig hierbei – die Datei wird tatsächlich ohne Umweg über den Papierkorb gelöscht, eine Wiederherstellung ist also sehr umständlich!

Sie können auch mehrere Dateien auf einen Schlag löschen, geben Sie diese durch ein Leerzeichen getrennt voneinander an:

```
rm dateiname.txt dateiname.jpg dateiname.pdf
```

Mit dem Sternchen »*« als Platzhalter löschen Sie alle Dateien, die sich im Verzeichnis befinden:

```
rm *
```

Indem Sie das Sternchen als Platzhalter nutzen und die Dateiendung folgen lassen, können Sie etwa alle ».pdf«-Dateien löschen und alle anderen Dateiformate unbehelligt lassen. Auch hier gilt natürlich die Groß- und Kleinschreibung:

```
rm *.pdf  
rm *.PDF
```

Um Dateien zu erstellen, nutzen Sie hingegen den Befehl `touch`, gefolgt von der zu erstellenden Datei. Um beispielsweise die Datei »dateiname.txt« zu erstellen:

```
touch dateiname.txt
```

Verzeichnisse löschen und erstellen

Um leere Verzeichnisse zu löschen, nutzen Sie den Befehl `rmdir`, gefolgt vom Verzeichnis und dem abschließenden Slash – möchten Sie zum Beispiel das Verzeichnis »Dokumente« löschen:

```
rmdir Dokumente/
```

Wie schon angedeutet, funktioniert dies nur bei leeren Verzeichnissen. Möchten Sie ein Verzeichnis löschen, in dem sich noch Daten vorhanden sind, nutzen Sie `rm` mit den Optionen `-rf`:

```
rm -rf Dokumente/
```

Möchten Sie Verzeichnisse erstellen, dient dazu der Befehl `mkdir`, weiterhin geben Sie den Namen des zu erstellenden Verzeichnisses an. Um das Verzeichnis »Dokumente« zu erstellen:

```
mkdir Dokumente
```

Beim Erstellen eines Verzeichnisses geben Sie keinen abschließenden Slash an.

Wollen Sie ein Unterverzeichnis erstellen in einem Verzeichnis, ohne in dieses zu wechseln, nutzen Sie zusätzlich die Option `-p` – um etwa das Verzeichnis »PDF« unter »Dokumente/Firma« zu erstellen:

```
mkdir -p Dokumente/Firma/PDF
```

Dateien und Verzeichnisse kopieren

Dateien und Verzeichnisse lassen sich auf der Shell mit dem Befehl `cp` kopieren. Sie geben die zu kopierende Datei an und den Pfad, also das Verzeichnis, in das die Datei kopiert werden soll:

```
cp dateiname.pdf Dokumente/Firma/PDF/
```

Dies gelingt auch für alle Dateien im Verzeichnis mit dem Platzhalter »*«:

```
cp * Dokumente/Firma/PDF/
```

Möchten Sie ein ganzes Verzeichnis kopieren, nutzen Sie zusätzlich die Option `-r`:

```
cp -r Neue-PDF-Dateien/ Dokumente/Firma/PDF/
```

Mit `cp` können Sie auch ganz einfach Backup-Dateien im selben Verzeichnis erstellen. Backup-Dateien haben unter Linux meist die Dateieindung »bak«. Im folgenden Beispiel wird von der Datei »dateiname.txt« ein Backup erstellt:

```
cp dateiname.txt dateiname.txt.bak
```

Auf dieselbe Weise können Sie das Backup wieder zurückspielen:

```
cp dateiname.txt.bak dateiname.txt
```

Dateien und Verzeichnisse verschieben

Zum Verschieben von Dateien und Verzeichnissen nutzen Sie den Befehl `mv`, auch hier geben Sie nachstehend das Verzeichnis an, in das verschoben werden soll:

```
mv PDF/ Dokumente/Firma/PDF/
```

Beim Verschieben von Verzeichnissen müssen Sie im Gegensatz zum Befehl `cp` keine weitere Option angeben.

Dasselbe mit einer Datei:

```
mv dateiname.pdf Dokumente/Firma/PDF/
```

`mv` kann auch zum Umbenennen von Dateien und Verzeichnissen genutzt werden, hierzu geben Sie zu Beginn die aktuelle Datei oder das aktuelle Verzeichnis an. Es folgt der neue Dateiname oder der neue Name des Verzeichnisses:

```
mv dateiname.pdf neuer_dateiname.pdf  
mv Verzeichnis/ neues_Verzeichnis
```

Stichwortverzeichnis

.bin 332
.run 332
.sh 332

A

Abhängigkeiten 312, 330
 deinstallieren 316
Active Directory 343
 ID 345
 Konfiguration 344
AD 343
Administrator 19
Administrator-Rechte 196
Adressbereich 219
 berechnen 227
Alias 207
Anacron 153, 154, 157
 Aufgaben erstellen 159
 Konfigurationsdatei 158
Apache 360
AppImage 325
 ausführbar machen 325
APT 313
Arbeitsspeicher 13, 57
Archiv 308
Argument
 auswerten 279
ash 103
at 153
 Aufgaben auflisten 153
Audio-Hardware 63
auskommentieren 187

B

Backup 27
 Skript starten 66
Bash 18, 204, 271, 365
 Grundkonfiguration 204
 Prompt anpassen 207

Bash-Skript 271, 365
 ausführbar machen 273
 Ausgabe speichern 275, 276
 Bedingung 281
 Dateiendung 273
 Datum 277
 Exit-Code 285
 Fallentscheidungen 281
 Fehler finden 298
 grafische Oberfläche 287
 Hilfe 286
 starten 273
 Syntax prüfen 298
 testen 298
 Uhrzeit 277
 Variable 277
 Zeilenumbruch 284
Befehl
 Alias 206
 anpassen 34
 ausführliche Hilfe 31
 Fehler 35
 Hilfe 30
 Navigation 35
 neu starten 34
 Optionen 24
 Pfad 41
 Pfad zur Konfigurationsdatei hin-
 zufügen 42
 suchen 30, 34
 vervollständigen 22
Befehle werden nicht gefunden 41
Benutzer 43
 anlegen 46
 anzeigen 44
 bearbeiten 48
 löschen 47
Benutzer-Verzeichnis 16
Benutzereingabe 279

Benutzername 46
 Berechtigungen 50
 vergeben 52
 Besitzer 53
 Betriebssystem 365
 Bibliotheken 300
 bin 15
 BIOS 99
 boot 15
 Boot-Optionen 102
 Boot-Order 100
 Bootloader 15, 101, 187
 Bootloader GRUB
 Konfiguration 187
 Bootmenü 101
 Bourne Again Shell 365
 Broadcast 217, 218
 Broadcast-Adresse 216
 Btrfs 365
 btrfs 74
 BusyBox 103
 Bzip2 40

C

C/C++ 303
 cargo 328
 cat 28, 44
 chgrp 52
 chmod 52
 Cinnamon 268
 CLI 365
 Command Line Interface 365
 Container 351, 355
 aktualisieren 359
 Dateien speichern 356
 Image löschen 355
 Image selbst erstellen 359
 Konfiguration 357
 Port 357
 Container-Virtualisierung 351, 353
 CPU 99
 Cron 153, 154
 Skript 156
 Crontab 155
 crypt-luks 350
 CUPS 338

D

Datei
 erstellen 25
 Hardware 59
 kopieren 26
 löschen 25
 ohne Dateieindung 36
 suchen 29
 umbenennen 27
 verschieben 27
 versteckte 24
 versteckte anzeigen 213
 Dateisystem 15, 22, 69, 73, 365
 ermitteln 78
 ext4 366
 Metadaten 75
 Navigation 22
 reparieren 96
 USB-Stick 74
 virtuelles 17
 Dateisystemfehler 96
 Daten retten 349
 DE 365
 DEB 307
 DEB-Paket 316
 Debian 19
 Debug-Funktion 299
 aktivieren 299
 deaktivieren 299
 deluser 47
 Desktop 365
 Desktop-Umgebung 261, 265, 365
 mehrere 265
 Desktop Environment 365
 dev 15, 59
 DHCP 224
 Dialog 288, 291
 Dialog anzeigen 292
 Textdatei anzeigen 297
 Dienst 16, 121, 125
 Discover 308
 Flatpak 319
 Snap 323
 Distribution 366
 DNF 317
 DNS 221
 freie Server 225
 Root-Server 222

- Docker 353, 354, 365
 - Image 355
- docker-compose 358
- docker.io 354
- Domäne 221
- Drucker 48, 63, 338
 - Konfiguration 340
- E**
- Editor 36
 - Nano 36
 - Vi 37
 - Vim 37
 - wechseln 205
- Einhängepunkt 67
- Entwicklungs Umgebung 301, 303
- etc 16
- eth0 216
- Ethernet 366
- exFAT 73, 74
- Exit-Code 291
 - auswerten 293
- ext3 73
- ext4 73, 366
- F**
- Fast Boot 100
- FAT32 73, 74
- Fdisk 73
- Fehlerausgabe 35
- Fenster-Manager 265, 366
 - mehrere 265
- Festplatte 61
 - anzeigen 62
 - Daten löschen 92
 - Funktionsweise 69
 - Geräte datei 62
 - Geschwindigkeit 114
 - Informationen anzeigen 61
 - Last 113
 - Name 61
 - virtuelle 62
- file 36
- find 29
- Firewall 234
 - Grundlagen 235
 - Log 241
 - Regeln speichern 249
 - Server 238
- Firmware 16, 57, 59, 99, 332
 - Installation 334
- Flatpak 308, 318
 - Abhängigkeiten deinstallieren 321
 - aktivieren 319
 - Berechtigungsprobleme 322
 - Discover 319
 - GNOME 320
 - installieren 319
 - installierte Pakete anzeigen 321
 - Paket aktualisieren 321
 - Paket deinstallieren 321
 - Paket installieren 320
 - Probleme lösen 322
 - Sicherheit 322
 - suchen 320
 - XFCE 320
- Formatieren 73
- fstab 192
- G**
- Gateway 223
- Geräte
 - Block 60
 - Charakter 60
 - Pipe 60
 - Socket 60
- Geräte-Mapper 62
- Geräte datei 15, 59
 - anzeigen 59
 - Berechtigungen 60
 - Festplatte 62
 - kopieren 60
 - RAID 93
 - Terminal 63
- gid 43
- GNOME 267
- GNOME-Software 310
- Gparted 71, 74
- GPG-Schlüssel
 - Installation 203
- GPT 71
- Grafische Oberfläche 261
 - deaktivieren 124
- Grafisches User Interface 366

graphical.target 123
 grep 29
 GRUB 101

- Hintergrundbild 190
- Konfiguration 187
- optische Einstellungen 190
- Schriftfarbe ändern 191

 GRUB-Shell 103
 Gruppe 43

- Administrator 53
- anzeigen 43
- Benutzer hinzufügen 53
- Dateie übergeben 53
- erstellen 52
- Zugehörigkeit anzeigen 44

 Gruppen

- anzeigen 45

 GUPFW 236

- aktivieren 236
- Regeln definieren 237

 GUI 366
 gz 242
 Gzip 39

H

Hardlink 54, 55

- löschen 55

 Hardware 15, 57, 215, 216

- Adresse 216
- anzeigen 57
- eingebaute 57
- Hersteller-Identifikations-ID 58
- Identifikationsnummer 57, 58
- Probleme anzeigen 333
- Regel anpassen 65
- USB 58

 Hardware-Erkennung 64
 Hardware-Log 176
 Hardware-Virtualisierung 351
 Hauptplatine 57
 Hauptspeicher 14, 57
 head 28
 Heimnetzwerk 224
 help 30
 History

- Befehl anpassen 34
- durchsuchen 34

history 34
 home 16
 Home-Verzeichnis 18, 23
 Host 221
 htop 111

I

id 43
 IDE 301
 if-then 281

- Syntax 281

 info 32
 Info-Page 32
 Init 366
 Init-System 121
 Inode 75

- Verfügbarkeit prüfen 76

 Internetverbindung 223
 iotop 113
 IP-Adresse 216, 217, 218

- anzeigen 222
- konvertieren zu 221
- öffentliche 219
- private 219
- statische 224

 Iptables 235, 242

- Logs aktivieren 248
- Regel erstellen 245
- Regelketten 243
- Regeln anzeigen 245
- Regeln löschen 245
- Support 235
- Syntax 244
- Tabellen 243

J

journalctl 163
 Journaling 73

K

KDE Plasma 266
 Kdialog 288

- Befehle kombinieren 295
- Dialog anzeigen 292
- Ja-/Nein-Abfrage 290
- Spalten 295
- Textdatei anzeigen 298

- Kernel 59, 366
 - Konfiguration laden 200
 - Modul 198
 - nicht gefunden 103, 104
 - Optionen anpassen 102
 - Parameter 102
 - Priorität 118
 - Start 106
- Kernel-Modul
 - anzeigen 200
 - ausschließen 199
 - deaktivieren 199
- Kernel-Space 14, 168
- kill
 - Signale 117
- Kommentar 274
- Kompilieren 329
 - C-Programme 329
 - C++-Programme 331
- Komprimierte Archive 39
- Komprimierung 40
- Konfiguration 16, 185
 - sichern 186
- Konfigurationsdatei 186
 - Syntax 187
- Konsole 366
- Kontrollleiste 266
- L**
- Laufwerk 63
- LDAP 343
- Leerzeichen 32
- less 28, 174
- lib 16
- Linux
 - Reparatur 97
 - startet nicht 96
- Linux-Kernel 13
 - Aufgaben 13
- Linux automatisch entschlüsseln 346
- Live-System
 - Reparatur 97
- ln 54
- lo 216
- Log 161, 164
 - aktuellsten anzeigen 163, 175
 - Aufbau 162
 - automatisch löschen 172
 - beim Herunterfahren 165
 - Benutzer-Anmeldungen 178
 - Datum anzeigen 175
 - durchsuchen 164, 174
 - einer Systemanwendung anzeigen 167
 - eines Benutzers anzeigen 166
 - filtern 164
 - Firewall 241
 - komprimiertes 174
 - leeren 176
 - Level 167
 - live anzeigen 168, 174
 - löschen 172, 175
 - seit dem letzten Start 164
 - Speicherplatz anzeigen 172
 - Speicherplatz begrenzen 172
 - Stufe 241
 - Systemd 161, 162
 - SysVinit 161
 - Verzeichnis 171
 - wichtige Dateien 176
 - zu bestimmtem Zeitpunkt 166
 - zu Hardware anzeigen 166
- log 17
- Log-Datei 367
- Log-Meldung 161
 - eingrenzen 164
- Logical Volume 80
 - anzeigen 89
 - erstellen 84
 - Größe anzeigen 90
 - löschen 86
 - vergrößern 81
 - verkleinern 82
 - verschieben 85
- Logical Volume Group 80
 - Attribute 88
- Logical Volume Manager 79, 366
- Logs der Desktop-Umgebung 182
- ls 24
- Lsof 251
- Luks 346
- LVM 62, 79, 80, 366
 - auf Festplatte installieren 83
 - Festplatte hinzufügen 83

- Partitionen 80
 - schematischer Aufbau 79
 - weitere Befehle 86
- M**
- MAC-Adresse 216
- Malware 255
- man 31
- Manpage 25, 31
 - beenden 32
 - durchsuchen 31
- Mate 269
- MBR 71
- media 16
- mkfs 75
- mnt 16
- Motherboard 57, 99
 - Firmware 99
- mount 66
- mounten
 - dauerhaft 78
 - temporär 76
- Mountpoint 77
- multi-user.target 123
- Multicast 217
- mv 27
- N**
- Nano 36, 205
- NAS 366
- Netfilter 235
- Netplan 225
- Netstat 250
- Network Attached Storage 366
- Network File System 366
- Netzwerk 215
 - Grundlagen 215
 - Hardware 215
 - Informationen 215
 - Konfiguration 222
 - lokales 220
 - Route ändern 223
 - Schnittstelle 215
 - überwachen 255
- Netzwerk-Karte 215
 - konfigurieren 218
- neu starten 96
- Nextcloud 357
- NFS 366
- Nftables 235, 242
 - Regel erstellen 245
 - Regelkette 244
 - Regelkette erstellen 247
 - Regeln anzeigen 245
 - Regeln löschen 245
 - Syntax 245
 - Tabelle erstellen 247
 - Tabellen 243
- nice 119
- Nmap 252, 253
- nmcli 227
- NTFS 73, 74
- NVIDIA-Treiber 335
- O**
- Offene Ports anzeigen 249
- OpenLDAP 343
- opt 16
- Option 24
 - auflisten 30
- Oracle 74
- P**
- Paket 307
 - Abhängigkeiten 312
 - Aktualisierung 313, 317
 - automatisch aktualisieren 202
 - Cache 316
 - DEB 316
 - deinstallieren 315, 318
 - installieren 314, 318
 - installierte anzeigen 315
 - manuell heruntergeladenes instal-
lieren 316
 - Quelle 201
 - suchen 314
 - universelles 340
- Paket-Format 307
 - Flatpak 319
 - Snap 323
- Paket-Log
 - unter Debian 179
 - unter Fedora 181

- Paket-Verwaltung 308
 - APT 313
 - Discover 308
 - DNF 317
 - GNOME-Software 310
 - grafische 308
 - Snap 323
 - Synaptic 311
 - verwalten 313
 - Zypper 317
 - Paketverwaltung 201
 - Konfiguration 201
 - Parted 72
 - Partition 60, 62, 69, 366
 - automatisch einhängen 192
 - Backup 194
 - dauerhaft einhängen 78
 - formatieren 74
 - Linux 70
 - Rechte 193
 - reparieren 96
 - temporär einhängen 76
 - vergrößern 79
 - verkleinern 79
 - verschlüsseln 349
 - Partitionieren 70, 73
 - Partitionstabelle 70
 - erstellen 71
 - GPT 71
 - MBR 71
 - PATH 205
 - PATH= 42
 - pcap 258
 - PCI 57
 - Perl 366
 - PHP 302, 367
 - PID 108, 117
 - Pipe 283
 - Platzhalter 27
 - Podman 353, 354, 365, 367
 - Image 355
 - podman-compose 359
 - poedit 304
 - Port 235, 249
 - Datenbank 236
 - freigeben 246
 - Liste aller 240
 - offene anzeigen 250, 253
 - sperren 247
 - SSH 235
 - Webbrowser 235
 - Zustand 251
 - Portscan 254
 - Positionsparameter 284
 - Priorität
 - höchste 119
 - niedrigste 119
 - proc 17
 - Produkt-ID 67
 - Programmieren 271, 300
 - Programmiersprachen 300
 - Installation 300
 - Prompt 18, 207
 - anpassen 207
 - Farbe zurücksetzen 211
 - Hintergrundfarbe 210
 - Schriftfarbe 210
 - Prozess 13, 107
 - anzeigen 107, 108
 - beenden 108, 114, 116
 - ID 117
 - im Hintergrund anzeigen 115
 - im Hintergrund starten 115
 - in den Hintergrund verschieben 115
 - mit Priorität starten 119
 - Priorität 109, 111, 114
 - Priorität ändern 118, 120
 - Reihenfolge beim Systemstart 121
 - Signal 117
 - Status 109
 - steuern 111, 114
 - suchen 110
 - Zombie 116
 - Prozess-Management 13
 - Prozessmonitor 111, 113
 - ps 108
 - Python 300, 326, 367
 - Software installieren 326
 - Python-Repository 326
- Q**
- Qemu 352, 353
 - Quelltext 329

R

RAID 91

- Fake-RAID 92
- Festplattenausfall 94
- Hardware-RAID 91
- Kombi 92
- Mirroring 91
- Parity 91
- Software-RAID 91
- Striping 91

RAID 1 92

RAM 57

RAR 39

reboot.target 124

Rechte

- administrative 19, 20, 46
- Dateien und Verzeichnisse 49
- root 20
- vergeben 48

Regex 367

renice 120

Repository 202, 307

- integrieren 202

Resolver 221

rm 25

root 15, 17, 19, 20, 196

root-Passwort 196

Root-Server 222

Router 218

Routing-Tabelle 218

- prüfen 223

RPM 307, 317

run 17

Runlevel 123, 146

- anzeigen 146
- wechseln 147

Rust 303

Rust-Repository 328

Rust-Software 327

S

Samba 367

sbin 17

Scanner 341

Schnellstart 100

Schnittstelle

- drahtlose 216

kabelgebundene 216

virtuelle 216

scp 231

sda 62

Secure Boot 100

Secure Shell 229, 367

Server 220

Dateisystem 74

Firewall 238

Verbindung herstellen 220

Service 16, 121, 125

aktivieren 128

anzeigen 125, 147

automatischen Start verhindern 128

Backup 129

bearbeiten 129

erstellen 131

erstellen mit SysVinit 149

maskieren 129

neu starten 129

Start-Zeit 169

starten 128, 140

Status 126, 148

steuern 128, 149

stoppen 128

Setgid Bit 52

Shebang 272

Shell 17, 271, 365, 367

Alias 207

Befehle 17

Benutzer anlegen 46

grundlegende Befehle 21

Konfiguration 204

Konfigurationsdatei 42

Leerzeichen 33

ohne grafische Oberfläche 18

Prompt 208

Software installieren 311

Sonderzeichen 32

Shell-Interpreter 17, 368

Shell-Skript 272

ShellCheck 299

Signed-By 203

Skript 149, 271

mit grafischer Oberfläche 287

selbst extrahierendes 332

SMB 367

- Snap 308, 323
 - Discover 323
 - GNOME-Software 323
 - installieren 323
 - installierte Pakete anzeigen 324
 - Pakete deinstallieren 324
 - Pakete installieren 323
 - Probleme lösen 324
 - Rechte 325
 - Software deinstallieren 324
 - Software suchen 324
 - Snapshot 74
 - Sockel 99
 - Socket 140
 - anzeigen 141
 - erstellen 143
 - Startzeit 170
 - steuern 142
 - Softlink 54
 - Software
 - auf der Shell verwalten 313
 - ausführbar machen 325
 - installieren, aktualisieren und deinstallieren 307
 - kompilieren 329
 - Paket 307
 - Quelle 307
 - Rust 327
 - Shell 311
 - Start-Optionen 129
 - Sonderzeichen 32
 - Sound-Karte 63
 - Speichermedium 61, 69
 - Einhängepunkt 67
 - optisches 63
 - srv 17
 - SS 251
 - SSD 61, 62, 69
 - Funktionsweise 69
 - Partitionen 62
 - SSH 229, 367
 - abgebrochene Sitzungen fortführen 232
 - abmelden 230
 - absichern 233
 - aktivieren 229
 - Anmeldung 230
 - Dateien kopieren 231
 - grafische Anwendungen 230
 - Port 235
 - Sicherheitsschlüssel 234
 - verbinden 230
 - Start-Analyse 168
 - Startprobleme 100
 - Startreihenfolge 100
 - Sticky Bit 52
 - su 19, 20
 - Subnetz-Maske 227
 - sudo 19, 21, 49
 - Swap 367
 - Swap-Partition 70
 - Symlink 54
 - Synaptic 311
 - sys 17
 - System
 - Konfiguration 185
 - Systemaufrufe 15
 - Systemd 122, 366, 367
 - Log 161, 162
 - Start-Analyse 168
 - Timer 135
 - Systemd-Logs begrenzen 171
 - Systemmonitor 107, 110
 - Systemstart 99
 - Analyse 168
 - anzeigen 165
 - Dauer anzeigen 168
 - Problembhebung 102
 - System V 122
 - Systemzustand 123, 146
 - SysVinit 122, 145
 - Log 161, 173
- ## T
- tail 29, 174
 - Tar 40
 - Target 123
 - anzeigen 123
 - nicht aktiv 124
 - Start-Zeit 170
 - Status 124
 - Systemstart 125
 - wechseln 124

- Tar und Tar.gz
 - Pakete 308
- tcp 246
- TCP/IP 217
- tcpdump 255
- temporäre Datei 17
- Terminal 17, 367
 - Befehle 17
 - öffnen 18
 - virtuelles 18, 63
 - Zeilenumbruch 284
- test 281
- Text-based User Interface 367
- Textdatei
 - anzeigen 28
 - durchsuchen 28, 29
- Texteditor 36
- Timer 135
 - erstellen 137
 - Status 136
 - steuern 137
- tmp 17
- Tmux 232
- top 110
- TPM 367
- TPM-Chip 346
 - Hash 348
 - prüfen 347
- Treiber 13, 15, 57, 59, 198, 332
 - ausschließen 199
 - Datenbank 333
 - Drucker 338
 - fehlende anzeigen 333
 - Grafikkarte 335
 - Installation 334
 - Log 198
 - Multifunktionsgeräte 338
 - NVIDIA 335
 - Probleme erkennen 333
 - prüfen 333
 - Scanner 338
- Trusted Platform Module 367
- TUI 367
- Turboprint 340
- U**
 - übersetzen 304
- udev 64
 - eigene Regeln schreiben 66
 - Manpage 66
 - Operator 66
 - Regeln 64, 65
- UEFI 99
- UFW 235, 238
 - Logs 241
 - Regeln erstellen 239
- uid 43
- Umgebungsvariable 326
- Unicast 217, 218
- USB-Stick 61
 - bootfähigen erstellen 60
- usermod 48
- usr 17
- UUID 193
- V**
 - var 17
 - Variable 277
 - Syntax 278
 - Vendor-ID 67
 - Verknüpfung 54
 - Verschlüsselung 346
 - Daten retten 349
 - Versteckte Datei 213
 - Verzeichnis 50
 - Inhalt anzeigen 23
 - kopieren 26
 - umbenennen 27
 - verschieben 27
 - verstecktes 24
 - wecheln 22
 - Verzeichnis-Hierarchie 15
 - Vi 37
 - Vim 37, 205
 - Virt-Manager 353
 - VirtualBox 352
 - Virtualisierung 351, 368
 - Container 353
 - Hardware 351, 352
 - Virtuelle Festplatte 62
 - Virtuelle Maschine 352, 368
 - Eigenschaften 353
 - Virtuelles Dateisystem 17
 - visudo 196

Volume 62, 368
Vuescan 341

W

Wayland 230, 262, 264, 368
 Konfiguration 264
Webserver
 Logs 171
 Verzeichnis 17
Weitere Logs 182
Window Manager 366
Windows 351
Wireshark 255
 auf der Shell 258
 Filter 258
 tcpdump 258
wlan0 216
wlo1 216
Wurzelverzeichnis 15, 23
 aushängen 96

X

X11 262, 368
X-Server 261
 Konfiguration 262
X.ORG 368

XFCE 267
XFS 74

Z

Z-Shell 204, 212, 368
Zeitgesteuerte Aufgaben 135, 153
Zenity 288
 Befehle kombinieren 294
 Dialog anzeigen 292
 Fenster benennen 293
 Ja-/Nein-Abfrage 290
 Spalten 294
 Textdatei anzeigen 298
 Texteingabe 296
ZFS 74
ZIP 39, 41
zless 174
Zombie 107, 116
Zsh 18, 204, 212, 368
 Befehlsvervollständigung 213
 Erweiterung aktivieren 214
 Erweiterungen anzeigen 213
 Grundkonfiguration 204
 Theme 213
Zugangsrechte 49
Zypper 317