

Inhaltsverzeichnis

Dateiverwaltung

cat	verbindet mehrere Dateien zu einer Gesamtdatei	24
cd	wechselt das aktuelle Verzeichnis	25
chgrp	ändert die Gruppenzugehörigkeit einer Datei	26
chmod	ändert die Zugriffsbits einer Datei	27
chown	ändert den Besitzer einer Datei	28
cp	kopiert Dateien	31
file	versucht, den Typ einer Datei festzustellen	51
ln	stellt feste und symbolische Links zu Dateien her	80
ls	zeigt das Inhaltsverzeichnis an	83
mkdir	erzeugt ein neues Verzeichnis	92
mv	verschiebt Dateien bzw. ändert ihren Namen	110
rm	löscht Dateien	123
rmdir	löscht Verzeichnisse	123
rsync	synchronisiert Verzeichnisse	127
tee	dupliziert die Standardeingabe	148

Access Control Lists (ACLs) und Extended Attributes (EAs)

attr	verwaltet die Zusatzattribute einer Datei	22
chacl	verwaltet die ACLs einer Datei	25
getfacl	ermittelt die ACLs einer Datei	60
getfattr	ermittelt die Zusatzattribute einer Datei	61
setfacl	verändert die ACLs einer Datei	130
setfattr	verändert die Zusatzattribute einer Datei	132

Dateien suchen

find	sucht Dateien nach Name, Datum, Größe etc.	51
grep	sucht Text in einer Textdatei	61
locate	sucht Dateien in einer dafür vorbereiteten Datenbank	81
updatedb	aktualisiert die Suchdatenbank für locate	157
whereis	sucht Dateien in vordefinierten Verzeichnissen	163
which	durchsucht die PATH-Verzeichnisse nach Kommandos	163

Bearbeitung von Textdateien

cat	zeigt die Datei an bzw. vereint mehrere Texte	24
csplit	zerlegt den Text an vorgegebenen Stellen in Einzeldateien	34
cut	extrahiert Spalten aus jeder Zeile des Textes	36
diff	vergleicht zwei Texte	39
expand	ersetzt Tabulator- durch Leerzeichen	47
fold	zerlegt lange Textzeilen in kürzere	54
grep	sucht Texte innerhalb der Datei	61
head	zeigt die ersten Zeilen der Datei an	69
iconv	ändert den Zeichensatz von Textdateien	69
less	zeigt Dateien seitenweise an (mit Rückwärtsbewegung)	78
more	zeigt Dateien seitenweise an	98
paste	vereint mehrere Texte zeilenweise	113
patch	ändert Textdateien gemäß einer diff-Datei	113
recode	konvertiert zwischen verschiedenen Zeichensätzen	120
sed	Stream-Editor (programmierbarer Editor)	129
sort	sortiert Dateien	139
split	zerlegt eine Datei in Teildateien mit vorgegebener Größe	140
strings	zeigt Zeichenketten innerhalb einer Binärdatei an	142
tac	zeigt Text in inverser Reihenfolge an (die letzte Zeile zuerst)	144
tail	zeigt das Ende der Datei an	144
tr	ersetzt vorgegebene Zeichen durch andere Zeichen	151
uniq	eliminiert mehrfach auftretende Zeilen in einer Textdatei	155
zcat	zeigt eine komprimierte Textdatei an	171
zless	zeigt eine komprimierte Textdatei an (auch rückwärts)	171
zmore	zeigt eine komprimierte Textdatei seitenweise an	171

Dateien komprimieren und archivieren

bunzip2	dekomprimiert Dateien, die mit bzip2 komprimiert wurden	23
bzip2	komprimiert Dateien; leistungsfähiger als gzip	23
cpio	überträgt Archivdateien zwischen Dateisystemen	33
compress	komprimiert Dateien	29
gunzip	dekomprimiert Dateien, die mit gzip komprimiert wurden	67
gzip	komprimiert Dateien; leistungsfähiger als compress	67
mt	steuert den Streamer (Vor- und Rückspulen etc.)	109
tar	vereint mehrere Dateien (und Verzeichnisse) in einer Datei	145
uncompress	dekomprimiert durch compress komprimierte Dateien	154
unxz	dekomprimiert Dateien, die mit xz komprimiert wurden	155
unzip	dekomprimiert ein Windows-kompatibles ZIP-Archiv	155
xz	komprimiert Dateien; leistungsfähiger als bzip2	168
zip	erzeugt ein Windows-kompatibles ZIP-Archiv	171
zipinfo	zeigt den Inhalt eines ZIP-Archivs an	171

CDs/DVDs schreiben

cdrdao	schreibt eine CD im Disk-at-once-Modus	25
dvd+rw-format	formatiert DVD+RW- und DVD-RW-Medien	43
dvd+rw-mediainfo	liefert Informationen über ein optisches Datenmedium	43
genisoimage	erzeugt ein ISO-Image (ehemals mki sofs)	57
growisofs	schreibt DVDs und Blu-Ray-Discs	65
mkudffs	richtet ein UDF-Dateisystem ein (z. B. auf einer DVD-RAM)	97
wodim	schreibt eine CD oder DVD (ehemals cdrecord)	164

Konverter

a2ps	konvertiert Textdateien in das PostScript-Format	17
convert	konvertiert Grafikdateien zwischen verschiedenen Formaten	29
convmv	ändert den Zeichensatz von Dateinamen	31
dvips	wandelt eine DVI-Datei in das PostScript-Format um	44
enscript	konvertiert Textdateien in das PostScript-Format	45
epstopdf	konvertiert EPS-Dateien in PDF-Dateien	46
ffmpeg	konvertiert Video-Dateien	49
iconv	ändert den Zeichensatz von Textdateien	69
mogrify	verändert Parameter einer Bilddatei	97
mpage	konvertiert Textdateien in das PostScript-Format	109
pdf2ps	konvertiert PDF-Dateien in PostScript-Dateien	114
pdftk	manipuliert PDF-Dateien	114
pdftops	Alternative zu pdf2ps	115
pdftotext	wandelt ein PDF-Dokument in eine reine Textdatei um	116
ps2pdf	konvertiert PostScript-Dateien in PDF-Dateien	118
recode	ändert den Zeichensatz von Textdateien	120

Programm-/Prozessverwaltung

bg	setzt einen Prozess im Hintergrund fort	23
chroot	startet eine Shell mit einem veränderten Wurzelverzeichnis	28
fg	setzt einen Prozess im Vordergrund fort	50
fuser	ermittelt das Programm, das auf eine Datei zugreift	57
halt	beendet Linux und hält den Rechner an	67
iotop	zeigt die Prozesse mit der größten IO-Aktivität an	73
kill	versendet Signale (meist zum Beenden von Prozessen)	77
killall	wie kill; der Prozess wird mit Namen genannt	78
ldconfig	aktualisiert die Cache-Datei zur Bibliothekssuche	78
ldd	liefert alle erforderlichen Bibliotheken eines Programms	78
lsof	listet offene Dateien und die zugeordneten Prozesse auf	86
nice	startet ein Programm mit verringerter Priorität	111
nohup	startet einen »unzerstörbaren« Prozess	112

pidof	ermittelt die Prozessnummer eines laufenden Programms	116
powertop	analysiert den Energieverbrauch der laufenden Prozesse	117
ps	zeigt die Liste der laufenden Prozesse an	117
pstree	wie ps; macht die Abhängigkeiten besser sichtbar	118
reboot	beendet Linux und startet den Rechner neu	120
renice	verändert die Priorität eines laufenden Prozesses	121
sudo	führt ein Programm als root aus	143
top	zeigt alle fünf Sekunden eine Liste aller Prozesse an	150
watch	führt ein Kommando periodisch aus und zeigt die Ausgaben an	160

Verwaltung von Benutzern und Gruppen

adduser	richtet einen neuen Benutzer ein (Debian)	18
addgroup	richtet eine neue Gruppe ein (Debian)	18
chgrp	ändert die Gruppenzugehörigkeit einer Datei	26
chown	ändert den Besitzer einer Datei	28
chsh	verändert die Default-Shell eines Benutzers	28
delgroup	löscht eine Gruppe (Debian)	38
deluser	löscht einen Benutzer (Debian)	38
groupadd	richtet eine neue Gruppe ein	64
groupdel	löscht eine Gruppe	64
groupmod	verändert Gruppeneigenschaften	64
groups	zeigt die Gruppen des aktuellen Benutzers an	64
id	zeigt die aktuelle Benutzer- und Gruppen-ID-Nummer an	69
makepasswd	erzeugt ein neues, zufälliges Passwort	89
newgrp	ändert die aktive Gruppe eines Benutzers	111
newusers	richtet mehrere neue Benutzer ein	111
passwd	verändert das Passwort eines Benutzers	113
pwgen	generiert leicht merkbare Passwörter	120
useradd	richtet einen neuen Benutzer ein	158
userdel	löscht einen Benutzer	158
usermod	verändert Benutzereigenschaften	158

Administration des Dateisystems

badblocks	testet, ob Datenträger defekte Sektoren enthalten	22
blkid	liefert die UUID, den Namen und andere Partitionsdaten	23
cryptsetup	richtet ein Crypto-Device ein	34
dd	kopiert Datenblöcke zwischen Devices (low level)	37
df	zeigt den freien Speicher auf der Festplatte an	39
du	ermittelt den Platzbedarf eines Verzeichnisses	42
dumpe2fs	zeigt interne Informationen über ein ext-Dateisystem an	43
eject	wirft eine CD oder Diskette aus	45

fdformat	formatiert eine Diskette	48
fdisk	partitioniert die Festplatte	48
hdparm	verändert Parameter der Festplatte	68
luksformat	richtet ein Crypto-Device samt Dateisystem ein	87
mdadm	verwaltet RAID-Partitionen	90
mkfifo	erzeugt eine FIFO-Datei (eine benannte Pipe)	92
mkfs	richtet ein Dateisystem ein	93
mknod	erstellt Device-Dateien	96
mkswap	richtet eine Datei oder eine Partition als Swap-Bereich ein	96
mount	bindet ein Dateisystem in den Verzeichnisbaum ein	98
resize2fs	verändert die Größe eines ext-Dateisystems	121
smartctl	steuert die SMART-Funktionen der Festplatte	136
sfdisk	partitioniert die Festplatte	134
swapoff	deaktiviert eine Swap-Datei oder -Partition	143
swapon	aktiviert eine Swap-Datei oder -Partition	143
sync	führt alle gepufferten Schreiboperationen aus	144
tune2fs	verändert Systemparameter eines ext-Dateisystems	152
umount	löst ein Dateisystem aus dem Verzeichnisbaum	154
volname	liefert den Partitionsnamen einer CD	160

LVM-Kommandos

lvcreate	richtet ein neues LV (<i>logical volume</i>) ein	87
lvdisplay	liefert Detailinformationen zu einem LV	88
lvextend	vergrößert ein LV	88
lvreduce	verkleinert ein LV	88
lvremove	löscht ein LV	89
lvrename	gibt dem LV einen neuen Namen	88
lvscan	listet alle LVs auf	89
pvcreate	kennzeichnet eine Partition als PV (<i>physical volume</i>)	119
pvdisplay	liefert Detailinformationen zu einem PV	119
pvremove	entfernt die PV-Kennzeichnung eines ungenutzten PVs	119
pvscan	listet alle PVs auf	119
vgchange	ändert die Attribute einer VG (<i>volume group</i>)	158
vgcreate	erzeugt eine neue VG aus einem oder mehreren PVs	159
vgdisplay	liefert Detailinformationen zu einer VG	159
vgextend	vergrößert eine VG um ein PV	159
vgmerge	vereint zwei VGs	159
vgreduce	verkleinert eine VG um ein ungenutztes PV	159
vgrename	gibt einer VG einen neuen Namen	159
vgscan	listet alle VGs auf	160

Paketverwaltung

add-apt-repository	richtet eine PPA-Paketquelle ein (Ubuntu)	18
alien	wandelt Pakete zwischen verschiedenen Formaten um	19
alternatives	richtet Links in /etc/alternatives ein (Fedora, Red Hat)	19
apt-cache	liefert Informationen über installierte/verfügbare Pakete	19
apt-cdrom	richtet eine CD/DVD als APT-Paketquelle ein	20
apt-get	hilft bei der DEB-Paketverwaltung	20
apt-key	richtet einen Schlüssel für eine APT-Paketquelle ein	22
aptitude	hilft bei der DEB-Paketverwaltung	22
dpkg	(de)installiert oder aktualisiert DEB-Pakete	40
rpm	(de)installiert oder aktualisiert RPM-Pakete	125
tasksel	(de)installiert DEB-Paketgruppen	147
update-alternatives	richtet Links in /etc/alternatives ein	155
yum	hilft bei der RPM-Paketverwaltung (Fedora, Red Hat)	168
yumdownloader	lädt den Quellcode von Paketen herunter	170
zypper	hilft bei der RPM-Paketverwaltung (SUSE-spezifisch)	171

Netzwerkfunktionen

ftp	überträgt interaktiv Dateien via FTP	56
host	löst IP-Nummern bzw. Netzwerknamen auf	69
hostname	liefert bzw. verändert den Namen des lokalen Rechners	69
ifconfig	konfiguriert Netzwerkschnittstellen bzw. ermittelt deren Status	70
ifdown	deaktiviert eine Netzwerkschnittstelle	71
ifup	aktiviert eine Netzwerkschnittstelle	71
ip	zeigt Netzwerkeinstellungen an bzw. verändert sie	73
iptables	konfiguriert den Linux-Paketfilter (Firewall)	74
iwconfig	wie ifconfig, aber für WLAN-Schnittstellen	77
iwlist	liefert Informationen zum WLAN-Controller und -Netz	77
netstat	analysiert die Netzwerkaktivität auf dem lokalen Rechner	110
nmap	analysiert die Netzwerkaktivität auf einem fremden Rechner	112
ping	überprüft die Netzwerkverbindung zu einem anderen Rechner	116
rdiff-backup	erstellt inkrementelle Backups	122
route	verändert bzw. zeigt die IP-Routing-Tabelle	124
rsync	synchronisiert Netzwerkverzeichnisse	127
scp	überträgt Dateien verschlüsselt via SSH	129
sftp	überträgt Dateien via SFTP	135
smbclient	überträgt Dateien aus/in Windows-Netzwerkverzeichnisse	138
smbpasswd	ändert ein Samba-Passwort	138
smbtree	liefert eine Liste aller Windows-Netzwerkverzeichnisse	139
ssh	ermöglicht Logins auf anderen Rechnern im Netzwerk	141
telnet	kommuniziert interaktiv mit einem Netzwerkdienst	148
traceroute	liefert die Zwischenstationen zu einer Netzwerkadresse	151
wget	lädt selbstständig Dateien oder Verzeichnisse herunter	161

Druckeradministration

<code>lpadmin</code>	richtet neue Drucker ein bzw. löscht sie wieder (CUPS)	81
<code>lpinfo</code>	listet alle Druck-Devices, Druckertreiber etc. auf (CUPS)	81
<code>lpoptions</code>	zeigt die Optionen von Druckern an bzw. verändert sie (CUPS)	82
<code>lpq</code>	zeigt den Inhalt einer Druckerwarteschlange an	82
<code>lpr</code>	druckt eine Datei aus	82
<code>lprm</code>	löscht einen Druck-Job aus der Warteschlange	82
<code>lpstat</code>	liefert Informationen über Drucker, Druck-Jobs etc. (CUPS)	83

Hardware-Verwaltung

<code>acpi</code>	liefert Informationen über den Batteriezustand	17
<code>free</code>	zeigt den freien Speicherplatz (RAM und Swap-Speicher) an	55
<code>lshal</code>	liefert Informationen des <i>Hardware Abstraction Layer</i> (HAL)	85
<code>lspci</code>	liefert Informationen über PCI-Komponenten	86
<code>lsscsi</code>	liefert Informationen über angeschlossene SCSI-Geräte	87
<code>lsusb</code>	liefert Informationen über angeschlossene USB-Geräte	87

Kernel und Module

<code>depmod</code>	aktualisiert die Modulabhängigkeitsdatei <code>modules.dep</code>	39
<code>insmod</code>	lädt ein Modul in den Kernel	72
<code>lsmod</code>	liefert die Liste aller geladenen Module	86
<code>modinfo</code>	liefert Informationen über ein Modul	97
<code>modprobe</code>	lädt ein Modul und alle abhängigen Module in den Kernel	97
<code>rmmod</code>	entfernt ein Modul aus dem Kernel	124
<code>sysctl</code>	verändert Kernelparameter	144

Systemstart und -stopp, Init-V-Prozess, Upstart

<code>chkconfig</code>	richtet Init-V-Links ein (Red Hat, Fedora)	26
<code>grub</code>	startet die GRUB-Shell (GRUB 0.97 <i>legacy</i>)	66
<code>grub-install</code>	installiert GRUB in den Bootsektor (GRUB 2)	66
<code>invoke-rc</code>	führt ein Init-V-Script aus (Debian, Ubuntu)	73
<code>lilo</code>	richtet den Boot-Loader LILO ein	79
<code>init</code>	wechselt in einen anderen Runlevel (Init-V-System)	72
<code>initctl</code>	steuert Upstart und generiert Upstart-Ereignisse	72
<code>insserv</code>	richtet Init-V-Links ein (SUSE)	72
<code>mkinitrd</code>	erzeugt eine Initrd-Datei (Fedora, Red Hat, SUSE)	95
<code>service</code>	führt ein Init-V-Script aus (Fedora, Red Hat, Ubuntu)	130
<code>shutdown</code>	beendet Linux	136
<code>start</code>	startet einen durch Upstart verwalteten Hintergrundprozess	142
<code>stop</code>	stoppt einen durch Upstart verwalteten Hintergrundprozess	142

telinit	ändert den Runlevel	148
update-grub	aktualisiert die GRUB-Konfigurationsdatei menu.lst (Debian)	156
update-initramfs	erzeugt oder aktualisiert eine Initrd-Datei (Debian, Ubuntu)	156
update-rc.d	richtet Init-V-Links ein (Debian, Ubuntu)	157

Terminal und Textkonsole

echo	gibt eine Zeile Text aus	45
printf	ermöglicht eine formatierte Ausgabe wie unter C	117
reset	führt einen Reset für das Terminal durch	121
setfont	verändert die Terminal-Schriftart	133
setterm	verändert diverse Terminaleinstellungen	133

Online-Hilfe

apropos	sucht Kommandos zu einem Thema	19
help	zeigt die Beschreibung eines Shell-Kommandos an	69
info	startet das info-System	71
man	zeigt die Beschreibung eines Kommandos an	89
what is	zeigt eine Kurzbeschreibung (eine Zeile) eines Kommandos an	163

X

fc-list	listet alle skalierbaren Schriften auf	48
xkill	beendet ein Programm per Mausklick	166
xrandr	ändert die Auflösung oder andere Parameter des Grafiksystems	167
xset	ändert Benutzereinstellungen des Grafiksystems	168

Sonstiges

alias	definiert eine Abkürzung	19
basename	ermittelt den Dateinamen eines Pfads	22
cksum	berechnet die CRC-Prüfsumme zu einer Datei	29
date	zeigt Datum und Uhrzeit an	37
dirname	ermittelt das Verzeichnis eines Pfads	40
dmesg	zeigt die Kernelmeldungen des Bootvorgangs an	40
expr	führt Berechnungen und Mustervergleiche durch	47
hash	zeigt die Hash-Tabelle an	68
ldd	zeigt die erforderlichen Libraries für ein Programm an	78
md5sum	berechnet eine Prüfsumme zu einer Datei	90
printenv	zeigt nur die Umgebungsvariablen an	117
set	zeigt alle der Shell bekannten Variablen an	130
shasum	berechnet eine Prüfsumme zu einer Datei	135

sleep	wartet eine vorgegebene Zeit	136
tty	zeigt den Device-Namen des aktuellen Terminals an	152
type	gibt den Typ eines Kommandos an (z. B. Shell-Kommando)	153
unalias	löscht eine Abkürzung	154
uname	zeigt den Betriebssystemnamen und die Versionsnummer an	154
xargs	leitet die Standardeingabe an ein anderes Kommando weiter	166

bash-Variablenverwaltung

alias	definiert eine Abkürzung	19
declare	definiert eine (Umgebungs-)Variable	38
export	definiert eine Umgebungsvariable	47
local	definiert lokale Variablen in einer Funktion	80
read	liest eine Variable ein	119
readonly	zeigt alle schreibgeschützten Variablen an	120
shift	verschiebt die Parameterliste	135
unalias	löscht eine Abkürzung	154
unset	löscht eine Variable	155

bash-Programmierung

break	beendet eine Schleife vorzeitig	23
case	leitet eine Fallunterscheidung ein	24
continue	überspringt den Schleifenkörper	29
exit	beendet das Shell-Programm	47
for	leitet eine Schleife ein	54
function	definiert eine neue Funktion	57
if	leitet eine Verzweigung ein	70
local	definiert lokale Variablen in einer Funktion	80
source	führt die angegebene Shell-Datei aus	140
test	wertet eine Bedingung aus	149
until	leitet eine Schleife ein (Variante 1)	155
while	leitet eine Schleife ein (Variante 2)	163

bash-Kommandos

dirs	zeigt die Liste der gespeicherten Verzeichnisse an	40
eval	wertet das angegebene Kommando aus	46
popd	wechselt in das letzte gespeicherte Verzeichnis	117
pushd	speichert das aktuelle und wechselt in ein neues Verzeichnis	118
trap	führt beim Eintreten eines Signals ein Kommando aus	152
ulimit	kontrolliert die von der Shell beanspruchten Ressourcen	153
wait	wartet auf das Ende eines Hintergrundprozesses	160

Einleitung

Dieses Buch enthält Kurzbeschreibungen der wichtigsten Linux-Kommandos zur Verwaltung des Dateisystems, zum Starten und Beenden von Prozessen, zur Bearbeitung von Textdateien, für andere administrative Aufgaben sowie zur bash-Programmierung.

Auf eher selten benötigte Kommandos sowie auf eine vollständige Beschreibung aller Optionen habe ich bewusst verzichtet: einerseits aus Platzgründen, andererseits aber auch, um die Referenz übersichtlich und anwenderfreundlich zu halten.

Haben Sie daher bitte Verständnis dafür, wenn ausgerechnet Ihr Lieblingskommando in dieser Referenz fehlt! Wenn Sie zu einem Kommando weitere Details wissen möchten, erhalten Sie diese in der Regel mit `man name` oder `info name`.

Je nachdem, welche Distribution Sie einsetzen, kann es vorkommen, dass einige Kommandos standardmäßig nicht zur Verfügung stehen und extra installiert werden müssen. Zudem gibt es distributionsspezifische Kommandos, die nur unter bestimmten Distributionen zur Verfügung stehen – z. B. die Paketverwaltungskommandos `dpkg` (Debian, Ubuntu), `rpm` (Red Hat, SUSE) und `zypper` (SUSE). In der Kommandobeschreibung weise ich auf diesen Umstand jeweils hin.

Was ist ein Kommando?

Linux unterscheidet nicht zwischen Kommandos, wie sie in diesem Buch beschrieben werden, und Programmen wie Firefox, OpenOffice oder Gimp. »Kommando« meint hier Programme ohne grafische Benutzeroberfläche, die in der Regel in einem Terminalfenster ausgeführt werden.

Ich beschreibe in diesem Buch auch einige Kommandos, die gar keine echten Programme sind, sondern nur Befehle der gerade aktiven Shell. Dabei gehe ich davon aus, dass Sie die `bash` (Bourne Again Shell) verwenden, die standardmäßig bei nahezu allen Linux-Distributionen für das interaktive Ausführen von Kommandos eingesetzt wird. Ein typisches Beispiel ist das häufig eingesetzte Kommando `cd` zum Wechseln des aktuellen Verzeichnisses. Derartige Kommandos sind in der Referenz mit dem Hinweis (`bash`) in der Überschrift gekennzeichnet.



Optionen

Die meisten in diesem Buch beschriebenen Kommandos werden durch Optionen gesteuert. Die Angabe der Optionen erfolgt *vor* allen weiteren Parametern. Bei vielen Kommandos gibt es zwei Schreibweisen: `-x` für kurze Optionen (ein Buchstabe) und `--xyz` für lange Optionen (mehrere Buchstaben).

Die beiden folgenden `ls`-Kommandos sind gleichwertig und zeigen jeweils alle Dateien und Verzeichnisse im `/usr`-Verzeichnis an. (Eine genaue Beschreibung des Kommandos und seiner Optionen finden Sie auf Seite 83.)

```
user$ ls -l -A /usr
user$ ls --format=long --almost-all /usr
```

Bei manchen Kommandos können mehrere Optionen als Gruppe angegeben werden (also `-ab` statt `-a -b`). Manche Kommandos kommen auch mit Optionen zurecht, die hinter dem oder den eigentlichen Parametern angegeben werden. Das sollte Sie aber nicht zu dem Schluss verleiten, dass das für alle Kommandos gilt!

```
user$ ls -lA /usr
user$ ls /usr -lA
```

Bei einigen wenigen Kommandos hat die Reihenfolge der Parameter einen Einfluss darauf, wie das Kommando ausgeführt wird. Wenn Optionen angegeben werden, die einander gegenseitig logisch ausschließen, gilt in der Regel die letzte angegebene Option.

man, info und help

Um das Kapitel nicht unnötig aufzublähen, beschreibe ich nur die wichtigsten Optionen. Eine vollständige Übersicht aller Optionen liefert bei der Mehrzahl der Kommandos `kommandoname --help`. Ausführlichere Informationen sind zumeist in den Manual-Seiten enthalten, die mit `man name` bzw. mit `man 1 name` angezeigt werden können. Bei manchen Kommandos enthalten die `man`-Seiten lediglich einen Verweis auf die `info`-Texte, die entsprechend mit `info name` angezeigt werden.

Bei Kommandos, die direkt in die `bash` integriert sind (z. B. `cd`), führt `man name` zur `man`-Seite der `bash`. Dort ist das Kommando zwar tatsächlich beschrieben, aber die Suche in der sehr langen Dokumentation ist mühsam. Hilfreicher ist hier `help name`.

Kommandos von A bis Z

a2ps [optionen] textdatei -o psdatei

a2ps wandelt Textdateien in das PostScript-Format um. Standardmäßig wird der Text über zwei Spalten auf eine Seite im Querformat verteilt. Bei Programmcode führt a2ps automatisch eine Syntaxhervorhebung durch.

- fn stellt die Schriftgröße auf *n* Punkt ein. Damit ändert sich automatisch auch der Zeilenabstand und damit die Anzahl der Zeilen pro Seite. *n* ist eine Fließkommazahl.
- R Längsformat (Portrait) statt Querformat (Landscape).
- highlight-level=none
keine Syntaxhervorhebung durchführen.
- list=defaults
liefert eine Liste aller Standardeinstellungen von a2ps.
- M *papierformat* bzw. --medium=*papierformat*
verwendet das angegebene Papierformat (z. B. A3, A4, A5 oder Letter).
- o dateiname
speichert den resultierenden PostScript-Code in der angegebenen Datei. Ohne diese Option wird der Text direkt am Standarddrucker ausgedruckt.
- X *charset*
gibt den Zeichensatz des Texts an (z. B. ASCII, *ison*, *latinn*). Unicode wird leider nicht unterstützt!

acpi [optionen]

acpi liefert Informationen zur Energieversorgung des Rechners.

- a zeigt an, ob der Rechner an das Stromnetz angeschlossen ist.
- b zeigt den Batteriezustand an.
- t zeigt die Temperatur des Akkus an.



add-apt-repository ppa:name

Das Ubuntu-spezifische Kommando `add-apt-repository` richtet eine neue Paketquelle für ein *Personal Package Archive* (PPA) ein. PPAs sind nichtoffizielle Paketquellen von Ubuntu-Entwicklern, in denen oft besonders aktuelle Versionen populärer Programme zu finden sind.

Beispiel

Das folgende Kommando richtet eine Paketquelle für die jeweils aktuellsten Versionen von Firefox und Thunderbird ein:

```
root# add-apt-repository ppa:ubuntu-mozilla-daily
```

addgroup [optionen] name

`addgroup` richtet auf Debian-Systemen eine neue Gruppe ein und berücksichtigt dabei die Einstellungen aus `/etc/adduser.conf`. Bei Fedora und Red Hat ist `addgroup` ein Link auf das Kommando `groupadd`, dessen Syntax auf Seite 64 beschrieben ist.

`--gid n`

weist der Gruppe die angegebene GID-Nummer zu (*group identification*).

adduser [optionen] name**adduser** name group

`adduser` richtet auf Debian-Systemen einen neuen Benutzer ein und berücksichtigt dabei die Einstellungen aus `/etc/adduser.conf`. Insbesondere wird für jeden Benutzer auch eine gleichnamige Gruppe erzeugt und dem neuen Benutzer zugeordnet. In der zweiten Syntaxvariante fügt `adduser` den bereits vorhandenen Benutzer einer zusätzlichen Gruppe hinzu.

Bei Fedora und Red Hat ist `adduser` ein Link auf das Kommando `useradd`, dessen Syntax auf Seite 158 beschrieben ist.

`--disabled-login`

verzichtet auf die Passwort-Frage. Ein Login ist unmöglich, bis das Passwort eingestellt wird.

`--gecos 'vollständiger Name'`

verzichtet auf die Fragen nach dem vollständigen Namen, dem (Büro-)Raum, den Telefonnummern und anderen Informationen.

`--group`

richtet nur eine neue Gruppe ein (keinen Benutzer). `adduser --group` entspricht `addgroup`.

`--system`

richtet einen System-Benutzer ein. Dabei wird die UID aus dem Bereich für System-Benutzer gewählt (in der Regel 100–999) und der Login blockiert (Shell `/bin/false`). Es kann kein Passwort angegeben werden.

`--uid n`

weist dem Benutzer die angegebene UID-Nummer zu (*user identification*).

Beispiel

Das folgende Kommando richtet auf einem Debian-System den neuen Benutzer `kofler` ein:

```
root# adduser --gecos 'Michael Kofler' kofler
```

```
alias abkürzung='kommando' (bash)
```

`alias` definiert eine neue Abkürzung bzw. zeigt eine vorhandene Abkürzung an. Wenn `alias` ohne weitere Parameter verwendet wird, werden alle definierten Abkürzungen angezeigt.

Beispiel

Das folgende Kommando definiert die Abkürzung `ll` für das Kommando `ls -l`:

```
user$ alias ll='ls -l'
```

```
alien [optionen] paketdatei
```

`alien` wandelt die angegebene Paketdatei in ein anderes Format um. Das gewünschte Format wird durch die Optionen `--to-deb` (Debian), `--to-rpm` (RPM) oder `--to-tgz` (tar-Archiv) angegeben. `alien` muss von `root` ausgeführt werden, damit die Besitzer und Zugriffsrechte der neuen Pakete richtig eingestellt werden.

```
alternatives [optionen] kommando
```

`alternatives` ist die Red-Hat- bzw. Fedora-Variante von `update-alternatives`. Die Syntax der beiden Kommandos ist weitgehend identisch und ist auf Seite 155 beschrieben.

```
apropos thema
```

`apropos` liefert eine Liste aller `man`-Texte, die Informationen zum angegebenen Thema enthalten. Wenn `apropos` nicht funktioniert, fehlen wahrscheinlich die zugrunde liegenden Datenbanken, die mit `mandb` bzw. mit `makewhatis` erzeugt werden können.

```
apt-cache kommando
```

`apt-cache` liefert Informationen über verfügbare bzw. bereits installierte Debian-Pakete.

```
show paketname
```

liefert eine kurze Paketbeschreibung. Das funktioniert auch für nicht installierte Pakete, sofern sich die Paketbeschreibung im Cache befindet.

`showpkg paketname`

zeigt die Abhängigkeitsinformationen für das Paket an.

`search muster`

liefert eine Liste aller verfügbaren Pakete (egal, ob sie bereits installiert sind oder nicht), in deren Paketnamen oder Paketkurzbeschreibung der Suchtext *muster* vorkommt. Die zusätzliche Option `--names-only` schränkt die Suche auf den Paketnamen ein.

`stats`

liefert eine Statistik über die Anzahl der installierten und verfügbaren Pakete.

`apt-cache` ist nicht in der Lage, Informationen über den genauen Inhalt eines Pakets zu geben bzw. eine Zuordnung zwischen einer Datei und dem dazugehörigen Paket herzustellen. Wenn Sie an diesen Informationen interessiert sind, müssen Sie `dpkg` einsetzen (siehe Seite 40).

Beispiel

Das folgende Kommando liefert eine sortierte Liste aller Pakete, deren Namen `gimp` enthält:

```
root# apt-cache search --names-only gimp | gimp
```

`apt-cdrom [optionen] add`

`apt-cdrom` ermittelt alle auf der CD verfügbaren Pakete und trägt sie in eine APT-Cache-Datei ein. Gleichzeitig wird die CD als zusätzliche Paketquelle in `/etc/apt/sources.list` eingefügt.

Damit `apt-cdrom` richtig funktioniert, muss die CD Metadaten für das APT-Paketverwaltungssystem enthalten. Wenn `apt-cdrom` die CD/DVD nicht findet, geben Sie deren `mount`-Verzeichnis mit der Option `-d` an.

`apt-get [optionen] kommando`

`apt-get` installiert, aktualisiert und entfernt Debian-Pakete. `apt-get` lädt dabei die Pakete von den in `/etc/apt/sources.list` definierten Paketquellen herunter. Sämtliche Paketverwaltungsfunktionen werden in Form von Kommandos durchgeführt (z. B. `apt-get install paketname`). Die folgende Aufzählung beschreibt die wichtigsten Kommandos:

`update`

aktualisiert die Paketlisten der in `sources.list` angegebenen Archive. Dabei werden nur die Metadaten der Paketquellen gelesen und in APT-Cache-Dateien eingetragen. Es werden aber keine Pakete heruntergeladen oder aktualisiert! Der einzige Zweck dieses Kommandos ist, dass `apt-get` weiß, welche Pakete im Internet zur Verfügung stehen. Das Kommando sollte vor jedem anderen `apt`-Kommando ausgeführt werden.

`upgrade`

aktualisiert alle installierten Pakete, soweit in den Paketquellen neuere Versionen zur Verfügung stehen.

dist-upgrade

funktioniert ähnlich wie `upgrade`, installiert bei Bedarf aber auch neue Pakete.

install name1 name2 ...

sucht die Pakete `name1`, `name2` etc. auf allen APT-Paketquellen, lädt sie herunter und installiert sie. Gegebenenfalls werden auch weitere Pakete geladen und installiert oder aktualisiert, um Paketabhängigkeiten zu erfüllen. Bei `name1`, `name2` etc. darf es sich auch um lokale Debian-Dateien handeln. Damit werden diese Pakete installiert, wobei zur Auflösung der Paketabhängigkeiten weiterhin alle APT-Paketquellen genutzt werden.

remove name1 name 2 ...

deinstalliert die angegebenen Pakete.

autoremove

deinstalliert alle nicht mehr benötigten Pakete, die aufgrund von Abhängigkeiten installiert wurden.

source name

installiert den Quellcode des Pakets in das aktuelle Verzeichnis.

check

aktualisiert den Cache aller installierten Pakete und stellt eventuell vorhandene Paketkonflikte und ungelöste Abhängigkeiten fest. Das ist nur erforderlich, wenn Pakete ohne APT (de)installiert wurden und in der Folge Konflikte aufgetreten sind.

clean

entfernt alle heruntergeladenen Pakete aus dem Paket-Cache.

autoclean

entfernt nur solche Pakete aus dem Paket-Cache, zu denen bereits eine neuere Version verfügbar ist.

Das Detailverhalten bei der Ausführung der Kommandos wird durch Optionen gesteuert:

-d bzw. --download-only

lädt die Pakete nur in das Verzeichnis `/var/cache/apt/archives` herunter, installiert sie aber nicht.

--no-install-recommends

verzichtet auf die Installation empfohlener Pakete.

-s bzw. --simulate

simuliert die Installation, führt aber keine tatsächlichen Veränderungen durch.

-y bzw. --assume-yes

beantwortet alle Fragen mit `yes` und ermöglicht so die Verwendung des Kommandos in einem Script.

Beispiel

Die folgenden Kommandos zeigen die Anwendung von `apt-get`:

```
root# apt-get update           (Paketquellen aktualisieren)
root# apt-get dist-upgrade    (alle installierten Pakete aktualisieren)
root# apt-get install jmacs    (das Paket jmacs installieren)
```

apt-key kommando

apt-key verwaltet die öffentlichen Schlüssel von APT-Paketquellen. apt-key kennt die folgenden Kommandos:

add schlüsseldatei.gpg

fügt den Schlüssel aus der angegebenen Datei zur Liste der APT-Schlüssel hinzu.

del id

löscht den anhand der ID-Nummer spezifizierten Schlüssel aus der Liste der APT-Schlüssel.

list

liefert eine Liste aller dem APT-System bekannten Schlüssel.

aptitude [optionen] kommando

aptitude installiert, aktualisiert und entfernt Debian-Pakete und greift dabei wie apt-get auf die APT-Infrastruktur zurück. Der Vorteil im Vergleich zu apt-get besteht darin, dass sich aptitude bei der Installation abhängige Pakete merkt und diese bei der Deinstallation automatisch wieder entfernt.

Sämtliche Paketverwaltungsfunktionen werden in Form von Kommandos durchgeführt (z. B. aptitude install paketname). Die elementaren Kommandos stimmen mit denen von apt-get überein und sind auf Seite 20 beschrieben. Die Kommandos check und autoremove sind in aptitude allerdings nicht verfügbar.

attr [optionen] dateien

attr ermittelt bzw. verändert die erweiterten Zugriffsattribute der angegebenen Dateien bzw. Verzeichnisse. Das funktioniert nur, wenn das Dateisystem EAs (*Extended Attributes*) unterstützt. Bei ext3/ext4-Dateisystemen muss dazu die mount-Option user_xattr verwendet werden.

Statt attr sollten Sie nach Möglichkeit getfattr bzw. setfattr einsetzen (siehe Seite 61 bzw. 132). attr steht nur aus Kompatibilitätsgründen zu IRIX SGI zur Verfügung.

badblocks device blocks

badblocks überprüft die durch block angegebene Anzahl von Blöcken des Datenträgers. Wenn Sie vorhaben, auf der Festplatte(npartition) ein ext-Dateisystem einzurichten, ist es besser, die Überprüfung mit mkfs.extn -c im Rahmen der Formatierung durchzuführen.

basename zeichenkette [endung]

basename liefert den Dateinamen des übergebenen Pfads. basename /etc/X11/Xwrapper.per.config führt also zum Ergebnis Xwrapper.config. Wenn als zusätzlicher Parameter eine

Dateiendung angegeben wird, so wird diese Dateiendung (falls vorhanden) aus dem Dateinamen entfernt.

bg [prozess]

(bash)

bg setzt einen unterbrochenen Prozess im Hintergrund fort. Wenn keine Prozessnummer angegeben wird, gilt bg für den zuletzt (meist mit `[Strg]+[Z]`) unterbrochenen Prozess. Andernfalls muss der Prozess durch seinen Namen oder durch die bash-interne Jobnummer (nicht PID) angegeben werden.

blkid devicename

blkid liefert Informationen über die durch den Device-Namen angegebene Partition: den Dateisystemtyp, die UUID und den Namen (Label) des Dateisystems.

Beispiel

Die Partition `/dev/sda1` enthält ein ext4-Dateisystem mit der UUID 2716...19e4.

```
root# blkid /dev/sda1
/dev/sda1: UUID="27162884-8fe9-4fa9-8b5e-712ab82d19e4" TYPE="ext4"
```

break [n]

(bash)

break bricht in bash-Scripts eine for-, while- oder until-Schleife vorzeitig ab. Das Shell-Programm wird beim nächsten Kommando nach dem Schleifenende fortgesetzt. Durch die Angabe eines optionalen Zahlenwerts können *n* Schleifenebenen abgebrochen werden.

bunzip2 datei.bz2

bunzip2 dekomprimiert eine zuvor mit bzip2 komprimierte Datei. Dabei wird automatisch die Kennung .bz2 im Dateinamen entfernt.

bzip2 datei

bzip2 komprimiert die angegebenen Dateien. Das Kommando liefert im Regelfall um 20 bis 30 Prozent kleinere Dateien als gzip (Seite 67). Allerdings ist die zum Komprimieren erforderliche Rechenzeit deutlich größer.

-c bzw. --stdout oder --to-stdout

lässt die zu (de)komprimierende Datei unverändert und leitet das Ergebnis auf die Standardausgabe (in der Regel auf den Bildschirm) um. Von dort kann es mit > in eine beliebige Datei umgeleitet werden.

-d bzw. --decompress oder --uncompress

dekomprimiert die angegebene Datei, anstatt sie zu komprimieren (entspricht bunzip2).

-1 bis -9

gibt an, wie viel Speicherplatz (RAM) der Komprimieralgorithmus nutzen darf. Die Grundeinstellung lautet -9 und liefert die besten Ergebnisse. Wenn nur wenig RAM zur Verfügung steht, sollten Sie einen kleineren Wert wählen; allerdings wird dann auch die Komprimierung etwas schlechter.

```
case ausdruck in (bash)
  muster1 ) kommandos;;
  muster2 ) kommandos;;
  ...
esac
```

case bildet in bash-Scripts Mehrfachverzweigungen, wobei als Kriterium für die Verzweigung eine Zeichenkette angegeben wird (zumeist eine Variable oder ein Parameter, der dem Shell-Programm übergeben wird). Diese Zeichenkette wird der Reihe nach mit den Mustern verglichen, wobei in diesen Mustern die Jokerzeichen für Dateinamen (*?[]) verwendet werden können. In einem case-Zweig können auch mehrere durch | getrennte Muster angegeben werden. Sobald ein Muster zutrifft, werden die Kommandos ausgeführt, die zwischen der runden Klammer) und den beiden Strichpunkten folgen. Anschließend wird das Programm nach esac fortgesetzt.

```
cat dateien
```

cat zeigt den Inhalt der angegebenen Textdatei an. Bei längeren Texten sollten Sie statt cat das Kommando less verwenden; damit können Sie zeilen- bzw. seitenweise durch den Text blättern. cat wird häufig auch dazu verwendet, mehrere Dateien zu einer größeren Datei zusammzusetzen. Dazu muss die Standardausgabe mit > in eine Datei umgeleitet werden (siehe Beispiel). Zu cat existiert auch die Variante tac. Dieses Kommando gibt die Zeilen der Textdatei in umgekehrter Reihenfolge aus (die letzte Zeile zuerst).

- s reduziert mehrere leere Zeilen auf eine einzige leere Zeile.
- T zeigt Tabulatorzeichen in der Form ^I an.
- v zeigt nicht druckbare Zeichen in der ^xxx-Schreibweise an.

Beispiele

Das folgende Kommando setzt die Einzeldateien teil1.tex, teil2.tex etc. zu einer Gesamtdatei total.tex zusammen. Die Einzeldateien werden in alphabetischer Reihenfolge verarbeitet.

```
user$ cat teil*.tex > total.tex
```

Im folgenden Kommando wird die Standardeingabe in eine neue Datei umgeleitet. Nachdem Sie das Kommando mit `[↵]` bestätigt haben, werden alle weiteren Eingaben in die neue Datei geschrieben. `[Strg]+[D]` beendet die Eingabe. In dieser Form kann `cat` also dazu verwendet werden, um ohne einen Editor eine neue Textdatei zu erzeugen.

```
root# cat > neuedatei
Zeile 1
Zeile 2 <Strg>+<D>
```

cd [verzeichnis]

(bash)

`cd` wechselt in das angegebene (Unter-)Verzeichnis. Wenn kein Verzeichnis angegeben wird, wechselt `cd` in das Heimatverzeichnis. Wenn als Verzeichnis `-` angegeben wird, wechselt `cd` in das zuletzt gültige Verzeichnis. `pwd` zeigt den Pfad des gerade aktuellen Verzeichnisses an.

cdrdao kommando [optionen] toc-datei

`cdrdao` schreibt eine CD im Disk-at-once-Modus (DAO). In der Praxis besteht die gebräuchlichste Anwendung von `cdrdao` darin, Audio-CDs zu kopieren. Das erste `cdrdao`-Kommando erzeugt die Dateien `data.bin` (Inhalt der CD) und `data.toc` (Inhaltsverzeichnis). Das zweite Kommando schreibt diese Daten auf eine CD.

```
user$ cdrdao read-cd --device /dev/sg0 data.toc
user$ cdrdao write --device /dev/sg0 --buffers 64 data.toc
```

chac1 [optionen] dateien

`chac1` ermittelt bzw. verändert die erweiterten Zugriffsrechte der angegebenen Dateien bzw. Verzeichnisse. Das funktioniert nur, wenn das Dateisystem ACLs (*Access Control Lists*) unterstützt. Bei `ext3/ext4`-Dateisystemen muss dazu die `mount-Option` `ac1` verwendet werden.

Statt `chac1` sollten Sie nach Möglichkeit `getfac1` bzw. `setfac1` einsetzen. `chac1` steht nur aus Kompatibilitätsgründen zu IRIX SGI zur Verfügung.

chattr [optionen] +=[ASacDdIijsTtu] dateien

In den Dateisystemen `ext2`, `ext3` und `ext4` können mit jeder Datei neben den Benutzerinformationen (siehe `chmod` und `chown`) einige zusätzliche Attribute gespeichert werden. Diese Attribute enthalten beispielsweise Informationen über den Journaling-Status oder über eine eventuelle Komprimierung der Datei. Eine kurze Beschreibung der Attribute gibt man `chattr`. Allerdings werden momentan nur wenige der vorgesehenen Attribute tatsächlich genutzt.

chgrp [optionen] gruppe dateien

chgrp ändert die Gruppenzugehörigkeit von Dateien. Der Besitzer einer Datei kann diese Datei nur seinen eigenen Gruppen zuordnen. root kann beliebige Zuordnungen treffen.

-R bzw. --recursive

verändert auch die Gruppenzuordnung von Dateien in allen Unterverzeichnissen. Die Option ist nur dann sinnvoll, wenn die Dateien durch Jokerzeichen beschrieben werden (etwa *.tex).

chkconfig optionen

chkconfig ist ein Red-Hat- bzw. Fedora-spezifisches Kommando zum Einrichten bzw. Löschen von Init-V-Runlevel-Links. Sofern xined installiert ist, können auch dessen Dienste mit chkconfig gesteuert werden. Das Kommando steht aus Kompatibilitätsgründen vereinzelt auch für andere Distributionen in gleicher oder ähnlicher Syntax zur Verfügung. (Werfen Sie aber unbedingt einen Blick in die man-Seiten!)

--list

liefert eine Liste aller installierten Init-V-Scripts sowie Informationen darüber, in welchen Runleveln die Scripts gestartet werden.

--add name

richtet Links auf das Init-V-Script in den dafür vorgesehenen Standard-Runleveln ein. (Diese Information stammt aus den Kommentarzeilen am Beginn des Scripts.) Die neuen Links werden erst beim nächsten Runlevel-Wechsel bzw. Neustart berücksichtigt. Das Init-V-Script wird also nicht gestartet.

--del name

entfernt alle Init-V-Links für das Script.

--level n name on|off

richtet Init-V-Links nur für die angegebenen Runlevel ein bzw. entfernt sie.

Beispiel

Das folgende Kommando richtet Start-Links für den Webserver Apache (Dämon httpd) in den Runleveln 3 und 5 ein:

```
root# chkconfig --level 35 httpd on
```

Dabei werden diese zwei Links erzeugt:

```
/etc/rc3.d/S85httpd -> ../init.d/httpd  
/etc/rc5.d/S85httpd -> ../init.d/httpd
```

chmod [optionen] änderungen dateien

chmod ändert die neun Zugriffsbits von Dateien. Zusammen mit jeder Datei wird gespeichert, ob der Besitzer (*user*), die Gruppenmitglieder (*group*) und andere Benutzer (*others*) die Datei lesen, schreiben und ausführen dürfen. Die Änderung der Zugriffsbits erfolgt durch die Zeichenkombination *Gruppe +/- Zugriffstyp*, also beispielsweise *g+w*, um allen Gruppenmitgliedern eine Schreiberlaubnis zu geben. Die Gruppe geben Sie durch *u* (*user*), *g* (*group*), *o* (*others*) oder *a* (*all*) an, den Zugriffstyp durch *r* (*read*), *w* (*write*) oder *x* (*execute*).

Setuid, setgid und sticky-Bits

Das *setuid*-Bit (oft auch *suid*-Bit genannt) bewirkt, dass Programme so ausgeführt werden, als hätte der Besitzer selbst das Programm gestartet. Wenn der Besitzer eines Programms *root* ist, dann kann jeder das Programm ausführen, als wäre er selbst *root*.

Das *setgid*-Bit hat bei Programmen dieselbe Funktion wie *setuid*, aber eben für die Gruppenzugehörigkeit. Bei Verzeichnissen bewirkt das *setgid*-Bit, dass in diesem Verzeichnis neu erzeugte Dateien der Gruppe des Verzeichnisses angehören (und nicht, wie sonst üblich, der Gruppe des Benutzers, der die Datei erzeugt).

Das *sticky*-Bit bewirkt bei Verzeichnissen, in denen alle die Dateien ändern dürfen, dass jeder nur seine eigenen Dateien löschen darf (und nicht auch Dateien anderer Benutzer). Das Bit ist beispielsweise beim */tmp*-Verzeichnis gesetzt. In diesem Verzeichnis darf jeder Benutzer temporäre Dateien anlegen. Es muss aber vermieden werden, dass auch jeder Benutzer nach Belieben fremde Dateien umbenennen oder löschen kann.

Um mit *chmod* die Spezialbits *setuid*, *setgid* und *sticky* zu setzen, sind die folgenden Zeichenkombinationen vorgesehen:

```
setuid: u+s
setgid: g+s
sticky: +t
```

Damit *setuid* wirkt, muss auch das *x*-Bit für den Besitzer gesetzt sein (*u+x*).

Damit *setgid* wirkt, muss auch das *x*-Bit für die Gruppe gesetzt sein (*g+x*).

Oktale Schreibweise

Statt mit Buchstaben kann der Zugriffstyp auch durch eine maximal vierstellige Oktalzahl angegeben werden. Bei den Zugriffsbits ist *u*, *g* und *o* jeweils eine Ziffer zugeordnet. Jede Ziffer ist aus den Werten 4, 2 und 1 für *r*, *w* und *x* zusammengesetzt. 660 bedeutet daher *rw-rw----*, 777 steht für *rxwxrwx*. Die drei Spezialbits *setuid*, *setgid* und *sticky* haben die Oktalwerte 4000, 2000 und 1000.

-R bzw. *--recursive*

verändert auch die Zugriffsrechte von Dateien in allen Unterverzeichnissen.

Beispiele

Die Datei `sichere` kann nun von allen Benutzern ausgeführt werden. `sichere` kann etwa ein Shell-Script zur Erstellung eines Backups sein.

```
user$ chmod a+rx sichere
```

Das folgende Kommando entzieht allen Benutzern außerhalb der eigenen Gruppe die Lese- und Schreiberlaubnis für alle `*.doc`-Dateien im aktuellen Verzeichnis:

```
user$ chmod o-rw *.doc
```

```
chown [optionen] user[:gruppe] dateien
```

`chown` ändert den Besitzer und (optional) auch die Gruppenzugehörigkeit einer Datei. Der Besitzer einer Datei kann nur von `root` verändert werden, während die Gruppe auch von anderen Benutzern eingestellt werden kann (siehe `chgrp`).

`-R` bzw. `--recursive`

verändert auch die Gruppenzuordnung von Dateien in allen Unterverzeichnissen.

Beispiel

Das folgende Kommando stellt sicher, dass alle Dateien innerhalb von `/var/www` dem Benutzer und der Gruppe `www-data` zugeordnet sind. (Bei Debian- und Ubuntu-Systemen läuft Apache unter dem Account `www-data`.)

```
root# chown -R www-data:www-data /var/www
```

```
chroot verzeichnis [kommando]
```

Ohne weitere Parameter startet `chroot` eine neue Shell, die das angegebene Verzeichnis als Wurzelverzeichnis `/` verwendet. In dieser Shell können Sie interaktiv arbeiten. `exit` führt zurück in die ursprüngliche Shell.

Wenn Sie optional ein Kommando angeben, wird dieses Kommando statt der Shell gestartet. Während der Ausführung des Kommandos gilt abermals das angegebene Verzeichnis als Wurzelverzeichnis.

```
chsh [user] shell
```

`chsh` verändert die Standard-Shell, die automatisch nach dem Einloggen aufgerufen wird. Zur Auswahl stehen alle in `/etc/shells` eingetragenen Shells, normalerweise `/bin/bash`, `/bin/csh` und `/bin/ksh`. Das Kommando `chsh` verändert die Datei `/etc/passwd` und trägt dort die neue Shell ein. Die Shell eines anderen Anwenders kann nur von `root` verändert werden

(während jeder Anwender seine eigene Shell nach Belieben verändern kann). Die neue Shell muss mit dem vollständigen Verzeichnis angegeben werden.

cksum datei

cksum ermittelt die Prüfsumme und die Länge der Datei in Bytes. Die Prüfsumme kann verwendet werden, um rasch festzustellen, ob zwei Dateien identisch sind. cksum liefert zuverlässigere Ergebnisse als das verwandte Kommando sum. Mathematisch noch sicherer sind md5sum oder sha512sum.

clear

clear bzw. `[Strg]+[L]` löschen den Inhalt der Konsole.

cmp [optionen] datei1 datei2

cmp vergleicht zwei Dateien Byte für Byte und liefert die Position der ersten Abweichung. Wenn die Dateien identisch sind, zeigt das Kommando überhaupt keine Meldung an (siehe auch diff auf Seite 39).

-c bzw. --show-chars

zeigt das jeweils erste Textzeichen an, bei dem sich die Dateien voneinander unterscheiden.

-l bzw. --verbose

liefert eine Liste aller Abweichungen.

compress [optionen] datei

compress komprimiert bzw. dekomprimiert die angegebene Datei. Bei komprimierten Dateien wird die Kennung .Z an den Dateinamen angehängt. compress existiert nur noch aus Kompatibilitätsgründen. Wesentlich leistungsfähiger sind bzip2 (siehe Seite 23) und gzip (Seite 67).

continue [n]

(bash)

continue überspringt in bash-Scripts den Körper einer for-, while- oder until-Schleife und setzt die Schleife mit dem nächsten Durchlauf fort. Durch den optionalen Zahlenwert kann dieser Vorgang auch für äußere Schleifenebenen durchgeführt werden.

convert [optionen] bildalt bildneu

convert aus dem Image-Magick-Paket konvertiert Bilddateien von einem Format in ein anderes. In der einfachsten Form wird es in der Art `convert name.tif name.jpg` aufgerufen, um die angegebene TIF-Datei in eine JPEG-Datei zu konvertieren. Die ursprüngliche Datei bleibt

dabei erhalten. Mit über 100 Optionen können gleichzeitig diverse Bildparameter verändert werden. Die folgende Liste ist daher nur eine Auswahl.

- blur *radius*
verwischt das Bild.
- colors *n*
reduziert die Anzahl der RGB-Farben auf *n*.
- colorspace CMYK|GRAY|RGB|Transparent|YUV
gibt das gewünschte Farbmodell an (wobei zahlreiche weitere Modelle zur Auswahl stehen).
- compress None|BZip|Fax|Group4|JPEG|JPEG2000|Lossless|LZW|RLE|Zip
gibt das gewünschte Kompressionsformat an. Welche Formate tatsächlich zur Auswahl stehen, hängt allerdings vom Bildformat ab.
- contrast bzw. +contrast
verringert bzw. vergrößert den Kontrast des Bilds.
- crop *geometry*
schneidet den gewünschten Teil des Bilds aus. Beispielsweise beschreibt -crop 50x50+100+100 ein 50 mal 50 Pixel großes Gebiet, das an der Koordinatenposition (100, 100) beginnt.
- filter Point|Box|Triangle|Hermite|Hanning|Hamming ...
wendet den gewünschten Filter auf das Bild an.
- gaussian *radius*
verwischt den Filter mit dem Gauß-Operator.
- normalize
normalisiert die Farbverteilung im Bild.
- quality *n*
gibt die gewünschte Kompressionsqualität an. Die zulässigen Werte für *n* hängen vom Bildformat ab (z. B. 0 bis 100 bei JPEG).
- resize *NxN* bzw. -resize *n%*
verändert die Auflösung des Bildes.
- rotate *winkel*
dreht das Bild im Uhrzeigersinn um den angegebenen Winkel in Grad.
- trim
schneidet einfarbige Bildränder ab, wenn sie dieselbe Farbe wie die Eckpunkte des Bilds haben.