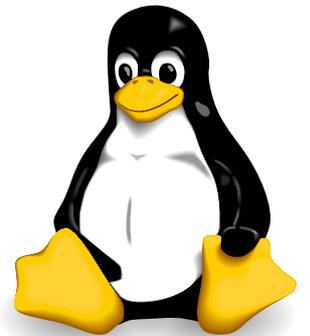


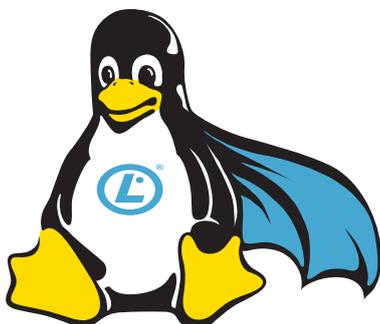


Landeshauptstadt
München
Referat für
Bildung und Sport

Linux Essentials



**Städtische Berufsschule
für Informationstechnik**



Autoren

Michael Niedermair, Joachim Wolf

Städtische Berufsschule für Informationstechnik

<http://www.bsinfo.musin.de/>

Quellen, Material, Bilder, Unterstützung, ...

- Markus Kohn
- Elke Niedermair
- Michael Niedermair
- Axel Stiegelbauer
- Dominik Wagenfuehr
- Joachim Wolf
- LPI (Linux Professional Institute)
- freie Quellen (siehe Quellenangaben)

Satz

Michael Niedermair

Version

8.7.2012 (Revision: 893) – **BETA**

Lizenz

Frei für **nicht kommerzielle Nutzung** in öffentlich/rechtlichen Schulen.
Andere Lizenzvereinbarungen in Absprache möglich!

Inhaltsverzeichnis

1	Einführung	7
2	Einheit I – Erste Schritte mit Linux	11
2.1	Linux	12
2.1.1	Geschichte von Unix	12
2.1.2	Geschichte von Linux	12
2.1.3	Distributionen	12
2.1.4	Einsatz	14
2.1.5	Wichtige Open Source Anwendungen	16
2.1.6	Wichtige Open Source Programmiersprachen	17
2.2	Terminal und Konsole, erste Schritte auf der Kommandozeile	17
2.2.1	Konsole bzw. Terminal starten	17
2.2.2	Was ist eine Shell?	20
2.3	Hilfen zu Kommandos	22
2.4	Kommandos	23
2.4.1	Einfache Kommandos	23
2.4.2	Kommandos mit Argumenten	24
2.4.3	Laufende Kommandos/Prozesse anzeigen	25
2.5	Grundlagen der Dateiverwaltung	27
2.5.1	Namenskonventionen im Dateisystem	27
2.5.2	Jokerzeichen und Reguläre Ausdrücke der Shell	27
2.5.3	Dateiarten	31
2.5.4	Verzeichnisstruktur von Linux-Systemen (FHS)	32
2.5.5	Begriffe zum Dateisystem	33
2.5.6	Arbeiten mit Dateien	36
2.6	Texteditoren	40
2.6.1	Nano	40
2.6.2	Vi	41
2.7	Icon auf dem Desktop erstellen	43
2.8	Fragen	44
3	Einheit II – Arbeiten mit Benutzern und Berechtigungen	47
3.1	Die Benutzerverwaltung unter Linux	48
3.1.1	Systemverwalter, Standardbenutzer und Systembenutzer	48
3.1.2	Primär-Gruppen und Sekundär-Gruppen	48
3.1.3	Benutzerkonten anlegen	48
3.1.4	Als anderer Benutzer arbeiten - su und sudo	51
3.1.5	Gruppenverwaltung	52
3.1.6	Fragen	56
3.2	Dateiberechtigungen und Dateieigentum	58
3.2.1	Benutzerklassen und Zugriffsrechte	58
3.2.2	Zugriffsrechte und Eigentümerschaft ändern	59
3.2.3	Fragen	65
4	Einheit III – Skripte erleichtern das Arbeiten	67
4.1	Einführung in die Skriptprogrammierung	68
4.1.1	Das erste Skript	68
4.1.2	Skript ausführen	68
4.1.3	Variablen	68
4.1.4	Parameter	69
4.2	Gruppen anlegen	74
4.2.1	Datei zeilenweise auslesen	74
4.2.2	Schleifen	74
4.2.3	Bedingungen	77

4.2.4	Mehrfachauswahl	77
4.2.5	Zeichen umwandeln	79
4.2.6	Vorhandene Gruppen ermitteln.....	79
4.3	Benutzer anlegen.....	80
4.3.1	Generieren der Schülerkennung.....	80
4.3.2	Passwörter generieren.....	80
4.3.3	Schülerkennung hinzufügen.....	81
4.3.4	Lehrerkennung hinzufügen.....	82
4.3.5	Superuser einrichten.....	83
4.4	Fragen	85
5	Einheit IV – Bilder vom „Stick“ verarbeiten.....	87
5.1	Externe Medien	88
5.1.1	LOG-Meldungen	88
5.1.2	USB-Geräte anzeigen.....	91
5.1.3	Bilder kopieren	92
5.2	Medien prüfen, einhängen, formatieren,	96
5.2.1	Dateisysteme	96
5.2.2	Dateisysteme ein-/aushängen.....	96
5.2.3	Dateisysteme prüfen.....	97
5.2.4	Dateisystem formatieren	98
5.3	Bilddaten auswerten.....	99
5.4	Filter für Textdateien	101
5.4.1	Textdateien ausgeben.....	101
5.4.2	Textdateien sortieren	102
5.4.3	Zeilen von Textdateien auswählen	103
5.4.4	Textdateien in Spalten und Felder aufteilen	104
5.4.5	Zeilen und Wörter in Textdateien zählen	104
5.4.6	Zeichen austauschen	104
5.5	Bilder kopieren	107
5.5.1	Datum und Uhrzeit ermitteln	107
5.5.2	Datum und Uhrzeit formatieren.....	108
5.5.3	Bilder mit gleichem Datum.....	108
5.5.4	Bilder automatisch drehen	109
5.6	Thumbnails erzeugen	109
5.7	Spielereien mit Bildern	110
5.7.1	Rahmen um Bilder	110
5.7.2	Bilder beschneiden.....	111
5.8	Fragen	113
6	Einheit V – Mit dem Netzwerk arbeiten	115
6.1	Das Internet und die TCP/IP-Protokoll-Familie.....	116
6.2	IPv4	117
6.3	IPv6	119
6.4	IP-Adressen und Rechnernamen mit DNS.....	120
6.5	Manuelle Netzwerkkonfiguration	120
6.6	Automatische Netzwerkkonfiguration mit DHCP.....	123
6.7	Fragen	127
7	Einheit VI – Daten sichern und komprimieren	129
7.1	Datenkomprimierung	130
7.1.1	gzip	130
7.1.2	bzip2.....	131
7.1.3	zip.....	132
7.1.4	tar	134
7.1.5	Andere Komprimierer	135
7.1.6	7-Zip.....	135
7.1.7	Fragen	136
7.2	Datensicherung.....	137
7.2.1	Sicherungsarten	137
7.2.2	Generationenprinzip.....	137
7.2.3	Backup	137

7.2.4	Restore	140
7.2.5	Fragen	140
8	Einheit VII – Webseite für die Bilder erstellen	141
8.1	Bildergalerie erstellen	142
8.1.1	HTML	142
8.1.2	Thumbnails anzeigen	143
8.1.3	Fragen	149
8.2	Galerie über einen Webserver veröffentlichen	150
8.3	Lizenzen.....	151
8.3.1	Übersicht	151
8.3.2	Welche Lizenz für eigene Bilder	154
8.3.3	Fragen	154
8.4	Firefox als sicherer Browser	155
8.4.1	JavaScript	155
8.4.2	Datenschutz.....	155
8.4.3	Werbung unterdrücken	155
8.4.4	Webseitenverfolgung verhindern.....	156
8.4.5	Verschlüsselte Verbindung verwenden	156
8.4.6	Speichern von Inhalten.....	156
8.4.7	Websuche.....	157
9	Einheit VIII – OSS Gimp & LibreOffice	159
9.1	GIMP	160
9.2	LibreOffice.....	161
10	Themenübersicht: Linux-Essentials.....	163
	Topic: 1 Die Linux-Community und Karriere im Open-Source-Umfeld	165
	Topic: 1.1 Linux Entwicklung und gängige Betriebssysteme	165
	Topic: 1.2 Wichtige Open Source Anwendungen	165
	Topic: 1.3 Open-Source-Software und Lizenzen verstehen	165
	Topic: 1.4 IuK-Kenntnisse und Arbeiten mit Linux	166
	Topic: 2 Sich auf einem Linux-System zurechtfinden	166
	Topic: 2.1 Erste Schritte auf der Kommandozeile	166
	Topic: 2.2 Die Kommandozeile benutzen, um Hilfe zu finden	167
	Topic: 2.3 Verzeichnisse und Protokolldateien nutzen	167
	Topic: 2.4 Erstellen, Verschieben und Löschen von Dateien	167
	Topic: 3 Die Macht der Kommandozeile	168
	Topic: 3.1 Archivierung von Dateien auf der Kommandozeile.....	168
	Topic: 3.2 Suche und Entnahme von Daten aus Dateien.....	168
	Topic: 3.3 Befehle in ein Skript umwandeln	169
	Topic: 4 Das Linux Betriebssystem	169
	Topic: 4.1 Wahl eines Betriebssystems	169
	Topic: 4.2 Computer-Hardware verstehen	169
	Topic: 4.3 Datenspeicherung	170
	Topic: 4.4 Den Computer im Netzwerk einbinden	170
	Topic: 5 Sicherheit und Dateiberechtigungen.....	171
	Topic: 5.1 Grundlegende Sicherheit und Identifizieren von Benutzertypen.....	171
	Topic: 5.2 Benutzer und Gruppen erstellen.....	171
	Topic: 5.3 Einstellungen für Dateiberechtigungen und Dateieigentum.....	171
	Topic: 5.4 Besondere Verzeichnisse und Dateien	172
11	Anhang	173
	Programme / Befehle	175
	Verzeichnisse / Dateien	177
	Stichwortverzeichnis.....	179

1 Einführung

Dieses Arbeitsheft dient dazu, dass sich Schülerinnen und Schüler der Städtischen Berufsschule für Informationstechnik, schrittweise anhand eines Szenarios in Linux einarbeiten. Ziel ist es, dass sie für die LPI Linux Essentials Prüfung das Wissen erhalten, um die Zertifizierungsprüfung mit gutem Erfolg zu bestehen

Einige Linux Essentials Inhalte sind Basisinhalte für IT-Berufe und diese werden im normalen Berufsschulunterricht vermittelt. Auf Seite 163 finden sich die Themeninhalte.

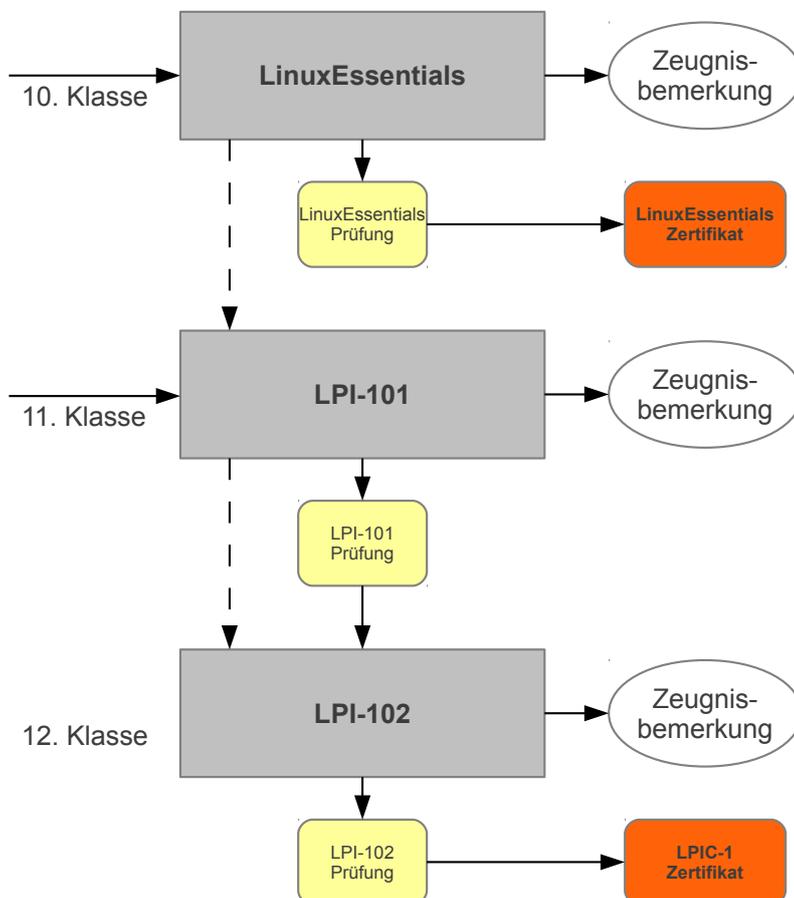
Linux Essentials richtet sich vor allem an Schülerinnen und Schüler, deren Schwerpunkt nicht auf der technischen Seite liegt oder an die, die Linux Grundkenntnisse erlangen wollen.

Es gibt allgemeine Informationen und Übungen, sowie Übungen, die auf das Szenario ausgelegt sind. Dazu ist das Arbeitsheft in mehrere Bereiche aufgeteilt, die Arbeitseinheiten für einen Termin (ca. vier Unterrichtsstunden) darstellen. Zusätzlich ist noch eine Vor- und Nachbereitung zu Hause notwendig, um sicher und ohne Probleme die kursinternen Tests und abschließend die Zertifizierungsprüfung zu bestehen.

Jede Einheit wird mit einem Fragenblock abgeschlossen, so dass das erworbene Wissen hier abgefragt und zusammengefasst werden kann.

Das Niveau entspricht dem eines IT-Auszubildenden und geht in manchen Bereichen über das Niveau der LPI-Prüfung hinaus. Zusätzlich sind ein paar Themen dazu gekommen, um das Szenario zu ergänzen und interessante „Spielereien“ zu zeigen.

Kursprogramm



Die Berufsschule für Informationstechnik bietet insgesamt drei Linux-Wahlfächer an. Linux Essentials richtet sich dabei an Schülerinnen und Schüler der 10ten Klassen. LPI-101 dagegen an Schüler der 11ten Klasse, die dann in der 12ten Klasse LPI-102 besuchen können.

Jeder Kurs ist so ausgelegt, dass im laufenden Jahr mehrere Tests abzulegen sind, deren Ergebnisse dann in einer Zeugnisbemerkung zusammengefasst werden. Anschließend kann dann bei uns im Haus die LPI-Zertifizierungsprüfung abgelegt werden.

Zu Beginn jedes Termins (siehe Tabelle 1.1 auf der nächsten Seite) wird eine kurze Zusammenfassung des Inhalts der jeweiligen Einheit präsentiert. Danach haben die Schülerinnen und Schüler Zeit, die Einheit zu bearbeiten, die Fragen zu beantworten und die Übungen selbst am Rechner durchzuführen. Kurz vor Ende des Termins wird ein kurzer Test (max. 10 Minuten) geschrieben, der Fragen zu dem vorherigen Thema enthält. Die Fragen sind so wie die eigentliche LPI-Prüfung aufgebaut.

Kurstermin 10 ist ein spezieller Crash-Kurs zur Prüfungsvorbereitung. Hier werden zwei Sätze mit Prüfungsfragen (40 Fragen) bearbeitet und dann besprochen. Die Fragen entsprechen im Umfang und Art der LPI-Zertifizierungsprüfung.

Sie sind von uns ausgedacht worden und sind keine echten Prüfungsfragen! Die Fragen werden entsprechend der Gewichtung aus einem von uns erstellten Fragenpool zufällig ausgewählt.

Termin	Datum	Thema / Inhalt	Test
1		Einführung, Verteilung der Unterlagen	-
2		Einheit I – Erste Schritte zu Linux	-
3		Einheit II – Arbeiten mit Benutzern und Berechtigungen	Test zu I
4		Einheit III – Skripte erleichtern das Arbeiten	Test zu II
5		Einheit IV – Bilder vom „Stick“ verarbeiten	Test zu III
6		Einheit V – Mit dem Netzwerk arbeiten	Test zu IV
7		Einheit VI – Daten sichern und komprimieren	Test zu V
8		Einheit VII – Webseite für die Bilder erstellen	Test zu VI
9		Einheit VIII – OSS Gimp & LibreOffice (Optional)	Test zu VII
10		Crash-Kurs zur Prüfungsvorbereitung	-
11		LPI-Prüfung Linux Essentials	-

Abb. 1.1: Zeitplan

Erklärungen

- Kommandos** Die Beispiele haben bzgl. des Ausführens von Kommandos folgenden Aufbau:
Wird ein Kommando direkt angegeben, so kann dies jeder Anwender ausführen. Wird bei dem Kommando `sudo` voran gesetzt, so sind für das Ausführen Administratorrechte (`root`-Rechte) notwendig.
- Skriptbeispiele** Bei einigen Skriptbeispielen finden Sie direkt die erzeugte Konsolenausgabe neben dem eigentlichen Code.
- Tasten** Sind bestimmte Tasten zu betätigen, so werden diese über einen speziellen Tastenkappenfont dargestellt.
 steht für die Return-Taste
 ist eine Kombination von ,  und 
- Maus** Für das Betätigen der Maus (rechte Maustaste, linke Maustaste, ...) werden auch entsprechende Symbole verwendet.
 steht für die rechte Maustaste

Distributionen

Für dieses Skript haben wir die Distribution Ubuntu 12.04 LTS verwendet.

Dies können Sie wie folgt verwenden:

- In der Schule auf den Laborrechnern: nach dem Booten: Virtuelle Bootdisketten – Linux Wahlfach
- In der Schule auf den Pädagogischen Rechnern: über VirtualBox: Das Image hat den Namen „Venus32“
- Zu Hause: über eine Boot-CD-Rom – Live-System
siehe <http://www.ubuntu.com/download/desktop>
- Zu Hause: über einen bootbaren USB-Stick – Live-System
siehe <http://www.ubuntu.com/download/help/create-a-usb-stick-on-windows>
- Zu Hause: selber installieren
siehe <http://www.ubuntu.com/download/help/install-ubuntu-desktop>
- über VirtualBox auf dem eigenen Rechner:
 - In der Schule das VirtualBox-Image „Venus32“ kopieren und zu Hause verwenden.
 - Ein VirtualBox-Image von <http://virtualboximages.com> verwenden
siehe <http://virtualboximages.com/Ubuntu+12.04+i386+Desktop+VirtualBox+VDI+Virtual+Computer>
 - Selbst ein VirtualBox-Image erstellen
siehe <http://www.mannipage.de/virtual%20box.htm>

Material

Für das Wahlfach wird dabei folgendes Material verwendet:

- Kursskript Linux Essential** Dieses Skript! Die aktuelle Version kann als PDF über die Schul-Homepage heruntergeladen werden.
- Einführungspräsentation** Dient dazu, eine Einführung in den Kurs zu präsentieren (in Arbeit).
- Abschnittspräsentation** Für jeden Abschnitt wird zu Beginn eine Zusammenfassung der Inhalte präsentiert (in Arbeit).
- Wissenstest** Nach bestimmten Abschnitten erfolgt ein Wissenstest, um einmal den eigenen Stand einschätzen zu können, andererseits, um eine Grundlage für die Zeugnisbemerkung zu bekommen. Der Test umfasst nur Fragen aus dem angegebenen Bereich und hat einen Umfang von ca. 10 Minuten (in Arbeit).
- Skriptbeispiele** können per „Copy & Paste“ aus der Unterlage heraus kopiert werden.

Szenario



Die Berufsschulklassen (IT-Berufe) IT11FS, IT11FA und IT11SE haben ihre Jahrsabschlussfahrt nach Berlin gemacht. Dabei waren ca. 70 Schüler. Als Lehrer waren Viola Lorenz und Lars Greiner mit dabei.

Die Schüler sollen nun in verschiedenen Gruppen u. a.

- die Bilder jeweils von den entsprechenden Medien in ihre Klassenverzeichnisse kopieren,
- chronologisch sortieren und entsprechend benennen,
- in verschiedene Formate umwandeln,
- Vorschaubilder erstellen,
- Webseiten zur Ansicht erzeugen und
- Daten sichern.





2 Einheit I – Erste Schritte mit Linux

In dieser Einheit lernen Sie Linux, seine Geschichte, diverse Distributionen und die ersten Schritte mit dem Terminal kennen. Sie lernen, wie Befehle/Kommandos aufgerufen werden, Sie den Inhalt von Verzeichnissen anzeigen können und wie man sich sicher im Verzeichnisbaum bewegt. Anschließend legen Sie selbst Verzeichnisse an, kopieren, verschieben und löschen Dateien. Danach legen Sie ein Start-Icon für ein Terminal auf dem Desktop (hier Unity unter Ubuntu) an.

Wichtig ist: Notieren Sie sich bei allen Übungen die Befehle, Kommandos etc., so dass Sie bei der Wiederholung schnell das Wissen vertiefen können. Wenn etwas unklar ist, formulieren Sie entsprechende Fragen, die den nicht verstandenen Sachverhalt betreffen bzw. den Teil, den Sie nicht verstanden haben.

Inhalt

2.1	Linux.....	12
2.1.1	Geschichte von Unix	12
2.1.2	Geschichte von Linux	12
2.1.3	Distributionen.....	12
2.1.4	Einsatz.....	14
2.1.5	Wichtige Open Source Anwendungen	16
2.1.6	Wichtige Open Source Programmiersprachen.....	17
2.2	Terminal und Konsole, erste Schritte auf der Kommandozeile	17
2.2.1	Konsole bzw. Terminal starten.....	17
2.2.2	Was ist eine Shell?.....	20
2.3	Hilfen zu Kommandos	22
2.4	Kommandos	23
2.4.1	Einfache Kommandos.....	23
2.4.2	Kommandos mit Argumenten	24
2.4.3	Laufende Kommandos/Prozesse anzeigen.....	25
2.5	Grundlagen der Dateiverwaltung	27
2.5.1	Namenskonventionen im Dateisystem.....	27
2.5.2	Jokerzeichen und Reguläre Ausdrücke der Shell.....	27
2.5.3	Dateiarten	31
2.5.4	Verzeichnisstruktur von Linux-Systemen (FHS).....	32
2.5.5	Begriffe zum Dateisystem.....	33
2.5.6	Arbeiten mit Dateien.....	36
2.6	Texteditoren	40
2.6.1	Nano.....	40
2.6.2	Vi.....	41
2.7	Icon auf dem Desktop erstellen	43
2.8	Fragen	44

2.1 Linux

Linux ist ein modernes Multi-User-Multi-Tasking-Betriebssystem. Multi-User bedeutet, dass mehrere Nutzer gleichzeitig am System arbeiten können. Multi-Tasking steht dafür, dass mehrere Programme gleichzeitig laufen können. Beide Fähigkeiten hat Linux von seinem Ursprung Unix geerbt.

2.1.1 Geschichte von Unix

Die Entwicklung von Unix begann bereits Ende der 60er Jahre des 20. Jahrhunderts. Die Computer aus dieser Zeit waren große Rechner, die raum- oder sogar gebäudfüllend waren und über Terminals bedient wurden und nicht die Rechenleistung eines heutigen PCs erreichten.¹ Diese Terminals sind einfache Bildschirm/Tastatur-Einheiten, die alles an den Großrechner weiterleiten, was auf der Tastatur eingegeben wird und alles auf dem Bildschirm darstellen, was vom Großrechner kommt. Im Laufe der Zeit hat sich dies zu einer Client- und Serverumgebung weiterentwickelt. Bei einem Client-Server-Modell werden Aufgaben von einem Client (meist Desktop-PC) innerhalb eines Netzwerkes an Server weitergeleitet. Dies können beispielsweise Datei- und Verzeichnisdienste sein oder Datenbankabfragen etc.

Die wichtigsten Ereignisse in der UNIX-Entwicklung waren:

- 1969 Ken Thompson² entwickelte ein Spiel (Space Travel), welches er auf eine DEC-PDP/7 übertrug. Dafür musste ein kleines Multitasking-System entwickelt werden. Zusammen mit Dennis Ritchie wurde ein hierarchisches Dateisystem, eine Speicherverwaltung mit einer Größe von vier KWord des Rechners und eine Prozess- und Time-Sharing-Verwaltung entwickelt, die das System gleichzeitig für zwei Benutzer zur Verfügung stellte. Brian Kernighan bezeichnete dies als UNICS (Uniplexed Information and Computing System). Später wurde daraus der Name Unix, der keine Abkürzung, sondern ein Name ist.
- 1972 Das Betriebssystem wurde komplett neu in der Programmiersprache C implementiert und mit einem C-Compiler kostenfrei verteilt.
- 1977 Die Universität von Kalifornien in Berkeley entwickelt die Berkeley Software Distribution (BSD), ein Unix-Derivat, welches zum Preis eines Datenträger zur Verfügung gestellt wurde, zusammen mit der Erlaubnis, den Quelltext einzusehen und zu modifizieren.
- 1980 Unix (und Unix-Derivate) wird zum dominierenden Betriebssystem an den Universitäten.
- 1983 Start des GNU-Projekts (GNU: Gnu is Not Unix – GNU ist nicht Unix) GNU wird mit dem Ziel entwickelt, eine vollständig freie Alternative für Unix zu erstellen, und steht unter der GNU General Public License (GPL).
- 1991 Linus Torvalds beginnt mit der Entwicklung von Linux, einem UNIX-Derivat.

...

2.1.2 Geschichte von Linux

1991 begann Linus Torvalds in Helsinki mit der Entwicklung einer Terminal-Emulation, die sich allmählich zu einem Betriebssystem entwickelte und stellte es im September zur Verfügung. Der damalige FTP-Server-Administrator Ari Lemmke war mit dem von Torvalds vorgesehenen Namen Freax oder Buggix nicht einverstanden und stellte es daher in einem Verzeichnis mit dem Namen Linux zur Verfügung.

Im Januar 1992 wurde Linux unter die GNU GPL³ gestellt, was dazu führte, dass Linux in GNU integriert werden konnte. Dadurch wurde Linux für viele Entwickler noch interessanter, da es damit eines der ersten freien Betriebssysteme war (das erste für die x86-Plattform und später für unterschiedliche Plattformen) und somit die Modifizierung und Verbreitung vereinfachte.⁴ 1996 kündigte Torvalds an, dass er einen Pinguin als Maskottchen für Linux haben wolle und veranstaltete daher einen Wettbewerb - als Ergebnis wurde Tux präsentiert.

2.1.3 Distributionen

Eine Linux-Distribution ist eine Sammlung von Programm-Paketen, die für den Betrieb eines Linuxsystems notwendig sind. Es gibt eine Vielzahl von Distributionen, die meist alle einen anderen Schwerpunkt haben. Für die LinuxEssential-Zertifizierung ist es egal, welche Distribution verwendet wird, da die Grundlagen bei allen Systemen gleich sind.

¹ Heutige Groß- bzw. Super-Rechner erreichen auch wieder eine raum- oder gebäudfüllende Dimension, sind aber von Ihrer Rechenleistung um ein Zifaches höher als damals.

² Dennis MacAlistair Ritchie (* 9. September 1941 in Bronxville, New York; † vor 12. Oktober 2011 in Berkeley Heights, New Jersey) war ein amerikanischer Informatiker. Zusammen mit Ken Thompson und Brian Kernighan entwickelte er nicht nur UNICS, sondern auch die Programmiersprache C.

³ Die GNU General Public License ist eine von der Free Software Foundation (FSF) veröffentlichte Freie-Software-Lizenz mit Copyleft für die Lizenzierung von freier Software, die ihren Ursprung im GNU-Projekt hat.

⁴ Bereits 1990 gab es die erste Version 0.1 von MiNT, einem Betriebssystem für Atari-ST, dessen Quellen unter GPL standen.

⁵ Quelle: Wikimedia Commons – Autor: Larry Ewing, Simon Budig, Anja Gerwinski (CC-BY-SA-3.0)

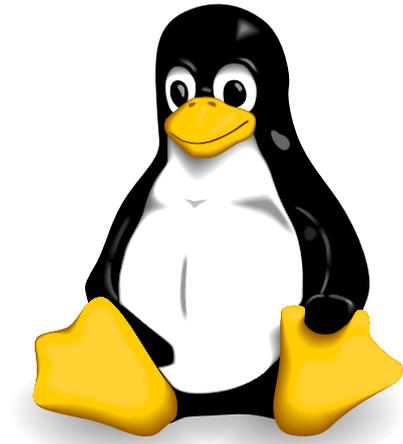


Abb. 2.1: Pinguin Tux, das Linux-Maskottchen⁵

Die gebräuchlichsten Distributionen sind:

- Debian
- Ubuntu
- Knoppix
- Fedora
- OpenSuSE
- Slackware
- Linux Mint
- ...

Abbildung 2.2 zeigt ein Mindmap einiger GNU/Linux-Distribution.

Für die Übungen verwenden wir hier die Distribution Ubuntu (Version 12.04 LTS) mit der grafischen Oberfläche Unity.

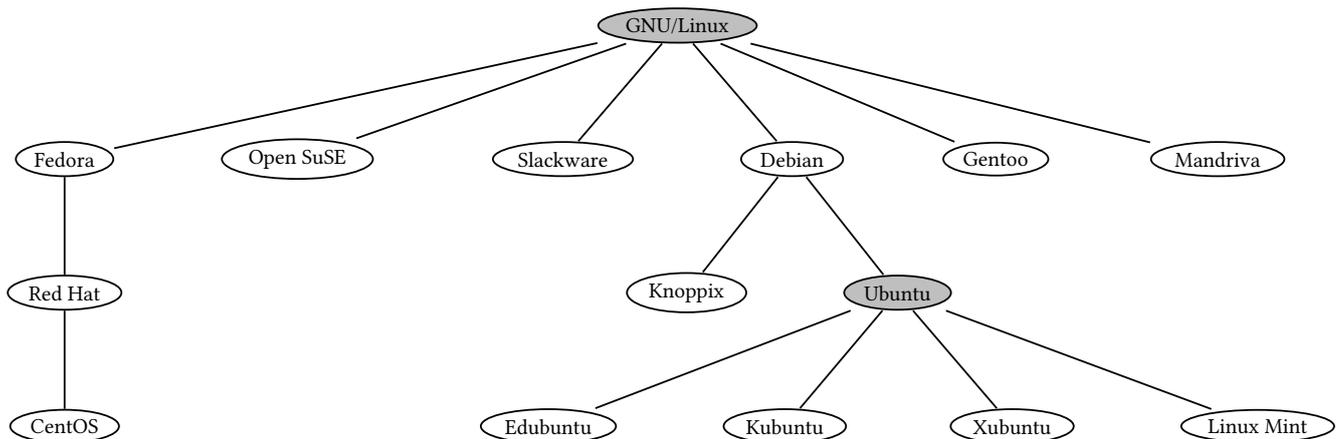


Abb. 2.2: Distributionen (Auswahl)

Distribution life cycle management

Bei Linux Programmen und Distributionen unterscheidet man verschiedene Versionen.

alpha-Version Eine Alpha-Version ist für Entwickler zum Testen gedacht. Ein unerfahrener Anwender sollte davon die Finger lassen.

beta-Version Eine Beta-Version hat noch kleine Fehler, zeigt aber oft neue Funktionen auf. Wenn man nicht gerade mit einem Produktivsystem arbeitet, kann man schon mal testen, welche Funktionen die neue Version hat und auch somit die Entwickler unterstützen.

testing Eine Version, die sich im Test befindet (wie beta-Version), die kurz vor der Veröffentlichung steht.

unstable Eine Version, an der ständig entwickelt wird. Diese wird irgendwann zu einer alpha- oder testing- Version.

stable Eine stable-Version gilt als stabil und im allgemeinen als fehlerfrei. Diese kann ohne Probleme im Produktivsystem eingesetzt werden.

LTS Normale Distributionen haben meist eine Lebensdauer von ein bis zwei Jahren (bei Ubuntu 18 Monate). Danach werden für die jeweilige Version kein Support geleistet oder Updates zur Verfügung gestellt. Im Gegensatz dazu werden LTS-Versionen (Long Term Support) bei Ubuntu fünf Jahre lang unterstützt.

2.1.4 Einsatz

Linux findet inzwischen in vielen Bereichen sein Einsatzgebiet. Sei es als Desktop-, Serversystem oder im Embedded-Bereich.

Desktop

Linux lässt sich als Betriebssystem auf dem Desktop-Rechner alleine oder neben anderen Systemen wie Windows installieren. Hier kommen Desktop-Manager wie Gnome, KDE oder viele andere wie Xfce, LXDE, twm, ... zum Einsatz.

Aus über vielen tausend kostenlosen Programmen kann eine individuelle Installation (es gibt auch viele Voreinstellungen für bestimmte Anwendungsbereiche) durchgeführt und so das System nach den eigenen Wünschen erstellt werden (z. B. Textverarbeitung, Tabellenkalkulation, Video- und Fotobearbeitung, Netzwerktools, Spiele oder wissenschaftliche Anwendungen, ...).

Für die LinuxEssential-Zertifizierung ist dies das Einsatzgebiet. Abbildung 2.3 zeigt den Ubuntu 11.10 Desktop mit Gnome und Unity.

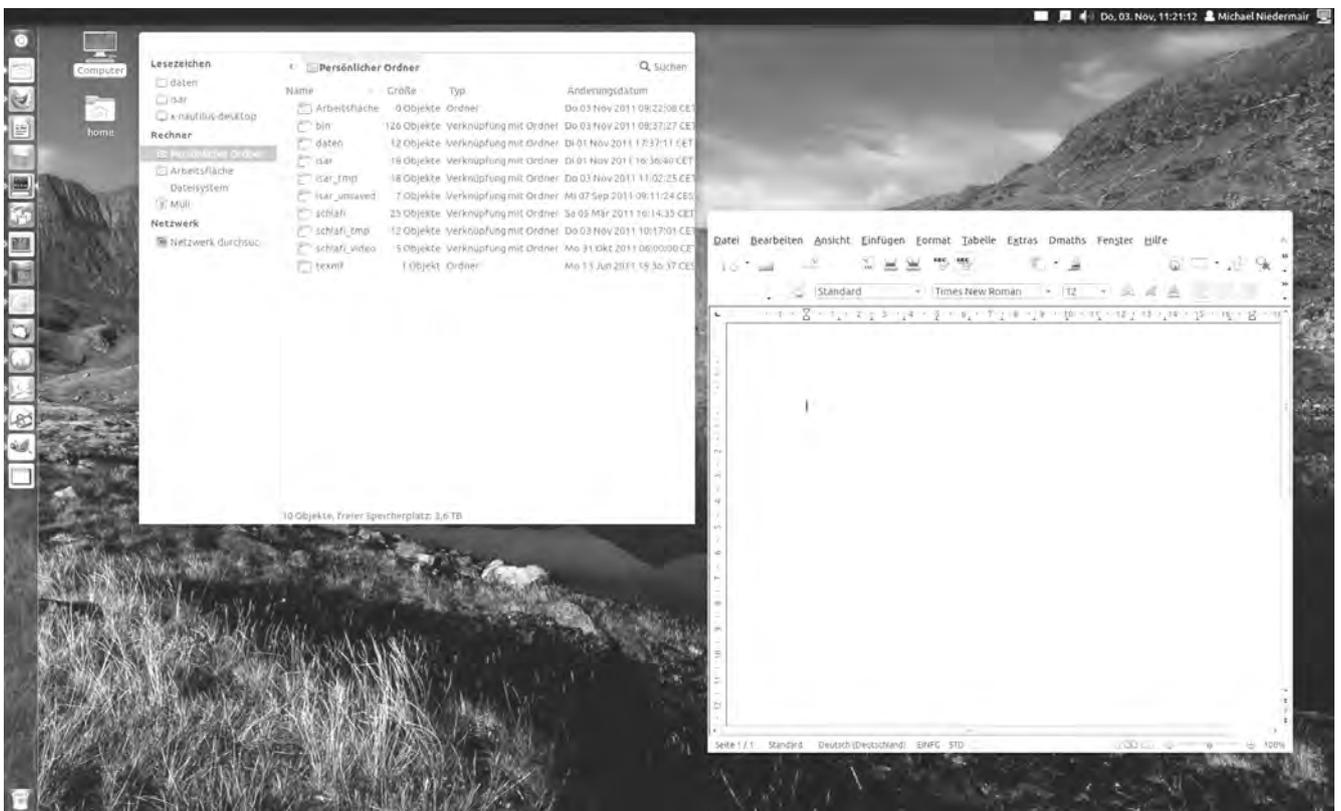


Abb. 2.3: Ubuntu 11.10 Desktop mit Gnome und Unity

Server

Durch die Kompatibilität von Linux mit anderen UNIX-Systemen hat es sich vor allem im Serverbereich besonders schnell etabliert, da es schon früh benötigte Serversoftware wie Webserver, Datenbankserver, Mailserver, Dateiserver oder Groupware kostenlos zur Verfügung stellte. Da Linux stabil läuft und einfach zu warten ist, wird es in vielen Rechenzentren verwendet. Der modulare Aufbau ermöglicht das Betreiben kompakter, dedizierter Server oder auch kompletter Clusterlösungen⁶, sowie die Bereitstellung einer Cloud⁷. Auch im Bereich der Virtualisierung (z. B. das Verwenden eines anderen Betriebssystems in einem Fenster mit VirtualBox⁸) hat sich Linux einen Namen gemacht.

Abbildung 2.4 zeigt die TOP500 Liste der schnellsten Computersysteme, welche von Linux mit 82 % angeführt wird.

⁶ Ein Rechnerverbund oder Computercluster, bezeichnet eine Anzahl von vernetzten Computern, die von außen als ein Computer gesehen werden. Fällt ein Computer aus, so übernehmen die anderen dessen Funktion, ohne dass man von außen etwas bemerkt.

⁷ Cloud Computing umschreibt den Ansatz, z. B. Rechenkapazität, Datenspeicher, Netzwerkkapazitäten oder auch fertige Software, dynamisch über ein Netzwerk zur Verfügung zu stellen.

⁸ siehe <https://www.virtualbox.org/>

⁹ Quelle: <http://top500.org/>

Operating System / Systems

June 2011



Abb. 2.4: TOP500 Liste der schnellsten Computersysteme⁹

Embedded Device

Als Embedded Linux (deutsch: „eingebettetes Linux“) bezeichnet man ein in ein Gerät eingebettetes System mit einem auf dem Linux-Kernel basierenden Betriebssystem.

Solche Systeme finden sich in HDTV-Receivern, DVD-Playern, DSL-Routern, industriellen Stromzählern, Firewall-Systemen und vielen mehr.

Abbildung 2.5 zeigt einen Linksys-DSL-Router mit Linux (hier mit DD-WRT – siehe <http://www.dd-wrt.com>).



Abb. 2.5: Linksys-Router mit DD-WRT-Linux

Smartphones & Tablets

Für Smartphones und Tablets (siehe Abbildung 2.6 und 2.7 auf der nächsten Seite) gibt es speziell optimierte Linux-Distributionen wie das Android-System, welches Telefonie- und Textnachrichten-Funktionen, diverse PIM-, Navigations- und Multimedia-Funktionen besitzt. Die Bedienung erfolgt meist über Multi-Touch-Bildschirme.



Abb. 2.6: Screenshot: HTC-Smartphone mit Linux



Abb. 2.7: Screenshot: Samsung Galaxy Tab mit Linux

2.1.5 Wichtige Open Source Anwendungen

Zu den wichtigen Open Source Anwendungen zählen:

- OpenOffice.org** Ein freies Office-Paket, das aus einer Kombination verschiedener Programme zur Textverarbeitung, Tabellenkalkulation, Präsentation und zum Zeichnen besteht. Es entstand aus den offengelegten Quelltexten des damaligen StarOffice im Jahr 2000 und wurde seither von Sun Microsystems entwickelt. Nach dem Kauf von SUN durch Oracle wurde die Entwicklung an die Apache Software Foundation weitergegeben.
- LibreOffice** LibreOffice ist aus OpenOffice.org im September 2010 entstanden, da viele Entwickler nicht mit der Entwicklerpolitik von Oracle einverstanden waren. Die Förderung wird durch eine Stiftung „The Document Foundation“ getragen. „Vereinfacht gesagt, ist das Ziel der Stiftung die Entwicklung eines Repertoires digitaler Produktivitäts- und Kreativitätswerkzeuge der nächsten Generation, indem eine nachhaltige, unabhängige und umfassende Community zur Entwicklung Freier und Open-Source-Software [FOSS] auf der Basis offener Standards, gefördert wird.“¹⁰
- Thunderbird** Mozilla Thunderbird (engl. für Donnervogel) ist ein Open-Source-E-Mail-Programm und -Newsreader des Mozilla-Projekts.
- Firefox** Mozilla Firefox (engl. für Feuerfuchs; bei uns Rotfuchs genannt) ist ein freier Webbrowser des Mozilla-Projektes.
- Blender** Blender ist eine freie 3D-Grafiksoftware mit Funktionen, um dreidimensionale Körper zu modellieren, texturieren, animieren und zu rendern.
- Gimp** GIMP (GNU Image Manipulation Program) ist ein Bildbearbeitungsprogramm, welches unter der GPL steht.
- Audacity** Audacity ist ein freier Audioeditor und -rekorder.
- ImageMagick** ImageMagick ist ein freies Softwarepaket zur Erstellung und Bearbeitung von Rastergrafiken.
- Apache** Der Apache HTTP Server ist der meistgenutzte Webserver im Internet.
- MySQL** MySQL ist ein relationales Datenbankverwaltungssystem; als Open-Source-Software sowie als kommerzielle Enterpriseversion für verschiedene Betriebssysteme verfügbar.
- PostgreSQL** PostgreSQL ist ein freies, objektrelationales Datenbankmanagementsystem (ORDBMS).

¹⁰ The Document Foundation: CommunityByLaws – Eine Satzung für die Community

NFS	Das Network File System ist ein von Sun Microsystems entwickeltes Protokoll, das den Zugriff auf Dateien über ein Netzwerk ermöglicht. Für Windows existieren NFS-Treiber, so dass Windows über NFS auf Linux-Server zugreifen kann.
Samba	Samba ist eine freie Software-Suite, die das Server-Message-Block-Protokoll (SMB) für Unix-Systeme verfügbar macht und unter anderem die Datei- und Druckdienste von Microsoft Windows unter Linux emuliert. Das Netzwerkprotokoll wird auch als Common Internet File System (CIFS), LAN-Manager- oder NetBIOS-Protokoll bezeichnet. Über Samba kann Windows ohne zusätzliche Treiberinstallation auf Linux-Server zugreifen und diese als Dateiablage genutzt werden.
OpenLDAP	OpenLDAP ist eine Implementierung des LDAP-Protokolls und ermöglicht eine Benutzerdatenverwaltung aufzubauen und zentral zu warten.
Postfix	Postfix ist ein Mail Transfer Agent (Mailserver).

2.1.6 Wichtige Open Source Programmiersprachen

Zu den wichtigen Open Source Programmiersprachen unter Linux (aber auch für Windows, ...) zählen:

C	C ist eine imperative ¹¹ Programmiersprache, die der Informatiker Dennis Ritchie in den frühen 1970er Jahren an den Bell Laboratories für die Systemprogrammierung des Betriebssystems Unix entwickelte.
C++	C++ ist eine höhere – objektorientierte, generische und prozedurale – Programmiersprache, die 1979 von Bjarne Stroustrup als Erweiterung von C entwickelt worden ist.
Perl	Perl ist eine freie, plattformunabhängige und interpretierte Programmiersprache (Skriptsprache), die von Linguist Larry Wall 1987 entwickelt worden ist. Als Vorlage diente dazu C, awk und Unix-Befehle. Ursprünglich war Perl dazu gedacht, Textdateien (insbesondere bei System- und Netzwerkadministration) schnell und effektiv zu verarbeiten.
Python	Python ist eine interpretierte, objektorientierte, aspektorientierte und funktionale Programmiersprache, die den Schwerpunkt auf Programmlesbarkeit legt.
PHP	PHP ist eine Skriptsprache zur Erstellung dynamischer Webseiten oder Webanwendungen.
Java	Java ist eine objektorientierte Programmiersprache die in alle Bereichen Einzug gefunden hat (Desktop, Server, Smartphones, ...).

2.2 Terminal und Konsole, erste Schritte auf der Kommandozeile

Linux stellt zwei Eingabemöglichkeiten bereit:

- Ein grafische Umgebung.
Hier wird meist mittels der Maus die jeweilige Funktion ausgewählt.
- Die Konsole bzw. das Terminal.
Hier werden die Befehle mit Hilfe der Tastatur eingegeben.
Die meiste Zeit werden wir hier mit der Konsole arbeiten.

2.2.1 Konsole bzw. Terminal starten



Wenn Sie eine aktuelle Ubuntu-Distribution verwenden, so steht Ihnen Unity zur Verfügung. Unity ist eine Arbeitsumgebung für PCs und Netbooks, die von Canonical entwickelte wurde und besonders sparsam mit Bildschirmplatz umgehen soll.

Auf der linken Seite des Desktops findet sich eine Starterleiste. Damit lassen sich diverse Programm, die man häufig benutzt, schnell aufrufen.

Das oberste Icon stellt den Unity-Starter dar, mit dem alle Programme gesucht und aufgerufen werden können (siehe Abbildung 2.8 auf der nächsten Seite). Dort kann man entweder die Bereiche „durchklicken“ oder mit einer Texteingabe das entsprechende Programm suchen und starten. Ist das Programm aufgerufen, so findet man ein entsprechendes Icon in der Startleiste. Durch einen -Klick auf das Icon erscheint ein Kontextmenu, mit dem man festlegen kann, dass das Startericon immer in der Starterleiste verbleiben soll (siehe Abbildung 2.9 auf der nächsten Seite).

¹¹ Bei Imperativer Programmierung, auch Prozedurale Programmierung genannt, wird ein Programm so entwickelt, dass der Quellcode aus einer Folge von Befehlen besteht, die vorgeben, in welcher Reihenfolge was vom Computer getan werden soll.



Abb. 2.8: Unity-Starter zum Suchen und Aufrufen von Programmen



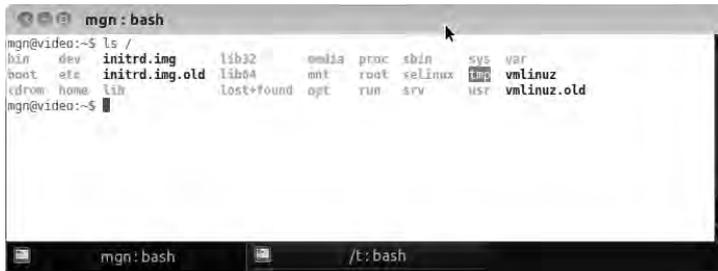
Abb. 2.9: Icons immer in der Starterleiste halten

Damit man auf der Konsole arbeiten kann, stellt Linux mehrere Möglichkeiten bereit:

Konsole X-Terminal

Paket: konsole

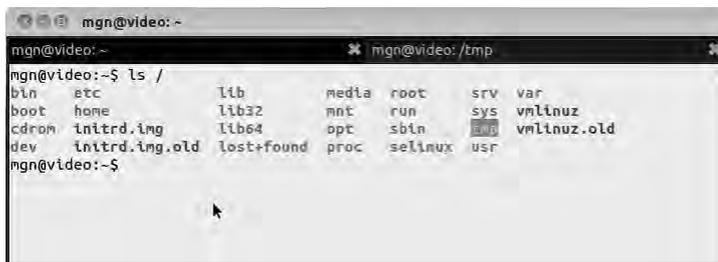
Konsole ist ein Terminal, der für die KDE-Plattform entwickelt worden ist. Es lassen sich damit mehrere Terminal-Sitzungen in einem Fenster ausführen. Ein umfangreiches Menü ermöglicht eine einfache Nutzung der vielen Funktionen.



Terminal GNOME-Terminal

Paket: gnome-terminal

Terminal ist ein Terminal, der für die Gnome-Plattform entwickelt worden ist. Auch hier lassen sich mehrere Terminal-Sitzungen verwalten und es bietet ähnliche Funktionen wie Konsole.



Xterm X-Terminal

Paket: xterm

Xterm ist ein sehr einfacher Terminal, der nicht speziell für eine Plattform entwickelt worden ist, sondern in allen grafischen Umgebungen (X-Window) arbeitet.

```
mgn@video:~$ ls /
bin  dev  etc  home  lib  lib32  lib64  lost+found  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
```

Welche Sie davon verwenden, ist für nachfolgende Übungen völlig egal, unser Favorit ist „Terminal“. Alle diese Terminals haben eins gemeinsam, sie führen die Standard-Shell aus (Voreinstellung).

Unter Ubuntu ist normalerweise die Standard-Shell die „dash“, es kann aber je nach Distribution auch die „bash“ sein.

Fehlende Programm-Pakete installieren

Ist ein Paket nicht installiert, so kann dies sehr einfach nachinstalliert werden. Öffnen Sie dazu ein Terminal (`xterm`, `konsole`, ...) und geben folgenden Befehl ein, wenn Sie z. B. GNOME-Terminalemulator installieren wollen:

```
sudo apt-get install gnome-terminal
```

Je nach System kann es sein, dass Sie dabei nach Ihrem Passwort gefragt werden.

Das Programm `sudo` sorgt dafür, dass das nachfolgende Programm unter root-Rechten (Rechte des Administrators) gestartet wird, da nur dieser Programme systemweit installieren darf. Das Programm `apt-get` installiert das nachfolgend genannte Paket unter einem Debian/Ubuntu-System.

Alternativ können Programmpakete auch mit dem GUI-Programm `synaptic` ein Paket installiert werden.

Übung:

1. Rufen Sie von Ihrer grafischen Umgebung ein Terminal Ihrer Wahl auf und installieren Sie das Paket „`kterm`“, einen Multi-lingualen Terminalemulator für X.
2. Installieren Sie das Paket „`synaptic`“, wenn es noch nicht standardmäßig installiert ist und rufen Sie `synaptic` auf. Installieren Sie jetzt mit `synaptic` den Terminalemulator „Xfce-Terminalemulator“.



2.2.2 Was ist eine Shell?

Die Shell (Muschel, Schale) ist ein Programm, das als Kommandointerpreter für den Dialog und die Interaktion mit dem Benutzer zuständig ist, z. B.:

DOS/WIN9x `command.com`

Windows `cmd.exe`

Linux/UNIX `bash, dash, sh, csh, zsh, ash` usw.

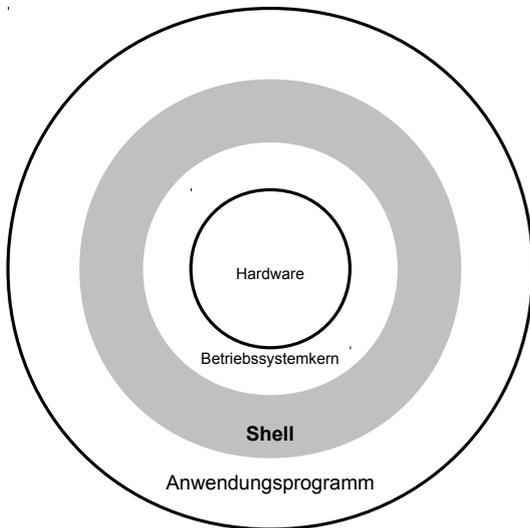


Abb. 2.10: Shell-Schale

Die Shell stellt an einem Terminal eine Kommandozeile zur Verfügung, setzt Benutzereingaben in entsprechende Betriebssystemaufrufe um, startet Systemprogramme und gibt die Ausgaben und Meldungen dieser Programme sowie die des Betriebssystems auf der Konsole aus.

Hilfe zur von uns verwendeten `bash`-Shell gibt es mit den Kommandos:

- `help`
- `help <Befehl>`
- `man bash`

Wichtige Eigenschaften der `bash`-Shell

- Eingabekomplettierung mit der Tabulator-Taste¹²
- History (Gedächtnis für Kommandos)
- Spezifizieren von Dateinamen mithilfe von Mustern und Jokerzeichen (wildcards) (mehr dazu auf Seite 27)
- Ein- und Ausgabeumleitung bei Programmen (mehr dazu auf Seite 71)
- Verbinden von Ein- und Ausgabedatenströmen mehrerer Programme (mehr dazu auf Seite 72)
- Kommandozeileneditor (mehr dazu auf Seite 40)
- Leistungsfähige Skriptsprache (shell script) (mehr dazu auf Seite 68)

Gedächtnis der `bash`-Shell

Die Bash speichert die letzten eingegebenen Kommandos und deren Parameter in einer History (Datei `~/.bash_history`). Mit wird dabei das zuletzt getippte Kommando angezeigt. Drückt man mehrmals, so kann man die Liste hinauf, mit hinab wandern. Mit `[Strg]+r` kann die History rückwärts gezielt nach Kommandos durchsucht werden. Die Anzahl an Kommandos, die gespeichert werden soll, wird in der Variablen `HISTSIZE` gespeichert. Ist die Grenze des Speichers erreicht, wird automatisch der älteste Eintrag gelöscht. Beim Ausloggen wird die History in die Datei `~/.bash_history` gespeichert. Soll ein anderer Dateiname verwendet werden, so muss dieser in der Variablen `HISTFILE` gespeichert werden.

Die wichtigsten Parameter von `history` sind dabei:

- `-n` fügt die neuen Einträge aus der Datei der History im Speicher hinzu
- `-r` fügt alle Einträge aus der Datei der History im Speicher hinzu

¹² Gibt man die ersten Buchstaben eines Kommando oder einer Datei ein und drückt die Tabulator-Taste, so versucht die Bash, automatisch die Eingabe zu vervollständigen. Drückt man ein zweites Mal die Tabulatortaste, so wird bei mehreren Möglichkeiten eine Auswahl angezeigt.

- a schreibt die aktuellen History-Einträge im Speicher in die Datei
- w schreibt alle History-Einträge im Speicher in die Datei
- c löscht die komplette Liste im Speicher. Wichtig ist dies besonders für `root` und für den Fall, dass man Kommandos mit Passwörtern in der Shell aufruft. Die Passwörter werden dann auch in der History gespeichert, was ein Sicherheitsproblem darstellen könnte. Viele schreiben daher in die `~/.bash_logout`, die beim Ausloggen aufgerufen wird, folgende Zeilen:

```
# Konsole löschen, wenn möglich
[ -x /usr/bin/clear_console ] && /usr/bin/clear_console -q
# History löschen
history -c
# Datei auf Größe 0 setzen (somit den Inhalt löschen)
> ~/.bash_history
```

Variablen ausgeben

Mit dem Kommando `echo` kann beliebiger Text auf der Konsole ausgegeben werden. Sind im Text Variablen enthalten, so werden diese von der Bash mit deren Inhalt ersetzt.

echo Variable HISTSIZE=\$HISTSIZE

Variable HISTSIZE=1000

Wird `-n` verwendet, so wird nach der Ausgabe kein Zeilenumbruch durchgeführt.

Tastenkürzel der bash-Shell

Nachfolgend einige Tastenkürzel:

- Strg**+**a** springt zurück an den Anfang der Eingabe
- Strg**+**e** springt ans Ende der Eingabe
- Esc**+**b** geht ein Wort zurück
- Esc**+**f** geht ein Wort vor
- Strg**+**b** springt ein Zeichen zurück
- Strg**+**f** springt ein Zeichen vor
- Strg**+**k** löscht alle Zeichen von der aktuellen Cursorposition bis zum Ende der Eingabe
- Strg**+**u** löscht alle Zeichen von der aktuellen Cursorposition bis zum Anfang der Eingabe
- Strg**+**w** löscht das Wort links vom Cursor
- Strg**+**t** vertauscht die beiden Zeichen vor und unter dem Cursor
- Esc**+**t** vertauscht die beiden vorangegangenen Wörter
- Strg**+**l** löscht den Inhalt des Terminal-Fensters
- Strg**+**s** stoppt die Bildschirmausgabe
- Strg**+**q** Bildschirmausgabe wieder fortsetzen
- Strg**+**c** Abbruch eines Kommandos

Alias-Kommandos

Werden Kommandos mit Parameter häufig verwendet, so ist es sinnvoll, diese als Alias zu definieren.

Mit nachfolgendem Kommando wird ein neues „Kommando“, ein Alias definiert, der eine Verzeichnisebene höher springt.

```
alias ..='cd ..'
```

Ein Alias kann mit dem Kommando `unalias` wieder gelöscht werden. Einige Aliase werden schon in der Grundkonfiguration definiert.

```
alias
alias dir='dir --color=auto'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
```

2.3 Hilfen zu Kommandos

Fast alle Kommandos geben eine kurze Hilfestellung zu ihren Argumenten und Optionen aus, wenn man sie folgendermaßen aufruft:

```
Kommandoname --help oder Kommandoname -?
```

Umfangreichere Hilfestellungen geben die Man-Pages (manual pages). Mit `man <Kommando>` (z. B. `man less`) wird vom Kommando `less` die entsprechende Man-Page aufgerufen. In den Man-Pages kann mit den Pfeil- und Bildlauf-Tasten manövriert werden.

Begriffe können durch Eingabe von `/Begriff` gesucht werden. Weitere Fundstellen springt man dann mit der Taste `[n]` an, vorherige mit `[N]`.

Mit `[q]` verlässt man die Man-Pages!¹³

```
man less
LESS(1) LESS(1)

NAME
    less - opposite of more

SYNOPSIS
    less -?
    less --help
    less -V
    less --version
    less [-[+]aABcCdeEfFgGiIJKLmMnNqQrRsSuUVvWx~]
        [-b space] [-h lines] [-j line] [-k keyfile]
        [-{oO} logfile] [-p pattern] [-P prompt] [-t tag]
        [-T tagsfile] [-x tab,...] [-y lines] [-[z] lines]
        [-# shift] [+][+]cmd] [--] [filename]...
    (See the OPTIONS section for alternate option syntax with long option
    names.)

DESCRIPTION
    Less is a program similar to more (1), but which allows backward - move
    ment in the file as well as forward movement. Also, less does not have
    to read the entire input file before starting, so with large input
    files it starts up faster than text editors like vi (1). Less uses
    termcap (or terminfo on some systems), so it can run on a variety of
    terminals. There is even limited support for hardcopy terminals. (On
    a hardcopy terminal, lines which should be printed at the top of the
    screen are prefixed with a caret.)
...

```

```
less --help
SUMMARY OF LESS COMMANDS

Commands marked with * may be preceded by a number, N.
Notes in parentheses indicate the behavior if N is given.

h H          Display this help.
q :q Q :Q ZZ Exit.
-----

MOVING

e ^E j ^N CR * Forward one line (or N lines).
y ^Y k ^K ^P * Backward one line (or N lines).
f ^F ^V SPACE * Forward one window (or N lines).
b ^B ESC-v * Backward one window (or N lines).
z * Forward one window (and set window to N).
w * Backward one window (and set window to N).
...

```

Sollen alle Man-Pages nach einem Begriff durchsucht werden, so hilft das Kommando `apropos` gefolgt von dem Begriff, den man sucht. Eine ähnliche Funktion bietet das Kommando `whatis`.

¹³ Installiert man das Paket `manpages-de`, so erhält man auch deutschsprachige manpage-Seiten, wenn diese vorhanden sind.

HOWTO's

HOWTOs sind umfassendere Dokumente, die nicht nur einzelne Kommandos beschreiben, sondern komplette Problemlösungen, wie z. B. das Samba-HOWTO, welches beschreibt, wie man eine Verzeichnisfreigabe für Windows-Rechner bereitstellt. Die HOWTOs befinden sich als Paket unter `/usr/share/doc/` bzw. `/usr/share/doc/howto/` und stehen als Text- oder HTML-Datei zur Verfügung. Aktuelle Versionen der HOWTOs finden sich auf den Seiten des Linux Documentation Project (<http://www.tldp.org>).

weitere Informationsquellen

- <http://www.linux.org>
- <http://linuxos.pro/>
- <http://www.linuxwiki.de>
- <http://lwn.net>
- <http://www.linuxforum.com>
- <http://freecode.com/>
- <http://www.linux-knowledge-portal.de>
- <http://www.linuxjournal.com/>
- <http://www.linuxfocus.org>
- <http://www.linux-center.org/en/>

Übung:

1. Starten Sie ein Terminal und rufen Sie die Hilfen zum Programm `ls` einmal mit `man` und mit `--help` auf.



2.4 Kommandos

Bei Kommandos/Befehlen unterscheidet man, ob sie mit Parameter aufgerufen werden oder nicht.

2.4.1 Einfache Kommandos

Einfache Kommandos/Befehle werden im Terminal direkt eingegeben.

Beispiele:

date gibt das aktuelle Datum aus

```
So 8. Jul 07:28:45 CEST 2012
```

cal Kalender des aktuellen Monats

```

      Juli 2012
So Mo Di Mi Do Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

```

who listet alle Benutzer auf, die im System angemeldet sind

```
mgn      pts/0      2012-07-08 07:05 (:0.0)
```

Die Spalten haben dabei folgende Bedeutung: Benutzername, Terminerkennung und Einlogzeit

2.4.2 Kommandos mit Argumenten

Syntax:

```
Kommandoname -Option [-weitere Optionen ... ] Argument [weitere Argumente ... ]
```

Angaben, die dem Kommandonamen folgen, werden als Parameter interpretiert. Man unterscheidet zwischen Argumenten und Optionen.

Argumente Dateinamen oder sonstige Angaben, die von den Kommandos verarbeitet werden.

Optionen Sind eine Art Schalter, die eine Feineinstellung der Kommandos bewirken.

Sie werden durch ein vorangestelltes Minuszeichen gekennzeichnet und werden meist vor den eigentlichen Argumenten angegeben. Optionen können auch Parameter enthalten.

Beispiele:

cal 12 2011 Kalender für Dezember 2011

```

Dezember 2011
So Mo Di Mi Do Fr Sa
                1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

```

ls -la /etc/sh* listet alle Dateien (mit zusätzlichen Angaben) im Verzeichnis `/etc` auf, die mit „sh“ beginnen

```

-rw-r----- 1 root shadow 1274 Jul  8 07:06 /etc/shadow
-rw----- 1 root root 1344 Jul  8 07:06 /etc/shadow-
-rw-r--r-- 1 root root  84 Apr 29 17:08 /etc/shells

```

Weitere Möglichkeiten des Kommandoaufrufes

Wird ein Kommando ganz normal aufgerufen, so wird es im Vordergrund ausgeführt und es können in dieser Zeit im Terminal keine weiteren Kommandos ausgeführt werden.

Kommando im Hintergrund ausführen

Wird am Ende des Kommandos (nach den Parametern) ein `&` angegeben, so wird das Kommando im Hintergrund ausgeführt und man kann sofort das nächste Kommando eingeben.

Beispiel:

gedit datei.txt &

Das Kommando `gedit` (ein grafischer Texteditor) wird in der Shell gestartet (in einem extra Fenster) und es kann sofort ein weiteres Kommando eingegeben werden. Als Information wird dabei die Job-ID ausgegeben (das Kommando `jobs` zeigt alle IDs an).

Mit dem Kommando `fg` kann ein Programm, welches im Hintergrund ist, wieder in den Vordergrund geholt werden. Ein laufendes Programm kann mit `Strg+Z` unterbrochen und dann mit `bg` in den Hintergrund verschoben werden. Gibt es mehrere Programme im Hintergrund, muss die Job-ID angegeben werden.

Kommandos hintereinander ausführen

Will man, dass mehrere Kommandos hintereinander ausgeführt werden, so trennt man diese durch einen Strichpunkt.

Beispiel:

cmd1 ; cmd2 ; cmd3

Dabei wird zuerst `cmd1` ausgeführt. Wird dieses beendet, startet `cmd2` und wenn dieses wiederum beendet wird, startet `cmd3`.

2.4.3 Laufende Kommandos/Prozesse anzeigen

Wird ein Kommando aufgerufen, so wird dafür vom Betriebssystem ein Prozess erzeugt. Mit dem Kommando `ps` lassen sich Prozesse anzeigen.

```
ps
  PID TTY          TIME CMD
 2961 pts/0    00:00:00 bash
 3028 pts/0    00:00:03 xterm
 3088 pts/0    00:00:00 xterm
 7415 pts/0    00:00:00 ps
```

Ruft man `ps` ohne Parameter auf, so werden nur die Prozesse angezeigt, die man selbst in der aktuellen Sitzung gestartet hat. In der Ansicht wird dabei die Prozessnummer (PID), der Name der verwendeten Konsole (TTY), die Zeit, wie lange der Prozess schon läuft und das entsprechende Kommando angezeigt.

Die wichtigsten Parameter von `ps` sind:¹⁴

- a wählt alle Prozesse mit einem Terminal aus, inklusive der anderer Benutzer
- x wählt Prozesse ohne kontrollierende TTYs (z. B. Systemprozesse) aus
- u zeigt den Besitzer (User) des Prozesses an
- H zeigt die Prozess-Hierarchie in einer Baum-Struktur an

```
ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0  24432  2412 ?        Ss   17:33   0:00 /sbin/init
root         2  0.0  0.0     0     0 ?        S    17:33   0:00 [kthreadd]
...
root        964  0.0  0.0  49948  2868 ?        Ss   17:33   0:00 /usr/sbin/sshd -D
syslog     977  0.0  0.0 249464  1668 ?        Sl   17:33   0:00 rsyslogd -c5
root       1020  0.0  0.0 234000  5344 ?        Ssl  17:33   0:00 NetworkManager
...
mgn       7486  0.0  0.0  18368  1272 pts/2    R+   20:43   0:00 ps aux
```

Alternativ kann auch das Kommando `ps tree` verwendet werden (siehe Abbildung 2.11).

```
init--NetworkManager--dhclient
|
|           |-dnsmasq
|           `--2*[{NetworkManager}]
|-accounts-daemon---{accounts-daemon}
|-lightdm--Xorg
|           |-lightdm--gnome-session--bluetooth-apple---2*[{bluetooth-apple}]
|           |           |
|           |           |           |-compiz--gtk-window-deco---2*[{gtk-window-deco}]
|           |           |           `--3*[{compiz}]
|           |           |           |-deja-dup-monito---2*[{deja-dup-monito}]
|           |           |           |-gdu-notificatio--{gdu-notificatio}
|           |           |           `--{gdu-notificatio}\247\310G\313\177
|           |           |           |-nautilus---2*[{nautilus}]
|           |           |           |-ssh-agent
|           |           |           |-telepathy-indic---2*[{telepathy-indic}]
|           |           |           |-update-notifier---3*[{update-notifier}]
|           |           |           |-xosview
|           |           |           |-zeitgeist-datah---{zeitgeist-datah}
|           |           |           `--3*[{gnome-session}]
|           |           `--{lightdm}
|           `--2*[{lightdm}]
|-zeitgeist-daemo---{zeitgeist-daemo}
`-zeitgeist-fts--cat
   `--{zeitgeist-fts}
```

Abb. 2.11: Anzeige der Prozesse in einer Baumansicht mit `ps tree` (Auszug)

Frage:

Ermitteln Sie über die man-Page, welche Informationen mit `ps aux` ausgegeben werden.

¹⁴ Das Kommando `ps` gehört zu den wenigen Programmen, die Parameter in verschiedenen Notationen zulassen: UNIX-Option mit einem `'--'`, BSD-Optionen ohne `'--'` und die GNU Langform mit `'--'`

Prozess beenden

Einen laufenden Prozess in der Konsole kann man meist über **Strg+C** beenden. Bei einem Prozess, der im Hintergrund läuft und an keine Konsole gebunden ist oder einem Systemprozess ist dies aber nicht möglich.

Mit dem Kommando `kill` können Prozesse beendet werden. Dabei kann der Anwender nur seine eigenen Prozesse beenden, `root` dagegen kann alle Prozesse beenden.

```
kill [signal] <PID> [weitere PIDs...]
```

Dabei wird mit Hilfe des Signals¹⁵ mitgeteilt, wie der Prozess zu beenden ist. Wird kein Parameter angegeben, so wird `SIGTERM` (-15) verwendet, welches dem Prozess „freundlich“ mitteilt, dass er sich beenden soll (normalerweise werden hier alle Daten gespeichert). Allerdings kann der Prozess dieses Signal ignorieren. Die „brutale“ Variante ist `SIGKILL` (-9), welches den Prozess sofort beendet (evtl. Daten, die der Prozess noch nicht gespeichert hat, gehen dabei verloren). Anstelle von `kill` kann auch `killall` verwendet werden. Hierbei wird nicht die PID verwendet, sondern der Kommandoname des Prozesses. Ist das Kommando mehrfach aufgerufen worden, so werden alle Prozesse mit diesem Namen beendet.

Beispiel:

```
ps
PID TTY          TIME CMD
3303 pts/3        00:00:00 bash
5177 pts/3        00:00:00 xterm
5236 pts/3        00:00:00 ps
kill -9 5177
```

Fortlaufende Anzeige von Prozessinformationen

Mit `ps` wird nur der aktuelle Zeitpunkt betrachtet. Sollen die Prozessinformationen ständig aktualisiert angezeigt werden, so hilft das Kommando `top`. Dabei wird standardmäßig der Prozess mit dem höchsten CPU-Verbrauch oben angezeigt. Zusätzlich werden noch weitere Informationen, wie z. B. die Speicherauslastung ausgegeben.

```
top - 21:02:41 up 3:29, 4 users, load average: 1.16, 1.08, 0.96
Tasks: 199 total, 2 running, 196 sleeping, 0 stopped, 1 zombie
Cpu(s): 0.8%us, 1.9%sy, 0.0%ni, 97.3%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 16365944k total, 3193964k used, 13171980k free, 130656k buffers
Swap: 3999740k total, 0k used, 3999740k free, 1243236k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2388	mgn	20	0	1402m	294m	44m	S	9	1.8	22:52.08	compiz
1234	root	20	0	185m	81m	38m	S	1	0.5	5:35.03	Xorg
2415	mgn	20	0	30740	2276	1864	S	1	0.0	1:24.93	xosview
2358	mgn	20	0	28324	3940	624	S	0	0.0	0:24.29	dbus-daemon
8042	mgn	20	0	17468	1360	952	R	0	0.0	0:00.01	top
1	root	20	0	24432	2412	1348	S	0	0.0	0:00.85	init
2	root	20	0	0	0	0	S	0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0	0.0	0:00.81	ksoftirqd/0
6	root	RT	0	0	0	0	S	0	0.0	0:00.00	migration/0
7	root	RT	0	0	0	0	S	0	0.0	0:00.03	watchdog/0
8	root	RT	0	0	0	0	S	0	0.0	0:00.00	migration/1
10	root	20	0	0	0	0	S	0	0.0	0:00.70	ksoftirqd/1
12	root	RT	0	0	0	0	S	0	0.0	0:00.03	watchdog/1
13	root	RT	0	0	0	0	S	0	0.0	0:00.00	migration/2
15	root	20	0	0	0	0	S	0	0.0	0:00.61	ksoftirqd/2
16	root	RT	0	0	0	0	S	0	0.0	0:00.05	watchdog/2

Speicherauslastung anzeigen

Will man nur Informationen über die Speicherauslastung, so verwendet man das Kommando `free`.

```
free
total used free shared buffers cached
Mem: 16365944 3103556 13262388 0 130704 1243232
-/+ buffers/cache: 1729620 14636324
Swap: 3999740 0 3999740
```

Ohne Parameter zeigt `free` die Speicherangaben in Bytes an (mit `-k` in Kilobytes, `-m` in Megabytes und `-g` in Gigabytes).

```
free -m
total used free shared buffers cached
Mem: 15982 3029 12952 0 128 1214
-/+ buffers/cache: 1687 14295
Swap: 3905 0 3905
```

¹⁵ Mit `kill -l` werden alle Signale angezeigt, die vom System unterstützt werden.

2.5 Grundlagen der Dateiverwaltung

Damit man auf der Konsole mit Dateien und Verzeichnissen arbeiten kann, ist es wichtig zu verstehen, wie Datei- und Pfadnamen aufgebaut sind und wie man damit arbeitet.

2.5.1 Namenskonventionen im Dateisystem

- Zwischen Groß- und Kleinschreibung wird abhängig vom Dateisystem unterschieden
Es gibt case-sensitive (strenge Unterscheidung), case-preserving (beim Anlegen wird die gegebene Schreibweise verwendet, beim Zugriff auf existierende Dateien spielt sie keine Rolle wird aber wie geschrieben gemeldet) und case-insensitive (beim Schreiben wird umgewandelt, beim Lesen ist Groß-/Kleinschreibung egal) Dateisysteme.
- Dateinamen können 255 Zeichen und länger sein, je nach verwendetem Dateisystem
- Sonderzeichen in Dateinamen sind erlaubt (Vorsicht!)
- Dateinamen, die Leerzeichen enthalten, müssen in Hochkommas eingeschlossen sein
- Dateinamen, die mit einem Punkt beginnen, sind versteckte Dateien
- Der forward slash „/“ ist das Trennzeichen für Verzeichnisse

2.5.2 Jokerzeichen und Reguläre Ausdrücke der Shell

Der Einsatz von Jokerzeichen (wildcards) bei der Angabe von Dateinamen in Kommandoargumenten veranlasst die Shell, die gesuchten Dateinamen zu finden und dann erst die kompletten Namen der gefundenen Dateien als Argumente dem Kommando zu übergeben. Kombinationen von Jokerzeichen werden Reguläre Ausdrücke genannt.¹⁶

Achtung:

- Die Auswertung der Regulären Ausdrücke erfolgt bereits durch die Shell, nicht durch das aufgerufene Kommando!
- Reguläre Ausdrücke von `grep`, `sed`, `python` usw. unterscheiden sich stark in ihrer Syntax zu denen der Shell!¹⁷

Bedeutung der Zeichen:

- ? steht für genau ein beliebiges Zeichen
- * steht für kein oder beliebig viele Zeichen
Der Stern „*“ erfasst alle Zeichen, auch Punkte, außer denjenigen am Anfang des Dateinamens
Der reguläre Ausdruck `*graf*` trifft zu auf:
`grafic.doc`, `junger-graf`, `steffigrafundherragassi`, `readme.graf`, `graf`
- [123] genau eines der angegebenen Zeichen
- [a-f] ein Zeichen aus dem angegebenen Bereich
- [!abc] keines der angegebenen Zeichen
- [^abc] keines der angegebenen Zeichen
- ~ Abkürzung für das Home-Verzeichnis des gerade angemeldeten Benutzers

Beispiele zum Kommando „ls“

Das Kommando `ls` zeigt Dateien und Verzeichnisse an.

- `ls [a-f]*` alle Dateien, die mit einem der Buchstaben von a bis f beginnen
- `ls *[_.-]` alle Dateien, die irgendwo im Namen ein „_“, „.“ oder „-“ haben
- `ls *.[!hc]` alle Dateien, die nicht mit `.h` oder `.c` enden

Hinweis: Der Stern „*“ erfasst nicht nur die Dateien, sondern auch die Verzeichnisse, `ls *` durchsucht auch die Dateien der Unterverzeichnisse. Dies kann durch die Option `-d` ausgeschaltet werden.

¹⁶ Im Englischen werden diese Zeichen auch „glob“ genannt, daher wird die Verwendung von Jokerzeichen auch gerne als „globbing“ bezeichnet.

¹⁷ Der Begriff „reguläre Ausdrücke“ wird sehr häufig als Synonym für PCRE (Perl Compatible Regular Expressions) oder extended regular expressions oder zumindest basic regular expressions verstanden. Bei dem einfachen globbing der Shell fehlen dafür auf jeden Fall die Auswahlmöglichkeiten für Ausdrücke, aber auch die Syntax für Zeichenwiederholungen (Quantoren). POSIX regular expressions sind allgemein etwas anderes als das, was die Shell bietet.

Einige der wichtigsten Linux/UNIX-Kommandos zur Dateiverwaltung

- ls** Dateien und Verzeichnisse anzeigen
- cd** Verzeichniswechsel
- mkdir** Erstellen von Verzeichnissen
Soll eine Verzeichniskette erstellt werden, so muss der Parameter `-p` verwendet werden.
- rmdir** Löschen von leeren Verzeichnissen
Soll ein Verzeichnis mit Inhalt gelöscht werden, so verwendet man:
`rm -r Verzeichnis`
- pwd** gibt das aktuelle Verzeichnis aus
- cp** Kopieren von Dateien/Unterverzeichnissen
- rm** Löschen von Dateien/Unterverzeichnissen
- mv** Umbenennen/Verschieben von Dateien/Unterverzeichnissen
- file** Analysieren des Inhalts von Dateien
- cat** Inhalt von (Text-)Dateien ausgeben
- less** Betrachteten von Textdateien
- test** universelles Kommando um zu prüfen, ob z. B. eine Datei existiert
- find** nach bestimmten Dateien suchen
- grep** Suchen von Zeichenketten in Dateien
- sort** Textdateien zeilenweise sortiert ausgeben
- echo** Meldung auf der Konsole ausgeben
- touch** setzt das letzte Zugriffsdatum einer Datei auf das aktuelle Datum. Ist die Datei nicht vorhanden, so wird eine Datei mit der Dateigröße 0 erstellt.

Quoting

Möchte man nicht, dass Jokerzeichen bereits durch die Shell ausgewertet werden, so muss man ihnen ihre Sonderbedeutung durch Quoting nehmen. Es gibt mehrere Möglichkeiten:

- Voranstellen eines „\“ vor jedes Jokerzeichen bzw.
- den Regulären Ausdruck in einfache Hochkommas (') schreiben.

ls *.??? ohne Quoting

```
test.???\n\ntest.aaa\n\ntest.bbb\n\ntest.ccc
```

ls *. '???' mit Quoting

```
test.???
```

Versteckte Dateien

Versteckte Dateien (z. B. `.profile`) beginnen mit einem Punkt und werden von der Shell nur angezeigt, wenn das Kommando `ls` mit der Option `-a` oder durch die explizite Angabe des Punktes z. B. mit `.*` aufgerufen wird.

ls * ohne versteckte Dateien/Verzeichnisse

```
test.aaa\n\ntest.bbb\n\ntest.ccc
```

ls -d .* versteckte Dateien/Verzeichnisse anzeigen¹⁸

```
.\n\n..\n\ntest.xxx
```

Das Verzeichnis „.“ stellt einen Verweis auf das aktuelle Verzeichnis dar. Das Dateisystem nutzt den Punkt-eintrag dazu, alle Dateinamen des entsprechenden Verzeichnisses zu verwalten. Das Verzeichnis „..“ ist ein Verweis auf das übergeordnete Verzeichnis (parent).

Beispiel: Eine Datei im aktuellen Verzeichnis kann über `./Datei` angesprochen werden. Eine Datei im übergeordneten Verzeichnis über `../Datei`.

ls -a alle Dateien/Verzeichnisse anzeigen

```
.\n\n..\n\ntest.aaa\n\ntest.bbb\n\ntest.ccc\n\ntest.xxx
```

Rekursives Auflisten

Sollen alle enthaltenen Unterverzeichnisse rekursiv durchsucht werden, so verwendet man den Parameter `-R`.

Achtung: Der Parameter `-r` liefert eine umgekehrte Reihenfolge beim Sortieren.

```
ls -R /tmp
```

Eine Alternative zum gezielten Durchsuchen aller Unterverzeichnisse bietet das Kommando `find` (siehe Abschnitt 5.1.3 auf Seite 92).

¹⁸ Mit „-d“ wird bei `ls` der Inhalt von Verzeichnissen (hier `./` und `../`) nicht angezeigt.



Übung:

1. Im Verzeichnis /home/auricka/bilder befinden sich folgende Bilder:

- | | | |
|----------------|----------------|----------------|
| ① RIMG0001.JPG | ⑥ RIMG0003.raw | ⑩ IMAG0023.jpg |
| ② RIMG0001.raw | ⑦ RIMG0004.JPG | ⑪ IMAG0023.raw |
| ③ RIMG0002.JPG | ⑧ RIMG0004.raw | ⑬ IMAG0024.jpg |
| ④ RIMG0002.raw | ⑨ IMAG0022.jpg | ⑭ IMAG0024.raw |
| ⑤ RIMG0003.JPG | ⑩ IMAG0022.raw | |

Erzeugen Sie mit `mkdir test` ein neues Verzeichnis, wechseln Sie mit `cd test` in das Verzeichnis und legen Sie die Dateien ①-⑭ jeweils mit dem Befehl `touch Dateiname` an.

Geben Sie jeweils die Nummern der Dateinamen an, die angezeigt werden, wenn der angegebene Befehl ausgeführt wird!

- | | | | |
|---------------------------------|-------|------------------------------------|-------|
| a) <code>ls *</code> | _____ | g) <code>ls ??????????????</code> | _____ |
| b) <code>ls RIM*</code> | _____ | h) <code>ls IMAG002?.raw</code> | _____ |
| c) <code>ls *JPG</code> | _____ | i) <code>ls RIMG000[1-3]*</code> | _____ |
| d) <code>ls IM*.jpg</code> | _____ | j) <code>ls IMAG00[1-4].jpg</code> | _____ |
| e) <code>ls *A*</code> | _____ | k) <code>ls [^R]*raw</code> | _____ |
| f) <code>ls IMAG002?.raw</code> | _____ | l) <code>ls *.[jJ][pP][gG]</code> | _____ |

2. Im Verzeichnis /tmp befinden sich zur Zeit folgende Dateien:

- | | | | | | |
|---|----------------------|----|-------------|----|------------------|
| 1 | Abc-datei-Nr.1 | 9 | File | 17 | Index.HTM |
| 2 | ABC-datei-Nr.1 | 10 | file02 | 18 | INDEX.html |
| 3 | acrobat.ps.pdf | 11 | file11 | 19 | INDEX.HTM |
| 4 | Command.com | 12 | file13 | 20 | index.html |
| 5 | Concurrent-file-No.1 | 13 | file19 | 21 | liste |
| 6 | Datei-File19 | 14 | File19 | 22 | schock.Datei.198 |
| 7 | Datei-File29 | 15 | Filiale28 | 23 | senseless-file88 |
| 8 | Datei File29 | 16 | Hifileasure | 24 | INDEXOMAT |

Ergänzen Sie die Argumente zu `ls`, damit die jeweils genannten Dateien ausgegeben werden!

Beispiel: `ls ?ile??` gibt die Dateien 10,11,12,13,14 aus.

- | | | | |
|--------------------------|---------------|--------------------------|----------|
| a) <code>ls _____</code> | (10,11,12,13) | f) <code>ls _____</code> | (6,7) |
| b) <code>ls _____</code> | (3,22) | g) <code>ls _____</code> | (7,8) |
| c) <code>ls _____</code> | (6,7,8) | h) <code>ls _____</code> | (6,7,23) |
| d) <code>ls _____</code> | (9,14,15) | i) <code>ls _____</code> | (17,19) |
| e) <code>ls _____</code> | (18,19,24) | j) <code>ls _____</code> | (1,2,22) |

2.5.3 Dateiarnten

Mit `ls -la` werden zusätzliche Informationen angezeigt.

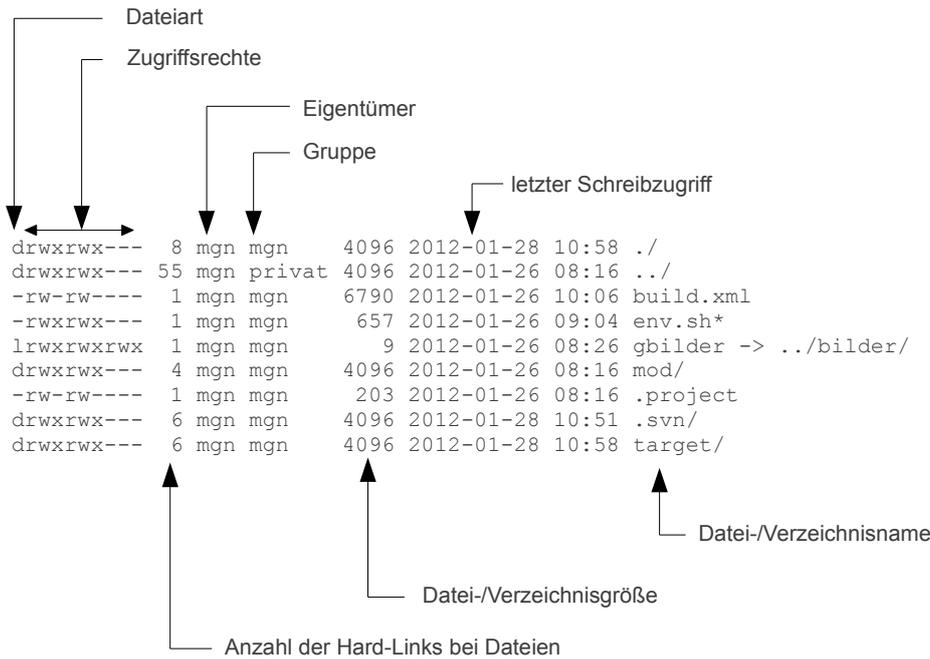


Abb. 2.12: Ansicht mit 'ls -la'

gewöhnliche Datei (ordinary file) Texte, Programme, Binärcode, Systembibliotheken, ...

Gerätefile (special file)

- logische Beschreibung der Hardware (z. B. Drucker, Monitor...)
- Über die Gerätefile wird auf die Gerätetreiber zugegriffen.
- lesender und schreibender Zugriff wie auf gewöhnliche Dateien
- Man unterscheidet zeichenorientierte Gerätefile (z. B. für Terminal) und
- blockorientierte Gerätefile (z. B. für Festplatte).

Beispiele wichtiger Gerätefile:

- `/dev/fd0` erstes Floppylaufwerk, meist ein Link auf die Gerätefile `/dev/fd0H1440`
- `/dev/sda` erste S-ATA oder SCSI-Festplatte
- `/dev/hda` IDE-Festplatte oder IDE-CD-ROM (Master am ersten IDE-Controller)
- `/dev/sda1` erste primäre Partition der ersten S-ATA oder SCSI-Festplatte
- `/dev/sda5` erstes log. Laufwerk in erweiterter Partition auf der ersten S-ATA oder SCSI-Festplatte
- `/dev/sr0` erstes S-ATA-CD/DVD-ROM (scsi-readable)
- `/dev/sde2` zweite primäre Partition der fünften S-ATA oder SCSI-Festplatte
- `/dev/sdb` zweite S-ATA oder SCSI-Festplatte, evtl. auch ein USB-Stick
- `/dev/tty1` erstes virtuelles Terminal

Verzeichnis (directory)

Kann Dateien und Unterverzeichnisse (subdirectories) enthalten.

Soft-Link

Verweis auf eine andere Datei (mehr dazu auf Seite 38)

Named Pipe

Datenstruktur zur Inter-Prozesskommunikation

Zeichen	Dateityp
d	Directory
l	Symbolische Links
b	Blockorientiertes Gerät
c	Zeichenorientiertes Gerät
p	Named Pipes
s	Sockets

Abb. 2.13: Anzeigekürzel für Dateiarnten

**Übung:**

Öffnen Sie ein Terminal und geben Sie bei nachfolgenden Dateien/Verzeichnissen den jeweiligen Typ an:

1. /etc/acpi _____
2. /etc/aliases _____
3. /etc/X11/X _____
4. /dev/zero _____
5. /dev/sda _____
6. /dev/log _____

2.5.4 Verzeichnisstruktur von Linux-Systemen (FHS)

Linux und UNIX verfügen über eine ähnliche Verzeichnisstruktur und verfügen nicht über Laufwerksbuchstaben. Die Verzeichnisse sind so organisiert, dass sich eine hierarchische Struktur ergibt (Verzeichnisbaum). Die Wurzel dieses umgedrehten Baums wird als *root* bzw. *root directory* oder „/“ bezeichnet.

In jeder Ebene des Baumes können sowohl Dateien als auch Verzeichnisse liegen, letztere bilden die nächste Ebene des Baumes. Name und Position jeder Datei kann durch eine entsprechende Pfadangabe spezifiziert werden, die an der Wurzel beginnt und alle zu durchlaufenden Verzeichnisse auflistet.

Um bei Linux einheitliche Verzeichnisstrukturen zu erhalten, wurde der Filesystems Hierarchy Standard (FHS) geschaffen. Einzelne Linux-Distributionen weichen in Details voneinander ab, die Akzeptanz der FHS-Empfehlungen steigt zusehends.

Das FHS hat u. a. folgende Verzeichnisse, die folgende Bedeutung haben, festgelegt.

/bin	Dienstprogramme und Kommandos, die von allen Benutzern ausgeführt werden können
/sbin	Dienstprogramme und Kommandos zur Systemverwaltung. Sie können nur von root ausgeführt werden
/boot	Kernel und Dateien, die vom Linux-Bootmanager benötigt werden
/dev	Geräte-dateien
/etc	Konfigurationsdateien für Systemdienste und systemweite Einstellungen
/etc/init.d	Skripten und Programme zur Systeminitialisierung
/home	Verzeichnisse der Benutzer
/lib	Systembibliotheken (shared libraries) und Kernel-Module
/lib...	alternative dynamische Bibliotheken, beispielsweise <code>/lib32</code> und <code>/lib64</code> für 32-Bit und 64-Bit
/media	Einhängepunkt für Wechseldatenträger
/mnt	temporärer Einhängpunkt für Dateisysteme
/opt	zusätzliche Anwendungsprogramme
/proc	Prozess-Datei-System, Schnittstelle zum Kernel z.B. zur Verwaltung von Prozessen
/root	Home-Verzeichnis des Systemadministrators <code>root</code>
/srv	Daten für Systemdienste
/tmp	temporäre Dateien
/usr	Anwendungsprogramme, X-Window-System, ... (statische Daten)
/usr/src	Quellen des Linux-Kernels (falls installiert)
/usr/share/doc	Hilfdateien und Dokumentationen zu Anwendungsprogrammen
/var	veränderliche Daten

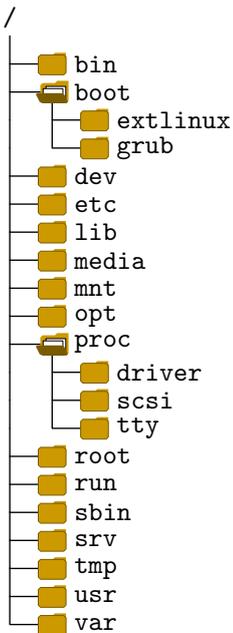


Abb. 2.14: Beispiel einer Linux-Verzeichnisstruktur (unvollständig)

2.5.5 Begriffe zum Dateisystem

- root directory** Die Wurzel des Dateisystems hat kein übergeordnetes Verzeichnis und wird mit „/“ bezeichnet.
- working directory** Momentaner Aufenthaltsort im Verzeichnisbaum (auch current directory). Er kann mit `pwd` (print working directory) angezeigt werden.

```
pwd
/var/local/daten/lpi/target/cp
```

- home directory** Heimatverzeichnis, Platz im Dateisystem, der einem Benutzer zugeordnet ist. Der Pfadname des Home-Directories steht in der Systemvariablen `HOME` bzw. kann über die Abkürzung `~` angesprochen werden.

```
echo $HOME
/home/mgn
echo ~
/home/mgn
```

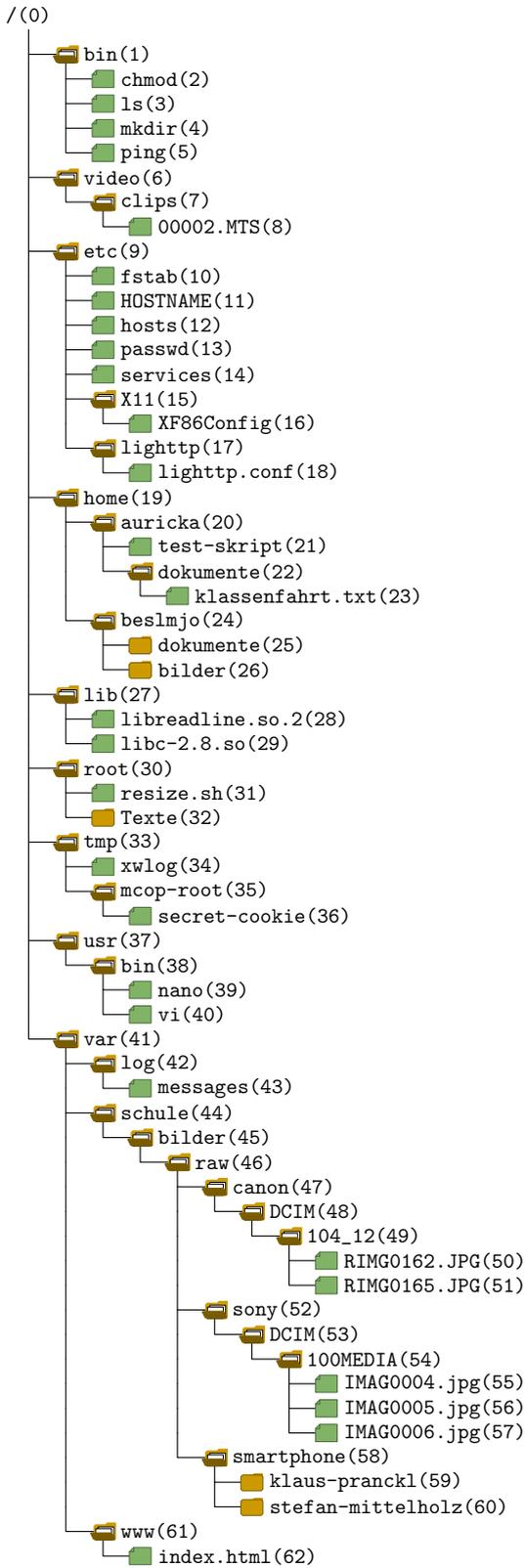
Das Home-Directory stellt die Privatsphäre eines Benutzers dar, es kann jederzeit mit `cd` erreicht werden.

- login directory** Anmeldeverzeichnis, Position im Dateibaum, an der sich der Anwender nach dem Anmelden befindet, im Normalfall wird dies sein Home-Directory sein.
- parent directory** Übergeordnetes Verzeichnis. Das parent directory kann vom aktuellen Verzeichnis aus mit `cd ..` erreicht werden.
- absoluter Pfadname** Vollständiger Pfadname, der ausgehend von „/“ den Weg durch den Dateibaum bis zur gewünschten Datei angibt (z. B.: `/home/herbert/briefe/maria.txt`)
- relativer Pfadname** Die Angaben eines solchen Pfadnamens beginnen im aktuell gesetzten Verzeichnis. Beispiel (für `/home/herbert`): `briefe/maria.txt`



Übung:

Gegeben ist ein Ausschnitt aus einer Linux-Verzeichnisstruktur:



Betrachten Sie auf Seite 34 den Ausschnitt der Verzeichnisstruktur eines Linux-PCs.

Lösen Sie dazu folgende Aufgaben:

Beispiel: Der absolute Pfadname der Datei 43 ist `/var/log/messages`

1. Geben Sie jeweils den absoluten Pfadnamen der Dateien 5, 8 und 56 an.

2. Sie möchten in das Verzeichnis `/home/auricka` wechseln.

- a) Geben Sie den Befehl an, mit dem Sie dorthin gewechselt sind (absolute Pfadangabe).

- b) Mit welchem Befehl geben Sie Ihr aktuell gesetztes Verzeichnis aus?

3. Jetzt ist Ihr aktuell gesetztes Verzeichnis `/home/auricka`

- a) Geben Sie den relativen Pfadnamen von Datei 23 an.

- b) Wechseln Sie in Verzeichnis 33
absolute Angabe

relative Angabe

4. Jetzt ist Ihr aktuell gesetztes Verzeichnis `/tmp`

Rufen Sie von dort aus das Programm 40 auf:

absolute Angabe

relative Angabe

5. Ihr aktuell gesetztes Verzeichnis ist `/var/schule/bilder`

Geben Sie jeweils den relativen Pfadnamen der folgenden Verzeichnisse an: 49, 53 und 58

2.5.6 Arbeiten mit Dateien

Mit Dateien kann viel gemacht werden. Nachfolgend die wichtigsten Kommandos dazu.

Dateien kopieren

Mit dem Kommando `cp` werden Dateien kopiert.

```
cp [Optionen] Quelle Ziel
```

Ist das `Ziel` ein Verzeichnis, so wird die Datei in dieses Verzeichnis kopiert. Der Dateiname bleibt dabei erhalten.

Die wichtigsten Parameter von `cp` sind:

- b ist die Zieldatei schon vorhanden, so wird diese in `Ziel~` umbenannt (die Tilde wird an den Dateinamen angehängt) und dann erst wird die Datei kopiert (backup)
- f wenn eine existierende Zieldatei nicht geöffnet werden kann, wird diese ohne Rückfrage gelöscht und das Kopieren wird erneut versucht (force)
- i fragt nach, ob eine bereits bestehende Dateien überschrieben werden soll (interactive)
- p behält möglichst alle Dateiattribute der Quelldatei bei
- R kopiert Verzeichnisse mit allen Unterverzeichnissen und allen darin enthaltenen Dateien (recursive), es kann auch `-r` verwendet werden.
- u die Quelldatei wird nur kopiert, wenn die Zieldatei älter ist (oder noch nicht existiert) (update)
- v zeigt alle Aktivitäten auf dem Bildschirm (verbose)
- s Die Datei wird nicht kopiert, stattdessen wird ein symbolischer Verweis erzeugt (symbolic).
- l wie `-s`, jedoch wird ein Hard-Link erzeugt
- d Links bleiben erhalten und werden als solche kopiert (preserve)

Beispiele

Kopieren einer Datei:

```
touch meineDatei1
cp meineDatei1 meineDatei2
ls -l meine*
# Dateien wieder löschen
rm -f meineDatei?
```

```
-rw-rw---- 1 mgn privat 0 Jul  8 07:27 meineDatei1
-rw-rw---- 1 mgn privat 0 Jul  8 07:27 meineDatei2
```

Kopieren einer Datei mit Backup-Funktion:

```
touch meineDatei1
cp -b meineDatei1 meineDatei2
ls -l meine*
echo '* mit Backup *'
cp -b meineDatei1 meineDatei2
ls -l meine*
# Dateien wieder löschen
rm -f meineDatei*
```

```
-rw-rw---- 1 mgn privat 0 Jul  8 07:27 meineDatei1
-rw-rw---- 1 mgn privat 0 Jul  8 07:27 meineDatei2
* mit Backup *
-rw-rw---- 1 mgn privat 0 Jul  8 07:27 meineDatei1
-rw-rw---- 1 mgn privat 0 Jul  8 07:27 meineDatei2
-rw-rw---- 1 mgn privat 0 Jul  8 07:27 meineDatei2~
```

Kopieren einer Datei, wenn diese neuer ist als die Zieldatei:

Damit die Datei wirklich ein anderes Datum hat, wird mit `sleep 2` zwei Sekunden gewartet. Damit die Uhrzeit nicht nur in Minuten angezeigt wird, wird mit dem Parameter `--full-time` die Zeit in voller Auflösung angezeigt.

Da die Datei `meineDatei1` älter ist als die Datei `meineDatei2`, bleibt der Aufruf von `cp` ohne Funktion.

```
echo "X" > meineDatei1
sleep 2
echo "XXX" > meineDatei2
cp -u meineDatei1 meineDatei2
ls -l --full-time meine*
# Dateien wieder löschen
rm -f meineDatei*
```

```
-rw-rw---- 1 mgn privat 2 2012-07-08 07:28:13.845619390 +0200 2
meineDatei1
-rw-rw---- 1 mgn privat 4 2012-07-08 07:28:15.845620887 +0200 2
meineDatei2
```

Dateien verschieben bzw. umbenennen

Mit dem Kommando `mv` werden Dateien/Verzeichnisse verschoben bzw. umbenannt.

```
mv [Optionen] Quelle Ziel
```

Die wichtigsten Parameter von `mv` sind:

- b ist die Zieldatei schon vorhanden, so wird diese in `Ziel~` umbenannt (die Tilde wird an den Dateinamen angehängt) und dann erst wird die Datei verschoben. (backup)
- f wenn eine existierende Zieldatei nicht geöffnet werden kann, wird diese gelöscht (ohne Rückfrage) und das Verschieben wird erneut versucht (force)
- i fragt nach, ob eine bereits bestehende Dateien überschrieben werden soll (interactive)
- v zeigt alle Aktivitäten auf dem Bildschirm (verbose)

Beispiel

```
touch meineDatei1
ls -l meine*
echo '* move *'
mv meineDatei1 meineDatei2
ls -l meine*
# Dateien wieder löschen
rm -f meineDatei*
```

```
-rw-rw---- 1 mgn privat 0 Jul  8 07:27 meineDatei1
* move *
-rw-rw---- 1 mgn privat 0 Jul  8 07:27 meineDatei2
```

Dateien löschen

Mit dem Kommando `rm` werden Dateien/Verzeichnisse gelöscht.

```
rm [Optionen] Datei
```

Die wichtigsten Parameter von `rm` sind:

- f alle Dateien, auch schreibgeschützte ohne Nachfrage löschen
- i fragt vor jedem Löschen nach
- R Verzeichnisse und ihre Inhalte rekursiv entfernen (es geht auch `-r`)
- v zeigt alle Aktivitäten auf dem Bildschirm an

Beispiel

```
mkdir meinDir
touch meinDir/meineDatei1
touch meinDir/meineDatei2
touch meinDir/meineDatei3
ls -l meinDir
# Dateien wieder löschen
rm -rf meinDir
```

```
insgesamt 0
-rw-rw---- 1 mgn privat 0 Jul  8 08:00 meineDatei1
-rw-rw---- 1 mgn privat 0 Jul  8 08:00 meineDatei2
-rw-rw---- 1 mgn privat 0 Jul  8 08:00 meineDatei3
```

Dateien verknüpfen – Links

Bei Linux-Dateisystemen können Sie Verweise auf Dateien anlegen und so auf eine Datei mit einem anderen Namen oder aus einem anderen Verzeichnis aus zugreifen. Verweise werden in Linux Links genannt.

```
ln [Optionen] Quelle Ziel
```

Standardmäßig wird der Link über sogenannte Inode-Verweise (Hard-Link) realisiert. Ein Inode (indirection node) ist ein Informationsblock im Dateisystem und enthält fast alle Informationen einer Datei (außer dem Dateinamen). Jeder Datei ist genau ein Inode zugeordnet. Ein Hard-Link kann nur innerhalb eines Dateisystems erstellt werden, da der Verweis über die Inode-Nummer realisiert wird, z. B. 4711 und diese Nummer auch auf einem anderen Dateisystem vorhanden ist (Abhilfe schafft hier ein symbolischer Link).

Ein symbolischer Link ist eine Verknüpfung, die über einen Pfadnamen realisiert wird. Symbolische Links können daher über Partitions Grenzen erstellt werden.

Ein Inode enthält u. a. hinaus:

- die Zugriffsrechte auf die Datei (Eigentümer-, Gruppen- und sonstige Rechte)
- den Eigentümer der Datei und die Hauptgruppe (UID, GID)
- den Dateityp (einfache Datei, Verzeichnis, Link, ...)
- die Größe der Datei (in Bytes)
- einen Referenzzähler (Anzahl der Hard-Links auf die Datei)
- das Datum
 - der letzten Inode-Änderung (change time, ctime),
 - des letzten Zugriffs auf die Datei (letzte Dateiöffnung/-ausführung, access time, atime) und
 - der letzten Änderung der Datei (modification time, mtime)
- einen oder mehrere Verweise auf die Cluster (Blöcke), in denen der Inhalt der Datei liegt.

Die wichtigsten Parameter von `ln` sind:

- `b` ist die Zieldatei schon vorhanden, so wird diese in `Ziel~` umbenannt (die Tilde wird an den Dateinamen angehängt) und dann erst wird der Link erzeugt
- `f` löscht vor dem Anlegen des Links eine evtl. vorhandene Zieldatei
- `i` fragt nach, ob eine bereits vorhandene Datei überschrieben werden soll
- `s` erstellt symbolische Links statt Hard-Links
- `v` zeigt alle Aktivitäten auf dem Bildschirm an (verbose)

Beispiele

In nachfolgendem Beispiel wird eine Datei erzeugt. Dabei hat der Datei-Referenzzähler (dritte Spalte) den Wert 1 (siehe Abbildung 2.12 auf Seite 31). In der ersten Spalte wird die Inodenummer angezeigt. Wird nun der Link angelegt, so wird ein neuer Dateieintrag im Verzeichnis angelegt, der auf die selbe Inode-Nummer verweist und der Datei-Referenzzähler wird um eins erhöht.

```
touch meineDatei1
ls -li meineDatei1
echo '* Link *'
ln meineDatei1 meineDatei2
ls -li meine*
# Dateien wieder löschen
rm -f meineDatei?
```

```
4458175 -rw-rw---- 1 mgn privat 0 Jul  8 07:26 meineDatei1
* Link *
4458175 -rw-rw---- 2 mgn privat 0 Jul  8 07:26 meineDatei1
4458175 -rw-rw---- 2 mgn privat 0 Jul  8 07:26 meineDatei2
```

In nachfolgendem Beispiel wird ein symbolischer Link erzeugt. Dabei wird nur ein Verzeichniseintrag in einer neuen Datei erzeugt, der auf die ursprüngliche Datei verweist.

```
touch meineDatei1
ls -l meineDatei1
echo '* Symbolischer Link *'
ln -s meineDatei1 meineDatei2
ls -l meine*
# Dateien wieder löschen
rm -f meineDatei?
```

```
-rw-rw---- 1 mgn privat 0 Jul  8 07:27 meineDatei1
* Symbolischer Link *
-rw-rw---- 1 mgn privat  0 Jul  8 07:27 meineDatei1
lrwxrwxrwx 1 mgn privat 11 Jul  8 07:27 meineDatei2 -> meineDatei1
```

**Übung:**

1. Florian Brunner möchte die Bilder, die sich auf seinen beiden Smartphones befinden, ebenfalls zur Verfügung stellen. Dazu müssen auf dem Linux-PC weitere Verzeichnisse erstellt werden (siehe auch die Verzeichnisstruktur auf Seite 34) Erstellen Sie dazu in `/var/schule/bilder/raw/smartphone` das Verzeichnis `florian-brunner` und darin die beiden Verzeichnisse `galaxy` und `iphone`.
Arbeiten Sie bei a) und b) mit absoluten Pfadangaben.
 - a) Legen Sie die benötigten Verzeichnisse nacheinander an.

 - b) Löschen Sie die neu angelegten Verzeichnisse wieder.

 - c) Um mit `mkdir` mit einem Befehlsaufruf komplette Unterverzeichnisstrukturen zu erstellen, ist ein Parameter nötig. Finden Sie diesen über die Hilfeseite (`man mkdir`) heraus.

 - d) Wechseln Sie jetzt nach `/var/schule/bilder/raw/smartphone` und legen Sie dort mit einem Befehl die benötigte Verzeichnisstruktur an.

2. Johann Beslmeisel möchte einige der unter `/var/schule/bilder/raw/sony` liegenden, unbearbeiteten Bilder der Sony-Kamera in sein Home-Verzeichnis `/home/beslmjo` kopieren.
Arbeiten Sie bei a), b) und c) mit absoluten Pfadangaben.
 - a) Mit welchem Kommando kopiert Johann die Datei `IMAG0004.jpg` in sein Home-Verzeichnis?

 - b) Geben Sie an, wie er mit einem Befehl die beiden Dateien `IMAG0004.jpg` und `IMAG0005.jpg` in sein Home-Verzeichnis kopiert.

 - c) Geben Sie an, wie Johann mit einem Befehl alle dort vorhandenen Bilder in sein Home-Verzeichnis kopieren könnte.

3. Um ein Bildkonvertierungsskript zu testen, muss die sich in `/var/schule/bilder/raw/canon` befindliche Unterverzeichnisstruktur der Canon-Kamera nach `/tmp/canon` kopiert werden.
 - a) Geben Sie den dazu benötigten Befehl an.

 - b) Nach dem Konvertierungsvorgang kann das Verzeichnis `/tmp/canon` komplett ohne Rückfrage gelöscht werden. Wie lautet das vollständige Kommando?

4. Frau Lorenz verlangt, dass die Verzeichnisstruktur des Linux-PCs einheitlicher wird. Mit welchem Kommando wird sie das komplette Verzeichnis `/video` nach `/var/schule` verschieben?

5. Probieren Sie das Beispiel zu den Links aus (Seite 38). Welche Inode-Nummer wird bei Ihnen verwendet?

6. Woran erkennen Sie, dass auf eine Datei ein Hard-Link besteht?

2.6 Texteditoren

Unter Linux gibt es eine Menge an Editoren für die unterschiedlichsten Bereiche. Einige sind nur reine Texteditoren, andere sind so erweitert, dass diese schon komplette Entwicklungsumgebungen, z. B. für die HTML-Seitenerstellung oder Java-Programmierung, darstellen.

Ein sehr einfacher zu bedienender Texteditor ist Nano. Ein weiterer Editor ist vi, der sehr viele Möglichkeiten bietet, aber dafür nicht mehr so einfach zu bedienen ist.

2.6.1 Nano

Nano ist ein Texteditor für die Kommandozeile und der Nachfolger von Pico. Im Gegensatz zu seinen „großen“ Verwandten vi und den Emacs-Editoren ist Nano nicht so mächtig, dafür aber einfach zu bedienen, was ihm vor allem bei Linux-Einsteigern zu einer großen Beliebtheit verholfen hat. Zusätzlich unterstützt er für viele Textformate Syntaxhighlighting.



```

GNU nano 2.2.6          Datei: nano-uebung.sh
#!/bin/bash
$output="Ausgabe"
$outputput="Taube"

echo $output
echo $outputput

[ 7 Zeilen gelesen ]
^G Hilfe          ^O Speichern     ^R Datei öffnen  ^Y Seite zurück  ^K Ausschneiden  ^C Cursor
^X Beenden       ^J Ausrichten   ^W Wo ist        ^V Seite vor     ^U Ausschn. rückg ^T Rechtschr.
  
```

Abb. 2.15: Nano mit Syntax Highlighting

Aufgerufen wird Nano durch die Eingabe von nano. Möchte man hingegen eine bestehende Datei bearbeiten, so gibt man beim Aufruf den Dateinamen an.

Die wichtigsten Tastenbefehle von Nano sind am unteren Rand des Terminals immer eingeblendet, wobei das Symbol ^ für die **[Strg]**-Taste steht. Eine vollständige Funktionsübersicht erhält man über die Hilfe mit **[Strg]+[g]**.



Übung:

1. Öffnen Sie Nano, geben Sie die nachfolgenden Zeilen ein, speichern Sie das Geschriebene in einer Datei mit dem Namen `foo.sh` und öffnen Sie die Datei in Nano wieder.

```

#!/bin/bash
$output="Ausgabe"
$outputput="Taube"
echo $output
echo $outputput
  
```

Was fällt Ihnen vor der Eingabe und nach dem erneuten Laden auf?

2. Nun ändern Sie die Reihenfolge der „Echo-Zeilen“, indem Sie sie eine davon ausschneiden und an anderer Stelle wieder einfügen.

2.6.2 Vi

vi ist das Urgestein der Text-Editoren für die Textkonsole und stammt aus den 1970er. Inzwischen ist er mehrfach erweitert und überarbeitet worden. Er ist auf fast allen Unix-/Linux-System installiert (z. B. als vim-Vi **IM**proved). Meist verwenden nur Systemadministratoren den vi, für Anwender ist Nano die richtige Wahl.

vi besitzt drei grundsätzlich unterschiedliche Arbeitsmodi.

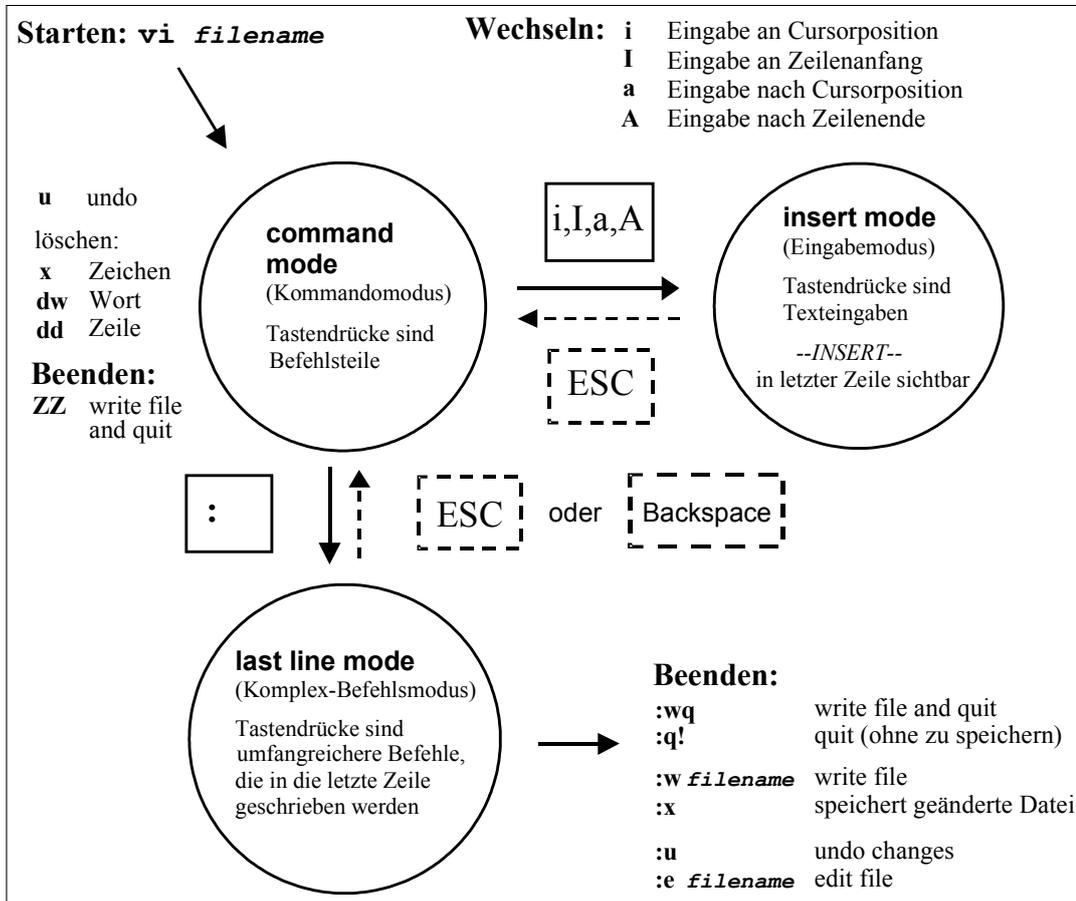


Abb. 2.16: Arbeitsmodi des vi

Die drei Modi sind:

1. **Kommandomodus** (command mode)

Beim Start von vi befindet man sich im Kommandomodus. Dort können durch verschiedene Tastendrücke einfache Befehle ausgeführt werden, wie z. B. „Wort suchen“, „Zeile löschen“ usw. Durch Drücken von **[Esc]** gelangt man in diesen Befehlsmodus.

Syntax: [Wiederholungsfaktor *n*]KommandoTexteinheit

Beispiele:

4dw löscht (delete) die nächsten 4 Wörter
 5dd löscht die nächsten 5 Zeilen
 y aktuelle Zeile in die Zwischenablage kopieren (yank)
 5y 5 Zeilen in die Zwischenablage kopieren
 p Zeile(n) aus der Zwischenablage einfügen (paste)
 G springt an den Beginn der letzten Zeile der Datei

[Strg]+[r] macht die durch u rückgängig gemachten Änderungen rückgängig (redo)

2. **Eingabemodus** (insert mode)

Im Einfügemodus ist die eigentliche Eingabe von Text möglich. Durch Befehle wie „i“, „a“ oder „o“ gelangt man aus dem Befehlsmodus in diesen Modus und mit **[Esc]** wieder zurück.

3. **Komplexbefehlsmodus** (last line mode)

Durch Eingabe von ':' (Doppelpunkt) gelangt man vom Befehlsmodus in den Komplexbefehlsmodus und mit `[Esc]` wieder zurück.

Syntax: : [Adresse]Kommando [Optionen]

Beispiele:

- :q! beendet den vi ohne zu speichern
- :/suchbegriff sucht im Text. Wenn etwas gefunden wurde, drückt man n für das nächste, N für das vorherige Vorkommen des Suchmusters
- :set number aktiviert die Zeilenanzeige in der aktuellen vi-Session
- :s/ö/oe/g ersetzt in der aktuellen Zeile alle „ö“ der durch „oe“
- :1, \$/\//\./g ersetzt alle „/“ in der gesamten Datei durch „.“ (Quoting nötig!)
- :5, 9s/Maier/Mair/g ersetzt in den Zeilen 5-9 jeweils alle Vorkommen von Maier durch Mair
- :50, \$s/[*#]/=/ ersetzt ab Zeile 50 bis zum Dokumentenende jeweils das erste „*“ oder „#“ durch „=“

Fragen und Übungen (nur) für angehende Administratoren

1. In welchem Modus befinden Sie sich nach dem Aufrufen des vi?

2. In welchen Modi können sie den vi wieder ordnungsgemäß verlassen?

3. Wie kommen sie in den insert mode
 - a) vom command mode bzw.

 - b) vom last line mode?

4. In welchem Modus können Sie den Cursor bewegen?

5. In welchem Modus können sie Texte löschen?

6. Speichern Sie im vi eine Datei unter einem anderen Namen. Wie gehen sie vor?

7. Durch Eingabe von /Suchbegriff im command mode können Sie im vi nach Begriffen suchen. Suchen Sie in der Konfigurationsdatei /etc/services nach dem Begriff lansource.

Geben Sie nachfolgend die erste Zeile an, in der dieser Begriff vorkommt.

8. Editieren Sie die Datei /etc/inittab und sorgen Sie dafür, dass die Tastenkombination `[Strg]+[Alt]+[Entf]` zum Herunterfahren des Systems nicht mehr funktioniert. Hinweis: Suchen Sie nach „ctrlaltdel“ und Kommentieren sie diese Zeile mit „#“ am Zeilenanfang aus.

2.7 Icon auf dem Desktop erstellen

Man kann ohne Probleme ein Terminal-Icon auf dem Desktop platzieren, welches beliebige Programme startet. Das Verzeichnis `~/Arbeitsfläche/` in Home-Verzeichnisses des Anwenders enthält pro Desktop-Icon eine Steuerdateien, die die Dateiendung `.desktop` hat und eine normale Textdateien ist.

```
#!/usr/bin/env xdg-open
[Desktop Entry]
Type=Application
Exec=gnome-terminal
Name=term
Icon=/usr/share/pixmaps/gnome-terminal.xpm
```

Ist ein Icon erstellt, so kann mit einem -Klick ein Kontext-Menü geöffnet werden. Über *EIGENSCHAFTEN* wird ein entsprechender Editor geöffnet werden, mit dem die Einstellungen angepasst werden können (siehe Abbildung 2.17).

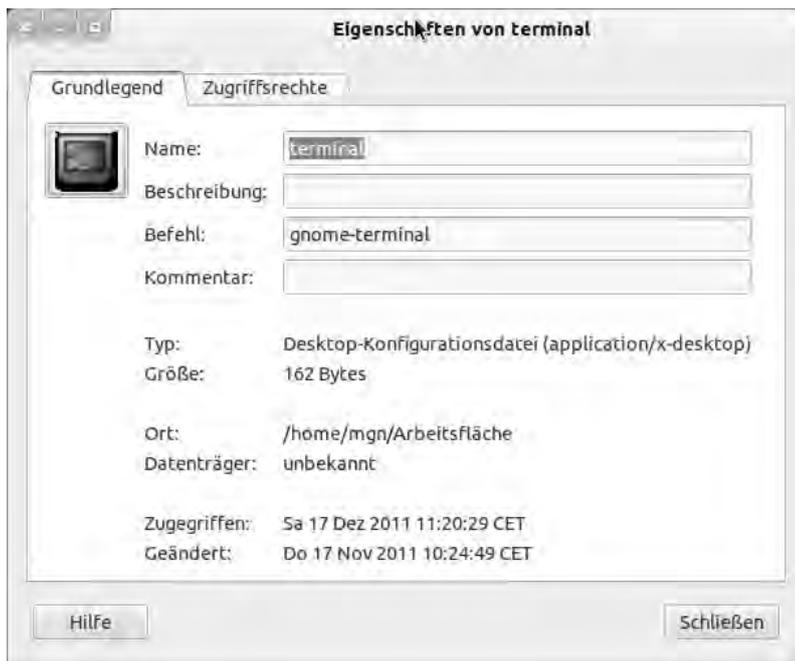


Abb. 2.17: Desktop-Icon anpassen

Übung:

Notieren Sie jeweils die entsprechenden Befehle:

1. Öffnen Sie ein Terminal und wechseln Sie in das Verzeichnis `~/Arbeitsfläche/`.

2. Erstellen Sie mit dem Editor `nano` die Datei `terminal.desktop`.

3. Damit das Programm gestartet werden kann, muss das Recht 'ausführen' gesetzt werden (mehr dazu auf Seite 58)


```
chmod +x terminal.desktop
```
4. Probieren Sie das Icon aus und starten Sie damit ein Terminal.
5. Falls Ihnen das Icon nicht gefällt, wechseln Sie dieses über -Klick – *EIGENSCHAFTEN* aus.



2.8 Fragen

1. Was versteht man unter alpha, beta und stable-Version?

2. Was bedeutet es, wenn bei einer Distribution LTS angegeben ist?

3. Aus welcher Distribution sind folgende Systeme entstanden?

a) Ubuntu _____

b) CentOS _____

4. Wo wird überall Linux eingesetzt? Nennen Sie dazu jeweils ein Beispiel!

5. Was ist eine Konsole, ein Terminal?

6. Wie installiert man das Programm „synaptic“ auf der Konsole?

7. Was ist eine Shell?

8. Geben Sie zwei Möglichkeiten an, um Hilfen zu Linux-Kommandos zu erhalten.

9. Worin unterscheiden sich Optionen und Argumente bei Kommandos?

10. Nennen Sie zwei Namenskonventionen bei einem Linux-Dateisystem.

11. Was versteht man unter Regulären Ausdrücken. Warum wurden sie erfunden?

12. Was versteht man unter Quoting?

13. Wie gibt man alle Dateinamen eines Verzeichnisses an, die aus genau 10 Zeichen bestehen?

14. Wie gibt man alle Dateinamen eines Verzeichnisses an, die nicht mit einem Großbuchstaben beginnen?

15. Erläutern Sie die fünf Dateiartern bei Linux.

16. Was versteht man unter FHS?

17. Welche besondere Eigenschaft hinsichtlich eines übergeordneten Verzeichnisses hat das `root` directory „/“ ?

18. Geben Sie an, welche Dateien sich bei Linux-Systemen in `/etc`, `/bin`, `/home`, `/root` befinden.

19. Worin unterscheiden sich die Verzeichnisstrukturen von Linux- und Windows-Systemen?

20. Was haben Verzeichnisstrukturen mit Bäumen zu tun?

21. Wie unterscheidet sich eine relative von einer absoluten Pfadangabe?

22. Welche Bedeutung haben die Parameter `-n`, `-e` vom Kommando `echo`?

23. Was bewirken folgende Sequenzen beim Kommando `echo` bzw. was stellen diese dar?

`\\` _____

`\f` _____

`\n` _____

`\r` _____

`\t` _____

`\v` _____

3 Einheit II – Arbeiten mit Benutzern und Berechtigungen

Eine wichtige Aufgabe der Systemadministration ist die Benutzerverwaltung: Neue Benutzerkonten müssen angelegt, alte gelöscht werden, Benutzer müssen einzelnen Gruppen zugeordnet oder ihre Gruppenmitgliedschaft soll verändert werden. Dazu kommt die Verwaltung der Dateiberechtigungen und des Dateieigentums.

Der folgende Abschnitt ist in zwei Abschnitte aufgeteilt. Im ersten Teil wird genau dargestellt, wie die Benutzer- und Gruppenverwaltung mit distributionsunabhängigen Linux-Kommandos durchgeführt werden kann. Dabei werden auch die Formate der Datenbanken für die Benutzer und deren Passwörter sowie die der Gruppenverwaltung erläutert. Nach einem Fragen- und Übungsteil zur Benutzerverwaltung folgt der zweite Teil, in dem erklärt wird, welche Dateiberechtigungen es bei Linux-Systemen gibt und wie diese verändert werden können. Auch dieser Abschnitt endet mit einem Fragen- und Übungsteil zu Dateiberechtigungen und Dateieigentum.

Inhalt

3.1	Die Benutzerverwaltung unter Linux.....	48
3.1.1	Systemverwalter, Standardbenutzer und Systembenutzer.....	48
3.1.2	Primär-Gruppen und Sekundär-Gruppen.....	48
3.1.3	Benutzerkonten anlegen.....	48
3.1.4	Als anderer Benutzer arbeiten - su und sudo.....	51
3.1.5	Gruppenverwaltung.....	52
3.1.6	Fragen.....	56
3.2	Dateiberechtigungen und Dateieigentum.....	58
3.2.1	Benutzerklassen und Zugriffsrechte.....	58
3.2.2	Zugriffsrechte und Eigentümerschaft ändern.....	59
3.2.3	Fragen.....	65

3.1 Die Benutzerverwaltung unter Linux

Eine der grundlegenden Eigenschaften von Linux ist es, dass mehrere Benutzer gleichzeitig und unabhängig voneinander an einem Linux-System arbeiten können. Dies erfordert jedoch eine Benutzer- und Zugriffsverwaltung, damit sich die Benutzer nicht gegenseitig beeinträchtigen.

Die Benutzerverwaltung legt fest, wer auf welche Dateien zugreifen darf, unter welchen Umständen ein Benutzer Daten eines anderen Benutzers lesen oder verändern darf, wer welche Programme ausführen kann und wer auf welche Hardware-Komponenten (über die Gerätedateien) zugreifen darf.

Jeder Benutzer verfügt über ein eigenes Benutzerkonto und ein persönliches Home-Verzeichnis, in dem seine eigenen Dateien und Konfigurationen gespeichert werden. Mit seinem Benutzernamen und einem persönlichen Passwort kann sich ein Benutzer am System anmelden. Alle Aktionen, die ein Benutzer während einer Session (d. h. zwischen Anmelden und Abmelden) durchführt, werden ihm vom System zugeordnet.

3.1.1 Systemverwalter, Standardbenutzer und Systembenutzer

Bei allen Linux-Distributionen werden meist bereits während der Installation zwei Benutzerkonten eingerichtet: das Administrator-Konto `root` und ein Konto für einen normalen Benutzer (Standardbenutzer).

Der Systemverwalter `root` besitzt alle Rechte im System. Er darf alle Dateien ansehen, verändern, löschen, alle Programme ausführen usw.

Mit Administrationsrechten kann ein System allerdings versehentlich zerstört werden. Deshalb müssen von `root` weitere Benutzer eingerichtet werden, die über eingeschränkte Rechte verfügen. Diese Standardbenutzer verwenden Linux, um damit zu arbeiten und haben uneingeschränkten Zugriff auf ihre eigenen Dateien, aber nur eingeschränkten Zugriff auf den Rest des Systems.

Für besondere Programme (z. B. Mail- und Webserver) oder für den Zugriff auf bestimmte Komponenten und Geräte (z. B. Drucker, Band- und Festplattenlaufwerke) existieren spezielle Benutzerkonten. Die Konten dieser sog. Systembenutzer (Pseudobnutzer) sind nicht für die interaktive Arbeit am Rechner vorgesehen.

3.1.2 Primär-Gruppen und Sekundär-Gruppen

Möchten mehrere Benutzer Dateien oder bestimmte Systemressourcen gemeinsam nutzen, kann dies über eine Gruppenmitgliedschaft realisiert werden.

Linux verlangt, dass jeder Benutzer Mitglied in mindestens einer Gruppe ist. Diese Gruppe wird Primär-Gruppe (Login-Gruppe, Hauptgruppe, Initial Group) genannt. Die Mitgliedschaft in der Primär-Gruppe wird in der Benutzerdatenbank `/etc/passwd` eingetragen.

Benutzer können aber auch feste Mitglieder zusätzlicher Gruppen sein, wenn sie von `root` in der Gruppendatenbank `/etc/group` in den entsprechenden Mitgliederlisten der einzelnen Gruppen eingetragen werden. Diese zusätzlichen Gruppen werden Sekundär-Gruppen (Supplementary Groups) genannt.

Nach der Anmeldung ist ein Benutzer zunächst seiner Primär-Gruppe zugeordnet. Er kann jedoch während einer Session jederzeit mit dem Kommando `newgrp` in die weiteren ihm zugeordneten Gruppen wechseln.

Bei Gruppen, die durch ein Passwort geschützt sind, können Benutzer auch ohne einen festen Eintrag in der Mitgliederliste vorübergehend die Gruppenmitgliedschaft erlangen, indem Sie sich mit `newgrp` unter Angabe des korrekten Gruppen-Passwortes in der Gruppe „einloggen“.

3.1.3 Benutzerkonten anlegen

Das Einrichten neuer Benutzerkonten ist im Prinzip immer gleich und erfolgt in mehreren Schritten:

- Einträge in der Benutzerdatenbank `/etc/passwd` und in der Passwortdatenbank `/etc/shadow` anlegen
- gegebenenfalls ein oder mehrere Einträge in der Gruppendatenbank `/etc/group` erstellen
- das Home-Verzeichnis anlegen und eine Grundausstattung an Dateien hineinkopieren und alles dem neuen Benutzer übereignen
- möglicherweise muss der neue Benutzer noch in weitere Listen eingetragen werden (z. B. Plattenkontingente (Quota), Zugriffsberechtigungen auf spezielle Applikationen oder Datenbanken usw.)

Zur Administration der Benutzer verfügt jede Linux-Distribution über ein eigenes Tool, mit dem weitere Benutzer bzw. Gruppen angelegt und verwaltet werden können (z. B.: SUSE: `yast`, KDE: `KDE-User-Manager`, Red Hat/Fedora: `system-config-users` ...).

Distributionsunabhängig und auf der Kommandozeile werden folgende Kommandos zur Benutzerverwaltung benötigt:

```
useradd neue Benutzer-Konten anlegen
usermod bearbeiten bereits bestehender Benutzer-Konten
userdel löschen von Benutzer-Konten
```

Wird `useradd` mit dem Argument `--help` aufgerufen, werden die verfügbaren Kommando-Optionen angezeigt:

```
useradd --help
usage:
useradd [-u uid [-o]] [-g group,..] [-G group]
        [-d home] [-s shell] [-c comment]
        [-m [-k template]] [-f inactive]
        [-e expire ] [-p passwd] name
```

Die wichtigsten Optionen für `useradd` sind:

<code>name</code>	Anmeldename, Kennung des neu anzulegenden Benutzers
<code>-c <Kommentar></code>	Kommentar
<code>-d <Verzeichnis></code>	Home-Verzeichnis, ohne diese Angabe wird <code>/home/Benutzername</code> angenommen
<code>-e</code>	Datum, an dem der Zugang automatisch gesperrt wird
<code>-g <Gruppe></code>	Primär-Gruppe des neuen Benutzers, als Name oder GID, die Gruppe muss bereits existieren
<code>-G <Gruppe1> [, Gruppe2]...</code>	weitere Gruppen (Sekundär-Gruppen), als Namen oder GIDs, die Gruppen müssen bereits existieren, mehrere Gruppen mit Kommas trennen
<code>-m</code>	legt das Home-Verzeichnis an und kopiert die Datei-Grundausrüstung (Vorlagen) aus <code>/etc/skel/</code> hinein
<code>-k <Verzeichnis></code>	ein anderes Verzeichnis für die Datei-Grundausrüstung angeben
<code>-s <Shell></code>	die Login-Shell des Benutzers
<code>-u <UID></code>	numerische Benutzerkennung des neuen Benutzers, die UID darf noch nicht anderweitig vergeben sein, es sei denn, die Option <code>-o</code> wurde angegeben

Mit dem Kommando

```
sudo useradd -c "Bittl,Erwin,IT11FA" -m -d /home/bittler -g it11fa -k /etc/skel.schueler bittler
```

wird von `root` für den Benutzer Erwin Bittl ein Benutzerkonto mit dem Anmeldennamen `bittler` angelegt und der Gruppe `it11fa` zugeordnet. Sein Home-Verzeichnis wird als `/home/bittler` angelegt und es werden die Dateien der Grundausrüstung aus `/etc/skel.schueler` dort hineinkopiert.

Benutzer-ID (UID) und Gruppen-ID (GID)

Für die systeminterne Identifizierung der Benutzer verwendet Linux nicht deren Namen, sondern eine eindeutige Identifikationsnummer, die sogenannte User-ID (UID, engl. user: der Benutzer). Die UID wird im Normalfall beim Anlegen des Benutzerkontos automatisch generiert. Neben der UID ist jedem Benutzer eine weitere Nummer zugeordnet, die sogenannte Gruppen-ID (GID), dabei handelt es sich um die Nummer seiner Primär-Gruppe.

Die UID und die GID von `root` sind immer 0! Dabei ist nicht der Name `root` entscheidend, sondern die UID von 0. Jeder Anwender, der die UID 0 hat, hat somit immer `root`-Rechte.

Die UID, die gerade aktive Primär-Gruppe und alle eingetragenen Sekundärgruppen und die jeweils dazugehörigen GIDs können mit dem Kommando `id` ausgegeben werden, dabei werden der Benutzername und die dem Benutzer zugeordneten Gruppennamen in Klammern angezeigt:

```
id
uid=524(bittler) gid=1010(it11fa) Gruppen=1010(it11fa),16(dialout),33(video)

id root
uid=0(root) gid=0(root) Gruppen=0(root)
```

Intern wird von Linux nahezu ausschließlich die UID bzw. GID verwendet. Selbst wenn bei Kommandos Benutzernamen und/oder Gruppennamen als Argumente anzugeben sind, findet zunächst eine Zuordnung der Namen auf die UID bzw. GID statt, bevor die gewünschte Aktion ausgeführt wird. Diese Zuordnung erfolgt durch die Auswertung der entsprechenden Einträge in der Benutzerdatenbank `/etc/passwd` und in der Gruppendatenbank `/etc/group`.

Die Benutzerdatenbank /etc/passwd

Die Datei /etc/passwd enthält die Liste aller Benutzer. Für jeden Benutzer existiert ein Eintrag, in dem mindestens der Anmelde-name, sein Passwort, die UID- und GID-Nummern, das Home-Verzeichnis und die Login-Shell eingetragen sind. In einem Kommentarfeld können zusätzliche Informationen, z. B. der vollständige Name des Benutzers angegeben werden.

Ein Eintrag in der /etc/passwd besteht aus 7 Feldern und hat folgendes Format:

Anmelde-name:Passwort:UID:GID:Kommentar:Home-Verzeichnis:Login-Shell

Das Kommentar-Feld wird manchmal auch als GECOS-Feld bezeichnet.

Da der Doppelpunkt als Feldtrenner dient, darf er in den einzelnen Feldern, z. B. im Kommentarfeld nicht vorkommen!

Die folgende (mit ...) gekürzte Befehlsausgabe zeigt einige Zeilen der /etc/passwd:

```
cat /etc/passwd
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
wwwrun:x:30:8:WWW daemon apache:/var/lib/wwwrun:/bin/false
...
bittler:x:1524:1010:Bittl,Erwin,IT11FA:/home/bittler:/bin/bash
brunnefl:x:1528:1010:Brunner,Florian,IT11FA:/home/brunnefl:/bin/bash
summerma:x:1630:1020:Summer,Maximilian,IT11SE:/home/summerma:/bin/bash
janist:x:1768:1030:Jani,Steffi,IT11FS:/home/janist:/bin/bash
greinela:x:1001:1000:Greiner,Lars,Lehrer:/home/greinela:/bin/bash
lorenzvi:x:1002:1000:Lorenz,Viola,Lehrer:/home/lorenzvi:/bin/bash
...
```

Früher wurden in /etc/passwd auch die verschlüsselten Passwörter gespeichert. Aus Sicherheitsgründen ist das heutzutage unüblich. Anstatt des Passworts enthält dann das Passwort-Feld das Zeichen x. Das weist darauf hin, dass die verschlüsselten Passwörter in der separaten Passwortdatenbank /etc/shadow gespeichert sind.

Passwörter und die Passwortdatenbank /etc/shadow

Für Linux-Passwörter sollten nur ASCII-Zeichen verwendet werden und aus Sicherheitsgründen sollten die Passwörter sowohl Groß- als auch Kleinbuchstaben sowie mindestens eine Ziffer enthalten.

Passwörter werden unter Linux verschlüsselt gespeichert. Dabei werden Einweg-Verschlüsselungsverfahren (Hash-Verfahren) angewendet. Hash-Verfahren haben den Vorteil, dass das Passwort nicht direkt entschlüsselt werden kann. Die Entschlüsselung mit Passwort-Crackern ist nur durch das zeitaufwändige Ausprobieren aller möglichen Kombinationen möglich (Brute-Force).

Da jedoch viele Benutzer oft zu schwache Passwörter verwenden, führt diese Angriffs-Methode leider häufig schnell zum Ziel.

Um Angriffe zu erschweren, werden die verschlüsselten Passwörter nicht direkt in /etc/passwd gespeichert, sondern getrennt davon in der Datei /etc/shadow. Diese Datei kann nur von root gelesen werden und ist somit für die anderen Benutzer nicht sichtbar.

Ein Eintrag in der /etc/shadow besteht aus 7 Feldern und hat folgendes Format:

Anmelde-name:Passwort:Änderung:MinD:MaxD:Warnung:Frist:Sperre:reserviert

Anmelde-name des Benutzers, der auch in der /etc/passwd-Datei in der ersten Spalte steht

Passwort verschlüsseltes Passwort des Benutzers, wenn dieses Feld leer ist, kann man sich unter dieser Kennung ohne Passwort anmelden wenn hier ein * oder ! steht, ist eine Anmeldung unter dieser Kennung nicht möglich

Änderung der Tag, an dem das Passwort zuletzt geändert wurde. Angegeben werden die Tage, die seit dem 1.1.1970 vergangen sind

MinD Minimale Anzahl der Tage, die das Passwort gültig ist, ist das Feld leer oder 0 so bedeutet das, dass es keine minimale Anzahl gibt

MaxD Maximale Anzahl der Tage, die das Passwort gültig ist, nach Ablauf dieser Frist muss der Benutzer sein Kennwort ändern.

Warnung Anzahl der Tage vor Ablauf der MaxD, an denen der Benutzer gewarnt wird, dass er sein Passwort demnächst ändern muss, weil die MaxD abläuft. Meist erscheint eine entsprechende Meldung bei der Anmeldung.

Frist Hier wird festgelegt, wie viele Tage das Passwort trotz Ablauf der MaxD noch gültig ist. Innerhalb dieser Frist kann der Benutzer sich anmelden, muss aber sofort sein Kennwort ändern.

Sperre Datum, ab dem das Benutzerkonto gesperrt wird (in Tagen seit 1.1.1970)
 reserviert dieses Feld hat momentan keine Bedeutung und ist für zukünftige Verwendung reserviert

Die folgende gekürzte Befehlsausgabe zeigt beispielhaft einige Zeilen einer `/etc/shadow`-Datei:

```
sudo cat /etc/shadow
root:$1$8jp/RdHb$D1x/6Xr2.puE0NX3nIgdX/:14617:0:99999:7:::
daemon*:13991:0:99999:7:::
bin*:13991:0:99999:7:::
sys*:13991:0:99999:7:::
...
bittler:$1$JDE7PpXQ$8bV7aOArT3P91NHFaI7vpg:15420:0:99999:7:::
brunnefl:$1$yktwzYKG$XmcUCqVtAO99GNGMnPJZ5p:15422:0:99999:7:::
summerma!:15591:0:99999:7:::
janist:$1$U3N08ahK$B0x7LHNRbsVuUnKHjzF4as:15425:0:99999:7:::
greinela:$1$CwunXsvI$rruro5WzWesKV6kXSkPtDi3:15317:0:99999:7:::
lorenzvi:$1$NqaUVjcl$jU5D342fouUdm7xu8051LA:15359:0:99999:7:::
...
```

Wenn ein Benutzer sein Passwort ändern möchte, gibt er das Kommando `passwd` ein. Dann wird er nach seinem alten Passwort gefragt und gibt dann zweimal hintereinander sein neues Passwort ein.

Normale Benutzer dürfen nur ihr eigenes Passwort ändern. Der Systemadministrator `root` darf alle Passwörter ändern, dabei wird er nicht nach dem alten Passwort gefragt:

```
root@linux ~# passwd bittler
New password: *****
Re-enter new password: *****
Password changed.
```

3.1.4 Als anderer Benutzer arbeiten - su und sudo

Ein Systemverwalter sollte nicht dauernd unter der `root`-Kennung arbeiten, da er versehentlich wichtige Systemdateien oder -Programme verändern könnte und so das System beschädigt würde. Das Arbeiten mit `root`-Rechten ist auch hinsichtlich der Internetnutzung kritisch, da möglicherweise unbemerkt Schadsoftware heruntergeladen und ausgeführt wird. Deshalb wird empfohlen, dass auch ein Systemverwalter normalerweise als nicht privilegierter Standardbenutzer arbeitet und nur wenn nötig administrative Rechte erhalten soll.

Werden temporär höhere Rechte benötigt, gibt es mehrere Möglichkeiten:

- sich als `root` anmelden,
- in der aktuellen Session mit dem Kommando `su` (switch user) die Identität wechseln oder
- das privilegierte Kommando mit `sudo` (switch user do) starten.

Mit `su` wird die Identität komplett gewechselt. Oft genügt es jedoch, nur für ein bestimmtes Kommando Administrationsrechte zu erhalten. Dies kann mit dem `sudo` Kommando erreicht werden. Manche Distributionen lassen eine Anmeldung als `root` nicht zu (z. B. Ubuntu). In diesem Fall müssen administrative Kommandos mit `sudo` ausgeführt werden.

Beispiel zu `su`:

```
lorenzvi@linux ~$ su
Passwort: *****
root@linux:/home/lorenzvi# su - bittler
bittler@linux: ~$ pwd
/home/bittler
```

Die Benutzerin `lorenzvi` hat zunächst die Identität von `root` angenommen. Wird bei `su` kein Benutzername angegeben, so wird das Passwort von `root` abgefragt und bei erfolgreicher Eingabe die Identität von `root` angenommen. Als `root` kann man ohne Passwort jede beliebige Identität annehmen.

Wird `su` mit der Option `-` aufgerufen, wird unter anderem das aktuelle Arbeitsverzeichnis auf das Home-Verzeichnis des neuen Benutzers gesetzt.

Das Programm `sudo` öffnet im Gegensatz zu `su` keine Shell mit der Identität des neuen Benutzers, sondern startet das angegebene Kommando sofort mit den entsprechenden Rechten. Ist über die Option `-u` kein Benutzer angegeben, wird `root` als neue Identität genommen:

```
sudo useradd -m janist
password for lorenzvi: *****
```

Normalerweise muss für `sudo` bei der ersten Verwendung ein Passwort eingegeben werden. Bei Ubuntu ist das das Passwort des Benutzers, der `sudo` aufruft. Ob und wie ein Standardbenutzer `sudo` verwenden kann, wird in der Datei `/etc/sudoers` konfiguriert.

3.1.5 Gruppenverwaltung

Um mehreren Benutzern den gemeinsamen Zugriff auf Ressourcen (z. B. Dateien, Applikationen, Datenbanken, usw.) zu erlauben, werden Gruppen benötigt. Jeder Linux-Benutzer kann Mitglied mehrerer Gruppen sein. Seiner Primär-Gruppe ist er bereits in der `/etc/passwd` zugeordnet.

Die Zuordnung zu beliebig vielen weiteren Sekundär-Gruppen wird über die Gruppendatenbank `/etc/group` erreicht. Sie enthält alle auf dem System verfügbaren Gruppen und die Zuordnung, welcher Benutzer in welcher Gruppe Mitglied ist. Diese Zuordnung wird über eine Benutzerliste vorgenommen. Dort stehen alle Benutzer, die zusätzliche Mitglieder dieser Gruppe sind.

Die Mitgliedschaften in Primär-Gruppen werden in der Gruppenliste dieser Gruppe nicht noch einmal aufgeführt!

Einträge in der Gruppendatenbank `/etc/group` haben folgendes Format:

Gruppenname:Passwort:GID:Benutzerliste

Die folgende (mit ...) gekürzte Befehlsausgabe zeigt einige Zeilen der Gruppendefinitionen in `/etc/group`:

```
cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
...
dialout:x:16:greinela,lorenzvi
...
admin:x:114:greinela,lorenzvi
...
it11fa:x:1010:
it11se:x:1020:
it11fs:x:1030:
lehrer:x:1000:
photo:x:2000:summerma,janist,brunnefl
...
```

Beim Benutzer `greinela` lautet die Primär-Gruppe `lehrer` (GID=1000). Da er dieser Gruppe bereits in der `/etc/passwd` zugeordnet wurde, ist sein Benutzername nicht noch einmal in der Benutzerliste in der `/etc/group` eingetragen. Zusätzlich ist `greinela` Mitglied in der Gruppe `dialout`, die es ihm ermöglicht, die Internet-Verbindung aufzubauen. Er ist auch der Gruppe `admin` zugeordnet. Auf Ubuntu-Systemen können Mitglieder dieser Gruppe Administrationsarbeiten durchführen, z. B. Programme installieren oder die Systemkonfigurationen ändern.

Strategien bei der Gruppenzuordnung

Um Benutzer ihrer Primär-Gruppe zuzuordnen, sind zwei Vorgehensweisen gebräuchlich:

Die einfachste Möglichkeit ist es, alle normalen Benutzer der primären Gruppe `users` zuzuordnen. Dies wird seit vielen Jahren bei SUSE-Linux umgesetzt. Ein Nachteil besteht darin, dass dann alle Benutzer Zugriff auf die Dateien anderer Benutzer haben.

Bei Red Hat- und Debian-basierten Linux-Distributionen wird jeder Benutzer einer eigenen Primär-Gruppe zugeordnet. Diese Primär-Gruppe trägt den selben Namen wie der Anmeldenamen des Benutzers. Die Gruppe `users` wird dadurch unnötig. Dieses Verfahren hat bei einer entsprechenden Systemadministration Vorteile, etwa dann, wenn mehrere Mitglieder einer sekundären Gruppe gemeinsame Dateien erzeugen.

Bei den GID-Nummern ist 0 für `root` vorgesehen, 1 bis 999 für die Gruppen der Systemdienste. GID=100 ist normalerweise für die Gruppe `users` reserviert. GIDs größer 1000 dürfen für eigene Zwecke definiert werden.

Kommandos für die Gruppenverwaltung

Folgende Kommandos werden bei der Gruppenverwaltung benötigt:

`groupadd` neue Gruppen erstellen

`groupmod` bearbeiten bereits bestehender Gruppen

`groupdel` löschen von Gruppeneinträgen

Bevor ein Benutzer einer Gruppe zugeordnet werden kann, muss diese erst erstellt werden. Dazu wird das Kommando `groupadd` verwendet. Die Syntax von `groupadd` lautet:

```
groupadd [-g GID] Gruppenname
```

Mit der Option `-g` kann eine bestimmte Gruppennummer vorgegeben werden. Dabei ist zu beachten, dass Werte bis 999 meist Systemgruppen vorbehalten sind. Wird diese Option weggelassen, wird die nächste freie GID verwendet.

Bereits bestehende Gruppen können mit `groupmod` bearbeitet werden:

```
groupmod [-g GID] [-n Name] Gruppenname
```

`-g GID` die GID der angegebenen Gruppe ändern

`-n Name` der angegebenen Gruppe einen neuen Gruppennamen zuweisen, die GID der Gruppe bleibt dabei unverändert

Um bereits bestehende Gruppen zu löschen, wird `groupdel` verwendet.

Beispiele zur Gruppenverwaltung

<code>groupadd -g 1010 it11fa</code>	Gruppe <code>it11fa</code> mit der GID 1010 anlegen
<code>groupmod -g 2011 it11fa</code>	der bestehenden Gruppe <code>it11fa</code> wird die neue GID 2011 zugewiesen
<code>groupmod -g 2100 -n teacher lehrer</code>	Gruppe <code>lehrer</code> nach <code>teacher</code> umbenennen und die neue GID 2100 zuweisen
<code>groupdel it11fa</code>	Gruppe <code>it11fa</code> löschen

Zuordnung von Benutzern zu Gruppen mit `usermod`

Neue Benutzer können direkt beim Anlegen mit `useradd` bereits bestehenden Gruppen zugeordnet werden. Sind die benötigten Gruppen zu diesem Zeitpunkt noch nicht vorhanden oder soll die Gruppenzugehörigkeit nachträglich verändert werden, wird `usermod` verwendet. Dieses Kommando ändert existierende Benutzerkonten und verwendet im wesentlichen die gleichen Optionen wie `useradd`.

```
usermod [-g Gruppe] [-G Gruppe1[,Gruppe2]... ] Benutzername
```

Die wichtigsten Optionen für die Gruppenverwaltung mit `usermod` sind:

`-g Gruppe` Primär-Gruppe des Benutzers, als Name oder GID,

`-G Gruppe1[,Gruppe2]...` weitere Gruppen (Sekundär-Gruppen) des Benutzers, als Namen oder GIDs, Gruppen mit Kommas getrennt

Beispiel, um die Benutzerin `lorenzvi` zusätzlich der Ubuntu-Administrations-Gruppe `admin` zuzuordnen:

```
usermod -G admin lorenzvi
```

**Übung:**

Die folgenden Aufgaben müssen Sie mit administrativen Rechten bearbeiten. Notieren Sie, wenn möglich, jeweils die entsprechenden Kommandos.

- Legen Sie die folgenden Gruppen mit der jeweils in Klammern angegebenen GID an:
it11fa (1010), it11se (1020), it11fs (1030) und lehrer (1000)

- Erstellen Sie das Vorlagenverzeichnis für die Datei-Grundausstattung der Schüler und Lehrer /etc/skel.schueler bzw. /etc/skel.lehrer und kopieren Sie dort alle Dateien aus /etc/skel hinein.

- Legen Sie die Benutzerkonten mit den in der folgenden Liste angegebenen Einstellungen an.

Anmeldename	Home-Verzeichnis	Primärgruppe	Vorlagenverzeichnis	Kommentar
brunnefl	/home/brunnefl	it11fa	/etc/skel.schueler	Brunner,Florian,IT11FA
summerma	/home/summerma	it11se	/etc/skel.schueler	Summer,Maximilian,IT11SE
janist	/home/janist	it11fs	/etc/skel.schueler	Jani,Steffi,IT11FS
lorenzvi	/home/lorenzvi	lehrer	/etc/skel.lehrer	Lorenz,Viola,Lehrer

Prüfen Sie durch einen Blick sowohl in die Gruppen- als auch in die Benutzerdatenbank, ob die neuen Konten angelegt wurden.

- Schauen Sie sich die Passwortdatenbank an. Woran erkennen Sie, dass für die neu angelegten Benutzer noch kein Passwort vergeben wurde?

Setzen Sie jetzt für die neu angelegten Benutzerkonten das Passwort „geheim“.

Wie haben sich die entsprechenden Einträge in der Passwortdatenbank verändert?

5. Melden Sie sich an weiteren Konsolen mit den Anmeldenamen `brunnefl`, `summerma` und `janist` an (Weitere Konsolen erhalten Sie im Textmodus mit den Tastenkombinationen `Alt+F2`, `Alt+F2` usw. bzw. `Strg+Alt+F2`, `Strg+Alt+F3` von der grafischen Oberfläche.).

- a) Wechseln Sie zurück auf die Konsole, an der Sie als `root` angemeldet sind und stellen Sie fest, welche Benutzer gerade angemeldet sind. Diese Informationen können Sie mit den Kommandos `w` und `who` erhalten.

- b) Setzen Sie für `janist` die Shell `/bin/dash`. Testen Sie den Erfolg Ihrer Arbeit, indem Sie `janist` ab- und wieder anmelden.

- c) Tragen Sie für `summerma` als Shell `/bin/false` ein und überprüfen Sie, ob unter dieser Benutzerkennung tatsächlich keine Anmeldung mehr möglich ist. Setzen Sie anschließend den Shell-Eintrag wieder auf `/bin/bash`.

- d) Erstellen Sie die Gruppe `photo` mit GID 2000 und ordnen Sie `summerma` und `brunnefl` zusätzlich dieser Gruppe zu. Schauen Sie sich die Änderungen in der Gruppendatenbank an und prüfen Sie dann, ob diese beiden Benutzer mit `newgrp` in diese Gruppe wechseln können.

- e) Prüfen Sie, ob auf es auf Ihrem System die Gruppe `admin` gibt. Falls ja, ordnen Sie `lorenzvi` zusätzlich dieser Gruppe zu.

- f) Löschen Sie diese neu angelegten Benutzerkonten wieder. Achten Sie darauf, dass auch die Home-Verzeichnisse gelöscht werden.

3.1.6 Fragen

1. Um welche besondere Art von Benutzern handelt es sich bei den Benutzernamen `daemon`, `bin`, `sys` und `wwwrun`?

2. Geben Sie das vollständige Kommando an, um den Benutzer Frank Meyer aus der Klasse IT11FS mit dem Anmeldenamen `meyerfr` anzulegen. Seine Primär-Gruppe soll `it11fs` sein, sein Home-Verzeichnis muss mit `/home/meyerfr` angelegt werden und die Grundausrüstung soll aus `/etc/skel.schueler` dort hineinkopiert werden. Ebenfalls soll dabei auch der Kommentar „Meyer, Frank, IT11FS“ in die Benutzerdatenbank mit eingetragen werden.

3. Erläutern Sie zwei Möglichkeiten, um die UID eines Benutzers zu erhalten.

4. Sie möchten die UID der Benutzerin `janist` ermitteln. Geben Sie das vollständige Kommando an, um diese Information zu erhalten.

5. Mit dem folgenden Kommando wird die GID der Primärgruppe der Benutzerin `janist` ausgegeben:

```
id -g janist  
1030
```


Erläutern Sie eine andere Möglichkeit, um diese Information zu erhalten.

6. Welches Zeichen im Passwort-Feld der `/etc/passwd` weist darauf hin, dass die verschlüsselten Passwörter in `/etc/shadow` gespeichert sind?

7. Für die folgenden fünf Aufgaben betrachten Sie bitte das Beispiel für die Passwortdatenbank `/etc/passwd` auf Seite 50.
 - a) Geben Sie die GID der Gruppe `it11se` an.

 - b) Erklären Sie, warum sich der Benutzer `summerma` nicht am System anmelden kann und wie Sie als Systemadministrator dieses Problem lösen.

 - c) Der Benutzer `bittler` hat sein Passwort das letzte Mal am Mittwoch, 21.03.2012 geändert. Wann wurde das Passwort des Benutzers `brunnefl` das letzte Mal geändert?

 - d) Welcher Eintrag müsste wie verändert werden, wenn die Benutzerin `lorenzvi` ihr Passwort alle 30 Tage ändern soll?

e) Betrachten Sie die folgende Bildschirmausgabe:

```
grep johannes /etc/passwd
johannes:x:1102:100:./home/johannes:
sudo grep johannes /etc/shadow
johannes:!:15591:0:99999:7:::
```

Erklären Sie, warum sich der Benutzer `johannes` nicht am System anmelden kann und wie Sie als Systemadministrator dieses Problem lösen.

8. Betrachten Sie die folgende Bildschirmausgabe:

```
sudo useradd johannes
grep johannes /etc/passwd
johannes:x:1102:100:./home/johannes:
sudo grep johannes /etc/shadow
johannes:!:15591:0:99999:7:::
ls /home/johannes
ls: /home/johannes: No such file or directory
```

Welches Problem besteht und wie kann es behoben werden?

9. Was könnten Sie tun, wenn Sie Ihr `root`-Passwort vergessen haben?

10. Wie könnte man mit `sudo` dauerhaft administrative Rechte erlangen?

11. Mit dem Kommando `last Benutzername` können Sie die Statistik der letzten Anmeldungen des angegebenen Benutzers ausgeben lassen. Wird der Benutzername weggelassen, wird die Statistik für alle Benutzer angezeigt. Mit `last reboot` kann auch festgestellt werden, wann der Rechner das letzte Mal neu gestartet wurde.

Probieren Sie beides aus, indem Sie sich auf einer anderen Konsole als `user1` an- und wieder abmelden und dann den Rechner neu starten.

3.2 Dateiberechtigungen und Dateieigentum

Die Benutzer- und Zugriffsverwaltung von Linux ermöglicht es, dass mehrere Benutzer unabhängig voneinander an einem Linux-System arbeiten können und sich dabei nicht gegenseitig beeinträchtigen.

Die Benutzerverwaltung legt über die Zuordnung der Benutzer zu verschiedenen Gruppen fest, wer auf welche Dateien zugreifen darf, unter welchen Umständen ein Benutzer Daten eines anderen Benutzers lesen oder verändern darf, wer welche Programme ausführen kann und wer welche Hardware-Komponenten benutzen darf.

Die Zugriffsverwaltung basiert auf der Benutzerverwaltung und setzt über Dateiberechtigungen die eingestellten Rechte durch.

Versucht beispielsweise der Benutzer `bittler` die Benutzerdatenbank zu löschen, muss von der Zugriffsverwaltung des Linux-Systems ermittelt werden, ob er dazu auch über die nötigen Rechte verfügt:

```
rm /etc/passwd
rm: Entfernen (unlink) von "/etc/passwd" nicht möglich: Keine Berechtigung
```

Hier wurde vom System (vernünftigerweise) das Löschen der Datei verwehrt, da der Benutzer `bittler` nicht das Recht hatte, die Datei zu löschen.

3.2.1 Benutzerklassen und Zugriffsrechte

Linux unterscheidet hinsichtlich der Dateiberechtigungen drei Klassen von Benutzern:

- der Eigentümer der Datei (`owner`)
- die Mitglieder der Gruppe, der die Datei zugeordnet ist (`group`)
- alle anderen Benutzer (`others`)

Für jede der drei Benutzerklassen können getrennt je drei Zugriffsrechte vergeben werden. Diese sind in Leserecht (`r`), Schreibrecht (`w`) und Ausführrecht (`x`) unterteilt.

Pro Datei werden zusätzlich noch die drei Spezialrechte SUID, SGID und Sticky-Bit verwaltet.

Die Zugriffsrechte werden für jede Datei (also auch für Verzeichnisse, Gerätedateien, Sockets, Pipes, usw.) im Dateisystem hinterlegt und dort in Gruppen zu jeweils drei Bits gespeichert.

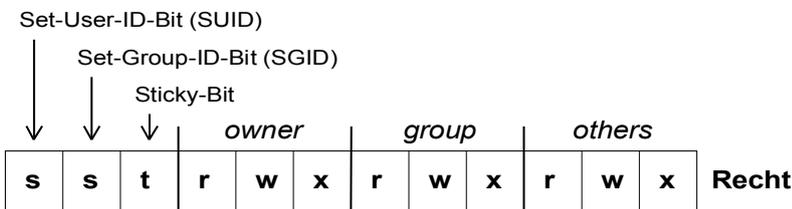


Abb. 3.1: Zugriffsrechte im Dateisystem

Mit `ls -l` Dateiname können die Zugriffsrechte angezeigt werden (siehe auch Abschnitt 2.5.3 auf Seite 31):

```
ls -l test.txt
-rw-r----- 1 bittler it11fa 208 10. Mär 21:55 test.txt
```

Die Datei `test.txt` ist eine gewöhnliche Datei (`-`). Sie gehört dem Benutzer `bittler`, der sie lesen und verändern darf (`rw-`). Die Datei ist der Gruppe `it11fa` zugeordnet. Alle Mitglieder dieser Gruppe dürfen die Datei lesen (`r--`). Alle anderen Benutzer haben keinerlei Rechte auf diese Datei (`---`).

Im folgenden Beispiel sind die Besonderheiten bei den Berechtigungen der Benutzer- und der Passwortdatenbank dargestellt:

```
ls -ld /etc/passwd /etc/shadow
-rw-r--r-- 1 root root 1561 22. Nov 12:45 /etc/passwd
-rw----- 1 root shadow 1187 22. Nov 18:33 /etc/shadow
```

Die Benutzerdatenbank `/etc/passwd` kann von allen Benutzern gelesen (`r--`), aber nur von `root` geschrieben (`rw-`) werden. Auf die Passwortdatenbank `/etc/shadow` hat nur `root` Lese-Schreibzugriff. Weder die Mitglieder einer Gruppe (`---`) noch irgendein anderer Benutzer (`---`) können darauf zugreifen.

Die Spezialrechte: SUID, SGID und das Sticky-Bit

Am folgenden Beispiel lassen sich die Spezialrechte gut erkennen:

```
ls -ld /tmp /usr/bin/passwd /bin/wall
drwxrwxrwt 1 root root 4096 15. Aug 08:18 /tmp
-rwsr-xr-x 1 root shadow 73640 15. Aug 22:33 /usr/bin/passwd
-rwxr-sr-x 1 root tty 14552 6. Jun 2008 /usr/bin/wall
```

Sticky-Bit Das Verzeichnis `/tmp` kann von allen Benutzern beschrieben werden. Das bedeutet, dass jeder dort Dateien lesen, erstellen und auch löschen kann. Da jedoch das Sticky-Bit gesetzt ist (`rwt`), hat dies zur Folge, dass jeder Benutzer nur die Dateien löschen darf, die ihm gehören.

SUID auf Programme Bei der Datei `/usr/bin/passwd` handelt es sich um das Programm zum Setzen des Passwortes. Wenn ein Benutzer sein Passwort ändern will, ruft er dieses Programm auf und gibt sein neues Passwort ein. Um das neue Passwort in die Passwortdatenbank `/etc/passwd` einzutragen, sind jedoch Schreibrechte nötig, über die aber nur `root` verfügt (s. o.). Normalerweise laufen Programme immer mit den Rechten des aktuellen Benutzers, der sie aufruft. Mit dem Set-User-ID (SUID) Spezialrecht läuft das `passwd`-Programm aber mit den Rechten des Dateieigentümers (hier: `root`) und ermöglicht so dem `passwd`-Programm den vollen Zugriff auf die Passwortdatenbank.

SGID auf Programme Das Set-Group-ID (SGID) Spezialrecht funktioniert ähnlich wie SUID. Bei der Programmausführung wird die GID der Programmdatei und nicht die des aktuellen Benutzers verwendet. Für das obige Beispiel bedeutet dies, dass wenn das Programm `/usr/bin/wall` von einem normalen Benutzer ausgeführt wird, läuft es mit den Rechten der Gruppe `tty` (mit `wall` können allen angemeldeten Benutzern Meldungen gesendet werden).

SGID auf Verzeichnisse Ist SGID bei Verzeichnissen gesetzt, werden die darin erstellten Dateien der Gruppe des Verzeichnisses zugeordnet und nicht wie normalerweise der Gruppe des Benutzers, der die Datei erzeugt hat.

Damit SUID, GUID und das Sticky-Bit wirken, muss bei SUID das Ausführungsrecht für den Eigentümer der Datei gesetzt sein, bei GUID das Ausführungsrecht für die Gruppe. Beim Sticky-Bit muss für alle anderen Benutzer (`others`) das Ausführungsrecht gesetzt sein.

Werden bei der Ausgabe von `ls -l` einzelne Spezialrechte groß angezeigt (z. B.: `-rwsr-xr-x`, `drwxrwxrwt`), wurde das entsprechende Ausführungsrecht nicht gesetzt und das Spezialrecht bleibt wirkungslos.

3.2.2 Zugriffsrechte und Eigentümerschaft ändern

Folgende Kommandos werden zum Setzen der Zugriffsrechte und zum Ändern der Eigentümerschaft einer Datei benötigt:

chmod Zugriffsrechte ändern

chown Datei-Eigentümer ändern

chgrp Gruppenzugehörigkeit ändern

Zugriffsrechte mit Buchstabennotation setzen

Wenn Zugriffsrechte mit der Buchstabennotation gesetzt werden sollen, können folgende Buchstaben und Symbole verwendet werden:

Benutzerklasse	Rechte	Symbole
u Eigentümer (user, owner)	r lesen (read)	+ Berechtigung hinzufügen
g Mitglieder der Gruppe (group)	w schreiben (write)	- Berechtigung entfernen
o alle anderen Benutzer (others)	x ausführen (execute)	= nur diese Rechte werden gesetzt,
a alle drei Benutzerklassen (all)	s SUID/SGID	alle anderen Rechte werden, sofern sie vergeben
	t Sticky-Bit	waren, entzogen.

Beispiel

Wenn der Benutzer `bittler` möchte, dass diese Datei nur noch von den anderen Mitgliedern seiner Primär-Gruppe `it11fa`, nicht aber von Anwendern außerhalb der Gruppe gelesen werden kann, dann muss er mit `chmod` das Leserecht für alle anderen Benutzer (`others`) entfernen:

```
ls -l skript.sh
-rw-r--r-- 1 bittler it11fa 904 10. Mär 21:55 skript.sh
chmod o-r skript.sh
ls -l skript.sh
-rw-r----- 1 bittler it11fa 904 10. Mär 21:55 skript.sh
```

Damit er und seine Gruppenmitglieder das Skript ausführen können, müssen noch Ausführungsrechte hinzugefügt werden:

```
chmod u+x,g+x skript.sh
ls -l skript.sh
-rwxr-x--- 1 bittler it11fa 904 10. Mär 21:55 skript.sh
```

Die beiden `chmod`-Kommandos hätten auch kombiniert werden können: `chmod u+x,g+x,o-r skript.sh`

Zugriffsrechte mit Oktalnotation setzen

Die Zugriffsrechte werden für jede Datei in Gruppen zu jeweils drei Bits gespeichert. Für ein vergebenes Recht wird das entsprechende Bit gesetzt.

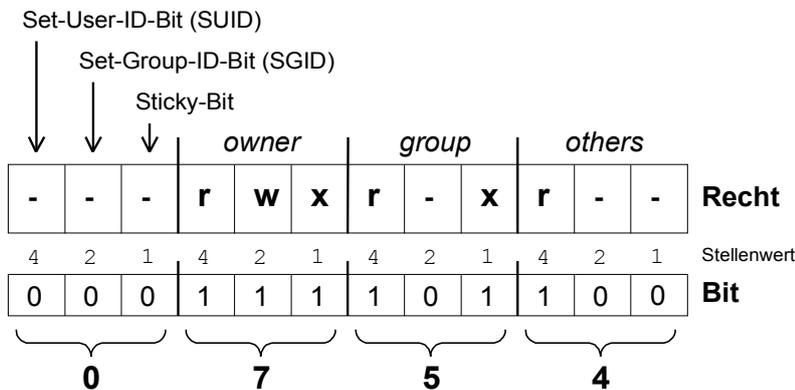


Abb. 3.2: Zugriffsrechte mit Oktalnotation

Beispiel: Für die Datei `test.txt` sollen folgende Zugriffsrechte vergeben werden:

- Eigentümer: (`rwX`) lesen, schreiben, ausführen
- Gruppe: (`r-x`) lesen, ausführen
- alle Anderen: (`r--`) lesen
- die Spezialrechte sollen nicht gesetzt sein

Werden von rechts nach links jeweils drei Bits zu einer Dezimalzahl gewandelt und die sich ergebenden Ziffern hintereinander geschrieben, so ergibt sich die Oktalzahl `0754`.

Um diese Zugriffsrechte zu setzen, muss dann `chmod 0754 test.txt` eingegeben werden. Da keine Spezialrechte vergeben werden, kann die führende Null auch weggelassen werden: `chmod 754 test.txt`.

Beispiele für Oktal- und Buchstabennotation

Buchstaben	Rechte	oktal	Bedeutung
<code>u=rw, g=r, o=</code>	<code>rw-r-----</code>	640	Nur der Eigentümer kann die Datei lesen und verändern. Die Gruppenmitglieder können die Datei lesen, alle anderen haben keinen Zugriff.
<code>a=r, u+w</code>	<code>rw-r--r--</code>	644	Alle Systembenutzer dürfen die Datei lesen, nur der Eigentümer darf sie verändern.
<code>u=rwx, g=rX, o=</code>	<code>rwxr-x---</code>	750	Der Eigentümer kann die Datei lesen, beschreiben und ausführen. Die Gruppenmitglieder dürfen die Datei lesen und ausführen, alle anderen haben keinen Zugriff.

Die vollständigen Kommandos für den Dateinamen `test.txt` lauten dann:

Oktalnotation

```
chmod 640 test.txt
chmod 644 test.txt
chmod 750 test.txt
```

Buchstabennotation

```
chmod u=rw,g=r,o= test.txt
chmod a=r,u+w test.txt
chmod u=rwx,g=rx,o= test.txt
```

Über die Option `-R` können mit `chmod` Zugriffsrechte rekursiv, d. h. auch in allen Unterverzeichnissen geändert werden.

Eigentümerschaft von Dateien ändern

Nur `root` darf die Eigentümerschaft von Dateien ändern!

Um eine Datei einer anderen Gruppe zuzuordnen, wird das Kommando `chown` verwendet:

```
chown [Optionen] Eigentümer[:Gruppe] Datei
```

Wenn eine Datei einer anderen Gruppe zugeordnet werden soll, wird das Kommando `chgrp` verwendet:

```
chgrp [Optionen] Gruppe Datei
```

Beispiel:

Die Benutzerin `janist` soll die neue Eigentümerin der Datei `/tmp/test.txt` werden:

```
sudo ls -l /tmp/test.txt
-rwxrw-r-- 1 bittler it11fa  2508 21. Apr 21:55 /tmp/test.txt
sudo chown janist /tmp/test.txt
ls -l /tmp/test.txt
-rwxrw-r-- 1 janist  it11fa  2508 21. Apr 21:55 /tmp/test.txt
```

Dann wird die Datei der Gruppe `it11fs` zugeordnet:

```
sudo chgrp it11fs /tmp/test.txt
ls -l /tmp/test.txt
-rwxrw-r-- 1 janist  it11fs  2508 21. Apr 21:55 /tmp/test.txt
```

Wenn sowohl die Eigentümerschaft einer Datei zu ändern und auch deren Gruppenzuordnung anzupassen sind, kann das mit einem einzigen Aufruf von `chown` erreicht werden, indem nach dem Benutzernamen, durch einen Doppelpunkt getrennt, der Gruppenname mit angegeben wird:

```
sudo chown janist:it11fs /tmp/test.txt
```

Auch `chown` und `chgrp` können über die Option `-R` rekursiv arbeiten.

**Übung:**

Vorbereitung für die Bearbeitung der weiteren Aufgaben:

Legen Sie die Gruppen `it11fa` und `it11se` und Benutzerkonten für `brunnefl` und `summerma` an. Führen Sie dazu als `root` die folgenden Kommandos aus:

```
groupadd -g 1010 it11fa
groupadd -g 1020 it11se
groupadd -g 2000 photo

useradd -m -g it11fa -G photo brunnefl
useradd -m -g it11se -G photo summerma

echo "brunnefl:geheim" | chpasswd
echo "summerma:geheim" | chpasswd
```

Hinweis: Das Programm `chpasswd` liest Benutzername und Passwort von der Standardeingabe. Die beiden Angaben werden durch einen Doppelpunkt getrennt, das Passwort kann dabei im Klartext angegeben werden.

Notieren Sie, wenn möglich, jeweils die entsprechenden Kommandos.

1. Ergänzen Sie in der Tabelle die fehlenden Angaben:

Anmeldename	Home-Verzeichnis	Primärgruppe	Sekundärgruppe	Passwort
brunnefl				geheim
summerma				geheim

2. Melden Sie sich jetzt an einer anderen Konsole als `brunnefl` an.

Wechseln Sie nach `/tmp` und erstellen Sie dort mit `echo Hallo >testdat` eine Datei.

Setzen Sie für `testdat` folgende Rechte:

Besitzer: lesen, schreiben, ausführen, Gruppe: lesen, Alle anderen: keine Rechte, die Spezialrechte sollen nicht gesetzt sein.

Geben Sie die passenden `chmod`-Kommandos für Buchstaben- und Oktalnotation an:

Buchstabennotation:

Oktalnotation:

3. Melden Sie sich an einer weiteren Konsole als `summerma` an.

Kann `summerma` den Inhalt der Datei `/tmp/testdat` lesen?

Falls ja, verhindern Sie dies als `brunnefl`, indem Sie die entsprechenden Rechte entziehen.

Falls nein, erlauben Sie dies als `brunnefl`, indem Sie die entsprechenden Rechte gewähren.

Geben Sie wieder die passenden `chmod`-Kommandos für Buchstaben- und Oktalnotation an:

Buchstabennotation:

Oktalnotation:

Wie prüfen Sie den Erfolg Ihrer Arbeit?

4. Erläutern Sie kurz, woran Sie erkennen, ob auf ein Verzeichnis das Sticky-Bit gesetzt ist.
-
-

5. Schreibrecht auf Verzeichnisse. Führen Sie nun als root folgendes aus:

```
mkdir /testverzeichnis
chmod 777 /testverzeichnis
echo hallo von root > /testverzeichnis/testdatei
chmod 644 /testverzeichnis/testdatei
```

6. Melden Sie sich jetzt als brunnefl an.

Warum können Sie z. B. mit `echo hallo >> /testverzeichnis/testdatei` nicht in `testdatei` schreiben?

7. Versuchen Sie mit `rm -f /testverzeichnis/testdatei` die Datei zu löschen. Begründen Sie das, was passiert!

8. Setzen Sie jetzt als root das Sticky-Bit auf `/testverzeichnis` Kommando:

Legen Sie dort als brunnefl eine Datei an: `echo brunnefl > /testverzeichnis/brunnefl`
 Legen Sie dort als summerma eine Datei an: `echo summerma > /testverzeichnis/summerma`

9. Kann summerma die Datei von brunnefl (bzw. umgekehrt) löschen? Begründen Sie Ihre Antwort.

10. Wechseln sie als summerma mit `newgrp photo` in die Gruppe `photo` und wiederholen Sie das folgende Kommando:

```
echo summerma > /testverzeichnis/summerma-photo
```

Welche Gruppe wurde dieser Datei zugeordnet?

Gewähren Sie der Gruppe Lese- und Schreibrechte und entziehen Sie allen anderen die Rechte auf diese Datei:

```
chmod g=rw,o= /testverzeichnis/summerma-photo
```

Melden Sie sich als brunnefl an, wechseln Sie mit `newgrp photo` in die Gruppe `photo` und prüfen Sie mit

```
echo brunnefl >> /testverzeichnis/summerma-photo
```

ob brunnefl Schreibrechte auf diese Datei besitzt.

11. Aufräumen: Löschen Sie als root die angelegten Gruppen, Benutzer und Verzeichnisse:

```
userdel -r brunnefl  
userdel -r summerma  
groupdel it11fa  
groupdel it11se  
groupdel photo  
rm -r /testverzeichnis
```

3.2.3 Fragen

1. Übersetzen Sie die folgenden Kommandos von der Buchstabennotation in die Oktalnotation.

`chmod o=,g=r,u=rw test.txt` _____

`chmod a=rw,g-w,o-r test.txt` _____

2. Übersetzen Sie die folgenden Kommandos von der Oktalnotation in die Buchstabennotation.

`chmod 751 test.txt` _____

`chmod 642 test.txt` _____

3. Das Kommando `ls -l testdatei` liefert aktuell folgende Ausgabe:

```
-r-xr-xr-x 1 janist it11fs 179 15. Dez 11:30 testdatei
```

Geben Sie das vollständige Kommando an, mit dem die Benutzerin `janist` die Rechte der Datei `testdatei` so ändert, dass die Gruppe nur lesen, der Eigentümer schreiben und lesen kann und alle anderen keine Rechte besitzen. Geben Sie das vollständige Kommando in Oktal- oder Buchstabennotation an:

4. Betrachten Sie folgende Bildschirmausgabe:

```
ls -l test.txt
-rwxr-xr-- 1 bittler it11fa 208 10. Mär 21:55 test.txt
Kommando1
ls -l test.txt
-rw-r--r-- 1 bittler it11fa 208 10. Mär 21:55 test.txt
Kommando2
ls -l test.txt
-rw----- 1 bittler it11fa 208 10. Mär 21:55 test.txt
```

Welche beiden Kommandos wurden eingegeben? Geben Sie diese vollständig an!

Kommando1 _____

Kommando2 _____

5. Wozu werden Sticky-Bit, SUID- und GUID-Bit benötigt?

6. Wie setzt man jeweils Sticky-, SUID- und GUID-Bit? Geben Sie Beispiele in Oktal- und Buchstabennotation an.

7. Betrachten Sie folgende Bildschirmausgabe:

```
ls -ld /testverzeichnis
drwxrwxrwt 1 root root 4096 12. Nov 19:25 /testverzeichnis
```

Welches Problem besteht und wie lösen Sie es?

8. Mit der folgenden, etwas sperrigen Kommandosequenz, wird die Ausgabe von `ls` an der linken Seite mit der Oktal-Darstellung der Berechtigungen ergänzt. Probieren Sie es aus!

```
ls -l | awk '{k=0; for(i=0;i<=8;i++)k+=((substr($1,i+2,1)~/[rwx]/)*2^(8-i)); if(k)printf("%0o ",k)};print}'
```

4 Einheit III – Skripte erleichtern das Arbeiten

Skripte erstellen und mit diesen arbeiten ist das Thema dieses Abschnitts. Dabei wird im ersten Schritt gezeigt, wie man ein Skript erstellt und ausführbar macht, so dass es direkt gestartet werden kann. Um „programmieren“ zu können, muss man Variablen anlegen und mit diesen arbeiten können, dabei werden auch Aufrufparameter behandelt.

Nach den ersten Grundkenntnissen kann es dann an das Anlegen der Gruppen für jede Klasse gehen, aber zuvor werden die Schleifen behandelt werden, da ja nicht nur eine Gruppe, sondern evtl. sehr viele Gruppen angelegt werden müssen. Damit man gewisse Prüfungen durchführen kann, wie z. B. ob eine Datei vorhanden ist, müssen Bedingungen geprüft und darauf reagiert werden.

Sind die Gruppen angelegt, so kann man die Kennungen für die Benutzer erstellen. Dazu werden die Schülernamen erst einmal auf eine gewisse Norm gebracht werden, bevor diese erzeugt werden können. Damit jeder Schüler sein individuelles Passwort bekommt, muss dies entsprechend generiert werden.

Damit der Lehrer auch alle Verzeichnisse etc. lesen und verändern kann, muss ein SysAdmin eingerichtet werden, der dies ermöglicht.

Inhalt

4.1	Einführung in die Skriptprogrammierung	68
4.1.1	Das erste Skript	68
4.1.2	Skript ausführen	68
4.1.3	Variablen	68
4.1.4	Parameter	69
4.2	Gruppen anlegen	74
4.2.1	Datei zeilenweise auslesen	74
4.2.2	Schleifen	74
4.2.3	Bedingungen	77
4.2.4	Mehrfachauswahl	77
4.2.5	Zeichen umwandeln	79
4.2.6	Vorhandene Gruppen ermitteln	79
4.3	Benutzer anlegen	80
4.3.1	Generieren der Schülerkennung	80
4.3.2	Passwörter generieren	80
4.3.3	Schülerkennung hinzufügen	81
4.3.4	Lehrerkennung hinzufügen	82
4.3.5	Superuser einrichten	83
4.4	Fragen	85

4.1 Einführung in die Skriptprogrammierung

In den letzten beiden Abschnitten haben Sie schon die Verwendung und Ausführung von Kommandos kennen und anwenden gelernt. Werden die Kommandos öfters verwendet, ist es mit sehr viel „Handarbeit“ verbunden, immer die gleichen Befehle, mit teilweise nur wenig geänderten Aufrufparametern, einzugeben.

Durch kleine Skripte ist es möglich, diese teils lästige „Handarbeit“ zu automatisieren.

Skripte in der einfachsten Form sind Befehle, die in eine Textdatei geschrieben werden und nacheinander abgearbeitet werden. Die Skripte, die wir hier in diesem Abschnitt erstellen, sind Bash-Skripte, beruhen also auf den Möglichkeiten der Bash-Shell.

Es gibt aber eine Vielzahl von Möglichkeiten, Skripte zu realisieren. Dazu kann jede beliebige Shell (Korn-Shell, Thompson-Shell, C-Shell, Bourne-Again-Shell, Z-Shell, ...), die auf dem System installiert ist, verwendet werden. Aber auch jede Interpreter-Sprache wie Perl, Python, Ruby, Lua, REXX, ... ist möglich. Je nach verwendeter Sprache ist der Funktionsumfang, die Möglichkeit, externe Bibliotheken einzubinden und natürlich die Ausführungsgeschwindigkeit unterschiedlich. Für unsere Beispiele reichen jedoch die Möglichkeiten der Bash vollkommen aus.

4.1.1 Das erste Skript

Das erste Skript gibt das aktuelle Datum und den Namen des Users aus, der das Skript ausführt.

```
#!/bin/bash
echo 'Heute ist      : ' `date`
echo 'Angemeldeter User:' $USER
```

```
Heute ist      : So 8. Jul 07:26:35 CEST 2012
Angemeldeter User: mgn
```

Die erste Zeile beginnt mit „#!“ und wird als Shebang (auch Magic Line oder hash-bang) bezeichnet. Damit wird festgelegt, mit welchem Programm das Skript ausgeführt wird. Danach muss ein Zeilenumbruch erfolgen. Ab Zeile zwei folgen dann die entsprechenden Skriptbefehle.

4.1.2 Skript ausführen

Damit das Skript direkt ausgeführt werden kann, muss noch das Ausführungsrecht („execute“) mit x gesetzt werden (siehe Abschnitt 3.2.1 auf Seite 58).

```
chmod +x meinskript
```

Das Skript kann dann direkt im Verzeichnis mit `./meinskript` aufgerufen werden. Ist das Skript in einem Verzeichnis, auf das die Variable `PATH` verweist, so kann die Angabe des Pfades weggelassen werden.

4.1.3 Variablen

Variablen haben Sie schon kurz auf Seite 20 kennengelernt. Zum Beispiel die Variable `HOME`, die standardmäßig die absolute Pfadangabe des Home-Verzeichnisses darstellt.

```
#!/bin/bash
echo 'Home-Verzeichnis : ' $HOME
```

```
Home-Verzeichnis : /home/mgn
```

Soll der Inhalt der Variablen ausgegeben werden, so wird dem Namen ein `$`-Zeichen vorangestellt.

Eine neue Variable wird mit Ihrem Namen und einer Zuweisung definiert.

```
#!/bin/bash
meineVariable="Inhalt der Variablen"
echo 'meineVariable : ' $meineVariable
```

```
meineVariable : Inhalt der Variablen
```

Sind z. B. Leerzeichen enthalten, so muss der Inhalt in Gänsefüßchen eingeschlossen werden.

Alle Variablen anzeigen

Mit dem Kommando `env` werden alle Variablen, die in der aktuellen Bash-Shell verwendet werden, angezeigt. Soll eine bestimmte Variable wieder gelöscht werden, so verwendet man das Kommando `unset`.

Welche Anführungszeichen soll ich verwenden?

Wird ein zusammenhängender Text, z. B. mit Leerzeichen verwendet, so muss dieser in Anführungszeichen zusammengefasst werden, da sonst die Shell nicht weiß, was zusammengehört. Dies wird auch Quoting genannt. Dazu gibt es zwei Möglichkeiten:

```
"Ein Text"
```

Mit doppelten Anführungszeichen (auch Gänsefüßchen genannt) wird der Text zusammengefasst. Die Shell interpretiert den Inhalt und ersetzt z. B. Variablen mit Ihrem Inhalt.

```
'Ein Text'
```

Mit einfachen Anführungszeichen wird der Text auch zusammengefasst, aber die Shell betrachtet den Inhalt nicht!

```
#!/bin/bash
echo 'doppelte Anführungszeichen:' "$HOME"
echo 'einfache Anführungszeichen:' '$HOME'
```

```
doppelte Anführungszeichen: /home/mgn
einfache Anführungszeichen: $HOME
```

Eindeutige Variablennamen

Sind die Variablennamen nicht eindeutig, so kann es zu Problemen kommen. Betrachten Sie nachfolgendes Beispiel:

```
#!/bin/bash
Linux="Linux"
Essentials="Essentials"
LinuxEssentials="Kurs $Linux $Essentials"
echo "$LinuxEssentials"
echo "${Linux}Essentials"
```

```
Kurs Linux Essentials
LinuxEssentials
```

Um eine Variable eindeutig anzusprechen, wird der Name in geschweifte Klammern eingeschlossen.

Variablen sind untypisiert!

In der Bash haben Variablen keinen speziellen Typ, wie z. B. in Java. Je nach dem, welcher Typ erwartet wird, wird versucht, z. B. die Zeichenkette in eine Zahl umzuwandeln. Klappt dies nicht, wird eine Fehlermeldung ausgegeben.

Wichtige Variablen in der Bash

- \$# Anzahl der Aufrufparameter
- \$0 der Name des Skripts
- \$1 – \$9 die ersten neun Aufrufparameter
- \$* alle Aufrufparameter (" \$* " verwenden)
- \$@ alle Aufrufparameter (" \$@" verwenden), jedoch mit „Quoting“, d. h. jeder Parameter ist in Anführungszeichen eingeschlossen.
- \$? der Rückgabewert (Exit-Status) eines Kommandos (mehr dazu auf Seite 75)
- \$! PID des letzten Kommandos (mehr dazu auf Seite 25)
- \$\$ PID des aktuellen Skriptes

4.1.4 Parameter

An ein Skript können, wie auch an jedes Kommando, Parameter übergeben werden (siehe Abschnitt 2.4.2 auf Seite 24). Diese bekommen fortlaufende Nummern. Der erste Parameter wird mit \$1 angesprochen, der zweite mit \$2 usw. Auf diese Weise können neun Parameter angesprochen werden. Die Variable \$0 enthält den Namen des Skripts.

Gegeben ist folgendes Skript:

```
#!/bin/bash
echo 0 : $0
echo 1 : $1
echo 2 : $2
echo 3 : $3
echo 4 : $4
```


Ein- / Ausgabeumleitung

Viele Kommandos verfügen über drei Ein- /Ausgabekanäle:

Kanal 0: Standard-Eingabekanal `stdin`

Der Standard-Eingabekanal ist die Tastatur.

Kanal 1: Standard-Ausgabekanal `stdout`

Der Standard-Ausgabekanal ist der Bildschirm.

Kanal 2: Standard-Fehlerausgabekanal `stderr`

Der Standard-Fehlerausgabekanal ist der Bildschirm.

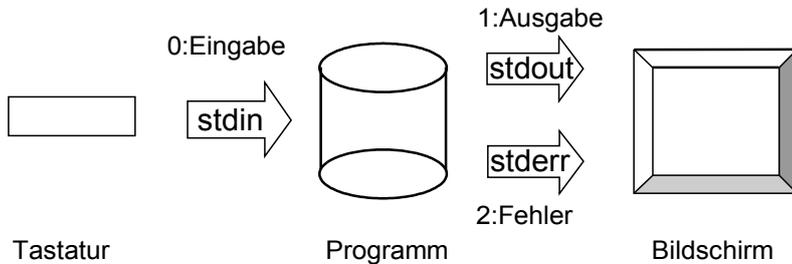


Abb. 4.1: Ein- / Ausgabeumleitung

Ausgabeumlenkung

Die Ausgabe kann mit den Zeichen `>` bzw. `>>` umgelenkt werden.

Beispiele:

`ls -l > Datei1` Die Ausgabe des Kommandos `ls -l` erfolgt nicht auf dem Bildschirm, sondern in `Datei1`. Wenn `Datei1` noch nicht existiert, wird sie neu angelegt. Eine bereits existierende Datei wird überschrieben.

`ls -l >> Datei1` Der alte Inhalt von `Datei1` wird nicht überschrieben. Die Ausgabe des Kommandos wird an den alten Inhalt von `Datei1` angehängt.

Eingabeumlenkung

Die Eingabe für ein Kommando kann von der Tastatur auf eine Datei umgelegt werden.

Beispiel:

Das Kommando `wc` liest Daten vom Standard-Eingabekanal (Tastatur) und gibt die Anzahl aller Zeilen, Wörter und Zeichen auf dem Bildschirm aus. Durch die Eingabeumlenkung können beispielsweise die Anzahl aller Zeilen, Wörter und Zeichen der Daten einer Datei ermittelt werden.

`wc < Datei1` gibt die Anzahl der Zeilen, Wörter und Zeichen von `Datei1` auf dem Bildschirm aus.

Ein- und Ausgabeumlenkung

`wc < Datei1 > Datei2` schreibt die Anzahl der Zeilen, Wörter und Zeichen der `Datei1` in `Datei2`.

Ausgabeumlenkung für Fehlermeldungen:

Trotz einer „normalen“ Ausgabeumlenkung in eine Datei werden Fehlermeldungen auf dem Bildschirm ausgegeben. Will man auch die Fehlermeldungen in eine Datei umlenken, muss man dies gesondert angeben.

Beispiel:

`wc < Datei1 > Datei2 2> Dateierr`

Der Ausdruck `2> Dateierr` lenkt Fehlermeldungen in die Datei `Dateierr` um. Die `2` von `2>` bezeichnet den Standardkanal 2 (Standard-Fehlerausgabekanal).

Mit dem Konstrukt `ls -l > Ausgabe 2>&1` werden beide Ausgabe-Kanäle in die Datei `Ausgabe` umgeleitet!

Umlenkung der Ausgabe eines Kommandos zur Eingabe eines Folgekommandos

Mit Hilfe einer Pipe (pipeline, Röhre) kann die Ausgabe eines Kommandos direkt als Eingabe eines weiteren Kommandos genutzt werden, ohne dass sie auf dem Bildschirm angezeigt wird. Das Symbol für die Pipe ist: „|“

Beispiel:

```
ls -l | wc
```

Die Ausgabe des Kommandos `ls -l` wird als Eingabe an das Kommando `wc` übergeben. Die Ausgabe von `wc` erfolgt auf dem Bildschirm.

Möchte man zur Fehlersuche trotzdem die Ausgabe betrachten, so kann man den Datenstrom mit dem Kommando `tee` auch duplizieren und in einer Datei speichern.

Beispiel: `ls | tee ls_ergebnis | sort`

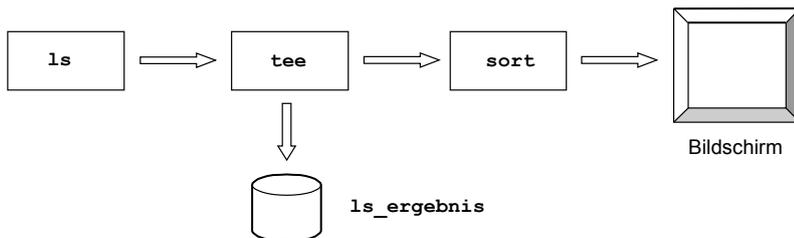


Abb. 4.2: Datenstrom duplizieren

Das Ergebnis, das von `ls` erzeugt wird, wird mit Hilfe des Kommandos `tee` in der Datei `ls_ergebnis` gespeichert. Außerdem wird es an das Kommando `sort` weitergeleitet (siehe Abbildung 4.2).

Hier-Dokumente

Die Standard-Eingabe wird auf ein Hier-Dokument, d. h. eine Zeilenfolge (die auch mehrere Zeilen enthalten kann) umgelenkt, die an der aktuellen Stelle eines Shell-Skripts (also „hier“) angegeben ist. Dadurch kann man die im Rahmen einer Eingabe-Umlenkung zu lesenden Daten im aktuellen Shell-Skript unterbringen und so auf eine externe Datei verzichten.

```
Kommando <<Marke
Eingabetext...
Eingabetext...
Eingabetext...
Marke
```

Die verwendete Marke muss dabei am Ende in einer Zeile stehen und darf keine Leerzeichen vor der Marke enthalten.

Das Hier-Dokument leitet Zeilenumbrüche (entspricht `\n`) direkt weiter. Wird `stdin` zum Lesen verwendet, so entspricht dies dem Tastendruck der `↵`-Taste (Enter).

Beispiel:

```
cat <<EOM
Hallo,
ich bin ein Hier-Dokument:
$(date)
EOM
```

```
Hallo,
ich bin ein Hier-Dokument:
So 8. Jul 07:27:27 CEST 2012
```

Übung:

1. Erstellen Sie ein kleines Skript, welche zwei Zahlen als Parameter enthält und diese einmal addiert, einmal subtrahiert und das Ergebnis jeweils ausgibt.
2. Suchen Sie mit Hilfe der man-Page und dem Internet weitere Standardoperatoren wie '+', '-', '**', '/', ..., notieren Sie diese und probieren Sie diese in einem kleinem Skript aus.
3. Erstellen Sie das unter 4.1.4 angegebene Skript `bspshowparam` und rufen Sie es so auf:

```
bspshowparam "A B C D"  
bspshowparam A 'D C' D  
bspshowparam 1 2 3 4 5 6 7 8 9 0 a b c d e
```

Was ist passiert?

4. Lösen Sie abwechselnd mit `let`, direkt in der Bash mit `((...))` und mit `expr` folgende Aufgaben:

```
2697 - 487  
2697 * 487  
2697 / 87  
2697 + 3954
```

5. Berechnen Sie mit `bc` folgende Aufgaben auf 5 Stellen hinter dem Komma:

```
2.48 + 1.86  
56.92 / 4.7
```

6. Weisen Sie der Variable `var` nacheinander folgende Werte (exakt so wie sie angegeben sind) zu. Geben Sie die Variable jeweils danach aus, um zu sehen, ob die Zuweisung funktioniert hat.

```
Inhalt der Variablen  
"Inhalt der Variablen"  
$var  
$HOME
```

4.2 Gruppen anlegen

Damit alle Lehrer und Schüler auf dem Linux-System eine Kennung erhalten, ist es sinnvoll, ihre Benutzerkonten automatisch mit einem Skript anzulegen. Die Schülerdaten werden aus der Schulverwaltung ausgespielt und haben folgendes Format.

Die Spalten werden hier durch „:“ getrennt und haben folgende Bedeutung:

Nachname:Vorname:Klasse

Pyka:Johannes:IT11FS	Manfred:Rolf:IT11FA	Wanzinger:Guido:IT11SE
Gerber:Dennis:IT11FS	Mammensohn:Friedl:IT11FA	Summer:Maximilian:IT11SE
Ewender:Nils:IT11FS	Bittl:Erwin:IT11FA	Schmied:Burkhard:IT11SE
Feiler:Matthias:IT11FS	Kruppweis:Hubert:IT11FA	Metzker:Alexander:IT11SE
Wilms:Norman:IT11FS	Vogg:Ralf:IT11FA	Zehnter:Herbert:IT11SE
Hollmann:Michael:IT11FS	Schmidbauer:Walter:IT11FA	Mittelholz:Stefan:IT11SE
Kreiselmeier:Jens:IT11FS	Brunner:Florian:IT11FA	Helff:Marcus:IT11SE
Mannweiler:Christopher:IT11FS	Daniels:Martin:IT11FA	Ruhland:Christian:IT11SE
Pranckl:Klaus:IT11FS	Reithmayer:Richard:IT11FA	Reiter:Gerhard:IT11SE
Liepold:Michael:IT11FS	Pimmer:Christian:IT11FA	Vaitl:Andreas:IT11SE
Aurich:Karl:IT11FS	Schlecht:Daniel:IT11FA	Burneleit:Andreas:IT11SE
Meyer:Frank:IT11FS	Hohmann:Michael:IT11FA	Ehmer:Marc:IT11SE
Jakob:Jens:IT11FS	Bode:Dirk:IT11FA	Sertl:Andor:IT11SE
Bischoff:Christian:IT11FS	Grill:Dominik:IT11FA	Hay:Thomas:IT11SE
Lehnert:Michael:IT11FS	Perner:Florian:IT11FA	Leyendecker:Bernd:IT11SE
Ibanez:Dennis:IT11FS	Saumweber:David:IT11FA	Grabisch:Rainer:IT11SE
Maier:Thomas:IT11FS	Kirsch:Bernhard:IT11FA	Dusl:Alfred:IT11SE
Brandes:Martin:IT11FS	Joachim:Maximilian:IT11FA	Friedrich:Stefan:IT11SE
Flaim:Hendrik:IT11FS	Karlstedt:Otto:IT11FA	Mair:Christian:IT11SE
Kappl:Tony:IT11FS	Lauxtermann:Conny:IT11FA	Beslmeisel:Johann:IT11SE
Volke:Christian:IT11FS	Lendner:Frank:IT11FA	Weidner:Hubert:IT11SE
Jani:Steffi:IT11FS	Bork:Johann:IT11FA	Figl:Ingo:IT11SE
Weniger:Herbert:IT11FS	Unterer:Justus:IT11FA	Krause:Markus:IT11SE
Gerlicher:Julia:IT11FS	Hetzenegger:Sven:IT11FA	Meschenmoser:Thomas:IT11SE

Datei: schueler.txt (enthält auch Leerzeilen)

4.2.1 Datei zeilenweise auslesen

Die Schülerdatei ist so aufgebaut, dass für jeden Schüler eine Zeile verwendet wird. Daher ist es notwendig, dass man die Datei zeilenweise bearbeitet und die Daten extrahiert.

Eine Schleife hilft dabei, die Datei zeilenweise auszulesen, dazu müssen aber zuerst die allgemeinen Möglichkeiten der Schleifen betrachtet werden.

Damit die Skripte und vor allem die Ausgaben kurz bleiben, wird in den folgenden Beispielen nur mit drei Schülern gearbeitet.

```
Pyka:Johannes:IT11FS
Manfred:Rolf:IT11FA
Wanzinger:Guido:IT11SE
Datei: sch.txt
```

4.2.2 Schleifen

In der Bash gibt es mehrere Möglichkeiten mit Schleifen zu arbeiten.

Die for-Schleife dient dazu, eine Liste abzuarbeiten und für jeden Eintrag alle Anweisungen im Schleifenkörper zu durchlaufen.

Die allgemeine Syntax dabei lautet:

```
for <Schleifenvariable> in <Werte>; do
  # Anweisungen
done
```

Hinweis: Die Liste der Werte wird dabei mit einem Strichpunkt abgeschlossen, wenn das `do` in derselben Zeile steht.

Im nachfolgenden Beispiel wird die Schleife mit den Werten von 1 bis einschließlich 5 durchlaufen.

```
#!/bin/bash
for i in 1 2 3 4 5; do
  echo -n "$i "
done
```

```
1 2 3 4 5
```

Es lassen sich aber auch beliebige Zeichenketten verwenden.

```
#!/bin/bash
for i in Merkur Venus Erde Mars Jupiter Saturn ↵
Uranus Neptun; do
    echo -n "$i "
done
```

```
Merkur Venus Erde Mars Jupiter Saturn Uranus ↵
Neptun
```

Ab Bash-Version 4 ist für Zählschleifen auch folgendes Konstrukt möglich. Die Syntax lautet dabei:

```
{<Startwert>..<Endwert>..<Schrittweite>}
```

Die Schrittweite kann auch weggelassen werden, wenn diese den Wert 1 hat.

```
#!/bin/bash
for i in {1..10..2}; do
    echo -n "$i "
done
```

```
1 3 5 7 9
```

Wird ein Verzeichnis und/oder eine Datei mit Wildcards angegeben, so wird eine Liste an Dateien abgearbeitet.

```
#!/bin/bash
for file in /etc/ali*; do
    echo -n "$file "
done
```

```
/etc/aliases
```

Die for-Schleife kann auch direkt, wie in anderen Programmiersprachen üblich, als Zählschleife verwendet werden.

Die Syntax lautet dabei:

```
((<Initialisierung>;<Bedingung>;<Schrittweite>))
```

```
#!/bin/bash
for (( i=1; i<=5; i++ )); do
    echo -n "$i "
done
```

```
1 2 3 4 5
```

Neben der for-Schleife, stellt die Bash auch die while-Schleife zur Verfügung.

Die allgemeine Syntax dabei lautet:

```
while [ Bedingung ] ; do
    # Anweisungen
done
```

Die Schleife wird dabei solange durchlaufen, wie die Bedingung erfüllt ist, d. h. den logischen Wert `true` zurückliefert. Ursprünglich war für solche Tests das Kommando `test` zuständig. Jedoch wurde dann die Schreibweise vereinfacht und durch eckige Klammern ersetzt, so dass dies u. a. in Schleifen verwendet werden kann. Tabelle 4.1 auf der nächsten Seite zeigt eine Auswahl an Test-Operatoren.

Auf Rückgabewert reagieren

Jedes Programm gibt einen sogenannten Rückgabewert (return code, unter Windows `ERRORLEVEL` genannt) zurück. Dieser wird in der Variablen `?` gespeichert und kann auch direkt für weitere Aktionen genutzt werden.

```
kommando && kommando1
```

Mit `&&` wird der Rückgabewert ausgewertet und nur dann das Programm *kommando1* gestartet, wenn der Rückgabewert 0 ist.

```
kommando || kommando2
```

Mit `||` wird nur dann das Programm *kommando2* gestartet, wenn der Rückgabewert ungleich 0 ist.

Nachfolgend ein paar Beispiele zu den Operatoren.

```
#!/bin/bash
a=7; b=5
test $a -gt $b && echo true || echo false
```

```
true
```

Stellt `test` fest, dass die Bedingung zutrifft, wird als Rückgabewert 0 zurückgegeben, andernfalls 1. Das Kommando nach `&&` wird dabei nur bei einem Rückgabewert von 0 ausgeführt, das Kommando nach `||` wenn der Rückgabewert ungleich 0 ist.

```
#!/bin/bash
a="XXX"; b="YYY"
test $a = $b && echo true || echo false
test $a != $b && echo true || echo false
```

```
false
true
```

Mit einer entsprechenden Bedingung kann die while-Schleife auch als Zählschleife verwendet werden.

```
#!/bin/bash
x=1
while [ $x -le 5 ] ; do
    echo -n "$x "
    x=$(( $x + 1 ))
done
```

```
1 2 3 4 5
```

! Eine andere Anwendung der while-Schleife ist, eine Datei zeilenweise auszulesen, dazu wird das Schlüsselwort read verwendet. Die zu lesende Datei wird als Eingabeumleitung nach done angegeben. Genau diese Form benötigen wir, um die Schülerdatei auszulesen.

Hinweis: Beachten Sie, wie die Datei sch.txt mit einer Eingabeumleitung gelesen wird!

```
#!/bin/bash
while read zeile ; do
    echo $zeile
done < sch.txt
```

```
Pyka:Johannes:IT11FS
Manfred:Rolf:IT11FA
Wanzinger:Guido:IT11SE
```

Legt man den Feldtrenner IFS (Internal Field Separator) fest, so kann auch gleich die Zeile entsprechend zerlegt werden. Dabei muss für jede Spalte eine Variable angegeben werden, in der der Inhalt der Spalte gespeichert wird.

```
#!/bin/bash
IFS=":"
while read nachname vorname klasse ; do
    echo "Schüler: $nachname, $vorname, $klasse"
done < sch.txt
```

```
Schüler: Pyka, Johannes, IT11FS
Schüler: Manfred, Rolf, IT11FA
Schüler: Wanzinger, Guido, IT11SE
```

Tab. 4.1: Auswahl an Test-Operatoren

Operator	Beispiel	Beschreibung
Zahlen		
-eq	[\$a -eq \$a]	prüft auf Gleichheit
-ne	[\$a -ne \$a]	prüft auf Ungleichheit
-lt	[\$a -lt \$a]	prüft auf Kleiner
-le	[\$a -le \$a]	prüft auf Kleiner-Gleich
-gt	[\$a -gt \$a]	prüft auf Größer
-ge	[\$a -ge \$a]	prüft auf Größer-Gleich
Zeichenketten		
-z	[-z \$z]	prüft, ob die Zeichenkette leer ist
-n	[-n \$z]	prüft, ob die Zeichenkette nicht leer ist
=	[\$x = \$z]	prüft, ob die Zeichenketten gleich sind
!=	[\$x != \$z]	prüft, ob die Zeichenketten ungleich sind
Dateien		
-d	[-d \$f]	prüft, ob es sich um ein Verzeichnis handelt
-e	[-e \$f]	prüft, ob die Datei / das Verzeichnis existiert
-f	[-f \$f]	prüft, ob es sich um eine normale Datei handelt
-r	[-r \$f]	prüft, ob die Datei lesbar ist bzw. ob man die Rechte dazu hat
-s	[-s \$f]	prüft, ob die Datei nicht leer ist
-w	[-w \$f]	prüft, ob die Datei schreibbar ist bzw. ob man Schreibrechte hat
-nt	[\$g -nt \$f]	prüft, ob die Datei neuer ist als die andere
-ot	[\$g -ot \$f]	prüft, ob die Datei älter ist als die andere



Übung:

1. Probieren Sie die dargestellten Beispiele aus!
2. Erstellen Sie eine Schleife, die folgendes ausgibt:

```
1 : 2 : 3 : 4 : 5 : 6 :
```

3. Geben Sie mit einer Schleife die Zahlen von 10 bis 100 aus.
4. Zählen Sie mit einer Schleife von 100 bis 10 abwärts, verwenden Sie `test` im Schleifenkörper und geben Sie die Zahl nur dann aus, wenn sie kleiner als 50 ist.
5. Schreiben Sie ein kleines Skript, welches Vor-/Nachname vertauscht ausgibt.
6. Schreiben Sie ein kleines Skript, welches nur Schüler der Klasse IT11SE ausgibt.

4.2.3 Bedingungen

Ist die Datei `sch.txt` nicht vorhanden, so kommt es zu einer Fehlermeldung. Mit einer entsprechenden Abfrage, kann vorher geprüft werden, ob die Datei vorhanden ist. Dazu dient die If-Else-Anweisung. Für die Bedingung können auch die Operatoren aus Tabelle 4.1 auf der vorherigen Seite verwendet werden.

Die allgemeine Syntax dabei lautet:

```
if [ Bedingung ]; then
    # true-Anweisungen
else
    # false-Anweisungen
fi
```

Dabei ist der `else`-Zweig optional und kann auch weg gelassen werden. Die Bedingungen unterliegen den selben Regeln, wie bei der `while`-Schleife.

```
#!/bin/bash
touch testdatei
if [ -r testdatei ]; then
    echo Datei ist vorhanden
else
    echo Datei ist nicht vorhanden
fi
```

```
Datei ist vorhanden
```

4.2.4 Mehrfachauswahl

Mit If-Else-Anweisungen lassen sich beliebige Verschachtelungen aufbauen. Bei einer hohen Verschachtelungstiefe kann dies aber sehr schnell unübersichtlich werden. Abhilfe schafft in einem solchen Fall die Case-Anweisung.

```
case $Variable in
    Wert1)
        # tue was
        ;;
    Wert2)
        # tue was
        ;;
    Wert3)
        # tue was
        ;;
    *)
        # Alternative
        ;;
esac
```

Der `case`-Block wird von oben nach unten durchlaufen. Wird der Inhalt der Variablen in der Aufzählung gefunden, so werden die entsprechenden Kommandos ausgeführt (bis zu `;;`) und dann die `case`-Anweisung verlassen. Wird kein Wert gefunden, so wird der Default-Block (mit `*`) ausgeführt.

Im nachfolgenden Beispiel wird die Variable `var` mit dem Wert `Hugo` belegt. Im `case`-Block wird dann das entsprechende Kommando (hier `echo Hugo ...`) ausgeführt. Die Blöcke `Peter`, `Uschi` und `*` werden dabei nicht ausgeführt.

```
var=Hugo
case $var in
  Peter)
    echo Peter ...
    ;;
  Hugo)
    echo Hugo ...
    ;;
  Uschi)
    echo Uschi ...
    ;;
  *)
    echo nicht gefunden ...
    ;;
esac
```

```
Hugo ...
```

Der Wert kann auch direkt über die Tastatur eingegeben werden. Mit dem Kommando `read` werden Eingaben von der Standard-Eingabe in einer Variablen gespeichert.

```
read var
case $var in
  Peter)
    echo Peter ...
    ;;
  Hugo)
    echo Hugo ...
    ;;
  Uschi)
    echo Uschi ...
    ;;
  *)
    echo nicht gefunden ...
    ;;
esac
```

Schülerdatei vorhanden?

Das Skript zum Einlesen der Schülerdatei kann dann wie folgt umgebaut werden.

```
#!/bin/bash
if [ -r sch.txt ]; then
  IFS=":"
  while read nachname vorname klasse ; do
    echo "Schüler: $nachname, $vorname, $klasse"
  done < sch.txt
else
  echo Datei sch.txt nicht gefunden!
fi
```

```
Schüler: Pyka, Johannes, IT11FS
Schüler: Manfred, Rolf, IT11FA
Schüler: Wanzinger, Guido, IT11SE
```

Sucht man bei diversen Suchmaschinen, so findet man auch andere Lösungen, um eine Datei zeilenweise auszulesen.

Nachfolgend ein gerne gezeigtes Beispiel, welches aber deutlich aufwändiger ist, externe Kommandos benötigt (`cut`, `echo`) und deshalb wesentlich langsamer ist. Darauf soll aber nicht weiter eingegangen werden.

```
#!/bin/bash
for i in $(cat sch.txt); do
  nachname=$(echo $i | cut -d: -f1)
  vorname=$(echo $i | cut -d: -f2)
  klasse=$(echo $i | cut -d: -f3)
  echo Schüler: $vorname, $nachname, $klasse
done
```

```
Schüler: Johannes, Pyka, IT11FS
Schüler: Rolf, Manfred, IT11FA
Schüler: Guido, Wanzinger, IT11SE
```



Übung:

Probieren Sie die dargestellten Beispiele aus!

4.2.5 Zeichen umwandeln

Jetzt sind die Daten vorhanden, um die Gruppen für die Schulklasse anzulegen. Es wurde vereinbart, dass die Gruppen (hier die Klassenbezeichnung) mit Kleinbuchstaben angelegt werden. Dabei wird aber davon ausgegangen, dass die Klassenbezeichnungen keine Sonderzeichen enthält (nur A-Z und 0-9).

Mit dem Kommando `tr` lassen sich Zeichen umwandeln (siehe Abschnitt 5.4.6 auf Seite 104). Der allgemeine Aufruf lautet:

```
tr Zeichenkette1 Zeichenkette2
```

Dabei wird ein Zeichen aus *Zeichenkette1* durch das Zeichen an der entsprechenden Position aus *Zeichenkette2* ersetzt. Im nachfolgenden Beispiel wird der Buchstabe A durch 1 ersetzt, B mit 2 usw.

```
#!/bin/bash
echo ABCDEF | tr 'ABCDEF' '123455'    123455
```

Für bestimmte Zeichenfolgen sind schon entsprechende Kürzel definiert. Nachfolgend eine Auswahl:

- [:alnum:]** Zeichen und Zahlen
- [:blank:]** „white spaces“ (Leerzeichen, Tabulatoren, ...)
- [:digit:]** Zahlen
- [:lower:]** Kleinbuchstaben
- [:upper:]** Großbuchstaben

Im folgenden Beispiel werden in einem Datenstrom alle Großbuchstaben in Kleinbuchstaben umgewandelt:

```
#!/bin/bash
str="IT11SE"
klein=`echo $str | tr '[:upper:]' '[:lower:]'`
echo $str ' -> ' $klein    IT11SE -> it11se
```

Übung:

Erstellen Sie eine Umsetzregel für `tr`, so dass aus „verqueren“ Dateinamen (mit Leerzeichen, Sonderzeichen oder wilder Groß- und Kleinschreibung) neue Dateinamen entstehen, die nur Kleinbuchstaben, aber keine Sonder- und Leerzeichen enthalten.



4.2.6 Vorhandene Gruppen ermitteln

In der Schülerdatei kommt die Klasse mehrmals vor. Dies bedeutet, dass nicht einfach jeder Eintrag verwendet und so die Gruppe angelegt werden kann. Ist die Gruppe schon vorhanden, so wird die Fehlermeldung „addgroup: Die Gruppe »xxx« existiert bereits.“ ausgegeben. Es muss daher vor jedem „hinzufügen“ geprüft werden, ob diese Gruppe schon vorhanden ist.

Wie in Abschnitt II beschrieben (mehr dazu auf Seite 52), werden alle Gruppen in der Datei `/etc/group` verwaltet.

Mit dem Kommando `grep` lässt sich feststellen, ob die Gruppe schon vorhanden ist. Dabei wird der reguläre Ausdruck `^it11se:` verwendet. Dieser bedeutet: Suche am Zeilenanfang den Eintrag `it11se` gefolgt von einem Doppelpunkt.

Mehr zu `grep` und regulären Ausdrücken auf Seite 103.

```
#!/bin/bash
str="it11se"
echo $str
grep "^$str:" /etc/group >/dev/null \
    || echo "Gruppe kann angelegt werden"    it11se
                                           Gruppe kann angelegt werden
```

`grep` liefert als Rückgabewert 0 zurück, wenn der Eintrag gefunden wurde. Somit kann mit `||` die Gruppe angelegt werden, wenn diese nicht gefunden worden ist.

```
#!/bin/bash
echo "Klassen anlegen"
if [ -r sch.txt ]; then
    IFS=":"
    while read nachname vorname klasse ; do
        kl=`echo $klasse | tr '[:upper:]' '[:lower:]'`
        grep "^$kl:" /etc/group >/dev/null \
            || sudo groupadd $kl
    done < sch.txt
fi
```

Übung:

Passen Sie das kleine Skript so an, dass alle Gruppen (Klassen), die in der Schülerdatei vorhanden sind, angelegt werden.



4.3 Benutzer anlegen

Nachdem alle Gruppen angelegt sind, können die Benutzer angelegt werden. Dabei müssen folgende Schritte durchgeführt werden:

1. Einlesen der Schülerdatei – bereits erledigt
2. Generieren der Benutzernamen der Schüler (Schülerkennung)
3. Benutzer hinzufügen und der entsprechenden Klasse zuordnen
4. Liste mit Benutzernamen und Passwörtern erstellen

4.3.1 Generieren der Schülerkennung

Es gibt mehrere Möglichkeiten, die Kennung für einen Benutzer zu erstellen. Zwei Möglichkeiten werden dabei gerne verwendet:

1. sechs Buchstaben des Nachnamens und 2 Buchstaben des Vornamens (alles in Kleinbuchstaben)
aus *Hollmann, Michael* wird dann `hollmami`
2. voller Nachname + Punkt + voller Vorname (alles in Kleinbuchstaben)
aus *Hollmann, Michael* wird dann `hollmann.michael`

Ist die Kennung schon vorhanden, so kann man eine Zahl anhängen, um die Kennung wieder eindeutig zu machen. In unserem Beispiel kommt dies aber nicht vor, daher arbeiten wir nur mit der einfachen Variante.

Wir verwenden hier die Variante 2 mit Nachname + Punkt + Vorname.

Dazu passen wir das Skript für das Anlegen der Gruppen entsprechend an:

```
#!/bin/bash
echo "Benutzer anlegen"
if [ -r sch.txt ]; then
  IFS=":"
  while read nachname vorname klasse ; do
    kl=`echo $klasse | tr '[:upper:]' '[:lower:]'`
    ke=`echo $nachname.$vorname | tr '[:upper:]' '[:lower:]'`
    echo $ke
  done < sch.txt
fi
```

```
Benutzer anlegen
pyka.johannes
manfred.rolf
wanzinger.guido
```

Auch hier muss geprüft werden, ob der neu anzulegende Benutzer schon vorhanden ist. Dabei wird der selbe Mechanismus verwendet, wie bei der Gruppe.



Übung (für Gute):

Erstellen Sie ein Skript, welches bei doppelten Einträgen (z. B. Müller Peter kann mehrmals vorkommen) eine Kennung mit zusätzlicher fortlaufender Zahl erzeugt.

4.3.2 Passwörter generieren

Bis auf das Passwort lassen sich alle Werte mit unseren bisherigen Kenntnissen festlegen. Es gibt mehrere Möglichkeiten, ein Passwort zu generieren. Gerne nimmt man für das erste Passwort das Geburtsdatum oder einen zufälligen Wert. Das Geburtsdatum ist aber in der Schülerdatei nicht enthalten, daher entfällt diese Möglichkeit.

Zufallswerte lassen sich aus der virtuellen Datei `/dev/urandom` auslesen. Es werden hier binäre Werte zwischen 0 und 255 erzeugt, also auch solche Zeichen, die man nicht anzeigen kann, da diese kein sichtbares Zeichen darstellen. Solche Zeichen lassen sich auch im Normalfall nicht über die Tastatur eingeben. Mit dem Programm `tr` sorgen wir dafür, dass alle Zeichen, die nicht einer bestimmten Regel entsprechen, nicht verwendet werden. Das generierte Passwort soll nur Zeichen von A-Z (groß- und kleingeschrieben) und Zahlen enthalten. Die Länge soll dabei auf acht Zeichen beschränkt sein.

```
#!/bin/bash
echo "Passwort generieren"
pw=$(tr -cd a-zA-Z0-9 < /dev/urandom | head -c 8)
echo "PW: $pw"
```

```
Passwort generieren
PW: CdJpccTh
```

Das Programm `tr` liest dabei Zeichen aus der virtuellen Datei aus. Mit dem Parameter `-c` wird festgelegt, dass `tr` nur eine Zeichenkette verwendet. Der Parameter `-d` sorgt dafür, dass Zeichen nicht ausgetauscht, sondern gelöscht werden. Dies bedeutet, alle Zeichen, die nicht der Regel `a-zA-Z0-9` entsprechen, werden beim Lesen gelöscht und somit nicht an das Programm `head` weitergegeben. Mit `head` wird dafür gesorgt, dass aus dem Eingabestrom nur acht Zeichen (Parameter `-c`) gelesen werden.

Das Programm `useradd` erwartet das angegebene Passwort nicht im Klartext, sondern verschlüsselt. Um das Passwort entsprechend zu verschlüsseln, verwendet man das Programm `md5pass`, welches als Verschlüsselungsalgorithmus `md5` verwendet.

Message-Digest Algorithm 5

MD5 ist eine kryptographische Hashfunktion, die aus einem Text mit beliebiger Länge einen 128-Bit-Hashwert (deutsch: Prüfsumme) erzeugt. Linux verwendet dieses Verfahren, um aus einem Passwort einen verschlüsselten Eintrag zu erzeugen. Aus dem Eintrag lässt sich das Klartextpasswort nicht wieder rekonstruieren. Allerdings lassen sich Wörterbücher und generierte Klartextpasswörter in einen Hashwert umwandeln. Speichert man Klartextpasswort und Hashwert dann in einer Datenbank, so kann man das Passwort ermitteln, wenn man den Hashwert hat. Allerdings ist dieses Verfahren sehr aufwändig und der Aufwand steigt deutlich, wenn das Klartextpasswort entsprechend „kompliziert“ ist. Anstelle von MD5 kann man auch einen anderen Hash-Algorithmus, wie z. B. SHA-2 verwenden.

```
#!/bin/bash
echo "Passwort generieren"
pw=$(tr -cd a-zA-Z0-9 < /dev/urandom | head -c 8)
pwmd5=$(md5pass $pw)
echo "PW: $pw"
echo $pwmd5
```

```
Passwort generieren
PW: nljeuRCq
$1$MBVYgj85$irOpg750ysSHLb5k5/Eq61
```

4.3.3 Schülerkennung hinzufügen

Jetzt sind wir in der Lage, mit dem Kommando `useradd` die Schülerkennungen anzulegen (mehr dazu auf Seite 48).

```
useradd -m -U -G <Gruppen> -p <Passwort> -s <Shell> -c <Kommentar> <Kennung>
```

Die Parameter haben dabei folgende Bedeutung:

- m Home-Verzeichnis anlegen
- U es wird eine Gruppe mit dem selben Namen wie die Kennung angelegt
- G <Gruppen> der Anwender wird den angegebenen Gruppen zugeteilt (Hier wird die Klasse verwendet)
- p <Passwort> das verschlüsselte Passwort für die Kennung
- s <Shell> die zu verwendende Shell (wir verwenden die Bash)
- c <Kommentar> der Kommentar für die Kennung (Wir tragen hier den Namen und die Klasse ein)
- <Kennung> die eigentliche Kennung

Nun können die Schülerkennungen angelegt werden, indem das bisherige Skript angepasst und mit den obigen Beispielen ergänzt wird. Damit auch angezeigt wird, welche Kennung angelegt worden ist, wird zusätzlich die entsprechende Gruppe bzw. Kennung ausgegeben. Dazu ist es aber notwendig, die Kommandos nach `||` mit runden Klammern zu einer Gruppe zusammenzufassen, so dass diese als ein Kommando für `||` betrachtet werden.

```
#!/bin/bash
echo "Benutzer anlegen"
if [ -r sch.txt ]; then
IFS=":"
while read nachname vorname klasse ; do
kl=`echo $klasse | tr '[:upper:]' '[:lower:]'`
# Gruppe anlegen, wenn nicht vorhanden
grep "^$kl:" /etc/group >/dev/null \
|| ( sudo groupadd $kl; echo Gruppe: $kl )
ke=`echo $nachname.$vorname | tr '[:upper:]' '[:lower:]'`
# PW generieren
pw=$(tr -cd a-zA-Z0-9 < /dev/urandom | head -c 8)
pwmd5=$(md5pass $pw)
# Kennung anlegen, wenn nicht vorhanden
grep "^$ke:" /etc/passwd >/dev/null \
|| ( sudo useradd -m -U \
-G $kl \
-p $pwmd5 \
-s /bin/bash \
-c "$nachname, $vorname ($klasse)" \
$ke ; echo Kennung: $ke )
done < sch.txt
fi
```

```
Benutzer anlegen
Gruppe: it11fs
Kennung: pyka.johannes
Gruppe: it11fa
Kennung: manfred.rolf
Gruppe: it11se
Kennung: wanzinger.guido
```

Nach dem Ausführen des Skripts findet man folgende Einträge in der Gruppendatenbank `/etc/group`:

```
it11fs:x:1002:pyka.johannes
it11fa:x:1003:manfred.rolf
it11se:x:1004:wanzinger.guido
pyka.johannes:x:1005:
manfred.rolf:x:1006:
wanzinger.guido:x:1007:
```

Die IDs können hier abweichen, da diese davon abhängig sind, wie viele Gruppen schon vorher angelegt worden sind. Die Benutzerdatenbank `/etc/passwd` zeigt folgende zusätzliche Einträge.

```
pyka.johannes:x:1001:1005:Pyka, Johannes (IT11FS):/home/pyka.johannes:/bin/bash
manfred.rolf:x:1002:1006:Manfred, Rolf (IT11FA):/home/manfred.rolf:/bin/bash
wanzinger.guido:x:1003:1007:Wanzinger, Guido (IT11SE):/home/wanzinger.guido:/bin/bash
```

Die entsprechenden Verzeichnisse sind in `/home/` angelegt worden.

```
drwxr-xr-x 2 manfred.rolf manfred.rolf 4096 2012-04-06 11:02 manfred.rolf/
drwxr-xr-x 2 pyka.johannes pyka.johannes 4096 2012-04-06 11:02 pyka.johannes/
drwxr-xr-x 2 wanzinger.guido wanzinger.guido 4096 2012-04-06 11:02 wanzinger.guido/
```

Betrachtet man die Zugriffsrechte, so fällt folgendes auf:

1. Die Rechte für das Home-Verzeichnis sind so gesetzt, dass jeder darin lesen darf. Dies muss später noch geändert werden, so dass nur der Benutzer lesen und schreiben kann (mehr dazu auf Seite 58).
2. Es soll noch eine Liste (eine Datei) erstellt werden, in der alle Kennungen mit deren Passwort aufgelistet sind, so dass man auch den Schülern die erstellten Kennungen und deren Passwort mitteilen kann.
3. Da das Passwort auf einer Liste zu finden ist, muss dafür gesorgt werden, dass nach der ersten Anmeldung das Passwort geändert wird.



Übung:

1. Bauen Sie das obige Skript so um, dass zusätzlich eine Datei mit dem Namen `schuelerkennungen.txt` erstellt wird, die z. B. folgende Einträge enthält.

```
it11fs, manfred.rolf, vFRIPVKI
it11fa, pyka.johannes, asd23AW3
```

Verwenden Sie dazu die Umleitung des Befehles `echo`:

```
echo $kl, $ke, $pw >> schuelerkennungen.txt
```

2. Sortieren Sie dann diese Datei mit dem Kommando `sort`, so dass die Liste nach der Klasse sortiert ist.
3. Mit dem Programm `passwd` und dem Parameter `-e` kann festgelegt werden, dass das Passwort abgelaufen (`expire`) ist und es nach der nächsten Anmeldung geändert werden muss. Lesen Sie in der `man`-Page nach, wie dies funktioniert und bauen Sie diese Möglichkeit in obiges Skript ein.

4.3.4 Lehrerkennung hinzufügen

Nachdem alle Schülerkennungen erstellt worden sind, müssen noch die Lehrerkennungen erstellt werden.

Dazu sind folgende Schritte notwendig:

1. Eine Gruppe für die Lehrer einrichten (Grupper: `lehrer`).
2. Für die Lehrer Viola Lorenz und Lars Greiner entsprechende Kennungen erstellen.
3. Die Lehrer der Gruppe `lehrer` und allen Gruppen der Klassen hinzufügen.



Übung:

1. Erstellen Sie ein entsprechendes Skript, welches obige Schritte durchführt. Dabei sollen die Informationen aus entsprechenden Dateien, wie z. B. der Datei `schueler.txt` entnommen werden.
2. Sind die Gruppen und Kennungen einmal eingerichtet, so „greifen“ die erstellten Skripte kein zweites Mal. Erstellen Sie ein kleines Skript, welches die erstellten Kennungen und danach die erstellten Gruppen wieder löscht.

4.3.5 Superuser einrichten

Der Lehrer Herr Greiner ist der SysAdmin und soll die Möglichkeit haben, mit dem Programm `sudo` root-Zugriff zu erhalten.

Unter Ubuntu wird der Zugriff mit `sudo` über entsprechende Gruppenmitgliedschaften (`admin` und `sudo`) geregelt. Der entsprechende Ausschnitt in der Konfigurationsdatei zu `sudo` mit den Namen `/etc/sudoers` sieht dabei wie folgt aus:

```
Defaults            env_reset

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo  ALL=(ALL:ALL) ALL
```

Beantworten Sie mit Hilfe der `man`-Page und den Infoseiten zu `sudo` dazu folgende Fragen und sorgen Sie dann dafür, dass der Lehrer Herr Greiner als Superuser (ohne extra Passworteingabe) arbeiten kann, wenn er dies möchte.

1. Was wird mit dem Eintrag „Defaults“ festgelegt? Welche Möglichkeiten gibt es hier?

2. Welcher Unterschied besteht zwischen der Gruppe `sudo` und `admin`?

3. Welche Bedeutung hat `ALL=(ALL:ALL) ALL`?

4. Standardmäßig muss bei der Verwendung von `sudo` das eigene Passwort angegeben werden. Wie lautet der Eintrag, wenn für den Benutzer „hans“ kein Passwort angegeben werden soll?

5. Überlegen Sie, welche Gefahren bestehen, wenn ein Anwender das Kommando `sudo` ohne Passwort ausführen kann.

**Übung:**

1. Die Werte „Peter“, „Uschi“, „Hans“ und „Gabi“ sollen in einer for-Schleife ausgegeben werden. Erstellen Sie dazu ein Skript.
2. In einer for-Schleife sollen die Werte 1 bis 20 (Schrittweite 5) ausgegeben werden. Zeigen Sie zwei Beispiele.
3. Zeigen Sie zu Frage 2 eine Lösung mit einer while-Schleife.
4. Die Datei `ausgabe` soll zeilenweise mit einer while-Schleife ausgegeben werden. Erstellen Sie dazu ein Skript.
5. Die Datei `ausgabe` enthält Zeilen mit fünf Spalten, die jeweils durch das Zeichen „;“ getrennt sind. Die Spalten 2 und 4 sollen ausgegeben werden. Erstellen Sie dazu ein Skript.
6. Die Zeichenkette „Berliner Zoo“ soll in Großbuchstaben umgewandelt werden. Erstellen Sie dazu ein Skript.
7. Sie haben ein Passwort generiert und in der Variablen `pw` gespeichert. Das Passwort soll nun so verschlüsselt und in der Variablen `pwmd5` gespeichert werden, dass es für das Kommando `useradd` verwendet werden kann. Erstellen Sie dazu ein Skript.
8. Erstellen Sie ein kleines Skript, welches für alle Standard-User (ID \geq 1000) die Kennung und den Kommentar ausgibt.
9. Erweitern Sie das Skript aus 8 so, dass auch alle Gruppen (primär und sekundär) ausgegeben werden. Verwenden Sie dazu das Kommando `grep`, um die sekundären Gruppen zu ermitteln.

4.4 Fragen

1. Wie lautet die erste Zeile eines Bash-Shell-Skripts?

2. Wie werden die ersten beiden Zeichen der ersten Zeile eines Bash-Shell-Skripts bezeichnet?

3. Was muss getan werden, damit ein Skript direkt als Kommando von der Konsole aus aufgerufen werden kann?

4. Wie wird eine Variable definiert und wie wird diese ausgegeben?

5. Was versteht man unter Quoting?

6. Welche Ausgabe erhält man mit `echo "$HOME"` und mit `echo '$HOME'`?

7. Was wird von der Bash in den folgenden Variablen gespeichert?

\$0 _____

\$1 _____

\$# _____

\$* _____

\$@ _____

\$? _____

\$_ _____

\$\$ _____

8. Der Inhalt der Variablen `i` soll um den Wert 2 erhöht werden.

a) Zeigen Sie zwei Möglichkeiten mit der Bash.

b) Zeigen Sie zwei Möglichkeiten mit dem Aufruf eines externen Kommandos.

9. Wenn die Datei `ausgabe` lesbar ist, und nur dann, soll diese gelöscht werden.

a) Wie sieht der Code dazu aus, wenn `if` verwendet wird?

b) Wie sieht der Code dazu aus, wenn `test` verwendet wird?

10. Die Datei `ausgabe` soll zeilenweise mit einer `while`-Schleife ausgegeben werden. Erstellen Sie dazu ein Skript.

11. Wie stellen Sie fest, ob die Gruppe „lehrer“ schon vorhanden ist?

5 Einheit IV – Bilder vom „Stick“ verarbeiten

Das Arbeiten mit externen Medien wie USB-Stick, SD-Card, Smartphone und Co. gehört zum Alltag, gerade wenn man mit Bildern arbeitet. Die Klassenfahrtbilder sollen von den verschiedenen Medien eingebunden werden. Dies klappt nicht immer einwandfrei. Linux erkennt sehr viele Datenträger automatisch, ist aber der Stick defekt oder will er einfach nicht, so muss man selbst Hand anlegen.

Linux stellt hierzu sehr umfangreiche Informationen zur Verfügung, warum ein Stick etc. nicht funktioniert. Wird die Hardware nicht erkannt, ist der Stick nicht formatiert oder ist er defekt, weil er z. B. zu lange im Auto gelegen ist, das in der Sonne stand und zu heiß geworden ist.

Hat man sehr viele Bilder, gerade von unterschiedlichen Kameras und Personen gemacht, muss man diese zuerst in eine chronologische Reihenfolge bringen, was oft nicht so ganz einfach ist. Linux kann hier sehr einfach die zusätzlichen Informationen im Bild, welche die Kamera zusätzlich speichert, auswerten und so kann man dann die Bilder entsprechend sortieren.

Damit eine Bilder-Übersicht auch ansprechend gestaltet werden kann, kann man optische Spielereien mit ImageMagick einbauen.

Inhalt

5.1	Externe Medien	88
5.1.1	LOG-Meldungen	88
5.1.2	USB-Geräte anzeigen	91
5.1.3	Bilder kopieren	92
5.2	Medien prüfen, einhängen, formatieren,	96
5.2.1	Dateisysteme	96
5.2.2	Dateisysteme ein-/aushängen	96
5.2.3	Dateisysteme prüfen	97
5.2.4	Dateisystem formatieren	98
5.3	Bilddaten auswerten	99
5.4	Filter für Textdateien	101
5.4.1	Textdateien ausgeben	101
5.4.2	Textdateien sortieren	102
5.4.3	Zeilen von Textdateien auswählen	103
5.4.4	Textdateien in Spalten und Felder aufteilen	104
5.4.5	Zeilen und Wörter in Textdateien zählen	104
5.4.6	Zeichen austauschen	104
5.5	Bilder kopieren	107
5.5.1	Datum und Uhrzeit ermitteln	107
5.5.2	Datum und Uhrzeit formatieren	108
5.5.3	Bilder mit gleichem Datum	108
5.5.4	Bilder automatisch drehen	109
5.6	Thumbnails erzeugen	109
5.7	Spielereien mit Bildern	110
5.7.1	Rahmen um Bilder	110
5.7.2	Bilder beschneiden	111
5.8	Fragen	113

5.1 Externe Medien

Linux unterstützt eine Vielzahl von externen Medien, z. B. Floppy-Laufwerk, DVD-Laufwerk, USB-Stick, SD-Card, Speicher in Kameras oder Smartphones. Bei vielen Systemen, wie zum Beispiel bei Ubuntu, legt man das Medium (z. B. DVD) ein oder steckt den USB-Stick an und das System erkennt automatisch das neue Medium und bindet dies sofort ein, so dass der aktuelle Anwender damit arbeiten kann.

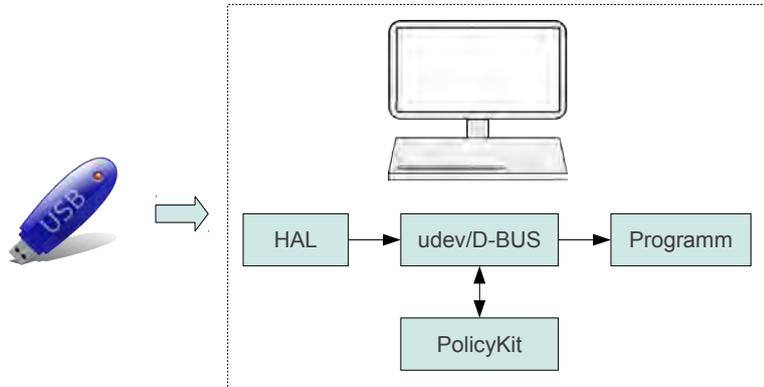


Abb. 5.1: Zusammenhang: USB-Stick-Programm

Wird der USB-Stick angesteckt, so reagiert der HAL (Hardware Abstraction Layer), der eine Zwischenschicht zwischen Hardware und Software bildet. Dies hat den Vorteil, dass die Software nicht speziell auf jede Hardware angepasst werden muss, sondern HAL als universeller Übersetzer arbeitet.

HAL seinerseits kooperiert mit `udev`¹ und `D-BUS`², sodass z. B. beim Anstecken des USB-Sticks sofort der Dateibrowser entsprechend geöffnet wird. Für die entsprechende Rechteverwaltung steht im Hintergrund ein `PolicyKit` zur Verfügung.

Als Benutzer bekommt man von dieser Kommunikation im Normalfall nichts mit, es sei denn, dass es Probleme mit dem USB-Stick gibt, falls dieser evtl. defekt ist.

5.1.1 LOG-Meldungen

Beim Einstecken des USB-Sticks werden entsprechende LOG-Meldungen geschrieben. LOG-Meldungen werden jeweils in unterschiedlichen Dateien (je nachdem, von wem die LOG-Meldungen stammen) im Verzeichnis `/var/log/` gespeichert.

Beispiele

`/var/log/syslog` Hier landen die normalen Systemmeldungen. Für die Verwaltung und Speicherung ist der Syslog-Daemon (`syslogd`) zuständig.

`/var/log/kern.log` Hier landen Meldungen des Kernels.

`/var/log/dmesg` Hier landen Meldungen des Kernels beim Booten. Diese Informationen werden erst in die Datei geschrieben, wenn auch das entsprechende Dateisystem eingebunden ist. Vorher kann man diese Meldungen mit dem Kommando `dmesg` betrachten. Dies ist vor allem dann wichtig, wenn beim Systemstart ein Fehler auftritt und das Dateisystem nicht eingebunden werden kann. Mit `dmesg` erhält man diese wichtigen Meldungen und kann meist schnell die eigentliche Fehlerursache finden.

`/var/log/auth.log` Hier landen Meldungen im Bezug auf Anmeldungen am System.

`/var/log/udev` Hier landen Meldungen von `udev`.

...

Konfiguriert wird das LOG-System über die Datei `/etc/rsyslog.conf` und die Dateien im Verzeichnis `/etc/rsyslog.d/`.

Übung:

Ermitteln Sie, was das folgende Skript macht:

```
sudo find /var/log -type f -exec wc -l {} \; | sort -nr | head
```

¹ `udev` ist ein Programm zur Überwachung und Auswertung von Hardware-Ereignissen für plugable Devices (Geräte, die zur Laufzeit angeschlossen und wieder entfernt werden können). Bekommt `udev` Informationen über ein neues Gerät, wertet es diese anhand konfigurierbarer Regeln aus und gibt dem Gerät einen Namen.

² `D-Bus` ist ein Software-System für die Interprozesskommunikation (IPC, welches insbesondere bei Desktopumgebungen verwendet wird). Über `D-Bus` können Programme miteinander kommunizieren, z. B. dass ein USB-Stick angeschlossen worden ist und nun der Dateibrowser geöffnet werden soll oder dass eine Audio-CD eingelegt worden ist und automatisch der Musik-Player startet.

dmesg-Meldung beim Booten

Die Meldungen beim Booten geben viele nützliche Informationen über den Bootvorgang, die Hardware und welche Treiber wie geladen werden. Wenn das System so funktioniert, wie man sich dies vorstellt, braucht man diese Informationen nicht, aber bei Fehlern oder Nichtfunktionieren eines Bereiches, können sie sehr hilfreich sein.

Hier ein Auszug mit Kommentaren:³

```
[ 0.000000] Initializing cgroup subsys cpuset
[ 0.000000] Initializing cgroup subsys cpu
[ 0.000000] Linux version 3.0.0-17-generic (buildd@yellow) (gcc version 4.6.1 (Ubuntu/Linaro 4.6.1-9ubuntu3) ) )
#30-Ubuntu SMP Thu Mar 8 20:45:39 UTC 2012 (Ubuntu 3.0.0-17.30-generic 3.0.22)
Der Kernel
[ 0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-3.0.0-17-generic root=UUID=1ac132a9-6bb5-40eb-92cb-2
a02ea616d320 ro quiet splash vt.handoff=7
Kernel-Parameter
[ 0.000000] BIOS-provided physical RAM map:
[ 0.000000] BIOS-e820: 0000000000000000 - 000000000009f800 (usable)
[ 0.000000] BIOS-e820: 000000000009f800 - 00000000000a0000 (reserved)
[ 0.000000] BIOS-e820: 00000000000e8000 - 0000000000100000 (reserved)
[ 0.000000] BIOS-e820: 0000000001000000 - 00000000dfba1640 (usable)
[ 0.000000] BIOS-e820: 00000000dfba3640 - 00000000dfba36a0 (ACPI NVS)
[ 0.000000] BIOS-e820: 00000000dfba36a0 - 00000000e0000000 (reserved)
[ 0.000000] BIOS-e820: 00000000f4000000 - 00000000f8000000 (reserved)
[ 0.000000] BIOS-e820: 00000000fec00000 - 00000000fed40000 (reserved)
[ 0.000000] BIOS-e820: 00000000fed45000 - 0000000100000000 (reserved)
[ 0.000000] BIOS-e820: 0000000100000000 - 000000041c000000 (usable)
Speicher, der vom BIOS gefunden worden ist
[ 0.000000] DMI: Hewlett-Packard HP Compaq 8000 Elite CMT PC/3647h, BIOS 786G7 v01.02 10/22/2009
Rechnername
[ 0.069731] CPU0: Intel(R) Core(TM)2 Quad CPU Q8400 @ 2.66GHz stepping 0a
CPU
[ 0.344038] Brought up 4 CPUs
[ 0.344041] Total of 4 processors activated (21278.92 BogoMIPS).
vier Kerne aktiviert
[ 1.314277] e1000e: Intel(R) PRO/1000 Network Driver - 1.3.10-k2
gefundene Netzwerkkarte
[ 1.832145] ata5.00: ATA-8: WDC WD20EARX-00PASB0, 51.0AB51, max UDMA/133
SATA Festplatte
[ 1.847780] ata3.00: ATA-8: Corsair Force 3 SSD, 1.3.2, max UDMA/133
SATA SSD
[ 1.873311] scsi 2:0:0:0: Direct-Access ATA Corsair Force 3 1.3. PQ: 0 ANSI: 5
[ 1.873503] sd 2:0:0:0: Attached scsi generic sg0 type 0
[ 1.873533] sd 2:0:0:0: [sda] 234441648 512-byte logical blocks: (120 GB/111 GiB)
[ 1.873593] sd 2:0:0:0: [sda] Write Protect is off
[ 1.873595] sd 2:0:0:0: [sda] Mode Sense: 00 3a 00 00
[ 1.873615] sd 2:0:0:0: [sda] Write cache: enabled, read cache: enabled, doesn't support DPO or FUA
[ 1.874071] sda: sda1 sda2
Einbindung SSD
[ 1.876863] scsi 3:0:0:0: CD-ROM hp DVD-RAM GH60L RD01 PQ: 0 ANSI: 5
[ 1.885228] sr0: scsi3-mmc drive: 40x/40x writer dvd-ram cd/rw xa/form2 cdda tray
[ 1.885232] cdrom: Uniform CD-ROM driver Revision: 3.20
[ 1.885325] sr 3:0:0:0: Attached scsi CD-ROM sr0
Einbindung DVD
[ 1.885514] scsi 4:0:0:0: Direct-Access ATA WDC WD20EARX-00P 51.0 PQ: 0 ANSI: 5
[ 1.885633] sd 4:0:0:0: [sdb] 3907029168 512-byte logical blocks: (2.00 TB/1.81 TiB)
[ 1.885635] sd 4:0:0:0: [sdb] 4096-byte physical blocks
Einbindung Festplatte 1
[ 1.885756] scsi 5:0:0:0: Direct-Access ATA WDC WD20EARX-00P 51.0 PQ: 0 ANSI: 5
[ 1.885859] sd 5:0:0:0: [sdc] 3907029168 512-byte logical blocks: (2.00 TB/1.81 TiB)
[ 1.885862] sd 5:0:0:0: [sdc] 4096-byte physical blocks
Einbindung Festplatte 2
[ 2.041003] md/raid0:md0: looking at sdc1
[ 2.041005] md/raid0:md0: comparing sdc1(3907021824) with sdc1(3907021824)
[ 2.041011] md/raid0:md0: looking at sdb1
[ 2.041013] md/raid0:md0: comparing sdb1(3907021824) with sdc1(3907021824)
RAID 0 für die zwei Festplatten
[ 9.585517] nvidia 0000:01:00.0: PCI INT A -> GSI 16 (level, low) -> IRQ 16
[ 9.585525] nvidia 0000:01:00.0: setting latency timer to 64
[ 9.585529] vgaarb: device changed decodes: PCI:0000:01:00.0,olddecodes=io+mem,decodes=none:owns=io+mem
[ 9.585671] NVRM: loading NVIDIA UNIX x86_64 Kernel Module 280.13 Wed Jul 27 16:53:56 PDT 2011
Grafikkarte
...
```

Übung:

Ermitteln Sie mit Hilfe von dmesg die wichtigsten Hardwaredaten Ihres Rechners.



³ Der Wert in eckigen Klammern ist die Zeit in Sekunden seit dem Systemstart.

Meldungen beim Einstecken eines USB-Sticks

Damit man sofort die Meldungen (z. B. beim Einstecken eines USB-Sticks) sehen kann, öffnet man ein Terminal und startet das Kommando `tail` mit dem Parameter `-f`, was dafür sorgt, dass der Inhalt der Datei fortlaufend angezeigt wird. Wird eine neue LOG-Meldung geschrieben, so wird diese sofort angezeigt:

```
1 tail -f /var/log/syslog
2 Apr 19 17:48:47 video kernel: [28621.335465] usb 2-1.7: new high speed USB device number 6 using ehci_hcd
3 Apr 19 17:48:47 video kernel: [28621.429996] scsi7 : usb-storage 2-1.7:1.0
4 Apr 19 17:48:47 video mtp-probe: checking bus 2, device 6: "/sys/devices/pci0000:00/0000:00:1d.0/usb2/2-1/2-1.7"
5 Apr 19 17:48:47 video mtp-probe: bus: 2, device: 6 was not an MTP device
6 Apr 19 17:48:48 video kernel: [28622.425091] scsi 7:0:0:0: Direct-Access          JetFlash Transcend 16GB      8.07 PQ: 0 ↵
   ANSI: 2
7 Apr 19 17:48:48 video kernel: [28622.727099] sd 7:0:0:0: Attached scsi generic sg3 type 0
8 Apr 19 17:48:48 video kernel: [28622.728375] sd 7:0:0:0: [sdc] 31375360 512-byte logical blocks: (16.0 GB/14.9 GiB)
9 Apr 19 17:48:48 video kernel: [28622.729368] sd 7:0:0:0: [sdc] Write Protect is off
10 Apr 19 17:48:48 video kernel: [28622.729373] sd 7:0:0:0: [sdc] Mode Sense: 03 00 00 00
11 Apr 19 17:48:48 video kernel: [28622.729943] sd 7:0:0:0: [sdc] No Caching mode page present
12 Apr 19 17:48:48 video kernel: [28622.729948] sd 7:0:0:0: [sdc] Assuming drive cache: write through
13 Apr 19 17:48:48 video kernel: [28622.732557] sd 7:0:0:0: [sdc] No Caching mode page present
14 Apr 19 17:48:48 video kernel: [28622.732562] sd 7:0:0:0: [sdc] Assuming drive cache: write through
15 Apr 19 17:48:48 video kernel: [28622.882479]          sdc: sdc1
16 Apr 19 17:48:48 video kernel: [28622.884530] sd 7:0:0:0: [sdc] No Caching mode page present
17 Apr 19 17:48:48 video kernel: [28622.884536] sd 7:0:0:0: [sdc] Assuming drive cache: write through
18 Apr 19 17:48:48 video kernel: [28622.884540] sd 7:0:0:0: [sdc] Attached SCSI removable disk
19 Apr 19 17:48:49 video ntfs-3g[7390]: Version 2011.4.12AR.4 external FUSE 28
20 Apr 19 17:48:49 video ntfs-3g[7390]: Mounted /dev/sdc1 (Read-Write, label "usb_stick", NTFS 3.0)
21 Apr 19 17:48:49 video ntfs-3g[7390]: Cmdline options: rw,nosuid,nodev,uhelper=udisks,uid=1000,gid=1000,dmask=0077,↵
   fmask=0177
22 Apr 19 17:48:49 video ntfs-3g[7390]: Mount options: rw,nosuid,nodev,uhelper=udisks,allow_other,nonempty,relatime,↵
   fsname=/dev/sdc1,blkdev,blksize=4096,default_permissions
23 Apr 19 17:48:49 video ntfs-3g[7390]: Global ownership and permissions enforced, configuration type 7
```

Was passiert hier:

Zeile: 2 Der USB-Treiber meldet dem Kernel, dass er ein neues USB-Gerät (ehci -> USB 2) gefunden hat.

Zeile: 3 Bei dem USB-Gerät handelt es sich um ein Speichermedium (usb-storage).

Zeile: 6 Der USB-Stick ist ein „JetFlash Transcend 16GB“.

Zeile: 8 Der Speicher-Stick hat dabei eine Blockgröße von 512 Bytes und es können netto 14.9 GiB darauf gespeichert werden.

Zeile: 15 Der USB-Stick wird über den Gerätenamen `/dev/sdc` angesprochen. Da der Stick nur eine Partition hat, wird diese mit `/dev/sdc1` angesprochen.

Zeile: 19 Es wurde erkannt, dass der Stick mit NTFS formatiert ist und der entsprechende Treiber wird geladen.

Zeile: 20 Der Stick hat den Namen „usb_stick“. Somit wird er beim Einbinden automatisch unter `/media/usb_stick/` eingebunden.

Zeile: 21 Die entsprechenden Optionen für das Kommando `mount` werden gesetzt.



Bei Ubuntu findet sich jetzt das USB-Stick-Icon in der Starterleiste auf der linken Seite. Mit einem Klick wird dann der entsprechende Dateibrowser geöffnet, wenn dieser nicht sowieso (je nach Einstellung) automatisch startet.

Ruft man mit -Klick das Kontextmenü auf und wählt „Sicher entfernen“, so wird das Dateisystem sicher ausgehängt, was bedeutet, dass vorher alle Daten, die sich noch im Write-Cache befinden, auf den Stick geschrieben werden. Im LOG findet man dann folgende Einträge.

```
Apr 19 18:33:59 video kernel: [31324.637429] usb 2-1.7: reset high speed USB device number 7 using ehci_hcd
Apr 19 18:33:59 video kernel: [31324.745228] usb 2-1.7: USB disconnect, device number 7
```

Übung:

- Legen Sie nacheinander eine DVD und eine Audio-CD in das DVD-Laufwerk ein und betrachten Sie jeweils die LOG-Datei.
- Schließen Sie nun eine Kamera per USB oder eine Kamera-SD-Card über einen Adapter an und betrachten Sie erneut die LOG-Datei.

Achten Sie darauf, welches Programm nach dem Einlegen automatisch gestartet wird.

5.1.2 USB-Geräte anzeigen

Will man alle USB-Geräte, die angesteckt sind, anzeigen lassen, so hilft das Kommando `lsusb`. Verwendet man zusätzlich den Parameter `-v` bzw. `-vv`, so werden deutlich mehr Informationen angezeigt.

```
lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 005 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 006 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 007 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 008 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 009 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 010 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 002 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 002 Device 003: ID 05e3:0608 Genesys Logic, Inc. USB-2.0 4-Port HUB
Bus 002 Device 004: ID 04f3:0234 Elan Microelectronics Corp.
Bus 002 Device 005: ID 04d9:1603 Holtek Semiconductor, Inc. Keyboard
Bus 003 Device 003: ID 9022:d660 TeVii Technology Ltd. DVB-S2 S660
Bus 004 Device 003: ID 9022:d660 TeVii Technology Ltd. DVB-S2 S660
Bus 002 Device 006: ID 0bb4:0ff9 High Tech Computer Corp. Desire / Desire HD / Hero (Charge Mode)
Bus 002 Device 007: ID 058f:6387 Alcor Micro Corp. Transcend JetFlash Flash Drive
```

Dabei wird der Bus angezeigt. Normalerweise hat ein PC mehrere USB-Anschlüsse, die über verschiedene Busse angesprochen werden. Auch können diese unterschiedliche Geschwindigkeiten unterstützen, z. B. USB-1, USB-2 oder neu auch USB-3. Die ID (z. B. `058f:6387`) identifiziert den Hersteller und somit die Hardware genau. Es gibt diverse Datenbanken im Internet, die mehr Auskünfte darüber geben.⁴

Mehr Hardwareinformationen

Sollen möglichst alle Hardwareinformationen des Systems angezeigt werden, so hilft das Kommando `lshw`. Dabei werden alle Pseudo-Dateisysteme wie `/proc` oder `/sys` vom Kernel durchsucht und so die Informationen in verschiedenen Gruppen angezeigt. Mit dem Kommando `lshw-gtk` steht auch eine grafische Oberfläche zur Verfügung.



Übung:

Untersuchen Sie die verwendeten USB-Geräte an Ihrem Rechnersystem.



⁴ siehe zum Beispiel <http://usb-ids.gowdy.us/read/UD/058f/6387>

5.1.3 Bilder kopieren

In nachfolgendem Beispiel wird eine Compact-Flash-Karte (CF-Card) mit 1 GB Speichergröße über einen Adapter angeschlossen.

```
Apr 19 19:02:46 video kernel: [33046.017118] usb 2-1.7: USB disconnect, device number 9
Apr 19 19:02:54 video kernel: [33053.615512] usb 2-1.7: new high speed USB device number 10 using ehci_hcd
Apr 19 19:02:54 video kernel: [33053.710967] scsi11 : usb-storage 2-1.7:1.0
Apr 19 19:02:54 video mtp-probe: checking bus 2, device 10: "/sys/devices/pci0000:00/0000:00:1d.0/usb2/2-1/2-1.7"
Apr 19 19:02:54 video mtp-probe: bus: 2, device: 10 was not an MTP device
Apr 19 19:02:55 video kernel: [33054.897268] scsi 11:0:0:0: Direct-Access Generic          CF 1.6E PQ: 0 ANSI: 0 CCS
Apr 19 19:02:55 video kernel: [33054.898729] scsi 11:0:0:1: Direct-Access Generic          MS 1.6E PQ: 0 ANSI: 0 CCS
Apr 19 19:02:55 video kernel: [33054.900084] scsi 11:0:0:2: Direct-Access Generic          MMC/SD 1.6E PQ: 0 ANSI: 0 CCS
Apr 19 19:02:55 video kernel: [33054.901422] scsi 11:0:0:3: Direct-Access Generic          SM 1.6E PQ: 0 ANSI: 0 CCS
...
Apr 19 19:02:55 video kernel: [33054.939668] sd 11:0:0:0: [sd] 2000880 512-byte logical blocks: (1.02 GB/976 MiB)
...
Apr 19 19:02:55 video kernel: [33054.961111]   sdc: sdc1
...
```

Die Karte wird über `/media/CANON_DC/` automatisch eingebunden.⁵

```
ls -la /media/CANON_DC/
drwx----- 5 mgn mgn 16384 2011-09-20 10:36 DCIM
```

Foto-Speicherkarten legen auf der Karte meist ein Verzeichnis mit dem Namen DCIM (Digital Camera Images) an. In diesem befinden sich eventuell mehrere Unterverzeichnisse, in denen die Bilder gespeichert werden. Diese haben das Namensschema: drei Ziffern + fünf Zeichen, z. B. 186CANON. Darin sind dann die Bilder enthalten, meist 100 Bilder pro Verzeichnis.

```
ls -la /media/CANON_DC/DCIM/186CANON
-rw-r--r-- 1 mgn mgn 3085238 2011-11-08 12:30 IMG_8691.JPG
-rw-r--r-- 1 mgn mgn 2169427 2011-11-08 12:30 IMG_8692.JPG
-rw-r--r-- 1 mgn mgn 4399074 2011-12-05 14:52 IMG_8693.JPG
-rw-r--r-- 1 mgn mgn 2448077 2011-12-05 15:13 IMG_8694.JPG
-rw-r--r-- 1 mgn mgn 2673685 2011-12-05 15:25 IMG_8695.JPG
-rw-r--r-- 1 mgn mgn 2880926 2011-12-05 15:44 IMG_8697.JPG
-rw-r--r-- 1 mgn mgn 2960835 2011-12-05 16:12 IMG_8698.JPG
-rw-r--r-- 1 mgn mgn 3283803 2011-12-05 16:12 IMG_8699.JPG
-rw-r--r-- 1 mgn mgn 2614610 2011-12-05 16:25 IMG_8700.JPG
ls -la /media/CANON_DC/DCIM/187CANON/
-rw-r--r-- 1 mgn mgn 2800427 2011-12-05 16:27 IMG_8701.JPG
-rw-r--r-- 1 mgn mgn 2995669 2011-12-05 16:28 IMG_8702.JPG
...
```

Nun können die Bilder z. B. mit `cp` kopiert werden. Sind es allerdings einige hundert oder mehr Bilder, so kann dies schon einiges an Handarbeit bedeuten, da hier viele Unterverzeichnisse verwendet werden.

Dateien finden

Mit dem Kommando `find` lassen sich Dateien nach einem bestimmten Schema suchen und anzeigen oder ein Kommando für diese Datei ausführen.

```
man find
FIND(1)                                Dienstprogramme für Benutzer                                FIND(1)

BEZEICHNUNG
  find - sucht in einer Verzeichnishierarchie nach Dateien

  Diese Handbuchseite ist eventuell veraltet. Im Zweifelsfall ziehen Sie die englischsprachige Handbuchseite zu Rate, indem Sie

      man -LC 1 find

  eingeben.

SYNTAX
  find [-H] [-L] [-P] [Verzeichnis...] [Ausdruck]

BESCHREIBUNG
  Diese Handbuchseite beschreibt die GNU Version von find. GNU find durchsucht eine oder mehrere - Verzeichnisse hierarchien und bewertet dabei den Ausdruck von links nach rechts entsprechend den Regeln der Priorität (siehe Abschnitt OPERATOREN) bis das Ergebnis bekannt ist (die linke Seite ist falsch für UND Operatoren, wahr für ODER Operatoren). Anschließend führt find die Suche mit der nächsten Datei fort.
...
```

⁵ Es wird hier eine Canon PowerShot verwendet.

Beispiel

```
find /media/CANON_DC/DCIM -name '*.JPG' -print
/media/CANON_DC/DCIM/186CANON/IMG_8691.JPG
/media/CANON_DC/DCIM/186CANON/IMG_8692.JPG
/media/CANON_DC/DCIM/186CANON/IMG_8693.JPG
/media/CANON_DC/DCIM/186CANON/IMG_8694.JPG
/media/CANON_DC/DCIM/186CANON/IMG_8695.JPG
...
/media/CANON_DC/DCIM/187CANON/IMG_8701.JPG
/media/CANON_DC/DCIM/187CANON/IMG_8702.JPG
...
```

Es werden im Verzeichnis `/media/CANON_DC/DCIM` und allen darin enthaltenen Unterverzeichnissen (rekursiv) alle Dateien angezeigt (`-print`), die dem regulären Ausdruck `*.JPG` entsprechen. Werden Wildcards verwendet, so muss die Zeichenkette in einfache Hochkommas gesetzt werden, da sonst die Shell die Wildcards auflöst.

Einige Optionen von `find`:

```
-name <Ausdruck>   Der Ausdruck, nach dem die Dateinamen ermittelt werden.
-print             Es wird der vollständige Pfadnamen der Datei ausgegeben.
-maxdepth <Ebenen> Es werden im Verzeichnisbaum nur maximal x Ebenen rekursiv durchlaufen.
-exec <Kommando>  Führt das Kommando aus. Evtl. Argumente werden direkt danach angegeben, bis ein Semikolon
                  \; folgt (dieser muss gequotet werden, da die Shell es sonst als Befehlstrenner interpretiert).
                  Der Platzhalter für die Datei ist hier {}.
```

Nun können die Bilder kopiert werden.

```
find /media/CANON_DC/DCIM -name '*.JPG' -print -exec cp {} . \;
/media/CANON_DC/DCIM/186CANON/IMG_8691.JPG
/media/CANON_DC/DCIM/186CANON/IMG_8692.JPG
...
```

Kopiert man die Bilder von vielen Kameras, so ergeben sich evtl. folgende Probleme:

- Der Bilddateiname, vor allem mit der Nummer, kommt bei mehreren Kameras öfters vor.
- Man weiß jetzt nicht mehr, von wem das Bild ist.
- Eine zeitliche Reihenfolge lässt sich aus dem Dateinamen nicht erkennen.

Viele Argumente

Mit `find` lassen sich über `-exec` weitere Kommandos aufrufen und die gefundenen Parameter werden dabei an das aufzurufende Kommando weitergegeben. Findet `find` sehr viele Dateien, so wird für jede gefundene Datei ein eigener Prozess gestartet.

Beispiel: Alle Dateien des Anwenders `killer` sollen gelöscht werden.

```
find / -user killer -exec rm {} \;
```

Findet das Kommando `find` z. B. 1.000 Dateien, so wird `rm` 1.000 mal aufgerufen. Dies dauert sehr lange.

Abhilfe schafft hier das Kommando `xargs`.

```
find / -user killer -print | xargs rm
```

Das Kommando `find` listet alle Dateien des Anwenders `killer` auf und reicht sie über die Pipe an `xargs` weiter. Dabei nimmt `xargs` die Liste von der Standardeingabe und erzeugt daraus ein `rm`-Kommando mit 1.000 Argumenten. Somit werden nur drei Prozesse (`find`, `xargs` und `rm`) gestartet. Wichtig dabei ist, dass die maximale Größe der Kommandozeile nicht überschritten wird.

Nachfolgendes Beispiel ermittelt die Größe:

```
getconf ARG_MAX                                2097152
```

Sind Leerzeichen im Dateinamen enthalten und werden als Parameter übergeben, so betrachtet die Bash diese als Parametertrenner. Daher muss man `find` und `xargs` mitteilen, dass beide Programme sogenannte „ASCII-NUL beendete Zeichenketten“ verwenden sollen.

```
find / -user killer -print0 | xargs -0 rm
```

**Übung:**

1. Ermitteln Sie mit Hilfe der man-Page zu `find` die Funktionen der folgenden Argumente und geben Sie jeweils ein Beispiel an:

`-type` _____

`-user` _____

`-group` _____

`-size` _____

`-atime` _____

`-mtime` _____

`-ctime` _____

`-perm` _____

`-iname` _____

`!` _____

`-a` _____

`-o` _____

2. Finden Sie alle Dateien, die in den letzten 24 Stunden geändert worden sind.

3. Finden Sie alle Dateien, die größer als 1 MiB sind und Ihnen gehören.

Dateien mit locate / slocate finden

Das Kommando `find` muss immer das entsprechende Verzeichnis durchsuchen, was bei vielen Dateien sehr lange dauern kann. `locate` verwendet einen anderen Ansatz. Zum Beispiel wird einmal pro Tag der gesamte Verzeichnisbaum durchsucht und jeder gefundene Dateiname wird in einer Datenbank gespeichert (mit dem Kommando `updatedb`). So braucht `locate` nur in der Datenbank zu suchen und findet so sehr viel schneller die entsprechende Datei. `slocate` prüft zusätzlich noch Eigentümer, Gruppe und Zugriffsrechte, so dass nur Dateien angezeigt werden, auf die der Anwender auch Zugriff hat.

`locate` hat aber auch Nachteile: Es werden nur Dateien angezeigt, die vor dem Aufruf von `updatedb` im System waren. Später erstellte Dateien werden nicht gefunden. Dies bedeutet aber auch, dass `locate` bei Wechselmedien, wie unserem USB-Stick, nicht funktioniert, da dieser sehr wahrscheinlich beim Aufruf von `updatedb` nicht eingesteckt war.

Programme mit which / whereis finden

Soll ein externes Programm bzw. Kommando gesucht werden, so sucht die Shell in allen Verzeichnissen, die in der Variablen `PATH` eingetragen sind:

echo \$PATH

```
/usr/local/texlive/2011/bin/x86_64-linux:/usr/local/jdk/bin:
/usr/local/texlive/2011/bin/x86_64-linux:/home/mgn/bin:/usr/local/sbin:
/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

Wollen Sie wissen, in welchem dieser Verzeichnisse das Kommando `find` zu finden ist, so kann man es mit `which` anzeigen lassen.

which find

```
/usr/bin/find
```

Dabei beginnt `which` beim ersten Verzeichnis in `PATH` und sucht die Datei, wird diese nicht gefunden wird beim nächsten Eintrag weiter gemacht usw., bis die Datei gefunden wird.

Das Kommando `whereis` zeigt nicht nur externe Kommandos an, sondern auch Dokumentationsdateien (Manpages), Quellcodedateien usw., die etwas mit dem Kommando zu tun haben.

whereis userdel

```
userdel: /usr/sbin/userdel /usr/share/man/man8/userdel.8.gz
```

Übung:

Für die weiteren Übungen benötigen wir entsprechende Verzeichnisse, um die Bilder der Kameras zu kopieren.

1. Legen Sie dazu folgende Verzeichnisse an, passen Sie aber dazu die Namen auf Ihre Kamera/Smartphone an.

```
/var/bilder/klassenfahrt/IT11FS/raw/canon
/var/bilder/klassenfahrt/IT11FS/raw/sony
/var/bilder/klassenfahrt/IT11FS/raw/iphone-klaus
/var/bilder/klassenfahrt/IT11FS/raw/iphone-stefan
...
```

2. Versetzen Sie das Klassenverzeichnis mit entsprechenden Rechten, so dass alle Schüler der Klasse Lese- und Schreib-Zugriff haben
3. und alle darin erzeugten Dateien und Verzeichnisse mit der Gruppe der Klasse versehen werden.



5.2 Medien prüfen, einhängen, formatieren, ...

Bis jetzt wurde der USB-Stick vom System erkannt und automatisch eingehängt. Es kann aber durchaus sein, dass dieser nicht eingehängt wird oder das verwendete Dateisystem nicht erkannt wird oder der Stick sogar defekt ist.

5.2.1 Dateisysteme

Bevor man ein Medium prüfen kann, ist es sinnvoll zu wissen, welches Dateisystem bei der Formatierung verwendet worden ist. Bei sehr vielen Sticks und Kamerakarten wird z. B. das Dateisystem FAT verwendet. Tabelle 5.1 zeigt eine Übersicht über gängige Dateisysteme und Ihren Einsatz.

Dateisystem	Linux	Windows	für ext. Medien	Beschreibung
FAT12	✓	✓	✓	für MS-DOS-Disketten
FAT16	✓	✓	✓	für kleine Festplatten und USB-Sticks bis 2 GiB
FAT32	✓	✓	✓	für Festplatten und größere USB-Sticks bis 2 TiB
VFAT	✓	✓	✓	Erweiterung eines FAT-Systems mit langen Dateinamen
exFAT	✗	✓	✓	für Flash-Speicher bis 63 ZiB
NTFS	✓	✓	✓	für Festplatten und größere USB-Sticks bis 256 TiB (mit Journal)
ext2	✓	✗(✓)	✗	für Festplatten bis 16 TiB
ext3	✓	✗	✗	für Festplatten bis 32 TiB (mit Journal)
ext4	✓	✗	✗	für Festplatten bis 1 EiB (mit Journal)
btrfs	✓	✗	✗	für Festplatten bis 16 EiB (mit Journal)
xfs	✓	✗	✗	für Festplatten bis 8 EiB (mit Journal)

Tab. 5.1: Übersicht einiger Dateisysteme

Journaling-Dateisystem

Ein Journaling-Dateisystem ist ein Dateisystem, welches alle Änderungen vor dem eigentlichen Schreiben der Datei in einem dafür besonderen Speicherbereich, dem Journal, aufzeichnet. Dabei werden nur die Änderungen geschrieben. Der eigentliche Dateiinhalt wird erst später geschrieben und möglichst lange im Cache-Speicher gehalten, um eine höhere Performance zu erreichen. Bei einem Stromausfall kann dann der Dateiinhalt aus dem Journal wieder rekonstruiert werden. Man vermeidet so einen inkonsistenten Zustand, d. h. eine fehlerhafte Datei.

5.2.2 Dateisysteme ein-/aushängen

Um ein Dateisystem einzuhängen, wird das Kommando `mount` verwendet. Im Normalfall darf dies direkt nur von `root` aufgerufen werden.

```
mount [-fnrsvw] [-t vfstype] [-o options] device dir
```

Beispiel

Es wird das Laufwerk (hier ein USB-Stick) in das allgemeine Verzeichnis `/mnt/` eingehängt. Dabei versucht `mount` eigenständig das Dateisystem zu ermitteln und ruft den entsprechenden Treiber auf. Kann das Dateisystem nicht automatisch erkannt werden, so muss es mit dem Parameter `-t` angegeben werden.

```
sudo mount /dev/sdvl /mnt
```

Ruft man `mount` ohne Parameter auf, so werden alle eingehängten Dateisysteme angezeigt. Dabei werden nicht nur „echte“ Dateisysteme angezeigt, sondern auch virtuelle Dateisysteme wie `proc` oder `sysfs`.

```
mount
/dev/sd1 on / type ext4 (rw,noatime,nodiratime,errors=remount-ro,commit=0)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
fusectl on /sys/fs/fuse/connections type fusectl (rw)
none on /sys/kernel/debug type debugfs (rw)
none on /sys/kernel/security type securityfs (rw)
udev on /dev type devtmpfs (rw,mode=0755)
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=0620)
```

```
tmpfs on /run type tmpfs (rw,noexec,nosuid,size=10%,mode=0755)
none on /run/lock type tmpfs (rw,noexec,nosuid,nodev,size=5242880)
none on /run/shm type tmpfs (rw,nosuid,nodev)
/dev/sda3 on /tmp type ext4 (rw,noatime,nodiratime,commit=0)
/dev/sda4 on /home type ext4 (rw,noatime,nodiratime,commit=0)
/dev/sdc1 on /mnt type vfat (rw)
```

Man sieht, der USB-Stick (/dev/sdc1) ist in das Verzeichnis mnt eingehängt. Das Dateisystem ist FAT mit der VFAT-Erweiterung.

Will man wissen, welche Gerätedateien den vorhandenen Partitionen zugeordnet sind, so kann man sich diese mit dem Kommando `fdisk -l` anzeigen lassen.

```
sudo fdisk -l
Disk /dev/sda: 1500.3 GB, 1500301910016 bytes
255 Köpfe, 63 Sektoren/Spur, 182401 Zylinder, zusammen 2930277168 Sektoren
Einheiten = Sektoren von 1 x 512 = 512 Bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifizier: 0x000aad0f

   Gerät boot.   Anfang           Ende       Blöcke   Id System
/dev/sda1 *      63      314584829   157292383+ 83 Linux
/dev/sda2        314584830   316705409   1060290   82 Linux Swap / Solaris
/dev/sda3        316705410   421577729   52436160  83 Linux
/dev/sda4        421577730   2930272064  1254347167+ 83 Linux

Platte /dev/sdc: 1040 MByte, 1040187392 Byte
1 Köpfe, 32 Sektoren/Spur, 63488 Zylinder, zusammen 2031616 Sektoren
Einheiten = Sektoren von 1 x 512 = 512 Bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifizier: 0xfd4d4282

   Gerät boot.   Anfang           Ende       Blöcke   Id System
/dev/sdc1 *      32      2031615    1015792   e  W95 FAT16 (LBA)
```

Für das Aushängen wird das Kommando `umount` verwendet. Als Parameter wird hier entweder der Gerätenamen oder das Verzeichnis angegeben, welches zum Einhängen verwendet worden ist.

```
sudo umount /mnt
```

Gerätedateien

Die meisten Festplatten werden über eine SCSI-Anbindung eingebunden (auch wenn es SATA-Platten, DVD-Laufwerke oder USB-Sticks sind). Dabei wird die erste Festplatte mit `sda` angesprochen, die zweite mit `sdb` usw. Die erste primäre Partition wird mit `1` angesprochen, die zweite mit `2` usw. Wird eine logische Partition verwendet, so beginnt die Nummer der ersten logischen Partition immer mit `5`.

erste primäre Partiton, zweite Festplatte -> /dev/sdb1
 zweite primäre Partiton, dritte Festplatte -> /dev/sdc2
 zweite logische Partiton, vierte Festplatte -> /dev/sdd6
 Mehr dazu auf Seite 31.

5.2.3 Dateisysteme prüfen

Ein Dateisystem wird mit dem Kommando `fsck` geprüft. `fsck` ist nur ein Wrapper, der für das entsprechende Dateisystem das eigentliche Prüfprogramm aufruft, z. B. für VFAT das Kommando `fsck.vfat`.

```
sudo fsck [-t vfstype] device
```

Soll hier unser USB-Stick geprüft werden, so lautet das Kommando:

```
sudo fsck -t vfat /dev/sdc1
fsck from util-linux 2.19.1
dosfsck 3.0.9, 31 Jan 2010, FAT32, LFN
/dev/sdc1: 50 files, 2479/253448 clusters
```

Wichtig: Das Dateisystem darf für die Prüfung nicht eingehängt sein.

Wird ein Fehler gefunden, so wird eine entsprechende Meldung ausgegeben und es wird versucht, diesen Fehler zu reparieren, wenn dies möglich ist.

5.2.4 Dateisystem formatieren

Mit dem Kommando `mkfs` kann ein Dateisystem angelegt werden.

```
mkfs: Usage: mkfs [-V] [-t fstype] [fs-options] device [size]
```

Auch hier wird wieder mit dem Parameter `-t` der Typ des Dateisystems angegeben und dann der entsprechende Wrapper aufgerufen, z. B. `mkfs.vfat`, `mkfs.ntfs`, `mkfs.ext3`, `mkfs.ext4`, ...

Soll der USB-Stick formatiert werden, so lautet der Aufruf:

```
sudo mkfs -t vfat /dev/sdc1
mkfs.vfat 3.0.9 (31 Jan 2010)
```

Auch hier darf das Dateisystem nicht eingehängt sein.

Steckt man den Stick ab und dann gleich wieder ein, so dass er automatisch eingehängt wird, so wird als Name eine generierte Nummer (z. B. A40B-7AA0) verwendet.

```
mount
..
.
/dev/sdc1 on /media/A40B-7AA0 type vfat (rw,nosuid,nodev,uid=1000,gid=1000,shortname=mixed,dmask=0077,2
utf8=1,showexec,flush,uhelper=udisks)
```

Möchte man einen anderen Namen verwenden, so kann man beim Formatieren diesen mit `-n` angeben.

```
sudo mkfs -t vfat -n sticklg /dev/sdc1
mkfs.vfat 3.0.9 (31 Jan 2010)
```

Somit wird der Stick dann automatisch unter `/media/sticklg` eingebunden.

Soll nachträglich der Name für einen Stick mit FAT-Dateisystem geändert werden, so verwendet man das Kommando `dosfslabel`.

```
sudo dosfslabel /dev/sdc1 sticklg
```

USB-Stick - Anzahl an Partitionen

Ein USB-Stick stellt ein normales Laufwerk dar. Daher ist es auch möglich, hier wie bei einer Festplatte mehrere Partitionen anzulegen. Allerdings unterstützt nicht jedes System (z. B. ältere Windows-Systeme) einen Stick mit mehreren Partitionen. Soll der Stick bei mehreren Systemen verwendet werden, so ist es ratsam, nur eine Partition (dies ist der Standard) zu verwenden.



Übung:

Für diese Übung verwenden Sie einen USB-Stick, der keine Daten enthält und **gelöscht** werden kann!

1. Stecken Sie einen USB-Stick an, der gelöscht werden kann. Wird dieser beim Einstecken automatisch eingehängt, so rufen sie das Kommando `umount` mit entsprechenden Parametern auf.

2. Formatieren Sie den Stick mit VFAT, ohne dass Sie einen Namen eingeben.

3. Setzen Sie nachträglich einen Namen.

4. Testen Sie den Stick, ob dieser Fehler enthält.

5.3 Bilddaten auswerten

Werden Bilder mit einer modernen Kamera oder einem Smartphone gemacht, so werden zu dem Bild auch entsprechende zusätzliche Informationen im Exif-Format (Exchangeable Image File Format) gespeichert. Exif-Daten werden direkt in die Datei (JPEG oder TIFF) im Header geschrieben.

Mit dem Kommando `exif` lassen sich diese Informationen anzeigen.⁶

Beispiele

exif IMG_8691.JPG

EXIF Einträge in 'IMG_8691.JPG' ('Intel' Byte-Reihenfolge):

```
-----+-----
Eintrag          |Wert
-----+-----
Hersteller       |Canon
Modell           |Canon PowerShot S70
Orientierung     |Links - Unten
Auflösung in X-Richt|180
Auflösung in Y-Richt|180
Maßeinheit der Auflö|Zoll
Datum und Uhrzeit |2011:11:08 13:30:15
YCbCr Positionierung|Zentriert
Kompression      |JPEG Kompression
Auflösung in X-Richt|180
Auflösung in Y-Richt|180
Maßeinheit der Auflö|Zoll
Belichtungszeit  |1/60 sek.
F Nummer        |f/2,8
Exif-Version     |Exif Version 2.2
Datum und Uhrzeit (o|2011:11:08 13:30:15
Datum und Uhrzeit (d|2011:11:08 13:30:15
Komponenten Konfigur|Y Cb Cr -
Komprimierte Bits pe| 5
Verschlusszeit   |5,91 EV (1/59 sek.)
Blende           |2,97 EV (f/2,8)
Belichtungsabweichun|0,00 EV
Maximaler Blendenwer|2,97 EV (f/2,8)
Messmodus       |Raster
Blitz            |Blitz ausgelöst, automatischer Modus, Rote Augen Reduktion
Brennweite       |5,8 mm
Anmerkungen des Hers|904 Byte(s) unbekannte Daten
Anmerkung des Nutzer|
FlashPixVersion  |FlashPix Version 1.0
Farbraum         |sRGB
Pixel X Dimension |3072
Pixel Y Dimension |2304
Fokusebene X-Auflösu|10816,901
Fokusebene Y-Auflösu|10816,901
Einheit der Angaben |Zoll
Bildsensor       |Einzel Chip Farbsensor
Dateiquelle      |DSC
Spezial Rendering |Normale Verarbeitung
Belichtungsart   |Automatische Belichtungszeit
Weißabgleich     |Automatischer Weißabgleich
Digitale Zoomrate |1,0000
Szenenmodus      |Standard
Interoperabilitäts I|R98
Interoperabilitäts V|0100
RelatedImageWidth |3072
RelatedImageLength|2304
-----+-----
EXIF Daten enthalten eine Vorschau (5859 Bytes).
```

exif IMAG0317.JPG

EXIF Einträge in 'IMAG0317.JPG' ('Motorola' Byte-Reihenfolge):

```
-----+-----
Eintrag          |Wert
-----+-----
Hersteller       |HTC
Modell           |HTC Sensation Z710e
Auflösung in X-Richt|72
Auflösung in Y-Richt|72
Maßeinheit der Auflö|Zoll
YCbCr Positionierung|Zentriert
```

⁶ Ist das Kommando nicht vorhanden, so installieren Sie das Paket `exif`.

```

Kompression           |JPEG Kompression
Auflösung in X-Richt |72
Auflösung in Y-Richt |72
Maßeinheit der Auflö |Zoll
ISO Geschwindigkeits |657
Exif-Version          |Exif Version 2.2
Datum und Uhrzeit (o |2012/03/31 12:41:45
Datum und Uhrzeit (d |2012/03/31 12:41:45
Komponenten Konfigur |Y Cb Cr -
Brennweite           |4,3 mm
FlashPixVersion       |FlashPix Version 1.0
Farbraum              |sRGB
Pixel X Dimension     |1840
Pixel Y Dimension     |3264
Interoperabilitäts I |R98
Interoperabilitäts V |0100

```

 EXIF Daten enthalten eine Vorschau (16608 Bytes).

Je nach Hersteller der Kamera oder des Smartphones werden unterschiedliche Informationen gespeichert. Auch das Format der Einträge kann unterschiedlich sein, z. B. beim Datum.

```

Datum und Uhrzeit     |2011:11:08 13:30:15
Datum und Uhrzeit (o |2012/03/31 12:41:45

```

Für uns ist das Datum wichtig, so dass die Bilder in eine chronologische Reihenfolge gebracht werden können.⁷ Anstelle der Namen können auch die TAG-IDs angezeigt werden.⁸ Dazu wird der Parameter `-i` angegeben.

Das Datum hat dabei folgende Tags:

```

0x132    DateTime
0x9004   DateTimeDigitized
0x9003   DateTimeOriginal
0x0007   GPS Zeit (Atom Uhr)
0x001d   GPS Zeit

```

Damit man nur die entsprechenden Einträge bekommt, ist es notwendig die Ausgabe zu filtern.



Übung:

Ermitteln Sie die Exif-Daten Ihrer Bilder. Findet sich darin das Datum der Aufnahme?

⁷ Das kann jedoch nur funktionieren, wenn bei allen verwendeten Kameras die Zeit korrekt eingestellt ist!

⁸ Mehr zu den Tags findet man unter http://www.jojosoftware.de/jojothumb/anleitung/jjt_doc802.htm

5.4 Filter für Textdateien

Von diversen Tools, wie z. B. dem Kommando `exif` erhält man sehr viele Informationen. Oft braucht man nur einen gewissen Teil davon, entweder nur eine Spalte oder eine Zeile, die mittels eines Schlüsselworts gefunden werden muss. Linux stellt eine Menge an Kommandos zur Verfügung, um solche Filteraktionen durchzuführen.

Nachfolgend sind die wichtigsten kurz beschrieben, so dass man anschließend die Informationen von `exif` auswerten kann.

5.4.1 Textdateien ausgeben

Mit dem Kommando `cat` lässt sich eine Datei oder auch mehrere ausgeben.

```
cat [Parameter] <Datei1> <Datei2> <...>
```

Die wichtigsten Parameter sind:

- n nummeriert alle Zeilen
- b nummeriert alle Zeilen, die nicht leer sind
- s unterdrückt aufeinander folgende leere Zeilen

Beispiel

```
cat -n sch.txt
```

```
1 Pyka:Johannes:IT11FS
2 Manfred:Rolf:IT11FA
3 Wanzinger:Guido:IT11SE
```

Alternative zu `cat`

Das Kommando `nl` dient auch dazu, eine Datei mit Zeilennummern auszugeben und stellt somit eine Alternative zu `cat` dar. Für die Nummerierung stellt es mehr Parameter zur Verfügung, z. B. die Möglichkeit, die Zeilennummern mit führenden Nullen auszugeben.

Beispiel

```
nl -nrz -w2 sch.txt
```

```
01 Pyka:Johannes:IT11FS
02 Manfred:Rolf:IT11FA
03 Wanzinger:Guido:IT11SE
```

Will man nur den ersten Teil einer Datei ausgeben, so verwendet man das Kommando `head`, für die letzten Zeilen einer Datei das Kommando `tail`.

```
head [Parameter] <Datei>
tail [Parameter] <Datei>
```

Die wichtigsten Parameter sind:

- c <x> die ersten/letzten x-Bytes werden ausgegeben
 <x> kann ein Vervielfachungssuffix haben: b für 512, k für 1KiB, m für 1MiB, g für 1GiB, kb für 1000, mb für 1000x1000, ...(T, P, E, Z, Y)
- n <x> die ersten/letzten x-Zeilen werden ausgegeben

Beispiel

```
head -n 5 schueler.txt | nl
```

```
1 Pyka:Johannes:IT11FS
2 Gerber:Dennis:IT11FS
3 Ewender:Nils:IT11FS
4 Feiler:Matthias:IT11FS
5 Wilms:Norman:IT11FS
```

Sollen die Zeilen 10-15 einer Datei ausgegeben werden, so können auch die Kommandos mit `|` verknüpft werden. Zu beachten ist, dass 10-15 insgesamt 6 Zeilen sind!

```
nl schueler.txt | head -n 15 | tail -n 6
```

```
10 Liepold:Michael:IT11FS
11 Aurich:Karl:IT11FS
12 Meyer:Frank:IT11FS
13 Jakob:Jens:IT11FS
14 Bischoff:Christian:IT11FS
15 Lehnert:Michael:IT11FS
```

Das Kommando `tail` bietet noch eine Besonderheit. Mit dem Parameter `-f` wird der Inhalt der Datei fortlaufend ausgegeben. Dies kann beim Anzeigen von LOG-Dateien sehr sinnvoll sein.

5.4.2 Textdateien sortieren

Mit dem Kommando `sort` lassen sich Textdateien sortieren. Die Sortierung erfolgt anhand von ASCII-Werten bzw. bei einem Unicode-Zeichensatz, wie UTF-8, nach dem Unicode-Codepoint (Reihenfolge in der Unicode-Tabelle). Wird die Variable `LANG`, `LC_ALL` oder `LC_COLLATE` mit `de`, `de_DE` oder `de_DE.UTF-8` gesetzt, so werden z. B. auch Umlaute entsprechend sortiert (ä -> ae, Ü, ue, ...).

```
sort [Parameter] <Datei>
```

Die wichtigsten Parameter sind:

- `-b` führende Leerzeichen werden ignoriert
- `-d` es wird nach Wörterbuch-Kriterien sortiert, d. h. nur Buchstaben, Ziffern und Leerzeichen werden berücksichtigt
- `-f` Kleinbuchstaben werden für die Sortierung wie Großbuchstaben behandelt
- `-i` nicht druckbare Zeichen werden ignoriert
- `-n` Zeichenketten, die Zahlen enthalten, werden als Zahlen sortiert
- `-r` umgekehrte Reihenfolge
- `-k Sp1, Sp2` es wird bei Spalte 1 gestartet und bei Spalte 2 aufgehört. Somit lässt sich nach Spalten sortieren
- `-t sep` legt das Trennzeichen für Spalten fest (siehe `-k`)
- `-u` es wird nur die erste Zeile von gleichen Zeilen ausgegeben
- `-o datei` die Ausgabe wird in einer Datei gespeichert, der Name darf auch der Name der Ursprungsdatei sein

Beispiele

```
sort -u schueler.txt | head -n 5
```

```
Aurich:Karl:IT11FS
Beslmeisel:Johann:IT11SE
Bischoff:Christian:IT11FS
Bittl:Erwin:IT11FA
```

Gibt die Schülerliste (eindeutig - doppelte Leerzeilen werden hier ignoriert) sortiert aus (nur die ersten 5 Einträge).

```
sort -t: -k3 -n /etc/passwd | head -n 5
```

```
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
sync:x:4:65534:sync:/bin:/bin/sync
```

Gibt die User des Systems sortiert aus. Dabei wird nach der UID (numerisch) sortiert (nur die ersten 5 Einträge).

Doppelte aufeinander folgende Zeilen ignorieren

Sind in einer Datei doppelte aufeinander folgende Zeilen vorhanden, so lassen sich diese mit dem Parameter `-u` ignorieren. Mit dem Kommando `uniq` lassen sich hier noch mehr Einstellungen vornehmen, z. B. ab welcher mehrfachen Anzahl ein Ignorieren stattfinden soll.

5.4.3 Zeilen von Textdateien auswählen

Mit dem Kommando `grep` (auch andere Kommandos unterstützen dies) lassen sich nach einem Muster nur bestimmte Zeilen auswählen, ähnlich wie die Wildcards beim Auffinden von Dateien. Dieses Muster wird auch als „Regulärer Ausdruck“ bzw. „regular expression“, kurz „regex“ bezeichnet.

Im nachfolgenden Beispiel werden alle Zeilen der Schülerdatei ausgegeben, bei denen „Michael“ enthalten ist.

```
grep 'Michael' schueler.txt
Hollmann:Michael:IT11FS
Liepold:Michael:IT11FS
Lehnert:Michael:IT11FS
Hohmann:Michael:IT11FA
```

Damit der Reguläre Ausdruck nicht bereits von der Shell interpretiert wird, muss dieser in einfache Hochkommas gesetzt werden.

Bei Regulären Ausdrücken wird zwischen verschiedenen Zeichen unterschieden. Dies sind Steuerzeichen, die eine besondere Bedeutung haben und normale Zeichen, die als solche verwendet werden können. Hier eine Auswahl:

<code>^</code>	steht für den Zeilenanfang
<code>\$</code>	steht für das Zeilenende
<code><</code>	Wortanfang
<code>></code>	Wortende
<code>\d</code>	Ziffern (0-9)
<code>\w</code>	alphanumerische Zeichen (a-z, A-Z, 0-9)
<code>\s</code>	Whitespace (Leerzeichen, Tabulatoren, Zeichenumbrüche, ... - <code>\t\n\r\f</code>)
<code>.</code>	genau ein beliebiges Zeichen
<code>+</code>	vorheriger Ausdruck mindestens ein Mal (1-n)
<code>*</code>	vorheriger Ausdruck beliebig oft (0-n)
<code>?</code>	vorheriger Ausdruck null oder ein Mal (0-1)
<code>{n, m}</code>	ein Wiederholungsintervall. Das der Klammer vorangehende Zeichen n bis m mal
<code>{n}</code>	ein genaues Wiederholungsintervall. Das der Klammer vorangehende Zeichen genau n mal
<code>[...]</code>	Zeichenklasse, eins der Zeichen aus der Klammer
<code>[...-...]</code>	Zeichenklasse von bis [0-9], [a-z], [A-Z], [0-9a-zA-Z], ...
<code>[^...]</code>	verneinte Zeichenklassen
<code>[^...-...]</code>	verneinte Zeichenklassen, keines der Zeichen aus der Klammer
<code>a1 a2</code>	alternative Ausdrücke, Ausdruck a1 oder a2
<code>(...)</code>	Gruppe von Ausdrücken
<code>\</code>	Ausblendung der Sonderbedeutung des nachfolgenden Zeichens <code>\.</code> für den Punkt, <code>\\$</code> für das Dollarzeichen, ...

Sollen z. B. alle Namen wie Maier (Mayr, Mair, Maier, Mayer, Meyer, Meier) unabhängig von der Groß/Kleinschreibung des ersten Buchstabens gefunden werden, so verwendet man folgenden Ausdruck:

```
[mM][ae][iy]e?r
```

`[mM]` m oder M

`[ae]` a oder e

`[iy]` i oder y

`e?` e ein oder kein Mal

`r` r

5.4.4 Textdateien in Spalten und Felder aufteilen

Mit dem Kommando `cut` lässt sich eine Datei in Spalten aufteilen. Dies ist immer dann notwendig, wenn man nur eine bestimmte Spalte einer Datei benötigt.

```
cut [Parameter] <Datei>
```

Die wichtigsten Parameter sind:

- d legt das Trennzeichen fest
- f es werden nur die angegebenen Spalten verwendet

Beispiel

Nachfolgendes Beispiel listet alle eingetragenen Shells der User (auch Systemuser) auf. Mit `sort` wird erreicht, dass doppelte Einträge ignoriert werden. Dazu wird die siebte Spalte der Datei `/etc/passwd` verwendet.

```
cut -d: -f7 /etc/passwd | sort -u
```

```
/bin/bash
/bin/false
/bin/sh
/bin/sync
/usr/sbin/nologin
```

5.4.5 Zeilen und Wörter in Textdateien zählen

Mit dem Kommando `wc` ist es möglich, Wörter, Bytes, Zeichen und Zeilen zu zählen.

Die wichtigsten Parameter sind dabei:

- c es werden Bytes gezählt
- m es werden Zeichen gezählt
- l es werden Zeilen gezählt
- w es werden Wörter gezählt

Beispiel

Im nachfolgenden Beispiel werden die `conf`-Dateien gezählt. Da `ls` für jede Datei eine Zeile erzeugt, müssen nur mit `wc` die Zeilen gezählt werden, um die Anzahl der Dateien zu erhalten.

```
cnt=$(ls /etc/*.conf | wc -l)
echo $cnt Dateien
```

```
37 Dateien
```

5.4.6 Zeichen austauschen

Mit dem Kommando `tr` lassen sich Zeichen austauschen (mehr dazu auf Seite 79).

```
tr [PARAMETER] Zeichenkette1 [Zeichenkette2]
```

Die wichtigsten Parameter sind:

- c das Komplement von Zeichenkette1 benutzen
- d Zeichen aus Zeichenkette1 löschen, nicht umwandeln
- t zuerst Zeichenkette1 auf die Länge von Zeichenkette2 kürzen

Hinweis: Es sind auch Auswahlen von Zeichen möglich, z. B. `[abc5-9]`

Beispiel

```
echo "01234" | tr '123' 'ABC'
```

```
0ABC4
```

**Übung:**

1. Probieren Sie folgende Befehlssequenz aus.

```
cat -n /etc/passwd | cut -d: -f1 > users
```

Erklären Sie, was passiert.

2. Ergänzen Sie die angegebene Befehlssequenz so, dass durch die Anwendung von `head` und `tail` die Zeilen 9 bis 14 der Datei `users` ausgegeben werden.

```
head          users | tail
```

3. Geben Sie mit einer Befehlssequenz durch Anwendung von `head` und `tail` die Zeile 2 von `users` aus.

4. Geben Sie die Datei `users` in umgekehrter Reihenfolge aus.

5. Geben Sie alle Zeilen der Datei `/etc/passwd` numerisch sortiert nach der User-ID (Feld 3, Feldtrenner ist ein `:`) aus. Dabei soll

a) die Zeile mit der höchsten User-ID zuerst ausgegeben werden,

b) die Zeile mit der niedrigsten User-ID zuerst ausgegeben werden.

6. Erstellen Sie ein Skript mit dem Namen `lss`, welches nur die Dateigröße und den Datennamen anzeigt (`ls -l`).

7. Geben Sie die Anzahl der Zeilen der Datei `/etc/passwd` aus.

8. Geben Sie zwei Möglichkeiten an, um die Zeilen der Datei `/etc/inittab` nummeriert auszugeben.

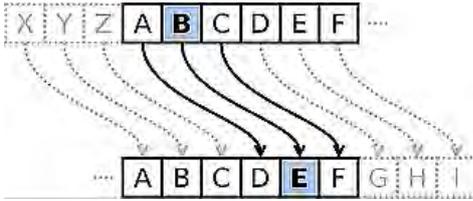
9. Nummerieren Sie mit `nl` die Zeilen der Datei `/etc/services` in Zweierschritten, beginnend bei 100.

10. Was macht das angegebene `tr`-Kommando mit dem berühmten Kinderlied?

```
echo DREI CHINESEN MIT DEM KONTRABASS | tr [aeiou] i
```

Versuchen Sie folgende Ausgabe zu erreichen:

```
DREI CHINESEN MIT DEM KONTRABASS ----> DRAA CHANASAN MAT DAM KANTRABASS
```

11. Cäsar-Chiffre⁹

Der berühmte römische Feldherr Julius Caesar verwendete angeblich für die Übermittlung geheimer Botschaften das folgende Verschlüsselungsverfahren:

Der Buchstabe A wurde ersetzt durch D, B durch E und so weiter; X entsprechend durch A, Y durch B und Z durch C. Wie lauten die `tr`-Aufrufe, um

- a) eine Botschaft des Feldherrn zu verschlüsseln und

- b) wieder zu entschlüsseln?

- c) Was bedeutet `WKH TXLFN EURZQ IRA MXPSV RYHU WKH ODCB GRJ?`¹⁰

für Gute!

12. Wie entfernen Sie in einer Textdatei alle Satzzeichen? Hinweis: `[:punct:]`

13. Wie ersetzen Sie alle Leerzeichen durch Zeilenumbrüche (Oktalzahl: `'\012'`)

14. Wie könnten Sie eine sortierte Liste aller Wörter in einer Textdatei aufstellen? Jedes Wort soll in der Liste nur einmal erscheinen.

15. Laden Sie mit `wget www.gutenberg.org/files/345/345.txt` den Text des Buches Dracula von Bram Stoker herunter¹¹ und führen Sie Folgendes aus:

```
cat 345.txt | tr ' '\012' | tr '[:upper:]' '[:lower:]' | tr -d '[:punct:]' | \
grep -v '^[a-z]' | sort | uniq -c | sort -rn >ausgabe.txt
```

Mit `less ausgabe.txt` können Sie sich die Ausgabe-Datei anzeigen lassen.

- a) Die Kommandosequenz ermittelt die Häufigkeit aller Wörter des Textes. Wie erreicht sie das?

Hinweise:

`'\012'` ist das Oktal-Zeichen für einen Zeilenumbruch

`grep -v '^[a-z]'` lässt nur Text durch

- b) Wie könnte die Kommandosequenz ergänzt werden, so dass die Gesamt-Anzahl aller Wörter ermittelt wird, die im Text nur je ein einziges Mal vorkommen?

⁹ siehe <http://de.wikipedia.org/wiki/Verschiebechiffre>

¹⁰ Dies ist ein Pangramm, diese wurden früher häufig zum Testen von Fernschreibverbindungen und Schreibmaschinen genutzt. <http://de.wikipedia.org/wiki/Pangramm>

¹¹ Aus dem Linux Journal Nr. 180, April 2009, <http://www.linuxjournal.com/article/10384>

5.5 Bilder kopieren

Nachdem jetzt die Möglichkeit besteht, Textdateien zu filtern, lassen sich die Bilder der Kameras kopieren. Beim Kopieren soll dabei der Dateiname so angepasst werden, dass dieser Datum, Uhrzeit und zur Sicherheit für doppelte Einträge eine fortlaufende Nummer enthält.

Im ersten Schritt sollen die Bilder kopiert werden. Für unser Beispiel wird das Verzeichnis `kamera` verwendet, wenn kein Parameter angegeben wird. Darin sind Bilder enthalten, die mit unterschiedlichen Kameras erstellt worden sind. Für das reale Skript muss hier natürlich das `/media`-Verzeichnis verwendet werden, in dem die USB-Sticks, Kameras, ... mit den Bildern eingebunden werden.

```
INPUT=$1
OUTPUT=$2

if [ -z $INPUT ] ; then
    INPUT=kamera
fi
if [ -z $OUTPUT ] ; then
    OUTPUT=.
fi

FILES=$(find ${INPUT} -iname '*.jpg' \
    -not -path '*thumbnails*' -print)
for FILE in $FILES ; do
    echo $FILE
done
```

```
kamera/img_cr2_0546.jpg
kamera/IMAG0433.jpg
kamera/IMG_8697.JPG
kamera/IMG_8693.JPG
```

Mit dem Kommando `find` werden hier die Dateien (rekursiv) gesucht. Mit `-iname` wird nicht zwischen Groß- und Kleinschreibung unterschieden. Da einige Kameras zusätzlich zu den Bildern auch noch Thumbnails für die Bildvorschau erzeugen, werden diese Verzeichnisse ignoriert. In der `for`-Schleife werden dann alle gefundenen Namen ausgegeben.

5.5.1 Datum und Uhrzeit ermitteln

Sind die Bilder gefunden, so kann versucht werden, das Datum und die Uhrzeit zu ermitteln.

Zur Erinnerung, das Datum hat dabei folgende Tags:

```
0x132    DateTime
0x9004   DateTimeDigitized
0x9003   DateTimeOriginal
0x0007   GPS Zeit (Atom Uhr)
0x001d   GPS Zeit
```

Wir verwenden hier die Tags `0x132` bzw. `0x9004`. Liefert die Kamera das Datum in anderen Tags, so muss das Skript entsprechend angepasst werden.

```
INPUT=$1
OUTPUT=$2

if [ -z $INPUT ] ; then
    INPUT=kamera
fi
if [ -z $OUTPUT ] ; then
    OUTPUT=.
fi

FILES=$(find ${INPUT} -iname '*.jpg' \
    -not -path '*thumbnails*' -print)
for FILE in $FILES ; do
    EXIF_DATE=$( \
        exif -i $FILE | \
        grep -e '^0x132' -e '^0x9004' | \
        cut -d'|' -f2
    )
    echo $EXIF_DATE
done
```

```
2011:04:20 10:50:51
2012/04/02 18:03:29
2011:12:05 16:44:10
2011:12:05 15:52:51
```

Das Kommando `exif` liefert mit dem Parameter `-i` eine Anzeige der Tag-Nummern, dann den Spaltentrenner „|“, gefolgt von Datum und Uhrzeit. Mit `grep` werden nur die Zeilen angezeigt, die das Tag `0x132` oder `0x9004` enthalten. Mit `cut` wird nur die zweite Spalte, also das Datum verwendet. Da das Zeichen „|“ von der Shell als Pipe interpretiert würde, muss es in einfache Hochkommas gesetzt werden.

5.5.2 Datum und Uhrzeit formatieren

Betrachtet man das Datum und die Uhrzeit, so fällt auf, dass diese einmal den „/“ und einmal den „:“ als Trenner verwenden. Zwischen Datum und Uhrzeit befindet sich ein Leerzeichen. Soll das Datum als Dateiname verwendet werden, so ist der „/“ nicht erlaubt. Auch ein Doppelpunkt ist nicht so gut, wenn man die Datei in einem Windowssystem betrachten möchte. Aus diesem Grund werden im nächsten Schritt diese Sonderzeichen durch einen Unterstrich ersetzt.

```
INPUT=$1
OUTPUT=$2

if [ -z $INPUT ] ; then
    INPUT=kamera
fi
if [ -z $OUTPUT ] ; then
    OUTPUT=.
fi

FILES=$(find ${INPUT} -iname '*.jpg' \
    -not -path '*thumbnails*' -print)
for FILE in $FILES ; do
    EXIF_DATE=$( \
        exif -i $FILE | \
        grep -e '^0x132' -e '^0x9004' | \
        cut -d'|' -f2 | \
        tr '/: ' '___' \
    )
    echo $EXIF_DATE
done
```

```
2011_04_20_10_50_51
2012_04_02_18_03_29
2011_12_05_16_44_10
2011_12_05_15_52_51
```

5.5.3 Bilder mit gleichem Datum

Die Auflösung des Datums/Uhrzeit beträgt hier eine Sekunde. Dies kann bedeuten, dass in einer Sekunde mehrere Bilder geschossen werden können. Daher muss als zusätzliches Unterscheidungskriterium eine fortlaufende Nummer eingebaut werden, damit Bilder, die in der selben Sekunde geschossen worden sind, einen unterschiedlichen Dateinamen enthalten.¹²

```
INPUT=$1
OUTPUT=$2
COUNTER=1

if [ -z $INPUT ] ; then
    INPUT=kamera
fi
if [ -z $OUTPUT ] ; then
    OUTPUT=.
fi

FILES=$(find ${INPUT} -iname '*.jpg' \
    -not -path '*thumbnails*' -print)
for FILE in $FILES ; do
    EXIF_DATE=$( \
        exif -i $FILE | \
        grep -e '^0x132' -e '^0x9004' | \
        cut -d'|' -f2 | \
        tr '/: ' '___' \
    )
    OFILE=img_${EXIF_DATE}_$( \
        printf "%04d" ${COUNTER} \
    ).jpg
    echo $OFILE
    COUNTER=$((COUNTER+1))
    cp "${FILE}" "${OUTPUT}/${OFILE}"
done
```

```
img_2011_04_20_10_50_51_0001.jpg
img_2012_04_02_18_03_29_0002.jpg
img_2011_12_05_16_44_10_0003.jpg
img_2011_12_05_15_52_51_0004.jpg
```

Mit dem Kommando `printf` wird dafür gesorgt, dass die Nummer immer mit vier Stellen und führenden Nullen ausgegeben wird. Beim Kopieren muss die Variable in doppelte Anführungszeichen gesetzt werden, da der Dateiname Leerzeichen enthalten kann (Leerzeichen -> Trenner von Parametern in der Shell).

Betrachtet man jetzt alle Bilder im Zielverzeichnis, so sind diese nach Datum und Uhrzeit sortiert und damit chronologisch angeordnet.

¹² Voraussetzung ist, dass bei den Kameras die Uhrzeit korrekt eingestellt worden ist.

5.5.4 Bilder automatisch drehen

Betrachtet man nun die Bilder, so stellt man fest, dass man bei einigen den Kopf drehen muss, da diese um 90° gedreht sind. Viele Bildanzeigeprogramme, aber nicht alle, drehen die Bilder automatisch, wenn ein entsprechender Eintrag in den exif-Informationen (Orientierung: Links - Unten) gefunden wird.

Mit dem Kommando `exifautotran` (Paket `libjpeg-progs`) lassen sich die Bilder automatisch drehen.

```
exifautotran [list of files]
```

Übung (für Gute):

1. Erweitern Sie das Skript so, dass nach dem Kopieren der Datei das Kommando `exifautotran` aufgerufen wird, so dass das Bild evtl. gedreht wird.
2. Erweitern Sie das Skript so, dass auch Videos (Dateiendung `avi`) und Bilder im raw-Format (Dateiendung `cr2`) kopiert werden. Da hier das Datum nicht mit Hilfe von `exif` ermittelt werden kann, soll das Dateidatum mit einer fortlaufenden Nummer für den neuen Namen verwendet werden.
3. Mit dem Kommando `ufraw` (Paket `ufraw`) lassen sich raw-Bilder in jpg-Bilder umwandeln.
 - a) Informieren Sie sich mit Hilfe der man-Pages, wie das Kommando `ufraw` bzw. `ufraw-batch` arbeitet. Betrachten Sie dabei die Parameter `--rotate`, `--out-type`, `--exif` und `--overwrite`.
 - b) Bauen Sie das Kommando `ufraw-batch` in das Skript ein, sodass beim Kopieren der raw-Bilder automatisch ein jpg-Bild erzeugt wird (mit exif-Informationen). Der Dateiname der jpg-Datei soll nach den vorherigen Regeln aufgebaut sein.



5.6 Thumbnails erzeugen

Für die übersichtliche Anzeige von Bildern ist es sinnvoll, kleine Vorschaubilder, sogenannte Thumbnails, zu erzeugen.

Das freie Programmpaket ImageMagick (Paket `imagemagick`) besitzt ein Kommando mit dem Namen `convert`, welches es ermöglicht, umfangreiche Umwandlungen an Bildern vorzunehmen.

Mit dem Parameter `-thumbnail 100` wird das Bild automatisch auf eine Breite von 100 Pixel skaliert. Dabei wird das Seitenverhältnis beibehalten.

```
convert -thumbnail 100 berlin_01.jpg thumb_berlin_01.jpg
```



Abb. 5.2: Original: Größe 2304x307



Abb. 5.3: Thumbnail: Größe 133x100

Übung:

Erstellen Sie ein Skript, welches aus allen jpg-Bildern Thumbnails erzeugt.



5.7 Spielereien mit Bildern

Mit Imagemagick sind noch einige interessante Spielereien möglich, um Bilder schöner zu präsentieren. Dieser Abschnitt hat dabei nichts mehr mit der Linux-Essentials-Prüfung zu tun, sondern dient als „nette“ Ergänzung. Probieren Sie einfach die vielen Parameter aus, dabei ist es meist nicht notwendig, alle Parameter zu verstehen.

Bei allen nachfolgenden Bildern verwenden wir das Originalbild `kastanie.jpg`, skalieren dieses, damit das Bild nicht so groß wird und wandeln dann das Ergebnis in ein png-Bild um. png-Bilder haben den Vorteil, dass diese z. B. auch Transparenz unterstützen und so mehr Effekte möglich sind als bei jpg-Bildern.¹³

5.7.1 Rahmen um Bilder

Bilder lassen sich schöner präsentieren, wenn man z. B. einen entsprechenden Rahmen um das Bild legt. Mit Imagemagick ist dies sehr leicht möglich. Lesen Sie dazu auch die Online-Dokumentation.

- einfacher Rahmen

```
convert kastanie.jpg -thumbnail 200 -mattecolor peru -frame 15x15+5+5 ka_s01.png
```



- 3D-Effekt mit `-raise`

```
convert kastanie.jpg -thumbnail 200 -raise 8 ka_s02.png
```



- 3D-Effekt mit `+raise`

```
convert kastanie.jpg -thumbnail 200 +raise 8 ka_s03.png
```



- 3D mit Verlauf (1) Verlauf erzeugen¹⁴

```
convert kastanie.jpg -thumbnail 200 -fill gray60 -colorize 100 \
  -raise 20 -normalize -blur 0x8 ka_s04.png
```



- 3D mit Verlauf (2) Bilder kombinieren

```
convert kastanie.jpg -thumbnail 200 -fill gray60 -colorize 100 \
  -raise 20 -normalize -blur 0x8 ka_s04.png
convert kastanie.jpg -thumbnail 200 ka_s04.png -compose hardlight \
  -composite ka_s05.png
```



¹³ Für die Beispiele sind Anregungen und Beispiele von der ImageMagick-Seite u. a. verwendet und entsprechend angepasst worden.

¹⁴ Dieser Schritt soll nur zur Verdeutlichung das graue Schattenbild zeigen.

- mit weichem Schatten

```
convert kastanie.jpg -thumbnail 200 \
  \( +clone -background Blue -shadow 80x3+5+5 \) \
  +swap -background none -layers merge +repage ka_s06.png
```



- mit hartem Schatten

```
convert kastanie.jpg -thumbnail 200 \
  \( +clone -background Blue -shadow 100x4+7+7 \) \
  +swap -background none -layers merge +repage ka_s07.png
```



- mit Rahmen

```
convert kastanie.jpg -thumbnail 200 \
  \( +clone -shave 20x20 -fill gray50 -colorize 100% \
  -mattecolor gray50 -frame 20x20+3+4 \) \
  -compose VividLight -composite ka_s08.png
```



5.7.2 Bilder beschneiden

- Ellipse

```
convert kastanie.jpg -thumbnail 200 -alpha set \
  -background none -vignette 0x4 ka_b01.png
```



- Rechteck mit Übergang

```
convert kastanie.jpg -thumbnail 200 \
  -alpha set -virtual-pixel transparent -channel A \
  -morphology Distance Euclidean:1,20\! +channel ka_b02.png
```



- Rechteck mit abgerundeten Ecken und Übergang

```
convert kastanie.jpg -thumbnail 200 \
  -alpha set -virtual-pixel transparent -channel A \
  -blur 0x20 -level 50%,100% +channel ka_b03.png
```



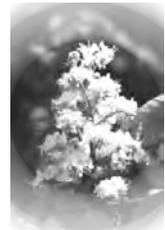
- Rechteck mit abgerundeten Ecken

```
convert kastanie.jpg -thumbnail 200 \
-alpha set -virtual-pixel transparent -channel A \
-blur 0x20 -threshold 50% +channel ka_b04.png
```



- Mit kreisförmiger Unschärfe

```
convert kastanie.jpg -thumbnail 200 \
-alpha set -virtual-pixel transparent -channel A \
-radial-blur 0x100 +channel ka_b05.png
```



- Tonpapier

```
convert kastanie.jpg -thumbnail 200 \
\(+clone -alpha extract -virtual-pixel black \
-spread 10 -blur 0x3 -threshold 50% -spread 4 -blur 0x.7 \) \
-alpha off -compose Copy_Opacity -composite ka_b06.png
```



- Polaroid 1

```
convert kastanie.jpg -thumbnail 200 \
-bordercolor white -border 6 \
-bordercolor grey60 -border 1 \
-background none -rotate 6 \
-background black \(+clone -shadow 60x4+4+4 \) +swap \
-background none -flatten ka_b07.png
```



- Polaroid 2

```
convert kastanie.jpg -thumbnail 200 \
-bordercolor snow -background black +polaroid ka_b08.png
```



- jedes Symbol kann verwendet werden¹⁵

```
# gezackte Sprechblase generieren
convert -size 230x250 -gravity center -font webdings.ttf label:{ \
-negate -channel A -combine +channel -fill pink -colorize 100% \
-background none -fill none -stroke black -strokewidth 5 label:{ \
-flatten +gravity -chop 0x10+0+0 -shave 0x10 +repage ka_b10a.png
# Schatten hinzufügen
convert ka_b10a.png \
\(+clone -background black -shadow 60x3+3+3 \) \
-background none -compose DstOver -flatten ka_b10b.png
# Bild skalieren
convert kastanie.jpg -thumbnail 200 ka_b10c.png
# Bilder zusammenfügen
convert ka_b10b.png ka_b10c.png \
-gravity center -compose DstOver -composite ka_b10.png
```



¹⁵ Es wird der Font „WebDings“ (webdings.ttf) verwendet.

- Rahmen ...¹⁶

```
convert kastanie.jpg -thumbnail 200 -matte -compose Copy \
  -bordercolor Black -border 2 \
  -bordercolor Sienna -border 3 \
  -bordercolor Black -border 1 \
  -bordercolor None -border 2 \
  -bordercolor Black -border 2 \
  -bordercolor Peru -border 3 \
  -bordercolor Black -border 1 \
  -compose Over \
  \( fancy_add.png          \) -gravity NorthWest -composite \
  \( fancy_add.png -flip    \) -gravity SouthWest -composite \
  \( fancy_add.png          -flop \) -gravity NorthEast -composite \
  \( fancy_add.png -flip -flop \) -gravity SouthEast -composite \
  -compose DstOut \
  \( fancy_sub.png          \) -gravity NorthWest -composite \
  \( fancy_sub.png -flip    \) -gravity SouthWest -composite \
  \( fancy_sub.png          -flop \) -gravity NorthEast -composite \
  \( fancy_sub.png -flip -flop \) -gravity SouthEast -composite \
  ka_b11.png
```



Viele weitere Beispiele und Möglichkeiten findet man über entsprechende Suchmaschinen im Netz.

5.8 Fragen

1. Welche Aufgabe hat HAL?

2. In welchem Verzeichnis finden sich die LOG-Dateien von Linux?

3. In welcher Datei landen allgemeine Systemmeldungen?

4. Wie können Meldungen, die beim Systemstart erzeugt werden, angezeigt werden?

5. Ein USB-Stick funktioniert nicht beim Anstecken. Wie erreichen Sie es, dass Systemmeldungen beim Anstecken kontinuierlich angezeigt werden.

6. Warum muss ein USB-Stick „Sicher entfernt“ werden?

7. Wie werden alle angesteckten USB-Geräte aufgelistet?

8. Wie können mit `find` Dateien kopiert werden, die nicht älter als 24 Stunden und größer als 10 KB sind?

9. Wie wird bei `find` verhindert, dass nicht mehr als 2 Verzeichnisebenen durchsucht werden?

10. Mit welchem anderen Kommando können Dateien gefunden werden und welche Vor-/Nachteil hat dieses Kommando gegenüber `find`?

¹⁶ Rahmenbilder aus „DIY Frames Section of Anthony’s Icon“

11. Mit welchem Kommando werden ausführbare Dateien (Kommandos) gefunden, die ohne Angabe eines Verzeichnisses gestartet werden können?
-
12. Mit welchen Variablen können Sie die Sucheinstellungen für ausführbare Dateien (Kommandos) erweitern?
-
13. Beim Einstecken eines USB-Sticks wird gemeldet, dass das Format nicht erkannt wurde bzw. fehlerhaft ist. Mit welchem Kommando testen Sie den Stick?
-
14. Was versteht man unter einem Journaling-Dateisystem?
-
15. Wie wird ein Stick „von Hand“ eingebunden und wie wieder entfernt?
-
16. Mit welchem Kommando (+ Parameter) können Sie feststellen, unter welcher Gerätedatei ein Stick ansprechbar ist?
-
17. Wie lautet die entsprechende Gerätedatei, wenn eine Festplatte über SCSI angesprochen wird?
- a) zweite primäre Partition der dritten Festplatte:
-
- b) erste logische Partition der ersten Festplatte:
-
18. Ein Stick ist mit VFAT formatiert. Wie prüfen Sie die Formatierung?
-
19. Bei einem Stick ist die Formatierung defekt. Wie wird dieser mit VFAT neu formatiert?
-
20. Mit welchem Kommando erhalten Sie zusätzliche Informationen zu Bildern?
-
21. Wie lautet die Kommando-Sequenz, mit der nur das „Modell“ der Kamera beim Bild IMG4711.JPG angezeigt wird?
-
22. Bei einer Datei sollen die Zeilen 7-12 angezeigt werden. Wie lautet der Aufruf?
-
23. Wie wird eine Datei /ect/passwd nach der dritten Spalte in umgekehrter Reihenfolge sortiert?
-
24. Eine Zeichenkette soll in Kleinbuchstaben umgewandelt werden. Wie lautet der Aufruf?
-
25. Wie speichern Sie das aktuelle Datum im Format „2012-06-03_18:45“ in der Variablen aktDate?
-

6 Einheit V – Mit dem Netzwerk arbeiten

Die Klassen möchten die Bilder der Jahresabschlussfahrt allen zur Verfügung stellen. Dazu soll im lokalen Netz auf einem Intranet-Server eine Bildergalerie bereitgestellt werden. Um dies realisieren zu können, müssen zunächst in der folgenden Einheit wichtige Netzwerkgrundlagen angesprochen und ein Webserver installiert werden. Die Erstellung der Bildergalerie ist dann Thema der nächsten Einheit.

Die folgenden Kapitel beschreiben, wie ein Linux-Rechner in ein lokales Netzwerk eingebunden wird, so dass eine Verbindung zum Internet hergestellt werden kann. Vor einer praktischen Umsetzung werden zunächst einige theoretische Grundlagen gelegt, indem nach einer kurzen Beschreibung der wichtigsten Protokolle der TCP/IP-Protokollfamilie die wesentlichen Eigenschaften von IPv4- und IPv6-Adressen erläutert werden. Dann wird an einem typischen Netzwerk-Szenario aufgezeigt, welche Schritte nötig sind, um bei einem Linux-Rechner das Netzwerk zu konfigurieren.

Im Fragenteil werden dann noch einmal die wichtigsten Aspekte der Linux-Netzwerkconfiguration angesprochen. Dann wird das Gelernte in einem Übungsteil praktisch umgesetzt, indem die Netzkonfiguration eines Linux-Rechners praktisch durchgeführt wird. Abschließend wird auf dem Intranet-Server ein Webserver installiert und getestet. Optional kann auf dem Intranet-Server auch ein SSH-Zugang bereitgestellt werden.

Inhalt

6.1	Das Internet und die TCP/IP-Protokoll-Familie	116
6.2	IPv4	117
6.3	IPv6	119
6.4	IP-Adressen und Rechnernamen mit DNS	120
6.5	Manuelle Netzwerkkonfiguration	120
6.6	Automatische Netzwerkkonfiguration mit DHCP	123
6.7	Fragen	127

6.1 Das Internet und die TCP/IP-Protokoll-Familie

Protokolle aus der TCP/IP-Familie sind weit verbreitet und werden heute in praktisch allen Bereichen der Kommunikationstechnik eingesetzt. Dabei handelt es sich um Verfahren, deren Entwicklung bereits in den 1960er Jahren durch das US Verteidigungsministerium (Department of Defense, DoD) betrieben wurde. Ziel war es, die Zusammenarbeit von militärischen Forschungseinrichtungen untereinander und mit Universitäten zu verbessern und die Verbindung verschiedener Netzwerke von verschiedenen Herstellern und somit den Aufbau eines „Netzes der Netze“ (Internet) zu entwickeln.

Beim Internet handelt es sich einerseits um den allgemeinen Begriff für die Zusammenschaltung mehrerer Netze, andererseits aber auch um ein weltumfassendes Kommunikationsnetz. Der Erfolg des Internets beruht auch darauf, dass eine große Anzahl von Teilnehmern auf zentrale Dienste zugreifen kann, z. B. Datenaustausch mit -FTP, elektronische Post über SMTP oder die Fernsteuerung von Rechnern mit SSH bzw. Telnet.

Die TCP/IP-Protokoll-Familie ist nach ihren Hauptprotokollen Transmission Control Protocol (TCP) und Internet Protocol (IP) benannt, die eine zentrale Rolle einnehmen. Im Fachjargon wird die Gesamtheit der Protokolle auch TCP/IP-Protokollstapel oder TCP/IP-Stack genannt.

TCP/IP-Protokolle sind so ausgelegt, dass ein möglichst stabiles Netz entsteht und dass der Ausfall von Übertragungswegen oder Netzknoten automatisch überbrückt werden kann. Zu den wesentlichen Vorteilen der herstellerunabhängigen TCP/IP-Protokolle gehört es, dass sie auf nahezu allen Rechnern implementierbar sind und sowohl zahlreiche Einsatzgebiete als auch eine weite Verbreitung aufweisen. Außerdem werden große Netzwerke mit dennoch geringem zentralen Management möglich.

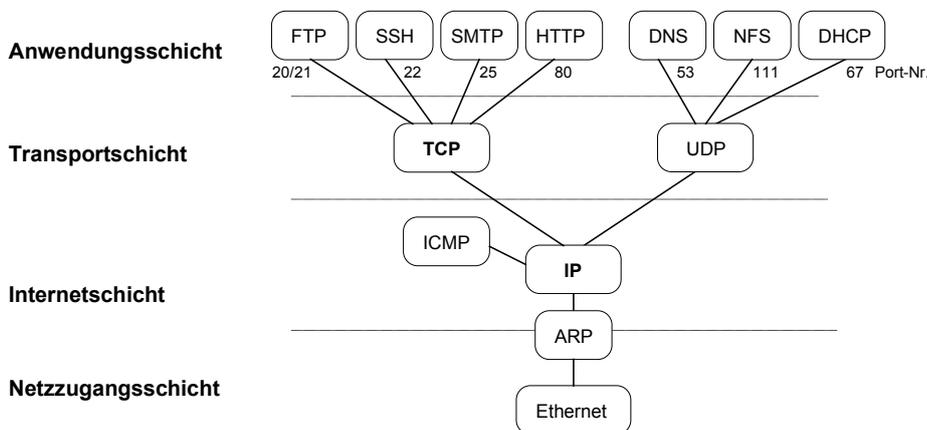


Abb. 6.1: TCP/IP-Schichten

Protokolle der Internetschicht

- IP** Internet Protocol
unbestätigtes, verbindungsloses Protokoll, gewährleistet die weltweit eindeutige Adressierung und übermittelt Datenpakete über ein Netz oder durch den Einsatz von Routern über mehrere Netze
- ICMP** Internet Control Message Protocol
regelt den Datenverkehr im Internet, überträgt Steuerinformationen zwischen Stationen und stellt für die Internet- und die Transportschicht Diagnoseinformationen bereit
- ARP** Address Resolution Protocol
dient zur Ermittlung der physikalischen MAC-Adresse bei bekannter logischer IP-Adresse

Eine der bekanntesten Anwendungen, die ICMP benutzt, ist das Kommando `ping`. Mit `ping` kann festgestellt werden, ob der Empfänger überhaupt angesprochen werden kann und wie lange eine Übertragung zu ihm dauert.

Protokolle der Transportschicht

- TCP** Transmission Control Protocol
bestätigtes, verbindungsorientiertes Protokoll, das eine zuverlässige Übertragung zusammenhängender Datenströme in beiden Richtungen ermöglicht
- UDP** User Datagramm Protocol
unbestätigtes, verbindungsloses Protokoll, das einzelne Datensegmente zwischen zwei Stationen in beiden Richtungen übermittelt, UDP ist wegen der fehlenden Bestätigung unzuverlässig, dafür aber schlanker und hat kürzere Reaktionszeiten als TCP

Protokolle der Anwendungsschicht

- FTP** File Transfer Protocol
ermöglicht schnelle Dateiübertragungen zwischen Client und Server
- SMTP** Simple Mail Transfer Protocol
Austausch von elektronischer Post zwischen zwei Systemen
- SSH** Secure Shell
erlaubt über ein Kommandofenster den interaktiven Zugriff auf einen externen Rechner (im Gegensatz zu Telnet ist SSH verschlüsselt)
- NFS** Network File System
Datei-Zugriff über ein Netzwerk
- HTTP** Hypertext Transfer Protocol
Hypertext-Seiten über das World Wide Web übertragen

Adressierung der Protokolle bzw. Dienste in der Anwendungsschicht

Um Daten an einen bestimmten Dienst der Anwendungsschicht zu senden, werden von TCP bzw. UDP sogenannte Ports zur Adressierung verwendet, z. B. FTP (20/21), SMTP (25), SSH (22), HTTP (80) und https (443).

6.2 IPv4

Mit der IP-Adresse wird ein Rechner (Host) eindeutig adressiert. Eine IPv4-Adresse hat eine Länge von vier Bytes (=32 Bit) und wird in zwei Teile unterteilt: Netz-Anteil (Netz-ID) und Host-Anteil (Host-ID).

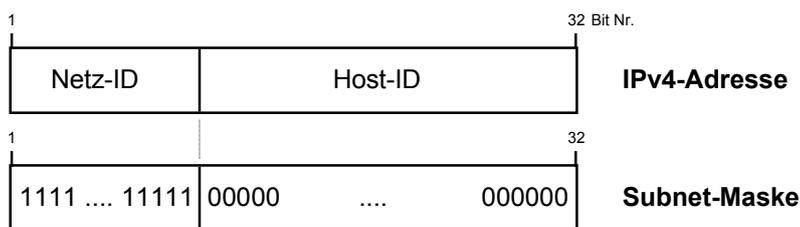


Abb. 6.2: IPv4-Adressformat

- Die Subnet-Maske trennt zwischen Netz-ID und Host-ID der IP-Adresse. Sie besteht ebenfalls aus 32 Bit.
- Die Bit-Stellen, an denen die Subnet-Maske eine binäre „1“ besitzt, gehören bei der IP-Adresse zur Netz-ID.
- Die Bit-Stellen, an denen die Subnet-Maske eine binäre „0“ besitzt, gehören bei der IP-Adresse zur Host-ID.
- Die binären Einsen der Subnet-Maske sind immer aneinanderhängend linksbündig angeordnet!

Gepunktete Dezimaldarstellung von IPv4-Adressen und Subnetz-Masken

Die IP-Adresse wird in 4 Gruppen zu je 8-Bit (Oktette) aufgeteilt. Die Oktette werden jeweils als Dezimalzahl dargestellt und durch einen Punkt getrennt.

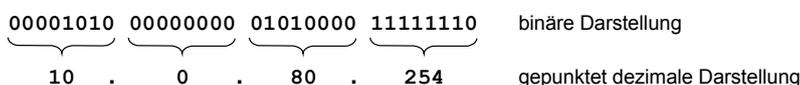


Abb. 6.3: Gepunktete Dezimaldarstellung von IPv4-Adressen

Subnet-Masken werden oft in der sogenannten CIDR-Darstellung angegeben: Nach dem „/“ wird die Anzahl der auf „1“ gesetzten Bits angegeben.
Beispiele: 255.0.0.0 entspricht /8 , 255.255.0.0 entspricht /16 , 255.255.255.128 entspricht /25

IPv4 Adressklassen

Eine IPv4 Adresse kann einer von insgesamt fünf Adressklassen angehören:

Klasse	IP-Adressbereich von - bis	Standard Subnet-Maske	
		dezimal	CIDR-Darstellung
A	0.0.0.0 - 127.255.255.255	255.0.0.0	/8
B	128.0.0.0 - 191.255.255.255	255.255.0.0	/16
C	192.0.0.0 - 223.255.255.255	255.255.255.0	/24
D	224.0.0.0 - 239.255.255.255	Multicastbereich	
E	240.0.0.0 - 255.255.255.255	reserviert	

Tab. 6.1: IPv4 Adressklassen

Die Klassen A, B und C sind für unterschiedliche Netzgrößen vorgesehen. Der Vorteil der Untergliederung in Adress-Klassen ist die Möglichkeit, die Adressen an vorhandene Organisationsstrukturen anzupassen.

Eine Ausnahme hiervon stellt die Klasse D dar, bei der es sich um eine Adressierung einer Gruppe von Hosts handelt, die logisch zusammenhängen (Multicast). Multicast-Übertragungen werden z. B. bei IP-TV und „Video on Demand“ eingesetzt.

Die Klasse E wird nicht verwendet und ist für zukünftige Anwendungen reserviert.

Besondere IPv4-Adressen

Netz- und Broadcast-Adresse

Die sog. Netzadresse ergibt sich, wenn alle Bits der Host-ID auf „0“ gesetzt werden. Die Broadcast-Adresse ergibt sich, wenn alle Bits der Host-ID auf „1“ gesetzt werden.

Loopback-Adresse

Eine weitere IPv4 Sonderadresse ist die Loopback-Adresse 127.0.0.1. Ein zu Testzwecken an diese Zieladresse gesendetes Paket verlässt den Rechner nicht und wird unmittelbar empfangen.

Achtung: Diese Sonderadressen dürfen nicht für die Adressierung von Rechnern vergeben werden!

Private IPv4-Adressen

In den Klassen A, B und C wurden Bereiche für private Adressierung festgelegt. Diese privaten IPv4-Adressen sind für lokale Netzwerke (LAN) gedacht, die keine öffentlichen Adressen benötigen. Private IP-Adressen werden im Internet nicht weitergeleitet (geroutet). Die folgenden IPv4-Adressbereiche sind privat:

Klasse	IP-Adressbereich von - bis	CIDR-Darstellung	Bemerkung
A	10.0.0.0 - 10.255.255.255	10.0.0.0/8	ein A-Netz
B	172.16.0.0 - 172.31.255.255	172.16.0.0/12	16 B-Netze
	169.254.0.0 - 169.254.255.255	169.254.0.0/16	ein B-Netz für Automatic Private IP Addressing (API-PA)
C	192.168.0.0 - 192.168.255.255	192.168.0.0/16	256 C-Netze

Tab. 6.2: Private IPv4-Adressen

6.3 IPv6

Da der verwendbare IPv4-Adressraum in nur drei Klassen unterteilt wurde, war es nicht immer möglich, mit IPv4-Adressen effektiv umzugehen. Dies führte bereits Mitte der 1990er Jahre zur Erkenntnis, dass in absehbarer Zeit ein Adressen-Engpass auftreten und deshalb ein neues Adressierungsschema nötig sein würde. Langfristig wird die Version 6 des Internet Protokolls alle derzeit erkennbaren Probleme mit der Verfügbarkeit von IP-Adressen sowie dem Routing im IP-Bereich lösen. IPv6 berücksichtigt bereits künftige Anwendungen wie beispielsweise Multimedia oder Video on Demand.

Vorteile von IPv6

- erweiterter Adressraum: Durch die Verwendung von 128 Bits sind theoretisch 2^{128} IPv6-Adressen möglich
- automatische Vergabe und Konfiguration von IPv6-Adressen (IPv6-Autokonfiguration)
- zusätzliche Sicherheitsoptionen für sicheren Datenaustausch über VPNs (IPsec)
- Unterstützung von Dienstgütemerkmalen (Quality of Service, QoS), z. B. für die Priorisierung von Datenübertragungen
- vereinfachtes Routing

IPv6 Adressarten

Innerhalb von IPv6 wurde eine Vielzahl verschiedener Arten von Adressen mit Sonderaufgaben und unterschiedlichen Eigenschaften definiert.

Die IPv6-Adressen sind nicht mehr in Klassen unterteilt. Stattdessen gibt es drei Arten von Adressen:

Unicast ein bestimmter Empfänger

Anycast ein beliebiger Empfänger aus einer Gruppe

Multicast an jeden Empfänger einer Gruppe

Broadcast-Adressen gibt es bei IPv6 nicht mehr!

IPv6 Adresstypen

Innerhalb der Unicast-Adressen bestehen wiederum verschiedene Typen von Adressen:

Global-Unicast	(2000::/3)	weltweit gültig, global routbar, werden durch den Netzbetreiber vergeben
Unique Local Unicast	(fc00::/7)	private Adressen, die im Internet nicht geroutet werden
Link-Local-Adressen	(fe80::/10)	werden nur innerhalb eines Subnetzes verwendet und überhaupt nicht geroutet

IPv6-Adressformat

Beispielhaft wird nachfolgend der Aufbau einer weltweit gültigen IPv6 Global-Unicast-Adresse gezeigt.

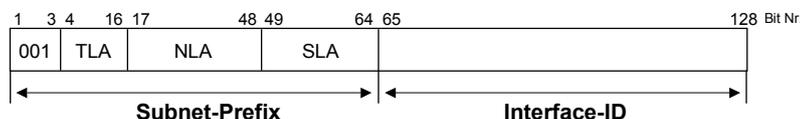


Abb. 6.4: IPv6-Adressformat

Die 128 Bit werden unterteilt in 64 Bit für das Subnet-Prefix und 64 Bit für die Interface-ID. Das Subnet-Prefix setzt sich zusammen aus 45 Bit für die Identifikation eines Unternehmens oder einer Organisation (Top Level Aggregation, TLA und Next Level Aggregation, NLA) sowie 16 Bit für die Subnetz-Adresse (Site Level Aggregation, SLA). Im Subnet-Prefix sind die ersten drei Bits für die Identifizierung des Adress-Typs reserviert (hier: 001 für Global Unicast). Die Interface-ID wird unter anderem aus der MAC-Adresse der Netzwerkkarte abgeleitet.

Darstellung von IPv6-Adressen

IPv6-Adressen werden in acht Blöcken zu je zwei Bytes dargestellt. Die Blöcke werden jeweils im Hexadezimalformat, getrennt durch Doppelpunkte, angegeben. Pro Block können führende Nullen entfallen, ebenso Blöcke, die nur Nullen beinhalten. Aufeinander folgende Doppelpunkte können durch einen Doppelpunkt ersetzt werden, solange die Eindeutigkeit gewährt bleibt.

Beispiel:

2001:0db8:0000:0000:0000:0000:1428:57ab wird zu 2001:0db8::1428:57ab

6.4 IP-Adressen und Rechnernamen mit DNS

Da für Nutzer IP-Adressen nicht besonders übersichtlich sind und sich Namen viel besser einprägen, können Rechner im Internet auch über ihren Namen angesprochen werden. Diese Namensauflösung wird über das Domain Name System (DNS) realisiert. Dabei handelt es sich um ein hierarchisches Namensschema, bei dem ein Rechnernamen von rechts nach links in Root-Domain, Top-Level-Domain, Second-Level-Domain, beliebig viele Subdomains und den eigentlichen Rechnernamen unterteilt wird. Die einzelnen Namensbestandteile werden jeweils durch einen Punkt getrennt. Vollständige DNS-Namen, die aus allen angegebenen Elementen bestehen, werden als Fully Qualified Domain Name (FQDN) bezeichnet.

Die Aufgabe eines DNS-Servers ist es, die Namen in Adressen und umgekehrt umzusetzen.

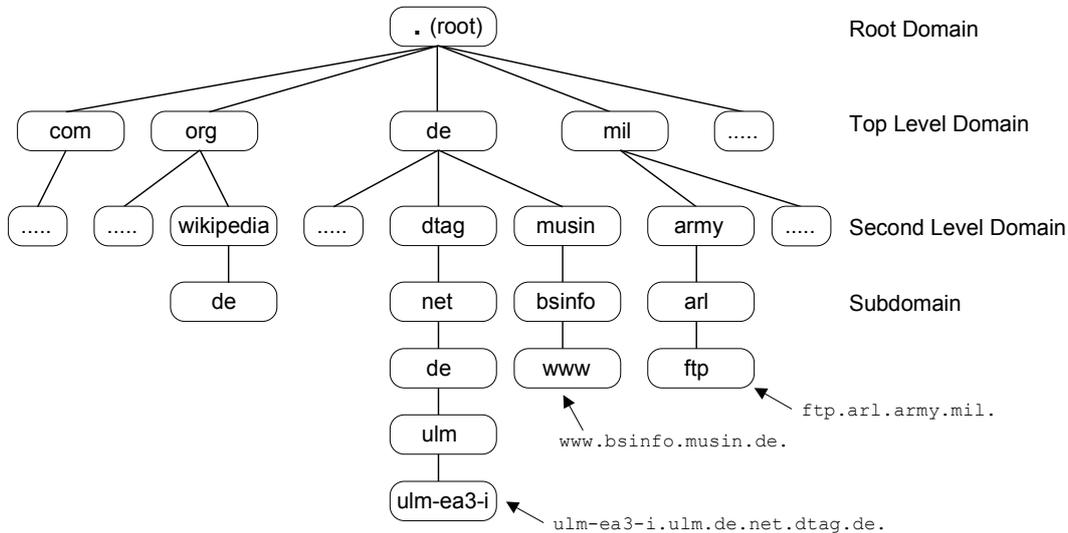


Abb. 6.5: DNS

Einige Beispiele für vollständige DNS-Namen:

`www.bsinfo.musin.de.` Städtische Berufsschule für Informationstechnik München
`ulm-ea3-i.ulm.de.net.dtag.de.` DNS-Name eines Routers bei der Telekom AG
`ftp.arl.army.mil.` einer der ersten FTP-Server im Internet

Der Punkt für die Root-Domain wird meist weggelassen!

Uniform Resource Locator (URL)

Die Verwendung eines Uniform Resource Locators (URL) ermöglicht über eine einheitliche Angabe den Zugriff auf viele Dienste im Netzwerk oder im Internet. Das URL-Format setzt sich zusammen aus: dem Dienst (z. B. `http`, `ftp`, ...), der Adresse oder dem DNS-Namen des Servers, auf dem der Dienst angeboten wird sowie gegebenenfalls dem Pfadnamen der angefragten Datei.

Beispiele für URLs:

`http://www.bsinfo.musin.de/downloads/Blockplan1112.pdf`
`ftp://ftp.debian.org/debian/README.html`
`http://192.168.9.184/files/VMs/Linux-vm.rar`

URL-Angaben mit IPv6-Adressen müssen mit [] geklammert werden:

`http://[2001:0db8:85a3:08d3:1319:8a2e:0370:7344]/`
`http://[2001:0db8:85a3:08d3:1319:8a2e:0370:7344]:8080/`

6.5 Manuelle Netzwerkkonfiguration

Viele Distributionen verwenden zur Netzwerkkonfiguration den NetworkManager, ein Programm, mit dem sich Netzwerkverbindungen unter Linux verwalten lassen und der besonders den Umgang mit drahtlosen Netzwerken erleichtert. Oft werden aber auch distributionseigene Tools (z. B. `yast` bei SUSE) eingesetzt. Der NetworkManager ist wegen seiner einfachen Benutzung vor allem im Desktop-Bereich verbreitet. Bei Servern wird die Netzwerkkonfiguration meist manuell durchgeführt.

An dem nachfolgend abgebildeten Netzwerk-Szenario soll schrittweise die manuelle Netzwerkkonfiguration des Linux-Client-PCs `linux1` dargestellt werden.

Zunächst wird dazu die IPv4-Adresse der Netzwerkkarte eingestellt, dann wird der DSL-Router als Standard-Gateway eingetragen und schließlich die Namensauflösung konfiguriert.

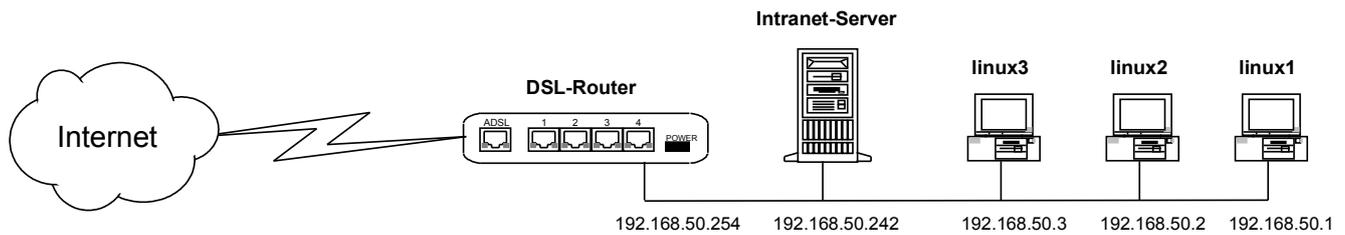


Abb. 6.6: Szenario

Adresskonfiguration der Netzwerkkarten

Die aktuelle Konfiguration der Netzwerkkarten wird mit `ifconfig` abgefragt:

```
ifconfig -a
eth0      Link encap:Ethernet  Hardware Adresse 00:0C:29:C7:1E:69
          inet Adresse:192.168.50.1  Bcast:192.168.50.255  Maske:255.255.255.0
          inet6 Adresse: fe80::20c:29ff:fec7:1e69/64  Gültigkeitsbereich:Verbindung
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 Sendewarteschlangenlänge:1000
          RX bytes:939 (939.0 b)  TX bytes:1378 (1.3 Kb)
          Interrupt:18 Basisadresse:0x1424

lo        Link encap:Lokale Schleife
          inet Adresse:127.0.0.1  Maske:255.0.0.0
          inet6 Adresse: ::1/128  Gültigkeitsbereich:Maschine
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:34 errors:0 dropped:0 overruns:0 frame:0
          TX packets:34 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 Sendewarteschlangenlänge:0
          RX bytes:2304 (2.2 Kb)  TX bytes:2304 (2.2 Kb)

wlan0     Link encap:Ethernet  Hardware Adresse 00:80:5a:54:48:cb
          UP BROADCAST MULTICAST  MTU:1500  Metrik:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

In diesem Beispiel sind drei Netzwerkschnittstellen vorhanden:

- Eine als `eth0` konfigurierte Ethernet-Karte mit der IPv4-Adresse `192.168.50.1` und der IPv6-Adresse `fe80::20c:29ff:fec7:1e69`.
- Zusätzlich ist eine noch nicht konfigurierte Funknetz-Karte (`wlan0`) vorhanden.
- Dazu kommt das Loopback-Interface `lo`, das auf den eigenen Rechner mit der IPv4-Loopback-Adresse `127.0.0.1` (bzw. IPv6-Loopback-Adresse `::1`) verweist.

Es wurden noch weitere Informationen ausgegeben: z.B. die Hardware-Adressen (MAC-Adresse) der Ethernet- und Funknetz-Karten, die zugeordneten Subnet-Masken, die IPv4 Broadcast-Adressen und einige Statusinformationen.

Die Option `-a` stellt sicher, dass `ifconfig` die Einstellungen aller vorhandenen Netzwerkkarten, auch die der noch nicht konfigurierten, ausgibt.

Mit `ifconfig` könnte der Ethernet-Karte eine andere IPv4-Adresse zugewiesen werden:

```
ifconfig eth0 192.168.1.1 up
```

Achtung: Die Konfiguration ist sofort wirksam!

Wenn eine nicht standardmäßige Subnetzmaske verwendet werden soll, müssen sowohl die zu benutzende Maske als auch die sich ergebende Broadcast-Adresse mit angegeben werden:

```
ifconfig eth0 192.168.2.131 netmask 255.255.255.240 broadcast 192.168.2.143 up
```

Eine CIDR-Angabe ist auch möglich:

```
ifconfig eth0 192.168.2.131/28
```

Wurde die IP-Adresse von Rechner `linux1` korrekt gesetzt, kann von dort aus mit `ping` geprüft werden, ob ein anderer Rechner in diesem IP-Netz, z. B. der DSL-Router, erreichbar ist:

```
ping 192.168.50.254
PING 192.168.50.254 (192.168.50.254) 56(84) bytes of data.
64 bytes from 192.168.50.254: icmp_seq=1 ttl=64 time=0.574 ms
64 bytes from 192.168.50.254: icmp_seq=2 ttl=64 time=0.053 ms
64 bytes from 192.168.50.254: icmp_seq=3 ttl=64 time=0.051 ms
^C
--- 192.168.50.254 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.051/0.226/0.574/0.246 ms
```

Hinweis: Hier wurde `ping` nach einigen Ausgabezeilen mit der Tastenkombination `[Strg]+[c]` abgebrochen.

Einstellen des Standardgateways

Soll der Rechner `linux1` über den DSL-Router an das Internet angebunden sein, muss ihm noch die IP-Adresse des zuständigen Routers mitgeteilt werden. Wenn es, wie im dargestellten Szenario, nur einen Router gibt, der für die Weiterleitung der Daten in alle anderen Netze zuständig ist, wird dieser Router auch Standard-Gateway oder Default-Gateway genannt.

Mit dem Kommando `route` wird der DSL-Router beim Rechner `linux1` als Standard-Gateway eingetragen:

```
route add default gw 192.168.50.254
```

Wenn mit `route` die Routingtabelle des Linux-Clients ausgegeben und bei der Ausgabe in der linken Spalte der Eintrag `default` vorhanden ist, so ist das Standard-Gateway korrekt eingetragen.

```
route
Kernel-IP-Routentabelle
Ziel          Router          Genmask          Flags Metric Ref    Use Iface
default       192.168.50.254  0.0.0.0          UG    0      0      0 eth0
link-local    *                255.255.0.0      U      1000   0      0 eth0
192.168.50.0  *                255.255.255.0    U      1      0      0 eth0
```

Jetzt sollte bereits eine Verbindung mit Rechnern im Internet möglich sein, dies kann z. B. mit einem `ping` auf den öffentlichen Google-DNS-Server geprüft werden:

```
ping 8.8.8.8
```

Konfiguration der Namensauflösung

Damit von `linux1` aus Rechner im Internet auch über Ihre DNS-Namen erreichbar werden, muss noch die Namensauflösung konfiguriert werden. Dazu wird der zu verwendende DNS-Server mit einem Editor in der Datei `/etc/resolv.conf` eingetragen. Im Normalfall wird der DSL-Router die Namensauflösung übernehmen, dann lautet der Eintrag in `/etc/resolv.conf` von `linux1` wie folgt:

```
nameserver 192.168.50.254
```

Ob die Namensauflösung funktioniert, kann man mit den Kommandos `nslookup` (veraltet) oder `dig` prüfen:

```
dig musin.muenchen.de

; <<>> DiG 9.7.3 <<>> musin.muenchen.de
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12028
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;musin.muenchen.de.          IN      A

;; ANSWER SECTION:
musin.muenchen.de.         5       IN      A       62.157.140.133
musin.muenchen.de.         5       IN      A       80.156.86.78

;; Query time: 4 msec
;; SERVER: 192.168.50.254#53(192.168.50.254)
;; WHEN: Mon Apr 16 18:27:54 2012
;; MSG SIZE rcvd: 67

nslookup musin.muenchen.de
Server:          192.168.50.254
Address:         192.168.50.254#53

Non-authoritative answer:
```

```
Name: musin.muenchen.de
Address: 62.157.140.133
Name: musin.muenchen.de
Address: 80.156.86.78
```

Sollte die Namensauflösung mit dem DSL-Router nicht funktionieren, muss möglicherweise anstatt der IP-Adresse die Adresse des DNS-Servers des Internet-Providers (z.B. 194.25.2.132) oder ein anderer DNS-Server in `/etc/resolv.conf` eingetragen werden.

Wenn in kleinen Netzen oder für Einzelrechner kein DNS-Server zur Verfügung steht, können auch über die Datei `/etc/hosts` einzelne Rechnernamen den zugehörigen IP-Adressen zugeordnet werden.

Netzwerkkonfiguration dauerhaft speichern

Um die Netzkonfiguration dauerhaft zu speichern, ist es am einfachsten, die distributionseigenen Administrationstools zu verwenden: z. B. bei Ubuntu über die Netzwerkeinstellungen im Gnome-Control-Center, bei SUSE mit `yast`.

Die Konfigurationsdateien der Netzwerkkonfiguration finden sich bei den verschiedenen Distributionen an unterschiedlichen Stellen:

Debian-Distributionen wie Ubuntu	<code>/etc/network/interfaces</code> Hilfe dazu mit <code>man 5 interfaces</code>
SUSE	Konfigurationsdateien im Verzeichnis <code>/etc/sysconfig/network/</code> für <code>eth0</code> z. B. <code>ifcfg-eth0</code>
Red Hat	Konfigurationsdateien im Verzeichnis <code>/etc/sysconfig/networkscripts/</code> für <code>eth0</code> z. B. <code>ifcfg-eth0</code>

6.6 Automatische Netzwerkkonfiguration mit DHCP

Das manuelle Einstellen der Netzkonfiguration ist umständlich und fehlerträchtig, eine automatische Konfiguration über das Dynamic Host Configuration Protocol (DHCP) ist wesentlich komfortabler. Bei DHCP wird die Netzwerkkonfiguration vom Client angefordert. Ein DHCP-Server antwortet auf die Anfrage und versorgt den Client mit allen notwendigen Informationen (z. B. IP-Adresse, Maske, Gateway, DNS-Server, usw.).

Alle Distributionen sind über ihre Administrationstools standardmäßig bereits nach der Installation auf DHCP eingestellt. Dann laufen die Programme `dhcpcd` bzw. `dhclient` im Hintergrund und fordern per DHCP die Netzkonfiguration an. Im oben abgebildeten Netzwerk-Szenario würde es also reichen, den in DSL- Routern normalerweise vorhandenen DHCP-Server zu aktivieren.

**Übung:**

Rechner in das Netzwerk einbinden und statisch konfigurieren

1. Fragen Sie die IP-Konfiguration Ihres Rechners ab und geben Sie folgendes an:

IPv4-Adresse: _____

IPv4-Netz: _____

IPv4-Broadcast: _____

IPv6-Adresse: _____

2. Ermitteln Sie die IPv4- und IPv6-Adressen Ihres Nachbarn und testen Sie mit `ping` die IPv4- und mit `ping6` die IPv6-Konnektivität zu ihm.
3. Ihr Rechner ist im Moment so eingestellt, dass er DHCP verwendet. Geben Sie das Kommando an, mit dem Sie das erste Ethernet-Interface auf die in der ersten Aufgabe ermittelte IPv4-Adresse einstellen.
4. Fragen Sie die IP-Konfiguration Ihres Rechners noch einmal ab und prüfen Sie, ob Sie Ihren Nachbarn nach wie vor über IPv4 erreichen können.
5. Stellen Sie das Standardgateway ein. Verwenden Sie die durch die Kursleiter angegebene IPv4-Adresse des Routers. Testen Sie das korrekte Routing durch einen `ping` auf 8.8.8.8.
6. Konfigurieren Sie die Namensauflösung. Verwenden Sie die durch die Kursleiter angegebene IPv4-Adresse des DNS-Servers. Testen Sie die Namensauflösung für `musin.muenchen.de`.
7. Jetzt soll die in den vorherigen Aufgaben durchgeführte Netzwerk-Konfiguration dauerhaft gespeichert werden. Dazu müssen Sie diese bei einem auf Debian-Linux basierenden Ubuntu-System in die Datei `/etc/network/interfaces` eintragen. Da der Rechner jetzt noch auf eine automatische Konfiguration über DHCP eingestellt ist, sieht in dieser Datei der Eintrag für das erste Ethernet-Interface folgendermaßen aus:

```
auto eth0
iface eth0 inet dhcp
```

Ändern Sie jetzt in der `/etc/network/interfaces` den Eintrag für das erste Ethernet-Interface und tragen Sie Ihre oben ermittelte IPv4-Konfiguration bei xxx ein:

```
auto eth0
iface eth0 inet static
    address 192.168.1.xxx # zu benutzende IPv4 Adresse
    netmask 255.255.255.0 # Netzwerkmaske
    gateway 192.168.1.xxx # Standard-Gateway
```

Starten Sie Ihren Rechner neu und testen Sie, ob er so wie Sie es eingestellt haben, konfiguriert wird.



Übung:

Webserver installieren

1. Für das Bereitstellen der Bilder der Klassenfahrt, soll auf dem im Netzwerk-Szenario (siehe Abbildung 6.6 auf Seite 121) dargestellten Intranet-Server der Webserver Lighttpd (gesprochen lighty) installiert werden. Hierbei handelt es sich um einen schlanken und sehr schnellen HTTP-Server, der darüber hinaus einfach zu konfigurieren ist.

Der Lighttpd-Server wird wie folgt installiert:

```
sudo apt-get install lighttpd
```

Nach der Installation ist der Server bereits aktiv und sollte auf allen Netzwerk-Interfaces Anfragen auf dem HTTP-Standard-Port 80 entgegen nehmen.

Dies kann auf dem Server an der Ausgabe des folgenden Kommandos gezeigt werden:

```
netstat -apn | grep ":80"
tcp        0      0  0.0.0.0:80          0.0.0.0:*        LISTEN     3939/lighttpd
```

Die Ausgabe bedeutet, dass das Programm lighttpd mit der Prozessnummer 3939 auf allen Interfaces (0.0.0.0) über TCP auf dem Port 80 (HTTP) lauscht.

Um den Server zu testen, kann man zunächst mit

```
echo '<HTML><BODY><P>es funktioniert !</P></BODY></HTML>' > /var/www/index.html
```

eine Startseite generieren und diese dann in einem Browser anzeigen lassen.

Dazu muss in der Adresszeile des Browsers lediglich die IP-Adresse dieses Servers angegeben werden.

Hat das nicht funktioniert, sollte mit

```
service lighttpd status
```

der Status des Servers überprüft werden, möglicherweise muss der Lighttpd-Server mit

```
service lighttpd start
```

erst gestartet werden.

Auftrag: Installieren und testen Sie den Lighttpd-Server wie oben dargestellt.

2. Lighttpd lässt sich durch Module um zusätzliche Funktionen erweitern. Beispielsweise können so Webseiten aus den Home-Verzeichnissen der lokalen Benutzer gelesen werden.

Vorher muss das entsprechende Modul von root mit dem Kommando

```
lighty-enable-mod userdir
```

aktiviert und der Webserver mit

```
service lighttpd start
```

neu gestartet werden.

Ein Benutzer, der diese Funktion nutzen möchte, legt in seinem Home-Verzeichnis das Verzeichnis `/public_html` an und stellt so die dort hinterlegten Dateien über `http://ip-des-servers/~benutzername/` zur Verfügung. Wichtig ist, dass das Benutzer-Konto `www-data`, unter dem der Webserver läuft, Leserechte darauf besitzt.

Weitere Informationen zur Konfiguration von Lighttpd lassen sich unter <http://wiki.ubuntuusers.de/lighttpd> finden.

Auftrag: Erweitern Sie Lighttpd so, dass Webseiten aus den Home-Verzeichnissen der lokalen Benutzer gelesen werden können.

**Übung:**

für Gute: SSH-Zugang auf dem Intranet-Server einrichten

1. Mit `which sshd` können Sie auf dem Intranet-Server testen, ob der SSH-Server auf diesem System vorhanden ist. Ist die Antwort negativ, installieren Sie dort den SSH-Server mit

```
sudo apt-get install openssh-server
```

2. Starten Sie den SSH-Server mit `sudo service ssh start` und melden Sie sich testweise als `user1` über SSH an:

```
ssh user1@192.168.50.242
```

und melden Sie sich gleich wieder ab:

```
exit
```

Um für einen Benutzer einen SSH-Zugang ohne Passwort einzurichten, sind folgende Schritte nötig:

- der Benutzer erzeugt auf dem Client ein Schlüsselpaar (öffentlicher und privater Schlüssel)
- der erzeugte öffentliche Schlüssel wird auf dem Server im Home-Verzeichnis des Benutzers an die Datei `~/.ssh/authorized_keys` angehängt

3. Erzeugen Sie als `user1` auf dem Client ein Schlüsselpaar:

```
ssh-keygen -t rsa
```

Übernehmen Sie den vorgeschlagenen Dateinamen `/home/user1/.ssh/id_rsa` und drücken Sie zweimal , um eine leere Passphrase zu erhalten.

4. Kopieren Sie als `user1` mit secure copy (`scp`) den öffentlichen Schlüssel `/home/user1/.ssh/id_rsa.pub` in das Home-Verzeichnis von `user1` auf dem SSH-Server:

```
scp /home/user1/.ssh/id_rsa.pub user1@192.168.50.242:~/.ssh/id_rsa.client
```

5. Melden Sie sich vom Client aus noch einmal als `user1` beim SSH-Server an

```
ssh user1@192.168.50.242
```

und machen Sie den öffentlichen Schlüssel von `user1` dort bekannt:

```
cat ~/.ssh/id_rsa.client >> ~/.ssh/authorized_keys
```

6. Nach dem Abmelden sollte sich `user1` vom Client aus ohne Passwort beim SSH-Server anmelden können. Probieren Sie es aus!

Wichtig:

Achten Sie darauf, dass das Verzeichnis `~/.ssh` im Home-Verzeichnis die Rechte `-rwx-----` besitzt, da sich sonst der SSH-Daemon weigert, mit dem Verzeichnis zu arbeiten und kein Login über `ssh` möglich ist.

6.7 Fragen

1. Was ist das Internet?

2. Nennen Sie einige Protokolle der TCP/IP-Protokollfamilie.

3. Wozu dient SSH und welche Vorteile hat es gegenüber Telnet?

4. Mit welchem Anwendungsprotokoll werden Daten von einem Webserver übertragen?

5. Was ist ein Port?

6. Aus wie vielen Bit besteht eine IPv4-Adresse?

7. Wozu wird die Subnet-Maske benötigt?

8. Geben Sie die binäre Darstellung der IP-Adresse 192.168.1.1 an.

9. Geben Sie die IPv4- und IPv6-Loopback-Adressen an.

10. Geben Sie die verkürzte Darstellung der IPv6-Adresse 2001:0db8:0000:0815:4711:0000:0028:03f1 an.

11. Wie können Sie die Konfiguration aller vorhandenen Netzwerkkarten abfragen?

12. Mit welchem Kommando prüfen Sie, ob ein Rechner erreichbar ist?

13. Was ist ein Standard-Gateway und mit welchem Kommando wird es gesetzt?

14. Wozu wird DNS benötigt?

15. In welcher Datei muss der zu benutzende DNS-Server eingetragen werden?

16. Geben Sie zwei verschiedene Kommandos an, mit denen die Namensauflösung getestet werden kann.

17. Über welches Anwendungs-Protokoll kann die Netzwerkkonfiguration automatisch erfolgen?

18. Geben Sie für das oben dargestellte Netzwerk-Szenario (siehe Abbildung 6.6 auf Seite 121) die Schritte und Kommandos an, mit denen Sie den Linux-Client-PCs `linux2` in das Netzwerk einbinden und wie dargestellt konfigurieren.

7 Einheit VI – Daten sichern und komprimieren

Datensicherung ist ein wichtiger Punkt, der gerne übersehen bzw. vernachlässigt wird. Erst wenn die Daten weg oder beschädigt sind, denkt man daran, ob und wann man das letzte Backup gemacht hat.

Einige Daten kann man sich wieder besorgen, aber z. B. die Bilder der Klassenfahrt sind dann weg. Eine nochmalige Fahrt wäre zwar schön, aber meist nicht möglich und die Wahrscheinlichkeit, dass man genau die selben Bilder wieder neu erstellen kann, geht gegen Null.

In diesem Abschnitt wird gezeigt, wie man Daten mit verschiedenen Verfahren komprimiert und dann nach verschiedenen Kriterien, z. B. auf einen USB-Stick sichert.

Inhalt

7.1	Datenkomprimierung	130
7.1.1	gzip	130
7.1.2	bzip2	131
7.1.3	zip	132
7.1.4	tar	134
7.1.5	Andere Komprimierer	135
7.1.6	7-Zip	135
7.1.7	Fragen	136
7.2	Datensicherung	137
7.2.1	Sicherungsarten	137
7.2.2	Generationenprinzip	137
7.2.3	Backup	137
7.2.4	Restore	140
7.2.5	Fragen	140

7.1 Datenkomprimierung

Man unterscheidet zwischen zwei Komprimierungsarten:

- **verlustbehaftete Komprimierung**

Dabei wird der Datenstrom komprimiert und ein gewisser Qualitätsverlust wird in Kauf genommen, um eine geringere Datenmenge zu erhalten. Eine Dekomprimierung zum Original ist nicht mehr möglich. Solch ein Verfahren wird gerne bei Filmen, Audiodateien oder Bildern verwendet, z. B. bei jpg-Bildern.

Eine verlustbehaftete Komprimierung ist für eine Datensicherung **nicht** geeignet!

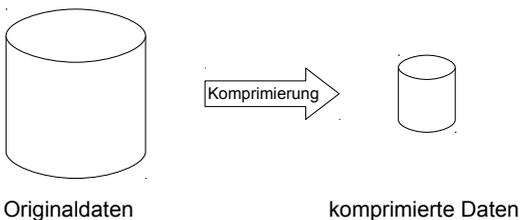
- **verlustfreie Komprimierung**

Dabei ist es jederzeit möglich, den komprimierten Datenstrom wieder in sein Original zurück zuführen.

Nur eine verlustfreie Komprimierung ist für eine Datensicherung geeignet.

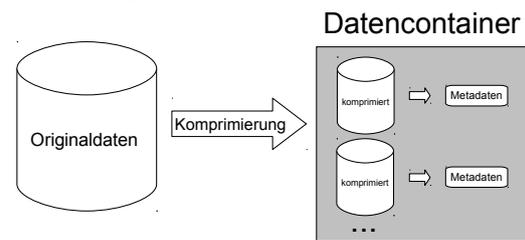
Die Daten werden auf zwei Arten komprimiert und weitergegeben:

als Datenstrom



Bei diesem Verfahren wird der Original-Datenstrom komprimiert und weitergegeben bzw. in einer Datei gespeichert. Dabei werden nur die reinen Daten komprimiert, Metadaten (bei einer Datei z. B. der Eigentümer, das Erstellungsdatum, etc.) gehen dabei verloren.

in einem Datencontainer



Hier werden nicht nur die Daten in komprimierter Form in einem Datencontainer gespeichert, sondern zu jeder gespeicherten Datei werden auch entsprechende Metadaten (z. B. Dateiname, Datum, Eigentümer, Zugriffsrechte, etc.) gespeichert. In einem Datencontainer können theoretisch beliebig viele Dateien und sogar Verzeichnisstrukturen gespeichert werden.

Eine Übersicht über viele Datenkomprimierungsprogramme findet man unter:

http://de.wikipedia.org/wiki/Liste_von_Datenkompressionsprogrammen

Beispieldatei

Für die Beispiele zur Datenkomprimierung und vor allem für den Vergleich wird die freie „Martin Luther Bibel von 1912“ als ASCII-Datei verwendet. Diese hat den Vorteil, dass es sich um eine reine ASCII-Datei handelt, frei von Lizenzen ist und eine Größe von 4,4 MiB hat. Diese kann unter http://bitflow.dyndns.org/german/MartinLuther/Martin_Luther_Uebersetzung_1912.txt heruntergeladen werden.

7.1.1 gzip

Das Kompressionsprogramm `gzip` (GNU zip) zählt zu den Datenstrom-Komprimierern und bietet einen guten Kompressionsgrad, ist frei von patentierten Algorithmen und findet sich in jedem Unix-/Linux-System. Es wird auch bei der Webseitenübertragung vom Server zum Client verwendet, um z. B. HTML-Seiten zu komprimieren, damit diese schneller übertragen werden können.

Zu dem Paket gehören insgesamt drei Kommandos:

```
gzip    komprimiert eine Datei bzw. einen Datenstrom
gunzip  dekomprimiert eine Datei bzw. einen Datenstrom
zcat    gibt eine komprimierte Textdatei aus.
```

Die wichtigsten Parameter von `gzip` sind:

```
-c      der Output wird nach stdout geschrieben und kann so von einem anderen Kommando weiter verwendet werden
-d      auspacken (wie gunzip)
--fast  es wird eine schnelle Komprimierung verwendet, die aber nicht so gut komprimiert
--best  es wird die beste Komprimierung verwendet, die aber mehr Zeit benötigt
```

Standardmäßig komprimiert `gzip` eine Datei und hängt die Dateiendung `.gz` an, danach wird das Original gelöscht. `gunzip` geht genau den umgekehrten Weg.

Im nachfolgenden Beispiel wird zuerst die Textdatei kopiert, so dass das Original nicht verändert wird. Die Datei wird dabei gepackt und wieder entpackt. Damit man sieht, wie gut die Datei komprimiert worden ist, wird mit `ls` jeweils die Dateigröße angezeigt¹.

```
cp bibel.txt Bcomp.txt
ls Bcomp.*
echo '*** packen *'
gzip Bcomp.txt
ls Bcomp.*
echo '*** entpacken *'
gunzip Bcomp.txt.gz
ls Bcomp.*
# alles wieder löschen
rm Bcomp.*
```

```
4364051 Bcomp.txt
*** packen *
1384697 Bcomp.txt.gz
*** entpacken *
4364051 Bcomp.txt
```

Die Originaldatei hat eine Größe von 4364051 Bytes und die gepackte Datei entspricht etwa 31,7%.

Möchte man, dass die Originaldatei von `gzip` nicht gelöscht wird, so muss man den Parameter `-c` verwenden und die Ausgabe umleiten.

```
cp bibel.txt Bcomp.txt
gzip -c Bcomp.txt > Bcomp.txt.gz
ls Bcomp.*
# alles wieder löschen
rm Bcomp.*
```

```
4364051 Bcomp.txt
1384697 Bcomp.txt.gz
```

7.1.2 bzip2

Das Kompressionsprogramm `bzip2` verhält sich ähnlich wie `gzip`, verwendet aber einen anderen Algorithmus, der meist besser komprimiert als `gzip`, jedoch deutlich mehr Zeit dafür braucht.

Diese Paket besteht aus:

```
bzip2    komprimiert eine Datei bzw. einen Datenstrom
bunzip2  dekomprimiert eine Datei bzw. einen Datenstrom
bzipcat  gibt eine komprimierte Textdatei aus.
```

Die wichtigsten Parameter sind dabei identisch zu `gzip`, so dass nur das entsprechende Kommando ausgetauscht werden muss.

```
cp bibel.txt Bcomp.txt
bzip2 -c Bcomp.txt > Bcomp.txt.bz2
ls Bcomp.*
# alles wieder löschen
rm Bcomp.*
```

```
4364051 Bcomp.txt
998272 Bcomp.txt.bz2
```

Man sieht hier deutlich, die komprimierte Datei ist kleiner geworden (22,9%).

Im nachfolgenden Beispiel wird mit dem Kommando `time` gemessen, wie lange die Ausführung dauert. Die Werte hängen stark davon ab, wie viele Prozesse noch parallel vom Betriebssystem verarbeitet werden, geben aber einen ganz guten Näherungswert ab.

```
cp bibel.txt Bcomp.txt
time gzip -c Bcomp.txt > Bcomp.txt.gz
time bzip2 -c Bcomp.txt > Bcomp.txt.bz2
# alles wieder löschen
rm Bcomp.*
```

```
real    0m0.260s
user    0m0.252s
sys     0m0.004s

real    0m0.401s
user    0m0.388s
sys     0m0.008s
```

Man sieht hier, `bzip2` braucht für die Komprimierung ca. 1,5 mal so viel Zeit.

¹ Das Kommando `ls` ist ein ganz einfaches Bash-Skript, welches das Kommando `ls -l` aufruft und nur die Spalten für die Dateigröße und den Dateinamen ausgibt. Wie man Spalten ausschneidet und verwendet, haben Sie bereits in Abschnitt III kennengelernt.

7.1.3 zip

Das Kompressionsprogramm `zip` (Gegenstück `unzip`) zählt zu den Container-Komprimierern und kann mehrere Dateien mit entsprechenden Metadaten in einem Archiv (Dateiendung `zip`) speichern.² Inzwischen werden hier mehrere Packalgorithmen verwendet, mit unterschiedlicher Packrate und Geschwindigkeit.

Die wesentlichen Vorteile von `zip` sind:

- es können theoretisch beliebig viele Dateien gespeichert werden
- Metadaten wie Dateiname, Eigentümer usw. werden mit gespeichert
- es können nachträglich Dateien hinzugefügt, gelöscht und erneuert werden
- es können Filter auf die Auswahl der Dateien oder das Datum verwendet werden
- die Daten können verschlüsselt werden³

Ruft man `zip --help` auf, so erhält man die wichtigsten Parameter.

```
zip --help
Copyright (c) 1990-2008 Info-ZIP - Type 'zip -L' for software license.
Zip 3.0 (July 5th 2008). Usage:
zip [-options] [-b path] [-t mmdyyyyy] [-n suffixes] [zipfile list] [-xi list]
The default action is to add or replace zipfile entries from list, which
can include the special name - to compress standard input.
If zipfile and list are omitted, zip compresses stdin to stdout.
-f  freshen: only changed files  -u  update: only changed or new files
-d  delete entries in zipfile    -m  move into zipfile (delete OS files)
-r  recurse into directories     -j  junk (don't record) directory names
-o  store only                   -l  convert LF to CR LF (-ll CR LF to LF)
-l  compress faster              -9  compress better
-q  quiet operation              -v  verbose operation/print version info
-c  add one-line comments        -z  add zipfile comment
-@  read names from stdin        -o  make zipfile as old as latest entry
-x  exclude the following names  -i  include only the following names
-F  fix zipfile (-FF try harder) -D  do not add directory entries
-A  adjust self-extracting exe   -J  junk zipfile prefix (unzipsfx)
-T  test zipfile integrity       -X  eXclude eXtra file attributes
-y  store symbolic links as the link instead of the referenced file
-e  encrypt                      -n  don't compress these suffixes
-h2 show more help
```

Da `zip` die Originaldatei im Normalfall nicht löscht, braucht diese nicht kopiert zu werden.

```
zip bibel.zip bibel.txt
lss bibel.*
# alles wieder löschen
rm bibel.zip
```

```
adding: bibel.txt (deflated 68%)
998344 bibel.tar.bz2
4364051 bibel.txt
1384837 bibel.zip
```

Mit `unzip --help` erhält man die wichtigsten Parameter.

```
unzip --help
UnZip 6.00 of 20 April 2009, by Debian. Original by Info-ZIP.

Usage: unzip [-Z] [-opts[modifiers]] file[.zip] [list] [-x xlist] [-d exdir]
Default action is to extract files in list, except those in xlist, to exdir;
file[.zip] may be a wildcard. -Z => ZipInfo mode ("unzip -Z" for usage).

-p  extract files to pipe, no messages      -l  list files (short format)
-f  freshen existing files, create none    -t  test compressed archive data
-u  update files, create if necessary       -z  display archive comment only
-v  list verbosely/show version info      -T  timestamp archive to latest
-x  exclude files that follow (in xlist)   -d  extract files into exdir

modifiers:
-n  never overwrite existing files         -q  quiet mode (-qq => quieter)
-o  overwrite files WITHOUT prompting     -a  auto-convert any text files
-j  junk paths (do not make directories)  -aa treat ALL files as text
-U  use escapes for all non-ASCII Unicode -UU ignore any Unicode fields
-C  match filenames case-insensitively   -L  make (some) names lowercase
-X  restore UID/GID info                  -V  retain VMS version numbers
-K  keep setuid/setgid/tacky permissions  -M  pipe through "more" pager
-O  CHARSET specify a character encoding for DOS, Windows and OS/2 archives
-I  CHARSET specify a character encoding for UNIX and other archives
```

See "unzip -hh" or `unzip.txt` for more help. Examples:

²Das Programm wurde von Phil Katz geschrieben, daher der ursprüngliche Name `pkzip` und `pkunzip`.

³Allerdings ist die Verschlüsselung nicht sehr gut und man findet diverse Tools im Internet, die versuchen, das Passwort zu entschlüsseln.

```
unzip data1 -x joe    => extract all files except joe from zipfile data1.zip
unzip -p foo | more  => send contents of foo.zip via pipe into program more
unzip -fo foo README => quietly replace existing README if archive file newer
```

Beim Entpacken mit `unzip` fallen ein paar Besonderheiten auf:

- das Datum der ursprünglichen Datei bleibt erhalten
- es können die UID, GID und die Rechte (SUID, GUID und das sticky-bit) der ursprünglichen Datei wieder hergestellt werden (nur `root` kann andere UID, GID etc. setzen)
- es kann verhindert werden, dass existierende Dateien überschrieben werden
- Textdateien können mit der Zeilenendemarkierung angepasst werden
Linux verwendet ein „line feed“-Zeichen (0x10), MAC ein „carriage return“ (0x13) und Windows beide Zeichen (0x13+0x10)

Sollen auch Unterverzeichnisse mit „eingepackt“ werden, so ist der Parameter `-r` zu verwenden.

```
zip -r bilder.zip kamera/
echo '*** Inhalt anzeigen ***'
unzip -l bilder.zip
# alles wieder löschen
rm bilder.zip
```

```
adding: kamera/ (stored 0%)
adding: kamera/img_cr2_0546.jpg (deflated 1%)
adding: kamera/IMAG0433.jpg (deflated 0%)
adding: kamera/IMG_8697.JPG (deflated 1%)
adding: kamera/IMG_8693.JPG (deflated 0%)
adding: kamera/img_cr2_0546.cr2 (deflated 8%)
*** Inhalt anzeigen ***
Archive: bilder.zip
  Length      Date    Time    Name
  -----
         0  2012-07-08  07:26   kamera/
    681938  2012-07-08  07:19  kamera/img_cr2_0546.jpg
   1366648  2012-07-08  07:19  kamera/IMAG0433.jpg
   2880926  2012-07-08  07:19  kamera/IMG_8697.JPG
   4399074  2012-07-08  07:19  kamera/IMG_8693.JPG
   8452365  2012-07-08  07:19  kamera/img_cr2_0546.cr2
  -----
  17780951                      6 files
```

Beim Packen von jpg-Bildern fällt besonders auf, dass die Kompressionsrate sehr klein ist. Dies liegt daran, dass das Bildformat jpg schon von Haus aus komprimiert ist und eine weitere Komprimierung wenig bis gar nichts mehr bringt. Einziger Vorteil, alle Bilder werden in eine Datei gepackt.

Hinweis: Wird der Schalter `-0` verwendet, so speichert `zip` die Datei nur im Container und komprimiert diese nicht.

Bei obigem Beispiel wurde auch die Datei `img_cr2_0546.cr2` mit gespeichert. Mit dem Parameter `-x` lassen sich bestimmte Dateien ausschließen, mit `-i` werden nur Dateien eingebunden, die dem Muster entsprechen. Dieser Parameter kann mehrmals verwendet werden, muss aber am Ende des Aufrufs stehen. Werden hier Wildcards verwendet, so muss der Eintrag mit einfachen Hochkommas versehen werden (Quoting).

```
zip -r bilder.zip kamera/ -x '*.cr2'
echo '*** Inhalt anzeigen ***'
unzip -l bilder.zip
# alles wieder löschen
rm bilder.zip
```

```
adding: kamera/ (stored 0%)
adding: kamera/img_cr2_0546.jpg (deflated 1%)
adding: kamera/IMAG0433.jpg (deflated 0%)
adding: kamera/IMG_8697.JPG (deflated 1%)
adding: kamera/IMG_8693.JPG (deflated 0%)
*** Inhalt anzeigen ***
Archive: bilder.zip
  Length      Date    Time    Name
  -----
         0  2012-07-08  07:26   kamera/
    681938  2012-07-08  07:19  kamera/img_cr2_0546.jpg
   1366648  2012-07-08  07:19  kamera/IMAG0433.jpg
   2880926  2012-07-08  07:19  kamera/IMG_8697.JPG
   4399074  2012-07-08  07:19  kamera/IMG_8693.JPG
  -----
   9328586                      5 files
```

7.1.4 tar

Das Archivierungsprogramm `tar` (tape archiver, Bandarchivierer) ist im Unix-/Linux-Umfeld sehr verbreitet und zählt zu den Container-Komprimierern. Es war ursprünglich dafür gedacht, Dateien auf Magnetbändern zu sichern. Inzwischen bietet `tar` die Möglichkeit, Dateien sequenziell in eine einzige Archiv-Datei zu schreiben bzw. auch wieder daraus zu extrahieren. Als Dateiergung wird hier `tar` verwendet⁴.

Ursprünglich war `tar` nicht in der Lage, das Archiv auch zu komprimieren, aber in Zusammenarbeit mit u. a. `gzip` bzw. `bzip2` lassen sich die Archive auch komprimieren (Dateiergung `tar.gz` bzw. `tar.bz2`).

```
tar [Parameter] <DATEI> ...
```

Die wichtigsten Parameter von `tar` sind:⁵

- c erstellt ein Archiv
- t listet den Inhalt auf
- x entpackt ein Archiv
- r Dateien werden angefügt
- u Dateien, die neuer sind als die im Archiv, werden hinzugefügt
- f es wird eine Datei zum Speichern verwendet
- v Ausgabe von Meldungen
- z es wird `gzip` für die Komprimierung verwendet
- j es wird `bzip2` für die Komprimierung verwendet
- C beim Entpacken vorher in das angegebene Verzeichnis springen
- p beim Entpacken Rechte wieder herstellen (nur `root` kann andere UID, GID usw. setzen)

Beispiele:

Dateien nur archivieren und nicht komprimieren.

```
tar cvf bibel.tar bibel.txt
ls bibel.*
# alles wieder löschen
rm bibel.tar
```

```
bibel.txt
4372480 bibel.tar
998344 bibel.tar.bz2
4364051 bibel.txt
```

Da `tar` zusätzlich noch die Metadaten speichert, ist das unkomprimierte Format größer als das Original.

Dateien archivieren und komprimieren mit `gzip`

```
tar cvzf bibel.tar.gz bibel.txt
ls bibel.*
# alles wieder löschen
rm bibel.tar.gz
```

```
bibel.txt
1384860 bibel.tar.gz
4364051 bibel.txt
```

Dateien archivieren und komprimieren mit `bzip2`

```
tar cvjf bibel.tar.bz2 bibel.txt
ls bibel.*
# alles wieder löschen
rm bibel.tar.bz2
```

```
bibel.txt
998344 bibel.tar.bz2
4364051 bibel.txt
```

Dateien archivieren, komprimieren mit `bzip2` und dann den Inhalt anzeigen lassen

```
tar cjf bibel.tar.bz2 bibel.txt
tar tvjf bibel.tar.bz2
# alles wieder löschen
rm bibel.tar.bz2
```

```
-rw-rw---- mgn/privat 4364051 2012-07-08 07:19 bibel.txt
```

Dateien archivieren, komprimieren mit `bzip2` und dann in ein anderes Verzeichnis entpacken (Rechte, Datum, etc. bleiben erhalten)

```
tar cjf bibel.tar.bz2 bibel.txt
mkdir test
tar tvjpf bibel.tar.bz2 -C test
ls -l test/
# alles wieder löschen
rm -r test
```

```
-rw-rw---- mgn/privat 4364051 2012-07-08 07:19 bibel.txt
insgesamt 0
```

⁴ Eine solche Datei wird auch gerne als Tarball (Teerklumpen oder Teerkugel) bezeichnet.

⁵ Wie `ps` unterstützt `tar` einmal Parameter mit `'-'`, mit `--` und ohne `'-'`

Kopieren von mehreren Dateien/Verzeichnissen

Normalerweise werden Dateien/Verzeichnisse mit `cp` kopiert, gibt man `-r` mit, sogar rekursiv. Voraussetzung ist aber, dass kein Verzeichnis des Zielbaumes schon existiert. Zusätzlich kommt hinzu, dass nur der Inhalt der Dateien, aber nicht z. B. die Zugriffsrechte kopiert werden..

Mit `tar` kann man diese Probleme sehr leicht umgehen:

```
tar cf - <Quell-Verzeichnis> | ( cd <Ziel-Verzeichnis>; tar xvf - )
```

Mit dem Parameter `-` wird nicht in eine Datei geschrieben, sondern in `stdout` und diese wird weitergeleitet (mit einer Pipe). Dort wird dann in das Zielverzeichnis gewechselt und der empfangene Datenstrom wieder entpackt (`-` sorgt hier dafür, dass von `stdin` gelesen wird). Wird zusätzlich beim Entpacken noch `-p` angegeben, so bleiben die Rechte etc. erhalten. Natürlich könnte man auch noch `gzip` bzw. `gunzip` verwenden, dies würde aber das Kopieren verlangsamen, da das Packen und Entpacken unnötig Zeit kostet und hier keinen Vorteil bringt.

7.1.5 Andere Komprimierer

Es kommt oft vor, dass gerade aus der Windowswelt Archive kommen, die mit anderen Komprimierern gepackt sind. In den meisten Fällen lassen sich auch diese Archive unter Linux verarbeiten.

rar

Das Kompressionsprogramm `rar` (Gegenstück `unrar`) zählt zu den Container-Komprimierern (Paket `rar` und `unrar`) und wurde von Jewgeni Roshal entwickelt, daher der Name **R**oshal **A**Rchive. Die Software ist nicht frei, sondern nur Shareware und muss nach 40 Tagen Benutzung registriert werden. Für Linux gibt es auch einen freien Unpacker (Paket `unrar-free`).

Rar konnte sich trotz höhere Komprimierungsraten gegenüber `zip` nur in einigen Bereichen durchsetzen, z. B. beim Austauschen über das Internet. Das Manko ist die nicht freie Lizenz.

```
rar [Parameter] <ARCHIV> <DATEI> ...
```

Die wichtigsten Parameter von `rar` sind:

- a fügt Dateien zu einem Archiv hinzu
- d löscht Dateien aus einem Archiv
- e entpackt ein Archiv
- x entpackt ein Archiv mit voller Verzeichnisstruktur
- f es wird eine Datei zum Speichern verwendet
- t testet ein Archiv

7.1.6 7-Zip

Das Kompressionsprogramm `7z` (7-Zip) zählt zu den Container-Komprimierern, wurde von Igor Wiktorowitsch Pawlow entwickelt und steht unter der LGPL frei zur Verfügung. Die Dateinamenserweiterung ist „7z“.

Die wichtigsten Eigenschaften sind:

- Unterstützung großer Dateien mit bis zu 16 Exabyte
- Archiv-Header-Komprimierung
- starke Verschlüsselung mit AES-256

Aber `7z` ist für ein Backup eines Linux-Systems nicht geeignet, da es sämtliche Zugriffsrechte ignoriert.

Die wichtigsten Parameter von `7z` sind:

- a Dateien/Verzeichnisse einem Archiv hinzufügen bzw. eine Archivdatei erstellen
- d Dateien/Verzeichnisse aus einem Archiv löschen
- e einfaches Entpacken aller Dateien aus dem Archiv
- x Archiv entpacken und dabei die Verzeichnisstruktur erhalten
- l Inhalt eines Archivs auflisten
- t Archiv auf Beschädigungen prüfen
- u Archivinhalt aktualisieren

Beispiel mit 7z.

```
7z a bibel.7z bibel.txt
echo '*****'
ls bibel.*
# alles wieder löschen
rm bibel.7z
```

```
7-Zip [64] 9.20 Copyright (c) 1999-2010 Igor Pavlov 2010-11-18
p7zip Version 9.20 (locale=de_DE.UTF-8,Utf16=on,HugeFiles=on,4 CPUs)
Scanning

Creating archive bibel.7z

Compressing bibel.txt

Everything is Ok
*****
1041165 bibel.7z
4364051 bibel.txt
```

7.1.7 Fragen

1. Was versteht man unter verlustbehafteter bzw. verlustfreier Komprimierung und welche Art ist für die Datensicherung geeignet?

2. Welche Vor- / Nachteile ergeben sich bei einem Datenstromkomprimierer im Gegensatz zu einem Komprimierer, der einen Datencontainer verwendet.

3. Wie erreicht man bei `gzip`, dass die Originaldatei nicht gelöscht wird?

4. Wie können bei `zip` auch Unterverzeichnisse mit gepackt werden?

5. Wie wird der Inhalt eines `zip`-Archivs angezeigt, ohne dies zu entpacken?

6. Wie wird verhindert, dass bei einem `zip`-Archiv auch Dateien mit der Endung `~` und `.backup` gepackt werden?

7. Wie lautet der Aufruf, wenn mit `tar` das Verzeichnis `hugo` mit kompletten Inhalt gesichert und mit `bzip2` gepackt werden soll?

8. Wie wird der Inhalt eine `tar`-Archivs angezeigt (mit `bzip2` gepackt), ohne dies zu entpacken?

9. Wie wird ein `tar`-Archiv entpackt, so dass die Original-Rechte wieder hergestellt werden?



Übung:

1. Probieren Sie obige Beispiele selber aus.
2. Erstellen Sie eine Übersichts-Tabelle, die verschiedene Dateiformate (Text, png-Bilder, jpg-Bilder, svg-Bilder, ...) komprimiert und jeweils den Komprimierungsfaktor und die verwendete Zeit auflistet.
Hinweis: Verwenden Sie ausreichend große Dateien, da sonst die Ergebnisse nicht wirklich aussagekräftig sind.

7.2 Datensicherung

Die Datensicherung dient dem Schutz vor Datenverlust. Diesen Satz sollte man beherzigen, denn erst wenn die Daten weg oder beschädigt sind, denken viele das erste Mal über ein Backup nach.

7.2.1 Sicherungsarten

Man unterscheidet:

Speicherabbild-Sicherung	Der komplette Datenträger (meist die Festplatte bzw. Partition) wird durch ein 1:1-Abbild gesichert. Somit werden nicht nur die Nutzerdaten, sondern das ganze Dateisystem, evtl. mit Betriebssystem und Benutzereinstellungen, gesichert.
Komplett- oder Vollsicherung	Die zu sichernden Daten (Verzeichnisse und Dateien) werden komplett auf das Sicherungsmedium, z. B. Bandlaufwerk, Wechselfestplatte oder USB-Stick, übertragen.
Differenzielle Sicherung	Alle Daten, die seit der letzten Komplettsicherung geändert wurden oder neu hinzugekommen sind, werden gespeichert.
Inkrementelle Sicherung	Hierbei werden nur die Dateien gespeichert, die seit der letzten inkrementellen Sicherung, oder beim ersten Mal seit der letzten Komplettsicherung, geändert wurden oder neu hinzugekommen sind.

7.2.2 Generationenprinzip

Eine Strategie der Datensicherung ist das Generationenprinzip (Großvater-Vater-Sohn-Prinzip) und gehört zu den am häufigsten verwendeten Rotationsschemata. Damit wird sichergestellt, dass in verschiedenen zeitlichen Abstufungen (Großvater, Vater, Sohn) verschiedene Versionen zur Verfügung stehen.

Ist die letzte Sicherung („Sohn“-Daten) beschädigt, so lassen sich die Daten aus der älteren Sicherung („Vater“-Daten) wieder herstellen, usw.

Beispiel

Bei einer täglichen Sicherung ergibt sich:

- 4 Medien für Montag bis Donnerstag bei täglicher Sicherung (Sohn)
- 4 Medien für vier Freitage eines Monats (Vater)
- 12 Medien für jeweils den letzten Tag im Monat (Großvater)

Dies bedeutet, mit 20 Speichermedien (vorausgesetzt, die Datensicherung passt jeweils auf ein Medium) hat man das Jahr abgedeckt.

Nach diesem Prinzip könnte man stets auf eine Sicherung von

- jedem der letzten vier Werktage (Sohn-Sicherungen)
- jedem der letzten vier Freitage (Vater-Sicherungen)
- jedem der letzten zwölf Monatsenden (Großvater-Sicherungen)

zugreifen.

7.2.3 Backup

Im vorherigen Abschnitt habe Sie die Möglichkeiten von Container-Komprimierern kennen gelernt. Diese eignen sich hervorragend für die Datensicherung, also um ein Backup zu erstellen. Man kann die Datensicherung immer von „Hand“ durchführen oder mit einem Skript einen gewissen Automatismus verwenden.

Für „unsere“ (private) Datensicherung wird `tar` mit `bzip2` verwendet. Als Speichermedium wird ein USB-Stick benutzt⁶. Für die nachfolgenden Beispiele wird das Unterverzeichnis `kamera` gesichert. Als Ziel wird das `tmp`-Verzeichnis verwendet.

⁶ Ein USB-Stick ist nicht wirklich ein gutes Backup-Mittel, da dieser meist in der Hosentasche aufbewahrt wird, schnell verloren wird und leicht beschädigt werden kann.

Vollsicherung

Die Vollsicherung wird mit dem Skript `sichvoll` durchgeführt.

Für den Datennamen der Sicherung soll folgendes gelten:

```
sich_<was>_voll_<datum>.tar.bz2
```

Somit lässt sich am Dateinamen erkennen, was, wann und wie gesichert worden ist.

```
#!/bin/bash
quelle=$1
ziel=/tmp

# Verzeichnis als Parameter?
if [ -z $1 ] ; then
    quelle=kamera
fi

# Sonderzeichen entfernen
was=$(echo $quelle | tr '/' '___')

# Datum
datum=$(date +%Y-%m-%d_%H-%M-%S)

# Dateiname
datei=sich_${was}_voll_${datum}.tar.bz2

# absoluter Pfad
tardatei=${ziel}/${datei}

# Sichern
tar cjf ${tardatei} ${quelle}

lss ${tardatei}
```

```
16129145 /tmp/sich_kamera_voll_2012-07-08_07-28-19.tar.bz2
```

Inkrementelle Sicherung

Mit `tar` ist es auch möglich, eine inkrementelle Sicherung durchzuführen. Dazu wird mit dem Parameter `-g` zusätzlich eine Snapshot-Datei erstellt. Die Inkrementelle Sicherung wird mit dem Skript `sichinc` durchgeführt.

```
#!/bin/bash
quelle=$1
ziel=/tmp

# Verzeichnis als Parameter?
if [ -z $1 ] ; then
    quelle=kamera
fi

# Sonderzeichen entfernen
was=$(echo $quelle | tr '/' '___')

# Datum
datum=$(date +%Y-%m-%d_%H-%M-%S)

# Dateiname
datei=sich_${was}_inc_${datum}.tar.bz2

# Snapshot-Datei
snap=sich_${was}_inc.snar

# absoluter Pfad
tardatei=${ziel}/${datei}

# Sichern
tar cjf ${tardatei} ${quelle} \
    -g ${ziel}/${snap}

lss ${ziel}/sich_${was}_inc*
```

```
16138855 /tmp/sich_kamera_inc_2012-07-08_07-27-58.tar.bz2
159 /tmp/sich_kamera_inc.snar
```

Damit man sieht, wie die Inkrementelle Sicherung arbeitet, wird schrittweise eine Sicherung durchgeführt und jeweils dabei eine neue Datei erzeugt und erneut gesichert. Damit bei einem schnellen Rechner nicht der Dateiname (die Auflösung des Datums beträgt ja hier eine Sekunde) doppelt vorkommt und so die letzte Sicherung überschrieben wird, wird bei diesem Beispiel mit `sleep` eine Pause von zwei Sekunden eingebaut.

```
# zur Sicherheit alle 'alten'
# Backups löschen
rm -f /tmp/sich_kamera*

echo '*** Backup 1 *'
bash sichinc

cp bibel.txt kamera/bibel.txt
sleep 2
echo '*** Backup 2 *'
bash sichinc

echo 'Ein beliebiger Text' \
  > kamera/text.txt
sleep 2
echo '*** Backup 3 *'
bash sichinc

# erzeugte Dateien wieder löschen
rm -f kamera/bibel.txt kamera/text.txt \
  /tmp/sich_kamera*
```

```
*** Backup 1 *
16138799 /tmp/sich_kamera_inc_2012-07-08_07-26-36.tar.bz2
159 /tmp/sich_kamera_inc.snar
*** Backup 2 *
16138799 /tmp/sich_kamera_inc_2012-07-08_07-26-36.tar.bz2
998462 /tmp/sich_kamera_inc_2012-07-08_07-26-40.tar.bz2
170 /tmp/sich_kamera_inc.snar
*** Backup 3 *
16138799 /tmp/sich_kamera_inc_2012-07-08_07-26-36.tar.bz2
998462 /tmp/sich_kamera_inc_2012-07-08_07-26-40.tar.bz2
274 /tmp/sich_kamera_inc_2012-07-08_07-26-43.tar.bz2
180 /tmp/sich_kamera_inc.snar
```

Betrachtet man die Dateigrößen nach „Backup 3“, so erkennt man, dass jeweils nur die Änderung gesichert worden ist.

Automatisches Backup

Schön wäre es natürlich, wenn beim Einstecken des USB-Sticks automatisch das Backup durchgeführt wird. Dies ist mit ein wenig Aufwand möglich, sprengt allerdings den Rahmen dieses Skripts. Zum Beispiel wird unter <http://wiki.ubuntuusers.de/udev> beschrieben, wie dies realisiert wird.

Tägliches Backup

Legt man die Sicherung nicht auf einem USB-Stick ab, sondern auf einem Verzeichnis, welches ständig verfügbar ist, wie z. B. ein Netzlaufwerk, so kann man auch mit einem zeitgesteuerten Dienst (cron-Job) die Sicherung durchführen lassen (hier mit dem Kommando `anacron`).

Für die Zeitsteuerung gibt es spezielle Verzeichnisse, bei denen der cron-Job dafür sorgt, dass darin enthaltene Skripte automatisch zur bestimmten Zeit ausgeführt werden.

In der Datei `/etc/crontab` wird festgelegt, wann genau die Ausführung erfolgt.

```
# m h dom mon dow user  command
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 )
```

`/etc/cron.hourly` jede Stunde (jeweils zur 17ten Minute)

`/etc/cron.daily` jeden Tag (jeden Tag um 06:25 Uhr)

`/etc/cron.weekly` jede Woche (jeden Sonntag um 06:47 Uhr)

`/etc/cron.monthly` jeden Monat (jeden Ersten im Monat um 06:52 Uhr)

Ist der Rechner zu diesem Zeitpunkt nicht eingeschaltet, so sorgt `anacron` dafür, dass der Auftrag nachgeholt wird.

7.2.4 Restore

Eine Vollsicherung kann recht einfach zurück gespielt werden. Dazu ist nur das entsprechende Archiv zu entpacken.

Bei einer inkrementellen Sicherung spielt man die einzelnen Sicherungen nacheinander ein. Wichtig: Es darf kein Schritt ausgelassen werden.

```
tar xjf /tmp/sich_kamera_inc_2012-05-20_06-17-30.tar.bz2 -g /tmp/sich_kamera_inc.snar
tar xjf /tmp/sich_kamera_inc_2012-05-21_06-17-10.tar.bz2 -g /tmp/sich_kamera_inc.snar
tar xjf /tmp/sich_kamera_inc_2012-05-22_06-17-20.tar.bz2 -g /tmp/sich_kamera_inc.snar
```



Übung:

1. Probieren Sie alle Beispiele aus.
2. Sichern sie Ihr eigenes Home-Verzeichnis auf einem USB-Stick.

7.2.5 Fragen

1. Welche Sicherungsarten werden häufig verwendet? Nennen Sie auch jeweils min. einen Vorteil und Nachteil.

2. Was versteht man unter dem Generationenprinzip und welchen Vorteil bringt es?

3. Wie wird eine Inkrementelle Datensicherung mit `tar` realisiert?

8 Einheit VII – Webseite für die Bilder erstellen

Durch die viele Vorarbeit ist man jetzt in der Lage, die Bilder zu präsentieren. Dazu wird im ersten Schritt eine Bildergalerie mit Vorschaubildern für eine Webseite erstellt und diese dann auf dem Web-Server veröffentlicht.

Damit man beim Veröffentlichen von Bildern sich und anderen keine Probleme macht, muss auch überlegt werden, welche Lizenz man für die Bilder verwendet. Hier gibt es eine Vielzahl von Möglichkeiten.

Zum Schluss geht es noch an die Einstellungen und Erweiterungsmöglichkeiten des Browsers, so dass deutlich sicherer im Netz gesurft werden kann und die Privatsphäre gesichert ist.

Inhalt

8.1	Bildergalerie erstellen	142
8.1.1	HTML	142
8.1.2	Thumbnails anzeigen	143
8.1.3	Fragen	149
8.2	Galerie über einen Webserver veröffentlichen	150
8.3	Lizenzen	151
8.3.1	Übersicht	151
8.3.2	Welche Lizenz für eigene Bilder	154
8.3.3	Fragen	154
8.4	Firefox als sicherer Browser	155
8.4.1	JavaScript	155
8.4.2	Datenschutz	155
8.4.3	Werbung unterdrücken	155
8.4.4	Webseitenverfolgung verhindern	156
8.4.5	Verschlüsselte Verbindung verwenden	156
8.4.6	Speichern von Inhalten	156
8.4.7	Websuche	157

8.1 Bildergalerie erstellen

Für eine Bildergalerie benötigt man eine Web-Seite (HTML) mit entsprechenden Vorschaubildern, welche ja schon in Abschnitt IV erstellt worden sind. Klickt man auf das Vorschaubild, so soll dann das Bild in voller Größe angezeigt werden.

8.1.1 HTML

Für HTML gibt es unter <http://de.selfhtml.org/> eine Web-Seite, die das Thema sehr gut und ausführlich behandelt. Diese soll hier als zusätzliche Hilfe und Ergänzung dienen.

Das HTML-Grundgerüst sieht dabei wie folgt aus:¹

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2//EN">
2 <html>
3 <head>
4   <title>HTML-Seite</title>
5 </head>
6 <body>
7   <p>Eine einfache HTML-Seite</p>
8 </body>
9 </html>

```

Eine einfache HTML-Seite

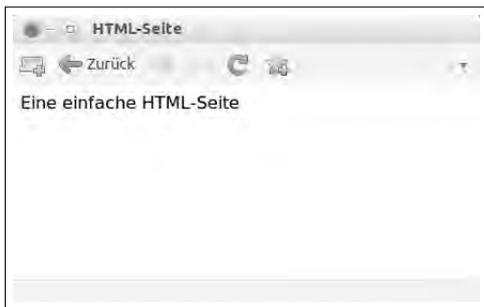


Abb. 8.1: Ansicht der Seite mit dem Midori-Browser

In Zeile 1 wird das Format (DOCTYPE) der HTML-Seite festgelegt. Hier wird HTML 3.2 als Grundlage genutzt. In Zeile 2 wird das HTML-Dokument mit dem Element `<html>` geöffnet und in Zeile 9 mit `</html>` wieder geschlossen. Der Titel (Der Text, der in der Browserleiste angezeigt wird) wird im Kopf (Element `<head>`) mit dem Element `<title>` definiert. Im Element `<body>` befindet sich der eigentliche Seiteninhalt. Der eigentliche Text wird mit dem Element `<p>` (Absatz) eingeschlossen.

Lesen Sie dazu auch

<http://de.selfhtml.org/html/allgemein/textauszeichnung.htm> und

<http://de.selfhtml.org/html/allgemein/grundgeruest.htm>.

Eine Übersicht über die HTML-Elemente finden Sie unter

<http://de.selfhtml.org/html/referenz/elemente.htm>,

über Attribute unter

<http://de.selfhtml.org/html/referenz/attribute.htm>.

¹ Die HTML-Seite wird mit dem Programm `wkhtmltopdf` in eine PDF-Seite umgewandelt und dann hier als Bild eingebunden (ohne Rahmen und Kopf).

8.1.2 Thumbnails anzeigen

Um eine einfache Übersicht zu erstellen, sollen die Bilder in einer Tabelle dargestellt werden.

Eine HTML-Tabelle hat dabei folgenden Aufbau:

```
<table border="1">
<tr>
  <td>A1</td>
  <td>A2</td>
  <td>A3</td>
  <td>A4</td>
</tr>
<tr>
  <td>B1</td>
  <td>B2</td>
  <td>B3</td>
  <td>B4</td>
</tr>
</table>
```

A1	A2	A3	A4
B1	B2	B3	B4

Baut man direkt Bilder ein, so wird das Element `` verwendet.

```
<table border="1">
<tr>
  <td></td>
  <td></td>
  <td></td>
  <td></td>
</tr>
<tr>
  <td></td>
  <td></td>
  <td></td>
  <td></td>
</tr>
</table>
```

A1	A2	A3	A4
B1	B2	B3	B4

Schritt 1: Thumbnails erzeugen

Im nächsten Schritt können aus den Bildern Thumbnails erzeugt werden (siehe Abschnitt 5.7 auf Seite 110). Die Bilder (22 Stück) befinden sich bei diesen Beispielen im Unterordner `berlin`. Beachten Sie, dass einige Effekte nur mit dem Bildformat `png` möglich sind. Mit dem Kommando `basename` wird das Verzeichnis und hier die Dateierweiterung `.jpg` abgeschnitten.

```
for file in berlin/berlin*.jpg; do
  destfile=berlin/thumb_$(basename $file .jpg).png
  echo -n "$file "
  convert $file -thumbnail 200 \
    \( +clone -alpha extract \
      -virtual-pixel black -spread 10 \
      -blur 0x3 -threshold 50% -spread 4 \
      -blur 0x.7 \) -alpha off \
    -compose Copy_Opacity -composite \
    $destfile
done
```

```
berlin/berlin_0001.jpg berlin/berlin_0002.jpg ↵
berlin/berlin_0003.jpg berlin/berlin_0004.jpg ↵
berlin/berlin_0005.jpg berlin/berlin_0006.jpg ↵
berlin/berlin_0007.jpg berlin/berlin_0008.jpg ↵
berlin/berlin_0009.jpg berlin/berlin_0010.jpg ↵
berlin/berlin_0011.jpg berlin/berlin_0012.jpg ↵
berlin/berlin_0013.jpg berlin/berlin_0014.jpg ↵
berlin/berlin_0015.jpg berlin/berlin_0016.jpg ↵
berlin/berlin_0017.jpg berlin/berlin_0018.jpg ↵
berlin/berlin_0019.jpg berlin/berlin_0020.jpg ↵
berlin/berlin_0021.jpg berlin/berlin_0022.jpg
```

Schritt 2: HTML Gerüst bauen

Nun kann das HTML-Gerüst gebaut werden. Dazu wird ein Hier-Dokument verwendet. In der Tabelle werden dabei in jeder Zeile vier Bilder angezeigt. Zur Vereinfachung wird in diesem Schritt nur der Dateiname angezeigt.

```
echo erstelle uebersicht1.html
# Header erzeugen
cat <<EOM >uebersicht1.html
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html>
<head>
  <title>Bilder-Galerie</title>
</head>
<body>
<table border="1">
EOM

cnt=0
for file in berlin/berlin*.jpg; do
  if [ $cnt -eq 0 ]; then
    echo "<tr>" >>uebersicht1.html
    fi
    echo "<td>$file</td>" >>uebersicht1.html
    cnt=$((cnt+1))
    mod=$((cnt % 4))
    if [ $mod -eq 0 ]; then
      echo "</tr>" >>uebersicht1.html
      cnt=0
    fi
  done
  if [ $mod -ne 0 ]; then
    echo "</tr>" >>uebersicht1.html
  fi

# Header erzeugen
cat <<EOM >>uebersicht1.html
</table>
</body>
</html>
EOM
```

```
erstelle uebersicht1.html
```

Damit immer die vier Spalten erstellt werden, wird mit einer Modulo-Berechnung festgestellt, ob schon Spalte vier erreicht ist, wenn ja, so wird die Tabellenzeile mit `</tr>` abgeschlossen. Für den Fall, dass die Anzahl der Bilder nicht einem Vielfachen von vier entspricht, muss nach der Schleife geprüft werden, ob noch ein `</tr>` eingefügt werden muss.

Hier das Ergebnis:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html>
<head>
  <title>Bilder-Galerie</title>
</head>
<body>
<table border="1">
<tr>
<td>berlin/berlin_0001.jpg</td>
<td>berlin/berlin_0002.jpg</td>
<td>berlin/berlin_0003.jpg</td>
<td>berlin/berlin_0004.jpg</td>
</tr>
<tr>
<td>berlin/berlin_0005.jpg</td>
<td>berlin/berlin_0006.jpg</td>
<td>berlin/berlin_0007.jpg</td>
<td>berlin/berlin_0008.jpg</td>
</tr>
<tr>
<td>berlin/berlin_0009.jpg</td>
<td>berlin/berlin_0010.jpg</td>
<td>berlin/berlin_0011.jpg</td>
<td>berlin/berlin_0012.jpg</td>
</tr>
<tr>
<td>berlin/berlin_0013.jpg</td>
<td>berlin/berlin_0014.jpg</td>
<td>berlin/berlin_0015.jpg</td>
<td>berlin/berlin_0016.jpg</td>
</tr>
<tr>
<td>berlin/berlin_0017.jpg</td>
<td>berlin/berlin_0018.jpg</td>
<td>berlin/berlin_0019.jpg</td>
<td>berlin/berlin_0020.jpg</td>
</tr>
<tr>
<td>berlin/berlin_0021.jpg</td>
<td>berlin/berlin_0022.jpg</td>
</tr>
</table>
</body>
</html>
```

Schritt 3: HTML Gerüst und Thumbnails verbinden

Jetzt muss das Gerüst mit dem Erstellen der Thumbnails kombiniert und die Bilder müssen über `` eingebunden werden.

```
echo erstelle uebersicht2.html
# Header erzeugen
cat <<EOM >uebersicht2.html
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html>
<head>
  <title>Bilder-Galerie</title>
</head>
<body>
<table border="1">
EOM

cnt=0
for file in berlin/berlin*.jpg; do
  if [ $cnt -eq 0 ]; then
    echo "<tr>" >>uebersicht2.html
  fi
  # Thumbnail erstellen
  destfile=berlin/thumb_$(basename $file .jpg).png
  echo -n "$file "
  convert $file -thumbnail 200 \
    \ ( +clone -alpha extract \
      -virtual-pixel black -spread 10 \
      -blur 0x3 -threshold 50% -spread 4 \
      -blur 0x.7 \) -alpha off \
      -compose Copy_Opacity -composite \
      $destfile
  echo "<td><img src=\"$destfile\" width=\"100\" >
  /></td>" >>uebersicht2.html
  cnt=$((cnt+1))
  mod=$((cnt % 4))
  if [ $mod -eq 0 ]; then
    echo "</tr>" >>uebersicht2.html
    cnt=0
  fi
done
if [ $mod -ne 0 ]; then
  echo "</tr>" >>uebersicht2.html
fi

# Header erzeugen
cat <<EOM >>uebersicht2.html
</table>
</body>
</html>
EOM
```

```
erstelle uebersicht2.html
berlin/berlin_0001.jpg berlin/berlin_0002.jpg ↵
berlin/berlin_0003.jpg berlin/berlin_0004.jpg ↵
berlin/berlin_0005.jpg berlin/berlin_0006.jpg ↵
berlin/berlin_0007.jpg berlin/berlin_0008.jpg ↵
berlin/berlin_0009.jpg berlin/berlin_0010.jpg ↵
berlin/berlin_0011.jpg berlin/berlin_0012.jpg ↵
berlin/berlin_0013.jpg berlin/berlin_0014.jpg ↵
berlin/berlin_0015.jpg berlin/berlin_0016.jpg ↵
berlin/berlin_0017.jpg berlin/berlin_0018.jpg ↵
berlin/berlin_0019.jpg berlin/berlin_0020.jpg ↵
berlin/berlin_0021.jpg berlin/berlin_0022.jpg ↵
```

Hier das Ergebnis:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html>
<head>
  <title>Bilder-Galerie</title>
</head>
<body>
<table border="1">
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
```

```

<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</table>
</body>
</html>

```



Abb. 8.2: Browser-Ansicht

Schritt 4: Link auf Bilder einfügen

Mit dem Element `<a>` lassen sich Links, z. B. auf die Bilder einfügen. Damit die Bilder jeweils in einem eigenen Tab im Browser geöffnet werden, wird das Attribut `target` verwendet.

```
echo erstelle uebersicht3.html
# Header erzeugen
cat <<EOM >uebersicht3.html
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html>
<head>
  <title>Bilder-Galerie</title>
</head>
<body>
<table border="1">
EOM

cnt=0
for file in berlin/berlin*.jpg; do
  if [ $cnt -eq 0 ]; then
    echo "<tr>" >>uebersicht3.html
    fi
    # Thumbnail erstellen
    destfile=berlin/thumb_${basename $file .jpg}.png
    echo -n "$file "
    convert $file -thumbnail 200 \
      \( +clone -alpha extract \
        -virtual-pixel black -spread 10 \
        -blur 0x3 -threshold 50% -spread 4 \
        -blur 0x.7 \) -alpha off \
        -compose Copy_Opacity -composite \
        $destfile
    echo "<td><a href=\"$file\" target=\"_blank\" >" \
      >>uebersicht3.html
    echo "<img src=\"$destfile\" width=\"100\" />" \
      >>uebersicht3.html
    echo "</a></td>" >>uebersicht3.html
    cnt=$((cnt+1))
    mod=$((cnt % 4))
    if [ $mod -eq 0 ]; then
      echo "</tr>" >>uebersicht3.html
      cnt=0
    fi
  done
  if [ $mod -ne 0 ]; then
    echo "</tr>" >>uebersicht3.html
  fi

# Header erzeugen
cat <<EOM >>uebersicht3.html
</table>
</body>
</html>
EOM
```

```
erstelle uebersicht3.html
berlin/berlin_0001.jpg berlin/berlin_0002.jpg ↵
berlin/berlin_0003.jpg berlin/berlin_0004.jpg ↵
berlin/berlin_0005.jpg berlin/berlin_0006.jpg ↵
berlin/berlin_0007.jpg berlin/berlin_0008.jpg ↵
berlin/berlin_0009.jpg berlin/berlin_0010.jpg ↵
berlin/berlin_0011.jpg berlin/berlin_0012.jpg ↵
berlin/berlin_0013.jpg berlin/berlin_0014.jpg ↵
berlin/berlin_0015.jpg berlin/berlin_0016.jpg ↵
berlin/berlin_0017.jpg berlin/berlin_0018.jpg ↵
berlin/berlin_0019.jpg berlin/berlin_0020.jpg ↵
berlin/berlin_0021.jpg berlin/berlin_0022.jpg ↵
```

Hier das Ergebnis:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html>
<head>
  <title>Bilder-Galerie</title>
</head>
<body>
<table border="1">
<tr>
<td><a href="berlin/berlin_0001.jpg" target="_blank" >

</a></td>
<td><a href="berlin/berlin_0002.jpg" target="_blank" >

</a></td>
<td><a href="berlin/berlin_0003.jpg" target="_blank" >

</a></td>
<td><a href="berlin/berlin_0004.jpg" target="_blank" >

</a></td>
</tr>
<tr>
```

```

<td><a href="berlin/berlin_0005.jpg" target="_blank" >

</a></td>
<td><a href="berlin/berlin_0006.jpg" target="_blank" >

</a></td>
<td><a href="berlin/berlin_0007.jpg" target="_blank" >

</a></td>
<td><a href="berlin/berlin_0008.jpg" target="_blank" >

</a></td>
</tr>
<tr>
<td><a href="berlin/berlin_0009.jpg" target="_blank" >

</a></td>
<td><a href="berlin/berlin_0010.jpg" target="_blank" >

</a></td>
<td><a href="berlin/berlin_0011.jpg" target="_blank" >

</a></td>
<td><a href="berlin/berlin_0012.jpg" target="_blank" >

</a></td>
</tr>
<tr>
<td><a href="berlin/berlin_0013.jpg" target="_blank" >

</a></td>
<td><a href="berlin/berlin_0014.jpg" target="_blank" >

</a></td>
<td><a href="berlin/berlin_0015.jpg" target="_blank" >

</a></td>
<td><a href="berlin/berlin_0016.jpg" target="_blank" >

</a></td>
</tr>
<tr>
<td><a href="berlin/berlin_0017.jpg" target="_blank" >

</a></td>
<td><a href="berlin/berlin_0018.jpg" target="_blank" >

</a></td>
<td><a href="berlin/berlin_0019.jpg" target="_blank" >

</a></td>
<td><a href="berlin/berlin_0020.jpg" target="_blank" >

</a></td>
</tr>
<tr>
<td><a href="berlin/berlin_0021.jpg" target="_blank" >

</a></td>
<td><a href="berlin/berlin_0022.jpg" target="_blank" >

</a></td>
</tr>
</table>
</body>
</html>

```

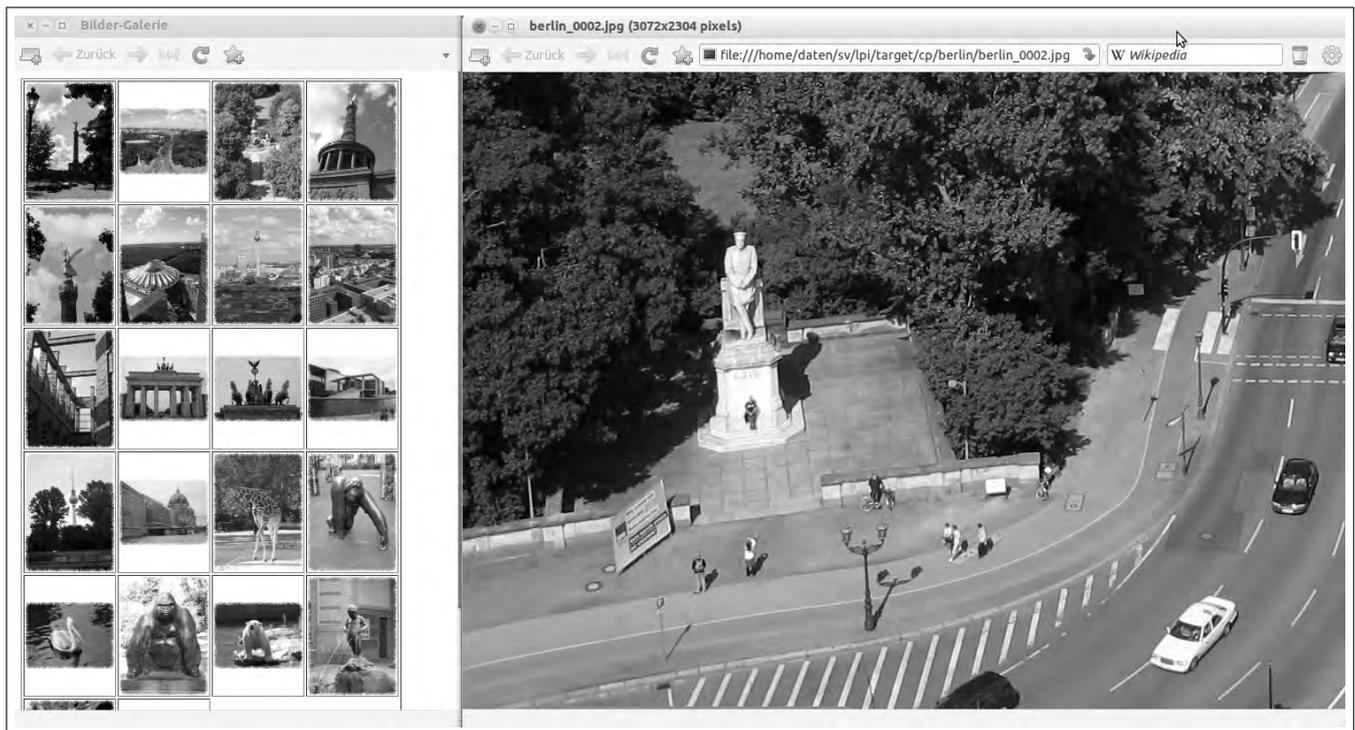


Abb. 8.3: Browser-Ansicht

Klickt man jetzt auf ein Vorschaubild, so öffnet sich je nach Browsereinstellung ein neues Fenster oder das Bild wird in einem neuen Tab angezeigt.

8.1.3 Fragen

1. Ermitteln Sie mit Hilfe von *selfhtml* die wichtigsten DOCTYPE-Definitionen und beschreiben Sie kurz deren Funktion.

2. Ermitteln Sie, welche Möglichkeiten mit einer Tabelle `<table>` und einem Bild `` noch möglich sind (z. B. Attribute, ...).

8.3 Lizenzen

Benötigt man Bilder, Texte oder andere Medien, so wird oft schnell die Suchmaschine „angeworfen“, nach dem Bild oder Text gesucht und per „copy & paste“ das Werk in sein eigenes Dokument übernommen. Die Meisten überlegen dabei nicht, unter welcher Lizenz so ein Werk steht. Kritisch wird es dann, wenn man fremde Werke in seinem Werk verwendet und der Rechteinhaber dies erfährt. Das kann zu einer Abmahnung führen.

Damit die Rechtesituation von Bildern etc. geklärt ist, sollte man beim Veröffentlichen seiner Werke deutlich kennzeichnen, unter welcher Lizenz diese veröffentlicht werden.

8.3.1 Übersicht

Nachfolgend werden einige wichtige Lizenzen genannt, die im Umfeld von Linux gerne verwendet werden. Es besteht kein Anspruch auf Vollständigkeit und Korrektheit bei rechtlichen Aussagen!

GNU GPL

Die „GNU General Public License“ (auch GNU GPL oder auch GPL genannt) ist eine von der Free Software Foundation (FSF) veröffentlichte Freie-Software-Lizenz mit Copyleft für die Lizenzierung von freier Software.²

Siehe hierzu:

- <http://www.gnu.org/licenses/gpl-3.0.html>
- <http://www.gnu.de/documents/gpl.de.html>³

Kurz gesagt, GPL bedeutet:

- Das Programm, der Code darf frei und ohne Lizenzgebühren etc. verwendet werden.
- Der Anwender
 - darf Einsicht in die Quelltexte eines Programms nehmen
 - darf die Quelltexte verändern, um die Software beispielsweise eigenen Bedürfnissen anzupassen
 - Teile des Codes für eigene Projekte verwenden
 - darf das Programm in unveränderter oder modifizierter Form weitergeben

Aber: Er muss für Veränderungen, Ergänzungen die in der GPL garantierten Rechte einräumen und somit den geänderten Quelltext wieder frei zur Verfügung stellen.

- Es besteht keinerlei Gewährleistung für das Programm.
- In keinem Fall ist der Entwickler für irgendwelche Schäden haftbar.

Copyleft

Das FSF versteht unter dem Begriff Copyleft die Kombination aus großen Freiheiten bei der Nutzung, mit der Verpflichtung, diese Freiheiten bei der Weitergabe oder Verwendung des Codes zu erhalten, d. h. Code, der unter der GPL steht, muss Open Source bleiben und Änderungen, Ergänzungen wieder unter der GPL gesetzt und zur Verfügung gestellt werden.

Open Source Philosophie

Zu Anfang verwendete man den Begriff „freie Software“. Durch Eric Raymond wurde der Begriff „Open Source“ geschaffen, da er glaubte, dass das Wort „frei“ die Entscheidungsträger aus der Wirtschaft abschrecken könnte.

Bei der freien Software steht mehr die politische Idee, also die Freiheit, im Vordergrund, wobei bei „Open Source“ in erster Linie ein Entwicklungsmodell der Hauptpunkt ist.

Richard Stallman sagte dazu folgendes:

„Angst vor Freiheit: Das Hauptargument für den Begriff 'Open-Source-Software' ist, dass 'Freie Software' manche Leute beunruhigt. Das ist wahr: Wenn man über Freiheit spricht, über ethische Fragen, über Verantwortung und Bequemlichkeit, dann verlangt man von den Menschen, dass sie über Dinge nachdenken, die sie vielleicht lieber ignoriert hätten. Das kann Unbehagen auslösen, und manche Leute lehnen die Vorstellung deswegen vielleicht ab. Daraus folgt nicht, dass die Gesellschaft besser aufgehoben wäre, wenn wir damit aufhören würden, über diese Themen zu sprechen.“⁴

² Die GNU GPL wurde im Januar 1989 von Richard Stallman, dem Gründer des GNU Projektes, geschrieben.

³ „Dies ist eine inoffizielle deutsche Übersetzung der GNU General Public License, die nicht von der Free Software Foundation herausgegeben wurde. Es handelt sich hierbei nicht um eine rechtsgültige Festlegung der Bedingungen für die Weitergabe von Software, die der GNU GPL unterliegt; dies leistet nur der englische Originaltext.“

⁴ Richard M. Stallman aus „The Selected Essays of Richard M. Stallman“

Auch für Unternehmen hat das Open-Source-Modell viel zu bieten.

- Es lassen sich offene Standards entwickeln. Dadurch können einzelne Personen und Unternehmen an einem Produkt arbeiten, was evtl. den Einzelnen überfordern würde. Der Nutzer bestimmt dabei den zeitlichen Ablauf der Entwicklung selbst; schnelle Fehlerkorrekturen und Anpassungen sind dadurch meist kein Problem mehr.
- Da der Code offen zugänglich ist, unterliegt er einer starken Kontrolle. Dies führt zu einer höheren Sicherheit. Eventuelle Sicherheitslücken werden durch viele Leute meist schneller erkannt und lassen sich so schnell beheben. Bei kommerzieller Software kommt es doch öfters vor, dass Lücken verschwiegen werden, bis sie von Hackern entdeckt und ausgenutzt werden.

Open-Source-Geschäftsmodelle

In der Literatur findet man eine Menge Ansätze für Geschäftsmodelle mit Open Source.

Viele Firmen setzen bei der Wertschöpfungskette im Bereich Beratung, zusätzliche Implementierung und vor allem im Training und Support an. Eine ausführliche Beschreibung dazu findet sich unter http://winfwiki.wi-fom.de/index.php/Gesch%C3%A4ftsmodelle_auf_Basis_des_Open_Source_Betriebssystems_LINUX.

GNU LGPL

Die „GNU Lesser General Public License“ (LGPL), wurde ursprünglich als „Library General Public License“ eingeführt, jedoch später umbenannt. Sie entspricht im Wesentlichen der GPL, mit der Ausnahme, dass Programme (auch nicht freie) mit einer LGPL-Bibliothek kombiniert (dynamisch verlinkt) werden können. Dies bedeutet, dass ein proprietäres Programm eine Bibliothek, die unter LGPL steht, verwenden kann. Somit ist die Verwendung von durch LGPL geschützten Bibliotheken zur Entwicklung kommerzieller bzw. nicht der GPL unterliegender Software ermöglicht. Änderungen an der Bibliothek selbst müssen allerdings wieder der LGPL unterliegen.

BSD

Die BSD Lizenz (Berkeley Software Distribution von der University of California) schränkt sowohl Programmierer als auch Anwender nur insofern ein, dass die ursprünglichen Autoren (Copyright-Vermerk) genannt werden müssen. Hauptpunkt der Lizenz ist der Ausschluss von Haftungsansprüchen gegenüber den Programmierern. Sourcecode, der der BSD-Lizenz unterliegt, kann in eigenen Entwicklungen benutzt werden, ohne dass diese wiederum freie Software sein müssen.

Public Domain

Public Domain stellt im Sinne von „Open Source“ keine Lizenz dar. Es sagt nur aus, dass der Autor auf jeglichen Einfluss auf sein Werk verzichtet, was bedeutet, dass jeder alles damit machen kann.

Das Public Domain der USA („Ich verzichte auf alle Rechte des Werkes.“) gilt in Deutschland nicht. In Deutschland ist Public Domain ein reines sehr weitreichendes Nutzungsrecht.

Siehe hierzu: <https://de.wikipedia.org/wiki/Gemeinfreiheit>

FLOSS/FOSS

Die Begriffe Free/Libre Open Source Software (FLOSS) bzw. Free and Open Source Software (FOSS) sind zusammengesetzte Begriffe für freie Software und Open-Source-Software.

Mehr dazu unter <http://www.ifossf.org/>.

Closed Source

Closed Source ist das Gegenteil von Open Source. Dies bedeutet, dass der Quellcode nicht verfügbar ist. Die Free Software Foundation empfiehlt, den Begriff Closed-Source-Software nicht zu verwenden und sie stattdessen als unfreie oder proprietäre Software zu bezeichnen.

Übersicht Softwarelizenzen

Lizenz	Kann mit kommerzieller Software verwendet werden	Eigene Veränderungen müssen wieder frei sein	Kann unter anderen Bedingungen veröffentlicht werden
GPL	Ja ⁵	Ja	Nein
LGPL	Ja	Ja	Nein
BSD	Ja	Nein	Ja ⁶
Public Domain	Ja	Nein	Ja

Weitere Übersichten in Englisch findet sich unter:

- http://en.wikipedia.org/wiki/Comparison_of_free_software_licences
- <http://www.gnu.org/licenses/gpl-faq>

Die Bundeszentrale für politische Bildung hat das Buch „Freie Software Zwischen Privat- und Gemeineigentum“ von Volker Grassmuck als PDF unter „Creative Commons Attribution-ShareAlike 2.0 License“ veröffentlicht:
<http://freie-software.bpb.de/Grassmuck.pdf>

Creative Commons

Creative Commons (schöpferisches Gemeingut) kurz CC ist eine gemeinnützige Organisation, die verschiedene Standard-Lizenzverträge veröffentlicht, mit denen ein Autor der Öffentlichkeit auf einfache Weise Nutzungsrechte an seinen Werken (z. B. Texte, Bilder, Musikstücke, Videoclips, ...) einräumen kann.

Es sind dabei folgende Rechtemodule vorgesehen:

- by** Der Name des Autors muss genannt werden.
- nc** Das Werk darf nicht für kommerzielle Zwecke verwendet werden.
- nd** Das Werk darf nicht verändert werden.
- sa** Weitergabe unter gleichen Bedingungen.

Zum Beispiel:

- CC-by** Namensnennung
- CC-by-sa** Namensnennung, Weitergabe unter gleichen Bedingungen
- CC-by-nd** Namensnennung, keine Bearbeitung
- CC-by-nc** Namensnennung, nicht kommerziell
- CC-by-nc-sa** Namensnennung, nicht kommerziell, Weitergabe unter gleichen Bedingungen
- CC-by-nc-nd** Namensnennung, nicht kommerziell, keine Bearbeitung

Mehr dazu unter <http://de.creativecommons.org/>.

GNU-Lizenz für freie Dokumentation

Die GFDL – GNU-Lizenz für freie Dokumentation (GNU Free Documentation License) ist eine Copyleft-Lizenz, die für freie Dokumentationen, ursprünglich für die Softwaredokumentation gedacht ist.

Die Lizenz gestattet

- die Vervielfältigung,
- Verbreitung und
- Veränderung des Werkes, auch zu kommerziellen Zwecken.

Im Gegenzug verpflichtet sich der Lizenznehmer zur Einhaltung der Lizenzbedingungen. Dies sind:

- Nennung des Autors
- abgeleitete Werke unter dieselbe Lizenz zu stellen (Copyleft-Prinzip)

⁵ Die GPL erlaubt ausdrücklich eine kommerzielle Verwertung der Programme. Natürlich unter den Regeln der GPL, sodass dem Käufer auf Anfrage der Quellcode ohne Mehrkosten (Kosten für Datenträger oder Versand ist z. B. okay.) verfügbar gemacht werden muss.

⁶ Die BSD-Lizenz (Modified-BSD-Lizenz bzw. die 3-Klausel-BSD-Lizenz, und die originale BSD-Lizenz mit Werbeklausel), ist kompatibel zur GPL. Als Folge davon darf man BSD-Code in ein GPL-Projekt einbringen und alles zusammen nur unter GPL veröffentlichen.

8.3.2 Welche Lizenz für eigene Bilder

Für die erstellte Galerie mit den Bildern würde sich eine Creative Commons Lizenz anbieten. Dabei muss man sich überlegen, ob hier eine kommerzielle Nutzung (nc) ausgeschlossen wird oder ob die Weitergabe unter gleichen Bedingungen (sa) stattfinden soll.

8.3.3 Fragen

1. Was ist der wesentliche Unterschied zwischen GPL und LPGL?

2. Was versteht man unter Copyleft?

3. Was versteht man unter FLOSS/FOSS?

4. Was bedeutet CC-by-nc?

5. Was bedeutet CC-by-nc-nd?

8.4 Firefox als sicherer Browser

Damit man sicher im Internet surfen kann, sind einige Einstellungen im Browser wichtig und zusätzliche Add-ons können einem das Leben erleichtern.

8.4.1 JavaScript

JavaScript ist eine Skriptprogrammiersprache, die im Browser integriert ist. Mit dieser Programmiersprache können Javascript-Befehle, die in einer Webseite integriert sind, lokal auf dem Rechner ausgeführt werden und so auch Aktionen ausführen, die man nicht möchte. Die Einstellung wird über *BEARBEITEN – EINSTELLUNGEN* vorgenommen

Deaktiviert man diese Einstellungen, so dass JavaScript nicht mehr funktioniert, so ist man von dieser Seite aus sicher, aber viele Webseiten, die JavaScript voraussetzen, funktionieren nicht mehr. Eine Alternative stellt die Erweiterung (Add-on) NoScript dar. Mit dieser Erweiterung kann individuell für jede Webseite festgelegt werden, ob für diese JavaScript aktiviert werden soll oder nicht.

Add-ons lassen sich sehr einfach über *EXTRAS – ADD-ONS* installieren.



Abb. 8.6: Einstellungen Javascript

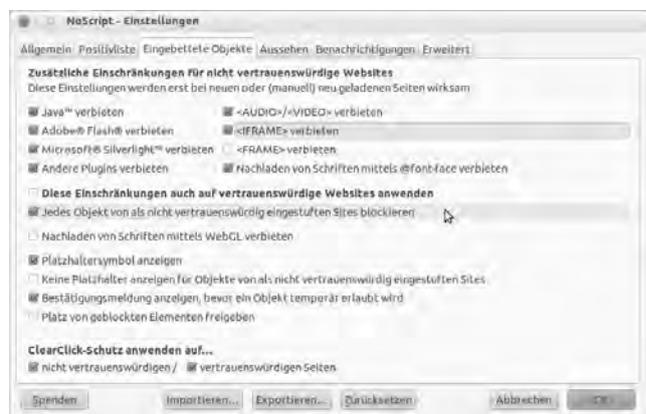


Abb. 8.7: NoScript schützt vor bössartigen Webseiten

8.4.2 Datenschutz

Firefox bietet die Möglichkeit, immer den privaten Modus zu verwenden (siehe Abbildung 8.8 auf der nächsten Seite). Dabei werden keinerlei Daten wie besuchte Seiten, Chroniken, Cookies, etc. gespeichert. Wählt man diesen Modus nicht, so kann man festlegen, wie und vor allem wann Firefox diese Daten wieder löschen soll. Eine gute Wahl ist z. B., diese Daten zu löschen, wenn Firefox geschlossen wird.

Folgende Add-ons helfen hier, diese löschen auf einfache Weise die History, Cookies, etc. und sogar Flash Local Shared Objects (LSO)

- Click&Clean
- BetterPrivacy

8.4.3 Werbung unterdrücken

Bei vielen Seiten nervt, dass ständig Werbung eingeblendet wird. Diese stört nicht nur, sondern man findet so auch oft nicht gleich das was man sucht und es werden zusätzliche Daten übertragen. Das Add-on Adblock Plus hilft hier und verhindert so unerwünschte Werbung. Über eine Auswahl können verschiedene Filter aktiviert werden, die jeweils Werbung, z. B. für Deutschland ausblenden (siehe Abbildung 8.9 auf der nächsten Seite).



Abb. 8.8: Datenschutzeinstellungen

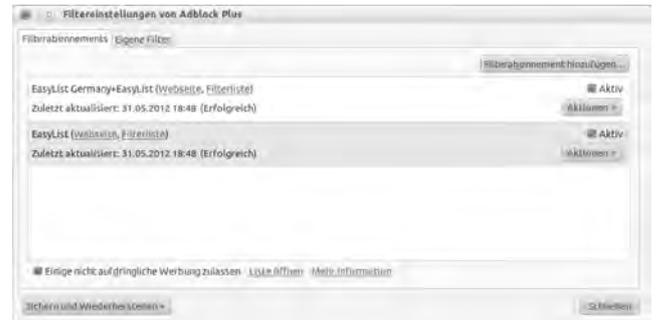


Abb. 8.9: Werbung verhindern

8.4.4 Webseitenverfolgung verhindern

Es gibt einige Möglichkeiten, wie versucht wird, herauszufinden, wer auf welche Seite surft. Eine Möglichkeit ist z. B. ganz kleine Bilder (diese haben eine Größe von einem Pixel) von einem bestimmten Server nachzuladen. Dieser kann dann genau in seiner LOG-Datei feststellen, wer (hier wird die IP-Adresse gespeichert) wann welche Seite besucht hat. Verhindern kann dies z. B. das Add-on Ghostery. Es zeigt einem an, welche Web-Elemente hier in einer Seite eingebaut sind und kann diese blockieren.

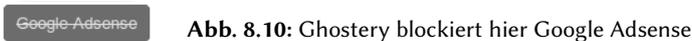


Abb. 8.10: Ghostery blockiert hier Google AdSense

Zusätzlich zeigt Ghostery an, welche Funktion das Web-Element hat. Bei Google AdSense z. B. „Google AdSense is an ad serving application run by Google. Website owners can enroll in this program to enable text, image, and video advertisements on their websites. These advertisements are administered by Google and generate revenue on either a per-click or per-impression basis. Primary products include AdSense for content (text or display) and AdSense for search.“

8.4.5 Verschlüsselte Verbindung verwenden

Viele Webseitenbetreiber bieten neben einer http-Verbindung auch die verschlüsselte Variante über https an. Somit ist die Verbindung zu dem Server verschlüsselt und kann nicht abgehört werden. Das Add-on HTTPS Finder ersetzt die http-Verbindung, wenn es dazu auch eine https-Verbindung gibt.

8.4.6 Speichern von Inhalten

Bietet eine Webseite eine Datei zum Download an, so kann diese meist über einen Klick oder durch „Ziel speichern unter“ gespeichert werden. Sind viele Dateien auf der Webseite, so kann dies schnell in Arbeit ausarten. Das Add-on DownThemAll (DTA) untersucht die Webseite und bietet dann nach bestimmten Filtern an, die Dateien alle am Stück herunterzuladen (siehe Abbildung 8.11 auf der nächsten Seite).



Abb. 8.11: Filter von DownThemAll

8.4.7 Websuche

Für die Websuche verwenden die meisten Leute eine Suchmaschine wie „Google“, „Yahoo“ oder „Bing“. Mit dem Add-on FoxyWebSearch werden mehrere Suchmaschinen befragt und das Ergebnis wird angezeigt. Eine kleine Übersicht am rechten Rand zeigt an, wie viele Treffer die jeweilige Suchmaschine gefunden hat.

Übung:

1. Probieren Sie alle vorgestellten Add-ons aus.
2. Suchen Sie unter <https://addons.mozilla.org/de/firefox/extensions/privacy-security/> weitere sicherheitsrelevante Add-ons.
3. Durchsuchen Sie die Kategorien von Add-ons. Wenn Sie ein interessantes Add-on finden, probieren Sie es aus.



9 Einheit VIII – OSS Gimp & LibreOffice

In diesem Abschnitt geht es um die Anwendung von Open Source Software.

Dieser Bereich ist optional, wenn noch Zeit ist oder für zu Hause.

Betrachtet man einige Bilder, so fällt auf, dass hier ein „Fleck“ auf dem Bild ist, ein Gegenstand im Bild stört oder eine Person rote Augen hat. Mit Gimp lassen sich solche Dinge nach etwas Übung schnell bereinigen und man hat deutlich bessere Bilder.

Im vorherigen Abschnitt wurde gezeigt, wie man die Klassenfotos in einer HTML-Gallery mit Vorschaubildern erzeugt. Eine andere Möglichkeit, die Bilder zu präsentieren ist eine Präsentation mit LibreOffice.

Inhalt

9.1	GIMP	160
9.2	LibreOffice	161

9.1 GIMP

Gimp (GNU Image Manipulation Program) ist ein kostenloses und freies Bildbearbeitungsprogramm, welches unter der GPL steht.

GIMP wird meist dazu genutzt, einzelne Bilder zu bearbeiten, viele Effekte auszunützen und Filter anzuwenden.

Installation

GIMP ist im normalen Ubuntu Repository enthalten und kann wie folgt installiert werden.

```
sudo apt-get install gimp
```

Leider ist bei Ubuntu 12.04 nur die Version 2.6 enthalten. Aktuell ist die Version 2.8, die vor allem den Einzelfenstermodus unterstützt. Diese lässt sich aber recht einfach nachinstallieren.

```
1 sudo add-apt-repository ppa:otto-kesselgulasch/gimp
2 sudo apt-get update
3 sudo apt-get install gimp
```

In Zeile 1 wird ein neues Repository hinzugefügt. Ein PPA (Personal Package Archive) bietet die Möglichkeit, Debian-Pakete (Ubuntu ist aus Debian entstanden) für unterschiedliche Architekturen (i386, amd64. ...) zu bauen und diese als eigene Paketquelle anderen zur Verfügung zu stellen.

Ist das Repository hinzugefügt, muss noch die Paktedatenbank aktualisiert werden (Zeile 2). Danach kann GIMP installiert werden (Zeile 3).



Abb. 9.1: GIMP im Einfenstermodus

GIMP hier zu beschreiben, würde den Rahmen sprengen. Es gibt aber viele Tutorials, die zeigen, wie GIMP funktioniert, z. B.

- <http://www.gimpusers.de/tutorials>
- <http://www.gimp-werkstatt.de/tutorien-grundlagen.php>

Übung:

Probieren Sie GIMP mit Ihren Bilder aus, lesen Sie die entsprechenden Tutorials und verbessern sie evtl. nicht so gute Bilder.



9.2 LibreOffice

LibreOffice ist ein freies Office-Paket für Textverarbeitung, Tabellenkalkulation, Präsentation und zum Erstellen von Zeichnungen. Zusätzlich stehen ein Datenbankmanagementsystem und ein Formeleditor zur Verfügung.

Für Präsentationen steht das Programm LibreOffice-Impress zur Verfügung.

Auch LibreOffice-Impress wird hier nicht beschrieben. Die folgenden Tutorials zeigen, wie damit eine Präsentation erstellt wird.

- <http://www.libreoffice.org/get-help/documentation/>
- http://help.libreoffice.org/Impress/Instructions_for_Using_Impress/de
- <http://gofree.com/Tutorials/LibreOfficeImpress1.php>
- <http://www.computerbild.de/artikel/cb-Downloads-Office-Buero-LibreOffice-Tipps-Anleitung-5951811.html>

Übung:

Probieren Sie LibreOffice-Impress aus und erstellen Sie eine „Dia-Show“ mit Ihren Bildern.



10 Themenübersicht: Linux-Essentials

Einführung

Die Beschreibung des gesamten LinuxEssentials Programms ist unter <http://wiki.lpi.org/wiki/LinuxEssentials> zusammengestellt (in Version 1.0).¹

Das Ziel des Linux Essentials Programms ist es, das Grundwissen zu definieren, das für die kompetente Nutzung eines Linux Betriebssystems auf einem Desktop-Computersystem oder auf einem mobilen Gerät notwendig ist. Dieses Programm wird Jugendlichen und denjenigen, für die Linux und Open Source neu ist, helfen und sie dabei unterstützen, den Platz von Linux und Open Source im größeren Kontext der IT-Branche zu verstehen.

Beschreibung des geeigneten Kandidaten für diese Prüfung (MQC)

Dies ist eine Beschreibung eines Kandidaten, der gerade soweit qualifiziert ist, um die LPI Linux Essentials Prüfung noch zu bestehen. Diese fiktive Person wird Minimally Qualified Candidate (MQC) genannt. Die Standards für die LPI Linux Essentials Prüfung sollten so gesetzt werden, dass diese Person, und jeder der kompetenter ist, die Prüfung besteht, aber jeder der weniger kompetent ist, durchfällt.

Der erfolgreiche Kandidat sollte die Linux- und Open-Source-Industrie verstehen und die gängigsten Open-Source-Anwendungen kennen. Der Kandidat sollte die Hauptkomponenten des Linux-Betriebssystems verstehen und das technische Wissen zum Arbeiten auf der Kommandozeile besitzen. Der MQC hat Grundkenntnisse zu Themen mit Bezug auf Sicherheit und Administration, wie dem Verwalten von Benutzern und Gruppen, dem Arbeiten auf der Kommandozeile und zum Thema Berechtigungen auf einem System. Der Inhaber eines Linux-Essentials-Zertifikats ist höchstwahrscheinlich Endnutzer eines weitgehend verwalteten Systems.

Ein geeigneter Kandidat (MQC) für die Linux-Essentials-Prüfung:

- Hat ein grundlegendes Verständnis von Freier und Open Source Software (FOSS), von den verschiedenen Communities und den Softwarelizenzen.
- Versteht die Grundlagen von Prozessen, Programmen und die Bestandteile eines Betriebssystems.
- Hat ein Grundwissen über Computer Hardware.
- Hat ein grundlegendes Verständnis von Systemsicherheit, von Benutzern und Gruppen und von Dateizugriffsrechten für öffentliche und private Verzeichnisse.
- Kann den eigenen Computer einem lokalen Netzwerk (LAN) hinzufügen und auf andere Computer zugreifen.
- Kennt die gängigen Open Source Anwendungen und weiß, welchen Closed Source Äquivalenten sie zugeordnet werden können.
- Versteht Navigationssysteme auf Linux-Desktopsystemen und weiß, wo Hilfe zu finden ist.
- Hat grundlegende Fähigkeiten zum Arbeiten auf der Kommandozeile und mit Dateien.
- Kann einfache Backups und Archive erstellen.
- Kann einen einfachen Kommandozeilen-Editor benutzen.
- Hat ein grundlegendes Verständnis von Dateiberechtigungen.
- Kann einfache Skripte schreiben und auszuführen.

¹ Quelle: <http://wiki.lpi.org/wiki/LinuxEssentials> (DE)

Übersicht

Thema	Beschreibung	Gewichtung
1	Die Linux-Community und Karriere im Open-Source-Umfeld	7
1.1	Linux Entwicklung und gängige Betriebssysteme	2
1.2	Wichtige Open Source Anwendungen	2
1.3	Open-Source-Software und Lizenzen verstehen	1
1.4	IuK-Kenntnisse und Arbeiten mit Linux	2
2	Sich auf einem Linux-System zurechtfinden	8
2.1	Erste Schritte auf der Kommandozeile	2
2.2	Die Kommandozeile benutzen, um Hilfe zu finden	2
2.3	Verzeichnisse und Protokolldateien nutzen	2
2.4	Erstellen, Verschieben und Löschen von Dateien	2
3	Die Macht der Kommandozeile	10
3.1	Archivierung von Dateien auf der Kommandozeile	2
3.2	Suche und Entnahme von Daten aus Dateien	4
3.3	Befehle in ein Skript umwandeln	4
4	Das Linux Betriebssystem	8
4.1	Wahl eines Betriebssystems	1
4.2	Computer-Hardware verstehen	2
4.3	Datenspeicherung	3
4.4	Den Computer im Netzwerk einbinden	2
5	Sicherheit und Dateiberechtigungen	7
5.1	Grundlegende Sicherheit und Identifizieren von Benutzertypen	2
5.2	Benutzer und Gruppen erstellen	2
5.3	Einstellungen für Dateiberechtigungen und Dateieigentum	2
5.4	Besondere Verzeichnisse und Dateien	1
Summe		40

1 Die Linux-Community und Karriere im Open-Source-Umfeld

Gewichtung: 7

1.1 Linux Entwicklung und gängige Betriebssysteme

Gewichtung	2
Beschreibung	Kenntnisse über Linux Entwicklung und wichtige Distributionen.

Hauptwissensgebiete:

- Open Source Philosophie
- Distributionen
- Embedded Systems

Hier ist eine auszugsweise Liste der verwendeten Dateien, Begriffe und Hilfsprogramme:

- Android
- Debian
- CentOS

1.2 Wichtige Open Source Anwendungen

Gewichtung	2
Beschreibung	Wichtige Anwendungen und deren Nutzung.

Hauptwissensgebiete:

- Desktop Anwendungen
- Server Anwendungen
- Mobile Anwendungen
- Entwicklungssprachen
- Werkzeuge für die Paketverwaltung

Hier ist eine auszugsweise Liste der verwendeten Dateien, Begriffe und Hilfsprogramme:

- OpenOffice.org, LibreOffice, Thunderbird, Firefox
- Blender, Gimp, Audacity, ImageMagick
- Apache, MySQL, PostgreSQL
- NFS, Samba, OpenLDAP, Postfix, DNS, DHCP
- C, Perl, Shell, Python, PHP

1.3 Open-Source-Software und Lizenzen verstehen

Gewichtung	1
Beschreibung	Open Source Communities und Lizenzen von freier und Open Source Software.

Hauptwissensgebiete:

- Lizenzen
- Free Software Foundation (FSF), Open Source Initiative (OSI)

Hier ist eine auszugsweise Liste der verwendeten Dateien, Begriffe und Hilfsprogramme:

- GPL, BSD, Creative Commons
- Free Software, Open Software, FOSS, FLOSS
- Open-Source-Geschäftsmodelle

Gut zu wissen:

- Geistiges Eigentum: Copyright, Markenzeichen und Patente
- Apache Lizenz, Mozilla Lizenz

1.4 IuK-Kenntnisse und Arbeiten mit Linux

Gewichtung	2
Beschreibung	Grundkenntnisse der Informations- und Kommunikationstechnologie (IuK) und Arbeiten mit Linux.

Hauptwissensgebiete:

- Desktop Kenntnisse
- Erste Schritte auf der Kommandozeile
- Gewerbliche Nutzung von Linux, Cloud Computing und Virtualisierung

Hier ist eine auszugsweise Liste der verwendeten Dateien, Begriffe und Hilfsprogramme:

- Nutzung eines Browsers, Sicherheitsbedenken, Einstellungsoptionen, Websuche, und Speichern von Inhalten
- Terminal und Konsole
- Passworteinstellungen
- Privatsphäreinstellungen und Werkzeuge
- Nutzung gängiger Open Source Anwendungen in Präsentationen und Projekten

2 Sich auf einem Linux-System zurechtfinden

Gewichtung: 8

2.1 Erste Schritte auf der Kommandozeile

Gewichtung	2
Beschreibung	Grundkenntnisse zur Benutzung der Linux Kommandozeile

Hauptwissensgebiete:

- Shell Grundkenntnisse
- Kommandos eingeben und strukturieren
- Mit Optionen arbeiten
- Variable
- Globbing, Suchmuster
- Quoting

Hier ist eine auszugsweise Liste der verwendeten Dateien, Begriffe und Hilfsprogramme:

- echo
- history
- PATH Umgebungsvariable
- which

Gut zu wissen:

- Variablenersetzung
- Steuerungsoperatoren |, && and ;

2.2 Die Kommandozeile benutzen, um Hilfe zu finden

Gewichtung	2
Beschreibung	Hilfekommandos benutzen und die Navigation durch die unterschiedlichen Hilfesysteme.

Hauptwissensgebiete:

- Man
- Info

Hier ist eine auszugsweise Liste der verwendeten Dateien, Begriffe und Hilfsprogramme:

- man
- info
- Handbuchseiten (man pages)
- /usr/share/doc
- locate

Gut zu wissen:

- apropos, whatis, whereis

2.3 Verzeichnisse und Protokolldateien nutzen

Gewichtung	2
Beschreibung	Navigieren in Home- und System-Verzeichnissen und Protokolldateien an verschiedenen Orten.

Hauptwissensgebiete:

- Dateien, Verzeichnisse
- Versteckte Dateien und Verzeichnisse
- Home
- Absolute und relative Pfade

Hier ist eine auszugsweise Liste der verwendeten Dateien, Begriffe und Hilfsprogramme:

- Gängige Optionen für ls
- Rekursives Auflisten
- cd
- . and ..
- Home and ~

2.4 Erstellen, Verschieben und Löschen von Dateien

Gewichtung	2
Beschreibung	Erstellen, Verschieben und Löschen von Dateien und Verzeichnissen im Home-Verzeichnis.

Hauptwissensgebiete:

- Dateien und Verzeichnisse
- Beachtung von Groß- und Kleinschreibung
- Einfaches Globbing und Quoting

Hier ist eine auszugsweise Liste der verwendeten Dateien, Begriffe und Hilfsprogramme:

- mv, cp, rm, touch
- mkdir, rmdir

3 Die Macht der Kommandozeile

Gewichtung: 10

3.1 Archivierung von Dateien auf der Kommandozeile

Gewichtung	2
Beschreibung	Archivierung von Dateien in deren eigenen Verzeichnissen (Home).

Hauptwissensgebiete:

- Dateien, Verzeichnisse
- Archive, Komprimieren

Hier ist eine auszugsweise Liste der verwendeten Dateien, Begriffe und Hilfsprogramme:

- tar
- Gängige tar-Optionen
- gzip, bzip2
- zip, unzip

Gut zu wissen:

- Einzelne Dateien aus Archiven extrahieren

3.2 Suche und Entnahme von Daten aus Dateien

Gewichtung	4
Beschreibung	Suche und Entnahme von Daten aus Dateien in den Home-Verzeichnissen.

Hauptwissensgebiete:

- Kommando-Pipelines
- Eingabe-/Ausgabe-Umleitung
- Wichtigste POSIX-konforme reguläre Ausdrücke (., [], *, ?)

Hier ist eine auszugsweise Liste der verwendeten Dateien, Begriffe und Hilfsprogramme:

- find
- grep
- less
- head, tail
- sort
- cut
- wc

Gut zu wissen:

- Grundlegende reguläre Ausdrücke gemäß POSIX (Auszug – [^], ^, \$)
- Erweiterte reguläre Ausdrücke gemäß POSIX (- +, (), |)
- xargs

3.3 Befehle in ein Skript umwandeln

Gewichtung	4
Beschreibung	Wiederkehrende Befehle in einfache Skripte umwandeln.

Hauptwissensgebiete:

- Grundlagen der Textbearbeitung
- Grundlagen von Shellskripten

Hier ist eine auszugsweise Liste der verwendeten Dateien, Begriffe und Hilfsprogramme:

- `/bin/sh`
- Variable
- Argumente
- `for`-Schleifen
- `echo`
- Rückgabewert

Gut zu wissen:

- `pico`, `nano`, `vi` (nur Grundlagen zur Erstellung von Skripten)
- Bash
- `if`-, `while`- und `case`-Kommandos
- Kommandos `read`, `test` und `[`

4 Das Linux Betriebssystem

Gewichtung: 8

4.1 Wahl eines Betriebssystems

Gewichtung	1
Beschreibung	Kenntnisse über wichtige Betriebssysteme und Linux Distributionen.

Hauptwissensgebiete:

- Unterschiede zwischen Windows, Mac und Linux
- Management des Lebenszyklus von Distributionen

Hier ist eine auszugsweise Liste der verwendeten Dateien, Begriffe und Hilfsprogramme:

- GUI versus Kommandozeile, Desktop Konfiguration
- Wartungszyklen, Beta und Stabil

4.2 Computer-Hardware verstehen

Gewichtung	2
Beschreibung	Vertrautheit mit den Komponenten für den Bau von Desktop- und Server-Computern.

Hauptwissensgebiete:

- Hardware

Hier ist eine auszugsweise Liste der verwendeten Dateien, Begriffe und Hilfsprogramme:

- Festplatten und Partitionen, Hauptplatinen, Prozessoren, Netzteile, optische Laufwerke, Peripheriegeräte
- Bildschirmarten
- Treiber

4.3 Datenspeicherung

Gewichtung	3
Beschreibung	Wo verschiedene Arten von Informationen in einem Linuxsystem gespeichert werden.

Hauptwissensgebiete:

- Kernel
- Prozess
- syslog, klog, dmesg
- /lib, /usr/lib, /etc, /var/log

Hier ist eine auszugsweise Liste der verwendeten Dateien, Begriffe und Hilfsprogramme:

- Programme, Bibliotheken, Pakete und Paketdatenbank, Systemeinstellungen
- Prozesse und Prozesstabellen, Speicheradressen, Systembenachrichtigungen und Protokollierung
- ps, top, free

4.4 Den Computer im Netzwerk einbinden

Gewichtung	2
Beschreibung	Abfragen wesentlicher Netzwerkeinstellungen und Bestimmen der Grundvoraussetzungen für einen Computer in einem Local Area Network (LAN).

Hauptwissensgebiete:

- Internet, Netzwerk, Router
- Domain Name Service
- Netzwerkeinstellungen

Hier ist eine auszugsweise Liste der verwendeten Dateien, Begriffe und Hilfsprogramme:

- route
- resolv.conf
- IPv4, IPv6
- ifconfig
- netstat
- ping

Gut zu wissen:

- ssh
- dig

5 Sicherheit und Dateiberechtigungen

Gewichtung: 7

5.1 Grundlegende Sicherheit und Identifizieren von Benutzertypen

Gewichtung	2
Beschreibung	Verschiedene Benutzertypen in einem Linux System.

Hauptwissensgebiete:

- Root und unprivilegierte Nutzer
- Systemnutzer

Hier ist eine auszugsweise Liste der verwendeten Dateien, Begriffe und Hilfsprogramme:

- /etc/passwd, /etc/group
- id, who, w
- sudo

Gut zu wissen:

- su

5.2 Benutzer und Gruppen erstellen

Gewichtung	2
Beschreibung	Benutzer und Gruppen in einem Linuxsystem erstellen.

Hauptwissensgebiete:

- Befehle zu Benutzern und Gruppen
- User-IDs

Hier ist eine auszugsweise Liste der verwendeten Dateien, Begriffe und Hilfsprogramme:

- /etc/passwd, /etc/shadow, /etc/group
- id, last
- useradd, groupadd
- passwd

Gut zu wissen:

- usermod, userdel
- groupmod, groupdel

5.3 Einstellungen für Dateiberechtigungen und Dateieigentum

Gewichtung	2
Beschreibung	Verstehen und Bearbeiten von Dateiberechtigungen und Eigentumseinstellungen.

Hauptwissensgebiete:

- Datei-/Verzeichnis-Rechte und Dateieigentümer

Hier ist eine auszugsweise Liste der verwendeten Dateien, Begriffe und Hilfsprogramme:

- ls -l
- chmod, chown

Gut zu wissen:

- chgrp

5.4 Besondere Verzeichnisse und Dateien

Gewichtung	1
Beschreibung	Besondere Verzeichnisse und Dateien in einem Linuxsystem, einschließlich besonderer Zugriffsrechte.

Hauptwissensgebiete:

- Systemdateien, Bibliotheken
- Symbolische Links

Hier ist eine auszugsweise Liste der verwendeten Dateien, Begriffe und Hilfsprogramme:

- /etc, /var
- /tmp, /var/tmp and Sticky Bits
- `ls -d`
- `ln -s`

Gut zu wissen:

- Hardlinks
- Setuid/Setgid

11 Anhang

weiterführende Literatur

- Linux Essentials – Die Einsteiger-Zertifizierung des LPI der Firma Linup-Front
Schulungsunterlage (Deutsch) unter Creative Commons
<http://shop-download.linupfront.de/cc/lxes-de-manual-cc.pdf>
Schulungsunterlage (Englisch) unter Creative Commons
<http://shop-download.linupfront.de/cc/lxes-en-manual-cc.pdf>
- Linux Essentials (Englisch)
Roderick W. Smith
ISBN: 978-1-1181-0679-2
Paperback, 368 pages, March 2012, 32,00 Euro
<http://eu.wiley.com/WileyCDA/WileyTitle/productCd-1118106792,miniSiteCd-SYBEX.html>
- Linux – Das umfassende Handbuch
Johannes Plötner, Steffen Wendzel
Galileo Computing
1282 Seiten, 5., aktualisierte Auflage 2012, geb., mit 2 DVDs
49,90 Euro, ISBN 978-3-8362-1822-1
Openbook (frei)
<http://openbook.galileocomputing.de/linux/>
- Ubuntu GNU/Linux (Version 11.04)
Marcus Fischer
Galileo Computing
1118 Seiten, 39,90 Euro ISBN 978-3-8362-1765-1
Openbook (frei)
<http://openbook.galileocomputing.de/ubuntu/>
- Linux-UNIX-Programmierung
Jürgen Wolf
Galileo Computing
1216 Seiten, mit CD, 49,90 Euro ISBN 3-89842-749-8 Openbook (frei)
http://openbook.galileocomputing.de/linux_unix_programmierung/
- Shell-Programmierung
Jürgen Wolf
Galileo Computing
782 Seiten., mit CD, 44,90 Euro
ISBN 3-89842-683-1
Openbook (frei)
http://openbook.galileocomputing.de/shell_programmierung/
- Wie werde ich UNIX-Guru?
Arnold Willemer
Galileo Computing
672 Seiten, 2003, geb.
34,90 Euro, ISBN 978-3-89842-240-6 (Vergriffen) Openbook (frei)
http://openbook.galileocomputing.de/unix_guru/
- IT-Handbuch für Fachinformatiker
Sascha Kersken
Galileo Computing
ca. 1172 Seiten, 5., aktualisierte und erweiterte Auflage, geb.
ca. 34,90 Euro, ISBN 978-3-8362-1744-6
Openbook (frei)
http://openbook.galileocomputing.de/it_handbuch/

Programme / Befehle

Symbole

7z 135

A

anacron 139

apropos 22

apt-get 19

ash 20

B

basename 143

bash 19, 20

bg 24

bunzip2 131

bzcat 131

bzip2 131, 134

C

cal 23

cat 28, 101

cd 28, 30, 33

chgrp 61

chmod 60, 61

chown 61

chpasswd 62

cmd.exe 20

command.com 20

convert 109

cp 28, 36, 92, 135

csch 20

cut 104, 107

D

dash 19, 20

date 23

dhclient 123

dhcpcd 123

dig 122

dmesg 88, 89

dosfstlabel 98

E

echo 21, 28

env 68

exif 99, 101, 107

exifautotran 109

F

fdisk 97

fg 24

file 28

find 28, 29, 92, 95

free 26

fsck 97

fsck.vfat 97

G

gedit 24

grep 27, 28, 79, 103, 107

groupadd 52

groupdel 53

groupmod 53

gunzip 130, 131

gzip 130, 131, 134

H

head 80, 101

help 20

history 20

I

id 49

ifconfig 121

J

jobs 24

K

kill 26

killall 26

L

less 22, 28

ln 38

locate 95

ls 23, 27, 28

lshw 91

lshw-gtk 91

lsusb 91

M

man 20

md5 81

md5pass 81

mkdir 28, 30

mkfs 98

mkfs.ext3 98

mkfs.ext4 98

mkfs.ntfs 98

mkfs.vfat 98

mount 90, 96

mv 28, 37

N

nano 40, 43

newgrp 48

nl 101

nslookup 122

P

passwd 51, 59, 82

python 27

ping 116, 122, 124

ping6 124

printf 108

ps 25, 26

pstree 25

pwd 28, 33

R	
rar	135
read	78
rm	28, 37, 93
rmdir	28
route	122
S	
scp	126
sed	27
sh	20
sleep	139
slocate	95
sort	28, 82, 102 , 104
ssh	126
su	51
sudo	19, 51 , 83
synaptic	19
syslogd	88
T	
tail	90, 101 , 102
tar	134
tee	72
test	28, 75
time	131
top	26
touch	28, 30
tr	79, 80, 104
U	
ufraw	109
ufraw-batch	109
umount	97, 98
unalias	21
uniq	102
unrar	135
unset	68
unzip	132
updatedb	95
useradd	49 , 53, 81
usermod	53
V	
vi	40, 41
vim	41
W	
w	55
wall	59
wc	104
whatis	22
whereis	95
which	95
who	23, 55
wkhtmltopdf	142
X	
xargs	93, 168
Y	
yast	120, 123
Z	
zcat	130
zip	132
zsh	20

Verzeichnisse / Dateien

Symbole

.bash_history (~/) 20
.bash_logout (~/) 21

A

~/Arbeitsfläche/ 43
auth.log (/var/log/) 88

C

crontab (/etc/) 139

D

dmesg (/var/log/) 88

E

/etc/rsyslog.d/ 88
/etc/skel/ 49
/etc/sysconfig/network/ 123
/etc/sysconfig/networkscripts/ 123

G

group (/etc/) 48, 52, 82

H

hosts (/etc/) 123

I

interfaces (/etc/network/) 123, 124

K

kern.log (/var/log/) 88

M

/mnt/ 96

P

passwd (/etc/) 48, 50, 52, 58, 59, 82, 104

R

resolv.conf (/etc/) 122
rsyslog.conf (/etc/) 88

S

shadow (/etc/) 48, 50, 51, 58
sudoers (/etc/) 51, 83
syslog (/var/log/) 88

U

udev (/var/log/) 88
urandom (/dev/) 80
/usr/share/doc/ 23
/usr/share/doc/howto/ 23

V

/var/log/ 88

Stichwortverzeichnis

Symbole

~	116
;	24
<	71
>	71
>>	71
&	24
&&	75
~	27
	72
	75
2>&1	71
7-Zip	135
7z	135

A

Add-on	155
Address Resolution Protocol	116
Alias	21
Android	15
Anycast	119
Apache	16
ARP	116
ASCII-NUL	93
Audacity	16
Ausgabeumlenkung	71

B

bash	
Alias	21
History	20
letzte Kommandos	20
Logout	21
Bedingungen	77
Befehl	23
Benutzerkonto	48
Benutzerverwaltung	48
Berkeley Software Distribution	<i>siehe</i> BSD
Bibliothek	152
Bilder	
kopieren	107
Blender	16
Brute-Force	50
BSD	152
Buggix	12
bzip2	131

C

C	17
C++	17
case-insensitive	27
case-preserving	27
case-sensitive	27
CC	153
CC-by	153
CC-by-nc	153
CC-by-nc-nd	153
CC-by-nc-sa	153
CC-by-nd	153

CC-by-sa	153
CIDR	121
Closed Source	152
Cloud	14
Cluster	14
Copyleft	12, 151
Creative Commons	153, 154
cron	139

D

D-BUS	88
Datei	31, 36
alle Anwenderdateien löschen	93
auf Größe 0 setzen	21
Benutzerklassen	58
Berechtigung	58
Eigentum	58
Eigentümer ändern	59
finden	92
Gerätedatei	31
group	58
Inhalt löschen	21
kopieren	92
löschen	28
Namenskonventionen	27
others	58
owner	58
SGID	59
Sticky-Bit	59
SUID	59
versteckt	27, 29
Verwaltung	27
Zugriffsrechte	58
Dateiberechtigung	163
Dateisystem	96
aushängen	97
Deviceübersicht	97
einbinden	96
formatieren	98
Journal	96
Namen festlegen	98
prüfen	97
Standardeinhängepunkt	97
verwendete anzeigen	96
Übersicht	96
Datenbankmanagementsystem	161
Datensicherung	137
DCIM	92
DEC-PDP/7	12
Default-Gateway	122
Department of Defense	116
Desktop	43
Start-Icon erstellen	43
DHCP	123, 124
directory	31
DNS	120
Namen	120
Server	120

DOCTYPE	142	User	52
DoD	116	Verwaltung	52
Domain		Zuordnung der Nummern	52
root	120	ändern	52
Second-Level	120	gzip	130
Subdomain	120		
Top-Level	120		
Domain Name System	120		
Donnervogel	16		
DSL-Router	121		
Dynamic Host Configuration Protocol	123		
E		H	
Eingabeumlenkung	71	HAL	88
Einweg-Verschlüsselungsverfahren	50	Hardware	121
else	77	Hardware Abstraction Layer	<i>siehe</i> HAL
Emacs	40	hash-bang	68
Embedded Device	15	Hash-Verfahren	50
ERRORLEVEL	75	Hier-Dokument	72
Exif	99	History	20
		Home-Verzeichnis	48
		Host	117
F		HTML	142
Feuerfuchs	16	HTTP	117
FHS	32	Server	125
fi	77	Testen des Servers	125
File Transfer Protocol	117	https	117
Filesystems Hierarchy Standard	<i>siehe</i> FHS	Hypertext Transfer Protocol	117
Firefox	16		
FLOSS	152		
Formeleditor	161		
forward slash	27		
FOSS	152, 163		
FQDN	120		
Freax	12		
Free and Open Source Software	152		
Free/Libre Open Source Software	152		
FSF	12, 151		
FTP	117		
Fully Qualified Domain Name	120		
G		I	
GECOS	50	ICMP	116
Generationenprinzip	137	Identifikationsnummer	49
Geschäftsmodelle	152	if	77
GFDL	153	IFS	76
GID	49	ImageMagick	16, 109
Gimp	16, 160	Informations- und Kommunikationstechnologie	166
globbing	27	Inode	38
Gnome	14	Internet	116
Gnome-Control-Center	123	Definition	116
GNU	12	Ursprung	116
GNU General Public License	<i>siehe</i> GPL	Internet Control Message Protocol	116
GNU GPL	<i>siehe</i> GPL	Internet Protocol	116
GNU Lesser General Public License	<i>siehe</i> LGPL	Interprozesskommunikation	88
GNU LGPL	<i>siehe</i> LGPL	IP	116
GPL	12, 151	IPC	88
Gruppen		IPsec	119
anlegen	48	IPv4	117
Datenbank	52	Adressklassen	118
groupadd	52	Broadcast-Adresse	118
groupdel	52	Darstellung	117
hinzufügen	52	Gepunktete Dezimaldarstellung	117
löschen	52	Host-ID	117
primär	48	Klasse	
root	52	A	118
sekundär	48	B	118
Strategien	52	C	118
Systemdienst	52	D	118
		E	118
		Loopback-Adresse	118
		Multicast	118
		Netz-ID	117
		Netzadresse	118
		Private Adressen	118
		IPv6	
		Adressarten	119
		Adressformat	119
		Adresstypen	119
		Darstellung	119
		Vorteile	119
		IuK	<i>siehe</i> Informations- und Kommunikationstechnologie
		J	
		Java	17
		JavaScript	155

- Jokerzeichen 27
- JPEG 99
- K**
- Kamera
 - Metadaten anzeigen 99
- KDE 14
- Kommando 23
 - gruppieren 81
 - im Hintergrund ausführen 24
- Kommandointerpreter 20
- Komprimierung 130
 - als Stream 130
 - im Kontainer 130
 - verlustbehaftet 130
 - verlustfrei 130
- Konsole 17, 18
- L**
- LAN 163
- Laufwerksbuchstaben 32
- LDAP 17
- LGPL 152
- LibreOffice 16, 161
- Lighttpd 125
- Linux
 - Eigenschaften 48
 - Embedded 15
 - Geschichte 12
 - Mindmap 13
- Linux Essential 163
 - Ziele 163
- Lizenz
 - BSD *siehe* BSD
 - Copyleft *siehe* Copyleft
 - Creative Commons *siehe* Creative Commons
 - FLOSS *siehe* FLOSS
 - FOSS *siehe* FOSS
 - GPL *siehe* GPL
 - LGPL *siehe* LGPL
 - Public Domain *siehe* Public Domain
 - Übersicht Softwarelizenzen 153
- LOG 88
 - BOOT-Meldungen 88
 - Kernmeldungen 88
 - Konfiguration 88
 - kontinuierlich anzeigen 90
 - Meldung 88
 - syslog 88
 - Systemmeldungen 88
 - USB-Stick 90
- Long Term Support *siehe* LTS
- Loopback-Adresse 118
- LSO 155
- LTS 13
- LXDE 14
- M**
- MAC 116, 121
- Magic Line 68
- Man-Pages 22
- Medien
 - automatisch einbinden 88
 - extern 88
- Minimally Qualified Candidate *siehe* MQC
- Mozilla
 - Firefox 16
 - Thunderbird 16
- MQC 163
- Multicast 118, 119
- Multimedia 119
- MySQL 16
- N**
- Namensauflösung 122
- Nano 40
- Network File System 117
- NetworkManager 120
- Netzadresse 118
- Netzwerkkonfiguration
 - automatisch 123
 - dauerhaft speichern 123
- NFS 17, 117
- NoScript 155
- O**
- Oktette 117
- Open Source 151
 - Geschäftsmodelle 152
 - Philosophie 151
- OpenLDAP 17
- OpenOffice.org 16
- ordinary file 31
- P**
- Pakete
 - installieren 19
- Pangramm 106
- Passwort
 - Brute-Force 50
 - Datenbank 50
 - Regeln 50
 - shadow 50
 - ändern 51
- PCRE 27
- Perl 17
- Pfadangabe 32
- PHP 17
- Pico 40
- PID 25
- Pipe 72, 93
- Piping 72
- PolicyKit 88
- Port
 - 20 117
 - 21 117
 - 22 117
 - 25 117
 - 80 117
 - 443 117
- Postfix 17
- PostgreSQL 16
- PPA 160
- Protokoll 116
- Präsentation 161
- Pseudobeneutzer 48
- Public Domain 152
- Python 17
- Q**
- QoS 119
- Quality of Service 119
- Quantoren 27
- Quoting 29, 69
- R**
- rar 135

- Rechte
 - Administrationsrechte 48
 - ausführen 58
 - eingeschränkt 48
 - lesen 58
 - r 58
 - schreiben 58
 - SGID 58
 - Sticky-Bit 58
 - SUID 58
 - w 58
 - x 58
- Referenzzähler 38
- regex *siehe* Regulärer Ausdruck
- regular expression *siehe* Regulärer Ausdruck
- Regulärer Ausdruck 27, 103
- Repository 160
 - hinzufügen 160
- root
 - sudo 83
 - SysAdmin einrichten 83
 - temporäre Rechte 51
- Rotfuchs 16
- S**
- Samba 17
- Schleife 74
 - for 74
 - while 75
- Secure Shell 117
- Server 14
 - Datei 14
 - Datenbank 14
 - Groupware 14
 - Mail 14
- Session 48
- SHA-2 81
- Shebang 68
- Shell 20
 - Definition 20
- Sicherung
 - Differenzielle Sicherung 137
 - Generationenprinzip 137
 - Inkrementelle Sicherung 137
 - Komplettsicherung 137
 - Speicherabbild-Sicherung 137
 - Vollsicherung 137
- SIGKILL 26
- SIGTERM 26
- Simple Mail Transfer Protocol 117
- Skript 163
 - ausführen 68
 - Variablen 68
- Smartphone 15
- SMTP 116, 117
- Softwarelizenz 163
- Space Travel 12
- special file 31
- SSH 116, 117
- Standard-Gateway 122
- Standardbenutzer 48
- Subnet-Maske 117
- Superuser 83
 - einrichten 83
 - sudo 83
- switch user 51
- switch user do 51
- Systembenutzer 48
- Systemverwalter 48, 51
- Systemicherheit 163
- T**
- Tabellenkalkulation 161
- Tablet 15
- tar 134
- Tarball 134
- TCP 116, 117
- TCP/IP 116
 - Protokoll-Familie 116
 - Protokollstapel 116
 - Schichten 116
 - Stack 116
 - Vorteile 116
- Telnet 116
- Terminal 17, 18
- Textdateien
 - ausgeben 101
 - erste x-Bytes ausgeben 101
 - erste Zeilen ausgeben 101
 - filtern 101
 - in Felder aufteilen 104
 - in Spalten aufteilen 104
 - kontinuierlich ausgeben 102
 - letzte Zeilen ausgeben 101
 - nummeriert ausgeben 101
 - sortieren 102
 - Wörter zählen 104
 - Zeichen austauschen 104
 - Zeilen auswählen 103
- Texteditor
 - Nano 40
 - Pico 40
 - vi 41
 - vim 41
- Textverarbeitung 161
- The Document Foundation 16
- Thumbnails 109
- Thunderbird 16
- TIFF 99
- Transmission Control Protocol 116, 117
- TTY 25
- Tux 12
- twm 14
- U**
- Ubuntu 13
- udev 88
- UDP 117
- UID 49
- Unicast 119
- UNICS 12
- Uniform Resource Locator 120
- Unity 13, 14, 17
- Unix 12
 - Geschichte 12
- URL 120
- User
 - Datenbank 50
 - Einstellungen ändern 49
 - Gruppenzuordnung 53
 - hinzufügen 49
 - löschen 49
 - su 51
 - sudo 51
 - UID 49
 - wechseln 51

User Datagram Protocol	117	VPN	119
V		W	
Variable		Wertschöpfungskette	152
\$!	69	white spaces	79
\$*	69	Wildcard	20, 27
\$0	69	Wurzel	32
\$1..9	69	X	
\$?	69, 75	X-Window	19
\$@	69	Xfce	14
\$HISTFILE	20	Xfce-Terminalemulator	19
\$HISTSIZ	20	Xterm	19
\$HOME	68	Z	
\$USER	68	Zeichnungen	161
\$#	69	Zeitsteuerung	139
\$\$	69	zip	132
alle anzeigen	68	Zugriffsrechte	
löschen	68	chgrp	59
Typ	69	chmod	59
Verzeichnis	31	chown	59
Baum	32	Oktal	60
FHS	32	ändern	59
Struktur	32	Zugriffsverwaltung	48, 58
vi	40		
Video on Demand	119		

