

Michael Kofler

# Linux Kommandoreferenz

Shell-Befehle von A bis Z



- ▶ Über 500 Befehle und Konfigurationsdateien immer griffbereit
- ▶ Schnell thematisch und alphabetisch nachschlagen
- ▶ Inkl. Shortcuts, Shell-Werkzeuge und Anwendungsbeispiele

5., aktualisierte Auflage



Rheinwerk  
Computing

Michael Kofler

# **Linux Kommandoreferenz**

Shell-Befehle von A bis Z

5., aktualisierte Auflage 2021



## **Impressum**

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

**Lektorat** Christoph Meister

**Korrektorat** Friederike Daenecke, Zülpich

**Covergestaltung** Eva Schmücker

**Herstellung E-Book** Norbert Englert

**Satz E-Book** Michael Kofler

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

**ISBN 978-3-8362-7860-7**

5., aktualisierte Auflage 2021

© Rheinwerk Verlag GmbH, Bonn 2021

## Liebe Leserin, lieber Leser,

dass sich Linux über die Shell mit wenig Übung äußerst schnell und effizient steuern, werden Sie wahrscheinlich bereits wissen. Über die Kommandozeile und Shellskripte automatisieren und strukturieren Sie Abläufe viel einfacher als mit wildem Herumgeklicke auf grafischen Benutzeroberflächen. Kurz: Wenn Sie den Umgang mit Shell-Kommandos beherrschen, dann liegt Ihnen Linux zu Füßen. Die Menge der Möglichkeiten hat natürlich eine Kehrseite: nämlich die Fülle der damit verbundenen Informationen. Es gibt sehr viele Linux-Kommandos und eine schier unüberschaubare Anzahl von Optionen – ein Blick in die man-Seiten zu den einzelnen Kommandos spricht Bände ...

Und das bringt uns zu dieser Kommandoreferenz: Sie begrenzt und ordnet gezielt die Fülle der Informationen und ermöglicht Ihnen so den schnellen Zugriff auf die wichtigsten Kommandos und häufig benötigte Optionen. Darüber hinaus gibt sie Ihnen geprüfte und sofort einsetzbare Anwendungsbeispiele an die Hand, weist auf distributionsspezifische Besonderheiten hin und bietet Ihnen neben einer Übersicht wichtiger Tastenkürzel auch eine Syntaxreferenz zentraler Linux-Konfigurationsdateien.

In dieser Auflage hat Michael Kofler, dessen Linux-Standardwerk Sie vermutlich kennen, zahlreiche weitere Kommandos für Sie beschrieben. Das betrifft besonders Informationen zur AWS-Administration oder dem Einsatz des Windows Subsystem for Linux. Denn wie mächtig die Linux-Shells sind, hat inzwischen sogar Microsoft erkannt und sie in aktuelle Windows-Versionen integriert. Noch mehr Gründe also, dieses Buch immer in Reichweite Ihres Linux-Rechners zu platzieren!

Noch ein Wort in eigener Sache: Dieses Werk wurde mit großer Sorgfalt geschrieben, geprüft und produziert. Sollte dennoch einmal etwas nicht so funktionieren, wie Sie es erwarten, freue ich mich,

wenn Sie sich mit mir in Verbindung setzen. Ihre Kritik und konstruktiven Anregungen sind uns jederzeit herzlich willkommen!

**Ihr Christoph Meister**

Lektorat Rheinwerk Computing

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

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

Rheinwerk Verlag • Rheinwerkallee 4 • 53227 Bonn

# Thematische Kommandoübersicht

## Dateien verwalten

---

cat	verbindet mehrere Dateien zu einer Gesamtdatei
cd	wechselt in ein anderes Verzeichnis
chgrp	ändert die Gruppenzugehörigkeit einer Datei
chmod	ändert die Zugriffsbits einer Datei
chown	ändert den Besitzer einer Datei
cp	kopiert Dateien
dircolors	hilft bei der Konfiguration der ls-Farben
file	versucht, den Typ einer Datei festzustellen
inotifywait	wartet auf die Veränderung von Dateiattributen
j	wechselt in ein anderes Verzeichnis
ln	stellt feste und symbolische Links zu Dateien her
ls	zeigt das Inhaltsverzeichnis an
mkdir	erzeugt ein neues Verzeichnis
mv	verschiebt Dateien bzw. ändert ihre Namen
namei	zeigt die Zugriffsrechte aller Verzeichnisse einer Datei
rdfind	sucht und beseitigt Doppelgänger
rename	gibt mehreren Dateien neue Namen
rm	löscht Dateien
rmdir	löscht Verzeichnisse
rsync	synchronisiert Verzeichnisse
stat	liefert detaillierte Informationen zu Dateien
tee	dupliziert die Standardeingabe
touch	erzeugt eine leere Datei bzw. ändert den Modify-Zeitpunkt
tree	stellt die Verzeichnishierarchie dar
truncate	verkleinert/vergrößert eine Datei
umask	steuert, welche Zugriffsrechte neue Dateien und Verzeichnisse erhalten

## Dateien suchen

---

ack	schnelle grep-Alternative für Programmierer
egrep	grep-Variante mit erweiterter Syntax
find	sucht Dateien nach Name, Datum, Größe etc.
grep	sucht Text in einer Textdatei
locate	sucht Dateien in einer dafür vorbereiteten Datenbank
updatedb	aktualisiert die Suchdatenbank für locate
whereis	sucht Dateien in vordefinierten Verzeichnissen
which	durchsucht die <i>PATH</i> -Verzeichnisse nach Kommandos

## Dateien komprimieren und archivieren

---

bunzip2	dekomprimiert *.bz2-Dateien
bzip2	komprimiert Dateien; leistungsfähiger als gzip
compress	komprimiert Dateien
cpio	überträgt Archivdateien zwischen Dateisystemen
gunzip	dekomprimiert *.gz-Dateien
gzip	komprimiert Dateien; leistungsfähiger als compress
lzop	komprimiert Dateien; deutlich schneller als gzip
tar	bildet ein *.tar-Archiv
uncompress	dekomprimiert durch compress komprimierte Dateien
unxz	dekomprimiert *.xz-Dateien
unzip	dekomprimiert *.zip-Dateien
xz	komprimiert Dateien; leistungsfähiger als bzip2
zip	erzeugt ein Windows-kompatibles ZIP-Archiv
zipinfo	zeigt den Inhalt eines ZIP-Archivs an

## Textdateien verarbeiten

---

awk	Programmiersprache zur Textauswertung
cat	gibt eine Datei aus bzw. vereint mehrere Texte
column	formatiert Informationen spaltenweise

<code>csplit</code>	zerlegt den Text an vorgegebenen Stellen in Einzeldateien
<code>cut</code>	extrahiert Spalten aus jeder Zeile des Textes
<code>diff</code>	vergleicht zwei Texte
<code>expand</code>	ersetzt Tabulator- durch Leerzeichen
<code>fold</code>	zerlegt lange Textzeilen in kürzere
<code>grep</code>	sucht Texte innerhalb der Datei
<code>head</code>	gibt die ersten Zeilen der Datei aus
<code>iconv</code>	ändert den Zeichensatz von Textdateien
<code>less</code>	zeigt Textdateien seitenweise an (auch rückwärts)
<code>more</code>	zeigt Textdateien seitenweise an
<code>multitail</code>	verfolgt die Änderungen mehrerer Dateien
<code>paste</code>	vereint mehrere Texte zeilenweise
<code>patch</code>	ändert Textdateien gemäß einer <code>diff</code> -Datei
<code>recode</code>	konvertiert zwischen verschiedenen Zeichensätzen
<code>sed</code>	Stream-Editor (programmierbarer Editor)
<code>sort</code>	sortiert Dateien
<code>split</code>	zerlegt eine Datei in Teildateien mit vorgegebener Größe
<code>strings</code>	extrahiert Zeichenketten aus Binärdateien
<code>tac</code>	gibt Text in umgekehrter Reihenfolge aus, also die letzte Zeile zuerst
<code>tail</code>	gibt das Ende einer Datei aus
<code>tr</code>	ersetzt vorgegebene Zeichen durch andere Zeichen
<code>unexpand</code>	ersetzt Leerzeichen durch Tabulatorzeichen
<code>uniq</code>	eliminiert mehrfach auftretende Zeilen in einer Textdatei
<code>zcat</code>	gibt eine komprimierte Textdatei aus
<code>zless</code>	<code>zcat</code> zeigt eine komprimierte Textdatei an (auch rückwärts)

**zmore** zcat zeigt eine komprimierte Textdatei seitenweise an

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

---

<b>attr</b>	verwaltet die Zusatzattribute einer Datei
<b>chacl</b>	verwaltet die ACLs einer Datei
<b>getcap</b>	ermittelt die Capabilities einer Datei
<b>getfacl</b>	ermittelt die ACLs einer Datei
<b>getfattr</b>	ermittelt die Zusatzattribute einer Datei
<b>setcap</b>	verändert die Capabilities einer Datei
<b>setfacl</b>	verändert die ACLs einer Datei
<b>setfattr</b>	verändert die Zusatzattribute einer Datei

## Konverter

---

<b>avconv</b>	konvertiert Videodateien
<b>convert</b>	konvertiert Grafikdateien
<b>dvips</b>	wandelt eine DVI-Datei in das PostScript-Format um
<b>enscript</b>	konvertiert Textdateien in das PostScript-Format
<b>epstopdf</b>	konvertiert EPS-Dateien in PDF-Dateien
<b>exiftool</b>	liest bzw. ändert EXIF-Daten in JPG-Dateien
<b>ffmpeg</b>	konvertiert Videodateien
<b>iconv</b>	ändert den Zeichensatz von Textdateien
<b>lame</b>	erzeugt MP3-kompatible Audiodateien
<b>mogrify</b>	verändert Parameter einer Bilddatei
<b>mpage</b>	konvertiert Textdateien in das PostScript-Format
<b>pandoc</b>	erzeugt Dokumente aus Markdown-Dateien
<b>paps</b>	konvertiert UTF-8-Textdateien in das PostScript-Format
<b>pdf2ps</b>	konvertiert PDF-Dateien in PostScript-Dateien
<b>pdfimages</b>	extrahiert Bilder aus PDF-Dateien
<b>pdftk</b>	manipuliert PDF-Dateien

<code>pdftops</code>	Alternative zu <code>pdf2ps</code>
<code>pdftotext</code>	wandelt ein PDF-Dokument in eine reine Textdatei um
<code>ps2pdf</code>	konvertiert PostScript-Dateien in PDF-Dateien
<code>recode</code>	ändert den Zeichensatz von Textdateien

## Prozesse verwalten

---

<code>at</code>	führt einen Job zu einem vordefinierten Zeitpunkt aus
<code>atq</code>	listet Jobs auf, die später ausgeführt werden sollen
<code>atrm</code>	löscht einen Job, der später ausgeführt werden soll
<code>batch</code>	führt einen Job aus, sobald das System im Leerlauf läuft
<code>bg</code>	setzt einen Prozess im Hintergrund fort
<code>chroot</code>	startet eine Shell in einem veränderten Wurzelverzeichnis
<code>crontab</code>	hilft bei der Administration eigener Crontab-Einträge
<code>disown</code>	löst einen Prozess von der Shell
<code>fg</code>	setzt einen Prozess im Vordergrund fort
<code>fuser</code>	ermittelt das Programm, das auf eine Datei zugreift
<code>glances</code>	moderne Alternative zu <code>top</code>
<code>halt</code>	beendet Linux und hält den Rechner an
<code>history</code>	zeigt die zuletzt in der <code>bash</code> ausgeführten Kommandos an
<code>htop</code>	zeigt alle fünf Sekunden eine Liste aller Prozesse an
<code>iftop</code>	beobachtet die Netzwerkaktivität
<code>ionice</code>	steuert die I/O-Priorität eines Prozesses
<code>iostop</code>	zeigt die Prozesse mit der größten IO-Aktivität an

<code>kill</code>	versendet Signale (meist zum Beenden von Prozessen)
<code>killall</code>	wie <code>kill</code> ; der Prozess wird mit Namen genannt
<code>ldconfig</code>	aktualisiert die Cache-Datei zur Bibliothekssuche
<code>ldd</code>	liefert alle erforderlichen Bibliotheken eines Programms
<code>lsof</code>	listet offene Dateien und die zugeordneten Prozesse auf
<code>nice</code>	startet ein Programm mit verringriger Priorität
<code>nohup</code>	startet einen Prozess, der von der Shell losgelöst ist
<code>parallel</code>	startet mehrere gleichartige Jobs parallel
<code>pidof</code>	ermittelt die Prozessnummer eines Programms
<code>powertop</code>	analysiert den Energieverbrauch der laufenden Prozesse
<code>ps</code>	zeigt die Liste der laufenden Prozesse an
<code>pstree</code>	wie <code>ps</code> ; macht die Abhängigkeiten besser sichtbar
<code>reboot</code>	beendet Linux und startet den Rechner neu
<code>renice</code>	verändert die Priorität eines laufenden Prozesses
<code>sudo</code>	führt ein Programm als root aus
<code>timeout</code>	limitiert die Ausführungszeit eines Kommandos
<code>top</code>	zeigt alle fünf Sekunden eine Liste aller Prozesse an
<code>uptime</code>	zeigt an, wie lange der Rechner schon läuft
<code>watch</code>	führt ein Kommando periodisch aus und zeigt die Ausgaben an

## Benutzer und Gruppen verwalten

---

<code>addgroup</code>	richtet eine neue Gruppe ein (Debian/Ubuntu)
<code>adduser</code>	richtet einen neuen Benutzer ein (Debian/Ubuntu)

<code>chage</code>	ändert das Ablaufdatum eines Kontos oder Passworts
<code>chgrp</code>	ändert die Gruppenzugehörigkeit einer Datei
<code>chown</code>	ändert den Besitzer einer Datei
<code>chpasswd</code>	verändert ein Benutzerpasswort ohne Interaktion
<code>chsh</code>	verändert die Default-Shell eines Benutzers
<code>delgroup</code>	löscht eine Gruppe (Debian/Ubuntu)
<code>deluser</code>	löscht einen Benutzer (Debian/Ubuntu)
<code>groupadd</code>	richtet eine neue Gruppe ein
<code>groupdel</code>	löscht eine Gruppe
<code>groupmod</code>	verändert Gruppeneigenschaften
<code>groups</code>	zeigt die Gruppen des aktuellen Benutzers an
<code>gpasswd</code>	ändert Gruppenpasswörter
<code>id</code>	zeigt die aktuelle Benutzer- und Gruppen-ID-Nummer an
<code>last</code>	verrät, wer zuletzt auf diesem Rechner eingeloggt war
<code>lastb</code>	listet auf, welche Login-Versuche zuletzt gescheitert sind
<code>makepasswd</code>	erzeugt ein neues, zufälliges Passwort
<code>mkpasswd</code>	erzeugt ein neues, zufälliges Passwort
<code>newgrp</code>	ändert die aktive Gruppe eines Benutzers
<code>newusers</code>	richtet mehrere neue Benutzer ein
<code>passwd</code>	verändert das Passwort eines Benutzers
<code>pwgen</code>	generiert leicht merkbare Passwörter
<code>useradd</code>	richtet einen neuen Benutzer ein
<code>userdel</code>	löscht einen Benutzer
<code>usermod</code>	verändert Benutzereigenschaften
<code>vigr</code>	<code>vipw</code> bearbeitet <i>/etc/groups</i>
<code>vipw</code>	bearbeitet <i>/etc/passwd</i>
<code>visudo</code>	<code>vipw</code> bearbeitet <i>/etc/sudoers</i>

`who` liefert Informationen über die eingeloggten Benutzer

## Dateisystem administrieren

---

<code>badblocks</code>	testet, ob Datenträger defekte Sektoren enthalten
<code>blkid</code>	liefert die UUID und den Namen eines Dateisystems
<code>btrfs</code>	administriert ein <code>btrfs</code> -Dateisystem
<code>cfdisk</code>	partitioniert eine Festplatte
<code>cryptsetup</code>	richtet ein verschlüsseltes Device ein
<code>dd</code>	kopiert Datenblöcke zwischen Devices
<code>df</code>	zeigt den freien Speicher auf der Festplatte an
<code>du</code>	ermittelt den Platzbedarf eines Verzeichnisses
<code>dumpe2fs</code>	zeigt interne Informationen über ein ext-Dateisystem an
<code>e4defrag</code>	defragmentiert Dateien eines ext4-Dateisystems
<code>exfatlabel</code>	ändert den Namen eines exFAT-Dateisystems
<code>findmnt</code>	liefert eine Liste aller aktiven Dateisysteme
<code>fstrim</code>	meldet der SSD alle freien Datenblöcke
<code>kpartx</code>	erzeugt oder löscht Device-Dateien für virtuelle Datenträger
<code>lsblk</code>	listet alle Block-Devices auf
<code>mdadm</code>	verwaltet RAID-Partitionen
<code>mkfifo</code>	erzeugt eine FIFO-Datei (eine benannte Pipe)
<code>mkfs</code>	richtet ein Dateisystem ein
<code>mknod</code>	erstellt Device-Dateien
<code>mkswap</code>	richtet eine Datei oder eine Partition als Swap-Bereich ein
<code>mount</code>	bindet ein Dateisystem in den Verzeichnisbaum ein
<code>ncdu</code>	interaktive, komfortable <code>du</code> -Variante
<code>parted</code>	partitioniert eine Festplatte

partprobe	informiert den Kernel über die geänderte Partitionierung
partx	liest Partitionen bzw. verändert die Partitionstabelle des Kernels
resize2fs	verändert die Größe eines ext-Dateisystems
sfdisk	partitioniert eine Festplatte mit MBR-Partitionstabelle
sgdisk	partitioniert eine Festplatte mit GUID-Partitionstabelle
smartctl	steuert die SMART-Funktionen der Festplatte
snapper	verwaltet btrfs-Snapshots (SUSE)
swapoff	deaktiviert eine Swap-Datei oder -Partition
swapon	aktiviert eine Swap-Datei oder -Partition
sync	führt alle gepufferten Schreiboperationen aus
tune2fs	verändert Systemparameter eines ext-Dateisystems
umount	löst ein Dateisystem aus dem Verzeichnisbaum
xfs_admin	xfsadmin verändert Parameter eines XFS-Dateisystems
xfs_growfs	xfsgrowfs vergrößert ein XFS-Dateisystem
xfs_info	xfsinfo zeigt die Eckdaten eines XFS-Dateisystems
xfs_repair	xfsrepair repariert ein defektes XFS-Dateisystem

## Logical Volume Manager (LVM)

---

lvcreate	richtet ein neues LV (Logical Volume) ein
lvdisplay	liefert Detailinformationen zu einem LV
lvextend	vergrößert ein LV
lvm	LVM-Basiskommando
lvreduce	verkleinert ein LV
lvremove	löscht ein LV
lvrename	gibt dem LV einen neuen Namen
lvscan	listet alle LVs auf

<code>pvcreate</code>	kennzeichnet eine Partition als PV (Physical Volume)
<code>pvdisplay</code>	liefert Detailinformationen zu einem PV
<code>pvremove</code>	entfernt die PV-Kennzeichnung eines ungenutzten PVs
<code>pvscan</code>	listet alle PVs auf
<code>vgchange</code>	ändert die Attribute einer VG (Volume Group)
<code>vgcreate</code>	erzeugt eine neue VG aus einem oder mehreren PVs
<code>vgdisplay</code>	liefert Detailinformationen zu einer VG
<code>vgextend</code>	vergrößert eine VG um ein PV
<code>vgmerge</code>	vereint zwei VGs
<code>vgreduce</code>	verkleinert eine VG um ein ungenutztes PV
<code>vgrename</code>	gibt einer VG einen neuen Namen
<code>vgscan</code>	listet alle VGs auf

## SELinux und AppArmor

---

<code>aa-complain</code>	protokolliert AppArmor-Regelverstöße, ohne sie zu ahnden
<code>aa-disable</code>	deaktiviert ein AppArmor-Regelprofil
<code>aa-enforce</code>	stellt die Einhaltung von AppArmor-Regeln sicher
<code>aa-status</code>	ermittelt den Zustand des AppArmor-Systems
<code>chcon</code>	verändert den SELinux-Kontext von Dateien
<code>getenforce</code>	ermittelt den SELinux-Modus ( <code>Enforcing</code> oder <code>Permissive</code> )
<code>restorecon</code>	stellt den Default-SELinux-Kontext wieder her
<code>sealert</code>	hilft bei der Analyse von SELinux-Regelverstößen
<code>sestatus</code>	ermittelt den Zustand des SELinux-Systems
<code>setenforce</code>	ändert den SELinux-Modus zwischen <code>Enforcing</code> und <code>Permissive</code>
<code>setsebool</code>	verändert boolesche Parameter der SELinux-Regeln

## Paketverwaltung

---

apk	verwaltet Pakete in Alpine Linux
apt	hilft bei der DEB-Paketverwaltung (Debian, Ubuntu)
add-apt-repository	richtet eine PPA-Paketquelle ein (Ubuntu)
alien	wandelt Pakete zwischen verschiedenen Formaten um
alternatives	richtet Links in <i>/etc/alternatives</i> ein (Fedora, Red Hat)
apt-cache	liefert Informationen über installierte/verfügbare Pakete
apt-get	hilft bei der DEB-Paketverwaltung (Debian, Ubuntu)
apt-key	richtet einen Schlüssel für eine APT-Paketquelle ein
aptitude	hilft bei der DEB-Paketverwaltung
cnf	verrät, welches Paket ein Kommando enthält (SUSE)
dnf	Alternative zu yum (Fedora)
dpkg	(de)installiert oder aktualisiert DEB-Pakete
flatpak	verwaltet Flatpak-Pakete
gdebi	installiert eine lokale Paketdatei (Debian, Ubuntu)
pip	verwaltet Python-Pakete
pkcon	verwaltet Pakete distributionsübergreifend (PackageKit)
ppa-purge	deaktiviert eine PPA-Paketquelle (Ubuntu)
repoquery	liefert Metadaten aus YUM-Paketquellen
rpm	(de)installiert oder aktualisiert RPM-Pakete
rpm2archive	wandelt ein Paket in ein TAR-Archiv um
rpm2cpio	wandelt ein Paket in ein CPIO-Archiv um
snap	verwaltet Snap-Pakete (Ubuntu)

<code>tasksel</code>	(de)installiert DEB-Paketgruppen
<code>ubuntu-security-status</code>	gibt den Support-Zeitraum der installierten Pakete an
<code>update-alternatives</code>	richtet Links in <i>/etc/alternatives</i> ein
<code>yum</code>	hilft bei der RPM-Paketverwaltung (Fedora, Red Hat)
<code>zypper</code>	hilft bei der RPM-Paketverwaltung (SUSE)

## Netzwerk administrieren

---

<code>cadaver</code>	überträgt interaktiv Dateien via WebDAV
<code>curl</code>	überträgt Dateien von/zu HTTP-, FTP- und SSH-Servern
<code>dhclient</code>	führt eine DHCP-Netzwerkkonfiguration durch
<code>dig</code>	führt DNS-Abfragen durch
<code>etherwake</code>	aktiviert ein Wake-on-LAN-Gerät
<code>ethtool</code>	verändert die Parameter eines Ethernet-Adapters
<code>exportfs</code>	meldet die NFS-Konfiguration an den NFS-Server
<code>firewall-cmd</code>	liest bzw. ändert die Firewall-Konfiguration (RHEL/Fedora)
<code>ftp</code>	überträgt interaktiv Dateien via FTP
<code>host</code>	löst IP-Nummern bzw. Netzwerknamen auf
<code>hostname</code>	liefert bzw. verändert den Namen des lokalen Rechners
<code>hostnamectl</code>	verändert den Hostnamen bleibend
<code>ifconfig</code>	konfiguriert Netzwerkschnittstellen
<code>ifdown</code>	deaktiviert eine Netzwerkschnittstelle
<code>ifup</code>	aktiviert eine Netzwerkschnittstelle
<code>ip</code>	zeigt Netzwerkeinstellungen an bzw. verändert sie
<code>ipcalc</code>	berechnet Netzwerkbereiche und -masken
<code>iptables</code>	konfiguriert eine Netfilter-Firewall
<code>iptables-restore</code>	<code>iptables-save</code> liest mehrere Firewall-Regeln ein

<code>iptables-save</code>	gibt alle Firewall-Regeln im Textformat aus
<code>iptables-xml</code>	<code>iptables-save</code> gibt alle Firewall-Regeln als XML-Dokument aus
<code>iw</code>	steuert WLAN-Schnittstellen (aktuelles Kommando)
<code>iwconfig</code>	steuert WLAN-Schnittstellen (veraltetes Kommando)
<code>iwlist</code>	liefert Informationen zum WLAN-Controller und -Netz
<code>mtr</code>	kombiniert ping- und traceroute-Ergebnisse
<code>nft</code>	konfiguriert eine Nftables-Firewall
<code>netplan</code>	steuert andere Netzwerk-Backends (Ubuntu)
<code>netstat</code>	analysiert die Netzwerkaktivität des lokalen Rechners
<code>networkctl</code>	liefert den Netzwerkstatus (systemd)
<code>newaliases</code>	meldet Änderungen in <i>/etc/aliases</i> an den Mail-Server
<code>nmap</code>	analysiert die Netzwerkaktivität eines fremden Rechners
<code>nmcli</code>	steuert den Network Manager
<code>openssl</code>	erzeugt und administriert SSL-Schlüsseldateien
<code>ping</code>	überprüft die Netzwerkverbindung zu einem Rechner
<code>pnuke</code>	<code>pssh</code> beendet ein Programm parallel auf mehreren Hosts
<code>postconf</code>	liest bzw. verändert die Postfix-Konfiguration
<code>postqueue</code>	zeigt die Postfix-Warteschlange
<code>pscp</code>	<code>pssh</code> kopiert Dateien parallel von/auf mehrere(n) Hosts
<code>pssh</code>	führt via SSH Kommandos auf mehreren Hosts aus
<code>rdiff-backup</code>	erstellt inkrementelle Backups

<code>rfkill</code>	(de)aktiviert Bluetooth-, WLAN- und Mobilfunk-Adapter
<code>route</code>	verändert bzw. zeigt die IP-Routing-Tabelle
<code>rpcinfo</code>	liefert Informationen über RPC- und NFS-Dienste
<code>rsync</code>	synchronisiert Netzwerkverzeichnisse
<code>scp</code>	überträgt Dateien verschlüsselt via SSH
<code>sftp</code>	überträgt Dateien via SFTP
<code>showmount</code>	listet NFS-Verzeichnisse auf
<code>smbclient</code>	überträgt Dateien aus Windows-Netzwerkverzeichnissen
<code>smbtree</code>	liefert eine Liste aller Windows-Netzwerkverzeichnisse
<code>ss</code>	analysiert die Netzwerkaktivität des lokalen Rechners
<code>ssh</code>	ermöglicht Logins auf anderen Rechnern im Netzwerk
<code>ssh-copy-id</code>	überträgt einen öffentlichen Schlüssel zum SSH-Server
<code>telnet</code>	kommuniziert interaktiv mit einem Netzwerkdienst
<code>traceroute</code>	liefert die Zwischenstationen zu einer Netzwerkadresse
<code>ufw</code>	konfiguriert die Firewall (Ubuntu)
<code>wakeonlan</code>	aktiviert ein Wake-on-LAN-Gerät (Debian, Ubuntu)
<code>wget</code>	lädt Dateien oder Verzeichnisse herunter
<code>whois</code>	führt DNS-Abfragen durch
<code>wol</code>	aktiviert ein Wake-on-LAN-Gerät (Fedora, Red Hat)
<code>wpa_passphrase</code>	wpapassphrase hilft bei der WLAN-Konfiguration

<code>arp-scan</code>	sendet ARP-Pakete an alle Adressen eines Netzwerks
<code>chkrootkit</code>	sucht nach bekannten Rootkits
<code>fail2ban-client</code>	administriert Fail2ban
<code>hydra</code>	Online-Passwort-Cracker
<code>john</code>	Offline-Passwort-Cracker
<code>nc</code>	Netcat, leitet Netzwerkdaten auf die Standardeingabe bzw. -ausgabe um
<code>ngrep</code>	filtert Netzwerk-Streams mit grep (Packet Sniffing)
<code>nmap</code>	Netzwerk- und Port-Scanner
<code>rkhunter</code>	sucht nach bekannten Rootkits
<code>tcpdump</code>	filtert Netzwerk-Streams (Packet Sniffing)

## Drucker-, Datenbank- und Server-Administration

---

<code>acme.sh</code>	verwaltet Let's-Encrypt-Zertifikate
<code>certbot</code>	verwaltet Let's-Encrypt-Zertifikate
<code>htpasswd</code>	speichert Apache-Login-Daten in einer Passworddatei
<code>lpadmin</code>	richtet neue Drucker ein bzw. löscht sie wieder
<code>lpinfo</code>	listet alle Druck-Devices, Druckertreiber etc. auf
<code>lpoptions</code>	zeigt die Optionen von Druckern an bzw. verändert sie
<code>lpq</code>	zeigt den Inhalt einer Druckerwarteschlange an
<code>lpr</code>	druckt eine Datei aus
<code>lprm</code>	löscht einen Druck-Job aus der Warteschlange
<code>lpstat</code>	liefert Informationen über Drucker, Druck-Jobs etc.
<code>mysql</code>	führt SQL-Kommandos auf einem MySQL-Server aus
<code>mysqladmin</code>	hilft bei der MySQL-Administration
<code>mysqlbinlog</code>	extrahiert Daten aus binären MySQL-Logging-Dateien
<code>mysqldump</code>	führt ein MySQL-Backup durch

<code>smbpasswd</code>	ändert ein Samba-Passwort
<code>sqlite3</code>	führt SQL-Kommandos in SQLite-Datenbanken aus

## Audio-Funktionen und Hardware-Verwaltung

---

<code>acpi</code>	liefert Informationen über den Batteriezustand
<code>alsactl</code>	speichert bzw. lädt alle Parameter des Audio-Systems
<code>alsamixer</code>	stellt die Audio-Kanäle interaktiv ein
<code>amixer</code>	steuert die Audio-Kanäle durch Optionen
<code>boltctl</code>	steuert Geräte an der Thunderbolt-Schnittstelle
<code>free</code>	zeigt den freien Speicherplatz an (RAM/Swap)
<code>fwupdmgmgr</code>	verwaltet Firmware-Updates
<code>hwclock</code>	liest bzw. stellt die Hardware-Uhr
<code>kbdrate</code>	stellt die Tastenwiederholungsrate ein
<code>localectl</code>	verändert die Sprach- und Tastatureinstellungen
<code>lscpu</code>	liefert Informationen über die CPU
<code>lshw</code>	bildet eine hierarchische Liste der Hardware-Komponenten
<code>lspci</code>	liefert Informationen über PCI-Komponenten
<code>lsscsi</code>	liefert Informationen über angeschlossene SCSI-Geräte
<code>lsusb</code>	liefert Informationen über angeschlossene USB-Geräte
<code>nproc</code>	liefert die Anzahl der CPU-Cores
<code>pactl</code>	steuert den PulseAudio-Server
<code>paplay</code>	spielt eine RAW-Datei via PulseAudio ab
<code>parecord</code>	paplay führt eine RAW-Audio-Aufnahme via PulseAudio durch
<code>powertop</code>	hilft, den Energieverbrauch von Notebooks zu optimieren
<code>speaker-test</code>	testet das Audio-System und die angeschlossenen Lautsprecher
<code>timedatectl</code>	stellt Datum, Uhrzeit und Zeitzone ein

`vcgencmd` liest bzw. verändert Hardware-Parameter des Raspberry Pi

## Bluetooth

---

<code>bluetoothctl</code>	konfiguriert Bluetooth-Geräte
<code>hciconfig</code>	(de)aktiviert lokale Bluetooth-Adapter
<code>hcitool</code>	verwaltet Bluetooth-Geräte
<code>l2ping</code>	sendet Echo-Anfragen an Bluetooth-Geräte
<code>rfkill</code>	(de)aktiviert Bluetooth-, WLAN- und Mobilfunk-Adapter
<code>sdptool</code>	ermittelt Detailinformationen über Bluetooth-Geräte

## Kernel

---

<code>canonical-livepatch</code>	administriert Kernel-Live-Patches (Ubuntu)
<code>depmod</code>	erzeugt eine Datei mit allen Modulabhängigkeiten
<code>dmesg</code>	zeigt die Meldungen des Kernels an
<code>dracut</code>	erzeugt eine neue Initrd-Datei (Fedora, RHEL, SUSE)
<code>insmod</code>	lädt ein Kernelmodul (Low-Level)
<code>kexec</code>	aktiviert einen anderen Kernel
<code>lsmod</code>	listet alle geladenen Kernelmodule auf
<code>modinfo</code>	liefert Informationen zu einem Kernelmodul
<code>modprobe</code>	lädt ein Kernelmodul samt Abhängigkeiten
<code>uname</code>	zeigt die aktuelle Kernelversion an
<code>update-initramfs</code>	erzeugt eine neue Initrd-Datei (Debian, Ubuntu)

## Systemstart und -stop, Init-System, Logging und GRUB

---

<code>efibootmgr</code>	liest bzw. verändert die Tabelle der EFI-Booteinträge
<code>grub-install</code>	installiert GRUB in den Bootsektor
<code>grub-mkconfig</code>	erzeugt eine neue GRUB-Konfigurationsdatei

<code>init</code>	wechselt in einen anderen Runlevel (Init-V-System)
<code>journalctl</code>	liest Nachrichten aus dem Journal
<code>logger</code>	protokolliert eine Nachricht
<code>logindctl</code>	steuert den systemd-Login-Manager
<code>needs-restarting</code>	verrät, ob das System oder einzelne Dienste neu gestartet werden müssen
<code>service</code>	führt ein Init-V-Script aus
<code>shutdown</code>	beendet Linux
<code>systemctl</code>	verwaltet systemd-Prozesse
<code>update-grub</code>	aktualisiert die GRUB-Konfiguration (Debian, Ubuntu)

## **Virtualisierung & Co. (Cloud, Docker, libvirt, KVM, Vagrant)**

---

<code>aws</code>	steuert Amazon-Cloud-Dienste
<code>docker</code>	verwaltet Container
<code>docker-compose</code>	richtet mehrere Container ein
<code>kvm</code>	führt eine virtuelle Maschine aus
<code>qemu-img</code>	erzeugt bzw. bearbeitet Image-Dateien
<code>qemu-kvm</code>	kvm führt eine virtuelle Maschine aus
<code>qemu-nbd</code>	bietet eine Image-Datei als Netzwerk-Block-Device an
<code>virsh</code>	führt libvirt-Kommandos aus
<code>virt-clone</code>	erstellt eine Kopie einer virtuellen Maschine
<code>virt-install</code>	richtet eine neue virtuelle Maschine ein
<code>virt-viewer</code>	erlaubt die Bedienung einer virtuellen Maschine via VNC
<code>wsl</code>	verwaltet Linux-Installationen unter Windows

## **Terminal und Textkonsole**

---

<code>echo</code>	gibt eine Zeile Text aus
<code>loadkeys</code>	lädt eine Tastaturtabelle für Textkonsolen
<code>printf</code>	ermöglicht eine formatierte Ausgabe wie unter C
<code>reset</code>	führt einen Reset für das Terminal durch

<code>screen</code>	verwaltet mehrere Sessions in einem Terminal
<code>setfont</code>	verändert die Terminal-Schriftart
<code>setterm</code>	verändert diverse Terminaleinstellungen

## Online-Hilfe

---

<code>apropos</code>	sucht Kommandos zu einem Thema
<code>help</code>	zeigt die Beschreibung eines Shell-Kommandos an
<code>info</code>	startet das <code>info</code> -System
<code>man</code>	zeigt die Beschreibung eines Kommandos an
<code>whatis</code>	zeigt eine Kurzbeschreibung eines Kommandos an

## Grafiksystem und Gnome

---

<code>chvt</code>	wechselt die aktive Textkonsole bzw. aktiviert das Grafiksystem
<code>dconf</code>	verändert die dconf-Datenbank (Low-Level)
<code>fc-list</code>	listet alle skalierbaren Schriften auf
<code>fgconsole</code>	liefert die Nummer der aktiven Konsole
<code>glxinfo</code>	überprüft die 3D-Funktionen des Grafiksystems
<code>gnome-session-quit</code>	initiiert einen Logout bzw. Shutdown
<code>gsettings</code>	liest bzw. ändert Einstellungen der dconf-Datenbank
<code>gtf</code>	berechnet die Parameter für einen neuen Grafikmodus
<code>nvidia-xconfig</code>	hilft bei der Konfiguration des NVIDIA-Grafiktreibers
<code>tvservice</code>	steuert den HDMI-Ausgang (Raspberry Pi OS)
<code>xdpinfo</code>	liefert Informationen zum laufenden X-Server
<code>xhost</code>	erlaubt oder sperrt Hosts für den X-Login
<code>xinput</code>	konfiguriert Eingabegeräte für X
<code>xkill</code>	beendet ein Programm per Mausklick
<code>xrandr</code>	ändert die Auflösung des Grafiksystems
<code>xset</code>	ändert Benutzereinstellungen des Grafiksystems

`zenity`

zeigt einfache grafische Dialoge an

## Sonstiges

---

`alias`

definiert eine Abkürzung

`basename`

ermittelt den Dateinamen eines Pfads

`cksum`

berechnet die CRC-Prüfsumme zu einer Datei

`date`

zeigt Datum und Uhrzeit an

`dirname`

ermittelt das Verzeichnis eines Pfads

`expr`

führt Berechnungen und Mustervergleiche durch

`git`

steuert das Versionsverwaltungssystem Git

`gpio`

steuert die Input/Output-Pins des Raspberry Pi

`hash`

zeigt die Hash-Tabelle an

`ldd`

zeigt die erforderlichen Libraries für ein Programm an

`lsb_release`

lsb-release stellt den Namen und die Version der Distribution fest

`mail`

übergibt eine E-Mail an den lokalen Mail-Server

`md5sum`

berechnet eine Prüfsumme zu einer Datei

`printenv`

zeigt nur die Umgebungsvariablen an

`qalc`

ist ein Taschenrechner für das Terminal

`raspi-gpio`

steuert die Input/Output-Pins des Raspberry Pi

`raspistill`

nimmt ein Foto auf (Raspberry Pi OS)

`raspivid`

nimmt ein Video auf (Raspberry Pi OS)

`seq`

liefert eine Zahlensequenz

`set`

zeigt alle der Shell bekannten Variablen an

`sha512sum`

berechnet eine Prüfsumme zu einer Datei

`sleep`

wartet eine vorgegebene Zeit

`strace`

verrät, welche Funktionen ein Programm aufruft

`svn`

steuert das Versionsverwaltungssystem Subversion

`time`

misst die Ausführzeit eines Kommandos

`tty`

zeigt den Device-Namen des Terminals an

`type`

gibt den Typ eines Kommandos an

<code>unalias</code>	löscht eine Abkürzung
<code>uname</code>	liefert den Betriebssystemnamen und die Kernelversion
<code>xargs</code>	leitet die Standardeingabe an ein Kommando weiter

## bash-Programmierung

---

<code>break</code>	beendet eine Schleife vorzeitig
<code>case</code>	leitet eine Fallunterscheidung ein
<code>continue</code>	überspringt den Schleifenkörper
<code>exit</code>	beendet das Shell-Programm
<code>for</code>	leitet eine Schleife ein
<code>function</code>	definiert eine neue Funktion
<code>if</code>	leitet eine Verzweigung ein
<code>local</code>	definiert lokale Variablen in einer Funktion
<code>source</code>	führt die angegebene Shell-Datei aus
<code>test</code>	wertet eine Bedingung aus
<code>until</code>	leitet eine Schleife ein (Variante 1)
<code>while</code>	leitet eine Schleife ein (Variante 2)

## bash-Variablenverwaltung

---

<code>alias</code>	definiert eine Abkürzung
<code>declare</code>	definiert eine (Umgebungs-)Variable
<code>export</code>	definiert eine Umgebungsvariable
<code>local</code>	definiert lokale Variablen in einer Funktion
<code>read</code>	liest eine Variable ein
<code>readonly</code>	zeigt alle schreibgeschützten Variablen an
<code>shift</code>	verschiebt die Parameterliste
<code>unalias</code>	löscht eine Abkürzung
<code>unset</code>	löscht eine Variable

## Weitere bash-Kommandos und -Sonderzeichen

---

<code>dirs</code>	zeigt die Liste der gespeicherten Verzeichnisse an
<code>disown</code>	löst einen Prozess von der Shell

<code>eval</code>	wertet das angegebene Kommando aus
<code>popd</code>	wechselt in das letzte gespeicherte Verzeichnis
<code>pushd</code>	speichert das Verzeichnis und wechselt in ein anderes
<code>trap</code>	führt beim Eintreten eines Signals ein Kommando aus
<code>ulimit</code>	kontrolliert die von der Shell beanspruchten Ressourcen
<code>wait</code>	wartet auf das Ende eines Hintergrundprozesses
<code>#&amp;%!</code>	bash-Sonderzeichen

## Konfigurationsdateien

---

<code>adduser.conf</code>	Einstellungen für neue Accounts (Debian, Ubuntu)
<code>aliases</code>	E-Mail-Weiterleitungen
<code>bashrc</code>	Defaulteinstellungen für die <code>bash</code>
<code>config.txt</code>	Raspberry-Pi-Hardware-Parameter (Raspberry Pi OS)
<code>crontab</code>	Prozesse periodisch ausführen
<code>deluser.conf</code>	Einstellungen für <code>deluser</code> und <code>delgroup</code>
<code>dhcpcd.conf</code>	Netzwerkkonfiguration (Raspberry Pi OS)
<code>dnf.conf</code>	Konfiguration der Paketverwaltung <code>dnf</code> (Fedora)
<code>fstab</code>	Dateisysteme/Partitionen automatisch einbinden
<code>group</code>	Gruppennamen und Gruppenzuordnungen
<code>grub</code>	Defaulteinstellungen für GRUB 2
<code>grub.cfg</code>	Konfiguration für GRUB 2
<code>gshadow</code>	Hash-Codes der Gruppenpasswörter
<code>host.conf</code>	Konfiguration der Resolver-Bibliothek
<code>hostname</code>	Hostname des Rechners
<code>hosts</code>	statische Liste von Hostnamen und IP-Adressen

<code>ifcfg-xxx</code>	Netzwerkparameter einer Schnittstelle (Fedora, Red Hat)
<code>inittab</code>	Defaulteinstellungen für das Init-V-System
<code>interfaces</code>	Netzwerkkonfiguration (Debian, Ubuntu)
<code>journald.conf</code>	Konfiguration des Logging-Diensts Journal
<code>locale.conf</code>	Lokalisierungseinstellungen (systemd)
<code>login.defs</code>	Optionen für das Anlegen neuer Benutzer und Gruppen
<code>mdadm.conf</code>	Software-RAID-Konfiguration
<code>modules</code>	Kernelmodule automatisch laden (Debian, Ubuntu)
<code>netplan.yaml</code>	Netzwerkeinstellungen (Ubuntu)
<code>networkd.network</code>	networkd-Konfiguration (systemd)
<code>nsswitch.conf</code>	Konfiguration der Name-Service-Switch-Funktionen
<code>os-release</code>	Namen und Versionsnummer der Distribution (systemd)
<code>passwd</code>	Liste aller Benutzer und Heimatverzeichnisse
<code>profile</code>	Konfiguration systemweiter Umgebungsvariablen
<code>rc.local</code>	Script, wird am Ende des Bootprozesses ausgeführt
<code>resolv.conf</code>	IP-Adresse des Nameservers
<code>rsyslog.conf</code>	Konfiguration des Syslog-Dienstes
<code>services</code>	Zuordnung zwischen Netzwerkdiensten und Ports
<code>shadow</code>	Hash-Codes der Login-Passwörter
<code>sources.list</code>	APT-Paketquellen (Debian, Ubuntu)
<code>sudoers</code>	Konfiguration für sudo
<code>sysctl.conf</code>	Defaulteinstellungen für Kernelparameter
<code>systemd.service</code>	Konfiguration von systemd-Diensten
<code>systemd.timer</code>	Konfiguration von periodischen systemd-Jobs
<code>vconsole.conf</code>	Tastatureinstellungen (systemd)

wpa_supplicant.conf	wpasupplicant.conf WLAN-Konfiguration (Raspberry Pi OS)
xorg.conf	Konfiguration des Grafiksystems
yum.conf	Konfiguration von yum (Fedora, Red Hat)

## Tastenkürzel

---

bash	Shell
emacs	Editor
fdisk	Partitionseditor
gnome-terminal	Terminal-Fenster unter Gnome
grub	Bootloader
info	Kommando zur Anzeige von Hilfetexten
joe	einfacher Editor
konsole	Terminal-Fenster unter KDE
less	Kommando zur Anzeige von Textdateien
man	Kommando zur Anzeige von Hilfetexten
mutt	E-Mail-Client für den Textmodus
nano	minimalistischer Editor
screen	Terminal-Multiplexer
----	Tastenkürzel in Textkonsolen
vi/vim	Editor

# 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.

Das Buch fasst außerdem die Syntax elementarer Konfigurationsdateien zusammen und enthält eine Tastenkürzelreferenz der Editoren Emacs, Nano und Vi sowie einiger anderer interaktiver Kommandos wie `less` oder `info`.

Das Ziel dieses Buches ist es, ein kompaktes Nachschlagewerk für die Arbeit mit Linux im Terminal zu bieten. Obwohl die Anzahl der Seiten in der hier vorliegenden Auflage weiter gestiegen ist, bleibt der Grundsatz dieses Buchs weiterhin: *Weniger ist mehr*. Dieses Buch kann und soll die `man`- und `info`-Seiten komplexer Kommandos nicht ersetzen! Exotische Optionen müssen Sie also weiterhin selbst nachsehen bzw. recherchieren.

Mit diesem Buch versuche ich aber, Ihnen die Arbeit abzunehmen, die oft über Dutzende Seiten reichende Originaldokumentation nach Optionen für den alltäglichen Gebrauch zu durchsuchen. Zahlreiche Beispiele zeigen zudem auf einen Blick den prinzipiellen Einsatz eines Kommandos.

Mitunter ist es so, dass man für eine bestimmte Aufgabe ein Kommando sucht, dessen Namen aber nicht kennt oder gerade vergessen hat. Für diese Fälle ist das thematisch organisierte Inhaltsverzeichnis gedacht.

Je nachdem, welche Distribution Sie einsetzen, stehen einige Kommandos standardmäßig nicht zur Verfügung und müssen extra installiert werden. Zudem gibt es distributionsspezifische Kommandos, die nur unter bestimmten Distributionen zur Verfügung stehen – z.B. die Paketverwaltungskommandos `dpkg` und `apt` (Debian, Ubuntu), `rpm` und `dnf` oder `yum` (Fedora, Red Hat) sowie

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, LibreOffice 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.

## 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:

```
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 die zuletzt angegebene Option.

### **man, info und help**

Um dieses Buch 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 Sie mit `man name` bzw. mit `man 1 name` lesen 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

## A

### **aa-complain** programm

Das Kommando aa-complain aktiviert den complain-Modus für das AppArmor-Profil des angegebenen Programms. Regelverstöße werden damit zwar protokolliert, aber nicht geahndet, d.h., das Programm läuft ungestört weiter.

aa-complain steht wie alle anderen aa-xxx-Kommandos nur bei Distributionen zur Verfügung, die das Sicherheitssystem AppArmor verwenden, also (open)SUSE und Ubuntu. Unter Ubuntu befindet sich das Kommando im Paket apparmor-utils, das extra installiert werden muss.

### **Beispiel**

Das folgende Kommando bewirkt, dass das Programm dovecot zwar weiterhin überwacht wird, Regelverstöße aber lediglich in einer Logging-Datei festgehalten werden:

```
root# aa-complain /usr/sbin/dovecot
Setting /usr/sbin/dovecot to complain mode.
```

### **aa-disable** programm

aa-disable deaktiviert das AppArmor-Profil für das angegebene Programm. Zur Reaktivierung verwenden Sie [aa-enforce](#) oder [aa-complain](#).

### **aa-enforce** programm

Das Kommando [aa-complain](#) aktiviert den enforce-Modus für das AppArmor-Profil des angegebenen Programms. AppArmor unterbindet damit Datei- oder Netzwerkzugriffe, die durch AppArmor-Regeln verboten sind.

### **aa-status** [option]

aa-status liefert eine Zusammenfassung über den Zustand des AppArmor-Systems. Durch die Angabe genau einer Option aus --

`complaining`, `--enabled`, `--enforced` oder `--profiled` kann die Ausgabe auf eine Detailinformation verkürzt werden.

## Beispiel

Auf dem Testrechner ist AppArmor aktiv. 25 Regelprofile sind geladen, aber nur vier davon überwachen tatsächlich laufende Programme.

```
root# aa-status
25 profiles are loaded.
25 profiles are in enforce mode.
  /sbin/dhclient
  /usr/bin/evince
  ...
  libreoffice-xpdfimport
0 profiles are in complain mode.
4 processes have profiles defined.
4 processes are in enforce mode.
  /sbin/dhclient (651)
  /usr/sbin/cups-browsed (537)
  /usr/sbin/cupsd (452)
  /usr/sbin/mysqld (612)
0 processes are in complain mode.
0 processes are unconfined but have a profile defined.
```

## ack [optionen] suchbegriff

ack ist eine für Programmierer und Administratoren optimierte Variante von [grep](#). Im Gegensatz zum Original ignoriert ack GIT- und SVN-Verzeichnisse, Backup-Dateien sowie binäre Dateien. Bei vielen Distributionen befindet sich ack im gleichnamigen Paket oder in ack-grep. Alternativ kann ack auch als Perl-Modul installiert werden (`cpan App::Ack`).

ack ist grundsätzlich kompatibel zu [grep](#), durchsucht aber standardmäßig rekursiv alle Dateien im aktuellen Verzeichnis (wie `grep -r`). Am häufigsten wird das Kommando in den Formen `ack suchbegriff` oder `ack -i suchbegriff` (Groß- und Kleinschreibung ignorieren) aufgerufen. Eine Referenz von ack-spezifischen Optionen liefern `ack --help` und `man ack`.

Eine Alternative zur ack-Variante ist das Kommando ag. Manche Distributionen bieten das Kommando im Paket `silver searcher-ag`

an. Alternativ können Sie das Programm von GitHub klonen und dann selbst kompilieren:

[https://github.com/ggreer/the\\_silver\\_searcher/wiki](https://github.com/ggreer/the_silver_searcher/wiki)

## Beispiel

Das folgende Beispiel sucht nach Dateien in `/etc`, die den Suchbegriff `localhost` enthalten, und gibt die entsprechenden Treffer samt Zeilennummern an:

```
root# cd /etc
root# ack localhost
security/pam_env.conf
 52:# to "localhost" rather than not being set at all
 53:#REMOTEHOST DEFAULT=localhost OVERRIDE=@{PAM_RHOST}
 64:#NNTPSERVER DEFAULT=localhost

speech-dispatcher/speechd.conf
 38:# for connections coming from localhost. If LocalhostAccessOnly is set
      to 0 it disables this
 199:#AudioNASServer "tcp/localhost:5450"

speech-dispatcher/modules/festival.conf
 7:#FestivalServerHost "localhost"
...

```

### acme.sh [optionen]

`acme.sh` ist ein Script, das bei der Installation von Let's-Encrypt-Zertifikaten hilft. Das Script ist eine handliche Alternative zum wesentlich bekannteren Kommando [certbot](#), dessen Python-Abhängigkeiten aber in der Vergangenheit oft zu Installationsproblemen führten.

Auch `acme.sh` muss installiert werden. Das gelingt mühelos:

```
user$ sudo apt/dnf/yum/zypper install curl socat
user$ curl https://get.acme.sh -o acme-setup
user$ less acme-setup    (Script-Code kurz kontrollieren)
user$ sudo sh acme-setup
...
Installed to /root/.acme.sh/acme.sh
Installing alias to '/root/.bashrc'
OK, Close and reopen your terminal to start using acme.sh
```

Im Rahmen der Installation wird `/var/spool/root` um eine Zeile erweitert. Dadurch kümmert sich Cron um den regelmäßigen Aufruf von `acme.sh --cron` und damit um die automatische Erneuerung aller Zertifikate, die mit `acme.sh` eingerichtet wurden.

- **--cron**  
erneuert alle mit acme.sh installierten Zertifikate, die älter als 60 Tage sind. Mit der zusätzlichen Option **--force** kann eine vorzeitige Erneuerung erzwungen werden.
- **--install-cert -d *domain***  
`[--cert-file path/to/certfile]`  
`[--key-file path/to/keyfile]`  
`[--fullchain-file path/to/fullchainfile]`  
`[--reloadcmd "systemctl restart httpd"]`  
 kopiert die zuvor mit acme.sh --issue erzeugten Zertifikate an einen Ort, wo der Webserver auf sie zugreifen kann, und startet den Webserver anschließend neu. Anstelle von `systemctl restart httpd` müssen Sie ein für Ihre Distribution geeignetes Kommando angeben, um den Webserver neu zu starten.  
 Beachten Sie, dass es nicht zielführend ist, die in `/root/.acme.sh` gespeicherten Zertifikate selbst in ein anderes Verzeichnis zu kopieren. acme.sh merkt sich die Pfade und das Reload-Kommando für die automatische Zertifikaterneuerung.  
 acme.sh verändert die Konfigurationsdateien Ihres Webservers nicht! Sie müssen die entsprechenden Anweisungen mit den Pfaden zu den Zertifikatsdateien selbst hinzufügen.
- **--issue -d *domain* -w *webrootdir***  
richtet ein Zertifikat für die angegebene Domäne ein. Die Option **-d** kann mehrfach verwendet werden, um ein Zertifikat für zusammengehörige Domänen zu erzeugen (also z.B. `-d eine-firma.de -d www.eine-firma.de`). Wenn Sie ein Wildcard-Zertifikat wünschen, übergeben Sie beispielsweise `-d eine-firma.de -d *.eine-firma.de`. Die Zertifikate werden im Verzeichnis `/root/.acme.sh` gespeichert.
- **--remove -d *domain***  
deaktiviert für die angegebene Domäne die Zertifikaterneuerung, die automatisch alle 60 Tage stattfindet.

Obwohl der Optionsname das Gegenteil vermuten lässt, werden die Zertifikate selbst nicht gelöscht. Sie können aber nun das entsprechende Verzeichnis `/root/.acme.sh/domain` selbst löschen.

- `--upgrade`  
aktualisiert das Script `acme.sh`.

## Beispiel

Die folgenden Kommandos erzeugen ein Zertifikat für die Domänen `eine-firma.de` und `www.eine-firma.de` und kopieren die Zertifikate in das für diesen Zweck eingerichtete Verzeichnis `/etc/acme-letsencrypt`:

```
root# mkdir /etc/acme-letsencrypt
root# acme.sh --issue -d eine-firma.de -w /var/www/html
Your cert is in /root/.acme.sh/eine-firma.de/eine-firma.de.cer
Your cert key is in /root/.acme.sh/eine-firma.de/eine-firma.de.key
The intermediate CA cert is in /root/.acme.sh/eine-firma.de/ca.cer
And the full chain certs is there: /root/.acme.sh/eine-firma.de/fullchain.cer
root# acme.sh --install-cert -d eine-firma.de \
    --cert-file /etc/acme-letsencrypt/eine-firma.de.cert \
    --key-file /etc/acme-letsencrypt/eine-firma.de.key \
    --fullchain-file /etc/acme-letsencrypt/eine-firma.de.fullchain
    --reloadcmd 'systemctl restart httpd'
Installing cert to:/etc/acme-letsencrypt/eine-firma.de.cert
Installing key to:/etc/acme-letsencrypt/eine-firma.de.key
Installing full chain to:/etc/acme-letsencrypt/eine-firma.de.fullchain
Run reload cmd: systemctl restart httpd
Reload success
```

## 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.
- `-V`  
zeigt alle verfügbaren Informationen an.

## add-apt-repository ppa:name

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

## Beispiel

Das folgende Kommando richtet eine Paketquelle für die jeweils aktuellste Version von LibreOffice ein:

```
root# add-apt-repository ppa:libreoffice/ppa
```

```
addgroup [optionen] name
```

`addgroup` richtet auf Debian- und Ubuntu-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](#) mit eigener Syntax.

- `--gid n`

weist der Gruppe die angegebene GID-Nummer zu (Group Identification).

```
adduser [optionen] name
```

```
adduser name group
```

`adduser` richtet auf Debian- und Ubuntu-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](#), sodass dort eine andere Syntax gilt. Unter (open)SUSE steht `adduser` nicht zur Verfügung, Sie müssen stattdessen [useradd](#) verwenden.

- **--disabled-login**  
verzichtet auf die Passwortabfrage. 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/Ubuntu-System den neuen Benutzer `kofler` ein und fügt diesen anschließend der Gruppe `libvирtd` hinzu:

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

Um einen Benutzer einzurichten, der sich nicht einloggen kann (z.B. für einen Mail-Account), gehen Sie wie folgt vor:

```
root# adduser --shell /bin/false --gecos ,,, mailuser
alias abkürzung='kommando'
```

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

Unter Fedora und Red Hat gibt es für `root` einige vordefinierte Aliase. Sie werden beim Login aus der Datei `/root/.bashrc` gelesen.

## 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.

alien funktioniert nur bei einfachen Paketen problemlos. Wenn ein Paket dagegen Installations-Scripts oder andere spezifische Eigenheiten des jeweiligen Paketformats nutzt, wird die Installation des zuvor umgewandelten Pakets sehr wahrscheinlich scheitern.

**alsactl** [optionen] [kommando] [card/id/device]

Mit alsactl können Sie den Zustand des Audio-Systems anzeigen, speichern und wiederherstellen. Außerdem bietet das Kommando einige fortgeschrittene Steuerungsmöglichkeiten, die im einfacheren Kommando [amixer](#) nicht zur Verfügung stehen.

alsactl gilt generell für alle Audio-Geräte, es sei denn, Sie geben beim Aufruf des Kommandos explizit die Nummer oder den Device-Namen des Audio-Geräts an (siehe auch die Datei `/proc/asound/cards`).

- `-b`

führt alsactl im Hintergrund aus. Diese Option hilft dem Init-System dabei, alsactl als Hintergrunddienst zu starten.

- `-f datei`

gibt an, welche Datei alsactl auslesen bzw. zum Speichern verwenden soll (standardmäßig `/var/lib/alsa/asound.state`).

- `-s`

verwendet den Syslog-Dienst zur Protokollierung von Nachrichten, Warnungen und Fehlern.

Die wichtigsten Kommandos von `alsactl` gehen aus der nachfolgenden Aufzählung hervor. Es gibt einige weitere Kommandos, die speziell für den Einsatz als Hintergrunddienst gedacht sind und für den interaktiven Aufruf bzw. für Scripts selten von Interesse sind.

- `init`  
initialisiert das Audio-System mit Defaulteinstellungen.
- `monitor`  
verfolgt alle Änderungen des Audio-Systems und zeigt diese an, bis das Kommando mit `Strg+C` beendet wird.
- `restore`  
stellt einen zuvor gespeicherten Zustand des Audio-Systems wieder her. Die Daten werden aus `/var/lib/alsa/asound.state` oder aus der durch `-f` angegebenen Datei gelesen.
- `store`  
speichert den Zustand des Audio-Systems in der Textdatei `/var/lib/alsa/asound.state` oder in einer anderen, durch `-f` angegebenen Datei.

### **alsamixer [optionen]**

`alsamixer` ist ein interaktives Kommando, in dem Sie durch Funktions- und Cursortasten die Lautstärke bzw. den Aufnahmepegel der Audio-Kanäle einstellen können. `Esc` beendet das Programm. `alsamixer` ist nicht zur Script-Programmierung geeignet. Dazu verwenden Sie [amixer](#).

- `-c n`  
wählt die gewünschte Audio-Karte aus, wenn es mehrere gibt. Die Audio-Karten sind durchnummieriert, beginnend mit 0. Werfen Sie gegebenenfalls einen Blick in die Datei `/proc/asound/cards`, die alle Audio-Karten aufzählt.
- `-D device`  
wählt das Audio-System aus. Beispielsweise können Sie mit `-D`

pulse Einstellungen des PulseAudio-Systems verändern. Ohne die Option steuert alsamixer direkt das ALSA-System.

- `-v playback|capture|all`

bestimmt, welche Regler anfänglich angezeigt werden sollen: Die zur Audio-Wiedergabe, die zur Aufnahme oder alle. Wenn alsamixer läuft, können Sie zwischen diesen Darstellungsformen mit `[F3]`, `[F4]` und `[F5]` wechseln.

#### **alternatives [optionen] kommando**

alternatives ist die Red-Hat- bzw. Fedora-Variante von update-alternatives. Die Syntax der beiden Kommandos ist weitgehend identisch und wird beim Kommando update-alternatives beschrieben.

#### **amixer [optionen] [kommando]**

Mit dem Kommando amixer können Sie durch Optionen die Lautstärke bzw. Pegel der Kanäle des Audio-Systems verändern. Wenn Sie die Lautstärke oder andere Einstellungen interaktiv regeln möchten, bietet sich dazu das Kommando alsamixer mit einer einfachen, textbasierten Benutzeroberfläche an.

- `-c n`

wählt die gewünschte Audio-Karte aus. Die Audio-Karten sind durchnummeriert, beginnend mit 0. Werfen Sie gegebenenfalls einen Blick in die Datei `/proc/#asound/cards`, die alle Audio-Karten aufzählt. Wenn die Option fehlt, bearbeitet amixer das Default-Audio-Device.

- `-D device`

wählt das Audio-System aus (z.B. `-D pulse` für das PulseAudio-System). Ohne die Option steuert amixer direkt das ALSA-System.

- `-q`

verzichtet darauf, nach der Veränderung von Einstellungen den neuen Zustand auszugeben.

- **-s**  
liest die auszuführenden Kommandos aus der Standardeingabe.  
Dabei werden ausschließlich die Kommandos sset und cset berücksichtigt.

Die folgende Liste beschreibt die wichtigsten amixer-Kommandos:

- **cget *id***  
zeigt den Zustand eines Schalters an. Die Bezeichnung des Schalters kann durch einen oder mehrere Parameter erfolgen, z.B. in der Form numid=3 oder iface=MIXER, name='Headphone Playback Switch'. Eine Liste aller Regler und Schalter samt entsprechender ID-Daten erstellt amixer controls.
- **controls|scontents|scontrols**  
listet alle Regler bzw. Schalter in einer unterschiedlich detaillierten Darstellung auf und gibt deren aktuellen Zustand an.
- **cset *id parameter***  
verändert den Zustand eines Schalters. Beispielsweise schaltet amixer cset numid=3 1 auf einem Raspberry Pi den Audio-Ausgang ein.
- **info**  
zeigt eine kurze Beschreibung der Audio-Karte an.
- **sget '*name*',*n***  
liefert den Zustand des angegebenen Reglers. Regler werden durch eine Kombination aus Zeichenkette und Zahl beschrieben, z.B. 'Master', 0. Die Zahl ist zumeist 0, es sei denn, eine Audio-Karte bietet mehrere gleichnamige Regler an, die dann durchnummeriert werden.
- **sset '*name*',*n parameter***  
verändert den Zustand des Reglers. Im einfachsten Fall geben Sie als Parameter einen Prozentwert an, z.B. 50%. Falls es sich um einen Stereo-Regler handelt, können Sie die beiden Kanäle durch ein Zahlenpaar getrennt einstellen, z.B. mit 40%, 55%. Alternativ

können viele Regler mit dem Suffix dB auch in der Einheit Dezibel eingestellt werden, z.B. -12.5dB.

Durch nachgestellte Plus- oder Minus-Zeichen können Sie Regler auch relativ verändern. 5%+ erhöht die Lautstärke oder den Aufnahmepiegel um fünf Prozentpunkte, 5%- reduziert Lautstärke bzw. Pegel um fünf Prozentpunkte.

Mit `mute`, `unmute` und `toggle` verändern Sie den Mute-Modus. `cap` aktiviert die Aufnahme (*Capture*), `nocap` deaktiviert diese Funktion.

Falls der Regler mehrere Kanäle unterstützt, können Sie diese mit den zusätzlichen Parametern `front`, `rear`, `center` und `woofer` auswählen.

## Beispiel

Auf dem Testrechner ist das dritte Audio-Gerät (-c 2, die Nummerierung beginnt mit 0) ein USB-Headset. Es gibt nur zwei *Simple Controls* zur Steuerung der Kopfhörer und des Mikrofons. `amixer sset` stellt nun die Kopfhörer auf eine mittlere Lautstärke.

```
user$ amixer -c 2 scontrols
Simple mixer control 'Headphone',0
Simple mixer control 'Mic',0
user$ amixer -c 2 sset 'Headphone',0 50% unmute
Simple mixer control 'Headphone',0
  Capabilities: pvolume pswitch pswitch-joined
  Playback channels: Front Left - Front Right
  Limits: Playback 0 - 41
  Mono:
    Front Left: Playback 21 [51%] [-20.00dB] [on]
    Front Right: Playback 21 [51%] [-20.00dB] [on]
```

## apk kommando

Das Kommando apk dient zur Paketverwaltung in Alpine Linux. Diese minimalistische Linux-Distribution ist vor allem im Docker-Kontext populär.

- add *name*  
installiert das angegebene Paket. Wenn an das Kommando die Option --update übergeben wird, aktualisiert apk vorher den Paket-Cache (wie durch apk update). --no-cache hat fast dieselbe

Wirkung, allerdings wird der Cache anschließend sofort wieder gelöscht, um Platz zu sparen.

- `del name`  
entfernt das angegebene Paket.
- `info`  
listet alle installierten Pakete auf.
- `search suchbegriff`  
sucht nach Paketen, auf deren Paketname der Suchbegriff zutrifft.
- `status`  
verrät, wie viele Pakete aktuell installiert sind.
- `update`  
aktualisiert den Cache aller in `/etc/apk/repositories` aufgezählten Paketquellen.
- `upgrade`  
aktualisiert alle installierten Pakete.

## Beispiel

Das folgende Kommando aktualisiert zuerst den Cache der Paketquellen und installiert danach die beiden Pakete `build-base` und `python-dev`:

```
root# apk add --update build-base python-dev  
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.

## Beispiel

`apropos editor` liefert eine Liste diverser installierter Edit-Kommandos:

```
user$ apropos editor  
ed (1)      - line-oriented text editor  
editor (1)  - Vi IMproved, a programmers text editor  
ex (1)      - Vi IMproved, a programmers text editor  
jmacs (1)   - Joe's Own Editor  
joe (1)     - Joe's Own Editor  
...  
...
```

## **apt** kommando

apt ist das populärste Kommando, um auf Debian-basierten Distributionen Pakete zu verwalten. Das gilt auch für Linux Mint, Raspberry Pi OS und Ubuntu. Mögliche Alternativen sind das früher sehr beliebte Kommando [apt-get](#) (es ist das weiterhin für automatisierten Installationen sowie zur Script-Programmierung vorzuziehen) sowie [aptitude](#) mit einer textbasierten grafischen Benutzeroberfläche.

apt installiert, aktualisiert und entfernt Pakete und berücksichtigt dabei alle Pakete in den in /etc/apt/sources.list definierten Paketquellen. Die folgende Aufzählung beschreibt die wichtigsten Kommandos:

- **autoclean**  
entfernt nur Pakete aus dem Paket-Cache, zu denen bereits eine neuere Version verfügbar ist.
- **autoremove**  
deinstalliert alle nicht mehr benötigten Pakete, die aufgrund von Abhängigkeiten installiert wurden.
- **full-upgrade**  
funktioniert ähnlich wie upgrade, führt aber auch solche Updates durch, für die Pakete entfernt werden müssen.
- **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.
- **list**  
listet alle verfügbaren Pakete auf. Mit der Option --installed

liefert das Kommando nur die aktuell installierten Pakete.

- `remove name1 name2 ...`  
deinstalliert die angegebenen Pakete.
- `search suchbegriff`  
listet alle Pakete auf, deren Name oder Beschreibung den Suchbegriff enthält.
- `show name1`  
zeigt Informationen zum angegebenen Paket an.
- `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 weiß, welche Pakete im Internet zur Verfügung stehen. Das Kommando sollte vor jedem anderen apt-Kommando ausgeführt werden.  
Hinweis: Wenn Sie nur ein Paket aktualisieren möchten (nicht alle Pakete), müssen Sie `apt install name` ausführen, nicht `apt update name!`
- `upgrade`  
aktualisiert alle installierten Pakete, soweit in den Paketquellen neuere Versionen zur Verfügung stehen. Bei Bedarf werden für das Update auch zusätzliche Pakete installiert. `apt upgrade` führt aber keine Updates durch, für die es notwendig ist, Pakete zu entfernen.

## Beispiel

Die folgenden Kommandos zeigen die Anwendung von apt:

```
root# apt update          (Paketquellen aktualisieren)
root# apt full-upgrade    (alle installierten Pakete aktualisieren)
root# apt install apache2  (den Apache-Webserver installieren)
```

**apt-cache** kommando

Das Debian- und Ubuntu-spezifische Kommando `apt-cache` liefert Informationen über verfügbare bzw. bereits installierte Pakete.

- `madison paketname`  
liefert eine einzeilige Zusammenfassung der Ergebnisse von apt-cache policy (siehe unten).
- `policy`  
zeigt eine detaillierte Liste aller Paketquellen.
- `policy paketname`  
zeigt an, aus welcher Paketquelle das gegenwärtig installierte Paket stammt und welche Paketquellen gegebenenfalls alternative Versionen des Pakets zur Verfügung stellen.
- `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.
- `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.
- `stats`  
liefert eine Statistik über die Anzahl der installierten und verfügbaren Pakete.  
  
apt-cache ist nicht in der Lage, detaillierte 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.

## Beispiele

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

```
root# apt-cache search --names-only gimp | sort
gimp-cbmplugs      - Erweiterungen für GIMP zum Import/Export von
                      Commodore-64-Dateien
gimp-data          - Daten für GIMP
gimp-data-extras   - Extra brushes and patterns for GIMP
...
```

Das zweite Beispiel zeigt, welche Version des Pakets `openjdk-11-jre` installiert ist und ob es alternativen Versionen gibt (was hier nicht der Fall ist):

```
user$ apt-cache policy openjdk-11-jre
openjdk-11-jre:
  Installed: 11.0.8+10-0ubuntu1~20.04
  Candidate: 11.0.8+10-0ubuntu1~20.04
  Version table:
*** 11.0.8+10-0ubuntu1~20.04 500
      500 http://at.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages
      500 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages
      100 /var/lib/dpkg/status
  11.0.7+10-3ubuntu1 500
      500 http://at.archive.ubuntu.com/ubuntu focal/main amd64 Packages
```

### **apt-get [optionen] kommando**

Das Debian- und Ubuntu-spezifische Kommando `apt-get` installiert, aktualisiert und entfernt Pakete. Für die interaktive Nutzung wird anstelle von `apt-get` heute das Kommando [apt](#) empfohlen. Es ist im Hinblick auf eine einfache Nutzung optimiert und stellt auch Subkommandos zur Paketsuche zur Verfügung (`apt show`, `apt search` und `apt list`), die bei `apt-get` fehlen.

`apt-get` ist damit aber nicht obsolet. Es kann weiterhin auf allen Debian- und Ubuntu-basierten Distributionen verwendet werden. Sein Einsatz ist insbesondere dann empfehlenswert, wenn Sie die Paketverwaltung in Scripts automatisieren möchten. Die folgende Aufzählung beschreibt die wichtigsten Kommandos:

- `autoclean`  
entfernt nur solche Pakete aus dem Paket-Cache, zu denen bereits eine neuere Version verfügbar ist.
- `autoremove`  
deinstalliert alle nicht mehr benötigten Pakete, die aufgrund von Abhängigkeiten installiert wurden.

- `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.
- `dist-upgrade`  
funktioniert ähnlich wie `upgrade`, installiert bei Bedarf aber auch neue Pakete.
- `install name1 name2 ...`  
installiert die Pakete `name1`, `name2` etc. und berücksichtigt dabei auch Paketabhängigkeiten. Wenn ein Paket bereits installiert ist, dann bewirkt `apt-get install name` die Aktualisierung dieses Pakets.
- `remove name1 name2 ...`  
deinstalliert die angegebenen Pakete.
- `source name`  
installiert den Quellcode des Pakets in das aktuelle Verzeichnis.
- `update`  
aktualisiert die Paketlisten der in `sources.list` angegebenen Archive. (Weitere Details sind bei [apt update](#) beschrieben.)
- `upgrade`  
aktualisiert alle installierten Pakete, soweit in den Paketquellen neuere Versionen zur Verfügung stehen.  
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 && apt-get dist-upgrade      (vollständiges Update durchführen)
root# apt-get remove -y emacs                      (emacs ohne Rückfragen entfernen)
```

### **apt-key** kommando

Das Debian- und Ubuntu-spezifische Kommando `apt-key` verwaltet die öffentlichen Schlüssel von APT-Paketquellen. `apt-key` kennt unter anderem die folgenden Befehle:

- **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.

## Beispiel

Die beiden folgenden Kommandos laden den Schlüssel für die VirtualBox-Paketquelle von Oracle herunter und fügen ihn zur Liste der APT-Schlüssel hinzu:

```
root# wget -q https://download.virtualbox.org/virtualbox/debian/oracle_vbox.asc
root# apt-key add oracle_vbox.asc
```

### **aptitude** [optionen] [kommando]

Das Debian-spezifische Kommando `aptitude` installiert, aktualisiert und entfernt Pakete und greift dabei wie `apt` und `apt-get` auf die

APT-Infrastruktur zurück. Der Vorteil im Vergleich zu diesen beiden Kommandos 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 bei diesem Kommando beschrieben. Die Kommandos `check` und `autoremove` sind in `aptitude` allerdings nicht verfügbar. Dafür gibt es das leistungsfähige Kommando `search`.

Alternativ können Sie das Programm auch mit einer Text-Benutzeroberfläche in einer Konsole nutzen, indem Sie das Programm einfach ohne weitere Parameter starten. Zur Menüauswahl verwenden Sie die Tastenkombination `Strg + T`. Wirklich intuitiv ist `aptitude` aber trotz des Menüs nicht zu bedienen; die meisten Anwender verwenden `aptitude` daher wie [apt-get](#) einfach zum Ausführen einzelner Kommandos. `aptitude` ist bei den meisten Distributionen nicht standardmäßig installiert und muss vor der ersten Verwendung mit [apt](#) oder [apt-get](#) installiert werden.

### **aptitude search**

`aptitude search` bietet umfassende Möglichkeiten, die Paketdatenbank zu durchsuchen. In der einfachsten Form suchen Sie mit `aptitude search suchbegriff` nach Paketen, in deren Name der Suchbegriff enthalten ist. Mit `-` und einem Buchstaben können Sie diverse Suchkriterien formulieren. Beispielsweise schränkt `-i` die Suche auf installierte Pakete ein. `-u` zeigt Pakete an, zu denen Updates verfügbar sind. Mit der Option `-F` steuern Sie, welche Details des Suchergebnisses in wie breiten Spalten ausgegeben werden sollen.

Eine Menge weiterer Suchoptionen sind in  
*/usr/share/doc/aptitude/README* sowie auf der folgenden Seite  
beschrieben:

<https://www.debian.org/doc/manuals/aptitude/ch02s04s05.en.html>

## Beispiele

Die beiden folgenden Kommandos installieren zuerst das Paket `mysql-server` und entfernen es dann wieder. Bemerkenswert ist, dass beim zweiten Kommando fast alle zusammen mit `mysql-server` installierten abhängigen Pakete wieder entfernt werden. (Erhalten bleiben allerdings die vom Server unabhängigen Client-Werkzeuge und -Bibliotheken.)

```
root# aptitude install mysql-server
```

Die folgenden NEUEN Pakete werden zusätzlich installiert:

```
 libaio1{a} libevent-core-2.1-7{a} libhtml-template-perl{a} libmecab2{a}  
 mecab-ipadic{a} mecab-ipadic-utf8{a} mecab-utils{a} mysql-client-8.0{a}  
 mysql-client-core-8.0{a} mysql-server mysql-server-8.0{a}  
 mysql-server-core-8.0{a}
```

0 Pakete aktualisiert, 12 zusätzlich installiert, 0 werden entfernt.

```
...
```

```
root# aptitude remove mysql-server
```

Die folgenden Pakete werden ENTFERNT:

```
 libaio1{u} libevent-core-2.1-7{u} libmecab2{u} mecab-ipadic{u}  
 mecab-ipadic-utf8{u} mecab-utils{u} mysql-client-8.0{u} ...  
 mysql-server-core-8.0{u}
```

0 Pakete aktualisiert, 0 zusätzlich installiert, 11 werden entfernt.

Das folgende Kommando liefert eine Liste aller installierten Pakete.

Dabei wird in der ersten Spalte jeweils die Paketgruppe bzw.

Paketquelle angegeben (%20s, *Source* mit einer Spaltenbreite von 20 Zeichen), in der zweiten Spalte der Paketname.

```
user$ aptitude search ~i -F "%20s %p"  
universe/text      a2ps  
gnome            accountsservice  
utils             acl  
admin             acpi-support  
...
```

Mit [egrep](#) filtern Sie diejenigen Pakete aus dem Ergebnis, die aus den Ubuntu-Paketquellen `universe` oder `multiverse` stammen.

```
user$ aptitude search ~i -F "%20s %p" | egrep "universe|multiverse"  
universe/text      a2ps  
universe/gnome     adwaita-icon-theme-full
```

universe/misc        chkrootkit

...

### **arp-scan [optionen] [hosts/ipadressen/netz]**

arp-scan sendet ARP-Pakete (*Address Resolution Protocol*) an die angegebenen Hosts oder IP-Adressen und zeigt die resultierenden Antworten an. arp-scan bietet damit eine äußerst schnelle Möglichkeit, um herauszufinden, an welchen Hosts bzw. IP-Adressen aktive Geräte laufen und welche MAC-Adressen diese aufweisen. Der Adressbereich kann wie üblich auch in der Form 10.17.0.0/16 angegeben werden.

Ähnliche Funktionen, wenn auch auf einer technisch vollkommen anderen Basis, bietet das Kommando [nmap](#). Kurz gesagt ist [nmap](#) langsamer, aber gründlicher.

- **-f dateiname** bzw. **--file=dateiname**  
liest die Hosts oder IP-Adressen aus der angegebenen Datei.
- **-I schnittstelle** bzw. **--interface=schnittstelle**  
wählt explizit die Netzwerkschnittstelle an, über die die ARP-Pakete versendet werden sollen. Wenn diese Option nicht angegeben wird, verwendet arp-scan einfach die erste Schnittstelle, wobei die Loopback-Schnittstelle ignoriert wird.
- **-l** bzw. **--localnet**  
durchsucht das gesamte lokale Netzwerk. Die Angabe von Hosts oder IP-Adressen entfällt dann.

### **Beispiel**

Mit dem folgenden arp-scan-Kommando wird ein Netzwerk-Scan für die IP-Adressen 10.0.0.\* durchgeführt:

```
root# arp-scan --interface=enp0s3 10.0.0.0/24
Interface: enp0s3, datalink type: EN10MB (Ethernet)
Starting arp-scan 1.9 with 256 hosts
10.0.0.9      00:16:b6:9d:ff:4b      Cisco-Linksys
10.0.0.22     b8:27:eb:11:44:2e      Raspberry Pi Foundation
10.0.0.39     ac:87:a3:1e:4a:87      (Unknown)
...
7 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9: 256 hosts scanned in 1.825 seconds
(140.27 hosts/sec). 7 responded
```

## **at** [optionen] zeitpunkt

Mit at geben Sie an, dass ein oder mehrere Kommandos zu einem späteren Zeitpunkt ausgeführt werden sollen. Die Kommandos werden im Anschluss an den at-Befehl interaktiv eingegeben; die Eingabe endet mit **[Strg]+D**. Sofern der Job eine Standardausgabe produziert, wird die Ausgabe nach der Erledigung des Auftrags per E-Mail an den Benutzer versendet, der at ausgeführt hat. Das setzt voraus, dass auf dem Rechner */usr/sbin/sendmail* installiert ist.

### **Zeitangabe**

Für die Zeitangabe gibt es eine Menge Varianten, von denen im Folgenden die wichtigsten anhand von Beispielen vorgestellt werden:

- **3:45 [tomorrow]**  
führt die Kommandos um 3:45 Uhr aus – und zwar heute, wenn es früher ist, oder morgen, wenn 3:45 Uhr bereits vorbei ist. Mit dem nachgestellten tomorrow können Sie explizit einen Zeitpunkt am nächsten Tag angeben.
- **[16:30] [2020-]12-31**  
führt die Kommandos am angegebenen Tag und zum angegebenen Zeitpunkt aus. Die Jahres- und die Zeitangabe sind optional. Wenn diese Daten fehlen, wird das aktuelle Jahr und die aktuelle Zeit verwendet. at 11-30 bedeutet also: am 30. November dieses Jahres zur gleichen Uhrzeit, zu der das at-Kommando ausgeführt wurde.
- **noon | teatime | midnight**  
entspricht den Zeitangaben 12:00 Uhr, 16:00 Uhr und 0:00 Uhr.
- **now**  
führt die Kommandos sofort aus. now wird oft mit + xxx kombiniert.
- **... + 2 days**  
fügt der vorangegangenen Zeitangabe zwei Tage hinzu. Zulässige

Zeiteinheiten sind `minutes`, `hours`, `days` und `weeks`. Die grammatisch fragwürdige Angabe `at now + 1 hours` bedeutet somit: in einer Stunde. (Zeiteinheiten sind immer im Plural anzugeben.)

## Optionen

- `-b`  
führt das Kommando aus, sobald das System eine geringe Auslastung hat. Die Zeitangabe entfällt bei dieser Option. `at -b` entspricht dem Kommando [batch](#).
- `-c n`  
gibt an, welche Kommandos durch den Job mit der angegebenen Nummer ausgeführt werden. Die Jobnummern können Sie mit [atq](#) ermitteln.
- `-d` oder `-r n`  
entfernt den durch seine Nummer angegebenen Job aus der Liste der vorgemerkt Aufträge. `at -d` bzw. `at -r` entspricht [atrm](#).
- `-f datei`  
liest die auszuführenden Kommandos aus der angegebenen Datei, nicht aus der Standardeingabe.
- `-l`  
listet alle Jobs auf. `at -l` entspricht dem Kommando [atq](#).
- `-m`  
Immer eine Mail senden, auch wenn der Job keine Ausgabe liefert.
- `-M`  
Keine Mail nach Vollendung des Jobs senden.
- `-qx`  
ordnet den Job der Warteschlange `x` zu. Dabei ist `x` ein Klein- oder Großbuchstabe. Die Buchstaben `a` und `b` werden für gewöhnliche at-Jobs sowie für [batch](#)-Jobs verwendet. Je höher der Warteschlangenbuchstabe gewählt wird, desto höher ist der [nice](#)-

Wert, mit dem der Job ausgeführt wird. -qz eignet sich also für Jobs mit sehr geringer Dringlichkeit.

- -t YYYYMMDDhhmm

führt den Job zur angegebenen Zeit aus. Die Option -t ist eine Alternative zu den anderen Formen der Zeitangabe.

## Beispiel

Es wurden Änderungen an der Konfiguration des Webservers durchgeführt, die erst mit einem Neustart wirksam werden. Um zu vermeiden, dass gerade aktive Benutzer der Website Session-Daten verlieren, soll der Webserver aber nicht sofort, sondern erst am nächsten Tag um 1 Uhr morgens neu gestartet werden. Der Service-Name apache2 gilt dabei für Debian und Ubuntu. Unter Fedora und RHEL geben Sie stattdessen httpd an.

```
root# at 1:00
service apache2 restart
<Strg>+<D>
```

### atq

atq listet alle Jobs auf, die mit [at](#) für die Ausführung zu einem zukünftigen Zeitpunkt eingerichtet wurden. Wenn atq von root ausgeführt wird, zeigt es die Jobs aller Benutzer an, andernfalls nur eigene Jobs.

### atrm n

atrm löscht den durch seine Nummer angegebenen Job. Die Nummern aller Jobs, die für eine zukünftige Ausführung eingerichtet wurden, ermitteln Sie mit [atq](#).

### 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.

Anstelle von attr sollten Sie die Kommandos getfattr bzw. setfattr vorziehen. attr steht nur aus Kompatibilitätsgründen zu IRIX SGI

zur Verfügung.

```
avconv [inopts] [-i infile] [outopts] outfile
```

Das Kommando avconv aus dem Paket libav-tools ersetzt auf manchen Distributionen das populäre Kommando [ffmpeg](#). avconv ist ein Fork von [ffmpeg](#). Die Syntax beider Kommandos ist nahezu identisch und wird bei [ffmpeg](#) beschrieben.

```
awk [optionen] 'code' textdateien  
awk [optionen] -f codedatei textdateien
```

awk ist kein simples Kommando, sondern eine eigene Programmiersprache, die bei der Verarbeitung und Auswertung strukturierter Texte hilft. Sie können damit z.B. eine Textdatei nach Schlüsselwörtern durchsuchen und aus dem nachfolgenden Textabschnitt eine Textspalte extrahieren. Den Programmcode geben Sie wahlweise direkt ein oder lesen ihn mit der Option -f aus einer Codedatei. awk verarbeitet dann alle als Parameter übergebenen Textdateien zeilenweise und schreibt das Ergebnis in die Standardausgabe.

Der von awk zu verarbeitende Programmcode folgt dieser vereinfachten Syntax:

```
/suchmuster1/ {aktionen}  
/suchmuster2/ {weitere aktionen}  
bedingung      {noch mehr aktionen}
```

Für den Aufbau des Suchmusters gelten dieselben Regeln, die beim Kommando [grep](#) beschrieben sind. Mehrere Aktionen werden durch Semikola getrennt.

Wenn das Suchmuster fehlt, gilt die Aktion für jede Zeile der Textdatei. Wenn umgekehrt die Aktion fehlt, wird der durch das Muster erfasste Text einfach ausgegeben.

Eine umfassende Beschreibung von awk und seinen Möglichkeiten ist hier aus Platzgründen ausgeschlossen. Stattdessen beschränke ich mich auf einige Beispiele. Wenn Sie sich intensiver in awk einarbeiten möchten, bietet sich dieses Tutorial als Einführung an:

<https://www.grymoire.com/Unix/Awk.html> Der offizielle *User's Guide* geht mehr ins Detail, kann mit fast 500 Seiten Umfang aber Einsteiger abschrecken:

<https://www.gnu.org/software/gawk/manual/gawk.pdf>

## Beispiele

Als Ausgangspunkt für die folgenden Beispiele dient dieser Auszug aus */etc/passwd*:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
...
...
```

Das erste awk-Beispiel liefert nur die erste und die siebte Spalte aus */etc/passwd*, also den Login-Namen und die zugeordnete Shell. Die Option -F gibt an, dass die Spalten durch Doppelpunkte getrennt sind und nicht durch Leer- und Tabulatorzeichen:

```
user$ awk -F'::' '{print $1,$7}' /etc/passwd
root /bin/bash
bin /sbin/nologin
daemon /sbin/nologin
...
...
```

Das zweite Beispiel ermittelt das Heimatverzeichnis und die Shell von kofler:

```
user$ awk -F'::' '/kofler/ {print $6; print $7;}' /etc/passwd
/home/kofler
/bin/bash
```

Das dritte Beispiel gibt die Login-Namen aller Benutzer mit einer UID größer gleich 1000 aus:

```
user$ awk -F'::' '$3>=1000 {print $1}' /etc/passwd
nfsnobody
kofler
test
```

Das letzte Beispiel extrahiert aus der Ausgabe von ls -l die Dateigröße und den Dateinamen. Beachten Sie aber, dass das nur für Dateinamen funktioniert, die keine Leerzeichen enthalten.

```
user$ ls -l *.txt | awk '{print $5,$9}'
123 datei1.txt
213231 datei2.txt
...
...
```

```
aws [optionen] hauptkommando [subkommando] [parameter]
```

Das Kommando aws hilft bei der Administration diverser Cloud-Dienste von Amazon. Da diese Dienste häufig von Linux-Rechnern aus genutzt werden, kommen viele Administratoren mit aws in Kontakt.

Dieser Abschnitt bezieht sich auf die von Amazon empfohlene aws-Version 2. Stellen Sie mit `aws --version` sicher, dass Sie nicht die noch weit verbreitete Version 1 installiert haben! Beide aws-Versionen verwenden den gleichen Kommandonamen und sind weitgehend kompatibel zueinander. Es gibt aber auch einige Unterschiede:

<https://docs.aws.amazon.com/cli/latest/userguide/cliv2-migration.html>

## Installation und Inbetriebnahme

aws ist standardmäßig nicht installiert und steht in der Regel auch nicht als Paket Ihrer Distribution zur Verfügung. Stattdessen müssen Sie zur Installation die folgenden Kommandos ausführen:

```
user$ sudo apt/dnf/yum/zypper install curl  
user$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" \  
      -o "awscliv2.zip"  
user$ unzip awscliv2.zip  
user$ sudo ./aws/install
```

Die ZIP-Datei enthält eine minimale Python-3-Distribution mit allen erforderlichen Paketen. Damit gibt es weder Abhängigkeiten von Python-Paketen Ihrer Distribution noch Konflikte aufgrund abweichender Versionsnummern. aws wird in das Verzeichnis `/usr/local/bin` installiert. Falls dieses Verzeichnis nicht Teil von PATH ist, müssen Sie den Pfad explizit angeben.

```
root# aws --version  
aws-cli/2.0.48 Python/3.7.3 Linux/5.4.0-26-generic exe/x86_64.ubuntu.20
```

Die Verbindung zu den Amazon-Cloud-Diensten erfolgt verschlüsselt. Bevor Sie im Terminal das Kommando aws für die weitere Nutzung konfigurieren können, müssen Sie sich auf der AWS-Seite **IDENTITY & ACCESS MANAGEMENT (IAM)** einen Benutzer und

eine Gruppe einrichten. Die Gruppe verbinden Sie mit der Policy AmazonS3FullAccess und mit dem neuen Benutzer. Sie finden die AWS-Seite unter:

<https://console.aws.amazon.com/iam>

Im Terminal führen Sie nun `aws configure` aus und geben dabei die *Access Key ID* und den *Secret Access Key* an, die Sie beim Einrichten des Benutzers erhalten haben. `aws` speichert diese Daten und die weiteren Optionen in `.aws/credentials` und `.aws/config`.

```
root# aws configure
AWS Access Key ID [None]: AKxxxxxxxxx
AWS Secret Access Key [None]: ZRxxxxxxxxxxxxxxxxxxxxxx
Default region name [None]: eu-central-1
Default output format [None]:
```

## Syntax und Online-Hilfe

Üblicherweise übergeben Sie an das Kommando `aws` ein Hauptkommando, das den gewünschten AWS-Dienst auswählt, sowie ein Subkommando, das spezifisch für den betreffenden Dienst gilt. Beispielsweise listet `aws s3 ls` Ihre Buckets im Dienst S3 auf. Ich beschränke mich an dieser Stelle aus Platzgründen auf einige wenige Kommandos für die Dienste EC2 (*Elastic Compute Cloud*) und S3 (*Simple Storage Service*).

Zu `aws` gibt es keine [man](#)-Seite. Einen Syntaxüberblick gibt stattdessen `aws help`. Einen Überblick zu den Kommandos für einen bestimmten Dienst gibt `aws <dienst> help`, also z.B. `aws ec2 help`. Details zu einem Subkommando verrät `aws <dienst> <subkommando> help`, also z.B. `aws s3 cp help`.

## S3-Administration

- `aws s3 ls [pfad]`

listet ohne weiteren Parameter alle S3-Buckets auf bzw. zeigt den Inhalt eines Bucket-Verzeichnisses oder einer dort befindlichen Datei an. Pfadangaben erfolgen in der Form

`s3://<bucketname>/<verzeichnis>/<datei>`, also z.B. `s3://my-bucket/my-dir/my-file`.

- `aws s3 cp pfade1 pfade2`  
kopiert eine lokale Datei in ein S3-Bucket bzw. von dort zurück in das lokale Dateisystem. Sie können auch eine S3-Datei an einen anderen S3-Ort kopieren. Mit der Option `--recursive` können Sie Verzeichnisse rekursiv kopieren. Zum Testen übergeben Sie `--dryrun`: Das Kommando zeigt dann an, was es machen würde, verändert aber nichts.  
Falls Sie S3 zur Datensicherung verwenden, ist es empfehlenswert, ausschließlich zuvor selbst verschlüsselte Dateien auf S3 hochzuladen. In der Vergangenheit ist es immer wieder zu Konfigurationsfehlern gekommen, sodass vermeintlich private Buckets doch öffentlich zugänglich waren. Und es kann auch nicht ausgeschlossen werden, dass sich Nachrichtendienste Zugriff verschaffen können.
- `aws s3 mv pfade1 pfade2`  
wie oben, aber verschiebt die Datei bzw. benennt eine bereits im S3-Speicher befindliche Datei um.
- `aws s3 sync pfade1 pfade2`  
synchronisiert zwei Verzeichnisse: ein lokales und eines im S3-Speicher. Das geht schneller als `aws s3 cp`, weil unveränderte Dateien nicht nochmals übertragen werden. Im Quellpfad gelöschte Dateien bleiben im Zielpfad erhalten. Wenn Sie das nicht wollen, übergeben Sie die Option `--delete`.
- `aws s3 mb newbucket`  
erzeugt einen neuen Bucket, wobei Sie den Namen in der üblichen Nomenklatur angeben müssen (also `s3://<bucketname>`). Der Name muss weltweit eindeutig sein.  
Mit der Option `--region` können Sie abweichend von der bei `aws configure` voreingestellten Region angeben, in welchem Rechenzentrum der Bucket eingerichtet wird.  
Der neue Bucket ist standardmäßig nicht öffentlich zugänglich. Zugriffsrechte und Zugriffsregeln (ACLs, also *Access Control Lists*)

müssen Sie über die AWS-Webkonsole einstellen.

- `aws s3 rb bucket`  
löscht den angegebenen Bucket. Die Operation gelingt nur, wenn der Bucket leer ist. Andernfalls können Sie das Löschen mit `--force` erzwingen.

## EC2-Administration

Während die aws-Kommandos zur S3-Administration einfach anzuwenden sind, gilt dies für EC2 leider nicht: Allein die Anzahl der Subkommandos – es sind über 400 – ist abschreckend. Viele Kommandos liefern bzw. erwarten zudem JSON-Datenstrukturen, sind also nur zur Script-Programmierung, aber nicht für die Anwendung durch Menschen gedacht. Als Startpunkt können Sie `aws ec2 describe instances` ausprobieren. Das Kommando liefert eine Auflistung aller Ihrer bereits eingerichteten Instanzen als JSON-Dokument:

```
root# aws ec2 describe-instances
{
    "Reservations": [
        {
            "Groups": [],
            "Instances": [
                {
                    "AmiLaunchIndex": 0,
                    "ImageId": "ami-02fc41eea1852",
                    ...

```

## Beispiel

Die beiden folgenden Kommandos richten einen neuen Bucket ein und kopieren dorthin alle Dateien aus dem `/etc`-Verzeichnis, die für den aktiven Benutzer lesbar sind:

```
user$ aws s3 mb s3://info.kofler.etcbak
user$ aws s3 sync /etc s3://info.kofler.etcbak
```

## B

### **badblocks device**

badblocks führt eine Low-Level-Überprüfung des Datenträgers durch. Das ist nur zweckmäßig, wenn Sie den Verdacht haben, dass Ihre Festplatte oder SSD defekt sein könnte. Wenn Sie vorhaben, auf der Festplatte(npartition) ein ext-Dateisystem einzurichten, ist es besser, die Überprüfung mit `mke2fs -c` im Rahmen der Formatierung durchzuführen.

badblocks zerstört normalerweise keine Daten und führt nur einen Read-only-Test durch. Gründlichere Tests können Sie mit den Optionen `-n` oder `-w` durchführen:

- `-n`  
führt auch Schreibtests durch, stellt den ursprünglichen Inhalt der Datenblöcke aber wieder her. Die Option kann nicht verwendet werden, wenn der Datenträger bzw. die Partition gerade verwendet wird, also mit [mount](#) in den Verzeichnisbaum eingebunden ist.
- `-w`  
führt Schreibtests mit Bitmustern durch. Vorsicht: Der Inhalt des Datenträgers bzw. der Partition wird dadurch gelöscht!

### **basename zeichenkette [endung]**

basename liefert den Dateinamen des übergebenen Pfads. basename /etc/X11/Xmodmap führt also zum Ergebnis *Xmodmap*. Wenn als zusätzlicher Parameter eine Dateiendung angegeben wird, so wird diese Dateiendung (falls vorhanden) aus dem Dateinamen entfernt.

### **Beispiel**

basename eliminiert den Pfad und die Kennung *.jpg*. Übrig bleibt `white`.

```
user$ basename /home/kofler/Bilder/Wallpapers/white.jpg .jpg  
white
```

### **batch**

`batch` merkt einen Job-Auftrag für die spätere Ausführung vor. Wie bei `at` werden die auszuführenden Kommandos anschließend interaktiv eingegeben. Die Eingabe wird mit `[Strg]+[D]` abgeschlossen. Anders als bei `at` entfällt die Zeitangabe, wann der Job ausgeführt werden soll. Vielmehr wartet der Hintergrunddämon `atd` so lange, bis das System eine geringe Auslastung hat. Die Schwelle ist mit einem `load`-Durchschnitt von 1,5 definiert. Wenn der Job eine Ausgabe liefert, wird diese per Mail an den Benutzer gesendet. Das erfordert, dass auf dem Rechner `/usr/sbin/sendmail` zur Verfügung steht.

`batch` verarbeitet weder Optionen noch Parameter. Wenn Sie den Jobauftrag aus einer Datei lesen möchten, führen Sie `at -qb -f datei now` aus.

### Beispiel

Mit der Ausführung des Backup-Scripts wird begonnen, sobald die Systemauslastung gering ist:

```
user$ batch  
./backup-script  
<Strg>+<D>  
bg [prozess]
```

Das Shell-Kommando `bg` setzt einen unterbrochenen Prozess im Hintergrund fort. Wenn keine Prozessnummer angegeben wird, gilt `bg` für den zuletzt mit `[Strg]+[Z]` unterbrochenen Prozess.

Andernfalls muss der Prozess durch seinen Namen oder durch die bash-interne Jobnummer (nicht durch die PID!) angegeben werden.

```
blkid [optionen] [device]
```

`blkid` liefert Informationen über das durch den Device-Namen angegebene Block-Device, also z.B. über Festplattenpartitionen, Logical Volumes oder RAID-Geräte. `blkid` gibt hierfür den Dateisystemtyp, die UUID und den Namen (Label) des Dateisystems an. Wenn die Device-Angabe beim Aufruf von `blkid` fehlt, liefert `blkid` diese Informationen für alle aktiven Partitionen und Logical

Volumes. Durch die Angabe von Optionen kann gezielt nach bestimmten Dateisystemen gesucht werden:

- **-k**  
listet alle vom Kernel unterstützten Dateisystemtypen auf.
- **-L *label***  
sucht nach Dateisystemen mit dem angegebenen Namen.
- **-n *fstyp1, fstyp2, fstyp3***  
sucht nach Dateisystemen, die den angegebenen Typen entsprechen.
- **-t *name=wert***  
sucht nach Dateisystemen, die dem Suchkriterium entsprechen. Zulässige Kriterien sind **TYPE** (Dateisystemtyp), **LABEL** (Name des Dateisystems), **PARTLABEL** (Name der Partition), **UUID** (UUID des Dateisystems) und **PARTUUID** (UUID der Partition). **blkid -t TYPE=ext4** entspricht **blkid -n ext4** und liefert eine Liste aller ext4-Dateisysteme.
- **-U *uuid***  
sucht nach Dateisystemen mit der angegebenen UUID.

## Beispiele

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"
```

Die Suche nach einer Partition mit dem Label `lvm2` liefert als Ergebnis `/dev/sdb2`:

```
root# blkid -t PARTLABEL=lvm2
/dev/sdb2: PARTLABEL="lvm2" PARTUUID="371cc374-847f-407a-bb5b-7ff015722383"
```

## bluetoothctl

Auf Desktop-Systemen erfolgt die Bluetooth-Konfiguration normalerweise über grafische Benutzeroberflächen. Wenn diese nicht zur Verfügung stehen (z.B. auf einem Raspberry Pi, an den kein Monitor angeschlossen ist) oder nicht funktioniert, kann die Konfiguration mit `bluetoothctl` auch im Textmodus erfolgen. Dazu

starten Sie mit `bluetoothctl` eine interaktive Sitzung und führen dann Kommandos aus, um das Bluetooth-Gerät zu finden und zu koppeln.

Innerhalb einer `bluetoothctl`-Sitzung können Sie die im Folgenden beschriebenen Kommandos ausführen. Wichtig ist, dass Sie bei Bluetooth-Geräten, die Sie neu konfigurieren wollen, immer wieder auf den Bluetooth-Knopf (Pairing-Knopf) drücken, damit das Gerät so der Umwelt signalisiert, dass es zum Verbindungsauftbau bereit ist. Bei vielen Geräten blinkt dann eine blaue Leuchtdiode. Wenn es keinen derartigen Knopf gibt, können Sie auch versuchen, das Gerät aus- und wieder einzuschalten.

- `agent on/off`  
(de)aktiviert den Bluetooth-Agenten. Diese Software-Komponente kümmert sich um die Autorisierung neuer Geräte und erlaubt beispielsweise die Tastatureingabe eines Paarungs-Codes.
- `connect xx:xx:xx`  
stellt die Verbindung her. (Das funktioniert nur, wenn vorher `pair` und `trust` ausgeführt wurden!) Wenn alles klappt, lautet die Reaktion *connection successful*. Das Gerät kann jetzt verwendet werden!
- `exit` oder `quit`  
beendet die `bluetoothctl`-Sitzung.
- `info xx:xx:xx`  
zeigt den Verbindungsstatus und diverse weitere Informationen zum Gerät an.
- `pair xx:xx:xx`  
initiiert den Verbindungsauftbau zu einem Gerät. Bei einer Tastatur werden Sie nun dazu aufgefordert, einen mehrstelligen Code einzutippen. Vergessen Sie nicht, die Eingabe mit  abzuschließen! Die erfolgreiche Kuppelung erkennen Sie an der

Meldung *pairing successful*. Bei Geräten ohne Eingabemöglichkeit (Maus, Headset, Lautsprecher etc.) gelingt das Pairing ohne Codeeingabe.

- **pairable on/off**  
aktiviert/deaktiviert den Kuppelungsmodus.
- **scan on/off**  
(de)aktiviert den Scan-Modus. Im Scan-Modus listet das Programm alle Geräte auf. Dieser Vorgang kann geraume Zeit dauern, einzelne Geräte werden dabei immer wieder angezeigt. Wenn Sie das gewünschte Gerät gefunden haben, schalten Sie den Modus mit `scan off` wieder aus.
- **trust xx:xx:xx**  
vertraut dem angegebenen Gerät.

`bluetoothctl` speichert die Bluetooth-Konfiguration für ein bestimmtes Gerät dauerhaft in `/var/lib/bluetooth/id1/id2/info`. Dabei ist `id1` der ID-Code des Bluetooth-Controllers und `id2` der ID-Code des Bluetooth-Geräts.

`bluetoothctl` ist eigentlich für die interaktive Benutzung gedacht. Um ein einzelnes Kommando unmittelbar auszuführen, können Sie wie folgt vorgehen:

```
root# echo -e "connect FC:58:FA:A0:4D:E7\nquit" | bluetoothctl
```

## Beispiel

Die folgenden Zeilen zeigen etwas gekürzt die Ein- und Ausgaben zur Konfiguration einer Bluetooth-Tastatur:

```
user$ bluetoothctl
[bluetooth]# agent on
[bluetooth]# pairable on
[bluetooth]# scan on
Discovery started
[CHG] Controller 00:1A:7D:DA:71:13 Discovering: yes
[NEW] Device 70:10:00:1A:92:20 70-10-00-1A-92-20
[CHG] Device 70:10:00:1A:92:20 Name: Bluetooth 3.0 Keyboard
...
[bluetooth]# scan off
[bluetooth]# pair 70:10:00:1A:92:20
Attempting to pair with 70:10:00:1A:92:20
[CHG] Device 70:10:00:1A:92:20 Connected: yes
[agent] PIN code: 963064
```

```
[CHG] Device 70:10:00:1A:92:20 Paired: yes
Pairing successful
[bluetooth]# trust 70:10:00:1A:92:20
[bluetooth]# connect 70:10:00:1A:92:20
[bluetooth]# info 70:10:00:1A:92:20
Device 70:10:00:1A:92:20
    Name: Bluetooth 3.0 Keyboard
    Paired: yes
    Trusted: yes
    Connected: yes
...
[bluetooth]# exit
```

### boltctl kommando

boltctl steuert über die Thunderbolt-Schnittstelle angeschlossene Geräte. Thunderbolt kommt aktuell vor allem in Notebooks mit Intel-CPUs zum Einsatz. In Zukunft wird die Verbreitung vermutlich größer werden, weil Thunderbolt Teil des neuen USB4-Standards ist.

- **authorize device**  
erlaubt einmalig die Nutzung eines Thunderbolt-Geräts. Eine Liste aller angeschlossenen Geräte ermitteln Sie mit `boltctl list`. Eine dauerhafte Erlaubnis geben Sie mit `boltctl enroll`.
- **config key**  
`config key newvalue`  
liest Konfigurationsoptionen aus bzw. verändert diese (erfordert root-Rechte). Eine Liste aller Optionen liefert `boltctl config --describe`.
- **enroll [--policy auto|manual|default] device**  
erlaubt dauerhaft die Kommunikation des Geräts über die Thunderbolt-Schnittstelle. Eine explizite Erlaubnis ist nur für Geräte erforderlich, die via Thunderbolt den PCI-Bus nutzen (z.B. für eine Docking Station), und nur dann, wenn im BIOS/EFI strenge Thunderbolt-Sicherheitsrichtlinien eingestellt sind.  
auto bedeutet, dass das Gerät in Zukunft ohne Rückfragen verwendet werden kann. manual erfordert bei jedem neuerlichen Anschluss eine Bestätigung mittels `boltctl authorize`. default übernimmt die Vorgaben des Thunderbolt-Dämons (üblicherweise auto).

- `forget device`  
entfernt die ID eines Geräts aus der Enroll-Datenbank und macht somit `boltctl enroll` rückgängig.
- `info device`  
zeigt Detailinformationen zum angegebenen Gerät an.
- `list -a`  
listet alle angeschlossenen Geräte auf.
- `monitor`  
verfolgt und protokolliert, welche Geräte angeschlossen bzw. entfernt werden, bis das Kommando mit `Strg + C` beendet wird.
- `power`  
aktiviert einen im Ruhezustand befindlichen Thunderbolt-Controller.

### **break [n]**

Das bash-Kommando `break` bricht in Shell-Skripts 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.

### **btrfs kommando**

`btrfs` ist das zentrale Administrationskommando für `btrfs`-Dateisysteme. Sofern keine Doppeldeutigkeiten auftreten, können Sie die im Folgenden beschriebenen Kommandos abkürzen und damit eine Menge Tipparbeit sparen. `btrfs fi sh` entspricht also `btrfs filesystem show`.

Die Administration von `btrfs`-Dateisystemen ist außergewöhnlich komplex. Im Folgenden sind nur die wichtigsten Kommandos zusammengefasst. Es fehlt hier aber der Platz, die `btrfs`-Grundlagen im Detail zu beschreiben.

- `balance start|pause|resume|cancel|status mountverzeichnis`  
 verteilt die Daten gleichmäßig über alle Devices des Dateisystems. Dieser Vorgang dauert sehr lange und wird im Hintergrund durchgeführt. Er kann mit `btrfs balance pause` unterbrochen, mit `btrfs balance resume` wieder aufgenommen bzw. mit `btrfs balance cancel` ganz gestoppt werden. `btrfs balance status` gibt Auskunft darüber, wie weit der Balance-Vorgang fortgeschritten ist.
- `device add devicename mountverzeichnis`  
 fügt einem aktiven `btrfs`-Dateisystem ein weiteres Device hinzu und vergrößert so dessen Datenpool.
- `device delete devicename mountverzeichnis`  
 entfernt ein Device aus dem Datenpool eines `btrfs`-Dateisystems. Die auf dem Device enthaltenen Daten werden dabei auf andere Devices des Dateisystems übertragen, was bei großen Dateisystemen natürlich lange dauert. Wenn Sie ein defektes (nicht mehr verfügbares) Device entfernen möchten, geben Sie als Device-Namen `missing` an. Bei RAID-1-Systemen müssen Sie vorher ein Ersatz-Device hinzufügen und `btrfs filesystem balance` ausführen!
- `filesystem df verzeichnis`  
 zeigt detaillierte Informationen darüber an, wie viel Platz für Daten, Metadaten und Systemdaten reserviert ist und wie viel Platz davon bereits genutzt wird. Das Kommando liefert allerdings keine Informationen darüber, wie viel nicht reservierter Speicherplatz noch zur Verfügung steht, was die richtige Interpretation der Daten schwierig macht.
- `filesystem resize [+/-]size mountverzeichnis`  
 vergrößert oder verkleinert das Dateisystem im laufenden Betrieb. Die neue Größe kann wahlweise absolut oder relativ angegeben werden. Dabei sind die Kürzel `k`, `m` und `g` für KiB, MiB und GiB erlaubt. Als Größenangabe ist auch `max` zulässig – dann

nutzt das Dateisystem in Zukunft die gesamte Größe des zugrunde liegenden Devices. Denken Sie daran, dass Sie die Größe des zugrunde liegenden Devices gegebenenfalls *vor* der Vergrößerung des Dateisystems, aber *nach* einer Verkleinerung anpassen müssen!

Anstelle von `resize` ist es bei btrfs-Dateisystemen einfacher, ein zusätzliches Device hinzuzufügen (`btrfs add device`).

- `filesystem show [verzeichnis/device]`  
zeigt Basisinformationen zu btrfs-Devices an. Das Kommando gibt unter anderem Auskunft darüber, wie groß die Devices sind und wie viel noch nicht für Daten, Metadaten oder Systemdaten reservierter Platz auf den Devices zur Verfügung steht.
- `filesystem sync verzeichnis`  
synchronisiert den Inhalt des Dateisystems bzw. eines seiner Verzeichnisse.
- `scrub start|cancel|resume|status verzeichnis/device`  
steuert den Scrub-Vorgang. Durch `scrub start` werden alle Dateien des Dateisystems gelesen, und ihre Integrität wird anhand von Prüfsummen kontrolliert. Dieser Prozess läuft im Hintergrund. Sein Fortschritt kann mit `scrub status` überprüft werden. `scrub cancel` unterbricht den Scrub-Vorgang, `scrub resume` setzt ihn wieder fort.
- `subvolume create verzeichnis/name`  
erzeugt das Subvolume `name`. Als Verzeichnis muss das [mount](#)-Verzeichnis oder ein Verzeichnis innerhalb des btrfs-Dateisystems angegeben werden. Wenn kein Verzeichnis angegeben wird, verwendet btrfs das aktuelle Verzeichnis.
- `subvolume delete verzeichnis/name`  
löscht das angegebene Subvolume oder den angegebenen Snapshot. Der belegte Speicher wird allerdings erst nach und nach durch einen Hintergrundprozess freigegeben.

- `subvolume list mountverzeichnis`  
listet alle Subvolumes und Snapshots auf. Als Verzeichnis wird das [mount](#)-Verzeichnis des Dateisystems angegeben. Die Ausgabe kann durch mehrere Optionen beeinflusst werden. Als empfehlenswert hat sich die Kombination `-a -p -t` herausgestellt. Sie bewirkt, dass der Ort des Subvolumes innerhalb der Verzeichnishierarchie exakt angegeben wird, dass zu jedem Subvolume bzw. Snapshot auch das Parent-Subvolume angegeben und das Ergebnis als Tabelle formatiert wird.
- `subvolume set-default id mountverzeichnis`  
bestimmt, welches Subvolume bzw. welcher Snapshot in Zukunft standardmäßig von [mount](#) verwendet wird. Die Volume-ID können Sie mit `btrfs subvolume list` ermitteln.
- `subvolume snapshot [-r] quellverzeichnis zielverzeichnis/name`  
erzeugt den Snapshot *name*. Als Quellverzeichnis muss das [mount](#)-Verzeichnis des Dateisystems oder das Verzeichnis eines vorhandenen Subvolumes oder Snapshots angegeben werden. Das Zielverzeichnis muss sich innerhalb des `btrfs`-Dateisystems befinden.  
Der Snapshot ist normalerweise veränderlich (d.h., das Dateisystem wird in zwei Zweige gespalten, die beide getrennt voneinander veränderlich sind). Wenn Sie für Backups einen Read-only-Snapshot wünschen, verwenden Sie die Option `-r`.

## Beispiel

`btrfs subvolume list` zeigt die in einem (open)SUSE-System standardmäßig eingerichteten Subvolumes sowie alle Snapshots:

```
root# btrfs subvolume list / -p -a -t
ID   gen   parent  top level  path
-   ---   -----  -----
257   106     5      5      <FS_TREE>/@
260   106    257    257      <FS_TREE>/@/boot/grub2/i386-pc
261   106    257    257      <FS_TREE>/@/boot/grub2/x86_64-efi
262   1305   257    257      <FS_TREE>/@/home
...
268   106    257    257      <FS_TREE>/@/var/lib/libvirt/images
269   106    257    257      <FS_TREE>/@/var/lib/mailman
```

```
270 106 257 257      <FS_TREE>/@/var/lib/mariadb
271 106 257 257      <FS_TREE>/@/var/lib/mysql
...
258 1158 257 257     <FS_TREE>/@/.snapshots
259 1310 258 258     <FS_TREE>/@/.snapshots/1/snapshot
283 106 258 258     <FS_TREE>/@/.snapshots/2/snapshot
304 669 258 258     <FS_TREE>/@/.snapshots/20/snapshot
...
```

Um im laufenden Betrieb ein Backup der MariaDB-Datenbankdateien zu erstellen, wird vorübergehend ein Read-only-Snapshot für das Subvolume `/var/lib/mariadb` erstellt:

```
root# btrfs subvolume snapshot -r /var/lib/mariadb/ /var/lib/mariadb-backup
Create a readonly snapshot of '/var/lib/mariadb/' in '/var/lib/mariadb-backup'
root# cd /var/lib/mariadb-backup
root# tar czf /home/kofler/mariadb.tgz .  (Backup erstellen)
root# cd                               (Verzeichnis verlassen)
root# btrfs subvolume delete /var/lib/mariadb-backup
Delete subvolume (no-commit): '/var/lib/mariadb-backup'
```

## Dokumentation

Die [man](#)-Seiten zu `btrfs` sind über mehrere Seiten verteilt, deren Namen sich jeweils aus `btrfs`, einem Bindestrich und einem Hauptkommando zusammensetzen. Details zu den diversen `btrfs`-scan-Kommandos können Sie daher mit `man btrfs-scan` nachlesen.

### **bunzip2** datei.bz2

`bunzip2` dekomprimiert eine zuvor mit [`bzip2`](#) komprimierte Datei. Dabei wird automatisch die Kennung `.bz2` im Dateinamen entfernt. `bunzip` ist ein Link auf [`bzip2`](#), wobei automatisch die Option `-d` aktiviert ist.

### **bzip2** datei

`bzip2` komprimiert die angegebenen Dateien und fügt ihnen die Endung `.bz2` hinzu. Das Kommando liefert im Regelfall um 20 bis 30 Prozent kleinere Dateien als durch [`gzip`](#) komprimierte Dateien. Allerdings ist die zum Komprimieren erforderliche Rechenzeit bei `bzip2` deutlich länger.

- `-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.

- **-c** bzw. **--stdout** oder **--to-stdout**  
lässt die zu (de)komprimierende Datei unverändert und leitet das Ergebnis auf die Standardausgabe 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](#)).

### Beispiel

Das erste Kommando komprimiert *datei.eps*. Der neue Dateiname lautet nun *datei.eps.bz2*. Das zweite Kommando stellt die ursprüngliche Datei wieder her.

```
user$ bzip2 datei.eps  
user$ bunzip2 datei.eps.bz2
```

# C

## **cadaver** [optionen] [dateien]

Das Kommando mit dem merkwürdigen Namen `cadaver` aus dem gleichnamigen Paket ist ein WebDAV-Client. Sie können damit ähnlich wie mit [ftp](#) Dateien herunter- oder hochladen. Selbst die Namen der meisten Kommandos sind gleichlautend, also `get`, `mget`, `put`, `mput` usw. (siehe [ftp](#)). Der entscheidende Unterschied besteht darin, dass das auf HTTP(S) basierende Protokoll WebDAV zum Einsatz kommt.

### **Beispiel**

Im folgenden Kommando wird zuerst eine WebDAV-Verbindung zum HTTP-Server mit der IP-Adresse 172.28.128.5 auf Port 8585 hergestellt. In das Verzeichnis `uploads` wird dann die Datei `test.php` hochgeladen.

```
user$ cadaver http://172.28.128.5:8585/uploads
dav:/uploads/> put test.php
  Uploading test.php to /uploads/test.php:
  Progress: ... 100,0% of 20 bytes succeeded.
dav:/uploads/> <Strg>+<D>
  Connection to 172.28.128.5 closed.
```

## **canonical-livepatch** [kommando]

Sofern das gleichnamige Snap-Paket installiert ist, steuert `canonical-livepatch` die Ubuntu-spezifischen Kernel-Livepatch-Funktionen.

- `config`  
richtet die Livepatch-Funktion ein bzw. verändert deren Konfiguration. Dieses Kommando erfordert einen Ubuntu-One-Account.
- `enable/disable`  
aktiviert bzw. deaktiviert das Livepatch-System.
- `refresh`  
überprüft, ob Updates zur Verfügung stehen, lädt diese

gegebenenfalls herunter und wendet sie auf den laufenden Kernel an. Das erfolgt in regelmäßigen Abständen automatisch.

- **status**  
zeigt den Status des Livepatch-Systems an.

## **Beispiel**

Auf diesem Server sind alle verfügbare Kernel-Patches bereits angewendet:

```
root# canonical-livepatch status
last check: 29 minutes ago
kernel: 4.15.0-72.81-generic
server check-in: succeeded
patch state: all applicable livepatch modules inserted
patch version: 71.1
case ausdruck in
  muster1 ) kommandos;;
  muster2 ) kommandos;;
  ...
esac
```

Das Konstrukt **case** der bash-Shell bildet in Scripts Mehrfachverzweigungen, wobei als Kriterium für die Verzweigung eine Zeichenkette angegeben wird – oft 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 schließenden runden Klammer ) und den beiden Strichpunkten folgen. Anschließend wird das Programm nach **esac** fortgesetzt.

## **Beispiel**

Im folgenden Listing wird **case** eingesetzt, um die übergebenen Parameter in Dateinamen und Optionen zu klassifizieren. Innerhalb einer Schleife wird jeder einzelne Parameter mit **case** analysiert. Wenn der Parameter mit einem Bindestrich beginnt, wird der

Parameter an das Ende der Variablen opt angefügt, andernfalls an das Ende von dat.

```
#!/bin/bash
for i do      # Schleife für alle übergebenen Parameter
    case "$i" in
        -* ) opt="$opt $i";;
        *  ) dat="$dat $i";;
    esac
done          # Ende der Schleife
echo "Optionen: $opt"
echo "Dateien:  $dat"
cat [optionen] [dateien]
```

cat zeigt den Inhalt der angegebenen Textdatei an. cat wird häufig auch dazu verwendet, mehrere Dateien zu einer größeren Datei zusammenzusetzen. Dazu muss die Standardausgabe mit > in eine Datei umgeleitet werden (siehe Beispiel). Wenn cat aus der Standardeingabe lesen soll, verzichten Sie auf die Nennung eines Dateinamens oder geben das Zeichen - an.

Bei längeren Texten sollten Sie statt cat das Kommando [less](#) verwenden: Damit können Sie zeilen- bzw. seitenweise durch den Text blättern.

Zu cat existiert auch die Variante [tac](#). Dieses Kommando gibt die Zeilen der Textdatei in umgekehrter Reihenfolge aus, also die letzte Zeile zuerst.

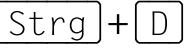
- -s  
reduziert mehrere leere Zeilen auf eine einzige leere Zeile.
- -T  
zeigt Tabulatorzeichen in der Form ^I an.
- -v  
zeigt nicht druckbare Zeichen in der ^xxx-Schreibweise an.

## 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 zweiten Beispiel wird die Standardeingabe in eine neue Datei umgeleitet.

Nachdem Sie das Kommando mit  bestätigt haben, werden alle weiteren Eingaben in die neue Datei geschrieben.  beendet die Eingabe. In dieser Form kann cat also dazu verwendet werden, ohne einen Editor eine neue Textdatei zu erzeugen.

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

## Varianten

Zu cat existieren zahllose Varianten:

- Das auf GitHub (<https://github.com/jingweno/ccat>) verfügbare Kommando ccat (*colorizing cat*) stellt Code-Dateien in gängigen Programmiersprachen farbig mit Syntaxhighlighting dar.
- Die Kommandos [more](#) und [less](#) ermöglichen es, zeilen- und Seitenweise durch den Text zu scrollen.
- Das Kommando [tac](#) gibt den Text in umgekehrter Reihenfolge aus, also die letzte Zeile zuerst.
- Die Kommandos [zcat](#) und [zless](#) ermöglichen es, auch komprimierte Textdateien direkt anzuzeigen.

**cd [verzeichnis]**

Das Shell-Kommando cd wechselt in das angegebene 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.

**certbot [kommando] [optionen]**

certbot hilft dabei, die von der Organisation *Let's Encrypt* kostenlos angebotenen Zertifikate auf dem lokalen Server einzurichten und in der Folge zu aktualisieren. certbot befindet sich bei vielen Distributionen im gleichnamigen Paket oder im Paket [python\[3\]-certbot](#). Wenn Sie weder in Ihrer Distribution noch in den Python-

Paketquellen ein aktuelles Paket finden, gibt es hier eine Anleitung zur manuellen Installation:

<https://certbot.eff.org>

certbot setzt die Installation von Python 3 samt diverser Zusatzpakete voraus, was in den vergangenen Jahren je nach Distribution immer wieder zu Problemen führte. Eine interessante Alternative ist in solchen Fällen das wesentlich schlankere Script [acme.sh!](#)

## Kommandos und Optionen

Mit certbot können Sie unter anderem die folgenden Kommandos ausführen. (Wenn Sie kein Kommando angeben, gilt automatisch run.)

- **certificates**  
listet alle installierten Zertifikate mit ihrem Ablaufdatum auf.
- **certonly**  
erzeugt oder erneuert die Zertifikate, verzichtet aber auf eine Webserver-Konfiguration.
- **renew**  
kontrolliert alle bereits installierten Zertifikate und aktualisiert alle, die in den nächsten 30 Tagen auslaufen. Eine Angabe der Zertifikate durch die Option -d ist nicht erforderlich. Sie können die Erneuerung von relativ frischen Zertifikaten mit der Option --force-renewal erzwingen.  
Es ist zweckmäßig, certbot renew täglich oder zumindest wöchentlich durch ein Cron-Script auszuführen.
- **run**  
erzeugt neue Zertifikate und installiert diese. Dabei wird überprüft, ob die Nameserver-Einträge der gewünschten Domänen auf den Host verweisen, auf dem das Kommando läuft. Die Zertifikate und diverse Konfigurationsdateien werden automatisch in `/etc/letsencrypt` gespeichert.

Einige Kommandos können Sie durch die folgenden Optionen ergänzen:

- **-a authenticator**  
wählt das gewünschte Verfahren aus, um zu überprüfen, dass Sie wirklich über die Domäne verfügen. Zur Auswahl stehen apache, standalone und webroot (siehe auch die Option `--webroot`).
- **--agree-tos**  
bringt zum Ausdruck, dass Sie mit den Nutzungsbedingungen (genau genommen den *ACME Server's Subscriber Agreement*) einverstanden sind.
- **--apache bzw. --nginx**  
richtet die Zertifikate für einen Apache- bzw. für einen Nginx-Webserver ein. Bei meinen Tests hat das nicht immer zuverlässig funktioniert. Gegebenenfalls müssen Sie die Apache- bzw. Nginx-Konfiguration anschließend manuell nachbearbeiten.
- **-d domäne1, domäne2, ...**  
gibt an, für welche Domänen (Hosts) die Zertifikate gelten sollen. Die Option kann auch mehrfach verwendet werden, also `-d domäne1 -d domäne2` etc.  
Auch bei Wildcard-Zertifikaten müssen Sie zumeist zwei Domänen angeben, einmal die Domäne an sich und das zweite Mal `*.domäne` für alle Subdomänen – also z.B. `-d firma.de, *.firma.de`. Weitere Tipps zum Umgang mit Wildcard-Zertifikaten finden Sie bei den Beispielen.
- **--dry-run**  
führt `renew` oder `certonly` probeweise durch, ohne irgendwelche Zertifikate tatsächlich zu verändern bzw. zu speichern.
- **-i installer**  
wählt das gewünschte Installations-Plugin aus. `-i apache` entspricht `--apache`.

- `-m mail@host`  
gibt an, an welche E-Mail-Adresse Warnungen gesendet werden sollen, wenn ein Zertifikat auszulaufen droht. Fehlt die Option, bittet certbot run interaktiv um die Angabe einer Mail-Adresse.
- `--manual`  
gibt Anweisungen zur manuellen Authentifizierung Ihres Servers. Die Option ist erforderlich, wenn Sie ein Wildcard-Zertifikat wünschen, aber kein DNS-Plugin nutzen können (siehe das entsprechende Beispiel).
- `--server letsencrypt-server`  
legt fest, mit welchem Let's-Encrypt-Server certbot kommunizieren soll. Die explizite Angabe eines Servers ist nur in Ausnahmefällen erforderlich, z.B. bei der Umstellung der Server-Architektur bei Let's Encrypt.
- `--test-cert`  
fordert Testzertifikate anstelle von echten Zertifikaten an. Das ist für Experimente zu empfehlen, weil Let's Encrypt nur relativ wenige echte Zertifikate pro Woche ausstellt.
- `--webroot`  
verwendet das Authenticator-Plugin webroot (entspricht `-a webroot`). Dieses Plugin erzeugt eine Datei und überprüft dann, ob diese Datei über den angegebenen Domainnamen zugänglich ist.
- `--webroot-path dir bzw. -w dir`  
gibt an, in welchem lokalen Verzeichnis das webroot-Plugin vorübergehend eine Datei einrichten soll.

Zu certbot gibt es zwar keine [man](#)-Seite, dafür liefert certbot `--help all` zu einer Auflistung aller Optionen. Noch mehr Details können Sie hier nachlesen:

<https://certbot.eff.org/docs/using.html>

## Beispiel

Das folgende Kommando fordert Zertifikate für *kofler.info* und *www.kofler.info* an und verändert die Apache-Konfiguration so, dass die Zertifikate auch genutzt werden:

```
root# certbot --apache -d kofler.info -d www.kofler.info
```

Falls es certbot nicht gelingt, Ihre Domäne zu verifizieren, ist die Verwendung des *webroot*-Plugins empfehlenswert:

```
root# certbot --apache -a webroot -w /var/www/html \
-d www.kofler.info
```

Da die Zertifikate nur 90 Tage gelten, müssen Sie außerdem ein Cron-Script einrichten, das regelmäßig *certbot renew* ausführt! Wesentlich komplizierter ist die Ausstellung von Wildcard-Zertifikaten. Um Missbrauch zu verhindern, muss während der Erstausstellung ein neuer DNS-TXT-Eintrag erstellt werden. Für manche DNS-Server kann dieser Prozess durch certbot-Plugins automatisiert werden:

<https://certbot.eff.org/docs/using.html#dns-plugins>

Wenn diese Plugins nicht zur Verfügung stehen oder der DNS-Server, auf dem die Einträge Ihrer Domäne gespeichert sind, dazu nicht kompatibel ist, können Sie die Authentifizierung manuell durchführen: Während certbot ausgeführt wird, müssen Sie zuerst einen TXT-Eintrag erzeugen, warten, bis der DNS-Eintrag auch sichtbar ist (das dauert rund fünf bis zehn Minuten, mit Pech aber auch ein, zwei Stunden), und schließlich eine Verifikationsdatei auf Ihrem Webserver einrichten.

Die folgende, stark gekürzte Ausgabe von certbot skizziert den Ablauf. Die Vorgehensweise hat leider den Nachteil, dass damit automatisierte Zertifikaterneuerungen (*certbot renew*) unmöglich sind. Sie müssen also den gesamten Vorgang ca. alle drei Monate wiederholen. Insofern sind Wildcard-Zertifikate nur empfehlenswert, wenn es für Ihren Domain-Provider ein certbot-Plugin gibt.

```
root# certbot certonly --manual --agree-tos \
-d centos-buch.info,*.centos-buch.info
Plugins selected: Authenticator manual, Installer None
```

```
...
Please deploy a DNS TXT record under the name
_acme-challenge.centos-buch.info with the following value:
bmwOk4...iAa8
Before continuing, verify the record is deployed.
Press Enter to Continue: <Return>
```

Create a file containing just this data:  
TXKgW0...5IL4

And make it available on your web server at this URL:  
<http://centos-buch.info/.well-known/acme-challenge/TXKgW0...adMw>  
Press Enter to Continue: <Return>

### **cfdisk [optionen] device**

cfdisk hilft bei der Formatierung von Festplatten und SSDs. Es unterstützt Partitionstabellen in den Formaten MBR und GPT. Die Bedienung ist im Vergleich zu den bekannteren Kommandos [fdisk](#) und [parted](#) etwas komfortabler: Die aktuelle Partitionstabelle wird als Menü angezeigt. Mit den Cursortasten kann nun die zu bearbeitende Partition ausgewählt werden. cfdisk ist nur für die interaktive Verwendung gedacht und ist nicht zur Script-Programmierung geeignet.

### **chacl [optionen] dateien**

chacl 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 acl verwendet werden.

Statt chacl sollten Sie nach Möglichkeit [getfacl](#) bzw. [setfacl](#) einsetzen. chacl steht nur aus Kompatibilitätsgründen zu IRIX SGI zur Verfügung.

### **chage [optionen] loginname**

Mit chage stellen Sie ein, wie lange ein Benutzerkonto verwendet werden darf, wann dessen Passwort abläuft und verändert werden muss und wie oft das Passwort verändert werden darf bzw. muss.

- -d *datum*

setzt das Datum fest, an dem das Passwort zuletzt verändert

wurde. Das Datum wird in der Schreibweise JJJJ-MM-TT (also z.B. 2020-12-31) angegeben. -d 0 bewirkt, dass der Benutzer sofort nach dem ersten Login sein Passwort ändern muss. Der Benutzer kann kein Kommando ausführen, bevor das Passwort verändert ist.

- **-E *datum***  
setzt das Datum fest, an dem das Konto abläuft. Ab diesem Datum ist kein Login mehr möglich. Mit **-E -1** gibt es kein Ablaufdatum. Abgelaufene Konten können mit **passwd -u** wieder aktiviert werden.
- **-I *n***  
gibt an, wie viele Tage nach Ablauf des Passworts ein Konto deaktiviert wird. **-I -1** verhindert eine Deaktivierung.
- **-l**  
zeigt die aktuellen Einstellungen an.
- **-m *n***  
legt fest, wie viele Tage mindestens verstreichen müssen, bevor ein geändertes Passwort neuerlich verändert werden kann. Die Standardeinstellung 0 bedeutet, dass der Benutzer sein Passwort jederzeit ändern darf.
- **-M *n***  
legt fest, nach wie vielen Tagen das Passwort spätestens geändert werden muss. **-1** bedeutet, dass das Passwort unbeschränkt gültig bleibt. **183** bedeutet, dass das Passwort zweimal pro Jahr verändert werden muss.
- **-W *n***  
gibt an, wie viele Tage vor dem Ablauf des Passworts eine Warnung angezeigt wird (standardmäßig 7).

## **Beispiel**

Mit den folgenden Kommandos richten Sie einen neuen Benutzer ein, legen dessen Startpasswort fest und zwingen ihn, sein Passwort sofort beim ersten Login sowie in Zukunft zumindest einmal

jährlich zu ändern. Die Warnung vor dem Ablauf des Passworts erscheint jeweils einen Monat im Voraus.

```
root# useradd peter
root# passwd peter
New password: *****
Retype new password: *****
root# chage -d 0 -M 365 -W 31 peter
chattr [optionen] +-=[aAcCdDeijsStTu] dateien
```

In Linux-Dateisystemen (ext2 bis ext4, btrfs, xfs etc.) können zusammen mit jeder Datei neben den Benutzerinformationen (siehe [chmod](#) und [chown](#)) einige zusätzliche Attribute gespeichert werden. Diese Attribute mit den Buchstaben-Codes aAcCdDeijsStTu steuern beispielsweise, ob Dateien komprimiert werden sollen, ob Änderungen an Dateien mit dem Copy-on-Write-Verfahren durchgeführt werden sollen etc.

Eine kurze Beschreibung der Attribute gibt `man chattr`. Allerdings werden momentan nur wenige der vorgesehenen Attribute tatsächlich genutzt. Zum Auslesen der Attribute verwenden Sie [lsattr](#).

- `-R`  
verändert die Attribute rekursiv in allen Dateien des Verzeichnisbaums.

## Beispiel

Eine wichtige Anwendung findet `chattr` bei [btrfs](#)-Dateisystemen: Dort können Sie mit `chattr +c` das automatisch aktive Copy-on-Write-Verfahren (COW) deaktivieren bzw. mit `chattr -c` wieder aktivieren. Die Deaktivierung von COW ist erforderlich, um in [btrfs](#)-Dateisystemen Image-Dateien von Virtualisierungsprogrammen oder große Datenbankdateien effizient zu speichern.

Dem Kommando übergeben Sie wahlweise einzelne Dateien oder ganze Verzeichnisse. Mit `-R` verändert das Kommando das Attribut rekursiv in Verzeichnisbäumen.

```
root# chattr +c -R /var/lib/mysql
```

Das Attribut gilt allerdings nur für neue Dateien. Um COW bei bereits existierenden Dateien mit einer Größe ungleich 0 Byte zu deaktivieren, müssen Sie die Dateien kopieren:

```
root# mv verz backup
root# mkdir verz
root# chattr +C verz
root# cp -a backup/* verz
root# rm -rf backup

chcon [optionen] kontext dateien
chcon [optionen] --reference=referenzdatei dateien
chcon [optionen] [-u benutzer] [-r rolle] [-t typ] [-l level]
dateien
```

**chcon** verändert den SELinux-Kontext der angegebenen Dateien. Der neue Kontext kann auf drei Arten angegeben werden: in der Schreibweise `benutzer:rolle:typ:level`, durch die Angabe einer Referenzdatei oder mit den Optionen `-u`, `-r`, `-l` und `-t`. Die dritte Variante hat den Vorteil, dass damit nur ein Teil des gesamten Kontexts verändert werden kann.

Eine manuelle Einstellung des SELinux-Kontexts ist normalerweise dann erforderlich, wenn Sie Dateien oder Verzeichnisse an anderen Orten speichern, als die SELinux-Regeln dies vorsehen – also beispielsweise Apache-Dateien in `/disk2/var/www/html` anstelle von `/var/www/html`.

Wenn es Ihnen nur darum geht, den SELinux-Kontext von Dateien zu korrigieren, die sich ohnedies im vorgesehenen Verzeichnis befinden, führt [restorecon](#) unkomplizierter zum Ziel. Um den SELinux-Kontext vorhandener Dateien festzustellen, verwenden Sie das Kommando [ls](#) mit der Option `-Z`.

- `-l level`

verändert den Level des SELinux-Kontexts. In den unter RHEL und Fedora üblicherweise geltenden Regeln (Targeted Security) werden Level nicht genutzt. Der Kontext-Level lautet generell `s0` (siehe auch die Datei `/etc/selinux/targeted/setrans.conf`).

- **-r role**  
verändert die Rolle des SELinux-Kontexts. SELinux-Rollen legen fest, welcher »Domäne« die Datei zugeordnet wird. SELinux-Domänen bestimmen wiederum, welche Rechte ein Prozess hat.
- **--reference=referenzdatei**  
liest den SELinux-Kontext der Referenzdatei und überträgt diesen auf die weiteren angegebenen Dateien.
- **-R**  
berücksichtigt auch alle Unterverzeichnisse (*recursive*).
- **-t typ**  
verändert den Typ des SELinux-Kontexts. SELinux-Typen bestimmen die Rechte einer Datei, so wie SELinux-Domänen die Rechte eines Prozesses steuern.
- **-u benutzer**  
verändert die Benutzerinformation des SELinux-Kontexts. Der SELinux-Benutzer ist eine den SELinux-Regeln bekannte Identität. Gewöhnliche Linux-Benutzer sind durch SELinux-Regeln mit SELinux-Identitäten verbunden (»gemappt«).
- **-v**  
zeigt die durchgeführten Änderungen an (*verbose*).

## Beispiele

Standardmäßig hat Apache unter CentOS/RHEL im Verzeichnis `/var/www/html` nur Leserechte. Damit der Webserver in einem bestimmten Verzeichnis auch Dateien verändern darf, führen Sie das folgende Kommando aus:

```
root# chcon -R system_u:object_r:httpd_sys_content_rw_t:s0 \
          /var/www/html/wordpress
```

Wenn Sie möchten, dass Apache auch auf Webdateien zugreifen kann (diesmal nur lesend), die sich in einem Verzeichnis auf einem anderen Datenträger befinden, gehen Sie wie folgt vor:

```
root# chcon -R system_u:object_r:httpd_sys_content_t:s0 /disk2/var/www/html/
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.

## Beispiel

Das folgende Kommando ordnet alle Dateien im Verzeichnis `/var/www` der Gruppe `www-data` zu. Unter diesem Account läuft unter Debian und Ubuntu der Webserver Apache.

```
root# chgrp -R www-data /var/www/  
chkrootkit [optionen]
```

Das Kommando `chkrootkit` aus dem gleichnamigen Paket durchsucht das System nach bekannten Rootkits. Die Vorgehensweise ist allerdings recht simpel: Das Kommando versucht, die Schadsoftware anhand einfacher Kriterien bzw. Signaturen (z.B. dem Vorkommen bestimmter Zeichenketten in Binärdateien) zu erkennen.

- **-n**  
ignoriert NFS-Verzeichnisse.
  - **-r *verzeichnis***  
verwendet das angegebene Verzeichnis als Startpunkt (als Root-Verzeichnis).

Das Projekt ist zwar nach wie vor aktiv, die Weiterentwicklung verläuft aber nur noch in kleinen Schritten. Sowohl Fehlalarme (*false positives*) als auch das Nichterkennen echter Bedrohungen sind möglich.

## Beispiel

```
root# chkrootkit
ROOTDIR is `/'
Checking `amd'...                                not found
Checking `basename'...                            not infected
Checking `biff'...                                not found
Checking `chfn'...                                not infected
```

```
Searching for sniffer's logs ...           nothing found
Searching for rootkit HiDrootkit's files... nothing found
Searching for rootkit t0rn's default files... nothing found
...
Checking `asp'...                          not infected
Checking `bindshell'...                    not infected
Checking `lkm'...                          chkproc: nothing detected
...
```

## Varianten

chkrootkit kann aufgrund seiner einfachen Anwendung der Startpunkt einer Rootkit-Suche sein. Alternative bzw. ergänzende Werkzeuge sind beispielsweise [rkhunter](#) sowie die kommerziellen Tools Lynis und Snort. Die beiden zuletzt genannten Programme haben jeweils einen Open-Source-Kern, der kostenlos genutzt werden kann.

### 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 `rwxrwxrwx`.

Die drei Spezialbits `setuid`, `setgid` und `sticky` haben die Oktalwerte `4000`, `2000` und `1000`. Beachten Sie, dass Sie diese drei Bits mit `chmod` zwar setzen können (`chmod 2770 verz`), aber nicht mehr löschen können. `chmod 0770 verz` bleibt wirkungslos. Sie müssen `chmod g-s verz` ausführen, um das Setgid-Bit zu löschen.

Um die Zugriffsbits einer vorhandenen Datei oder eines Verzeichnisses in Oktalschreibweise zu ermitteln, verwenden Sie das Kommando [stat](#).

## Optionen

- **-f** bzw. **--silent** bzw. **--quiet**  
zeigt keine Fehlermeldungen an.
- **-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+rwx sichere
```

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

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

Nur root darf die Datei *server.key* lesen, niemand darf sie verändern:

```
root# chmod 400 server.key
```

```
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

Die folgenden Kommandos stellen sicher, dass alle Dateien innerhalb von */var/lib/mysql* dem Benutzer und der Gruppe [mysql](#) zugeordnet sind, sodass nur der MySQL-Server-Prozess die Dateien lesen und verändern darf:

```
root# chown -R mysql:mysql /var/lib/mysql
root# chmod -R o-rwx /var/lib/mysql
```

```
chpasswd [optionen]
```

**chpasswd** ermöglicht es, Passwörter in Scripts zu ändern, also im Gegensatz zu [passwd](#) ohne Benutzerinteraktion. Das Kommando

liest die Benutzernamen und Passwörter aus der Standardeingabe, wobei die Informationen zeilenweise in der Form name:passwort angegeben werden müssen. Oft wird das Kommando in der Form chpasswd < pwfile.txt aufgerufen, um die in einer Datei gespeicherten Kombinationen aus Namen und Passwörtern zu verarbeiten.

Durch Optionen können Sie steuern, wie die Passwörter verschlüsselt werden (siehe `man chpasswd`). Normalerweise ist es aber nicht zweckmäßig, diese Optionen zu verwenden. chpasswd verwendet dann die von der Distribution vorgegebenen Defaulteinstellungen aus der Datei `/etc/logins.def`.

### Beispiel

Das folgende Script richtet mit `useradd` 30 Accounts ein, user01 bis user30. Für jeden Account wird ein mit `mkpasswd` generiertes Passwort eingestellt. Dazu werden der Name und das Passwort mit `echo` ausgegeben und die Ausgabe wird mit dem Pipe-Operator direkt an chpasswd weitergereicht. Alle Account-Namen und die dazugehörigen Passwörter werden außerdem in der Textdatei `students.txt` gespeichert. Zuletzt wird mit `chage` eingestellt, dass jeder Benutzer sofort nach dem ersten Login sein Passwort ändern muss.

```
#!/bin/bash
for i in {01..30}; do
    user="user$i"
    pw=$(mkpasswd -l 8 -d 0 -c 0 -s 0)
    echo "Account: $user  Password: $pw" >> students.txt
    useradd $user
    echo "$user:$pw" | chpasswd
    chage -d 0 $user
done
```

**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.

## Beispiel

chroot ist praktisch, um administrative Arbeiten in einem Dateisystem durchzuführen, das nicht beim Booten aktiviert wurde. Die folgenden Kommandos gehen davon aus, dass Sie einen Rechner von einem Rescue-System oder einer Live-DVD (Knoppix) gestartet haben. Nun binden Sie die Systempartition (im Beispiel `/dev/sda4`) in das Verzeichnis `/mnt/rescue` ein, machen dieses Verzeichnis zum neuen Wurzelverzeichnis und stellen dann mit [passwd](#) das vergessene root-Passwort neu ein:

```
root# mkdir /mnt/rescue
root# mount /dev/sda4 /mnt/rescue
root# chroot /mnt/rescue
root# passwd
Ändere Passwort für Benutzer root.
Geben Sie ein neues Passwort ein: ****
Geben Sie das neue Passwort erneut ein: ****
root# exit
root# reboot
```

**chsh -s shell [user]**

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/zsh`. Das Kommando chsh verändert die Datei [/etc/passwd](#) und trägt dort die neue Shell ein.

Jeder Anwender kann seine eigene Default-Shell nach Belieben verändern. Die Shell eines anderen Anwenders kann nur von root verändert werden. Die neue Shell muss mit dem vollständigen Verzeichnis angegeben werden. Die Änderung wird mit dem nächsten Login wirksam.

## Beispiel

Das folgende Kommando stellt anstelle einer Shell das Programm `nologin` ein. Es verhindert einen direkten Login und ist zweckmäßig,

wenn Sie einen Benutzer nur für automatisierte (Backup-)Scripts oder nur zum Zugriff auf einen E-Mail-Accounts eingerichtet haben:

```
root# chsh -s /usr/sbin/nologin mailuser
```

### chvt n

chvt aktiviert die Konsole n. Bei älteren Linux-Distributionen waren die Nummern 1 bis 6 für Textkonsolen reserviert, während auf der Konsole 7 das Grafiksystem des ersten Benutzers ausgeführt wurde. Für weitere parallele Logins wurden entsprechend die Konsolen 8, 9 usw. verwendet.

Bei neueren Distributionen läuft in der Regel auf der Konsole 1 der Display-Manager im Grafikmodus. Konsole 2 ist für das Grafiksystem des ersten eingeloggten Benutzers vorgesehen. Die Konsolen 3 bis 6 werden im Textmodus betrieben.

### 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 und kryptografisch wesentlich sicherer ist das Kommando sha512sum.

### clear

clear bzw. **[Strg]+[L]** löscht 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. Zum Vergleich von Textdateien eignet sich [diff](#) in der Regel besser als cmp.

- **-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.

## **cnf** kommando

**cnf** steht für *Command Not Found* und ist ein SUSE-spezifisches Kommando, das verrät, in welchem noch nicht installierten Paket das gesuchte Kommando enthalten ist.

Unter Red-Hat- und Fedora-Distributionen enthält das von der bash ausgeführte Kommando */usr/libexec/pk-command-not-found* eine vergleichbare Funktion, unter Ubuntu ist das */usr/lib/command-not-found* aus dem Paket *command-not-found*. Dieses Paket steht auch unter Debian zur Verfügung. Dort muss es aber manuell installiert und durch das Kommando *update-command-not-found* einmalig eingerichtet werden.

## **Beispiel**

```
user$ cnf jmacs
Das Programm 'jmacs' kann im folgenden Paket gefunden werden:
 * joe [ Pfad: /usr/bin/jmacs, Repository: zypp (repo-oss) ]
Try installing with:
 sudo zypper install joe
```

## **column** [optionen] datei

**column** liest die angegebene Datei, trennt jede Zeile in mehrere Teile auf und gibt dann alle Teile in eingerückten Spalten aus.

- **-o 'outseparator'**  
gibt an, wie die Spalten bei der Ausgabe voneinander getrennt werden sollen (standardmäßig durch zwei Leerzeichen).
- **-s 'inseparator'**  
gibt an, wie die einzulesenden Daten voneinander getrennt sind. Standardmäßig gilt *Whitespace* als Trennzeichen, also Leer- und Tabulatorzeichen.
- **-t**  
formatiert die Ausgabe als Tabelle.

## **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 die Alternativen [bzip2](#), [gzip](#) und [xz](#).

### **continue [n]**

Das bash-Konstrukt continue überspringt in Shell-Skripts den Körper einer [for](#)-, [while](#)- oder [until](#)-Schleife und setzt die Schleife mit dem nächsten Durchlauf fort. Durch den optionalen Zahlenwert kann dieser Vorgang für mehrere Schleifenebenen durchgeführt werden.

### **convert [optionen] bildalt bildneu**

convert aus dem ImageMagick-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.

## Beispiele

Die drei folgenden Beispiele zeigen einfache Anwendungen des Kommandos:

```
user$ convert -resize 100x100 in.jpg out.png  
user$ convert -type Grayscale in.jpg out.eps  
user$ convert -quality 80 in.bmp out.jpg
```

Das folgende Kommando macht aus einer PNG-Datei eine EPS-Datei gemäß der PostScript-Level-2-Spezifikation. (convert erzeugt normalerweise Dateien für PostScript Level 3. Diese sind dann oft deutlich größer, ohne dass sich daraus Vorteile ergeben.) Alle Bildecken werden übereinander platziert. Transparente Bereiche der PNG-Datei werden weiß dargestellt, nicht schwarz.

```
user$ convert -flatten -alpha remove -background white in.png EPS2:out.eps
```

## Dokumentation

Die [man](#)-Seite zu convert ist nicht besonders hilfreich. Dafür sind sämtliche Optionen ausführlich auf der folgenden Webseite beschrieben:

<https://imagemagick.org/script/command-line-processing.php>

```
cp [optionen] quelle ziel  
cp [optionen] dateien zielverzeichnis
```

cp kopiert Dateien und Verzeichnisse. Einzelne Dateien können beim Kopieren umbenannt werden. Bei der Bearbeitung mehrerer Dateien (z.B. durch die Angabe von Jokerzeichen) können diese lediglich in ein anderes Verzeichnis kopiert, nicht aber umbenannt werden. Anweisungen der Art cp \*.tex \*.bak sind nicht zulässig. Mit cp vergleichbare Kommandos sind [mv](#) zum Verschieben und Umbenennen von Dateien sowie [ln](#) zur Herstellung von Links. cp unterstützt unter anderem folgende Optionen:

- **-a bzw. --archive**  
behält möglichst alle Attribute der Dateien bei. -a ist eine Abkürzung für -dpR.
- **-b bzw. --backup**  
benennt bereits vorhandene Dateien in Backup-Dateien um (Dateiname plus ~), anstatt sie zu überschreiben.
- **-d bzw. --dereference**  
kopiert bei Links nur den Verweis, nicht aber die Datei, auf die der Link zeigt.
- **-i bzw. --interactive**  
fragt, bevor vorhandene Dateien überschrieben werden.
- **-l bzw. --link**  
erstellt feste Links (*Hard Links*), anstatt die Dateien zu kopieren. Wenn cp mit dieser Option verwendet wird, hat es dieselbe Funktionalität wie [ln](#).
- **-p bzw. --preserve**  
lässt die Informationen über den Besitzer, die

Gruppenzugehörigkeit, die Zugriffsrechte und den Zeitpunkt der letzten Änderung unverändert. Ohne diese Option gehört die Kopie demjenigen, der cp ausführt (Benutzer und Gruppe), und die Zeitangabe wird auf die aktuelle Zeit gesetzt.

- **-r bzw. -R bzw. --recursive**  
kopiert auch Unterverzeichnisse und die darin enthaltenen Dateien.
- **--reflink[=auto]**  
kopiert anfänglich nur die Metadaten der Datei, erstellt also eine sogenannte *Shallow Copy* der Datei. Die Kopie beansprucht deswegen fast keinen Platz im Dateisystem. Dennoch ist sie vom Original unabhängig: Sobald Datenblöcke des Originals oder der Kopie verändert werden, werden diese Blöcke neu alloziert.  
Shallow Copies funktionieren nur, wenn diese Funktion vom Dateisystem unterstützt wird. Das ist bei [btrfs](#) sowie bei aktuellen Versionen von *xfs* der Fall, nicht aber bei *ext4*. Bei nicht kompatiblen Dateisystemen führt die Option zu einem Fehler, es sei denn, Sie geben den Modus *auto* an: Dann entscheidet cp selbst, welche Art von Kopie möglich ist.
- **-s bzw. --symbolic-link**  
erstellt symbolische Links, anstatt die Dateien oder Verzeichnisse zu kopieren. cp hat damit die Funktionalität von [ln](#) -s.
- **-u bzw. --update**  
kopiert Dateien nur dann, wenn dabei keine gleichnamige Datei mit neuem Datum überschrieben wird.

## Verzeichnisse kopieren

Wenn Sie ein ganzes Verzeichnis mit allen darin enthaltenen Dateien und Unterverzeichnissen kopieren möchten, führen Sie cp *-r quellverzeichnis zielverzeichnis* aus. Damit werden auch versteckte Dateien und Unterverzeichnisse kopiert. Wenn Sie

möchten, dass beim Kopieren die Zugriffsrechte und -zeiten erhalten bleiben, verwenden Sie statt `-r` die Option `-a`.

Etwas diffizil ist die Frage, ob das Quellverzeichnis selbst oder nur sein Inhalt kopiert wird. Wenn es das Zielverzeichnis bereits gibt, wird darin das neue Unterverzeichnis *quellverzeichnis* erzeugt und der gesamte Inhalt des Quellverzeichnisses dorthin kopiert. Wenn es das Zielverzeichnis hingegen noch nicht gibt, wird es erzeugt; in diesem Fall wird nur der Inhalt des Quellverzeichnisses in das neu erzeugte Zielverzeichnis kopiert, nicht aber das Quellverzeichnis selbst. Zum Kopieren ganzer Verzeichnisbäume eignen sich auch [cpio](#), [rsync](#) und [tar](#).

## Beispiele

Das folgende Kommando kopiert alle `*.tex`-Dateien aus dem Unterverzeichnis *buch* in das aktuelle Verzeichnis. Der Punkt gibt dabei als Zielverzeichnis das aktuelle Verzeichnis an:

```
user$ cp buch/*.tex .
```

Das zweite Kommando erstellt eine Backup-Kopie des gesamten Verzeichnisses *buch*:

```
user$ cp -a buch bak-buch
```

`cp` kann nicht dazu verwendet werden, mehrere Dateien beim Kopieren umzubenennen. `cp *.*.xxx *.*.yyy` kopiert also nicht alle `*.*.xxx`-Dateien in `*.*.yyy`-Dateien. Um solche Operationen durchzuführen, müssen Sie [for](#) oder [sed](#) zu Hilfe nehmen. Im folgenden Kommando bildet [for](#) eine Schleife über alle `*.*.xxx`-Dateien. Der Ausdruck  `${i%.xxx}.yyy` entfernt die Endung `*.*.xxx` und ersetzt sie durch `.yyy`. Wenn Sie `cp` durch [mv](#) ersetzen, werden die Dateien nicht kopiert, sondern umbenannt.

```
user$ for i in *.*.xxx; do cp $i ${i%.xxx}.yyy; done
```

Etwas komplizierter ist die Vorgehensweise mit [sed](#): [ls](#) liefert die Liste der Dateien, die kopiert werden sollen, und gibt sie an [sed](#) weiter. [sed](#) bildet daraus mit dem Kommando `s` (*regular find and replace*) eine Liste von `cp`-Kommandos und gibt diese wiederum an eine neue Shell `sh` weiter, die die Kommandos schließlich ausführt.

```
user$ ls *.xxx | sed 's/(.*\)\.xxx$/cp & \1.yyy/' | sh  
cpio kommando [optionen]
```

cpio fasst mehrere Dateien zu einem Archiv zusammen und kopiert sie auf einen anderen Datenträger, auf einen Streamer oder in ein anderes Verzeichnis. Analog kann das Kommando auch zum Wiedereinlesen solcher Daten verwendet werden. Unter Linux ist cpio eher ungebräuchlich, stattdessen wird zumeist [tar](#) verwendet. Die zentralen cpio-Kommandos sind:

- -i  
(*input*) liest ein Archiv aus der Standardeingabe und extrahiert die darin enthaltenen Dateien.
- -o  
(*output*) fasst Dateien in ein Archiv zusammen und schreibt diese an die Standardausgabe. Die zu verarbeitenden Dateien werden üblicherweise durch [find](#) ermittelt und dann über eine Pipe an cpio weitergegeben.
- -p  
(*pass through*) überträgt Archive zwischen verschiedenen Verzeichnissen.
- -t  
zeigt den Inhalt eines Archivs aus der Standardeingabe an.

Details dieser Aktionen können durch weitere Optionen gesteuert werden.

## Beispiel

Mit den folgenden Kommandos werden zuerst alle Dateien aus dem Verzeichnis /etc archiviert. cpio -t zeigt danach als Kontrolle die Liste der gesicherten Dateien.

```
root# cd /etc  
root# find . | cpio -o > /tmp/etc-backup.cpio  
root# cpio -t < /tmp/etc-backup.cpio
```

```
crontab [optionen]
```

Das Kommando crontab hilft bei der Administration eigener Cron-Jobs in der Datei `/var/spool/cron/benutzername` (siehe `/etc/crontab`).

Cron-Jobs werden zu den vorgesehenen Zeiten automatisch im Hintergrund ausgeführt, z.B. einmal stündlich oder an jedem Sonntagabend.

- `-e`  
öffnet einen Editor zur Bearbeitung der Crontab-Datei. Als Editor kommt normalerweise `vi` zum Einsatz. Wenn Sie einen anderen Editor wünschen, müssen Sie die Umgebungsvariable `EDITOR` entsprechend einstellen. `crontab` überprüft beim Speichern die Einhaltung der Crontab-Syntaxregeln.
- `-l`  
listet die aktuellen Crontab-Einträge auf.
- `-r`  
löscht alle Crontab-Einträge.
- `-u benutzername`  
bearbeitet die Crontab-Datei des angegebenen Benutzers. Die Option `-u` steht nur `root` zur Verfügung.

### **cryptsetup [optionen] kommando**

`cryptsetup` aus dem gleichnamigen Paket greift auf Funktionen des Kernelmoduls `dm_crypt` zurück. Es richtet Crypto-Devices ein, aktiviert und deaktiviert sie. Anstelle eines Devices kann auch eine ausreichend große Image-Datei verwendet werden. Diese kann im Voraus z.B. mit `dd if=/dev/zero of=cryptoImage bs=1M count=n` erzeugt werden, wobei `n` die gewünschte Größe in MiB ist. An dieser Stelle werden nur die wichtigsten LUKS-spezifischen Kommandos von `cryptsetup` beschrieben:

- `luksAddKey device`  
richtet ein zusätzliches Passwort ein, das Zugriff auf den Crypto-Container gibt. Zur Ausführung des Kommandos muss ein bereits existierendes Passwort angegeben werden (egal welches). Insgesamt sind maximal acht Passwörter erlaubt.

- `luksClose device mappingname`  
deaktiviert ein Crypto-Device.
- `luksDump device`  
liefert Metainformationen über den Crypto-Container (z.B. den Verschlüsselungsalgorithmus).
- `luksFormat --type luks2 device`  
richtet im angegebenen Device einen Crypto-Container im aktuellen Format LUKS2 ein. (Ohne die Option `--type` verwendet das Kommando das weiterhin unterstützte, aber veraltete Format LUKS1.)  
In der Folge müssen Sie zweimal die *Passphrase* angeben, die aus Sicherheitsgründen zumindest 20 Zeichen lang sein sollte. Den gewünschten Verschlüsselungsalgorithmus können Sie mit `-c` auswählen, eine andere Schlüssellänge mit `-s` einstellen. Die zur Auswahl stehenden Algorithmen sind in der Pseudodatei `/proc/crypto` aufgelistet. Die Ausgabe des Kommandos `cryptsetup --help` verrät in den letzten Zeilen, welcher Algorithmus und welche Schlüssellänge standardmäßig zum Einsatz kommen.  
Aktuell ist dies der Algorithmus `aes-xts-plain64` mit einem 256-Bit-Schlüssel.
- `luksOpen device mappingname`  
aktiviert den Crypto-Container im angegebenen Device und weist ihm einen Namen zu. Das resultierende Crypto-Device kann nun über `/dev/mapper/mappingname` genutzt werden.

## Beispiel

Die folgenden Kommandos zeigen, wie Sie einen USB-Stick (`/dev/sdh1`) zuerst als Crypto-Device formatieren (`luksFormat`) und das Device dann unter dem willkürlich gewählten Namen `mycontainer` aktivieren (`luksOpen`). Naturgemäß sind Ihre Daten nur so sicher wie Ihr Passwort bzw. Ihre aus mehreren Wörtern bestehende Passphrase. Empfohlen wird eine Passwortlänge von zumindest 20 Zeichen.

Anschließend können Sie `/dev/mapper/mycontainer` wie eine Festplattenpartition oder ein Logical Volume nutzen – also ein Dateisystem einrichten, dieses in den Verzeichnisbaum einbinden etc. Nach [umount](#) müssen Sie daran denken, das Crypto-Device wieder zu deaktivieren (`luksClose`), um `/dev/sdh1` freizugeben. Erst jetzt dürfen Sie den USB-Stick ausstecken.

```
root# cryptsetup luksFormat --type luks2 /dev/sdh1
Daten auf /dev/sdh1 werden unwiderruflich überschrieben.
Are you sure? (Type uppercase yes): YES
Enter LUKS passphrase: ****
Verify passphrase: ****
Command successful.
```

```
root# cryptsetup luksOpen /dev/sdh1 mycontainer
Enter LUKS passphrase: ****
root# mkfs.ext4 /dev/mapper/mycontainer
root# mount /dev/mapper/mycontainer /test
root# ... das verschlüsselte Dateisystem nutzen ...
root# umount /test/
root# cryptsetup luksClose mycontainer
```

### **csplit [optionen] datei trennposition**

`csplit` zerlegt eine Textdatei an vorgegebenen Stellen in mehrere Einzeldateien. Die Trennposition kann entweder durch eine direkte Zeilenangabe oder durch ein Suchmuster angegeben werden. Das Kommando liefert als Ergebnis die Dateien `xx00`, `xx01` etc. und gibt auf dem Bildschirm deren Längen aus. Durch die Angabe entsprechender Optionen sind natürlich auch »schönere« Dateinamen möglich.

[cat](#) kann aus diesen Einzeldateien wieder die Originaldatei zusammensetzen. Anstelle von `csplit` bietet sich zum Zerlegen binärer Dateien das Kommando [split](#) an.

### **Angabe der Trennpositionen**

Die Trennpositionen werden entweder durch eine Zeilenanzahl oder durch ein Suchmuster angegeben. Im einen Fall wird die Datei nach  $n$  Zeilen zerlegt, im anderen Fall vor oder nach dem Auftreten des Suchmusters. Wenn `csplit` die Datei mehrfach zerlegen soll, was zumeist der Fall ist, muss hinter der Zeilenanzahl bzw. dem

Trennmuster angegeben werden, wie oft die Operation wiederholt werden soll.

- `/muster/`  
trennt die Datei in der Zeile vor dem Auftreten des Musters. (Die Zeile mit dem gefundenen Muster wird zur ersten Zeile der nächsten Datei.)
- `/muster/+n`  
`/muster/-n`  
trennt die Datei  $n$  Zeilen nach (+) oder vor (-) dem Auftreten des Musters.
- `n`  
trennt die Datei nach  $n$  Zeilen.
- `{n}`  
zerlegt die Datei in  $n+1$  Einzeldateien (und nicht nur in zwei Dateien).
- `{*}`  
zerlegt die Datei bei jedem Auftreten des Suchmusters bzw. jedes Mal nach  $n$  Zeilen in die entsprechende Anzahl von Einzeldateien.

## Optionen

- `-f datei` bzw. `--prefix=datei`  
verwendet den angegebenen Dateinamen zur Benennung der Ausgabedateien.
- `-k` bzw. `--keep-files`  
Bereits erzeugte Dateien werden beim Auftreten eines Fehlers nicht wieder gelöscht. Die Option muss insbesondere bei Musterangaben in der Form  $n \ {*}}$  verwendet werden. Die Musterangabe erfolgt wie bei [grep](#).
- `-z` bzw. `--elide-empty-files`  
verhindert die Erzeugung leerer Dateien. Ohne diese Option können leere Dateien insbesondere dann auftreten, wenn bereits die erste Zeile der Ausgangsdatei dem Suchmuster entspricht.

## Beispiel

`csplit` zerlegt *total.txt* in die Dateien *teil.00*, *teil.01* etc. Die Einzeldateien sind jeweils 100 Zeilen lang. `cat` bildet daraus anschließend eine Kopie der Originaldatei.

```
user$ csplit -k -f teil. total.txt 100 {*}  
user$ cat teil.* > kopie.txt
```

Im zweiten Beispiel zerlegt `csplit` die Datei *total.txt* in kleinere Dateien, wobei die Trennung immer dann erfolgt, wenn eine Zeile mit dem Text % === beginnt.

```
user$ csplit -k -f teil. total.txt '/^% ===/' {*} 
```

### **curl** [optionen] [url]

`curl` hilft bei der Übertragung von Dateien von oder zu einem Server, wobei alle erdenklichen Protokolle unterstützt werden (HTTP, HTTPS, FTP, SFTP, SCP etc.).

Die externe Datei bzw. das externe Verzeichnis wird durch eine URL-Zeichenkette (*Uniform Resource Locator*) angegeben, die mit dem Protokollnamen beginnt (z.B. `http://server.de/datei`).

- `--limit-rate n`  
begrenzt die Übertragungsgeschwindigkeit auf die angegebene Byteanzahl pro Sekunde. `n` kann der Buchstabe `k` oder `m` hintangestellt werden, um die Übertragungsrate auf `n` KiB oder MiB pro Sekunde zu limitieren.
- `-o datei`  
speichert die heruntergeladenen Daten in der angegebenen Datei, anstatt sie an die Standardausgabe weiterzuleiten.
- `-r n1-n2`  
überträgt den angegebenen Bytebereich der Datei.
- `-s bzw. --silent`  
verzichtet auf Statusausgaben.
- `-T datei`  
überträgt die angegebene Datei zum Server (*upload*). Statt des Dateinamens kann auch das Zeichen - angegeben werden, um Daten aus der Standardeingabe zu verarbeiten.

- `-u name:password`  
gibt den Login-Namen und das Passwort an.

## Beispiele

Das folgende Kommando überträgt die angegebene Datei zum FTP-Server backupserver und speichert sie im Verzeichnis `verz`:

```
user$ curl -T datei -u username:password ftp://backupserver/verz
```

Um Daten aus dem Standardeingabekanal zu verarbeiten, geben Sie mit `-T` als Dateinamen einen Bindestrich an. Das folgende Kommando speichert das aus dem [tar](#)-Kommando resultierende Ergebnis direkt in der Datei `name.tgz` auf dem FTP-Server:

```
user$ tar czf - verz/ | curl -T - -u usern:pw ftp://bserver/name.tgz
```

**cut [optionen] datei**

cut extrahiert aus jeder Zeile eines Textes die durch Optionen angegebenen Spalten.

- `-b liste` bzw. `--bytes liste`  
extrahiert die in einer Liste angegebenen Zeichen. Einzelne Einträge dürfen durch Kommata (aber nicht durch Leerzeichen) getrennt werden. Statt einzelner Zeichen dürfen auch ganze Bereiche angegeben werden, etwa `-b 3-6,9,11-15`.
- `-d zeichen` bzw. `--delimiter zeichen`  
gibt das Trennzeichen für `-f` an, das statt des Tabulatorzeichens verwendet werden soll.
- `-f liste` bzw. `--fields liste`  
wie `-b`, aber jetzt für Felder (Datensätze), die durch Tabulatorzeichen getrennt sein müssen.
- `-s` bzw. `--only-delimited`  
eliminiert alle Zeilen, die keine Daten enthalten, die der Option `-f` entsprechen. Kann nicht zusammen mit `-b` verwendet werden.

In der Praxis gelingt die Extraktion einzelner Textspalten mit [awk](#) oft wesentlich unkomplizierter als mit cut.

## Beispiel

Im folgenden Beispiel liest `cut` aus der Datei `/etc/passwd` die erste, die dritte und die siebte Spalte, also den Benutzernamen, die UID-Nummer und die Default-Shell. Das Trennzeichen in dieser Datei ist der Doppelpunkt.

```
root# cut -d\: -f1,3,7 /etc/passwd
root:0:/bin/bash
bin:1:/sbin/nologin
daemon:2:/sbin/nologin
adm:3:/sbin/nologin
...
```

## D

**date [optionen] [+format]**

date liefert das aktuelle Datum und die Uhrzeit bzw. verändert diese Daten. Eine Veränderung darf nur von root vorgenommen werden. date kann die Zeit in den verschiedensten Formaten anzeigen.

- **-s neuezeit**  
verändert das Datum und/oder die Uhrzeit.

Die folgende Liste nennt die wichtigsten Formatcodes. Die Formatzeichenkette muss in Anführungszeichen gesetzt werden.

- **%Y**  
das Jahr als vierstellige Zahl
- **%l**  
der Monat als Zahl (1 bis 12)
- **%m**  
der Monat als zweistellige Zahl (01 bis 12)
- **%b**  
der Monat als kurze Zeichenkette (Jan bis Dez)
- **%B**  
der Monat als lange Zeichenkette (Januar bis Dezember)
- **%d**  
der Tag des Monats als zweistellige Zahl (01 bis 31)
- **%e**  
der Tag des Monats als Zahl (1 bis 31), wobei einstelligen Zahlen ein Leerzeichen vorangestellt wird
- **%H**  
die Stunde als zweistellige Zahl (00 bis 23)
- **%M**  
die Minute als zweistellige Zahl (00 bis 59)

### Beispiel

Das folgende Script erzeugt ein komprimiertes [tar](#)-Archiv des Verzeichnisses `/home/kofler/data`. Das Archiv wird unter zwei

Dateinamen gespeichert: *mydata-day-dd.tar.gz* und *mydata-month-mm.tar.gz*.

Wenn das Script täglich ausgeführt wird, haben Sie mit der Zeit 43 Backup-Versionen, die den Zustand des Backup-Verzeichnisses für die letzten 28 bis 31 Tage sowie für die letzten 12 Monate widerspiegeln.

```
#!/bin/bash
fname1=/backup/mydata-day-$(date "+%d").tar.gz
fname2=/backup/mydata-month-$(date "+%m").tar.gz
tar czf $fname1 /home/kofler/data
cp $fname1 $fname2
chmod 600 $fname1 $fname2
```

### **dconf** kommando

**dconf** ist ein Low-Level-Kommando zur Veränderung der **dconf**-Datenbank, in der Gnome unzählige Einstellungen speichert. Anstelle von **dconf** sollten Sie in der Regel das Kommando [gsettings](#) verwenden, das mehr Kontrollmechanismen enthält.

- **dump** *load dir*  
liest ein ganzes Konfigurationsverzeichnis rekursiv aus und schreibt es an die Standardausgabe (**dump**) bzw. liest es von der Standardeingabe ein (**load**).
- **list** *dir*  
gibt die Einträge eines Verzeichnisses aus (nicht rekursiv).
- **read** *key*  
liest einen Eintrag aus.
- **reset** *key/dir*  
setzt einen Eintrag oder ein ganzes Konfigurationsverzeichnis auf den oder die Startwerte zurück. Bei Verzeichnissen muss die Option **-f** angegeben werden.
- **write** *key value*  
speichert bzw. überschreibt einen Eintrag.

### **Beispiele**

Die beiden folgenden Kommandos zeigen, wie Sie ein Backup aller Gnome-Einstellungen durchführen und später den so gespeicherten

Zustand wiederherstellen können:

```
user dconf dump / > dconf.bak  
user dconf load / < dconf.bak
```

Das folgende Kommando bewirkt einen totalen Reset aller Gnome-Einstellungen:

```
user$ dconf reset -f /  
dd optionen
```

dd überträgt Daten zwischen verschiedenen Speichermedien (Festplatte, USB-Stick etc.) und führt dabei auf Wunsch eine Konvertierung der Daten durch. Das Kommando kann dazu eingesetzt werden, Daten zwischen verschiedenen Rechnerarchitekturen auszutauschen.

dd kann nicht nur einzelne Dateien kopieren, sondern auch direkt auf Devices zugreifen. Damit können ganze Festplatten(partitionen) kopiert werden und kann der Bootsektor der Festplatte verändert werden etc. Auf dem Datenträger muss kein Dateisystem eingerichtet werden. Wenn dd ohne Optionen verwendet wird, liest es die Daten aus der Standardeingabe (Ende mit **Strg** + **Z**) und schreibt in die Standardausgabe. Die Optionen von dd werden ohne vorangehende Minus-Zeichen angegeben!

- **bs=n**

bestimmt die Blockgröße für die Ein- und Ausgabedatei. (Die Blockgröße gibt an, wie viele Bytes jeweils in einem Durchgang gelesen bzw. geschrieben werden.) Vorsicht: Standardmäßig verwendet dd eine Blockgröße von nur 512 Byte. Das ist extrem ineffizient! Wenn Sie mit dd Festplatten oder SSDs auslesen bzw. beschreiben, sollten Sie zumindest mit **bs=1M** arbeiten. Das Suffix **M** steht für  $1024 \times 1024$  Byte, **K** für 1024 Byte.

- **conv=modus**

konvertiert die Daten während des Kopierens. Für *modus* sind verschiedene Einstellungen erlaubt, unter anderem **lcase** (Großbuchstaben in Kleinbuchstaben umwandeln), **ucase** (Klein-

in Großbuchstaben umwandeln), swab (jeweils zwei Byte vertauschen) etc.

- **count=n**  
kopiert nur  $n$  Blöcke (und nicht die gesamten Daten).
- **ibs=n**  
bestimmt die Blockgröße der Quelldatei.
- **if=quelldatei**  
gibt die Quelldatei (statt der Standardeingabe) an.
- **obs=n**  
bestimmt die Blockgröße der Zieldatei.
- **of=zieldatei**  
gibt die Zieldatei (statt der Standardausgabe) an.
- **seek=n**  
überspringt  $n$  Blöcke, bevor die Ausgabe gestartet wird.
- **skip=n**  
überspringt  $n$  Blöcke, bevor mit dem Lesen begonnen wird.
- **status=progress**  
zeigt den Fortschritt des Kopierprozesses an. (Standardmäßig gilt status=none.)

Beachten Sie, dass der direkte Zugriff auf Devices (z.B. in der Form `cat /dev/xxx >`) oft schneller ist als dd. Außerdem müssen Sie sich dann keine Gedanken über die optimale Blockgröße machen.

## Beispiele

Durch das folgende Kommando wird die ISO-Datei `opensuse.iso` direkt auf den Datenträger mit dem Device `/dev/sdc` übertragen, also beispielsweise auf einen USB-Stick. Beachten Sie, dass dabei alle bisher gespeicherten Daten auf diesem Datenträger verloren gehen! Stellen Sie deswegen sicher, dass Sie den richtigen Device-Namen angeben. Wenn das mit `of` angegebene Device Ihre aktuelle Festplatte ist, verlieren Sie deren Inhalt!

```
root# dd if=opensuse.iso of=/dev/sdc bs=1M status=progress
```

Das zweite Kommando erzeugt eine 10 GiB große RAW-Image-Datei für ein Virtualisierungsprogramm (z.B. QEMU/KVM). Diese Image-Datei besteht einfach aus Nullen, die aus dem Device `/dev/zero` gelesen werden.

```
root# dd if=/dev/zero of=image.raw bs=1M count=10000
```

## Alternativen

Je nach Einsatzzweck bieten sich anstelle von `dd` die beiden folgenden Varianten an:

- `dd_rescue` bzw. je nach Distribution `ddrescue` (Paket `gddrescue`) versucht, eine Kopie auch dann zu erstellen, wenn einzelne Datensektoren des Quell-Devices defekt sind.
- `dcfldd` ist für forensische Aufgaben optimiert. Das Kommando kann unter anderem durch Prüfsummen verifizieren, dass die Kopie wirklich identisch ist.

Einen guten Vergleich zwischen `dd`, `dd_rescue` und `dcfldd` können Sie hier nachlesen:

<https://superuser.com/questions/355310>

```
declare [optionen] var[=wert]
```

Das bash-Kommando `declare` weist Shell-Variablen einen neuen Wert und/oder diverse Eigenschaften zu. Das Kommando wird vor allem in Shell-Scripts verwendet. Wenn es ohne Parameter aufgerufen wird, werden alle bekannten Variablen mit ihrem Inhalt aufgelistet. Vor und nach dem Gleichheitszeichen dürfen keine Leerzeichen angegeben werden!

- `-A`  
deklariert die Variable als assoziatives Array.
- `-r`  
deklariert die Variable read-only. Die Variable darf nur noch gelesen, aber nicht mehr verändert werden.
- `-x`  
deklariert die Variable als Umgebungsvariable (wie durch [export](#)).

Die Variable steht damit auch anderen Kommandos bzw. in Subshells zur Verfügung.

### **delgroup** [optionen] name

`delgroup` löscht auf Debian- und Ubuntu-Systemen die angegebene Gruppe und befolgt dabei die Regeln aus [`/etc/deluser.conf`](#). Bei Fedora und Red Hat ist `delgroup` ein Link auf das Kommando [`groupdel`](#), das aber andere Optionen kennt.

- `--only-if-empty`  
löscht die Gruppe nur, wenn ihr keine Benutzer zugeordnet sind.
- `--system`  
löscht die Gruppe nur, wenn es sich um eine Systemgruppe handelt.

### **deluser** [optionen] name

#### **deluser** name gruppe

`deluser` löscht auf Debian- und Ubuntu-Systemen den angegebenen Benutzer und befolgt dabei die Regeln aus [`/etc/deluser.conf`](#). Demzufolge wird das Heimatverzeichnis normalerweise nicht gelöscht. Bei Fedora und Red Hat ist `deluser` ein Link auf das Kommando [`userdel`](#) mit anderen Optionen.

- `--remove-all-files`  
löscht das Heimatverzeichnis sowie alle anderen Dateien des Benutzers (z.B. diverse Spool- und Mail-Dateien).
- `--remove-home`  
löscht auch das Heimatverzeichnis des Benutzers.
- `--system`  
löscht den Benutzer nur, wenn es sich um einen Systembenutzer handelt.

Wenn das Kommando in der Form `deluser name gruppe` ausgeführt wird, entfernt es den Benutzer aus der angegebenen Gruppe. Das Kommando agiert damit komplementär zu `adduser name gruppe`.

### **depmod** [optionen]

`depmod` erzeugt die Modulabhängigkeitsdatei `modules.dep` sowie diverse `*.map`-Dateien im Verzeichnis `/lib/modules/kernelversion/`. Die `*-map`-Dateien geben an, bei welcher Hardware-Komponente welches Kernelmodul geladen werden soll.

- `-A` bzw. `--quick`  
testet zuerst, ob bereits existierende `*.dep`- und `*.map`-Dateien noch aktuell sind. In diesem Fall verzichtet `depmod` darauf, die Dateien neu zu generieren.
- `-b verzeichnis` bzw. `--basedir verzeichnis`  
gibt an, für welches Verzeichnis die `*.dep`- und `*.map`-Dateien erzeugt werden sollen. Standardmäßig aktualisiert das Kommando die Dateien für den laufenden Kernel.

#### **df [optionen] [verzeichnis]**

`df` gibt Auskunft darüber, an welcher Stelle im Dateibaum Festplatten(partitionen) oder andere Laufwerke eingebunden sind und wie viel Speicherplatz darauf verfügbar ist. Normalerweise liefert `df` eine Liste aller aktiven Partitionen. Wenn Sie an das Kommando ein Verzeichnis übergeben, zeigt es hingegen nur die Daten der Partition an, in der sich dieses Verzeichnis physikalisch befindet. Bei Systemen mit vielen Partitionen können Sie auf diese Weise einfach feststellen, in welcher Partition ein bestimmtes Verzeichnis gespeichert wird.

- `-h`  
gibt den Speicherplatz in leicht lesbarer Form in MiB oder GiB an.
- `-i` bzw. `--inodes`  
gibt Informationen über die verfügbaren Inodes an (statt des freien Speicherplatzes in KiB).
- `-t fs`  
berücksichtigt nur Dateisysteme des angegebenen Typs, also beispielsweise mit `-t ext4` nur ext4-Dateisysteme.

- **-T**  
gibt für jede Partition bzw. für jeden Datenträger das Dateisystem an.
- **-x fs**  
ignoriert das angegebene Dateisystem.

## Beispiel

Auf dem Beispielsystem sind in der Systempartition / noch 15 GiB Speicherplatz frei. Die anderen Dateisysteme dienen zu Verwaltungszwecken (dynamische Darstellung von Devices) sowie zur Speicherung von Ubuntu-spezifischen Snap-Paketen. Wegen der Option **-x tmpfs** werden zahlreiche temporäre, also nur im RAM befindliche Dateisysteme nicht berücksichtigt.

```
root# df -h -T -x tmpfs
Filesystem      Type      Size  Used Avail Use% Mounted on
udev            devtmpfs  16G    0   16G  0% /dev
/dev/nvme0n1p1  vfat     1021M  254M  768M  25% /boot/efi
/dev/nvme0n1p4  ext4     119G   32G   81G  29% /
/dev/mapper/mycryptpart  ext4     1.5T  767G  634G  55% /crypt
/dev/loop0       squashfs 62M    62M    0  100% /snap/authy/5
/dev/loop1       squashfs 56M    56M    0  100% /snap/core18/1885
...

```

## dhclient Schnittstelle

Das Kommando **dhclient** aus dem **dhcp-client**-Paket bezieht bei den meisten Distributionen die Parameter für die Netzwerkkonfiguration von einem DHCP-Server. Das Kommando eignet sich ideal für eine Ad-hoc-Netzwerkkonfiguration bei neuen Server-Installationen. Bei manchen Distributionen kommen als Alternativen zu **dhclient** die Kommandos **dhclient3** oder **dhcpcd** zum Einsatz.

## Beispiel

Das folgende Kommando führt eine automatische Netzwerkkonfiguration für die Schnittstelle **eth0** durch:

```
root# dhclient eth0
diff [optionen] datei1 datei2
```

`diff` vergleicht normalerweise zwei Textdateien. Das Ergebnis ist eine Liste aller Zeilen, die voneinander abweichen. Das Kommando ist relativ »intelligent«, d.h., wenn in einer Datei gegenüber der anderen einige Zeilen eingefügt sind, so wird nur diese Abweichung gemeldet. Weitere Zeilen werden wieder als identisch erkannt, obwohl jetzt unterschiedliche Zeilenummern vorliegen. Das Kommando kann also dazu verwendet werden, die Abweichungen zwischen zwei Versionen eines Programmlistings rasch zu dokumentieren. `diff` kann in der Form `-r verz1 verz2` auch zwei Verzeichnisse vergleichen.

- `-b`  
betrachtet mehrfache Leerzeichen und -zeilen wie einfache Leerzeichen bzw. -zeilen.
- `-q`  
zeigt lediglich an, ob zwei Dateien unterschiedlich sind oder nicht. In dieser Variante verzichtet `diff` also auf die detaillierte Auflistung aller Unterschiede.
- `-r`  
vergleicht rekursiv zwei Verzeichnisse. Das Ergebnis listet auf, welche Dateien es nur im ersten Verzeichnis gibt und welche nur im zweiten Verzeichnis. Bei Dateien, die in beiden Verzeichnissen existieren, aber einen unterschiedlichen Inhalt haben, zeigt `diff` die Unterschiede an. `diff -r` lässt sich gut mit der Option `-q` kombinieren.
- `-w`  
ignoriert Leerzeichen und Leerzeilen ganz.

## **Beispiel**

Das folgende Kommando vergleicht den Inhalt der eigenen MySQL-Konfigurationsdatei mit der vom Paket vorgeschlagenen Konfiguration. Die mit `<` beginnenden Zeilen zeigen den aktuellen Zustand an (also Text aus der ersten an `diff` übergebenen Datei), die

mit > beginnenden Zeilen zeigen die entsprechende Passage aus der zweiten Datei.

```
root# diff /etc/mysql/my.cnf /etc/mysql/my.cnf.dpkg-dist
100,101c87,88
< server-id          = 100
< log_bin             = /var/log/mysql/mysql-bin.log
---
> #server-id          = 1
> #log_bin             = /var/log/mysql/mysql-bin.log
...
```

Das »Lesen« von `diff`-Ergebnissen ist gewöhnungsbedürftig.

Naturgemäß gibt es diverse Linux-Programme, die die Unterschiede grafisch visualisieren, z.B. `meld`. Zudem kennen die meisten besseren Editoren einen `diff`-Modus, der die Unterschiede zweier Texte farblich hervorhebt.

**dig** [optionen] [@dnsserver] host/ipaddr [typ]

`dig` führt ähnlich wie [host](#) und [whois](#) Abfragen auf einem Domain-Name-Server aus. Bei `dig` können Sie den Vorgang durch besonders viele Optionen steuern, die hier nicht einmal andeutungsweise wiedergegeben werden können (siehe `man dig`).

In der einfachsten Form `dig host` kontaktiert `dig` den in [/etc/resolv.conf](#) eingestellten DNS-Server und liefert den A-Record. Mit `@1.2.3.4` können Sie die IP-Adresse eines alternativen DNS-Servers angeben. Wenn Sie nicht am A-Record, sondern an anderen Daten interessiert sind, geben Sie deren Typ im dritten Parameter an.

- `-4 / -6`  
verwendet nur IPv4 bzw. nur IPv6.
- `+nocomments`  
verzichtet auf die Ausgabe von Kommentaren.
- `+nostats`  
verzichtet auf die Ausgabe statistischer Daten zur Abfrage (Abfragezeit, -größe etc.).
- `+short`  
formatiert die Ausgabe lesefreundlich. (Standardmäßig liefert `dig`

die Daten in der unübersichtlichen Notation des DNS-Servers bind.)

## Beispiel

Das folgende Kommando kontaktiert den öffentlichen Name-Server von Google, ermittelt dort den MX-Eintrag von `kofler.info` und zeigt diesen in Kurzschreibweise an:

```
user$ dig +short @8.8.4.4 kofler.info MX  
10 mail.kofler.info.
```

**dircolors [optionen] [datei]**

`dircolors` liefert den Programmcode zur Einstellung der Farben, die das Kommando `ls` zur Kennzeichnung verschiedener Dateitypen verwendet. Falls an `dircolors` eine Datei übergeben wird, zeigt das Kommando die in dieser Datei gespeicherten Einstellungen. Die Datei muss die durch `dircolors -p` vorgegebene Syntax einhalten.

- `-b`  
verwendet die Syntax der Shell `bash` (gilt standardmäßig).
- `-c`  
verwendet die Syntax der C-Shell.
- `-p`  
verwendet eine sehr ausführliche Syntax, die von `dircolors` selbst verarbeitet werden kann. Die Datei enthält Kommentare, die die Syntax und insbesondere die Farbcodes erläutern.

## Beispiel

Bei vielen Distributionen wird beim Start der `bash` die Datei `.dircolors` ausgewertet und zur Einstellung der Farben berücksichtigt. Um die `ls`-Farben zu ändern, erzeugen Sie zuerst mit `dircolors -p` eine eigene Version dieser Datei und verändern diese dann mit einem Editor. Die geänderten Einstellungen werden wirksam, sobald Sie ein neues Terminal-Fenster oder Dialogblatt öffnen.

```
user$ cd    (in das Heimatverzeichnis wechseln)  
user$ dircolors -p > .dircolors  
user$ ihrLieblingsEditor .dircolors
```

Sollte die Veränderung der Datei `.dircolors` bei Ihrer Distribution nicht ausreichen, erzeugen Sie daraus mit `dircolor` eine Anweisung zur Einstellung der Variablen `LS_COLORS` und fügen diese Anweisung am Ende der Datei `.bashrc` ein:

```
user$ dircolors .dircolors >> .bashrc
dirname zeichenkette
```

`dirname` liefert den Pfad eines vollständigen Dateinamens. `dirname /usr/bin/groff` liefert also `/usr/bin`.

### **dirs**

Das bash-Kommando `dirs` liefert die Liste aller durch [pushd](#) gespeicherten Verzeichnisse. Das Kommando ist primär zur Script-Programmierung gedacht.

### **disown [optionen] [jobspec]**

Das bash-Kommando `disown` löst den angegebenen Prozess von der Shell. Damit läuft der Prozess weiter, selbst wenn die Shell beendet wird. Der Prozess wird wahlweise durch die Prozess-ID-Nummer (PID) oder durch die Shell-Jobnummer angegeben, wobei der Shell-Jobnummer das Zeichen `%` vorangestellt werden muss. Ohne die Angabe einer Jobnummer ist der aktuelle (Hintergrund-)Prozess gemeint.

Wenn Sie von vornherein planen, ein Programm losgelöst von der Shell auszuführen, können Sie bereits beim Start auf [nohup](#) zurückgreifen. Sie sparen sich so den späteren Aufruf von `disown`.

- `-a`  
löst alle in der Shell gestarteten Prozesse.
- `-r`  
löst alle in der Shell gestarteten Prozesse, die zurzeit ausgeführt werden.

### **Beispiel**

Im folgenden Beispiel wird zuerst ein längerer Download-Prozess gestartet. `[Strg]+[Z]` unterbricht den Download, [bg](#) setzt ihn im

Hintergrund fort. `disown` löst den Download-Prozess von der Shell, sodass der Download auch nach dem Shell-Logout fortgesetzt wird.

```
user$ wget -q http://mirror.switch.ch/.../CentOS-n.n.iso  
user$ <Strg>+<Z>  
user$ bg  
user$ disown
```

### **dmesg [optionen]**

`dmesg` gibt die im Pufferspeicher enthaltenen Kernelmeldungen aus.

- `-C`  
löscht nach der Ausgabe den Pufferspeicher für die Kernelmeldungen. Diese Option erfordert root-Rechte.
- `--follow-new`  
wartet auf neue Kernelmeldungen und zeigt diese an, bis das Kommando mit `[Strg]+[C]` beendet wird. (Diese Option erfordert die erst seit Juli 2020 verfügbare Version 2.36 des Pakets `util-linux`.)
- `-H`  
führt die Ausgabe farbig durch und gibt Uhrzeiten in einer leichter lesbaren Form an (*human readable*).
- `-T`  
zeigt vor jeder Nachricht die absolute Uhrzeit an.
- `-w`  
zeigt alle Kernelmeldungen an und läuft dann weiter, bis das Kommando mit `[Strg]+[C]` beendet wird.

Bei einigen Linux-Distributionen erfordert `dmesg` root-Rechte. Wenn Sie allen Benutzern das Recht geben möchten, Kernelmeldungen zu lesen, tragen Sie `kernel.dmesg_restrict=0` in die Datei `/etc/sysctl.conf` ein und führen dann [`sysctl -p`](#) aus.

### **dnf [optionen] kommando**

`dnf` war ursprünglich das Paketverwaltungskommando der Distribution Fedora. Mittlerweile verwenden auch RHEL 8 und CentOS 8 `dnf`, wenngleich in diesen Distributionen der Aufruf oft

noch via [yum](#) erfolgt. (Intern ist dieses Kommando einfach ein Link auf `dnf`.)

`dnf` funktioniert somit unter Fedora und CentOS/RHEL identisch. Die folgende Zusammenfassung der Optionen gilt dementsprechend für alle aktuellen Red-Hat-verwandten Distributionen.

`dnf` lädt die Pakete von den Paketquellen herunter, die in `/etc/yum.repos.d/*.repo` definiert sind. (Der Name dieses Verzeichnis verdeutlicht die Herkunft von `dnf` und hat sich mit dem Wechsel von [yum](#) zu `dnf` nicht geändert.)

- `check-update`  
testet, ob Updates zu bereits installierten Paketen zur Verfügung stehen.
- `clean metadata`  
entfernt die Paketmetadaten aus dem Cache. Beim nächsten Start aktualisiert `dnf` die Daten, indem es die Metadaten aller Paketquellen neu einliest.
- `clean packages`  
entfernt heruntergeladene und bereits installierte Paketdateien aus dem Cache.
- `config-manager [optionen]`  
verwaltet DNF/YUM-Paketquellen:
  - `--add-repo url` fügt eine Paketquelle hinzu.
  - `--set-enabled name` bzw. -`--set-disabled name` (de)aktiviert eine Paketquelle.
- `copr [enable|disable|remove|list|search] [paketquelle]`  
verwaltet Copr-Paketquellen. Copr ist ein automatisiertes Build-System, das von manchen Entwicklern verwendet wird, um eigene Pakete öffentlich anzubieten. `copr` ist kein eingebautes `dnf`-Kommando. Es kann nur verwendet werden, wenn das Paket `dnf-plugins-core` installiert ist.

- `downgrade package`  
ersetzt ein Paket durch ein Paket aus einer älteren Distribution. Die Versionsnummer der Distribution geben Sie mit `--releasever=n` an.
- `download package`  
lädt ein Paket in das aktuelle Verzeichnis herunter, installiert es aber nicht. In Kombination mit der Option `--source` lädt `dnf` das Quellcodepaket herunter. Mit `--resolve` berücksichtigt das Kommando auch abhängige Pakete. Mit `--url` zeigt es nur die Links zu den Paketen an, lädt die Pakete aber nicht herunter.
- `grouplist`, `groupinfo`, `groupinstall`, `groupupdate`, `groupremove`  
liefert Informationen über Paketgruppen bzw. installiert, aktualisiert oder entfernt Paketgruppen. `dnf grouplist -v` listet zusammen mit jeder Paketgruppe in Klammern die dazugehörige interne englischsprachige Gruppen-ID auf.
- [history](#)  
liefert eine nummerierte Liste der zuletzt durchgeföhrten `dnf`-Aktionen. `dnf history info n` fördert Details zur Transaktion `n` zutage.
- `info name`  
liefert Informationen zum angegebenen Paket.
- `install name1 name2 ...`  
sucht die Pakete `name1`, `name2` etc. aus allen Paketquellen, lädt sie herunter und installiert sie. Gegebenenfalls werden auch weitere Pakete geladen und installiert oder aktualisiert, um Paketabhängigkeiten zu erfüllen.
- `install rpmfile`  
installiert ein RPM-Paket, das im lokalen Dateisystem gespeichert ist. Dabei werden Paketabhängigkeiten berücksichtigt. Fehlende Pakete werden gegebenenfalls aus den Paketquellen heruntergeladen und installiert.

- `install @paket:version`  
installiert das Paket in der angegebenen Version. Das ist eine Kurzschreibweise für `dnf module install paket:version`.
- `list`  
liefert eine zweiteilige Liste. Der erste Teil enthält (in alphabetischer) Reihenfolge alle bereits installierten Pakete, der zweite Teil alle noch nicht installierten Pakete, die in den Paketquellen zur Verfügung stehen. Durch einen optionalen Parameter kann das Ergebnis auf Pakete eingeschränkt werden, deren Name einem Muster entspricht (z.B. `dnf list xorg*`).
- `list available/updates/installed/extras/recent`  
schränkt die Ausgabe von `dnf list` auf bestimmte Pakete ein. Beispielsweise liefert `dnf list updates` eine Liste aller Pakete, zu denen Updates verfügbar sind. `dnf list recent` liefert eine Übersicht der Pakete, die kürzlich in die Paketquellen aufgenommen wurden.
- `makecache [fast]`  
erzeugt einen Cache mit dem Verzeichnis aller auf den Paketquellen verfügbaren Pakete. Mit dem zusätzlichen Schlüsselwort `fast` kontrolliert `dnf`, ob der Cache aktuell ist, und führt die Aktualisierung nur durch, wenn dies wirklich notwendig ist.  
Die Ausführung von `dnf makecache` ist nur in Ausnahmefällen erforderlich. `dnf` überprüft bei den anderen Kommandos automatisch, ob der Cache nicht zu alt ist, und führt gegebenenfalls selbst eine Aktualisierung durch.
- `module`  
`install/update/remove/enable/disable/lock/unlock/list/info`  
...  
verwaltet Pakete, die in unterschiedlichen Versionen zur Verfügung stehen. Diese Erweiterung des `dnf`-Kommandos ist Teil der mit Fedora 28 bzw. RHEL 8 eingeführten Module bzw.

Streams. Bei der Installation müssen Sie zusätzlich zum Paketnamen auch die gewünschte Version und das Profil angeben, also beispielsweise `dnf module install nodejs:13/development`.

- `remove name1 name2 ...`  
deinstalliert die angegebenen Pakete.
- `repoquery name`  
ermittelt den vollständigen Paketnamen für das angegebene Paket. Es spielt keine Rolle, ob das Paket bereits installiert ist oder nicht.

Mit der zusätzlichen Option `-i` liefert `dnf repoquery` in tabellarischer Form eine Menge weiterer Metadaten, z.B. die Versionsnummer, die Lizenz, die Paketquelle und die Projekt-URL. Mit der Option `--location` gibt das Kommando eine Adresse an, von der das Paket manuell heruntergeladen werden kann.

- `search suchbegriff`  
liefert eine Liste aller Pakete, die den Suchbegriff in der Beschreibung enthalten.
- `update`  
aktualisiert alle installierten Pakete.
- `update name1 name2 ...`  
aktualisiert nur die angegebenen Pakete.
- `upgrade`  
hat dieselbe Wirkung wie `update` in Kombination mit der Option `--obsoletes` (siehe unten).

Sie können das Verhalten von `dnf` durch einige (sehr selten benötigte) Optionen beeinflussen:

- `--enablerepo=name`  
aktiviert eine an sich deaktivierte Paketquelle. Die Option erspart es Ihnen, die `*.repo`-Datei zu ändern, wenn Sie nur ein einzelnes Paket einer nicht aktiven Paketquelle installieren möchten.

- **--exclude=paket**  
schließt das angegebene Paket von der gewählten Operation (z.B. einem Update) aus.
- **--obsoletes**  
bewirkt, dass `dnf` nicht mehr benötigte Pakete nach einem Update löscht. Das ist nur bei einem Distributions-Update (Version  $n$  zu  $n+1$ ) zweckmäßig.
- **-y**  
beantwortet alle Fragen von `dnf` mit *yes*. Damit kann `dnf` zur Installation ohne weitere Interaktion eingesetzt werden (z.B. in einem Script).

## Beispiele

Die beiden folgenden Kommandos aktualisieren zuerst alle bereits vorhandenen Pakete und installieren dann den Editor Emacs:

```
root# dnf update
...
Transaction Summary
Install      1 Package(s)
Update      42 Package(s)
Remove       0 Package(s)
Total download size: 52 M
Is this ok [y/N]: y
...

root# dnf install emacs
Installing:
 Package      Architecture Version      Repository      Size
 emacs        x86_64      1:26.1-5.el8   ol8_appstream  3.2 M
Installing dependencies:
 emacs-common x86_64      1:26.1-5.el8   ol8_appstream  38 M
 libXaw       x86_64      1.0.13-10.el8  ol8_appstream  194 k
 libblockfile x86_64      1.14-1.el8    ol8_appstream  32 k
 libotf       x86_64      0.9.13-11.el8  ol8_appstream  104 k
 m17n-db     noarch      1.8.0-3.el8   ol8_appstream  581 k
 m17n-lib     x86_64      1.8.0-2.el8   ol8_appstream  201 k
...
```

`dnf repoquery` (hier ausgeführt unter Oracle Linux 8) liefert eine Menge Metadaten zu einem Paket:

```
root# repoquery -i joe
Name      : joe
Version   : 4.6
Repository: epel
Summary   : An easy to use, modeless text editor
```

URL : <http://sourceforge.net/projects/joe-editor/>

...

Die folgenden beiden Kommandos ermitteln, in welchen Varianten das Paket nodejs zur Verfügung steht, und installieren dann die neueste Version im minimal-Profil:

```
root# dnf module list --all nodejs
Extra Packages for Enterprise Linux Modular 8 - x86_64
Name      Stream Profiles                               Summary
nodejs    12     default, development, minimal          Javascript runtime
nodejs    13     default, development, minimal          Javascript runtime

Oracle Linux 8 Application Stream (x86_64)
Name      Stream Profiles                               Summary
nodejs    10 [d] common [d], development, minimal, s2i Javascript runtime
nodejs    12     common [d], development, minimal, s2i Javascript runtime

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
```

```
root# dnf module install nodejs:13/minimal
```

Das letzte Kommando lädt das Quellcodepaket des Gnome-Texteditors herunter:

```
user$ dnf download --source gedit
docker [optionen] kommando
```

Das Kommando docker verwaltet Docker-Container. Vor der ersten Verwendung des Kommandos müssen Sie docker installieren, idealerweise aus einer aktuellen Docker-Paketquelle (siehe <https://docs.docker.com/install>).

- **build verzeichnis/url**  
liest die am angegebenen Ort befindliche Datei Dockerfile und erzeugt daraus ein entsprechendes lokales Image. Mit der Option **-t name:tag** kann das Image benannt und mit einem Tag gekennzeichnet werden.
- **create imagename/id [kommando]**  
erzeugt ähnlich wie docker run einen Container, startet diesen aber nicht.
- **exec containername/id kommando**  
führt das angegebene Kommando in einem bereits laufenden Container aus. Wie bei docker run ermöglichen die Optionen -i

und -t eine interaktive Bedienung. Mit -d wird das Kommando im Hintergrund ausgeführt.

- `image prune|pull|push|rm [imagename/id]`  
löscht alle ungenutzten Images, lädt Images vom Docker Hub herunter bzw. auf ihn hoch bzw. löscht ein einzelnes Image.  
(Docker Hub ist eine Sammlung öffentlicher Docker-Images, siehe <https://hub.docker.com>.)
- `images`  
listet lokale Images auf und gibt deren Größe an. Mit der Option -q werden nur die Image-IDs aufgelistet, was eine automatisierte Weiterverarbeitung ermöglicht.
- `inspect containername/id`  
liefert detaillierte Informationen zum angegebenen Container.
- `logs containername/id`  
zeigt die protokollierten Nachrichten des Containers an. Das funktioniert nur, wenn das zugrunde liegende Image entsprechend konfiguriert ist.
- `network create|connect|inspect|rm [networkname/id]`  
administriert Netzwerke für Docker-Container.
- `node ls|inspect|rm [nodename/host]`  
verwaltet Docker-Instanzen in einem Cluster (»Schwarm«).
- `port containername/id kommando`  
listet die Port-Zuordnungen eines Containers auf.
- `ps`  
gibt alle laufenden Container aus. Mit der Option -a werden auch gestoppte Container berücksichtigt. Fügen Sie -s hinzu, zeigt das Kommando auch die Container-Größe an.
- `rename altername/id neuername`  
benennt einen Container um.
- `rm containername/id`  
löscht den angegebenen Container. Mit -f wird das Kommando selbst dann ausgeführt, wenn der Container noch läuft.

- `rmi imagename/id`  
löscht das angegebene Image.
- `run imagename/id [kommando]` erzeugt aus dem angegebenen Image einen neuen Container, startet diesen und führt darin entweder das Default-Kommando oder das angegebene Kommando aus. Das erforderliche Image wird vom Docker Hub heruntergeladen, wenn es nicht schon lokal zur Verfügung steht. Die Eigenschaften des neuen Containers werden durch unzählige Optionen gesteuert. Die wichtigsten sind:
  - `d` führt den Container im Hintergrund aus.
  - `e var=value` definiert eine Umgebungsvariable für den Container.
  - `h hostname` weist dem Container einen Hostnamen zu.
  - `i -t` ermöglicht eine interaktive Bedienung des Containers.
  - `--name cname` weist dem Container einen Namen zu.
  - `--network netname` verwendet das angegebene Docker-Netzwerk.
  - `p localport:cport` verbindet einen lokalen Port mit einem Port des Containers.
  - `--rm` löscht den Container wieder, sobald die Ausführung endet.
  - `v localdir:cdir` verbindet ein lokales Verzeichnis mit einem Volume des Containers.
- `service create|inspect|logs|ls|ps|remove|update`  
administriert die Services in einem Docker-Cluster (Schwarm).
- `stack deploy ls|ps|rm|services`  
administriert Stacks (Gruppen von Services).
- `start/stop containername`  
startet bzw. stoppt einen bereits eingerichteten Container.
- `swarm init|join|leave`  
verwaltet einen Docker-Cluster (Schwarm).
- `system df`  
listet den Platzbedarf aller Images, Container und Volumes auf.

- `system info`  
zeigt umfassende Informationen zur Docker-Installation an.
- `system prune`  
löscht nicht laufende Container, ungenutzte Images und mit der Option `--volumes` auch Volumes.
- `tag containername/id neuename:tag`  
benennt einen Container neu.
- `volume ls|inspect|prune|rm`  
verwaltet Volumes.

## Beispiel

`docker run` erzeugt vom offiziellen MariaDB-Image auf Docker Hub einen abgeleiteten Container und gibt diesem den Namen `mydb`. Falls das Image noch nicht lokal heruntergeladen wurde, kümmert sich `docker run` darum. Die Ausführung des Kommandos dauert dann beim ersten Mal etwas länger. Alle Datenbankdateien werden in `/home/<username>/var/lib/mysql` gespeichert. Der Port 3306 des Containers ist im Hostsystem über den lokalen Port 13306 zugänglich.

```
root# mkdir /home/<username>/var/lib/mysql
root# docker run -d --name mydb -p 13306:3306 \
    -v /home/<username>/var/lib/mysql/:/var/lib/mysql mariadb
```

Unter CentOS/Fedora/RHEL müssen Sie das Argument für die Option `-v` um `:z` ergänzen. Andernfalls blockiert SELinux den Zugriff auf das Verzeichnis:

```
root# mkdir /home/<username>/var/lib/mysql
root# docker run -d --name mydb -p 13306:3306 \
    -v /home/<username>/var/lib/mysql/:/var/lib/mysql:z mariadb
```

Der Container läuft im Hintergrund, bis er mit `docker stop` beendet wird:

```
root# docker stop mydb
```

Wenn Sie den Container nicht mehrlich verwenden möchten, können Sie ihn nun löschen:

```
root# docker rm mydb
```

**docker-compose [optionen] kommando**

`docker-compose` wertet die Datei `docker-compose.yml` im lokalen Verzeichnis aus, erzeugt alle dort beschriebenen Container und startet diese. Wenn Sie mehrere Container in einer Gruppe kombinieren möchten, ist der Umgang mit `docker-compose` wesentlich effizienter als das manuelle Starten aller Container mit [`docker run`](#).

Alle `docker-compose`-Kommandos beziehen sich auf die im aktuellen Verzeichnis befindliche Datei `docker-compose.yml`. Fehlt diese Datei, liefert das Kommando eine Fehlermeldung.

- [`config`](#)  
testet, ob `docker-compose.yml` frei von Syntaxfehlern ist.
- [`down`](#)  
stoppt alle in `docker-compose.yml` beschriebenen Container, Netzwerke etc. und löscht sie. Das Kommando entspricht der Kombination aus `down` und [`rm`](#).
- [`logs servicename`](#)  
zeigt die Protokollausgaben des angegebenen Service (Containers).
- [`ps`](#)  
listet alle Container des Projekts auf.
- [`rm`](#)  
löscht die zuvor gestoppten Container.
- `stop|start|restart`  
beendet die Ausführung aller Container bzw. startet die Container wieder.
- [`top`](#)  
zeigt die Prozesse aller Container an.
- `up`  
erzeugt die in `docker-compose.yml` definierten Container und startet sie. Mit der üblicherweise verwendeten Option `-d` erfolgt die Ausführung im Hintergrund (*detached*).

## Beispiel

Als Ausgangspunkt für das Beispiel gilt die folgende Datei:

```
# Datei test/docker-compose.yml
version: '3'
services:
  db:
    image: mariadb:latest
    volumes:
      - /var/dc-test-db:/var/lib/mysql
    environment:
      MYSQL_ROOT_PASSWORD: geheim
      restart: always

  wordpress:
    image: wordpress:latest
    volumes:
      - /var/dc-test-www:/var/www/html
    ports:
      - "8082:80"
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_PASSWORD: geheim
      restart: always
```

Eine Referenz für die in docker-compose.yml zulässige Syntax finden Sie hier:

<https://docs.docker.com/compose/compose-file> Nun erzeugen Sie zuerst die Volume-Verzeichnisse und dann die Container:

```
root# cd test
root# mkdir /var/dc-test-www
root# mkdir /var/dc-test-db
root# docker-compose up -d
  Creating network "composetest_default" with the default driver
...
  Creating composetest_wordpress_1 ... done
```

Die Container werden sofort gestartet und laufen im Hintergrund, wovon Sie sich mit docker ps überzeugen können:

```
root# docker ps
  ID        IMAGE       PORTS          NAMES
  5211...  wordpress...  0.0.0.0:8082->80/tcp  test_wordpress_1
  d9dc...  mariadb...   3306/tcp        test_db_1
```

Das folgende Kommando stoppt und löscht beide Container:

```
root# docker-compose down
dpkg optionen [dateiname/paketname]
```

dpkg erledigt die Low-Level-Paketverwaltung in allen Debian- und Ubuntu-basierten Distributionen. Der Benutzer verwendet aber in

der Regel nicht `dpkg` zur Paketinstallation, sondern ein darauf aufbauendes Kommando, meist [apt-get](#).

- `--configure paketname`  
führt die Konfigurations-Scripts des angegebenen Pakets aus. Normalerweise geschieht das bereits während der Installation; in manchen Fällen kann es aber notwendig sein, diesen Schritt explizit auszuführen. Bei manchen Paketen gibt es zusätzlich zu den automatischen Konfigurations-Scripts interaktive Setup-Programme. Wenn Sie ein derartiges Setup-Programm später nochmals benötigen, müssen Sie `dpkg-reconfigure paketname` ausführen.
- `--get-selections`  
liefert ähnlich wie `--list` eine Liste aller installierten Pakete. Das Ergebnis enthält aber weniger Detailinformationen und ist damit viel übersichtlicher. Wenn Sie die Liste in eine Textdatei umleiten und speichern, können Sie alle Pakete später auf einem anderen Rechner mit `--set-selections` installieren.
- `-i bzw. --install dateiname.deb`  
installiert die angegebene(n) Paketdatei(en). Wenn bereits eine ältere Version installiert ist, wird diese deinstalliert und durch die neue Version ersetzt. Während der Installation werden auch die im Paket vorgesehenen Konfigurations-Scripts ausgeführt. Vor der Installation wird sichergestellt, dass alle Paketabhängigkeiten erfüllt sind. Wenn das nicht der Fall ist, geht aus der Fehlermeldung zumeist hervor, welche Pakete fehlen. Die fehlenden Abhängigkeiten werden zudem gespeichert. Sie können die fehlenden Pakete deswegen unkompliziert mit [apt install -f](#) aus den Paketquellen installieren und dann mit `dpkg -i name.deb` die Installation des lokalen Pakets wiederholen. Noch komfortabler ist es, die Installation einer lokalen Paketdatei mit [gdebi](#) durchzuführen.

`dpkg` lässt normalerweise nur die Installation von Paketen in der zur Distribution passenden Architektur zu. Wenn Sie mit einer 64-Bit-Linux-Distribution arbeiten, ein Programm aber nur als 32-Bit-Paket verfügbar ist, können Sie die Installation mit `--force-architecture` erzwingen. Ob das Programm dann tatsächlich funktioniert, ist eine andere Frage. Auf jeden Fall müssen Sie auch alle erforderlichen 32-Bit-Bibliotheken installieren.

- `-l` bzw. `--list`  
liefert eine Liste aller installierten Pakete.
- `-l` bzw. `--list 'muster'`  
liefert eine Liste der installierten und verfügbaren Pakete. Bei der Paketliste wird in der ersten Spalte ein Code aus zwei Buchstaben angegeben.

Der erste Buchstabe gibt den gewünschten Status des Pakets an:

`i` = installieren

`n` = nicht installieren

`r/p` = entfernen (*remove*)

`h` = halten (*hold*)

Der zweite Buchstabe beschreibt den tatsächlichen Status:

`i` = installiert

`n` = nicht installiert

`c` = konfiguriert (*configured*)

`u` = entpackt, aber noch nicht konfiguriert (*unpacked*)

`f` = fehlgeschlagen (*failed*)

Die Informationen stammen aus der Debian-Paketdatenbank, einer Sammlung von Dateien im Verzeichnis `/var/lib/dpkg`. Dort werden Meta-Informationen über alle installierten und verfügbaren Pakete gespeichert.

- `-L` bzw. `--listfiles paketname`  
liefert eine Liste aller Dateien des angegebenen Pakets. Das funktioniert nur für bereits installierte Pakete. Den Inhalt nicht

installierter Pakete ermitteln Sie mit dem Kommando `dpkg -d --contents dateiname`.

- **-P bzw. --purge paketname**  
entfernt das angegebene Paket inklusive aller Paketdateien (auch wenn diese von Ihnen verändert wurden).
- **-r bzw. --remove paketname**  
entfernt das angegebene Paket.
- **-s bzw. --search dateiname**  
ermittelt das Paket, aus dem die angegebene Datei stammt.

## Beispiel

Das erste dpkg-Kommando installiert ein neues Paket. `dpkg --search` ermittelt das Paket, von dem die Datei `/etc/sensors3.conf` stammt.  
`dpkg --listfiles` liefert eine Liste aller Dateien dieses Pakets.

```
root# dpkg --install paketname.deb
root# dpkg --search /etc/sensors3.conf
libsensors-config: /etc/sensors3.conf
root# dpkg --listfiles libsensors4
...
/
/etc
/etc/sensors.d
/etc/sensors.d/.placeholder
/etc/sensors3.conf
...
```

**dracut [optionen] initrd-datei kernelversion**

dracut ist unter CentOS, Fedora, RHEL und (open)SUSE für das Erzeugen einer Initrd-Datei zuständig. Wenn dracut ohne weitere Parameter ausgeführt wird, erzeugt es für den neuesten Kernel im Verzeichnis `/boot` eine Initrd-Datei mit dem Namen `/boot/initrd-kernelversion`. dracut berücksichtigt die Einstellungen in `/etc/dracut.conf` sowie zahllose Optionen, die in *man dracut* beschrieben sind. Hier sind nur die drei wichtigsten Optionen zusammengefasst:

- **-d a, b, c**  
baut die Kernelmodule a, b und c in die Initrd-Datei ein.  
Normalerweise ist diese Option nicht erforderlich. dracut erkennt

selbstständig, welche Module für den Startprozess erforderlich sind.

- **-f**  
überschreibt eine vorhandene Initrd-Datei.
- **--regenerate-all**  
erzeugt neue Initrd-Dateien zu allen auf dem System installierten Kernel-Versionen.

Debian und Ubuntu verwenden stattdessen [update-initramfs](#).

## Beispiel

Um für einen selbst kompilierten Kernel 5.7.3 in der Datei */boot/vmlinuz-5.7.3* manuell eine Initrd-Datei zu erzeugen, führen Sie das folgende Kommando aus:

```
root# dracut /boot/initrd-5.7.3 5.7.3
```

```
du [optionen] [verzeichnis]
```

du gibt Informationen über den Speicherbedarf von Dateien bzw. von Verzeichnissen aus. Wenn im Verzeichnisparameter eine Dateispezifikation angegeben wird (etwa \* oder \*.tex), dann liefert du eine Liste mit der Größe aller Dateien. Wird dagegen nur ein Verzeichnis angegeben, ermittelt du den Speicherbedarf für alle untergeordneten Verzeichnisse.

Die Speicherangaben umfassen auch den Speicherbedarf aller untergeordneten Verzeichnisse. Der letzte Zahlenwert gibt den Gesamtspeicherbedarf aller Dateien und Unterverzeichnisse ab dem angegebenen Verzeichnis an. Alle Angaben erfolgen in KiB.

- **-b bzw. --bytes**  
zeigt die Größenangaben in Byte (statt in KiB) an.
- **-c bzw. --total**  
zeigt als abschließenden Wert die Endsumme an. Diese Option ist nur notwendig, wenn du auf Dateien (und nicht auf Verzeichnisse) angewandt wird. Mit dieser Option kann relativ einfach festgestellt werden, wie viel Speicher alle Dateien mit einer bestimmten Kennung (z.B. \*.pdf) beanspruchen.

- **-h** bzw. **--human-readable**  
zeigt die Größenangaben in einer gut lesbaren Form an. **K**, **M** und **G** gelten als Abkürzungen für KiB, MiB oder GiB.
- **--max-depth=n** bzw. **--max n**  
gibt die Verzeichnisgröße nur für die angegebene Anzahl von Verzeichnisebenen aus.
- **-s** bzw. **--summarize**  
zeigt *nur* die Endsumme an. Diese Option ist nur dann sinnvoll, wenn der Speicherbedarf von Verzeichnissen angezeigt wird.
- **-s** bzw. **--dereference**  
zeigt nur den Speicherbedarf unmittelbar im Verzeichnis an. Der Speicherbedarf in Unterverzeichnissen wird *nicht* berücksichtigt.
- **-x** bzw. **--one-file-system**  
ignoriert Verzeichnisse, an denen andere Dateisysteme eingebunden sind. (Diese Option ist fast immer empfehlenswert. Auf meinem Rechner habe ich mit `alias du="du -x"` einen entsprechenden Alias definiert.)  
  
du bietet keine Möglichkeit, das Ergebnis zu sortieren. Genau das kann das interaktive Kommando `ncdu`. Es bietet darüber hinaus die Möglichkeit, mit den Cursortasten durch Unterverzeichnisse zu navigieren. Sehr empfehlenswert!

## Beispiel

du ermittelt, welche Verzeichnisse wie viel Platz beanspruchen. Mit `-max 1` werden Unterverzeichnisse nicht einzeln angeführt, sind aber im Ergebnis berücksichtigt.

```
user$ du -h --max 1
15M      ./Downloads
29M      ./Videos
69M      ./mozilla
...
```

## dumpe2fs device

dumpe2fs liefert unzählige interne Informationen über den Zustand eines ext-Dateisystems, insbesondere über die sogenannten

Superblöcke sowie über die Organisation der Blockgruppen des Datenträgers.

- **-b**  
listet nur defekte Blöcke des Datenträgers auf.
- **-h**  
liefert nur eine Zusammenfassung der Daten des Superblocks.

**dvips** [optionen] name.dvi

**dvips** erzeugt aus einer \*.dvi-Datei eine PostScript-Datei. Wenn **dvips** ohne die Option **-o** eingesetzt wird, leitet das Kommando die resultierende PostScript-Datei an den Standarddrucker weiter.

- **-A**  
wandelt nur ungerade Seiten um.
- **-B**  
wandelt nur gerade Seiten um.
- **-D n**  
verwendet bei der Erzeugung von LaTeX-Bitmap-Schriften eine Auflösung von *n* dpi (dots per inch). Die Standardauflösung beträgt meist 600 dpi (siehe /etc/texmf/config.ps). Alternativ darf *n* auch 300, 400 oder 1270 betragen (siehe /usr/bin/mktexpk). Die Option ist nur für Bitmap-Schriften relevant. PostScript-Schriften sind immer auflösungsunabhängig.
- **-E**  
erzeugt eine EPS-Datei (Encapsulated PostScript) mit einer Boundingbox, die nur den tatsächlich genutzten Teil der Seite umfasst. Das ist nur sinnvoll, wenn die DVI-Datei nur eine Seite umfasst und die resultierende EPS-Datei anschließend in ein anderes Dokument eingebettet werden soll.
- **-i -s n**  
zerlegt die Ausgabe in Dateien zu je *n* Seiten. Die Dateien werden automatisch durchnummieriert.

- **-l *letzteseite***  
beendet die Umwandlung mit der angegebenen Seite.
- **-o *zieldatei***  
schreibt das Ergebnis in die angegebene Datei (anstatt es an das Programm [lpr](#) weiterzuleiten).
- **-p *ersteseite***  
beginnt die Umwandlung mit der angegebenen Seite.
- **-pp *n1, n2-n3, n4, n5, n6-n7***  
druckt die angegebenen Seiten. Beachten Sie, dass in der Seitenliste keine Leerzeichen vorkommen dürfen.

Globale Standardeinstellungen für *dvips* sind in der Datei *config.ps* definiert. Diese Datei befindet sich je nach Distribution z.B. im Verzeichnis */etc/texmf* oder in */usr/share/texlive/texmf-dist/dvips/config*.

## E

### **e4defrag** [optionen] datei/verzeichnis/device

e4defrag defragmentiert wahlweise eine Datei, den gesamten Inhalt eines Verzeichnisses oder ein ganzes ext4-Dateisystem. Benutzer ohne root-Rechte können nur ihre eigenen Dateien defragmentieren.

Defragmentierung bedeutet, dass die Blöcke einer Datei möglichst aneinanderreihend auf dem Datenträger reserviert werden. Das beschleunigt besonders bei herkömmlichen Festplatten die Lesezugriffe. Bei SSDs sind die Effekte der Defragmentierung weniger stark ausgeprägt.

- -c  
zeigt einen Defragmentierungszähler für die betreffende(n) Datei(en) an, führt aber keine Änderungen durch. Wenn e4defrag mit dieser Option für das gesamte Dateisystem ausgeführt wird, liefert das Kommando nach geraumer Zeit eine Liste der Dateien, die am stärksten fragmentiert sind.
- -v  
zeigt Informationen vor und nach der Defragmentierung jeder Datei an.

### **echo** [optionen] zeichenkette

Das bash-Kommando echo gibt die angegebene Zeichenkette aus. Wenn die Zeichenkette Leer- oder Sonderzeichen enthält, muss sie in doppelte oder einfache Apostrophe eingeschlossen werden, je nachdem, ob Shell-Variablen ausgegeben werden sollen oder nicht.

- -e  
interpretiert diverse Backslash-Zeichenkombinationen, z.B. \a als Signaltón, \n als Zeilenende und \t als Tabulator (siehe auch help echo). Somit gibt echo -e "\a" einen Warnton aus.
- -n  
wechselt beim Ende der Ausgabe nicht in eine neue Zeile. Die

Ausgabe kann durch eine weitere echo-Anweisung fortgesetzt werden.

## Beispiel

Das folgende Kommando zeigt den aktuellen Inhalt der \$PATH-Variablen an:

```
user$ echo "Die PATH-Variable enthält: $PATH"  
efibootmgr [optionen]
```

In Linux-Systemen, die im EFI-Modus installiert und gestartet wurden, können über das Kommando `efibootmgr` diverse EFI-Einstellungen verändert werden. Diese Einstellungen werden in einem nichtflüchtigen Speicher (NVRAM) des Mainboards gespeichert. `efibootmgr` dient primär dazu, neue EFI-Booteinträge einzurichten, vorhandene Einträge zu entfernen und die Standard-Bootreihenfolge festzulegen. Das Kommando setzt voraus, dass das Kernelmodul `efivars` geladen ist. Sollte das nicht der Fall sein, führen Sie [modprobe efivars](#) aus.

Beachten Sie, dass das `efivars`-Modul nur verwendet werden kann, wenn Linux im EFI-Modus gebootet wurde (nicht im BIOS-Modus!). Verwenden Sie für Reparaturarbeiten gegebenenfalls ein Linux-Live-System, das sich im EFI-Modus starten lässt.

- `-b n -B`  
löscht den angegebenen EFI-Booteintrag. Die Nummern der EFI-Booteinträge ermitteln Sie, indem Sie `efibootmgr` ohne Parameter ausführen.
- `-c -l \EFI\distrib\grubdatei.efi -L name`  
erzeugt einen neuen EFI-Booteintrag (-c) und gibt an, an welchem Ort sich die dazugehörende Bootloader-Datei befindet. Die Pfadangabe ist relativ zur EFI-Partition `/boot/efi`, und als Verzeichnistrenner muss `\` verwendet werden. -L gibt den Namen des EFI-Booteintrags an (üblicherweise den Distributionsnamen, standardmäßig `Linux`). Der neue Eintrag wird automatisch zum Default-Eintrag. Optional können Sie mit -p die Nummer der EFI-

Bootpartition (normalerweise 1) und mit -d den Device-Namen der ersten Festplatte/SSD angeben (normalerweise `/dev/sda`).

- `-n n`  
bestimmt, welcher EFI-Booteintrag beim nächsten Neustart verwendet wird. Die Einstellung ist nicht bleibend, sondern gilt nur für den nächsten Rechnerstart.
- `-N`  
löscht die mit `-n` festgelegte Einstellung für den nächsten Neustart.
- `-o n1,n2,n3,...`  
legt die neue Reihenfolge der EFI-Booteinträge bleibend fest.  
Wenn Sie nur den Eintrag an der ersten Stelle festlegen möchten, reicht es aus, dessen Nummer anzugeben.
- `-p n`  
legt fest, in welcher Partition sich die EFI-Bootpartition befindet (normalerweise in der ersten).
- `-q`  
keine Ausgaben durchführen (*quiet*).
- `-t`  
gibt an, wie lange EFI bei einem Neustart mit der Aktivierung des Default-Booteintrags warten soll (in Sekunden). Es hängt allerdings von der EFI-Implementierung ab, ob das EFI-Menü in dieser Wartezeit automatisch angezeigt wird. Auf einem meiner Testrechner wird das EFI-Menü generell nur angezeigt, wenn ich eine entsprechende Taste drücke (`[F8]`, Asus-Mainboard P8H67-M Evo).
- `-T`  
löscht die mit `-t` eingestellte Wartezeiteinstellung, d.h., der Default-Booteintrag wird sofort aktiviert.

## Beispiel

Wenn das Kommando ohne weitere Optionen ausgeführt wird, listet es die EFI-Booteinträge sowie einige weitere Parameter des EFI-Bootloaders auf:

```
root# efibootmgr
BootCurrent: 0000
Timeout: 1 seconds
BootOrder: 0000,0005,0003,0001,0002
Boot0000* ubuntu
Boot0001* Hard Drive
Boot0002* CD/DVD-Laufwerk
Boot0003* Windows Boot Manager
Boot0005* Fedora
```

Beim nächsten Neustart soll Fedora gestartet werden (abweichend von der gespeicherten Bootreihenfolge, in der Ubuntu an erster Stelle steht):

```
root# efibootmgr -n 5
```

Das nächste Kommando erzeugt einen neuen EFI-Booteintrag. Die Verdoppelung der \ Zeichen ist erforderlich, weil die Shell ein einfaches \ Zeichen als Kennzeichnung von Sonderzeichen interpretiert.

```
root# efibootmgr -c -l \\EFI\\test\\abc.efi -L abc
egrep [optionen] suchmuster datei
```

egrep durchsucht die angegebene Textdatei nach einem Suchmuster und zeigt die gefundenen Textpassagen an. egrep ist kein eigenes Kommando, sondern lediglich ein Alias (oder ein kleines Script), das das Kommando [grep](#) mit der Option -E ausführt. Damit werden *Extended Regular Expressions* (EREs) aktiviert, also eine erweiterte Syntax für das Suchmuster. Weitere Informationen und Beispiele finden Sie bei [grep](#).

```
enscript [optionen] quelldatei -p zieldatei
```

Das Kommando [enscript](#) aus dem gleichnamigen Paket wandelt eine Textdatei wahlweise in die Formate PostScript, HTML oder RTF um.

- **--color**

Syntaxhervorhebung mit Farben durchführen (muss mit -E kombiniert werden).

- **-E**  
führt bei Programmcode eine Syntaxhervorhebung durch (fett, kursiv).
- **-f font**  
verwendet die angegebene Schriftart (standardmäßig Courier in einer Größe von 10 Punkt).
- **-M *papierformat***  
verwendet das angegebene Papierformat (z.B. A3, A4 oder Letter).
- **--n**  
formatiert den Text mit  $n$  Spalten. -2 bewirkt somit eine zweispaltige Formatierung.
- **-r bzw. --landscape**  
füllt das Blatt im Querformat.
- **-w *format* bzw. --language=*format***  
gibt das gewünschte Format an. Zur Auswahl stehen unter anderem PostScript (gilt standardmäßig), html und rtf.
- **-X *charset***  
gibt den Zeichensatz des Texts an (z.B. ascii, latin). Unicode wird leider nicht unterstützt. Um Unicode-Texte in das PostScript-Format zu konvertieren, verwenden Sie das Kommando [paps](#).

**epstopdf** [optionen] datei.eps

epstopdf wandelt die EPS-Datei in eine PDF-Datei um und speichert das Ergebnis unter dem Namen *datei.pdf*.

- **--exact**  
wertet die ExactBoundingBox aus (anstatt der normalen Boundingbox).
- **--hires**  
wertet die HiresBoundingBox aus (anstatt der normalen Boundingbox).

- `--nocompress`  
verzichtet auf eine Komprimierung der PDF-Daten.
- `--outfile=name`  
speichert das Ergebnis unter dem angegebenen Dateinamen.

### **etherwake [optionen] mac**

Das Kommando `etherwake` aus dem gleichnamigen Paket sendet ein spezielles Netzwerkpaket an eine MAC-Adresse und versucht damit, das Gerät aus dem Ruhezustand »aufzuwecken«. Das funktioniert nur, wenn das Gerät den Standard *Wake On LAN* unterstützt.

- `-i interface`  
versendet das Paket über die angegebene Netzwerkschnittstelle (standardmäßig `eth0`).

### **Beispiel**

Das folgende Kommando aktiviert in meinem Haushalt das NAS-Gerät, das im Keller steht und sich normalerweise aus Energiespargründen im Ruhezustand befindet:

```
root# etherwake -i wlp0s20f3 00:11:32:12:DD:0D
```

### **ethtool [optionen] device [parameter]**

`ethtool` ermittelt bzw. verändert (Hardware-)Parameter von Ethernet-Adaptoren. Wenn Sie an das Kommando einfach einen Schnittstellennamen ohne weitere Optionen übergeben, erhalten Sie eine Auflistung der wichtigsten Eckdaten. Dazu zählen die unterstützten Ports, die Link-Modi, die Geschwindigkeit, der Duplex-Modus etc.

- `-p device n`  
schaltet für *n* Sekunden die LED der Schnittstelle ein. Bei Servern mit mehreren Adaptoren vereinfacht das die Identifizierung der Steckerbuchse. Bei manchen Adaptoren blinkt die LED dann für die angegebene Zeit. Andere Adapter unterstützen gar keine LED-Steuerung – dann kommt es zur Fehlermeldung *operation not supported*.

- `-s device para1 value1 para2 value2 ...`  
verändert mehrere Parameter der Schnittstelle. Zulässige Parameternamen sind unter anderem speed, duplex, port, mdix, autoneg und advertise.

## Beispiel

Das folgende Kommando stellt die Geschwindigkeit des Netzwerkadapters explizit mit 100 MBit/Sekunde ein, aktiviert den Duplex-Modus und deaktiviert die Auto-Negotiate-Funktion:

```
root# ethtool -s eth0 speed 100 duplex full autoneg off  
eval $var
```

Das bash-Kommando eval interpretiert den Inhalt der Variablen als Kommandozeile, wertet diese Zeile mit allen bekannten Substitutionsmechanismen aus und führt das Kommando schließlich aus. eval ist immer dann erforderlich, wenn ein in einer Variablen gespeicherte Kommando ausgeführt werden soll und dieses Kommando diverse Sonderzeichen der Shell enthält.

## Beispiel

Erst mit der Verwendung von eval kann das in der Variablen kom gespeicherte Kommando ausgeführt werden. Der erste Versuch, das Kommando auszuführen, scheitert, weil die bash das Pipe-Zeichen | nicht mehr auswertet, nachdem sie \$kom durch seinen Inhalt ersetzt hat.

```
user$ kom="ls | more"  
user$ $kom  
ls: |: No such file or directory  
ls: more: No such file or directory  
user$ eval $kom  
exec kommando
```

Das bash-Kommando exec startet das im Parameter angegebene Kommando als Ersatz zur laufenden Shell. Das Kommando kann beispielsweise dazu verwendet werden, eine andere Shell zu starten. Die laufende Shell wird dadurch beendet. Bei einem normalen Kommandostart ohne exec läuft die Shell hingegen weiter und

nimmt nach dem Ende des Kommandos weitere Eingaben entgegen bzw. setzt das laufende Script fort.

**exfatlabel** device [label]

`exfatlabel` liest den Namen eines exFAT-Dateisystems aus bzw. verändert diesen. Das Kommando befindet sich bei vielen Distributionen im Paket `exfat-utils`. Zum Einrichten eines exFAT-Dateisystems verwenden Sie [`mkfs.exfat`](#).

**exiftool** [optionen] datei/verzeichnis

Das Perl-Script `exiftool` aus dem gleichnamigen Paket kann die Exif-Daten aus JPEG- und TIFF-Dateien extrahieren bzw. verändern.

Das *Exchangeable Image File Format* beschreibt, wie in Bitmapdateien zusätzliche Metadaten gespeichert werden, z.B. der Aufnahmezeitpunkt eines Fotos. Wird das Kommando ohne Optionen aufgerufen, listet es einfach sämtliche Exif-Informationen der angegebenen Datei auf.

- "`-tag`"  
liest nur die angegebene Information aus, z.B. mit `-CreateDate` das Aufnahmedatum.
- "`-tag=value`"  
überschreibt die Information durch den angegebenen Wert.
- "`-tag1<tag2`"  
überschreibt ein *Tag* durch ein anderes *Tag*. Dabei kann eine mit `-d` angegebene Formatierungszeichenkette berücksichtigt werden (siehe das folgende Beispiel).
- `-lang de`  
gibt die Exif-Elemente in deutscher Sprache aus.
- `-v`  
verrät, was gerade passiert (*verbose*).

Unzählige weitere Optionen und Anwendungsvarianten sind auf der Webseite des Tools dokumentiert:

<https://exiftool.org>

## Beispiel

Das folgende Kommando benennt alle Bilder im aktuellen Verzeichnis um. Der neue Dateiname ergibt sich aus dem Aufnahmepunkt (z.B. 2020-12-31-124500) plus einer durchlaufenden Nummer für mehrere Fotos, die in der gleichen Sekunde erstellt wurden (Formatzeichenkette %.c), plus der ursprünglichen Dateikennung (Formatzeichenkette %%e).

```
user$ exiftool -v '-Filename<CreateDate' -d '%Y-%m-%d-%H%M%S-.c.%e' .
user$ ls *.jpg
2020-10-27-161642-0.jpg 2020-10-27-194700-0.jpg ...
exit [rückgabewert]
```

Das bash-Kommando **exit** beendet ein Shell-Script bzw. die laufende Shell. Wenn kein Rückgabewert angegeben wird, gibt die Shell 0 zurück, also »OK«.

## **expand** [optionen] datei

**expand** ersetzt in der angegebenen Textdatei alle Tabulator-Zeichen durch eine entsprechende Zahl von Leerzeichen. Wenn keine Optionen angegeben werden, nimmt **expand** einen Tabulator-Abstand von acht Zeichen an. Das Ergebnis wird zur Standardausgabe geleitet.

- **-n**

verändert den Tabulator-Abstand auf *n* Zeichen.

## Beispiel

Das folgende Kommando entfernt die Tabulatorzeichen in *test.java* und speichert das Ergebnis in *test-no-tabs.java*. Die ursprüngliche Datei wurde mit einem Editor verfasst, der Tabulatorzeichen mit vier Leerzeichen gleichsetzte.

```
user$ expand -n 4 test.java > test-no-tabs.java
```

## **export** [optionen] variable [=wert]

Das bash-Kommando **export** deklariert die angegebene Shell-Variable als Umgebungsvariable. Damit ist die Variable auch in allen aufgerufenen Kommandos und Subshells verfügbar. Optional kann dabei auch eine Variablenzuweisung erfolgen. Wenn das

Kommando ohne Parameter aufgerufen wird, werden alle Umgebungsvariablen angezeigt.

- -n

macht eine Umgebungsvariable wieder zu einer normalen Shell-Variablen. Das Kommando hat damit genau die umgekehrte Wirkung wie bei der Verwendung ohne Optionen.

### **exportfs [optionen]**

Das Kommando `exportfs -a` meldet Änderungen in der NFS-Konfigurationsdatei `/etc(exports` an den laufenden NFS-Server. Das Kommando muss also ausgeführt werden, damit in `/etc/exportfs` durchgeführte Änderungen wirksam werden.

**expr ausdruck**

**expr Zeichenkette : Muster**

`expr` wertet den angegebenen Ausdruck arithmetisch aus oder führt einen Mustervergleich für Zeichenketten aus. Zwischen den angegebenen Variablen, Zahlen und Operatoren müssen jeweils Leerzeichen stehen! Eine kurze Beschreibung aller zulässigen Operatoren finden Sie in den [man](#)-Seiten zu `expr`. Beachten Sie, dass viele Operatorzeichen in der Shell durch das Zeichen \ vor der unmittelbaren Auswertung geschützt werden müssen.

Wenn Sie die `bash` als Shell verwenden, können Sie darin auch ohne `expr` rechnen: Arithmetische Ausdrücke können dort in der Form `$[ausdruck]` angegeben werden. Im Muster können die bei [grep](#) beschriebenen Jokerzeichen verwendet werden. Das Muster muss normalerweise in Apostrophe gestellt werden, um eine Auswertung der Sonderzeichen durch die Shell zu vermeiden.

### **Beispiele**

Das erste `expr`-Kommando führt eine einfache Berechnung durch. Das zweite Kommando liefert die Anzahl der Zeichen, die dem Musterausdruck maximal entsprechen. Das dritte Kommando extrahiert den geklammerten Teil des Musters aus einer Zeichenkette.

```
user$ expr 3 + 7 \* 19
136
user$ expr abcdefghi : 'a.*g'
7
user$ expr abc_efg_hij : '.*_\(\.*\)_.*'
efg
```

## F

**fail2ban-client** [optionen] kommando

Fail2ban ist ein Sicherheitsprogramm, das Logins überwacht und nach wiederholten fehlerhaften Logins die betreffende IP-Adresse für einige Zeit sperrt. Sofern Fail2ban eingerichtet wurde, kann seine Funktion mit **fail2ban-client** überprüft und verändert werden.

- **get parameter**  
liest einen Konfigurationsparameter aus.
- **reload [jailname]**  
liest die gesamte Konfiguration bzw. die Konfiguration eines Jails neu ein. Als *Jail* gilt in Fail2ban ein zu überwachendes Programm.
- **set [jailname] parameter wert**  
verändert einen allgemeinen Konfigurationsparameter bzw. einen Parameter eines Jails.
- **start/stop [jailname]**  
startet oder stoppt Fail2ban komplett bzw. nur ein Jail.
- **status [jailname]**  
zeigt die Liste aller aktiven Jails bzw. Details zu einem Jail an.

Wenn **fail2ban-client** ohne Kommandos nur mit der Option **-d** ausgeführt wird, gibt der Befehl die gesamte aktuell gültige Konfiguration aus, wobei nur die Filter, Jails und Actions berücksichtigt werden, die tatsächlich im Einsatz sind. Aufgrund der unübersichtlichen Konfigurationsdateien kann das ein wichtiges Hilfsmittel bei der Fehlersuche sein.

### Beispiele

Die folgenden Kommandos zeigen, dass auf dem Testrechner vier Jails aktiv sind. Das Jail für den SSH-Server hat in der Vergangenheit bereits 1418 IP-Adressen vorübergehend blockiert. Aktuell sind drei Adressen blockiert.

```
root# fail2ban-client status
Status
Number of jail: 4
```

```
Jail list:      dovecot, postfix, sshd, sshd-ddos
root# fail2ban-client status sshd
Status for the jail: sshd
  Filter
    Currently failed: 4
    Total failed:    151362
    Journal matches: _SYSTEMD_UNIT=sshd.service + _COMM=sshd
  Actions
    Currently banned: 3
    Total banned:     1418
    Banned IP list:   58.218.nnn.nnn 58.242.nnn.nnn 180.76.nnn.nnn
```

Eine irrtümlich blockierte IP-Adresse können Sie mit `fail2ban-client set` wieder freigeben (hier für das SSH-Jail):

```
root# fail2ban-client set sshd unbanip 1.2.3.4
```

**fc-list [muster]**

`fc-list` listet unter X alle skalierbaren Schriftarten auf, die dem (optionalen) Suchmuster entsprechen. Das Ergebnis kann mit `| sort` sortiert werden.

**fdisk [optionen] [device]**

`fdisk` ist ein veraltetes Programm zur Partitionierung von Festplatten.

Es kann nur Datenträger mit MBR-Partitionstabellen bearbeiten! Freunden Sie sich daher besser mit [parted](#) an, das sowohl zum MBR-Verfahren (*Master Boot Record*) als auch zu den moderneren GPTs (*GUID Partition Tables*) kompatibel ist.

**ffmpeg [inopts] [-i infile] [outopts] outfile**

Das Kommando `ffmpeg` aus dem gleichnamigen Paket konvertiert Videodateien von einem Format in ein anderes. Bei der Angabe von Optionen ist zu beachten, dass diese nur für die als Nächstes angegebene Videodatei gelten. Die Reihenfolge der Optionen ist daher entscheidend für die korrekte Funktion des Kommandos. Sofern Sie keine abweichenden Einstellungen vornehmen, verwendet `ffmpeg` für die Ergebnisdatei dieselben Codecs und Einstellungen wie in der Quelldatei.

Neben den hier aufgezählten Optionen gibt es unzählige weitere. Die optimale Einstellung der Optionen ist eine Wissenschaft für

sich. Wenn Sie sich damit nicht auseinandersetzen möchten, ist es empfehlenswert, zur Video-Recodierung eine grafische Benutzeroberfläche einzusetzen, z.B. HandBrake.

- **-acodec/-vcodec *type***  
legt den gewünschten Audio- bzw. Video-Codec fest. Eine Liste der verfügbaren Codecs liefert `ffmpeg -codecs`.
- **-b *n***  
gibt die gewünschte Bitrate in Bits pro Sekunde an (standardmäßig 200 kBit/s).
- **-f *format***  
verwendet das angegebene Format.
- **-formats**  
listet die unterstützten Formate, Codecs, Filter etc. auf.
- **-r *n***  
gibt die gewünschte Frame-Rate an (standardmäßig 25 Frames pro Sekunde).
- **-s *size***  
gibt die gewünschte Frame-Größe in der Form *wxh* an (z.B. 800x600). Alternativ können Sie eine vordefinierte Zeichenkette angeben (z.B. vga oder hd720, siehe `man ffmpeg`).
- **-ss *position***  
startet die Konvertierung zum angegebenen Zeitpunkt. Die Zeitangabe erfolgt wahlweise in Sekunden oder in der Form *hh:mm:ss[.xxx]*.
- **-t *duration***  
konvertiert nur die angegebene Zeitspanne (und nicht die gesamte Datei). Die Angabe der Zeitspanne erfolgt wie bei der Option `-ss`.
- **-target *type***  
gibt das gewünschte Zieldateiformat an (z.B. vcd, dvd etc.).
- **-vpre *quality***  
wählt eine Datei mit vordefinierten Optionen (Presets) für den

angegebenen Codec aus (z.B. default, medium, max oder fast). Die zur Auswahl stehenden Preset-Dateien befinden sich üblicherweise im Verzeichnis `/usr/share/ffmpeg`.

- `-y`  
überschreibt bereits vorhandene Dateien.

## Beispiel

Das folgende Kommando erstellt eine Filmdatei in DVD-Auflösung:

```
user$ ffmpeg -i in.avi -target pal-dvd -y out.avi
```

Das zweite Beispiel wandelt eine OGV-Filmdatei in eine YouTube-taugliche MPEG-Datei um:

```
user$ ffmpeg -i in.ogv -vcodec libx264 -vpre medium -acodec copy -y out.mpg
```

### **fg [prozess]**

Das bash-Kommando `fg` setzt einen mit `Strg+Z` unterbrochenen Prozess im Vordergrund fort. Wenn kein Prozess angegeben wird, gilt `fg` für den zuletzt unterbrochenen bzw. für den zuletzt im Hintergrund gestarteten Prozess. Andernfalls muss der Prozess durch seinen Namen oder durch die bash-interne Jobnummer (nicht PID) angegeben werden.

In der bash kann anstelle von `fg process` auch die Kurzschreibweise `%prozess` verwendet werden.

### **fgconsole [--next-available]**

`fgconsole` gibt die Nummer der gerade aktiven Konsole aus. Mit der Option `--next-available` ermittelt das Kommando die Nummer der ersten freien (ungenutzten) Konsole. Zum Wechsel der aktiven Konsole können Sie das Kommando [chvt](#) verwenden.

### **file [optionen] datei**

`file` versucht festzustellen, welchen Dateityp die angegebene Datei hat. Als Ergebnis liefert `file` eine Zeichenkette mit dem Dateinamen und dem Typ der Datei. `file` wertet nicht den Dateinamen bzw. dessen Kennung aus, sondern den Inhalt der Datei!

- `-z`  
versucht, den Datentyp einer komprimierten Datei zu erkennen.

## Beispiel

`name.jpg` enthält, wie der Dateiname erwarten lässt, eine JPEG-Bitmap-Datei:

```
user$ file name.jpg  
name.jpg: JPEG image data, JFIF standard 1.02  
find [pfadangabe] [suchoptionen]
```

`find` hilft bei der Suche nach Dateien. Dabei können verschiedene Suchkriterien (Muster für den Dateinamen, Dateigröße, Datum der Erstellung oder des letzten Zugriffs etc.) bei der Suche berücksichtigt werden. Es besteht sogar die Möglichkeit, auf alle Dateien, die diese Bedingungen erfüllen, mit einem anderen Programm (etwa [grep](#)) weitere Selektionskriterien anzuwenden. Auf diese Weise könnten beispielsweise alle `*.tex`-Dateien gefunden werden, die in den letzten drei Tagen bearbeitet wurden und die den Text »Grafikprogrammierung« enthalten. Aufgrund der großen Anzahl möglicher Suchkriterien liefert `man find` eine viele Seiten lange Beschreibung des Kommandos.

Die folgende Syntaxbeschreibung bezieht sich explizit auf die unter Linux gebräuchliche GNU-`find`-Implementierung. Sie zeichnet sich durch einige kleinere Syntaxunterschiede gegenüber den `find`-Varianten anderer Unix-Systeme aus: Dort *muss* eine Pfadangabe erfolgen, während GNU-`find` das aktuelle Verzeichnis als Startverzeichnis verwendet. Außerdem verlangen die meisten `find`-Implementierungen den Einsatz der Option `-print` zur Anzeige der Suchergebnisse, was bei GNU-`find` nicht erforderlich ist.

## Allgemeine Optionen

Im Gegensatz zu den meisten anderen Kommandos durchsucht `find` automatisch alle Unterverzeichnisse. Wenn das nicht erwünscht ist, muss die Anzahl der Unterverzeichnisse durch `-maxdepth` eingeschränkt werden.

- `-depth`

bearbeitet zuerst das aktuelle Verzeichnis und erst dann die

Unterverzeichnisse. Je nachdem, wo Sie die gesuchte Datei vermuten, führt diese Vorgehensweise erheblich schneller zu ersten Ergebnissen.

- `-follow`  
bearbeitet auch Verzeichnisse, die durch symbolische Links erfasst werden.
- `-maxdepth n`  
schränkt die Suche auf  $n$  Verzeichnisebenen ein. Mit `-maxdepth 1` werden überhaupt keine Unterverzeichnisse berücksichtigt.

## Suchkriterien

Es können mehrere Suchkriterien gleichzeitig genannt werden. Diese Kriterien werden logisch mit UND verknüpft. Die Suche wird abgebrochen, sobald das erste Kriterium nicht erfüllt ist – die Reihenfolge der Kriterien kann also Einfluss auf die Geschwindigkeit des Kommandos haben. Kriterien können mit `\(` und `\)` gruppiert, mit `!` negiert und mit `-o` logisch ODER-verknüpft werden (siehe die Beispiele).

- `-anewer referenzdatei`  
funktioniert wie `-newer` (siehe dort), aber berücksichtigt den Zeitpunkt des letzten Lesezugriffs.
- `-atime n`  
berücksichtigt den Zeitpunkt des letzten Lesezugriffs (*access*). Die Angabe erfolgt in Tagen, siehe `-mtime`.
- `-cnewer referenzdatei`  
funktioniert wie `-newer` (siehe dort), aber berücksichtigt den Zeitpunkt der letzten Veränderung von Metadaten.
- `-ctime n`  
wertet den Zeitpunkt aus, zu dem zuletzt die Metadaten der Datei (Zugriffsrechte etc.) verändert wurden.
- `-executable`  
findet Dateien, die ausführbar sind, bei denen also das Execute-Bit gesetzt ist.

- `-group gruppenname` oder `-nogroup gruppenname`  
findet Dateien, die der angegebenen Gruppe angehören (bzw. die ihr nicht angehören).
- `-mmin n`  
funktioniert wie `mtime`, allerdings erfolgt die Zeitangabe in Minuten.
- `-mtime n`  
findet Dateien, deren Inhalt zuletzt vor genau *n* Tagen verändert wurde (*modify*). Wenn vor der Zahl ein + angegeben wird, dann werden alle Dateien erfasst, die älter als *n* Tage sind. Ein vorangestelltes - liefert Dateien, die jünger als *n* Tage sind. `-mtime 0` liefert Dateien, die in den letzten 24 Stunden verändert wurden.
- `-name suchmuster`  
findet Dateien, die dem angegebenen Suchmuster entsprechen. Wenn das Suchmuster Jokerzeichen enthält, muss es in Apostrophe gestellt werden. Wenn `find` nicht zwischen Groß- und Kleinschreibung unterscheiden soll, verwenden Sie statt `-name` die Option `-iname`.
- `-newer referenzdatei`  
findet Dateien, die neuer sind als die Referenzdatei, die also nach der Referenzdatei erzeugt oder verändert wurden. Sie können beispielsweise beim Beginn eines Backups mit `touch` eine timestamp-Datei erzeugen. Bei einem inkrementellen weiteren Backup zu einem späteren Zeitpunkt berücksichtigen Sie dann nur die Dateien, für die `-newer timestamp` gilt.
- `-path suchmuster`  
findet Dateien, die dem angegebenen Suchmuster entsprechen. Die Option geht über `-name` hinaus, weil das Suchmuster jetzt nicht nur den Dateinamen, sondern auch den Pfad dorthin betrifft. Diese Option ist flexibler als die direkte Pfadangabe im

ersten Argument von `find`, weil hier die Jokerzeichen auch das Zeichen / erfassen.

- `-perm zugriffsbits`

findet Dateien, deren Zugriffsbits genau dem angegebenen Oktalwert (siehe [chmod](#)) entsprechen. Wenn dem Oktalwert ein - vorangestellt wird, darf die Datei auch darüber hinausgehende Zugriffsrechte haben. Wenn ein / vorangestellt wird, reicht es aus, wenn zumindest eines der gesuchten Zugriffsbits gesetzt ist.

- `-size dateigröße`

gibt die gewünschte Dateigröße vor. Die Angabe erfolgt standardmäßig in Vielfachen von 512. `-size 3` bezeichnet also Dateien zwischen 1024 und 1536 Bytes. Durch die zusätzlichen Zeichen c oder k kann die Größe in Byte oder KiB angegeben werden. Ein vorangestelltes + erfasst alle größeren Dateien, ein vorangestelltes - alle kleineren Dateien. `-size +10k` liefert daher alle Dateien, die größer als 10 KiB sind.

- `-type zeichen`

schränkt die Suche auf bestimmte Dateitypen ein. Die wichtigsten Zeichen sind f für reguläre Dateien, d für Verzeichnisse (*directories*) und l für symbolische Links.

- `-user username oder -nouser username`

findet Dateien, die dem angegebenen Benutzer gehören (bzw. die ihm nicht gehören).

## Aktionen beim Finden einer Datei

- `[-exec kommando [optionen] {} \;]`

ruft das angegebene Kommando auf und übergibt den Dateinamen der gefundenen Datei, die alle bisher verarbeiteten Kriterien erfüllt hat. Das Kommando kann nun einen Test durchführen, ob die Datei weiteren Kriterien entspricht. Ein typisches Programm, das durch `-exec` aufgerufen wird, ist [grep](#). Alternativ können Sie hier aber auch Kommandos aufrufen, die

die gefundenen Dateien verschieben, kopieren oder anderweitig verarbeiten.

{} steht dabei als Platzhalter für den Dateinamen. ; schließt den Kommandoaufruf ab, d.h., dahinter können weitere `find`-Optionen angegeben werden. \ ist innerhalb der Shell erforderlich, um die Interpretation von ; als Sonderzeichen zu verhindern.

- `[-exec kommando [optionen] {} +]`  
wie oben, allerdings wird das angegebene Kommando nicht für jede gefundene Datei aufgerufen. Vielmehr versucht `find`, möglichst viele Ergebnisse auf einmal an das Kommando zu übergeben. Das ist deutlich effizienter. Die Anzahl der Dateien ist nur durch die maximale Länge eines Kommandos limitiert. Diese beträgt in aktuellen Linux-Distributionen ca. 2 Millionen Zeichen.
- `-ls`  
liefert detaillierte Angaben zu jeder gefundenen Datei (so ähnlich wie [ls -l](#)).
- `-print`  
gibt die gefundenen Dateinamen auf dem Bildschirm aus. Diese Option ist die Standardeinstellung, sofern nicht `-exec` verwendet wird.
- `-print0`  
trennt die Dateinamen der Ergebnisliste durch 0-Bytes. Das ermöglicht eine Weiterverarbeitung von Dateinamen mit Leerzeichen durch [xargs](#).
- `-printf format`  
gibt die gefundenen Dateinamen und andere Informationen auf dem Bildschirm aus. In der Formatzeichenkette kann angegeben werden, in welcher Form die Ausgabe erfolgt und welche Zusatzinformationen mit ausgegeben werden (etwa die Dateigröße, das Datum der letzten Änderung etc.). Die Syntax für die Formatzeichenkette ist in den [man](#)-Seiten beschrieben.

## Beispiele

Das erste `find`-Kommando liefert alle Dateien im aktuellen Verzeichnis (auch versteckte Dateien, aber keine Dateien aus Unterverzeichnissen); das zweite Kommando liefert nur gewöhnliche Dateien (aber keine unsichtbaren Dateien):

```
user$ find -maxdepth 1 -type f -name '*'  
user$ find -maxdepth 1 -type f -name '[!.]*'
```

Das folgende Kommando durchsucht das `/etc`-Verzeichnis nach Dateien, die in den letzten zwei Wochen verändert wurden. Das Kommando muss mit root-Rechten ausgeführt werden, weil einige Dateien in `/etc` nicht von gewöhnlichen Benutzern gelesen werden dürfen:

```
root# sudo find /etc -mtime -14
```

Das folgende Kommando löscht alle Backup-Dateien, die durch das Zeichen `~` am Ende des Dateinamens gekennzeichnet sind. `find` berücksichtigt dabei auch alle Unterverzeichnisse. Die Liste der zu löschen Dateien wird durch die Kommandosubstitution mit `$(kommando)` an `rm` weitergeleitet.

```
root# rm $(find . -name '*~')
```

Falls es sich um sehr viele Dateien handelt, tritt bei der Ausführung des obigen Kommandos ein Fehler auf: Die Kommandozeile mit allen `*~`-Dateien wird so lang, dass sie die maximale Kommandozeilenlänge überschreitet. In solchen Fällen müssen Sie entweder die `-exec`-Option des `find`-Kommandos oder das Kommando `xargs` zu Hilfe nehmen. Die `xargs`-Variante hat den Vorteil, dass sie auch mit Dateinamen zurechtkommt, die Leerzeichen enthalten.

```
user$ find -name '*~.jpg' -exec rm {} \;
```

Im nächsten Beispiel sollen in einem Verzeichnis alle Dateien für jeden lesbar gemacht werden (siehe [chmod](#)). Etwas schwieriger ist es, mit `find` für alle Unterverzeichnisse (aber nicht für gewöhnliche Dateien) das Execute-Bit zu setzen, sodass alle Personen auf dem Rechner mit `cd` in die Verzeichnisse wechseln dürfen. Das Zeichen `+` am Ende des Kommandos bewirkt, dass [chmod](#) nicht für jedes

Verzeichnis einzeln aufgerufen wird, sondern dass möglichst viele Verzeichnisse auf einmal an [chmod](#) übergeben werden.

```
root# cd /verzeichnis/fuer/alle  
root# chmod -R a+r *  
root# find . -type d -exec chmod a+x {} +
```

Das letzte Beispiel erstellt zuerst mit [touch](#) eine Referenzdatei mit dem Datum 1.1.2020 sowie das Verzeichnis *../neue-bilder*. Dorthin werden nun alle Bilder aus dem lokalen Verzeichnis kopiert, die ab dem 1.1.2020 entstanden sind. *neue-bilder* darf nicht im aktuellen Verzeichnis erstellt werden, sonst erfasst `find` auch die dorthin kopierten Dateien. Beachten Sie auch die Oder-Verknüpfung für die Dateikennungen *\*.jpg* und *\*.png*.

```
user$ touch 2020 -t 202001010000  
user$ mkdir ../neue-bilder  
user$ find . -newer 2020 \(` -name \*.png -o -name \*.jpg `) \  
      -exec cp {} ../neue-bilder +;
```

Weitere `find`-Beispiele finden Sie bei der Beschreibung der Kommandos [grep](#) und [xargs](#).

**findmnt** [optionen] [device/mountpoint]

`findmnt` ohne Parameter liefert eine baumförmige Liste aller im Verzeichnisbaum integrierten Dateisysteme. Das Ergebnis enthält auch diverse */dev-*, */proc-*, */sys-* und */run*-Dateisysteme zum internen Informationsaustausch.

Durch die Angabe eines Devices oder [mount](#)-Punktes reduziert sich die Liste auf alle so ausgewählten Dateisysteme. Weitere zulässige Angaben sind die UUID oder der Name (Label) eines Dateisystems bzw. die Major- und Minor-Nummer des Devices in der Schreibweise `major:minor`.

- `-A`  
zeigt alle Dateisysteme an.
- `-l`  
zeigt das Ergebnis als Tabelle an (nicht als Baum).
- `--mtab bzw. --fstab`  
berücksichtigt nur die Einträge von */etc/mtab* bzw. */etc/fstab*.

Standardmäßig berücksichtigt `findmnt` die Mount-Tabelle des Kernels in der Pseudo-Datei `/proc/self/mountinfo`.

- `-o spalten` bzw. `--output spalten`  
gibt nur die in `spalten` angegebenen Informationen spaltenweise aus. Zulässige Spaltencodes sind unter anderem:  
`SOURCE`: Device  
`TARGET`: [mount](#)-Pfad  
`FSTYPE`: Dateisystemtyp  
`OPTIONS`: alle [mount](#)-Optionen  
`SIZE`: Größe des Dateisystems  
Weitere Codes liefert `findmnt --help`. Die Spaltencodes werden durch Kommas getrennt, beispielsweise `-o SOURCE,TARGET,FSTYPE`. Leerzeichen sind nicht zulässig!
- `-t typliste` bzw. `--types typliste`  
liefert nur Dateisysteme des angegebenen Typs. Mit `-t ext2,ext3,ext4` werden somit nur ext-Dateisysteme in den Versionen 2, 3 und 4 berücksichtigt.

## Beispiel

`findmnt` zeigt alle aktiven Dateisysteme des Testrechners:

```
root# findmnt
TARGET                SOURCE      FSTYPE      OPTIONS
/                     /dev/sda1   ext4        rw,relatime,errors=...
 /sys                  sysfs       sysfs       rw,nosuid,nodev,noexec,...
 /sys/kernel/security securityfs securityfs  rw,nosuid,nodev,noexec,...
 /sys/fs/cgroup         tmpfs       tmpfs       ro,nosuid,nodev,noexec,...
```

**firewall-cmd optionen**

`firewall-cmd` liest bzw. verändert die Firewall-Konfiguration unter CentOS, Fedora, RHEL sowie bei aktuellen SUSE-Distributionen. `firewall-cmd` kommuniziert mit dem im Hintergrund laufenden Programm `firewalld`. Dieses Programm ordnet jeder Netzwerkschnittstelle eine Zone zu. Eine Zone im Sinne von `firewalld` ist eine Sammlung von Regeln für einen bestimmten

Anwendungszweck. Die folgende Aufzählung beschreibt die wichtigsten Zonen:

- **block**  
blockiert jeden Netzwerkverkehr; der Absender erhält eine ICMP-Fehlermeldung.
- **drop**  
blockiert jeden Netzwerkverkehr; der Absender wird nicht informiert.
- **external**  
blockiert die meisten Ports und aktiviert das Masquerading (IPv4). Bei einem Router ist diese Zone für die Schnittstelle vorgesehen, die die Verbindung zum Internet herstellt.
- **Fedoraworkstation und FedoraServer**  
kommen standardmäßig für Netzwerkschnittstellen von Fedora-Installationen zum Einsatz. **FedoraServer** blockiert alle Dienste außer SSH und DHCP-Client. **Fedoraworkstation** ist deutlich liberaler und akzeptiert auch Samba-Client-Verbindungen sowie den Verkehr über alle Ports zwischen 1025 und 65535.
- **home und internal**  
blockiert die meisten Ports, akzeptiert aber Samba (nur als Client), CUPS und Zeroconf/Avahi/mdns. Beide Zonen sind für Rechner in einem als einigermaßen sicher geltenden lokalen Netzwerk gedacht. Wenn Sie diese Zone nutzen und selbst Windows-Netzwerkverzeichnisse freigeben möchten, müssen Sie außerdem den Dienst **SAMBA** freischalten.
- **public**  
ähnlich wie **home**, blockiert aber auch CUPS und Samba-Client-Funktionen. Die Zone ist für die Internet-Nutzung in unsicheren Netzwerken gedacht, z.B. in einem öffentlichen WLAN.
- **trusted**  
erlaubt jeden Netzwerkverkehr. Diese Zone ist für gut gesicherte

lokale Netzwerke gedacht, aber nicht für WLAN-Verbindungen. `firewall-cmd` führt die Änderungen normalerweise nur dynamisch durch. Wenn Änderungen dauerhaft gelten sollen, müssen Sie die Option `--permanent` angeben. Die Regeln werden dann im Verzeichnis `/etc/firewalld` gespeichert.

## Optionen

- `--add-interface=name` bzw. `--remove-interface=name` ordnet die angegebene Schnittstelle der Default-Zone oder der durch die Option `--zone` angegebenen Zone zu bzw. löst diese Zuordnung wieder. Bevor eine Schnittstelle einer neuen Zone zugeordnet werden kann, muss die bisherige Zuordnung explizit entfernt werden!
- `--add-prot=name` bzw. `--remove-port=name` erlaubt die Nutzung eines TCP-Ports in der mit `--zone` angegebenen Zone bzw. blockiert den Port.
- `--add-service=name` bzw. `--remove-service=name` erlaubt die Nutzung eines Netzwerkdiensts (z.B. `https` oder `ssh`) in der mit `--zone` angegebenen Zone bzw. blockiert den Dienst.
- `--change-interface=iface` ordnet die aktive Zone der angegebenen Netzwerkschnittstelle zu. Um eine andere Zone mit einer Schnittstelle zu verbinden, übergeben Sie zusätzlich die Option `--zone=name`.
- `--get-active-zones` listet alle Zonen auf, die momentan in Verwendung sind, und gibt an, welche Netzwerkschnittstellen den Zonen zugeordnet sind.
- `--get-default-zone` ermittelt die Default-Zone.
- `--get-services` listet alle Netzwerkdienste auf, die `firewall-cmd` kennt. Die resultierenden Namen (`http`, `ssh` usw.) können bei `--add-interface` angegeben werden.

- `--get-zones`  
listet alle Zonen auf, die `firewall-cmd` kennt.
- `--get-zone-of-interface=name`  
ermittelt die Zone, die der angegebenen Netzwerkschnittstelle zugeordnet ist.
- `--list-all-zones`  
liefert die Eckdaten aller definierten Firewall-Zonen und gibt an, welche Netzwerkschnittstellen den Zonen zugeordnet sind.
- `--list-services`  
listet alle Netzwerkdienste auf, die in der gerade aktiven Firewall-Zone erlaubt sind (also nicht blockiert werden).
- `--permanent`  
speichert Änderungen an der Firewall-Konfiguration bleibend.  
Die Änderungen werden aber nicht direkt angewendet. Wenn Sie eine Änderung sofort *und* bleibend durchführen möchten, müssen Sie das betreffende Kommando *zweimal* ausführen:  
einmal mit `--permanent` und einmal ohne diese Option.
- `--set-default-zone=name`  
definiert die Default-Zone. Diese gilt für alle Netzwerkschnittstellen, denen nicht explizit eine andere Zone zugewiesen wird.
- `--state`  
ermittelt, ob `firewalld` läuft oder nicht.
- `--zone=name`  
gibt an, welche Zone verändert, bearbeitet oder gelesen werden soll.

## Beispiele

Die folgenden Kommandos ermitteln zuerst, welcher Zone die Netzwerkschnittstelle `enp0s3` zugeordnet ist, und ordnen die Schnittstelle dann der Zone `trusted` zu. Damit können sämtliche Dienste über diese Schnittstelle genutzt werden. Die Änderungen

sind nicht permanent, gelten also nur bis zum nächsten Neustart des Rechners bzw. des Firewall-Dämons.

```
root# firewall-cmd --get-zone-of-interface=enp0s3
public
root# firewall-cmd --zone=public --remove-interface=enp0s3
success
root# firewall-cmd --zone=trusted --add-interface=enp0s3
success
```

Beim zweiten Beispiel dient das erste Kommando dazu, herauszufinden, welcher Zone die Netzwerkschnittstelle enp0s5 zugeordnet ist. Über diese Schnittstelle soll der Rechner einen Webserver nach außen anbieten. Aber anstatt die Schnittstelle generell der Zone `trusted` zuzuordnen und so auch alle anderen erdenklichen Dienste freizuschalten, werden jetzt die Regeln für die Zone geändert (hier `FedoraWorkstation`), und das dauerhaft: Diese Zone erlaubt jetzt die Protokolle HTTP und HTTPS. Das `reload`-Kommando aktiviert die Änderungen an der Zone.

```
root# firewall-cmd --get-zone-of-interface=enp0s5 (aktive Zone herausfinden)
FedoraWorkstation
root# firewall-cmd --permanent --zone=FedoraWorkstation --add-service=http
root# firewall-cmd --permanent --zone=FedoraWorkstation --add-service=https
root# firewall-cmd --reload
```

### **flatpak [optionen] kommando**

flatpak hilft beim Installieren, Entfernen und Administrieren von Flatpak-Paketen. Dabei handelt es sich um ein neuartiges Paketformat, das die Installation von Desktop-Anwendungen losgelöst von den Paketquellen einer Distribution vereinfachen soll. Das größte offizielle Flatpak-Repository ist <https://flathub.org>. Im Vergleich zu herkömmlichen Paketen gibt es zusätzliche Sicherheitsmechanismen, z.B. Sandboxing. Allerdings sind Flatpak-Pakete oft riesig, weil sie alle erforderlichen Bibliotheken inkludieren und nicht auf eventuell schon vorhandene traditionelle Pakete Rücksicht nehmen. Die Installation mehrerer Flatpaks ist deswegen oft mit großen Redundanzen verbunden. Flatpak-Pakete werden üblicherweise direkt im Desktop installiert, z.B. im Gnome-Programm *Software* oder im Browser. Der

Programmstart erfolgt im Startmenü bzw. in der Gnome-Ansicht AKTIVITÄTEN. Das Kommando `flatpak` ist insofern nur selten erforderlich.

Flatpak steht aktuell (2020) nur unter Fedora, CentOS und RHEL standardmäßig zur Verfügung. Das Kommando kann aber unkompliziert in anderen Distributionen installiert und genutzt werden und soll eine Plattform zur distributionsunabhängigen Weitergabe von Linux-Software werden. Eine Alternative zu Flatpaks sind Snap-Pakete, die von Ubuntu favorisiert werden (siehe [snap](#)). Das Kommando `flatpak` erfordert zwar keine root-Rechte, allerdings müssen Sie sich während der Ausführung immer wieder mit Ihrem eigenen Passwort authentifizieren (mitunter ein Dutzend Mal für eine einzige Operation, z.B. die Installation eines Pakets).

- `info paketname`  
zeigt Details zu einem bereits installierten Paket an. Beachten Sie, dass der Paketname immer aus mehreren Teilen zusammengesetzt ist, z.B. `org.gnome.gedit2`.
- `install paketquelle paketname`  
installiert das gewünschte Paket aus einer zuvor eingerichteten Paketquelle. Häufig werden die Quelle und der Paketname gemeinsam mit der Option `--from` als URL übergeben, also z.B.:  
`--from https://flathub.org/repo/appstream/io.atom.Atom.flatpakref`
- `list [--system bzw. --user]`  
listet alle installierten Flatpaks auf (gegebenenfalls nur solche, die auf Systemebene bzw. für den aktuellen Benutzer installiert wurden). Mit der zusätzlichen Option `-d` zeigt `flatpak` auch die Paketgröße sowie einige weitere Details an.
- `remote-add [--if-not-exists] paketquelle url`  
richtet eine Paketquelle ein. Die Option `--if-not-exists` vermeidet Fehlermeldungen, wenn die Paketquelle bereits früher definiert wurde.

- `search suchbegriff`  
durchsucht die eingerichteten Paketquellen nach Paketen, die den Suchbegriff im Paketnamen oder in der Beschreibung enthalten.
- `uninstall paketname`  
entfernt das angegebene Paket ohne Rückfrage. Sie müssen dabei unter Umständen den vollständigen Paketnamen angeben, wie er von `flatpak list` angezeigt wird, z.B. `flatpak uninstall org.gimp.GIMP/x86_64/stable`.
- `update [--app paketname]`  
aktualisiert alle installierten Flatpaks bzw. nur das mit `--app` angegebene Paket.

## Beispiele

Die beiden folgenden Kommandos richten die Paketquelle für <https://flathub.org> ein und installieren dann den Editor Atom auf Systemebene, sodass alle Benutzer des Rechners ihn verwenden können. Beachten Sie die irrwitzigen Download-Größen für die Plattformpakete. Es ist nicht lange her, da waren die Downloads für eine ganze Linux-Distributionen kleiner!

```
root# flatpak remote-add --if-not-exists flathub \
    https://dl.flathub.org/repo/flathub.flatpakrepo
root# flatpak install flathub io.atom.Atom
Required runtime for io.atom.Atom/x86_64/stable (runtime/org.freedesktop.Sdk/
x86_64/19.08) found in remote flathub
Do you want to install it? [Y/n]: y
...
      ID                  Branch   Download
1. org.freedesktop.Platform.GL.default 19.08     89.1 MB
2. org.freedesktop.Platform.openh264    2.0       1.5 MB
3. org.freedesktop.Sdk.Locale         19.08     322.4 MB
4. org.freedesktop.Sdk                19.08     578.9 MB
5. io.atom.Atom                      stable    184.1 MB
```

Mit `flatpak search` suchen Sie in allen Flatpak-Quellen nach einem Paket:

```
user$ flatpak search spotify
Name      Description          Application ID      Version      Branch  Remotes
Spotify   Online music strea... com.spotify.Client  1.1.26.501  stable  flathub
fold [optionen] datei
```

`fold` umbreicht Textzeilen bei einer Länge von 80 Zeichen und zeigt das Ergebnis auf dem Bildschirm an.

- `-s` bzw. `--spaces`  
versucht, den Umbruch an der Stelle eines Leerzeichens (also zwischen zwei Wörtern) durchzuführen. Die Zeilen werden damit kürzer oder gleich  $n$  Zeichen.
- `-w n` bzw. `--width n`  
stellt eine maximale Textbreite von  $n$  Zeichen ein. Ohne diese Option gelten standardmäßig 80 Zeichen.

```
for var [in liste;] do  
    kommandos  
done
```

`for` bildet Schleifen in bash-Skripts. In die angegebene Variable werden der Reihe nach alle Listenelemente eingesetzt. Die Liste kann auch mit Jokerzeichen für Dateinamen oder mit `{...}`-Elementen zum Zusammensetzen von Dateinamen gebildet werden. Wenn auf die Angabe der Liste verzichtet wird, durchläuft die Variable alle Parameter, die der Shell-Datei übergeben wurden (also `in $*`).

### Beispiel

`for` kann nicht nur in Shell-Skripts verwendet werden, sondern auch im Terminal, um einige Kommandos auf eine Reihe von Dateien anzuwenden. Das folgende Kommando gibt die Dateinamen aller `*.jpg`-Dateien aus. Sie können nun [echo](#) durch ein anderes Kommando ersetzen, das die `*.jpg`-Dateien druckt, skaliert etc.

```
user$ for i in *.jpg; do echo $i; done
```

Ein weiteres `for`-Beispiel finden Sie bei der Beschreibung des [cp](#)-Kommandos.

```
free [optionen]
```

`free` zeigt an, wie der verfügbare Speicherplatz (RAM und Swap-Speicher) genutzt ist.

- **-h**  
wählt selbst geeignete Einheiten für die Angabe des Speichers (*human readable*).
- **-m / -g**  
zeigt den Speicherbedarf in MiB oder GiB an (statt in KiB).

## Beispiel

Im folgenden Beispiel wird zur Ausführung von `free` die Lokalisierungsvariable `LANG` zurückgesetzt. Das verhindert die langatmigen deutschen Übersetzungen der Spaltennamen, die das Spaltenlayout durcheinander bringen.

In der virtuellen Maschine mit knapp 2 GiB RAM werden 1,1 GiB genutzt. Dass dennoch nur 170 MiB als frei gelten, liegt daran, dass das Betriebssystem fast 700 MiB zur Zwischenspeicherung von Dateien verwendet. Dieser Speicher kann bei Bedarf rasch freigegeben werden, so dass für neue Prozesse noch rund 680 MiB zur Verfügung stehen.

Der Swap-Speicherplatz wird bis auf wenige MiB nicht genutzt.

```
user LANG= free -h
```

	total	used	free	shared	buff/cache	available
Mem:	1.9G	1.1G	171M	10M	686M	677M
Swap:	947M	6.5M	940M			

## fsck [optionen] device

`fsck` überprüft die Konsistenz des Dateisystems und führt gegebenenfalls Reparaturen durch. Es darf nur von `root` ausgeführt werden. Je nach Typ des Dateisystems ruft `fsck` die Programme `fsck.ext3`, `fsck.ext4` etc. auf. Lesen Sie vor der Ausführung von `fsck` unbedingt die [man](#)-Seiten dieses und des dateisystemspezifischen Kommandos!

Vor der Überprüfung eines Dateisystems müssen Sie [`umount`](#) ausführen. Zur Überprüfung des root-Dateisystems führen Sie zuerst [`touch`](#)/`forcefsck` aus und starten dann den Rechner neu.

Fehlerhafte Dateien bzw. Dateifragmente werden im Verzeichnis *lost+found* gespeichert.

- **-A**  
überprüft alle in */etc/fstab* genannten Dateisysteme.
- **-t typ**  
gibt den Typ des Dateisystems an (etwa *ext3, xfs*).

```
fsck.ext2 [optionen] device  
fsck.ext3 [optionen] device  
fsck.ext4 [optionen] device
```

*fsck.ext2/3/4* bzw. *e2fsck* überprüft die Konsistenz eines ext-Dateisystems und führt gegebenenfalls Reparaturen durch. Das Kommando kann auch via *fsck -t ext2/ext3/ext4* gestartet werden.

- **-b n**  
liest den alternativen Superblock *n*.
- **-n**  
beantwortet alle Rückfragen mit *n (no)*, führt keine Änderungen durch.
- **-p**  
führt Reparaturen (Änderungen) im Dateisystem ohne Rückfrage durch.
- **-y**  
beantwortet alle Rückfragen mit *yes*, führt Änderungen durch.

```
fstrim [optionen] mountverzeichnis
```

*fstrim* führt ein Trim-Kommando für das durch den [mount](#)-Punkt angegebene Dateisystem aus. Dabei werden standardmäßig alle nicht mehr vom Dateisystem genutzten Datenblöcke an den Datenträger gemeldet.

Das Kommando ist nur zweckmäßig, wenn es sich bei dem Datenträger um eine Solid State Disk (SSD) handelt und das Dateisystem nicht ohnedies bei jedem Löschvorgang automatisch Trim-Kommandos ausführt (siehe die [mount](#)-Option `discard` bei den

Dateisystemen ext4, [btrfs](#) und xfs). `fstrim` kann täglich oder wöchentlich durch einen Cron-Job ausgeführt werden, um Performance-Verluste durch zu häufige oder aber gar nicht ausgeführte Trim-Kommandos zu vermeiden.

- `-l` bzw. `--length n`  
meldet ungenutzte Blöcke nur für den angegebenen Bereich.
- `-m` bzw. `--minimum n`  
meldet ungenutzte zusammenhängende Blöcke nur, wenn sie zumindest n Byte groß sind. Das kann die Ausführung des Kommandos erheblich beschleunigen.
- `-o` bzw. `--offset n`  
meldet ungenutzte Blöcke erst ab dem angegebenen Offset (gerechnet vom Beginn des Dateisystems). Die Angabe kann durch ein entsprechendes Suffix in binären KiB, MiB, GiB oder TiB (K, M, G oder T) oder in dezimalen Kilobyte, Meabyte etc. angegeben werden (kB, MB, GB oder TB).
- `-v` bzw. `--verbose`  
liefert die Summe der mit dem Trim-Kommando gemeldeten Bytes.

## Beispiel

Das folgende Kommando meldet alle zumindest 64 KiB großen freien Datenblöcke an die SSD:

```
root# fstrim -m 64K /
/: 13493051392 bytes were trimmed
ftp [optionen] ftpserver
```

ftp stellt eine Verbindung zum angegebenen FTP-Server her. Nach dem Login können Sie interaktiv Dateien zwischen dem lokalen Rechner und dem FTP-Server übertragen. Im Folgenden sind die wichtigsten Kommandos für die interaktive Nutzung zusammengefasst:

- ?  
zeigt eine Liste aller FTP-Kommandos an.

- **!**  
ermöglicht die Ausführung von Shell-Kommandos.
- **ascii**  
wechselt in den Textmodus.
- **binary**  
wechselt in den Binärmodus.
- **bye oder quit**  
beendet FTP.
- **cd *verz***  
wechselt in das angegebene FTP-Verzeichnis.
- **get *datei***  
überträgt die Datei vom FTP-Archiv in das aktuelle Verzeichnis.
- **help *kommando***  
zeigt eine kurze Info zum angegebenen Kommando an.
- **lcd *verz***  
wechselt das aktuelle Verzeichnis auf dem lokalen Rechner.
- **ls**  
zeigt die Liste der Dateien auf dem FTP-Server an.
- **lls**  
zeigt die Liste der Dateien auf dem lokalen Rechner an.
- **mget *\*.muster***  
überträgt alle passenden Dateien vom FTP-Archiv in das aktuelle Verzeichnis (siehe auch **prompt**).
- **open**  
stellt die Verbindung zum fremden Rechner her (wenn es beim ersten Versuch nicht geklappt hat).
- **prompt**  
aktiviert bzw. deaktiviert die Rückfrage vor der Übertragung jeder Datei durch **mget**.
- **put *datei***  
überträgt die Datei vom aktuellen Verzeichnis in das FTP-Archiv

(*upload*).

- **reget** *datei*  
setzt die Übertragung einer bereits teilweise gesendeten Datei fort.

```
[function] name  
{kommandos}
```

Das optionale Schlüsselwort **function** definiert in bash-Scripts eine Subfunktion, die wie ein neues Kommando aufgerufen werden kann. Innerhalb der Funktion können Sie mit [local](#) lokale Variablen definieren. Funktionen können auch rekursiv aufgerufen werden. Funktionen müssen *vor* ihrem ersten Aufruf deklariert werden! An Funktionen können Parameter übergeben werden. Anders als bei vielen Programmiersprachen dürfen die Parameter nicht in Klammern gestellt werden. Innerhalb der Funktion können die Parameter den Variablen \$1, \$2 etc. entnommen werden.

## Beispiel

Das folgende Mini-Script gibt *Hello World, abc!* aus:

```
#!/bin/bash  
function myfunc {  
    echo "Hello World, $1!"  
}  
  
myfunc "abc"
```

```
fuser dateiname
```

**fuser** ermittelt das Programm, das auf die angegebene Datei zugreift und diese möglicherweise für andere Programme blockiert. **fuser** liefert normalerweise nur die Prozessnummer (PID) des Programms.

- **-k**  
beendet alle Prozesse, die auf die Datei zugreifen.
- **-m**  
listet alle Prozesse auf, die auf *irgendeine* Datei des Dateisystems zugreifen, das als Device oder durch eine Datei spezifiziert ist.

- **-v**  
gibt auch an, welcher Benutzer den Prozess ausführt (*verbose mode*).

Bei der Ausgabe folgt der Prozessnummer ein Buchstabe, der die Art des Zugriffs angibt:

- c  
Der Prozess hat das Verzeichnis als aktuelles Verzeichnis eingestellt (*current directory*).
- e  
Die Datei wird ausgeführt (*executable being run*).
- f  
Die Datei ist für den Lesezugriff geöffnet (*open file*). Dieser Buchstabe wird nur im *verbose mode* angezeigt.
- F  
Die Datei ist für den Schreibzugriff geöffnet. Auch F wird nur im *verbose mode* angezeigt.
- m  
Die Datei wird als Bibliothek oder durch Memory-Mapping genutzt.
- r  
Das Verzeichnis wird als Wurzelverzeichnis genutzt (*root directory*).

Eine Alternative zu fuser ist das Kommando [lsof](#).

## Beispiel

Der Versuch, eine DVD durch [umount](#) aus dem Dateisystem zu lösen, scheitert. Schuld ist der Prozess 32664 des Benutzers kofler, der mit [less](#) gerade eine Textdatei der DVD liest.

```
root# umount /media/kofler/myBackupDvd
umount: /media/kofler/myBackupDvd: target is busy.
root# fuser -v -m /media/kofler/myBackupDvd
                BEN.          PID ZUGR.  BEFEHL
/run/media/kofler/myBackupDvd
                    kofler    32664 f.... less
```

**fwupdmg** kommando

Der Kommandoname `fwupdmgr` steht für *Firmware Update Manager*. Das Kommando gibt Ihnen bei modernen Linux-Distributionen die Möglichkeit, Firmware-, BIOS- oder EFI-Updates zu initiieren.

`fwupdmgr` setzt voraus, dass der Hintergrunddienst `fwupdmgr` läuft.

- `get-devices`

liefert eine detaillierte Liste aller Komponenten, die mit Updates versorgt werden können, sowie eine Aufstellung, welche Updates zuletzt durchgeführt wurden und welche aktuell zur Verfügung stehen. Im Anschluss haben Sie die Möglichkeit, einen Statusbericht an die Website <https://fwupd.org> hochzuladen.

- `downgrade device`

macht ein Update rückgängig und reaktiviert eine ältere Version.

- `refresh`

aktualisiert die Update-Informationen und lädt dazu Daten von <https://fwupd.org> herunter.

- `update device`

initiiert ein Update für alle Komponenten bzw. nur für die angegebene Komponente. `fwupdmgr` bereitet das Update nur vor. In den meisten Fällen kann das Update erst beim nächsten Neustart des Rechners durchgeführt werden.

## Beispiel

Das erste Kommando aktualisiert die Update-Informationen. Das zweite Kommando zeigt einen Statusbericht für das Lenovo-Notebook mit der Typenbezeichnung 20MD0001GE.

```
root# fwupdmgr refresh
Fetching metadata https://cdn/fwupd.org/downloads/firmware.xml.gz
Fetching signature https://cdn/fwupd.org/downloads/firmware.xml.gz.asc
Successfully downloaded new metadata: 5 local devices supported
```

```
root# fwupdmgr get-devices
20MD0001GE
Thunderbolt Controller:
  Device ID:          02dcf173032b1c4f136f185c0564b059a2e3d305
  Summary:            Unmatched performance for high-speed I/O
  Current version:   39.00
  ...
PM981 NVMe Samsung 2048GB: ...
```

```
SAMSUNG MZVLB256HAHQ-000L7: ...
System Firmware: ...
UEFI Device Firmware: ...
Devices that were not updated correctly:
```

```
* ThinkPad P1 Thunderbolt Controller (23.00 -> 39.00)
```

Devices that have been updated successfully:

```
* UEFI Device Firmware (192.24.1314 -> 192.35.1427)
* 20MD0001GE System Firmware (0.1.24 -> 0.1.25)
* UEFI Device Firmware (192.35.1427 -> 192.47.1524)
```

Uploading firmware reports helps hardware vendors to quickly identify failing and successful updates on real devices. Upload report now?  
(Requires internet connection):

0. Do not upload reports at this time, but prompt again for future updates
1. Do not upload reports, and never ask to upload reports for future updates
2. Upload reports just this one time, but prompt again for future updates
3. Upload reports this time and automatically upload reports after completing future updates

Das letzte Kommando initiert das Update für den Thunderbolt-Controller:

```
root# fwupdmgr update
Upgrade available for Thunderbolt Controller from 39.00 to 50.00
Thunderbolt Controller and all connected devices may not be usable while
updating. Continue with update? [Y|n]: y
```

## G

```
gdebi [optionen] paket.deb
```

Das Kommando `gdebi` aus dem Paket `gdebi-core` ermöglicht es unter Debian, Ubuntu und verwandten Distributionen, ein lokales Paket zu installieren. Im Unterschied zu `dpkg -i` werden dabei auch abhängige Pakete aus den eingerichteten Paketquellen heruntergeladen und installiert.

- `--apt-line`  
führt die Installation nicht durch, sondern gibt nur das erforderliche [apt-get](#)-Kommando aus.
- `-n`  
installiert das Paket ohne Rückfrage.
- `-p "apt-options"`  
übergibt die angegebenen Optionen an [apt](#).
- `-q`  
zeigt keine Fortschrittsinformationen an (*quiet*).

### Beispiel

Das folgende Kommando installiert das Datenbank-Tool DBeaver, wobei die zuvor heruntergeladene Paketdatei verwendet wird:

```
root# gdebi Downloads/dbeaver-ce_7.1.3_amd64.deb
```

```
getcap [optionen] dateiname
```

`getcap` aus dem Paket `libcap` oder `libcap-progs` ermittelt bei ausführbaren Dateien, welche Operationen für das Programm zulässig sind (*Capabilities*). Obwohl der Linux-Kernel bereits seit vielen Jahren Capabilities unterstützt, wird diese Funktion nur sehr sporadisch genutzt.

Capabilities setzen ein Dateisystem mit Extended Attributes voraus. Bei ext-Dateisystemen muss die [mount](#)-Option `user_xattr` verwendet werden.

### Beispiel

Einige Distributionen nutzen Capabilities für das Kommando [ping](#). Die Ausführung dieses Kommandos erfordert eigentlich root-Rechte. Anstatt wie in der Vergangenheit einfach das SetUID-Bit zu setzen, was oft zu Sicherheitsproblemen führt, ist man dazu übergegangen, [ping](#) mit [setcap](#) das Recht cap\_net\_raw+ep zuzuweisen. Damit darf das Kommando, auch wenn es von gewöhnlichen Benutzern ausgeführt wird, elementare Netzwerkfunktionen nutzen.

Das folgende getcap-Ergebnis kam unter Fedora zustande. Sie sehen, dass nur ganz wenige Kommandos und Gnome-Programme Capabilities nutzen:

```
user$ getcap /usr/bin/*
/usr/bin/arping = cap_net_raw+p
/usr/bin/clockdiff = cap_net_raw+p
/usr/bin/gnome-keyring-daemon = cap_ipc_lock+ep
/usr/bin/gnome-shell = cap_sys_nice+ep
/usr/bin/newgidmap = cap_setgid+ep
/usr/bin/newuidmap = cap_setuid+ep
```

Wie Sie Capabilities selbst einstellen, zeigt ein Beispiel bei der Beschreibung des [setcap](#)-Kommandos.

### getenforce

getenforce liefert den Zustand des SELinux-Systems. Die möglichen Ergebnisse sind Enforcing, Permissive oder Disabled. Zur Veränderung des Zustands verwenden Sie das Kommando [setenforce](#).

### getfacl [optionen] dateiname

getfacl ermittelt 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 acl verwendet werden. Ein Beispiel finden Sie bei der Beschreibung des Kommandos [setfacl](#).

- -d  
zeigt die Standard-ACLs an.

- **-R**  
zeigt die ACLs aller Dateien in allen Unterverzeichnissen an.
- **--skip-base**  
liefert keine Ergebnisse bei Dateien, für die nur die gewöhnlichen Unix-Zugriffsrechte gelten, aber keine ACL-Regeln.
- **--tabular**  
zeigt die ACLs in Tabellenform an.

### **getfattr [optionen] dateiname**

`getfattr` ermittelt die erweiterten Attribute der angegebenen Dateien bzw. Verzeichnisse. Das funktioniert nur, wenn das Dateisystem *Extended Attributes* unterstützt ([mount](#)-Option `user_xattr` bei ext-Dateisystemen). Ein Beispiel finden Sie bei der Beschreibung des Kommandos [`setfattr`](#).

- **-d**  
liefert eine Liste aller Benutzerattribute und ihrer Einstellungen. Als Standardmuster für den Attributnamen gilt »user«.
- **-m attributmuster**  
liefert die Attribute, deren Namen dem angegebenen Muster entsprechen.
- **-n attributname**  
liefert den Wert des angegebenen Attributs. Es muss der vollständige Attributname angegeben werden, beispielsweise `user.attrname`.
- **-R**  
zeigt die EAs aller Dateien in allen Unterverzeichnissen an.

### **git kommando [optionen/parameter]**

Das Kommando `git` aus dem gleichnamigen Paket steuert das Versionsverwaltungssystem Git. Es wurde ursprünglich nur zur Kernelentwicklung konzipiert, ist heute aber das am weitesten verbreiteten Versionsverwaltungssystemen auch außerhalb der Open-Source-Welt. Anders als bei Subversion hat jeder Git-Nutzer

ein eigenes, lokales Git-Repository, die sogenannte *working copy*.

Auch Repository-Zweige (*branches*) sind lokal!

Dateien in Git-Repositories können drei Zustände annehmen:

- *committed*: Die lokale Datei ist im Repository gespeichert und wurde seither nicht mehr verändert.
- *modified*: Die lokale Datei wurde verändert, die Veränderungen wurden aber noch nicht durch einen Commit im Repository gespeichert.
- *staged*: Eine veränderte Datei (*modified*) soll beim nächsten Commit berücksichtigt werden.

## Kommandos

- `add dateien/verzeichnisse`  
fügt Dateien oder Verzeichnisse dem Repository hinzu bzw. kennzeichnet bereits im Repository befindliche lokal veränderte Dateien für den nächsten Commit. Damit ändert sich der Status der Dateien von *modified* zu *staged*.
- `branch [-v]`  
listet die lokalen Repository-Zweige auf und markiert den aktuellen Zweig mit einem vorangestellten Stern. Mit der Option `-v` zeigt git zusätzlich den letzten Commit in jedem Branch an.
- `branch zweigname`  
erzeugt einen neuen Repository-Zweig. Um in diesen Zweig zu wechseln, müssen Sie außerdem `git checkout zweigname` ausführen.
- `branch -d zweigname`  
löscht den angegebenen Zweig. Das Kommando wird nur ausgeführt, wenn der Zweig vorher mit einem anderen Zweig oder mit `HEAD` verbunden wurde. Um einen Zweig unabhängig vom Merge-Status zu löschen, verwenden Sie die Option `-D`.
- `branch -u reponame/zweigname`  
stellt das Upstream-Repository für den aktuellen Zweig bleibend ein und speichert die Verknüpfung in `.git/config`. Alternativ kann

das Upstream-Repository auch beim ersten Push-Vorgang mit `git push -u repo zweig` eingestellt werden.

Im Regelfall stimmen der lokale und der externe Zweigname überein; das ist aber nicht zwingend erforderlich. Die Verknüpfung bewirkt, dass die betreffenden lokalen Zweige durch `git pull` und `git push` automatisch verbunden werden, und erspart Ihnen bei diesen Kommandos die wiederholte Angabe des gewünschten Repositorys.

- `checkout [-b] zweigname`

wechselt in den angegebenen Zweig. Damit werden alle Dateien im Arbeitsverzeichnis in den Zustand versetzt, den sie beim letzten Commit in diesem Zweig hatten. Mit der Option `-b` wird der angegebene Branch zuerst erzeugt und dann aktiviert. `git checkout -b b1` entspricht also `git branch b1; git checkout b1`.

- `clone repo-url [verzeichnis]`

lädt ein externes Git-Repository herunter und erstellt eine lokale Kopie im aktuellen bzw. im angegebenen Verzeichnis. Je nachdem, ob die Kommunikation via HTTPS oder via SSH erfolgt, haben die URLs die folgende Form:

`https://github.com/<account>/<reponame>.git`  
`git@github.com:<account>/<reponame>.git`

Je nachdem, wo sich Ihre Repositories befinden, müssen Sie natürlich `github.com` durch Ihre Git-Plattform ersetzen. Das externe Repository wird zukünftig unter dem Namen `origin` angesprochen. Git erlaubt die Verwendung mehrerer externer Repositories, die mit `git remote` verwaltet werden.

- `commit [-a] -m 'snapshot beschreibung'`

erstellt einen neuen lokalen Snapshot (einen »Commit«) im Git-Repository. Darin werden alle Änderungen seit dem letzten Commit gespeichert, aber nur, sofern der Status der Dateien `staged` ist. Sie müssen also vorher alle zu speichernden Dateien mit `git add` als `staged` kennzeichnen.

Mit der zusätzlichen Option `-a` können Sie sich diese Mühe sparen -- `git commit` berücksichtigt nun alle Änderungen an bereits versionierten Dateien. (Neue Dateien werden aber ignoriert.)

Vor dem ersten Commit müssen Sie Ihren Namen und Ihre E-Mail-Adresse mit `git config` in der Git-Konfiguration speichern. Diese Daten werden zusammen mit anderen Metadaten im Commit gespeichert.

Der letzte Snapshot im aktuellen Zweig hat immer den Namen `HEAD`. Der vorletzte Snapshot kann mit `HEAD^` angesprochen werden, der vorvorletzte mit `HEAD^2` etc. (Weitere Beispiele für die Referenz auf andere Commits gibt [Tabelle 1](#).)

- `config --list`  
ermittelt alle im aktiven Verzeichnis gültigen Git-Einstellungen und zeigt diese an. Mit der zusätzlichen Option `--show-origin` gibt das Kommando auch die Herkunft der Einstellung an, d.h., ob die Einstellung lokal für das Repository oder global vorgenommen wurde.
- `config [--global] parameter wert`  
speichert die Einstellung eines Git-Parameters in der Datei `.git/config` des lokalen Git-Repositorys. Mit `--global` wird der Parameter stattdessen in der globalen Git-Konfigurationsdatei `.git/config` im Home-Verzeichnis gespeichert und gilt somit als Default für mehrere Repositories.
- [diff](#)  
ohne weitere Parameter zeigt `git diff` alle seit dem letzten Commit durchgeführten Änderungen. Standardmäßig vergleicht `git diff` das Projektverzeichnis mit dem letzten Commit. Wenn Sie stattdessen den Stage-Bereich mit dem letzten Commit vergleichen wollen, geben Sie zusätzlich die Option `--stage` an. `git diff` zeigt die Änderungen ähnlich wie [diff](#) in einer speziellen Patch-Syntax an. Mit den Optionen `--compact-summary`

oder `--numstat` oder `--shortstat` erhalten Sie stattdessen eine kompakte Zusammenfassung der Anzahl der geänderten Zeilen.

- `diff file`  
zeigt nur die Veränderungen der angegebenen Datei an.
- `diff revision [file]`  
zeigt die Änderungen zwischem dem angegebenen Commit und dem aktuellen Zustand im Projektverzeichnis an.
- `diff revision1..revision2 [file]`  
zeigt die Änderungen zwischen zwei Commits an.
- `fetch [reponame]`  
lädt die zuletzt durchgeföhrten Änderungen von einem externen Git-Repository herunter. Standardmäßig wird dazu das bei `clone` angegebene Repository ausgelesen, also `origin`. Der Befehl `git fetch` aktualisiert also das lokale Repository und integriert alle neuen externen Commits, Zweige und Tags. Um Namenskonflikte zu vermeiden, bekommen neue externe Zweige den Namen `origin/zweigname` bzw. `reponame/zweigname`.  
Zu `git fetch` gibt es die Variante `git pull`: Damit werden die externen Änderungen nicht nur heruntergeladen, sondern auch gleich mit dem lokalen Zweig verbunden. `git pull` entspricht `git fetch; git merge`. Dabei verbindet `merge` lokale Zweige mit den verfolgten Remote-Zweigen (*tracked remote branch*, siehe auch `git branch -u`).
- `grep muster`  
durchsucht alle versionierten Textdateien im Projektverzeichnis nach dem Suchausdruck. *muster* ist ein regulärer Ausdruck wie bei [grep](#).
- `init`  
richtet für das aktuelle Verzeichnis ein Git-Repository ein.
- `log [--oneline] [--graph] [zweig/tag [^nichtzweig]]`  
liefert eine Liste der zuletzt durchgeföhrten Commits. Die Option `--oneline` verkürzt die Ausgabe pro Commit auf eine Zeile. --

graph visualisiert, wie Zweige miteinander verbunden wurden. Wenn Sie als optionalen Parameter einen Zweig oder einen Markierungspunkt (Tag) angeben, erscheinen nur solche Commits, die Einfluss auf diesen Zweig hatten. Dabei kann es sich durchaus auch um Commits handeln, die vor dem Anlegen dieses Zweigs erfolgten, weil diese den Ausgangspunkt des Zweiges formten. Um die von `git log zweigname` produzierte Commit-Liste weiter zu reduzieren, können Sie mit `^nichtzweig` Commits des angegebenen Zweigs ausblenden. `git log bugfix ^master` liefert somit nur solche Commits des `bugfix`-Zweigs, die noch nicht mit `master` verbunden wurden.

Die Ausgabe wird durch `less` geleitet, sodass Sie mit den Cursortasten durch das zumeist seitenlange Ergebnis blättern können. Dieses Verhalten können Sie bei allen `git`-Kommandos durch die Option `--no-pager` unterbinden.

Die Ausgabe von `git log` lässt sich durch unzählige weitere Optionen steuern. Werfen Sie gegebenenfalls einen Blick auf <https://git-scm.com/docs/git-log> mit einem Umfang von ca. 40 Textseiten!

- **`merge zweigname`**

verbindet den angegebenen Zweig mit dem gerade aktuellen Zweig. Die sich daraus ergebenden Änderungen werden wie bei `git commit` in einem neuen Commit für den aktuell aktiven Zweig gespeichert.

Falls bei dem Merge-Prozess Konflikte auftreten, wird der Vorgang unterbrochen. Sie können die Konflikte nun manuell beheben und den Merge-Prozess dann mit `git merge --continue` fortsetzen. Alternativ bricht `git merge --abort` den Prozess ab.

- **`mv bisher neu`**

benennt eine versionierte Datei um bzw. verschiebt sie in ein anderes Verzeichnis.

- **`pull [reponame]`**

überträgt alle Commits von einem externen Repository (wie `git fetch`) und integriert sie in das lokale Repository (wie `git merge`). Dabei wird nur der gerade aktive Zweig berücksichtigt.

`git pull` ohne weitere Parameter setzt voraus, dass das Remote Repository für den aktuellen Zweig in `.git/config` konfiguriert ist. Mit der Option `--rebase` führt `git` keinen gewöhnlichen Merge-Prozess aus, sondern ein Rebasing.

- `push [-u] [reponame [zweigname]]`  
lädt lokal durchgeführte Commits für `zweigname` in das externe Git-Repository `reponame` hoch und verbindet sie dort (`merge`). Beispielsweise verbindet `git push origin master` alle eigenen Commits im `master`-Zweig mit dem gleichnamigen Zweig des externen Repositorys `origin`. Wenn `zweigname` im externen Repository nicht existiert, wird der Zweig dort automatisch eingerichtet. Durch die Option `-u` werden der lokale Zweig und der gleichnamige externe Zweig verknüpft (wie durch `git branch -u`).

Das `push`-Kommando scheitert, wenn es externe Commits gibt, die lokal noch nicht durchgeführt sind. Führen Sie daher vor `git push` immer `git pull` aus.

Wenn Sie `git push` ausführen, ohne Repository- und Zweignamen anzugeben, kommuniziert `git` mit dem Default-Repository (zumeist `origin`) und synchronisiert den aktuellen verknüpften Zweig (siehe das Kommando `branch -u`).

- `rebase zweigname`  
ist eine Variante zu `git merge`. Dabei werden die eigenen Commits so verändert (also zu neuen Commits umgebaut), als wären sie von Anfang an auf Basis der Commits des externen Zweigs entstanden. `git merge` hat den Vorteil, dass die resultierende Commit-Abfolge »schöner« aussieht, also nicht durch ständige Merge-Commits durchbrochen ist. Allerdings entstehen lokale

Commits, die es ursprünglich nie gab. Das erschwert das Nachvollziehen von Fehlern.

- `remote [-v]`  
listet alle externen Repositories auf. Mit `-v` zeigt `git` auch die URL jedes Repositorys an. Das Stamm-Repository `origin` wird durch `git clone` definiert. Weitere Repositories können mit Kommandos wie `git remote add` oder `git remote remove` verwaltet werden.
- `reset HEAD - datei`  
ändert den Status einer irrtümlich als *staged* gekennzeichneten Datei zurück zu *modified*.
- `reset [--hard] revision`  
macht Änderungen rückgängig und führt zurück in eine alte Version des Repositorys. `xxx` gibt die gewünschte Version an: `HEAD` führt zurück zum letzten Commit, `HEAD^` zum vorletzen, `HEAD^2` zum vorvorletzen etc. Alternativ können Sie auch die Anfangsziffern der hexadezimalen SHA1-Prüfsumme des Commits angeben (siehe [Tabelle 1](#)). `reset origin/master` widerruft alle lokalen Änderungen und synchronisiert den lokalen Zweig mit dem `master`-Zweig des Default-Repositorys. Die Option `--hard` bewirkt, dass lokale Änderungen unwiderruflich überschrieben werden.
- `reset datei`  
entfernt die Datei aus dem Stage-Bereich und macht so `git add datei` rückgängig.
- `restore datei`  
stellt den Zustand der Datei aus dem letzten Commit wieder her. Seither durchgeführte Änderungen werden ohne Rückfrage überschrieben. Dieses Kommando gibt es erst ab der Git-Version 2.23 (August 2019). Wenn Sie anstelle einer Datei ein Verzeichnis angeben, gilt das Kommando für alle im Verzeichnis veränderten Dateien.

- `rm datei`  
löscht die Datei aus dem lokalen Verzeichnis und entfernt sie aus dem *staging*-Bereich. Wenn die Datei seit dem letzten Commit verändert wurde, müssen Sie den Löschvorgang mit `--force` erzwingen.
- `show revision:datei`  
zeigt die Datei in dem Zustand, den sie beim angegebenen Commit hatte.
- `status [-s]`  
liefert Informationen über den aktuellen Git-Zweig sowie über den Status veränderter Dateien. Die Option `-s` macht die Statusausgabe kompakter. Allen Dateinamen werden zwei Buchstaben vorangestellt, wobei der erste Buchstabe den *staging*- und der zweite den *modifying*-Zustand angibt.
- `tag`  
ohne weitere Parameter listet `git tag` alle *Tags* (Markierungen, Versions-Marker) auf.
- `tag -a 'marker-text' [-m 'kommentar'] [revision]`  
fügt dem letzten Commit ein *Tag* hinzu, also eine Markierung, eine Art Lesezeichen. Diese Funktion eignet sich gut, um z.B. eine Release-Version zu kennzeichnen. Die Angabe einer Revision ermöglicht es, nachträglich einen älteren Commit zu markieren. Git unterscheidet zwischen zwei Arten von Tags: Mit den Optionen `-a` oder `-m` wird ein annotiertes Tag erzeugt, das auch Metadaten speichert. Ohne diese Option entsteht ein Lightweight-Tag ohne Metadaten.

Zu `git` gibt es nicht eine [man](#)-Seite, sondern eine ganze Menge: Jedes `git`-Kommando ist auf einer eigenen Seite dokumentiert. Beispielsweise gibt `man git-clone` eine umfassende Beschreibung zu `git clone`.

## Revisionssyntax

In git-Kommandos können Sie auf vielfältige Art und Weise auf vergangene Commits bzw. Revisionen verweisen (siehe [Tabelle 1](#)). Als *Head* gilt dabei der jeweils neueste Commit eines Zweigs. Die Schreibweise `@{xxx}` bezieht sich auf das lokale Referenz-Log und kann daher nur auf zuletzt lokal durchgeführte Aktionen angewendet werden.

Beispiel	Bedeutung
HEAD	Head des aktuellen Zweigs
@	Head des aktuellen Zweigs (Kurzschreibweise)
develop	Head eines anderen Zweigs
develop:readme.txt	Readme-Datei am Head eines anderen Zweigs
refs/remots/origin/feature	Head eines entfernten Zweigs
v1.3	Commit mit Tag v1.3
HEAD@{2 days ago}	Commit im aktuellen Zweig vor zwei Tagen
master@{7}	Commit im master-Zweig vor zwei Aktionen
HEAD~	Vorgänger (Parent) des letzten Commits
HEAD~3	Vor-Vor-Vorgänger des letzten Commits
v1.3~3	Vor-Vor-Vorgänger des letzten Commits mit Tag v1.3
234ae33^	erster Elternteil des Merge-Commits 234ae33

Beispiel	Bedeutung
234ae33^^	zweiter Elternteil des Merge-Commits 234ae33

**Tabelle 1** Beispiele für die Git-Revisionssyntax  
**Beispiele**

Im ersten Beispiel speichern Sie zuerst Ihren Namen und Ihre E-Mail-Adresse in Ihrer persönlichen Git-Konfigurationsdatei, sofern Sie das nicht schon früher erledigt haben. Danach laden Sie ein existierendes Repository von GitHub herunter:

```
user$ git config --global user.name "Hermann Huber"
user$ git config --global user.email "huber@meine-firma.de"
user$ git clone https://github.com/git-buch/hello-world.git
user$ cd hello-world
```

Das zweite Beispiel zeigt, wie Sie schnell einen Bugfix im master-Zweig eines vorhandenen Projekts durchführen. Mit `git checkout` wechseln Sie in den master-Zweig. `git pull` lädt eventuell im Remote-Repository gespeicherte Änderungen herunter. Nun korrigieren Sie einen Fehler, speichern Ihre Änderungen mit `git commit` und übertragen die Änderungen mit `git push` zurück in das origin-Repository:

```
user$ cd verzeichnis/mit/vorhandenem/git/repo
user$ git checkout master
user$ git pull
user$ meinlieblingseditor dateien ...
user$ git commit -a -m 'Bugfix Login-Authentifizierung'
user$ git push
```

### glances [optionen]

Das Kommando `glances` aus dem gleichnamigen Paket ist je nach Anwendung eine moderne Alternative zu [top](#) oder ein simples Monitoring-Programm, das im Client/#Server-Modus läuft. Zweiteres erfordert eine entsprechende Konfiguration, ermöglicht dafür aber den Zugriff auf die Monitoring-Daten auch über einen Webbrowser.

Beim direkten Aufruf liefert das Kommando im Vergleich zu [top](#) auch einen Überblick über Netzwerk- und I/O-Aktivitäten sowie

über den Zustand der Dateisysteme. Die Farbdarstellung ist für ein dunkles Terminal optimiert. Wenn Sie eine helle Hintergrundfarbe eingestellt haben, führen Sie `glances` mit der Option `--theme-white` aus.

Diverse Tasten sortieren die Prozessliste nach unterschiedlichen Kriterien: `C` nach CPU-Nutzung, `I` nach I/O-Aktivität, `M` nach Speichernutzung, `P` nach Name oder `T` nach CPU-Laufzeit. Mit `←` können Sie die Prozessliste nach einem Suchausdruck filtern. Weitere Tastenkürzel blenden Informationen ein und aus bzw. aktivieren verschiedene Darstellungsmodi. So wechselt `U` zwischen dem aktuellen Netzwerkdurchsatz und kumulierten Datenmengen. Noch mehr Tastenkürzel zeigt die Hilfeseite, die Sie mit `H` aufrufen.

### **glxinfo [optionen]**

Mit dem Programm `glxinfo`, das sich je nach Distribution im Paket `mesa-utils`, `glx-utils` oder `Mesa-demos-x` versteckt, können Sie die 3D-Funktionen Ihres Systems überprüfen. Das Programm liefert eine Menge Detailinformationen über das laufende GLX-System, also über die OpenGL-3D-Erweiterungen des Grafiksystems. Aus den sehr detaillierten Ergebnissen filtern Sie die relevanten Zeilen am besten mit `grep` heraus.

### **Beispiele**

Die folgenden Zeilen zeigen die 3D-Rendering-Informationen von zwei Testrechnern:

```
user$ glxinfo | grep render (Intel-Treiber, X)
  direct rendering: Yes
  OpenGL renderer string: Mesa DRI Intel(R) Sandybridge Desktop
  ...
user$ glxinfo | grep render (NVIDIA-Treiber, X)
  direct rendering: Yes
  OpenGL renderer string: Quadro P1000/PCIe/SSE2  direct rendering: Yes
```

Wenn kein 3D-beschleunigter Treiber läuft, sieht die Ausgabe wie in einem der drei folgenden Beispiele aus. Im ersten Beispiel stehen gar keine 3D-Funktionen zur Verfügung, im zweiten Fall werden sie

per Software durch die Mesa-Bibliothek nachgebildet und im dritten Fall durch die llvmpipe-Bibliothek. Bei der llvmpipe-Bibliothek kümmert sich die CPU um die 3D-Funktionen. Das erhöht zwar den Rechenaufwand der CPU beträchtlich, ermöglicht aber die Nutzung von 3D-Funktionen auch ohne einen richtigen 3D-Grafiktreiber (z.B. in virtuellen Maschinen).

```
user$ glxinfo | grep render
    Xlib: extension "GLX" missing on display ":0.0"
user$ glxinfo | grep render
    direct rendering: No
    OpenGL renderer string: Mesa GLX Indirect
user$ glxinfo | grep render
    direct rendering: Yes
    OpenGL renderer string: llvmpipe (LLVM 10.0.1, 256 bits)
```

### **gnome-session-quit [optionen]**

Das Programm gnome-session-quit zeigt einen Dialog an, in dem Sie sich abmelden bzw. den Rechner ausschalten oder neu starten können.

- **--force**  
führt den Logout auch dann durch, wenn es noch ungespeicherte Dateien gibt. Diese Option kann nur in Kombination mit **--logout** verwendet werden.
- **--logout**  
fragt den Benutzer, ob er sich abmelden möchte.
- **--no-prompt**  
führt den Logout ohne Rückfrage aus. Diese Option kann nur in Kombination mit **--logout** verwendet werden.
- **--power-off**  
fragt den Benutzer, ob er den Rechner ausschalten möchte.
- **--reboot**  
fragt den Benutzer, ob er den Rechner neu starten möchte.

### **gpasswd [optionen] gruppe**

gpasswd dient eigentlich dazu, Gruppenpasswörter einzustellen. Allerdings gelten Passwörter, die mehreren Mitgliedern einer Gruppe bekannt sein müssen, als unsicher und sind daher nicht

mehr gebräuchlich. gpasswd erfüllt dennoch eine wichtige Funktion. In der folgenden Variante entfernt es einen Benutzer aus einer Gruppe:

```
root# gpasswd -d benutzer gruppe
```

Damit ist gpasswd -d das Gegenstück zu [usermod](#) -a -G gruppe benutzer.

### **gpg bzw. gpg2 [optionen/kommandos]**

gpg verschlüsselt, entschlüsselt oder signiert Dateien und verwaltet Schlüssel. gpg wird durch unzählige Optionen gesteuert und ist für den manuellen Einsatz nur bedingt geeignet. Das Kommando wird unter anderem von E-Mail-Clients zur Verschlüsselung und Signierung von Nachrichten sowie von Paketverwaltungswerkzeugen zur Verifizierung der Integrität von Paketen und Paketquellen eingesetzt.

Mit den Optionen -c und -d kann gpg zur symmetrischen Ver- und Entschlüsselung von Dateien verwendet werden. Das ist einfach, aber sicherheitstechnisch nicht optimal. Alle anderen Optionen gehen davon aus, dass Sie eine asymmetrische Verschlüsselung verwenden möchten: Die Verschlüsselung erfolgt dann mit dem öffentlichen Teil des Schlüssels Ihres Kommunikationspartners, die Entschlüsselung mit dem privaten Teil Ihres eigenen Schlüssels. Das setzt voraus, dass Sie gpg auch zur Schlüsselverwaltung verwenden. gpg ist für den Server- und Embedded-Einsatz gedacht und greift auf wenige externe Bibliotheken zurück. gpg2 ist hingegen für den Desktop-Einsatz konzipiert. Diese Version unterstützt zusätzlich S/MIME. Davon abgesehen, unterstützen beide Kommandos dieselben Optionen.

- --batch

aktiviert den Batch-Modus. In diesem sind interaktive Kommandos nicht zulässig, es erfolgt keine Interaktion. Der Batch-Modus ist zur Script-Programmierung gedacht.

- **-c datei** bzw. **--symmetric datei**  
fordert zur Eingabe eines Passworts auf, verschlüsselt dann die angegebene Datei und speichert das Ergebnis unter dem Namen *datei.gpg*.
- **--cipher-algo zip/zlib/bzip2/none**  
legt den Verschlüsselungsalgorithmus fest. Standardmäßig verwendet gpg den Algorithmus CAST5. Die zur Auswahl stehenden Algorithmen ermittelt das Kommando gpg --version.
- **--compress-algo zip/zlib/bzip2/none**  
gibt an, ob bzw. wie die Daten komprimiert werden sollen. Standardmäßig wird das ZIP-Verfahren eingesetzt. Aus Kompatibilitätsgründen werden andere Einstellungen als [zip](#) oder [none](#) nicht empfohlen. [none](#) bietet sich dann an, wenn die Ausgangsdaten bereits komprimiert sind. Die neuerliche Komprimierung kostet dann unnötig Zeit, ohne den Platzbedarf zu reduzieren.
- **-d datei** bzw. **--decrypt datei**  
fordert zur Eingabe eines Passworts auf, entschlüsselt dann die angegebene Datei und schreibt das Ergebnis in die Standardausgabe.
- **--decrypt datei > klartext**  
entschlüsselt die angegebene Datei und leitet die Ausgabe in die Datei *klartext* um. Die Datei muss mit dem öffentlichen Teil Ihres eigenen Schlüssels verschlüsselt sein. Zum Entschlüsseln müssen Sie das Passwort Ihres Schlüssels angeben.
- **--encrypt datei**  
verschlüsselt die angegebene Datei und speichert sie als *datei.gpg*. Die E-Mail-Adresse des Schlüssels geben Sie interaktiv an. Der Schlüssel muss gegebenenfalls zuvor mit **--import** in der Schlüsseldatenbank gespeichert werden. Wenn Sie vor der Option **--encrypt** zusätzlich **--armor** angeben, wird die verschlüsselte

Datei im ASCII-Format gespeichert. Das erfordert mehr Platz, vereinfacht aber die Weitergabe per E-Mail.

- `--export name@host > public.bin`  
speichert eine Binärdarstellung des öffentlichen Teils des Schlüssels für `name@host` in der angegebenen Datei. Diese Datei müssen Sie an Personen weitergeben, die Ihnen verschlüsselte Dateien oder E-Mails senden sollen. Wenn Sie vor der Option `--export` zusätzlich `--armor` angeben, wird der öffentliche Schlüssel in einer E-Mail-tauglichen Textdarstellung gespeichert.
- `--gen-key`  
erzeugt einen neuen Schlüssel und speichert diesen in `.gnupg/pubring.gpg`. Dabei müssen Sie diverse Daten angeben – die gewünschte Art und Länge des Schlüssels, Ihren Namen, die E-Mail-Adresse, ein Passwort etc.
- `--import public-key-file`  
importiert den öffentlichen Teil eines Schlüssels einer anderen Person in die eigene Schlüsselsammlung. Das ist notwendig, um dieser Person verschlüsselte Dateien bzw. E-Mails senden zu können.
- `-k` bzw. `--list-keys`  
listet alle in `.gnupg/pubring.gpg` gespeicherten Schlüssel auf.
- `-no-tty`  
unterdrückt die Ausgabe von Warnungen.
- `--passphrase-file datei`  
liest das Verschlüsselungspasswort aus der angegebenen Datei.
- `--passphrase-repeat n`  
gibt an, wie oft die Passworteingabe wiederholt werden muss (standardmäßig einmal). Wenn das Passwort in Scripts aus einer Datei gelesen wird (siehe die obige Option), deaktivieren Sie die wiederholte Eingabe mit dem Parameter 0.

- **-q** bzw. **--quiet**  
verzichtet (weitgehend) auf Status-Ausgaben.
- **--verify *signaturdatei* *datei***  
überprüft, ob die Signaturdatei und die Ausgangsdatei zusammenpassen.
- **--version**  
zeigt die Version von gpg und liefert eine Liste aller unterstützten Verschlüsselungs- und Hash-Algorithmen.

## Beispiele

Das folgende Kommando führt eine symmetrische Verschlüsselung des Dokuments *geheim.odt* durch:

```
user gpg -c geheim.odt
Geben Sie die Passphrase ein: ****
Geben Sie die Passphrase nochmals ein: ****
```

gpg erzeugt damit die neue Datei *geheim.odt.gpg*. Die ursprüngliche Datei *geheim.odt* wird nicht verändert. Wenn Sie den Zugriff auf diese Datei im lokalen Dateisystem verhindern möchten, müssen Sie die Datei sowie deren Backups explizit löschen! Um die verschlüsselte Datei wiederherzustellen, gehen Sie so vor:

```
user gpg -d geheim.odt.gpg > geheim.odt
Geben Sie die Passphrase ein: ****
```

Im zweiten Beispiel wird gpg in zwei Scripts zum Ver- und Entschlüsseln von Dateien mit dem AES256-Algorithmus eingesetzt. Voraussetzung für die Scripts ist eine Schlüsseldatei, die mit [openssl](#) erzeugt wird. (32 Bytes erscheinen wenig, aber das sind 256 Bits. Für symmetrische Verfahren gelten bereits 128 Bits als ausreichend sicher.)

```
root# openssl rand 32 > key.txt
root# chmod 400 key.txt
```

Das Verschlüsselungs-Script *mycrypt* sieht so aus:

```
#!/bin/sh
# Verwendung: mycrypt < in > out.crypt
gpg -c --batch --cipher-algo AES256 --compress-algo none --passphrase-file key.txt
--passphrase-repeat 0
```

Um die verschlüsselte Datei wieder zu dechiffrieren, verwenden Sie *myuncrypt*:

```
#!/bin/sh
# Verwendung: myuncrypt < in.encrypt > out
gpg -d --batch --no-tty -q --cipher-algo AES256 --compress-algo none --passphrase-file key.txt --passphrase-repeat 0
```

Die Anwendung sieht so aus:

```
root# ./mycrypt < backup.tar.gz      > backup.tar.gz.crypt
root# ./myuncrypt < backup.tar.gz.crypt > backup-copy.tar.gz
root# file backup.tar.gz.crypt
backup.tar.gz.crypt: GPG symmetrically encrypted data (AES256 cipher)
root# diff backup.tar.gz backup-copy.tar.gz
(keine Ausgabe, die Dateien sind identisch)
```

Das letzte Beispiel zeigt die Überprüfung von Signaturen. Auf vielen Websites werden außer den eigentlichen Downloads auch Textdateien mit Prüfsummen angegeben. Aber wie können Sie sich vergewissern, dass Ihnen nicht nur keine falschen Downloads, sondern auch keine manipulierten Prüfsummen untergeschoben werden? Auf der Download-Seite von Kali Linux ist die Prüfsummendatei deswegen mit einem Schlüssel eines Kali-Entwicklers signiert. Um die Signatur zu überprüfen, importieren Sie zuerst diesen Schlüssel und rufen dann gpg --verify auf:

```
user$ wget -q -O - https://www.kali.org/archive-key.asc | gpg --import
gpg: Schlüssel 7D8D0BF6: Öffentlicher Schlüssel
      "Kali Linux Repository <devel@kali.org>" importiert ...
Primary key fingerprint: 44C6 513A 8E4F B3D3 0875 F758 ED44 4FF0 7D8D 0BF6
```

```
user$ gpg --verify SHA256SUMS.gpg SHA256SUMS
Signature made Thu 09 Nov 2017 05:21:22 PM CET
using RSA key ID 7D8D0BF6
Good signature from "Kali Linux Repository <devel@kali.org>"
```

```
WARNING: This key is not certified with a trusted signature!
There is no indication that the signature belongs to the owner.
Primary key fingerprint: 44C6 513A 8E4F B3D3 0875 F758 ED44 4FF0 7D8D 0BF6
```

Die obige Ausgabe wirkt auf den ersten Blick nicht vertrauenerweckend. Tatsächlich ist aber alles in Ordnung: Das gpg-Kommando bestätigt, dass die Signatur und die Prüfsummendatei übereinstimmen. Die dritte Zeile der gpg-Ausgabe beginnt mit Good signature. Fatal wäre, wenn an dieser Stelle die Meldung wrong signature angezeigt würde!

Die nachfolgende Warnung bezieht sich darauf, dass der Schlüssel der Kali-Entwickler selbst signiert ist, diese Signatur aber nicht

überprüft werden konnte. Diese Warnung können Sie ignorieren: Sie haben den Schlüssel von <https://www.kali.org> heruntergeladen und sich vergewissert, dass sein Fingerabdruck (bei den obigen Kommandos also die hexadezimale Ziffernfolge 44c6 ...) mit den erwarteten Werten übereinstimmt. Um außerdem die Signatur des Schlüssels selbst zu überprüfen, müssten Sie und die Kali-Entwickler Ihre persönlichen GPG-Schlüssel entweder bei einem persönlichen Treffen oder über das sogenannte *Web of Trust* austauschen.

### **gpio** [optionen] kommando

Dieses in der Vergangenheit sehr beliebte Kommando ist veraltet und steht auf aktuellen Installationen von Raspberry Pi OS nicht mehr zur Verfügung. Verwenden Sie stattdessen `raspi-gpio`.

### **grep** [optionen] suchmuster datei

### **grep** -R [optionen] suchmuster

`grep` durchsucht die angegebene Textdatei nach einem Suchmuster und zeigt die gefundenen Textpassagen an oder gibt einfach nur an, wo bzw. in wie vielen Zeilen das Muster gefunden wurde. `grep` kann mit [find](#) kombiniert werden, um alle Dateien, die bestimmten Bedingungen entsprechen, nach Texten zu durchsuchen.

- **-n**  
zeigt nicht nur die Zeile mit dem gefundenen Text an, sondern zusätzlich die *n* unmittelbar vorangehenden und nachfolgenden Zeilen.
- **-c**  
gibt nur die Anzahl der Zeilen an, in denen das Suchmuster gefunden wurde, nicht aber die Zeilen selbst.
- **--color=auto**  
kennzeichnet die Übereinstimmungen bei der Ausgabe farbig. Das funktioniert nur, wenn Sie das grep-Ergebnis direkt in einer Konsole anzeigen, nicht aber, wenn Sie das Ergebnis mit einem anderen Kommando weiterverarbeiten (z.B. mit [sort](#) oder [less](#)).

Um die Farbinformationen auch in diesem Fall in die Ausgabe einzubetten, verwenden Sie `--color=always`.

- `-E`  
aktiviert *Extended Regular Expressions* (EREs). Diese erweiterte Syntax erlaubt die Zeichen ? (null oder einmal), + (mindestens einmal) und | (logisches Oder) im Suchmuster. Anstelle von grep `-E` können Sie auch einfach egrep aufrufen.
- `-f datei`  
liest die hier aufgezählten Optionen der angegebenen Datei (für komplexe oder häufig benötigte Suchmuster).
- `-i`  
unterscheidet nicht zwischen Groß- und Kleinschreibung.
- `-l`  
zeigt nur die Dateinamen an, in denen das Suchmuster gefunden wurde.
- `-n`  
gibt bei der Ausgabe jeder Zeile auch deren Zeilennummer an.
- `-P`  
interpretiert das reguläre Muster gemäß Perl-Regeln.
- `-q`  
liefert keine Bildschirmausgaben und gibt lediglich den Rückgabewert 0 (Suchtext gefunden) oder 1 (nicht gefunden) zurück. Diese Option ist sinnvoll, wenn grep von anderen Programmen aufgerufen wird (siehe das Beispiel bei [find](#)).
- `-R`  
wendet grep rekursiv auf alle Dateien des Verzeichnisbaums an.
- `-v`  
inverse Wirkung: grep liefert alle Zeilen, die dem Suchmuster *nicht* entsprechen. grep `-v '^#' datei | cat -s` liefert alle Zeilen der Datei, die nicht mit # beginnen (wobei cat `-s` mehrere Leerzeilen auf eine einzige reduziert).

- `-w`

findet nur ganze Wörter. Wenn diese Option angegeben ist, wird das Suchmuster »die« im Wort »dieser« nicht mehr erkannt. Das Suchmuster besteht aus zwei Komponenten: aus der Angabe, wonach gesucht wird, und aus der Angabe, wie oft der Suchausdruck auftreten darf. [Tabelle 2](#) fasst die Bedeutungen der wichtigsten Zeichen zusammen. Das Kürzel ERE markiert dabei Zeichen für die *Extended Regular Expressions* (Option `-E` oder Kommando `egrep`). Wenn Sonderzeichen wie `? * + [ ] ( ) #` oder `!` im Suchmuster verwendet werden sollen, muss `\` vorangestellt werden. Für manche Zeichengruppen sind bereits Muster vordefiniert, etwa `[:digit:]` für Ziffern oder `[:space:]` für *White Space* (also Leer- und Tabulatorzeichen, siehe [man](#)-Seite).

Zeichen	Bedeutung
abc	die Zeichenkette »abc«
[abc]	eines der Zeichen a, b oder c
[^abc]	keines der Zeichen a, b oder c (also ein beliebiges anderes Zeichen)
[a-z]	eines der Zeichen zwischen a und z
.	ein beliebiges Zeichen
?	Das vorangegangene Zeichen darf gar nicht oder einmal auftreten (ERE).
*	Das Zeichen darf beliebig oft (auch gar nicht) auftreten.
+	Das Zeichen darf beliebig oft, muss aber mindestens einmal auftreten (ERE).
{n}	Das Zeichen muss genau $n$ -mal auftreten.
{, n}	Das Zeichen darf höchstens $n$ -mal auftreten.

Zeichen	Bedeutung
{n, }	Das Zeichen muss mindestens n-mal auftreten.
{n, m}	Das Zeichen muss mindestens n- und höchstens m-mal auftreten.
^	Zeilenanfang
\$	Zeilenende
	Logisches Oder (ERE)

**Tabelle 2** Aufbau von grep-Suchmustern (ERE = Extended Regular Expression)

Einige Beispiele für Suchmuster: 'abc' sucht nach genau dieser Zeichenkette. '[a-z][0-9]+' sucht nach Zeichenketten, die mit einem Kleinbuchstaben beginnen und in denen eine oder mehrere Ziffern folgen. '\(.\*\)' sucht nach beliebigen Zeichenketten, die in zwei runde Klammern eingeschlossen sind.

grep wendet das Suchmuster nur zeilenweise an. Textpassagen, die durch einen Zeilenumbruch unterbrochen sind, können nicht erkannt werden.

## Beispiele

Das folgende Kommando sucht aus der Apache-Logging-Datei `/var/log/apache2/access.log` alle Zeilen heraus, die die IP-Adresse 1.2.3.4 enthalten. Das Kommando muss mit root-Rechten ausgeführt werden, weil gewöhnliche Benutzer kein Zugriffsrecht auf die Apache-Logging-Dateien haben.

```
root# grep 1.2.3.4 /var/log/apache2/access.log
```

Das folgende Kommando ermittelt, wie oft die Zeichenkette `mysql` in beliebiger Groß- und Kleinschreibung in einer PHP-Datei verwendet wurde:

```
user$ grep -c -i mysql code.php
```

Im nächsten Beispiel werden ausgehend von einem Startverzeichnis rekursiv alle Apache-Konfigurationsdateien durchsucht. grep zeigt

die Dateinamen an, in denen das Schlüsselwort `ErrorDocument` vorkommt:

```
root# cd /etc/apache2  
root# grep -R -l ErrorDocument
```

Das folgende grep-Kommando filtert aus den Kernelnachrichten alle Zeilen heraus, die in beliebiger Groß- und Kleinschreibung das Suchmuster `eth` enthalten:

```
root# dmesg | grep -i eth  
e1000: eth0 NIC Link is Down  
e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX  
...
```

Die Ausgabe des Kommandos `df` enthält diverse Einträge zu temporären Dateisystemen, die oft uninteressant sind. `grep -v` eliminiert diese Zeilen. (`df -x tmpfs` würde ebenfalls zum Ziel führen.)

```
root# df | grep -v tmpfs  
Dateisystem 1K-Blöcke Benutzt Verfügbar Verw% Eingehangt auf  
/dev/mapper/fedora-root 16080304 4820060 10420360 32% /  
/dev/sda1 487652 109669 348287 24% /boot
```

Manche Konfigurationsdateien im Verzeichnis `/etc` sehen so aus, als hätten die Autoren die Datei mit dem Handbuch verwechselt. Zwischen Hunderten, bisweilen Tausenden von Kommentarzeilen sind wenige tatsächlich wirksame Einstellungen versteckt. Die folgenden Kommandos eliminieren alle Zeilen, die mit `#` oder `;` beginnen bzw. die leer sind:

```
root# cd /etc/samba  
root# cp smb.conf smb.conf.orig  
root# grep -Ev '^#|^;|^$' smb.conf.orig > smb.conf
```

Um auch Kommentare zu eliminieren, bei denen `#` oder `;` nicht am Zeilenanfang steht, sowie leere Zeilen mit Whitespace (Leerzeichen, Tabulator-Zeichen), formulieren Sie das grep-Kommando so:

```
root# grep -Ev '^[[[:space:]]*#|^[[[:space:]]*;|^[[[:space:]]*$' \  
smb.conf.orig > smb.conf
```

Das folgende Beispiel durchsucht eine Apache-Konfigurationsdatei nach Zeilen, die das Suchmuster `SSL*File` enthalten:

```
user$ grep 'SSL.*File' /etc/httpd/conf.d/ssl.conf  
SSLCertificateFile      /etc/letsencrypt/live/centos-buch.info/cert.pem  
SSLCertificateKeyFile   /etc/letsencrypt/live/centos-buch.info/privkey.pem  
SSLCertificateChainFile /etc/letsencrypt/live/centos-buch.info/chain.pem
```

Sie können [find](#) und grep auch kombinieren, um besonders wirkungsvolle Suchen durchzuführen. Im folgenden Kommando sucht [find](#) nach \*.php-Dateien und grep überprüft, ob in ihnen die Zeichenkette [mysql](#) vorkommt. Wenn das der Fall ist, wird der Dateiname auf dem Bildschirm ausgegeben. Beachten Sie, dass die Option -print nach -exec angegeben werden muss.

```
root# find -name '*.php' -exec grep -q -i mysql {} \; -print
```

## Varianten zu grep

- [ack](#) und ag sind für Programmierer optimierte grep-Varianten. Sie ignorieren Backup-Dateien, binäre Dateien sowie Verzeichnisse von Versionsverwaltungssystemen.
- egrep entspricht grep -E und unterstützt eine erweiterte Syntax für das Suchmuster. Beispielsweise liefert egrep 'muster1|muster2' datei alle Zeilen der angegebenen Datei, die eines der beiden Muster enthält.
- fgrep entspricht grep -F und liest die Suchmuster zeilenweise aus einer angegebenen Datei.
- grepmail hilft dabei, E-Mail-Archive im mbox-Format nach Zeichenketten zu durchsuchen. Der Vorteil gegenüber grep besteht darin, dass nicht eine einzelne Zeile, sondern die gesamte E-Mail aus der Datei extrahiert wird.
- zgrep durchsucht komprimierte Dateien.

### **groupadd name**

groupadd richtet eine neue Gruppe ein.

- -g n

verwendet n als neue GID-Nummer (Group Identification).

### **groupdel name**

groupdel löscht die angegebene Gruppe.

### **groupmod [optionen] name**

groupmod verändert die GID-Nummer und/oder den Namen der Gruppe.

- **-g n**  
bestimmt die neue GID-Nummer (Group Identification).
- **-n name**  
bestimmt den neuen Gruppennamen.

## Beispiel

Das folgende Kommando gibt der Gruppe `mysqladmins` den neuen Namen `dbadmins`:

```
root# groupmod -n dbadmins mysqladmins
```

```
groups [benutzername]
```

`groups` zeigt eine Liste aller Gruppen an, denen der aktuelle bzw. der angegebene Benutzer angehört. Dabei handelt es sich um die in [/etc/passwd](#) angegebene Hauptgruppe sowie um die optional in [/etc/group](#) angegebenen Gruppen.

## Beispiel

Der aktuelle Benutzer gehört zu den Gruppen `kofler`, `docuteam` und `wheel`. Mitglieder der `wheel`-Gruppe dürfen in RHEL 6 administrative Aufgaben erledigen.

```
user$ groups
kofler docuteam wheel
grub-install [device]
grub2-install [device]
```

Das Kommando `grub-install` (Ubuntu, Debian) bzw. `grub2-install` (CentOS, Fedora, SUSE, RHEL) installiert den Bootloader in den Bootsektor des angegebenen Festplatten-Devices bzw. bei EFI-Rechnern in die EFI-Partition. `grub-install` setzt voraus, dass zuvor die GRUB-Konfigurationsdatei `/boot/grub/grub.cfg` eingerichtet wurde, in der Regel durch den Aufruf von [update-grub](#) (Debian, Ubuntu) oder mit `grub2-mkconfig` (CentOS, Fedora, SUSE, RHEL).

## Beispiel

Unter Ubuntu erstellt das Kommando [update-grub](#) die GRUB-Konfigurationsdatei `/boot/grub/grub.cfg`, wobei automatisch Einträge für alle auf dem Rechner gefundenen Betriebssysteme

angelegt werden. *grub-install* installiert GRUB 2 bei BIOS-Rechnern in den Bootsektor der ersten Festplatte:

```
root# update-grub  
root# grub-install /dev/sda
```

Bei einem EFI-Rechner entfällt die Device-Angabe. *grub-install* setzt voraus, dass die EFI-Partition im Verzeichnis */boot/efi* in den Verzeichnisbaum eingebunden ist.

```
root# update-grub  
root# grub-install /dev/sda
```

Unter CentOS, Fedora, SUSE und RHEL müssen Sie stattdessen diese Kommandos ausführen, mit oder ohne Device-Angabe bei *grub2-install*, je nachdem, ob es sich um eine BIOS- oder um eine EFI-Installation handelt:

```
root# grub2-mkconfig -o /boot/grub2/grub.cfg  
root# grub2-install [/dev/sda]
```

```
grub-mkconfig [optionen]  
grub2-mkconfig [optionen]
```

*grub-mkconfig* (Ubuntu, Debian) bzw. *grub2-mkconfig* (CentOS, Fedora, SUSE, RHEL) wertet die GRUB-Konfigurationsdateien in */etc/grub.d* aus und analysiert alle Festplattenpartitionen. Aus diesen Informationen erzeugt das Kommando eine neue GRUB-Konfigurationsdatei und gibt diese aus.

- *-o datei*  
speichert die resultierende Konfiguration in die angegebene Datei.

### **gsettings** kommando

*gsettings* liest Werte aus der [dconf](#)-Datenbank bzw. speichert Einstellungen dort. Die [dconf](#)-Datenbank befindet sich üblicherweise in der Datei *.config/dconf/user* und liegt in einem Binärformat vor. Die Datenbank wird zur Speicherung diverser Einstellungen von neueren Gnome-Programmen verwendet.

- *get schema schlüssel*  
ermittelt den Wert des Parameters *schlüssel* für das angegebene Schema. Das »Schema« enthält den Software-Anbieter, den

Programmnamen und (optional) ein Einstellungsverzeichnis.  
Beispielsweise sind alle Nautilus-Einstellungen für die Icon-Ansicht dem Schema `org.gnome.nautilus.icon-view` zugeordnet.

- `list-keys schema`  
liefert eine Liste aller Schlüssel für das angegebene Schema.
- `list-schemas`  
liefert eine nicht sortierte Liste aller Schemata.
- `set schema schlüssel wert`  
weist dem Parameter `schlüssel` einen neuen Wert zu.

## Beispiel

Das folgende Kommando stellt den Textskalierungsfaktor ein.  
Damit werden alle Texte um 66 Prozent vergrößert dargestellt. Das ist z.B. bei einem HiDPI- bzw. Retina-Monitor zweckmäßig, auf dem die Texte sonst unleserlich klein sind.

```
user$ gsettings set org.gnome.desktop.interface text-scaling-factor 1.66  
gtf x y freq
```

`gtf` berechnet eine Generalized Timing Formula, also eine Sequenz von Zahlen, die einen VESA-Grafikmodus beschreibt. Die Zahlensequenz kann zur Definition eines Grafikmodus in der Konfigurationsdatei </etc/X11/xorg.conf> bzw. durch das Kommando [xrandr](#) verwendet werden.

## Beispiel

Das folgende Kommando berechnet die Parameter für den Grafikmodus 1280×720 bei einer Bildfrequenz von 60 Hertz:

```
user$ gtf 1280 720 60  
# 1280x720 @ 60.00 Hz (GTF) hsync: 44.76 kHz; pclk: 74.48 MHz  
Modeline "1280x720_60.00" 74.48 1280 1336 1472 1664 720 721 724 746 \  
-HSync +VSync
```

```
gunzip datei
```

Dekomprimiert die angegebene Datei, unabhängig davon, ob sie mit [gzip](#) oder mit [compress](#) komprimiert wurde. Dabei wird automatisch die Kennung `.gz` bzw. `.Z` im Dateinamen entfernt.

`gunzip` ist ein Link auf [gzip](#), wobei automatisch die Option `-d` aktiviert ist.

**gzip [optionen] datei**

`gzip` komprimiert bzw. dekomprimiert die angegebene Datei. Komprimierten Dateien wird automatisch die Kennung `.gz` angehängt. `gzip` eignet sich nur zur Komprimierung einzelner Dateien. Wenn Sie mehrere Dateien (oder ganze Verzeichnisse) in einer komprimierten Datei speichern möchten, müssen Sie zusätzlich das Kommando [tar](#) verwenden.

- **-1 bis -9, --fast, --best**  
steuert die Geschwindigkeit und Qualität der Kompression. `-1` entspricht `--fast` und bewirkt eine schnelle Kompression, aber etwas größere Dateien. `-9` entspricht `--best` und führt zu höheren Rechenzeiten, aber kleineren Dateien. Die Grundeinstellung ist `-6`.
- **-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 (siehe Beispiel unten).
- **-d bzw. --decompress oder --uncompress**  
dekomprimiert die angegebene Datei, anstatt sie zu komprimieren (entspricht [gunzip](#)).
- **-r bzw. --recursive**  
(de)komprimiert auch Dateien in allen Unterverzeichnissen.

## Beispiele

Das folgende Kommando komprimiert alle `*.tif`-Dateien des aktuellen Verzeichnisses. Das Ergebnis sind lauter `*.tif.gz`-Dateien.

```
user$ gzip *.tif
```

Im zweiten Beispiel komprimiert `gzip` die angegebene Datei, lässt sie aber unverändert und schreibt das Ergebnis in `backup.gz`:

```
user$ gzip -c datei > backup.gz
```

# H

## **halt** [optionen]

halt beendet alle laufenden Prozesse und fährt den Rechner dann herunter. halt entspricht [shutdown -h now](#).

- -p

schaltet den Rechner nach dem Shutdown aus (*poweroff*). Bei vielen Systemen erfolgt das Ausschalten automatisch. Die Option ist nur erforderlich, wenn dies nicht funktioniert – z.B. in vielen virtuellen Maschinen.

## **hash** [option]

Das bash-Kommando hash zeigt den Inhalt der Hash-Tabelle an. Dabei handelt es sich um eine Tabelle, in der sich die Shell die Pfadnamen aller bereits ausgeführten Kommandos merkt. Dadurch wird die abermalige Ausführung eines bereits bekannten Kommandos beschleunigt, weil jetzt nicht mehr alle PATH-Verzeichnisse nach dem Programm durchsucht werden müssen.

- -r

löscht die Hash-Tabelle der bash. Das ist notwendig, wenn sich das Verzeichnis eines Programms ändert, das sich in der Hash-Tabelle befindet. Die bash findet das Kommando sonst nicht mehr. In der tcsh muss statt hash -r das Kommando *rehash* verwendet werden.

## **hciconfig** [hcidevice] [kommando]

hciconfig aus dem Paket bluez hilft bei der Konfiguration lokaler Bluetooth-Adapter über das Host Controller Interface (HCI). Sofern kein HCI-Device-Name angegeben wird (üblicherweise `hci0` oder `hci1`), kommuniziert hciconfig mit allen lokalen Bluetooth-Geräten. Wenn das Kommando ohne Parameter ausgeführt wird, listet es Informationen über alle lokalen Bluetooth-Adapter auf.

- reset  
initiiert einen Neustart des Bluetooth-Adapters.
- up/down  
aktiviert bzw. deaktiviert den Bluetooth-Adapter. Wenn dabei die Fehlermeldung *Operation not possible due to RF-kill* angezeigt wird, können Sie versuchen, den Adapter vorher mit `rfkill unblock` einzuschalten.

### **hcitool [optionen] [kommando]**

Das Kommando `hcitool` aus dem Paket `bluez` hilft beim Scannen und Einrichten von Bluetooth-Geräten.

- `-h`  
listet alle unterstützten Kommandos auf.
- `-i hciX`  
wendet das nächste Kommando auf das angegebene Bluetooth-Device an. Ohne diese Option wird das Kommando an das erste verfügbare Gerät gesendet.

## **Kommandos**

- `cc bt-mac`  
stellt eine Verbindung zum Bluetooth-Gerät mit der angegebenen MAC-Adresse her.
- `dc bt-mac`  
beendet die angegebene Verbindung wieder.
- `dev`  
liefert eine Liste der lokalen Bluetooth-Geräte. Normalerweise handelt es sich dabei um den eingebauten Bluetooth-Adapter, dem in der Regel der Device-Name `hci0` zugewiesen ist.
- `scan`  
listet alle in Funkreichweite befindlichen externen Bluetooth-Geräte inklusive ihrer MAC-Adressen auf.

## **Beispiel**

Es gibt nur einen lokalen Bluetooth-Adapter mit dem Device-Namen `hci0`. Alle weiteren Kommandos werden daher automatisch an diesen Adapter gesendet; die Option `-i` kann entfallen. In Funkreichweite befinden sich unter anderem ein Android-Smartphone und eine Maus.

```
user$ hcitool dev  
Devices:  
      hci0    00:1F:CF:41:00:A2  
user$ hcitool scan  
  60:FB:42:FC:BB:8C  Michael Koflers Maus  
  10:68:3F:25:68:18  Nexus 4  
  ...
```

### **head** [optionen] datei

`head` gibt die ersten zehn Zeilen einer Textdatei auf dem Bildschirm aus.

- `-n zeilen`

gibt die angegebene Anzahl von Zeilen aus.

### **help** name

`help` zeigt eine kurze Beschreibung des angegebenen bash-Kommandos an. `help` funktioniert nur für Kommandos, die in die bash integriert sind, beispielsweise [alias](#), [cd](#) oder [type](#).

### **history** [optionen] [n]

Das bash-Kommando `history` zeigt die zuletzt ausgeführten Kommandos mit einer durchlaufenden Nummer an. Der Parameter `n` beschränkt die Ausgabe auf die zuletzt ausgeführten Kommandos. Mit `!n` können Sie das Kommando mit der angegebenen Nummer wiederholen. `!-1`, `-2` etc. führt das letzte, vorletzte Kommando etc. nochmals aus.

- `-c`

löscht die Abfolge der gespeicherten Kommandos im RAM.  
(`.bash_history` bleibt erhalten.)

- `-r [dateiname]`

liest die in der angegebenen Datei bzw. in `.bash_history` gespeicherten Kommandos in den History-Speicher.

- `-w [dateiname]` speichert die Kommandoabfolge (standardmäßig in `.bash_history`).

**host** [optionen] name/ip-adresse

`host` liefert die IP-Adresse zum angegebenen Netzwerknamen bzw. den Netzwerknamen zur angegebenen IP-Adresse. Bei vielen Distributionen ist `host` ein Bestandteil eines bind-Pakets, z.B. `bind-utils` oder `bind9-host`.

- **-a**  
liefert zusätzliche Informationen, die eventuell helfen, Fehler in der Nameserver-Konfiguration zu finden.
  - **-t typ**  
liefert DNS-Einträge des gewünschten Typs.

## Beispiel

Dem Hostnamen *kofler.info* sind die IP-Adressen 5.9.22.29 und 2a01:4f8:161:107::4 zugewiesen. Der Mail-Server hat den Domainnamen `mail.kofler.info`. Die SPF-Informationen des Mail-Servers gehen aus dem DNS-Texteintrag hervor.

hostname liefert den aktuellen Netzwerknamen des Systems bzw. verändert ihn bis zum nächsten Neustart. Die Veränderung wird also nicht bleibend gespeichert! Wenn Sie das möchten, führen Sie [hostnamectl](#) aus oder verändern die entsprechende Konfigurationsdatei.

- -d liefert den Domainnamen anstelle des Hostnamens.

- **-i**  
liefert die dem Hostnamen zugeordnete IP-Adresse. Da es möglich ist, dass dem Host mehrere IP-Adressen zugeordnet sind, sollten Sie statt dieser Option nach Möglichkeit die Option **-I** verwenden.
- **-I**  
liefert alle dem Hostnamen zugeordneten IP-Adressen in einer durch Leerzeichen getrennten Liste. Die `localhost`-Adresse sowie lokale IPv6-Adressen werden dabei nicht berücksichtigt. Die Option **-I** steht bei älteren Versionen von `hostname` nicht zur Verfügung.

### **hostnamectl [optionen] [kommando]**

Mit dem Kommando `hostnamectl` kann bei Distributionen mit dem Init-System `systemd` der Hostname neu eingestellt werden. Anders als mit dem [`hostname`](#)-Kommando wird die neue Einstellung bleibend gespeichert.

- **set-hostname name**  
stellt den neuen Hostnamen ein. Damit die Änderung wirksam wird, müssen Sie sich aus- und neu einloggen. Der neue Hostname wird gleichzeitig in `/etc/hostname` gespeichert.
- **status**  
zeigt den aktuellen Hostnamen sowie diverse Angaben über das laufende System an, darunter den Distributionsnamen und die Kernelversion.

### **Beispiel**

Das folgende Kommando stellt den Hostnamen `arbeitsplatz` und den Domainnamen `mylan` ein:

```
root# hostnamectl set-hostname arbeitsplatz.mylan
```

### **htop [optionen]**

`htop` aus dem gleichnamigen Paket ist eine Variante zu [`top`](#). Der Hauptvorteil gegenüber [`top`](#) besteht darin, dass Sie mit den Cursortasten horizontal und vertikal durch die Prozessliste scrollen

können. Der ausgewählte Prozess kann unkompliziert mit **K** beendet werden. Mit den Funktionstasten können Sie außerdem zwischen verschiedenen Darstellungsformen wählen.

- **-d n**  
gibt an, nach wie vielen Zehntelsekunden die Prozessliste aktualisiert werden soll.
- **-p n1, n2, n3, ...**  
zeigt nur die Prozesse mit den angegebenen PIDs an.
- **-s name**  
sortiert die Prozessliste nach dem angegebenen Kriterium. Eine Liste aller erlaubten Schlüsselwörter gibt `htop -s help`.
- **-u name**  
zeigt nur Prozesse des ausgewählten Benutzers.

**htpasswd [optionen] passworddatei benutzername [passwort]**

`htpasswd` bzw. bei manchen Distributionen `htpasswd2` erzeugt eine Passwortdatei für den Apache-Webserver oder ändert Einträge in einer bereits vorhandenen Passwortdatei.

Die Datei kann zur Basic-Authentifizierung verwendet werden (`AuthType Basic`). Ihr Ort muss in einer Apache-Konfigurationsdatei (z.B. `httpd.conf` oder `.htaccess`) mit dem Schlüsselwort `AuthUserFile` angegeben werden. Aus Sicherheitsgründen sollten Sie unbedingt darauf achten, dass der Webserver die Datei zwar lesen kann, aber nicht via HTTP ausliefern darf!

- **-b**  
erwartet das Passwort als Parameter. Das vereinfacht die Erzeugung von Passworteinträgen per Script, ist aber unsicher.
- **-c**  
erzeugt eine neue Passwortdatei.
- **-D**  
löscht einen Benutzer aus der Passwortdatei.

- **-l**  
sperrt den Account vorübergehend.
- **-u**  
aktiviert einen gesperrten Account wieder.

## Beispiel

Mit den folgenden Kommandos wird die neue Passwortdatei *passwords.pwd* erzeugt und dort ein Eintrag für den Benutzer `name1` eingefügt. Weitere Benutzername/Passwort-Paare werden ohne die Option `-c` hinzugefügt:

```
user$ htpasswd -c passwords.pwd name1
New password: *****
user$ htpasswd passwords.pwd name2
New password: *****
```

## hwclock [optionen]

Ohne weitere Parameter liest `hwclock` die Uhrzeit aus der Hardware-Uhr des Rechners und zeigt sie an.

- **-s bzw. --hctosys**  
liest die Hardware-Uhr aus und setzt damit die Uhrzeit des Rechners.
- **-w bzw. --systohc**  
speichert die aktuelle Uhrzeit des Rechners in der Hardware-Uhr.  
Das Kommando unterstützt darüber hinaus eine Menge Spezialfunktionen, die in `man hwclock` beschrieben sind.

## hydra [optionen] [hostname/ipadresse] dienst

Das Hacking- bzw. Penetration-Testing-Kommando `hydra` aus dem gleichnamigen Paket liest Passwörter aus einer Datei bzw. generiert diese selbst und versucht, mit ihnen einen Login bei einem Netzwerkdienst durchzuführen. `hydra` unterstützt eine Menge Dienste, darunter FTP, HTTP(S), IMAP, MySQL, Microsoft SQL, POP3, PostgreSQL, SMTP, Telnet und VNC.

Das Kommando kann auch Logins in Webformularen versuchen (GET, PUT, POST). Die Liste der zulässigen Dienstnamen ermitteln Sie mit `hydra -h`.

- **-6**  
verwendet nach Möglichkeit IPv6.
- **-C *dateiname***  
verwendet die in der Datei angegebenen Kombinationen aus Login-Name und Passwort. Die Logins und Passwörter müssen zeilenweise in der Form `login:passwort` enthalten sein.
- **-e nsr**  
probiert zusätzlich ein leeres Passwort (*n* wie *null*), den Login-Namen als Passwort (*s* wie *same*) und den umgekehrten Login-Namen (*r* wie *reverse*).
- **-f**  
beendet das Kommando, sobald eine gültige Login/Passwort-Kombination gefunden wurde.
- **-l *loginname***  
verwendet den angegebenen Login-Namen.
- **-L *userdatei***  
liest die Login-Namen zeilenweise aus der angegebenen Textdatei.
- **-m *optionen***  
übergibt zusätzliche Optionen, die spezifisch für den Netzwerkdienst gelten. Zulässige Optionen können Sie mit `hydra -U dienst` ermitteln, also beispielsweise mit `hydra -U http-get` für Logins, die mit einem HTTP-GET-Request durchgeführt werden sollen.
- **-M *hostdatei***  
liest die anzugreifenden Hostnamen bzw. IP-Adressen aus der Datei und greift alle Hosts parallel an.
- **-o *ergebnisdatei***  
speichert die erfolgreichen Login-Passwort-Kombinationen in der angegebenen Datei, anstatt sie auf der Standardausgabe auszugeben.

- **-p *password***  
verwendet das angegebene Passwort.
- **-P *pwdatei***  
probiert die Passwörter aus der angegebenen Textdatei der Reihe nach aus.
- **-R**  
setzt den zuletzt mit **[Strg]+C** unterbrochenen hydra-Aufruf fort, sofern es die Datei `hydra.restore` gibt. Es müssen keine weiteren Optionen angegeben werden; diese sind in `hydra.restore` enthalten.
- **-s *portnr***  
verwendet den angegebenen Port anstelle des Default-Ports des jeweiligen Dienstes.
- **-t *n***  
führt *n* Tasks (Threads) parallel aus. Die Standardeinstellung lautet 16. Das kann zu hoch sein, weil manche Dienste bei zu vielen parallelen Anfragen (noch dazu von derselben IP-Adresse) den Login blockieren.
- **-x *min:max:chars***  
generiert Passwörter, die zwischen *min* und *max* Zeichen lang sind und die angegebenen Zeichen enthalten. Dabei gilt a als Kurzschrifweise für Kleinbuchstaben, A für Großbuchstaben und 1 für Ziffern. Alle anderen Zeichen, unter anderem äöüß, müssen einzeln angegeben werden.  
Beispiel: Mit `-x '4:6:aA1-_\$%'` verwendet `hydra` Passwörter, die vier bis sechs Zeichen lang sind und neben Buchstaben und Ziffern auch die Zeichen -, \_, \$ und % enthalten. Mit `-x '4:4:1'` probiert `hydra` alle vierstellige Zahlen. Das ergibt 10.000 Möglichkeiten.  
Die Option `-x` ist nur in Ausnahmefällen sinnvoll, nämlich wenn Sie (fast) unendlich viel Zeit haben und Ihr Opfer unbegrenzt

viele Login-Versuche toleriert.

## Beispiel

Im folgenden Beispiel versucht `hydra`, auf einem Linux-Server einen Account mit trivialem oder gar keinem Passwort für einen SSH-Login zu finden. Dazu erzeugt zuerst [cut](#) eine Liste aller Accounts. Dieses Kommando führen Sie idealerweise auf einem Rechner aus, auf dem dieselbe Distribution wie auf dem Zielrechner läuft:

```
user$ cut -d: -f1 /etc/passwd > logins.txt
```

Anschließend soll `hydra` für alle in `logins.txt` gespeicherten Accounts einen SSH-Login ausprobieren, wobei als Passwort der Accountname, der umgedrehte Accountname sowie eine leere Zeichenkette verwendet werden:

```
user$ hydra -L logins.txt -t 4 -e nsr 10.0.0.36 targethost
```

## Alternativen

Wenn Ihnen Passwörter in Form von Hashcodes bekannt sind, können Sie versuchen, die Klartextpasswörter mit den Offline-Passwort-Cracker [john](#) herauszufinden. Noch viel schneller ist `hashcat`. Dieses Kommando erfordert aber die mitunter komplizierte Installation geeigneter GPU-Treiber.

|

```
iconv -f zeichensatz1 -t zeichensatz2 in.txt > out.txt
```

iconv führt eine Zeichensatzkonvertierung von Zeichensatz 1 nach Zeichensatz 2 durch. iconv --list liefert eine umfangreiche Liste aller unterstützten Zeichensätze. Das folgende Kommando erzeugt aus einer Latin-1-codierten Textdatei eine entsprechende UTF-8-Datei:

```
user$ iconv -f latin1 -t utf-8 latin1dat > utf8dat
```

```
id
```

id gibt den Namen und die ID-Nummer des Benutzers, seiner primären Gruppe und der weiteren zugeordneten Gruppen an. Unter CentOS, Fedora und RHEL liefert das Kommando auch den SELinux-Kontext.

## Beispiel

Der Benutzer kofler hat die UID 1000, gehört der primären Gruppe kofler mit der GID 1000 an und ist Mitglied der Gruppe wheel mit der GID 10:

```
root# id
uid=1000(kofler) gid=1000(kofler) Gruppen=1000(kofler),10(wheel)
Kontext=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

if bedingung; then
    kommandos
[elif bedingung; then
    kommandos]
[else
    kommandos]
fi
```

if bildet Verzweigungen in bash-Skripts. Der Block nach then wird nur ausgeführt, wenn die Bedingung erfüllt ist. Andernfalls werden beliebig viele elif-Bedingungen ausgewertet. Gegebenenfalls wird der ebenfalls optionale else-Block ausgeführt.

Als Bedingung können mehrere Kommandos angegeben werden. Nach dem letzten Kommando muss ein Strichpunkt folgen. Als

Kriterium gilt der Rückgabewert des letzten Kommandos.

Vergleiche und andere Tests können mit dem Kommando [test](#) durchgeführt werden. Statt [test](#) ist auch eine Kurzschreibweise in eckigen Klammern zulässig. Dabei muss aber nach [ und vor ] jeweils ein Leerzeichen angegeben werden.

```
ifconfig [-a]  
ifconfig schnittstelle  
ifconfig schnittstelle [optionen] [ip-adresse]
```

In der ersten Syntaxvariante liefert `ifconfig` Informationen über alle Netzwerkschnittstellen (ohne `-a` nur für aktive Schnittstellen, mit `-a` auch für noch nicht aktive Schnittstellen). In der zweiten Syntaxvariante zeigt `ifconfig` Informationen über die angegebene Netzwerkschnittstelle an. In der dritten Syntaxvariante richtet das Kommando eine neue Schnittstelle ein bzw. entfernt diese wieder. Die folgenden Optionen gelten nur für die dritte Syntaxvariante:

- `up/down`  
aktiviert bzw. deaktiviert die Schnittstelle.
- `mtu n`  
stellt den Parameter *maximum transfer unit* ein.
- `netmask n`  
gibt die Netzwerkmaske an. Das ist nur notwendig, wenn die Maske von der Standardmaske für die gewählte Adresse abweicht.

Beachten Sie, dass das `ifconfig`-Kommando als veraltet gilt.

Verwenden Sie stattdessen das Kommando [ip](#), das in dieser Kommandoreferenz wesentlich ausführlicher beschrieben wird!

## Beispiel

Die beiden folgenden Kommandos aktivieren die Netzwerkschnittstelle `eth0` und weisen ihr die Adresse 192.168.0.2 zu. Für eine manuelle Einbindung des Rechners in ein lokales Netzwerk ist darüber hinaus auch eine Nameserver-Konfiguration in der Datei `/etc/resolv.conf` sowie die Einrichtung einer Defaultroute mit dem Kommando [route](#) erforderlich.

```
root# ifconfig eth0 up  
root# ifconfig eth0 192.168.0.2
```

### **iftop [optionen]**

iftop beobachtet den Netzwerkverkehr einer bzw. aller Netzwerkschnittstellen und zeigt auf einer Seite, die alle drei Sekunden aktualisiert wird, zu welchen Hosts bzw. IP-Adressen die meisten Daten fließen. iftop läuft wie [top](#), bis es mit  beendet wird.

- **-B**  
rechnet in Bytes/s statt in Bits/s.
- **-F *ipadr/mask***  
berücksichtigt nur Verkehr von der bzw. zu der angegebenen Adresse.
- **-G *ip6addr/mask***  
berücksichtigt nur Verkehr von der bzw. zu der angegebenen IPv6-Adresse.
- **-i *name***  
berücksichtigt nur die angegebene Netzwerkschnittstelle.
- **-n**  
zeigt IP-Adressen statt Hostnamen.

### **ifup schnittstelle** **ifdown schnittstelle**

ifup aktiviert die angegebene Schnittstelle, ifdown deaktiviert sie wieder. Die Kommandos werden vom Init-System zur Netzwerkinitialisierung aufgerufen und greifen auf die distributionsspezifischen Konfigurationsdateien zurück. Daher variiert die Implementierung der Kommandos je nach Distribution; auch die verfügbaren Optionen und deren Bedeutung hängen von der Distribution ab (siehe `man ifup/ifdown`). In einigen aktuellen Distributionen (Ubuntu, Raspberry Pi OS) stehen die Kommandos gar nicht mehr zur Verfügung oder funktionieren nur noch mit großen Einschränkungen.

## Beispiel

Die beiden folgenden Kommandos fahren die Netzwerkschnittstelle eth0 zuerst herunter und dann wieder hoch – beispielsweise, um eine geänderte Konfiguration zu aktivieren:

```
root# ifdown eth0  
root# ifup eth0
```

### info [kommandoname]

info startet das gleichnamige Online-Hilfesystem. Zur Navigation im Hilfetext verwenden Sie die in [Tabelle 14](#) zusammengefassten Tastenkürzel. info-Texte können Sie alternativ auch mit dem Kommando pinfo aus dem gleichnamigen Paket, mit dem Editor Emacs oder in den Hilfesystemen von Gnome und KDE lesen. Alle Varianten bieten mehr Komfort als das Original.

- -f datei

lädt die angegebene Datei statt einer Datei aus `/usr/share/info`.

Wenn der info-Text auf mehrere Dateien verteilt ist, muss die erste Datei angegeben werden (etwa *elisp-1.gz*).

### init [n]

init aktiviert den durch *n* angegebenen Runlevel. Das funktioniert nur bei Distributionen, die das traditionelle Init-V-System einsetzen. Bei Distributionen mit dem Init-System systemd verändern Sie den Runlevel bzw. genau genommen das »Target« mit dem Kommando [systemctl isolate](#).

### inotifywait [optionen] [dateien/verzeichnisse]

inotifywait aus dem Paket inotify-tools überwacht die Veränderungen von Dateien bzw. deren Metadaten. In der einfachsten Form übergeben Sie einen oder mehrere Dateinamen an das Kommando. In diesem Fall wartet das Kommando, bis für eine dieser Dateien ein inotify-Event auftritt, also z.B. eine Veränderung der Datei, ein Lesezugriff etc. Damit endet das Kommando.

Das Kommando wird häufig in Scripts eingesetzt, um automatisiert auf Änderungen von Dateien zu reagieren. Alternativ können Sie die Überwachung auch unbegrenzt durchführen und die aufgetretenen Ereignisse protokollieren.

- **-d bzw. -m**  
arbeitet als Hintergrundprozess (*-d, daemon*) oder im Vordergrund (*-m, monitor*). `inotifywait` endet nun nicht beim ersten auftretenden Ereignis, sondern läuft, bis es explizit beendet wird, z.B. durch [kill](#) oder `Strg + C`.
- **-e event**  
reagiert nur auf das angegebene Ereignis. Standardmäßig verarbeitet das Kommando alle `inotify`-Ereignisse. Die Option `-e` kann mehrfach angegeben werden, um mehrere Ereignisse auszuwählen. Zu den wichtigsten Ereignissen zählen `access`, `close`, `create`, `delete`, `modify`, `move` und `open`. Eine detaillierte Beschreibung aller Ereignisse gibt die [man](#)-Seite.
- **--fromfile datei.txt**  
liest die Liste der zu überwachenden Dateien oder Verzeichnisse zeilenweise aus *datei.txt*.
- **-q**  
verzichtet auf unnötige Ausgaben (*quiet*).
- **-r**  
beobachtet rekursiv auch alle Unterverzeichnisse des angegebenen Startverzeichnisses. Dabei wird für jede einzelne Datei eine `inotify`-Überwachung eingerichtet. Bei Verzeichnissen mit vielen Dateien dauert das eine Weile und erfordert relativ hohe Ressourcen. Die Maximalanzahl der Überwachungen ist normalerweise mit 8192 festgelegt. Dieser Wert kann bei Bedarf in der Datei `/proc/sys/fs/inotify/max_user_watches` verändert werden.

- `-t n`  
endet in jedem Fall nach  $n$  Sekunden, auch wenn kein Ereignis auftritt.

## Beispiel

Das folgende Shell-Script überwacht die Dateien `*.md` im aktuellen Verzeichnis. Bei jeder Veränderung in einer dieser Dateien überprüft es, ob es eine `*.md`-Datei gibt, die aktueller als die entsprechende `*.pdf`-Datei ist. In diesem Fall wird das betreffende PDF-Dokument mit [pandoc](#) neu erzeugt. Das Script läuft endlos, bis es mit Strg + C beendet wird.

```
#!/bin/bash
while :
do
    for mdfile in *.md; do
        pdffile=${mdfile%.*}.pdf
        if [ $mdfile -nt $pdffile ]; then
            echo $mdfile
            pandoc -t beamer -H header.tex $mdfile -o $pdffile
        fi
    done
    inotifywait -e modify -q *.md
done
```

**insmod** [optionen] moduldatei [parameter=wert ...]

`insmod` lädt das angegebene Kernelmodul. Dabei muss der vollständige Dateiname übergeben werden. Zusätzlich können Parameter (Optionen) an das Modul übergeben werden. Falls Sie hexadezimale Werte angeben möchten, müssen Sie `0x` voranstellen, also etwa `option=0xff`. Die zur Auswahl stehenden Parameter des Moduls können Sie mit [modinfo](#) ermitteln.

- `-f`  
versucht, das Modul selbst dann zu laden, wenn es nicht für die laufende Kernelversion kompiliert wurde. Ob das tatsächlich funktioniert, hängt davon ab, ob es zwischen der Kernel- und der Modulversion irgendwelche Inkompatibilitäten gibt. Die Option ist vor allem dann sinnvoll, wenn Hardware-Hersteller ein Modul nur als Binärversion (ohne Quellcode) zur Verfügung stellen. Die

Option ist aber natürlich keine Garantie dafür, dass das Modul tatsächlich kompatibel zu Ihrer Kernelversion ist.

### **ionice [optionen] [kommando]**

ionice führt das angegebene Kommando mit einer veränderten I/O-Priorität aus. ionice hat damit eine ähnliche Funktion wie [nice](#), beeinflusst aber I/O-Operationen und nicht die CPU-Auslastung.

- **-c n**  
gibt die gewünschte Scheduling-Klasse an. Zulässige Einstellungen sind:
  - 0: keine Präferenzen
  - 1: *realtime*, also maximale I/O-Geschwindigkeit
  - 2: *best-effort*, gilt standardmäßig.
  - 3: *idle*, also I/O-Operationen nur durchführen, wenn das System gerade nicht ausgelastet ist.
- **-n n**  
gibt die Prioritätsstufe innerhalb der gewählten Scheduling-Klasse an. Die zulässigen Werte reichen von 0 (maximale Priorität) bis 7 (minimale Priorität). Prioritätsstufen sind nur für die Scheduling-Klassen 1 und 2 vorgesehen und erlauben eine Differenzierung innerhalb der Klasse.
- **-p pid**  
verändert die I/O-Priorität des durch die ID-Nummer angegebenen Prozesses.

### **Beispiel**

Das folgende Kommando startet ein Backup-Script mit minimaler I/O-Priorität:

```
root# ionice -c 3 backupscript
```

### **iotop**

iotop aus dem gleichnamigen Paket zeigt die I/O-Aktivität aller laufenden Prozesse an. Das hilft bei der Suche nach Prozessen, die die Festplatte oder andere Datenträger besonders stark beanspruchen.

- **-o**  
zeigt nur Prozesse an, die tatsächlich I/O-aktiv sind (und nicht standardmäßig alle laufenden Prozesse).
- **-u bzw. --user=user**  
zeigt nur die Prozesse des angegebenen Benutzers.

### **ip [optionen] objekt kommando**

ip ist ein ungemein vielseitiges Kommando, um Informationen über Netzwerk-Devices, Tunnel, Routing-Regeln etc. zu ermitteln bzw. diese Einstellungen zu ändern. ip sollte anstelle von [ifconfig](#) und [route](#) verwendet werden, da diese beiden Kommandos als veraltet gelten.

- **-f fam bzw. -family fam**  
bestimmt das gewünschte Netzwerkprotokoll (inet, inet6 oder link). Statt -f inet ist die Kurzschreibweise -4 zulässig, statt -f inet6 die Option -6, statt -f link die Option -0.
- **-o bzw. -oneline**  
fasst zusammengehörende Ausgaben in einer Zeile zusammen. Das reduziert die Lesbarkeit, vereinfacht aber die Weiterverarbeitung durch [grep](#) oder [wc](#).
- **-r bzw. -resolve**  
löst IP-Adressen auf und zeigt stattdessen die Hostnamen an. Das erfordert einen Nameserver.

Als objekt muss eines der folgenden Schlüsselwörter angegeben werden: addr, addrlabel, link (also eine Netzwerkschnittstelle), maddr (eine Multicast-Adresse), mroute, monitor, neighbor (ein ARP- oder NDISC-Cache-Eintrag), [route](#), rule oder tunnel.

Diese Schlüsselwörter dürfen abgekürzt werden. Für die meisten Objekte stehen die Kommandos add, delete und list = show zur Auswahl. Die weiteren Kommandos sind objektspezifisch. In der folgenden Referenz beschränke ich mich auf die wichtigsten Kommandos für die Objekte addr, link und [route](#):

- `ip addr [show dev xxx]`  
zeigt die IP-Adressen aller Schnittstellen. Die Ausgabe umfasst normalerweise mehrere Zeilen. Die mit `link/ether` beginnende Zeile gibt die MAC-Adresse der Schnittstelle an. Die mit `inet` beginnende Zeile enthält die IPv4-Adresse samt Maske in der Kurzschreibweise `/n` sowie die Broadcast-Adresse. Die mit `inet6` beginnenden Zeilen geben die IPv6-Adressen an; das können mehrere sein. Mit `-4` oder `-6` kann die Ausgabe auf IPv4 oder IPv6 eingeschränkt werden. `ip addr show dev xxx` liefert nur Informationen zur angegebenen Schnittstelle.
- `ip addr add n/m dev xxx`  
fügt die IP-Adresse `n` mit der Maske `m` der Schnittstelle `xxx` hinzu. Eine zulässige IPv4-Adresse samt Maske wäre z.B. `10.0.45.34/24`.
- `ip addr del n/m dev xxx`  
macht die Adresszuweisung zur Schnittstelle `xxx` rückgängig. Es müssen exakt dieselben Parameter wie bei `ip addr add` angegeben werden.
- `ip addr flush dev xxx`  
löscht *alle* Adresszuweisungen der Schnittstelle `xxx`.
- `ip link [show dev xxx]`  
liefert eine Liste aller Netzwerkschnittstellen, im Gegensatz zu `ip addr show` aber ohne die Angabe von IP-Adressen.
- `ip link set xxx up/down`  
aktiviert bzw. deaktiviert die Netzwerkschnittstelle.
- `ip neigh`  
liefert eine Liste aller anderen im lokalen Netzwerk bekannten IP-Adressen, also eine Aufzählung der »Nachbarn«.
- `ip route [list]`  
gibt die IPv4-Routing-Tabelle aus. Wenn Sie IPv6-Daten wünschen, müssen Sie die Option `-6` angeben. Die Gateway-Adresse geht aus der Zeile hervor, die mit `default` beginnt.

- `ip route add default via n`  
legt die IP-Adresse *n* als Default-Gateway fest.
- `ip route add n1/m via n2 dev xxx`  
definiert für den Adressbereich *n1/m* die Routing-Adresse *n2*. Die IP-Pakete werden über die Schnittstelle *xxx* geleitet.
- `ip route del ...`  
entfernt den angegebenen Routing-Eintrag. Die Parameter müssen exakt mit denen des Kommandos `ip route add` übereinstimmen.

Die Dokumentation des `ip`-Kommandos ist über mehrere [man](#)-Seiten verteilt. `man ip` gibt lediglich einen Überblick. `ip-address` liefert Details zu `ip addr`, `ip-route` zu `ip route` etc. Außerdem können Sie mit `ip object command help` eine Syntaxbeschreibung eines bestimmten Kommandos ermitteln, also beispielsweise mit `ip addr del` die Syntax zum Auflösen einer Adresszuordnung.

## Beispiel

Das folgende Kommando zeigt die aktuelle Routing-Tabelle an. Bei vielen modernen Distributionen kommen dabei anstelle von `eth0` Schnittstellennamen wie `enp0s3` zur Anwendung.

```
user$ ip route show
10.0.0.0/24 dev eth0 proto kernel scope link src 10.0.0.41 metric 1
default via 10.0.0.138 dev eth0 proto static
```

Um der Schnittstelle `eth0` die Adresse `10.0.0.41` zuzuweisen und das Gateway `10.0.0.138` einzurichten, führen Sie die folgenden Kommandos aus:

```
root# ip route add 10.0.0.41/24 dev eth0
root# ip route add default via 10.0.0.138 dev eth0
```

Die folgenden Kommandos zeigen eine IPv6-Konfiguration:

```
root# ip -6 addr add 2a01:4f8:107::2/64 dev eth0
root# ip -6 route add default via fe80::1 dev eth0
```

Eine kompakte Liste aller Netzwerkschnittstellen liefert `ip -o link`:

```
root# ip -o link
1: lo:      <LOOPBACK,UP,LOWER_UP>          mtu 16436 ...
2: eth0:    <BROADCAST,MULTICAST,UP,LOWER_UP>  mtu 1500 ...
3: br0:     <BROADCAST,MULTICAST,UP,LOWER_UP>  mtu 1500 ...
4: virbr0:  <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 ...
5: vnet0:   <BROADCAST,MULTICAST,UP,LOWER_UP>  mtu 1500 ...
```

### **ipcalc** [ipaddress [netmask]]

ipcalc aus dem gleichnamigen Paket ermittelt aus einer gegebenen IPv4-Adresse und der Netzmase alle weiteren Parameter, also die Netzwerkadresse, die Broadcast-Adresse etc.

```
user$ ipcalc 10.11.12.13/16
Address: 10.11.12.13          00001010.00001011. 00001100.00001101
Netmask: 255.255.0.0 = 16      11111111.11111111. 00000000.00000000
Wildcard: 0.0.255.255         00000000.00000000. 11111111.11111111

Network: 10.11.0.0/16          00001010.00001011. 00000000.00000000
HostMin: 10.11.0.1             00001010.00001011. 00000000.00000001
HostMax: 10.11.255.254         00001010.00001011. 11111111.11111110
Broadcast: 10.11.255.255       00001010.00001011. 11111111.11111111
Hosts/Net: 65534               Class A, Private Internet
```

### **iptables** [optionen]

### **ip6tables** [optionen]

iptables konfiguriert den Filter für Netzwerkpakete (kurz *Netfilter*) des Linux-Kernels. Die iptables-Optionen folgen einem einfachen Schema: Eine Option in Großbuchstaben gibt die durchzuführende Aktion an (beispielsweise -P zur Einstellung des Standardverhaltens). Weitere Optionen in Kleinbuchstaben steuern die Details dieser Aktion. Diese Syntaxzusammenfassung ist nach Aktionen gegliedert.

Beachten Sie, dass der Linux-Kernel vollkommen getrennte Filtertabellen für IPv4 und für IPv6 verwaltet. Dementsprechend gibt es auch zwei Konfigurationskommandos: iptables für IPv4 und ip6tables für IPv6. Die folgende Beschreibung gilt gleichermaßen für beide Kommandovarianten.

Bevor Sie iptables aufrufen, sollten Sie sich jedoch vergewissern, dass Ihre Distribution kein eigenes Firewall-System hat. CentOS, Fedora, RHEL und SUSE setzen z.B. auf *Firewalld* (siehe [firewall-cmd](#)). Selbst definierte Firewall-Regeln führen dann leicht zu Konflikten. Moderne Distributionen verwenden anstelle von *Netfilter* das neue Firewall-System *Nftables*. Zum Glück macht dieser Umstieg das Kommando iptables nicht obsolet – es kann dank einer Kompatibilitätschicht weiterverwendet werden. (Unzählige

Firewall-Scripts würden sonst nicht mehr funktionieren.) Es gibt aber natürlich auch die Möglichkeit, *Nftables* direkt zu steuern – und zwar mit dem Kommando [nft](#).

#### **iptables -P chain policy [-t table]**

`iptables -P (policy)` definiert das Standardverhalten für die angegebene Regelkette. Mögliche Verhalten sind:

ACCEPT: Paket weiterleiten (Grundeinstellung)

DROP: Paket löschen

RETURN: Paket zurücksenden (selten)

QUEUE: Paket an ein Programm außerhalb des Kernels weiterleiten (selten)

Standardmäßig gilt das Kommando für *Filter*-Regelketten oder für selbst definierte Regelketten. Falls eine *NAT*- oder *Mangle*-Regelkette verändert werden soll, muss der Tabellenname mit der Option `-t` angegeben werden, z.B. `iptables -P POSTROUTING ACCEPT -t nat`. Es ist nicht möglich, ein Standardverhalten für selbst definierte Regelketten zu definieren. Sie können das Standardverhalten aber bei Bedarf durch die letzte Regel definieren, z.B. durch `iptables -A mychain -j DROP`.

#### **iptables -A chain [-t table] options**

`iptables -A (add)` fügt der angegebenen Regelkette eine neue Regel hinzu. Generell gilt eine Regel für alle möglichen Fälle (d.h. für alle IP-Protokolle, für alle Ports, für alle Absender- und Zieladressen, für alle Interfaces etc.).

Durch Optionen kann die Gültigkeit eingeschränkt werden. Die meisten Optionen können mit einem Ausrufezeichen auch verneint eingesetzt werden. Mit `-p udp` gilt eine Regel also beispielsweise nur für UDP-Pakete. Mit `-p ! udp` gilt sie hingegen für alle Pakete außer für UDP-Pakete.

Nicht alle möglichen Kombinationen der Optionen sind zulässig.

Beispielsweise dürfen die Optionen `-d` und `-s` nur für tcp-Pakete verwendet werden, also in Kombination mit `-p tcp`.

- **-d *ipadresse***  
gibt die Zieladresse an (*destination*). Adressbereiche können in der Form 192.168.0.0/24 oder 192.168.0.0/255.255.255.0 angegeben werden. In beiden Fällen sind alle IP-Nummern 192.168.0.\* gemeint.
- **[--dport *port[:port]*]** gibt den Port oder Port-Bereich (z.B. 0:1023) der Zieladresse an.
- **-i *interface***  
gibt das Interface an, aus dem das IP-Paket kommt (nur für *Input*-, *Forward*- und *Prerouting*-Regelketten). Beim Interface-Namen ist das Sonderzeichen + als Platzhalter für alle Interface-Nummern erlaubt, also ppp+ für ppp0, ppp1 etc.
- **-j ACCEPT/DROP/mychain/..**  
gibt an, was mit dem Paket geschehen soll (*jump*). Hier wird meistens eines der vorgegebenen Verfahren (ACCEPT, DROP etc.) angegeben. Für Spezialanwendungen sieht [iptables](#) REDIRECT oder MASQUERADE vor.  
Anstelle eines der vordefinierten Schlüsselwörter kann auch eine selbst definierte Regelkette angegeben werden. In diesem Fall werden alle Regeln dieser Kette angewandt. Falls keine Regel der selbst definierten Regelkette zutrifft, kommt die nächste Regel der ursprünglichen Regelkette zur Anwendung. In der prozeduralen Programmierung würde das einem Unterprogrammaufruf entsprechen.
- **-m *module***  
gibt an, dass ein Zusatzmodul verwendet werden soll. In der Folge dürfen spezielle Optionen verwendet werden, die durch dieses Zusatzmodul definiert sind. Ein besonders wichtiges Zusatzmodul ist state. Damit können Pakete nach ihrem Verbindungsstatus ausgewählt werden. Beispielsweise gilt eine Regel mit **-m state --state NEW** nur für IP-Pakete, die neue

Verbindungen initiieren. Mit `--state` können folgende Statusschlüsselwörter angegeben werden:

- NEW:** Das Paket initiiert eine neue Verbindung.
- ESTABLISHED:** Das Paket gehört zu einer schon existierenden Verbindung.
- RELATED:** Das Paket initiiert eine neue Verbindung, gehört aber zu einer schon existierenden Verbindung.
- INVALID:** Das Paket gehört zu keiner vorhandenen Verbindung und initiiert auch keine neue Verbindung.

- **-o *interface***  
gibt das Interface an, zu dem das IP-Paket unterwegs ist (nur für *Output-, Forward- und Postrouting*-Regelketten).
- **-p *protocol***  
bestimmt das Protokoll (z.B. `tcp`, `udp` oder `icmp`).
- **-s *ipadresse***  
gibt die Absenderadresse an (*source*).
- **[--sport *port[:port]*]** gibt den Port oder Port-Bereich für den Absender an.
- **--syn**  
gibt an, dass die Regel nur für solche TCP-Pakete gelten soll, bei denen das SYN-Bit gesetzt ist. Derartige Pakete werden verwendet, um eine Verbindung zu initiieren (etwa für alle TCP-Wrapper-Funktionen, für HTTP etc.).

[iptables](#) bietet die Möglichkeit, die Wirksamkeit einzelner Regeln durch `syslogd` zu protokollieren. Dazu geben Sie bei der Regel als Aktion `LOG` an. Damit eine Regel sowohl wirksam ist als auch protokolliert wird, muss sie zweimal angegeben werden: einmal mit `-j LOG` und ein zweites Mal mit `-j ACCEPT` bzw. `-j DROP!` Beachten Sie, dass durch Logging-Regeln sehr rasch riesige Protokolldateien entstehen können.

### **iptables -N mychain**

iptables -N (*new*) erzeugt eine neue Regelkette mit dem Namen *mychain*.

### **iptables -L [chain] [-t table] [-v]**

iptables -L (*list*) liefert ohne weitere Optionen eine Liste aller Regeln für die drei Regelketten der *Filter*-Tabelle sowie für alle selbst definierten Regelketten.

Mit den weiteren Optionen können Sie die gewünschte Regelkette genau spezifizieren (z.B. iptables -L mychain oder iptables -L POSTROUTING -t nat). Die Zusatzoption -v bewirkt detailliertere Informationen. -n führt dazu, dass bei der Ausgabe IP- und Port-Nummern angezeigt werden (statt Netzwerk- bzw. Port-Namen).

### **iptables -D chain [-t table] options**

iptables -D (*delete*) löscht die Regel aus der Regelkette. Es müssen exakt dieselben Optionen wie bei iptables -A angegeben werden.

### **iptables -F chain [-t table]**

iptables -F (*flush*) löscht alle Regeln aus der angegebenen Regelkette.

### **iptables -X [mychain]**

iptables -x löscht die angegebene eigene Regelkette. Wenn keine Regelkette angegeben wird, werden alle selbst definierten Regelketten gelöscht.

## **Beispiel**

Die folgenden iptable-Kommandos definieren eine Mini-Firewall für IPv4. Dabei werden eintreffende Pakete nur dann akzeptiert, wenn sie entweder einer bereits existierenden Verbindung zuzuordnen sind oder *nicht* von der Schnittstelle eth0 stammen, über die der Rechner mit dem Internet verbunden ist.

Das Beispiel geht davon aus, dass sich die [iptables](#)-Tabellen anfänglich im Defaultzustand befinden, also alle Pakete akzeptieren, und dass es nur eine Schnittstelle zum Internet gibt.

```
root# iptables -N wall
root# iptables -A wall -m state --state ESTABLISHED,RELATED -j ACCEPT
root# iptables -A wall -m state --state NEW ! -i eth0 -j ACCEPT
root# iptables -A wall -j DROP
root# iptables -A INPUT -j wall
root# iptables -A FORWARD -j wall
ip[6]tables-save [optionen]
ip[6]tables-restore [optionen]
ip[6]tables-xml [optionen]
```

Die Kommandos `iptables-xxx` bzw. `ip6tables-xxx` helfen dabei, die Regeln einer Paketfilter-Firewall in einer Datei zu speichern bzw. daraus zu lesen. Jedes im Folgenden kurz beschriebene Kommando gibt es auch in einer IPv6-Variante.

- `iptables-save` gibt alle Regeln aller Firewall-Filter aus. Dabei wird eine gut lesbare Syntax verwendet. Die Ausgabe kann mit `> in` eine Datei umgeleitet werden. Die Option `-t name` bewirkt, dass nur die Regeln des angegebenen Filters gespeichert werden.
- `iptables-restore` liest Regeln aus der Standardeingabe. Die Regeln ersetzen normalerweise bereits vorhandene Filter-Regelketten. Wenn Sie das nicht möchten, geben Sie die Option `-n` an. Mit `-T name` werden nur die Regeln des angegebenen Filters berücksichtigt.
- `iptables-xml` funktioniert ähnlich wie `iptables-save`, erzeugt aber ein XML-Dokument.

### **iw** objekt kommando

Mit dem Kommando `iw` steuern Sie WLAN-Adapter, die die nl80211-Schnittstelle unterstützen. Das ist bei den meisten aktuellen WLAN-Adaptoren der Fall, deren Treiber auf dem mac80211-Framework basieren.

Es gibt mehrere Möglichkeiten, um das Objekt anzugeben, das Sie steuern wollen. Dabei darf das Kürzel `dev` bzw. `phy` weggelassen werden, wenn der Schnittstellen- oder Gerätename eindeutig ist.

- `dev name`  
gibt den Schnittstellennamen an (z.B. `dev wlan0`).

- `phy name` bzw. `phy #n`  
gibt den Namen bzw. die Indexnummer des Geräts an. Bei Notebooks mit einem WLAN-Adapter lautet der Gerätename immer `phy0`.
- `reg`  
steuert den *regulatory agent*, also ein Regelwerk für nationale Funkstandards.

Die zur Auswahl stehenden Kommandos hängen vom Objekttyp ab.  
Im Folgenden stelle ich nur einige ausgewählte Kommandos vor:

- `dev name connect ssid`  
stellt eine Verbindung zum angegebenen WLAN-Netzwerk her.  
Das gelingt nur bei Netzwerken ohne Verschlüsselung. Wenn das Funknetzwerk durch WEP abgesichert ist, geben Sie den Schlüssel durch den optionalen Parameter `keys` an (z.B. `keys 0:0011223344`). Der Schlüssel wird wahlweise in Form von 5 oder 13 ASCII-Zeichen bzw. durch 10 oder 26 hexadezimale Ziffern angegeben.  
Wenn das Netzwerk durch WPA abgesichert ist, müssen Sie vor der Ausführung von `iw dev connect` den Schlüssel in [`/etc/wpa\_supplicant/wpa\_supplicant.conf`](#) angeben und sicherstellen, dass `wpa_supplicant` als Hintergrunddienst läuft.
- `dev name del`  
entfernt (löscht) die Schnittstelle. Wenn die Schnittstelle später wieder verwendet werden soll, muss sie mit `interface add` neu eingerichtet werden.
- `dev name disconnect`  
beendet die Verbindung.
- `dev name info`  
gibt allgemeine Informationen über die Schnittstelle an.
- `dev name link`  
liefert Informationen zur aktiven Netzwerkverbindung bzw. *not connected*.

- `dev name scan`  
liefert detaillierte Informationen zu allen in Reichweite befindlichen Funknetzen.
- `phy phy0 interface add wlan0 type managed`  
richtet die Schnittstelle `wlan0` für das Gerät `phy0` ein. Andere Schnittstellentypen sind `monitor`, `wds`, `mesh` bzw. `mp` sowie `ibss` bzw. `adhoc`. Die neue Schnittstelle muss anschließend mit `ifconfig wlan0 up` aktiviert werden.

Zur Fehlersuche ist es häufig zweckmäßig, in einem zweiten Fenster oder einer Konsole das Kommando `iw event` auszuführen. Es liefert bis zum Ende durch `Strg+C` alle Status- und Fehlermeldungen.

## Beispiel

Die folgenden Kommandos stellen manuell eine Netzwerkverbindung zu einem Funknetz her, das nicht durch ein Passwort geschützt ist:

```
root# iw phy phy0 interface add wlan0 type managed
root# ifconfig wlan0 up
root# iw dev wlan0 connect hotel-wlan
root# dhclient wlan0
```

Bei vielen aktuellen Distributionen heißen die WLAN-Schnittstellen nicht mehr `wlan0`, `wlan1` etc., sondern `wlpns*`.

```
iwconfig [schnittstelle]
iwconfig schnittstelle [optionen]
```

Das Kommando `iwconfig` zählt zu den veralteten Linux-Wireless-Tools. Es wird dennoch bei den meisten Linux-Distributionen standardmäßig installiert. Nach Möglichkeit (d.h., wenn es für Ihren WLAN-Adapter einen modernen Treiber gibt, der die `nl80211`-Schnittstelle unterstützt) sollten Sie aber das neuere Kommando `iw` vorziehen.

In der ersten Syntaxvariante liefert `iwconfig` Informationen über alle WLAN-Schnittstellen bzw. über die angegebene Schnittstelle. In der zweiten Syntaxvariante stellt das Kommando die Parameter der WLAN-Schnittstelle ein (z.B. den Netzwerknamen, den WEP-

Schlüssel etc.). `iwconfig` ist Teil der Wireless-Tools. Die eigentliche Aktivierung der Schnittstelle erfolgt anschließend wie bei LAN-Schnittstellen durch `ip` oder `ifconfig`.

- `channel n`  
wählt den Frequenzkanal aus. Mit `channel auto` sucht der WLAN-Controller selbst einen geeigneten Kanal. Der Befehl `iwlist channel` liefert bei Bedarf eine Liste aller Kanäle
- `essid name`  
gibt den Namen des WLAN-Netzes an. Oft funktioniert auch die Einstellung `any`.
- `key schlüssel`  
stellt den aktuellen WEP-Schlüssel ein. Der Schlüssel wird normalerweise als hexadezimale Zahl ohne vorangestelltes `0x` angegeben. Mit `key [n]` wählen Sie den gerade aktuellen Schlüssel, wobei `n` zwischen 1 und 4 liegt. `ifconfig key` ist ungeeignet, um WPA-Schlüssel einzustellen! Der WPA-Schlüsselaustausch erfolgt durch das Hinterprogramm `wpa_supplicant`, das separat konfiguriert werden muss.
- `mode modus`  
bestimmt den Netzwerkmodus. Zur Auswahl stehen je nach Hardware `Managed`, `Ad-Hoc`, `Master`, `Repeater`, `Secondary`, `Monitor` oder `Auto`. Wenn Sie mit Ihrem WLAN-Controller auf einen WLAN-Router oder -Access-Point zugreifen möchten, lautet die richtige Einstellung `Managed`.
- `power modus`  
steuert den Energiesparmodus. `period n` steuert die Zeit zwischen Wake-ups. Die Zeitangabe erfolgt standardmäßig in Sekunden, mit angehängtem `m` oder `s` oder `u` in Milli- bzw. Mikro-Sekunden (also z.B. `period 20m`). `timeout n` stellt ein, nach welcher Zeit der Inaktivität die Schnittstelle in den Ruhezustand versetzt werden soll. Die Einstellung `on` bzw. `off` aktiviert bzw. deaktiviert alle Energiesparfunktionen.

## Beispiel

Das folgende Kommando deaktiviert alle Energiesparfunktionen des WLAN-Adapters. Bei unausgereiften Treibern kann das die Stabilität der Verbindung erhöhen.

```
root# iwconfig wlan0 power off  
iwlist [schnittstelle] modus
```

`iwlist` liefert für alle bzw. für die angegebene WLAN-Schnittstelle die möglichen Frequenzkanäle, die zulässigen Verschlüsselungsverfahren etc.

Die wichtigsten Schlüsselwörter für `modus` sind:

<code>channel</code>	Frequenzkanäle
<code>frequency</code>	Frequenzen
<code>key</code>	zulässige Verschlüsselungsverfahren und eingestellte Schlüssel
<code>rate</code>	unterstützte Bruttoübertragungsraten des WLAN-Controllers
<code>scan</code>	Liste der erreichbaren Netze mit ESSID, Qualität, Frequenz etc.

## J

### j verzeichnis

Das Kommando `j` aus dem Paket `autojump` hilft dabei, besonders effizient in ein anderes Verzeichnis zu wechseln. `j` ist gewissermaßen eine mitlernende Variante zum [cd](#)-Kommando. Wenn Sie beispielsweise einmal `j /etc/X11/xorg.conf.d` ausgeführt haben, reicht beim zweiten Mal `j xorg.conf.d`, also die Angabe des letzten Teils des Verzeichnispfads (vorausgesetzt, dieser ist eindeutig). Mit der Vervollständigung durch können Sie die Eingabe weiter verkürzen, z.B. in der Form `j xorg` . Wenn mehrere Verzeichnisse passen, drücken Sie einfach mehrfach . Eine Statistik aller zuletzt besuchten Verzeichnisse liefert bei Bedarf das Kommando `jumpstats`.

### john [optionen] [hashdatei]

Das Programm *John the Ripper* (Kommando- und Paketname `john`) ist ein Offline-Passwort-Cracker. Damit Sie das Programm anwenden können, müssen Ihnen die Passwörter in Form von Hashcodes vorliegen. `john` testet dann, ob selbst generierte Passwörter bzw. Passwörter aus einer vorgegebenen Liste den Hashcodes entsprechen.

`john` kommt mit den meisten gängigen Hash-Algorithmen zurecht. Im einfachsten Fall übergeben Sie an das Kommando einfach nur den Namen einer Textdatei, die zeilenweise Hash-Codes enthält. Die Zeilen der Textdatei können auch in der Form `name:hashcode:xxx` vorliegen.

`john` probiert als Passwort zuerst den Account-Namen (wenn verfügbar), dann Passwörter aus einer eingebauten Wortliste und zuletzt Passwörter, die es selbst generiert (Modus `--single`, `--wordlist` und `--incremental`). Geknackte Passwörter werden in `john/john.pot` gespeichert.

- `--format=hashname`  
gibt an, welches Hash-Verfahren `john` anwenden soll. Das Kommando unterstützt unter anderem die folgenden Formate: `afs`, `bcrypt`, `bsdicrypt`, `crypt`, `descrypt`, `dummy`, `lm` `md5crypt` und `tripcode`. Die Option ist nur erforderlich, wenn `john` das Hash-Verfahren nicht selbst erkennt.  
Eine kurze Beschreibung der Hash-Formate finden Sie auf der folgenden Webseite. Beachten Sie, dass diese Webseite auch Formate beschreibt, die nur in der inoffiziellen Jumbo-Variante enthalten sind (siehe den Abschnitt »Alternativen«). Google Chrome zeigt beim Besuch der folgenden Seite eine Warnung an, dass die Seite schädliche Inhalte enthalten kann. Meines Wissens war das zuletzt aber nicht der Fall.  
<http://pentestmonkey.net/cheat-sheet/john-the-ripper-hash-formats>
- `--incremental[:lower|lowernum|alpha|digits|alnum]`  
generiert selbst Passwörter, wobei zuerst kurze und dann immer längere Passwörter ausprobiert werden. Achtung, in diesem Modus läuft `john` endlos, es sei denn, die gesuchten Passwörter sind trivial.  
Standardmäßig werden in diesem Modus die Zeichen a--z, A--Z sowie 0--9 berücksichtigt (entspricht `alnum`, 62 Zeichen). Optional können Sie den Zeichenvorrat einschränken: `lower` entspricht a--z, `lowernum` umfasst a--z und 0--9, `alpha` entspricht a--z und A--Z, `digits` berücksichtigt nur Ziffern. Einige weitere Modi sind unter Umständen in `/etc/john/john.conf` definiert. (Unter Ubuntu funktionieren viele Modi nicht, weil die entsprechenden Zeichensatzdateien in `/usr/share/john` fehlen.)  
Schließlich können Sie mit `john --make-charset` eine eigene Zeichensatzdatei generieren, die Sie dann mit `--incremental=charsetfile` nutzen. Tipps dazu können Sie hier nachlesen: <https://security.stackexchange.com/questions/66106>

- `--restore`  
setzt die mit `[Strg]+[C]` unterbrochene Ausführung von `john` fort.
- `--show`  
zeigt bereits geknackte Passwörter an. Dazu wird die Datei `.john/john.pot` ausgewertet.
- `--wordlist dateiname`  
probiert die zeilenweise in der Datei enthaltenen Passwörter aus.

## Beispiel

Der Ausgangspunkt für dieses Beispiel ist ein Linux-Rechner mit mehreren Accounts. Das mit `john` mitgelieferte Kommando `unshadow` fasst `/etc/passwd` und `/etc/shadow` zu einer neuen Datei `hashes` zusammen. (Der Zugriff auf die shadow-Datei erfordert root-Rechte.) Die Hash-Codes in der resultierenden Datei sind hier aus Platzgründen verkürzt wiedergegeben:

```
root# unshadow /etc/passwd /etc/shadow > hashes
root# chown user hashes
user$ cat hashes
...
peter:$6$U.zGFB1F$LdNTE...:1001:1001::/home/peter:/bin/bash
maria:$6$gSJg6.d8$mN.en...:1002:1002::/home/maria:/bin/bash
hans:$6$UiunuQqJY$iD59.N...:1003:1003::/home/hans:/bin/bash
```

`john` findet zwei besonders unsichere Passwörter innerhalb von Sekunden. `peter` hat als Passwort seinen eigenen Namen verwendet, `hans` das beliebte Passwort 123456. Das Passwort von `Maria` ist aber nicht auf Anhieb zu knacken, weswegen der Vorgang nach einer Weile mit `[Strg]+[C]` gestoppt wird:

```
user$ john hashes
Loaded 3 password hashes with 3 different salts
Press 'q' or Ctrl-C to abort, almost any other key for status
peter          (peter)
123456         (hans)
<Strg>+<C>
```

Von GitHub können Sie nun mit [git](#) Listen beliebter Passwörter herunterladen. (Achtung, der Platzbedarf für die Listen beträgt rund 750 MiB!) Eines dieser Wörterbücher enthält tatsächlich auch das Passwort von `Maria`! Es lautet `secret`:

```
user$ git clone https://github.com/danielmiessler/SecLists.git
user$ john --wordlist=SecLists/Passwords/Common-Credentials/ \
           10-million-password-list-top-10000.txt hashes
Loaded 3 password hashes with 3 different salts
Remaining 1 password hash with 1 different salt
secret          (maria)
```

## Alternativen

Auf GitHub gibt es die stark erweiterte, von der Community gepflegte »Jumbo«-Variante von john:

<https://github.com/openwall/john>

Wenn Sie das Cracken von Hash-Codes mit GPU-Unterstützung enorm beschleunigen möchten, sollten Sie den Einsatz von hashcat in Erwägung ziehen. Die größte Hürde ist in diesem Fall die Installation geeigneter GPU-Treiber. Um Netzwerkdienste im Hinblick auf unsichere Passwörter zu überprüfen, verwenden Sie den Online-Passwort-Cracker [hydra](#).

**journalctl** [optionen] [suchausdruck]

Viele Distributionen, die systemd als Init-System verwenden, ersetzen oder ergänzen das traditionelle Syslog-System durch die systemd-Implementierung, das sogenannte *Journal*. Die Logging-Dateien werden dabei in einem speziellen binären Format gespeichert und können nur noch mit dem hier beschriebenen journalctl-Kommando ausgelesen werden.

Ein oder mehrere Suchausdrücke können als Pfad zu einem Programm oder in der Syntax `field=value` formuliert werden. `journalctl` zeigt dann nur Logging-Einträge, auf die alle Suchausdrücke zutreffen. Suchausdrücke mit logischem Oder können Sie mit `+` formulieren, also `field1=value1 + field2=value2`. Für `field` sind unter anderem die Schlüsselwörter `MESSAGE`, `PRIORITY`, `ERRNO` sowie `_PID`, `_UID`, `_GID` oder `_SELINUX_CONTEXT` zulässig. Eine Referenz weiterer Suchparameter sowie eine genaue Beschreibung ihrer Bedeutungen können Sie mit `man systemd.journal-fields` nachlesen.

Bei einem Aufruf ohne weitere Parameter liefert `journalctl` eine zumeist fast endlose Liste aller protokollierten Meldungen. Mit Optionen können Sie die Ausgabe filtern:

- `-b`  
zeigt nur die Nachrichten seit dem letzten Neustart des Rechners.
- `--disk-usage`  
zeigt an, wie viel Platz die Dateien des Journals in `/var/log` beanspruchen.
- `-e`  
springt sofort an das Ende der anzuzeigenden Nachrichten.
- `-f`  
startet `journalctl` im Dauerbetrieb, wobei ständig die gerade eintreffenden Nachrichten angezeigt werden. `Strg + C` beendet das Kommando.
- `-k`  
zeigt nur Kernelnachrichten.
- `-n n`  
zeigt nur die letzten *n* Zeilen.
- `--no-pager`  
schreibt die Ausgaben direkt auf die Standardausgabe, anstatt einen Pager (normalerweise das Kommando [less](#)) zur Anzeige zu verwenden.
- `-p n`  
zeigt nur Nachrichten in einer bestimmten Prioritätsstufe. Der Zahlenbereich reicht von 0 bis 7 für emerg, alert, crit, err, warning, notice, [info](#) und debug. -p 2 bewirkt, dass nur Nachrichten der Stufen 0, 1 und 2 berücksichtigt werden.
- `-r`  
zeigt die neuesten Nachrichten zuerst.
- `--since 2020-12-31 19:30:00`  
zeigt nur Nachrichten an, die nach dem angegebenen Zeitpunkt

protokolliert wurden.

- `-t name`  
zeigt nur Nachrichten für das angegebene Syslog-Stichwort (Tag, wie bei [logger -t](#)).
- `-u name`  
zeigt nur Nachrichten für den angegebenen systemd-Dienst (Unit, z.B. avahi-daemon).
- `--user`  
zeigt Nachrichten aus dem User-Journal statt aus dem System-Journal an. Das User-Journal wird bei modernen Linux-Distributionen verwendet, um Meldungen des Grafiksystems, von Gnome bzw. von anderen Desktop-Systemen sowie der darin laufenden Programme zu protokollieren.
- `--until 2020-12-31 19:30:00`  
zeigt nur Nachrichten, die bis zum angegebenen Zeitpunkt protokolliert wurden.

## Beispiele

Das folgende Kommando zeigt alle Meldungen des Open-SSH-Servers in umgekehrter Reihenfolge, d.h., die neuesten Nachrichten zuerst:

```
user$ journalctl -u sshd -r
```

Wenn Sie die Logging-Nachrichten live verfolgen möchten, rufen Sie `journalctl` wie folgt auf:

```
user$ journalctl -u sshd -f
```

Das dritte Beispiel zeigt alle Meldungen an, in denen die IP-Adresse 10.0.0.2 vorkommt:

```
user$ journalctl | grep 10.0.0.2
```

# K

## **kbdrate** [optionen]

kbdrate stellt die Verzögerungszeit und die Geschwindigkeit von Tastenwiederholungen der Tastatur ein. Ohne Optionen zeigt das Kommando die aktuell gültigen Einstellungen an.

- **-d n** bzw. **--delay=n**  
gibt an, nach wie vielen Millisekunden die Tastenwiederholungen einsetzen. Zulässige Werte sind 250, 500, 750 und 1000.
- **-r n** bzw. **--rate=n**  
steuert, wie viele Zeichen pro Sekunde die Tastatur bei einem längeren Tastendruck weiterleitet. *n* darf eine Fließkommazahl sein, allerdings ist nicht jeder Wert erlaubt (siehe `man kbdrate`). Das Kommando entscheidet sich automatisch für den nächstbesten zulässigen Wert.

## **kexec** [optionen]

Das Kommando kexec aus dem Paket `kexec-tools` ermöglicht es, einen neuen Kernel ohne einen Neustart des Rechners zu aktivieren. Dabei müssen alle Prozesse sowie das Init-System komplett heruntergefahren werden. Im Vergleich zu einem echten Reboot ist ein Kernel-Neustart mit kexec dennoch um ein paar Sekunden schneller: Insbesondere entfallen die BIOS/EFI-Initialisierung und das Durchlaufen des GRUB-Prozesses. Allerdings kann es unter Umständen Probleme bei der Neuinitialisierung diverser Hardware-Komponenten geben.

kexec wird in zwei Schritten verwendet: Zuerst bereiten Sie den Kernel-Neustart mit `kexec -l` vor, danach führen Sie den Kernel-Austausch mit `kexec -e` oder mit `systemctl isolate kexec` tatsächlich durch.

- **--append=kernelparameter**  
übergibt die Parameter zusätzlich zu `--command-line` bzw. `--`

`reuse-cmdline` an den Kernel.

- `--command-line=kernelparameter`  
übergibt die Parameter an den Kernel.
- `-e`  
startet den zuvor eingerichteten Kernel neu (*exec*).
- `--initrd=initrddatei`  
gibt den Ort der Initrd-Datei an.
- `-l kerneldatei`  
gibt den Ort des neu zu ladenden Kernels an (*load*).
- `--reuse-cmdline`  
übernimmt die beim letzten Kernel-Start verwendeten Parameter.

## Beispiel

Das folgende Kommando bereitet den Kernel-Neustart vor:

```
root# kexec -l /boot/vmlinuz-5.6.7-300 \
           --initrd=/boot/initrd.img-5.6.7-300 --reuse-cmdline
```

Im zweiten Schritt aktivieren Sie den neuen Kernel – entweder unmittelbar mit `kexec -e` oder nachdem Sie via `systemd` alle laufenden Dienste heruntergefahren haben. Auf einem Server ist die zweite Variante unbedingt vorzuziehen! Nur sie stellt sicher, dass z.B. ein Datenbank-Server alle offenen Transaktionen zu Ende führen und alle Dateien ordnungsgemäß schließen kann.

```
root# kexec -e          (unmittelbarer Neustart)
root# systemctl isolate kexec (zuerst Dienste herunterfahren, dann Neustart)
kill [-s signal] prozessnr
```

Das bash-Kommando `kill` versendet Signale an einen laufenden Prozess. Wenn `kill` ohne die `-s`-Option verwendet wird, wird standardmäßig das SIGTERM-Signal (15) gesendet, um den Prozess zu beenden (zu *killen*, daher auch der Name des Kommandos). Bei besonders hartnäckigen Fällen hilft `-9` bzw. `-s SIGKILL` oder `-KILL`. Der Prozess hat dann allerdings keine Chance, noch irgendwelche Aufräumarbeiten zu erledigen.

`kill` kann aber auch zum Versenden harmloserer Signale verwendet werden. Recht häufig wird `-1` bzw. `-s SIGHUP` bzw. `-HUP` verwendet, um einen Dämon dazu aufzufordern, seine Konfigurationsdateien neu einzulesen. Auf diese Weise können Sie bei manchen Programmen eine neue Konfiguration aktivieren, ohne den Dämon vollständig stoppen und neu starten zu müssen.

Die erforderliche Prozessnummer (PID) wird am einfachsten mit dem Kommando [ps](#) ermittelt. Unter X gibt es mit `xkill` eine bequeme Variante zu `kill`: Das Programm, das beendet werden soll, kann damit einfach per Maus »abgeschossen« werden.

**killall** [-signal] prozessname

`killall` funktioniert beinahe wie das [kill](#)-Kommando. Der Unterschied besteht darin, dass nicht die Prozessnummer (PID), sondern der Name des Prozesses angegeben wird. Wenn es mehrere Prozesse dieses Namens gibt, erhalten alle das angegebene Signal (standardmäßig wieder SIGTERM). Das gewünschte Signal wird entweder als Nummer `-n` oder mit einem Namen wie `-HUP` angegeben. Eine Liste aller Signallamen erhalten Sie mit `killall -l`.

## Beispiel

Das folgende Beispiel beendet alle laufenden Firefox-Instanzen des aktuellen Benutzers. Wird das `killall`-Kommando von `root` ausgeführt, beendet es alle laufenden Firefox-Prozesse *aller* Benutzer.

```
user$ killall firefox
```

**kpartx** [diskdevice]

Das Low-Level-Kommando `kpartx` aus dem gleichnamigen Paket ermittelt alle Partitionen des angegebenen Datenträgers und erzeugt die dazugehörigen Device-Dateien. Normalerweise werden die Device-Dateien durch das `udev`-System automatisch erzeugt, sobald ein neuer Datenträger erkannt wird – z.B. beim Anschließen einer USB-Festplatte. `kpartx` ist primär zur Bearbeitung von

virtuellen Datenträgern bzw. von Image-Dateien virtueller Maschinen gedacht.

- **-a**  
erzeugt neue Device-Dateien für den angegebenen Datenträger (*add*).
- **-d**  
entfernt die Device-Dateien für den Datenträger (*delete*).
- **-l**  
liest die Partitionen des Datenträgers, erzeugt aber keine Device-Dateien.
- **-u**  
aktualisiert die Device-Dateien für einen veränderten Datenträger (*update*).
- **-v**  
gibt Informationen über die durchgeführten Aktionen aus.

## Beispiel

Das folgende Kommando verbindet alle in der RAW-Image-Datei enthaltenen Partitionen mit Loop-Devices:

```
root# kpartx -av image.raw
add map loop0p1 (252:12): 0 1024000 linear /dev/loop0 2048
add map loop0p2 (252:13): 0 19945472 linear /dev/loop0 1026048
```

Die ganze virtuelle Festplatte kann anschließend über das Device */dev/mapper/loop0* angesprochen werden.

```
kvm [optionen] [imagedatei]
qemu-kvm [optionen] [imagedatei]
```

**kvm** (Debian, Ubuntu) bzw. **qemu-kvm** (Fedora, Red Hat) sind winzige Scripts, die im Emulator **qemu** eine virtuelle Maschine mit KVM-Unterstützung ausführen. Unter CentOS/RHEL befindet sich das Kommando **qemu-kvm** im Verzeichnis */usr/libexec*.

Im einfachsten Fall übergeben Sie an **kvm** bzw. **qemu-kvm** nur den Namen einer Image-Datei. KVM emuliert dann eine virtuelle Maschine mit Standardeinstellungen, unter anderem mit einer IDE-Festplatte. Wünschen Sie andere Einstellungen oder mehrere

Datenträger, verwenden Sie dazu die Optionen `-drive` oder `-hda`, `-hdb` etc. In diesem Fall kann die direkte Angabe der Image-Datei im KVM-Kommando entfallen.

- `-accel kvm`

aktiviert KVM. Bei den meisten Distributionen ist diese Option automatisch aktiv, aber nicht bei allen! Ohne diese Option wird die virtuelle Maschine ohne CPU-Unterstützung emuliert, was ineffizient und langsam ist. Der zugrunde liegende Emulator `qemu` kommt auch mit anderen Beschleunigungssystemen zurecht, deswegen unterstützt die Option auch die Einstellungen `xen`, `hax` oder `tcg`.

- `-boot order=xxx,once=xxx,menu=on/off`

gibt an, in welcher Reihenfolge die Datenträger für den Bootprozess berücksichtigt werden sollen. Dabei ist `xxx` eine Buchstabenfolge, die die Reihenfolge der Datenträger ausdrückt (z.B. `adc`: zuerst das Diskettenlaufwerk, dann das CD/DVD-Laufwerk, danach die erste Festplatte). Die Buchstaben `a` bis `d` entsprechen den DOS/Windows-Laufwerksbuchstaben.

`once=xxx` gibt die Bootreihenfolge nur für den *ersten* Bootvorgang an. Wenn die virtuelle Maschine also beispielsweise beim ersten Versuch vom CD/DVD-Laufwerk booten soll, bei weiteren Neustarts aber von der Festplatte, geben Sie `-boot order=c,once=d` oder schlicht `-boot once=d` an.

`menu=on` zeigt zum Beginn des Bootmenüs die Meldung *Press F12 for boot menu* an. Mit `F12` kann dann der Bootdatenträger interaktiv ausgewählt werden.

- `-cdrom iso-datei`

verwendet die angegebene ISO-Datei als Datenquelle für das virtuelle CD/DVD-Laufwerk. Die Option entspricht `-drive file=iso-datei,index=2,media=cdrom`.

- **-cpu host**  
gibt alle Eigenschaften der Host-CPU an den Gast weiter. Standardmäßig ist das nicht der Fall: Es wird nur ein Subset weitergegeben, um die Kompatibilität virtueller Maschinen zwischen unterschiedlichen CPUs zu maximieren. Wenn Sie auf einem 64-Bit-Host arbeiten, dem Gast aber nur eine 32-Bit-CPU zur Verfügung stellen möchten, verwenden Sie `-cpu kvm32`.
- **-device gerät**  
fügt der virtuellen Maschine ein zusätzliches Gerät hinzu. Eine Liste aller unterstützten Geräte liefert `kvm -device ?`. Beim Gerätenamen wird zwischen Groß- und Kleinschreibung unterschieden! Die für ein bestimmtes Gerät verfügbaren Optionen ermitteln Sie mit `kvm -device gerät,?`, also z.B. `kvm -device isa-serial,?`. Beachten Sie, dass Sie die meisten Komponenten einer virtuellen Maschine auf zwei Arten definieren können: Mit der hier beschriebenen, sehr universellen Option `-device` oder mit gerätespezifischen Optionen (z.B. `-drive`, `-soundhw`, `-usb-device` oder `-vga`).
- **-drive details**  
definiert die Eigenschaften einer virtuellen Festplatte. Die Detailparameter werden nur durch Kommata voneinander getrennt (ohne Leerzeichen!). Die Option kann mehrfach verwendet werden, wenn die virtuelle Maschine mit mehreren Datenträgern ausgestattet werden soll.  
`boot=on/off` gibt an, ob der Datenträger beim Booten berücksichtigt werden soll. Bei IDE- und SCSI-Laufwerken gilt automatisch `boot=on`. Damit KVM auch von einem virtio-Laufwerk booten kann, muss `boot=on` explizit angegeben werden.  
`cache=writethrough/writeback/none` gibt an, ob und wie Schreibzugriffe zwischengespeichert werden. Standardmäßig gilt `writethrough`: Im Gastsystem erscheint ein Schreibzugriff erst

dann als abgeschlossen, wenn das Hostsystem den Speichervorgang quittiert hat.

`file=fname` gibt den Dateinamen der Image- oder ISO-Datei bzw. den Device-Namen eines Logical Volumes an.

`if=ide/scsi/virtio` gibt an, über welche Schnittstelle die virtuelle Maschine den Datenträger ansprechen soll (standardmäßig `ide`). Bei Linux-Gästen ist `virtio` effizienter.

`index=n` bestimmt die Nummerierung der Datenträger einer Schnittstelle. Der Parameter ist nur erforderlich, wenn die Datenträger nicht der Reihe nach angegeben werden.

`media=disk/cdrom` gibt an, ob es sich um eine Festplatte (gilt standardmäßig) oder um ein CD/DVD-Laufwerk handeln soll.

- `-hda/-hdb/-hdc/-hdd details`  
gibt eine virtuelle IDE-Festplatte an.  
`-hda fname` entspricht `-drive file=fname, index=0, media=disk`,  
`-hdb fname` entspricht `-drive file=fname, index=1, media=disk` etc.
- `-k sprachkürzel`  
verwendet das angegebene Tastaturlayout. Zulässige Kürzel sind unter anderem `de` (Deutsch) und `en-us` (US-Englisch). Die Option ist nur erforderlich, wenn die virtuelle Maschine durch einen externen VNC-Client bedient wird. Die VNC-Clients des Virtual Machine Managers bzw. des Kommandos [`virt-viewer`](#) erkennen die Tastatureinstellung selbstständig.
- `-localtime`  
initialisiert die virtuelle CMOS-Uhr des Gastsystems mit der lokalen Zeit (statt standardmäßig mit der UTC-Zeit).
- `-m n`  
stellt die Speichergröße der virtuellen Maschine ein (in MiB). Die Defaulteinstellung variiert je nach Distribution.
- `-machine name[, para1=wert1, para2=wert2 ...]`  
gibt an, welche Hardware emuliert werden soll. Eine Liste der zulässigen Typennamen liefert `-machine help`. Mit den

Parametern können abweichend von den Grundeinstellungen des jeweiligen Typs Zusatzeigenschaften aktiviert bzw. deaktiviert werden.

- **-monitor device**  
leitet die Ein- und Ausgabe des QEMU-Monitors in das angegebene Device um. Wenn Sie den Monitor über die aktuelle Konsole bedienen möchten, geben Sie als Device `stdio` an. Mit `pty` legt `kvm` beim Start ein neues Pseudo-TTY-Device an und verwendet es für die Kommunikation.
- **-net nic, details**  
konfiguriert einen virtuellen Netzwerkadapter. Wenn diese Option nicht angegeben wird, emuliert KVM standardmäßig eine RTL-8139-kompatible Netzwerkkarte.  
`model=ne2k_pci/i82551/i82557b/i82559er/rtl8139/e1000/pcnet/virtio` legt fest, welchen Netzwerkadapter KVM emulieren soll. Für Linux-Gäste erzielen Sie mit `model=virtio` die besten Resultate.  
`macaddr=52:54:00:nn:nn:nn` gibt die gewünschte MAC-Adresse an.
- **-net user, details**  
verwendet Usermode-Networking (gilt standardmäßig): Das Gastsystem kann zwar dank NAT und Masquerading die Internetverbindung des Hostsystems nutzen, es ist aber keine direkte Netzwerkverbindung zwischen Gast und Host möglich.
- **-rtc base=utc/localtime**  
gibt an, welche Startzeit die Uhr der virtuellen Maschine haben soll. `utc` ist die korrekte Einstellung für Linux-Gäste, während `localtime` für Windows-Gäste geeignet ist. Standardmäßig ist die Uhr immer synchron mit jener des Hostrechners. Wenn Sie das nicht wünschen, können Sie den zusätzlichen Parameter `clock=vm` angeben.
- **-smp n bzw. -smp cores=c, threads=t, sockets=s**  
gibt in der Kurzform an, wie viele CPUs bzw. Cores der virtuellen Maschine zugewiesen werden sollen (standardmäßig nur ein

Core). Bei Hostsystemen mit mehreren CPUs gibt `c` an, wie viele Cores pro CPU genutzt werden sollen. `t` gibt die gewünschte Anzahl der Threads pro Core an; sinnvoll ist hier zumeist der Wert 2 bei Intel-CPUs, die Hyperthreading unterstützen. `s` legt schließlich fest, wie viele CPUs (Sockets) verwendet werden sollen.  $c*t*s$  ergibt die Anzahl der CPUs, die die virtuelle Maschine sieht.

- `-soundhw ac97/es1370/hda/sb16/all`  
fügt der virtuellen Maschine eines der angegebenen Audio-Geräte hinzu (oder alle, wenn Sie `all` verwenden). `ac97` steht für Intel 82801AA AC97, `es1370` für Ensoniq AudioPCI ES1370, `hda` für Intel High Definition Audio und `sb16` für Creative Sound Blaster 16.
- `-spice port=n[,optionen]`  
aktiviert das Grafiksystem Spice. Dazu muss zumindest der gewünschte Port angegeben werden. Mit `password=xxxx` kann die Verbindung zudem durch ein Passwort abgesichert werden. Wenn kein Passwort verwendet werden soll, muss explizit die Option `disable-ticketing` angegeben werden.
- `-usb`  
aktiviert den USB-Treiber, wenn dies nicht standardmäßig der Fall ist.
- `-usbdevice mouse/tablet/disk/host...`  
fügt der virtuellen Maschine ein USB-Gerät hinzu. Am häufigsten werden Sie die Option `-usbdevice tablet` benötigen. Sie ersetzt die standardmäßig emulierte PS/2-Maus durch ein virtuelles USB-Zeigergerät, das absolute Koordinaten versteht und so die Synchronisation der Mausposition des Gasts mit dem VNC- oder Spice-Client ermöglicht.
  - `usbdevice disk` ermöglicht es, eine Image-Datei des Hosts so an den Gast weiterzugeben, dass dieser einen USB-Datenträger sieht.
  - `usbdevice host:bus.addr` bzw. `-usbdevice host:vendorid:productid` leitet ein USB-Gerät des Hosts an den

Gast weiter. Das USB-Gerät darf vom Host nicht genutzt werden. Die Bus- und Device-Nummer bzw. die Vendor- und Produkt-IDs ermitteln Sie auf dem Hostrechner am einfachsten mit [lsusb](#).

- `-vga cirrus/qxl/std/virtio/vmware`  
gibt den gewünschten Typ der virtuellen Grafikkarte an. Standardmäßig emuliert KVM eine Cirrus-kompatible Grafikkarte mit einer Auflösung von bis zu 1024×768 Pixeln. Dieses Grafiksystem wird von nahezu allen Gastsystemen korrekt erkannt und in einer akzeptablen Geschwindigkeit ausgeführt. Die Grafikkarte `qxl` kann nur in Kombination mit `-spice` eingesetzt werden.
- `-vnc n.n.n.n:n[, optionen]`  
führt einen VNC-Server aus, über den Clients den Inhalt der virtuellen Grafikkarte darstellen können. Mit `n.n.n.n` geben Sie an, von welcher IP-Adresse aus der Verbindungsauftbau zum VNC-Server erfolgen darf (z.B. 127.0.0.1 für Verbindungen von `localhost`). `:n` gibt die Display-Nummer an. Der Port für den VNC-Server ergibt sich aus `n+5900`. Wenn Sie nur die Display-Nummer ohne IP-Adresse angeben (also z.B. `:0`), kann der Verbindungsauftbau durch jeden beliebigen Rechner erfolgen.  
`kvm` bzw. `qemu-kvm` wird nur in Ausnahmefällen direkt ausgeführt. Üblicher ist es, zum Einrichten und Starten von virtuellen Maschinen auf das Kommando [`virsh`](#) aus dem `libvirt`-Paket zurückzugreifen bzw. ein Oberfläche wie `virt-manager` oder ein Cloud-Framework wie OpenStack zu verwenden.

## Beispiel

Im folgenden Beispiel wird zuerst eine Image-Datei mit einer maximalen Größe von 10 GiB erzeugt, die dann als virtuelle Festplatte zur Installation eines Ubuntu-Servers von einer ISO-Datei genutzt wird:

```
root# qemu-img create -f qcow2 disk.img 10G
root# kvm -accel kvm -m 1024 -smp 2 -boot once=d -cdrom ubuntu-server.iso \
      -drive file=disk.img,if=virtio,format=qcow2 \
```

```
-net user -net nic,macaddr=52:54:00:12:e4:4e,model=virtio \
-vga cirrus -vnc 127.0.0.1:0 -k de -usb -usbdevice tablet
```

Um die virtuelle Maschine zu bedienen, müssen Sie nun noch einen VNC-Client starten, z.B. das Programm vncviewer:

```
user$ vncviewer localhost:0
```

## L

**l2ping** [optionen] bluetoothmac

l2ping aus dem bluez-Paket sendet L2CAP-Echo-Anfragen an das durch seine MAC-Adresse angegebene Bluetooth-Gerät. Auf diese Weise kann getestet werden, ob prinzipiell eine Verbindung zwischen dem lokalen Bluetooth-Adapter und dem externen Gerät möglich ist. Die MAC-Adresse externer Bluetooth-Geräte ermitteln Sie mit [hcitool scan](#). l2ping muss mit root-Rechten ausgeführt werden.

**lame** [optionen] in out.mp3

Das Acronym LAME steht für *LAME Ain't an MP3 Encoder*. Tatsächlich dient das Kommando lame aber sehr wohl dazu, WAV-Dateien in komprimierte Audiodateien umzuwandeln, die zum MP3-Format kompatibel sind. Alternativen zu lame sind toolame oder twolame, die beide Dateien im Format MPEG-1 Layer 2 (also MP2) erzeugen. Auch derartige Dateien werden von den meisten MP3-Playern problemlos abgespielt.

- -r  
die Quelldatei liegt im RAW-Format vor (nicht als WAV-Datei).
- -s 8/11.025/12/16/22.05/24/32/44.1/48  
gibt die Sampling-Frequenz innerhalb der RAW-Datei an. Die Option ist bei WAV-Dateien nicht erforderlich.

### Beispiel

Das folgende Kommando erzeugt die MP3-kompatible Audiodatei *title.mp3*:

```
user$ lame title.wav title.mp3
```

**last** [optionen]

last liefert eine Liste der Benutzer, die zuletzt auf diesem Rechner eingeloggt waren. Das Kommando wertet die Datei */var/log/wtmp* aus.

- **-i**  
zeigt bei SSH-Logins die IP-Adresse anstelle des Hostnamens des entfernten Rechners an.
- **-n oder -n n**  
steuert, wie viele Einträge angezeigt werden sollen, z.B. 100 mit **-100**.
- **-t YYYYMMDDhhmmss**  
gibt an, wer zum angegebenen Zeitpunkt eingeloggt war.

## Beispiel

Auf dem Testrechner hat zuletzt nur der Benutzer **kofler** gearbeitet:

```
user$ last
kofler    pts/0    62-47-230-2.adsl Tue Sep 24 13:51  still logged in
kofler    pts/0    212-183-46-83.ad Mon Sep 23 19:50 - 20:22  (00:32)
kofler    pts/1    91-115-236-11.ad Fri Sep 13 15:39 - 17:51  (02:11)
kofler    pts/0    91-115-236-11.ad Fri Sep 13 15:31 - 17:51  (02:20)
```

## **lastb** [optionen]

**lastb** zeigt an, welche Login-Versuche zuletzt gescheitert sind. Das Kommando wird mit den gleichen Optionen wie [last](#) gesteuert. Es darf nur von **root** ausgeführt werden. Es wertet die Datei **/var/log/btmp** aus.

## **ldconfig**

**ldconfig** aktualisiert die Links zu allen Bibliotheken und erstellt die Cache-Datei **/etc/ld.so.cache**, die bei der effizienten Suche nach Bibliotheken hilft. **ldconfig** wertet die Konfigurationsdatei **/etc/ld.so.conf** aus. Das Kommando muss nach der manuellen Installation von Bibliotheken ausgeführt werden.

## **ldd** programm

**ldd** liefert eine Liste aller Bibliotheken, die zur Ausführung des angegebenen Programms erforderlich sind. Mit dem Kommando stellen Sie auch fest, ob auf dem Rechner alle notwendigen Bibliotheken verfügbar sind.

## Beispiel

Der Gnome-Editor greift unter Fedora 32 auf rund 80 Bibliotheken zurück! Die folgenden Zeilen zeigen die ersten drei davon in alphabetischer Ordnung:

```
root# ldd /usr/bin/gedit | sort
/lib64/ld-linux-x86-64.so.2 (0x00007fc2b68ca000)
libamtk-5.so.0 => /lib64/libamtk-5.so.0 (0x00007fc2b5707000)
libatk-1.0.so.0 => /lib64/libatk-1.0.so.0 (0x00007fc2b5911000)
...
```

### **less** [optionen] datei

less zeigt die angegebene Textdatei seitenweise an. Das Kommando wird häufig als Filter verwendet, z.B. `ls -l | less` zur Seitenweisen Anzeige einer langen Dateiliste.

- `-m`  
zeigt in der Statuszeile die aktuelle Textposition in Prozent an.
- `-M`  
zeigt in der Statuszeile den Dateinamen und die Textposition in Zeilen an.
- `-p suchtext`  
zeigt die erste Zeile an, in der der zu suchende Text gefunden wurde.
- `-r`  
zeigt Formatier- und Farbkontrollcodes an (*raw*). Die Option ist in seltenen Fällen erforderlich, um eine Datei korrekt anzuzeigen, die solche Kontrollcodes enthält.
- `-s`  
reduziert mehrere Leerzeilen zu einer Zeile.

Während less läuft, zeigt `H` einen Hilfetext an. Die Cursortasten ermöglichen die Navigation im Text. Die Leertaste scrollt eine Seite vor, `B` eine Seite zurück. `<` und `>` springen zum Anfang bzw. Ende des Texts. `/` ermöglicht die Eingabe eines Suchtextes.

`Q` beendet less.

### **letsencrypt** [optionen]

In der Vergangenheit gab es das Kommando `letsencrypt`, um Let's-Encrypt-Zertifikate anzufordern. Dieses Kommando wird aber nicht mehr gewartet. Verwenden Sie [acme.sh](#) oder [certbot!](#)

## **lftp [optionen] [site]**

`lftp` ist ein interaktiver FTP-Client. `lftp` kann aber auch in Scripts eingesetzt werden und durch Kommandos gesteuert werden.

- `-c "kommando"`  
führt die in einer Zeichenkette angegebenen und durch Strichpunkte getrennten FTP-Kommandos aus (z.B. `lftp -c "open -u user,passw server; mirror -R verz"`).
- `-f datei`  
liest die auszuführenden Kommandos zeilenweise aus einer Datei.

`lftp`-Login-Einstellungen können in der Datei `.netrc` zeilenweise für diverse Server gespeichert werden. Einträge in dieser Datei sehen wie dieses Muster aus:

```
machine backup.hostname.com    login u123456    password x234b0CT2xCb
```

## **Beispiel**

Das folgende Kommando lädt eine Datei auf einen FTP-Server hoch:

```
root# lftp -c "open -u username,password backupserver; put datei"
```

Um statt einer Datei ein ganzes Verzeichnis zum Backup-Server zu übertragen, verwenden Sie das Kommando `mirror -R`. `mirror` kopiert normalerweise Verzeichnisse vom FTP-Server auf den lokalen Rechner. `-R` dreht die Übertragungsrichtung um. Auch hierzu ein Beispiel:

```
root# lftp -c "open -u usern,password bserver; mirror -R verzeichnis"
```

## **ll [optionen] dateien**

`ll` ist bei vielen Distributionen ein vordefinierter Alias, der das Kommando [`ls`](#) mit der Option `-l` und eventuell mit weiteren Optionen ausführt.

## **ln [optionen] quelle [ziel]**

## **ln [optionen] dateien zielverzeichnis**

`ln` richtet feste oder symbolische Links auf Dateien und Verzeichnisse ein. Die gleiche Funktionalität wie `ln` hat auch das Kommando [cp](#), wenn dort die Optionen `-l` oder `-s` angegeben werden.

- **-b bzw. --backup**  
benennt bereits vorhandene gleichnamige Dateien in Backup-Dateien (Name plus ~-Zeichen) um, anstatt sie zu überschreiben.
- **-d bzw. --directory**  
erzeugt einen festen Link für ein Verzeichnis. Diese Operation ist nur root gestattet. Alle anderen Anwender können symbolische Links auf Verzeichnisse erzeugen.
- **-s bzw. --symbolic**  
erzeugt symbolische Links. (Ohne diese Option liefert `ln` feste Links.)

## Beispiel

Das folgende Kommando erzeugt den symbolischen Link `xyz` auf die bereits vorhandene Datei `abc`:

```
user$ ln -s abc xyz  
loadkeys [optionen] filename
```

Das Low-Level-Kommando `loadkeys` lädt eine Tastaturtabelle für den Textmodus. Die erforderlichen Einstellungsdateien befinden sich üblicherweise im Verzeichnis `/lib/kbd/keymaps`. Wenn Sie einen Dateinamen ohne Pfad angeben, versucht das Kommando selbst, eine geeignete Datei mit der Kennung `.map.gz` zu finden.

Die so veränderten Tastatureinstellungen gelten also nur, wenn Sie in einer Textkonsole arbeiten. Die Einstellungen für den Grafikmodus müssen hingegen im Rahmen der X-Konfiguration verändert werden; bei Distributionen mit `systemd` können Sie dazu auch das Kommando [`localectl`](#) verwenden.

- **-d**  
lädt die Default-Tastaturtabelle, in der Regel die Datei `/lib/kbd/keymaps/defkeymap.map`.

## Beispiel

Das folgende Kommando aktiviert in den Textkonsolen das deutsche Tastaturlayout:

```
root# loadkeys de-latin1  
Lade /lib/kbd/keymaps/i386/qwertz/de.map.gz  
local var [=wert]
```

Das bash-Kommando `local` definiert in Funktionen innerhalb von Shell-Scripts eine lokale Variable. Das Kommando kann nur in selbst definierten Funktionen verwendet werden (siehe [function](#)). Vor und nach dem Gleichheitszeichen dürfen keine Leerzeichen angegeben werden.

```
localectl [optionen] [kommando]
```

Das Kommando `localectl` dient bei Distributionen mit dem Init-System `systemd` zur Steuerung der Sprach- und Tastatureinstellungen. Es verändert die Dateien [/etc/locale.conf](#), [/etc/vconsole.conf](#) sowie [/etc/X11/xorg.conf](#).

## Optionen

- `--no-convert`  
verändert nur die Tastatur für die Konsole (Kommando `set-keymap`) bzw. für den Grafikmodus (Kommando `set-x11-keymap`). Standardmäßig versucht `localectl`, die Tastatureinstellungen sowohl für den Text- als auch für den Grafikmodus zugleich zu verändern. Die betreffenden Dateien haben aber eine unterschiedliche Syntax, weswegen die Konvertierung der Parameter mitunter scheitert.
- `--no-pager`  
führt Ausgaben direkt auf der Standardausgabe aus, anstatt [`less`](#) zu verwenden.

## Kommandos

- `list-keymaps`  
ermittelt eine Liste aller möglichen Tastatur-Layouts für den Textmodus.

- `list-locales`  
ermittelt eine Liste aller möglichen Spracheinstellungen.
- `list-x11-keymap-models`  
`list-x11-keymap-layouts`  
`list-x11-keymap-variants [layout]`  
`list-x11-keymap-options`  
ermittelt die zulässigen Parameter für die Einstellung des Tastaturlayouts im Grafiksystem X.
- `set-keymap name`  
aktiviert das angegebene Tastaturlayout.
- `set-locale name`  
aktiviert die angegebene Spracheinstellung.
- `set-x11-keymap layout [modell variante optionen]`  
aktiviert das angegebene Tastaturlayout für den Grafikmodus.
- `status`  
zeigt die aktuellen Sprach- und Tastatureinstellungen an.

## Beispiel

Die beiden folgenden Kommandos stellen die Sprache Deutsch, den Zeichensatz UTF8 sowie ein deutsches Tastaturlayout ein, wobei die Besonderheiten einer Apple-Tastatur berücksichtigt werden:

```
root# localectl set-locale de_DE.UTF-8
root# localectl set-x11-keymap de de mac grp:alt_shift_toggle
```

### **locate muster**

locate ermöglicht eine besonders schnelle Suche nach Dateien. Es durchsucht eine zumeist einmal täglich aktualisierte Dateidatenbank, in der das angegebene Muster im vollständigen Dateinamen (inklusive Pfad) vorkommt. Dateien, die nach der letzten Datenbankaktualisierung entstanden sind oder verändert wurden, können aber natürlich nicht gefunden werden.

Zumeist müssen locate und das zugrunde liegende [updatedb](#)-System extra installiert (Paket `mlocate`) und oft auch konfiguriert werden.

### **logger [optionen] nachricht**

`logger` protokolliert eine Nachricht mit dem Syslog-Dienst. Je nach Distribution ist dies zumeist `rsyslogd` oder bei moderneren Distributionen das Journal (siehe [journalctl](#)).

- `-p n`  
protokolliert die Nachricht in der angegebenen Prioritätsstufe zwischen 0 = emerg und 7 = debug.
- `-p facility.level`  
protokolliert die Nachricht für den angegebenen Dienst und in der angegebenen Prioritätsstufe. Standardmäßig verwendet `logger` die Einstellung `user.notice`.
- `-t tag`  
speichert zusammen mit der Nachricht das angegebene Schlüsselwort. Das vereinfacht eine spätere Suche nach entsprechenden Nachrichten.

## Beispiel

Das folgende `logger`-Kommando könnte am Ende eines Backup-Scripts stehen. Es protokolliert das erfolgreiche Ende des Backups.

```
user$ logger -p local0.notice -t mybackuptools "Backup completed"
```

**logind** [optionen] [kommando] [name]

`logind` gehört wie [systemctl](#) und [journalctl](#) zur `systemd`-Familie. Mit dem Kommando können Sie den Login-Manager (siehe `man systemd-logind.service`) steuern bzw. Daten abfragen. Ohne Optionen oder Parameter zeigt das Kommando einfach alle aktiven Logins (Sessions) an.

- `-p property`  
gibt an, welche Eigenschaft abgefragt werden soll. Ohne diese Option liefern Subkommandos wie `show-session` oder `show-user` sehr viele Details.
- `active sessionname`  
aktiviert die angegebene Session. Das Kommando wechselt gegebenenfalls auch die aktive Konsole (wie durch die Tastenkürzel `Strg+Alt+F1`, `Alt+F2` etc.).

- `list-sessions`  
zeigt alle aktiven Sessions an (Defaultverhalten).
- `list-users`  
zeigt alle eingeloggten Benutzer an.
- `session-status sessionname`  
zeigt an, welche Prozesse innerhalb einer Session gestartet wurden und welche Logging-Meldungen protokolliert wurden.
- `show-session sessionname`  
zeigt Details zu einer Session an.
- `show-user username`  
zeigt Details zu einem Benutzer an.
- `terminate-session sessionname`  
beendet eine Session. Alle zugeordneten Prozesse erhalten ein Kill-Signal.

## Beispiel

Auf dem Beispielrechner gibt es vier Sessions. Auf Konsole 1 (`tty1`) läuft der Gnome Display Manager und zeigt eine Login-Box an. Ein Benutzer ist lokal auf Konsole 7 (`tty7`) eingeloggt. Außerdem gibt es zwei weitere Logins, die via SSH erfolgt sind (was die Ausgabe von `logindctl` aber nicht verrät).

```
root# logindctl
   SESSION   UID   USER     SEAT    TTY
      5     1000  kofler
      c1    119   gdm      seat0   tty1
      2     1000  kofler   seat0   tty7
      4     1000  kofler
```

Das zweite `logindctl`-Kommando zeigt, dass das Grafiksystem in Konsole 7 Wayland nutzt. Kommt stattdessen Xorg zum Einsatz, lautet die Ausgabe `Type=x11`.

```
root# logindctl show-session 2 -p Type
  Type=wayland
```

## logname

`logname` zeigt den Login-Namen (Benutzernamen) an.

## logout

`logout` oder noch kürzer einfach `Strg`+`D` beendet die Sitzung in einer Konsole oder in einem Terminalfenster.

### **lpadmin [optionen]**

`lpadmin` richtet einen neuen Drucker für das CUPS-Drucksystem ein, verändert dessen Zugriffsrechte für den Netzwerkbetrieb bzw. löscht ihn wieder.

- `-d name`  
definiert den angegebenen Drucker als Standarddrucker.
- `-E name`  
aktiviert den angegebenen Drucker.
- `-p name`  
richtet einen neuen Drucker ein. Mit diversen weiteren Optionen geben Sie die Konfigurationsparameter an (siehe `man lpadmin`).
- `-x name`  
löscht die Konfiguration für den angegebenen Drucker.

### **lpinfo [optionen]**

`lpinfo` listet die für CUPS verfügbaren Devices und Treiber auf.

- `-l`  
liefert besonders ausführliche Informationen. Die Option muss mit `-m` oder `-v` kombiniert werden.
- `-m`  
liefert eine Liste aller verfügbaren Druckertreiber.
- `-v`  
liefert eine Liste aller bekannten Druck-Devices.

### **lpoptions [optionen]**

`lpoptions` zeigt die Optionen von CUPS-Druckern an bzw. verändert sie.

- `-l`  
liefert eine Liste der verfügbaren Optionen und deren aktueller Einstellung.

- `-o optionsname=wert`  
verändert die Einstellung der angegebenen Option.
- `-p name`  
gibt den gewünschten Drucker an. (Ohne die Option bezieht sich `lpoptions` auf den Standarddrucker.)

### **lpq [optionen]**

`lpq` liefert eine Liste aller zwischengespeicherten Dateien bzw. Druckaufträge. Dabei werden auch die Größe der Datei sowie eine Jobnummer angegeben. Diese Jobnummer können Sie als Parameter von [`lprm`](#) angeben, um eine Datei aus dem Drucker-Spooler zu entfernen.

- `-a`  
zeigt die Druckjobs aller Warteschlangen an.
- `-Pname`  
zeigt die Druckjobs der angegebenen Warteschlange *name* an.

### **lpr datei**

`lpr` druckt die angegebene Datei aus.

- `-l`  
umgeht das Filtersystem und sendet die Druckerdaten unverändert an den Drucker. Die Option ist dann sinnvoll, wenn eine Druckdatei bereits im druckerspezifischen Format vorliegt.
- `-o options`  
übergibt diverse Zusatzparameter, z.B. `-o media=A4` oder `-o page-ranges=23-27,29,31`. Zahlreiche Beispiele finden Sie in der CUPS-Dokumentation: <https://www.cups.org/doc/options.html>
- `-Pname`  
verwendet die Warteschlange *name* statt des Standarddruckers.  
Beachten Sie, dass der Option kein Leerzeichen folgt!

### **lprm [optionen] [id]**

`lprm` bricht den aktuellen bzw. den durch die ID-Nummer angegebenen Druckjob ab.

- **-Pname**  
gibt die Warteschlange an.

### **lpstat [optionen]**

`lpstat` zeigt Informationen über CUPS-Klassen, -Drucker und deren Druckjobs an. `lpstat` funktioniert gleichermaßen für lokale Drucker wie für Netzwerkdrucker.

- **-a**  
zeigt für alle Drucker an, ob sie bereit sind, Druckaufträge entgegenzunehmen.
- **-c**  
zeigt alle Klassen an.
- **-d**  
zeigt den Standarddrucker an.
- **-s**  
zeigt eine Statusübersicht an (Standarddrucker, Liste aller Klassen und Drucker etc.).
- **-t**  
zeigt alle verfügbaren Informationen an.
- **-v**  
zeigt alle Drucker an.

### **ls [-optionen] [pfad]**

`ls` zeigt eine Liste aller Dateien und Verzeichnisse an. Wenn `ls` ohne weitere Parameter oder Optionen verwendet wird, liefert das Kommando eine mehrspaltige, nach Dateinamen sortierte Tabelle, in der alle Dateien, Links und Verzeichnisse im aktuellen Verzeichnis angezeigt werden.

- **-a bzw. -all**  
zeigt auch Dateien an, die mit `.` beginnen. Eine ganz ähnliche Wirkung hat die Option `-A`. Der einzige Unterschied besteht darin, dass die Dateien `.` und `..` (Verweis auf das aktuelle und das übergeordnete Verzeichnis) nicht angezeigt werden.

- **--color [=never|auto|always]**  
verwendet unterschiedliche Farben für unterschiedliche Dateitypen (Links, Verzeichnisse etc.) bzw. deaktiviert die Farbdarstellung. Mit [dircolors](#) können Sie die Farben selbst konfigurieren.
- **-d bzw. --directory**  
zeigt nur den Namen des Verzeichnisses, nicht aber seinen Inhalt an. Die Option ist beispielsweise dann sinnvoll, wenn als Pfad ein Verzeichnisname angegeben wird und die Zugriffsrechte dieses Verzeichnisses überprüft werden sollen (und nicht sein Inhalt).
- **-h**  
zeigt die Dateigrößen in KiB, MiB und GiB an (*human readable*).
- **-i bzw. --inode**  
zeigt zusätzlich zu den restlichen Informationen auch den I-Node der Datei an. (Der I-Node ist eine interne Identifikationsnummer der Datei, die zur Linux-internen Dateiverwaltung benötigt wird.) Die Option kann zur Erkennung von Links eingesetzt werden. (Durch Links verknüpfte Dateien haben denselben I-Node.)
- **-I *datei* bzw. --ignore *datei***  
**-Imuster**  
a schließt die angegebenen Dateien von der Anzeige aus. -I\*ps verhindert beispielsweise, dass Dateien mit der Endung ps angezeigt werden. Wenn nach -I nicht eine einzelne Datei, sondern ein Dateimuster angegeben wird, dann darf zwischen -I und dem Muster kein Leerzeichen eingegeben werden!
- **-l bzw. --format=long oder --format=verbose**  
zeigt zusätzlich zum Dateinamen weitere Informationen an: die Dateigröße in Bytes, die Zugriffsrechte etc. Zur Anzeige jeder Datei wird eine eigene Zeile verwendet (statt der platzsparenden mehrspaltigen Auflistung).

- **-L bzw. --dereference**  
zeigt bei einem symbolischen Link auf ein Verzeichnis nicht den Pfad des Links, sondern den Inhalt des Ursprungsverzeichnisses an.
- **-o bzw. --no-color**  
verzichtet auf unterschiedliche Farben oder Schriftarten.
- **-p bzw. -F**  
hängt an die Dateinamen ein Sonderzeichen an, das den Typ der Datei kennzeichnet. Diese Option wird bei manchen Linux-Distributionen in */etc/profile* durch eine *alias*-Abkürzung standardmäßig aktiviert. Die wichtigsten Sonderzeichen sind: / für Verzeichnisse, @ für symbolische Links, \* für ausführbare Dateien und | für FIFOs.
- **-r bzw. --reverse**  
dreht die Sortierreihenfolge um. Die Option wird oft in Kombination mit -t oder -s verwendet.
- **-R bzw. --recursive**  
erfasst auch Dateien in Unterverzeichnissen.
- **-s bzw. --sort=size**  
sortiert die Dateien nach ihrer Größe (die größte Datei zuerst).
- **-t bzw. --sort=time**  
sortiert die Dateien nach Datum und Uhrzeit der letzten Änderung (die neueste Datei zuerst).
- **-u bzw. --sort=access**  
sortiert die Dateien nach Datum und Uhrzeit des letzten Lesezugriffs. Die Option muss zusammen mit -t angegeben werden (sonst sortiert ls überhaupt nicht).
- **-v**  
sortiert Nummern in Dateinamen vernünftig, also z.B. *tty2* vor *tty10*.

- **-x bzw. --sort=extension**  
sortiert die Dateien nach ihrer Kennung (d.h. nach der Buchstabenkombination, die auf den letzten . im Dateinamen folgt).
- **-z bzw. --context**  
zeigt die SELinux-Kontextinformationen. Mehr Informationen liefert --lcontext, weniger Informationen gibt --scontext.  
`ls` zeigt leider nicht den gesamten Speicherbedarf aller aufgelisteten Daten an. Diese Aufgabe übernimmt das Kommando [du](#). `ls` ist nicht in der Lage, die Zugriffsbits einer Datei oktal anzuzeigen. Dazu verwenden Sie das Kommando [stat](#).

## Beispiel

Das folgende Kommando zeigt alle Dateien im aktuellen Verzeichnis an und sortiert sie nach dem Datum (die neueste Datei zuletzt):

```
user$ ls -ltr
-rw-r--r-- 1 kofler kofler 1681276 2020-03-02 10:01 cimg3079.jpg
-rw-r--r-- 1 kofler kofler 1582496 2020-03-02 10:01 cimg3014.jpg
-rw-r--r-- 1 kofler kofler 1615070 2020-03-02 10:01 cimg2965.jpg
...
```

Kurz einige Anmerkungen zur Interpretation des `ls`-Ergebnisses:  
Die zehn Zeichen am Beginn der Zeile geben den Dateityp und die Zugriffsbits an. Als Dateityp kommen infrage: der Bindestrich - für eine normale Datei, *d* für ein Verzeichnis (*directory*), *b* oder *c* für eine Device-Datei (*block* oder *char*) oder *l* für einen symbolischen Link.

Die nächsten drei Zeichen (rwx) geben an, ob der Besitzer die Datei lesen, schreiben und ausführen darf. Analoge Informationen folgen für die Mitglieder der Gruppe sowie für alle anderen Systembenutzer.

Die Zahl im Anschluss an die zehn Typ- und Zugriffszeichen gibt an, wie viele Hard-Links auf die Datei verweisen. Die weiteren Spalten geben den Besitzer und die Gruppe der Datei an (hier jeweils `kofler`),

die Größe der Datei, Datum und Uhrzeit der letzten Änderung und zuletzt den Dateinamen.

### **lsattr [optionen] dateien**

`lsattr` zeigt den Zustand der Zusatzattribute von Dateien oder Verzeichnissen in Linux-Dateisystemen an (ext2 bis ext4, [btrfs](#), xfs etc.). Diese Zusatzattribute steuern beispielsweise, ob Dateien komprimiert werden sollen, ob Änderungen an Dateien mit dem Copy-on-Write-Verfahren durchgeführt werden sollen etc. Es hängt von der Implementierung des jeweiligen Dateisystems ab, welche Attribute tatsächlich beachtet werden – bei den meisten Dateisystemen nur ganz wenige (siehe auch die Beschreibung des Kommandos [chattr](#) zur Veränderung dieser Attribute).

- -a  
zeigt auch versteckte Dateien an.
- -d  
zeigt die Attribute von Verzeichnissen an, nicht die Attribute der Dateien in den Verzeichnissen.
- -R  
zeigt rekursiv die Attribute aller Dateien im Verzeichnisbaum an.

### **Beispiel**

Auf dem Testsystem (eine openSUSE-Leap-Installation mit [btrfs](#)-Dateisystem) ist für die Verzeichnisse `/var/lib/mysql` und `/var/lib/libvirt/images` das Attribut c gesetzt. Es deaktiviert die Copy-on-Write-Funktion für Dateien in diesem Verzeichnis.

```
root# lsattr -d /var /var/lib /var/lib/libvirt/images /var/lib/mariadb
----- /var
----- /var/lib/
-----c /var/lib/libvirt/images
-----c /var/lib/mariadb/
```

### **lsblk [optionen] [iodevice]**

`lsblk` liefert eine hierarchische Liste sämtlicher Block-Devices bzw. aller Devices, die sich auf einem Datenträger befinden. Das

Kommando ist ausgezeichnet geeignet, um rasch einen guten Überblick über alle Partitionen und Logical Volumes zu erhalten.

- -a  
bezieht auch leere Devices in das Ergebnis ein.
- -b  
gibt Größenangaben in Byte an.
- -f  
zeigt zu jedem Device auch den Dateisystemtyp, den Label und die UUID an.
- -p  
gibt vollständige Device-Namen an, also z.B. *dev/mapper/fedora-swap* bei Logical Volumes.

## Beispiel

Das folgende `lsblk`-Ergebnis ist auf einem Fedora-Rechner mit zwei Festplatten und einem DVD-Laufwerk entstanden. Das root-Dateisystem und die Swap-Partition befinden sich in Logical Volumes.

```
root# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda       8:0    0 232,9G  0 disk
  sda1    8:1    0   487M  0 part /boot/efi
  sda2    8:2    0  14,8G  0 part
    fedora-swap 253:0    0  500M  0 lvm  [SWAP]
    fedora-root 253:1    0  14,3G  0 lvm  /
  sda3    8:3    0  93,2G  0 part /virt
sdb       8:16   0 931,5G  0 disk
  sdb1    8:17   0   128M  0 part
  sdb2    8:18   0 146,4G  0 part
  sdb3    8:19   0 783,5G  0 part /data
sr0     11:0    1 1024M  0 rom
```

## lsb\_release [optionen]

`lsb_release` ermittelt distributionsspezifische Informationen, also z.B. den Namen und die Versionsnummer des laufenden Linux-Systems. LSB steht für *Linux Standard Base*. Wenn keine Option übergeben wird, gilt -v.

- **-a**  
zeigt alle zur Verfügung stehenden Informationen an: den Distributionsnamen, die vollständige Beschreibung, die Versionsnummer und den Code-Namen (siehe das Beispiel unten).
- **-v**  
zeigt die LSB-Versionsnummer an – allerdings nur, wenn außer dem Kommando auch die LSB-Module installiert sind (Paket `lsb-core`). Oft ist das nicht der Fall – dann liefert das Kommando die Fehlermeldung *no LSB modules are available*. In der Praxis ist die LSB-Versionsnummer freilich selten von Interesse. Wenn Sie nur die Version des installierten Linux-Systems wissen wollen, verwenden Sie die Option `-a`.

## Beispiel

Im folgenden Beispiel wurde `lsb_release` auf einer Installation von Ubuntu 20.04 ausgeführt:

```
root# lsb_release -a
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.1 LTS
Release:        20.04
Codename:       focal
```

## **lscpu [optionen]**

`lscpu` zeigt Detailinformationen zur CPU, deren Cores und Cache-Speicher sowie gegebenenfalls zum Virtualisierungssystem an.

- **-e**  
zeigt nur eine Zusammenfassung der Eckdaten in einer gut lesbaren Form an.
- **-p**  
wie `-e`, liefert die Daten aber in einer Form, die einfach in Scripts verarbeitet werden kann.

## **lshw [optionen]**

Das Kommando `lshw` aus dem gleichnamigen Paket erstellt eine hierarchisch geordnete Liste aller Hardware-Komponenten des

Rechners.

- **-businfo**  
listet die Komponenten zeilenweise auf und gibt zu jeder Komponente eine ID-Nummer für den verwendeten SCSI-, USB- oder PCI-Bus an.
- **-html**  
formatiert die Ausgabe als HTML-Dokument.
- **-short**  
liefert nur eine kompakte Zusammenfassung.
- **-xml**  
formatiert die Ausgabe als XML-Dokument.

## Beispiel

Die folgende Hardware-Übersicht ist auf einem Raspberry Pi 4B entstanden.

```
root# lshw -businfo
Bus info          Device  Class      Description
=====
                                         system    Raspberry Pi 4 Model B Rev 1.1
                                         bus       Motherboard
cpu@0              processor cpu
cpu@1              processor cpu
cpu@2              processor cpu
cpu@3              processor cpu
                     memory   855MiB System memory
pci@0000:00:00.0  bridge    Broadcom Limited
pci@0000:01:00.0  bus       VL805 USB 3.0 Host Controller
usb@1              usb1     bus       xHCI Host Controller
usb@1:1            bus       bus       USB2.0 Hub
usb@1:1.4          bus       bus       Keyboard Hub
usb@1:1.4.2        input     input     Apple Keyboard
usb@1:1.4.3        input     input     USB-PS/2 Optical Mouse
usb@2              usb2     bus       xHCI Host Controller
                     eth0     network   Ethernet interface
                     wlan0    network   Wireless interface
```

## lsmod

`lsmod` liefert eine Liste aller Module, die momentan in den Kernel geladen sind.

## lsof [optionen] [datei/schnittstelle]

Ohne weiteren Parameter liefert `lsof` eine zumeist sehr lange Liste aller momentan offenen Dateien bzw. Schnittstellen und der ihnen

zugeordneten Prozesse. `lsof` liefert Informationen zum Prozess, der diese Datei offen hält.

- `-i adresse`

liefert Informationen zu Prozessen, die die angegebene Netzwerkadresse nutzen. Die Adresse setzt sich in der Form [46] `[protocol][@hostname|hostaddr][:service|port]` zusammen. Beispielsweise liefert `-i 4tcp` Informationen zu allen Prozessen, die das Protokoll TCP in der IP-Version 4 nutzen.

- `-n`

verzichtet auf die Auflösung von Netzwerknamen.

- `-N`

berücksichtigt auch NFS-Dateien.

- `-u user`

liefert nur Informationen zu Dateien/Schnittstellen, die vom angegebenen Benutzer genutzt werden. Es können auch mehrere Benutzer angegeben werden (Name oder UID, getrennt durch Kommas).

- `-X`

ignoriert alle offenen TCP- und UDP-Dateien.

## Beispiele

Die beiden folgenden Kommandos zeigen alle Prozesse, die das Protokoll UDP bzw. den Port 22 nutzen:

```
root# lsof -i udp
ntpd      3696     ntp    16u   IPv4    9026      UDP *:ntp
portmap    4745  daemon    3u   IPv4   12931      UDP *:sunrpc
rpc.statd  4764  statd    5u   IPv4   12962      UDP *:700
...
root# lsof -i :22
COMMAND  PID USER    FD   TYPE DEVICE SIZE NODE NAME
sshd     5559 root    3u   IPv6  14097      TCP *:ssh (LISTEN)
sshd     7729 root    3r   IPv6  33146      TCP mars.sol:ssh->merkur.sol:45368
                                                (ESTABLISHED)
```

## **`lspci` [optionen]**

`lspci` liefert Informationen über den PCI-Bus und sämtliche daran angeschlossenen Geräte.

- **-k**  
zeigt zu jedem PCI-Gerät an, welches Kernelmodul dafür verantwortlich ist.
- **-tv**  
liefert eine baumförmige Geräteliste, die deutlich macht, wie die Geräte miteinander verbunden sind.
- **-v bzw. -vv bzw. -vvv**  
liefert noch mehr Details.

### **lsscsi [optionen]**

`lsscsi` liefert Informationen über alle angeschlossenen SCSI- und SATA-Geräte.

- **-c**  
zeigt die Daten im selben Format wie `/proc/scsi/scsi` an.
- **-H**  
liefert eine Liste der SCSI-Hosts (statt der SCSI-Geräte).
- **-l**  
liefert ausführliche Informationen zu jedem Gerät.

### **Beispiel**

Das folgende Ergebnis ist auf einem Rechner mit zwei PCIe-SSDs und einer externen USB-Festplatte entstanden.

```
root# lsscsi
[0:0:0:0]    disk    WD Elements 25A2 1026      /dev/sda
[N:0:4:1]    disk    PM981 NVMe Samsung 2048GB_1  /dev/nvme0n1
[N:1:4:1]    disk    SAMSUNG MZVLB256HAHQ-000L7_1 /dev/nvme1n1
```

### **lsusb [optionen]**

`lsusb` liefert Informationen über alle angeschlossenen USB-Geräte.

- **-t**  
rückt die Ausgabe baumartig ein und verdeutlicht so, an welchem Bus welches Gerät angeschlossen ist.
- **-v**  
liefert ausführliche Informationen zu jedem Gerät.

### **Beispiel**

Das folgende Kommando zeigt, dass an den Computer unter anderem eine Apple-Tastatur angeschlossen ist:

```
user$ lsusb
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 004: ID 1058:25a2 Western Digital Technologies, Inc. Elements 25A2
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 007: ID 06cb:009a Synaptics, Inc.
Bus 001 Device 019: ID 05ac:021e Apple, Inc. Aluminum Mini Keyboard (ISO)
...
```

### **lvcreate [optionen] name**

Das LVM-Kommando `lvcreate` erzeugt ein neues Logical Volume (LV) innerhalb der durch `name` angegebenen Volume Group (VG).

- `-i n`  
verteilt das LV gleichmäßig über die angegebene Anzahl von Physical Volumes (PVs). Das setzt voraus, dass die VG aus zumindest so vielen PVs besteht. Sofern sich die PVs auf unterschiedlichen Festplatten befinden, erzielen Sie damit einen ähnlichen Effekt wie mit RAID-0 (Striping).
- `-L size`  
gibt die gewünschte Größe an. Die Größenangabe erfolgt standardmäßig in MiB, mit dem entsprechenden Suffix aber auch in Byte, KiB, GiB oder TiB (z.B. `-L 2G`).
- `-n lvname`  
gibt den LV-Namen an. Wenn diese Option fehlt, verwendet das Kommando den Namen `lvoln`.
- `-s`  
erzeugt einen Snapshot, also eine statische Kopie eines vorhandenen LVs. `name` gibt in diesem Fall nicht die VG, sondern das zugrunde liegende LV an. Die Größenangabe `-L` gibt den Speicher an, der für kopierte Sektoren benötigt wird, wenn sich das zugrunde liegende LV ändert. Ist dieser Speicher erschöpft, endet die Lebensdauer des Snapshots.

### **Beispiel**

Das folgende Kommando richtet in der VG `myvg1` ein neues Logical Volume mit dem Namen `myvol1` in der Größe von 50 GiB ein:

```
root# lvcreate -L 50G -n myvol1 myvg1
Logical volume "myvol1" created
```

Durch das Kommando wird gleichzeitig auch die Datei `/dev/myvg1/myvol2` erzeugt. Dabei handelt es sich um einen Link auf die Datei `/dev/mapper/myvg1-myvol2`. Das LV kann jetzt unter einem dieser beiden Device-Namen wie eine gewöhnliche Festplattenpartition verwendet werden. Das folgende Kommando richtet dort ein Dateisystem ein:

```
root# mkfs.ext4 /dev/myvg1/myvol1
```

### **lvdisplay** [optionen] lvname

`lvdisplay` zeigt Detailinformationen zu einem LV an.

### **lvextend** [optionen] lvname [pvname]

`lvextend` vergrößert den Speicher für das angegebene LV-Device. Den neuen Gesamtspeicher geben Sie mit der Option `-L` an (siehe [lvcreate](#)). Alternativ können Sie die Schreibweise `-L +size` verwenden, um die gewünschte Änderung anzugeben. Der benötigte Speicher wird innerhalb derselben VG reserviert, in der sich das LV befindet. Wenn die VG aus mehreren Physical Volumes (PVs) zusammengesetzt ist, können Sie durch die optionale Angabe von `pvname` auswählen, welche PV den Speicher zur Verfügung stellt.

Ein Beispiel zur Anwendung von `lvextend` finden Sie bei der Beschreibung des Kommandos [resize2fs](#), mit dem Sie die Größe von ext-Dateisystemen ändern.

### **lvm** [kommando]

`lvm` ist das zentrale Administrationskommando für den Logical Volume Manager. `lvm` befindet sich üblicherweise im Paket `lvm2` und muss bei vielen Distributionen extra installiert werden, wenn nicht bereits bei der Installation ein LVM-System eingerichtet wurde.

Wenn Sie `lvm` ohne weitere Kommandos ausführen, gelangen Sie in eine Shell, in der Sie interaktiv LVM-Kommandos ausführen

können. [help](#) liefert eine Liste aller zur Auswahl stehenden Kommandos. Alternativ können Sie an `lvm` ein Kommando übergeben, das dann sofort ausgeführt wird.

Alle LVM-Kommandos können wahlweise via `lvm` oder direkt ausgeführt werden. Die Kommandos `lvm lvcreate xxx` und `lvcreate xxx` sind daher gleichwertig.

Aus Gründen der Übersichtlichkeit habe ich mich in diesem Buch dazu entschlossen, die wichtigsten LVM-Kommandos separat zu beschreiben. Kommandos zur Bearbeitung von Logical Volumes beginnen mit den Anfangsbuchstaben `lv`, solche zur Administration von Physical Volumes beginnen mit `pv`, und Kommandos zum Verwalten von Volume Groups beginnen mit `vg`.

#### **lvreduce [optionen] lvname**

`lvreduce` verkleinert den Speicherplatz des angegebenen LV-Device. Wenn das LV ein Dateisystem enthält, muss dieses *vorher* verkleinert werden, andernfalls kommt es zu Datenverlusten! Sie können wahlweise die neue Gesamtgröße in der Form `-L size` angeben oder die gewünschte Änderung in der Form `-L -size`.

#### **lvremove [optionen] lvname**

`lvremove` löscht das angegebene LV.

- `-f`

verzichtet auf Rückfragen.

#### **lvrename oldlvname newlvname**

`lvrename` gibt dem LV einen neuen Namen.

#### **lvscan [optionen]**

`lvscan` listet alle LVs auf.

#### **lzop [optionen] [dateiname]**

`lzop` aus dem gleichnamigen Paket komprimiert bzw. dekomprimiert Dateien. `lzop` erreicht deutlich schlechtere Kompressionsraten als [gzip](#), [bzip2](#) oder [xz](#), ist dafür aber *viel* schneller. Die Syntax von `lzop` orientiert sich am [gzip](#)-Kommando.

`lzop` ist das perfekte Kommando, wenn Sie Dateien platzsparend sichern möchten (z.B. für ein Backup), dafür aber nicht allzu viel CPU-Leistung investieren möchten. Dabei werden Sie `lzop` zumeist im Pipe-Modus verwenden, wie bei den beiden [tar](#)-Beispielen unten. In den meisten Fällen sparen Sie durch das Komprimieren nicht nur Platz, sondern auch Zeit!

Bei vielen Dateien ist die Zeitersparnis bei der Datenspeicherung (I/O) bzw. bei der Netzwerkübertragung größer als die zum Komprimieren erforderliche Zeit.

- `-c` bzw. `--stdout` oder `--to-stdout`  
leitet das Ergebnis auf die Standardausgabe um.
- `-d` bzw. `--decompress` oder `--uncompress`  
dekompriamt die angegebene Datei, anstatt sie zu komprimieren.
- `-n`  
steuert die Geschwindigkeit und Qualität der Kompression. `-1` ist am schnellsten, `-9` liefert die kleinsten Dateien. Standardmäßig gilt `-3`.
- `-U`  
überschreibt die vorhandene Datei. (Standardmäßig bleibt die ursprüngliche Datei erhalten. Die neue, komprimierte Datei erhält das Suffix `.lzo`.)

## Beispiele

Das erste Kommando komprimiert die Datei und liefert `datei.lzo`.

Das zweite Kommando dekomprimiert die Datei wieder:

```
user$ lzop datei  
user$ lzop -d datei.lzw
```

Die beiden folgenden Beispiele zeigen die Anwendung von `lzop` in Kombination mit [tar](#):

```
user$ tar cf -c verz | lzop -c > backup.tar.lzo      (Archiv erstellen)  
user$ lzop -d < backup.tar.lzo | tar xf -           (Archiv auspacken)
```

## M

```
mail [optionen] mail@hostname [< nachtentext]
```

Mit dem Kommando `mail` aus dem Paket `mailx` oder `mailutils` (Debian, Ubuntu) können Sie lokale, in `/var/mail/name` gespeicherte E-Mails lesen. Die Bedienung ist aber gelinde gesagt gewöhnungsbedürftig. Wenn Sie einen Mail-Client für den Textmodus suchen, installieren Sie besser `mutt`.

Die Stärke von `mail` liegt vielmehr darin, dass Sie damit unkompliziert E-Mails an einen lokalen Mail-Server übergeben können, wahlweise interaktiv oder in Scripts. Interaktiv führen Sie einfach `mail empfänger@site` aus. Sie können nun weitere Empfänger (CC-Adressen), das Subject und schließlich den Nachrichtentext angeben, den Sie mit `[Strg]+D` abschließen.

Damit wird die E-Mail an den lokalen Mail-Server übergeben, wobei als Absender `loginname@hostname` verwendet wird.

Für den Script-gesteuerten Mail-Versand müssen Sie sich mit den `mail`-Optionen anfreunden, von denen hier nur die wichtigsten drei vorgestellt werden:

- `-a "header"`  
gibt eine zusätzliche Header-Zeile an. Das kann dazu verwendet werden, um eine BCC-Adresse hinzuzufügen oder um den Zeichensatz der Nachricht einzustellen (siehe das Beispiel). `mail` erkennt in vielen Konfigurationen den richtigen Mail-Hostnamen nicht und setzt die Absenderadresse dann falsch zusammen. Das gilt selbst auf Debian- und Ubuntu-Systemen, wo `/etc/mailname` sehr einfach auszuwerten wäre. Abhilfe schafft die explizite Angabe der Absenderadresse mit `-a "From:name@myhost.com"`.
- `-A datei`  
fügt der E-Mail eine Datei hinzu.

- **-s "subject"**  
gibt den Text für die Betreffzeile an.

## Beispiel

Das folgende Kommando sendet den UTF-8-Text aus *nachricht.txt* an die angegebenen Empfänger:

```
user$ mail name1@firma-abc.de, name1@firma-abc.de \
    -s "Subject 1 2 3 äöü" \
    -a "BCC: blindcopy@firma-abc.de" \
    -a "Content-Type: text/plain; charset=UTF-8" < nachricht.txt
```

## makepasswd [optionen]

`makepasswd` aus dem gleichnamigen Paket erzeugt ein neues, zufälliges Passwort. Das Kommando ist gut geeignet, wenn Sie für einen neuen Account ein Passwort benötigen, aber nicht Ihre eigene Fantasie strapazieren möchten.

Unter CentOS, Fedora und RHEL steht `makepasswd` nicht zur Verfügung. Dort installieren Sie stattdessen das Paket `expect`, das das Kommando `mkpasswd` enthält. Unter (open)SUSE steht auch `mkpasswd` nicht zur Verfügung. Dort können Sie auf [pwgen](#) ausweichen.

- **--chars *n***  
gibt die gewünschte Passwortlänge an (standardmäßig zehn Zeichen).
- **--count *n***  
liefert *n* Passwörter (standardmäßig nur eines).
- **--strings *zeichenkette***  
verwendet die in der Zeichenkette enthaltenen Zeichen, um das Passwort zusammenzusetzen. Standardmäßig verwendet das Kommando nur Groß- und Kleinbuchstaben (ASCII) und Ziffern, aber keine Sonderzeichen.

## Beispiel

Das folgende Kommando liefert ein Passwort mit zwölf Zeichen:

```
user$ makepasswd --chars 12
TaQKD1VGYmax
```

## man [gruppe] [optionen] name

`man` zeigt Online-Informationen zum angegebenen Kommando bzw. zur angegebenen Datei an. Durch die Angabe einer Gruppe kann die Suche eingeschränkt werden. Wichtige Gruppen sind 1 (Benutzerkommandos), 5 (Konfigurationsdateien) und 8 (Kommandos zur Systemadministration). Bei den meisten Distributionen werden die `man`-Texte durch [less](#) angezeigt, d.h., es gelten die beim Kommando [less](#) in [Tabelle 17](#) zusammengefassten Tastenkürzel zum Blättern bzw. Suchen im Hilfetext.

- `-a`  
zeigt der Reihe nach alle gleichnamigen `man`-Seiten an. Ohne diese Option wird gewöhnlich nur die erste von mehreren gleichnamigen Dateien aus unterschiedlichen Themengebieten angezeigt. Bei manchen Linux-Distributionen gilt `-a` als Standardeinstellung.
- `-f schlüsselwort`  
zeigt die Bedeutung eines Schlüsselworts an (einen einzeiligen Text). Mit dieser Option entspricht `man` dem Kommando `whatis thema`.
- `-k schlüsselwort`  
zeigt eine Liste aller vorhandenen `man`-Texte an, in denen das Schlüsselwort vorkommt. Dabei wird allerdings keine Volltextsuche durchgeführt. Vielmehr werden nur die Schlüsselwörter jedes `man`-Textes analysiert. Mit dieser Option entspricht `man` dem Kommando `apropos thema`.

### **md5sum dateien**

`md5sum` berechnet Prüfsummen zu allen angegebenen Dateien. Prüfsummen des *Message-Digest Algorithm 5* werden oft dazu verwendet, um nach der Übertragung einer Datei sicherzustellen, dass die Datei unverändert ist. Beachten Sie, dass der MD5-Algorithmus als nicht mehr sicher gilt. Kryptografisch besser sind die `shasum`-Kommandos.

## **mdadm** [optionen]

Das Kommando `mdadm` aus dem gleichnamigen Paket hilft bei der Verwaltung von Software-RAID-Verbunden. Das Kommando wird durch unzählige Optionen gesteuert. Dabei gibt zumeist eine Option den Modus an (z.B. `--create` bzw. `-c`, um einen neuen RAID-Verbund einzurichten). Die weiteren Optionen steuern Details der gewünschten Operation.

- `mddevice --add device`  
fügt dem Verbund eine Partition hinzu.
- `--assemble mddevice device1 device2 ... devicem`  
fügt den Verbund `mddevice` aus den Festplattenpartitionen `device1` bis `devicem` neu zusammen. Das funktioniert nur, wenn `device1` bis `devicem` vorher mit `--create` als Bestandteile eines RAID-Verbunds initialisiert wurden.
- `--assemble --scan mddevice`  
setzt die RAID-Partition `mddevice` aus den in `mdadm.conf` angegebenen Festplattenpartitionen zusammen.
- `--create mddevice --level=n --raid-devices=m device1 device2 ... devicem`  
erzeugt den neuen Verbund `mddevice` für den RAID-Level `n`, der aus den `m` Festplattenpartitionen `device1` bis `devicem` besteht.
- `--detail mddevice`  
liefert Detailinformationen über die RAID-Partition.
- `--examine device`  
liefert Detailinformationen über die angegebene Festplattenpartition, die Bestandteil eines RAID-Verbunds sein muss.
- `--examine --scan`  
liefert eine Zusammenfassung über alle verfügbaren RAID-Partitionen in der Syntax von `mdadm.conf`.
- `mddevice --fail device`  
markiert die Festplattenpartition `device` als fehlerhaft und

deaktiviert sie so.

- `--grow mddevice --size=max`  
vergrößert den Verbund, sodass die zugrunde liegenden Festplattenpartitionen optimal genutzt werden. Das Kommando ist in dieser Form nur nach einer Vergrößerung der Festplattenpartitionen sinnvoll.
- `--grow mddevice --raid-devices=n`  
vergrößert die Anzahl der in den Verbund integrierten Festplattenpartitionen (nur für die RAID-Level 1, 5 und 6).
- `--monitor device`  
überwacht die aktiven RAID-Verbunde. Das Kommando wird üblicherweise beim Hochfahren des Rechners gestartet und sendet bei Problemen eine Mail an den Administrator (Option `-m`).
- `--query device`  
testet, ob die Partition eine RAID-Partition bzw. Bestandteil eines RAID-Verbunds ist.
- `mddevice --remove device`  
entfernt die Festplattenpartition `device` aus dem RAID-Verbund. `device` darf nicht aktiv sein und muss gegebenenfalls vorher mit `--fail` deaktiviert werden.
- `--stop mddevice`  
deaktiviert den angegebenen Verbund. Der Verbund kann nun mit `--assemble` neu zusammengesetzt werden.
- `--zero-superblock device`  
löscht die RAID-Metadaten einer Festplattenpartition.

## Beispiele

Die beiden folgenden Kommandos bilden aus den beiden Partitionen `/dev/sda3` und `/dev/sdb3` einen RAID-0-Verbund und richten dort ein ext4-Dateisystem ein. Die Partitionen müssen zuvor als RAID-Partitionen gekennzeichnet werden. Wenn Sie zum Partitionieren [fdisk](#) verwenden, stellen Sie die Partitions-ID-

Nummer mit dem Kommando `[T]` auf den hexadezimalen Wert fd.

Bei [parted](#) führen Sie `set partitionsnummer raid on` aus.

```
root# mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/sda3 /dev/sdb3
mdadm: array /dev/md0 started.
root# mkfs.ext4 /dev/md0
```

Die Vorgehensweise beim Einrichten eines RAID-1-Verbunds ist exakt dieselbe wie bei RAID-0. Einzig das Kommando zum Einrichten des RAID-Systems sieht ein wenig anders aus und enthält nun `--level=1` statt `--level=0`:

```
root# mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sda3 /dev/sdb3
mdadm: array /dev/md0 started.
root# mkfs.ext4 /dev/md0
```

Um die Funktionsweise eines RAID-1-Verbunds zu testen, markieren Sie eine Partition als defekt:

```
root# mdadm /dev/md0 --fail /dev/sdb3
```

Sie können den Verbund weiter nutzen; alle Änderungen werden nun aber nur noch auf der verbleibenden Festplattenpartition gespeichert. Um `/dev/sdb3` wieder zu `/dev/md0` hinzuzufügen, müssen Sie die als defekt gekennzeichnete Partition zuerst explizit entfernen:

```
root# mdadm --remove /dev/md0 /dev/sdb3
root# mdadm --add      /dev/md0 /dev/sdb3
```

Nun beginnt die automatische Resynchronisation der beiden Partitionen, die je nach der Größe des Verbunds geraume Zeit dauert (Richtwert: ca. 20 Minuten pro 100 GiB bei herkömmlichen SATA-Festplatten). Immerhin können Sie in dieser Zeit weiterarbeiten. Das Dateisystem wird allerdings langsamer als sonst reagieren. `cat /proc/mdstat` gibt Auskunft darüber, wie weit der Synchronisationsprozess fortgeschritten ist.

Um alle durchgeführten Änderungen in der RAID-Konfigurationsdatei [`/etc/mdadm/mdadm.conf`](#) zu speichern, führen Sie das folgende Kommando aus. Anschließend müssen Sie mit einem Editor eventuell schon früher eingetragene RAID-Verbunde aus der Konfigurationsdatei entfernen, damit die Datei frei von Doppelgängern ist.

```
root# mdadm --examine --scan >> /etc/mdadm/mdadm.conf
```

## **mkdir** verzeichnis

**mkdir** erstellt ein neues Verzeichnis. Die wichtigsten Optionen lauten:

- **-m modus** bzw. **--mode=modus**  
setzt die Zugriffsrechte des neuen Verzeichnisses wie durch *modus* angegeben (siehe [chmod](#)).
- **-p** bzw. **--parents**  
erstellt auch Zwischenverzeichnisse. Wenn Sie **mkdir -p a/b/c** ausführen und die Verzeichnisse *a* sowie *a/b* noch nicht existieren, werden auch diese Verzeichnisse erstellt.
- **-Z**  
aktiviert für das neue Verzeichnis den Defaultkontext von SELinux.

## **mkfifo** datei

**mkfifo** richtet eine FIFO-Datei ein (*First In, First Out*). FIFO-Dateien funktionieren im Prinzip wie Pipes und ermöglichen den Datenaustausch zwischen zwei Programmen.

### **Beispiel**

Im folgenden Beispiel richtet **mkfifo** eine FIFO-Datei ein. [ls](#) leitet das Inhaltsverzeichnis auf diese Datei um. [more](#) liest es von dort. [ls](#) muss mit & als Hintergrundprozess gestartet werden, weil der Prozess [ls](#) erst dann zu Ende ist, wenn [more](#) alle Daten aus *fifo* gelesen hat.

```
user$ mkfifo fifo
user$ ls -l > fifo &
user$ more < fifo
```

Die Anzeige des Inhaltsverzeichnisses via [more](#) könnte natürlich viel einfacher auch ohne eine FIFO-Datei bewerkstelligt werden, nämlich durch [ls -l | more](#).

## **mkfs** [optionen] device [blocks]

**mkfs** richtet auf einer zuvor partitionierten Festplatte ein Dateisystem ein. **mkfs** kann nur von root ausgeführt werden. Das Programm verzweigt je nach angegebenem Dateisystem in das

Kommando `mkfs.dateisystemtyp`. Die wichtigsten dateisystemspezifischen `mkfs.***`-Kommandos werden im Folgenden beschrieben.

- `-t dateisystemtyp`  
gibt den Typ des Dateisystems an. Infrage kommen z.B. `ext4` oder `xfs`. Die Option `-t` muss als erste Option angegeben werden! Alle weiteren Optionen werden an das Kommando weitergegeben, das das Dateisystem tatsächlich einrichtet. Sie sind vom Typ des Dateisystems abhängig.

```
mkfs.btrfs [optionen] device1 [device2 device3 ...]
```

`mkfs.btrfs` richtet ein [btrfs](#)-Dateisystem ein. Das Kommando ist je nach Distribution Teil des Pakets `btrfs-tools`, `btrfs-progs` oder `btrfsprogs`, das unter Umständen extra installiert werden muss.

- `-A bzw. --alloc-start n`  
lässt am Beginn des Devices *n* Byte ungenutzt (standardmäßig gilt *n*=0).
- `-d bzw. --data type`  
gibt den gewünschten RAID-Typ für die Daten an. Erlaubte Werte sind momentan `raid0`, `raid1`, `raid10`, `raid5`, `raid6` oder `single` (gilt standardmäßig). Wenn Sie einen [btrfs](#)-RAID-Verbund einrichten, müssen Sie entsprechend viele Devices angeben (zwei für RAID-1, vier für RAID-10).
- `-L bzw. --label name`  
gibt dem Dateisystem einen Namen.
- `-m bzw. --data type`  
gibt den gewünschten RAID-Typ für die Metadaten an. Neben den bei der Option `-d` angegebenen RAID-Varianten gibt es zusätzlich `dup`. Diese Variante gilt standardmäßig bei Dateisystemen, die nur ein Device umfassen. Alle Metadaten werden daher aus Sicherheitsgründen doppelt gespeichert. Eine Ausnahme sind SSDs, wo bei Ein-Device-Dateisystemen per Default `single` gilt.

`dup` ist hier nicht zweckmäßig, weil die interne Optimierung vieler SSDs die Redundanz der Daten erkennt und diese nur einfach abspeichert. Damit geht der durch die Duplizierung beabsichtigte Sicherheitsgewinn wieder verloren.

Wenn Sie ein [btrfs](#)-Dateisystem ohne weitere Angabe von Optionen über mehrere Devices verteilen, gilt standardmäßig der RAID-Typ `raid1`. Somit werden die Daten einfach gespeichert, die Metadaten aber mehrfach!

Beim Erstellen eines RAID-Verbunds geben Sie in der Regel mit `-m` den gleichen RAID-Typ wie bei der Option `-d` an. Wenn Ihr Dateisystem so schnell wie möglich sein soll, verwenden Sie `-d raid0 -m raid0`, um alle Redundanzen zu eliminieren.

## Beispiel

Ein RAID-1-Dateisystem, das die beiden gleich großen Partitionen `/dev/sdb1` und `/dev/sdc1` nutzt, richten Sie mit diesem Kommando ein:

```
root# mkfs.btrfs -d raid1 -m raid1 /dev/sdb1 /dev/sdc1
```

Sollte ein Device ausfallen, können Sie das Dateisystem mit der zusätzlichen [mount](#)-Option `degraded` weiterverwenden:

```
root# mount -o degraded /dev/sdb1 /media/btrfs
```

Um den RAID-Verbund wiederherzustellen, fügen Sie dem Dateisystem wieder ein neues Device hinzu. Im folgenden Beispiel ist das wieder `/dev/sdc1`, wobei diese Partition nun aber von einer neuen Festplatte oder SSD stammt. Um das Dateisystem wieder über beide Devices zu verteilen und somit die RAID-1-Redundanz wiederherzustellen, müssen Sie außerdem `btrfs filesystem balance` ausführen. Bei großen Dateisystemen dauert die Ausführung dieses Kommandos naturgemäß sehr lange. Immerhin kann das Dateisystem in dieser Zeit genutzt werden, wenn auch mit stark verminderter Geschwindigkeit.

```
root# btrfs device add /dev/sdc1 /media/btrfs
root# btrfs filesystem balance /media/btrfs
```

Erst jetzt kann das defekte Device aus dem Dateisystem entfernt werden. Dabei verwenden Sie zur Device-Angabe das Schlüsselwort `missing`:

```
root# btrfs device delete missing /media/btrfs  
mkfs.exfat [optionen] device
```

`mkfs.exfat` aus dem Paket `exfat-utils` richtet ein exFAT-Dateisystem ein. Dieses Dateisystem ist speziell für große SD-Karten optimiert. Es kommt standardmäßig bei SDXC-Speicherkarten zum Einsatz und wird von SDXC-kompatiblen Digitalkameras und Smartphones unterstützt.

- `-i id`

bestimmt eine 32-Bit-ID-Nummer für das Dateisystem. Die Zahl wird als hexadezimaler Code angegeben. Bei der Ausführung des Kommandos ohne die Option wird eine zufällige ID festgelegt.

- `-n name`

legt den Namen (Label) für das Dateisystem fest. Standardmäßig bleibt der Name leer. Der Name kann bei Bedarf nachträglich mit [exfatlabel](#) verändert werden.

```
mkfs.ext2 [optionen] device [blocks]  
mkfs.ext3 [optionen] device [blocks]  
mkfs.ext4 [optionen] device [blocks]  
mke2fs [optionen] device [blocks]
```

`mke2fs` bzw. `mkfs.ext2`, `mkfs.ext3`- oder `mkfs.ext4` richten ein ext2-, ext3 oder ext4-Dateisystem ein. Alle vier Kommandos verweisen durch Links auf dasselbe Programm, verwenden aber unterschiedliche Standardoptionen, die in `/etc/mk2efs.conf` gespeichert sind.

- `-b n`

bestimmt die Blockgröße (normalerweise 4096 Bytes). Für `n` sind die Werte 1024, 2048, 4096 etc. erlaubt (also  $2^n \times 1024$ ).

- `-c`

testet vor dem Einrichten des Datenträgers mit [badblocks](#), ob

defekte Blöcke existieren. Derartige Blöcke werden für das Dateisystem nicht genutzt. Der Test erfordert, dass *jeder* Datenblock verändert und wieder gelesen werden muss, und verlängert daher die Zeit zum Einrichten des Dateisystems erheblich.

- **-i n**  
gibt an, nach jeweils wie vielen Bytes ein I-Node eingerichtet wird. (I-Nodes sind interne Verwaltungseinheiten eines Dateisystems. In einem I-Node werden alle Verwaltungsinformationen einer Datei mit Ausnahme des Namens – also Zugriffsbits, Besitzer, Datum des letzten Zugriffs etc. – gespeichert. Die Anzahl der I-Nodes wird beim Formatieren unveränderlich festgelegt.) Beispielsweise gibt es bei  $n=4096$  pro MiB  $1.048.576 / 4096 = 256$  I-Nodes, d.h., es können pro MiB maximal 256 Dateien oder Verzeichnisse gespeichert werden, egal wie klein die Dateien sind. Wenn Sie sehr viele sehr kleine Dateien speichern möchten, sollten Sie  $n$  kleiner wählen. Der Minimalwert beträgt 1024 (das entspricht 2048 Dateien pro MiB). Es ist nicht sinnvoll, den Wert kleiner als die Blockgröße zu wählen, da pro Block ohnedies nur eine Datei gespeichert werden kann. Wenn Sie dagegen nur sehr wenige sehr große Dateien speichern möchten, können Sie  $n$  auch größer wählen. Sie verringern damit den Verwaltungs-Overhead.
- **-J**  
ermöglicht die Übergabe zusätzlicher Parameter, etwa um die Größe der Journaling-Datei einzustellen oder um die Journaling-Datei in einem anderen Device einzurichten. Das kann unter Umständen die Geschwindigkeit des Dateisystems ein wenig verbessern, macht aber gleichzeitig die Administration und die Wiederherstellung des Dateisystems nach einem Crash komplizierter. Die zur Auswahl stehenden Optionen werden in der Manual-Seite von `mke2fs` ausführlich beschrieben.

- **-m n**  
gibt an, wie viel Prozent des Datenträgers für Daten von root reserviert werden sollen (standardmäßig 5 Prozent). Dieser Reservespeicher ermöglicht es root, selbst dann noch zu arbeiten, wenn das Dateisystem für alle anderen Benutzer bereits restlos voll ist. Es handelt sich also um eine Sicherheitsreserve.
- **-t ext2/ext3/ext4**  
gibt die gewünschte ext-Version an (nur bei `mke2fs` erforderlich).

## Beispiel

Die beiden folgenden Kommandos richten zuerst ein neues ext4-Dateisystem in der Partition `/dev/sda5` ein und deaktivieren dann die automatische Überprüfung des Dateisystems (siehe auch die Beschreibung des Kommandos [tune2fs](#)):

```
root# mkfs.ext4 /dev/sda5
root# tune2fs -i 0 -c 0 /dev/sda5
```

**mkfs.ntfs** [optionen] device

`mkfs.ntfs` richtet ein NTFS-Dateisystem ein, das kompatibel mit allen aktuellen Windows-Versionen ist. Das Kommando ist bei vielen Distributionen Teil des Pakets `ntfsprogs` und muss extra installiert werden.

- **-f bzw. --fast**

verzichtet darauf, jeden Datenblock mit 0-Bytes vollzuschreiben. Die Option beschleunigt die Formatierung um ein Vielfaches und ist unbedingt empfehlenswert.

**mkfs.vfat** [optionen] device

`mkfs.vfat` richtet ein Windows-Dateisystem ein, das kompatibel mit allen gängigen Windows-Versionen ist. Dieses Format bietet sich vor allem für USB-Sticks und SD-Karten an, die auch unter Windows oder in Kameras verwendet werden.

- **-F 12/16/32**

gibt den Typ der FAT-Tabellen an (12, 16 oder 32 Bit).

Normalerweise wählt `mkfs.vfat` bei großen Datenträgern automatisch den 32-Bit-Modus. Wenn nicht, hilft diese Option.

- `-I`  
erlaubt die Formatierung des gesamten Datenträgers, ohne dass eine Partitionstabelle eingerichtet wird (`mkfs.vfat -I /dev/sdg`). Mitunter kommt diese Methode bei USB-Sticks zur Anwendung.
- `-n name`  
gibt dem Dateisystem einen maximal 11 Zeichen langen Namen (Label).

## Beispiel

Das folgende Kommando richtet in der Partition `/dev/sdd1` ein 32-Bit-VFAT-Dateisystem mit dem Namen `MeineFotos` ein:

```
root# mkfs.vfat -F 32 -n MeineFotos /dev/sdd1  
mkfs.xfs [optionen] device
```

`mkfs.xfs` richtet ein `xfs`-Dateisystem ein. Dieser Prozess kann durch unzählige Optionen gesteuert werden (`man mkfs.xfs`), von denen hier die wichtigsten herausgegriffen sind:

- `-b size=n`  
gibt die gewünschte Blockgröße an (standardmäßig 4096 Byte).
- `-l logdev=device`  
speichert die Journaling-Daten in einer eigenen Partition.
- `-l size=n`  
gibt die Größe des Bereichs für die Journaling-Daten an. Ohne diese Option wählt `mkfs.xfs` selbst eine geeignete Größe.
- `-L label`  
gibt dem Dateisystem den gewünschten Namen (maximal 12 Zeichen).

```
mknod devicedatei [b|c] major minor
```

`mknod` richtet eine neue Device-Datei ein. Device-Dateien befinden sich im Verzeichnis `/dev` und ermöglichen den Zugriff auf diverse Hardware-Komponenten. Device-Dateien werden durch drei Informationen charakterisiert: `major` und `minor` geben durch zwei

Zahlen den Treiber an, mit dessen Hilfe auf die Geräte zugegriffen werden kann (Major- und Minor-Device-Nummer). Die Zeichen b oder c geben an, ob das Gerät gepuffert oder ungepuffert arbeitet. Bei allen aktuellen Linux-Distributionen werden Device-Dateien automatisch durch das udev-System eingerichtet. Eine manuelle Ausführung von `mknod` ist daher nicht erforderlich. Eine Liste der wichtigsten Linux-Devices und der dazugehörigen Device-Nummern finden Sie hier:

<https://www.kernel.org/doc/Documentation/admin-guide/devices.txt>

### **mkpasswd [optionen]**

`mkpasswd` aus dem Paket `expect` (CentOS, Fedora, RHEL) generiert zufällige Passwörter.

- `-c n`  
gibt an, wie viele Kleinbuchstaben mindestens im Passwort enthalten sein müssen (Default: 2).
- `-C n`  
gibt an, wie viele Großbuchstaben mindestens enthalten sein müssen (Default: 2).
- `-d n`  
gibt an, wie viele Ziffern mindestens enthalten sein müssen (Default: 2).
- `-l n`  
gibt die gewünschte Zeichenanzahl des Passworts an (standardmäßig 9).
- `-s n`  
gibt an, wie viele Sonderzeichen mindestens enthalten sein müssen.

### **Beispiel**

Das folgende Kommando erzeugt ein Sechs-Zeichen-Passwort ohne Sonderzeichen:

```
root# mkpasswd -l 6 -s 0  
SX39vz
```

### **mkswap** device/datei

`mkswap` richtet ein Device (z.B. eine Festplattenpartition) oder eine Datei als Swap-Bereich ein. `mkswap` kann nur von root ausgeführt werden. Der Swap-Bereich kann anschließend mit [swapon](#) aktiviert werden. Damit der Swap-Bereich bei jedem Rechnerstart automatisch genutzt wird, muss er in die Datei `/etc/fstab` eingetragen werden.

### **Beispiel**

Die folgenden Kommandos füllen zuerst eine 512 MiB große Datei mit Nullen, machen die Datei zur Swap-Datei und aktivieren diese schließlich:

```
root# dd if=/dev/zero of=swapfile bs=1M count=512
root# chmod 600 swapfile
root# mkswap swapfile
root# swapon swapfile
```

### **mkudffs** [optionen] device

`mkudffs` richtet ein UDF-Dateisystem ein.

- `--media-type=medientyp`  
gibt den Medientyp an. Zulässige Werte sind unter anderem `hd` (Festplatte), `cdrw`, `dvd` (für DVD-R[W] und DVD+R[W]), `dvDRAM` sowie `bdr` (Blu-ray Disc Recordable).
- `--udfrev=version`  
gibt die gewünschte UDF-Revisionsnummer an. Zulässige Werte sind `0x0102`, `0x0150`, `0x0200` und `0x0201` (gilt standardmäßig).
- `--vid=name`  
gibt den Datenträgernamen an (Volume ID).

### **modinfo** modulname

`modinfo` liefert Informationen über das angegebene Modul. Dazu zählen der vollständige Dateiname, der Autor, die Lizenz, abhängige Module, eine Kurzbeschreibung sowie eine Liste aller Parameter, die das Modul kennt.

### **modprobe** [optionen] modulname [parameter=wert ...]

`modprobe` lädt das angegebene Modul in den Kernel. Das Kommando ist eine erweiterte Variante von [insmod](#). Es beachtet die in `/usr/lib/modules/kernel-version/modules.dep` definierten Modulabhängigkeiten und lädt gegebenenfalls auch Module, die vom gewünschten Modul vorausgesetzt werden. Außerdem berücksichtigt es die in `/etc/modprobe.d/*` angegebenen Modulparameter.

- `-c`  
liefert eine schier endlose Liste aller zurzeit gültigen Moduloptionen und Moduleinstellungen. Die Parameter ergeben sich sowohl aus dem Quelltext des `modutils`-Pakets als auch aus den Einstellungen in `/etc/modprobe.d/*`.
- `-f`  
erzwingt das Laden des Moduls selbst dann, wenn es für eine andere Kernelversion kompiliert wurde. Ob das tatsächlich funktioniert, hängt davon ab, ob es zwischen der Kernel- und der Modulversion irgendwelche Inkompatibilitäten gibt.
- `-r`  
entfernt das Modul aus dem Kernel (anstatt es zu laden).

### **mogrify [optionen] bilddatei**

`mogrify` verändert Parameter einer Bilddatei, beispielsweise die Auflösung oder die Anzahl der Farben. Dabei kommen dieselben Optionen wie bei [convert](#) zum Einsatz. Im Gegensatz zu [convert](#), das eine neue Datei erzeugt, überschreibt `mogrify` die ursprüngliche Datei. Das Kommando ist Teil des Image-Magick-Pakets.

### **Beispiel**

Das folgende Kommando reduziert die Auflösung eines Bilds auf 800×600 Pixel:

```
user$ mogrify -resize 800x600 foto.jpg
```

### **more datei**

`more` zeigt den Inhalt einer Textdatei seitenweise an. Nach jeder Seite wird die Anzeige unterbrochen. `more` wartet jetzt auf eine

Tastatureingabe. Die wichtigsten Eingabemöglichkeiten sind ↵ (eine Zeile nach unten), die ⌘ (eine Seite nach unten), ⌘ B (eine Seite nach oben) und ⌘ Q (Quit). Statt more wird in der Regel das leistungsfähigere Kommando [less](#) eingesetzt.

```
mount
mount [optionen] device verzeichnis
```

mount ohne Parameter liefert eine Liste aller momentan eingebundenen Dateisysteme inklusive des Dateisystemtyps und der mount-Optionen. In der zweiten Syntaxvariante bindet mount einen Datenträger (Partition einer Festplatte, USB-Stick ...) in das Linux-Dateisystem ein. Dabei müssen Sie die Device-Bezeichnung des Datenträgers (beispielsweise `/dev/sdc1`) und das Zielverzeichnis im Verzeichnisbaum angeben.

Das Einbinden von Dateisystemen beim Start von Linux wird durch die Datei `/etc/fstab` gesteuert. Für alle in `/etc/fstab` verzeichneten Devices kann mount in einer Kurzform verwendet werden, bei der nur die Device-Datei oder das Mount-Verzeichnis angegeben wird. mount liest die fehlenden Daten und Optionen selbstständig aus `fstab`. Datenträger werden mit umount wieder aus dem Dateisystem entfernt (siehe umount). Sowohl mount als auch umount dürfen im Regelfall nur von *root* ausgeführt werden. Eine Ausnahme stellen Laufwerke dar, die in `fstab` mit der Option *user* oder *users* gekennzeichnet sind.

Die folgende Beschreibung allgemeiner und dateisystemspezifischer mount-Optionen erstreckt sich über mehrere Seiten. Im Anschluss daran folgen einige Beispiele.

### Allgemeine mount-Optionen

- `--move olddir newdir`  
ändert das Verzeichnis, an dem ein Dateisystem eingebunden ist.
- `-o optionen`  
ermöglicht die Angabe zusätzlicher Optionen, z.B. in der Form `-o`

`acl, user_xattr`. Die so angegebenen Optionen dürfen nicht durch Leerzeichen getrennt werden! Eine Referenz der Dateisystemoptionen folgt unten.

- `-r`  
verhindert Schreiboperationen auf dem Datenträger (*read-only*).
- `-t dateisystem`  
gibt das Dateisystem an. Infrage kommen unter anderem `ext2/3/4` sowie [`btrfs`](#) oder `xfs` für Linux-Partitionen, `vfat` und `ntfs` für Windows-Partitionen sowie `iso9660` und `udf` für Daten-CDs und -DVDs.

## Allgemeine Dateisystemoptionen

Mit `-o` können Sie unzählige Optionen übergeben, die das Verhalten des eingebundenen Dateisystems beeinflussen. Im Folgenden werden die Schlüsselwörter für allgemeine Optionen zusammengefasst. Dateisystemspezifische Optionen folgen etwas weiter unten.

Alle Optionen für `mount -o` sind auch in der vierten Spalte von `/etc/fstab` zulässig. Diese Datei wird während des Bootprozesses ausgewertet, unter Ubuntu durch das Kommando `mountall`. Die folgende Zusammenstellung gilt daher auch für `/etc/fstab` und für `mountall` und wurde deshalb um einige `fstab`- bzw. `mountall`-spezifische Schlüsselwörter ergänzt, die für das Kommando `mount` nicht unmittelbar von Bedeutung sind. Dazu zählen `defaults`, `noauto`, `nobootwait`, `optional`, `owner` und `users`.

- `atime / noatime / relatime / strictatime`  
gibt an, unter welchen Umständen die *Inode Access Time* einer Datei aktualisiert wird: bei jedem Lesezugriff (`atime` bzw. `strictatime`, POSIX-konform), maximal einmal pro Tag (`relatime`) oder nie (`noatime`). Bei Schreibzugriffen wird die *Inode Access Time* in jedem Fall geändert.

Die Option hat einen relativ großen Einfluss auf die Geschwindigkeit von Leseoperationen und auf die Lebensdauer von Flash-Datenspeichern. Standardmäßig gilt bis Kernel 2.6.29 `atime`, bei aktuelleren Kernelversionen aber die schnellere Einstellung `relatime`. Wenn Sie das E-Mail-Programm `mutt` einsetzen, dürfen Sie die Einstellung `noatime` nicht verwenden! `relatime` verursacht hingegen keine Probleme.

- `defaults`

gibt an, dass das Dateisystem mit den Standardoptionen in den Verzeichnisbaum eingebunden werden soll. `defaults` wird in der vierten Spalte von `/etc/fstab` angegeben, wenn keine anderen Optionen verwendet werden sollen. (Die Spalte darf nicht leer bleiben.)

- `dev / nodev`

bewirkt bzw. verhindert, dass Device-Dateien als solche interpretiert werden. Unter Linux können Dateien als Block- oder Character-Devices gekennzeichnet werden. Die Option ist aus Sicherheitsgründen für CD/DVD-Laufwerke sowie für externe Datenträger sinnvoll.

- `discard`

aktiviert bei den Dateisystemen `ext4`, [`btrfs`](#) und `xfs` die Trim-Funktion. Der Kernel informiert nun die Solid State Disk, welche Datenblöcke gelöscht wurden, und gibt der SSD so die Möglichkeit, die interne Verwaltung der Speicherzellen zu optimieren. Die Trim-Funktion funktioniert auch in Kombination mit LVM und Software-RAID (`mdraid`).

Ob man die `discard`-Option einsetzen soll, ist umstritten: Sie hilft zwar bei der langfristigen Optimierung der SSD, kann aber zu einer starken Verlangsamung von Löschoperationen führen (etwa wenn in kurzer Zeit viele Dateien gelöscht werden sollen, siehe beispielweise <https://patrick-nagel.net/blog/archives/337>). Eine

gute Alternative zur `discard`-Option ist ein Cron-Job, der einmal täglich oder einmal wöchentlich das Kommando [`fstrim`](#) ausführt und auf diese Weise alle gelöschten Sektoren an die SSD meldet.

- `exec / noexec`  
legt fest, ob auf dem Dateisystem befindliche Programme unter Linux ausgeführt werden dürfen oder nicht. Bei den meisten Dateisystemen gilt standardmäßig die Einstellung [`exec`](#). Sicherheitsbewusste Administratoren werden für CD/DVD-Laufwerke und externe Datenträger aber die Option `noexec` angeben. (Wenn Sie die Option `user` oder `users` verwenden, gilt `noexec`. Das kann durch eine [`exec`](#)-Option wieder geändert werden.)
- `noauto`  
ist nur in `/etc/fstab` zweckmäßig und bewirkt dort, dass der in dieser Zeile genannte Datenträger beim Systemstart *nicht* automatisch eingebunden wird. Dennoch ist es sinnvoll, den Datenträger in `fstab` einzutragen, weil die Anwender nun bequem `mount name` durchführen können, ohne dabei alle weiteren `mount`-Optionen explizit anzugeben. `noauto` kommt beispielsweise für selten benötigte Datenpartitionen zum Einsatz.
- `nobootwait`  
hält den Bootprozess nicht auf, wenn das Device nicht zur Verfügung steht. Das Schlüsselwort wird unter Ubuntu von `mountall` ausgewertet und kann in anderen Distributionen nicht verwendet werden. Es bietet sich vor allem für externe Datenträger an, die nicht immer angeschlossen sind, z.B. für eine USB-Festplatte.
- `nofail`  
meldet keinen Fehler, wenn das Device gar nicht existiert. Die Option ist praktisch, wenn `mount` in einem Script versuchen soll, auf einen USB-Stick oder ein Netzwerkverzeichnis zuzugreifen.

- **optional**  
ignoriert den *fstab*-Eintrag, wenn der Dateisystemtyp während des Bootprozesses nicht zur Verfügung steht. Das Schlüsselwort wird unter Ubuntu von `mountall` ausgewertet und kann in anderen Distributionen nicht verwendet werden.
- **owner**  
erlaubt es jedem Benutzer, das betreffende Dateisystem selbst einzubinden bzw. wieder zu lösen. Die Option ist nur in */etc/fstab* zweckmäßig. Der Unterschied zu `user` besteht darin, dass `mount` bzw. `umount` nur dann ausgeführt werden darf, wenn der Benutzer Zugriffsrechte auf die betreffende Device-Datei hat (z.B. `/dev/fd0`).
- **remount**  
ändert die Optionen eines bereits in den Verzeichnisbaum eingebundenen Dateisystems. Diese Option kann nur bei einem direkten Aufruf von `mount` angegeben werden und ist in */etc/fstab* nicht zulässig.
- **ro / rw**  
gibt an, dass Dateien nur gelesen bzw. auch verändert werden dürfen. Normalerweise gilt für die meisten Dateisystemtypen `rw` (also *read-write*).
- **suid / nosuid**  
erlaubt bzw. verhindert die Auswertung des sid- und gid-Zugriffsbits. Diese Zugriffsbits erlauben es gewöhnlichen Benutzern, Programme mit `root`-Rechten auszuführen. Dies ist häufig ein Sicherheitsrisiko. Die Option `nosuid` verhindert die Auswertung derartiger Zugriffsbits und sollte für CDs, DVDs und externe Datenträger genutzt werden.
- **sync**  
bewirkt, dass Änderungen sofort gespeichert werden, anstatt sie für einige Sekunden im RAM zwischenspeichern und erst später auf den Datenträger zu übertragen. `sync` minimiert die

Gefahr des Datenverlusts, wenn Sie einen Datenträger (USB-Stick) irrtümlich ohne `umount` entnehmen bzw. die Kabelverbindung lösen. `sync` hat allerdings den Nachteil, dass das Schreiben von Daten je nach Datenträger sehr viel ineffizienter erfolgt. Insbesondere bei billigen USB-Sticks und traditionellen Festplatten sinkt die Geschwindigkeit um den Faktor zehn und mehr!

- `user`  
ermöglicht es normalen Benutzern, das betreffende Dateisystem mit `umount` ab- und mit `mount` wieder anzumelden. Dazu ist ansonsten nur `root` in der Lage. Datenträger dürfen nur von dem Benutzer wieder aus dem Dateisystem gelöst werden, der `mount` veranlasst hat. Die Option ist nur in `/etc/fstab` zweckmäßig und vor allem für externe Datenträger gedacht. Wenn Sie `user` verwenden, gelten automatisch auch die Optionen `noexec`, `nosuid` und `nodev`.
- `users`  
hat dieselbe Bedeutung wie `user`, aber mit einem kleinen Unterschied: Jeder Benutzer darf mit `users` gekennzeichnete Datenträger wieder aus dem Dateisystem entfernen (`umount`). `users` erlaubt also, dass der Benutzer A `mount` ausführt und der Benutzer B später `umount` aufruft.

## btrfs-Optionen

Viele `mount`-Optionen hängen vom jeweiligen Dateisystem ab. Im Folgenden sind daher die wichtigsten dateisystemspezifischen Optionen zusammengefasst.

Wir beginnen mit [btrfs](#).

- `compress=zlib|lzo`  
komprimiert alle neuen bzw. veränderten Dateien. Vorhandene Dateien bleiben unverändert, solange sie nur gelesen werden. Was die Kompressionsmethode betrifft, haben Sie die Wahl zwischen dem etwas langsameren, aber besser komprimierenden

`zlib`-Algorithmus und dem schnelleren, aber weniger platzsparenden `lzo`-Algorithmus. In Zukunft wird [`btrfs`](#) voraussichtlich weitere Kompressionsverfahren unterstützen. Beinahe fertig ist die Integration des `snappy`-Algorithmus, der bei ähnlichen Kompressionsraten etwas schneller als das `lzo`-Verfahren sein soll.

- `compress-force=zlib|lzo`  
komprimiert Dateien selbst dann, wenn dies nicht erfolgversprechend ist. Normalerweise erkennt das [`btrfs`](#)-Dateisystem bereits komprimierte Dateien und verzichtet darauf, diese nochmals zu komprimieren, weil dies in der Regel keine Platzersparnis mit sich bringt, aber eine Menge CPU-Zeit kostet. Die Verwendung von `compress-force` statt [`compress`](#) deaktiviert diese Erkennung.
- `degraded`  
bindet das Dateisystem selbst dann ein, wenn ein Teil der RAID-Devices fehlt. Das funktioniert nur, wenn aufgrund von RAID-Redundanzen alle Daten zur Verfügung stehen, also ein degraderter RAID-Verbund vorliegt.
- `noacl`  
deaktiviert die ACL-Zugriffsverwaltung.
- `nobarrier`  
deaktiviert die sogenannten *Write Barriers*. Eine Write Barrier blockiert weitere Schreibvorgänge, bis alle bisherigen Schreibvorgänge physikalisch abgeschlossen sind. Ohne Barriers ist das Dateisystem schneller, gleichzeitig steigt aber das Risiko eines Datenverlusts bei einem Crash, insbesondere bei Dateisystemen, die über mehrere Devices verteilt sind (RAID). Barriers funktionieren nur, wenn die Festplatte die Schreibaufforderung auch tatsächlich befolgt – was (aus Performance-Gründen) häufig nicht der Fall ist.

- `ssd`  
optimiert Dateioperationen für Solid State Disks (SSDs). Die Option ist automatisch aktiv, wenn der `btrfs`-Treiber feststellen kann, dass es sich beim Datenträger um eine SSD handelt. Die Option hat allerdings keinen Einfluss auf das Trim-Verhalten, das separat durch die Option `discard` gesteuert wird.
- `subvolume=name` oder `subvolid=n`  
verwendet das angegebene Subvolume (und nicht das Default-Subvolume). Die Volume-ID für `subvolid` können Sie mit `btrfs subvolume list` ermitteln.

## CIFS-Optionen

Mit dem *Common Internet File System* (CIFS) können Sie Windows-Netzwerkverzeichnisse in das Dateisystem integrieren. CIFS löst SMBFS ab. Damit Sie CIFS nutzen können, müssen Sie bei manchen Distributionen ein eigenes Paket installieren, z.B. `cifs-utils` oder `smbfs`.

Einige Worte zu den Zugriffsrechten: Wenn die Dateien von einem gewöhnlichen Windows-PC stammen, werden sie unter Linux dem Benutzer `root` zugeordnet. Alle Benutzer dürfen die Dateien lesen, aber nur `root` darf sie verändern. Wenn ein gewöhnlicher Benutzer Zugriff auf die Dateien bekommen soll, erreichen Sie das durch die korrekte Einstellung der Optionen `uid`, `gid`, `file_mode` und `dir_mode`. Wenn die Dateien hingegen von einem Datei-Server stammen, der die CIFS-Unix-Erweiterung unterstützt (das ist der Fall, wenn Samba auf einem Linux-Rechner läuft), dann werden die UIDs, GIDs und Zugriffsbits der Dateien unverändert weitergeleitet. Die Optionen `uid`, `gid`, `file_mode` und `dir_mode` bleiben dann wirkungslos, es sei denn, Sie deaktivieren die Unix-Erweiterungen mit der Option `nounix`.

Weitere Details zu den CIFS-spezifischen `mount`-Optionen gibt `man mount.cifs`.

- `credentials=datei`  
gibt eine Datei an, die den Benutzernamen und das Passwort für den Login beim Windows- bzw. Samba-Server enthält. Damit vermeiden Sie die Angabe von Passwörtern in `/etc/fstab`. Die Datei enthält drei Zeilen mit `username=xxx` (nicht `user=xxx!`), `password=xxx` und bei Bedarf `domain=xxx`.
- `dir_mode=n` (`dmask` bei SMBFS)  
gibt an, welche Zugriffsrechte für den Zugriff auf Verzeichnisse gelten. `dir_mode=0770` gibt dem durch `uid` angegebenen Benutzer sowie allen Mitgliedern der Gruppe `gid` uneingeschränkte Rechte (Verzeichnis lesen, öffnen und ändern).
- `domain=workgroup`  
gibt die Domäne oder Arbeitsgruppe an.
- `file_mode=n` (`fmask` bei SMBFS)  
gibt an, welche Zugriffsrechte für den Dateizugriff gelten. `n` muss mit einer Null beginnen und enthält die Rechte in der von [chmod](#) vertrauten Oktalschreibweise. `file_mode=0444` gibt allen Benutzern Lese- und Schreibrechte. Die Einstellung wird ignoriert, wenn der Datei-Server die CIFS-Unix-Erweiterung unterstützt.
- `iocharset=name`  
gibt den gewünschten Zeichensatz an. Wenn deutsche Sonderzeichen oder andere Nicht-ASCII-Zeichen in Dateinamen falsch angezeigt werden, hilft in der Regel `iocharset=utf8`.
- `nodfs`  
deaktiviert die DFS-Funktionen. DFS steht für *Distributed File System* und erlaubt es, auf die Netzwerkverzeichnisse unterschiedlicher Server einheitlich zuzugreifen. Einige alte NAS-Festplatten verwenden ebenso alte Samba-Versionen mit bekannten DFS-Fehlern, die insbesondere beim Einsatz des

Kommandos [rsync](#) zu Problemen führen. Abhilfe schafft in solchen Fällen die `nodfs`-Option.

- `nounix`

deaktiviert den Unix-Kompatibilitätsmodus. Zum Verständnis dieser Option sind ein paar Hintergrundinformationen unumgänglich. Der CIFS-Standard wurde nämlich um die sogenannten *Unix Extensions* erweitert, um eine bessere Kompatibilität mit Unix/Linux-Systemen zu erzielen. Bei manchen NAS-Geräten sind diese Extensions auch aktiv. In diesem Fall übernimmt `mount` die Benutzer- und Gruppeninformationen sowie die Zugriffsrechte vom NAS-Gerät und ignoriert die Optionen `uid`, `gid`, `dir_mode` und `file_mode`. Das klingt auf den ersten Blick praktisch, ist es aber zumeist nicht: Die Unix Extensions ergeben nur dann einen Sinn, wenn die UIDs und GIDs des lokalen Rechners und des Samba-Servers abgestimmt sind. Das ist selten der Fall, bei NAS-Geräten praktisch nie. Daher muss der Kompatibilitätsmodus deaktiviert werden, damit `uid`, `gid`, `dir_mode` und `file_mode` wirksam bleiben. Mitunter können Sie mit `nounix` auch Inkompatibilitäten aus dem Weg gehen, die sich in der vagen Fehlermeldung *operation not supported* äußern.

- `password=pw`

gibt das Passwort für die Authentifizierung beim Windows- bzw. Samba-Server an. Die direkte Passwortangabe ist unsicher! Besser ist es, das Passwort in einer Datei anzugeben und auf diese mit dem Schlüsselwort `credentials` zu verweisen.

- `sec=modus`

gibt den Authentifizierungsmodus an. Dieser Modus steuert die Authentifizierung beim Samba-Server. Die Defaulteinstellung lautet `ntlmssp`. Dieses Schlüsselwort bezeichnet ein Windows-typisches Verfahren mit Passwort-Hashing, wobei die Daten in eine NTLMSSP-Nachricht verpackt werden. Alternative

Einstellungen wie `ntlmssp` oder `krb5i` (Kerberos) beschreibt man `mount.cifs`.

- `uid=u,gid=g`  
geben wie beim `vfat`-Treiber den Besitzer und die Gruppe der Dateien an. An die Optionen können statt der UID/GID-Nummern auch die Benutzer- bzw. Gruppennamen übergeben werden.
- `user=name` (`username` bei SMBFS)  
gibt den Benutzernamen für die Authentifizierung beim Windows- bzw. Samba-Server an. An die Option können gleichzeitig auch der Arbeitsgruppenname und das Passwort übergeben werden (`user=workgroup/name%password`).
- `vers=n.n`  
gibt an, welche Version des SMB-Protokolls verwendet werden soll. Aktuelle `mount`-Versionen verwenden standardmäßig die Version 3, weil ältere Versionen bekannte Sicherheitsprobleme aufweisen. Um mit einem alten Server zu kommunizieren, müssen Sie die gewünschte Version dann explizit angeben, z.B. mit `vers=2.1`.

## exFAT-Optionen

Das exFAT-Dateisystem (-t `exfat`) unterstützt die Optionen `dmask`, `fmask`, `gid`, `noatime`, `ro` (Read-only) `uid` und `umask`. Die Optionen haben dieselbe Wirkung wie die gleichnamigen VFAT-Optionen, die einige Seiten weiter beschrieben werden. Einzig `noatime` steht bei VFAT-Dateisystemen nicht zur Verfügung. Diese Option verhindert, dass bei Lesezugriffen die Access-Time-Information aktualisiert wird, und verringert so die Anzahl der Schreiboperationen.

## ext3/ext4-Optionen

Die folgenden Optionen gelten für das Linux-Standarddateisystem `ext3` bzw. `ext4`:

- **acl**  
aktiviert die ACL-Zugriffsverwaltung. Das erlaubt die Speicherung von zusätzlichen Zugriffsinformationen in Form von *Access Control Lists*. Die Option ist insbesondere beim Einsatz von Samba in heterogenen Netzwerken sinnvoll.
- **barrier=0 (nur für ext4)**  
deaktiviert die sogenannten *Write Barriers*, die sicherstellen, dass Änderungen in der richtigen Reihenfolge auf die Festplatte geschrieben werden. Das erhöht die Geschwindigkeit des Dateisystems, ist aber nur sicher, wenn der Festplatten-Cache durch eine Batterie gegen Stromausfälle und Abstürze abgesichert ist.
- **commit=n**  
synchronisiert das Journal alle  $n$  Sekunden (standardmäßig alle 5 Sekunden). Das Paket `laptop-mode` vergrößert diese Zeitspanne bei Notebooks im Batteriebetrieb, um Energie zu sparen. Bei ext3 mit `data=ordered` werden gleichzeitig mit dem Journal auch alle Dateiänderungen synchronisiert. Bei ext3 mit `data=writeback` und bei ext4 mit *Delayed Allocation* gilt das `commit`-Intervall hingegen nur für das Journal.
- **data=journal/ordered/writeback**  
bestimmt den Journaling-Modus (Details dazu folgen gleich). Standardmäßig gilt für ext4 der Modus `ordered`. Für ext3 gilt standardmäßig für die Kernelversionen 2.6.30 bis 2.6.35 der Modus `writeback`, für alle Versionen davor und danach der sichere, aber etwas langsamere Modus `ordered`.
- **errors=continue/remount-ro/panic**  
bestimmt das Verhalten, wenn bei der Überprüfung des Dateisystems ein Fehler auftritt. Das Defaultverhalten kann mithilfe des Befehls [`tune2fs`](#) mit der Option `-e` eingestellt werden.

- `grpid`  
bewirkt, dass neue Dateien automatisch die Gruppen-ID des Verzeichnisses übernehmen, in dem sie erzeugt wurden. Damit verhält sich das Dateisystem wie unter BSD bzw. so, als wäre in allen Verzeichnissen das SetGID-Bit gesetzt. (Standardmäßig ordnet Linux neuen Dateien die Gruppen-ID des Benutzers zu, der die Datei erzeugt.)
- `nodealloc` (nur für ext4)  
deaktiviert die *Delayed Allocation*.
- `noload`  
ignoriert die vorhandene Journaling-Datei beim `mount`-Kommando. Das kann zweckmäßig sein, wenn die Journaling-Datei defekt ist.
- `sb=n`  
verwendet Block  $n$  als Superblock (statt standardmäßig Block 1). Damit gelingt es in manchen Fällen, ein beschädigtes Dateisystem noch zu lesen. Normalerweise erstellt `mke2fs` alle 8192 Blöcke eine Kopie des Superblocks. Daher empfiehlt es sich, für  $n$  die Werte 8193, 16.385 etc. einzusetzen.
- `user_xattr`  
aktiviert die Unterstützung für erweiterte Dateiattribute.

### ext3/ext4-Journaling-Modi

Das ext-Dateisystem kennt drei Verfahren, wie das Journaling durchgeführt wird:

- `data=ordered`  
Bei diesem Modus werden im Journal nur Metadaten gespeichert, also Informationen *über* Dateien, keine Inhalte. Im Journal werden Dateien erst dann als *committed* gekennzeichnet, wenn sie vollständig gespeichert wurden. Nach einem Crash kann das Dateisystem rasch wieder in einen konsistenten Zustand gebracht werden, weil unvollständig gespeicherte Dateien

anhand des Journals sofort erkannt werden. Es ist aber unmöglich, solche Dateien wiederherzustellen.

Im Modus `data=ordered` wird das Journal alle fünf Sekunden mit der Festplatte synchronisiert. Bei `ext3` hat das zur Folge, dass sämtliche Änderungen an irgendwelchen Dateien innerhalb von fünf Sekunden physikalisch auf der Festplatte gespeichert werden. Dieses Standardverhalten ist zwar nicht besonders effizient, dafür aber sehr sicher: Selbst bei Totalabstürzen und Stromausfällen sind massive Datenverluste äußerst selten.

`data=ordered` hat bei `ext3` aber eine unerfreuliche Nebenwirkung: Bei jedem Aufruf der `fsync`-Funktion wird nicht nur eine bestimmte Datei, sondern das gesamte Dateisystem synchronisiert. Das kann zu spürbaren Verzögerungen führen. Bei `ext4` wird das Journal zwar ebenfalls alle fünf Sekunden synchronisiert, die eigentlichen Datenänderungen werden aber aufgrund der *Delayed Allocation* oft erst viel später gespeichert. Nur ein expliziter Aufruf der `fsync`-Funktion stellt die sofortige physikalische Speicherung einer Datei sicher! Glücklicherweise erfordert `fsync` bei `ext4` nicht, dass das gesamte Dateisystem synchronisiert werden muss. Die Funktion wird daher wesentlich schneller ausgeführt.

- **`data=writeback`**

Dieser Modus ähnelt dem `ordered`-Modus. Der einzige Unterschied besteht darin, dass das Journal und die Dateioperationen nicht immer vollständig synchron sind. Das Dateisystem wartet mit den *committed*-Einträgen im Journal nicht auf den Abschluss der Speicheroperation auf der Festplatte. Damit ist das Dateisystem etwas schneller als im `ordered`-Modus. Nach einem Crash ist die Integrität des Dateisystems weiterhin sichergestellt. Allerdings kann es vorkommen, dass veränderte Dateien alte Daten enthalten. Dieses Problem tritt nicht auf, wenn Anwendungsprogramme -- wie im POSIX-Standard

vorgesehen – den Speichervorgang mit `fsync` abschließen (siehe dort).

- `data=journal`

Anders als in den beiden anderen Modi werden jetzt im Journal auch die tatsächlichen Daten gespeichert. Dadurch müssen alle Änderungen *zweimal* gespeichert werden (zuerst in das Journal und dann in die betroffene Datei). Deswegen ist `ext3` in diesem Modus deutlich langsamer. Dafür können nach einem Crash Dateien wiederhergestellt werden, deren Änderungen bereits vollständig in das Journal (aber noch nicht in die Datei) eingetragen worden sind.

Grundsätzlich wird das Journal alle fünf Sekunden physikalisch auf der Festplatte gespeichert. Diese Zeitspanne kann durch die `mount`-Option `commit` verändert werden. Intern kümmert sich der in den Kernel integrierte Journaling-Dämon `kjournald` um die regelmäßige Aktualisierung der Journaling-Datei.

### **ISO-9660-Optionen (Daten-CDs/DVDs)**

Daten-CDs und viele DVDs verwenden das Dateisystem ISO 9660 (`-t iso9660`).

- `iocharset=name`

gibt den Zeichensatz an, mit dem Joliet-Dateinamen unter Linux verarbeitet werden sollen (standardmäßig ISO 8859-1). Die Option ist nur bei CDs sinnvoll, die die Joliet-Erweiterung nutzen.

- `norock, nojoliet`

deaktiviert die Rock-Ridge- bzw. die Joilet-Erweiterung. (Der ISO-9660-Treiber erkennt und berücksichtigt diese Erweiterungen automatisch.)

- `session=n`

gibt an, welche Session gelesen werden soll (nur bei Multi-Session-CDs).

- `uid=u, gid=g`

geben den Besitzer und die Gruppe der Dateien an

(standardmäßig jeweils root). Die Zugriffsbits können nicht eingestellt werden.

- `utf8`  
verwendet Linux-seitig den Zeichensatz UTF8, um Joliet-Dateinamen zu verarbeiten. Auch diese Option ist nur bei Joliet-CDs sinnvoll.

## NFS-Optionen

Das Dateisystem NFS (`-t nfs`) ist der übliche Weg zur Freigabe bzw. Nutzung von Netzwerkverzeichnissen unter Unix/Linux. Beachten Sie, dass sich `-t nfs` auf die NFS-Versionen 2 und 3 bezieht. Wenn Sie NFS4 nutzen möchten, müssen Sie im `mount`-Kommando `-t nfs4` angeben.

Die meisten NFS-Optionen gelten gleichermaßen für alle NFS-Versionen. Eine umfassende Zusammenfassung aller Optionen inklusive einer Differenzierung zwischen NFS2/3 und NFS4 gibt `man nfs`.

- `bg`  
bewirkt, dass der `mount`-Prozess im Hintergrund fortgesetzt wird, wenn der NFS-Server nicht erreichbar ist. Diese Option ist nur für `/etc/fstab` relevant.
- `hard`  
bewirkt, dass ein Programm, das auf eine NFS-Datei zugreift, hängen bleibt, wenn der NFS-Server nicht mehr zur Verfügung steht. (Die Alternative zu `hard` ist `soft`. Sie bewirkt, dass der Kernel es nach einer Weile aufgibt, die NFS-Datei wiederzufinden. Das klingt zwar sicherer, verursacht aber in der Realität noch mehr Probleme als `hard`.)
- `intr`  
ermöglicht es, ein Programm auch dann durch `kill` bzw. `[Strg]+[C]` zu stoppen, wenn eine offene NFS-Datei nicht mehr zur

Verfügung steht. Die Option ist nur in Kombination mit `hard` gültig.

- `nfsvers=2/3`  
gibt die gewünschte NFS-Version an (nur für `-t nfs`).
- `rsize=n, wsize=n`  
gibt die Größe der Puffer für Lese- und Schreiboperationen an (in KiB). Der Standardwert lautet jeweils 4096.

Die richtigen `mount`-Optionen entscheiden maßgeblich über die mit NFS erzielbare Geschwindigkeit! Eine hohe Geschwindigkeit erzielen Sie mit den Optionen `hard, intr, rsize=8192, wsize=8192`. Wenn die NFS-Geschwindigkeit dennoch unter Ihren Erwartungen liegt, sollten Sie sicherstellen, dass `/etc(exports` auf dem NFS-Server die Option `async` verwendet.

## NTFS-Optionen

Die folgenden Optionen gelten für den `ntfs3g`-Treiber, der bei den meisten Distributionen für den Zugriff auf NTFS-Dateisysteme zum Einsatz kommt.

- `locale=name`  
gibt den lokalen Zeichensatz an (beispielsweise in der Form `locale=de_DE.UTF-8`). Die Option ist nur in `/etc/fstab` notwendig, wenn der Zeichensatz beim automatischen Einbinden der Dateisysteme durch den Init-V-Prozess noch nicht konfiguriert ist.
- `show_sys_files`  
zeigt auch NTFS-Systemdateien an.
- `streams_interface=none|windows|xattr`  
gibt an, wie der Zugriff auf Datei-Streams erfolgen soll: In der Standardeinstellung `xattr` erfolgt der Zugriff über die Kommandos `getfattr` bzw. `setfattr`. `windows` erlaubt den Zugriff in der unter Windows üblichen Schreibweise `dateiname:streamname`. Die Standardeinstellung lautet `none`.

- `uid=u, gid=g, umask=m | fmask=f, dmask=d`  
 haben dieselbe Bedeutung wie beim `vfat`-Treiber (siehe unten).  
 Als Dateibesitzer gilt der Benutzer, der `mount` ausgeführt hat. Alle Benutzer haben Lese- und Schreibzugriff!

## UDF-Optionen (DVDs)

Auf DVDs kommen zumeist die Dateisysteme ISO 9660 (siehe oben) oder UDF (-t `udf`) zum Einsatz. UDF ist auch das optimale Dateisystem für DVD-RAMs. UDF verwendet überwiegend dieselben Optionen wie ISO 9660. Im Folgenden sind deswegen nur zwei UDF-spezifische Optionen beschrieben:

- `undelete`  
 gibt Zugriff auf gelöschte Dateien.
- `unhide`  
 zeigt auch Dateien, die eigentlich versteckt sind.

## VFAT-Optionen

Mit `mount -t vfat` binden Sie Windows-VFAT-Partitionen in allen Varianten (FAT12, FAT16, FAT32) in den Verzeichnisbaum ein. Zahlreiche Optionen steuern, wer welche Zugriffsrechte auf die Dateien hat, wie die unterschiedlichen Zeichensätze in den Dateinamen verarbeitet werden etc.

- `codepage=name`  
 gibt die Codeseite an (eine Art DOS-Zeichensatz), die VFAT-intern für kurze Dateinamen gilt (maximal 8+3 Zeichen). Lange Dateinamen werden auf jeden Fall im Unicode-Format gespeichert; für sie ist diese Option irrelevant. Aktuelle Windows-Versionen speichern auch kurze Dateinamen im VFAT-Format für lange Dateinamen, um so die korrekte Groß- und Kleinschreibung zu erhalten. Aus diesem Grund ist die Option zunehmend unwichtig. Der Standardwert der Option kann beim Kompilieren bestimmt werden und beträgt zumeist `cp437`.
- `flush`  
 bewirkt, dass der Treiber sofort damit beginnt,

zwischengespeicherte Daten auf dem Datenträger zu speichern. Die Option minimiert bei externen Datenträgern (z.B. einem USB-Memorystick) die Zeit zum Speichern veränderter Daten. Im Gegensatz zur Option [sync](#) wartet das Dateisystem nicht auf den Abschluss der Synchronisation. Deswegen hat `flush` keinen negativen Einfluss auf die Geschwindigkeit von Schreiboperationen.

- `fmask=f, dmask=d`

stellt die Maske für Dateien (`fmask`) und Verzeichnisse (`dmask`) trennt ein. Das ist vor allem deswegen praktisch, weil Sie oft für Verzeichnisse das x-Zugriffsbit setzen möchten, für Dateien aber nicht. Damit kann der Benutzer in alle Verzeichnisse wechseln, aber keine Dateien bzw. Programme ausführen. Eine mögliche Einstellung ist `fmask=177, dmask=077`: Damit kann der Besitzer (uid) Dateien lesen und schreiben (w-----) sowie in Verzeichnisse wechseln (rwx-----). Alle anderen Benutzer haben keinerlei Zugriff auf die Dateien. Ebenfalls oft zweckmäßig ist `fmask=133, dmask=022`: Damit gilt für Dateien rw-r--r-- und für Verzeichnisse rwxr-xr-x, d.h., jeder darf alles lesen, aber nur der Besitzer darf etwas verändern.

- `iocharset`

gibt den Zeichensatz an, mit dem Windows-Dateinamen unter Linux verarbeitet werden sollen. Die Standardeinstellung lautet `iso8859-1`. Die Option kommt in der Regel nur dann zum Einsatz, wenn Sie unter Linux mit einem 8-Bit-Zeichensatz arbeiten. (VFAT-intern wird für lange Dateinamen in jedem Fall der Unicode-Zeichensatz verwendet. Die Option bestimmt nur, wie die Dateinamen unter Linux dargestellt werden.) Wenn Sie unter Linux mit dem Zeichensatz UTF8 arbeiten, sollten Sie statt `iocharset` die im Folgenden beschriebene Option `utf8` verwenden! `iocharset=utf8` funktioniert zwar auf den ersten Blick ebenfalls, ist aber nicht empfehlenswert: Der VFAT-Treiber

beachtet dann die exakte Groß- und Kleinschreibung, was den Windows-Gepflogenheiten widerspricht.

- `shortname=lower|win95|winnt|mixed`  
gibt an, in welcher Groß- und Kleinschreibung kurze Dateinamen (maximal 8+3 Zeichen) präsentiert bzw. gespeichert werden. Das diesbezügliche Verhalten variierte unter Windows je nach Version. Unter Linux gilt standardmäßig die Einstellung `lower`, d.h., kurze Dateinamen werden in Kleinbuchstaben präsentiert. Bei neuen Dateien wird der Name im VFAT-Format für lange Dateinamen gespeichert (auch wenn die DOS-Grenze von 8+3-Zeichen gar nicht überschritten wird), sodass die Groß- und Kleinschreibung erhalten bleibt. Ausgenommen von dieser Regel sind nur kurze Dateinamen, die ausschließlich aus Großbuchstaben bestehen; derartige Namen werden DOS-typisch als kurze Dateinamen gespeichert. Die Einstellung hat keinerlei Einfluss auf lange Dateinamen, deren Groß- und Kleinschreibung nach dem Erzeugen der Datei auf jeden Fall erhalten bleibt.

- `uid=u, gid=g, umask=m`  
bestimmen, wem (`uid`) und welcher Gruppe (`gid`) die Windows-Dateien »gehören«, d.h., wer die Dateien lesen bzw. verändern darf. Üblicherweise geben Sie mit `uid` und `gid` die gewünschte Benutzer- bzw. Gruppennummer an. An [`umask`](#) übergeben Sie einen oktalen Zahlenwert mit der Bitmaske der *invertierten* Zugriffsbits. `umask=0` bedeutet daher, dass jeder alle Dateien lesen, schreiben und ausführen darf (`rwxrwxrwx`). `umask=022` entspricht `rwxr-xr-x`: Alle dürfen alle Dateien lesen, aber nur der Besitzer darf Dateien ändern.

Wenn diese Optionen nicht angegeben werden, gelten für das `vfat`-Dateisystem die folgenden Standardwerte:

- `uid`: UID des Benutzers, der `mount` ausführt. Normalerweise darf nur `root` das Kommando `mount` ausführen, es sei denn, die

entsprechende Zeile in `/etc/fstab` enthält die Optionen `user`, `users` oder `owner`.

- `gid`: GID des Benutzers, der `mount` ausführt.
- `umask`: 022, d.h., alle dürfen alles lesen, aber nur der Besitzer darf Dateien bzw. Verzeichnisse verändern.

Oft enthält `fstab` die Einstellung `gid=users`. Damit gilt für `gid` die Standardgruppe des Benutzers, der `mount` ausführt. Ohne diese Option ist es möglich, zuerst mit `newgrp` eine andere Gruppe zu aktivieren, die dann auch für `mount` gilt.

- `utf8`

bewirkt, dass der Treiber die Dateinamen als UTF8-Zeichenketten an Linux zurückgibt. Die Option ist zur korrekten Darstellung von Nicht-ASCII-Zeichen erforderlich, wenn Sie unter Linux als Zeichensatz UTF8 verwenden.

## XFS-Optionen

Die meisten Optionen für das `xfs`-Dateisystem betreffen selten benutzte Funktionen (siehe `man mount`). ACLs und *Extended Attributes* stehen beim `xfs`-Dateisystem standardmäßig zur Verfügung und werden nicht durch Optionen gesteuert.

- `logdev=device`

gibt die Partition an, in der die Journaling-Daten gespeichert sind. Die Option ist nur notwendig, wenn beim Einrichten des Dateisystems eine externe Journaling-Partition angegeben wurde.

- `norecovery`

verhindert die Auswertung der Journaling-Daten beim Einbinden des Dateisystems. Gleichzeitig müssen Sie die Option `ro` verwenden (*Read-only*-Zugriff).

## Beispiele

Linux-Partitionen können Sie in der Regel ohne explizite Angabe des Dateisystems (also ohne die Option `-t`) in das Dateisystem integrieren:

```
root# mkdir /windows  
root# mount /dev/sda7 /media/backup-partition
```

Das folgende Beispiel zeigt den Zugriff auf die Daten eines USB-Sticks über das Verzeichnis */windows*:

```
root# mkdir /usbstick  
root# mount -t vfat /dev/sdc1 /usbstick
```

Damit nicht nur root, sondern auch ein gewöhnlicher Benutzer mit der User-ID 1000 und der Gruppen-ID 1000 Zugriff auf den USB-Stick hat und Dateien schreiben oder verändern kann, sind mehr Optionen erforderlich:

```
root# mount -t vfat -o uid=1000,gid=1000,fmask=0022,dmask=0022 \  
      /dev/sdc1 /usbstick
```

Das nächste Kommando bindet das CD-ROM-Laufwerk mit einer Daten-CD (ISO-9660-Dateisystem) im Verzeichnis */media/cdrom* in das Dateisystem ein:

```
root# mount -t iso9660 /dev/sr0 /media/cdrom
```

Das folgende Kommando bindet ein Netzwerkverzeichnis eines Windows- oder Samba-Servers, z.B. von einem NAS-Gerät, in den lokalen Verzeichnisbaum ein. Der Benutzer mit der UID/GID 1000 kann die Daten lesen und verändern.

```
root# mount -t cifs -o username=name,uid=1000,gid=1000,iocharset=utf8,nounix \  
      //jupiter/myshare /media/winshare  
Password for name: *****
```

Mit `mount -o remount` können Sie Optionen eines bereits eingebundenen Dateisystems verändern. Das folgende Kommando aktiviert beispielsweise die [exec](#)-Option für eine DVD, sodass darauf enthaltene Programme ausgeführt werden können:

```
root# mount /media/dvd -o remount,exec
```

Falls beim Einbinden der Systempartition während des Rechnerstarts Probleme auftreten, wird die Partition nur read-only eingebunden. Um die Fehlerursache – etwa einen falschen Eintrag in */etc/fstab* – zu beheben, ist es aber oft erforderlich, Änderungen im Dateisystem durchzuführen. Dazu führen Sie das folgende Kommando aus. Mit ihm wird die Systempartition neu eingebunden, wobei jetzt auch Schreibzugriffe möglich sind.

```
root# mount -o remount,rw /
```

## **mpage** [optionen] quelldateien > zieldatei

**mpage** wandelt eine Textdatei in das PostScript-Format um. **mpage** kann außerdem vorhandene PostScript-Dateien platzsparend neu formatieren und anschließend beispielsweise mit zwei Seiten pro Blatt ausdrucken. Das funktioniert allerdings nicht mit allen PostScript-Dateien. **mpage** ist leider nicht Unicode-kompatibel.

- **-1        -2        -4        -8**  
ordnet eine, zwei, vier oder acht Seiten auf einem Blatt an (normalerweise vier Seiten). Bei zwei oder acht wird das Querformat verwendet.
- **-bA4**  
Ausgabe im DIN-A4-Format (normalerweise US-Letter).
- **-C ISO-Latin.1**  
verwendet die ISO-Latin-1-Kodierung für den Zeichensatz. Die Option ist für den Ausdruck von Texten mit deutschen Sonderzeichen erforderlich. Andere Kodierungstabellen finden Sie im Verzeichnis */usr/lib/mpage*.
- **-f**  
verteilt lange Zeilen auf mehrere Zeilen (statt sie abzuschneiden).
- **-j n-m**  
druckt nur Blatt *n* bis *m* (normalerweise alle).
- **-l**  
druckt im Querformat mit 55 Zeilen und 132 Spalten und einer entsprechend geringeren Schriftgröße (normalerweise Hochformat mit 66 Zeilen und 80 Spalten).
- **-L n**  
druckt *n* Zeilen pro Seite und passt die Zeichensatzgröße entsprechend an (normalerweise 66 Zeilen).
- **-o**  
verzichtet auf Rahmenlinien um die Seiten.

- **-w n**  
druckt *n* Spalten pro Seite und passt die Zeichensatzgröße entsprechend an (normalerweise 80 Spalten).
- mtr [optionen] hostname**
- Das Kommando **mtr** sendet regelmäßig Netzwerkpakete zum angegebenen Host und analysiert die Antworten. Die Ergebnisliste kombiniert Daten von [ping](#) und [traceroute](#). Beachten Sie, dass es zwei Versionen dieses Programms gibt: das hier beschriebene Textkommando und eine GTK-Variante mit grafischer Benutzeroberfläche. Bei Desktop-Installationen von Debian und Ubuntu ist standardmäßig die GTK-Variante installiert. Um stattdessen die Textversion zu installieren, führen Sie `apt-get install mtr-tiny` aus.
- **-c n**  
führt *n* Tests aus. (Standardmäßig läuft das Programm, bis es mit **[Strg]+[C]** beendet wird.)
  - **-n bzw. --no-dns**  
verzichtet auf die Namensauflösung und gibt alle Zwischenstationen in Form von IP-Adressen an.
  - **-r bzw. --report**  
liefert nach Abschluss des Tests einen Bericht in Textform. Diese Option muss mit -c verbunden werden.

## Beispiel

Die folgenden Zeilen zeigen ein typisches **mtr**-Ergebnis, wobei die Verbindung zwischen einem lokalen ADSL-Anschluss und [google.de](#) getestet wurde:

```
user$ mtr -c 10 -r google.de
HOST: michaels-computer          Loss%   Snt   Last    Avg   Best Wrst StDev
1. |-- speedtouch.lan           0.0%    10   42.6  48.5   6.0  95.9  28.9
2. |-- 178-191-207-254.adsl.high 0.0%    10   18.9  20.4  18.6  23.2   1.9
3. |-- 195.3.74.129             0.0%    10   19.4  18.8  17.9  19.4   0.5
4. |-- AUX10-GRAZBC10.highway.te 0.0%    10   21.2  21.3  20.7  22.0   0.3
5. |-- 195.3.70.154              0.0%    10   21.2  27.3  20.9  81.2  18.9
...
```

```
8.|-- 216.239.46.88      0.0%   10   25.8  26.3  25.8  27.7  0.6  
9.|-- bud01s08-in-f23.1e100.net 0.0%   10   25.7  25.8  25.0  26.8  0.5
```

### **multitail [optionen] datei1 [datei2 datei3 ...]**

`multitail` hilft dabei, mehrere Logging-Dateien gleichzeitig zu überwachen. In der einfachsten Form führen Sie `multitail datei1 datei2 datei3 ...` aus. Damit wird das Terminal in mehrere gleich große Bereiche geteilt, die jeweils die letzten Zeilen der angegebenen Dateien zeigen. Wenn Sie den Dateien `-I` voranstellen, werden Änderungen mehrerer Dateien in einem Bereich gemeinsam ausgegeben. Das hat allerdings den Nachteil, dass schwer nachzuvollziehen ist, aus welcher Datei welche Änderungen stammen.

### **mv quelle ziel** **mv dateien zielverzeichnis**

`mv` benennt eine Datei oder ein Verzeichnis um bzw. verschiebt (eine oder mehrere) Dateien in ein anderes Verzeichnis.

- **-b bzw. --backup**  
benennt bereits vorhandene gleichnamige Dateien in Backup-Dateien (Name plus ~-Zeichen) um, anstatt sie zu überschreiben.
- **-i bzw. --interactive**  
fragt, bevor vorhandene Dateien überschrieben werden.

### **Beispiel**

Das folgende Kommando verschiebt alle PDF-Dateien im aktuellen Verzeichnis in das Unterverzeichnis `pdf`:

```
user$ mv *.pdf pdf/
```

`mv` kann nicht dazu verwendet werden, mehrere Dateien umzubenennen. `mv * .xxx * .yyy` funktioniert also nicht. Um solche Operationen durchzuführen, müssen Sie [for](#) oder [sed](#) zu Hilfe nehmen. Entsprechende Beispiele finden Sie bei [cp](#).

### **mysql [optionen] [datenbankname] [ < name.sql]**

Der Kommandointerpreter `mysql` führt SQL-Kommandos auf einem MySQL- oder MariaDB-Server aus, wahlweise interaktiv oder im Batch-Betrieb aus der Standardeingabe. Die Kommandos müssen

mit einem Strichpunkt abgeschlossen werden. In der Batch-Variante wird `mysql` häufig zum Einlesen von Backups verwendet, die mit [`mysqldump`](#) erstellt wurden. Wenn Sie beim Aufruf von `mysql` keinen Datenbanknamen angeben, müssen Sie die gewünschte Datenbank mit dem Kommando `USE` einstellen.

Die folgenden Optionen betreffen lediglich den Verbindungsauflauf und gelten in gleicher Form auch für die im Folgenden beschriebenen Kommandos [`mysqladmin`](#) und [`mysqldump`](#). In der Praxis sind zumeist nur die Optionen `-u`, `-p` und `-h` erforderlich.

- `--default-character-set=name`  
gibt an, welcher Zeichensatz für die Kommunikation mit dem MySQL- oder MariaDB-Server verwendet werden soll (z.B. `latin1` oder `utf8`).
- `--defaults-extra-file=dateiname`  
lädt Verbindungsoptionen aus der angegebenen Datei. Das ist beispielsweise zweckmäßig, wenn `mysql` oder [`mysqldump`](#) regelmäßig durch Cron ausgeführt werden soll und eine interaktive Passworteingabe unmöglich ist.
- `-h bzw. --host=hostname`  
gibt den Namen oder die IP-Nummer des Rechners an, auf dem der Server läuft (standardmäßig `localhost`).
- `--login-path=gruppe`  
wertet die Daten von [gruppe] in `.mylogin.cnf` aus.
- `-P bzw. --port=n`  
gibt die Port-Nummer für die TCP/IP-Verbindung an (standardmäßig 3306).
- `-p bzw. --password`  
fordert unmittelbar nach dem Start des Kommandos zur Eingabe des Passworts auf. Wenn die Option fehlt, liest `mysql` das Passwort aus der lokalen Konfigurationsdatei `.my.cnf`. Wenn diese Datei

nicht existiert oder keine Passwortdaten enthält, versucht `mysql` den Login ohne Passwort.

- `-pxxx` bzw. `--password=xxx`  
übergibt das Passwort direkt. Das ist allerdings unsicher, denn das Passwort erscheint in der Prozessliste im Klartext! Bei der Option `-p` darf im Gegensatz zu anderen Optionen kein Leerraum angegeben werden.
  - `--protocol=tcp/socket`  
gibt an, welches Protokoll zur Kommunikation zwischen dem MySQL-Client und dem Datenbank-Server verwendet werden soll. Standardmäßig gilt `socket`, wenn Client und Server auf demselben Rechner laufen, sonst `tcp`.
  - `-s` bzw. `--socket=name`  
gibt den Ort der Socket-Datei an. Diese Option ist nur erforderlich, wenn `mysql` die Socket-Datei nicht selbstständig findet. Die Datei hat bei den meisten Distributionen den Namen `/var/run/mysqld/mysqld.sock`. Ihr Ort kann durch die Konfigurationsdatei `/etc/my.cnf` bzw. `/etc/mysql/my.cnf` eingestellt werden.
  - `-u` bzw. `--user=name`  
gibt den MySQL- oder MariaDB-Benutzernamen an (standardmäßig den aktuellen Benutzernamen).
- Außerdem gibt es einige `mysql`-spezifische Optionen:
- `-B` bzw. `--batch`  
trennt bei der Ausgabe der Ergebnisse die Spalten durch Tabulatorzeichen (statt durch Leerzeichen und Liniengrafik). Außerdem werden nur die Ergebnisse von Abfragen angezeigt, aber keinerlei Statusinformationen.
  - `-e` bzw. `--execute='sql-kommando'`  
führt das angegebene Kommando aus. Es dürfen so auch

mehrere, durch Strichpunkte getrennte Kommandos angegeben werden.

- **-h bzw. --html**  
formatiert das Abfrageergebnis als HTML-Code.
- **-N bzw. --skip-column-names**  
verzichtet auf die Beschriftung der Ergebnisspalten.
- **-r bzw. --raw**  
gibt die Zeichen O, Tabulator, Newline und \ in Abfrageergebnissen unverändert aus. (Normalerweise werden diese Zeichen als \0, \t, \n und \\ ausgegeben.) Die Option ist nur in Kombination mit --batch wirksam.
- **-s bzw. --silent**  
gibt nur das Ergebnis an sich aus.
- **-x bzw. --xml**  
formatiert das Abfrageergebnis als XML-Code.

## Beispiel

Das folgende Kommando spielt ein Datenbank-Backup auf dem lokalen MySQL- oder MariaDB-Server ein. Die Datenbank dbname muss bereits existieren.

```
user$ mysql -u name -p dbname < backup.sql  
Enter password: *****
```

Wenn die Backup-Datei komprimiert war, gehen Sie so vor:

```
user$ gunzip -c backup.sql.gz | mysql -u name -p dbname
```

Die folgenden Zeilen zeigen ein kleines Script, das an mysql ein SQL-Kommando über gibt. Dieses testet, ob eine Datenbank existiert oder nicht:

```
#!/bin/bash  
dbname="ebooks"  
sql="    SELECT schema_name FROM information_schema.schemata "  
sql="$sql WHERE schema_name='$dbname';"  
result=$(mysql -s -N -e "$sql")  
if [[ -n "$result" ]]; then  
    echo "Datenbank existiert"  
else  
    echo "unbekannte Datenbank"  
fi
```

```
mysqladmin [optionen] kommando1 kommando2 ...
```

`mysqladmin` hilft bei diversen administrativen Aufgaben auf einem MySQL- oder MariaDB-Server – etwa beim Anlegen neuer Datenbanken. An `mysqladmin` können mehrere Kommandos übergeben werden, die der Reihe nach ausgeführt werden. Die Namen von `mysqladmin`-Kommandos können so weit abgekürzt werden, wie sie noch eindeutig erkennbar sind (z.B. `flush-l` statt `flush-logs`). Die meisten `mysqladmin`-Kommandos können auch als SQL-Kommandos ausgeführt werden – etwa durch `CREATE DATABASE`, `DROP DATABASE`, `FLUSH` etc.

Damit `mysqladmin` eine Verbindung zum Datenbank-Server herstellen kann, müssen die beim [mysql](#)-Kommando beschriebenen Verbindungsoptionen verwendet werden, also `-u`, `-p`, `-h` etc. Außerdem kennt `mysqladmin` einige weitere Optionen:

- `-f` bzw. `--force`  
führt das Kommando ohne Rückfragen aus. Die Ausführung mehrerer Kommandos wird auch dann fortgesetzt, wenn bei einem Kommando ein Fehler auftritt.
- `-i` bzw. `--sleep=n`  
wiederholt das Kommando alle  $n$  Sekunden (beispielsweise zur regelmäßigen Statusanzeige). `mysqladmin` läuft nun endlos und muss mit `Strg+C` beendet werden.
- `-r` bzw. `--relative`  
zeigt in Kombination mit `-i` und dem Kommando `extended-status` die Veränderung gegenüber dem vorherigen Zustand an.

Im Folgenden sind die wichtigsten `mysqladmin`-Kommandos beschrieben:

- `create dbname`  
erzeugt eine neue Datenbank.
- `drop dbname`  
löscht eine vorhandene Datenbank unwiderruflich.

- **extended-status**  
zeigt zahllose Statusvariablen des Servers an.
- **flush-logs**  
schließt alle Logging-Dateien und öffnet sie wieder .
- **flush-privileges**  
liest die Datenbank mit den Zugriffsrechten neu ein.
- **kill *id1, id2 ...***  
beendet die angegebenen Threads.
- **ping**  
testet, ob eine Verbindung zum Datenbank-Server hergestellt werden kann.
- **shutdown**  
beendet den MySQL- bzw. MariaDB-Server.
- **status**  
zeigt diverse Statusvariablen des Datenbank-Servers an.
- **variables**  
liefert eine Liste der Systemvariablen des Datenbank-Servers.
- **version**  
ermittelt die Version des MySQL- oder MariaDB-Servers.

## Beispiel

Das folgende Kommando erzeugt eine neue Datenbank auf dem lokalen MySQL- oder MariaDB-Server:

```
user$ mysqladmin -u name -p create neuedb
Enter password: *****
```

```
mysqlbinlog [optionen] loggingdatei1 datei2 ...
```

MySQL bzw. MariaDB protokollieren bei entsprechender Konfiguration alle Änderungen, also insbesondere alle **INSERT**-, **UPDATE**- und **DELETE**-Kommandos, in binäre Logging-Dateien. Diese Dateien können im Bedarfsfall als inkrementelles Update dienen. **mysqlbinlog** hilft beim Auslesen der Logging-Dateien. Für den Verbindungsaufbau gelten dieselben Optionen wie bei [\*\*mysql\*\*](#) (-u, -p etc.).

- `--database=name`  
berücksichtigt nur Kommandos, die eine bestimmte Datenbank betreffen.
- `-j n`  
wertet die erste Logging-Datei erst ab einer bestimmten Position aus.
- `--no-defaults`  
berücksichtigt Konfigurationsdateien wie `.my.cnf` nicht. Die Option ist in der Praxis relativ häufig erforderlich, wenn `.my.cnf` eine Option enthält, die für alle MySQL-Client-Kommandos gilt, aber für `mysqlbinlog` nicht zulässig ist (z.B. `default-character-set=utf8`). Sie vermeiden dann die Fehlermeldung *unknown variable xy*.

## Beispiel

Nach dem Einspielen des letzten vollständigen Backups werden die inkrementellen Updates aus zwei Binlog-Dateien angewendet:

```
root# cd /var/log/mysql
root# mysqlbinlog -j 120 mariadb-bin.001380 mariadb-bin.001380 | mysql -u root -p
mysqldump [optionen] dbname [tabelle1 tabelle2 ...]
mysqldump [optionen] --databases [dbname1 dbname2 ...]
mysqldump [optionen] --all-databases
```

`mysqldump` ist ein einfaches Backup-Werkzeug für die Datenbanksysteme MySQL und MariaDB. Das Kommando liefert eine lange Liste aller SQL-Kommandos, aus denen die Datenbank exakt wiederhergestellt werden kann. `mysqldump` kennt drei Syntaxvarianten, je nachdem, ob eine Datenbank, mehrere aufgezählte Datenbanken oder alle von MySQL verwalteten Datenbanken gespeichert werden sollen. Nur bei der ersten Variante kann das Backup auf einzelne Tabellen beschränkt werden. Damit `mysqldump` eine Verbindung zum Datenbank-Server herstellen kann, müssen die beim [mysql](#)-Kommando beschriebenen Verbindungsoptionen verwendet werden, also `-u`, `-p`, `-h` etc. Eine Menge weiterer Optionen steuern die Details des Backups.

- `--add-locks`  
fügt vor dem ersten `INSERT`-Kommando `LOCK TABLE` und nach dem letzten `INSERT`-Kommando `UNLOCK` ein. Das beschleunigt das Wiedereinlesen von MyISAM-Tabellen. Die Option ist aber für InnoDB-Tabellen ungeeignet.
- `--create-options`  
inkludiert MySQL-spezifische Optionen in das `CREATE TABLE`-Kommando.
- `--disable-keys`  
deaktiviert die automatische Schlüsselaktualisierung während des Wiedereinlesens der Daten. Dadurch werden Indizes erst am Ende des Einfügeprozesses aktualisiert, was schneller ist.
- `--extended-insert`  
erzeugt `INSERT`-Kommandos, mit denen jeweils mehrere Datensätze gleichzeitig eingefügt werden.
- `--lock-all-tables`  
führt für die gesamte Datenbank `LOCK TABLE READ` aus. Das stellt sicher, dass während des gesamten Backups keine Tabelle verändert werden kann. Diese Option ist nur bei MyISAM-Tabellen zweckmäßig. Bei InnoDB-Tabellen verwenden Sie besser `--single-transaction`.
- `--lock-tables`  
führt während des Backups für jede einzelne Tabelle `LOCK TABLE READ` aus, bevor die Daten gelesen werden.
- `--no-create-info`  
speichert nur die Daten, aber nicht die Struktur der Datenbank. Im resultierenden Backup gibt es also nur `INSERT`-Kommandos, aber kein `CREATE TABLE`.
- `--no-data`  
sichert nur die Struktur (das Schema) der Datenbank, nicht aber die darin enthaltenen Daten.

- **--no-tablespaces**  
verhindert, dass `mysqldump` die SQL-Kommandos `CREATE LOGFILE GROUP` oder `CREATE TABLESPACE` in das Backup einbaut. Diese Option muss bei aktuellen MySQL-Server-Versionen verwendet werden, wenn Sie das Kommando in einem Account ohne das Privileg `Process` ausführen. Andernfalls kommt es zu einem Access-denied-Fehler.
- **--opt**  
ist die Kurzschreibweise für die Optionen `--add-drop-table`, `--add-locks`, `--create-options`, `--disable-keys`, `--extended-insert`, `--lock-tables`, `--quick` und `--set-charset`. Aus historischen Gründen ist `--opt` standardmäßig aktiv. Tatsächlich sind die resultierenden Einstellungen aber nur dann optimal (daher kommt der Name dieser Option!), wenn Sie ein Backup von MyISAM-Tabellen durchführen.
- **--quick**  
gibt die zu sichernden Datensätze ohne Zwischenspeicherung im RAM direkt aus.
- **--routines**  
sichert auch den Code von Stored Procedures.
- **--set-charset**  
bewirkt, dass am Beginn der `mysqldump`-Ausgabe der aktive Zeichensatz verändert und am Ende der bisher geltende Zeichensatz wiederhergestellt wird.
- **--single-transaction**  
bewirkt, dass das gesamte Backup im Rahmen einer einzigen Transaktion durchgeführt wird. Das ist bei InnoDB-Tabellen zweckmäßig und schließt aus, dass einzelne Tabellen während des Backups verändert werden. Somit wird sichergestellt, dass das Backup insgesamt konsistent ist.
- **--skip-opt**  
deaktiviert die standardmäßig aktive Option `--opt`.

- **--triggers**  
sichert auch den Code von Triggern.

## Beispiele

Für ein Backup einer Datenbank mit MyISAM-Tabellen sind nur wenige Optionen erforderlich. Wenn Sie ausschließen möchten, dass einzelne Tabellen während des Backups geändert werden, fügen Sie dem Kommando noch die Option **--lock-all-tables** hinzu. Damit sind während des Backups allerdings alle Schreiboperationen in der Datenbank blockiert.

```
user$ mysqldump -u user -p dbname > backup.sql  
Enter password: *****
```

Wenn Sie ein Backup komprimiert speichern möchten, gehen Sie so vor:

```
user$ mysqldump -u user -p dbname | gzip -c > backup.sql.gz
```

Die optimale Durchführung eines Backups mit InnoDB-Tabellen erfordert wesentlich mehr Optionen, weil die Standardeinstellungen von `mysqldump` nicht für diesen Fall optimiert sind:

```
user$ mysqldump -u user -p --skip-opt --single-transaction \  
    --create-options --quick --extended-insert \  
    --disable-keys --add-drop-table dbname > backup.sql
```

Wenn Sie das Backup automatisiert durch einen Cron-Job ausführen, ersetzen Sie `-u user -p` durch die folgende Option und geben die Verbindungsparameter in der nur für `root` lesbaren Datei `/root/.my.cnf` an:

```
# im Backup-Script, --defaults-extra-file muss die erste Option sein  
mysqldump --defaults-extra-file=/root/.my.cnf ...
```

Um das Backup später wieder einzuspielen, führen Sie diese Kommandos aus:

```
user$ mysqladmin -u user -p create dbname  
user$ mysql -u user -p --default-character-set=utf8 dbname < backup.sql
```

## N

**namei** [optionen] datei

**namei** zeigt die Zugriffsrechte aller Verzeichnisse, die den Pfad der angegebenen Datei bilden:

```
user$ namei -l ~/.ssh
f: /home/kofler/.ssh
drwxr-xr-x root root /
drwxr-xr-x root root home
drwxr-xr-x kofler kofler kofler
drwx----- kofler kofler .ssh
```

**nc** [optionen] [hostname/ip-adresse] [port]

Das Kommando **nc** ist unter der Bezeichnung *Netcat* bekannter. Es leitet TCP/UDP-Ports auf die Standardeingabe/-ausgabe um und bietet eine Menge weiterer Funktionen. Sie können mit dem Kommando ähnlich wie [telnet](#) interaktiv Netzwerkprotokolle wie HTTP oder SMTP ausprobieren. Mit **nc** können Sie aber auch Dateien über einen beliebigen Netzwerkport kopieren, einen Chat realisieren oder eine simple Backdoor einrichten, die über einen Port Kommandos entgegennimmt und ausführt. Da verwundert es nicht, dass **nc** unter Hackern und Penetration-Testern außerordentlich beliebt ist.

Bei einigen Linux-Distributionen ist das Kommando **nc** im gleichnamigen Paket enthalten, bei anderen Distributionen müssen Sie **netcat** installieren. Beachten Sie, dass es unterschiedliche Implementierungen von Netcat gibt. So kommt unter Debian und Ubuntu **netcat-traditional** zum Einsatz, während CentOS/RHEL eine Variante der [nmap](#)-Entwickler anbieten (<https://nmap.org/ncat>). In der Praxis ergeben sich daraus keine großen Unterschiede, allerdings sind möglicherweise einzelne Optionen je nach Version anders (oder gar nicht) implementiert.

- **-4** oder **-6**

verwendet ausschließlich IPv4 oder IPv6.

- `-l`  
wartet am angegebenen Port auf einen Verbindungsauftbau (*listen*).
- `-p portnr`  
legt den lokalen Port (Source-Port) fest. Der üblicherweise am Ende des nc-Kommandos angegebene Port ist hingegen der Ziel-Port (Destination-Port).
- `-x proxyadr:portnr`  
verwendet die angegebene Proxy-Adresse und den dazugehörigen Port.

Eine Menge weiterer Optionen beschreibt `man nc`. Eine mögliche Alternative zu nc ist das Kommando socat, das in diesem Buch aber nicht behandelt wird. Es unterstützt auch das Protokoll SCTP, kann über Proxyserver arbeiten, auch serielle Schnittstellen bedienen und die Daten für die Übertragung verschlüsseln:

<http://www.dest-unreach.org/socat>

## Beispiele

Um eine Datei über einen beliebigen Port (hier 1234) von Host 1 nach Host 2 zu kopieren, starten Sie zuerst auf Host 2 (rechte Spalte) den Empfänger und initiieren die Übertragung der Datei dann auf Host 1 (linke Spalte).

```
host2$ nc -l 1234 > datei
host1$ nc host2 1234 < datei
```

Ebenso einfach führen Sie einen Chat durch. Sie müssen sich mit Ihrem Gesprächspartner lediglich auf einen Port einigen. Der Chat wird auf einem Rechner mit nc `-l` initiiert (rechte Spalte). Damit überwacht nc den angegebenen Port 1234 und wartet auf einen Verbindungsauftbau.

Auf dem zweiten Rechner wird nc ohne Optionen gestartet, um die Verbindung zum ersten Host herzustellen. Eine sichtbare Bestätigung des Verbindungsauftbaus gibt es zwar nicht, aber sobald nun einer der beiden Gesprächspartner Text eingibt

(Standardeingabe) und mit  bestätigt, erscheint der Text im Terminal des anderen Gesprächspartners (Standardausgabe).

```
host1$ nc -l 1234
wie geht's?
gut
<Strg>+<C>
```

```
host2$ nc host1 1234
wie geht's?
gut
```

Das Gefahrenpotential von Netcat zeigt sich im dritten Beispiel: Hier wird nc auf Host 1 so eingerichtet, dass es alle auf Port 1234 empfangenen Eingaben an die Shell bash weitergibt. Deren Ausgaben werden wieder zurück an den Sender übertragen. Von einem zweiten Host können nun Shell-Kommandos auf Host 1 ausgeführt werden. [ls](#) zeigt also Dateien, die sich auf Host 1 befinden!

```
host1$ nc -l 1234 -e /bin/bash
host2$ nc host1 1234
ls
datei1
datei2
datei3
```

Die Option -e zur Ausführung eines Kommandos steht allerdings nicht bei allen Netcat-Versionen zur Verfügung. Sie fehlt insbesondere bei der unter Debian und Ubuntu üblichen netcat-traditional-Implementierung. Abhilfe:

Installieren Sie das Paket [nmap](#), und führen Sie das dort enthaltene Kommando ncat aus.

```
ncdu [optionen] [verzeichnis]
```

ncdu ist eine interaktive Variante des Kommandos [du](#).

Normalerweise wird es einfach ohne Optionen und Parameter aufgerufen. Es zeigt dann die Unterverzeichnisse im aktuellen Verzeichnis an, die am meisten Platz beanspruchen.

Mit den Cursortasten und  können Sie nun durch die Unterverzeichnisse navigieren. Diverse weitere Funktionen sind über Tastenkürzel zugänglich. Eine Referenz über die wichtigsten

Kürzel gibt `[?]`. Vorsicht, `[D]` löscht nach einer Rückfrage die aktuelle Datei bzw. das aktuelle Verzeichnis.

- `-r`  
führt das Kommando im Read-only-Modus aus. Ein versehentliches Löschen von Dateien oder Verzeichnissen ist damit ausgeschlossen.
- `-x`  
bleibt im aktuellen Dateisystem, berücksichtigt also keine [mount](#)-Verzeichnisse.

## Beispiel

Der Platzbedarf der Verzeichnisse im Heimatverzeichnis wird auch durch Balken symbolisiert.

```
user$ ncd
--- /home/kofler ---
1,5 GiB [#####] /Dropbox
1,0 GiB [#####] /Downloads
528,5 MiB [##] /Bilder
209,4 MiB [#] /Dokumente
148,7 MiB [ ] /.dropbox-dist
...
```

## needs-restarting [optionen]

Das unter CentOS, Fedora, Red Hat und verwandten Distributionen verfügbare Kommando `needs-restarting` gibt an, ob der Rechner oder einzelne Programme aufgrund eines Updates des Kernels, einer grundlegenden Bibliothek oder einer Firmware-Datei neu gestartet werden müssen.

Bei Distributionen auf der Basis von Debian oder Ubuntu steht das Kommando nicht zur Verfügung. Dort weist die Datei `/var/run/reboot-required` auf einen erforderlichen Neustart hin.

- `-r`  
zeigt an, ob und warum ein Rechnerneustart erforderlich ist.
- `-u`  
listet Prozesse des aktuellen Benutzers auf, die neu gestartet werden müssen. Diese Option gilt standardmäßig.

## Beispiel

Die folgenden Kommandos zeigen den Zustand eines Rechners, der schon zu lange unbeaufsichtigt läuft und dringend neu gestartet werden sollte:

```
root# needs-restarting -r
Core libraries or services have been updated:
  kernel -> 3.10.0-693.21.1.el7
  kernel -> 3.10.0-693.17.1.el7
  linux-firmware -> 20170606-58.gitc990aae.el7_4
  systemd -> 219-42.el7_4.10
Reboot is required to ensure that your system benefits from these updates.
More information: https://access.redhat.com/solutions/27943

root# needs-restarting | sort -n
1 : /usr/lib/systemd/system --system --deserialize 23
469 : /usr/lib/systemd/systemd-journald
485 : /usr/sbin/lvmetad -f
629 : /usr/lib/polkit-1/polkitd --no-debug
...
root# uptime
13:29:22 up 96 days, 21:50, 1 user, load average: 0,00, 0,01, 0,05
netplan [kommando]
```

Netplan (<https://netplan.io>) ist ein von Ubuntu entwickeltes Framework, das andere Netzwerk-Backends wie den NetworkManager sowie networkd (eine systemd-Komponente, siehe [networkctl](#)) konfiguriert und einbindet. Es kommt in Ubuntu zum Einsatz und wertet die Konfigurationsdateien aus, die sich in */etc/netplan* befinden (siehe [/etc/netplan/netplan.yaml](#)).

- **apply**  
führt netplan generate aus und aktiviert dann die geänderte Konfiguration.
- **generate**  
liest die Netplan-Konfiguration aus den YAML-Dateien in */etc/netplan*, */lib/netplan* und */var/netplan* und generiert die entsprechenden Konfigurationsdateien für die Netzwerk-Backends. Die neuen Dateien landen in */run/NetworkManager* bzw. in */run/systemd/network*.  
Die neue Konfiguration wird nicht aktiviert! Die geänderten Einstellungen werden erst wirksam, wenn das jeweilige Backend

aufgefordert wird, die Dateien auszuwerten (bzw. beim nächsten Neustart).

- `ip leases schnittstelle`  
gibt die aktuellen DHCP-Lease-Daten für die betreffende Schnittstelle aus. Das funktioniert nur, wenn das Backend networkd und eine DHCP-Konfiguration im Einsatz sind. Andernfalls liefert das Kommando in die Irre führende Fehlermeldungen.

### **netstat [optionen]**

netstat liefert Informationen über die Netzwerkaktivität auf dem lokalen Rechner. Wird das Kommando ohne Optionen aufgerufen, liefert es eine Liste aller offenen Internetverbindungen und Sockets.

- `-a`  
berücksichtigt auch nichtaktive Sockets, also Dienste, die auf einem Netzwerk-Port auf eine Verbindung warten (Zustand `LISTEN` bzw. bei deutscher Lokalisierung `HÖREN`).
- `-e`  
gibt bei Internetverbindungen zusätzlich den Benutzer an.
- `-n`  
liefert numerische Adressen und Port-Nummern anstelle von Host- und Netzwerkdienstnamen.
- `-p`  
gibt zusätzlich die Prozessnummer (PID) und den Prozessnamen des Programms an, das für die Verbindung verantwortlich ist.  
netstat benötigt root-Rechte, damit es Informationen über nichteigene Prozesse liefern kann.
- `-t / -u / -w / -x`  
schränkt die Ausgabe auf Verbindungen ein, die das Protokoll TCP, UDP, Raw oder Unix nutzen.

### **Beispiel**

Das folgende Kommando listet alle aktiven Verbindungen (established) bzw. überwachten Ports (LISTEN) auf. Die Ausgabe wurde aus Platzgründen stark gekürzt.

```
root# netstat -atupe
Active Internet connections (servers and established)
Proto Local Address          Foreign Addr      State    User        PID/Prog name
tcp   *:nfs                  *:*                LISTEN   root       -
tcp   *:ldap                 *:*                LISTEN   root       5842/slapd
tcp   *:3142                 *:*                LISTEN   root       5904/perl
tcp   localhost:mysql         *:*                LISTEN   mysql     5785/mysqld
tcp6  [::]:ssh               [::]:*              LISTEN   root       5559/sshd
tcp6  mars.sol:ssh           merkur.so...    ESTAB    root       7729/0
udp   *:nfs                  *:*                LISTEN   root       -
udp   mars.local:netbios-ns  *:*                LISTEN   root       6231/nmbd
```

### networkctl [kommando]

networkctl gehört zu den Administrationskommandos aus der systemd-Familie. Sofern networkd als Netzwerk-Backend eingesetzt wird, hilft networkctl bei der Analyse des Netzwerkstatus.

Standardmäßig ist networkd zwar bei vielen Distributionen installiert, aber nur bei wenigen auch aktiv. Zu den Ausnahmen zählt Ubuntu Server, wo networkd als Backend für das Ubuntu-eigene Netplan-System dient (siehe [netplan](#)). networkd wertet Konfigurationsdateien aus den Verzeichnissen

`/etc/systemd/network`, `/lib/systemd/network` sowie in `/run/systemd/network` aus (siehe

[/etc/systemd/network/networkd.network](#)).

- `list`  
listet alle Netzwerkschnittstellen auf. Die `SETUP`-Spalte des Ergebnisses zeigt, ob die Schnittstelle durch networkd gesteuert wird (`configured`) oder nicht (`unmanaged`).
- `lldp`  
zeigt andere Geräte im Netzwerk, die über das *Link Layer Discovery Protocol* (LLDP) entdeckt wurden.
- `status [schnittstelle]`  
liefert detaillierte Informationen über den Status einer Netzwerkschnittstelle. Wenn keine Schnittstelle angegeben wird,

versucht das Kommando, den gesamten Netzwerkstatus zusammenzufassen.

## Beispiele

Die folgenden Ausgaben von `networkctl` sind auf einem Ubuntu-Server in einer virtuellen Maschine entstanden:

```
root# networkctl list
IDX LINK      TYPE      OPERATIONAL   SETUP
  1 lo        loopback   carrier       unmanaged
  2 ens3      ether      routable     configured

root# networkctl status ens3
    Link File: /lib/systemd/network/99-default.link
    Network File: /run/systemd/network/10-netplan-ens3.network
        Type: ether
        State: routable (configured)
        ...
    HW Address: 52:54:00:0a:8d:fc
    Address: 138.201.20.182
              fe80::5054:ff:fe0a:8dfc
    Gateway: 138.201.20.176 (Fujitsu Technology Solutions GmbH)
    DNS: 213.133.100.100
          213.133.98.98
Search Domains: ubuntu-buch.info
```

Das zweite Listing zeigt die Statuszusammenfassung einer anderen Server-Installation mit IPv6-Konfiguration:

```
root# networkctl status
WARNING: systemd-networkd is not running, output will be incomplete.
        State: n/a
        Address: 138.201.20.187 on eth0
                  2a01:4f8:171:2baf::4 on eth0
                  fe80::5054:ff:feb9:3d5f on eth0
        Gateway: 138.201.20.176 (Fujitsu Technology Solutions GmbH) on eth0
                  2a01:4f8:171:2baf::2 (Fujitsu Technology Solutions GmbH) on eth0
```

## newaliases

Die Datei [/etc/aliases](#) enthält eine Alias-Liste für den E-Mail-Server, die z.B. dafür sorgt, dass alle E-Mails, die an `postmaster` adressiert sind, an `root` weitergeleitet werden. Damit Änderungen an dieser Datei vom Mail-Server berücksichtigt werden, müssen Sie `newaliases` ausführen.

## newgrp [gruppenname]

Das Kommando `newgrp` bestimmt die gerade aktive Gruppe eines Benutzers, der mehreren Gruppen angehört. Die aktive Gruppe bestimmt, welcher Gruppe neu erzeugte Dateien angehören. Die zur

Auswahl stehenden Gruppen können mit [groups](#) ermittelt werden. Wenn kein Gruppenname angegeben wird, wird die primäre Gruppe verwendet. Diese Gruppe gilt auch nach einem Login automatisch als aktive Gruppe.

### **Beispiel**

Im folgenden Beispiel macht das newgroup-Kommando docuteam zur aktiven Gruppe des Benutzers kofler. Die neu erzeugte Datei *newfile* wird daher der Gruppe docuteam zugeordnet und kann von anderen Mitgliedern des Dokumentationsteams bearbeitet werden.

```
user$ groups  
kofler docuteam wheel  
user$ newgroups docuteam  
user$ touch newfile
```

**newusers** datei

**newusers** liest eine Textdatei und erzeugt für jede Zeile einen neuen Benutzer. Die Textdatei weist prinzipiell dasselbe Format auf wie [/etc/passwd](#). Allerdings müssen die Passwörter unverschlüsselt in der zweiten Spalte angegeben werden. Die meisten weiteren Parameter (UID, GID etc.) sind optional. **newusers** erzeugt für jeden angegebenen Benutzer einen neuen Account, wobei bei Bedarf auch die dazugehörigen Gruppen angelegt werden. Für fehlende Parameter wählt **newusers** geeignete Defaultwerte, wobei die Einstellungen in [/etc/login.defs](#) berücksichtigt werden.

Beachten Sie, dass **newusers** zwar neue Heimatverzeichnisse erzeugt, wenn deren Ort in der sechsten Spalte angegeben ist. Das Kommando kümmert sich aber nicht darum, den Inhalt von [/etc/skel](#) dorthin zu kopieren.

### **Beispiel**

Die folgenden Zeilen zeigen die minimalistische Textdatei *users.txt*, die den Anforderungen von **newusers** entspricht. Da *users.txt* Klartextpasswörter enthält, müssen Sie darauf achten, dass niemand außer root die Datei lesen kann, bzw. die Datei nach der Ausführung von **newusers** wieder löschen.

```
huber:geheim1:::Hermann Huber:/home/huber:/bin/bash  
moser:geheim2:::Gabi Moser:/home/moser:/bin/bash  
schmidt:geheim3:::Peter Schmidt:/home/schmidt:/bin/bash
```

newusers erzeugt nun die drei neuen Benutzer huber, moser und schmidt sowie gleichnamige primäre Gruppen. newusers entscheidet sich selbst für geeignete UIDs und GIDs.

```
root# newusers users.txt
```

### **nft** kommando [optionen]

Die meisten aktuellen Linux-Distributionen verwenden mittlerweile das neue Firewall-System *Nftables*, das seinen Vorgänger *Netfilter* ablöst. Zur Konfiguration der Firewall kommt jedoch oft noch das alte Kommando [iptables](#) zum Einsatz, das dank einer Kompatibilitätsschicht weiterhin funktioniert.

Neue bzw. *Nftables*-spezifische Funktionen können aber nur mit dem Kommando nft gesteuert werden. Die folgende Beschreibung gibt einen groben Überblick über die wichtigsten Unterkommandos und Optionen dieses Kommandos. Viel mehr Details zur Funktionsweise von *Nftables* sowie zur Syntax von nft liefern `man nft` sowie das folgende Wiki:

<https://wiki.nftables.org>

Bevor Sie nft aufrufen, sollten Sie sich vergewissern, dass auf Ihrer Distribution nicht schon eine Firewall läuft. Unter CentOS, Fedora, RHEL und SUSE ist z.B. *Firewalld* aktiv (siehe [firewall-cmd](#)). Selbst definierte Firewall-Regeln führen dann oft zu Konflikten mit der Firewall Ihrer Distribution.

### **nft-Nomenklatur**

Für nft gilt eine eigene Nomenklatur: Firewall-Regeln (*rules*) sind immer Teil einer Kette (*chain*). Tabellen (*tables*) bestehen wiederum aus mehreren Ketten. Anders als bei [iptables](#) gibt es weder vordefinierte Tabellen noch vordefinierte Ketten. Sie können somit beliebig viele eigene Tabellen aus ebenfalls selbst definierten Ketten zusammenstellen.

Tabellen, Ketten und Regeln gelten für verschiedene Typen von Netzwerkpaketen, die den folgenden Familien (*families*) zugeordnet sind:

- `ip` (nur IPv4)
- `ip6` (nur IPv6)
- `inet` (für Regeln, die gleichermaßen für IPv4 und IPv6 gelten)
- `arp` (Level-2-Regeln, werden vor Level-3-Regeln ausgewertet)
- `bridge` (Switching-Regeln)
- `netdev` (Low-Level-Regeln, werden vor allen anderen Regeln ausgewertet und ermöglichen z.B. eine besonders effiziente Abwehr von DDOS-Angriffen)

## nft-Optionen

- `-a`  
baut in die Ausgaben von `nft list` sogenannte *handles* ein. Das sind Nummern, die jedes Objekt innerhalb einer Gruppe (also z.B. eine Regel innerhalb einer Kette) eindeutig bezeichnen. Handles ermöglichen es, gezielt einzelne Objekte zu löschen bzw. zu ersetzen oder die Position anzugeben, wo neue Objekte eingefügt werden sollen.
- `-f fname`  
liest die auszuführenden Kommandos aus der angegebenen Datei. Dabei sind zwei Syntaxvarianten zulässig: Die Datei kann entweder zeilenweise gewöhnliche nft-Kommandos enthalten oder eine hierarchisch strukturierte Abfolge von Tabellen, Ketten und Regeln. Der Aufbau der Datei für die zweite Variante entspricht der Ausgabe von `nft list ruleset`.
- `-S`  
verwendet in Ausgaben die in `/etc/services` angegebenen Namen von Netzwerkdiensten anstelle von Port-Nummern (also z.B. `ssh` anstelle von 22).
- `-v`  
zeigt die Versionsnummer an.

## nft-Kommandos

- `add chain family tname cname`  
erzeugt eine neue Kette mit dem Namen *cname* in der angegebenen Tabelle.
- `add rule family tname cname matches statements`  
fügt der angegebenen Kette *cname* eine neue Regel hinzu. Dabei gibt *matches* an, für welche Pakete die Regel gilt. *statements* beschreibt, was mit den betreffenden Paketen passieren soll. Anstelle von `add` sind auch die Schlüsselwörter `insert` und `replace` erlaubt, um eine Regel an einer bestimmten Stelle in die Regelliste einzubauen bzw. um eine vorhandene Regel durch eine neue zu ersetzen.
- `add table family tname`  
erzeugt eine neue Tabelle für die angegebene Familie.
- `delete/flush chain family tname cname`  
entfernt die angegebene Kette aus der Tabelle. `flush` löscht alle Regeln der Kette, nicht aber die Kette selbst.
- `delete rule family tname cname handle handle`  
löscht die durch *handle* spezifizierte Regel aus der Kette *cname*.
- `delete/flush table family tname`  
löscht die angegebene Tabelle. Bei `flush` werden zwar alle Ketten und Regeln der Tabelle gelöscht; die nun leere Tabelle bleibt aber erhalten.
- `flush ruleset`  
löscht sämtliche Tabellen inklusive aller darin enthaltenen Ketten und Regeln. Das entspricht einem kompletten Reset der Firewall. Die Firewall akzeptiert nun jedes Paket.
- `list tables`  
`list table family name`  
`list chain family tname cname`  
listet Tabellen, Ketten und Regeln auf.

- `list ruleset`  
listet sämtliche Tabellen, Ketten und Regeln auf. Die durch geschwungene Klammern strukturierte Ausgabe erfüllt die Syntaxregeln für `nft -f`, kann also als Grundlage für eine neue Regeldatei verwendet werden. Mit der zusätzlichen Option `-j` erfolgt die Ausgabe im JSON-Format.
- `monitor [filterkriterien]`  
zeigt *Nftables*-Ereignisse an (z.B. als Debugging-Hilfe).

## Beispiele

Die beiden ersten Kommandos zeigen, welche Firewall-Tabellen auf dem Testrechner (Fedora 32) definiert sind und welche Ketten und Regeln für die Tabelle `firewalld` innerhalb der Familie [ip](#) gelten.  
(*Nftables* erlaubt die Verwendung übereinstimmender Tabellennamen für unterschiedliche Familien. Deswegen muss beim zweiten `list`-Kommando auch die Familie angegeben werden.)

```
root# nft list tables
table bridge filter
table bridge nat
table inet firewalld
table ip firewalld
table ip6 firewalld

root# nft list table ip firewalld
table ip firewalld {
    chain nat_PREROUTING {
        type nat hook prerouting priority dstnat + 10; policy accept;
        jump nat_PREROUTING_ZONES
    }

    chain nat_PREROUTING_ZONES {
        iifname "enp1s0" goto nat_PRE_FedoraWorkstation
        goto nat_PRE_FedoraWorkstation
    }
    ...
}
```

[`systemctl`](#) deaktiviert nun den vorgegebenen Firewall-Dämon von Fedora:

```
root# systemctl disable --now firewalld
```

Jetzt kann mit `nft` eine eigene Firewall zusammengesetzt werden. Es ist nicht üblich, die Regeln durch den wiederholten Aufruf von `nft` zu definieren. Wesentlich eleganter ist es, die Regeln in einer

Textdatei zu formulieren und diese Datei wie ein Script auszuführen. Dazu geben Sie in der Zeile `nft -f` wie einen Shell-Interpreter an.

```
#!/sbin/nft -f
# vorhandene Firewall löschen
flush ruleset

# neue Tabelle für IPv4 und IPv6
table inet myfilter {
    chain input {
        type filter hook input priority 0; policy drop;
        # fehlerhafte Pakete blockieren
        ct state invalid drop
        # Pakete von selbst erzeugten Verbindungen immer akzeptieren
        ct state {established, related} accept
        # internen Netzwerkverkehr akzeptieren
        iif lo accept
        # Pakete an Loopback aber von externen Adressen blockieren
        iif != lo ip daddr 127.0.0.1/8 drop
        iif != lo ip6 daddr ::1/128 drop
        # ICMP-Pakete akzeptieren
        ip protocol icmp accept
        ip6 nexthdr icmpv6 accept
        # SSH-Verbindung von außen akzeptieren
        tcp dport 22 accept
        # bei Server-Konfiguration: weitere Regeln für
        # HTTP + HTTPS (Port 80 + 443), Samba usw.
        # ...
    }
    # kein Forwarding
    chain forward {
        type filter hook forward priority 0; policy drop;
    }
}
```

Nun können Sie das Script ausführen:

```
root# chmod +x myfirewall
root# ./myfirewall
```

`nft -f` führt zuerst einen Syntax-Check durch. Wenn es dabei Fehler feststellt, bricht es den Vorgang mit einer Fehlermeldung ab. Die bisherige Firewall bleibt in diesem Fall erhalten.

Das obige Script wurde leicht modifiziert vom Gentoo-Wiki übernommen. Dort sowie im ArchLinux-Wiki finden Sie noch mehr Beispiele:

<https://wiki.gentoo.org/wiki/Nftables/Examples>

<https://wiki.archlinux.org/index.php/Nftables>

**ngrep** [optionen] [grep-Suchausdruck] [pcap-Filterausdruck]

ngrep ist ein sogenannter *Packet Sniffer*. Das Kommando liest den Netzwerkverkehr eines Ports mit und filtert ihn. ngrep bietet ähnliche Funktionen wie das Kommando [tcpdump](#) und greift wie dieses auf die pcap-Bibliothek zurück, um die Netzwerkpakete auszulesen. Im Unterschied zu [tcpdump](#) berücksichtigt ngrep aber auch den Inhalt der Pakete. Das funktioniert naturgemäß nur bei nicht verschlüsselten Protokollen, also z.B. bei HTTP oder FTP.

Der Suchausdruck ist wie bei [grep](#) als reguläres Muster zu formulieren. Für den pcap-Filterausdruck gilt die bei [tcpdump](#) zusammengefasste Syntax.

- `-d schnittstelle|any`  
bestimmt die Netzwerkschnittstelle.
- `-i`  
ignoriert die Groß- und Kleinschreibung im [grep](#)-Suchausdruck.
- `-v`  
invertiert die Suche. ngrep liefert somit nur die Pakete, in denen das [grep](#)-Suchmuster *nicht* erkannt wurde.
- `-w`  
interpretiert den [grep](#)-Suchausdruck als Wort.
- `-W byline`  
berücksichtigt bei der Ausgabe Zeilenumbrüche, was zu besser lesbaren Ausgaben führt.

## Beispiel

Das folgende Beispiel lauscht auf allen Schnittstellen nach HTTP-Paketen, in denen die Schlüsselwörter user, pass usw. vorkommen:

```
root# ngrep -d any -i 'user|pass|pwd|mail|login' port 80
interface: any
filter: (ip or ip6) and ( port 80 )
match: user|pass|pwd|mail|login

T 10.0.0.87:58480 -> 91.229.57.14:80 [AP] POST /index.php
HTTP/1.1..Host: ... user=name&pass=geheim&login=Login
...
nice [optionen] programm
```

`nice` startet das angegebene Programm mit einer verringerten oder erhöhten Priorität. Das Kommando kann dazu eingesetzt werden, nicht zeitkritische Programme mit kleiner Priorität zu starten, um das restliche System nicht zu stark zu beeinträchtigen.

- `-n +/ -n`

gibt den `nice`-Wert vor. Standardmäßig (also ohne `nice`) werden Programme mit dem `nice`-Wert 0 gestartet. Ein Wert von --20 bedeutet »höchste Priorität«, ein Wert von +19 bedeutet »niedrigste Priorität«. Werte kleiner 0 dürfen nur von `root` angegeben werden, d.h., die meisten Anwender können mit `nice` nur Programme mit reduzierter Priorität starten. Wenn auf diese Option verzichtet wird, startet `nice` das Programm mit dem `nice`-Wert von +10.

Beachten Sie, dass `nice` nur die CPU-Belastung steuert. Wenn Sie die IO-Belastung eines Kommandos reduzieren möchten, setzen Sie besser `ionice` ein.

## Beispiel

Das folgende Kommando startet das Backup-Script `sichere` mit niedrigerer Priorität:

```
user$ nice -n 10 sichere  
nl [optionen] datei
```

`nl` nummeriert alle nichtleeren Zeilen der angegebenen Textdatei und schreibt das Ergebnis in die Standardausgabe. Durch die Einstellung der zahlreichen Optionen kann eine Seitenweise Nummerierung, eine Nummerierung von Kopf- und Fußzeilen etc. erreicht werden.

```
 nmap [optionen] hostname/ip-adresse/ip-adressbereich
```

Das Kommando `nmap` (*Network Mapper*) aus dem gleichnamigen Paket führt einen Port-Scan durch und versucht festzustellen, welche Netzwerkdienste auf dem angegebenen Rechner bzw. im angegebenen Netzwerk aktiv sind. `nmap` sollte ausschließlich zur Analyse eigener Rechner bzw. nach Rücksprache mit dem jeweiligen

Administrator eingesetzt werden. Ein Port-Scan fremder Rechner kann als Einbruchsversuch gewertet werden!

- **-A**  
führt einen ausführlichen (»aggressiven«) Scan durch, entspricht `-sV -O -sC --traceroute`.
- **-F**  
berücksichtigt nur die 100 wichtigsten Ports aus `/usr/share/nmap/nmap-services` (schneller Scan).
- **-iL datei**  
scannet die in der Datei angegebenen IP-Adressen.
- **-oN datei / -oG datei / -ox datei.xml**  
schreibt die Ergebnisse in eine normale Textdatei, in eine Textdatei, die mit [grep](#) gut weiterverarbeitet werden kann, oder in eine XML-Datei. Ohne die Option verwendet `nmap` die Standardausgabe und das normale Textformat.
- **-O**  
versucht, das Betriebssystem zu erkennen; diese Option muss mit einer Scan-Option kombiniert werden, z.B. mit `-sS`, `-sT` oder `-sF`.
- **-p1-10,22,80**  
berücksichtigt nur die angegebenen Ports.
- **-Pn**  
verzichtet auf einen Ping-Test. Damit betrachtet `nmap` alle Hosts als online und führt auf jeden Fall einen Scan durch (langsam!).
- **-sL**  
listet alle Ports auf und gibt in der Vergangenheit zugeordnete Hostnamen an. Das gelingt besonders schnell, liefert aber veraltete Daten auch von Geräten, die aktuell gar nicht mehr online sind.
- **-sP**  
führt nur einen Ping-Test durch (schnell).

- **-sS**  
führt einen TCP-SYN-Scan durch (gilt per Default).
- **-sU**  
berücksichtigt auch UDP. Diese Option darf zusammen mit einer anderen **-s**-Option verwendet werden.
- **-sV**  
versucht bei offenen Ports herauszufinden, welcher Service dort angeboten wird (*Service Version Detection*) bzw. welches Programm in welcher Version für den Dienst zuständig ist.
- **-T0 bis -T5**  
wählt ein Timing-Schema. **-T5** ist am schnellsten. **-T3** gilt per Default. **-T0** und **-T1** sind extrem langsam, minimieren aber das Risiko, dass der Scan bemerkt wird.
- **-v**  
gibt detaillierte Informationen aus (*verbose*).

Sie müssen sich beim Aufruf für *eine* **-s**-Option entscheiden. Einzig **-su** darf mit anderen **-s**-Optionen kombiniert werden. Generell ist die richtige Wahl der Optionen ein Kompromiss zwischen Gründlichkeit und Geschwindigkeit.

In vielen Fällen reicht `nmap -v -A name`, um einen ersten Überblick über die Netzwerkdienste des angegebenen Rechners zu bekommen. Fortgeschrittene `nmap`-Anwender finden weitere Details in der [man](#)-Seite sowie auf den folgenden Webseiten:

<https://insecure.org/nmap>

<https://nmap.org/book> Zu `nmap` existieren auch grafische Benutzeroberflächen, z.B. `nmap-frontend` oder `zenmap`.

## Beispiele

Das folgende Kommando führt einen schnellen Netzwerk-Scan im lokalen Netzwerk durch (256 IP-Adressen). Dank der Konzentration auf die wichtigsten 100 Ports ist die Sache in gut zwei Sekunden

erledigt. Die nmap-Ausgaben wurden aus Platzgründen stark gekürzt und zeigen nur die Ergebnisse von zwei der gefundenen Geräte:

```
root# nmap -F -T4 10.0.0.0/24
Nmap scan report for imac (10.0.0.2)
  Host is up (0.00019s latency).
  PORT      STATE SERVICE
  22/tcp    open  ssh
  88/tcp    open  kerberos-sec
  445/tcp   open  microsoft-ds
  548/tcp   open  afp
  MAC Address: AC:87:A3:1E:4A:87 (Apple)

Nmap scan report for raspberrypi (10.0.0.22)
  Host is up (0.00038s latency).
  Not shown: 99 closed ports
  PORT      STATE SERVICE
  22/tcp    open  ssh
  MAC Address: B8:27:EB:11:44:2E (Raspberry Pi Foundation)
```

...

```
Nmap done: 256 IP addresses (6 hosts up) scanned in 2.42 seconds
```

Das zweite Beispiel zeigt das wiederum stark gekürzte nmap-Ergebnis für ein NAS-Gerät, das sich in einem anderen lokalen Netz befindet:

```
root# nmap -v -A 10.0.0.89
Nmap scan report for DiskStation.fritz.box (192.168.178.28)
  Host is up (0.0023s latency).
  ...
  Discovered open port 445/tcp on 192.168.178.28
  Discovered open port 139/tcp on 192.168.178.28
  Discovered open port 80/tcp on 192.168.178.28
  Discovered open port 443/tcp on 192.168.178.28
  Discovered open port 22/tcp on 192.168.178.28
  Discovered open port 5001/tcp on 192.168.178.28
  Discovered open port 5000/tcp on 192.168.178.28
  Discovered open port 548/tcp on 192.168.178.28

  PORT      STATE SERVICE VERSION
  22/tcp    open  ssh      OpenSSH 7.4 (protocol 2.0)
  ssh-hostkey:
    2048 5a:e7:3a:66:f4:99:9f:0a:0a:... (RSA)
    256 06:1a:bf:9f:e9:d0:64:3a:92:49:... (ECDSA)
    256 ad:b7:7d:ab:ae:70:0a:c9:a6:0c:... (ED25519)
  ...
  80/tcp    open  http      nginx
  139/tcp   open  netbios-ssn Samba smbd 4.6.2
  445/tcp   open  netbios-ssn Samba smbd 4.6.2
  ...

Nmap done: 1 IP address (1 host up) scanned in 64.19 seconds
```

**nmcli [optionen] con|dev|nm** kommando

Der Network Manager wird üblicherweise durch ein Menü im Gnome- oder KDE-Panel gesteuert. Mit dem Kommando `nmcli` können Sie Netzwerkverbindungen auch über die Kommandozeile bzw. durch ein Script steuern.

Die zur Auswahl stehenden Kommandos hängen davon ab, auf welches Objekt sie sich beziehen. Mit den Optionen `-t`, `-p` etc. kann die Form der `nmcli`-Ausgaben gesteuert werden – je nachdem, ob die Ausgabe ordentlich formatiert oder durch ein Script weiterverarbeitet werden soll (Details siehe `man nmcli`).

- `con down name` bzw. `con down uuid n`  
deaktiviert die angegebene Verbindung.
- `con show`  
listet alle konfigurierten Verbindungen auf, wobei zu jeder Verbindung ihr Name und ihre UUID angegeben wird.
- `con up name` bzw. `con up uuid n`  
aktiviert die angegebene Verbindung.
- `dev disconnect name`  
beendet die Verbindung für die angegebene Schnittstelle.
- `dev`  
listet alle dem Network Manager bekannten Netzwerkschnittstellen auf und gibt deren Eigenschaften an.
- `dev show`  
zeigt ausführliche Informationen zu allen Netzwerkschnittstellen an.
- `dev wifi list`  
liefert eine Liste aller in Reichweite befindlichen WLAN-Netzwerke, wobei auch die Signalstärke angegeben wird. Das Kommando setzt voraus, dass der `wpa_supplicant`-Dienst läuft.
- `general [status]`  
zeigt den Status des Network Managers an.

- **networking [on|off]**  
zeigt an, ob eine Netzwerkverbindung besteht oder nicht. Durch **off** bzw. **on** können alle Netzwerkverbindungen unterbrochen bzw. wiederhergestellt werden.
- **radio [wifi|wwan|wimax|all] [on|off]**  
zeigt den Status der Funknetzwerke an bzw. verändert deren Status. **wifi** bezeichnet WLAN-Verbindungen, **wwan** Mobilfunkverbindungen und **wimax** die bei uns unübliche WiMAX-Technik.

## Beispiel

Das erste Kommando listet zuerst alle dem Network Manager bekannten Verbindungen auf. Die folgenden beiden Kommandos de- und reaktivieren die Verbindung mit dem Namen

*Kabelgebundene Verbindung 1:*

```
root# nmcli con show
NAME           UUID            TYP      GERÄT
Kabelgebundene Verbindung 1  effd094d-...  802-3-ethernet  enp0s3
virbr0          207d9f4f-...  bridge       virbr0
root# nmcli con down  id 'Kabelgebundene Verbindung 1'
root# nmcli con up    id 'Kabelgebundene Verbindung 1'
```

Das zweite Beispiel filtert aus den detaillierten Schnittstelleninformationen die IP-Adressen des Nameservers heraus. Das ist vor allem unter Ubuntu praktisch, wo diese Adressen nicht aus */etc/resolv.conf* hervorgehen. Vielmehr verweist *resolv.conf* bei dieser Distribution auf einen lokalen Nameserver, also auf die Adresse 127.0.0.1.

```
root# nmcli dev show | grep DNS
IP4.DNS[1]:  10.0.0.138
IP4.DNS[2]:  4.4.8.8
```

**nohup** kommando

Wenn Sie ein Kommando als Hintergrundprozess in einem Shell-Fenster starten und das Fenster dann schließen oder wenn Sie das Kommando in einer Textkonsole starten und sich dann ausloggen, so wird der Hintergrundprozess automatisch beendet. Das ist im Regelfall ein sinnvolles Verhalten.

Manchmal möchten Sie aber einen Prozess starten, der nach dem Ausloggen weiterläuft – und genau dazu dient `nohup`. Das Kommando muss mit seinem vollständigen Pfad angegeben werden. Es kann keine Textausgaben an die Standardausgabe schreiben. Gegebenenfalls werden derartige Ausgaben in die Datei `nohup.out` im lokalen Verzeichnis umgeleitet.

## Beispiel

Im folgenden Beispiel loggt sich ein Administrator via `ssh` auf einem Server ein, startet dort ein Backup-Script im Hintergrund und loggt sich dann wieder aus. Das Backup-Script läuft weiter.

```
someone@localhost$ ssh user@remotehost
user@remotehost$ nohup backup-script &
user@remotehost$ exit
```

## nproc

`nproc` gibt an, wie viele CPU-Cores zur Verfügung stehen. Bei CPUs mit Hyper-Threading werden auch die virtuellen Cores mitgerechnet. Genaue Informationen über die CPU verraten das Kommando `lscpu` sowie der Inhalt der Datei `/proc/cpuinfo`.

## nvidia-xconfig [optionen]

`nvidia-xconfig` hilft dabei, den proprietären NVIDIA-Grafiktreiber zu konfigurieren. Das Kommando steht nur zur Verfügung, wenn dieser Treiber installiert ist.

Wenn das Kommando ohne Parameter ausgeführt wird, erzeugt bzw. verändert es die Datei `/etc/X11/xorg.conf` so, dass in Zukunft der NVIDIA-Grafiktreiber verwendet wird. Wenn nach einem Neustart des Rechners das Grafiksystem funktioniert, kann zur weiteren Konfiguration die grafische Benutzeroberfläche `nvidia-settings` verwendet werden.

- `-c datei`  
verwendet die angegebene Datei anstelle von `/etc/X11/xorg.conf`.
- `--mode=WxH`  
fügt `xorg.conf` einen Grafikmodus für  $W \times H$  Pixel hinzu.

- `--query-gpu-infos`  
zeigt Details zu allen vorhandenen Grafikkarten und angeschlossenen Monitoren an.
- `-t`  
liest `xorg.conf` und zeigt die darin enthaltenen Einstellungen in einer übersichtlichen Baumdarstellung an.

# O

## **openssl** kommando

`openssl` aus dem gleichnamigen Paket hilft beim Erzeugen und der weiteren Administration von Zertifikaten sowie privaten und öffentlichen Schlüsseln. Das Kommando greift dabei auf die OpenSSL-Bibliothek zurück. Neue Zertifikate oder Schlüssel werden an die Standardausgabe geschrieben, wenn Sie nicht mit `-out` eine Zielfile angeben.

- `dparam [optionen] [n]`  
erzeugt bzw. manipuliert Parameterdateien für das Diffie-Hellman-Verfahren. Dabei gibt der Parameter *n* die Schlüsselgröße in Bit an. Im einfachsten Fall generiert `openssl dparam -out keyfile 512` einen Diffie-Hellman-Schlüssel in einer Länge von 512 Bit.
- `enc [optionen]`  
liest Daten aus der Standardeingabe, codiert bzw. decodiert (Option `-d`) diese symmetrisch und schreibt das Ergebnis an die Standardausgabe. Den dabei einzusetzenden Algorithmus und den Schlüssel geben Sie mit Optionen an, z.B. in der Form `-aes-256-cbc -pass file:mykeyfile`. Beachten Sie, dass `openssl enc` nur für relativ kleine Datenmengen geeignet ist. Um große Dateien zu verschlüsseln, verwenden Sie besser das Kommando [gpg](#).
- `genpkey [optionen]`  
erzeugt einen privaten Schlüssel. Um einen 2048-Bit-RSA-Schlüssel in eine Datei zu schreiben, sind die folgenden Optionen erforderlich:  
`openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out server.key`  
Wenn der Schlüssel selbst verschlüsselt werden soll, geben Sie den Namen des Verschlüsselungsalgorithmus durch eine weitere

Option an, z.B. -aes256. Bei der Ausführung fragt openssl nun zweimal nach einem Verschlüsselungspasswort.

- `list-standard-commands` | `list-message-digest-commands` |  
`list-cipher-commands` | `list-cipher-algorithms` |  
`list-message-digest-algorithms` | `list-public-key-algorithms`  
listet die von openssl unterstützten Kommandos, Algorithmen etc. auf.
- `rand [optionen] n`  
schreibt *n* Byte mit pseudo-zufälligen Daten an die Standardausgabe. Beachten Sie, dass es sich bei der Ausgabe um binäre Daten handelt, nicht um ASCII-Text! Mit der Option -base64 erreichen Sie, dass openssl eine Base64-Codierung in das ASCII-Format durchführt. Die Länge der resultierenden Zeichenkette verlängert sich dadurch um circa ein Drittel.
- `req [optionen]`  
erzeugt eine Anfrage zur Signierung eines Zertifikats (also ein *Certificate Signing Request*, eine CSR-Datei). Mit -new geben Sie an, dass ein neues Zertifikat eingerichtet werden soll. -key gibt die dabei zu verwendende Schlüsseldatei an. Ohne diese Option erzeugt openssl zu diesem Zweck einen neuen RSA-Schlüssel, den Sie mit -keyout in einer eigenen Datei speichern können.  
`openssl req -new -sha256 -key server.key -out server.csr`  
openssl fragt dann interaktiv nach den Eckdaten des Zertifikats, also nach dem Ländercode, dem Ort, Ihrem Namen, dem Hostnamen (Common Name) etc.  
Mit den zusätzlichen Optionen -x509 -days *n* erzeugt openssl ein selbst signiertes Zertifikat mit einer Gültigkeit von *n* Tagen anstelle einer Zertifikatsanfrage.
- `rsa [optionen]`  
verarbeitet RSA-Schlüssel und konvertiert sie zwischen unterschiedlichen Formen und Formaten. -in datei und -out

`datei` geben an, aus welcher Datei der ursprüngliche Schlüssel gelesen und wohin der neue Schlüssel geschrieben wird.

Wenn Sie keine weiteren Optionen angeben, entfernt `openssl` die Verschlüsselung (*pass phrase*) des Schlüssels. Mit `-inform` und `-outform` kann die Form des Schlüssels spezifiziert werden (standardmäßig `PEM`, alternativ `DER` oder `NET`). Mit `-in datei -text` können Sie die Eigenschaften eines vorhandenen Schlüssels ermitteln, z.B. die Schlüssellänge (die Anzahl der Bits).

- `sclient [optionen]`

agiert als SSL/TLS-Client, um Verbindungen zu einem verschlüsselten Server zu testen. Ein Anwendungsbeispiel finden Sie hier:

<https://www.heise.de/security/artikel/StartTLS-785453.html>

- `speed`

führt mehrere Benchmarktests durch, die alle jeweils drei Sekunden dauern. Das Ergebnis besagt, wie viele Byte für den jeweiligen Algorithmus verarbeitet wurden (je mehr es sind, desto schneller läuft bzw. desto besser optimiert ist die Hardware, auf der der Test durchgeführt wird).

- `x509 [optionen]`

zeigt Zertifikatdaten an, signiert Zertifikate und konvertiert Zertifikate zwischen verschiedenen Formen. `-in` und `-out` geben wieder die Dateinamen der Quell- und Zielzertifikate an. Mit `-req` erwartet `openssl` eine Zertifikatsanfrage als Eingabe, nicht das Zertifikat selbst. `-CA` gibt an, welches CA-Zertifikat für die Signatur verwendet werden soll. `-CAkey` gibt den dazugehörigen privaten Schlüssel an.

Zu den meisten `openssl`-Kommandos gibt es eigene `man`-Seiten. Beispielsweise beschreibt `man req` die unzähligen Optionen von `openssl req`, `man rsa` beschreibt die Optionen von `openssl rsa` etc.

## Beispiele

Wenn Sie für Ihren Webserver ein Zertifikat von einer offiziellen Zertifizierungsstelle (z.B. Thawte) unterzeichnen lassen möchten, erzeugen Sie zuerst einen Schlüssel (wenn Sie noch keinen haben) und dann ein Certificate Signing Request. Auf eine Verschlüsselung des Schlüssels wird hier verzichtet, weil der Webserver sonst bei jedem Start nach dem Verschlüsselungspasswort fragen müsste.

```
root# genkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out server.key  
root# chmod 400 server.key  
root# openssl req -new -key server.key -out server.csr  
...
```

Common Name (eg server FQDN or YOUR name) []: **firma-abc.de**

Zur Kontrolle können Sie einen Blick in die CSR-Datei werfen:

```
root# openssl req -in server.csr -noout -text
```

Die CSR-Datei übermitteln Sie nun an die Zertifizierungsstelle. Von dort bekommen Sie dann gegen gutes Geld das signierte Zertifikat. Die übliche Dateikennung lautet *.pem* oder *.crt*. Diese Datei, Ihren eigenen Schlüssel sowie ein CA-Zertifikat der Zertifizierungsstelle müssen Sie dann bei der Apache-Konfiguration angeben.

```
# Apache-Konfigurationsdatei  
SSLCertificateFile      /etc/apache2/server.pem  
SSLCertificateKeyFile   /etc/apache2/server.key  
SSLCertificateChainFile /etc/apache2/sub.class1.server.ca.pem  
SSLCACertificateFile    /etc/apache2/ca.pem
```

Wenn Sie die Kosten für eine offizielle Signatur scheuen, können Sie sich Ihr Zertifikat selbst signieren:

```
root# openssl x509 -req -days 1900 -in server.csr \  
        -signkey server.key -sha256 -out server.pem  
Signature ok  
subject=/C=DE/L=Berlin/O=Max Muster/CN=www.firma-abc.de/  
emailAddress=webmaster@firma-abc.de  
Getting Private key
```

Um einen Blick in die Zertifikatsdatei zu werfen, führen Sie `openssl x509 -text` aus:

```
root# openssl x509 -text -in server.pem  
Certificate:  
Data:  
    Version: 1 (0x0)  
    Serial Number: 12669601459972319941 (0xafd37766c36baac5)  
    Signature Algorithm: sha256WithRSAEncryption  
    Issuer: C=DE, L=Berlin, O=Max Muster,  
           CN=www.firma-abc.de/emailAddress=webmaster@firma-abc.de  
Validity  
    Not Before: Sep 28 14:48:03 2020 GMT
```

```
Not After : Dec 10 14:48:03 2025 GMT
```

```
...
```

Das Erzeugen eines Schlüssels, einer Zertifikatsanfrage und das Selbst-Signieren kann auch in einem einzigen Kommando erledigt werden: Das folgende Kommando erzeugt ein für 10 Jahre gültiges, selbst signiertes Zertifikat für einen Mail-Server. Entscheidend ist, dass Sie bei der Ausführung von `openssl` als Common Name den Hostnamen Ihres Servers angeben. Da das Zertifikat selbst signiert ist, wird Ihr Mail-Client bei der Konfiguration darauf hinweisen, dass das Zertifikat nicht vertrauenswürdig ist.

```
root# openssl req -new -x509 -days 3650 -nodes \
    -out      /etc/ssl/certs/postfix.pem \
    -keyout  /etc/ssl/private/postfix.key
...
Common Name (eg server FQDN or YOUR name) []: firma-abc.de
root# chmod 400 /etc/ssl/private/postfix.key
```

## P

### **pactl** kommando [parameter]

`pactl` steuert das PulseAudio-System. Dieser netzwerkfähige Sound-Server ist auf den meisten Linux-Distributionen standardmäßig installiert und verwendet zumeist ALSA als Basis. PulseAudio macht es möglich, dass mehrere Programme unkompliziert parallel das Audio-System nutzen können. `pactl` kennt rund 30 Kommandos, von denen hier nur die wichtigsten präsentiert werden:

- `info|stat`  
zeigt die Eckdaten bzw. den Speicherverbrauch des PulseAudio-Systems an.
- `list [short] [modules|sinks|sources|clients|cards|...]`  
liefert eine detaillierte Beschreibung aller Module, Audio-Quellen und -Ausgänge, Streams etc., die von PulseAudio verwaltet werden. Das zusätzliche Schlüsselwort `short` reduziert die Ausgabe auf eine Zeile pro Element. Die Angabe von `modules`, `sinks` bewirkt, dass `pactl` nur die betreffenden Elemente auflistet.
- `set-default-sink n`  
legt den Default-Audio-Ausgang fest. Anstelle der Nummer des Audio-Systems *n* (siehe `/proc/asound/cards`) können Sie auch dessen Namen angeben. Die zulässigen Namen ermitteln Sie mit `pactl list short cards`.
- `set-sink-mute n 0|1|toggle`  
deaktiviert, aktiviert bzw. verändert die Mute-Funktion für den Audio-Ausgang *n*.
- `set-sink-volume n vol`  
stellt die Lautstärke für den Audio-Ausgang *n* ein. Die Lautstärke kann als Integerzahl, als Prozentwert oder als Dezibel-Wert angegeben werden (z.B. 10% oder 20dB).

### **Beispiel**

Die beiden folgenden Kommandos führen dazu, dass das erste Audio-Gerät als Standardausgabegerät bei einer Lautstärke von 20% verwendet wird:

```
user$ pactl set-default-sink 0  
user$ pactl set-sink-volume 0 20%  
pandoc [optionen] in1 [in2 in3 ...] [> out]
```

Die primäre Aufgabe von pandoc aus dem gleichnamigen Paket besteht darin, Markdown-Textdateien in andere Formate umzuwandeln, beispielsweise in HTML-Dateien, PDF-Dokumente oder Office-Dateien. Mit Einschränkungen kann pandoc anstelle von Markdown-Dateien auch andere Eingabeformate verarbeiten, unter anderem HTML und LaTeX.

- **-c cssdatei**  
bietet die angegebene CSS-Datei in das HTML-Dokument ein. Die Option kann mehrfach angegeben werden, wenn mehrere CSS-Dateien erforderlich sind.
- **-f format**  
gibt das Quellformat an (*from*). Wichtige Quellformate sind asciidoc, docbook, latex und markdown. pandoc versucht anhand der Dateikennung der Ein- und Ausgabedateien die gewünschten Quell- und Zielformate selbst zu erkennen. Die Optionen **-f** und **-t** sind nur erforderlich, wenn dies nicht gelingt.
- **-H headerdatei**  
fügt die angegebene Header-Datei am Beginn des HTML- oder LaTeX-Quelltextes ein. Die Option kann mehrfach angegeben werden, wenn mehrere Header-Dateien erforderlich sind.
- **-o output.kennung**  
schreibt das Ergebnis in die angegebene Datei anstatt zur Standardausgabe. Die Option ist bei binären Ausgabeformaten (DOCX, ODT, PDF) zwingend erforderlich.
- **-s**  
erzeugt ein eigenständiges Dokument (*standalone*). Diese Option

ist vor allem bei den Ausgabeformaten HTML, LaTeX und RTF relevant. Ohne diese Option erzeugt pandoc Dateien ohne Header-Informationen; solche Dateien können nicht für sich verwendet werden, sondern müssen in ein anderes Dokument eingebettet werden.

- **-t format**  
gibt das Zielformat an (*to*). Wichtige Zielformate sind `docx`, `latex`, `html`, `html5`, [man](#), `odt` und `rtf`. Beachten Sie, dass `pdf` nicht zur Liste der unterstützten Formate zählt. Um PDF-Dokumente zu erzeugen, müssen Sie die Option `-o out.pdf` angeben. pandoc wandelt den Quelltext dann zuerst in das LaTeX-Format um und erzeugt anschließend die PDF-Datei mit `pdflatex`.
- **--toc**  
bietet ein Inhaltsverzeichnis in das resultierende Dokument ein.

## Beispiele

Das erste Kommando erzeugt aus der Markdown-Datei `input.text` ein eigenständiges HTML-Dokument. Das zweite Kommando erzeugt eine EPUB-Datei aus einem Buch, dessen Kapitel in eigenen Dateien gespeichert sind. Das dritte Kommando erzeugt eine PDF-Datei, wobei hinter den Kulissen LaTeX zum Einsatz kommt.

```
user$ pandoc -s -c mein.css input.text > output.html  
user$ pandoc kap1.text kap2.text kap3.text -t epub3 -o buch.epub  
user$ pandoc vorwort.text kap01.text kap02.text -o out.pdf
```

```
paplay [optionen] [datei]  
parecord [optionen] [datei]
```

`paplay` spielt die angegebene Audiodatei über das PulseAudio-System ab. `parecord` führt eine Aufnahme durch und speichert den Audio-Stream als Datei.

- **-d *id***  
bezeichnet das Ausgabegerät (`paplay`) bzw. die Audio-Quelle (`parecord`). Ohne diese Option wird das jeweilige Default-Gerät verwendet.

- **--file-format=name**  
gibt das gewünschte Audio-Format an, z.B. `raw` oder `wav`.
- **--list-file-formats**  
listet alle unterstützten Audio-Formate auf.
- **--volume=n**  
gibt die gewünschte Wiedergabelautstärke an. Der Wertebereich geht von 0 bis 65.535 für maximale Lautstärke.

## Beispiel

Das folgende Kommando spielt eine der Audiodateien ab, die mit LibreOffice mitgeliefert werden:

```
user$ paplay /usr/lib/libreoffice/share/gallery/sounds/train.wav
paps [optionen] textdateien > out.ps
```

paps konvertiert die angegebenen UTF-8-Textdateien in das PostScript-Format und schreibt das Ergebnis in die Standardausgabe. Die UTF-8-Zeichen werden nicht als Buchstaben, sondern durch Linienzüge dargestellt. Aus diesem Grund sieht die Darstellung der PostScript-Datei in einem PostScript-Viewer (Evince, Okular etc.) unter Umständen pixelig aus. Eine Textauswahl ist nicht möglich. Die Druckqualität ist aber gut.

- **--columns=n**  
bestimmt die Anzahl der Textspalten (standardmäßig eine).
- **--cpi=n**  
steuert die Anzahl der Zeichen pro Inch (2,54 cm) und somit die Schriftgröße.
- **--font=name**  
gibt den gewünschten Zeichensatz an (standardmäßig Monospace 12).
- **--landscape**  
formatiert den Text im Querformat.

## Beispiel

Das folgende Kommando produziert die zweispaltige PostScript-Datei `text.ps`:

```
user$ paps --landscape --columns 2 text.txt > druck.ps
```

```
parallel [optionen] ['kommando']
```

parallel aus dem gleichnamigen Paket hilft dabei, mehrere Kommandos parallel auszuführen, unter Umständen auch auf unterschiedlichen Rechnern. parallel unterstützt teilweise dieselben Optionen wie [xargs](#).

- **-eta**  
gibt Informationen über den Status der Parallelverarbeitung sowie den voraussichtlichen Zeitpunkt an, wann parallel fertig ist (*estimated time of arrival*). Diese Zeitabschätzung setzt voraus, dass jeder Job in etwa gleich lange dauert.
- **-j n**  
maximal *n* Instanzen parallel ausführen.
- **-j +n bzw. -j -n**  
wie oben, aber die Angabe erfolgt relativ zur Anzahl der CPUs/Cores. Auf einer CPU mit 8 Cores führt -j +0 dazu, dass maximal 8 Instanzen des Kommandos parallel ausgeführt werden. -j -1 beschränkt die Anzahl der parallelen Aufrufe entsprechend auf 7.
- **-S host1,host2,host3**  
verteilt die Jobs auf die Hosts host1, host2, host3 etc. Wenn die Host-Liste mit einem Komma endet, wird zusätzlich auch der lokale Rechner verwendet.  
-S setzt voraus, dass die Kommunikation mit den Hosts via [ssh](#) und [scp](#) ohne Passwort möglich ist; Sie müssen also vorher mit [ssh-copy-id](#) entsprechende Schlüsseldateien einrichten.  
Außerdem muss das auszuführende Kommando natürlich auf allen Hosts zur Verfügung stehen. Schließlich sollte auf allen Hosts das Kommando parallel installiert sein. Es wird benötigt, um die Anzahl der CPU(-Cores) festzustellen.  
-S wird überlicherweise mit den Optionen --transfer, --return { . } .ergebnis und --cleanup kombiniert, um die lokale Datei zum

externen Host zu kopieren, die neue Datei mit der Kennung `*.ergebnis` zurückzuübertragen und schließlich die temporären Daten auf dem Host wieder zu löschen. Eine Kurzschreibweise für diese drei Optionen lautet `-trc { . }.ergebnis`.

In der Theorie verspricht `-s` eine einfache Verteilung von Jobs über mehrere via SSH erreichbare Hosts. Bei meinen Tests hat sich die Option allerdings als äußerst fehleranfällig erwiesen. Wenn Sie Aufgaben via SSH parallelisieren und beispielsweise ein Kommando auf Dutzenden von Hosts oder Cloud-Instanzen ausführen möchten, bieten sich anstelle von `parallel` die Kommandos [pssh](#) (Parallel SSH) oder [clusterssh](#) an, die in diesem Buch allerdings nicht behandelt werden.

## Beispiele

Eine einfache Anwendung von `parallel` besteht darin, eine Reihe von Dateien zu komprimieren, zu dekomprimieren oder sonst wie zu verarbeiten. Anstatt `gunzip *.gz` auszuführen, verwenden Sie `parallel`, um das jede Datei eine eigene Instanz von [gunzip](#) zu starten. Auf einem Rechner mit mehreren CPUs/Cores wird dies deutlich schneller sein als die serielle Verarbeitung der Dateien:

```
user$ ls *.gz | parallel gunzip
```

Im zweiten Beispiel soll eine Reihe von PNG-Bildern mit [convert](#) in das EPS-Format umgewandelt werden. Dabei reicht es aber nicht aus, einfach die Dateinamen zu übergeben. Vielmehr erwartet [convert](#) einmal den ursprünglichen Dateinamen und einmal den Name abzüglich der Kennung `*.png`, aber zuzüglich der neuen Kennung `*.eps`. `parallel` ersetzt dazu `{}` in der Kommandozeichenkette durch den Dateinamen und `{.}` durch den Namen ohne Kennung.

```
user$ ls *.png | parallel -eta -j +0 \
    'convert -density 90 -flatten -background white {} {.}.eps'
```

**parted** [optionen] [device [kommando [optionen]]]

`parted` hilft bei der Partitionierung von Festplatten und SSDs. Im Gegensatz zu veralteten Kommando [fdisk](#) kommt `parted` auch mit

GUID-Partitionstabellen (GPT) zurecht.

Wenn Sie an `parted` kein Kommando übergeben, können Sie das Programm interaktiv verwenden und der Reihe nach mehrere Kommandos eingeben. Die Kommandos dürfen abgekürzt werden, sofern die Eingabe eindeutig ist (also `q` statt `quit`).

Beachten Sie dabei aber, dass die Kommandos anders als bei [`fdisk`](#) sofort ausgeführt werden und nicht mehr rückgängig gemacht werden können! Anstelle des wenig komfortablen Programms `parted` können Sie auch dessen grafische Benutzeroberfläche `gparted` einsetzen.

- **-a bzw. --align**

gibt an, wie neue Partitionen ausgerichtet werden sollen.

Zulässige Einstellungen sind `none`, `cylinder` (Ausrichtung an Zylindergrenzen), `minimal` (Ausrichtung an Blockgrenzen) oder `optimal` (Ausrichtung an Vielfachen von einem MiB). Sie sollten unbedingt die Option `-a optimal` verwenden, um den Datenträger in maximaler Geschwindigkeit zu nutzen.

- **-l bzw. --list**

listet alle Partitionen auf.

- **-s bzw. --script**

führt alle als Parameter übergebenen Kommandos ohne Rückfragen aus.

Im Folgenden sind die wichtigsten `parted`-Kommandos kurz beschrieben:

- **align-check min/opt nr**

testet, ob die angegebene Partition an einer optimalen Position startet. `parted align-check min` testet lediglich, ob die Partition an einer physikalischen Blockgrenze startet. Bei `parted align-check opt` versucht `parted` aus den Eckdaten des Datenträgers die optimale Startposition von Partitionen auszurechnen. Gelingt dies nicht, gelten Vielfache von einem MiB als optimal.

- `mklabel msdos/gpt`  
richtet eine neue Partitionstabelle im MBR- oder GPT-Format ein.  
Vorsicht: Dabei geht der gesamte Inhalt der Festplatte verloren!  
parted unterstützt auch diverse andere Partitionsformate, die  
aber im Linux-Umfeld selten von Bedeutung sind (siehe `man parted`).
- `mkpart primary/logical/extended/name [dateisystem] start ende`  
richtet eine neue Partition ein. Beachten Sie, dass Sie die Start- und Endposition angeben müssen, nicht die Startposition und Größe! Die Endposition kann auch als negative Zahl angegeben werden und wird dann vom Ende des Datenträgers an gerechnet. Die Angabe der Partitionsart (primary, logical oder extended) ist nur bei Datenträgern mit MBR-Partitionstabellen zweckmäßig. Wenn sich auf dem Datenträger eine GUID-Partitionstabelle (GPT) befindet, geben Sie der Partition mit dem ersten `mkpart`-Parameter einen Namen.  
Die optionale Angabe des Dateisystems dient nur zur Festlegung des Partitionstyps. Auf der neuen Partition wird aber kein Dateisystem eingerichtet. Zulässige Typen sind `fat16`, `fat32`, `ext2`, `HFS`, `linux-swap`, `NTFS` und `ufs`.
- `print`  
listet alle Partitionen auf.
- `quit`  
beendet parted.
- `resize nr start ende`  
verändert die Größe der angegebenen Partition. Das Kommando kann leider nur für die erweiterte Partition eines MBR-Datenträgers verwendet werden. Um die Größe anderer Partitionen zu verändern, müssen Sie diese löschen und dann mit dem exakt gleichen Startpunkt neu einrichten. Das ist natürlich eine fehleranfällige Operation, die zudem voraussetzt, dass hinter der Partition freier Platz ist.

- `rm nr`  
löscht die angegebene Partition.
- `set nr flag on/off`  
verändert die Flags (Zusatzattribute) der Partition. `parted` kennt unter anderem die Flags `boot`, `root`, `swap`, `hidden`, `raid`, [`lvm`](#), `lba` und `bios_grub`.
- `unit einheit`  
legt die Einheit für Positions- und Größenangaben fest. Zur Auswahl stehen:
  - `s` (Sektoren)
  - `B` (Byte)
  - `kB`, `MB`, `GB` und `TB` ( $10^3$ ,  $10^6$ ,  $10^9$  und  $10^{12}$  Byte)
  - `KiB`, `MiB`, `GiB` und `TiB` ( $2^{10}$ ,  $2^{20}$ ,  $2^{30}$  und  $2^{40}$  Byte)
  - `%` (Prozent, relativ zur Gesamtgröße des Datenträgers)
  - `compact` (dezimale Megabyte bei Eingaben, lesefreundliche Ausgabe)

Standardmäßig gilt die Einstellung `compact`. Wenn `parted` mit der Option `--align` gestartet wurde, werden beim Einrichten neuer Partitionen die Positions- und Größenangaben entsprechend angepasst.

## Beispiele

Die folgenden Zeilen zeigen, wie Sie mit `parted` eine neue LVM-Partition auf einer SSD mit GPT einrichten:

```
root# parted -a optimal /dev/sda
(parted) print
Modell: ATA SAMSUNG SSD 830 (scsi)
Festplatte /dev/sda: 512GB
Sektorgröße (logisch/physisch): 512B/512B
Partitionstabelle: gpt
Nummer Anfang Ende Größe Dateisys Name Flags
 1 17,4kB 1000MB 1000MB fat32 EFI System Part... boot, versteckt
 2 1000MB 21,0GB 20,0GB ext4 versteckt
 3 21,0GB 21,1GB 134MB Microsoft reser... versteckt, msftres
 4 21,1GB 61,2GB 40,1GB ntfs Basic data part... versteckt
 5 61,2GB 82,2GB 21,0GB ext4 versteckt
 6 82,2GB 103GB 21,0GB ext4 boot, versteckt
 7 103GB 124GB 21,0GB ext4 boot
```

```

8      124GB   281GB   157GB   ext4
9      281GB   386GB   105GB
10     386GB   407GB   21,0GB   ext4
(parted) mkpart lvm-partition 407GB 450GB
(parted) set 11 lvm on
(parted) print
...
11     407GB   450GB   42.7GB           lvm-partition      lvm
(parted) quit

```

parted kann Kommandos auch direkt verarbeiten, ohne manuelle Interaktion. Die folgenden Kommandos setzen voraus, dass `/dev/sdb` das Device eines USB-Sticks oder einer SD-Karte ist. Das erste parted-Kommando richtet dort eine neue MBR-Partitionstabelle ein und löscht damit alle vorhandenen Daten. Das zweite Kommando erzeugt eine primäre Partition, wobei am Beginn und am Ende des Datenträgers jeweils 1 MiB frei bleibt. [mkfs.vfat](#) richtet in dieser Partition ein VFAT-Dateisystem ein.

```

root# parted /dev/sdb mklabel msdos
  Warnung: Die bestehende Partitionstabelle und alle Daten
  auf /dev/sdb werden gelöscht. Wollen Sie fortfahren?
  Ja/Yes/Nein/No? ja
root# parted /dev/sdb 'mkpart primary fat32 1mib -1mib'
root# mkfs.vfat -F 32 -n FOTOS /dev/sdb1

```

Das letzte Beispiel zeigt die Verwendung von parted im Script-Modus. Das Kommando richtet auf dem angegebenen Datenträger eine GPT-Partitionstabelle ein. Vorsicht, alle bisher gespeicherten Daten geben dabei verloren!

```

root# mkpart /dev/sdb -s mklabel gpt
partprobe [optionen] [devices]

```

partprobe wird üblicherweise ohne Optionen oder andere Parameter nach der Veränderung der Partitionierung lokaler Festplatten oder SSDs ausgeführt, z.B. nachdem [fdisk](#) beendet wurde. Das Kommando informiert nun den Kernel über die durchgeführten Änderungen, sodass die neuen bzw. veränderten Partitionen ohne einen Neustart des Rechners verwendet werden können.

```
partx [optionen] [partition] [disk]
```

partx ist ein Low-Level-Kommando, das Informationen über die Partitionierung von Datenträgern ermittelt bzw. an den Kernel

weitergibt. partx ist gleichermaßen für die Partitionstabellen MBR und GPT geeignet.

Die zu bearbeitende Partition wird normalerweise direkt durch eine Device-Datei angegeben. Alternativ können Sie den Datenträger durch seine Device-Datei und die Partitionsnummer durch die Option -n spezifizieren, z.B. mit partx -n 3 /dev/sda. Falls eine einzelne Partition selbst wieder Sub-Partitionen enthält und daher wie ein selbstständiger Datenträger behandelt werden soll, stellen Sie dem Partitions-Device ein freistehendes Minuszeichen voran, also z.B. partx - /dev/sda4.

Anders als [fdisk](#) und [parted](#) verändert partx den Datenträger nicht, eignet sich also *nicht* zum Einrichten oder Löschen physikalischer Partitionen! Die Optionen -a, -d und -u verändern lediglich die Partitionstabelle des Linux-Kernels, belassen den Datenträger aber unverändert. Der Zweck dieser Optionen besteht ausschließlich darin, dem Kernel Informationen über eine mit einem anderen Programm veränderte Partitionierung mitzuteilen. Einfacher gelingt dies meist mit dem Kommando [partprobe](#). In der Praxis wird partx oft zur Bearbeitung virtueller Datenträger verwendet, also z.B. für Loop-Devices oder für Image-Dateien virtueller Maschinen.

- -a  
fügt neue Partitionen hinzu. Die Partition muss bereits physikalisch auf dem Datenträger existieren. -a aktualisiert lediglich die Partitionstabelle des Kernels!
- -b  
verarbeitet Größenangaben in Byte.
- -d  
löscht Partitionen aus der Partitionstabelle des Kernels.
- -n M:N  
gibt die zu bearbeitenden Partitionen des Datenträgers numerisch an. Dabei sind verschiedene Schreibweisen möglich, wie die folgenden Beispiele verdeutlichen:

- n 4 bezeichnet die vierte Partition.
- n -2 bezeichnet die vorletzte Partition.
- n 3:6 meint die Partitionen 3 bis 6.
- n 3: bezieht sich auf alle Partitionen, beginnend mit der dritten.
- n :5 umfasst die Partitionen 1 bis 5.
- -o spalten bzw. --output spalten  
zeigt ausgewählte Informationen zum ausgewählten Datenträger bzw. zur Partition an. spalten kann unter anderem die folgenden Schlüsselwörter enthalten: NR (Partitionsnummer), START (Startsektor), END (Endsektor), SIZE (Größe in Byte), TYPE (Partitionstyp als Hexcode oder UUID) und FLAGS (Zusatzinformationen). partx -o NR,START,END liefert somit eine Liste der Partitionen jeweils mit ihren Start- und Endsektoren. Als Sektorgröße gilt generell 512 Byte, unabhängig von der tatsächlichen Sektorgröße des Datenträgers.
- -s bzw. --show  
zeigt Informationen zum ausgewählten Datenträger bzw. zur Partition an. Die mit -s oder -o angezeigten Daten werden vom physikalischen Datenträger ermittelt und stimmen unter Umständen nicht mit der Partitionstabelle des Linux-Kernels überein.
- -u  
verändert die Größe oder andere Daten einer Partition in der Partitionstabelle des Kernels. Die Änderungen müssen vorher physikalisch auf dem Datenträger durchgeführt werden.

## Beispiel

Das folgende Kommando liefert Detailinformationen zu einer Festplatte mit einer MBR-Partitionstabelle:

```
root# partx -o NR,TYPE,FLAGS,START,END,SECTORS,SIZE /dev/sda
NR TYPE FLAGS START END SECTORS SIZE
 1 0x83 0x80    2048  411647  409600  200M
 2 0x8e 0x0    411648 31457279 31045632 14,8G
```

**passwd [optionen] [username]**

`passwd` ohne Parameter ermöglicht es, das Passwort des aktuellen Benutzers zu ändern. Dazu muss zuerst das alte und dann zweimal hintereinander das neue Passwort eingegeben werden. Das neue Passwort wird in verschlüsselter Form in die Datei [`/etc/shadow`](#) eingetragen. Allerdings muss das neue Passwort bei manchen Distributionen bestimmten Sicherheitsregeln entsprechen, damit es akzeptiert wird.

`root` kann mit `passwd name` auch das Passwort anderer Benutzer verändern. Das alte Passwort muss nicht angegeben werden, d.h., `root` kann das Passwort auch dann verändern, wenn der Benutzer sein Passwort vergessen hat. Für `root` gelten die obigen Passwortrestriktionen nicht, er kann also auch ein Passwort aus nur einem einzigen Zeichen definieren. Es ist aber nicht einmal `root` gestattet, überhaupt kein Passwort (also einfach  ) anzugeben. Durch die Angabe von Optionen kann `passwd` einzelne Accounts auch sperren, wieder aktivieren sowie Ablaufzeiten für das Konto bzw. sein Passwort einstellen (siehe auch die Beschreibung des Kommandos [`chage`](#), das in dieser Hinsicht noch mehr Einstellmöglichkeiten bietet).

- `-g`  
stellt das Passwort einer Gruppe ein. Diese Funktion wird nur selten genutzt. Sie können damit das Betreten nichtprimärer Gruppen durch [`newgrp`](#) mithilfe eines Passworts absichern. Da das Gruppenpasswort mehreren Benutzern bekannt sein muss, ist das Verfahren inhärent unsicher.
- `-l`  
deaktiviert das Konto (*lock*).
- `-u`  
aktiviert ein deaktiviertes Konto wieder (*unlock*).
- `-x n`  
gibt an, wie viele Tage ein Passwort maximal gültig bleibt. `-x 180`

bewirkt also, dass circa alle sechs Monate ein neues Passwort eingestellt werden muss.

`passwd` kann nur interaktiv verwendet werden. Es besteht keine Möglichkeit, das Passwort als Parameter oder Option zu übergeben. Um Passwörter automatisiert zu setzen, verwenden Sie das Kommando `chpasswd`.

## Beispiel

Im folgenden Beispiel stellt `root` ein neues Passwort für den Benutzer `huber` ein. Dieser muss in Zukunft sein Passwort einmal jährlich ändern.

```
root# passwd huber
ändere Passwort für Benutzer huber.
Geben Sie ein neues Passwort ein: ****
Geben Sie das neue Passwort erneut ein: ****
passwd: alle Authentifizierungsmerkmale erfolgreich aktualisiert.
root# passwd -x 365 huber
```

```
paste datei1 datei2 ...
```

`paste` setzt die Zeilen der angegebenen Dateien zu neuen (längereren) Zeilen zusammen und gibt das Ergebnis auf dem Bildschirm aus. Die erste Zeile des resultierenden Textes ergibt sich also aus der ersten Zeile der ersten Datei plus der ersten Zeile der zweiten Datei etc. Zwischen den Bestandteilen der neuen Zeile werden Tabulatorzeichen eingefügt. Durch `> zieldatei` kann das Ergebnis in einer Datei gespeichert werden.

```
patch [optionen] < patchdatei
```

`patch` wendet die in einer [diff](#)-Datei zusammengefassten Änderungen an. Das Kommando wird in der Regel dazu genutzt, Codeänderungen anzuwenden (beispielsweise für den Kernelcode).

- `-b`  
erzeugt Backup-Dateien für alle geänderten Dateien.
- `--dry-run`  
testet den Patch, führt aber keine Änderungen aus. Generell sollten Sie vor der Anwendung jedes Patches mit der Option `--dry-run` sicherstellen, dass dabei keine Probleme auftreten. Nichts

ist ärgerlicher als ein fehlerhaft oder nur teilweise angewendeter Patch!

- **-pn**  
entfernt *n* Verzeichnisebenen aus den Dateinamen der Patch-Datei. Wenn der ursprüngliche Dateiname */a/b/name.c* lautet, macht **-p1** daraus *a/b/name.c*, **-p2** liefert *b/name.c*. Der richtige Wert *n* hängt davon ab, in welchem Verzeichnis patch ausgeführt wird.
- **-R bzw. --reverse**  
wendet den Patch invers an. Auf diese Weise wird ein bereits angewandter Patch wieder rückgängig gemacht.

## Beispiel

Die folgenden Kommandos zeigen, wie der Kernelcode von Version 5.7.5 auf 5.7.6 gepatcht wird. Dazu muss zuerst der Patch von Version 5.7 auf 5.7.5 rückgängig gemacht werden, bevor dann der Patch für Version 5.7.6 angewendet werden kann. Kernel-Patches beziehen sich üblicherweise auf die letzte Major-Version, also hier auf die Version 5.7.

```
root# cd /usr/src/linux-5.7.5
root# bunzip2 -c patch-5.7.5.bz2 | patch -R -p1 --dry-run (Invers-Patch testen)
... keine Fehlermeldungen
root# bunzip2 -c patch-5.7.5.bz2 | patch -R -p1          (5.7.5 -> 5.7)
root# bunzip2 -c patch-5.7.6.bz2 | patch -p1 --dry-run    (Patch testen)
... keine Fehlermeldungen
root# bunzip2 -c patch-5.7.6.bz2 | patch -p1          (5.7 -> 5.7.6)
root# cd /usr/src
root# mv linux-5.7.5 linux-5.7.6
```

**pdf2ps quelle.pdf [ziel.ps]**

pdf2ps erzeugt aus einem PDF-Dokument eine PostScript-Datei. Wenn Sie auf die Angabe von *ziel.ps* verzichten, erhält die neue PDF-Datei den Namen *quelle.pdf*. Eine Alternative zu pdf2ps ist das Kommando [pdftops](#) aus der Poppler-Bibliothek. Um PostScript-Dateien in PDF-Dokumente umzuwandeln, verwenden Sie [ps2pdf](#).

- **-dLanguageLevel=n**  
gibt an, welcher PostScript-Level für die Ausgabe verwendet

werden soll (standardmäßig PostScript Level 2, alternativ werden auch die Level 1 und 3 unterstützt).

### **pdfimages [optionen] datei.pdf [bildname]**

pdfimages versucht, Bilder aus PDF-Dateien zu extrahieren. Das funktioniert aber nicht immer perfekt. Oft werden unzählige winzige Bilder erkannt. Die relevanten Bilder erkennen Sie am ehesten an der Dateigröße.

Sie übergeben an das Kommando eine PDF-Datei und einen Bildnamen. Das Kommando erzeugt dann Bilder, die es in der Form *bildname-nnn.typ* benennt. Der Befehl schreibt standardmäßig Bilder in den unüblichen Formaten PBM und PPM, es sei denn, Sie wählen mit **-f** ein anderes Format aus.

- **-all**  
speichert eingebettete Bilder nach Möglichkeit im vorgegebenen Format. Wo dies nicht möglich ist, werden PNG-Bilder erzeugt.
- **-f n und -t n**  
gibt an, von welcher bis zu welcher Seite die PDF-Datei ausgewertet werden soll. Ohne diese Optionen berücksichtigt das Kommando das gesamte Dokument.
- **-j**  
belässt Bilder, die im JPEG-Format in die PDF-Datei eingebettet sind, in diesem Format.
- **-list**  
listet die gefundenen Bilder auf, erzeugt aber keine Bilddateien.
- **-png bzw. -tiff**  
speichert alle Bilder im PNG- bzw. im TIFF-Format.

### **Beispiel**

pdfimages **-list** liefert umfassende Informationen zu allen gefundenen Bildern. Die Ausgabe ist hier aus Platzgründen gekürzt.

```
user$ pdfimages -list buch.pdf
page type width height color comp bpc enc interp object ... size ratio
 1 stencil 1 1 - 1 1 image no [inline] 1B -
 4 stencil 1 1 - 1 1 image no [inline] 1B -
```

```

4 stencil    1     1  -      1   1  image  no  [inline]      1B  -
4 stencil    1     1  -      1   1  image  no  [inline]      1B  -
4 image     415   324  rgb    3   8  image  no       71   11.5K 2.9%
5 image     626   441  rgb    3   8  image  no       81   19.3K 2.4%
7 image    1173   635  rgb    3   8  jpeg   no      134   59.6K 2.7%
...

```

Im folgenden Beispiel extrahiert pdfimages elf Bitmaps aus der PDF-Datei:

```

user$ pdfimages -all buch.pdf tst
user$ ls tst*
tst-000.png  tst-001.png  tst-002.png  tst-003.png  tst-004.png  tst-005.png
tst-006.jpg  tst-007.png  tst-008.png  tst-009.png  tst-010.png
pdftk datei1.pdf datei2.pdf ... kommando

```

pdftk manipuliert PDF-Dateien. Sie können damit Seiten extrahieren, mehrere PDF-Dokumente zusammenführen, PDF-Dokumente verschlüsseln bzw. die Verschlüsselung entfernen (wenn Sie das Passwort kennen) etc. Aus Platzgründen ist die Syntax hier vereinfacht dargestellt. Die komplette Syntax liefert man pdftk. Sie können mehrere Kommandos kombinieren, müssen dabei aber die Reihenfolge einhalten, in der die Kommandos hier beschrieben sind:

- **input\_pw *password***  
gibt das Passwort zu *datei1.pdf* an.
- **cat**  
fügt alle PDF-Dateien zusammen.
- **cat *seitenliste***  
extrahiert die angegebenen Seiten. Mehrere Seitenbereiche werden durch Leerzeichen getrennt, also beispielsweise **1-5 7-10**. end bezeichnet die letzte Seite. Um die Seiten aus mehreren PDF-Dateien zusammenzufügen, geben Sie jeder Eingabedatei ein Kürzel und verweisen bei den Seitenangaben darauf (also beispielsweise **pdftk A=datei1.pdf B=datei2.pdf cat A2-4 B5 B7**). Wenn Sie nur Seiten mit geraden bzw. ungeraden Seitennummern wünschen, fügen Sie der Seitenangabe even oder odd hinzu, also etwa **1-10odd**.

- **background *watermark.pdf***  
hinterlegt jede Ausgabeseite mit einer Seite von *watermark.pdf*. Wenn *watermark.pdf* weniger Seiten als die Ausgabe hat, wird die letzte Seite aus *watermark.pdf* wiederholt als Wasserzeichen verwendet. Die Ausgangs-PDF-Datei muss transparent sein, sonst ist das Wasserzeichen nicht sichtbar.
- **stamp *stamp.pdf***  
schreibt über jede Ausgabeseite eine Seite aus *stamp.pdf*. Wenn *stamp.pdf* weniger Seiten als die Ausgabe hat, wird die letzte Seite aus *stamp.pdf* wiederholt. *stamp.pdf* muss transparent sein, sonst überdeckt sie die ursprüngliche PDF-Datei.
- **burst**  
erzeugt für jede Seite der PDF-Dateien eine eigene Datei mit dem Namen *page\_n.pdf*, wobei *n* die Seitennummer ist.
- **output *ergebnis.pdf***  
speichert die resultierende PDF-Datei unter dem Namen *ergebnis.pdf*.
- **owner\_pw *passwort* bzw. user\_pw *passwort***  
verschlüsselt *ergebnis.pdf* mit dem angegebenen Passwort. Das owner\_pw-Passwort gilt für den Ausdruck und andere PDF-Operationen, das user\_pw-Passwort für das Öffnen der Datei.

## Beispiele

Das folgende Kommando liest die Seiten 10 bis 20 sowie 30 bis 40 aus *in.pdf* und schreibt sie in die neue Datei *out.pdf*:

```
user$ pdftk in.pdf cat 10-20 30-40 output out.pdf
```

Auch um mehrere PDF-Dateien aneinanderzufügen, verwenden Sie das Kommando [cat](#):

```
user$ pdftk in1.pdf in2.pdf in3.pdf cat output out.pdf
```

Das folgende Beispiel erzeugt für jede einzelne Seite in *in.pdf* eine eigene PDF-Datei mit dem Namen *pg\_n*, wobei *n* die Seitennummer ist:

```
user$ pdftk in.pdf burst
```

Das nächste Beispiel erzeugt eine verschlüsselte PDF-Datei. Die Datei kann zwar ohne das Passwort xxx gelesen, nicht aber ausgedruckt oder sonstwie bearbeitet werden. Wenn Sie selbst das Lesen der Datei schützen möchten, verwenden Sie statt owner\_pw das Kommando user\_pw.

```
user$ pdftk in.pdf output encrypted.pdf owner_pw xxx
```

Zuletzt wird pdftk dazu verwendet, eine PDF-Datei mit einem Wasserzeichen zu markieren:

```
user$ pdftk in.pdf background watermark.pdf output out.pdf  
pdftops [optionen] quelle.pdf [ziel.ps]
```

pdftops aus der Poppler-Bibliothek, die unter Debian und Ubuntu im Paket poppler-utils bereitgestellt wird, erzeugt eine PostScript-Datei aus einem PDF-Dokument. Im Vergleich zu [pdf2ps](#) unterstützt es zahlreiche Optionen zur Beeinflussung der PostScript-Datei.

- **-eps**  
erzeugt eine EPS-Datei. Bei mehrseitigen PDF-Dokumenten muss mit **-f** und **-l** eine Seite ausgewählt werden.
- **-f n und -l n**  
gibt die erste und letzte Seite an (*first* und *last*).
- **-leveln**  
gibt den gewünschten PostScript-Level an (1 bis 3).
- **-levelnsep**  
führt zusätzlich eine Farbseparierung durch. Dazu werden alle Farben in das CMYK-Format umgewandelt.
- **-opw password bzw. -upw password**  
gibt das Owner- bzw. User-Passwort an, um passwortgeschützte PDF-Dokumente zu verarbeiten.
- **-paper format**  
gibt das gewünschte Papierformat an (A3, A4, letter oder legal).
- **-paperw n und -paperh n**  
gibt die Papiergröße in Punkt an.

```
pdftotext [optionen] quelle.pdf [ziel.txt]
```

`pdftotext` extrahiert den Text aus einer PDF-Datei und speichert ihn in einer reinen Textdatei (standardmäßig `quelle.txt`). Die Formatierung sowie alle Bilder gehen dabei verloren. Auch `pdftotext` gehört zur Poppler-Bibliothek.

Die Optionen `-f`, `-l`, `-opw` und `-upw` haben dieselbe Bedeutung wie bei [`pdftops`](#) (siehe dort).

- `-layout`  
versucht, das Seitenlayout zu erhalten.
- `-nopgbrk`  
verzichtet darauf, das Seitenende mit einem Sonderzeichen zu kennzeichnen.

### **pidof programmname**

`pidof` ermittelt die Prozessnummern aller Instanzen eines namentlich genannten Prozesses.

- `-o %PPID`  
ignoriert den Elternprozess, bei Shell-Scripts also die Shell, in der das Script gerade ausgeführt wird.
- `-s`  
liefert nur den ersten passenden Prozess (*single shot*).

### **Beispiel**

Das folgende Kommando liefert die PIDs aller laufenden `bash`-Shells:

```
root# pidof /bin/bash  
32329 21636 21600 3351 1739 922
```

### **ping [optionen] adresse** **ping6 [optionen] adresse**

`ping` sendet einmal pro Sekunde ein Netzwerkpaket (ECHO\_REQUEST gemäß dem Protokoll ICMP) an die angegebene Adresse. Wenn die Adresse erreichbar ist und Echo-Pakete nicht durch eine Firewall blockiert werden, empfängt `ping` Antwortpakete und gibt an, wie lange die Kommunikation hin und zurück gedauert hat. Standardmäßig läuft `ping` unbeschränkt, bis es durch `[Strg]+[C]` beendet wird.

- **-4 bzw. -6**  
verwendet ausschließlich IPv4 bzw. IPv6. Alternativ können bei einigen Distributionen die Kommandos `ping4` bzw. `ping6` aufgerufen werden.
- **-c n**  
sendet nur  $n$  Pakete und endet dann.
- **-i n**  
gibt die Intervallzeit zwischen zwei Paketen in Sekunden an.  $n$  ist eine Fließkommazahl, d.h., auch `ping -i 0.1` ist möglich.
- **-n**  
zeigt nur die IP-Adresse, aber nicht den Hostnamen des Empfängers an.

### **pip** kommando

Das Kommando `pip` aus dem gleichnamigen Paket hilft dabei, Python-Pakete aus dem *Python Package Index* (PyPI) zu installieren. Auf vielen Distributionen sind die Python-Versionen 2 und 3 parallel nebeneinander installiert oder zumindest verfügbar. Dann gilt `pip` für Python-2-Pakete und `pip3` für Python-3-Pakete.

Im Gegensatz zu Linux-Paketverwaltungskommandos wird `pip` normalerweise ohne root-Rechte bzw. ohne [`sudo`](#) ausgeführt. Die Pakete stehen dennoch allen Benutzern zur Verfügung (es sei denn, Sie verwenden die gleich beschriebene Option `--user`).

- `install [--upgrade] [--user] paketname`  
installiert bzw. aktualisiert das angegebene Paket. Mit der Option `--user` wird das Paket so installiert, dass es nur dem aktuellen Benutzer zur Verfügung steht.
- `list [--outdated]`  
liefert eine Liste aller installierten Pakete bzw. aller Pakete, die aktualisiert werden können.
- `search suchbegriff`  
sucht in PyPI nach Paketen mit dem angegebenen Suchbegriff.

- `show [--files] paketname`  
zeigt Informationen zu einem installierten Paket an bzw. listet dessen Dateien auf.
- `uninstall paketname`  
entfernt das angegebene Paket.

## Beispiel

Mit den folgenden Kommandos wird zuerst `pip3` selbst und dann die `wiringpi`-Bibliothek auf einem Raspberry Pi so installiert, dass die Bibliothek für alle Nutzer zugänglich ist.

```
user$ sudo apt install pip3
user$ sudo pip3 install wiringpi2
Installing collected packages: wiringpi, wiringpi2
Successfully installed wiringpi-2.44.4 wiringpi2-2.32.3
```

## pkcon kommando

PackageKit ist eine allgemeine Schnittstelle zu den verschiedenen Linux-Paketverwaltungssystemen. PackageKit wird überwiegend von grafischen Benutzeroberflächen wie *Gnome Software* verwendet, kann aber auch über das Kommando `pkcon` gesteuert werden. Je nachdem, auf welches Backend PackageKit zurückgreift (APT, DNF, YUM etc.), kann es sein, dass manche der folgenden Kommandos nicht zur Verfügung stehen.

- `get-groups`  
ermittelt Paketgruppen.
- `get-updates`  
liefert eine Liste aller Pakete, die aktualisiert werden können.
- `install paketname`  
installiert ein Paket.
- `install-local datei`  
installiert eine lokale Datei.
- `refresh [force] [optionen]`  
aktualisiert den Cache mit den Paketinformationen. In der Form `pkcon refresh force -c -1` wird das zulässige Alter für die Cache-

Dateien auf kleiner null gesetzt. Das bewirkt, dass der gesamte Cache gelöscht wird, was eine Menge Platz sparen kann.

- `remove paketname`  
deinstalliert ein Paket.
- `search [name|details|group|file] suchbegriff`  
sucht nach Paketen.
- `update [paketname]`  
aktualisiert das angegebene Paket bzw. alle installierte Pakete.

### **popd**

Das bash-Kommando `popd` wechselt in ein zuvor mit [pushd](#) gespeichertes Verzeichnis zurück. Das Verzeichnis wird aus der Verzeichnisliste entfernt. `popd` und [pushd](#) werden üblicherweise nur in Shell-Scripts eingesetzt.

### **postconf [optionen] [parameter [=wert]]**

`postconf` hilft dabei, die Konfiguration des Mail-Servers Postfix auszulesen und im laufenden Betrieb zu verändern. Wenn das Kommando ohne Optionen aufgerufen wird, zeigt es einfach die aktuell gültige Einstellung der rund 1000 Parameter von Postfix an.

- `parameternname`  
zeigt den aktuellen Zustand des Parameters an.
- `parameternname=neuerwert`  
stellt den Parameter neu ein und führt die Änderung auch in der Konfigurationsdatei `/etc/postfix/main.cf` durch.
- `-d bzw. -df`  
zeigt den Defaultwert aller Parameter an (auch dann, wenn die Parameter aktuell anders eingestellt sind). Die Option `-df` anstelle von `-d` bewirkt, dass lange Einträge über mehrere Zeile umbrochen werden (*fold*).
- `-d parametername`  
zeigt den Defaultzustand des Parameters an (gegebenenfalls abweichend von der aktuellen Einstellung).

- **-n** bzw. **-nf**  
listet alle Parameter auf, die nicht den Defaultwert haben. **-nf** umbricht lange Zeilen.

### **postqueue [optionen]**

postqueue gibt Informationen über noch nicht ausgelieferte E-Mails des Mail-Servers Postfix bzw. veranlasst einen neuerlichen Sendevorschuss.

- **-f**  
versucht, alle E-Mails in der Warteschlange sofort neuerlich zu versenden (*flush*). Postfix kümmert sich darum an sich automatisch, hält zwischen mehreren Sendeversuchen aber Wartezeiten ein.
- **-i id**  
versucht, die angegebene E-Mail sofort nochmals zu versenden.
- **-p** bzw. **-j**  
zeigt alle in der Warteschlange befindlichen E-Mails an. Mit **-j** erfolgt die Ausgabe im JSON-Format.

### **powertop**

Das interaktive Programm **powertop** aus dem gleichnamigen Paket analysiert, welche Prozesse am meisten Energie beanspruchen bzw. die CPU oder Festplatte am häufigsten aus einem Ruhemodus aufwecken. Während das Programm läuft, können Sie mit zwischen mehreren Ergebnisseiten wechseln.

Die aus der Sicht der Energiesparfunktionen interessanteste Seite heißt **TUNABLES**. Dort zeigt **powertop** eine Liste von Einstellungen an, deren aktueller Zustand **BAD** bzw. **GOOD** sein kann. Mit den Cursortasten können Sie nun einzelne Punkte auswählen und durch umstellen. **powertop** zeigt dabei an, welches Kommando es ausführt.

Sie können nun schrittweise versuchen, einzelne Einstellungen zu ändern, und dann testen, welche Auswirkungen dies hat: Sinkt der

mit einem Strommessgerät erfasste Energieverbrauch des zuvor aufgeladenen Notebooks tatsächlich spürbar?

Verursachen die deaktivierten Funktionen Probleme? Lassen sich USB-Geräte weiterhin nutzen, funktioniert das Ein- und Ausschalten des WLAN-Adapters weiterhin, funktioniert das Audio-System ohne Störgeräusche etc.?

Die mit `powertop` durchgeführten Änderungen gelten nur bis zum nächsten Neustart des Rechners. Um die Energiesparmaßnahmen dauerhaft zu aktivieren, tragen Sie die von `powertop` angezeigten Kommandos (z.B. `echo '1' > /sys/xxx`) in eine Datei ein, die bei jedem Systemstart ausgeführt wird. Bei den meisten Distributionen eignet sich dazu `/etc/rc.d/rc.local`. Gegebenenfalls müssen Sie diese Datei erzeugen und mit `chmod a+x` als ausführbar kennzeichnen. Je nach Distribution müssen Sie auch `systemctl daemon-reload` ausführen, damit `systemd` die Datei berücksichtigt.

Eine radikale Lösung besteht darin, in `rc.local` anstelle einzelner Tuning-Kommandos `powertop --auto-tune` einzutragen. Dann führt `powertop` einfach alle bekannten Optimierungsmaßnahmen aus.

Leider schießt `powertop` damit oft über das Ziel hinaus: Was nützt es, wenn das Notebook eine Stunde länger als bisher läuft, aber die Netzwerkverbindung nur noch unzuverlässig funktioniert?

Eine Liste aller möglichen Tuning-Kommandos erhalten Sie, wenn Sie `powertop` mit der Option `--html` ausführen. Das Kommando erzeugt dann nach einer Messzeit von ca. 20 Sekunden die HTML-Datei `powertop.html`, die neben diversen statistischen Daten auch eine Zusammenfassung aller Tuning-Parameter gibt.

**ppa-purge** ppa:ppaowner [/ppaname]

Das Ubuntu-spezifische Kommando `ppa-purge` aus dem gleichnamigen Kommando deaktiviert die angegebene private Paketquelle (PPA, siehe auch die Beschreibung des Kommandos [add-apt-repository](#)). Sofern die PPA alternative Versionen offizieller

Pakete zur Verfügung stellt, werden diese Pakete entfernt und durch die Original-Pakete ersetzt.

## Beispiel

Das folgende Kommando deaktiviert die libreoffice-Paketquelle und ersetzt die von dieser Paketquelle stammenden Pakete durch Ubuntu-Standardpakete:

```
root# ppa-purge ppa:libreoffice/ppa  
printenv [variable]
```

Gibt den Inhalt der angegebenen Umgebungsvariablen bzw. aller Umgebungsvariablen zeilenweise aus.

```
printf format para1 para2 para3 ...
```

printf erlaubt es, Ausgaben in der Syntax des C-Kommandos printf zu formatieren. Detaillierte Informationen zu den Formatierungsmöglichkeiten erhalten Sie mit `man 3 printf`.

## Beispiel

Das folgende Kommando gibt eine ganze Zahl, eine Fließkommazahl mit zwei Nachkommastellen sowie ein Zeilenende-Zeichen aus. Wenn Sie das Kommando auf einem System mit englischer Lokalisierung ausführen, müssen Sie die Fließkommazahl mit einem Dezimalpunkt anstelle des Kommas angeben.

```
root# printf "%d %.2f\n" 123 3,1415927  
123 3,14
```

```
ps [optionen]
```

ps zeigt die Liste der laufenden Prozesse (Programme) und Kernel-Threads an. Das Kommando ist insbesondere im Zusammenspiel mit [kill](#) sehr praktisch, um hängende Programme gewaltsam zu beenden. ps ist mit zahllosen Optionen ausgestattet, die im Online-Manual (`man ps`) ausführlich beschrieben sind. Dort finden Sie auch Erklärungen dazu, was die zahlreichen Informationen bedeuten, die von ps ausgegeben werden. Eine Variante zu ps ist [pstree](#), das den Prozessbaum anzeigt und so auf einen Blick die Hierarchie der Prozesse klarmacht.

Im Gegensatz zu anderen Kommandos gibt es bei ps Optionen mit und ohne vorangehenden Bindestrich. Diese haben teilweise sogar eine unterschiedliche Bedeutung. (ps -a und ps a sind nicht gleichwertig!) Beide Optionstypen können in Gruppen gemischt werden, z.B. ps -A ul.

- a  
zeigt auch Prozesse anderer Benutzer an (nicht nur die eigenen).
- -A  
zeigt alle Prozesse an.
- f  
zeigt den Prozessbaum an.
- -f -l oder l  
zeigt diverse Zusatzinformationen an (Speicherbedarf, Priorität etc.).
- -p n oder -p n1,n2,n3  
zeigt nur Prozesse mit den angegebenen Prozessnummern (PIDs) an.
- --ppid n  
zeigt nur Prozesse an, die n als Parent-Prozess-ID haben, die also vom Prozess n gestartet wurden.
- u  
zeigt auch den Namen der Benutzer der jeweiligen Prozesse an.  
Die Option kann nicht mit l kombiniert werden.
- x  
zeigt auch Prozesse an, denen kein Terminal zugeordnet ist. Dabei handelt es sich unter anderem um Linux-interne Prozesse zur Verwaltung des Systems (engl. *Daemons*).
- z  
zeigt zusätzlich den SELinux-Kontext der Prozesse an.

## Beispiele

Die folgenden Beispiele zeigen elementare Anwendungen des ps-Kommandos:

```
root# ps ax          (zeigt alle Prozesse)
root# ps ax | grep ssh (zeigt Prozesse, in deren Beschreibung ssh vorkommt)
root# ps aux          (zeigt alle Prozesse inklusive Besitzer/Account)
```

Unter Linux hat der Kernel-Thread-Dämon kthreadd immer die Prozessnummer 2. Alle weiteren Kernel-Threads werden von kthreadd gestartet. Wenn Sie also nur die Kernel-Threads auflisten möchten, verwenden Sie ps --ppid 2:

```
root# ps --ppid 2      (zeigt nur Kernel-Threads)
```

```
ps2pdf [optionen] quelle.ps [ziel.pdf]
```

ps2pdf erzeugt aus einer beliebigen PostScript- oder EPS-Datei eine PDF-Datei. Wenn Sie auf die Angabe von ziel.pdf verzichten, bekommt die PDF-Datei denselben Namen wie die PostScript-Datei (aber die Kennung .pdf, also *quelle.pdf*).

- -dEncodeColorImages=false  
verhindert jegliches Komprimieren von Bildern. Die resultierenden PDF-Dateien werden nun sehr groß. In Kombination mit der /prepress-Einstellung liefert ps2pdf nun PDFs in optimaler Druck- bzw. Belichtungsqualität.
- -dPDFSETTINGS=/default | /screen | /printer | /prepress  
gibt an, ob das PDF-Dokument für eine beliebige Verwendung, für die Bildschirmdarstellung, für einen gewöhnlichen Ausdruck oder für eine Belichtung (Buchdruck etc.) optimiert werden soll. Der Preis für die höhere Qualität ist die zunehmende Dateigröße. Diese vier Voreinstellungen ersparen es Ihnen, zahlreiche Optionen einzeln einstellen zu müssen.
- -rn  
gibt die Auflösung für Bitmap-Fonts an (Einheit DPI, also *dots per inch*).

Unzählige weitere Optionen sind hier beschrieben:

<http://web.mit.edu/ghostscript/www/Ps2pdf.htm>

## Beispiel

Das folgende Kommando erzeugt aus der PostScript-Datei *buch.ps* das PDF-Dokument *buch.pdf*:

```
user$ ps2pdf buch.ps
pssh [optionen] kommando
pscp [optionen] lokale_datei entfernte_datei
pnuke [optionen] muster
```

Das Python-Script pssh aus dem gleichnamigen Paket führt ein Kommando parallel auf mehreren Servern aus. pscp kopiert Dateien von allen oder auf alle angegebenen Hosts. pnuke beendet auf allen Hosts mit `kill -9` ein Programm, dessen Name oder Muster angegeben wurde. Alle drei Kommandos setzen voraus, dass eine Authentifizierung mit Schlüsseln möglich ist oder dass sich das Hintergrundprogramm ssh-agent um die Authentifizierung kümmert.

- `-h hosts.txt`  
gibt den Ort der Textdatei an, die zeilenweise die Hostnamen der Rechner enthält, auf denen das Kommando ausgeführt wird.
- `-o verzeichnis`  
gibt ein Verzeichnis an, in dem pssh die Ausgaben der Kommandos bzw. pscp die zu kopierenden Dateien speichert. pssh und pscp richten für jeden Host eigene Ausgabedateien ein, aus deren Namen der Host hervorgeht.

## Beispiel

Das Beispiel geht davon aus, dass es sich bei allen Hosts um CentOS- oder RHEL-Systeme handelt. Auf diesen Rechnern wird nun mit [yum](#) ein Software-Update durchgeführt.

```
root# pssh -h hosts.txt yum update
pstree [optionen] [pid]
```

pstree aus dem Paket psmisc oder pstree gibt einen Baum mit allen Prozessen auf dem Bildschirm aus. Der Baum macht deutlich, welcher Prozess von welchem anderen Prozess gestartet wurde.

Wenn eine Prozessnummer angegeben wird, beginnt der Baum an

dieser Stelle, andernfalls beginnt er bei [init](#) oder [systemd](#), also dem ersten Prozess, der beim Systemstart ausgeführt wird.

- **-h**  
kennzeichnet den aktuellen Prozess und all seine Parent-Prozesse.
- **-p**  
zeigt zu jedem Prozess auch die Prozessnummer (PID).
- **-u**  
zeigt bei allen Prozessen, deren UID (User-ID) von der UID des Parent-Prozesses abweicht, den Benutzer- oder Account-Namen.

### **pushd** verzeichnis

Das bash-Kommando **pushd** speichert das aktuelle Verzeichnis und wechselt anschließend in das angegebene Verzeichnis. Mit [popd](#) gelangen Sie in das ursprüngliche Verzeichnis zurück. [dirs](#) zeigt die Liste der gespeicherten Verzeichnisse an. **pushd** und [popd](#) werden überwiegend in Shell-Scripts eingesetzt.

### **pvcreate** [optionen] device

Das LVM-Kommando **pvcreate** deklariert eine Partition bzw. ein Device als Physical Device (PV) zur späteren Nutzung in einer Volume Group (VG) (siehe auch [vgcreate](#) und [vgextend](#)).

**pvcreate** setzt voraus, dass die Partition bzw. das Device vorher als LVM-Partition gekennzeichnet wurde. In [fdisk](#) verwenden Sie dazu das Kommando **T** und den Code 8e. Bei [parted](#) lautet das erforderliche Kommando `set partitionsnummer lvm on`.

### **Beispiel**

Das folgende Kommando macht die Partition `/dev/sdc1` zu einem Physical Volume für LVM:

```
root# pvcreate /dev/sdc1
Physical volume "/dev/sdc1" successfully created
```

### **pvdisplay** device

**pvdisplay** zeigt Detailinformationen zum angegebenen PV an.

### **pvremove** device

`pvremove` entfernt die PV-Kennzeichnung eines ungenutzten PVs.

### **pvscan**

`pvscan` listet alle PVs auf.

### **pwd**

Das bash-Kommando `pwd` gibt das aktuelle Verzeichnis an.

Neben der in die bash inkludierten Version steht `pwd` auch als eigenständiges Kommando zur Verfügung, oft unter dem Dateinamen `/bin/pwd`. Beachten Sie, dass es Fälle geben kann, in denen die beiden Varianten von `pwd` nicht zum gleichen Ergebnis kommen. Das liegt daran, dass die bash-Version von `pwd` relativ arbeitet, das externe Kommando aber absolut. Symbolische Links auf Verzeichnisse können die bash-Version von `pwd` gewissermaßen überlisten.

### **Beispiel**

Im folgenden Beispiel richtet [`ln`](#) einen symbolischen Link auf das Verzeichnis `/tmp` ein. [`cd`](#) wechselt in dieses Verzeichnis. Die bash glaubt, dass dieses Verzeichnis das aktuelle Verzeichnis ist, während `/bin/pwd` erkennt, dass tatsächlich `/tmp` das aktuelle Verzeichnis ist.

```
user$ ln -s /tmp symlink
user$ cd symlink
user$ pwd          (bash-Version von pwd)
/home/kofler/symlink
user$ /bin/pwd      (eigenständiges pwd-Kommando)
/tmp
```

### **pwgen [optionen] [n]**

`pwgen` aus dem gleichnamigen Paket generiert eine ganze Liste zufälliger, aber leicht merkbarer Passwörter. Die Idee ist die, dass der Benutzer eines der Passwörter auswählt und dann den Bildschirm mit `[Strg]+[L]` löscht, bevor ihm jemand über die Schulter blicken und das Passwort mitlesen kann. Im Vergleich zu den von [`makepasswd`](#) erzeugten Passwörtern sind die `pwgen`-Passwörter weniger zufällig und somit auch weniger sicher, aber für viele Anwendungsfälle sind sie immer noch ausreichend gut.

Die Länge der Passwörter beträgt standardmäßig acht Zeichen. Längere Passwörter generieren Sie, indem Sie den optionalen Parameter *n* angeben.

- -0 verwendet in den Passwörtern keine Ziffern.
- -1 liefert nur ein Passwort und nicht gleich 100.
- -B vermeidet Buchstaben bzw. Ziffern, die je nach Schriftart leicht zu verwechseln sind, also O und 0 sowie l und 1. Das macht die Passwörter unsicherer.
- -y baut in die Passwörter auch Sonderzeichen ein.

## **Beispiel**

Wie das Beispiel beweist, sind bereits die in den Standardeinstellungen erzeugten Passwörter keineswegs trivial:

```
user$ pwgen  
0hshu3yo Ea1wedoe OhCh2Zua Aili8ooc Xu4iiyix eug3Chee Gaesh2pu Eeth6mah  
eMee0jio xieL6oob ob8uYah9 shaif0Ed uep8Eive lang3Eho thaiS7xa Sah3See4  
...
```

Das folgende Kommando erzeugt ein 80 Zeichen langes Passwort, das auch Sonderzeichen enthält:

```
root# pwgen -1 -y 80  
AhB9eiv]oo~s0go2ievaF7eePe6ooyoo7Cim%oh8eishiloosaeSeex+eeza9Iem7Ahchuh[i1
```

## Q

### **qalc** [ausdruck]

qalc aus dem gleichnamigen Paket berechnet den übergebenen mathematischen Ausdruck und liefert das Ergebnis zurück. Um die Verarbeitung von Sonderzeichen durch die bash zu vermeiden, müssen Sie die meisten Ausdrücke in Anführungszeichen stellen:

```
user$ qalc "(2+3)*4^5"  
(2 + 3) * (4^5) = 5120
```

Wenn kein Ausdruck übergeben wird, aktiviert qalc einen interaktiven Modus. Es führt dann so lange Berechnungen aus, bis das Kommando mit **Strg**+**D** beendet wird.

```
user$ qalc  
>2^64  
2^64 = approx. 1.8446744E19  
>sin(0.2)  
sin(0.2 * radian) = approx. 0.19866933  
>sin(pi)  
sin(pi * radian) = 0  
>sqrt(2)  
sqrt(2) = approx. 1.4142136  
>10 EUR to USD  
10 * euro = $10.651
```

### **qemu-img** kommando

qemu-img hilft bei der Erzeugung und Verwaltung von Image-Dateien für virtuelle Festplatten. Als erster Parameter muss ein Befehl angegeben werden, z.B. [create](#) oder [convert](#). Anschließend folgen je nach Befehl weitere Optionen, der Dateiname der Image-Datei etc.

- `convert [-f quellformat] -o zielformat quelldatei zieldatei`  
wandelt eine Image-Datei von einem Format in ein anderes um.  
Die ursprüngliche Datei bleibt dabei erhalten. Die Image-Datei darf aber während der Umwandlung nicht von einer virtuellen Maschine genutzt werden. Mit der zusätzlichen Option -c wird die Zieldatei komprimiert.
- `convert -f qcow2 -s snapshotname -o zielformat quelldatei zieldatei`

überträgt nur den mit -s ausgewählten Snapshot eines QCOW2-Images in die neue Image-Datei.

- `create [-f raw/qcow2/qed] [-o opt1=val1,opt2=val2,...] datei  
größe`  
erzeugt eine neue Image-Datei im angegebenen Typ und in der gewünschten Größe. Die Größenangabe erfolgt standardmäßig in Byte. Optional können die Suffixe `k` bzw. `K`, `M`, `G` oder `T` verwendet werden (für KiB, MiB, GiB oder TiB). Je nach Image-Format können diverse Zusatzoptionen angegeben werden. Die zur Auswahl stehenden Parameter ermitteln Sie mit `-o '?'`, wobei diese Option in einem vollständigen, syntaktisch korrekten Kommando angegeben werden muss:  
`qemu-img create -f qcow2 -o '?' test.img 1G`
- `create -f qcow2 -o backing_file=basis.img overlay.img`  
erzeugt eine Overlay-Datei. Bei der Ausführung der virtuellen Maschine werden alle Änderungen in der Overlay-Datei gespeichert; die Basis-Datei bleibt unverändert.
- `info datei`  
liefert Informationen über die angegebene Image-Datei. Informationen über Snapshots in QCOW2-Images erhalten Sie mit `qemu-img snapshot` (siehe unten).
- `resize datei +/-größe`  
vergrößert bzw. verkleinert die Image-Datei um die angegebene Größe. Das Kommando ist nur für RAW- und QCOW2-Images geeignet.
- `snapshot [optionen] snapshotname datei`  
bearbeitet Snapshots. Dieses Kommando ist nur für QCOW2-Images verfügbar. `-c` erzeugt einen neuen Snapshot, `-a`wendet einen Snapshot auf das Image an (widerruft also alle Änderungen, die seit der Erstellung des Snapshots erfolgt sind), `-d` löscht den Snapshot, und `-l` listet alle Snapshots auf. Mit Ausnahme von `-l` dürfen Sie diese Kommandos allerdings nicht nutzen, während

die Image-Datei von einer laufenden virtuellen Maschine genutzt wird!

## Beispiele

Das erste Kommando erzeugt eine QCOW2-Image-Datei. Ein großer Vorteil dieses Image-Formats besteht darin, dass die Datei anfänglich kaum Speicher auf dem Datenträger beansprucht und erst nach und nach wächst.

```
root# qemu-img create -f qcow2 disk.img 10G
```

Das zweite Kommando erzeugt aus einer QCOW2-Datei ein gleichwertiges Raw-Image:

```
user$ qemu-img convert -f qcow2 image.qcow2 -O raw image.raw
```

**qemu-kvm** [optionen] [imagedatei]

qemu-kvm führt eine virtuelle Maschine aus. Das Kommando steht je nach Distribution unter dem Namen [kvm](#) zur Verfügung und ist in diesem Buch dort dokumentiert.

**qemu-nbd** [optionen] imagedatei

qemu-nbd bietet eine Image-Datei als Network Block Device an. [kvm](#) auf einem anderen Rechner kann dann über das Netzwerk auf das Image zugreifen, z.B. mit `kvm -hda nbd:nbd-hostname:1024`.

- `-b n.n.n.n`  
verbindet die Netzwerkschnittstelle mit der angegebenen IP-Adresse. Standardmäßig gilt die Adresse 0.0.0.0, d.h., der Netzwerkzugriff ist nicht auf das lokale Netzwerk limitiert.
- `-c device`  
verbindet die Image-Datei mit der angegebenen Device-Datei (*connect*).
- `-d device`  
löst die Verbindung zur angegebenen Device-Datei (*disconnect*).
- `-p n`  
verwendet den Port n (standardmäßig 1024).

## Beispiel

`qemu-nbd` wird hier dazu verwendet, um ein QCOW2-Image über eine NBD-Device-Datei anzubieten. Das vereinfacht die Verwendung des Images auf dem lokalen Rechner, ist aber für den Netzwerkeinsatz ungeeignet. Mit [parted](#) kann der virtuelle Datenträger nun wie eine lokale Festplatte bearbeitet werden. Das letzte Kommando gibt die Device-Datei wieder frei.

```
root# qemu-img create -f qcow2 disk.img 1G
root# modprobe nbd
root# qemu-nbd -c /dev/nbd0 disk.img &
root# parted /dev/nbd0
...
root# qemu-nbd -n /dev/nbd0
```

# R

## **raspi-gpio** kommando

Das Kommando ermittelt bzw. verändert den Zustand der GPIO-Pins des Raspberry Pi. Es ersetzt das veraltete [gpio](#)-Kommando des WiringPi-Projekts.

- **funcs [nr]**  
zeigt an, welche Funktionen der angegebene GPIO übernehmen kann. Wenn *nr* fehlt, liefert **raspi-gpio func** diese Daten für alle GPIOs.  
*nr* bezieht sich auf die Nummerierung der GPIOs gemäß der Dokumentation des Herstellers Broadcom (entspricht [gpio -h](#)). Die Nummern stimmen nicht mit den Pin-Nummern des J8-Headers überein!
- **get [nr]**  
zeigt den Zustand aller GPIOs bzw. des durch *nr* angegebenen GPIOs an.
- **help**  
zeigt einen relativ ausführlichen Hilfetext an. Eine [man](#)-Seite steht leider nicht zur Verfügung.
- **set nr zustand**  
aktiviert den gewünschten Zustand für den GPIO. Mögliche Werte sind:  
[ip](#) = Input  
[op](#) = Output  
[dl](#) = Zustand 0 (drive low)  
[dh](#) = Zustand 1 (drive high)  
[a0](#) bis [a5](#) = alternative Funktion 0 bis 5  
[pu](#) = Pull-up-Widerstand aktiv  
[pd](#) = Pull-down-Widerstand aktiv  
[pn](#) = Pull-up/down-Widerstand deaktiviert (no pull)

Soweit sich sinnvolle Kombinationen ergeben, dürfen mehrere der obigen Schlüsselwörter auf einmal genannt werden, jeweils getrennt durch Leerzeichen. Welche alternativen Funktionen ein GPIO unterstützt und wie diese nummeriert sind, geht aus `raspi-gpio func` hervor.

## Beispiel

Das erste Kommando ermittelt, welche Funktionen der GPIO mit der Nummer 23 unterstützt. Auf dem Raspberry Pi ist dieser GPIO mit Pin 16 des J8-Headers verbunden. Das zweite Kommando nutzt diesen GPIO als simplen Signalausgang, wobei der Pegel auf High gesetzt wird (3,3 V). Eine über einen Vorwiderstand mit Pin 16 verbundene LED würde nun leuchten. Das dritte Kommando verifiziert den Zustand von GPIO 23.

```
pi$ raspi-gpio funcs 23
GPIO, DEFAULT PULL, ALT0, ALT1, ALT2, ALT3, ALT4, ALT5
23, DOWN, SD0_CMD, SD15, DPI_D19, SD1_CMD, ARM_RTCK, -
pi$ raspi-gpio set 23 op dh
pi$ raspi-gpio get 23
GPIO 23: level=1 fsel=1 func=OUTPUT
```

## **raspistill [optionen]**

Das unter Raspberry Pi OS verfügbare Kommando `raspistill` nimmt ein Foto mit der Kamera auf, die als Erweiterung für den Raspberry Pi erhältlich ist.

- `-e jpg|bmp|gif|png`  
legt das gewünschte Bildformat fest (standardmäßig gilt `jpg`).
- `-h n`  
gibt die Bildhöhe an (standardmäßig 1944 Pixel).
- `-n`  
zeigt während der Aufnahme kein Vorschaufenster an.
- `-o datei`  
gibt den Namen der Bilddatei an.
- `-q n`  
gibt die JPEG-Bildqualität an (zwischen 0 = minimal und 100 = maximal).

- **-rot *n***  
gibt die gewünschte Drehung des Bilds in Grad an (0 bis 359).
- **-t *n***  
gibt an, nach welcher Zeitspanne in Millisekunden das Foto erstellt werden soll (standardmäßig nach 5000 ms, also nach 5 Sekunden). Um das Bild schnellstmöglich zu machen, geben Sie **-t 1** an. Die naheliegende Einstellung **-t 0** funktioniert nicht.
- **-tl *n***  
gibt an, nach wie vielen Millisekunden jeweils weitere Fotos gemacht werden sollen.
- **-w *n***  
gibt die Bildbreite an (standardmäßig 2592 Pixel).

Zur Steuerung der Bildqualität stehen unter anderem die folgenden Optionen zur Auswahl:

- **-br 0 bis 100**  
stellt die gewünschte Helligkeit ein.
- **-co -100 bis 100**  
steuert den Kontrast.
- **-sa -100 bis 100**  
gibt ein Maß für die Farbsättigung an.
- **-sh -100 bis 100**  
gibt die gewünschte Bildschärfe an.

## Beispiel

Das folgende Kommando erstellt bei minimaler Wartezeit und ohne die Anzeige eines Vorschaufensters ein Foto in einer Auflösung von 800×600 Pixel:

```
pi$ raspistill -n -t 1 -w 800 -h 600 -o foto.jpg
```

**raspivid** [optionen]

Das unter Raspberry Pi OS verfügbare Kommando **raspivid** nimmt ein Video im H264-Format mit der Kamera auf, die als Erweiterung für den Raspberry Pi erhältlich ist. Die Auflösung von Videos beträgt standardmäßig 1920×1080 Pixel.

Viele Steuerungsoptionen stimmen mit dem oben beschriebenen Kommando [raspistill](#) überein, unter anderem -br, -co, -h, -n, -o, -rot, -sa, -sh und -w. An dieser Stelle konzentriere ich mich auf zwei raspivid-spezifische Optionen:

- **-fps *n***  
gibt an, wie viele Frames pro Sekunde aufgenommen werden sollen (standardmäßig 30, bei einer reduzierten Auflösung sind bis zu 90 Frames pro Sekunde möglich).
- **-t *n***  
gibt die gewünschte Länge des Videos in Millisekunden an (standardmäßig 5000, also 5 Sekunden).

## Beispiel

Das folgende Kommando nimmt ein Video in der Länge von 30 Sekunden bei einer Auflösung von 1024×768 Pixeln auf:

```
pi$ raspivid -n -w 1024 -h 768 -t 30000 -o video.h264
```

```
rdfind [optionen] verzeichnis1 verzeichnis2
```

**rdfind** aus dem gleichnamigen Paket durchsucht zwei Verzeichnisse nach darin enthaltenen Doppelgängern (also identischen Dateien). Das Kommando unterstützt verschiedene Verfahren, wie es die Gleichheit zweier Dateien feststellt und wie es beurteilt, welche Datei das Original und welche Datei das Duplikat ist. Je nach Optionen werden die Doppelgänger aufgelistet, durch Links ersetzt oder sogar gelöscht.

Die folgenden Optionen beeinflussen das Suchverhalten:

- **-checksum md5|sha1|sha256**  
berechnet bei zwei gleich großen Dateien Prüfsummen mit dem angegebenen Verfahren. Standardmäßig kommt sha1 zum Einsatz.
- **-followsymlinks**  
berücksichtigt beim Vergleich symbolische Links.
- **-ignoreempty**  
ignoriert leere Dateien (mit einer Dateilänge von 0).

- `-minsize n`  
ignoriert Dateien, die kleiner als  $n$  Byte sind.

Mit weiteren Optionen legen Sie fest, was mit Duplikaten passieren soll. Standardmäßig werden deren Namen in die Datei *results.txt* geschrieben.

- `-deleteduplicates true|false`  
löscht Duplikate (Default: `false`).
- `-dryrun true|false`  
läuft im Testmodus, führt keine Änderungen durch (Default: `false`).
- `-makesymlinks true|false`  
ersetzt das Duplikat durch einen symbolischen Link auf das Original (Default: `false`).
- `-makehardlinks true|false`  
ersetzt das Duplikat durch einen festen Link (Default: `false`).
- `-makesresultfile true|false`  
schreibt die Dateinamen übereinstimmender Dateien in *results.txt* (Default: `true`). Die Ergebnisdatei enthält nicht einfach nur die Dateinamen beider Versionen jedes Duplikats, sondern auch Schlüsselwörter, die Aufschluss geben, welche Datei `rdfind` als Original betrachtet:

`DUPTYPE_FIRST_OCCURRENCE`: Original

`DUPTYPE_WITHIN SAME_TREE`: Duplikat in `verzeichnis1`

`DUPTYPE_OUTSIDE_TREE`: Duplikat in `verzeichnis2`

## Beispiele

Das folgende Beispiel ermittelt alle identischen Dateien in den Verzeichnissen für die 4. und 5. Auflage dieses Buchs:

```
user$ rdfind li-ref4 li-ref5
Now scanning "li-ref4", found 211 files.
Now scanning "li-ref5", found 215 files.
Now have 426 files in total.
Removed 0 files due to nonunique device and inode.
Total size is 85754591 bytes or 82 MiB
Removed 103 files due to unique sizes from list. 323 files left.
Now eliminating candidates based on first bytes:
```

```
removed 4 files from list. 319 files left.  
Now eliminating candidates based on last bytes:  
removed 0 files from list. 319 files left.  
Now eliminating candidates based on sha1 checksum:  
removed 2 files from list. 317 files left.  
It seems like you have 317 files that are not unique  
Totally, 39 MiB can be reduced. Now making results file results.txt.
```

Das zweite Kommando durchsucht zwei Verzeichnisse mit Fotos. Duplikate, die zumindest 100 Byte groß sind, werden durch feste Links ersetzt:

```
user$ rdiff -minsize 100 -makehardlinks true fotos1 fotos2
```

**rdiff-backup** [optionen] quellverzeichnis zielverzeichnis

rdiff-backup synchronisiert ähnlich wie [rsync](#) das Zielverzeichnis mit dem Quellverzeichnis. Im Unterschied zu [rsync](#) archiviert rdiff-backup bei wiederholter Ausführung auch alte Versionen von geänderten bzw. gelöschten Dateien, wobei aus Platzgründen nur die Änderungen in komprimierter Form gespeichert werden, also in Form inkrementeller Backups. Das ermöglicht die Rekonstruktion von irrtümlich gelöschten oder überschriebenen Dateien.

Das Quell- und das Zielverzeichnis können sich auf Netzwerk-Servern befinden. In diesem Fall erfolgt die Kommunikation standardmäßig via [ssh](#). Das Kommando rdiff-backup muss auch auf den externen Rechnern installiert sein. Bei der Angabe externer Verzeichnisse gilt nahezu dieselbe Syntax wie bei [rsync](#). Der einzige Unterschied besteht darin, dass nach dem Hostnamen *zwei* Doppelpunkte angegeben werden müssen, also z.B. in der Form user@backupserver::verzeichnis.

- **-r zeit bzw. --restore-as-of zeit**

rekonstruiert die Daten, wie sie zum gegebenen Zeitpunkt waren. Die Wiederherstellung alter Dateien verursacht allerdings mit zunehmender Versionsanzahl einen erheblichen CPU-Aufwand und ist entsprechend langsam!

- **--remove-older-than zeit**

löscht inkrementelle Backup-Dateien, die älter als der angegebene Zeitpunkt sind. Den Zeitpunkt können Sie wahlweise absolut

(z.B. 2020-12-31) oder relativ in Stunden (h), Tagen (d), Wochen (w) etc. angeben (siehe auch `man rdiff-backup` im Abschnitt TIME FORMATS). Statt eines konkreten Zeitpunkts können Sie mit `nB` auch angeben, wie viele Backup-Versionen maximal archiviert bleiben sollen. Standardmäßig löscht `rdiff-backup` immer nur *eine* Backup-Version. Wenn Sie mehrere Backup-Versionen auf einmal löschen möchten, müssen Sie zusätzlich die Option `--force` angeben.

## Beispiele

Das folgende Kommando erstellt ein Backup von `/home` im Verzeichnis `/home-backup`. Wenn dieses Kommando regelmäßig ausgeführt wird (z.B. täglich), werden automatisch auch alle Änderungen archiviert. Der Platzbedarf im Backup-Verzeichnis wächst damit entsprechend an.

```
root# rdiff-backup /home /home-backup
```

Das folgende Kommando rekonstruiert die gesicherten Daten in ein temporäres Verzeichnis:

```
root# rdiff-backup -r now /home-backup /tmp/home-aktuell
```

Das folgende Kommando stellt den Zustand des `/home`-Verzeichnisses so wieder her, wie er vor zehn Tagen war:

```
root# rdiff-backup -r 10D /home-backup/ /tmp/home-historisch
```

```
read [var1 var2 var3 ...]
```

`read` liest in bash-Scripts eine Zeile Text in die angegebenen Variablen. `read` erwartet die Daten aus der Standardeingabe. Wenn keine Variable angegeben wird, schreibt `read` die Eingabe in die Variable `REPLY`. Wenn genau eine Variable angegeben wird, schreibt `read` die gesamte Eingabe in diese eine Variable. Wenn mehrere Variablen angegeben werden, schreibt `read` das erste Wort in die erste Variable, das zweite Wort in die zweite Variable ... und den verbleibenden Rest der Eingabe in die letzte Variable. Wörter werden dabei durch Leer- oder Tabulatorzeichen getrennt.

Das `read`-Kommando sieht keine Möglichkeit vor, einen Infotext als Eingabeaufforderung auszugeben. Deswegen ist es zweckmäßig, den

Anwender vor der Ausführung von read-Kommandos mit echo -n über den Zweck der Eingabe zu informieren.

### **readonly**

Das bash-Kommando readonly zeigt die schreibgeschützten Variablen der Shell an. Variablen können mit [declare](#) -r vor Veränderungen geschützt werden.

### **reboot [optionen]**

reboot beendet alle laufenden Prozesse und startet den Rechner anschließend neu. reboot entspricht shutdown -r now.

### **recode zeichensatz1..zeichensatz2 datei recode zeichensatz1..zeichensatz2 < quelle > ziel**

recode führt eine Zeichensatzkonvertierung von Zeichensatz 1 nach Zeichensatz 2 durch. recode -l liefert eine umfangreiche Liste aller unterstützten Zeichensätze.

## **Beispiele**

Das folgende Kommando konvertiert die DOS-Datei *dosdat* in eine Linux-Datei mit dem Latin-1-Zeichensatz:

```
user$ recode ibmpc..latin1 < dosdat > linuxdat
```

Das zweite recode-Beispiel ersetzt in der Datei *windowsdat* alle Zeilenenden (CR plus LF, also *Carriage Return* und *Line Feed*) durch das unter Linux übliche Zeilenende (nur LF). Der eigentliche Zeichensatz wird nicht geändert. Die resultierende Datei wird in *linuxdat* gespeichert.

```
user$ recode latin1/cr-lf..latin1 < windowsdat > linuxdat
```

recode liest die im Zeichensatz Latin-1 codierte Textdatei *latin1dat* und speichert sie als UTF-8-Datei (Unicode):

```
user$ recode latin1..u8 < latin1dat > utf8dat
```

### **rename**

Bei den meisten Distributionen ist ein rename-Kommando vorinstalliert, das bei der Umbenennung mehrerer Dateien hilft. Allerdings kommen je nach Distribution unterschiedliche Varianten mit unterschiedlicher Syntax zur Anwendung. Im Folgenden beschreibe ich die beiden gängigsten Varianten, zuerst das unter

Debian, Raspberry Pi OS und Ubuntu übliche Perl-Script aus dem Paket `rename` und dann das unter CentOS, Fedora, RHEL und SUSE gängige Kommando `rename` aus dem Paket `util-linux`.

## **rename unter Debian, Raspberry Pi OS und Ubuntu**

Die Syntax dieser `rename`-Variante sieht so aus:

```
rename [optionen] rename_expression dateien
```

Dabei gibt die `rename_expression` in einer [sed](#)-ähnlichen Syntax an, wie mit den Dateien verfahren werden soll, die in den weiteren Parametern genannt sind. Die Details der Syntax können Sie mit `man perlexpr` nachlesen. Ich beschränke mich an dieser Stelle auf zwei Beispiele:

- `-f` bzw. `-force`  
überschreibt vorhandene Dateien.
- `-n`  
zeigt an, welche Dateien umbenannt würden, führt aber keine Änderungen durch.

Das folgende Beispiel benennt alle `*.jpg`-Dateien in `*.jpeg`-Dateien um:

```
user$ rename -n s/^.jpg/.jpeg/ *.jpg
```

Das zweite Kommando ersetzt alle Großbuchstaben durch Kleinbuchstaben:

```
user$ rename -n y/A-Z/a-z/ *
```

## **rename unter CentOS, Fedora, RHEL und SUSE**

Die `rename`-Syntax unter CentOS, Fedora, RHEL und SUSE ist simpler:

```
rename [optionen] find replace dateien
```

Das Kommando ersetzt einfach den Ausdruck [find](#) im Dateinamen durch `replace`. Es gibt keine Möglichkeit, die Wirkung von `rename` vorweg auszuprobieren.

- `-s`  
verändert auch die Namen von Dateien, auf die symbolische Links zeigen.
- `-v`  
zeigt die durchgeführten Namensänderungen an.

Um wieder *.jpg* in *.jpeg*-Dateien umzubenennen, führen Sie `rename` so aus:

```
user$ rename jpg jpeg *.jpg  
renice n pid
```

`renice` verändert die Priorität des Prozesses mit der angegebenen PID-Nummer. *n* ist wahlweise die neue Priorität (ein Wert zwischen -20 und 20) oder ein Deltawert mit positivem oder negativem Vorzeichen (z.B. +3 oder --2). Nur root darf die Priorität von Prozessen erhöhen. Die höchste Priorität ist --20, die kleinste ist 20.

**repoquery [optionen]**

Das ehemals eigenständige Kommando `repoquery` ist mittlerweile als Subkommando von [dnf](#) implementiert. `dnf repoquery` ermittelt Metadaten zu Paketen aus YUM-Paketquellen.

**reset**

`reset` stellt die Schrift im Textterminal wieder her, wenn diese durch die Ausgabe von Sonderzeichen zerstört worden ist. `reset` setzt auch die Terminalerstellungen auf die früher mit `setterm -store` gespeicherte Grundeinstellung zurück.

**resize2fs [optionen] device [größe]**

`resize2fs` verändert die Größe eines ext2/ext3/ext4-Dateisystems. Für die Größenangabe sind die Schreibweisen *nK*, *nM* und *nG* für KiB, MiB oder GiB zulässig. Wenn Sie keine Größe angeben, passt `resize2fs` das Dateisystem an die Größe der zugrunde liegenden Partition bzw. des Logical Volumes an (ideal für Vergrößerungen). Vergrößerungen des Dateisystems sind im laufenden Betrieb möglich. Für eine Verkleinerung müssen Sie das Dateisystem aushängen und vorher [fsck](#) ausführen. Beachten Sie, dass Sie bei einer Vergrößerung *vorher* die zugrunde liegende Partition oder das LV vergrößern müssen, bei einer Verkleinerung die Partition oder das LV aber erst *nachher* verkleinern dürfen!

## Beispiel

Im folgenden Beispiel wird zuerst ein Logical Volume mit [lvextend](#) vergrößert. Anschließend wird das darin enthaltene Dateisystem mit [resize2fs](#) vergrößert:

```
root# lvextend -L 40G /dev/mapper/vg1-test
  Extending logical volume test to 40,00 GB
  Logical volume test successfully resized
root# resize2fs /dev/mapper/vg1-test
Das Dateisystem auf /dev/mapper/vg1-test ist auf /test eingehängt;
  Online-Größenveränderung nötig
Führe eine Online-Größenänderung von /dev/mapper/vg1-test
  auf 10485760 (4k) Blöcke durch.
Das Dateisystem auf /dev/mapper/vg1-test ist nun 10485760 Blöcke groß.
```

### **restorecon [optionen] dateien**

restorecon stellt den für ein Verzeichnis vorgesehenen SELinux-Kontext wieder her. Das ist erforderlich, wenn SELinux vorübergehend deaktiviert war oder nachdem Sie Dateien so verschoben oder kopiert haben (z.B. mit `cp -a`), dass keine automatische Einstellung des SELinux-Kontexts möglich war. Um den SELinux-Kontext verzeichnisunabhängig festzulegen, verwenden Sie das Kommando [chcon](#).

- **-0**  
erwartet, dass die Dateinamen durch 0-Bytes und nicht durch Tabulator- oder Leerzeichen getrennt sind. Das hilft in Kombination mit `find -print0` bei der Verarbeitung von Dateien, deren Namen Leerzeichen enthalten.
- **-e verzeichnis**  
überspringt dieses Verzeichnis. Die Option kann wiederholt werden, um mehrere Verzeichnisse von den Kontext-Änderungen auszunehmen.
- **-r bzw. -R**  
berücksichtigt auch alle Unterverzeichnisse (*recursive*).
- **-v**  
zeigt die durchgeführten Änderungen an (*verbose*).

## **Beispiel**

Das folgende Kommando stellt den korrekten SELinux-Kontext für alle Dateien im DocumentRoot-Verzeichnis des Webservers Apache ein:

```
root# restorecon -R -v /var/www/html/*
```

### rfkill kommando

rfkill aus dem gleichnamigen Paket ermöglicht es, Bluetooth-, WLAN- und Mobilfunkadapter ein- und auszuschalten, beispielsweise um bei einem Notebook Strom für gerade nicht benötigte Funktionen zu sparen.

- **block index/typ**  
deaktiviert das durch die Index-Nummer spezifizierte Gerät bzw. alle Geräte eines bestimmten Typs. Zulässige Typen sind `all`, `wifi = wlan`, `bluetooth`, `uwb = ultrawideband`, `wimax`, `wwan`, `gps` und `fm`.
- **event**  
wartet auf rfkill-Ereignisse und gibt diese aus.
- **list**  
listet alle Funk-Adapter und deren Zustand auf. `soft blocked` bedeutet, dass das Gerät durch rfkill ein- und ausgeschaltet werden kann. `hard blocked` bedeutet, dass rfkill das Gerät nicht steuern kann, z.B. weil es mit einem mechanischen Schalter verbunden ist.
- **unblock index/typ**  
aktiviert das bzw. die angegebenen Geräte, die wie bei `block` angegeben werden.

### Beispiel

Auf dem Testrechner stehen zwei Bluetooth- und ein WLAN-Adapter zur Verfügung. Einer der Bluetooth-Adapter wird aktiviert:

```
root# rfkill list
0: tpacpi_bluetooth_sw: Bluetooth
    Soft blocked: yes
    Hard blocked: yes
1: hci0: Bluetooth
    Soft blocked: yes
    Hard blocked: no
2: phy0: Wireless LAN
    Soft blocked: yes
```

```
      Hard blocked: no  
root# rfkill unblock 1
```

## rkhunter [optionen]

Das Shell-Script rkhunter (*Rootkit Hunter*) versucht, auf dem Rechner installierte Rootkits zu erkennen. Im Vergleich zu [chkrootkit](#) führt rkhunter noch mehr Tests durch. Wie bei [chkrootkit](#) ist das Programm aber nicht unfehlerbar: Sie müssen sowohl mit *false positives* rechnen als auch damit, dass das Programm ein modernes Rootkit gar nicht erkennt.

Bei einigen Distributionen wird mit der Installation des Pakets rkhunter ein Cron-Script eingerichtet, das rkhunter einmal täglich aufruft und eine E-Mail mit den Ergebnissen versendet. Unter Debian und Ubuntu ist diese Funktion ebenfalls vorgesehen, in der Konfigurationsdatei */etc/default/rkhunter* aber deaktiviert.

- -c bzw. --check  
führt eine Rootkit-Suche durch und zeigt die Ergebnisse an.
- --update  
aktualisiert die Musterdateien zur Erkennung von Rootkits.

## Beispiel

Die stark gekürzte Ausgabe des Programms hat die folgende Form:

```
root# rkhunter --check  
[ Rootkit Hunter version 1.4.2 ]  
Checking system commands  
...  
Performing file properties checks  
  Checking for prerequisites [ OK ]  
  /usr/sbin/adduser [ OK ]  
  /usr/sbin/chroot [ OK ]  
  ...  
  Performing check of known rootkit files and directories  
    55808 Trojan - Variant A [ Not found ]  
    ADM Worm [ Not found ]  
    AjaKit Rootkit [ Not found ]  
  ...  
  Performing system configuration file checks  
    Checking for an SSH configuration file [ Found ]  
    Checking if SSH root access is allowed [ Warning ]  
    Checking if SSH protocol v1 is allowed [ Not allowed ]  
  
System checks summary:  
File properties checks: Files checked: 151, Suspect files: 1
```

```
Rootkit checks:          Rootkits checked: 380, Possible rootkits: 0
...
All results have been written to the log file:
/var/log/rkhunter.log
```

### rm [optionen] dateien

rm löscht die angegebenen Dateien. Verzeichnisse werden – sofern nicht die Option -r verwendet wird – nicht gelöscht. Zum Löschen einzelner Verzeichnisse ist das Kommando [rmdir](#) vorgesehen. Wenn Dateien mit Sonderzeichen gelöscht werden sollen, müssen die Dateinamen in einfache Apostrophe eingeschlossen werden. Die wichtigsten Optionen vom rm sind:

- **-f**  
löscht ohne Rückfragen (auch Verzeichnisse). Vorsicht!
- **-i** bzw. **--interactive** oder **-v** bzw. **--verbose**  
zeigt vor dem Löschen jeder einzelnen Datei eine Rückfrage an.
- **-r** bzw. **-R** bzw. **--recursive**  
löscht auch Dateien in allen Unterverzeichnissen (Vorsicht!).  
Wenn dabei der gesamte Inhalt des Unterverzeichnisses gelöscht wird, wird auch das Unterverzeichnis selbst eliminiert.

### Beispiele

Das folgende Kommando löscht alle Backup-Dateien (Dateien, die mit dem ~-Zeichen enden) im aktuellen Verzeichnis:

```
user$ rm *\char126
```

rm löscht die Datei oder das Verzeichnis *backup*. Wenn *backup* ein Verzeichnis ist, werden auch alle darin enthaltenen Unterverzeichnisse und Dateien gelöscht!

```
user$ rm -r backup
```

rm löscht alle Dateien, die mit dem Doppelkreuz # beginnen. Die Apostrophe sind erforderlich, damit die Shell # nicht als Kommentar interpretiert.

```
user$ rm '#'*
```

### rmdir [optionen] verzeichnis

rmdir löscht das angegebene Verzeichnis. rmdir kann nur ausgeführt werden, wenn das Verzeichnis leer ist. Eventuell vorhandene

Dateien müssen vorher mit [rm](#) gelöscht werden. Die wichtigste Option lautet:

- **-p bzw. --parents**

löscht auch Unterverzeichnisse im angegebenen Verzeichnis, sofern die Verzeichnisse – mit Ausnahme von Unterverzeichnissen – leer sind.

```
rmmmod [optionen] modulname
```

**rmmmod** entfernt das angegebene Modul aus dem Kernel. Das funktioniert nur, wenn das Modul nicht mehr benötigt wird.

```
route [optionen]
```

```
route add/del [-net/-host] zieladresse [optionen]
```

**route** zeigt die Routing-Tabelle für das Netzwerkprotokoll IP an, fügt einen neuen Eintrag hinzu (**add**) bzw. entfernt einen Eintrag (**del**). Die Zieladresse wird wahlweise in numerischer Form oder als Netzwerkname angegeben. Als Zieladresse ist auch das Schlüsselwort **default** erlaubt, um eine Standardregel zu definieren (z.B. für das Gateway). Das Paket **route** gilt als veraltet. Verwenden Sie stattdessen nach Möglichkeit das Kommando [ip route](#)!

- **-host**

adressiert einen einzelnen Rechner.

- **-n**

zeigt nur numerische IP-Adressen an und kontaktiert keinen Nameserver zur Auflösung der zugeordneten Namen.

- **-net**

adressiert ein Netzwerk.

- **dev schnittstelle**

gibt die gewünschte Netzwerkschnittstelle an (z.B. **eth0**). Wenn **dev** die letzte Option ist, kann auf das Schlüsselwort verzichtet werden (also **route add -net 192.34.34.34 netmask 255.255.255.0 eth1**).

- **gw routingadresse**

leitet die Pakete an den angegebenen Router weiter.

- netmask *maske*

gibt die gewünschte Netzwerkmaske an (z.B. 255.255.255.0).

## **Beispiel**

Das erste route-Kommando definiert die IP-Adresse 192.168.0.1 als Standard-Gateway. Das zweite Kommando zeigt die neue Routing-Tabelle an.

```
root# route add default gw 192.168.0.1
root# route -n
Kernel IP routing table
Destination Gateway     Genmask      Flags Metric Ref Use Iface
192.168.0.0 0.0.0.0    255.255.255.0 U         0      0      0 eth0
0.0.0.0    192.168.0.1  0.0.0.0    UG        0      0      0 eth0
```

### **rpcinfo [optionen]**

rpcinfo führt einen Remote Procedure Call (RPC) durch und liefert das vom RPC-Server zurückgegebene Ergebnis. rpcinfo wird unter anderem dazu eingesetzt, um die NFS-Server-Konfiguration zu überprüfen.

- -p [*hostname*]

ermittelt eine Liste aller aktiven NFS- und RPC-Dienste auf dem lokalen Rechner bzw. auf einem externen Rechner.

### **rpm optionen [dateiname/paketname]**

rpm installiert, aktualisiert oder entfernt RPM-Pakete bzw. zeigt Informationen zu diesen Paketen an. rpm steht auf allen Distributionen zur Verfügung, die auf RPM-Paketen basieren, z.B. Fedora, Red Hat und SUSE.

- -e bzw. --erase

entfernt ein vorhandenes Paket. Wenn Sie zusätzlich die Option -nodeps verwenden, wird das Paket selbst dann entfernt, wenn andere Pakete davon abhängig sind.

- -i bzw. --install

installiert die angegebene(n) Paketdatei(en). Der Installationsort kann durch die zusätzliche Option --root verzeichnis verändert werden. Wenn es sich beim angegebenen Paket um ein Quellpaket handelt, werden der Programmcode und die

Konfigurationsdateien in `/usr/src` installiert. Der Installationsort kann in diesem Fall durch eine Veränderung von `/etc/rpmrc` eingestellt werden.

Mit der Zusatzoption `--test` führt `rpm` keine Veränderungen durch, sondern gibt nur aus, was bei einer tatsächlichen Installation passieren würde.

`rpm` versucht, automatisch die Integrität signierter Pakete zu überprüfen. Dazu muss aber der öffentliche Schlüssel der Signatur zur Verfügung stehen (siehe die Option `--import`). Wenn `rpm` auf die Kontrolle verzichten soll, übergeben Sie die Option `--nosignature`.

Mit der Zusatzoption `--nodeps` können Sie ein Paket selbst dann installieren, wenn `rpm` glaubt, dass diverse Abhängigkeiten nicht erfüllt sind. (Falls zwei Pakete gegenseitig voneinander abhängig sind, besteht keine Notwendigkeit, `--nodeps` einzusetzen.)

Installieren Sie einfach beide Pakete gleichzeitig!)

Die Zusatzoption `--ignorearch` ermöglicht eine Installation auch dann, wenn `rpm` erkennt, dass die CPU-Architektur nicht stimmt. Das ist bisweilen erforderlich, um 32-Bit-Pakete auf einer 64-Bit-Distribution zu installieren.

Die Zusatzoption `--force` erzwingt eine Installation in jedem Fall, selbst dann, wenn `rpm` glaubt, dass das Paket bereits installiert ist. Mit der Zusatzoption `--noscripts` verzichtet `rpm` auf die automatische Ausführung von Installationsprogrammen. Unter Umständen funktioniert das gerade installierte Programm dann aber nicht ordnungsgemäß.

- `--import`

installiert den angegebenen GPG-Schlüssel. `rpm` kann anhand dieses Schlüssels die Integrität signierter Pakete überprüfen. Die erforderlichen Schlüsseldateien finden Sie auf allen Websites, die signierte RPM-Pakete zur Verfügung stellen.

- **-q bzw. --query**  
liefert Informationen über installierte Pakete bzw. über den Inhalt einer Paketdatei. In der einfachsten Form gibt `rpm -q paketname` den exakten Paketnamen inklusive Versionsnummer des angegebenen Pakets aus.  
Mit `rpm -qf datei` stellen Sie fest, zu welchem Paket eine bestimmte Datei Ihres Systems gehört. Das ist besonders bei Konfigurationsdateien sehr nützlich. Beispielsweise liefert `rpm -qf /sbin/hwclock` den Paketnamen `util-linux-n`.  
`rpm -qp paket.rpm` liefert Informationen über ein noch *nicht* installiertes Paket. Als Parameter muss in diesem Fall der Dateiname einer `*.rpm`-Datei verwendet werden.  
`rpm -q --whatprovides attribut` bzw. `rpm -q --whatrequires attribut` sucht nach allen installierten Paketen, die das angegebene Abhängigkeitsattribut zur Verfügung stellen bzw. erfordern. Diese zwei Optionen sind nicht geeignet, um noch nicht installierte Pakete zu durchsuchen!  
`rpm -q --provides paket` bzw. `rpm -q --requires` erstellt eine Liste aller Attribute, die das angegebene Paket zur Verfügung stellt bzw. benötigt. Die Optionen können auch für noch nicht installierte Paketdateien (zusätzliche Option `-p`) eingesetzt werden.  
Bei allen Abfragevarianten, die als Ergebnis ein Paket liefern, steuern Sie durch Zusatzaoptionen die Informationsfülle zum gefundenen Paket:
  - i Kurzbeschreibung des Pakets, komprimierte Größe
  - l Liste aller Dateien des Pakets
  - lv wie oben, aber mit Dateigröße, Zugriffsrechten etc.
  - c Liste aller Konfigurationsdateien des Pakets
  - d Liste aller Dateien mit Online-Dokumentation zum Paket
  - scripts Liste aller Installationsprogramme

Wenn Sie einfach nur eine unsortierte Liste aller installierten Pakete benötigen, führen Sie `rpm -qa` ohne weitere Parameter aus. `rpm -qa --last` sortiert alle Pakete nach dem Installationsdatum (das zuletzt installierte Paket zuerst).

- **-U bzw. --upgrade**  
aktualisiert ein bereits vorhandenes Paket. Von den veränderten Konfigurationsdateien des bisherigen Pakets werden automatisch Backup-Dateien erzeugt; alle anderen Dateien des bisherigen Pakets werden durch die neue Version ersetzt bzw. gelöscht. Mit der zusätzlichen Option `--oldpackage` können Sie ein neueres Paket durch ein älteres ersetzen. Auch beim Upgrade erreichen Sie durch `--nodeps`, dass `rpm` Paketabhängigkeiten ignoriert.
- **-V bzw. --verify**  
überprüft, ob sich irgendwelche Dateien eines Pakets seit der Installation verändert haben. Das Kommando liefert eine Liste aller veränderten Dateien. Dokumentationsdateien werden nicht überprüft.

## Beispiele

Das folgende Kommando installiert das angegebene Paket:

```
root# rpm -i abc-2.0.7-1.x86_64.rpm
```

Das zweite `rpm`-Beispiel liefert eine Liste aller installierten Pakete, wobei die Pakete nach dem Installationsdatum sortiert werden (das zuletzt installierte Paket steht zuoberst in der Liste):

```
root# rpm -qa --last
```

`rpm -q --whatprovides` verrät, welches bereits installierte Paket das Attribut `mysqli.so` zur Verfügung stellt:

```
root# rpm -q --whatprovides mysqli.so
php-mysql
```

Wenn Sie nicht wissen, zu welchem Paket die Datei `/etc/magic` gehört, ermitteln Sie diese Informationen mit `rpm -qf`.

Anschließend liefert `rpm -qi` eine Beschreibung des Pakets:

```
root# rpm -qf /etc/magic
file-n.n
```

```
root# rpm -qi file
```

...  
The file command is used to identify a particular file according to the type of data contained by the file.

```
rpm2archive name.rpm  
rpm2cpio name.rpm
```

**rpm2archive** erzeugt aus einem als Datei vorliegenden RPM-Paket ein komprimiertes TAR-Archiv (`name.rpm.tgz`) mit allen Dateien des Pakets. **rpm2archive** ist praktisch, um einzelne Dateien aus einem Paket zu extrahieren, ohne das Paket zu installieren.

**rpm2archive** steht erst ab der RPM-Version 4.12 zur Verfügung. Bei älteren Distributionen müssen Sie sich mit dem äquivalenten Kommando **rpm2cpio** behelfen. Es erzeugt eine [cpio](#)-Datei, deren Inhalt Sie mit [cpio](#) extrahieren können.

```
rsync [optionen] quelle  
rsync [optionen] quelle ziel
```

**rsync** kopiert Dateien vom Quell- in das Zielverzeichnis bzw. synchronisiert die beiden Verzeichnisse. Bei der ersten Syntaxvariante (kein Zielverzeichnis) werden die Dateien in das lokale Verzeichnis kopiert. **rsync** überträgt die Dateien wahlweise lokal oder verschlüsselt über das Netzwerk. Die Netzwerkvariante setzt voraus, dass auf dem zweiten Rechner ein SSH- oder RSYNC-Server läuft. Bevor das Kopierkommando startet, muss das Passwort angegeben werden. Für die Angabe der Quell- und Zielverzeichnisse bestehen die folgenden Syntaxvarianten:

datei1 datei2	lokale Dateien
verzeichnis	lokales Verzeichnis
host:verz	Verzeichnis auf dem Rechner <a href="#">host</a>
user@host:verz	wie oben, aber SSH-Login unter dem Namen <a href="#">user</a>
rsync://user@host/verz	Kommunikation mit RSYNC-Server
rsync://user@host:port/verz	RSYNC-Server am angegebenen Port

Die Entwickler des SSH-Projekts empfehlen, **rsync** anstelle des Kommandos [scp](#) einzusetzen, da die Implementierung von [scp](#)

mangelhaft und eine Behebung der [scp](#)-Probleme nicht absehbar sei. Sofern auch auf dem externen Host rsync installiert ist und keine besonderen [scp](#)-Optionen im Spiel sind, können Sie [scp](#) eins zu eins durch rsync ersetzen. Entsprechende Beispiele finden Sie bei [scp](#).

- **-4 bzw. -6**  
erzwingt das IPv4- bzw. das IPv6-Protokoll, falls beide Varianten zur Auswahl stehen.
- **-a bzw. --archive**  
kopiert den Verzeichnisinhalt rekursiv und erhält alle Dateiinformationen. Die Option ist eine Kurzfassung für `-rlptgoD`.
- **--delete**  
löscht im Zielverzeichnis Dateien bzw. Unterverzeichnisse, die im Quellverzeichnis nicht mehr existieren (Vorsicht!).
- **-D bzw. --devices**  
kopiert auch Zeichen- und Block-Device-Informationen.
- **-e kommando bzw. --rsh=kommando**  
gibt an, welches externe Shell-Programm eingesetzt wird, um rsync auf dem Zielrechner auszuführen. In der Regel wird dies [ssh](#) sein. Wenn Sie an das Kommando zusätzliche Optionen übergeben möchten, setzen Sie es in Hochkommas, z.B. `-e "ssh -p 1234"`.
- **--exclude=muster**  
gibt ein Muster für Dateien an, die nicht kopiert werden sollen. `--exclude="*.o"` schließt beispielsweise \*.o-Dateien aus. (Zur Formulierung komplexerer Muster siehe `man rsync`.)
- **-g bzw. --group**  
gibt den Dateien im Zielverzeichnis dieselbe Gruppenzugehörigkeit wie bei Quelldateien.

- **-l bzw. --links**  
dupliciert symbolische Links des Quellverzeichnisses auch im Zielverzeichnis.
- **-o bzw. --owner**  
gibt den Dateien im Zielverzeichnis denselben Besitzer, den auch die Dateien im Quellverzeichnis haben.
- **-p bzw. --perms**  
gibt den Dateien im Zielverzeichnis dieselben Zugriffsrechte wie bei Quelldateien.
- **-r bzw. --recursive**  
kopiert rekursiv auch alle Unterverzeichnisse.
- **-t bzw. --times**  
gibt den Dateien im Zielverzeichnis dieselbe Änderungszeit wie bei den Quelldateien. Auf diese Weise kann rsync bei einem neuerlichen Aufruf eindeutig erkennen, welche Dateien sich geändert haben.
- **-u bzw. --update**  
ignoriert Dateien, die im Zielverzeichnis bereits existieren und neuer sind als die Quelldatei.
- **-v bzw. --verbose**  
liefert ausführliche Informationen darüber, was gerade passiert.
- **-w bzw. --whole-file**  
kopiert bei Änderungen die gesamte Datei. Bei schnellen Netzwerkverbindungen kann das die Synchronisationsgeschwindigkeit ein wenig erhöhen. Standardmäßig überträgt rsync nur die Änderungen und minimiert so die Menge der übertragenen Daten. Die Option -w vergrößert die zu übertragende Datenmenge, reduziert aber den Aufwand zur Feststellung der Änderungen. -w gilt automatisch, wenn es sich bei den Quell- und Zielverzeichnissen um lokale Verzeichnisse handelt. In diesem Fall erreichen Sie mit --no-

`whole-file`, dass rsync dennoch den inkrementellen Änderungsmodus nutzt.

## Beispiele

Das folgende Kommando synchronisiert alle Dateien aus dem Verzeichnis `mydata` mit dem gleichnamigen Backup-Verzeichnis. Dateien, die in `mydata` gelöscht wurden, werden auch im Backup-Verzeichnis gelöscht.

```
user$ rsync -av --delete mydata/* /backup/mydata/
```

Das zweite Beispiel synchronisiert das lokale Verzeichnis `verz1` mit dem Verzeichnis `verz2` auf dem Rechner `mars.sol`. Für die Passworteingabe ist [ssh](#) verantwortlich. (Es muss also das Login-Passwort des Benutzers `username` auf dem Rechner `mars.sol` eingegeben werden.)

```
user@saturn.sol$ rsync -e ssh -az verz1/ mars.sol:verz2/  
user@mars.sol's password: *****
```

rsync kann auch dazu verwendet werden, eine abgebrochene [scp](#)-Übertragung einer großen Datei fortzusetzen. ([scp](#) ist dazu nicht in der Lage.)

```
user$ rsync --partial --progress --rsh=ssh local_file user@host:remote_file
```

# S

**scp** [optionen] quelle ziel

Das weit verbreitete Kommando scp kopiert eine Datei zwischen zwei Rechnern in einem Netzwerk. Die Übertragung erfolgt verschlüsselt. scp setzt voraus, dass auf dem zweiten Rechner ein SSH-Server läuft. Die Quell- und Zieldatei bzw. das Zielverzeichnis werden wie folgt angegeben:

datei	lokale Datei
verzeichnis	lokales Zielverzeichnis
host:	Zielrechner (kopiert Dateien in das Heimatverzeichnis)
host:datei	Datei auf dem angegebenen Rechner
host:verzeichnis	Zielverzeichnis auf dem angegebenen Rechner
user@host:datei	Datei des Nutzers user auf dem angegebenen Rechner
user@host:verzeichnis	Zielverzeichnis des Nutzers user@host

Leider weist scp diverse Implementierungsschwächen auf, die sich nicht ohne Weiteres beheben lassen. Die Entwickler des SSH-Projekts empfehlen daher allen Anwendern, von scp auf das wesentlich flexiblere Kommando [rsync](#) umzusteigen. Sofern auch auf dem anderen Host [rsync](#) installiert ist und keine besonderen scp-Optionen im Spiel sind, können Sie scp eins zu eins durch [rsync](#) ersetzen. In den folgenden Beispielen habe ich daher immer auch das äquivalente [rsync](#)-Kommando angegeben.

## Beispiele

Das folgende Kommando kopiert die lokale Datei *abc.txt* in das Verzeichnis *~/efg* des Rechners *mars*:

```
user@uranus$ scp abc.txt mars:efg/
user@uranus$ rsync abc.txt mars:efg/
user@mars's password: *****
```

Mit -r wird ein ganzes Verzeichnis übertragen. Die Option --links stellt sicher, dass [rsync](#) auch symbolische Links berücksichtigt, was normalerweise nicht der Fall ist (bei scp dagegen schon).

```
user$ scp -r local-dir/ mars:other-dir
user$ rsync -r --links local-dir/ mars:other-dir
user@mars's password: *****
```

Bei scp können Sie mit der Option -P eine Port-Nummer übergeben, wenn der SSH-Server auf der Gegenseite nicht den Standardport 22 nutzt. Diese Option steht bei [rsync](#) nicht zur Verfügung. Vielmehr müssen Sie mit -e angeben, wie das hinter den Kulissen aufgerufene Kommando [ssh](#) zusammengesetzt werden soll:

```
user$ scp -P 2222 datei mars:
user$ rsync -e 'ssh -p 2222' datei mars:
user@mars's password: *****
```

### **screen [optionen] [tty-device]**

screen aus dem gleichnamigen Paket ist ein Screen-Manager, der mehrere Prozesse (Sessions) mit einem Terminal verbindet. Sie können damit z.B. innerhalb einer SSH-Verbindung mehrere Kommandos parallel ausführen und zwischen den Sessions hin- und herschalten. screen kann auch dazu verwendet werden, serielle Schnittstellen (TTY-Devices) in ein Terminal gleichsam »umzuleiten«.

screen wird durch zahlreiche Tastenkürzel gesteuert, die alle mit **[Strg]+[A]** beginnen. **[Strg]+[A]**, **[?]** zeigt eine Zusammenfassung der wichtigsten Kürzel an, **[Strg]+[A]**, **[X]** beendet screen. Weitere Kürzel fasst [Tabelle 20](#) im Abschnitt »Tastenkürzel« zusammen.

Länger laufende Prozesse können Sie mit **[Strg]+[A]**, **[D]** (*detach*) von screen lösen. Der Prozess läuft dann im Hintergrund weiter. Wenn Sie zu einem späteren Zeitpunkt screen mit der Option -r neuerlich starten, stellt screen die Verbindung zur Session wieder her.

- **-ls**

listet alle von screen getrennten Sessions auf.

- **-L**  
protokolliert alle in screen ausgeführten Kommandos sowie deren Ausgaben in den Dateien *screenlog.0* (erste Session), *screenlog.1* (zweite Session) etc. Das Logging kann auch mit **[Strg] + [A]**, **[↑] + [H]** ein- und ausgeschaltet werden. Wenn Sie die Logging-Dateien mit [less](#) lesen möchten, müssen Sie die Option **-r** verwenden, damit die mitprotokollierten Steuerzeichen korrekt dargestellt werden.
- **-r [session-id]**  
stellt die Verbindung zu einer getrennten Session wieder her. Wenn es nur eine Session gibt, können Sie auf die Nennung der Session-ID verzichten.

## Beispiel

Die Funktionsweise von screen lässt sich schwer in Textform darstellen. Wenn Sie noch nie mit screen gearbeitet haben, sollten Sie sich ein kurzes Video anzusehen. Suchen Sie z.B. nach *screen command video* oder nach *screen command video tutorial*!

**sdptool** [optionen] kommando

Mit dem **sdptool** aus dem **bluez**-Paket können Sie Anfragen gemäß dem *Service Discovery Protocol* (SDP) an Bluetooth-Geräte senden. Am häufigsten kommt dabei das Kommando **browse** zum Einsatz.

- **browse bt-mac**  
liefert Informationen über das externe Bluetooth-Gerät, das durch seine MAC-Adresse angegeben wird. Die MAC-Adresse ermitteln Sie mit **hcitool scan**.
- **browse local**  
liefert detaillierte Informationen über den lokalen Bluetooth-Adapter.

## Beispiel

**sdptool** stellt fest, dass es sich beim angegebenen Bluetooth-Gerät um ein Eingabegerät (Human Interface Device) handelt, das die

Protokolle L2CAP und HIDP unterstützt, also das *Logical Link Control and Adaptation Protocol* und das *Human Interface Device Protocol*.

```
user$ sdptool browse 60:FB:42:FC:BB:8C
Service Name: Wireless Mouse
Service Class ID List:
  "Human Interface Device" (0x1124)
Protocol Descriptor List:
  "L2CAP" (0x0100)
    PSM: 17
  "HIDP" (0x0011)
...
```

### **sealert [optionen]**

sealert hilft bei der Diagnose von SELinux-Regelverstößen. Das Kommando setzt voraus, dass der Hintergrunddienst `setroubleshootd` läuft. Dieser Dienst protokolliert in das systemd-Journal:

```
root# journalctl | grep setrouble
...
setroubleshoot[2685]: SELinux is preventing httpd from read access
on the file test.html. For complete SELinux messages run
sealert -l ca111078-1aba-4481-bbfc-fbd59d8ca6b9
...
```

Wenn Sie nun, wie in der Protokolldatei angegeben, sealert ausführen, erhalten Sie einen kunterbunt aus deutschen und englischen Begriffen zusammengewürfelten Text. Besser lesbar wird der Text, wenn Sie die Lokalisierung auf Englisch umstellen:

```
root# LANG=; sealert -l ca111078-1aba-4481-bbfc-fbd59d8ca6b9
SELinux is preventing httpd from read access on the file test.html.
```

```
** Plugin catchall_boolean (89.3 confidence) suggests

If you want to allow httpd to read user content, then you must tell
SELinux about this by enabling the 'httpd_read_user_content' boolean.
You can read 'httpd_selinux' man page for more details. Do
```

```
setsebool -P httpd_read_user_content 1
```

```
** Plugin catchall (11.6 confidence) suggests
```

```
If you believe that httpd should be allowed read access on the test.html
file by default, then you should report this as a bug ...
```

Leider ist das Ergebnis selten eine echte Hilfe.

- **-a** *logdatei*  
analysiert die angegebene Loggingdatei und liefert Informationen zu allen darin protokollierten SELinux-Ereignissen.
- **-l** *id*  
gibt an, zu welchem SELinux-Ereignis sealert Informationen liefern soll. **-l '\*'** behandelt alle protokollierten SELinux-Regelverstöße.

**sed** [optionen] kommando [< quelle > ziel]

sed ist ein sogenannter Stream-Editor. Das Kommando wird normalerweise als Textfilter verwendet, um bestimmte Zeichen oder Zeichenkombinationen im Ausgangstext zu finden und zu bearbeiten (löschen, durch andere Zeichen ersetzen etc.). Die Steuerung des sed erfolgt durch Kommandos, die entweder auf alle Zeilen des Textes angewendet werden oder nur auf solche Zeilen, die bestimmten Bedingungen entsprechen.

- **-i** *datei*  
verändert eine Datei direkt (*inline*). Normalerweise liest sed aus der Standardeingabe und schreibt an die Standardausgabe. Die direkte Veränderung einer Datei ist dann unmöglich.
- **-i.**backup-kennung *datei*  
wie oben, die originale Datei wird aber als Backup mit der angegebenen Kennung gespeichert.

Der Umgang mit sed ist leider nicht ganz einfach. Selbst eine Beschreibung der elementaren Syntaxoptionen würde den Rahmen dieser Referenz bei Weitem sprengen. Deswegen beschränke ich mich in diesem Buch auf einige einfache Beispiele. Eine ausgezeichnete englischsprachige Einführung in den Umgang mit sed finden Sie hier:

<https://www.grymoire.com/Unix/Sed.html>

## Beispiel

Im ersten Beispiel löscht sed die erste bis dritte Zeile der Datei *test* und leitet den Rest der Datei an die Standardausgabe. Dabei steht **d** für das sed-Kommando *delete*. **1,3** gibt den Adressbereich an, in dem dieses Kommando wirkt.

```
user$ sed 1,3d < test
```

Im zweiten Beispiel ersetzt sed alle »a« durch »A«. Dabei steht **s** für das Kommando *regular find and replace*. Die in / eingeschlossenen Texte sind das Suchmuster und der einzusetzende Text. Da vor dem Kommando keine Adressangabe steht, wird es auf alle Zeilen angewandt.

```
user$ sed s/a/A/ < test
```

Das dritte Beispiel löscht aus einer Markdown-Datei alle LaTeX-Anweisungen, die mit \ beginnen. Ein Backup der ursprünglichen Datei wird unter *test.md.bak* gespeichert.

```
user$ sed -i.bak '/\\newpage.*/d' test.md
```

Wie Sie sed zum Kopieren bzw. Umbenennen von Dateien einsetzen können, zeigt ein weiteres Beispiel bei der Beschreibung des [cp](#)-Kommandos.

```
seq [optionen] ende  
seq [optionen] start ende  
seq [optionen] start delta ende
```

seq liefert eine Sequenz von Ziffern. Die Funktionsweise geht aus den folgenden Beispielen hervor. Beachten Sie, dass bei Fließkommazahlen der Endwert aufgrund von Rundungsfehlern unter Umständen gerade nicht erreicht wird, wie dies im letzten Beispiel demonstriert ist.

- **-s zeichenkette**

gibt an, durch welches Zeichen die Ergebnisse getrennt werden. Standardmäßig geschieht das durch \n, also durch einen Zeilenumbruch.

## Beispiele

Die folgenden Beispiele zeigen die Anwendung von seq. Dabei kommt aus Platzgründen als Trennzeichen ein Leerzeichen zum

Einsatz. Normalerweise (also ohne die Option `-s`) wird jede Zahl in einer eigenen Zeile ausgegeben.

```
user$ seq -s ' ' 5
1 2 3 4 5
user$ seq -s ' ' 4 9
4 5 6 7 8 9
user$ seq -s ' ' 10 5 30
10 15 20 25 30
user$ seq -s ' ' 1 0.1 2
1 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9
```

Wenn Sie [for](#)-Schleifen in bash-Scripts formulieren, sollten Sie nach Möglichkeit die Schreibweisen `{start..ende}` oder `{start..ende..delta}` nutzen, also beispielsweise:

```
user$ {bfs for i in {100..150..10}; do echo $i; done}
100 110 120 130 140 150
```

**service** dienst kommando

service erlaubt es unter vielen Distributionen, Dienste zu starten bzw. zu stoppen. Der Clou an service besteht darin, dass das Kommando unabhängig vom Init-System funktioniert und sowohl zum alten Init-V-System als auch zum mittlerweile weit verbreiteten systemd kompatibel ist.

## Beispiele

Das folgende Kommando startet unter Fedora den Webserver Apache neu:

```
root# service httpd restart
```

Unter den Distributionen Debian und Ubuntu ist für Apache die systemd-Service-Datei apache2 zuständig:

```
root# service apache2 restart
```

**sestatus** [optionen]

sestatus zeigt den Zustand des SELinux-Systems an. Wenn das Kommando ohne weitere Optionen verwendet wird, liefert es eine Zusammenfassung des aktuellen Zustands.

- `-b`

zeigt zusätzlich den Zustand aller booleschen Parameter der SELinux-Regeln an. Diese Parameter können mit [setsebool](#) verändert werden.

- -v

zeigt zusätzlich Kontextinformationen zu den in */etc/sestatus.conf* aufgelisteten Dateien und Prozessen an.

## Beispiel

Auf dem Testrechner ist SELinux aktiv, die Einhaltung der Regeln wird sichergestellt (`current mode = enforcing`). Es kommt das Regelset `targeted` zum Einsatz.

```
root# sestatus -b
SELinux status:          enabled
SELinuxfs mount:         /sys/fs/selinux
SELinux root directory: /etc/selinux
Loaded policy name:     targeted
Current mode:           enforcing
Mode from config file: enforcing
Policy MLS status:      enabled
Policy deny_unknown status: allowed
Memory protection checking: actual (secure)
Max kernel policy version: 32

Policy booleans:
abrt_anon_write        off
abrt_handle_event       off
abrt_upload_watch_anon_write on
antivirus_can_scan_system off
...
```

## set

Das bash-Kommando `set` zeigt alle der Shell bekannten Variablen an (inklusive der Umgebungsvariablen, die auch mit [printenv](#) angezeigt werden können). `set -x` bewirkt, dass die bash vor der Ausführung jedes Kommandos anzeigt, wie die interne Kommandozeile nach der Berücksichtigung von [alias](#)-Abkürzungen und der Expansion der Dateinamen aussieht.

## setcap [optionen] capability dateiname

`setcap` aus dem Paket `libcap-ng-utils` legt bei ausführbaren Dateien fest, welche Operationen für das Programm zulässig sind (Capabilities). Wenn statt `capability` die Option `-r` übergeben wird, werden alle früher festgelegten Capabilities gelöscht. Capabilities setzt ein Dateisystem mit Extended Attributes voraus. Bei ext-

Dateisystemen muss also die [mount](#)-Option `user_xattr` verwendet werden.

## Beispiel

Das Netzwerkkommando [ping](#) war früher bei vielen Distributionen mit dem setuid-Bit ausgestattet, sodass es von gewöhnlichen Benutzern verwendet werden konnte. Sobald Sie dieses Bit löschen, kann nur noch root mit [ping](#) arbeiten:

```
user$ ls -l /usr/bin/ping
-rwsr-xr-x 1 root root 72776 Jan 31 2020 /usr/bin/ping
user$ sudo chmod u-s /bin/ping
user$ ping yahoo.de
ping: icmp open socket: Die Operation ist nicht erlaubt
```

Anstatt nun das unsichere setuid-Bit wieder zu setzen, reicht es auch, dem Kommando [ping](#) mit setcap den Zugriff auf Netzwerkfunktionen des Kernels zu geben. Mit [getcap](#) können Sie nachsehen, welche Capabilities ein Kommando hat:

```
user$ sudo setcap cap_net_raw=ep /bin/ping
user$ getcap /bin/ping
/bin/ping = cap_net_raw+ep
user$ ping yahoo.de
PING yahoo.de (212.82.102.24) 56(84) bytes of data.
...
rtt min/avg/max/mdev = 58.054/58.054/58.054/0.000 ms
```

Bei den meisten aktuellen Distributionen ist [ping](#) mittlerweile standardmäßig mit Capabilities ausgestattet, z.B. unter Debian, Fedora, openSUSE und Ubuntu.

**setenforce 0|1|Enforcing|Permissive**

setenforce schaltet SELinux zwischen den Modi Enforcing (1) und Permissive (0) um. Im Permissive-Modus werden Regelübertretungen zwar protokolliert, das betroffene Programm kann aber ungehindert weiterarbeiten. Im Enforcing-Modus hindert SELinux das Programm hingegen, durch SELinux-Regeln verbotene Vorgänge auszuführen.

Wenn Sie SELinux vollständig deaktivieren möchten, müssen Sie in `/etc/selinux/config` die Einstellung `SELINUX=disabled` vornehmen und den Rechner neu starten. Beachten Sie, dass eine spätere

Reaktivierung schwierig ist, weil Sie dann den SELinux-Kontext aller neuen Dateien richtigstellen müssen. Deswegen ist der Permissive-Modus in der Regel vorzuziehen.

**setfacl [optionen] [aclaktion] dateiname**

setfacl 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 acl verwendet werden. Das Kommando wird üblicherweise zur Durchführung einer der vier folgenden Aktionen eingesetzt:

- **-m aclregel**  
fügt den vorhandenen ACLs eine weitere Regel (eine *Access Control Entity*, kurz ACE) hinzu. (m steht hier für *modify*.)
- **-M aclregeldatei**  
wie -m, liest die Regel aber aus einer Datei. Regeldateien müssen wie die Ausgabe von [getfacl](#) aussehen.
- **-x aclregel**  
löscht die angegebene ACL-Regel.
- **-X aclregeldatei**  
wie -x, liest die Regel aber aus einer Datei.

Der vereinfachte Aufbau einer ACL-Regel (einer ACE) geht aus der folgenden Tabelle hervor. Die vollständige Syntax ist in `man setfacl` dokumentiert.

[u:]uid [:rechte]	ändert Rechte für einen Benutzer.
g:gid [:rechte]	ändert Rechte für eine Gruppe.
o[:][:rechte]	ändert die Rechte für alle anderen Benutzer.
m[:][:rechte]	stellt die ACL-Maske ( <i>Effective Right Mask</i> ) ein.

Die rechte bestehen aus bis zu drei Buchstaben: r für *read*, w für *write* und x für *execute*. Um einem Benutzer oder einer Gruppe alle Rechte zu entziehen, geben Sie - an.

Wenn Sie der gesamten Regel d: voranstellen, gilt sie für die Standard-ACLs. Statt der Buchstaben u, g, o und m können Sie auch

die Schlüsselwörter `user`, `group`, `other` und `mask` verwenden.

`setfacl` kann durch weitere Optionen gesteuert werden:

- `-b` bzw. `--remove-all`  
entfernt alle ACL-Regeln.
- `-d`  
wendet die übergebene Regel auf die Standard-ACL an.
- `-k` bzw. `--remove-default`  
entfernt die Standard-ACL-Regeln.
- `-n`  
verzichtet auf die sonst bei jeder ACL-Regeländerung  
automatische Neuberechnung der ACL-Maske.
- `--restore=datei`  
wendet die in der Datei angegebenen ACL-Regeln auf die Dateien  
im aktuellen Verzeichnis an. Eine ACL-Backup-Datei erstellen Sie  
mit [`getfacl -R`](#).
- `-R`  
wendet die angegebene Regel rekursiv auf alle Dateien und  
Unterverzeichnisse an.

## Beispiele

Bei einem Dateisystem mit ACLs gelten normalerweise die Standardzugriffsrechte, die oft auch als *minimale ACL* bezeichnet werden. [`getfacl`](#) zeigt diese Rechte in ACL-Form an:

```
user$ touch datei1
user$ getfacl datei1
# file: datei1
# owner: kofler
# group: kofler
user::rw-
group::r--
other::r--
user$ ls -l datei1
-rw-r--r-- 1 kofler kofler ... datei2
```

Mit `setfacl` definieren Sie nun zusätzliche Zugriffsregeln. Die folgenden Kommandos geben der Benutzerin `gabi` sowie allen

Mitgliedern der Gruppe docuteam Schreib- und Lesezugriff auf die Datei, verbieten aber der Benutzerin kathrin jeglichen Zugriff:

```
user$ setfacl -m gabi:rw datei1
user$ setfacl -m g:docuteam:rw datei1
user$ setfacl -m kathrin:- datei1
```

Die Rechteliste von [getfacl](#) ist nun schon etwas länger. Bei `ls -l` folgt den üblichen Zugriffsbuchstaben das Zeichen +, um darauf hinzuweisen, dass es ACL-Regeln gibt.

```
user$ getfacl datei1
# file: datei1
# owner: kofler
# group: kofler
user::rw-
user:gabi:rw-
user:kathrin:---
group::r--
group:docuteam:rw-
mask::rw-
other::r--

user$ ls -l datei1
-rw-rw-r--+ 1 kofler kofler ... datei1
```

### **setfattr [optionen] dateiname**

setfattr ändert die erweiterten Attribute der gewählten 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.

- `-n attributname` bzw. `--name=attributname`  
gibt den Namen des zu verändernden Attributs an. Dem eigentlichen Namen muss `user.` vorangestellt werden (also `-n user.meinattribut`).
- `-v wert` bzw. `--value=wert`  
gibt den Wert an, der im Attribut gespeichert werden soll.
- `-x attributname`  
löscht das angegebene Attribut.
- `--restore=datei`  
wendet die in der Datei angegebenen EA-Definitionen auf die

Dateien im aktuellen Verzeichnis an. Eine EA-Backup-Datei erstellen Sie mit `getfattr -R -d`.

## Beispiele

Die folgenden Beispiele zeigen, wie Sie mit `setfattr` Attribute speichern und diese mit `getfattr` auslesen. Die Anzahl der Attribute pro Datei ist in ext-Dateisystemen beschränkt.

```
user$ touch datei2
user$ setfattr -n user.language -v de datei2
user$ setfattr --name=user.charset --value=utf8 datei2
user$ getfattr -d datei2
# file: datei2
user.charset="utf8"
user.language="de"
```

`getfattr` liefert normalerweise nur Attribute, deren Name mit »user.« beginnt. Wenn Sie andere Attribute sehen möchten, müssen Sie deren Namen durch `-n` oder deren Muster durch `-m` angeben:

```
user$ getfattr -n security.selinux -d tst
# file: tst
security.selinux="user_u:object_r:user_home_t:s0^000"
setfont fontfile
```

`setfont` liest bei einigen Distributionen eine Schriftartdatei und aktiviert diesen Font im VGA-Textmodus. `setfont` kann nur in einer Textkonsole und nicht unter X verwendet werden. Brauchbare Schriften sind *default8x16* (für den 25-zeiligen Textmodus) und *default8x9* (für den 50-zeiligen Textmodus).

**setsebool [optionen] bool1=wert1 bool2=wert2 ...**

`setsebool` verändert boolesche Parameter zur Laufzeit. Mit booleschen Parametern lässt sich das Verhalten einiger SELinux-Regeln verändern. Welche boolesche Parameter es gibt, können Sie mit `sestatus -b` herausfinden.

- `-N`

speichert die Veränderung, aktiviert sie aber nicht. Die Änderung wird somit erst mit dem nächsten Neustart des Rechners wirksam.

- **-P**  
führt die Veränderung durch und speichert die Einstellung bleibend. Damit gilt die Änderung sofort und auch in Zukunft, also nach dem nächsten Neustart.

## Beispiel

Das folgende Kommando erlaubt dem Programm Squid, als transparenter Proxy zu agieren:

```
root# setsebool -P squid_use_tproxy 1
```

**setterm [option]**

`setterm` verändert diverse Einstellungen des Terminals. Wenn das Kommando ohne die Angabe einer Option ausgeführt wird, zeigt es eine Liste aller möglichen Optionen an. Nützliche Optionen zur Shell-Programmierung sind:

- **--background *farbe***  
stellt die Hintergrundfarbe ein. Dabei sind die Farben `black`, `red`, `green`, `yellow`, `blue`, `magenta`, `cyan` und `white` erlaubt. Damit die Einstellung auch nach Farbausgaben durch `ls` oder `grep` gültig bleibt, muss die Option mit `--store` kombiniert werden.
- **--blank *n***  
aktiviert nach *n* Minuten ohne Eingabe den Bildschirmschoner (nur für Textkonsolen).
- **--bold on|off**  
aktiviert bzw. deaktiviert die fette Schrift. In Textkonsolen erscheint der Text zwar nicht fett, aber immerhin in einer anderen Farbe als der sonstige Text.
- **--clear**  
löscht den Inhalt des Terminals.
- **--default**  
stellt Farben und Textattribute auf die Standardeinstellung zurück.
- **--foreground *farbe***  
stellt die Textfarbe ein.

- **--half-bright** on|off  
stellt hervorgehobene Schrift ein/aus.
- **--inversescreen** on|off  
invertiert die Darstellung in der Textkonsole (schwarzer Text auf weißem Hintergrund) bzw. stellt wieder den Normalzustand her.
- **--reverse** on|off  
stellt inverse Schrift ein/aus.
- **--store**  
merkt sich die Farbeinstellungen (Optionen **--foreground** und **--background**), sodass diese auch erhalten bleiben, wenn Kommandos wie [ls](#) oder [grep](#) vorübergehend andere Farben verwenden. **--store** ist nur in den Linux-Textkonsolen gültig, nicht aber innerhalb von Terminalfenstern. Auch dort werden die Einstellungen nicht in einer Datei gespeichert, sondern gehen mit dem Logout verloren.
- **--underline** on|off  
stellt unterstrichene Schrift ein/aus.

## Beispiel

Wenn die folgende Zeile in *.bashrc* eingefügt wird, verwenden Textkonsolen nach dem Login einen weißen Hintergrund und schwarze Schrift:

```
# in .bashrc einfügen
setterm -background white -foreground black -store
```

**sfdisk [optionen]**

sfdisk listet die Partitionen einer Festplatte auf bzw. partitioniert die Festplatten neu. Das Kommando eignet sich besonders zur automatisierten Partitionierung (z.B. in einem Script) bzw. dazu, eine vorhandene Partitionierung auf eine zweite Festplatte zu übernehmen, etwa beim Einrichten eines RAID-Verbunds.

sfdisk kommt nur mit MBR-Partitionstabellen zurecht und ist für Datenträger mit einer GUID-Partitionstabelle (GPT) ungeeignet!

- **-d [device]**  
listet alle Partitionen aller Festplatten bzw. der angegebenen Festplatte auf. Das Ausgabeformat eignet sich zur Weiterverarbeitung durch `sfdisk`.
- **--force**  
führt die Partitionierung auch dann durch, wenn `sfdisk` glaubt, dass die Partition bzw. Festplatte verwendet wird, bzw. wenn Partitionsgrenzen nicht mit Zylindergrenzen übereinstimmen.
- **-l [device]**  
listet alle Partitionen aller Festplatten bzw. der angegebenen Festplatte in einer gut lesbaren Form auf. Mit der zusätzlichen Option `-uS`, `-uB`, `-uC` oder `-uM` verwendet `sfdisk` Sektoren (512 Byte), Blöcke (1024 Byte), Zylinder oder MiB als Maßeinheit.
- **-s device**  
liefert die Anzahl der Blöcke (zu je 1024 Byte) der angegebenen Festplatte oder Partition.
- **-uS | -uB | -uC | -uM**  
verwendet Sektoren (512 Byte), Blöcke (1024 Byte), Zylinder oder MiB als Maßeinheit.
- **device < partitionierungstabelle>**  
partitioniert die Festplatte anhand der übergebenen Tabelle neu. Die Partitionierungstabelle muss dasselbe Format wie die Ausgabe von `sfdisk -d` haben.

## Beispiel

Das folgende Kommando partitioniert die zweite Festplatte (`/dev/sdb`) exakt wie die erste Festplatte (`/dev/sda`). Das setzt voraus, dass die zweite Festplatte zumindest genauso groß wie die erste ist. Alle Partitionen bzw. Daten auf der zweiten Festplatte gehen durch dieses Kommando verloren.

```
root# sfdisk -d /dev/sda | sfdisk /dev/sdb
sftp [optionen] sshserver
sftp [optionen] user@sshserver:filename
```

`sftp` ist eine Alternative zum unsicheren [ftp](#)-Kommando. In der ersten Syntaxvariante stellt `sftp` eine Verbindung zum angegebenen SSH-Server her. Nach dem Login können Sie interaktiv Dateien zwischen dem lokalen Rechner und dem SSH-Server übertragen. In der zweiten Syntaxvariante überträgt `sftp` die angegebene Datei nach der Passworteingabe direkt auf den lokalen Rechner.

- `-b batchdatei`

führt die in der Batch-Datei angegebenen Kommandos aus.

### **sgdisk [option] device**

`sgdisk` hilft bei der Bearbeitung von GUID-Partitionstabellen auf Festplatten und SSDs. Anders als [parted](#) wird das Kommando ausschließlich durch Optionen gesteuert und ist nicht für den interaktiven Einsatz gedacht. Vielmehr können Sie mit `sgdisk` Partitionstabellen Script-gesteuert erstellen oder manipulieren. Trotz des ähnlich klingenden Namens erwartet `sgdisk` ganz andere Optionen als [sfdisk](#). `sgdisk` arbeitet mit MiB und GiB in der IEC-Schreibweise.

- `-a n`

gibt an, zu welchem Vielfachen von Sektoren neue Partitionen ausgerichtet werden. Standardmäßig verwendet `sgdisk` den Wert 2048. Das entspricht einem MiB.

- `-b datei`

schreibt ein Backup der Partitionstabelle in eine binäre Datei.

- `-d nr`

löscht die angegebene Partition.

- `-G`

weist allen Partitionen neue, zufällige GUIDs zu. Das ist nach dem Klonen einer Festplatte erforderlich.

- `-l datei`

partitioniert den Datenträger entsprechend der in der Datei

gespeicherten Partitionstabelle. Die Datei muss vorher mit der Option `-b` erzeugt werden. Alle bisher auf dem Datenträger vorhandenen Partitionen werden gelöscht. Die Kombination der Kommandos `sgdisk -b` und `sgdisk -l` kann dazu verwendet werden, um das Partitionslayout einer Festplatte auf einen zweiten gleich großen Datenträger zu übertragen, z.B. beim Einrichten eines RAID-Systems.

- `-n nr:start:end`  
legt die neue Partition mit der Nummer `nr` an, die von der angegebenen Start- bis zur Endposition reicht. Standardmäßig wird `start` und `end` in Sektoren zu 512 Byte angegeben. Alternativ sind auch Positionsangaben in MiB und GiB möglich. `sgdisk -n 3:10G:35G` erzeugt eine neue Partition, die 25 GiB groß ist. Den Start- und Endpositionen können die Zeichen `+` und `-` vorangestellt werden. `+` ermöglicht Angaben, die relativ zum Default-Startsektor sind. `-` ermöglicht Angaben, die vom Ende der Festplatte gerechnet werden. `sgdisk -n 4:+1M:-20G` erzeugt eine Partition, die 1 MiB hinter der letzten Partition beginnt und 20 GiB am Ende des Datenträgers frei lässt.

## Beispiel

Die folgenden Kommandos erstellen erst ein Backup der aktuellen Partitionstabelle auf dem Datenträger `/dev/sdb` und richten dann die neue Partition 3 ein:

```
root# sgdisk -l partbackup /dev/sdb  
root# sgdisk -n 3:70G:430G /dev/sdb
```

**shasum** dateien

`sha1sum`, `sha224sum`, `sha256sum`, `sha384sum` und `sha512sum` berechnen Prüfsummen zu allen angegebenen Dateien. Die Zahl im Kommando gibt die Bitlänge der Prüfsumme an (160 Bits bei `sha1sum`). Die Prüfsummen werden verwendet, um nach der Übertragung einer Datei sicherzustellen, dass die Datei unverändert ist.

## Beispiel

Das folgende Kommando berechnet eine 256-Bit-Prüfsumme für eine ISO-Datei:

```
root# sha256sum image.iso  
6ad62fe91eefa4315f852eb1bc8732bc341d95b47f6b4d1f5e362d3629fc981b image.iso  
shift [n]
```

Das bash-Kommando `shift` schiebt die dem Shell-Script übergebene Parameterliste durch die vordefinierten Variablen \$1 bis \$9. Wenn `shift` ohne Parameter verwendet wird, werden die Parameter um eine Position verschoben, andernfalls um *n* Positionen.

`shift` wird nur zur Script-Programmierung eingesetzt. Das Kommando ist eine wertvolle Hilfe, wenn mehr als neun Parameter angesprochen werden sollen. Einmal mit `shift` aus den Variablen geschobene Parameter können nicht mehr angesprochen werden. Sie werden auch aus der Variablen `$*` entfernt.

```
showmount [option]
```

Ohne weitere Optionen liefert das Kommando `showmount`, wenn es auf einem NFS-Server ausgeführt wird, eine Liste aller Clients, die die freigegebenen Verzeichnisse per NFS 3 nutzen. Beachten Sie, dass NFS-4-Verzeichnisse nicht in das Ergebnis einfließen und nur bei der Variante `showmount -e` berücksichtigt werden.

- `-a`  
liefert zu jedem NFS-3-Client auch die IP-Adresse.
- `-e [hostname]`  
liefert eine Liste aller Verzeichnisse, die per NFS genutzt werden können, unabhängig davon, ob dieser Dienst momentan genutzt wird. Wenn Sie einen Hostnamen angeben, wird die Liste der Verzeichnisse ermittelt, die auf diesem Rechner verfügbar sind.  
`showmount -e` funktioniert auch für NFS 4.

## Beispiel

```
root# showmount -e  
/nfsexport      *.lan,10.0.0.0/24  
/nfsexport/fotos *.lan,10.0.0.0/24  
/nfsexport/iso  *
```

## **shutdown** [optionen] zeitpunkt [nachricht]

shutdown fährt den Rechner herunter. Als Zeitpunkt muss entweder eine Uhrzeit (hh:mm), die Anzahl der Minuten, gerechnet von der aktuellen Zeit (+m), oder das Schlüsselwort now (also sofort) angegeben werden. shutdown kann nur von root ausgeführt werden. Oft ist Linux so konfiguriert, dass Anwender ohne root-Rechte den Rechner mit **Alt** + **Strg** + **Entf** neu starten können.

shutdown informiert alle anderen Benutzer darüber, dass das System in Kürze abgeschaltet wird, und lässt keine neuen Logins mehr zu. Anschließend werden alle Prozesse gewarnt, dass sie in Kürze gestoppt werden. Einige Programme (etwa emacs, vi ...) nutzen diese Warnung und speichern alle offenen Dateien in Backup-Kopien.

- -c

versucht, einen bereits eingeleiteten shutdown-Vorgang abzubrechen. Das ist dann möglich, wenn die Zeit bis zum Beginn des shutdown-Prozesses noch nicht abgelaufen ist.

- -f

wie -r, aber schneller.

- -F

bewirkt eine Überprüfung der Dateisysteme beim nächsten Neustart. shutdown erzeugt zu diesem Zweck die Datei */forcefsck*. Ist sie vorhanden, wird bei den meisten Distributionen beim Neustart eine Dateisystemüberprüfung ausgelöst.

- -h

hält das System nach dem Herunterfahren an. Der Rechner reagiert dann nicht mehr auf Eingaben. Auf dem Bildschirm erscheint die Meldung *system halted*. Anschließend wird der Rechner durch ein ACPI-Signal ausgeschaltet.

- -n

führt den Shutdown unter Umgehung des Init-Systems besonders schnell aus.

- **-r**  
veranlasst nach dem Herunterfahren des Systems einen Neustart.
- **-t sekunden**  
bestimmt, wie lange zwischen der Warnnachricht und dem Kill-Signal für die Prozesse gewartet werden soll (normalerweise 20 Sekunden).

### **sleep zeit**

`sleep` versetzt das laufende Programm für die angegebene Zeit in den Ruhezustand. Das Programm konsumiert in dieser Zeit praktisch keine Rechenzeit.

Die Zeitangabe erfolgt normalerweise in Sekunden. Optional können die Buchstaben `m`, `h` oder `d` angehängt werden, um die Zeit in Minuten, Stunden oder Tagen anzugeben.

### **smartctl optionen device**

`smartctl` steuert die SMART-Funktionen der Festplatte. Damit können Sie mögliche Probleme der Festplatte frühzeitig erkennen.

- **-a bzw. --all**  
liefert alle verfügbaren SMART-Informationen.
- **-d ata/scsi/3ware, n**  
gibt den Festplattentyp an. Die Option ist nur erforderlich, wenn `smartctl` den Typ nicht selbst erkennen kann. Das gilt insbesondere für SATA-Festplatten, bei denen `smartctl` oft glaubt, es würde sich um SCSI-Festplatten handeln. Die Einstellung `3ware, n` ermöglicht die Nutzung der SMART-Funktionen für Festplatten, die an einen 3ware-RAID-Controller angeschlossen sind (zu den Details siehe `man smartctl`).
- **-H bzw. --health**  
gibt an, ob die Festplatte momentan in Ordnung ist und voraussichtlich die nächsten 24 Stunden noch funktionieren wird.

- **-i bzw. --info**  
liefert Basisinformationen zur Festplatte.
- **-l error/selftest bzw. --log,error/selftest**  
liefert Informationen zu den letzten fünf Fehlern bzw. zu den Ergebnissen der letzten 21 Selbsttests.
- **-s on/off bzw. --smart=on/off**  
aktiviert bzw. deaktiviert die SMART-Funktionen.
- **-t short/long bzw. --test=short/long**  
führt einen kurzen bzw. langen Selbsttest durch. Das Testergebnis sehen Sie sich später mit **-l selftest** an.

## Beispiel

`smartctl -H` bzw. `smartctl --health` gibt an, ob die Festplatte momentan in Ordnung ist. Sollte `smartctl` hier nicht *PASSED* als Ergebnis liefern, sollten Sie *sofort* damit beginnen, ein komplettes Backup durchzuführen!

```
root# smartctl -H /dev/sda
```

```
...
```

```
SMART overall-health self-assessment test result: PASSED
```

Für die Festplatte `/dev/sda` wurden bisher drei Selbsttests ausgeführt: einer unmittelbar nach dem Einbau der Platte (nach 40 Betriebsstunden), die anderen beiden nach ca. 2600 Stunden. Es sind keine Probleme aufgetreten.

```
root# smartctl -t short /dev/sda
```

```
root# smartctl -t long /dev/sda
```

```
...
```

```
root# smartctl -l selftest /dev/sda
```

Num	Test_Description	Status	Remaining	Lifetime	LBA
# 1	Extended offline	Completed without error	00%	2592	-
# 2	Short offline	Completed without error	00%	2591	-
# 3	Short offline	Completed without error	00%	40	-

```
smbclient [optionen]
```

```
smbclient verzeichnis [passwort] [optionen]
```

`smbclient` ermöglicht einen interaktiven Zugriff auf Windows-Netzwerkverzeichnisse. In der ersten Syntaxvariante ermitteln Sie mit dem Programm verfügbare Windows-Ressourcen. In der zweiten Variante geben Sie das gewünschte Netzwerkverzeichnis an

(z.B. `//mars/data`, wenn Sie das Verzeichnis *data* auf dem Rechner *mars* nutzen möchten). Nach dem Verbindungsauftbau gelangen Sie in eine Shell. Anschließend können Sie wie beim Kommando [ftp](#) Verzeichnisse mit [ls](#) ansehen, mit [cd](#) wechseln, mit get Dateien auf den lokalen Rechner übertragen (*download*) und mit put Dateien auf dem externen Rechner speichern (*upload*). Einen Überblick über die wichtigsten Kommandos bekommen Sie mit [help](#).

- `-L rechnername`  
liefert die Liste der Netzwerkressourcen auf dem angegebenen Windows-Rechner bzw. Samba-Server.
- `-N`  
verzichtet auf die automatische Passwortabfrage. Die Option ist nur zweckmäßig, wenn Sie wissen, dass für den Zugriff auf eine bestimmte Netzwerkressource kein Passwort erforderlich ist.
- `-U benutzername`  
gibt den Benutzernamen an (nur notwendig, wenn dieser vom aktuellen Login-Namen abweicht).
- `-W arbeitsgruppenname`  
gibt den Namen der Windows-Workgroup an (nur notwendig, wenn diese nicht in `/etc/samba/smb.conf` enthalten ist).

## Beispiel

Zuerst stellt `smbclient` eine Verbindung zur NAS-Festplatte her, anschließend liefert [ls](#) die Liste der dort gespeicherten Verzeichnisse und Dateien:

```
user$ smbclient -U name -W wgname \\\\mynas\\\\myshare  
Password: xxxxxx  
Domain=[wgname] OS=[Unix] Server=[Samba 3.5.2]  
smb: > ls  
data D 0 Apr 5 18:17:11 2019  
file.xy AR 226 Sep 14 00:00:00 2020  
...  
smbpasswd [optionen] [name]
```

`smbpasswd` verändert das Passwort des aktiven bzw. des angegebenen Samba-Benutzers.

- **-a**  
erzeugt einen neuen Samba-Account. Es muss einen gleichnamigen Linux-Account geben.
- **-n**  
entfernt das Passwort für den angegebenen Account. Die Netzwerkverzeichnisse sind nun ohne Passwort erreichbar, falls der Samba-Server entsprechend konfiguriert ist (`null passwords = yes`).
- **-x**  
löscht den angegebenen Samba-Account.

### **smbtree [optionen]**

`smbtree` liefert eine baumförmige Aufstellung aller im Netzwerk erreichbaren Windows-Netzwerkverzeichnisse bzw. Samba-Objekte.

- **-b**  
durchsucht das Netzwerk mit *Broadcast Requests*, anstatt sich auf die Ergebnisse des lokalen Master-Browsers zu verlassen.
- **-dn**  
gibt den Debug-Level zwischen 0 (keine Debugging-Informationen) und 10 (maximales Logging) an. Zur Fehlersuche führen Sie `smbtree` mit der Option `-d10` aus. Sie erhalten dann alle möglichen Debugging-Ausgaben.  
Mitunter scheitert die Verbindung zu Windows-Netzwerkverzeichnissen daran, dass es im Netzwerk keinen Computer bzw. kein Gerät gibt, das als *Master-Browser* agiert. Abhilfe kann dann entweder ein *echter* Windows-Rechner schaffen oder natürlich ein Linux-Rechner, auf dem Samba läuft.
- **-D**  
liefert nur eine Liste der erreichbaren Domänen bzw. Arbeitsgruppen.
- **-N**  
verzichtet auf die Passwortabfrage. Das ist nur sinnvoll, wenn es

Netzwerkressourcen gibt, die ohne Passwort erreichbar sind.

- **-S**  
liefert nur eine Liste der erreichbaren Server.
- **[ -U bzw. --user=*name*[%*password*] ]** gibt den Namen (und optional das Passwort) an.

## Beispiel

Das folgende Kommando liefert eine Liste aller Windows- bzw. Samba-Netzwerkressourcen, auf die der aktuelle Benutzer zugreifen kann:

```
user$ smbtree
Enter user's password:
WORKGROUP
    \\WD-NAS                               My Book World Edition Network Storage
        \\WD-NAS\Public                      Public Share
        \\WD-NAS\Download                    Download Share
        \\WD-NAS\multimedia
        \\WD-NAS\archive
...
...
```

**snap [optionen] kommando**

Ubuntu hat vor einigen Jahren ein neues System zur Installation von Software-Paketen eingeführt. Es arbeitet parallel zum weiterhin dominierenden Debian-Paketesystem (siehe [apt](#) und [dpkg](#)) und hat eine ähnliche Zielsetzung wie das von Red Hat unterstützte Verfahren Flatpak (siehe [flatpak](#)). Snap steht ebenso wie Flatpak allen Distributionen offen, hat bisher aber außerhalb von Ubuntu keine nennenswerte Verbreitung gefunden.

Snap-Pakete werden in der Regel mit dem Gnome-Programm *Software* installiert. Das Kommando `sudo snap` gibt Ihnen die Möglichkeit, Snap-Pakete auch im Terminal zu administrieren.

- **find *suehbegriff***  
sucht den angegebenen Begriff in den Paketbeschreibungen in der Snap-Standardpaketquelle <https://snapcraft.io>.
- **install [--classic] *name***  
installiert das angegebene Paket (erfordert root-Rechte bzw. [sudo](#)).  
Die Option `--classic` muss nur bei manchen Paketen angegeben

werden, die uneingeschränkten Zugriff auf das Dateisystem benötigen und daher mit weniger strikten Sicherheitsregeln ausgeführt werden.

Mit den Optionen `--edge`, `--beta` oder `--candidate` installieren Sie Pakete aus den angegebenen Testkanälen. (Ohne diese Optionen kommt der Release-Kanal zum Einsatz.)

- `list`  
listet bereits installierte Snap-Pakete auf.
- `refresh`  
aktualisiert alle installierten Pakete.
- `remove name`  
deinstalliert das angegebene Paket.

## Beispiel

Das folgende Kommando installiert den Editor Atom:

```
user$ sudo snap install --classic atom
atom 1.51.0 aus Snapcrafters installiert
snapper [optionen] kommando
```

SUSE-Distributionen verwenden standardmäßig ein [btrfs](#)-Dateisystem für die Systempartition. YaST erstellt automatisch vor und nach jedem administrativen Vorgang einen Snapshot. Diese Snapshots können wahlweise mit dem YaST-Modul *Snapper* oder mit dem hier kurz vorgestellten Kommando `snapper` verwaltet werden. Dem auszuführenden `snapper`-Kommando können Sie eine oder mehrere Optionen voranstellen:

- `--iso`  
formatiert Zeiten im ISO-Format.
- `--utc`  
zeigt UTC-Zeiten an (Universal Time Coordinated).

Die Aktionen, die eigentlich ausgeführt werden sollen, werden durch Subkommandos angegeben:

- `cleanup number|timeline|empty-pre-post`  
löscht alte Snapshots, wobei als Kriterium wahlweise eine

Maximalanzahl von Snapshots, deren Alter oder deren Inhalt verwendet wird (siehe auch die Konfigurationsdatei `/etc/snapper/configs/root`).

- `create`  
erzeugt einen neuen Snapshot.
- `delete n bzw. n1-n2`  
löscht die angegebenen Snapshots. Die zu verwendenden Nummern gehen aus `snapper list` hervor.
- `list`  
listet alle Snapshots auf.
- `rollback [n]`  
macht den angegebenen oder den aktuellsten verfügbaren Snapshot zum aktiven Snapshot (genau genommen: zum [btrfs](#)-Default-Subvolume). Ein anschließendes [reboot](#)-Kommando versetzt das System in den Zustand, den es zum Zeitpunkt der Snapshot-Erstellung innehatte.
- `undochange n1..n2 [dateien]`  
macht die zwischen dem Snapshot *n1* und dem Snapshot *n2* durchgeführten Änderungen rückgängig – wahlweise für alle betroffenen Dateien oder nur für die als Parameter angegebenen Dateien.

### **sort [optionen] datei**

`sort` sortiert die angegebene Datei und gibt das Ergebnis auf dem Bildschirm aus. Die Sortierordnung ist durch die Spracheinstellungen und speziell durch die Umgebungsvariable `LC_COLLATE` definiert.

- `-c`  
überprüft, ob die Datei sortiert ist oder nicht.
- `-f`  
behandelt Klein- und Großbuchstaben als gleichwertig.

- **-k *n1, n2***  
berücksichtigt nur die Zeichen zwischen der *n1*-ten und der *n2*-ten Spalte. Die Spaltennummerierung beginnt mit 1. Spalten werden normalerweise durch Leer- oder Tabulatorzeichen voneinander getrennt, siehe aber **-tz**. Wenn *n2* nicht angegeben wird, dann werden alle Zeichen ab *n1* bis zum Zeilenende berücksichtigt.
- **-m**  
fügt zwei oder mehrere vorsortierte Dateien zu einer großen sortierten Datei zusammen: Das geht schneller, als die Dateien vorher zusammenzufügen und erst dann zu sortieren.
- **-n**  
sortiert Zahlen numerisch (also 10 nach 9 und nicht schon nach 1).
- **-o *ergebnisdatei***  
schreibt das Ergebnis in die angegebene Datei. Die Datei darf mit der zu sortierenden Datei übereinstimmen.
- **-r**  
sortiert in umgekehrter Reihenfolge.
- **-tz**  
gibt das Trennzeichen zwischen zwei Spalten an. Die Standardeinstellung ist *white space*, also eine beliebige Mischung aus Leer- und Tabulatorzeichen.
- **-u**  
eliminiert Doppelgänger aus dem Ergebnis (wie `sort | uniq`).

## Beispiel

Das folgende Kommando sortiert das von [ls](#) übergebene Inhaltsverzeichnis nach Anwender- und Gruppennamen. Dateien mit übereinstimmenden Anwender- und Gruppennamen werden ihrer Größe nach sortiert.

```
user$ ls -l | sort -k 3
```

```
source datei
```

`source` führt in einem bash-Script die angegebene Datei so aus, als befänden sich die darin enthaltenen Kommandos an der Stelle des `source`-Kommandos. Nach der Ausführung der Datei wird das laufende Shell-Programm in der nächsten Zeile fortgesetzt. Zur Ausführung der angegebenen Datei wird keine neue Shell gestartet. Alle Variablen inklusive der Parameterliste gelten daher auch für die angegebene Datei. Wenn in dieser Datei `exit` ausgeführt wird, kommt es *nicht* zu einem Rücksprung in das Programm mit dem `source`-Kommando, sondern zu einem sofortigen Ende der Programmausführung.

Zu `source` existiert die Kurzform `._datei`. Dabei steht `_` für ein Leerzeichen.

### **speaker-test [optionen]**

Das Kommando `speaker-test` aus dem Paket `alsa-utils` hilft dabei, Problemen des Audio-Systems auf die Spur zu kommen. Dazu generiert es Testtöne, die über das Audio-System ausgegeben werden.

- `-c n`  
verwendet *n* Audio-Kanäle.
- `-f n`  
spielt einen Sinus-Ton in der angegebenen Frequenz in Hertz. Gleichzeitig muss die Option `-t sine` verwendet werden.
- `-s n`  
testet nur den Audio-Kanal mit der Nummer *n*.
- `-t pink|sine|wav`  
gibt rosa Rauschen, eine Sinuswelle oder ein Sprechmuster aus einer WAV-Datei aus.
- `-w datei`  
spielt die angegebene WAV-Datei ab. Gleichzeitig muss die Option `-t wav` verwendet werden.

### **Beispiel**

Das folgende Kommando gibt über den linken und den rechten Audio-Kanal die Wörter *front left* bzw. *front right* aus. Das Sprechmuster wird aus einer WAV-Datei geladen. Die Ausgabe endet erst mit **Strg**+**C** wieder.

```
user$ speaker-test -t wav -c2
```

```
split [optionen] datei [zieldatei]
```

**split** zerlegt die angegebene Datei in mehrere Einzeldateien. Die Ausgangsdatei wird dabei (je nach Optionen) alle *n* Byte oder Zeilen getrennt. Wenn keine Zieldatei angegeben wird, liefert das Kommando die Dateien *xaa*, *xab* etc. Wenn eine Zieldatei angegeben wird, dann wird für die Ergebnisdateien dieser Dateiname zusammen mit der Zeichenkombination *aa*, *ab* etc. verwendet.

- **-n** bzw. **-l n** bzw. **--lines=n**  
zerlegt die Ausgangsdatei in Einzeldateien mit je *n* Zeilen.
- **-b n** bzw. **--bytes=n**  
trennt die Ausgangsdatei alle *n* Bytes. Die Größenangabe kann auch in KiB, MiB, GiB bzw. in Vielfachen von 1000 Byte angegeben werden: **KB** (1000 Byte), **K** (1024 Byte), **MB** ( $10^6$  Byte), **M** ( $1024^2$  Byte), **GB** ( $10^9$  Byte) oder **G** ( $1024^3$  Byte).
- **-c n** bzw. **--line-bytes=n**  
wie **-b**, allerdings werden die Dateien an Zeilengrenzen zerlegt und sind daher zumeist einige Byte kleiner als *n*.

## Beispiel

Im folgenden Beispiel zerlegt **split** die Archivdatei *backup.tgz* in Einzeldateien zu je 4 GiB und benennt sie mit *dvd.aa*, *dvd.ab* etc. Diese Dateien könnten anschließend auf DVDs gebrannt werden.

```
user$ split -C 4G backup.tgz dvd.
```

[cat](#) setzt die Einzeldateien wieder zu einer Gesamtdatei zusammen:

```
user$ cat dvd.* > total.tar
```

```
sqlite3 [optionen] datenbankdatei
```

```
sqlite3 [optionen] datenbankdatei 'sqlkommando1;sql2;...'
```

`sqlite3` wird normalerweise als Shell zur Bearbeitung von SQLite-Datenbanken genutzt. Nach dem Start können Sie interaktiv SQL-Kommandos ausführen, wobei alle Kommandos mit einem Semikolon abgeschlossen werden müssen. Neben den üblichen SQL-Kommandos kennt `sqlite3` einige spezifische Kommandos, die mit einem Punkt beginnen, z.B. `.help` zur Anzeige eines Hilfetexts oder `.tables` zur Auflistung aller Tabellen in der Datenbank. `.quit` oder `[Strg] + [D]` beenden `sqlite3`.

In der zweiten Syntaxvariante führt `sqlite3` die angegebenen SQL-Kommandos direkt aus.

## Beispiel

Das folgende Kommando ermittelt, welche Musikalben in einer ownCloud-Datenbank gespeichert sind:

```
user$ sqlite owncloud.db
sqlite> select * from oc_media_albums;
1|Nightclub|1|
2|The Monroe Doctrine|2|
3|The Best Of Herbie Mann|3|
...
sqlite> .quit
ss [optionen]
```

`ss (socket statistics)` analysiert die Netzwerkaktivität des lokalen Rechners. Wenn das Kommando ohne Optionen ausgeführt wird, liefert es eine Liste aller *non-listening* TCP-Sockets mit einer aktiven Verbindung. `ss` bietet eine ähnliche Funktionalität wie [netstat](#), gilt aber als das modernere Kommando und liefert in manchen Fällen detailliertere Informationen.

- -4 bzw. -6  
berücksichtigt nur IPv4- bzw. nur IPv6-Verbindungen.
- -a  
zeigt alle Sockets an, sowohl *listening* als auch *non-listening*.
- -e  
liefert noch mehr Details (*extended*).

- **-n**  
zeigt numerische Portnummern an, nicht die Namen der Dienste.
- **-p**  
gibt an, welche Prozesse welche Sockets nutzen.
- **-r**  
ersetzt Portnummern durch die Namen der Dienste.
- **-s**  
zeigt nur zusammenfassende Informationen an (*summary*).
- **-t bzw. -u**  
berücksichtigt nur TCP- bzw. nur UDP-Verbindungen.

## Beispiel

Das folgende Kommando liefert eine Liste aller aktiven TCP-Sockets für IPv6-Verbindungen:

```
user$ ss -6 -t -a
State      Recv-Q Send-Q    Local Address:Port          Peer Address:Port
LISTEN      0      100          :::pop3                  ::::*
LISTEN      0      100          :::imap2                  ::::*
LISTEN      0      128          :::http                  ::::*
LISTEN      0      100          :::smtp                  ::::*
LISTEN      0      128          :::https                 ::::*
LISTEN      0      100          :::imaps                 ::::*
LISTEN      0      100          :::pop3s                ::::*
TIME-WAIT    0       0          ::ffff:5.9.22.28:http   ::ffff:188.192.181.195:50045
TIME-WAIT    0       0          ::ffff:5.9.22.28:http   ::ffff:188.192.181.195:50047
ESTAB       0     46200         ::ffff:5.9.22.28:http   ::ffff:46.10.92.30:22893
FIN-WAIT-1  0       1          ::ffff:5.9.22.28:https  ::ffff:203.134.55.243:51106
FIN-WAIT-2  0       0          ::ffff:5.9.22.28:http   ::ffff:46.10.92.30:24262
...

```

```
ssh [optionen] rechnername
ssh [optionen] rechnername kommando
```

In der ersten Syntaxvariante öffnet ssh eine Shell auf einem anderen Rechner und ermöglicht es so, auf diesem Rechner interaktiv zu arbeiten. Die zweite Syntaxvariante führt ein einzelnes Kommando auf dem zweiten Rechner aus. ssh funktioniert nur, wenn auf dem zweiten Rechner ein SSH-Server läuft.

- **-l loginname**  
verwendet auf dem zweiten Rechner den angegebenen Login-Namen statt des aktuellen Benutzernamens. Der gewünschte

Login-Name kann auch in der Form `ssh loginname@rechnername` angegeben werden.

- `-L localport:host:hostport`  
bildet einen Tunnel zwischen dem lokalen Rechner (Port `localport`) und dem zweiten Rechner (Port `hostport`).
- `-o option=wert`  
ermöglicht die Übergabe weiterer Optionen in der gleichen Syntax wie in der Datei `/etc/ssh/ssh_config` (siehe `man ssh_config`). Beispielsweise bewirkt `-o StrictHostKeyChecking=no`, dass SSH darauf verzichtet, die Host-Daten mit bereits gespeicherten Einträgen in `.ssh/known_hosts` zu überprüfen. Das kann bei der automatisierten Administration vieler virtueller Maschinen hilfreich sein.
- `-t`  
erzwingt die Verwendung eines Pseudo-Terminals für die Ausführung von Kommandos in der Form `ssh -t hostname "kommandos"`. Das ermöglicht es, interaktive Programme wie [top](#) via SSH zu starten. Die Verbindung wird erst gestoppt, wenn das externe Kommando endet (bei [top](#) durch die Eingabe von `Q`).
- `-T`  
verhindert die Aktivierung eines Pseudo-Terminals. Eine Anwendung dieser Option finden Sie bei den folgenden Beispielen.
- `-v`  
liefert umfassende Debugging-Informationen. Wenn Sie Login-Probleme haben, hilft `ssh -v` oft dabei, die Ursache herauszufinden. Die Option kann bis zu dreimal angegeben werden (also `-v -v -v`), `ssh` liefert dann noch mehr Debugging-Informationen.
- `-X`  
ermöglicht die Ausführung von X-Programmen.

## Beispiele

Das folgende Kommando zeigt, wie Sie sich auf einem externen Rechner einloggen, wobei der Benutzername unverändert bleibt. Anschließend können Sie auf dem externen Rechner (z.B. Ihrem Root-Server) Kommandos ausführen.

```
kofler@uranus$ ssh kofler.info  
kofler@kofler.info's password: *****
```

Wenn Sie mit ssh zum ersten Mal eine Verbindung zu einem anderen Rechner herstellen, erscheint oft eine Warnung nach dem folgenden Muster:

```
The authenticity of host 'kofler.info (1.2.3.4)' can't be established.  
RSA1 key fingerprint is 1e:0e:15:ad:6f:64:88:60:ec:21:f1:4b:b7:68:f4:32.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'kofler.info,1.2.3.4' to the list of known hosts.
```

Das folgende Kommando zeigt, wie ich den gesamten /var/www-Verzeichnisbaum meines Webservers kofler.info in das lokale Verzeichnis ~/bak kopiere. Dabei wird ssh dazu genutzt, ein einzelnes Kommando ([tar](#)) auszuführen. Das setzt voraus, dass kofler@kofler.info alle Dateien aus /var/www lesen kann.

Grundsätzlich wäre es auch möglich, das [tar](#)-Kommando mit ssh -l root mit root-Rechten auszuführen. Allerdings sind die meisten SSH-Server aus Sicherheitsgründen so konfiguriert, dass ein direkter root-Login unmöglich ist.

Die Ergebnisse von [tar](#) werden via SSH abhörsicher auf den lokalen Rechner übertragen und vom zweiten [tar](#)-Kommando weiterverarbeitet.

```
kofler@uranus$ ssh kofler.info tar -cf - /var/www | tar -xC ~/bak/ -f -  
kofler@kofler.info's password: *****
```

Um in einem Script mehrere Kommandos via SSH auf einem anderen Server auszuführen, können Sie sich der HEREDOC-Syntax der bash in Kombination mit der ssh-Option -t bedienen. Das folgende Kommando ändert das Passwort von someaccount auf einem externen Host und führt danach einige weitere Kommandos aus.

```
#!/bin/bash  
pw=strengeheim
```

```
...
ssh -T root@host <<ENDSSH
echo someaccount:$pw | chpasswd
rm -f /etc/file1
cp /root/file2 /userxy/file3
ENDSSH
```

### **ssh-copy-id [optionen] user@host**

ssh-copy-id fügt den öffentlichen Schlüssel des lokalen Rechners in die Datei *.ssh/authorized\_keys* des Host-Rechners ein. Wenn es mehrere Schlüssel gibt, entscheidet das Kommando ssh-add -L, welcher Schlüssel übertragen wird.

- **-i [schlüsseldatei]**

überträgt die angegebene Schlüsseldatei bzw. *.ssh/id\_rsa.pub*. Sollte das Script ssh-copy-id nicht zur Verfügung stehen oder die Schlüsselübertragung an den Sicherheitseinstellungen des externen SSH-Servers scheitern, muss die öffentliche Schlüsseldatei (in der Regel *.ssh/id\_rsa.pub*) manuell zum Server übertragen und dort der Datei *.ssh/authorized\_keys* hinzugefügt werden.

Wenn auf dem Host-System Fedora, RHEL oder ein RHEL-Derivat läuft (CentOS etc.), kann der Einsatz von ssh-copy-id dazu führen, dass das Verzeichnis *.ssh* und die Datei *authorized\_keys* ohne die SELinux-Kontextinformationen erzeugt werden. Abhilfe schafft eines der beiden folgenden Kommandos auf dem Hostrechner:

```
root# /sbin/restorecon -r /root/.ssh
root# /sbin/restorecon -r /home/username/.ssh
```

### **ssh-keygen [optionen]**

ssh-keygen erzeugt, verwaltet und konvertiert kryptografische Schlüssel, die zur Authentifizierung mit SSH verwendet werden können. Wenn ssh-keygen ohne weitere Optionen aufgerufen wird, erzeugt es ein RSA-Schlüsselpaar mit einer Schlüssellänge von 2048 Bits. Der private und der öffentliche Teil des Schlüssels wird in den Dateien *.ssh/id\_rsa* und *.ssh/id\_rsa.pub* gespeichert.

Der Zugriff auf den Schlüssel wird standardmäßig durch ein Passwort abgesichert; wenn Sie statt der Passworteingabe einfach  drücken, entfällt dieses Sicherheitsmerkmal. Wenn ein

privater, nicht durch ein Passwort abgesicherter Schlüssel einem Dritten in die Hände fällt, kann dieser Dritte sich ohne Weiteres auf allen Rechnern anmelden, auf denen Sie den öffentlichen Teil des Schlüssels installiert haben!

Durch zusätzliche Optionen können Sie beim Erzeugen neuer Schlüssel unterschiedliche Verschlüsselungsverfahren, Schlüssellängen und Speicherorte angeben, vorhandene Schlüssel konvertieren etc.

### **stat [optionen] dateien**

stat liefert ausführliche Informationen über die angegebenen Dateien, unter anderem die Zugriffsrechte in der [ls](#)-Schreibweise und als Oktalzahl, die Anzahl der genutzten Blöcke, den Zeitpunkt der letzten Änderung und des letzten Lesezugriffs etc.

- **-c formatstring**  
formatiert die Ausgabe entsprechend der Formatzeichenkette.
- **-f**  
liefert Informationen über das Dateisystem, in dem sich die Datei befindet (anstelle von Informationen über die Datei). In der Zeichenkette können unter anderem die folgenden Codes verwendet werden:
  - %a: Zugriffsrechte in Okatalschreibweise
  - %A: Zugriffsrechte in [ls](#)-Schreibweise
  - %c: SELinux-Kontext
  - %F: Dateityp (Datei, Verzeichnis, Block-Device etc.)
  - %g: GID-Nummer der Gruppe, der die Datei zugeordnet ist
  - %G: Name der Gruppe, der die Datei zugeordnet ist
  - %h: Anzahl der Hard-Links
  - %n: Dateiname
  - %s: Dateigröße in Bytes
  - %u: UID-Nummer des Dateibesitzers
  - %U: Name des Dateibesitzers

## Beispiel

Ohne Optionen liefert stat eine übersichtliche Zusammenstellung diverser Metadaten einer Datei. Mit -c kann das Ergebnis auf Einzelinformationen reduziert werden, im zweiten Beispiel auf die oktalen Zugriffsrechte und den Dateinamen.

```
user$ stat druck.pdf
  Datei: druck.pdf
  Größe: 390758          Blöcke: 768          EA Block: 4096   reguläre Datei
Gerät: fd01h/64769d      Inode: 397971        Verknüpfungen: 1
Zugriff: (0664/-rw-rw-r--) Uid: ( 1000/ kofler)  Gid: ( 1000/ kofler)
Kontext: unconfined_u:object_r:user_home_t:s0
Zugriff   : 2013-09-24 11:35:17.316000000 +0200
Modifiziert: 2013-08-06 17:53:16.083000000 +0200
Geändert   : 2013-08-06 17:53:16.083000000 +0200
Geburt     : -
user$ stat -c "%a %n" druck.pdf
664 druck.pdf
```

### **strace [optionen] kommando**

strace führt das angegebene Kommando oder Programm aus und zeigt alle Systemfunktionen an, die das Kommando aufruft. strace ist damit ein wichtiges Werkzeug zur Fehlersuche in kompilierten Programmen. Durch diverse Optionen kann gesteuert werden, welche (Zusatz-)Informationen das Programm ausgeben soll.

### **strings [optionen] datei**

Das Kommando extrahiert Zeichenketten aus einer Binärdatei und zeigt diese an. Das ist beispielsweise dann praktisch, wenn Sie einen Text (z.B. eine Fehlermeldung oder einen Dateinamen) in einer Programmdatei suchen.

## Beispiele

Das folgende Kommando ermittelt alle Zeichenketten, die sich in der Binärdatei /bin/ls befinden und das Wort error enthalten:

```
root# strings /bin/ls | grep error
error
error initializing month strings
write error
```

Im folgenden Beispiel wollte ich ergründen, ob bzw. auf welche Tastaturkonfigurationsdatei(en) das systemd-Programm systemd-vconsole-setup unter openSUSE Leap zugreift:

```
user$ strings /usr/lib/systemd/systemd-vconsole-setup | grep keyboard  
/etc/sysconfig/keyboard  
Failed to read /etc/sysconfig/keyboard: %s
```

### **su [optionen] [user]**

su (*substitute user*) ohne Optionen wechselt mit Passworteingabe in den root-Modus. Damit ist bis zum nächsten [exit](#)-Kommando root der aktive User.

Optional kann bei su statt des Standard-Users root auch ein anderer User angegeben werden. Wenn su von root ausgeführt wird, muss beim User-Wechsel nicht einmal ein Passwort angegeben werden.

- -c '*kommando*'  
führt nur das angegebene Kommando mit root-Rechten aus.
- -l bzw. --login  
beim User-Wechsel wird die neue Shell als Login-Shell gestartet. Das heißt, es werden alle Login-Dateien mit eingelesen. Das ist erforderlich, damit Umgebungsvariablen wie PATH korrekt konfiguriert werden.

### **sudo [optionen] [var1=wert1 var2=wert2] kommando**

sudo führt ein Kommando aus, als würde es von einem anderen Benutzer (normalerweise root) ausgeführt. Damit können dank sudo gewöhnliche Benutzer administrative Aufgaben übernehmen bzw. systemkritische Kommandos ausführen, ohne dazu das root-Passwort zu kennen.

Bevor sudo die Ausführung eines Programms erlaubt, muss dieses Recht für einen bestimmten Benutzer und für ein bestimmtes Programm in der Datei [/etc/sudoers](#) angegeben werden.

- -b  
startet das angegebene Kommando im Hintergrund.
- -E  
erhält den aktuellen Inhalt der Umgebungsvariablen. Die Option ist erforderlich, um unter Wayland ein Programm mit root-Rechten im Grafikmodus zu starten.

- **-H**  
trägt das Home-Verzeichnis des Benutzers, für dessen Account das Kommando ausgeführt wird, in die \$HOME-Umgebungsvariable ein (also üblicherweise /root).
- **-i bzw. --login**  
startet die Shell des Zielbenutzers (-u) bzw. von root. Dabei werden die Konfigurationsdateien des Zielbenutzers ausgewertet, also .profile, .bashrc oder .zshrc etc.
- **-K**  
löscht das gespeicherte Passwort. Beim nächsten Aufruf von sudo muss neuerlich das Passwort angegeben werden.
- **-s**  
startet die in der lokalen Umgebungsvariablen \$SHELL angegebene Shell. Das ermöglicht es, mehrere Kommandos in einer Shell auszuführen, ohne jedes Mal sudo voranzustellen. Die Shell kann mit [exit](#) oder **[Strg]+[D]** verlassen werden.
- **-u benutzer**  
startet das Kommando für den angegebenen Benutzer (statt für root).

Im sudo-Kommando definierte Variablen werden als Umgebungsvariablen weitergegeben. Daher ist sudo var1=x var2=y cmd eine Kurzschreibweise für die folgenden Kommandos:

```
user$ sudo -s
root# export var1=x
root# export var2=y
root# cmd
root# exit
```

## Beispiele

Im folgenden Beispiel wird sudo dazu verwendet, das Kommando apt install mit root-Rechten auszuführen:

```
user$ sudo apt install gimp
[sudo] password for user: *****
```

...

Das zweite Beispiel ermöglicht ein interaktives Arbeiten als Benutzer martin mit dessen Shell und mit dessen lokalen Shell-

Einstellungen:

```
user$ sudo -u martin -i  
martin$ pwd  
/home/martin  
martin$ exit  
user$
```

## Beispiel mit Ein-/Ausgabeumleitung

Wenn Sie versuchen, mit sudo ein Kommando auszuführen, das auf die Ein- oder Ausgabeumleitung zurückgreift, tritt unter Umständen ein Fehler auf:

```
user$ cd /etc/samba  
user$ sudo grep -Ev '^#|^$' smb.conf > smb.conf.condensed  
-bash: smb.conf.condensed: Keine Berechtigung
```

Beabsichtigt war, aus *smb.conf* alle Kommentare zu eliminieren und die resultierende Datei in *smb.conf.condensed* zu speichern. Das Problem ist aber, dass die Ausgabeumleitung nicht mit sudo-Rechten ausgeführt wird, sondern innerhalb der bash mit den Rechten des aktuellen Benutzers läuft. Dieser darf in */etc* natürlich keine Dateien verändern.

Damit derartige Konstrukte funktionieren, müssen Sie mit sudo eine Shell starten und das Kommando an diese Shell übergeben:

```
user$ sudo sh -c 'grep -Ev "^#|^$" smb.conf > smb.conf.condensed'  
svn kommando
```

Das Kommando **svn** aus dem Paket **subversion** steuert das gleichnamige Programm zur Versionsverwaltung. Subversion (SVN) wird häufig eingesetzt, wenn mehrere Personen zugleich ein Programm entwickeln. Subversion verwaltet ein zentrales Codearchiv (das Subversion-Repository auf dem Subversion-Server) und ermöglicht es, jede Änderung am Code zurückzuverfolgen und bei Bedarf rückgängig zu machen.

- **add [-N] datei/verzeichnis**  
fügt die Datei der Subversion-Kontrolle hinzu. Die Datei wird mit dem nächsten SVN-Commit in das Subversion-Repository übertragen. Bei Verzeichnissen bewirkt **-N**, dass der Inhalt des Verzeichnisses *nicht* berücksichtigt wird.

- `checkout/co [--username name] url`  
 überträgt erstmalig alle Dateien vom SVN-Server in das aktuelle Verzeichnis auf dem lokalen Rechner. Dabei muss in der Regel ein Passwort angegeben werden (je nach Konfiguration des SVN-Servers). `svn` speichert einen Hash des Passworts in `.subversion/auth/*`, sodass die Passwortangabe in Zukunft nicht wiederholt werden muss.
- `commit/ci -m 'kommentar'`  
 lädt die zuletzt durchgeführten Änderungen in das Subversion-Repository hoch. Vor jedem Commit sollte `svn update` ausgeführt werden. Wenn Sie vor dem Commit überprüfen möchten, welche Änderungen gespeichert werden, führen Sie vorher `svn status` aus.
- `copy oldurl newurl -m 'kommentar'`  
 erzeugt eine Kopie des SVN-Verzeichnisses, wobei die Kopie unter der Kontrolle von Subversion bleibt. Kopien werden häufig zur Verwaltung von Zweigen (*branches*) verwendet. Es ist üblich, das Unterverzeichnis *branches* zur Speicherung der Zweige zu verwenden.
- `diff -r n1:n2 [datei]`  
 zeigt an, wie sich die angegebene Datei von Revision *n1* bis zur Revision *n2* verändert hat. Wenn keine Datei angegeben wird, ermittelt `svn diff` die Auflistung *aller* Änderungen.
- `diff -r url1[@n1] url2[@n2]`  
 ermittelt die Unterschiede zwischen zwei SVN-Zweigen. Wenn keine Revisionsnummern angegeben werden, berücksichtigt `svn` jeweils die aktuellste Version jedes Zweigs.
- `help [kommando]`  
 liefert eine Liste aller SVN-Kommandos bzw. einen detaillierten Hilfetext zum angegebenen Kommando.
- `list/ls url`  
 liefert eine Liste aller unter SVN-Kontrolle befindlichen Dateien

auf dem SVN-Server. Das Kommando ist vor allem dann zweckmäßig, wenn Sie die Dateien nicht herunterladen möchten.

- `log [datei]`  
listet alle SVN-Commits auf (die letzte Änderung zuerst). Wenn eine Datei angegeben wird, liefert `svn log` nur solche Commits, durch die diese Datei verändert wurde.
- `move/mv/ren datei1 datei2`  
gibt einer unter SVN-Kontrolle stehenden Datei einen neuen Namen.
- `revert datei`  
widerruft alle Änderungen, die seit dem letzten SVN-Update an der lokalen Datei durchgeführt wurden.
- `rm datei`  
entfernt die Datei aus Subversion. Die Datei wird mit dem nächsten SVN-Commit auf dem Subversion-Server gelöscht.
- `status/st`  
fasst die Änderungen zusammen, die seit dem letzten SVN-Update an den lokalen Dateien vorgenommen wurden.
- `switch url`  
aktiviert den Zweig (*branch*), den Sie durch seine URL angeben.
- `update/up`  
aktualisiert die lokalen Dateien mit der neuesten Version vom Subversion-Server.
- `update -r n datei`  
ersetzt die angegebene Datei durch eine ältere Version (Revision *n*).

## Beispiel

Die folgenden Zeilen zeigen, wie Sie einem Projekt eine neue PHP-Datei hinzufügen:

```
user$ svn update  
user$ svn add modules/html.php  
user$ svn commit -m 'neues HTML-Modul'
```

```
swapon device  
swapoff device
```

`swapon` aktiviert das angegebene Device (zumeist eine Festplattenpartition) bzw. die stattdessen angegebene Datei als Swap-Bereich.

Die Partition bzw. Datei muss vorher mit [mkswap](#) als Swap-Bereich formatiert werden. `swapon` wird beim Hochfahren von Linux für alle in */etc/fstab* aufgezählten Swap-Bereiche automatisch ausgeführt. Ein Beispiel zu `swapon` finden Sie bei der Beschreibung von [mkswap](#). `swapoff` deaktiviert die angegebene Swap-Datei oder -Festplattenpartition.

```
sync
```

Führt alle gepufferten Schreiboperationen auf den Festplatten aus. Wenn aus irgendeinem Grund ein geordnetes Beenden von Linux nicht möglich ist – d.h., wenn die Kommandos [shutdown](#), [reboot](#) und [halt](#) nicht ausführbar sind und der Rechner nicht auf **[Alt]+[Strg]+[Entf]** reagiert –, dann sollte `sync` unmittelbar vor dem Ausschalten ausgeführt werden. Das ist aber nur eine Notlösung!

```
sysctl optionen
```

`sysctl` liest den Zustand von Kernelparametern bzw. verändert ihren Wert im laufenden Betrieb. Dauerhafte Änderungen müssen Sie in die Datei */etc/sysctl.conf* eintragen.

- **-a**  
liefert eine Liste aller verfügbaren Kernelparameter inklusive der aktuellen Einstellung.
- **-n parameter**  
liefert den aktuellen Wert des angegebenen Kernelparameters.
- **-p [dateiname]**  
verarbeitet */etc/sysctl.conf* bzw. die angegebene Datei und aktiviert alle dort gespeicherten Einstellungen.

- **-w parameter=wert**  
verändert den angegebenen Parameter. Die Option **-w** darf auch weggelassen werden.

## Beispiel

Die beiden folgenden Kommandos aktivieren zuerst die Forwarding-Funktionen des Kernels und dann das Masquerading für die Netzwerkschnittstelle `eth0`:

```
root# sysctl -w net.ipv4.ip_forward=1  
root# iptables -A POSTROUTING -t nat -o eth0 -j MASQUERADE
```

### **systemctl** Kommando

`systemctl` dient zur Administration des Init-Prozesses bei Distributionen, die `systemd` als Init-System verwenden. Das ist bei nahezu allen aktuellen Distributionen der Fall. Sie können mit `systemctl` eine Liste aller durch `systemd` verwalteten Prozesse ermitteln, einzelne Hintergrunddienste beenden oder neu starten etc.

`systemctl` unterscheidet unter anderem zwischen gewöhnlichen Diensten (`services`), Sockets, Devices, Mount-Diensten und Zielen (`targets`, vergleichbar mit Init-V-Runleveln).

- `daemon-reload`  
liest die Konfigurationsdateien ein. Dieses Kommando muss ausgeführt werden, damit Konfigurationsänderungen wirksam werden.
- `enable/disable name`  
aktiviert den angegebenen Dienst dauerhaft bzw. deaktiviert ihn wieder. Bei der Ausführung des Kommandos werden die entsprechenden Links eingerichtet bzw. wieder entfernt. Bei aktuellen `systemd`-Versionen bewirkt die zusätzliche Option `--now`, dass der betreffende Dienst auch gleich gestartet bzw. gestoppt wird.
- `get-default`  
ermittelt das Default-Target.

- `halt/poweroff/reboot`  
fährt das System herunter bzw. startet es neu.
- `is-active name`  
testet, ob der angegebene Dienst aktiv ist.
- `isolate name.target`  
aktiviert das angegebene Target sowie alle davon abhängigen Dienste. Nicht benötigte Dienste werden gestoppt. `isolate` hat damit eine ähnliche Funktion wie ein Runlevel-Wechsel bei einem Init-V-System.
- `list-timers`  
gibt einen Überblick über aktive Timer, also periodisch durch `systemd` auszuführende Jobs. Bei jedem Job wird der Zeitpunkt der letzten und der geplanten nächsten Ausführung angegeben.
- `list-units`  
liefert eine Liste aller von `systemd` verwalteten Dienste, Sockets, Targets etc. Mehrseitige Ausgaben werden durch [`less`](#) geleitet, sodass sie seitenweise gelesen werden können. Wenn Sie das nicht wünschen, geben Sie die Option `--no-pager` an. Das Ergebnis kann mit `--type=...` gefiltert werden. Zulässige Unit-Typen sind [`service`](#), [`socket`](#), [`target`](#), [`device`](#), [`mount`](#), [`automount`](#) und [`snapshot`](#).
- `reload/restart name`  
fordert den Prozess zum Neueinlesen der Konfiguration auf bzw. startet den Dienst gänzlich neu.
- `set-default name.target`  
legt das Default-Target für künftige Bootprozesse fest.
- `show name`  
liefert detailliertere Daten als `status`. Das Ergebnis ist zeilenweise strukturiert und kann verhältnismäßig einfach durch ein Script verarbeitet werden.
- `start/stop name`  
startet bzw. stoppt den angegebenen Init-Prozess bzw. Dienst.

- **status name**

liefert Statusinformationen zum angegebenen Dienst in einer gut lesbaren Form.

`systemctl` gilt standardmäßig für Dienste, die auf Systemebene ausgeführt werden. Um Dienste auf Benutzerebene zu bearbeiten, müssen Sie die Option `--user` übergeben. Das ist z.B. auf modernen Desktopsystemen sinnvoll, wo Gnome auf `systemd` zurückgreift.

## Beispiele

Die folgenden Zeilen zeigen typische `systemctl`-Kommandos:

```
root# systemctl start cups (CUPS-Dämon starten)
root# systemctl stop cups (CUPS-Dämon stoppen)
root# systemctl restart cups (CUPS-Dämon neu starten)
root# systemctl reload cups (Konfiguration des CUPS-Dämons neu einlesen)
root# systemctl status cups (Status des CUPS-Dämons ermitteln)
root# systemctl enable cups (CUPS-Dämon in Zukunft automatisch starten)
        Created symlink /etc/systemd/system/sockets.target.wants/cups.socket ->
                      /lib/systemd/system/cups.socket.
Created symlink /etc/systemd/system/multi-user.target.wants/cups.path ->
                      /lib/systemd/system/cups.path.
root# systemctl disable cups (CUPS-Dämon nicht mehr automatisch starten)
        Removed /etc/systemd/system/sockets.target.wants/cups.socket.
        Removed /etc/systemd/system/multi-user.target.wants/cups.path.
root# systemctl isolate reboot.target (Neustart des Rechners)
```

# T

## **tac** datei

tac gibt die angegebene Textdatei in umgekehrter Reihenfolge auf dem Bildschirm aus, d.h. die letzte Zeile zuerst. Der merkwürdige Kommandoname ergibt sich aus der Verdrehung der Buchstaben des [cat](#)-Kommandos, das Textdateien in richtiger Reihenfolge ausgibt.

## **tail** [optionen] datei

tail gibt die letzten zehn Zeilen einer Textdatei auf dem Bildschirm aus.

- **-n zeilen**  
gibt die angegebene Anzahl von Zeilen aus.
- **-n +offset**  
beginnt die Ausgabe mit der angegebenen Zeile. tail -n +2 datei gibt somit die gesamte Datei mit Ausnahme der ersten Zeile aus.
- **-f**  
liest die Datei regelmäßig aus und gibt alle neuen Zeilen aus. In dieser Form eignet sich tail vor allem zur Beobachtung von Protokolldateien.

## **Beispiel**

Das folgende Kommando zeigt die letzten zehn Zeilen von *syslog* an. Wenn neue Zeilen hinzukommen, werden auch diese angezeigt, sodass nach kurzer Zeit der ganze Bildschirm genutzt wird – und nicht nur zehn Zeilen. Die Ausführung endet mit **[Strg]+[C]**.

```
root# tail -f /var/log/syslog
```

## **tar** aktion [optionen] dateien **tar** aktion [optionen] verzeichnisse

tar vereint mehrere Dateien oder ganze Verzeichnisse in einem sogenannten Archiv bzw. extrahiert aus diesem Archiv wieder ihre Bestandteile. tar war ursprünglich als Tool zum Lesen und Schreiben von Daten auf einem Streamer konzipiert. tar greift

deswegen standardmäßig auf den installierten Streamer (zumeist `/dev/tape` oder `/dev/rmt0`) zu. Wenn Sie ein Archiv in einer Datei anlegen möchten (beispielsweise, um diese Datei anschließend auf eine CD zu brennen), müssen Sie die Option `-f datei` angeben.

Da `tar` je nach Angabe der Optionen die zu archivierenden Dateien auch komprimiert, ist es in seiner Funktion mit dem unter Windows verbreiteten Programm WinZIP zu vergleichen. Die typische Kennung für Archivdateien ist `.tar`. Wenn die Archivdatei komprimiert ist, lauten die Kennungen zumeist `.tgz`, `.tar.gz`, `.tar.bz2` oder `.tar.xz`.

Die Steuerung von `tar` erfolgt in zwei Stufen: Zum einen muss eine Aktion angegeben werden, die `tar` ausführen soll, und zum anderen kann diese Aktion durch eine oder mehrere Optionen gesteuert werden. Auch wenn Aktionen und Optionen formal gleich aussehen, besteht ein wesentlicher Unterschied: Es muss genau eine Aktion vor allen anderen Optionen angegeben werden. Während in den nächsten Zeilen alle Aktionen kurz beschrieben sind, werden anschließend nur die wichtigsten Optionen aufgezählt (siehe `man tar`).

Auf vielen Unix-Systemen erfolgt die Steuerung von `tar` zwar mit denselben Kommandos und Optionen, die Syntax ist aber anders: Es werden alle Kommandos und Optionen als ein Block ohne die üblichen Optionsstriche angegeben, etwa `tar cvf name.tar pfad`. GNU-`tar` versteht beide Syntaxvarianten, d.h., es ist Ihnen überlassen, ob Sie Optionsstriche verwenden oder nicht.

## Aktionen

- **-A bzw. --catenate bzw. --concatenate**  
hängt an ein vorhandenes Archiv ein weiteres Archiv an. Diese Option ist nur für Streamer geeignet (nicht für Archivdateien).
- **-c bzw. --create**  
erzeugt ein neues Archiv, d.h., ein eventuell vorhandenes Archiv

wird überschrieben.

- **-d bzw. --diff bzw. --compare**  
vergleicht die Dateien des Archivs mit den Dateien des aktuellen Verzeichnisses und stellt eventuelle Unterschiede fest.
- **--delete**  
löscht Dateien aus dem Archiv. Nur für Archivdateien (nicht für Streamer) geeignet.
- **-r bzw. --append**  
erweitert das Archiv um zusätzliche Dateien.
- **-t bzw. --list**  
zeigt das Inhaltsverzeichnis des Archivs an.
- **-u bzw. --update**  
erweitert das Archiv um neue oder geänderte Dateien. Die Option kann nicht für komprimierte Archive verwendet werden.  
Vorsicht: Das Archiv wird immer größer, weil bereits vorhandene Dateien nicht überschrieben werden! Die neuen Dateien werden einfach an das Ende des Archivs angehängt.
- **-x bzw. --extract**  
extrahiert die angegebenen Dateien aus dem Archiv und kopiert sie in das aktuelle Verzeichnis. Die Dateien werden dabei nicht aus dem Archiv gelöscht.

## Optionen

- **-C verzeichnis**  
extrahiert die Dateien in das angegebene (statt in das aktuelle) Verzeichnis.
- **-f datei**  
verwendet die angegebene Datei als Archiv (anstatt auf den Streamer zuzugreifen). Wenn statt eines Dateinamens ein einfacher Bindestrich angegeben wird (-f -), werden die Daten an den Standardausgabekanal weitergeleitet bzw. von der Standardeingabe gelesen. Das ist vor allem dann sinnvoll, wenn

zwei tar-Kommandos durch | verknüpft werden sollen  
(typischerweise zum Transportieren ganzer Dateibäume).

- **-g timestamp-datei**  
speichert in der binären Timestamp-Datei Metadaten über die mit tar gesicherten Dateien und berücksichtigt bei weiteren Aufrufen von tar nur die Dateien, die sich seither verändert haben. Die Option bietet eine einfache Möglichkeit, inkrementelle Backups zu erzeugen (siehe auch das letzte der folgenden Beispiele).
- **-j bzw. --bzip2**  
komprimiert bzw. dekomprimiert das gesamte Archiv durch [bzip2](#) (siehe auch [-z!](#)).
- **-J bzw. --xz**  
komprimiert bzw. dekomprimiert das gesamte Archiv durch [xz](#). Das setzt voraus, dass das Kommando [xz](#) zur Verfügung steht. Bei vielen Distributionen muss dazu das Paket `xz-utils` installiert werden.
- **-L n bzw. --tape-length n**  
gibt die Kapazität des Streamers in KiB an. Wenn die Kapazität durch die Größe des Archivs überschritten wird, fordert tar zu einem Wechsel des Magnetbands auf.
- **-N datum bzw. --after-date datum bzw. --newer datum**  
archiviert nur Dateien, die aktueller als das angegebene Datum sind. Damit lassen sich unkompliziert inkrementelle Backups erstellen.
- **-p bzw. --preserve-permissions**  
belässt beim Extrahieren der Dateien die Zugriffsrechte unverändert. Die Option gilt standardmäßig, wenn tar von root ausgeführt wird.
- **-T datei bzw. --files-from datei**  
archiviert bzw. extrahiert die in der Datei angegebenen

Dateinamen.

- **-v** bzw. **--verbose**  
zeigt während der Arbeit alle Dateinamen auf dem Bildschirm an.  
Wenn **-v** in Kombination mit dem Kommando **t** verwendet wird,  
werden zusätzliche Informationen zu den Dateien angezeigt  
(Dateigröße etc.). Wenn die Option zweimal angegeben wird,  
werden die Informationen noch ausführlicher.
- **-w** bzw. **--verify**  
überprüft nach dem Schreiben die Korrektheit der gerade  
archivierten Dateien. Kann nicht für komprimierte Archive  
verwendet werden.
- **-z** bzw. **--gzip**  
komprimiert bzw. dekomprimiert das gesamte Archiv durch [gzip](#).  
Für das Anlegen von \*.tgz-Dateien ist diese Option sehr  
praktisch. Wenn dagegen tatsächlich Daten auf einem Streamer  
gespeichert werden, kann die Option gefährlich sein: Ein einziger  
Fehler auf dem Magnetband kann das gesamte Archiv  
unbrauchbar machen! Ohne Kompression werden dann zwar  
auch Dateien zerstört, im Regelfall ist der Schaden aber deutlich  
geringer. DAT-Streamer sind in der Lage, die zu verarbeitenden  
Daten selbst zu komprimieren. Das geht schneller, ist aber nicht  
so effektiv wie [gzip](#).

## Beispiele

Im folgenden Beispiel archiviert **tar** alle Dateien aus dem  
Verzeichnis *Dokumente* und aus allen Unterverzeichnissen in der  
komprimierten Datei *meinarchiv.tgz*:

```
user$ tar -czf meinarchiv.tgz Dokumente
```

**tar -tzf** liefert ein Inhaltsverzeichnis des Archivs. Die Dateien  
innerhalb des Archivs sind willkürlich geordnet.

```
user$ tar -tzf meinarchiv.tgz | less
```

...

**tar -xzf** packt das Archiv aus und extrahiert alle enthaltenen  
Dateien:

```
user$ cd anderes-verzeichnis/  
user$ tar -xzf meinarchiv.tgz
```

Das folgende Kommando kopiert alle Dateien aus /verz1 nach /verz2. Der Vorteil gegenüber einem normalen [cp](#)-Kommando besteht darin, dass symbolische Links als solche kopiert werden (und nicht die Daten, auf die die Links verweisen). Die obigen zwei Kommandos eignen sich besonders dazu, um ganze Dateibäume von einer Partition auf eine andere zu übertragen.

```
root# (cd /verz1 ; tar cf - .) | (cd /verz2 ; tar xvf -)
```

Das letzte Beispiel zeigt die Erzeugung inkrementeller Backups für das Verzeichnis *Bilder*. Beim ersten Backup wird eine eventuell schon vorhandene Timestamp-Datei gelöscht, um ein vollständiges Backup durchzuführen (z.B. immer sonntags nachts). Die weiteren Backups in der Nacht der folgenden Wochentage berücksichtigen dann nur die Änderungen, die seither erfolgt sind.

```
user$ rm backup.timestamps  
user$ tar czvf backup-0.tgz -g backup.timestamps Bilder (So: vollst. Backup)  
user$ tar czvf backup-1.tgz -g backup.timestamps Bilder (Mo: nur Änderungen)  
user$ tar czvf backup-2.tgz -g backup.timestamps Bilder (Di: nur Änderungen)
```

Wenn Sie beispielsweise in einem Cron-Script sämtliche Ausgaben von tar unterdrücken möchten (auch Fehlermeldungen), ergänzen Sie das Kommando um die folgende Ein-/Ausgabeumleitung:

```
user$ tar czf backup.tgz verzeichnis >/dev/null 2>&1
```

## tasksel

tasksel installiert oder deinstalliert vordefinierte Paketgruppen. Das Kommando steht nur unter Debian- und Ubuntu-Distributionen zur Verfügung.

- `install gruppenname`  
installiert alle Pakete der angegebenen Paketgruppe.
- `--list-tasks`  
ermittelt eine Liste aller definierten Paketgruppen.
- `remove gruppenname`  
entfernt alle Pakete der angegebenen Paketgruppe. Vorsicht: Bei älteren tasksel-Versionen werden die Pakete ohne Rücksicht auf Abhängigkeiten zu anderen Paketen entfernt. Das führt dazu,

dass oft wesentlich mehr Pakete als beabsichtigt deinstalliert werden.

- `--task-packages gruppenname`  
listet alle Pakete der angegebenen Paketgruppe auf.

## Beispiel

Die beiden folgenden Kommandos installieren unter Debian bzw. Ubuntu den Apache-Webserver mit oft benötigten Erweiterungen und Zusatzprogrammen:

```
root# tasksel install web-server      (Debian)
root# tasksel install lamp-server     (Ubuntu)
```

**tcpdump [optionen] [filterausdruck]**

tcpdump aus dem gleichnamigen Paket liest den Netzwerkverkehr einer Schnittstelle mit, filtert ihn nach Kriterien, gibt ihn aus oder speichert ihn in einer Datei. Das Kommando kommt nicht nur mit TCP-Paketen, sondern auch mit UDP- und ICMP-Paketen zurecht.

tcpdump greift intern auf die pcap-Bibliothek zurück, um die Netzwerkpakete auszulesen und zu filtern.

## Optionen

- `-a`  
zeigt Paketinhalte in Textform an (ASCII).
- `-c n`  
beendet das Programm nach *n* Paketen
- `-i schnittstelle`  
berücksichtigt nur Pakete, die über die angegebene Schnittstelle fließen. Eine Liste der infrage kommenden Schnittstellen erhalten Sie mit `tcpdump -D`.
- `-n`  
zeigt IP-Adressen statt Hostnamen an.
- `-q`  
zeigt weniger Informationen an (*quiet*).
- `-r datei`  
liest die Pakete aus einer zuvor mit `tcpdump -w` gespeicherten

Datei.

- **-w datei**  
speichert die Pakete in binärer Form (*raw*) in die angegebene Datei. Die Datei kann später durch `tcpdump -r` oder von anderen Programmen (z.B. Wireshark) wieder gelesen und ausgewertet werden.
- **-x**  
zeigt Paketinhalte in hexadezimaler Form an.

## Filterausdruck

Den Optionen kann ein Filterausdruck folgen, der unter anderem aus den folgenden Schlüsselwörtern zusammengesetzt sein kann:

- **greater n**  
berücksichtigt nur Pakete, die größer als *n* Byte sind.
- **host *ipadr* oder host *hostname***  
berücksichtigt nur Pakete, die die angegebene IP-Adresse bzw. den entsprechenden Host als Quelle oder als Ziel verwenden.
- **less n**  
berücksichtigt nur Pakete, die kleiner als *n* Byte sind.
- **net *cldr***  
berücksichtigt nur Pakete, deren Quelle oder Ziel dem angegebenen Adressbereich in CIDR-Notation entspricht (z.B. `10.0.0.0/24`).
- **port *n* oder portrange *n1-n2***  
berücksichtigt nur Pakete, die die angegebenen Port-Nummern als Quelle oder Ziel verwenden.
- **proto ether|fddi|tr|wlan|ip|ip6|arp|rarp|decnet|tcp|udp**  
berücksichtigt nur Pakete des angegebenen Protokolls. Dem Ausdruck kann `ip` oder `ip6` vorangestellt werden, wenn nur IPv4- bzw. IPv6-Pakete analysiert werden sollen (z.B. `ip6 proto udp`).

Den Schlüsselwörtern `host`, `net` und `port` kann wahlweise `dst` oder `src` vorangestellt werden, wenn sich die Angabe nur auf das

Paketziel bzw. die Paketquelle bezieht.  
Mehrere Filterbedingungen verknüpfen Sie mit and oder or.  
Komplexe Ausdrücke müssen Sie mit \(` und `) klammern.  
Alternativ können Sie auch einfache Klammern verwenden, dann  
müssen Sie aber den gesamten Filterausdruck in Apostrophe stellen  
(z.B. '(port 1 or port 2)').

Zahlreiche weitere Filtermöglichkeiten beschreibt man `pcap-filter`.

## Beispiele

Das folgende Kommando gibt Informationen über alle HTTP-Pakete aus, die über die Schnittstelle `wlan0` fließen:

```
root# tcpdump -i wlan0 port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlan0, link-type EN10MB (Ethernet)
10:34:33.681218 IP imac.57402 > bpf.tcpdump.org.http:
Flags [S], seq 755525464, win 65535, options [mss 1460,nop,wscale 5,nop,nop,
TS val 595975353 ecr 0,sackOK,eol], length 0
10:34:33.681793 IP imac.57403 > bpf.tcpdump.org.http:
Flags [S], seq 2954861158, win 65535, ...
```

Das zweite Kommando zeichnet in der Datei `dump.pcap` die nächsten 100 HTTP-Pakete auf, die von der oder zu der IP-Adresse 192.139.46.66 fließen:

```
root# tcpdump -i wlan0 -n -c 100 -w dump.pcap port 80 and host 192.139.46.66
tee datei
```

`tee` dupliziert die Standardeingabe, zeigt eine Kopie an und speichert die andere in einer Datei. In der Praxis ist das dann sinnvoll, wenn die Ausgabe eines Kommandos auf dem Bildschirm beobachtet, aber gleichzeitig auch in einer Datei gespeichert werden soll. Eine einfache Umleitung mit `>` in eine Datei hätte zur Folge, dass auf dem Bildschirm nichts zu sehen ist.

## Beispiel

Das folgende Kommando zeigt das Inhaltsverzeichnis des aktuellen Verzeichnisses auf dem Bildschirm an und speichert es gleichzeitig in der Datei `inhalt`:

```
user$ ls -l | tee inhalt
telnet [optionen] host [port]
```

`telnet` erlaubt die interaktive Kommunikation mit einem Netzwerkdienst über das Telnet-Protokoll. Das Kommando eignet sich in erster Linie dazu, um elementare Netzwerkfunktionen zu testen. Da die Kommunikation unverschlüsselt erfolgt, ist `telnet` zu unsicher, um Kommandos auf externen Servern auszuführen. Für solche Aufgaben verwenden Sie besser [ssh](#)!

- `-4 bzw. -6`  
akzeptiert ausschließlich IPv4- bzw. IPv6-Adressen.
- `-l user`  
verwendet den angegebenen Benutzernamen (anstelle des aktuellen Benutzernamens).

## Beispiel

Das folgende Beispiel zeigt die Kommunikation mit einem Mail-Server, der auf dem lokalen Rechner läuft:

```
user$ telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^].
220 kofler.info ESMTP Postfix (Ubuntu)
he lo kofler.info
250 kofler.info
mail from:<kofler@kofler.info>
250 2.1.0 Ok
...
```

Oft sind Mail-Server so konfiguriert, dass nur die ersten Kommandos unverschlüsselt übertragen werden. Danach handeln Server und Client die Eckdaten der Verschlüsselung aus (STARTTLS-Verfahren). `telnet` muss an dieser Stelle passen; es eignet sich also nur dazu, um grundsätzlich zu testen, ob der Mail-Server auf Port 25 aktiv ist.

### test ausdruck

`test` wird in bash-Scripts zur Formulierung von Bedingungen verwendet und zumeist in [if](#)-Abfragen und Schleifen eingesetzt. Je nachdem, ob die Bedingung erfüllt ist, liefert es den Wahrheitswert 0 (wahr) oder 1 (falsch). Statt `test` kann auch die Kurzschreibweise [

`ausdruck` ] verwendet werden. Dabei ist es wichtig, dass Sie Leerzeichen vor und nach dem Ausdruck angeben!

Wenn Sie test oder die Kurzschreibweise [ `ausdruck` ] als Bedingung in einer Verzweigung oder Schleife verwenden, müssen Sie die Bedingung mit einem Strichpunkt abschließen, also z.B. `if [ "$1" = "abc" ] ; then ...`

`if`-Abfragen können manchmal durch die Formulierung [ `"$1" = "abc"` ] && cmd ersetzt werden. In diesem Fall ist kein Strichpunkt erforderlich. Das Kommando wird nur ausgeführt, wenn die vorherige Bedingung erfüllt war.

## Zeichenketten

[ <code>zk</code> ]	wahr, wenn die Zeichenkette nicht leer ist
[ <code>-n zk</code> ]	wahr, wenn die Zeichenkette nicht leer ist (wie [ <code>zk</code> ])
[ <code>-z zk</code> ]	wahr, wenn die Zeichenkette leer ist (0 Zeichen)
[ <code>zk1 = zk2</code> ]	wahr, wenn die Zeichenketten übereinstimmen
[ <code>zk1 != zk2</code> ]	wahr, wenn die Zeichenketten voneinander abweichen

Die Zeichenketten bzw. Variablen sollten in Hochkommata gestellt werden (z.B. [ `"$1" = "abc"` ] oder [ `"$a" = "$b"` ]). Andernfalls kann es bei Zeichenketten mit mehreren Wörtern zu Fehlern kommen!

## Zahlen

[ <code>z1 -eq z2</code> ]	wahr, wenn die Zahlen gleich sind ( <i>equal</i> )
[ <code>z1 -ne z2</code> ]	wahr, wenn die Zahlen ungleich sind ( <i>not equal</i> )
[ <code>z1 -gt z2</code> ]	wahr, wenn z1 größer z2 ist ( <i>greater than</i> )
[ <code>z1 -ge z2</code> ]	wahr, wenn z1 größer gleich z2 ist ( <i>greater equal</i> )
[ <code>z1 -lt z2</code> ]	wahr, wenn z1 kleiner z2 ist ( <i>less than</i> )
[ <code>z1 -le z2</code> ]	wahr, wenn z1 kleiner gleich z2 ist ( <i>less equal</i> )

## Dateien (auszugsweise)

[ <code>-d dat</code> ]	wahr, wenn es sich um ein Verzeichnis handelt ( <i>directory</i> )
-------------------------	--

[ -e <i>dat</i> ]	wahr, wenn die Datei existiert ( <i>exist</i> )
[ -f <i>dat</i> ]	wahr, wenn es sich um eine einfache Datei (und nicht um ein Device, ein Verzeichnis ...) handelt ( <i>file</i> )
[ -L <i>dat</i> ]	wahr, wenn es sich um einen symbolischen Link handelt
[ -r <i>dat</i> ]	wahr, wenn die Datei gelesen werden darf ( <i>read</i> )
[ -s <i>dat</i> ]	wahr, wenn die Datei mindestens 1 Byte lang ist ( <i>size</i> )
[ -w <i>dat</i> ]	wahr, wenn die Datei verändert werden darf ( <i>write</i> )
[ -x <i>dat</i> ]	wahr, wenn die Datei ausgeführt werden darf ( <i>execute</i> )
[ <i>dat1</i> -ef <i>dat2</i> ]	wahr, wenn beide Dateien denselben I-Node haben ( <i>equal file</i> )
[ <i>dat1</i> -nt <i>dat2</i> ]	wahr, wenn Datei 1 neuer als Datei 2 ist ( <i>newer than</i> )

### Verknüpfte Bedingungen

[ ! <i>bed</i> ]	wahr, wenn die Bedingung nicht erfüllt ist
[ <i>bed1</i> -a <i>bed2</i> ]	wahr, wenn beide Bedingungen erfüllt sind ( <i>and</i> )
[ <i>bed1</i> -o <i>bed2</i> ]	wahr, wenn mindestens eine der Bedingungen erfüllt ist ( <i>or</i> )

### bash-Variante [[ ausdruck ]]

Während test und die Kurzschreibweise [ ausdruck ] in dieser Form in den meisten Shells funktionieren, steht in der bash auch eine Variante mit zwei eckigen Klammern zur Auswahl. Dort sind unter anderem die folgenden zusätzlichen Vergleiche bzw. Schreibweisen erlaubt:

[[ <i>zk</i> = <i>muster*</i> ]]	wahr, wenn die Zeichenkette mit <i>muster</i> beginnt
[[ <i>zk</i> == <i>muster*</i> ]]	wie oben
[[ <i>zk</i> =~ <i>regex</i> ]]	wahr, wenn die Zeichenkette dem regulären Muster entspricht
[[ <i>bed1</i> && <i>bed2</i> ]]	wahr, wenn beide Bedingungen erfüllt sind ( <i>and</i> )

`[[ bed1 || bed2 ]]` wahr, wenn mindestens eine der Bedingungen erfüllt ist (*or*)

Ein weiterer Unterschied besteht darin, dass Variablen bei der Auswertung in `[[ ]]` nicht ausgewertet werden (keine Berücksichtigung von Joker-Zeichen wie \* und ?, korrekte Behandlung von Dateinamen mit Leerzeichen). Deswegen funktioniert `[[ -f $filename ]]` selbst dann, wenn die Variable Leer- oder Sonderzeichen enthält.

## Beispiel

Das folgende Mini-Script erfüllt keine sinnvolle Aufgabe, sondern demonstriert nur die test-Syntax in der Schreibweise `[ bedingung ] && kommando`. In der [for](#)-Schleife soll die Variable `i` die Werte von 1 bis 10 durchlaufen. `rest` enthält den Rest der ganzzahligen Division durch 2. Mit `[ $rest -eq 1 ]` testet das Script, ob der Rest 1 beträgt, ob es sich bei `i` also um eine ungerade Zahl handelt. In diesem Fall werden die folgenden Kommandos mit [continue](#) übersprungen. [echo](#) gibt den Inhalt von `i` aus. Wenn `i` den Wert 6 erreicht, wird die Schleife mit [break](#) beendet. Somit gibt das Programm die Zahlen 2, 4 und 6 aus.

```
#!/bin/bash
for i in {1..10}; do
    rest=$((i%2))
    [ $rest -eq 1 ] && continue
    echo $i
    [ $i -eq 6 ] && break
done
```

## time kommando

`time` führt das angegebene Kommando aus und gibt anschließend an, wie lange die Ausführung gedauert hat. `time` ersetzt nicht nur die Stoppuhr, sondern schlüsselt die Ausführungszeit auch in drei Komponenten auf:

- `real`  
gibt die tatsächliche Ausführungszeit aus.

- user
 

gibt die CPU-Zeit an, die während der Ausführung des Programms außerhalb des Kernels gebraucht wurde.
- sys
 

gibt die CPU-Zeit an, die während der Ausführung des Programms innerhalb des Kernels gebraucht wurde.

Die Summe aus user und sys gibt an, wie viel CPU-Aufwand die Ausführung des Kommandos verursacht hat. real ist aber oft (viel) größer als die Summe aus user und sys. Das kann zwei Gründe haben:

Einerseits laufen parallel zum durch time gestarteten Kommando in der Regel eine Menge anderer Prozesse, die natürlich auch Rechenzeit kosten. Zum anderen kann es bei der Ausführung des Kommandos zu Wartezeiten kommen, etwa beim Lesen oder Schreiben von Dateien oder bei der Übertragung von Daten im Netzwerk.

user und sys beziehen sich jeweils auf *einen* CPU-Core. Wenn das ausgeführte Kommando z.B. mehrere Cores voll auslastet, kann die Summe aus user und sys größer als real sein.

## **Beispiel**

Das folgende Kommando liest den Inhalt eines Logical Volumes aus und komprimiert ihn mit dem Kommando 7zr (Paket p7zip). Der Vorgang dauert auf einem Multi-Core-System im Leerlauf 24 Minuten. Würde nur ein Core zur Verfügung stehen, würde die Backup-Erstellung noch länger dauern (ca. 40 Minuten).

```
root# time dd if=/dev/vg830/lv3 bs=4M | 7zr a -si lv3.img.7z
real 24m53.424s
user 38m37.841s
sys   0m30.446s
```

**timedatectl [optionen] [kommando]**

Das Kommando timedatectl verändert bei Distributionen mit dem Init-System systemd die Uhrzeit und die aktive Zeitzone. Die

geänderten Einstellungen werden sofort wirksam; sie werden außerdem dauerhaft in `/etc/localtime` gespeichert.

- `list-timezones`  
liefert eine Liste aller bekannten Zeitzonen.
- `set-local-rtc 0|1`  
gibt an, ob die Uhr des Rechners die lokale Zeit oder die GMT-Zeit enthält.
- `settime datetime`  
stellt das Datum und die Uhrzeit neu ein. Dabei wird auch die Hardware-Uhr des Rechners entsprechend verändert. Die Zeitangabe muss in der Form `2020-12-31 23:59:59` erfolgen.
- `set-timezone name`  
stellt die gewünschte Zeitzone ein.
- `status`  
zeigt die Uhrzeit, die Zeitzone sowie diverse weitere Zeitdaten an, z.B. den nächsten Wechsel zwischen Sommer- und Winterzeit.

## Beispiel

Die folgenden Kommandos zeigen zuerst die aktuellen Zeitdaten und verändern dann die Zeitzone:

```
root# timedatectl status
    Local time: 2020-05-18 09:08:37 CEST
    Universal time: 2020-05-18 07:08:37 UTC
        RTC time: 2020-05-17 18:51:42
        Time zone: Europe/Vienna (CEST, +0200)
...
root# timedatectl set-timezone Europe/Paris
```

**timeout [optionen] zeitspanne kommando**

timeout führt das angegebene Kommando aus und beendet es automatisch, wenn es länger als die angegebene Zeitspanne läuft. Die Zeitspanne wird als Fließkommazahl mit dem nachgestellten Buchstaben s, m, h oder t in Sekunden, Minuten, Stunden oder Tagen angegeben (z.B. `0.5h` für eine halbe Stunde).

- `-k`  
beendet das zu lange laufende Programm durch ein `KILL`-Signal.

Standardmäßig verwendet `timeout` ein `TERM`-Signal, das aber ignoriert werden kann.

### **top [q]**

`top` zeigt alle fünf Sekunden die Liste aller laufenden Prozesse an, wobei die Prozesse nach ihrem Anteil an der Rechenzeit geordnet werden. Wenn beim Aufruf der optionale Parameter `q` angegeben wird, aktualisiert `top` die Liste ständig und beansprucht die gesamte freie Rechenzeit. `Q` beendet das Programm.

Eine komfortablere Alternative zu `top` ist das Kommando `htop`, das bei den meisten Distributionen separat installiert werden muss. Es erlaubt ein horizontales und vertikales Scrollen in der Prozessliste. Wenn Sie sich nicht für den Speicherverbrauch und die CPU-Nutzung interessieren, sondern für die IO-Aktivität oder die Energieeffizienz der laufenden Prozesse, sollten Sie einen Blick auf die Beschreibung von `iostop` oder `powertop` werfen.

### **touch [optionen] dateien**

`touch` verändert den zusammen mit der Datei gespeicherten Zeitpunkt der letzten Änderung. Wenn das Kommando ohne Optionen verwendet wird, wird als Änderungszeit die aktuelle Zeit gespeichert. Wenn die Datei noch nicht existiert, wird eine neue, 0 Byte lange Datei erzeugt.

- `-r datei`  
verwendet die gespeicherte Änderungszeit der angegebenen Datei.
- `-t zeit`  
speichert den angegebenen Zeitpunkt. Für die Zeitangabe müssen Sie das Format `[[cc]YY]MMDDhhmm[.ss]` verwenden, also z.B. als `202012311730` für den 31.12.2020 um 17:30 Uhr. Alternativ können Sie den Zeitpunkt auch als Zeichenkette mit der Option `-d` angeben. Dabei sind mehr besser lesbare Formate erlaubt, über die Sie sich mit `info touch` kundig machen können.

## Beispiel

In diesem Beispiel wird touch dazu verwendet, die Änderungszeiten aller Dateien eines Verzeichnisses mit den Dateien eines zweiten Verzeichnisses abzulegen. Das ist z.B. dann sinnvoll, wenn Sie bei einem größeren [cp](#)-Kommando die Option -a vergessen haben und die kopierten Dateien nun alle das aktuelle Datum aufweisen.

```
user$ cd zielverzeichnis  
user$ find . -type f -exec touch -r /quellverzeichnis/{} {} \\;  
tr [optionen] zk1 [zk2] [ < quelle > ziel]
```

tr ersetzt in der angegebenen Quelldatei alle Zeichen der Zeichenkette 1 durch die entsprechenden Zeichen der Zeichenkette 2. Die beiden Zeichenketten sollten gleich lang sein. Zeichen, die in der ersten Zeichenkette nicht vorkommen, bleiben unverändert. Es ist nicht möglich, ein einzelnes Zeichen durch mehrere Zeichen zu ersetzen (etwa ö durch "o) – dazu müssen Sie Kommandos wie [recode](#) oder [sed](#) verwenden.

- -d  
löscht die in Zeichenkette 1 angegebenen Zeichen. Zeichenkette 2 braucht nicht angegeben zu werden.

## Beispiel

Im folgenden Beispiel ersetzt tr alle Klein- durch Großbuchstaben und zeigt das Ergebnis im Terminal (an der Standardausgabe) an:

```
user$ tr a-zäöü A-ZÄÖÜ < textdatei  
traceroute[6] [optionen] zieladresse
```

traceroute liefert eine Liste aller Stationen, die auf dem Weg eines TCP/IP-Pakets vom lokalen Rechner zur Zieladresse liegen. Zu jeder Station wird die Gesamtlaufzeit angegeben (standardmäßig für drei Versuche). traceroute funktioniert nur, wenn der UDP-Port 33434 auf dem Weg zur Zieladresse nicht durch eine Firewall blockiert wird. In diesem Fall liefert traceroute statt der Angaben zur Zwischenstation nur drei Sterne.

- -4 bzw. -6  
nutzt ausschließlich das Protokoll IPv4 bzw. IPv6. Standardmäßig

verwendet traceroute automatisch das geeignete Protokoll. Wenn ein Hostname sowohl mit einer IPv4- als auch mit einer IPv6-Adresse verbunden ist, zieht traceroute IPv4 vor. Für traceroute6 gilt automatisch -6.

- **-m n**  
bestimmt die maximale Anzahl von Zwischenstationen (standardmäßig 30).
- **-n**  
zeigt bei den Zwischenstationen nur die IP-Nummer an (nicht den Hostnamen).
- **-p n**  
verwendet den angegebenen UDP-Port (standardmäßig 33434).
- **-q n**  
sendet *n* Pakete und misst für jedes Paket die Antwortzeiten (standardmäßig 3).

## Beispiel

Das folgende Kommando zeigt die Stationen zwischen meinem lokalen Rechner und meinem Webserver. Bei einigen Stationen werden die IP-Pakete über unterschiedliche Router geleitet, was auf eine redundante Netzwerkanbindung hindeutet.

```
root# traceroute kofler.info
traceroute to kofler.info (213.239.211.2), 30 hops max, 60 byte packets
 1 dsldevice.lan (10.0.0.138)  14.116 ms  13.376 ms  13.112 ms
 2 62.47.95.239 (62.47.95.239)  25.650 ms  27.489 ms  29.245 ms
...
 9 hos-tr3.ex3k16.rz6.hetzner.de (213.239.252.8)  48.315 ms
  hos-tr4.ex3k16.rz6.hetzner.de (213.239.252.136)  49.034 ms
  hos-tr2.ex3k16.rz6.hetzner.de (213.239.229.136)  51.231 ms
10 kofler.info (213.239.211.2)  52.069 ms  54.640 ms  55.757 ms
```

```
trap [kommando] signal
```

Das bash-Kommando trap führt den angegebenen Befehl aus, wenn die Shell das angegebene Signal empfängt. Wenn kein Kommando angegeben wird, ignoriert das Programm bzw. die bash das betreffende Signal. trap -l liefert eine Liste aller möglichen Signale

und der ihnen zugeordneten Kennnummern. `trap` wird üblicherweise zur Fehlerabsicherung in Shell-Scripts eingesetzt.

## Beispiel

Das folgende Mini-Script läuft in einer Endlosschleife, die sich nicht durch `Strg+C` unterbrechen lässt. Sie können das Script aber z.B. durch [kill](#) beenden.

```
#!/bin/bash
trap 'echo "Strg+C wird ignoriert!"' SIGINT
while true
do
    sleep 1
done
```

**tree [optionen] [startverzeichnis]**

`tree` aus dem gleichnamigen Paket stellt die Hierarchie des Verzeichnisbaums ausgehend vom aktuellen bzw. vom angegebenen Startverzeichnis dar.

- `-a`  
gibt auch Dateien aus anstatt standardmäßig nur Verzeichnisse.
- `-L n`  
berücksichtigt nur die ersten  $n$  Hierarchieebenen.
- `-x`  
bleibt im aktuellen Dateisystem, berücksichtigt keine [mount](#)-Verzeichnisse oder Links in andere Dateisysteme.

**truncate [optionen] dateien**

`truncate` verkleinert oder vergrößert Dateien. Beim Verkleinern wird die Datei einfach abgeschnitten. Vorsicht: Dabei gehen Daten verloren! Beim Vergrößern werden der Datei entsprechend viele 0-Bytes hinzugefügt. Bei den meisten Dateisystemen ändert sich dadurch der Platzbedarf auf dem Datenträger nicht, weil die Datei als *Sparse File* erkannt wird. Wenn Sie eine Datei physikalisch mit Nullen füllen möchten, müssen Sie das Kommando [dd](#) einsetzen. Wenn an `truncate` der Name einer noch nicht vorhandenen Datei übergeben wird, erzeugt das Kommando die Datei in der gewünschten Größe.

- **-s n**  
gibt die gewünschte Größe in Byte an. Wenn der Zahl die Buchstaben K, M, G oder T folgen, multipliziert truncate die Zahl mit 1024, 1.048.576 etc. Alternativ stehen kB, MB, GB oder TB für 1000, 1.000.000 etc. 10K entspricht also 10.240 Byte, 10kB entspricht 10.000 Byte.

## Beispiel

Im folgenden Beispiel wird [file](#) zuerst mit 1 MiB Zufallsdaten gefüllt. Danach wird die Datei mit truncate auf 2 MiB vergrößert. ls -l zeigt tatsächlich eine Größe von 2 MiB, aber [dd](#) beweist, dass die Datei nur 1 MiB Platz auf der Festplatte oder SSD beansprucht. Der Platzbedarf steigt erst, wenn in die Datei später andere Daten als Nullen geschrieben werden.

```
user$ dd if=/dev/urandom of=file bs=1024 count=1024
user$ truncate -s 2M file
user$ ls -l file
-rw-r--r-- 1 kofler users 2097152 19. Nov 14:30 file
user$ du -h file
1,0M    file
```

## tty

tty zeigt den Device-Namen des aktiven Terminals an (für die Textkonsolen: /dev/tty1 bis tty6, für Shell-Fenster unter X /dev/pts/n).

## tune2fs [optionen] device

Mit tune2fs können Sie diverse Systemparameter eines ext2-/ext3-/ext4-Dateisystems verändern.

- **-c n**  
gibt an, nach wie vielen [mount](#)-Vorgängen die Partition beim Booten auf Fehler kontrolliert werden soll. 0 bedeutet, dass nie eine Überprüfung erfolgen soll.
- **-i n**  
gibt an, wie oft (in Tagen) die Partition beim Booten auf Fehler kontrolliert werden soll. 0 bedeutet, dass nie eine Überprüfung erfolgen soll.

- **-l**  
zeigt Informationen zum angegebenen Dateisystem an, ändert aber nichts.
- **-m n**  
gibt an, wie viel Prozent des Datenträgers für Daten von root reserviert werden sollen (bei `mke2fs` üblicherweise 5 Prozent).
- **-U neue-uuid** bzw. **-U random**  
gibt dem Dateisystem eine neue UUID.

## Beispiel

Das folgende Kommando bewirkt, dass das Dateisystem in der Partition `/dev/sda1` nur noch einmal jährlich bzw. nach 200 [mount](#)-Vorgängen überprüft wird:

```
root# tune2fs -i 365 -c 200 /dev/sda1
Setze die maximale Mount-Anzahl auf 200
Setze das Intervall zwischen Checks auf 31536000 Sekunden
```

## **tvservice [optionen]**

Unter Raspberry Pi OS können Sie mit dem Kommando `tvservice` Informationen über den HDMI-Status des Raspberry Pi ermitteln und das HDMI-Signal aus- und wieder einschalten.

- **-e "group mode drive"** bzw. **--explicit "group mode drive"**  
schaltet den HDMI-Ausgang ein und aktiviert den angegebenen Modus. Anstelle von `group` geben Sie `CEA` oder `DMT` an, anstelle von `drive` das Schlüsselwort `HDMI` oder `DVI`. Die Liste der zur Auswahl stehenden Modi liefert `tvservice --modes CEA|DMT`.
- **-m CEA|DMT** bzw. **--modes=CEA|DMT**  
listet die vom Grafiksystem unterstützten Modi gemäß den Spezifikationen der *Consumer Electronics Association* (CEA) bzw. aus der Liste der von der VESA definierten *Display Monitor Timings* (DMT) auf. CEA-Modi sind auf TV-Geräten vorzuziehen, DMT-Modi auf Computer-Monitoren.
- **-o** bzw. **--off**  
schaltet den HDMI-Ausgang aus. Die meisten HDMI-Monitore

aktivieren nun einen Energiesparmodus.

- **-p bzw. --preferred**  
aktiviert den HDMI-Ausgang und verwendet HDMI-Defaulteinstellungen. Leider funktioniert das Wiedereinschalten des Ausgangs auf dem Raspberry Pi nur bedingt: Der Monitor schaltet sich zwar wieder ein, das Bild bleibt aber schwarz. Abhilfe schafft nun ein zweimaliger Wechsel der aktiven Konsole mit dem Kommando [chvt](#). Dabei müssen Sie zuerst eine Textkonsole und dann wieder die Grafikkonsole aktivieren (`chvt 1 && chvt 7`). Einfacher gelingt das Ein- und Ausschalten des HDMI-Ausgangs mit [vcgencmd display\\_power 0/1](#).
- **-s bzw. --status**  
zeigt den aktuellen HDMI-Status an (HDMI-Modus, Auflösung etc.).

### **type** kommando

Das bash-Kommando `type` ermittelt, ob es sich beim angegebenen Kommando um ein Shell-Kommando (beispielsweise [cd](#)), eine selbst definierte Funktion oder um eine [alias](#)-Abkürzung handelt. Die Meldung *command is hashed* bedeutet, dass es sich um ein in dieser Sitzung bereits ausgeführtes Linux-Kommando handelt, dessen Pfadnamen sich die bash in einem Hash-Verzeichnis gemerkt hat.

### **Beispiel**

[cd](#) ist ein in die Shell integriertes Kommando:

```
root# type cd  
cd is a shell builtin
```

## U

### **ubuntu-security-status [optionen]**

ubuntu-security-status liefert einen Überblick, wie viele Pakete über welchen Zeitraum gewartet werden. Diese Information ist deswegen wichtig, weil die fünfjährige Update-Garantie für Ubuntu-LTS-Versionen nur für die Paketquellen *main* und *restricted* gilt. Eine typische Ubuntu-Installation enthält einen Mix aus Paketen verschiedener Paketquellen.

Wenn das Kommando ohne Optionen aufgerufen wird, gibt es einen Überblick ohne allzu viele Details.

- **--third-party**  
listet Pakete auf, die aus Ubuntu-fremden Paketquellen stammen.
- **--unavailable**  
listet Pakete auf, zu denen es keine aktiven Paketquellen gibt.

### **Beispiele**

Die folgenden Ausgaben sind auf einer Desktop-Installation von Ubuntu 20.04 entstanden:

```
user$ ubuntu-security-status
3263 packages installed, of which:
2024 receive package updates with LTS until 4/2025
1225 could receive security updates with ESM Apps until 4/2030
  4 packages are from third parties
  10 packages are no longer available for download
...
user$ ubuntu-security-status --thirdparty
...
4 packages are from third parties:
  code google-chrome-stable spotify-client virtualbox-6.1
```

Im Gegensatz zu dem auf älteren Ubuntu-Versionen verfügbaren Kommando `ubuntu-support-status` ist `ubuntu-security-status` nicht in der Lage, alle Pakete aufzulisten, die nicht aus den Quellen *main* und *restricted* stammen. `ubuntu-security-status` verrät nur deren Anzahl, was unbefriedigend ist.

Abhilfe schafft das Kommando [aptitude](#) für die Paketquellen *universe* und *multiverse*:

```
user$ aptitude search -F "%s# %p" "~i ?section(universe)"
...
user$ aptitude search -F "%s# %p" "~i ?section(multiverse)"
multiverse/web          flashplugin-installer
multiverse/gnome         nautilus-dropbox
multiverse/x11           ttf-mscorefonts-installer
multiverse/metapackages  ubuntu-restricted-addons
multiverse/metapackages  ubuntu-restricted-extras
multiverse/utils          unrar
```

### **ufw [optionen] kommando**

`ufw` ist ein Ubuntu-spezifisches Kommando zur Firewall-Konfiguration. `ufw` ist unter Ubuntu und seinen Derivaten zwar standardmäßig installiert, aber nicht aktiviert. Wenn Sie `ufw` zur Firewall-Konfiguration verwenden möchten, müssen Sie die Firewall zuerst mit `ufw enable` aktivieren.

Zur Konfiguration der Firewall stehen unter anderem die folgenden Subkommandos zur Auswahl. Sie können mit der Option `--dry-run` kombiniert werden. In diesem Fall werden keine Änderungen durchgeführt. Die Option ist praktisch, um die Syntax komplexer Kommandos zu testen.

- `allow|deny [on interface] port/protocol/"app name"`  
erlaubt bzw. blockiert einen Port (z.B. 22), ein Protokoll (z.B. [ssh](#)) oder ein Programm, für das es eigene Regeldateien gibt (siehe `ufw app list`).  
Die Regel kann optional nur für eine bestimmte Netzwerkschnittstelle formuliert werden, also z.B. `on eth0`. `man ufw` zählt diverse weitere Optionen auf, um Regeln zu formulieren, die nur für eintreffende oder ausgehende Pakete bzw. nur für bestimmte Protokolle oder IP-Versionen gelten (z.B. für `tcp`, `udp`, `ipv6`).
- `app info "app name"`  
liefert detaillierte Informationen zu einer `ufw`-Regeldatei.

- **app list**  
liefert eine Liste aller Programme, für die es ufw-Regeldateien im Verzeichnis `/etc/ufw/applications.d` gibt.
- **default allow|deny|reject [incoming|outgoing|routed]**  
bestimmt das Defaultverhalten der Firewall. Ohne die Angabe von `incoming|outgoing|routed` wird das Verhalten des Eingangsfilters verändert. Standardmäßig gilt – sofern die Firewall überhaupt aktiviert ist – für den `incoming`-Filter das Verhalten `deny`, für den `outgoing`-Filter das Verhalten `allow`.
- **disable**  
deaktiviert die Firewall. Der gesamte Netzwerkverkehr kann ungehindert fließen.
- **enable**  
aktiviert die Firewall.
- **reload/reset**  
lädt alle Firewall-Regeln neu bzw. führt einen kompletten Neustart der Firewall durch.
- **status [verbose]**  
zeigt den Zustand der Firewall an.

### Beispiel

Die folgenden Kommandos erlauben zuerst den Dienst `ssh` und aktivieren dann die ufw-Firewall. Eine vorhandene SSH-Verbindung sollte dadurch nicht unterbrochen werden. Alle anderen Server-Dienste werden aber von nun an blockiert.

```
root# ufw allow ssh
root# ufw enable
root# ufw status verbose
Status: Aktiv
Protokollierung: on (low)
Voreinstellung: deny (eingehend),
                allow (abgehend),
                disabled (gesendet)
Neue Profile: skip
Zu          Aktion      Von
-
22          ALLOW IN   Anywhere
22 (v6)     ALLOW IN   Anywhere (v6)
ulimit option grenzwert
```

`ulimit` begrenzt die Systemressourcen, die von der Shell und den aus ihr gestarteten Prozessen in Anspruch genommen werden.

Größenangaben erfolgen generell in KiB. `ulimit` wird oft in [`/etc/profile`](#) voreingestellt.

- **-c speicher**  
beschränkt die Größe von Coredumps, also des Speicherabilds, das bei einem Programmabsturz automatisch auf der Festplatte gespeichert wird.
- **-d speicher**  
beschränkt den Speicher für das Datensegment von Prozessen.
- **-f dateigröße**  
verhindert die Erzeugung von Dateien, die größer als der angegebene Grenzwert sind. Funktioniert nicht mit allen Dateisystemen.
- **-s speicher**  
beschränkt den Stack-Speicher.

**umask [maske]**

Das bash-Kommando `umask` steuert, mit welchen Zugriffsrechten neue Dateien angelegt werden. `umask` merkt sich hierfür einen oktalen Zahlenwert, der von den Defaultzugriffsrechten neuer Dateien bzw. Verzeichnisse abgezogen wird.

Linux sieht eigentlich vor, dass neue Dateien die Zugriffsbits `rw-rw-rw` (oktal 666) bekommen. Neue Verzeichnisse und Programmdateien, die von einem Compiler erzeugt werden, bekommen automatisch die Zugriffsbits `rwxrwxrwx` (777). Für die praktische Arbeit sind diese Grundeinstellungen zu

freifügig. Deswegen sehen alle Linux-Shells die sogenannte umask-Einstellung vor. Dabei handelt es sich um einen Zahlenwert, der die Zugriffsbits angibt, die von den Standardzugriffsbits *abgezogen* werden. Die umask-Einstellung ist systemweit vorgegeben und wird je nach Distribution in [/etc/profile](#) oder in [bashrc](#) eingestellt. Oft gelten für `root` und für alle anderen Benutzer unterschiedliche Werte. Bei Ubuntu ist der umask-Wert durch das PAM-System festgelegt; der Defaultwert ist in [/etc/login.defs](#) gespeichert.

### Beispiele

`umask` ohne weitere Parameter zeigt die aktuelle Einstellung an:

```
user$ umask (Debian, openSUSE, Ubuntu)  
0022  
user$ umask (Fedora, RHEL)  
0002
```

Das zweite Beispiel zeigt, welche Auswirkungen es auf neue Dateien und Verzeichnisse hat, wenn Sie den umask-Wert auf 027 ändern. Neue Dateien erhalten so die Zugriffsrechte  $666 - 026 = 640 = \text{rw-r}---$   $--$ , neue Verzeichnisse  $777 - 027 = 750 = \text{rwxr-x}---$ .

```
user$ umask 27  
user$ touch neue-datei  
user$ mkdir neues-verzeichnis  
user$ ls -ld neu*  
-rw-r----- ... user user ... neue-datei  
drwxr-x--- ... user user ... neues-verzeichnis  
umount device  
umount verzeichnis
```

`umount` entfernt ein Dateisystem aus dem Linux-Verzeichnisbaum. Die Angabe des Dateisystems erfolgt entweder durch den Device-Namen des Datenträgers oder durch die Angabe des Verzeichnisses, bei dem das Dateisystem in den Verzeichnisbaum eingebunden ist. Das Kommando kann nur von `root` ausgeführt werden. Es führt zu einer Fehlermeldung, wenn es auf dem Dateisystem noch geöffnete Dateien gibt.

- `-f`

erzwingt den `umount`-Vorgang (zweckmäßig bei nicht mehr erreichbaren NFS-Verzeichnissen).

### `unalias` abkürzung

`unalias` löscht die angegebene Abkürzung. Wenn das Kommando mit der Option `-a` aufgerufen wird, löscht es alle bekannten Abkürzungen. Der Umgang mit Abkürzungen wird bei der Beschreibung des [alias](#)-Kommandos erläutert.

### `uname` [optionen]

`uname` zeigt den Namen des Betriebssystems an (also Linux). Durch die Angabe von Optionen können auch andere Informationen angezeigt werden.

- `-a`

zeigt alle verfügbaren Informationen an, nämlich das Betriebssystem, die Versionsnummer, Datum und Uhrzeit sowie den Prozessor.

- `-m`

liefert nur die CPU-Plattform (z.B. `i686` oder `x86_64` bei 64-Bit-Systemen).

- `-r`

liefert nur die Kernelversion.

### Beispiel

Auf diesem Rechner läuft ein 64-Bit-Kernel, der im April 2020 kompiliert wurde:

```
root# uname -a  
Linux fedora32 5.6.7-300.fc32.x86_64 #1 SMP  
Thu Apr 23 14:13:50 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
```

### `uncompress` datei

`uncompress` dekomprimiert eine mit [compress](#) komprimierte Datei.

Dabei wird die Dateikennung `.Z` automatisch entfernt. `uncompress` ist ein Link auf [compress](#), wobei automatisch die Option `-d` aktiviert wird.

```
unexpand textdatei > ergebnis
```

unexpand ersetzt mehrfache Leerzeichen in der angegebenen Textdatei durch Tabulatorzeichen und schreibt das Ergebnis in die Standardausgabe. Das Kommando ist das Gegenstück zu [expand](#), das Tabulatorzeichen durch Leerzeichen ersetzt.

- **-a**  
ersetzt alle Leerzeichen, nicht nur die am Beginn einer Zeile.
- **-t n**  
gibt die Anzahl der Zeichen je Tabulatorposition an (standardmäßig 8).

```
uniq [optionen] datei
```

uniq gibt die Zeilen einer Textdatei auf der Standardausgabe aus, wobei unmittelbar aufeinanderfolgende gleichlautende Zeilen eliminiert werden. Bei vorsortierten Dateien eliminiert uniq alle mehrfach auftretenden Zeilen.

- **-c**  
gibt bei jeder Zeile an, wie oft sie gefunden wurde.
- **-d**  
gibt nur die Doppelgänger aus (nicht aber Zeilen, die nur einmal vorkommen).
- **-u**  
gibt nur Zeilen aus, die *keine* Doppelgänger haben (also *unique* sind).

## Beispiel

[sort](#) sortiert die Datei *test*, und uniq eliminiert doppelte Zeilen und speichert das Resultat in *test1*:

```
user$ sort test | uniq > test1  
unset variable
```

Das bash-Kommando **unset** löscht die angegebene Variable.

```
until bedingung; do  
    kommandos  
done
```

until bildet Schleifen in bash-Skripts. Die Schleife wird so lange ausgeführt, wie die angegebene Bedingung erfüllt ist. Das Schleifenkriterium ist der Rückgabewert des Kommandos, das als Bedingung angegeben wird. Vergleiche und Tests werden mit dem Kommando [test](#) oder dessen Kurzform in eckigen Klammern durchgeführt.

## Beispiel

Das Beispiel zeigt die Formulierung einer simplen Schleife mit until:

```
user$ i=1; until [ $i -gt 3 ]; do echo $i; i=$[$i+1]; done  
1  
2  
3
```

```
unxz [optionen] dateien
```

unxz aus dem Paket xz-utils dekomprimiert die zuvor mit [xz](#) komprimierten Dateien. Dabei wird die Dateiendung .xz entfernt.

```
unzip [optionen] archiv.zip [datei1 datei2]
```

Das Kommando extrahiert Dateien aus einem vor allem in der Windows-Welt sehr verbreiteten ZIP-Archiv. Wenn die zu extrahierenden Dateien nicht explizit aufgeführt werden, packt **unzip** alle Dateien des Archivs aus. Wenn Sie unter Linux selbst ZIP-Archive bilden möchten, verwenden Sie hierfür [zip](#).

## Beispiel

Das folgende Kommando extrahiert die Datei *druck.pdf* aus *archiv.zip*. Sollte es in diesem Archiv noch weitere Dateien geben, werden diese nicht angerührt.

```
root# unzip archiv.zip druck.pdf  
Archive: ./archiv.zip  
  inflating: druck.pdf  
update-alternatives [optionen] kommando
```

`update-alternatives` verwaltet die Links im Verzeichnis `/etc/alternatives`. Diese Links bestimmen die aktive Version mehrerer parallel installierter Programme mit derselben Funktion (z.B. Editoren oder Java-Umgebungen). Das Kommando wird in der Regel von den (De-)Installations-Scripts der betreffenden Pakete ausgeführt, kann aber natürlich auch interaktiv verwendet werden. Unter Red Hat bzw. Fedora ist das Kommando auch unter dem Namen [alternatives](#) verfügbar.

- `--auto name`  
aktiviert den Automatikmodus für das Kommando. Damit ist automatisch das Kommando aktiv, das den höchsten Prioritätswert hat.
- `--config name`  
zeigt die zur Auswahl stehenden Alternativen zum angegebenen Kommando an. Anschließend geben Sie interaktiv an, welche Alternative in Zukunft aktiv sein soll. `update-alternatives` passt die Links anschließend entsprechend an und wechselt für das Kommando in den manuellen Modus.
- `--display name`  
liefert eine Liste aller installierten Alternativen zum angegebenen Kommando.
- `--install ... bzw. --remove ...`  
installiert bzw. entfernt eine Alternative für ein Kommando. Diese Kommandos kommen meist nur im (De-)Installations-Script eines Pakets zur Anwendung.
- `--set name kommandopfad`  
richtet das mit `kommandopfad` angegebene Kommando als Standardprogramm für `name` ein.

### Beispiel

Das folgende Kommando definiert `/usr/bin/jmacs` als Standardeditor:

```
root# update-alternatives --set editor /usr/bin/jmacs  
update-grub
```

Dieses Script steht nur unter Ubuntu und Debian zur Verfügung. Es führt die Konfigurations-Scripts `/etc/grub.d/*` aus und erstellt die neue GRUB-Menüdatei `/boot/grub/grub.cfg`. Auf anderen Distributionen führen Sie statt `update-grub` das Kommando `grub2-mkconfig -o /boot/grub2/grub.cfg` aus.

### `update-initramfs [optionen]`

`update-initramfs` ist bei Debian und Ubuntu für das Erzeugen, Aktualisieren oder Löschen von Initrd-Dateien zuständig. Die Initrd-Dateien enthalten Kernelmodule, die während des Systemstarts durch GRUB geladen werden.

Bei Fedora, Red Hat und SUSE verwenden Sie statt `update-initramfs` das Kommando [dracut](#).

Intern greift `update-initramfs` auf das Script `mkinitramfs` zurück, dessen direkte Verwendung aber nicht empfohlen wird. Die Konfiguration der Initrd-Dateien erfolgt durch die Dateien des Verzeichnisses `/etc/initramfs-tools`.

- `-c`  
erzeugt eine neue Initrd-Datei für die mit `-k` angegebene Kernelversion.
- `-d`  
löscht die Initrd-Datei für die mit `-k` angegebene Kernelversion.
- `-k versionsname`  
gibt die zu bearbeitende Kernelversion an. `-k all` bewirkt, dass `update-initramfs` die Initrd-Dateien aller installierten Kernelversionen bearbeitet.
- `-u`  
aktualisiert die Initrd-Datei der aktuellsten Kernelversion bzw. der mit `-k` angegebenen Kernelversion.

### Beispiel

Das folgende Kommando erzeugt eine neue Initrd-Datei für die Kernelversion 3.12.3. Die Kerneldatei hat den Dateinamen `/boot/vmlinuz-5.6.7-generic`, und die resultierende Initrd-Datei heißt

```
/boot/initrd.img-5.6.7-generic.  
root# update-initramfs -c -k 5.6.7-generic  
updatedb
```

updatedb erstellt ein Indexverzeichnis für das [locate](#)-Kommando. Der Index enthält eine Liste aller im gesamten Dateisystem enthaltenen Dateien. Das Kommando wird üblicherweise einmal täglich automatisch durch einen Cron-Job ausgeführt. Die Ausführung erfordert root-Rechte. Je nach Distribution wird die Dateidatenbank im Verzeichnis `/var/lib/`, `/var/lib/slocate` oder `/var/lib/mlocate` gespeichert.

#### **uptime [optionen]**

Ohne weitere Optionen gibt `uptime` an, wie lange der Rechner schon läuft, wie viele Personen gerade eingeloggt sind und wie die durchschnittliche Auslastung in der letzten Minute, in den letzten fünf Minuten bzw. in den letzten 15 Minuten war (*load average*).

- `-p`  
gibt den Uptime-Zeitraum in gut lesbarer Form aus (*pretty*).
- `-s`  
verrät, wann der Rechner hochgefahren wurde (*since*).

#### **Beispiel**

Der Server, auf dem das folgende Kommando ausgeführt wurde, läuft schon seit über drei Monaten. Derart lange Uptimes sind nur vernünftig, wenn es in dieser Zeit keine sicherheitsrelevanten Kernel-Patches gab bzw. wenn diese Patches mit einem Live-Patch-System in den Kernel integriert wurden. Auf Ubuntu-Servern können Sie das mit [canonical-livepatch](#) feststellen.

```
user$ uptime  
15:48:11 up 104 days, 23 min,  1 user,  load average: 0.15, 0.12, 0.05
```

#### **useradd [optionen] name**

useradd richtet einen neuen Benutzer ein.

- `-b "basedir"`  
legt die Basis für die Benutzerverzeichnisse fest (standardmäßig `/home`).
- `-c "vollständiger Name"`  
gibt den vollständigen Namen des neuen Benutzers an.
- `-g gruppe`  
gibt die Hauptgruppe (primäre Gruppe) des Benutzers an.
- `-G gruppeA,gruppeB,gruppeC`  
bestimmt alle Zusatzgruppen (Supplementary Groups) des Benutzers.
- `-m`  
falls noch kein Heimatverzeichnis existiert (`/home/name`), wird es erzeugt. Alle Dateien aus `/etc/skel` werden dorthin kopiert. Normalerweise gilt `-m` aufgrund der Voreinstellungen in [/etc/login.defs](#) standardmäßig und muss nicht angegeben werden. In solchen Fällen kann die Erzeugung des Heimatverzeichnisses durch `-M` verhindert werden.
- `-u n`  
weist dem Benutzer die angegebene UID-Nummer zu (User Identification), sofern diese noch verfügbar ist.

#### **Beispiel**

Die folgenden Kommandos richten den neuen Benutzer Gerald Gersin mit dem Login-Namen `gersin` ein, definieren dann ein initiales Passwort und zwingen den Benutzer, sofort beim ersten Login und in der Folge alle 100 Tage ein neues Passwort einzustellen:

```
root# useradd gersin -c "Gerald Gersin"  
root# passwd gersin  
Geben Sie ein neues Passwort ein: *****  
Geben Sie das neue Passwort erneut ein: *****  
root# chage -d 0 -M 100 gersin
```

```
userdel name
```

**userdel** löscht den angegebenen Benutzer-Account.

- -r

löscht auch das gesamte Heimatverzeichnis sowie die Mail-Inbox des Benutzers.

```
usermod [optionen] name
```

**usermod** verändert diverse Eigenschaften des Benutzer-Accounts, z.B. das Heimatverzeichnis, die Gruppenzugehörigkeit, die Standard-Shell oder die UID. Die meisten Optionen sind mit denen von [useradd](#) identisch. Änderungen am Benutzer-Account werden erst nach einem neuerlichen Login wirksam.

- -a -G *gruppe*

fügt den Benutzer der angegebenen Gruppe hinzu.

- -L

blockiert den Account vorübergehend. Dazu wird in */etc/shadow* vor dem Hashcode des Passworts das Zeichen »!« gestellt, weswegen ein Login nicht mehr möglich ist.

- -U

gibt einen durch -L blockierten Account wieder frei.

### **Beispiel**

Das folgende Kommando fügt den Benutzer `gersin` der Gruppe `docuteam` hinzu:

```
root# usermod -a -G docuteam gersin
```

Wenn Sie den Benutzer später wieder aus der Gruppe entfernen möchten, führen Sie [gpasswd](#) -d `gersin` `docuteam` aus.

## V

### **vcgencmd** kommando

Mit dem spärlich dokumentierten Kommando `vcgencmd` können Sie unter Raspberry Pi OS Eckdaten des Geräts bzw. seiner CPU auslesen und einige wenige Parameter auch verändern.

- `codec_enabled H264|MPG2|WVC1|MPG4|MJPEG|WMV9`  
gibt an, ob der betreffende Hardware-Codec genutzt werden kann. Die MPEG2- und VC1-Codecs erfordern einen Lizenzschlüssel, der in [`/boot/config.txt`](#) eingetragen werden muss.
- `commands`  
listet alle `vcgencmd`-Kommandos auf.
- `display_power 0|1`  
schaltet den HDMI-Ausgang aus bzw. wieder ein.
- `get_config parametername|int|str`  
liefert den Zustand des angegebenen Parameters bzw. aller Integer- oder String-Parameter. Die Parameter werden in der Datei `/boot/config.txt` eingestellt.
- `measure_clock arm|core|hdmi|uart`  
liefert die CPU-Frequenz, die Frequenz der Grafik-Cores sowie die Frequenz diverser weiterer Komponenten.
- `measure_temp`  
liefert die Temperatur der CPU.
- `measure_volts core|sdram_c|sdram_i|sdram_p`  
gibt an, mit welcher Spannung die Grafik-Cores und der Speicher versorgt werden.
- `version`  
liefert die Firmware-Version.

### **Beispiel**

Die folgenden Kommandos zeigen den Betriebszustand eines Raspberry Pi, der gerade im Leerlauf ist:

```
pi$ vcgencmd measure_clock arm  
frequency(45)=600000000  
pi$ vcgencmd measure_temp  
temp=44.4'C
```

### **vgchange [optionen] [vgname]**

Das LVM-Kommando vgchange verändert die Attribute einer Volume Group (VG). Die wichtigste Anwendung besteht darin, VGs zu aktivieren bzw. zu deaktivieren.

- -a y|n  
aktiviert (y) bzw. deaktiviert (n) alle VGs bzw. die angegebene VG.

### **vgcreate [optionen] vgname pvname1 [pvname2 ...]**

vgcreate erzeugt eine neue VG aus einem oder mehreren Physical Volumes (PVs).

## **Beispiel**

Im folgenden Beispiel markiert zuerst [pvcreate](#) die jeweils erste Partition der Festplatten oder SSDs `/dev/sdb1` und `/dev/sdc1` als Physical Volumes. vgcreate bildet daraus die Volume Group `vg1`. Der Befehl [lvcreate](#) reserviert nun 100 GiB für das Logical Volume `lv1`. `mkfs.ext4` richtet darin ein Dateisystem ein, und [mount](#) bindet dieses am gerade erzeugten Verzeichnis `/mnt/lv1` in den Verzeichnisbaum ein.

```
root# pvcreate /dev/sdb1  
root# pvcreate /dev/sdc1  
root# vgcreate vg1 /dev/sdb1 /dev/sdc1  
root# lvcreate -L 100G -n lv1 vg1  
root# mkfs.ext4 /dev/mapper/vg1-lv1  
root# mkdir /mnt/lv1  
root# mount /dev/mapper/vg1-lv1 /mnt/lv1
```

### **vgdisplay vgname**

vgdisplay zeigt Detailinformationen zur angegebenen VG an.

### **vgextend vgname pvname**

vgextend fügt ein PV zu einer VG hinzu.

## **Beispiel**

Die folgenden Kommandos definieren die Partition `/dev/sdc2` als neues Physical Volume und fügen dieses dann der vorhandenen, zu klein gewordenen Volume Group `myvg1` hinzu:

```
root# pvcreate /dev/sdc2
      Physical volume "/dev/sdc2" successfully created
root# vgextend myvg1 /dev/sdc2
      Volume group "myvg1" successfully extended
root# vgdisplay myvg1
...
VG Size          180,64 GB
Alloc PE / Size 6402 / 22,50 GB
Free  PE / Size 41324 / 158,14 GB
...
```

### **vgmerge** vgname1 vgname2

**vgmerge** fügt VG2 zu VG1 hinzu. Nach der erfolgreichen Ausführung des Kommandos gibt es also nur noch VG1. Sie besteht aus allen PVs, die bisher den Speicherpool für VG1 und VG2 bildeten.

### **vgreduce** [optionen] vgname [pvname1 pvname2 ...]

**vgreduce** entfernt die angegebenen PVs aus dem Speicherpool der VG. Das funktioniert nur, wenn die PVs ungenutzt sind. Mit der Option -a werden alle inaktiven PVs entfernt.

### **vgrename** oldvgname newvgname

**vgrename** gibt einer VG einen neuen Namen. Statt `oldvgname` kann auch die UUID der VG angegeben werden.

### **vgscan**

**vgscan** listet alle Volume Groups auf.

### **vipw** [optionen] **vigr** [optionen] **visudo** [optionen]

Die drei Kommandos `vipw`, `vigr` und `visudo` starten den Editor `vi/vim` (siehe Abschnitt »Tastenkürzel«) und öffnen darin die Datei [/etc/passwd](#), [/etc/group](#) bzw. [/etc/sudoers](#). Beim Speichern erfolgt eine Kontrolle, ob Sie die Syntax der jeweiligen Datei einhalten. Solange das nicht der Fall ist, können Sie nicht speichern.

- `-r verzeichnis`

verwendet das angegebene Verzeichnis als Root-Verzeichnis und bearbeitet die Dateien relativ zu diesem Verzeichnis.

- `-s`

öffnet die Datei [/etc/shadow](#) (Kommando `vipw`) bzw. [/etc/gshadow](#)

(Kommando `vigr`).

Sie können die drei Kommandos auch dann verwenden, wenn Sie kein Fan des Editors `vi` sind. In diesem Fall müssen Sie vorweg die Umgebungsvariable `VISUAL` oder `EDITOR` einstellen und dort den Pfad zu Ihrem Lieblingseditor angeben.

```
virsh [[-c connection] kommando]
```

Mit dem Kommando `virsh` administrieren Sie virtuelle KVM- und Xen-Maschinen. `virsh` kann auf zwei Arten verwendet werden: entweder zur direkten Ausführung eines `virsh`-Befehls oder interaktiv als Shell. Bei der ersten Variante können Sie mit der Option `-c` eine Verbindungszeichenkette angeben:

```
root# virsh -c qemu:///session list --all
```

```
...
```

Im Folgenden sind die wichtigsten `virsh`-Kommandos kurz beschrieben. Sollten Sie weitere Details in der [man](#)-Seite nachlesen, beachten Sie bitte, dass virtuelle Maschinen in der `virsh`-Nomenklatur *Domänen* heißen.

- `attach-device name device.xml [--persistent]`  
fügt einer virtuellen Maschine eine zusätzliche Hardware-Komponente hinzu (z.B. ein USB-Gerät), die im libvirt-XML-Format beschrieben ist. Die Option `--persistent` bewirkt, dass das Gerät bleibend mit der virtuellen Maschine verbunden wird und in der XML-Datei der virtuellen Maschine gespeichert wird.
- `attach-disk name source target`  
fügt einer virtuellen Maschine einen Datenträger hinzu. Dabei ist `source` der Device-Name auf dem Host-System, `target` der Device-Name im Gast. Das Kommando kann durch zahlreiche Optionen ergänzt werden, die den Treiber (`--driver`), das Caching-Verfahren (`--cache`) etc. angeben. Die Option `--persistent` verbindet den Datenträger bleibend mit der virtuellen Maschine.
- `attach-interface name type source`  
fügt einer virtuellen Maschine eine Netzwerkschnittstelle hinzu.

*type* gibt den Typ der Schnittstelle an, z.B. `network` oder `bridge`.  
*source* gibt den Schnittstellennamen auf dem Hostrechner an (z.B. `br0`). Die Details der Schnittstelle können durch weitere Optionen eingestellt werden (`--mac`, `--model` etc.).

- `autostart [--disable] name`  
gibt an, dass die virtuelle Maschine während des Bootprozesses des Hostrechners automatisch gestartet werden soll. Mit der Option `--disable` wird der automatische Start wieder abgestellt.
- `connect qemu:///session`  
stellt eine Benutzerverbindung zur `libvirtd`-Instanz des aktuellen Benutzers her. Auf diese Weise können eigene virtuelle Maschinen verwaltet werden.
- `connect qemu:///system`  
stellt eine Verbindung zur Systeminstanz von `libvirtd` her. Wenn Sie `virsh` mit `root`-Rechten ausführen, stellt `virsh` diese Verbindung automatisch her.
- `connect qemu+ssh://user@hostname/system`  
stellt eine Verbindung zur `libvirtd`-Instanz eines anderen Rechners ([hostname](#)) her. Die Kommunikation erfolgt über einen SSH-Tunnel.
- `console name`  
ermöglicht die Bedienung der angegebenen virtuellen Maschine direkt in der Konsole. Das setzt voraus, dass in der virtuellen Maschine ein `getty`-Prozess für die serielle Schnittstelle `/dev/ttys0` läuft. Um die Verbindung zu beenden, drücken Sie `[Strg]` zusammen mit der Tastenkombination für das Zeichen `]`.
- `define xmldatei`  
richtet eine neue virtuelle Maschine ein, deren Eckdaten in der angegebenen XML-Datei zusammengefasst sind. Vorsicht: Wenn bereits eine gleichnamige virtuelle Maschine existiert (gemäß

dem `<name>`-Element in der XML-Datei), wird deren Definition überschrieben!

- `destroy name`  
beendet die virtuelle Maschine sofort. Das ist so, als würden Sie bei Ihrem Rechner das Stromkabel ausstecken, und es kann dieselben Folgen haben (also ein zerstörtes Dateisystem etc.)!
- `detach-device name device.xml`  
entfernt die durch eine XML-Datei beschriebene Hardware-Komponente von der virtuellen Maschine.
- `detach-disk name target`  
entfernt den Datenträger von der virtuellen Maschine. `target` gibt den Device-Namen im Gast an.
- `detach-interface name type --mac=xxx`  
löst eine Netzwerkschnittstelle von der virtuellen Maschine.
- `domstatus name` und `dominfo name`  
liefern Informationen zu einer virtuellen Maschine.
- `edit name`  
lädt die XML-Datei zur Beschreibung der virtuellen Maschine in einen Editor, wobei die Umgebungsvariable `$EDITOR` beachtet wird.
- `help` bzw. `help kommando`  
liefert eine Liste aller Kommandos bzw. die Syntaxbeschreibung eines bestimmten Kommandos.
- `list [--inactive oder --all]`  
listet alle laufenden virtuellen Maschinen auf. Wenn Sie nur die gerade nicht aktiven oder aber alle Maschinen auflisten möchten, geben Sie die Optionen `--inactive` oder `--all` an.
- `managedsave name`  
`managedsave-remove name`  
speichert den Zustand der virtuellen Maschine (also den Inhalt des RAMs, die CPU-Register etc.) in einer Datei im Verzeichnis `/var/lib/libvirt/save/` und stoppt dann die Ausführung der

Maschine. Zur Reaktivierung der virtuellen Maschine verwenden Sie einfach `start`. Die Zustandsdatei wird dann automatisch gelöscht. Wenn Sie den gespeicherten Zustand verwerfen und die virtuelle Maschine von der Festplatte neu starten möchten, löschen Sie die Zustandsdatei mit `managedsave-remove`.

- `net-create xmlfile`  
`net-start netname`  
`net-destroy netname`  
`net-undefine netname`  
`net-list`

helfen bei der Verwaltung virtueller Netzwerke. Dabei handelt es sich um private Netzwerkbereiche, die via NAT mit dem Hostsystem verbunden werden können – beispielsweise das `default`-Netzwerk der `libvirt`-Werkzeuge.

- `pool-define xmlfile`  
`pool-define-as poolname type --target path`  
`pool-start poolname`  
`pool-auto-start xmlfile`  
`pool-destroy poolname`  
`pool-delete poolname`  
`pool-list`  
`pool-info poolname`

helfen bei der Administration von `libvirt`-Speicher-Pools. `pool-define` erzeugt einen neuen Pool, dessen Eigenschaften in einer XML-Datei beschrieben sind. `pool-define-as` erzeugt ebenfalls einen neuen Pool, wobei die Eckdaten direkt als Parameter übergeben werden. Erlaubte Pool-Typen sind unter anderem `dir` (ein lokales Verzeichnis), `netfs` (ein Netzwerkverzeichnis), `logical` (eine Volume Group), `disk` (eine Festplatte) oder `iscsi` (ein iSCSI-Server).

Der neue Pool muss anschließend mit `pool-start` gestartet werden. Wenn der Pool in Zukunft automatisch gestartet werden soll, müssen Sie außerdem `pool-autostart` ausführen.

Etwas verwirrend sind die Kommandos zum Löschen eines Pools: Ein Pool muss vor dem Löschen mit `pool-destroy` deaktiviert werden. Obwohl das Kommando Schlimmes vermuten lässt, wird der Pool dadurch lediglich gestoppt und kann später mit `pool-start` wieder gestartet werden. Erst `pool-delete` löscht den Pool. `pool-delete` setzt voraus, dass zuerst alle Volumes des Pools gelöscht wurden. Es kann also nur ein leerer Pool gelöscht werden.

- `qemu-monitor-command --hmp name 'kommando'`  
führt das angegebene QEMU-Monitor-Kommando für die durch *name* angegebene virtuelle Maschine aus. Die Option `--hmp` ist erforderlich, weil das Kommando andernfalls im JSON-Format angegeben werden muss.
- `restore datei`  
aktiviert eine mit `save` gespeicherte virtuelle Maschine wieder. Die Zustandsdatei kann anschließend gelöscht werden.
- `save name datei`  
speichert den Zustand der virtuellen Maschine (also im Wesentlichen den Inhalt des RAMs) in einer Datei und stoppt dann die Ausführung der Maschine. Zur Reaktivierung der virtuellen Maschine verwenden Sie `restore`.
- `schedinfo [optionen] name`  
zeigt die Scheduler-Parameter der virtuellen Maschine an bzw. verändert diese (`--set parameter=wert`). Mit diesen Parametern kann gesteuert werden, wie viele Hardware-Ressourcen eine virtuelle Maschine nutzen darf. Die Ressourcensteuerung erfordert die Aktivierung der cgroups-Funktionen auf dem Host-System.
- `shutdown/reboot name`  
fährt die virtuelle Maschine herunter bzw. startet sie neu. Die virtuelle Maschine erhält via ACPI ein Shutdown-Signal. In der

virtuellen Maschine muss der ACPI-Dämon `acpid` installiert sein, damit das Signal auch verarbeitet wird.

- `snapshot-create vmname [xml-datei [--redefine]]`  
`snapshot-create-as vmname snapshotname [beschreibung]`  
`snapshot-list vmname`  
`snapshot-delete vmname`  
`sname`  
`snapshot-revert vmname sname [--running]`  
erzeugt einen Snapshot einer laufenden virtuellen Maschine, listet alle Snapshots auf, löscht einen Snapshot wieder bzw. wendet den Inhalt eines Snapshots auf die Image-Datei an.  
Die Snapshot-Funktion kann nur für virtuelle Maschinen verwendet werden, die QCOW2-Image-Dateien nutzen.
- `start name`  
startet die angegebene virtuelle Maschine. Wenn Sie mit der Maschine im Grafikmodus kommunizieren möchten, verwenden Sie dazu entweder einen VNC-Client (die Verbindungsdaten ermittelt das `virsh`-Kommando `vncdisplay`, siehe unten) oder das Programm [`virt-viewer`](#).
- `suspend/resume name`  
stoppt die angegebene virtuelle Maschine vorübergehend bzw. setzt die Ausführung wieder fort. Die gestoppte virtuelle Maschine beansprucht jedoch weiterhin RAM! Es wird also nur die virtuelle CPU angehalten.
- `ttyconsole name`  
gibt an, über welches Device des Hostcomputers die serielle Schnittstelle des Gastsystems zugänglich ist (z.B. `/dev/pts/5`).
- `undefine name`  
löscht die XML-Datei, die die virtuelle Maschine beschreibt. Die Image-Datei mit der virtuellen Festplatte bleibt erhalten. `undefine` kann erst ausgeführt werden, nachdem alle Snapshots der virtuellen Maschine gelöscht wurden.

- **`vcpuinfo name`**  
liefert Informationen zu den (virtuellen) CPUs, die der virtuellen Maschine zur Verfügung stehen, sowie Angaben zur bisher beanspruchten CPU-Zeit. Mit dem Kommando kann auch das CPU-Pinning überprüft werden.
- **`vcpuin name gast-cpu-nr host-cpu-list`**  
verknüpft die physikalischen CPU-Cores des Hostsystems mit den virtuellen CPUs des Gasts. `vcpuin vm1 0 2` ordnet der ersten virtuellen CPU der virtuellen Maschine `vm1` den dritten CPU-Core des Hosts fix zu.
- **`vol-create xmlfile`**  
`vol-create-as poolname newvolname size`  
`vol-delete volname`  
`vol-list`  
`vol-info volname`  
helfen bei der Administration von Datenträgern in Speicher-Pools. `vol-create` erzeugt einen neuen Datenträger, dessen Eigenschaften in der angegebenen XML-Datei beschrieben sind. `vol-create-as` erzeugt einen neuen Datenträger in der gewünschten Größe, wobei die Suffixe `k`, `M`, `G` und `T` für KiB, MiB, GiB und TiB zulässig sind. Wenn Sie eine Image Disk erzeugen, können Sie mit `--format raw/qcow2/qed` das gewünschte Format angeben (standardmäßig `raw`).  
`--allocation size` bestimmt, wie viel des Speichers im Voraus alloziert werden soll; bei RAW-Volumes wird immer der gesamte Speicher reserviert.  
Wenn es mehrere Speicher-Pools gibt, müssen Sie bei allen Kommandos mit der Option `--pool poolname` angeben, auf welchen Pool Sie sich beziehen.
- **`vncdisplay name`**  
liefert die IP-Adresse (leer für `localhost`) und Portnummer für die VNC-Anzeige der virtuellen Maschine. Sie können nun einen

beliebigen VNC-Client starten, um mit der virtuellen Maschine zu interagieren. Aus Sicherheitsgründen funktioniert der VNC-Zugang standardmäßig nur von `localhost` (siehe die Datei `/etc/libvirt/qemu.conf`).

`vncdisplay` liefert kein Ergebnis, wenn die virtuelle Maschine ihr Grafiksystem gar nicht über VNC freigibt, sondern stattdessen das modernere Spice-System verwendet. In diesem Fall können Sie die virtuelle Maschine mit dem Programm [virt-viewer](#) bedienen. An dieses Programm können Sie direkt den Namen der virtuellen Maschine übergeben.

Sollte sich dennoch die Notwendigkeit ergeben, die Spice-Portnummer zu ermitteln, wird es schwierig. In `virsh` fehlt ein Kommando, um ähnlich wie mit `vncdisplay` den Spice-Port einer virtuellen Maschine zu ermitteln. Abhilfe schafft das folgende Kommando, das ich in einem Forum von [ubuntuusers.de](#) gefunden habe. Es extrahiert die Portnummer aus der Prozessliste.

```
\verb%spice_port=$(ps aux | grep vm_name | grep -oP "(?<-- spice port=).*(?=,)")%
```

## Beispiele

Nachdem Sie mit dem Kommando `virsh` die libvirt-Shell gestartet haben, können Sie Kommandos zur Verwaltung aller virtuellen Maschinen auf dem lokalen Rechner ausführen:

```
root# virsh
virsh# list --all
  Id  Name      Status
  --
  13 fedora    laufend
  - ubuntu     ausschalten
  - windows    ausschalten
virsh# start windows
Domain windows gestartet
virsh# vncdisplay windows
:1
virsh# exit
```

Via SSH können Sie auch eine Verbindung zum Dämon `libvирtd` auf einem anderen Rechner herstellen. Wenn auf dem KVM-Host aus

Sicherheitsgründen ein root-Login mit Passwortangabe via SSH unmöglich ist, müssen Sie vor dem ersten Verbindungsauftbau Ihren öffentlichen SSH-Schlüssel auf dem KVM-Host einrichten:

```
virsh# connect qemu qemu+ssh://user@hostname/system  
user@hostname's password: *****
```

### **virt-clone [optionen]**

`virt-clone` kopiert eine heruntergefahrenen virtuelle Maschine, die durch libvirt-Werkzeuge verwaltet wird. Dabei wird eine neue XML-Definitionsdatei sowie eine Kopie der Image-Datei erstellt. Die sonstigen Hardware-Komponenten bleiben weitgehend unverändert.

- `--auto-clone`  
gibt der neuen Maschine den Namen *bisher-clone* und der neuen Image-Datei den Namen *bisher-clone.img*. Mit dieser Option kann auf die Angabe von `--name` und `--file` verzichtet werden.
- `--connect hypervisor`  
stellt eine Verbindung zum angegebenen Virtualisierungssystem her.
- `-f datei bzw. --file=datei`  
gibt den gewünschten Namen der neuen Image-Datei an. Wenn die ursprüngliche virtuelle Maschine mehrere virtuelle Festplatten besitzt, müssen Sie diese Option mehrfach verwenden. Anstelle einer Image-Datei können Sie auch den Namen einer Device-Datei angeben, z.B. wenn Sie Logical Volumes des Hostsystems als Datenspeicher verwenden.
- `--mac nn:nn:nn:nn:nn:nn`  
gibt die gewünschte neue MAC-Adresse des Netzwerkadapters an. Wenn diese Angabe entfällt, verwendet `virt-clone` automatisch eine eindeutige zufällige MAC-Adresse.
- `--name name`  
gibt den Namen der neuen virtuellen Maschine an.

- `--original name`  
gibt den Namen der ursprünglichen virtuellen Maschine an.
- `--preserve-data`  
verhindert, dass die Image-Datei kopiert wird. Die neue virtuelle Maschine verwendet also dieselbe Image-Datei wie die bisherige virtuelle Maschine. Es ist daher nicht zulässig, die alte und die neue virtuelle Maschine gleichzeitig auszuführen! Die Option ist dann zweckmäßig, wenn Sie ein System mit einer neuen virtuellen Hardware- oder Netzwerkkonfiguration testen möchten.

## Beispiel

Das folgende Kommando kopiert die Ubuntu-Server-Installation `userver5`. Die neue virtuelle Maschine erhält den Namen `userver6`, und die neue Image-Datei wird in der Datei `/var/lib/libvirt/images/userver6.img` gespeichert. Achten Sie darauf, die neue Image-Datei in einem `libvirt`-Speicherpool anzulegen! Andernfalls verhindern die SELinux-Regeln unter RHEL/Fedora die Ausführung der virtuellen Maschine.

```
root# virt-clone --original userver5 --name userver6 \
      --file /var/lib/libvirt/images/userver6.img
```

## **virt-install [optionen]**

`virt-install` ist ein Python-Script, das beim Einrichten neuer virtueller Maschinen hilft.

- `--arch architektur`  
gibt die gewünschte CPU-Architektur an, z.B. `i386`, `i686` oder `x86_64`. Standardmäßig verwendet `virt-install` dieselbe Architektur wie auf dem Hostsystem.
- `--cdrom datei`  
gibt den Dateinamen der ISO-Datei bzw. den Device-Namen des CD/DVD-Laufwerks an, von dem die Installationsdaten gelesen werden.

- **--connect *hypervisor***  
stellt eine Verbindung zum angegebenen Virtualisierungssystem her. Wenn `virt-install` auf einem KVM-Host mit root-Rechten gestartet wird, stellt das Script automatisch eine Verbindung zu `qemu:///system` her.
- **--disk *datei***  
gibt den Dateinamen der Image-Datei an, in der die virtuelle Maschine gespeichert werden soll. Wenn Sie mehrere virtuelle Speichergeräte verwenden möchten, müssen Sie die Option mehrfach angeben.
- **--disk *opt1=wert1, opt2=wert2, ...***  
ermöglicht die Angabe mehrerer Image-Dateien aus unterschiedlichen libvirt-Speicherpools. Zulässige Optionen sind unter anderem `path` zur Angabe einer Image-Datei oder eines Block-Devices, `pool` zur Angabe eines zuvor eingerichteten libvirt-Speicher-Pools, `vol` zur Angabe einer bereits existierenden Image-Datei in einem Speicher-Pool (`vol=poolname/imagename`), `size` zur Angabe der gewünschten Größe neuer Image-Dateien (in GiB), `bus` zur Auswahl des Bussystems (`ide`, `scsi` oder `virtio`) und `cache` zur Auswahl des Cachings (`writethrough`, `writeback` oder `none`).
- **--graphics *typ, opt1=wert1, opt2=wert2 ...***  
gibt an, wie das Grafiksystem der virtuellen Maschine realisiert werden soll. Als Typ kommen `vnc` oder `spice` infrage. Standardmäßig kommt VNC zum Einsatz. Mit den weiteren Optionen können diverse Verbindungsparameter angegeben werden z.B. `port`, `listen` und `password`.
- **--import**  
bewirkt, dass `virt-install` keine Neuinstallation durchführt, sondern die neue virtuelle Maschine auf Basis einer bereits existierenden Image-Datei anlegt.

- `--name name`  
gibt den Namen der virtuellen Maschine an.
- `--network typ, opt1=wert1, opt2=wert2, ...`  
gibt die Eckdaten eines Netzwerk-Devices der virtuellen Maschine an. Die Option kann bei Bedarf mehrfach angegeben werden. `typ` gibt an, wie die Netzwerkverbindung zum Hostrechner erfolgen soll. Zulässige Werte sind `bridge=name` für eine zuvor eingerichtete Netzwerkbrücke, `network=name` für ein mit [virsh](#) definiertes virtuelles Netzwerk oder einfach `user`, wenn die virtuelle Maschine via NAT mit dem Hostsystem kommunizieren soll. Die weiteren Optionen bestimmen die Parameter des Netzwerk-Devices: `model` gibt an, welcher Adapter emuliert werden soll, z.B. `e1000`, `rtl8139` oder `virtio`. `mac` bestimmt die MAC-Adresse des Devices. Wenn dieser Parameter nicht angegeben wird, generiert `virt-install` selbst eine zufällige MAC-Adresse. MAC-Adressen für KVM müssen mit `52:54:00` beginnen!
- `--nodisk`  
ermöglicht eine Installation ohne Image-Datei (z.B., wenn die Installation in einen Netzwerkspeicher erfolgen soll).
- `--noreboot`  
verhindert den automatischen Neustart nach dem Abschluss der Installation.
- `--os-type name`  
gibt die Art des Betriebssystems an, das installiert werden soll, z.B. `linux`, `unix`, `windows` oder `other`. Die Information wird zur Optimierung diverser Hardware-Parameter genutzt.
- `--os-variant name`  
gibt an, welches Betriebssystem installiert werden soll, z.B. `fedora28`, `virtio26`, `rhel7` oder `win10`. Eine vollständige Liste aller bekannten Betriebssystemnamen liefert `man virt-install`. Wenn diese Option verwendet wird, kann auf die Angabe von `--os-type` verzichtet werden.

- `--ram n`  
gibt die Größe des RAMs in MiB an.
- `-serial pty`  
stattet die virtuelle Maschine mit einer seriellen Schnittstelle aus.  
Beim Start der virtuellen Maschine wird die serielle Schnittstelle mit einem Pseudo-TTY-Device des Hostrechners verbunden. In [virsh](#) können Sie den Device-Namen mit `ttyconsole vmname` ermitteln.
- `--soundhw=ac97/es1370/sb16`  
stattet die virtuelle Maschine mit einer Sound-Karte aus.
- `--vcpus=n`  
gibt die gewünschte Anzahl von CPU-Cores an (standardmäßig einer).
- `--video=cirrus/vga/vmvga`  
gibt an, welcher Grafikadapter emuliert werden soll (standardmäßig `cirrus`).

## Beispiel

Bevor Sie `virt-install` ausführen, müssen Sie eine Image-Datei für die virtuelle Festplatte erzeugen:

```
root# virsh
virsh# vol-create-as default disk.qcow2 10G --format qcow2
virsh# exit
```

Mit dem Kommando `virt-install` richten Sie nun eine neue virtuelle Maschine ein:

```
root# virt-install --name myvmname --ram 2048 --cdrom install.iso \
--os-variant rhel8 --disk vol=default/disk.qcow2 --graphics vnc \
--noreboot
```

**`virt-top` [optionen]**

`virt-top` zeigt an, wie viel CPU-Kapazität und Speicherplatz die laufenden virtuellen Maschinen benötigen. Während `virt-top` läuft, können Sie die Anzeige mit `0`, `1`, `2` und `3` nach virtuellen Maschinen, CPU-Cores, Netzwerkschnittstellen oder Datenträger-Devices ordnen. `Q` beendet das Kommando.

- `--connect hypervisor`  
stellt die Verbindung zum angegebenen Virtualisierungssystem her. Standardmäßig kommuniziert `virt-top` mit `qemu:///system`, sofern es mit root-Rechten gestartet wurde.
- `-o sort`  
gibt die gewünschte Sortierordnung an, z.B. `cpu`, `mem` oder `time`, um die Liste nach der CPU-Leistung, dem Speicherbedarf oder der gesamten Rechenzeit zu ordnen.

## Beispiel

```
root# virt-top
virt-top 14:36:17 - x86_64 8/8CPU 1600MHz 15961MB
5 domains, 3 active, 3 running, 0 sleeping, 0 paused, 2 inactive D:0 O:0 X:0
CPU: 0.1% Mem: 6144 MB (6144 MB by guests)
```

ID	S	RDRQ	WRRQ	RXBY	TXBY	%CPU	%MEM	TIME	NAME
3	R	0	22	0	0	1.3	12.0	35:20:50	kofler.info
4	R	0	0	0	0	0.1	12.0	274:52.67	pi-buch.info
1	R	0	0	0	0	0.0	12.0	39:11:52	ubuntu-buch.info
	-								(centos-buch.info)
	-								(michael-kofler.com)

**`virt-viewer [optionen] name/id/uuid`**

`virt-viewer` ist ein VNC- und Spice-Client, der das Grafiksystem einer virtuellen Maschine anzeigt. An das Kommando wird normalerweise einfach der Name, die ID- oder die UUID-Nummer der virtuellen Maschine übergeben.

- `-c hypervisor`  
stellt eine Verbindung zum angegebenen Virtualisierungssystem her. Die Option ist nur erforderlich, wenn es sich um einen externen KVM-Host handelt.

## W

### **wait** [prozessnummer]

Das bash-Kommando `wait` wartet auf das Ende des angegebenen Hintergrundprozesses. Wenn keine Prozessnummer angegeben wird, wartet das Kommando auf das Ende aller laufenden Hintergrundprozesse, die von der Shell gestartet wurden.

### **Beispiel**

Der Einsatz von `wait` bietet sich vor allem in Shell-Scripts an, wenn zuerst mehrere Hintergrundprozesse gestartet werden und dann auf das Ende all dieser Prozesse gewartet werden soll:

```
#!/bin/bash
kommando1 &
kommando2 &
kommando3 &
kommando4 &
wait
# hier erst fortsetzen, wenn Kommando 1 bis Kommando 4 fertig sind
...
```

### **wakeonlan** [optionen] mac

`wakeonlan` aus dem gleichnamigen Paket (verfügbar für Debian und Ubuntu) sendet ein spezielles Netzwerkpaket für eine MAC-Adresse und versucht, das so identifizierte Gerät aus dem Ruhestand »aufzuwecken«. Das funktioniert nur, wenn das Gerät den Standard *Wake On LAN* unterstützt.

- `-i host`

sendet das Wake-on-LAN-Paket nur an den angegebenen Host bzw. an die angegebene Adresse und nicht an das gesamte Netzwerk.

### **watch** kommando

`watch` führt das angegebene Kommando periodisch aus und zeigt die Ausgabe an.

- `-d` bzw. `--differences`

markiert die Unterschiede im Vergleich zum vorherigen Ergebnis.

- **-e bzw. --errexit**  
beendet `watch`, wenn das aufgerufene Kommando einen Fehler zurückgibt.
- **-n n**  
gibt an, nach wie vielen Sekunden das Kommando jeweils neu ausgeführt werden soll (standardmäßig alle zwei Sekunden).

## Beispiel

Das folgende Kommando zeigt alle Änderungen in der Liste der laufenden `mysqld`-Prozesse an:

```
user$ watch -d 'ps aux | grep mysqld'
```

**wc dateien**

`wc` zählt die Anzahl der Zeilen, Wörter und Zeichen in den angegebenen Dateien. Wenn durch Jokerzeichen mehrere Dateien erfasst werden, berechnet `wc` auch die Gesamtsumme der drei Angaben. `wc` ist auch gut für die Kombination mit anderen Programmen geeignet.

## Beispiel

[find](#) liefert für jede reguläre Datei eine Zeile. `wc` zählt die Zeilen und liefert somit die Gesamtanzahl aller Dateien im Verzeichnisbaum.

```
user$ find / -type f -print | wc
```

**wget [optionen] url1 url2 ...**

`wget` lädt Dateien von HTTP-, HTTPS- und FTP-Servern herunter. `wget` kann unterbrochene Downloads wieder aufnehmen, rekursiv Links verfolgen und eignet sich zur Automatisierung von Downloads. Die zu übertragenden Dateien werden in Form von URLs (Uniform Resource Locator) angegeben, also beispielsweise als `http://meinserver.de/datei.iso`. Der Befehl `wget` wird durch unzählige Optionen gesteuert, von denen hier nur die wichtigsten zusammengefasst werden:

- **-b**  
führt den Download als Hintergrundprozess durch und schreibt alle Statusmeldungen in die Datei `wget-log`.

- **-B *url* bzw. --base=*url***  
stellt allen relativen Links innerhalb der Datei, die Sie durch **-i *datei*** angegeben haben, die Basisadresse *url* voran.
- **-c**  
nimmt einen unterbrochenen Download wieder auf. Vorsicht:  
Wenn sich die Datei in der Zwischenzeit auf dem Server geändert hat, ist die heruntergeladene Datei fehlerhaft und enthält teilweise Daten aus der alten Datei und teilweise Daten aus der neuen Datei!
- **--force-html**  
interpretiert die mit **-i** angegebene Datei als HTML-Datei und lädt alle Dateien, auf die Links der HTML-Datei zeigen.
- **--ftp-user=*user* --ftp-password=*pw***  
gibt den Benutzernamen und das Passwort für FTP-Downloads an. Diese Informationen können auch in der URL übergeben werden (`ftp://user:password@server/datei.txt`).
- **-i *datei* bzw. --input-file=*datei***  
liest die zu übertragenden Dateien (URLs) aus der angegebenen Textdatei. Wenn Sie statt einer Datei das Zeichen **-** übergeben, erwartet wget die URLs aus der Standardeingabe. Die Option **-i** akzeptiert als Parameter nur lokale Dateien, keine URLs. (`-i http://server/name.html` funktioniert also nicht!)
- **--limit-rate=*n***  
limitiert die Download-Menge pro Sekunde. Die Buchstaben **k** und **m** bezeichnen Kilo- bzw. Megabytes (also etwa `--limit-rate=0.25m`).
- **--no-check-certificate**  
verzichtet auf die Überprüfung von HTTPS-Zertifikaten. Das ist dann sinnvoll, wenn Sie wissen, dass der betreffende Server selbst signierte Zertifikate verwendet.

- **-q**  
verzichtet auf die Ausgabe von Statusmeldungen (*quiet*).
- **--retry-connrefused**  
unternimmt auch nach dem Fehler *connection refused* weitere Versuche, die Datei herunterzuladen. Die Option ist nur bei unzuverlässigen Download-Servern zweckmäßig, die gelegentlich aus dem Netz verschwinden und wenig später wieder auftauchen.
- **--spider**  
testet, ob alle Links in der durch `-i htmldatei --force-html` angegebenen Datei noch gültig sind. Die durch Links angegebenen Dateien werden aber nicht heruntergeladen.
- **-t n** bzw. **--tries=n**  
unternimmt bei einem Verbindungsabbruch *n* Versuche, sich neuerlich zu verbinden (standardmäßig 20). Mit `-t 0` versucht wget sein Glück so lange, bis der Download gelungen ist oder Sie das Kommando abbrechen.
- **-w n** bzw. **--wait=n**  
gibt an, wie viele Sekunden wget warten soll, bevor es die nächste Datei herunterlädt. Die Option verhindert, dass der Download-Server durch unzählige, nahezu gleichzeitige Download-Anfragen zu stark belastet wird.

## Rekursive Downloads

Die folgenden Optionen steuern rekursive Downloads:

- **-E** bzw. **--html-extension**  
fügt bei allen heruntergeladenen Dateien des Typs *application/xhtml+xml* oder *text/html*, deren Namen nicht mit *.html* oder *.htm* enden, die Endung *.html* an den Dateinamen an.
- **-H** bzw. **--span-hosts**  
verfolgt auch Links auf andere Websites.
- **-k** bzw. **--convert-links**  
ändert in den heruntergeladenen HTML-Dateien die Links so,

dass sie auf die lokalen Dateien verweisen. Das ermöglicht es später, die Seiten offline anzusehen.

- **-l n** bzw. **--level n**  
limitiert die Rekursionsebene (standardmäßig 5). **-l inf** deaktiviert die Limitierung.
- **-L** bzw. **--relative**  
verfolgt nur relative Links (aber keine absoluten Links auf die Start-Website).
- **-r**  
aktiviert rekursive Downloads. Bei HTTP-Downloads verfolgt wget alle HTTP-Links der Startseite, lädt die so angegebenen Dateien herunter, verfolgt auch deren Links etc. wget berücksichtigt dabei nur relative Links sowie Links auf Seiten der Start-Website. Bei FTP-Downloads liest wget alle Unterverzeichnisse.
- **-p**  
lädt alle Dateien herunter, die zum Betrachten der Ausgangsdatei erforderlich sind (auch Dateien, bei denen die maximale Rekursionsebene überschritten wird).

## Beispiele

In der Grundform lädt wget die angegebene Datei einfach herunter:

```
user$ wget ftp://myftpserver.de/name.abc
```

Wenn der Download aus irgendeinem Grund unterbrochen wird, kann er mit **-c** ohne Umstände wieder aufgenommen werden:

```
user$ wget -c ftp://myftpserver.de/name.abc
```

Um eine Website später offline zu lesen bzw. um ihren aktuellen Zustand zu archivieren, hilft das folgende rekursive Download-Kommando (Option **-r**).

Die Rekursionstiefe wird durch **-l 4** auf vier Ebenen limitiert.

```
user$ wget -r -l 4 -p -E -k http://website.de
```

```
whatis datei
```

**whatis** gibt eine kurze Beschreibung (meist einzeilig) des angegebenen Kommandos bzw. Schlüsselworts aus. **whatis**-Beschreibungen existieren nur zu Themen, zu denen [man](#)-Texte

installiert sind. Wenn `whatis` nicht funktioniert, fehlen wahrscheinlich die zugrunde liegenden Datenbanken, die mit `mandb` bzw. mit `makewhatis` erzeugt werden können.

### **whereis datei**

`whereis` durchsucht alle üblichen Pfade für Binärdateien, [man](#)-Dateien und Quellcode nach dem angegebenen Dateinamen. `whereis` ist damit weniger gründlich als [find](#), dafür aber deutlich schneller. Die [man](#)-Seite zu `whereis` zählt auf, welche Verzeichnisse durchsucht werden.

### **which kommando**

`which` durchsucht alle in `PATH` angegebenen Pfade nach dem Kommando. `which` liefert als Antwort den vollständigen Namen des Kommandos, das ausgeführt würde, wenn das Kommando ohne Pfadinformationen aufgerufen würde. Das ist vor allem dann eine Hilfe, wenn zu einem Kommando mehrere Versionen in unterschiedlichen Verzeichnissen existieren.

In der `bash` kann statt `which` auch [type](#) verwendet werden. [type](#) hilft Ihnen dabei, herauszufinden, ob `kommando` in Wirklichkeit ein eingebautes Shell-Kommando, ein Alias oder eine selbst definierte Funktion ist.

## **Beispiel**

Das folgende Kommando ermittelt, wo das Kommando [ls](#) im Dateisystem gespeichert ist:

```
user$ which ls  
/bin/ls
```

```
while bedingung; do  
    kommandos  
done
```

`while` bildet Schleifen in `bash`-Scripts. Die Schleife wird so lange ausgeführt, bis die angegebene Bedingung zum ersten Mal nicht mehr erfüllt ist. Das Schleifenkriterium ist der Rückgabewert des Kommandos, das als Bedingung angegeben wird. Vergleiche und

Tests werden mit dem Kommando [test](#) oder dessen Kurzform in eckigen Klammern durchgeführt.

## Beispiele

Die folgende Schleife gibt die Zahlen 0 bis 5 aus:

```
#!/bin/bash
i=0
while [ $i -le 5 ]; do
    echo $i
    i=$[$i+1]
done
```

Um eine Textdatei zeilenweise zu verarbeiten, formulieren Sie Ihr bash-Script so:

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

### who [optionen]

who zeigt eine Liste aller zurzeit eingeloggten Systembenutzer an. Auch wenn Sie allein mit Ihrem Rechner arbeiten, können Sie sich mit unterschiedlichen Namen an verschiedenen Textkonsolen einloggen. Wenn Sie wissen möchten, wer zuletzt auf diesem Rechner eingeloggt war, führen Sie das Kommando [last](#) aus.

- **-a**  
liefert detaillierte Informationen zu jedem Benutzer, inklusive der verwendeten Konsole und dem Login-Zeitpunkt. Bei SSH-Verbindungen zeigt who auch an, von welchem Rechner der Login erfolgte.
- **-m**  
gibt den Benutzernamen der gerade aktiven Konsole an. Das Kommando who am i hat dieselbe Bedeutung.

## Beispiel

Auf dem Testrechner gibt es neben den TTY-Prozessen für die sechs Konsolen zwei aktive Shell-Sessions des Benutzers kofler:

```
user$ who -a
          system boot 2020-09-09 08:57
          run-level 2 2020-09-09 08:57
LOGIN      tty5        2020-09-09 08:57          971 id=5
```

```
LOGIN      tty2          2020-09-09 08:57          976 id=2
LOGIN      tty3          2020-09-09 08:57          977 id=3
LOGIN      tty1          2020-09-09 08:57          1802 id=1
kofler    + pts/1        2020-09-24 11:45 02:02      351 (xxx.telekom.at)
kofler    + pts/2        2020-09-24 13:47          .       2354 (xxx.telekom.at)
```

### w~~hois~~ [optionen] hostname

whois führt DNS-Abfragen für den angegebenen Hostnamen durch und zeigt das Ergebnis an.

### Beispiel

```
user$ whois kofler.info
Domain Name: KOFLER.INFO
...
Domain Status: ok https://icann.org/epp#ok
Registrant Name: Kofler Michael
...

```

### w~~ol~~ [optionen] mac

w~~ol~~ aus dem gleichnamigen Paket (verfügbar für CentOS, Fedora und RHEL) sendet ein spezielles Netzwerkpaket für eine MAC-Adresse und versucht damit, das Gerät aus dem Ruhestand »aufzuwecken«. Das funktioniert nur, wenn das Gerät den Standard *Wake On LAN* unterstützt.

- -h *host* bzw. -i *host*  
sendet das Wake-on-LAN-Paket nur an den angegebenen Host bzw. an die angegebene Adresse und nicht an das gesamte Netzwerk.

### w~~pa\_passphrase~~ ssid

w~~pa\_passphrase~~ hilft dabei, eine Konfigurationsdatei für den WLAN-Authentifizierungsdienst w~~pa\_supplicant~~ einzurichten.

### Beispiel

w~~pa\_supplicant~~ erwartet als Parameter den Namen eines WLAN-Netzes (also den sogenannten *Service Set Identifier*). Anschließend müssen Sie das Passwort für das WLAN eingeben. Das Kommando gibt dann mehrere Zeilen in der Syntax von w~~pa\_supplicant~~ aus. Sie müssen diese Zeilen nun nur noch an das Ende von */etc/wpa\_supplicant/wpa\_supplicant.conf* anfügen.

```
user$ wpa_passphrase mein-wlan-name
# reading passphrase from stdin
```

```
strengGeheim
network={
    ssid="mein-wlan-name"
    #psk="strengGeheim"
    psk=9a184914197f550e1c6b350cc49b09a5fab532a8ec991f997ee77fd0a5e78d96
}

```

### **write** username

**write** ermöglicht es, einem anderen Benutzer eine Nachricht zu senden.

Nach der Ausführung des Kommandos werden alle eingegebenen Zeichen bis **Strg** + **D** zum Terminal des angegebenen Benutzers übertragen.

### **wsl** [optionen] [kommando]

**wsl** ist kein Linux-, sondern ein Windows-Kommando. Es hat deswegen Aufnahme in dieses Buch gefunden, weil es dabei hilft, das *Windows-Subsystem für Linux* zu verwalten, also Linux-Installationen unter Windows zu starten, zu entfernen etc. Das Kommando steht nur zur Verfügung, wenn Sie im Programm *Windows Features aktivieren* die Option **WINDOWS-SUBSYSTEM FÜR LINUX** aktivieren.

Ohne weitere Optionen startet **wsl** die Default-Linux-Installation bzw. führt dort das an **wsl** übergebene Kommando aus.

- **-d** bzw. **--distribution name**  
startet die angegebene Distribution. Die weitere Nutzung erfolgt interaktiv in einer Shell.
- **-l** bzw. **--list**  
listet alle Linux-Installationen auf. Mit der zusätzlichen Option **--verbose** verrät das Kommando auch, welche Linux-Instanzen gerade ausgeführt werden und ob diese WSL1- oder WSL2-kompatibel sind. (WSL2 verwendet zur Ausführung der jeweiligen Distribution einen echten Linux-Kernel, was Geschwindigkeitsvorteile hat. Allerdings gibt es bei WSL2 im Vergleich zu WSL1 Einschränkungen bezüglich der Netzwerkanbindung.)

- **-s bzw. --set-default *name***  
legt fest, welche Linux-Installation als Default-Linux-System gelten soll.
- **--set-default-version *name* 1|2**  
bestimmt, ob neue Linux-Installationen standardmäßig mit WSL1 oder mit WSL2 ausgeführt werden.
- **--set-version *name* 1|2**  
legt fest, wie die angegebene Linux-Distribution ausgeführt werden soll (WSL1 oder WSL2).
- **--shutdown**  
fährt alle laufenden Linux-Distributionen herunter.
- **-t bzw. --terminate *name***  
stoppt die Ausführung der angegebenen Linux-Distribution.
- **-u bzw. --user *name***  
führt das Kommando bzw. den Login für den angegebenen Benutzer aus.
- **--unregister *name***  
löscht die angegebene Linux-Installation.

`wsl` bietet keine Möglichkeit, eine neue Linux-Distribution zu installieren. Dazu müssen Sie den Microsoft Store zu Hilfe nehmen, in dem diverse Linux-Distributionen zum kostenlosen Download zur Auswahl stehen. Beim ersten Start einer WSL-Distribution müssen Sie einen Benutzer samt Passwort einrichten. Dieser Benutzer hat [sudo](#)-Rechte.

## Beispiele

Auf dem Testrechner gibt es vier Linux-Installationen, wobei Ubuntu als Defaultdistribution gilt:

```
> wsl --list --verbose
  NAME          STATE      VERSION
* Ubuntu        Stopped    2
  kali-linux    Stopped    1
  docker-desktop-data  Stopped    2
  docker-desktop  Stopped    2
```

wsl führt zweimal das Kommando [id](#) aus, einmal im Standardaccount und einmal mit root-Rechten. Dazu ist kein Passwort erforderlich.

```
> wsl id  
uid=1000(kofler) gid=1000(kofler) groups=1000(kofler),4(adm),...
```

```
> wsl -u root id  
uid=0(root) gid=0(root) groups=0(root)
```

Das letzte Beispiel startet Kali Linux für die interaktive Nutzung:

```
> wsl -d kali-linux  
$ cat /etc/os-release  
PRETTY_NAME="Kali GNU/Linux Rolling"  
NAME="Kali GNU/Linux"  
...  
  
$ ip a  
8: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 group default qlen 1  
    link/ether 04:d4:c4:92:65:f5  
    inet 192.168.178.52/24 brd 192.168.178.255 scope global dynamic  
        ...  
16: eth1: <BROADCAST,MULTICAST,UP> mtu 1500 group default qlen 1  
    link/ether 00:15:5d:7e:85:98  
    inet 172.31.96.1/20 brd 172.31.111.255 scope global dynamic  
        ...
```

## X

### xargs kommando

xargs leitet die von der Standardeingabe kommenden Daten an das als Parameter angegebene Kommando weiter. xargs wird meist in Kombination mit einem per Pipe vorangestellten Kommando verwendet, also in der Form `kommando1 | xargs kommando2`. Auf diese Weise können die Resultate des ersten Kommandos mit dem zweiten Kommando verarbeitet werden. Falls die Ergebnisse des ersten Kommandos so umfangreich sind, dass sie nicht in einer Kommandozeile übergeben werden können, wird `kommando2` in mehreren Schritten aufgerufen.

Der Unterschied zu `kommando1 | kommando2` besteht darin, dass xargs die Standardeingabe in Form von Parametern an `kommando2` übergibt. xargs muss eingesetzt werden, wenn `kommando2` nur die übergebenen Parameter verarbeitet. Gewöhnliche Pipes eignen sich dagegen, wenn `kommando2` die Daten aus der Standardeingabe verarbeitet.

- `--null`

erwartet 0-Bytes zur Trennung von Dateinamen (nicht aber Leer- und Tabulatorzeichen). Diese Option eignet sich zur Verarbeitung von [find](#)-Ergebnissen, wenn [find](#) mit der Option `-print0` ausgeführt wurde.

### Beispiel

Das folgende Kommando durchsucht das aktuelle Verzeichnis und alle Unterverzeichnisse nach Backup-Dateien, die mit dem Zeichen ~ enden, und löscht diese. Das funktioniert auch für Dateinamen, die Leerzeichen enthalten.

```
user$ find -name '*.*' -print0 | xargs --null rm
```

### xdpyinfo [optionen]

`xdpyinfo` liefert im Grafikmodus umfassende Informationen zum laufenden X-Server. Mit [grep](#) können Sie daraus die für Sie

relevanten Details extrahieren.

## Beispiel

Die folgenden Kommandos ermitteln die Release-Nummer des X-Servers, die Auflösung des Bildschirms sowie dessen Pixeldichte (DPI, *Dots per Inch*):

```
user$ xdpinfo | grep release
vendor release number: 11702000
root# xdpinfo | grep -C 1 dimensions
screen #0:
dimensions: 1600x1200 pixels (411x311 millimeters)
resolution: 99x98 dots per inch
```

### xfs\_admin [optionen] device/imagedatei

xfs\_admin zeigt bzw. verändert Parameter eines XFS-Dateisystems.

- **-c 0|1**  
deaktiviert bzw. aktiviert die sogenannte *Lazy-Counter*-Funktion des Dateisystems. Damit werden die Superblocks des Dateisystem seltener aktualisiert, was gewisse Dateioperationen spürbar beschleunigt.
- **-f**  
gibt an, dass sich das Dateisystem in einer Image-Datei und nicht auf einem Datenträger-Device befindet.
- **-l**  
gibt den Label des Dateisystems aus.
- **-L name**  
stellt den Namen (Label) des Dateisystems neu ein.
- **-u**  
zeigt die UUID des Dateisystems an.
- **-U uuid**  
weist dem Dateisystem eine neue UUID zu.

### xfs\_growfs [optionen] mount-verzeichnis

xfs\_growfs vergrößert ein XFS-Dateisystem. Das setzt voraus, dass das zugrunde liegende Device (z.B. eine Partition oder ein Logical Volume) vorher vergrößert wurde.

Die Vergrößerung erfolgt im laufenden Betrieb. Das Dateisystem muss also in den Verzeichnisbaum eingebunden sein. Eine Verkleinerung von XFS-Dateisystemen ist nicht vorgesehen. `xfs_growfs` kennt diverse Optionen, mit denen Sie steuern können, welche Bereiche des Dateisystems wie stark vergrößert werden sollen. Im Regelfall ist es aber nicht notwendig, diese Optionen anzugeben. `xfs_growfs` erkennt, wie groß der zugrunde liegende Datenträger ist, und entscheidet selbst, wie groß die verschiedenen Bereiche des XFS-Dateisystems werden sollen.

#### **`xfs_info` device**

`xfs_info` gibt eine Zusammenfassung der Eckdaten eines XFS-Dateisystems aus. Das Kommando ist gleichwertig zu `xfs_growfs -n`.

#### **`xfs_repair` [optionen] device/imagedatei**

`xfs_repair` versucht, ein beschädigtes XFS-Dateisystem wieder in einen konsistenten Zustand zu bringen.

- **-d**  
repariert ein aktives Read-only-Dateisystem.  
Normalerweise kann `xfs_repair` nur Dateisysteme reparieren, die momentan nicht genutzt werden. Diese Option ist dann zweckmäßig, wenn Sie Ihren Linux-Rechner im Single-User- oder Emergency-Modus gestartet haben und die Systempartition reparieren müssen. Nach der Reparatur müssen Sie das System neu starten. Die [man](#)-Seite beschreibt die Option als *dangerous*. Sicherer ist es, den Rechner mit einem Live- oder Emergency-System zu starten und die Reparatur von dort auszuführen.
- **-f**  
gibt an, dass sich das Dateisystem in einer Image-Datei und nicht auf einem Datenträger-Device befindet.
- **-n**  
führt keine Änderungen durch, sondern zeigt an, welche

Reparaturen durchgeführt werden, wenn das Kommando ohne diese Option ausgeführt wird.

### **xhost** +/-hostname

Mit **xhost** +hostname akzeptiert der Grafik-Server X Verbindungen vom angegebenen Hostnamen. Analog blockiert X nach **xhost** - hostname Verbindungen von diesem Host.

Aus Sicherheitsgründen erlaubt das Grafiksystem X zumeist nur lokale Verbindungen. Damit das Arbeiten von einem externen Rechner gelingt, muss für diesen Host mit **xhost** +name explizit eine Ausnahme definiert werden. Noch liberaler ist **xhost** +: Damit wird die Zugriffskontrolle komplett deaktiviert, ein Login ist von jedem Host aus möglich.

### **Beispiel**

In der Praxis ist das Ausführen von **xhost** vor allem dann oft erforderlich, wenn Sie zuerst als normaler Benutzer via VNC eine Verbindung zum Grafik-Server herstellen und dann mit [sudo](#) oder [su](#) ein grafisches Programm mit root-Rechten ausführen möchten.

Standardmäßig ist das nicht erlaubt. Abhilfe schafft **xhost** +localhost (weil ja dank VNC die Verbindung schon hergestellt wurde und es nun nur noch darum geht, Programme auch unter einem anderen Account auszuführen).

Das folgende Beispiel setzt voraus, dass Sie zuerst eine VNC-Verbindung herstellen, dann in einem Terminalfenster zuerst **xhost** ausführen und anschließend mit [sudo](#) ein Programm im Grafikmodus – hier den Editor **gedit** – starten.

```
user$ xhost +localhost  
user$ sudo gedit      (Editor mit root-Rechten starten)
```

### **xinput** [kommando]

**xinput** hilft bei der Konfiguration von Eingabegeräten (z.B. Tastatur, Maus und Touchpad) für das X-Window-System.

- **list**  
listet alle erkannten Eingabegeräte auf.

- **list-props *geräteid***  
listet die Parameter eines Geräts auf. Die Geräte-ID ermitteln Sie vorher mit `xinput list`.
- **set-button-map *geräteid mapbutton1 mapbutton2 ...***  
verändert die Zuordnung der Buttons einer Maus.

### **xkill [optionen]**

`xkill` beendet ein hängen gebliebenes oder halb abgestürztes X-Programm. Das betreffende Fenster muss nach dem Start des Kommandos mit der Maus angeklickt werden.

### **xrandr [optionen]**

`xrandr` ändert im laufenden Betrieb die Bildschirmauflösung, die Bildfrequenz und andere Einstellungen.

- **--addmode *ausgang name***  
fügt für den angegebenen Display-Ausgang einen neuen, eventuell zuvor mit `--newmode` definierten Grafikmodus hinzu.
- **--dpi *n***  
gibt an, wie viele Pixel pro Zoll angezeigt werden.
- **--left-of *output name* bzw. `--right-of name` bzw. `--below name` bzw. `--above name`**  
aktiviert den durch `--output` angegebenen Signalausgang und gibt an, wie die Bildschirme relativ zueinander positioniert sind. Das funktioniert nur, wenn in `xorg.conf` eine ausreichend große virtuelle Auflösung eingestellt wurde, damit beide Bildschirme abgedeckt werden können.
- **--newmode *name freq x1 x2 x3 x4 y1 y2 y3 y4***  
definiert einen neuen Modus (eine neue Auflösung). Die Syntax entspricht der von `ModeLine`-Zeilen in `xorg.conf`. Zur Ermittlung der richtigen Zahlenkombination können Sie das Kommando [gtf](#) zu Hilfe nehmen.
- **--off**  
schaltet den mit `--output` angegebenen Signalausgang aus.

- `--orientation x`  
ändert die Bildlage. Zulässige Einstellungen sind `normal`, `inverted`, `left` und `right`.
- `--output name`  
gibt an, dass sich die weiteren Optionen auf einen bestimmten Signalausgang beziehen. Das ist zweckmäßig, wenn mehrere Monitore angeschlossen sind. Die Namen der aktiven Signalausgänge ermitteln Sie mit `xrandr -q`.
- `--primary`  
macht den mit `--output` ausgewählten Bildschirm zum primären Bildschirm, auf dem Gnome oder KDE das Panel, das Dock etc. anzeigen.
- `-q`  
gibt an, welche Einstellmöglichkeiten zur Verfügung stehen. Das Ergebnis ist stark vom eingesetzten Grafiktreiber abhängig.
- `--rate n`  
ändert die Bildfrequenz für die aktuelle Auflösung. *n* ist die gewünschte Bildfrequenz in Hertz.
- `--size n`  
ändert die Auflösung. *n* ist eine Nummer der Ergebnisliste von `xrandr -q` oder die gewünschte Auflösung in der Form *breite**höhe*.

## Beispiel

Das folgende Kommando stellt die Auflösung auf 1280×1024 Punkte ein:

```
user$ xrandr --size 1280x1024
```

Das nächste Kommando aktiviert sowohl den DVI- als auch den VGA-Signalausgang. (Die Namen der Ausgänge variieren je nach Grafikkarte und Treiber.) Auf beiden Monitoren wird dasselbe Bild angezeigt. Die Option `--auto` bewirkt, dass jeder Monitor in der für ihn optimalen Auflösung und Bildfrequenz betrieben wird.

```
user$ xrandr --output DVI-I-0 --auto --output VGA-0 --auto
```

Die folgenden drei Kommandos definieren einen neuen Grafikmodus in der Auflösung von 1280×720 Pixel und aktivieren diesen für den HDMI1-Ausgang:

```
user$ xrandr --newmode 1280x720 74.18 1280 1390 1430 1650 720 725 730 750  
user$ xrandr --addmode HDMI1 1280x720  
user$ xrandr --size 1280x720
```

### xset [kommando]

xset verändert diverse Einstellungen des X-Servers.

- +dpms bzw. -dpms  
aktiviert bzw. deaktiviert die Energiesparfunktionen (Display Power Management Signaling).
- dpms *n1* [*n2* [*n3*]]  
gibt an, nach wie vielen Sekunden die DPMS-Modi *Standby*, *Suspend* und *Off* des Monitors aktiviert werden sollen.
- q  
liefert eine Liste mit den aktuellen Einstellungen.
- s *n*  
aktiviert nach *n* Sekunden den Bildschirmschoner.

### xz [optionen] dateien

xz aus dem Paket `xz-utils` komprimiert die angegebenen Dateien. xz ist ein relativ neues Komprimierkommando, das noch kleinere Dateien als [bzip2](#) liefert. Die komprimierten Dateien erhalten standardmäßig die Endung `.xz`.

- -*0* bis -9  
gibt an, wie gut xz komprimieren soll. Die besten Ergebnisse (also die kleinsten Dateien) liefert -9, allerdings beansprucht xz dann am meisten Arbeitsspeicher. Standardmäßig gilt -6. In diesem Fall beansprucht xz beim Komprimieren maximal 100 MiB, beim Dekomprimieren maximal 10 MiB.
- -d bzw. --decompress  
dekompriamt die angegebene Datei (entspricht `unxz`).

## Y

**yum** [optionen] kommando

`yum` installiert, aktualisiert und entfernt unter CentOS und RHEL RPM-Pakete. Hinter den Kulissen kommt seit Version 8 aber das zuerst in Fedora etablierte Kommando [dnf](#) zum Einsatz. Deswegen sind in diesem Buch alle Kommandos und Optionen bei [dnf](#) beschrieben.

## Z

```
zcat datei.gz  
zless datei.gz  
zmore datei.gz
```

Die drei Kommandos funktionieren wie [cat](#), [less](#) und [more](#). Der einzige Unterschied besteht darin, dass mit [gzip](#) komprimierte Dateien direkt gelesen werden können, also ohne vorheriges Dekomprimieren mit [gunzip](#).

Statt [zless](#) kann bei vielen Distributionen direkt [less](#) eingesetzt werden: Dieses Kommando ist meist so konfiguriert, dass es ohne Unterschied für komprimierte und nichtkomprimierte Dateien verwendet werden kann.

```
zenity [optionen]
```

[zenity](#) zeigt einfache GTK-Dialoge zur Texteingabe, zur Auswahl eines Tags im Kalender oder zur Dateiauswahl an. Das Kommando gibt das Ergebnis zur Weiterverarbeitung zurück. Mit [zenity](#) können Sie Ihren bash-Scripts einen moderneren Anstrich geben – vorausgesetzt, das Script wird in einem grafischen Desktop-System ausgeführt.

Die folgenden Optionen illustrieren lediglich das Konzept des Kommandos. Das Kommando bietet mit weit über 50 Optionen erheblich mehr Eingabe- und Steuerungsmöglichkeiten.

- **--calendar**  
zeigt einen Kalender an und ermöglicht die Auswahl eines Tags.
- **--entry**  
ermöglicht eine Texteingabe.
- **--file-selection**  
erlaubt die Auswahl einer Datei.
- **--password**  
ermöglicht eine Passworteingabe.

- `--text text`  
gibt an, welcher Text innerhalb des Dialogs angezeigt werden soll.
- `--title text`  
stattet den Dialog mit einem Titel aus.

## Beispiel

Die folgenden Zeilen eines Scripts fordern den Benutzer zuerst zur Passworteingabe auf und zeigen das Passwort dann an:

```
pw=$(zenity --password --title 'Ihr Passwort bitte ...')
echo "Das war Ihr Passwort: $pw"
```

**zip** [optionen] archiv.zip datei1 datei2 ...

Das Kommando bildet das ZIP-Archiv *archiv.zip* und fügt darin alle angegebenen Dateien ein. Das Archiv eignet sich besonders für den Datenaustausch mit Windows-Anwendern. Um die Dateien unter Linux wieder zu extrahieren, verwenden Sie `unzip`.

- `-r`

archiviert rekursiv den Inhalt von Verzeichnissen.

## Beispiel

Das folgende Kommando fügt alle als Parameter übergebenen HTML-Dateien in *meinarchiv.zip* ein:

```
user$ zip meinarchiv.zip *.html
zipinfo [optionen] datei.zip
```

Das Kommando liefert das Inhaltsverzeichnis eines ZIP-Archivs.

**zypper** [optionen] kommando

Das SUSE-spezifische Kommando `zypper` hilft bei der RPM-Paketverwaltung. Es installiert und aktualisiert Pakete, richtet Paketquellen ein etc. `zypper` setzt die `libzypp`-Bibliothek voraus, die ein integraler Bestandteil aller aktuellen SUSE-Distributionen ist. Die im Folgenden beschriebenen Kommandos können abgekürzt werden, sofern die Bedeutung eindeutig ist, also z.B. `zypper in name` statt `zypper install name`.

- `addrepo [optionen] uri name`  
richtet eine neue Paketquelle ein und gibt ihr einen Alias-Namen.

Der *Uniform Resource Identifier* (URI) beschreibt die Paketquelle, beispielsweise in der folgenden Form:

`http://download.opensuse.org/distribution/leap/15.0/repo/oss/`  
Standardmäßig werden Paketquellen aktiviert (Spalte *enabled* in `zypper repos`) und zur Aktualisierung vorgemerkt (Spalte *refresh*). Die Paketquelle wird allerdings nicht erstmalig in den Cache eingelesen. Dazu müssen Sie `zypper refresh` ausführen.

Bei Paketquellen, deren Inhalt sich nicht ändert, können Sie die regelmäßige Aktualisierung durch die Option `-n` (*no refresh*) vermeiden.

Während Sie eine Paketquelle einrichten, haben Sie die Möglichkeit, den Schlüssel zu importieren, mit dem die Pakete signiert sind. Sofern Sie von der Authentizität der Paketquelle überzeugt sind, sollten Sie den diesbezüglichen Rückfragen zustimmen.

`zypper addrepo` bietet keine Möglichkeit, den Namen einer Paketquelle anzugeben. `zypper repos` zeigt in der Namensspalte deswegen nochmals den Alias an. Wenn Sie Namen und Alias getrennt einstellen möchten, müssen Sie die Datei zur Beschreibung der Paketquelle im Verzeichnis `/etc/zypp/repos.d/` direkt ändern.

- `clean [--all]`  
löscht heruntergeladene Pakete aus dem Cache. Mit der Option `--all` wird auch der Metadaten-Cache gelöscht, also die Beschreibung der in den Repositories verfügbaren Pakete. Diese Daten müssen vor der Installation des nächsten Pakets wieder heruntergeladen werden müssen.
- `dup`  
führt ein Distributions-Update durch, also beispielsweise von openSUSE 15.0 auf Version 15.1. Vorher müssen Sie ein reguläres Update durchführen und dann die Paketrepositories auf die neue

Version umstellen. Nach dem Update ist ein Neustart erforderlich.

- `info name`  
liefert Informationen zum angegebenen Paket/Patch/Produkt etc.
- `install [-y] name`  
installiert das angegebene Paket. Statt eines Pakets darf auch der Name eines Produkts, eines Patches oder einer Sprache angegeben werden. Um alle Pakete eines vordefinierten Schemas (Patterns) zu installieren, führen Sie `zypper install -t pattern name` aus. Mit der zusätzlichen Option `-y` verzichtet `zypper` auf Rückfragen.
- `lifecycle`  
liefert eine tabellarische Übersicht, wie lange die installierten Pakete noch mit Updates versorgt werden und für welche Pakete dies nicht mehr der Fall ist.
- `list-updates [-t package]`  
zeigt alle verfügbaren Patches an, mit `-t package` sowohl Patches als auch Updates.
- `patches`  
listet alle verfügbaren Patches auf.
- `patch-check`  
ermittelt die Anzahl aller verfügbaren Patches.
- `patch-info [name]`  
liefert Informationen zu allen Patches bzw. zum angegebenen Patch.
- `refresh`  
testet, ob sich die Inhaltsverzeichnisse der Paketquellen geändert haben, und liest diese gegebenenfalls neu ein – auch bei solchen Paketquellen, die mit der Option `-n (no refresh)` eingerichtet wurden. Mit der Option `-f (force)` erreichen Sie eine Neuerfassung

des Inhaltsverzeichnisses selbst dann, wenn zypper glaubt, dass sich nichts geändert hat.

- `remove [-y] name`

entfernt das angegebene Paket. Mit der zusätzlichen Option `--clean-deps` werden auch abhängige Pakete deinstalliert, die nun nicht mehr benötigt werden. Mit `-y` verzichtet zypper auf Rückfragen.

- `removerrepo name`

entfernt die durch den URI oder Alias angegebene Paketquelle.

- `renamerepo old new`

gibt der Paketquelle einen neuen Alias-Namen.

- `repos`

listet alle eingerichteten Paketquellen auf. Das Kommando zeigt dabei den Namen und den Alias jeder Paketquelle an, nicht aber die Adresse (*uri*). Bei mit YaST eingerichteten Paketquellen wird die Adresse als Alias verwendet, weswegen die Alias-Spalte oft wie eine Adressspalte aussieht. Für Paketquellen, die mit zypper `addrepo` eingerichtet werden, gilt dies aber nicht.

- `search [optionen] ausdruck`

sucht nach Paketen mit dem angegebenen Suchausdruck.

Standardmäßig werden nur die Paketnamen durchsucht. Mit `-d` durchsucht zypper auch die Paketbeschreibung. `-i` schränkt die Suche auf bereits installierte Pakete ein, `-u` auf noch nicht installierte Pakete. Mit `-t pattern` ermitteln Sie eine Liste vordefinierter Paketgruppen.

- `update [-t package] [-y]`

aktualisiert alle Pakete, zu denen Patches verfügbar sind. Mit der Option `-t package` berücksichtigt zypper auch Updates, also neuere Programmversionen. Dazu eine kurze Erläuterung: Updates sind komplette Pakete, die in einer neueren Version als der installierten zur Verfügung stehen. Patches sind dagegen

Ergänzungs- bzw. Aktualisierungspakete (Delta-RPMs), die nur die Änderungen enthalten und somit wesentlich kleiner sind.

## Beispiel

Das erste Kommando aktualisiert alle Paketquellen Quellen, das zweite installiert den Editor nano, und das dritte stellt fest, welche Updates zur Verfügung stehen.

```
root# zypper refresh
```

```
All repositories have been refreshed.
```

```
root# zypper install nano
```

```
Reading installed packages...
```

```
The following NEW package is going to be installed: nano
```

```
Overall download size: 335.0 K. After the operation, additional 1.2 M  
will be used.
```

```
Continue? [YES/no]: yes
```

```
root# zypper list-updates
```

```
...
```

Repository:	Name	Version	Category	Status
Haupt-Update Repos	MozillaFirefox	4572-0	security	Needed
Haupt-Update Repos	NetworkManager	4548-0	recommended	Needed

```
...
```

### #&%! (bash-Sonderzeichen)

Sowohl bei der Eingabe von Kommandos als auch bei der Shell-Programmierung können Sie eine unüberschaubare Fülle von Sonderzeichen für diverse Aktionen verwenden. [Tabelle 3](#) fasst die Bedeutung der wichtigsten Sonderzeichen zusammen.

Zeichen	Bedeutung
;	trennt mehrere Kommandos.
:	Shell-Kommando, das nichts tut
.	Shell-Programm ohne eigene Subshell starten (. datei, entspricht source datei)
#	leitet einen Kommentar ein.
#!/bin/sh	identifiziert die gewünschte Shell für das Shell-Programm.
&	führt das Kommando im Hintergrund aus (kom &).

Zeichen	Bedeutung
&&	bedingte Kommandoausführung (kom1 && kom2)
&>	Umleitung von Standardausgabe und -fehler (entspricht >&)
	bildet Pipes (kom1   kom2).
	bedingte Komandoausführung (kom1    kom2)
*	Jokerzeichen für Dateinamen (beliebig viele Zeichen)
?	Jokerzeichen für Dateinamen (ein beliebiges Zeichen)
[abc]	Jokerzeichen für Dateinamen (ein Zeichen aus abc)
[ ausdruck ]	Kurzschreibweise für test ausdruck
[ [ ausdr ] ]	erweiterte <a href="#">test</a> -Syntax, bash-spezifisch
( . . . )	Kommandos in derselben Shell ausführen ((kom1; kom2))
{ . . . }	Kommandos gruppieren
{ , , }	Zeichenketten zusammensetzen (a{1,2,3} → a1 a2 a3)
{a..b}	Zeichenketten zusammensetzen (b{4..6} → b4 b5 b6)
~	Abkürzung für das Heimatverzeichnis
>	Ausgabeumleitung in eine Datei (kom > dat)
>>	Ausgabeumleitung; an vorhandene Datei anhängen
>&	Umleitung von Standardausgabe und -fehler (entspricht >&)
2>	Umleitung der Standardfehlerausgabe

Zeichen	Bedeutung
<	Eingabeumleitung aus einer Datei (kom < dat)
<< ende	Eingabeumleitung aus der aktiven Datei bis zu ende
\$	Kennzeichnung von Variablen (echo \$var)
\$!	PID des zuletzt gestarteten Hintergrundprozesses
\$\$	PID der aktuellen Shell
\$0	Dateiname des gerade ausgeführten Shell-Scripts
\$1 bis \$9	die ersten neun dem Kommando übergebenen Parameter
\$#	die Anzahl der dem Shell-Programm übergebenen Parameter
\$* oder \$@	die Gesamtheit aller übergebenen Parameter
\$?	Rückgabewert des letzten Kommandos (0 = OK oder Fehlernummer)
\$(...)	Kommandosubstitution (echo \$(ls))
\${...}	diverse Spezialfunktionen zur Bearbeitung von Zeichenketten
\$[ ... ]	arithmetische Auswertung (echo \${2+3})
"..."	Auswertung der meisten Sonderzeichen verhindern
'...'	Auswertung aller Sonderzeichen verhindern
`...`	Kommandosubstitution (echo `ls`, entspricht echo \$(ls))
\zeichen	hebt die Wirkung des Sonderzeichens auf.

**Tabelle 3** bash-Sonderzeichen

# Konfigurationsdateien

Dieser Abschnitt der *Linux Kommandoreferenz* fasst die Syntax der wichtigsten Linux-Konfigurationsdateien zusammen. Die Dateien sind nach ihrem Namen geordnet, nicht nach dem Pfad des Verzeichnisses, in dem sie sich befinden.

Die meisten der hier präsentierten Dateien gelten für alle Distributionen. Auf distributionsspezifische Eigenheiten weise ich explizit hin. Weitere Details zu Konfigurationsdateien erhalten Sie mit `man 5 name`. Dabei gibt die Zahl 5 an, dass Sie die Dokumentation einer Konfigurationsdatei nachlesen möchten, nicht die eines womöglich gleichnamigen Kommandos.

## /etc/adduser.conf

Die Debian/Ubuntu-spezifische Datei `/etc/adduser.conf` enthält Defaulteinstellungen für das Einrichten neuer Benutzer und Gruppen durch die Kommandos [adduser](#) und [addgroup](#). Die Datei `adduser.conf` enthält zeilenweise Einstellungen in der Form `parameter=wert`.

- `ADD_EXTRA_GROUPS=0|1`  
gibt an, ob neuen Benutzern automatisch die in `EXTRA_GROUPS` aufgezählten sekundären Gruppen zugeordnet werden sollen.
- `DHOME`  
gibt das Basisverzeichnis für die Heimatverzeichnisse an (üblicherweise `/home`).
- `DIRMODE`  
gibt an, mit welchen Zugriffsbits neue Heimatverzeichnisse eingerichtet werden sollen (standardmäßig 755).
- `DSHELL`  
gibt die Default-Shell an.
- `EXTRA_GROUPS="gruppe1,gruppe2,..."`  
enthält eine Liste von sekundären Gruppen, denen neue Benutzer

standardmäßig (`ADD_EXTRA_GROUPS=1`) oder durch die [adduser](#)-Option `--add_extra_groups` zugeordnet werden.

- `FIRST_GID` und `LAST_GID`  
bestimmt den Bereich der GID-Nummern für gewöhnliche Gruppen.
- `FIRST_SYSTEM_GID` und `LAST_SYSTEM_GID`  
bestimmt den Bereich der GID-Nummern für die Gruppen von System-Accounts.
- `FIRST_SYSTEM_UID` und `LAST_SYSTEM_UID`  
bestimmt den Bereich der UID-Nummern für System-Accounts.
- `FIRST_UID` und `LAST_UID`  
bestimmt den Bereich der UID-Nummern für gewöhnliche Benutzer.
- `GROUPHOMES=yes|no`  
legt fest, ob die Heimatverzeichnisse von neuen Benutzern am Ort `/home/gruppenname/benutzername` eingerichtet werden sollen.
- `LETTERHOMES=yes|no`  
bestimmt, ob Heimatverzeichnisse in Unterverzeichnisse mit dem Anfangsbuchstaben platziert werden sollen, also z.B. `/home/k/kofler`. Das kann bei Rechnern mit sehr vielen Accounts zweckmäßig sein.
- `NAME_REGEX=yes|no`  
enthält einen regulären Ausdruck (siehe [grep](#)), dem neue Benutzernamen entsprechen müssen.
- `SKEL`  
gibt an, aus welchem Verzeichnis Defaultdateien in ein neues Benutzerverzeichnis kopiert werden sollen (in der Regel `/etc/skel`).
- `USERGROUPS=yes|no`  
gibt an, ob für jeden Benutzer eine eigene Gruppe eingerichtet werden soll.

## Beispiel

Die folgenden Zeilen zeigen auszugsweise die unter Ubuntu gültigen Defaulteinstellungen in *adduser.conf*:

```
DSHELL=/bin/bash
DHOME=/home
GROUPHOMES=no
LETTERHOMES=no
SKEL=/etc/skel
FIRST_SYSTEM_UID=100
LAST_SYSTEM_UID=999
FIRST_SYSTEM_GID=100
LAST_SYSTEM_GID=999
FIRST_UID=1000
LAST_UID=29999
FIRST_GID=1000
LAST_GID=29999
USERGROUPS=yes
USERS_GID=100
DIR_MODE=0755
...
```

### /etc/aliases

Die Datei */etc/aliases* listet für lokale E-Mail-Accounts Weiterleitungsadressen auf. Für die Auswertung der Datei ist der E-Mail-Server verantwortlich, d.h., die Datei ist nur dann relevant, wenn auf dem Rechner ein Mail Transfer Agent (MTA) läuft, z.B. *sendmail* oder *postfix*.

Im Regelfall ist *aliases* primär für lokale Weiterleitungen gedacht, beispielsweise um E-Mails von *webmaster@hostname* an *adminxy@hostname* weiterzuleiten. Syntaktisch ist es auch zulässig, lokale Mails an externe Hosts weiterzuleiten; in der Praxis scheitert dies aber oft an Spam-Schutzmaßnahmen der externen Hosts. Viele Mail-Server werten */etc/aliases* nicht direkt aus, sondern berücksichtigen stattdessen eine Datenbankdatei, die aus der *aliases*-Datei generiert wird. Deswegen werden Änderungen in der *aliases*-Datei oft erst wirksam, nachdem das Kommando *newaliases* ausgeführt wird.

*/etc/aliases* enthält zeilenweise die Umleitungsregeln:

```
name: alias1, alias2, alias3 ...
```

Dabei muss *name* ein lokaler Mail-Account sein (ohne *@hostname!*).

Zur Angabe des Alias gibt es mehrere Möglichkeiten, wobei es vom

Mail-Server abhängt, welche Varianten unterstützt werden.

- name: Die E-Mail wird an einen anderen lokalen Mail-Account weitergeleitet.
- adresse@host: Die E-Mail wird an eine externe E-Mail-Adresse weitergeleitet.
- /pfad/datei: Die E-Mail wird am Ende der angegebenen Datei hinzugefügt.
- | kommando: Die E-Mail wird an das angegebene Kommando übergeben.
- :include:/pfad/datei: Die [alias](#)-Liste wird aus der angegebenen Datei gelesen.

Wenn mehrere Aliase angegeben werden, wird die E-Mail an alle Adressen weitergegeben.

### Beispiel

Im folgenden Beispiel werden E-Mails, die an `root`, `postmaster` und `webmaster` gerichtet sind, an die lokalen Benutzer `michael` und `bernd` weitergeleitet. E-Mails an `support` werden sowohl im support-Mailfach gespeichert als auch an `michael` sowie an `peter.proell@firma-abc.de` weitergeleitet.

```
root:          michael
postmaster:    michael
webmaster:    michael, bernd
support:       support, michael, peter.proell@firma-abc.de
```

### /etc/bashrc

`/etc/bashrc` enthält Defaulteinstellungen für die `bash`. Die Datei wird als `bash`-Script ausgeführt. Es sind daher alle von der `bash` unterstützten Sprachkonstrukte erlaubt. Üblicherweise wird in `bashrc` die Prompt-Umgebungsvariable `PS1` eingestellt, bei manchen Distributionen auch der [umask](#)-Wert.

Die Einstellungen in `/etc/bashrc` werden durch jene in `/etc/profile` ergänzt. `/etc/profile` gilt für alle Shells, `bashrc` enthält nur die Einstellungen, die spezifisch für die `bash` gelten sollen. Zudem werden benutzerspezifische Einstellungen in `.bashrc` berücksichtigt.

## /boot/config.txt

*config.txt* ist eine Raspberry-Pi-spezifische Datei zur Konfiguration der Hardware. Sie muss sich zusammen mit anderen Bootdateien in der ersten Partition der SD-Karte befinden. Unter Raspberry Pi OS ist die Datei im Verzeichnis */boot* zugänglich. Die Datei wird während des Bootprozesses noch vor dem Start des Kernels direkt durch den SoC (System-on-a-Chip) des Raspberry Pi ausgewertet.

- `arm_freq=n`  
gibt die gewünschte maximale CPU-Frequenz in MHz an. Beim Raspberry Pi 3B+ beträgt die Frequenz standardmäßig 1,4 GHz, wobei die Frequenz automatisch reduziert wird, wenn die CPU wenig zu tun hat. Anders als bei älteren Modellen ist ein *Overclocking*, also die Einstellung höherer Frequenzen, nicht empfehlenswert!
- `cmdline=zeichenkette`  
übergibt die Zeichenkette an den Kernel. Der Parameter wird anstelle der Datei */boot/cmdline.txt* ausgewertet.
- `config_hdmi_boost=n`  
steuert die HDMI-Signalstärke (0 ist der normale Pegel, 7 ist der maximale Pegel).
- `core_freq=n`  
gibt die Taktfrequenz der Grafik- und Encoding-Codes in MHz an.
- `decode MPG2=0x12345678`  
`decode WVC1=0x12345678`  
gibt Codes an, um den MPG-2- und den VC-1-Decoder freizuschalten. Die Schlüssel können unter <https://www.raspberrypi.com> erworben werden und müssen zur Seriennummer der CPU passen. Die Seriennummer geht aus der Datei */proc/cpuinfo* hervor.
- `disable_camera_led=1`  
deaktiviert die Leuchtdiode, die die Aktivität der Raspberry-Pi-Kamera anzeigt.

- `disable_overscan=1`  
deaktiviert die Overscan-Funktion. Solange diese aktiv ist, umgibt das Grafiksystem des Raspberry Pi das eigentliche Bild mit einem schwarzen Rand. Das erhöht die Kompatibilität zu manchen HDMI-Monitoren, führt bei anderen Modellen aber zu einem schwarzen »Trauerrand«. Abhilfe schafft dann `disable_overscan=1`. (Wenn die Overscan-Funktion erwünscht ist, können Sie mit den Parametern `overscan_left`, `-right`, `-top` und `-bottom` die Overscan-Breite auf allen vier Seiten einstellen.)
- `display_rotate=1|2|3`  
rotiert die Display-Darstellung um 90, 180 bzw. 270 Grad.
- `dtoverlay=xxx`  
aktiviert den Device-Tree-Overlay für eine Hardware-Komponente und lädt die entsprechenden Treibermodule.
- `dtparam=audio=on`  
aktiviert das Audio-System, lädt den Treiber `snd_bcm28xx`.
- `dtparam=i2c_arm=on`  
aktiviert den Bus I<sup>2</sup>C.
- `dtparam=spi=on`  
aktiviert den SPI-Bus.
- `gpu_mem=n`  
gibt an, wie viel RAM dem Grafiksystem zugewiesen wird. RAM wird zwischen der CPU und dem Grafikprozessor geteilt. Die Aufteilung muss beim Start endgültig festgelegt werden, wobei Sie dem Grafiksystem 16, 64, 128 oder 256 MiB zuweisen können. 16 MiB sind für den normalen Betrieb ausreichend. Für grafikintensive Anwendungen (3D-Grafik, HD-Filme abspielen etc.) benötigt das Grafiksystem hingegen 128 MiB Speicher. Auch die Nutzung der Kamera erfordert ein Minimum von 128 MiB Grafikspeicher.

- `hdmi_force_hotplug=1`  
aktiviert den HDMI-Ausgang auch dann, wenn kein Monitor erkannt wird.
- `hdmi_group=1|2`  
gibt an, wie die `hdmi_mode`-Einstellung zu interpretieren ist: 1 bedeutet CEA (*Consumer Electronics Association*, Modi für TV-Geräte), 2 gilt für DMT (*Display Monitor Timings* für Computer-Monitore).
- `hdmi_mode=n`  
gibt den gewünschten Display-Modus an. Dabei wird auch `hdmi_group` berücksichtigt. Die Liste der möglichen Modi können Sie mit `vservice -m CEA` bzw. `vservice -m DMT` feststellen.
- `sdram_freq=n`  
gibt die Taktfrequenz für den Arbeitsspeicher (RAM) im MHz an.
- `start_x=1`  
aktiviert das Kameramodul.

Veränderungen an `config.txt` werden erst mit dem nächsten Neustart wirksam. Viele `config.txt`-Parameter können Sie im laufenden Betrieb mit [`vcgencmd get\_config`](#) auslesen. Noch mehr `config.txt`-Details können Sie hier nachlesen:

[`https://elinux.org/RPiConfig`](https://elinux.org/RPiConfig)

### **/etc/crontab**

Die Datei `/etc/crontab` wird vom Hintergrundprozess `crond` ausgewertet. `crontab` enthält zeilenweise Informationen darüber, wann welche Kommandos für welchen Benutzer-Account ausgeführt werden sollen. Das Cron-System ermöglicht es so, regelmäßig zu vorgegebenen Zeitpunkten Aufgaben automatisch durchzuführen, z.B. an jedem ersten Montag im Januar, April, Juli und Oktober um 2:30 eine Spiegelung des Dateisystems auf einen externen Backup-Server zu starten.

Jeder *crontab*-Eintrag besteht aus sieben Spalten in einer Zeile. Die Spalten sind durch Leer- oder Tabulatorzeichen voneinander getrennt. Die letzte Zeile *muss* mit einem Zeilenumbruchzeichen abgeschlossen werden, andernfalls wird sie ignoriert. Kommentare werden mit dem Zeichen # am Zeilenanfang eingeleitet. Es ist nicht zulässig, einen Kommentar im Anschluss an einen *crontab*-Eintrag anzugeben.

- Erste Spalte (Minute): gibt an, in welcher Minute (0–59) das Programm ausgeführt werden soll.
- Zweite Spalte (Stunde): gibt die gewünschte Stunde an (0–23).
- Dritte Spalte (Tag): gibt den Tag im Monat an (1–31).
- Vierte Spalte (Monat): gibt den Monat an (1–12).
- Fünfte Spalte (Wochentag): gibt den Wochentag an (0–7, 0 und 7 bedeuten jeweils Sonntag).
- Sechste Spalte (Benutzer/Account): gibt an, für welchen Benutzer das Kommando ausgeführt wird (oft root).
- Siebte Spalte (Kommando): enthält das auszuführende Kommando. Das Kommando darf Leerzeichen enthalten; der gesamte Text bis zum Ende der Zeile wird als Kommando interpretiert.

Wenn in den ersten fünf Spalten statt einer Zahl ein \* angegeben wird, wird dieses Feld ignoriert. 15 \* \* \* \* bedeutet beispielsweise, dass das Kommando immer 15 Minuten nach der ganzen Stunde ausgeführt werden soll, in jeder Stunde, an jedem Tag, in jedem Monat, unabhängig vom Wochentag. 29 0 \* \* 6 bedeutet, dass das Kommando an jedem Samstag um 0:29 Uhr ausgeführt wird.

Für die Zeitfelder ist auch die Schreibweise \*/*n* erlaubt. Das bedeutet, dass das Kommando jede *n*-te Minute/Stunde etc. ausgeführt wird. \*/15 \* \* \* \* würde also bedeuten, dass das Kommando viertelstündlich (n:00, n:15, n:30 und n:45) ausgeführt wird.

Bei den Zeitfeldern sind zudem durch Kommata getrennte Mehrfachangaben sowie Von-bis-Bereiche zulässig. 1,13 in der

zweiten Spalte bedeutet, dass das Kommando um 1 Uhr morgens sowie um 13 Uhr ausgeführt werden soll. 2, 4, 6, 8-18, 20, 23 in der zweiten Spalte bewirkt, dass das Kommando um 2 Uhr, 4 Uhr, 6 Uhr, 20 Uhr, 23 Uhr sowie stündlich zwischen 6 und 18 Uhr ausgeführt wird.

Anstelle der fünf Zeitspalten dürfen auch die in [Tabelle 1](#) zusammengefassten @-Kürzel verwendet werden. Eine weitere Zusatzregel besagt, dass ein Minuszeichen am Beginn der ersten Spalte verhindert, dass Syslog die Kommandoausführung protokolliert. Das ist allerdings nur erlaubt, wenn die sechste Spalte root enthält.

Kürzel	Code	Bedeutung
@reboot	–	nach jedem Reboot ausführen
@yearly	0 0 1 1 *	einmal im Jahr ausführen
@annually	0 0 1 1 *	wie @yearly
@monthly	0 0 1 * *	einmal pro Monat ausführen
@weekly	0 0 * * 0	einmal pro Woche ausführen
@daily	0 0 * * *	einmal pro Tag ausführen
@hourly	0 * * * *	einmal pro Stunde ausführen

**Tabelle 1** crontab-Intervallkürzel ersetzen die ersten fünf Spalten.

### SHELL, PATH und MAIL

Vor den gerade beschriebenen Zeilen zum periodischen Start von Programmen enthält */etc/crontab* üblicherweise drei Zeilen zur Einstellung von Umgebungsvariablen. Unter CentOS sehen die Defaulteinstellungen so aus:

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
```

Das bedeutet, dass die Kommandozeichenkette durch die `bash` ausgeführt wird, dass bei der Suche nach Kommandos die `PATH`-

Verzeichnisse berücksichtigt werden und dass die E-Mails mit Ausgaben oder Fehlermeldungen an `root` gesendet werden.

## Beispiel

Die folgenden drei Zeilen in `/etc/crontab` bewirken, dass 15 Minuten nach jeder vollen Stunde ein Maintenance-Script aufgerufen wird, dass jeden Sonntag um 0:30Uhr ein Backup-Werkzeug gestartet wird und dass täglich um 3:15Uhr eine Sicherheitskopie einer MySQL-Datenbank erstellt wird:

```
15 * * * * root /usr/bin/maintenance-script
30 0 * * 0 root /usr/bin/backup-tool
15 3 * * * mysql /usr/bin/mysql-backup
```

## Weitere Cron-Dateien

Außer mit der systemweiten Datei `/etc/crontab` haben Sie je nach Distribution verschiedene weitere Möglichkeiten, Cron-Jobs zu definieren:

- `/etc/cron.d`: Ergänzend zu `/etc/crontab` wertet der Cron-Dämon `crond` bei den meisten Distributionen auch alle Dateien im Verzeichnis `/etc/cron.d` aus. Für diese Dateien gelten dieselben Syntaxregeln wie für `/etc/crontab`.
- Benutzerspezifische crontab-Dateien: Je nach Distribution werden im Verzeichnis `/var/spool/cron/` bzw. `/var/spool/cron/tabs` benutzerspezifische Crontab-Dateien gespeichert. Der Name der Datei gibt an, für welchen Benutzer die Datei gilt. Dafür entfällt in diesen Dateien die sechste Spalte. Zur Veränderung benutzerspezifischer Cron-Einträge ist das Kommando `crontab` vorgesehen.
- `cron.hourly`, `cron.daily`, `cron.weekly` und `cron.monthly`: Bei den meisten Distributionen enthält die Defaultkonfiguration in `/etc/crontab` einige Einträge, die bewirken, dass einmal pro Stunde alle Script-Dateien in `/etc/cron.hourly/*` mit `root`-Rechten ausgeführt werden und einmal pro Tag die Script-Dateien in `/etc/cron.daily/*` etc. Bei diesen Dateien handelt es sich um ganz gewöhnliche Script-Dateien, für die keine Crontab-Syntaxregeln

gelten. Denken Sie aber daran, das *execute*-Bit zu setzen (`chmod a+x datei`)!

Die Dateinamen eigener Scripts in *cron.daily*, *cron.weekly* und *cron.monthly* dürfen ausschließlich aus Zahlen, Buchstaben und Binde- und Unterstrichen bestehen. Sobald der Dateiname auch nur einen Punkt enthält, wird das Script ignoriert! Die Scripts aus */etc/cron.daily*, *-weekly* und *-monthly* werden nicht ausgeführt, wenn das Programm Anacron installiert ist.

#### **/etc/deluser.conf**

Die Debian/Ubuntu-spezifische Datei */etc/deluser.conf* enthält Defaulteinstellungen für das Löschen von Benutzern und Gruppen durch die Kommandos [deluser](#) und [delgroup](#). Die Datei *deluser.conf* besteht aus zeilenweisen Einträgen in der Form *parameter=wert*.

- **BACKUP=0|1**  
gibt an, ob Backups von den Dateien gemacht werden sollen, die durch *REMOVE*-Parameter gelöscht werden.
- **BACKUP\_TO**  
gibt den Ort des Backup-Verzeichnisses an.
- **EXCLUDE\_FSTYPES="(fs1, fs2, ...)"**  
gibt an, welche Dateisysteme ignoriert werden sollen, wenn nach Dateien des Benutzers gesucht wird, um ein Backup zu erstellen.
- **ONLY\_IS\_EMPTY=1**  
bewirkt, dass Gruppen nur gelöscht werden, wenn sie keine Mitglieder haben.
- **REMOVE\_HOME=0|1**  
gibt an, ob mit dem Account auch dessen Heimatverzeichnis gelöscht werden soll.
- **REMOVE\_ALL\_FILES=0|1**  
gibt an, ob Benutzerdateien außerhalb des Heimatverzeichnisses gelöscht werden sollen, z.B. Spooling-Dateien und die E-Mail-Inbox.

## Beispiel

Die folgenden Zeilen zeigen die unter Ubuntu gültigen Defaulteinstellungen in *deluser.conf*:

```
REMOVE_HOME = 0
REMOVE_ALL_FILES = 0
BACKUP = 0
BACKUP_TO = "."
ONLY_IF_EMPTY = 0
EXCLUDE_FSTYPES = "(proc|sysfs|usbfs|devpts|tmpfs|afs)"
```

### /etc/dhcpcd.conf

Das Programm *dhcpcd.conf* ist ein DHCP-Client. Unter Raspberry Pi OS spielt dieses Programm eine zentrale Rolle für die gesamte Netzwerkkonfiguration des Raspberry Pi. Es übernimmt standardmäßig die Kontrolle über alle Netzwerkschnittstellen. Für WLAN-Verbindungen startet *dhcpcd* das Programm *wpa\_supplicant* (siehe auch die Beschreibung der Konfigurationsdatei */etc/wpa\_supplicant/wpa\_supplicant.conf*).

Die Defaultkonfiguration von */etc/dhcpcd.conf* sieht wie folgt aus:

```
# /etc/dhcpcd.conf unter Raspberry Pi OS
hostname
clientid
persistent
option rapid_commit
option domain_name_servers, domain_name, domain_search, host_name
option classless_static_routes
option ntp_servers
option interface_mtu
require dhcp_server_identifier
slaac private
```

Obwohl es der Natur eines DHCP-Clients widerspricht, kann in *dhcpcd.conf* auch eine statische Konfiguration durchgeführt werden. Dabei orientieren Sie sich nach dem folgenden Muster:

```
# statische Konfiguration von eth0
interface eth0
static ip_address=192.168.0.10/24
static ip6_address=fd51:42f8:caae:d92e::ff/64
static routers=192.168.0.1
static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1
```

Die Netzwerkkonfiguration in */etc/network/interfaces* ist nach wie vor zulässig, d.h., das ENI-Framework steht in Raspberry Pi OS weiterhin zur Verfügung. Die betreffende Schnittstelle muss dann

aber in *dhpcd.conf* deaktiviert werden. An *denyinterfaces* dürfen auch mehrere Schnittstellen übergeben werden, wahlweise getrennt durch Leerzeichen oder Kommata.

```
# dhpcd soll wlan0 ignorieren, diese Schnittstelle wird durch ENI verwaltet
denyinterfaces wlan0
```

### /etc/dnf/dnf.conf

Die Grundeinstellungen des unter Fedora und Red Hat üblichen Paketverwaltungssystems DNF befinden sich in */etc/dnf/dnf.conf*. Außerdem enthalten die Dateien */etc/yum.repos.d/\*.repos* Detaileinstellungen für die einzelnen Paketquellen.

## Grundeinstellungen

- `clean_requirements_on_remove = true/false`  
gibt an, ob Pakete, die aufgrund von Abhängigkeiten installiert wurden, automatisch wieder entfernt werden sollen, sobald alle Pakete deinstalliert werden, die auf das Paket verweisen. Auch wenn diese Option `true` lautet, entfernt [dnf](#) nie ungefragt Pakete, die explizit mit `dnf install` installiert wurden.
- `color=never`  
bewirkt, dass das Kommando [dnf](#) im Terminal keine Farben nutzt.  
Wenn Sie das möchten, verwenden Sie `color=always`.
- `exactarch=1`  
bewirkt, dass [dnf](#) nur Updates berücksichtigt, bei denen die Architektur mit dem bereits installierten Paket übereinstimmt.  
i386-Pakete können also nicht durch neuere x86\_64-Pakete ersetzt werden.
- `gpgcheck=1`  
bewirkt, dass [dnf](#) mit einem Schlüssel die Authentizität der Pakete sicherstellt. `gpgcheck` kann abweichend von der Einstellung in *dnf.conf* auch individuell für jede Paketquelle eingestellt werden. `plugins` entscheidet, ob [dnf](#) Plug-ins berücksichtigt.
- `installonly_limit=n`  
steuert, wie viele alte Versionen von Kernelpaketen gleichzeitig

installiert werden. Die Einstellung betrifft alle Pakete, zu denen Updates in Form von neuen Paketen installiert werden. Diese Vorgehensweise stellt sicher, dass immer auch ältere Versionen des betroffenen Pakets als Fallback zur Verfügung stehen.

- `keepcache=0`

bewirkt, dass heruntergeladene Pakete nach der Installation nicht archiviert werden. In der Regel ist das eine zweckmäßige Einstellung, weil der Platzbedarf für die Pakete im Laufe der Zeit recht groß wird und normalerweise kein Grund besteht, die Pakete ein zweites Mal zu installieren. Es kann allerdings passieren, dass [dnf](#) während der Installation ein Problem feststellt und die Installation abbricht. Auch in diesem Fall werden die heruntergeladenen Pakete gelöscht. Wenn Sie das Problem beheben können und anschließend das Update wiederholen, müssen alle Pakete neuerlich heruntergeladen werden. Diese Situation vermeiden Sie mit `keepcache=1`. Um die heruntergeladenen Pakete in `/var/cache/dnf` explizit zu löschen, führen Sie `dnf clean packages` aus.

In aktuellen Fedora-Versionen enthält die Datei nur wenige Einstellungen:

```
# Datei /etc/dnf/dnf.conf unter Fedora
[main]
gpgcheck=1
installonly_limit=3
clean_requirements_on_remove=True
best=False
skip_if_unavailable=True
```

## Paketquellen

Die Konfiguration der Paketquellen erfolgt durch \*.conf-Dateien im Verzeichnis `/etc/yum.repos.d`. Von einer Umbenennung des Verzeichnisses in `/etc/dnf/repos.d` hat man beim Wechsel von [yum](#) zu [dnf](#) abgesehen.

- `[repovariante]`

gibt an, auf welchen Teilbereich der Paketquelle sich die

folgenden Einstellungen beziehen.

- `baseurl=url` oder `mirrorlist=url`  
gibt an, wo sich die Paketdateien befinden. Bei `mirrorlist` verweist die `url` auf eine Liste von Mirror-Servern. [dnf](#) entscheidet sich selbstständig für einen der Mirrors. [dnf](#) ersetzt in der `url` die Variablen `$releasever`, `$arch` und `$basearch` durch die Versionsnummer der Linux-Distribution und deren Architektur.
- `enabled=0/1`  
gibt an, ob die Paketquelle aktiv ist.
- `name=reponame`  
gibt den Namen der Paketquelle an.
- `metadata_expires`  
steuert, wie lange die von einer Paketquelle heruntergeladenen Metadaten gültig sind. [dnf](#) speichert die Metadaten in einem Cache und verzichtet auf ein neuerliches Herunterladen, wenn die Metadaten noch nicht veraltet sind. Das spart Zeit und Download-Volumen, kann aber dazu führen, dass [dnf](#) kürzlich durchgeführte Änderungen in der Paketquelle ignoriert.  
Gegebenenfalls erzwingen Sie durch `dnf clean metadata` das Löschen der lokalen Metadaten. Damit ist [dnf](#) beim nächsten Mal gezwungen, die Metadaten aller Paketquellen neu einzulesen.

Die folgenden Zeilen zeigen einen Ausschnitt aus der Datei `/etc/yum.repos.d/#fedora.repo`. Diese Datei enthält die Definition der Paketquelle mit den Basispaketen der Fedora-Distribution:

```
[fedora]
name=Fedora $releasever - $basearch
metalink=https://mirrors.fedoraproject.org/metalink?
repo=fedora-$releasever&arch=$basearch
enabled=1
countme=1
metadata_expire=7d
repo_gpgcheck=0
type=rpm
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$releasever-$basearch
skip_if_unavailable=False
```

```
[fedora-source]
name=Fedora $releasever - Source
...
```

## /etc/fstab

*/etc/fstab* enthält zeilenweise Einträge für alle Dateisysteme und Swap-Partitionen, die beim Rechnerstart in den Verzeichnisbaum integriert bzw. aktiviert werden sollen. Jeder Eintrag besteht aus sechs Spalten, die durch Leer- oder Tabulatorzeichen voneinander getrennt sind. Kommentare werden mit dem Zeichen # eingeleitet.

- Die erste Spalte (Device) enthält den Device-Namen des Datenträgers. Statt des Device-Namens können Sie mit `LABEL=xxx` oder mit `UUID=xxx` auch den Namen bzw. die ID-Nummer des Dateisystems angeben, oder mit `PARTUUID=xxx` die ID-Nummer der Partition.
- Die zweite Spalte (Pfad) gibt an, bei welchem Verzeichnis der Datenträger in den Dateibaum eingebunden wird. Die in der zweiten Spalte angegebenen Verzeichnisse müssen bereits existieren. Bei Swap-Partitionen geben Sie hier das Schlüsselwort `none` an.
- Die dritte Spalte (Dateisystemtyp) gibt das Dateisystem an, z.B. `ext4`, [btrfs](#) oder `vfat`. Einen Überblick über die wichtigsten Linux-Dateisysteme finden Sie bei der Beschreibung des Kommandos [mount](#). Mit dem Eintrag `auto` versucht Linux, das Dateisystem selbst zu erkennen. Bei Swap-Partitionen, die intern kein Dateisystem enthalten, geben Sie das Schlüsselwort `swap` an.
- Die vierte Spalte (Optionen) enthält die gewünschten [mount](#)-Optionen. Eine Referenz der Optionen für alle wichtigen Dateisysteme finden Sie in diesem Buch beim [mount](#)-Kommando. Wenn Sie keine Optionen benötigen, geben Sie stattdessen das Schlüsselwort `defaults` an. Mehrere Optionen werden nur durch Kommata getrennt, nicht durch Leerzeichen!
- Die fünfte Spalte (dump) enthält Informationen für das Unix-Programm `dump`. Unter Linux wird diese Spalte ignoriert. Es ist

üblich, für die Systempartition 1 und für alle anderen Partitionen oder Datenträger 0 einzutragen.

- Die sechste Spalte ([fsck](#)) gibt an, ob und in welcher Reihenfolge die Dateisysteme beim Systemstart überprüft werden sollen. Oft wird 1 für die Systempartition und 0 für alle anderen Partitionen eingetragen. Das bedeutet, dass beim Rechnerstart nur die Systempartition auf Fehler überprüft und gegebenenfalls durch [fsck](#) repariert wird. Falls Sie möchten, dass weitere Partitionen automatisch überprüft werden, geben Sie bei diesen Partitionen die Ziffer 2 an. Das heißt, die Überprüfung soll nach der Kontrolle der Systempartition erfolgen. Wenn Einträge in der fünften und sechsten Spalte in [/etc/fstab](#) fehlen, wird 0 angenommen.

## Beispiel

Das folgende Beispiel zeigt die [fstab](#)-Datei eines Servers mit LVM und RAID. Die Bootpartition befindet sich auf dem RAID-Device [/dev/md/0](#). Die System- und Datenverzeichnisse nutzen Logical Volumes als Speicherort. Die Dateisysteme für [/](#), [/var](#) und [/backup](#) werden beim Neustart überprüft. Sollte im Root-Dateisystem ein Fehler auftreten, wird das Dateisystem im Read-only-Modus eingebunden.

```
/dev/md/0      /boot    ext2    defaults        0  0
/dev/vg0/root   /       ext4    errors=remount-ro  0  1
/dev/vg0/var    /var    ext4    defaults        0  2
/dev/vg0/backup /backup  ext4    defaults        0  2
/dev/vg0/swap   swap    swap    defaults        0  0
```

## [/etc/group](#)

[/etc/group](#) enthält eine Liste aller Gruppennamen mit den zugehörigen Gruppenidentifikationsnummern (GIDs) sowie mit einer Aufzählung aller Benutzer, die der Gruppe angehören. Zur Administration der Gruppen verwenden Sie üblicherweise die Kommandos [groupadd](#), [groupmod](#) und [groupdel](#).

Die zeilenweisen Einträge in der [group](#)-Datei bestehen aus vier Spalten, die jeweils durch Doppelpunkte voneinander getrennt sind:

- Die erste Spalte (Gruppenname) gibt den Namen der Gruppe an.
- Die zweite Spalte (GID) enthielt in älteren Linux-Versionen den Hash-Code des Gruppenpassworts. Aktuelle Linux-Distributionen speichern hier das Zeichen x. Sofern Gruppenpasswörter definiert sind, werden diese in /etc/gshadow gespeichert.
- Die dritte Spalte (GID) enthält die Gruppenidentifikationsnummer. Bei den meisten Distributionen sind GIDs kleiner 1000 für System-Accounts reserviert.
- Die vierte Spalte (Accounts) enthält eine durch Kommata getrennte Liste aller Benutzer, die Mitglied dieser Gruppe sind. Dabei werden allerdings nur sekundäre Mitgliedschaften berücksichtigt. Die primäre Gruppe jedes Benutzers wird hingegen in der vierten Spalte von /etc/passwd gespeichert.

## Beispiel

Die folgenden Zeilen zeigen Ausschnitte aus */etc/group* auf einem Rechner, der als Web- und Entwicklungsserver dient. Jeder System-Account und jeder reguläre Benutzer hat seine eigene Gruppe. Die Mitglieder der `admin`-Gruppe dürfen mit [sudo](#) Administrationsarbeiten erledigen. Die Mitglieder der `devel`-Gruppe haben besondere Zugriffsrechte auf das Versionsverwaltungssystem.

```
root:x:0:
bin:x:1:
admin:x:109:kofler,huber
...
kofler:x:1000:
huber:x:1001:
mueller:x:1002:
devel:x:1023:kofler,huber,mueller,gruber,schmiedt
...
```

## /etc/default/grub

*/etc/default/grub* enthält Defaulteinstellungen für den Linux-Bootloader GRUB 2. Diese Einstellungen werden erst berücksichtigt, wenn die GRUB-Konfigurationsdatei */boot/grub/grub.cfg* neu erzeugt wird – entweder automatisch bei einem Kernel-Update oder

manuell durch das Kommando `update-grub` (Debian, Ubuntu) bzw. `grub-mkconfig` (Fedora, openSUSE).

Die folgende Auflistung beschreibt die wichtigsten Schlüsselwörter in `/etc/default/grub`:

- `GRUB_CMDLINE_LINUX` und `GRUB_CMDLINE_LINUX_DEFAULT` werden von `10_linux` berücksichtigt und geben an, welche Optionen an den Kernel übergeben werden sollen. Die `GRUB_CMDLINE_LINUX`-Optionen gelten für jeden Start; die `GRUB_CMDLINE_LINUX_DEFAULT`-Optionen werden zusätzlich für den Standardstart hinzugefügt, aber nicht für den Recovery Mode.
- `GRUB_DEFAULT` gibt an, welcher GRUB-Menueintrag standardmäßig ausgewählt werden soll. Die Einstellung "saved" bedeutet, dass der zuletzt ausgewählte Menueintrag aktiviert wird. Das funktioniert allerdings nur, wenn sich die GRUB-Dateien in einer gewöhnlichen Partition befinden! Ist dagegen LVM oder RAID im Spiel, kann GRUB nach der Menüauswahl keine Umgebungsvariablen speichern.  
Eine weitere Möglichkeit besteht darin, `GRUB_DEFAULT` die `menuentry`-Zeichenkette des gewünschten Menueintrags zuzuweisen. Dabei müssen Sie aber darauf achten, die Schreibweise exakt einzuhalten.
- `GRUB_DISABLE_LINUX_UUID=true` bewirkt, dass GRUB das Root-Verzeichnis als Device-Namen an den Kernel übergibt. Ohne diese Option übergibt GRUB die UUID-Nummer des Dateisystems. Diese Einstellung gilt nur für den Start der aktiven Distribution (Script `10_linux`), nicht für andere Distributionen.
- `GRUB_DISABLE_RECOVERY=true` verhindert, dass `update-grub` bzw. `grub2-mkconfig` Menueinträge zum Start von Linux im Recovery Mode in `grub.cfg` einbaut.

- **GRUB\_DISTRIBUTOR**  
wird vom Script `/etc/grub.d/10_linux` ausgewertet und gibt den Namen der aktuellen Distribution an.
- **GRUB\_GFXMODE**  
gibt an, in welchem Grafikmodus GRUB laufen soll (standardmäßig in einer Auflösung von 640×480 Pixel).
- **GRUB\_HIDDEN\_TIMEOUT**  
ist dann von Bedeutung, wenn GRUB während der Installation nur eine einzige Linux-Distribution auf Ihrem Rechner erkennt. In diesem Fall gibt `GRUB_HIDDEN_TIMEOUT` an, wie lange der Benutzer Zeit hat, um mit das GRUB-Menü anzuzeigen. Während dieser Wartezeit bleibt der Bildschirm schwarz. Wenn mehrere Betriebssysteme installiert sind, ignoriert GRUB die `GRUB_HIDDEN_TIMEOUT`-#Einstellung und zeigt das Menü an.
- **GRUB\_HIDDEN\_TIMEOUT\_QUIET=true**  
verhindert, dass während der `GRUB_HIDDEN_TIMEOUT`-Wartezeit ein Countdown-Zähler angezeigt wird. Wenn Sie möchten, dass das GRUB-Menü immer angezeigt wird, stellen Sie den Zeilen `GRUB_HIDDEN_TIMEOUT=...` und `GRUB_HIDDEN_TIMEOUT_QUIET=...` jeweils das Kommentarzeichen # voran.
- **GRUB\_TIMEOUT=n**  
gibt an, wie viele Sekunden GRUB auf die Auswahl eines Menüeintrags wartet. Wenn diese Zeit ohne Benutzereingaben verstreicht, startet GRUB das ausgewählte Betriebssystem. Die hier eingestellte Zeit kommt nur zur Geltung, wenn das GRUB-Menü überhaupt erscheint.
- **GRUB\_TERMINAL=console**  
bewirkt, dass GRUB im Textmodus läuft. Dort können allerdings keine Unicode-Zeichen angezeigt werden.

## Beispiel

In Ubuntu enthält die Konfigurationsdatei die folgenden Einstellungen:

```
# Datei /etc/default/grub
GRUB_DEFAULT=0
GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=
```

### /boot/grub/grub.cfg

/boot/grub/grub.cfg bzw. /boot/grub2/grub.cfg ist die Konfigurationsdatei der aktuellen GRUB-Version 2. Diese Datei wird automatisch generiert, direkte Veränderungen sind nicht empfehlenswert. Um Einfluss auf das Verhalten von GRUB zu nehmen, können Sie in /etc/default/grub diverse Defaulteinstellungen verändern oder in /etc/grub.d/\*custom Anweisungen einfügen, die dann direkt in grub.cfg übernommen werden. Um grub.cfg anschließend neu zu erstellen, führen Sie eines der folgenden Kommandos aus:

```
root# update-grub                                (Debian und Ubuntu)
root# grub2-mkconfig -o /boot/grub2/grub.cfg      (Fedora, openSUSE)
```

### /etc/gshadow

Die Datei /etc/gshadow enthält die Hashcodes der Gruppenpasswörter sowie weitere Daten zur Gruppenadministration. Aus Sicherheitsgründen darf nur root diese Datei lesen und verändern.

Die zeilenweisen Einträge in der gshadow-Datei bestehen aus vier Spalten, die jeweils durch Doppelpunkte voneinander getrennt sind:

- Die erste Spalte (Gruppenname) gibt den Namen der Gruppe an.
- Die zweite Spalte (Hash-Code) enthält den Hash-Code des Gruppenpassworts oder einen Stern (kein Passwort). Die Verwendung von Gruppenpasswörtern ist aus Sicherheitsgründen unüblich. Wenn Sie sich doch dafür entscheiden, legen Sie das Gruppenpasswort mit dem

Kommando [gpasswd](#) fest. Jeder, der das Gruppenpasswort kennt, kann nun mit [newgrp](#) in diese Gruppe wechseln, auch wenn er nicht Mitglied der Gruppe ist.

- Die dritte Spalte (Administratoren) enthält eine durch Kommata getrennte Aufzählung von Gruppenadministratoren (Account-Namen). Gruppenadministratoren dürfen das Gruppenpasswort ändern und andere Mitglieder zur Gruppe hinzufügen bzw. wieder aus ihr entfernen.
- Die vierte Spalte (Mitglieder) enthält eine durch Kommata getrennte Aufzählung von Gruppenmitgliedern, die die Gruppe *ohne* Passwortangabe aktivieren dürfen.

Auf vielen Linux-Systemen ist die Datei *gshadow* ungenutzt und sieht dann so aus:

```
root:*:::  
daemon:*:::  
bin:*:::  
sys:*:::  
...
```

### /etc/host.conf

*/etc/host.conf* steuert die Resolver-Bibliothek, die bei Netzwerkoperationen angibt, wie unbekannte Hostnamen aufgelöst werden.

- `multi on|off`  
gibt an, ob die Resolver-Bibliothek für jeden Host alle Einträge (on) oder nur den ersten passenden Eintrag zurückgibt (off).
- `order a, b, c...`  
gibt an, in welcher Reihenfolge welche Verfahren zur Hostname-Ermittlung genutzt werden sollen. `bind` bedeutet, dass die Resolver-Bibliothek auf den in */etc/resolv.conf* eingestellten Nameserver zurückgreift. `hosts` lässt die Auswertung von */etc/hosts* zu. `nis` bewirkt, dass der veraltete Network Information Service zum Einsatz kommt.

## Beispiel

Die Standardkonfiguration für *host.conf* sieht bei den meisten Distributionen so wie im folgenden Listing aus. Die Resolver-Bibliothek wertet zuerst die Datei */etc/hosts* aus und kontaktiert dann den in *resolv.conf* eingestellten Nameserver. Die *multi*-Zeile erlaubt, dass einem in */etc/hosts* angegebenen Hostnamen mehrere IP-Adressen zugeordnet werden dürfen.

```
order hosts, bind  
multi on
```

### /etc/hostname

Die Datei */etc/hostname* enthält bei nahezu allen gängigen Linux-Distributionen den vollständigen Namen des Rechners, der aus dem Host- und dem Domainnamen besteht. */etc/hostname* auf der Ubuntu-Server-Installation für meinen Webserver enthält daher diesen Text:

kofler.info

*/etc/hostname* wird beim Rechnerstart ausgewertet. Änderungen im laufenden Betrieb können mit dem Kommando *hostname* durchgeführt werden. Bei systemd-kompatiblen Distributionen kann der Hostname zudem dauerhaft durch das Kommando [hostnamectl](#) eingestellt werden.

### /etc/hosts

*/etc/hosts* ordnet IP-Adressen ihren Hostnamen zu. Die Datei enthält normalerweise nur Einträge für *localhost* sowie bei statisch konfigurierten Servern für den Hostnamen. In der Vergangenheit wurde die Datei verwendet, um andere Rechner im lokalen Netzwerk aufzulisten. Diese Funktion übernimmt heute aber selbst in kleinen Netzen ein Router bzw. ein lokaler Nameserver, z.B. Bind oder Dnsmasq.

Die Datei */etc/hosts* enthält zeilenweise Einträge gemäß dem folgenden Muster:

```
ip-addr hostname1 hostname2 hostname3 ...
```

Die Minimalvariante sieht so aus:

```
127.0.0.1 localhost
```

Bei den meisten Linux-Distributionen ist localhost auch für IPv6 definiert. Die folgenden Zeilen zeigen die Defaulteinstellungen unter Fedora und RHEL:

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1          localhost localhost.localdomain localhost6 localhost6.localdomain6
```

Bei einer statischen Netzwerkkonfiguration, z.B. auf einem Root-Server, kann *hosts* auch einen Eintrag mit der IP-Adresse und dem Hostnamen des Rechners enthalten:

```
211.212.213.214 firma-abc.de firma-abc
```

### /etc/sysconfig/network-scripts/ifcfg-xxx

Auf Fedora- und Red-Hat-Systemen enthält die Datei */etc/sysconfig/network-scripts/ifcfg-xxx* die Netzwerkkonfiguration für die Schnittstelle xxx. Die Einstellungen der Schnittstelle *eth0* befinden sich somit in *ifcfg-eth0*. Die Datei enthält zeilenweise Variablenzuweisungen. Bei einfachen Zeichenketten ohne Leerzeichen sind die Anführungszeichen optional.

In aktuellen Versionen von CentOS, Fedora und RHEL werden die Dateien nicht mehr von distributionsspezifischen Scripts ausgewertet, sondern vom Plugin *ifcfg-rh* des NetworkManagers. Die Syntax der Konfigurationsdateien beschreibt `man nm-settings-ifcfg-rh`.

## Grundeinstellungen

- `DEVICE="devname"`  
enthält den Schnittstellennamen, z.B. `eth0` oder `enp0s5`.
- `HWADDR="nn:nn:nn:nn:nn:nn"`  
gibt die MAC-Adresse der Schnittstelle an.
- `NM_CONTROLLED="yes | no"`  
gibt an, ob die Schnittstelle durch den NetworkManager gesteuert werden soll. In der Defaulteinstellung `yes` werden alle weiteren Einstellungen in *ifcfg-xxx* ignoriert. Wenn Sie die Schnittstelle manuell in *ifcfg-xxx* konfigurieren möchten, muss die Datei daher unbedingt `NM_CONTROLLED="no"` enthalten!

- **ONBOOT="yes | no"**  
gibt an, ob die Schnittstelle beim Rechnerstart automatisch aktiviert werden soll.
- **TYPE="Ethernet"**  
gibt die Art der Schnittstelle an. Außer **Ethernet** sind z.B. **Bridge** für Netzwerkbrücken oder **IPsec** für IP-Secure-Schnittstellen zulässig. An dieser Stelle berücksichtige ich allerdings nur die Konfiguration von Ethernet-Schnittstellen.

## IPv4-Konfiguration

- **BOOTPROTO="dhcp | none"**  
gibt an, ob die Konfiguration der Schnittstelle durch DHCP oder statisch erfolgt. Die weiteren Parameter sind nur für **BOOTPROTO="none"** relevant.
- **IPADDR="n.n.n.n"**  
bestimmt die IP-Adresse.
- **PREFIX="n"**  
legt das Netzwerk-Präfix fest. **PREFIX="24"** entspricht **NETMASK="255.255.255.0"**.
- **NETMASK="n.n.n.n"**  
legt die Netzwerkmaske fest (nur, wenn eine **PREFIX**-Angabe fehlt).
- **NETWORK="n.n.n.n"**  
definiert die Netzwerkadresse. Der Parameter gilt als veraltet, weil die Netzwerkadresse aus **IPADDR** und **NETMASK** automatisch ermittelt werden kann.
- **BROADCAST="n.n.n.n"**  
gibt die Broadcast-Adresse an. Der Parameter gilt als veraltet, weil die Broadcast-Adresse aus **IPADDR** und **NETMASK** automatisch ermittelt werden kann.
- **GATEWAY="n.n.n.n"**  
gibt die Gateway-Adresse an. Diese Adresse kann alternativ auch in der Datei **/etc/sysconfig/network** eingestellt werden, wenn es ein zentrales Gateway für mehrere Schnittstellen gibt.

- `ZONE="name"`  
gibt an, welcher FirewallD-Zone die Schnittstelle zugewiesen ist.  
Die Einstellung betrifft nur Distributionen, die FirewallD verwenden, z.B. Fedora ab Version 19. Wenn die Einstellung fehlt, gelten für die Schnittstelle die Regeln der Default-Firewall-Zone. Diese Zone können Sie mit `firewall-cmd --get-default-zone` ermitteln.

## IPv6-Konfiguration

- `IPV6INIT="yes|no"`  
aktiviert IPv6 für diese Schnittstelle (Default no).
- `IPV6_AUTOCONF="yes|no"`  
wertet das Router Advertisement des IPv6-Routers aus und führt eine automatische Konfiguration durch (Default yes).
- `DHCPV6C="yes|no"`  
bezieht die IP-Konfiguration von einem DHCPv6-Server (Default no).
- `DHCPV6C="-x -y"`  
enthält Optionen für den DHCP-Client. Eine Liste der zulässigen Optionen liefert `man dhclient`.
- `IPV6ADDR="nnn"`  
gibt die IPv6-Adresse statisch an.
- `IPV6_DEFAULTGW="nnn"`  
gibt die Gateway-Adresse für IPv6 statisch an.

## Beispiele

Auf Desktop-Systemen kümmert sich in der Regel der NetworkManager um die Netzwerkkonfiguration. Die einzige Voraussetzung besteht darin, dass `ifcfg-xxx` nicht die Zeile `NM_CONTROLLED="no"` enthält. Alle weiteren Parameter trägt der NetworkManager selbst in `ifcfg-xxx` ein.

Wenn Sie die IPv4-Konfiguration selbst durchführen und es einen DHCP-Server im lokalen Netzwerk gibt, sieht `ifcfg-xxx` typischerweise so aus:

```
DEVICE=enp0s5
HWADDR=00:1C:42:85:09:A1
NM_CONTROLLED=no
ONBOOT=yes
BOOTPROTO=dhcp
TYPE=Ethernet
ZONE="trusted"
```

Bei einer statischen Konfiguration muss die Datei dem folgenden Muster entsprechen:

```
DEVICE=eth0
HWADDR=00:1C:42:85:09:A1
NM_CONTROLLED=no
ONBOOT=yes
BOOTPROTO=none
TYPE=Ethernet
IPADDR=10.0.17.33
NETMASK=255.255.255.0
GATEWAY=10.0.17.1
ZONE="trusted"
```

Wenn Sie IPv6 nutzen möchten, ergänzen Sie die *ifcfg*-Datei um die folgenden Zeilen, damit die Schnittstelle das *Router Advertisement* des IPv6-Routers auswertet und die IPv6-Parameter automatisch einstellt:

```
...
IPV6INIT=yes
IPV6_AUTOCONF=yes
```

Wenn die IPv6-Parameter von einem DHCPv6-Server bezogen werden sollen, sieht die Konfiguration so aus:

```
...
IPV6INIT=yes
DHCPOV6C=yes
```

Zu guter Letzt folgt hier noch die Variante für eine statische IPv6-Konfiguration:

```
...
IPV6INIT=yes
IPV6ADDR=2a01:4f8:161:107::2/64
IPV6_DEFAULTGW=fe80::1
```

### /etc/inittab

Die Datei */etc/inittab* enthält Defaulteinstellungen für das klassische Init-V-System. Die Datei ist nur für Distributionen relevant, die dieses Init-System verwenden, z.B. für Debian bis Version 7 sowie für CentOS/RHEL bis Version 5.

Nahezu alle aktuellen Linux-Distributionen verwenden als Init-System systemd, wo es keine mit *inittab* vergleichbare zentrale Konfigurationsdatei gibt. Aus diesem Grund wird hier auf eine genauere Beschreibung der *inittab*-Syntax verzichtet.

### /etc/network/interfaces

Die Datei */etc/network/interfaces* (Kurzschreibweise ENI) enthält bei Debian-Distributionen sowie bei älteren Versionen von Raspian und Ubuntu die Netzwerkkonfiguration.

Unter Raspberry Pi OS ist hingegen *dhcpcd* für die Netzwerkkonfiguration zuständig. Die Konfiguration befindet sich in */etc/dhcpcd.conf*. Das ENI-Framework steht aber weiterhin zur Verfügung. Wenn eine Schnittstelle in */etc/network/interfaces* konfiguriert werden soll, muss sie in *dhcpcd.conf* deaktiviert werden (z.B. *denyinterfaces wlan0*).

Ubuntu ist mit Version 17.10 vollständig auf Netplan umgestiegen (siehe [netplan](#) sowie */etc/netplan/netplan.yaml*). Das ENI-Framework wurde entfernt.

### Zusammenspiel mit dem NetworkManager

Bei Desktop-Installationen ist die Datei normalerweise fast leer und enthält nur die Einstellungen für die Loopback-Schnittstelle:

```
# Loopback-Schnittstelle
auto lo
iface lo inet loopback
```

Für alle weiteren Schnittstellen ist der NetworkManager verantwortlich.

### IPv4-Konfiguration

Ganz anders sieht es bei Server-Installationen bzw. auf Rechnern aus, deren Netzwerkkonfiguration statisch eingestellt ist. In diesem Fall leitet die Zeile *auto name* einen Konfigurationsblock für jede Netzwerkschnittstelle ein, die beim Rechnerstart aktiviert werden soll. Die folgende Zeile *iface name optionen* beschreibt die Basiskonfiguration der Schnittstelle. Je nach Konfigurationstyp folgen dann die weiteren Parameter. Wenn die Schnittstelle die IP-

Daten von einem DHCP-Server bezieht, fällt die Konfiguration sehr knapp aus:

```
# IPv4-Netzwerkschnittstelle mit DHCP
auto eth0
iface eth0 inet dhcp
```

Bei einer statischen Konfiguration werden die Netzwerkparameter zeilenweise durch mehrere Schlüsselwörter angegeben, deren Bedeutung selbsterklärend ist. Das Schlüsselwort `dns-nameservers` dient nur unter Ubuntu zur Konfiguration der Nameserver. Für die Auswertung ist das Paket `resolvconf` zuständig. Unter Debian müssen Sie den Nameserver hingegen in `/etc/resolv.conf` selbst einstellen.

```
# IPv4-Netzwerkschnittstelle, statische Konfiguration
auto eth0
iface eth0 inet static
    address      211.212.213.37
    netmask      255.255.255.224
    gateway      211.212.213.1
    dns-nameservers 211.222.233.244 212.232.234.245
```

Änderungen an einer einzelnen Schnittstelle aktivieren Sie mit `ifdown xxx; ifup xxx`, wobei Sie `xxx` durch den Schnittstellennamen ersetzen. Um die gesamte Konfiguration neu zu aktivieren, führen Sie `service network restart` aus.

## IPv6-Konfiguration

Wenn Sie auch IPv6 nutzen möchten, definieren Sie die betreffende Schnittstelle in `/etc/network/interfaces` einfach ein zweites Mal mit dem Schlüsselwort `inet6`. Das Schlüsselwort `auto` gibt an, dass die IPv6-Konfiguration das sogenannte *Router Advertisement* des Gateways bzw. IPv6-Routers berücksichtigt.

```
# IPv4-Konfiguration via DHCP, IPv6-Konfiguration via Router Advertisement
auto eth0
iface eth0 inet dhcp
iface eth0 inet6 auto
```

Wenn das IPv6-Gateway einen DHCPv6-Server verwendet, lautet die korrekte Methode `dhcp`. Wenn außerdem die Router-Adresse per *Router Advertisement* konfiguriert werden soll, ist die Zusatzoption

`accept_ra 1` erforderlich. Das ist beispielsweise der Fall, wenn Sie als DHCP-Server dnsmasq mit der Option `enable-ra` einsetzen.

```
# IPv4- und IPv6-Konfiguration via DHCP
auto eth0
iface eth0 inet dhcp
iface eth0 inet6 dhcp
    accept_ra 1
```

Bei einer statischen Konfiguration muss *interfaces* so aussehen:

```
# IPv4- und IPv6-Konfiguration statisch
auto eth0
iface eth0 inet static
    ... (IPv4-Konfiguration wie bisher)
iface eth0 inet6 static
    address 2a01:4f8:161:107::2
    netmask 64
    gateway fe80::1
```

## WLAN- und WPA-Konfiguration

Um WLAN-Schnittstellen und die WPA-Authentifizierung kümmert sich auf Notebooks normalerweise der NetworkManager. Es besteht aber auch die Möglichkeit, die Konfiguration direkt in */etc/interfaces* vorzunehmen. Besonders häufig ist dies auf Embedded-Geräten oder Minicomputern wie dem Raspberry Pi erforderlich.

Die Konfiguration der WLAN-Schnittstellen erfolgt grundsätzlich wie die von Ethernet-Schnittstellen. Dabei steuern Sie das Zusammenspiel mit dem Programm `wpa_supplicant` für die Authentifizierung durch `wpa`-Parameter. Im einfachsten Fall kann die Konfiguration wie folgt aussehen:

```
# Datei /etc/network/interfaces
...
auto wlan0
iface wlan0 inet dhcp
    wpa-ssid "wlan-name"
    wpa-psk "strenge geheim"
```

Mit `wpa-ssid` geben Sie also den Namen Ihres WLAN-Netzes an. (SSID steht für *Service Set Identifier*.) `wpa-psk` gibt das WLAN-Passwort an, wobei vorausgesetzt wird, dass die Verschlüsselung mit dem WPA-Verfahren (*Wi-Fi Protected Access*) mit Pre-shared Keys erfolgt. Im Heimbereich ist das der Regelfall.

Wenn mehr Parameter eingestellt werden müssen, ist es üblich, die Authentifizierungsdaten in eine externe Datei auszulagern. Die *interfaces*-Datei hat dann den folgenden Aufbau:

```
# /etc/network/interfaces (manuelle WLAN-Konfiguration mit WPA)
auto wlan0
iface wlan0 inet dhcp
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Dabei gibt `wpa-conf` den Ort einer WPA-Konfigurationsdatei an, die der Syntax von `/etc/wpa_supplicant/wpa_supplicant.conf` entsprechen muss.

Wenn es wie unter Raspberry Pi OS eine grafische Benutzeroberfläche zur WLAN-Konfiguration gibt, um die Verbindungsdaten zu *mehreren* WLAN-Netzen zu verwalten, müssen Sie das Schlüsselwort `wpa-roam` verwenden:

```
# /etc/network/interfaces (WLAN-Defaultkonfiguration unter Raspberry Pi OS)
allow-hotplug wlan0
iface wlan0 inet manual
    wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

Die Zeile `allow-hotplug` erlaubt es anderen Programmen, die WLAN-Schnittstelle zu steuern. Die mit `wpa-roam` beginnende Zeile gibt an, in welcher Datei die WPA-Schlüssel verschiedener WLAN-Netze gespeichert sind. Sobald ein passendes Netz in Reichweite ist, wird automatisch eine Verbindung hergestellt, wobei die IP-Daten dann via DHCP bezogen werden.

Die Anweisung `iface default inet dhcp` besagt, dass jedes in `wpa_supplicant.conf` definierte Netzwerk standardmäßig DHCP zur IP-Konfiguration verwenden soll.

## Netzwerkbrücken und Spezialeinstellungen

Zur Konfiguration virtueller Brücken oder um spezielle Konfigurationswünsche zu erfüllen, gibt es eine Menge weiterer Schlüsselwörter, die auf der [man](#)-Seite zu `interfaces` dokumentiert sind. Insbesondere können Sie mit den Schlüsselwörtern `pre-up`, `up` und `post-up` sowie `pre-down`, `down` und `post-down` Kommandos angeben, die unmittelbar vor, während bzw. nach dem Einrichten

bzw. Stoppen einer Netzwerkschnittstelle ausgeführt werden sollen. Die folgenden Zeilen zeigen eine Netzwerkbrücke, die mit der physischen Schnittstelle `eth0` verbunden ist und Netzwerkpakete zu vier virtuellen Maschinen weiterleitet:

```
# IPv4-Netzwerkbrücke
auto br0
iface br0 inet static
    # statische Konfiguration
    address      210.211.212.18
    broadcast    210.211.212.31
    netmask      255.255.255.224
    gateway      210.211.212.1
    pointopoint  210.211.212.1

    # Brücke
    bridge_ports  eth0
    bridge_stp    off
    bridge_fd     0
    bridge_maxwait 0
    # zusätzliche IPs für KVM
    up route add -host 210.211.212.26 dev br0
    up route add -host 210.211.212.27 dev br0
    up route add -host 210.211.212.28 dev br0
    up route add -host 210.211.212.29 dev br0

    # statische Routen für IPv6
    up ip -6 addr add 2a01:1234:567:890::2/64      dev br0
    up ip route  add default via fe80::1           dev br0
    down ip route del default via fe80::1          dev br0
    down ip -6   del add 2a01:1234:567:890::2/64 dev br0
```

### /etc/systemd/journald.conf

Der zu systemd gehörende Protokolldienst *Journal* wird durch die folgenden Konfigurationsdateien gesteuert:

```
/etc/systemd/journald.conf
/etc/systemd/journald.conf.d/*.conf
/run/systemd/journald.conf.d/*.conf
[/usr]/lib/systemd/journald.conf.d/*.conf
```

Bei vielen Distributionen existiert ausschließlich

`/etc/systemd/journald.conf`, und auch diese Datei enthält nur auskommentierte Beispielzeilen. In diesem Fall gelten die Defaulteinstellungen.

Zur Veränderung der Konfiguration können Sie innerhalb der Konfigurationsgruppe `[Journal]` die folgenden Schlüsselwörter verwenden:

- **ForwardToSyslog**  
gibt an, ob durch das Journal protokollierte Nachrichten auch an einen traditionellen Syslog-Dienst weitergegeben werden sollen.
- **MaxFileSec**  
gibt an, nach welcher Zeit spätestens eine neue Logging-Datei gestartet werden soll. Die Defaulteinstellung sieht einen Monat vor. Diese Einstellung ist nur relevant, wenn die Logging-Dateien langsamer wachsen, als die Limits `SystemMaxUse`, `SystemMaxFileSize` und `SystemKeepFree` vorgeben.
- **MaxLevelStore**  
gibt die Prioritätsstufe an, bis zu der Nachrichten im Journal gespeichert werden. Die Defaulteinstellung lautet `debug`.
- **SystemKeepFree**  
gibt an, wie viel Prozent des Dateisystems frei bleiben müssen. Die Defaulteinstellung beträgt 15 Prozent.
- **SystemMaxFileSize**  
gibt die maximale Größe einer Logging-Datei an. Die Defaulteinstellung beträgt ein Achtel von `SystemMaxUse`. Das führt zu einem automatischen *Rotating*, wobei neben der aktuellen Logging-Datei maximal sieben ältere Dateien entstehen.
- **SystemMaxUse**  
gibt an, wie viel Prozent des Dateisystems die Logging-Dateien maximal beanspruchen können. Bevor dieses Limit überschritten wird, werden alte Logging-Dateien gelöscht. Die Defaulteinstellung beträgt 10 Prozent.

Unzählige weitere Optionen können Sie mit `man journald.conf` nachlesen.

#### /etc/locale.conf

Die Datei `/etc/locale.conf` enthält bei Distributionen, die eine aktuelle systemd-Version als Init-System verwenden, die Sprach- und Zeichensatzeinstellungen. Bei der Veränderung der

Einstellungen hilft das Kommando [localectl](#). Die folgende Aufzählung beschreibt die wichtigsten Parameter, die eingestellt werden können:

- `LANG=xxx`  
bestimmt den Standardwert für alle nicht eingestellten LC-Variablen.
- `LC_ALL=xxx`  
überschreibt alle individuellen LC-Einstellungen.
- `LC_COLLATE=xxx`  
bestimmt die Sortierordnung.
- `LC_CTYPE=xxx`  
bestimmt den Zeichensatz.
- `LC_MESSAGES=xxx`  
bestimmt die Darstellung von Nachrichten, Fehlermeldungen etc.
- `LC_MONETARY=xxx`  
bestimmt die Darstellung von Geldbeträgen.
- `LC_NUMERIC=xxx`  
bestimmt die Darstellung von Zahlen.
- `LC_PAPER=xxx`  
bestimmt die Papiergröße.
- `LC_TIME=xxx`  
bestimmt die Darstellung von Datum und Uhrzeit.

Eine Liste aller möglichen Einstellungen liefert `locale -a`.

Üblicherweise wird die Schreibweise `spr_land.zs` verwendet, wobei `spr` durch zwei Buchstaben die Sprache und `land` durch zwei Buchstaben das Land bezeichnet. `zs` gibt den Zeichensatz an. Im deutschen Sprachraum sollten Sie `de_DE.utf8` verwenden.

Je nach Distribution werden diese bzw. weitere Lokalisierungseinstellungen an anderen Orten gespeichert:  
Debian, Ubuntu: `/etc/default/locale`  
SUSE: `/etc/sysconfig/language`

## Beispiel

Mit der folgenden Einstellung in *locale.conf* gilt Deutsch als Defaultsprache und UTF-8 als Defaultzeichensatz:

```
LANG="de_DE.utf8"
```

### /etc/login.defs

*/etc/login.defs* enthält diverse Einstellungen, die beim Anlegen neuer Benutzer und Gruppen berücksichtigt werden. Die Einstellungen gelten für die Kommandos aus dem shadow-utils-Paket, also z.B. für [adduser](#) und [useradd](#) sowie für [addgroup](#) und [groupadd](#). Die *login.defs*-Parameter haben hingegen keinen Einfluss auf Kommandos, die auf die Pluggable Authentication Modules (PAM) zurückgreifen; dazu zählt insbesondere [passwd](#)!

Die Einstellungen in *login.defs* erfolgen zeilenweise. In jeder Zeile werden zuerst ein Schlüsselwort, dann ein oder mehrere Leer- oder Tabulatorzeichen und schließlich die gewünschte Einstellung angegeben.

- **CREATE\_HOME**  
gibt an, ob beim Anlegen neuer Benutzer automatisch auch ein Heimatverzeichnis erzeugt werden soll (yes/no).
- **ENCRYPT\_METHOD**  
gibt an, welcher Algorithmus zur Speicherung der Hash-Codes in */etc/shadow* verwendet werden soll. Zulässige Einstellungen sind DES, MD5, SHA256 und SHA512 (zurzeit am sichersten).
- **ENV\_PATH und ENV\_SUPATH**  
enthält eine Voreinstellung für die PATH-Umgebungsvariable für gewöhnliche Benutzer und für root bzw. [su](#). Weitere Einstellungen für die PATH-Variable erfolgen bei den meisten Distributionen in */etc/profile* oder in */etc/bashrc*.
- **GID\_MIN und GID\_MAX**  
gibt die minimale und maximale Gruppen-ID-Nummer für gewöhnliche Gruppen an.
- **LOG\_OK\_LOGINS**  
zeichnet erfolgreiche Logins mit Syslog auf (yes/no).

- **MAIL\_DIR**  
gibt den Ort des Verzeichnisses zur Speicherung lokaler E-Mails an.
- **PASS\_MAX\_DAYS, PASS\_MIN\_DAYS und PASS\_WARN\_AGE**  
gibt an, wie lange Passwörter gelten und nach welchem Zeitraum sie frühestens verändert werden dürfen (siehe [chage](#)).
- **PASS\_MIN\_LEN**  
definiert die minimale Passwortlänge.
- **SYS\_GID\_MIN und SYS\_GID\_MAX**  
gibt die minimale und maximale GID für System-Accounts an.
- **SYS\_UID\_MIN und SYS\_UID\_MAX**  
gibt die minimale und maximale UID für System-Accounts an.
- **SYSLOG\_SU\_ENAB und SYSLOG\_SG\_ENAB**  
zeichnet [su](#)- und [newgrp](#)-Kommandos mit Syslog auf (yes/no).
- **UID\_MIN und UID\_MAX**  
gibt die minimale und maximale Benutzer-ID-Nummer für gewöhnliche Benutzer an.
- **USERGROUPS\_ENAB**  
steuert, ob zusammen mit jedem Benutzer eine eigene Gruppe erzeugt werden soll (yes/no).

## Beispiel

Die folgenden Zeilen zeigen die Defaulteinstellungen in *login.defs* unter Fedora. Beachten Sie aber, dass die Einstellungen für **PASS\_MIN\_LEN** und **UMASK** für den Betrieb von Fedora nicht relevant sind! Die Mindestanforderungen für Passwörter sind durch das PAM-Modul **pwquality** vorgegeben, und [umask](#) wird in */etc/profile* mit 022 voreingestellt.

```
MAIL_DIR      /var/spool/mail
PASS_MAX_DAYS      99999
PASS_MIN_DAYS      0
PASS_MIN_LEN       5
PASS_WARN_AGE       7
UID_MIN           1000
UID_MAX           60000
```

SYS_UID_MIN	201
SYS_UID_MAX	999
GID_MIN	1000
GID_MAX	60000
SYS_GID_MIN	201
SYS_GID_MAX	999
CREATE_HOME	yes
UMASK	077
USERGROUPS_ENAB	yes
ENCRYPT_METHOD	SHA512

### /etc/mdadm/mdadm.conf

Die Datei */etc/mdadm/mdadm.conf* fasst die Konfiguration des Software-RAID-Systems des Rechners zusammen. Wenn die Datei fehlt oder verloren gegangen ist, kann sie aus den in RAID-Partitionen gespeicherten Metadaten mit dem folgenden Kommando teilweise neu erstellt werden:

```
root# mdadm --examine --scan > /etc/mdadm/mdadm.conf
```

Die folgende Auflistung nennt die wichtigsten Schlüsselwörter in *mdadm.conf*:

- **ARRAY device1 metadata=... UUID=... name=... devices=...**  
gibt an, aus welchen Komponenten ein RAID-Verbund besteht.
- **CREATE owner=... group=... mode=... auto=...**  
gibt an, welche Defaulteinstellungen beim Aktivieren eines RAID-Devices gelten sollen. Der **auto**-Parameter entspricht der Option **--auto** des Kommandos [mdadm](#).
- **DEVICE device1 device2 | partitions | containers**  
gibt an, welche Festplatten oder Partitionen RAID-Komponenten enthalten können. Wenn die **DEVICE**-Angabe fehlt, gilt **DEVICE partitions containers**. Der Befehl [mdadm](#) durchsucht damit alle SSD- und Festplattenpartitionen sowie alle Logical Volumes.
- **MAILADDR name**  
gibt an, an welche lokale E-Mail-Adresse Meldungen, Warnungen und Fehlermeldungen gesendet werden sollen, sofern [mdadm](#) im Monitor-Modus läuft.

### Beispiel

Die folgenden Zeilen zeigen die Konfigurationsdatei eines Linux-Servers mit zwei RAID-1-Devices:

```
CREATE owner=root group=disk mode=0660 auto=yes  
MAILADDR root  
ARRAY /dev/md/0 metadata=1.2 UUID=0860...f0a2 name=rescue:0  
ARRAY /dev/md/1 metadata=1.2 UUID=2546...e6e2 name=rescue:1
```

### /etc/modules

Die Debian- und Ubuntu-spezifische Datei */etc/modules* enthält zeilenweise die Namen von Kernelmodulen, die beim Rechnerstart geladen werden sollen. Normalerweise erkennt Linux selbst, welche Kernelmodule geladen werden müssen. Nur wenn das nicht funktioniert, müssen Sie der Datei */etc/modules* die betreffenden Modulnamen hinzufügen.

### Alternative Verfahren

Kernelmodule, die bereits während des Bootprozesses erforderlich sind, müssen in die Initrd-Datei eingebaut werden. Dabei helfen die Kommandos [dracut](#) (Fedora, CentOS, RHEL) und [update-initramfs](#) (Debian, Ubuntu).

Eine andere Möglichkeit besteht darin, das Kernelmodul am Ende des Init-Prozesses durch das Script */etc/rc.local* zu laden.

Raspberry Pi OS verwendet Device Trees, um die zur Hardware passenden Kernelmodule zu verwalten. Die Konfiguration erfolgt durch */boot/config.txt*. Die Device-Tree-Dateien mit der Kennung *\*.dtb* befinden sich in den Verzeichnissen */boot* und */boot/overlays*.

### /etc/netplan/netplan.yaml

Netplan (<https://netplan.io>) kommt in Ubuntu zum Einsatz. Das Framework konfiguriert und steuert andere Netzwerk-Backends wie den NetworkManager sowie die systemd-Komponente networkd. Die Konfigurationsdateien für Netplan befinden sich in */etc/netplan*, */lib/netplan* und */run/netplan* und haben die Endung *.yaml*. Damit Konfigurationsänderungen wirksam werden, müssen Sie [netplan apply](#) ausführen. Die Konfigurationsdateien werden in alphabetischer Reihenfolge ausgewertet.

Auf Desktop-Systemen enthält `/etc/netplan` standardmäßig nur eine minimale Konfigurationsdatei, die alle weiteren Arbeiten an den NetworkManager delegiert:

```
# Datei /etc/netplan/01-network-manager-all.yaml (Desktop-Systeme)
network:
  version: 2
  renderer: NetworkManager
```

Bei Server-Installationen in lokalen Netzwerken, die auf DHCP zurückgreifen, sieht die Konfiguration wie in den folgenden Zeilen aus. Für die eigentliche Realisierung der Netzwerkanbindung ist networkd zuständig. (Die Zeile `renderer: networkd` kann entfallen, weil networkd als Default-Backend für Netplan gilt.)

```
# Datei /etc/netplan/01-netcfg.yaml (Server/DHCP)
# Schnittstelle enp0s3 via DHCP konfigurieren
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: true
```

Zur statischen Netzwerkkonfiguration inklusive Nameserver-Spezifikation können Sie sich an diesem Muster orientieren:

```
# Datei /etc/netplan/01-netcfg.yaml (Server/statisch)
# Schnittstelle ens3 statisch konfigurieren
network:
  version: 2
  renderer: networkd
  ethernets:
    ens3:
      addresses: [ 138.201.20.182/26 ]
      gateway4: 138.201.20.176
      nameservers:
        search: [ ubuntu-buch.info ]
        addresses:
          - "213.133.100.100"
          - "213.133.98.98"
```

In den Konfigurationsdateien sind natürlich auch IPv6-Adressen erlaubt. Teilweise müssen Sie IPv4- und IPv6-spezifische Schlüsselwörter nennen, z.B. `gateway4` versus `gateway6` oder `dhcp4` versus `dhcp6`. Eine Menge weiterer Beispiele samt Routing, Bridging, Bonding, VLANs etc. finden Sie hier:

<https://netplan.io/examples>

## /etc/systemd/network/networkd.network

Das zu systemd gehörende Netzwerk-Backend networkd ist aktuell zwar bei vielen Distributionen installiert, aber nur selten aktiv. Die wichtigste Ausnahme ist Ubuntu Server, wo networkd durch Netplan gesteuert wird.

networkd wertet Konfigurationsdateien in den Verzeichnissen `/etc/systemd/network`, `/lib/systemd/network` und `/run/systemd/network` aus. Die Dateien müssen die Endung `*.network` haben.

Die folgenden Zeilen zeigen beispielhaft die Einrichtung einer Ethernet-Schnittstelle mit statischer IP-Adresse in einer virtuellen Maschine:

```
# Datei /etc/systemd/network/01-mystatic.network
[Match]
Name=ens3

[Network]
Address=138.201.20.182/26
Gateway=138.201.20.176
DNS=213.133.100.100
DNS=213.133.98.98
Domains=ubuntu-buch.info
```

Das zweite Beispiel zeigt ebenfalls eine statische Konfiguration, diesmal für einen Root-Server bei Hetzner inklusive Routing und IPv6-Setup:

```
[Match]
MACAddress=90:1b:0e:8e:48:86
```

```
[Network]
Address=2a01:4f8:171:2baf::2/64
Gateway=fe80::1
Address=138.201.20.176/26
Gateway=138.201.20.129
```

```
[Route]
Destination=138.201.20.128/26
Gateway=138.201.20.129
```

Wenn ein DHCP-Server zur Verfügung steht, sieht eine mögliche Konfiguration so aus:

```
# Datei /etc/systemd/network/01-dhcpclient.network
[Match]
Name=enp0s3
```

```
[Network]
DHCP=ipv4
```

```
[DHCP]
UseMTU=true
RouteMetric=100
```

Weitere Details zur Syntax der networkd-Konfiguration gibt man `man systemd.network`. Sehr informativ ist auch das Wiki von Arch Linux: <https://wiki.archlinux.org/index.php/systemd-networkd>

### /etc/nsswitch.conf

Die Datei `/etc/nsswitch.conf` steuert, wie Host-, Benutzer- und Gruppennamen durch die Name-Switch-Funktionen in der GNU-C-Bibliothek aufgelöst werden. Änderungen an dieser Datei sind selten erforderlich – es sei denn, im lokalen Netzwerk wird LDAP oder ein anderer Netzwerkdienst zur Benutzerverwaltung und -authentifizierung eingesetzt.

Die Datei enthält zeilenweise Einträge, wobei die erste Spalte jeweils den Namen einer Datenbank angibt. Mit »Datenbank« ist hierbei eine Gruppe von Informationen gemeint, also z.B. die Liste aller Benutzer samt Login-Namen, vollständigem Namen, Heimatverzeichnis, Default-Shell etc. Zulässige Schlüsselwörter für die erste Spalte sind unter anderem `aliases` (Mail-Aliasse), `group`, `hosts`, `networks`, [passwd](#), `rpc` und `shadow`.

Dem Datenbanknamen folgt ein Doppelpunkt und dann eine durch Leerzeichen getrennte Auflistung von Diensten. Die wichtigsten Dienste sind:

- `files` liest die Daten aus lokalen Konfigurationsdateien, z.B. `/etc/passwd` oder `/etc/group`.
- `compat` hat eine ähnliche Bedeutung wie `files`, erlaubt aber die Interpretation von Zusatzinformationen in `/etc/passwd`.
- `db` liest die Daten aus Datenbankdateien.
- `nis` wertet den veralteten Network Information Service aus.
- `dns` fragt beim Nameserver nach (für Hostnamen).

- `mdns4_minimal` verwendet Zeroconf zur Auflösung von Hostnamen.
- `ldap` kommuniziert mit einem LDAP-Server.
- `wins` greift auf einen WINS-Server zurück.

Außerdem können in der Form [STATUS=ACTION] bzw. [!STATUS=ACTION] Kommandos angegeben werden, die dann beim (Nicht-)Eintreten eines bestimmten Ereignisses ausgeführt werden sollen. Zulässige STATUS-Werte sind `success`, `notfound`, `unavail` und `tryagain`. Mögliche Aktionen sind `return` (Ergebnis sofort zurückgeben, nicht auf andere Optionen zurückgreifen) und `continue` (den nächsten Lookup-Dienst verwenden).

## Beispiel

Die folgenden Zeilen zeigen einen Auszug aus der Defaultkonfiguration unter Fedora. Für die Benutzeraccounts werden ausschließlich die Dateien `/etc/passwd`, `/etc/shadow` und `/etc/group` berücksichtigt. Hostnamen werden aus `/etc/hosts` gelesen, beim Nameserver erfragt sowie via Zeroconf (Avahi) ermittelt.

```
passwd:      files
shadow:     files
group:      files
hosts:      files dns mdns4_minimal
...

```

### /etc/os-release

Bei aktuellen Distributionen, die systemd als Init-System verwenden, enthält `/etc/os-release` Informationen über die installierte Distribution. Die folgenden Zeilen zeigen die Syntax der Datei:

```
NAME="Oracle Linux Server"
VERSION="8.2"
ID="ol"
ID_LIKE="fedora"
VARIANT="Server"
VARIANT_ID="server"
VERSION_ID="8.2"
PLATFORM_ID="platform:el8"
PRETTY_NAME="Oracle Linux Server 8.2"
...
```

Die Datei ist genau genommen keine Konfigurations-, sondern eine Informationsdatei. Es ist also nicht sinnvoll, die Datei zu ändern. Sie wird aber unter Umständen gelegentlich im Zuge von Updates aktualisiert.

### /etc/passwd

Die Datei `/etc/passwd` enthält Daten zu allen lokalen Accounts auf dem Rechner, darunter den Login-Namen, den vollständigen Namen, die Standard-Shell und das Heimatverzeichnis. *Nicht* enthalten ist hingegen der Hash-Code des Passworts – dieser ist aus Sicherheitsgründen in der separaten Datei `/etc/shadow` gespeichert. Zur Verwaltung der Accounts sollten Sie nach Möglichkeit die dafür vorgesehenen Kommandos verwenden, z.B. [useradd](#), [usermod](#) oder [userdel](#).

Die `passwd`-Datei besteht aus sieben Spalten, die jeweils durch Doppelpunkte voneinander getrennt sind:

- Die **erste Spalte** (Account-Name) enthält den Login-Namen. Dieser darf keine Leerzeichen enthalten und sollte frei von Sonderzeichen sein. Es ist üblich, ausschließlich Kleinbuchstaben zu verwenden.
- Die **zweite Spalte** (Passwort) enthielt in der Vergangenheit den Hash-Code des Passworts. Alle aktuellen Linux-Distributionen speichern in dieser Spalte nun das Zeichen x, die Hash-Codes befinden sich in `/etc/shadow`.
- Die **dritte Spalte** (UID) enthält die interne Benutzer-ID-Nummer, die für die Verwaltung der Zugriffsrechte und Prozesse verwendet wird. Bei den meisten Distributionen sind UIDs kleiner 1000 für System-Accounts reserviert und UIDs ab 1000 für reguläre Benutzer vorgesehen.
- Die **vierte Spalte** (primäre Gruppe) gibt die Nummer der primären Gruppe des Benutzers an. Der Benutzer kann weiteren

Gruppen zugeordnet sein; sekundäre Gruppen werden in `/etc/group` gespeichert.

- Die fünfte Spalte (`Kommentar`) ist für eine Kommentar-Zeichenkette vorgesehen, die aus mehreren Teilen bestehen kann. In der Praxis wird hier zumeist nur der vollständige Benutzername gespeichert. Wenn dieser beim Anlegen des Benutzers nicht angegeben wurde, stimmt die fünfte Spalte mit der ersten überein.
- Die sechste Spalte (`Heimatverzeichnis`) enthält den Pfad zum Heimatverzeichnis.
- Die siebte Spalte (`Shell`) gibt die Shell an, die nach dem Login automatisch aktiv ist – bei gewöhnlichen Benutzern oft `/bin/bash`. Bei System-Accounts, für die kein Login vorgesehen ist, wird statt der Shell oft `/sbin/nologin` oder `/bin/false` verwendet. Die Default-Shell kann mit dem Kommando [`chsh`](#) verändert werden.

## Beispiel

Die folgenden Zeilen zeigen einige System-Accounts sowie einen Benutzer-Account auf einem Fedora-System:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
...
kofler:x:1000:1000:Michael Kofler:/home/kofler:/bin/bash
```

### /etc/profile

`/etc/profile` sowie `/etc/profile.d/*.sh` enthalten systemweite Einstellungen von Umgebungsvariablen. Dort werden Variablen wie PATH, MAIL, HOSTNAME voreingestellt. Außerdem wird hier bei vielen Distributionen mit [`umask`](#) eingestellt, welche Zugriffsbits bei neu erzeugten Dateien gesetzt werden sollen.

`/etc/profile` ist eine Shell-Script-Datei. Syntaktisch ist somit alles erlaubt, was der gemeinsame Nenner der wichtigsten Shells zulässt. Unter Ubuntu ist ergänzend zu `/etc/profile` die Datei

*/etc/environment* für die systemweite Einstellung von Umgebungsvariablen vorgesehen. Tatsächlich enthält diese Datei üblicherweise aber nur die Einstellung der `PATH`-Variablen. Die `profile`-Einstellungen werden durch benutzerspezifische Einstellungen in `.profile` sowie durch Shell-spezifische Einstellungen ergänzt. Die `bash` liest z.B. die Dateien `/etc/bashrc`, `/etc/bash.bashrc`, `.bashrc` und `.alias`. Je nach Distribution kann es weitere Konfigurationsdateien geben. Beispielsweise wird `PATH` in Debian und Ubuntu durch */etc/environment* eingestellt. Für die im Grafiksystem X gültigen Umgebungsvariablen werden auch die Dateien des Verzeichnisses `/etc/X11/Xsession.d/` verarbeitet, falls dieses existiert.

```
/etc/rc.local  
/etc/rc.d/rc.local  
/etc/init.d/boot.local
```

Einige Distributionen führen zwecks Kompatibilität zu alten Init-V-Systemen die Script-Datei `rc.local` aus. Je nach Distribution hat die Datei unterschiedliche Orte:

Debian, Raspberry Pi OS, Ubuntu:	<code>/etc/rc.local</code>
CentOS, Fedora, RHEL:	<code>/etc/rc.d/rc.local</code>
openSUSE/SUSE:	<code>/etc/init.d/boot.local</code>

Die Datei muss ausführbar sein! Außerdem müssen Sie bei einigen Distributionen nach dem Einrichten der Datei `systemctl daemon-reload` ausführen, damit die Datei berücksichtigt wird.

## Beispiel

Das folgende Script speichert den Zeitpunkt des letzten Boot-Vorgangs in `/var/log/boottime`:

```
#!/bin/bash  
# Datei /etc/rc.local oder /etc/rc.d/rc.local oder /etc/init.d/boot.local  
date > /var/log/boottime
```

Damit das Script in `/etc` bzw. `/etc/rc.d` bzw. `/etc/init.d` von `systemd` berücksichtigt wird, führen Sie folgende Kommandos aus:

```
root# chmod +x /etc/rc.local      (Dateiname je nach Distribution ändern!)  
root# systemctl reload-daemon
```

```
root# reboot
```

### /etc/resolv.conf

*/etc/resolv.conf* steuert, wie die IP-Adressen für unbekannte Netzwerknamen (Hostnamen) ermittelt werden. »Unbekannt« bedeutet, dass die Namen nicht in *hosts* definiert sind. Die Datei besteht normalerweise nur aus zwei oder drei Zeilen mit den folgenden Schlüsselwörtern:

- **nameserver *ipaddr***  
gibt die IP-Adresse eines Nameservers an. Das Schlüsselwort kann mehrfach verwendet werden, um alternative Nameserver anzugeben. Insgesamt sind maximal je drei IPv4- und IPv6-Nameserver erlaubt.
- **domain *mydomain***  
gibt den lokalen Domainnamen an. Das ermöglicht die verkürzte Angabe von Hostnamen, also von *name* anstelle von *name . mydomain*.
- **search *domain***  
gibt einen Domainnamen an, der bei Suchanfragen berücksichtigt wird. Insgesamt können bis zu sechs Domainnamen angegeben werden, jeweils durch einen eigenen search-Eintrag. Bei der Namensauflösung werden dann der Reihe nach die angegebenen Domainnamen getestet, bei *ping name* also zuerst *name*, dann *name . domain1*, dann *name . domain2* etc. Die Defaulteinstellung für *search* ist der lokale Domainname. *domain* und *search* schließen sich gegenseitig aus. Wenn *resolv.conf* beide Schlüsselwörter enthält, gilt das zuletzt genannte.

*/etc/resolv.conf* wird häufig dynamisch erzeugt, insbesondere dann, wenn die IP-Konfiguration über DHCP erfolgt.

Ubuntu richtet standardmäßig einen lokalen Nameserver ein. Bei aktuellen Versionen kümmert sich das Programm *systemd-resolv* darum. Wenn Sie die IP-Adressen des DNS-Servers wissen möchten, führen Sie *systemd-resolv --status* aus.

## Beispiel

Die folgenden Zeilen zeigen eine manuelle Name-Server-Konfiguration für IPv4 und IPv6:

```
# /etc/resolv.conf
domain mylan                      # Domainname des LANs
nameserver 211.212.213.1            # IPv4-DNS
nameserver 211.212.214.1            # Ersatz-DNS (falls der erste ausfällt)
nameserver 2001:4860:4860::8888    # IPv6-DNS
```

### /etc/rsyslog.conf

Viele Linux-Distributionen verwenden für die Protokollierung der Meldungen des Kernels sowie diverser Netzwerkdienste das Syslog-kompatible Programm `rsyslogd`. Die Konfiguration dieses Programms erfolgt durch die Datei `/etc/rsyslog.conf` sowie die Ergänzungsdateien `/etc/rsyslog.d/*.conf`. Diese Konfigurationsdateien enthalten Regeln, die aus zwei Teilen bestehen:

- **Selektor**: Der erste Teil jeder Regel gibt an, was protokolliert werden soll.
- **Aktion**: Der zweite Teil steuert, was mit der Meldung geschehen soll.

Regeln können mit dem Zeichen \ über mehrere Zeilen verteilt werden. Es ist möglich, dass auf eine Meldung mehrere Regeln zutreffen. In diesem Fall wird die Meldung mehrfach protokolliert bzw. weitergegeben. Damit Änderungen an der Syslog-Konfiguration wirksam werden, muss der Syslog-Dienst mit `service rsyslog restart` neu gestartet werden!

### Selektoren

Jeder Selektor besteht aus zwei durch einen Punkt getrennten Teilen: *dienst.prioritätsstufe*. Es ist erlaubt, mehrere durch einen Strichpunkt separierte Selektoren anzugeben. Des Weiteren können in *einem* Selektor mehrere Dienste durch Kommas getrennt werden. Alle Linux-Programme, die Syslog verwenden, müssen ihren Meldungen einen Dienst und eine Priorität zuordnen.

Syslog kennt die folgenden Dienste (Facilities): auth, authpriv, cron, daemon, [ftp](#), kern, [lpr](#), [mail](#), news, syslog, user, uucp sowie local0 bis local7. Das Zeichen \* umfasst alle Dienste.

Syslog kennt außerdem diese Prioritätsstufen (in steigender Wichtigkeit): debug, [info](#), notice, warning = warn, err = error, crit, alert und emerg = panic. Die Schlüsselwörter warn, error und panic gelten als veraltet – verwenden Sie stattdessen warning, err und emerg. Das Zeichen \* umfasst alle Prioritätsstufen. Das Schlüsselwort none gilt für Nachrichten, denen keine Priorität zugeordnet ist. Die Angabe einer Prioritätsstufe schließt alle höheren (wichtigeren) Prioritätsstufen mit ein. Der Selektor mail.err umfasst also auch crit-, alert- und emerg-Meldungen des Mail-Systems. Wenn Sie explizit nur Nachrichten einer bestimmten Priorität wünschen, stellen Sie das Zeichen = voran (also etwa mail.=err).

## Aktionen

Als Aktion wird normalerweise der Name einer Logging-Datei angegeben. Normalerweise werden Logging-Dateien nach jeder Ausgabe synchronisiert. Wenn dem Dateinamen ein Minuszeichen vorangestellt ist, verzichtet Syslog auf die Synchronisierung. Das ist wesentlich effizienter, allerdings gehen dann bei einem Absturz noch nicht physikalisch gespeicherte Meldungen verloren.

Syslog kann Nachrichten auch an FIFO-Dateien (*First In, First Out*) oder Pipes weiterleiten. In diesem Fall stellen Sie dem Dateinamen das Zeichen | voran. Die Datei `/dev/xconsole`, die im folgenden Listing vorkommt, ist eine besondere FIFO-Datei zur Weitergabe von Meldungen an das Grafiksystem X.

Das Zeichen \* bedeutet, dass die Nachricht an alle in Konsolen bzw. via SSH eingeloggten Benutzer gesendet wird. Da das sehr störend ist, wird es standardmäßig nur für kritische Meldungen verwendet.

## Journal-Konfiguration

Aktuelle Distributionen verwenden statt oder ergänzend zu `rsyslogd` das zum systemd-Projekt gehörige Programm *Journal*. Dessen Konfiguration erfolgt durch `/etc/systemd/journal.conf` sowie durch einige weitere Dateien.

## Beispiel

Die folgenden Zeilen geben die Syslog-Standardkonfiguration von Ubuntu leicht gekürzt und etwas übersichtlicher formatiert wieder:

```
# Datei /etc/rsyslog.d/50-default.conf bei Ubuntu
# Selektor          Aktion
auth,authpriv.*    /var/log/auth.log
*.*;auth,authpriv.none  -/var/log/syslog
kern.*             -/var/log/kern.log

mail.*             -/var/log/mail.log
mail.err           /var/log/mail.err
*:emerg            :omusrmmsg:*

daemon.*;mail.*; news.err; *.=debug;*.=info; *.=notice;*.=warn      |/dev/xconsole
```

Im Klartext bedeutet die obige Konfiguration:

- */var/log/auth.log* enthält Authentifizierungsmeldungen aller Prioritätsstufen. Dazu zählen gescheiterte und erfolgreiche Login-Versuche (auch via SSH), PAM-Meldungen, [sudo](#)-Kommandos etc. Als einzige Logging-Datei wird *auth* bei jeder Meldung sofort synchronisiert.
- */var/log/syslog* enthält *alle* via Syslog protokollierten Meldungen (inklusive Authentifizierungsmeldungen, denen keine Priorität zugewiesen ist). Der allumfassende Ansatz ist zugleich ein Vorteil und ein Nachteil: Einerseits können Sie so aus einer einzigen Datei alle erdenklichen Informationen extrahieren; andererseits ist es in diesem Sammelsurium natürlich besonders schwierig, relevante Einträge zu finden.
- */var/log/kern.log* enthält alle Kernelmeldungen.
- Die Nachrichten des Mail-Systems (z.B. Postfix) werden über mehrere Dateien verteilt. In *mail.log* werden *alle* Nachrichten gespeichert, in *mail.err* nur Fehlermeldungen.

- Kritische Systemmeldungen, z.B. über einen bevorstehenden Shutdown oder über Kernelfehler, werden durch :omusrmmsg:\* an alle Benutzer weitergeleitet, genau genommen an alle Terminalfenster und Konsolen. omusrmmsg ist ein rsyslog-Modul, um Nachrichten an Benutzer zu senden.
- Außerdem werden diverse Warnungen und Fehlermeldungen an das X-System weitergeleitet. Um diese Meldungen unter X zu verfolgen, starten Sie das Programm xconsole. Es sieht wie ein kleines Terminalfenster aus, erlaubt aber keine Eingaben.

### /etc/services

Die Datei */etc/services* enthält eine Liste aller üblichen Netzwerkdienste und der ihnen zugeordneten IP-Ports und Protokolltypen. Veränderungen an dieser Datei sind nur in Ausnahmefällen erforderlich. Die Syntax der Datei geht aus den folgenden Beispielzeilen klar hervor:

```
# service-name    port/protocol [aliases]      [# comment]
tcpmux          1/tcp                  # TCP port service multiplexer
tcpmux          1/udp                  # TCP port service multiplexer
rje             5/tcp                  # Remote Job Entry
rje             5/udp                  # Remote Job Entry
echo            7/tcp
echo            7/udp
...
ftp              21/tcp
ftp              21/udp      fsp  fspd
ssh              22/tcp                  # The Secure Shell (SSH) Protocol
ssh              22/udp                  # The Secure Shell (SSH) Protocol
telnet           23/tcp
telnet           23/udp
...
```

### /etc/shadow

*/etc/shadow* ergänzt die Benutzerinformationen aus */etc/passwd* um den Passwort-Hash sowie um Daten, die angeben, wie lange der Account gültig ist bzw. wann das Passwort erneuert werden soll. Zur Veränderung dieser Daten verwenden Sie üblicherweise die Kommandos [passwd](#) und [chage](#).

Die shadow-Datei besteht aus neun Spalten, die durch Doppelpunkte getrennt sind:

- Die erste Spalte (Account-Name) stimmt mit der ersten Spalte von `/etc/passwd` überein.
- Die zweite Spalte (Passwort-Hash) ermöglicht die Überprüfung des Passworts. Eine Rekonstruktion des Passworts aus dem Hash-Code ist hingegen unmöglich. Bei aktuellen Linux-Distributionen wird der Hash-Code mit `sha512sum` erzeugt. Da beim Erzeugen jedes Hash-Codes ein zufälliger Initialisierungswert verwendet wird (das sogenannte *salt*), haben zwei Benutzer, die dasselbe Passwort verwenden, dennoch unterschiedliche Hash-Codes. Damit sind auch Wörterbuchangriffe unmöglich, bei denen vorgenerierte Hash-Codes mit dem Inhalt von `/etc/shadow` verglichen werden.

Bei Accounts, die keinen Login vorsehen, enthält die zweite Spalte einfach einen Stern. Bei gesperrten Accounts werden dem Hashcode zwei Ausrufezeichen vorangestellt. Der Code wird damit ungültig. Zur Reaktivierung des Accounts werden die Ausrufezeichen wieder entfernt.

- Die dritte Spalte (letzte Änderung) gibt an, wann das Passwort zuletzt verändert wurde. Dabei wird in Tagen ab dem 1.1.1970 gerechnet. Der Wert 0 bedeutet, dass der Benutzer das Passwort sofort beim nächsten Login ändern muss.
- Die vierte Spalte (Mindestalter) gibt an, nach wie vielen Tagen ein Passwort frühestens verändert werden darf. 0 erlaubt eine jederzeitige Änderung.
- Die fünfte Spalte (Höchstalter) gibt an, nach wie vielen Tagen ein Passwort spätestens verändert werden muss. 0 erlaubt eine unbeschränkte Nutzung des Passworts.
- Die sechste Spalte (warnzeit) gibt an, wie viele Tage vor dem Ablauf des Passworts der Benutzer darauf hingewiesen wird. 0 deaktiviert diese Warnungen.

- Die siebte Spalte (Sperrzeit) gibt an, wie viele Tage nach Ablaufen des Passworts das Konto gesperrt wird. Es kann dann nur noch vom Administrator durch [passwd](#) -u reaktiviert werden.
- Die achte Spalte (Ablaufzeit) gibt an, wann der Account abläuft. Der Zeitpunkt wird in Tagen ab dem 1.1.1970 angegeben. Wenn der Account unbegrenzt gelten soll, bleibt diese Spalte leer. Der Wert 0 wird nicht verwendet!
- Die neunte Spalte ist für zukünftige Erweiterungen reserviert.

## Beispiel

Die folgenden Zeilen zeigen einen Auszug aus */etc/shadow*, wobei die langen Hash-Codes gekürzt wurden:

```
root:$6$.cc11McOB::0:99999:7:::
bin:*:18095:0:99999:7:::
daemon:*:18095:0:99999:7:::
adm:*:18095:0:99999:7:::
...
kofler:$6$Xax3eEarK::18526:0:99999:7:::
```

Um herauszufinden, wann das Passwort zuletzt verändert wurde, führen Sie [date](#) aus:

```
user$ date -d "1970-01-01 + 18526 days"
Mon Sep 21 00:00:00 CEST 2020
```

Die weiteren [chage](#)-Angaben für die Benutzer root und kofler bedeuten, dass das Passwort jederzeit verändert werden kann (0) und nahezu unbegrenzt gültig ist (99999 = ca. 273 Jahre). Sieben Tage vor dem Ablauf des Passworts wird der Benutzer beim Login gewarnt.

## /etc/apt/sources.list

In */etc/apt/sources.list* sowie */etc/apt/sources.list.d/\** sind die APT-Paketquellen definiert. Die Syntax jeder Zeile sieht so aus:

```
pakettyp uri distribution [komponente1] [komponente2]
[komponente3] ...
```

Der Pakettyp lautet deb für gewöhnliche Debian-Pakete bzw. deb-src für Quellcodepakete. Die zweite Spalte gibt das Basisverzeichnis der Paketquelle an. Neben HTTP- und FTP-Verzeichnissen unterstützt

APT auch gewöhnliche Verzeichnisse, RSH- oder SSH-Server sowie CDs bzw. DVDs.

Die dritte Spalte bezeichnet die Distribution. Alle weiteren Spalten geben die Komponenten der Distribution an, die berücksichtigt werden können. Die Komponentennamen sind von der Distribution und von der Paketquelle abhängig! Beispielsweise unterscheidet Ubuntu zwischen *main*-, *restricted*-, *universe*- und *multiverse*-Paketen, während Debian hingegen zwischen den Komponenten *main*, *contrib*, *non-free* etc. differenziert.

Die zuerst genannten Paketquellen werden bevorzugt: Wenn ein bestimmtes Paket also in mehreren Quellen zum Download zur Verfügung steht, lädt APT es von der ersten Quelle herunter.

## Beispiel

Das folgende Listing zeigt einige Ubuntu-Paketquellen. Aus Platzgründen wurde dabei jeder Eintrag über zwei Zeilen verteilt.

```
# Datei /etc/apt/sources.list
deb http://de.archive.ubuntu.com/ubuntu/ focal main restricted universe
multiverse
deb http://de.archive.ubuntu.com/ubuntu/ focal-updates main restricted universe
multiverse
deb http://security.ubuntu.com/ubuntu focal-security main restricted universe
multiverse
```

## /etc/sudoers

Die Datei */etc/sudoers* sowie die Zusatzdateien in */etc/sudoers.d* legen fest, welche Benutzer welche Kommandos mit welchen Rechten durch [sudo](#) ausführen dürfen. Die Datei enthält außerdem diverse allgemeine Einstellungen, die das Grundverhalten von [sudo](#) steuern. Die Einstellungen fallen je nach Distribution recht unterschiedlich aus.

Die sudoers-Datei sollte nur mit dem Kommando `visudo` verändert werden. Es ruft den durch die Umgebungsvariablen `VISUAL` oder `EDITOR` definierten Editor auf (standardmäßig `vi`), führt vor dem Speichern einen Syntaxcheck durch und stellt so sicher, dass Sie sich nicht durch eine fehlerhafte sudoers-Datei selbst von weiteren

Administrationsarbeiten ausschließen. Besonders wichtig ist das bei Distributionen wie Ubuntu, die keinen root-Login vorsehen.

## Grundeinstellungen

- `Defaults always_set_home`  
verändert beim Benutzerwechsel die `HOME`-Umgebungsvariable, sodass diese auf das Heimatverzeichnis des neuen Benutzers verweist. Normalerweise geschieht dies automatisch. Diese Option ist nur dann relevant, wenn `!env_reset` gilt oder `env_keep` die Variable `PATH` enthält.
- `Defaults env_keep="var1 var2 var3"`  
gibt an, welche Umgebungsvariablen beim Benutzerwechsel erhalten bleiben sollen und somit von `env_reset` ausgenommen sind.
- `Defaults env_reset`  
bewirkt, dass beim Benutzerwechsel alle Umgebungsvariablen zurückgesetzt werden. Diese Einstellung gilt standardmäßig. Um sie zu deaktivieren, geben Sie `Defaults !env_reset` an.
- `Defaults mail_badpass`  
führt dazu, dass nach einem fehlerhaften Login-Versuch eine Warn-E-Mail an den Administrator versandt wird.
- `Defaults secure_path="pfad1:pfad2:pfad3"`  
legt den Inhalt der `PATH`-Umgebungsvariablen für [sudo](#)-Kommandos fest.
- `Defaults targetpw`  
bedeutet, dass grundsätzlich das Passwort für den Account angegeben werden muss, in dem das Kommando ausgeführt werden soll, in der Regel also das root-Passwort. Ohne diese Einstellung erwartet [sudo](#) das Passwort des aktuellen Benutzers.
- `Defaults timestamp_timeout=n`  
gibt an, nach welcher Zeit [sudo](#) neuerlich nach dem Passwort fragt. Die Standardeinstellung beträgt fünf Minuten.

## Rechte

Dreispaltige Einträge in `/etc/sudoers` legen fest, welche Benutzer von welchem Rechner aus welche Programme ausführen dürfen. Die folgende Zeile bedeutet, dass die Benutzerin `kathrin` auf dem Rechner `uranus` das Kommando `/sbin/fdisk` ausführen darf. Das Schlüsselwort `ALL` bedeutet, dass `kathrin` das Kommando unter jedem beliebigen Account ausführen darf, also als `root`, als `news`, als `lp` etc.

```
kathrin uranus=(ALL) /sbin/fdisk
```

Wenn der ersten Spalte von `sudoers` das Zeichen `%` vorangestellt wird, gilt der Eintrag für alle Mitglieder der angegebenen Gruppe. Die folgende Zeile bedeutet, dass alle Benutzer der Gruppe `wheel` von jedem beliebigen Rechner aus alle Kommandos als beliebiger Benutzer ausführen dürfen:

```
%wheel    ALL=(ALL) ALL
```

Es besteht die Möglichkeit, einem bestimmten Benutzer das Ausführen von [sudo](#) ohne Passwortangabe zu erlauben. Das `NOPASSWD`-Schlüsselwort ist allerdings nur gültig, wenn es keine anderen `sudoers`-Zeilen gibt, die vom selben Benutzer ein Passwort verlangen. Das gilt auch für Gruppeneinträge, also z.B. `%admin`.

```
kofler ALL=(ALL) NOPASSWD: ALL
```

Äußerst liberal ist die folgende Regel, die sinngemäß lautet: Jeder darf alles. Sinnvoll ist diese Regel nur in Kombination mit der Option `Defaults targetpw`, die grundsätzlich die Eingabe des Passworts des Benutzers verlangt, in dessen Namen ein Kommando ausgeführt werden soll.

```
ALL ALL=(ALL) ALL
```

## Beispiel

Die folgenden Zeilen zeigen die Defaultkonfiguration unter Ubuntu. Bei dieser Distribution ist `root` ohne gültiges Passwort eingerichtet. Ein `root`-Login ist damit unmöglich. Auch [su](#) oder `ssh -l root` funktionieren nicht. Die einzige Möglichkeit zur Ausführung administrativer Kommandos bietet somit [sudo](#). Dieses Recht haben alle Mitglieder der Gruppen `admin` und [sudo](#), wobei die `admin`-Gruppe

nur noch aus Gründen der Kompatibilität zu älteren Ubuntu-Versionen enthalten ist. Bei aktuellen Ubuntu-Versionen und bei den meisten anderen Distributionen ist [sudo](#) die übliche Gruppe für Benutzer mit [sudo](#)-Rechten.

```
# Defaultkonfiguration in /etc/sudoers bei Ubuntu
Defaults    env_reset
Defaults    mail_badpass
Defaults    secure_path=
"/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
root        ALL=(ALL:ALL) ALL
%admin      ALL=(ALL) ALL
%sudo       ALL=(ALL:ALL) ALL
```

### /etc/sysctl.conf

*/etc/sysctl.conf* enthält Defaulteinstellungen für Kernelparameter. Die Datei wird beim Systemstart ausgewertet, und alle dort aufgezählten Parameter werden entsprechend eingestellt. Bei nachträglichen Änderungen erreichen Sie mit [sysctl](#) -p, dass alle Einstellungen aus *sysctl.conf* ausgelesen und gesetzt werden. Die einzustellenden Kernelparameter werden zeilenweise in der Form name.name.name=wert angegeben. Kommentarzeilen beginnen mit ; oder mit #.

### Beispiel

Die folgende Einstellung in *sysctl.conf* bewirkt, dass beim Rechnerstart das Forwarding für IPv4 und für IPv6 aktiviert wird. Das ist erforderlich, wenn der Rechner als Router arbeiten soll.

```
net.ipv4.ip_forward=1
net.ipv6.conf.all.forwarding=1
```

### systemd.service

Das Init-System *systemd* wird durch eine ganze Palette von Konfigurationsdateien gesteuert, die sich in den drei folgenden Verzeichnissen befinden:

```
/etc/systemd/system
/run/systemd/system
[/usr]/lib/systemd/system
```

Wenn *systemd* auch zur Verwaltung benutzerspezifischer Dienste verwendet wird, gibt es auch \*.service-Dateien in

`/home/name/.config/systemd/user` sowie in einigen weiteren Verzeichnissen.

Die in diesem Abschnitt beschriebenen `*.service`-Dateien legen fest, wie ein Dienst – beispielsweise ein SSH-Server – durch systemd gestartet, überwacht und bei Bedarf auch wieder gestoppt wird. `*.service`-Dateien bestehen aus mehreren Abschnitten, die durch Zeilen mit dem Inhalt `[Unit]`, `[Service]`, `[Install]` eingeleitet werden.

### **[Unit]-Schlüsselwörter**

- `Before/After=name`  
nennt andere Dienste bzw. Units, die vorher oder nachher gestartet werden sollen. `Before` und `After` geben systemd Hinweise, in welcher Reihenfolge die Dienste gestartet werden sollen.
- `Description=beschreibung`  
beschreibt den Dienst durch eine Zeichenkette.
- `OnFailure=name`  
nennt einen oder mehrere Dienste, die ausgeführt werden sollen, wenn der Start des aktuellen Dienstes scheitert.
- `Wants/Requires=name`  
nennt andere Dienste, die vorher gestartet werden sollen bzw. müssen. Diese Schlüsselwörter definieren Abhängigkeiten. systemd versucht, die betreffenden Dienste vorher zu starten. Bei `Wants` wird ein eventuelles Scheitern stillschweigend ignoriert, während es bei `Requires` dazu führt, dass der aktuelle Dienst nicht gestartet und stattdessen ein Fehler ausgelöst wird.

### **[Service]-Schlüsselwörter**

- `Environment=var1='wert1' var2='wert2'`  
definiert mehrere Umgebungsvariablen, die an den zu startenden Prozess weitergegeben werden.
- `EnvironmentFile=dateiname`  
liest die Datei und berücksichtigt die dort zeilenweise

enthaltenden Variablenzuweisungen.

- **ExecReload=/pfad/programm optionen**  
führt das angegebene Kommando aus, um die Konfiguration des Dienstes im laufenden Betrieb neu zu laden.
- **ExecStart=/pfad/programm optionen**  
führt das angegebene Kommando aus, um den Dienst zu starten. In Kombination mit `Type=oneshot` sind mehrere `ExecStart`-Zeilen erlaubt, die dann der Reihe nach ausgeführt werden.
- **ExecStop=/pfad/programm optionen**  
führt das angegebene Kommando aus, um den Dienst zu beenden. Wenn Sie dieses Schlüsselwort nicht verwenden, wird der Dienst durch ein KILL-Signal beendet. Welches Signal wie versendet werden soll, beschreiben bei Bedarf die Schlüsselwörter `KillMode` und `killSignal`, die in der [man](#)-Seite `systemd.kill` beschrieben sind. `ExecStop` ist nur in Kombination mit `Type=oneshot` möglich.
- **RemainAfterExit**  
wird oft in Kombination mit `Type=oneshot` verwendet. `systemd` merkt sich den gerade aktivierten Zustand. Ohne diese Option glaubt `systemd` nach `systemctl start name`, dass die Aktion mit dem Ende des letzten `startExec`-Kommandos beendet ist, und setzt den Status sofort wieder auf `stop`. Ein explizites Ausführen von `systemctl stop` würde dann wirkungslos bleiben.
- **Type=simple|forking|oneshot|dbus|notify|idle**  
gibt an, um welche Art von Dienst es sich handelt bzw. wie das Programm gestartet werden soll.  
Wenn die Typangabe fehlt, gilt `simple`. `systemd` nimmt dann an, dass das mit `ExecStart` angegebene Kommando einen Hintergrunddienst startet. Erst wenn dieses Kommando endet, betrachtet `systemd` den Dienst als regulär beendet.

Type=notify funktioniert so ähnlich, allerdings erwartet systemd eine explizite Benachrichtigung darüber, dass der Startprozess abgeschlossen ist. Der Hintergrunddienst muss dazu die Funktion `sd_notify` oder eine gleichwertige Funktion aufrufen. Auch Type=forking hat Ähnlichkeiten zu simple, allerdings ist hier das ExecStart-Kommando dafür verantwortlich, einen vom Startkommando losgelösten Hintergrundprozess zu starten. Sobald das ExecStart-Kommando abgeschlossen ist, nimmt systemd an, dass der Dienst in einem eigenen Hintergrundprozess läuft.

Beim Typ oneshot nimmt systemd an, dass das auszuführende Kommando so rasch beendet wird, dass darauf gewartet werden kann, bevor andere Aktionen durchgeführt werden. Der Status des Dienstes wird mit dem Ende wieder auf stop gesetzt, es sei denn, die Service-Datei enthält das Schlüsselwort `RemainAfterExit`.

Details zu den weiteren Type-Varianten entnehmen Sie bitte `man systemd.service`.

- `User=username` und `Group=groupname` bestimmt, unter welchem Account das Kommando ausgeführt wird. Systemdienste laufen standardmäßig mit root-Rechten.

## [Install]-Schlüsselwörter

- `WantedBy=` bzw. `RequiredBy=target` gibt an, für welches Target der Dienst wünschenswert oder erforderlich ist, z.B. für das `multi-user.target` oder für das `reboot.target`. In der Praxis wird zumeist `wantedBy` eingesetzt. Das führt dazu, dass das Target auch dann erreicht wird, wenn der Start einzelner Dienste scheitert. `RequiredBy` führt bei Problemen hingegen zu einem Fehler. Das betreffende Target kann dann nicht aktiviert werden.

## Beispiel

Die folgenden Zeilen zeigen die Datei *httpd.service* in der Defaultkonfiguration von Fedora. Sie ist für den Start des Webservers Apache verantwortlich. Voraussetzung für den Start ist die erfolgreiche Aktivierung des Netzwerks und eventueller Netzwerkdateisysteme. Der Webserver soll automatisch im Rahmen des Multi-User-Targets gestartet werden.

```
# Datei /lib/systemd/system/httpd.service (Fedora)
[Unit]
Description=The Apache HTTP Server
Wants=httpd-init.service
After=network.target remote-fs.target nss-lookup.target httpd-init.service
Documentation=man:httpd.service(8)

[Service]
Type=notify
Environment=LANG=C

ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND
ExecReload=/usr/sbin/httpd $OPTIONS -k graceful
# Send SIGWINCH for graceful stop
KillSignal=SIGWINCH
KillMode=mixed
PrivateTmp=true

[Install]
WantedBy=multi-user.target
```

## Weitere systemd-Konfigurationsdateien

Neben den \*.service-Dateien gibt es zur systemd-Konfiguration auch Dateien mit den Kennungen \*.mount, \*.path, \*.socket, \*.target und \*.wants, auf die ich in diesem Buch nicht näher eingehe. Ihre Syntax ist in den [man](#)-Seiten *service.mount*, *service.path* etc. dokumentiert.

Alle diese Dateien zusammen gelten als Unit-Dateien. Die allgemeine Syntax derartiger Units erläutert [man systemd.unit](#). Wenn Sie Jobs einrichten wollen, die durch systemd periodisch ausgeführt werden, müssen Sie sich mit dem Format der \*.timer-Dateien auseinandersetzen (siehe [systemd.timer](#)).

### systemd.timer

systemd kann sich ähnlich wie Cron auch um die regelmäßige Ausführung von Prozessen kümmern. Die Konfiguration erfolgt durch eine *name.timer*-Datei, der eine gleichnamige Service-Datei

gegenüberstehen muss (also *name.service*). Bei selbst definierten Jobs befinden sich beide Dateien üblicherweise im Verzeichnis */etc/systemd/system*.

Der Aufbau einer \*.timer-Datei ist ähnlich wie der einer \*.service-Datei (siehe [systemd.service](#)). Er besteht üblicherweise aus drei Abschnitten: den schon bekannten [Unit]- und [Install]-Abschnitten sowie einem [Timer]-Abschnitt, dessen Einstellungen steuern, wann bzw. wie häufig der Job ausgeführt wird.

### [Timer]-Schlüsselwörter

- **AccuracySec=zeitspanne**  
gibt an, innerhalb welcher Genauigkeit die Jobs ausgeführt werden sollen. Standardmäßig ist dafür ein Zeitfenster von einer Minute vorgesehen, innerhalb dessen die Startzeit zufällig bestimmt wird. Kleinere Zeitspannen sind möglich, sollten aber nur dann verwendet werden, wenn dies wirklich erforderlich ist.
- **OnBootSec=zeitspanne**  
führt den Job erstmals nach der angegebenen Zeitspanne nach dem Bootprozess aus. Wenn die Zeitangabe ohne Einheit erfolgt, sind Sekunden gemeint. Die Syntax der Zeitangaben ist in `man systemd.time` dokumentiert. Zulässig ist z.B. `2h 15min` oder `2weeks` oder `4months`.
- **OnCalendar=zeitangabe**  
gibt an, zu welcher Zeit der Job ausgeführt werden soll. `12:30` bedeutet beispielsweise täglich um 12:30 Uhr. Die Syntax für absolute Zeitangaben ist einigermaßen komplex. Die Details sind in `man systemd.time` im Abschnitt »Calendar Events« ausführlich dokumentiert. So bewirkt `oncalendar=Sun 2020-*-* 17:15` beispielsweise, dass ein Job an jedem Sonntag des Jahres 2020 um 17:15 Uhr ausgeführt wird.
- **OnUnitActiveSec=zeitspanne**  
startet den Job neuerlich, sobald die angegebene Zeitspanne nach dem letzten Start des Jobs vergangen ist.

- `OnUnitInactiveSec=zeitspanne`  
startet den Job neuerlich, sobald die angegebene Zeitspanne nach der letzten Erledigung des Jobs vergangen ist.

## Beispiel

Die folgende Datei `dnf-automatic.timer` kümmert sich darum, dass eine Stunde nach dem Bootvorgang und in der Folge einmal täglich der in der Datei `dnf-automatic.service` beschriebene Dienst ausgeführt wird. (Beide Dateien sind Bestandteil des Fedora-Pakets `dnf-automatic`, das automatisch Updates mit [dnf](#) durchführt.)

```
# Datei /usr/lib/systemd/system/dnf-automatic.timer
# (Fedora, wenn das Paket dnf-automatic installiert ist)
[Unit]
Description=dnf-automatic timer
ConditionPathExists=!/run/ostree-booted
[Timer]
OnBootSec=1h
OnUnitInactiveSec=1d

[Install]
WantedBy=basic.target
```

## /etc/vconsole.conf

Die Datei `/etc/vconsole.conf` enthält bei Distributionen, die eine aktuelle systemd-Version als Init-System verwenden, die Tastatur- und Font-Einstellungen für die Arbeit in Textkonsolen. Bei der Veränderung der Einstellungen hilft das Kommando [localectl](#). Dieses Kommando hat gegenüber einer direkten Veränderung von `vconsole.conf` den Vorteil, dass es das gewünschte Tastaturlayout auch in `/etc/X11/xorg.conf/00-keyboard.conf` speichert, sodass die Einstellung auch für das Grafiksystem gilt.

`vconsole.conf` besteht üblicherweise nur aus zwei Zeilen mit den folgenden Parametern:

- `FONT=name`  
gibt die Schriftart an, die in Textkonsolen verwendet werden soll.  
Passende Font-Dateien befinden sich bei den meisten Distributionen im Verzeichnis `/lib/kbd/consolefonts`.

- KEYMAP=*name*  
gibt das Tastaturlayout an. Die zur Auswahl stehenden Einstellungen können mit `localectl list-keymaps` ermittelt werden.

## Beispiel

Bei einer deutschen Fedora-Installation enthält `vconsole.conf` diese Einstellungen:

```
KEYMAP="de-nodeadkeys"
FONT="eurlatgr"
```

### /etc/wpa\_supplicant/wpa\_supplicant.conf

Der Hintergrunddienst `wpa_supplicant` ist für die Authentifizierung in Funknetzen (WLAN) zuständig. Bei Desktop-Distributionen wird das Programm vom NetworkManager als Backend verwendet; die Konfiguration erfolgt über Einstellungsdialoge von KDE, Gnome etc. Anders sieht die Lage aus, wenn ein Raspberry Pi oder ein anderes Gerät (Embedded Device) ohne grafische Benutzeroberfläche eine WLAN-Verbindung nutzen soll: Dann müssen die WLAN-Authentifizierungsdaten in die Datei

`/etc/wpa_supplicant/wpa_supplicant.conf` eingetragen werden. Die folgenden Zeilen geben dafür ein Beispiel:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=DE

network={
    ssid="wlan-sol3"
    psk="strengGeheim"
    key_mgmt=WPA-PSK
}
```

Die ersten drei Zeilen enthalten die Defaulteinstellungen unter Raspberry Pi OS. Wichtig ist die korrekte `country`-Einstellung. Sie stellt sicher, dass sich der WLAN-Adapter an die Bestimmungen des jeweiligen Landes hält.

Unter Raspberry Pi OS deaktiviert der systemd-Dienst `/lib/systemd/system/wifi-country.service` den WLAN-Controller,

wenn diese Zeile fehlt! Der Länder-Code muss aus zwei Großbuchstaben bestehen.

Anschließend folgen beliebig viele network-Gruppen, für jedes Funknetz eine. Raspberry Pi OS verwendet dann automatisch das stärkste Netzwerk in Funkreichweite. Bei Bedarf können Sie für das Funknetz eine Menge zusätzlicher Parameter definieren:

```
network={  
    ssid="noch-ein-wlan"  
    psk="ganzGeheim"  
    proto=RSN  
    key_mgmt=WPA-PSK  
    pairwise=CCMP  
    auth_alg=OPEN  
}
```

Achten Sie beim Verfassen von `wpa_supplicant.conf` darauf, dass Sie vor und nach den Gleichheitszeichen *keine* Leerzeichen angeben dürfen! Eine Menge Details zu den erlaubten Schlüsselwörtern sowie weitere Beispiele können Sie mit `man wpa_supplicant.conf` sowie in der folgenden Dokumentationsdatei bzw. auf der folgenden Webseite nachlesen:

`/usr/share/doc/wpasupplicant/README.modes.gz`  
[https://wiki.ubuntuusers.de/WLAN/wpa\\_supplicant](https://wiki.ubuntuusers.de/WLAN/wpa_supplicant)

In einfachen Fällen können Sie die erforderlichen Konfigurationzeilen auch mit [wpa\\_passphrase](#) erstellen oder die Konfiguration direkt in `/etc/network/interfaces` durchführen.

### **/etc/X11/xorg.conf**

Die Dateien `/etc/X11/xorg.conf` und `/etc/X11/xorg.conf.d/*.conf` steuern zusammen die Konfiguration des Grafiksystems X. In der Vergangenheit war eine korrekte `xorg.conf`-Datei eine Grundvoraussetzung für den Start des Grafiksystems. Mittlerweile ist `xorg.conf` hingegen optional: Der X-Server ermittelt beim Start die Parameter aller Hardware-Komponenten und führt die Konfiguration selbstständig durch. Bei Bedarf können viele Einstellungen im laufenden Betrieb durch das Kommando [xrandr](#) verändert werden. Das gilt insbesondere für die

Bildschirmauflösung sowie für die Konfiguration mehrerer Bildschirme.

Nur wenn all das nicht funktioniert, z.B. in virtuellen Maschinen oder bei exotischen Hardware-Komponenten, ist eine manuelle Konfiguration erforderlich. Diese müssen Sie in einer Textkonsole durchführen, weil das Grafiksystem ja noch nicht funktioniert.

Insofern erschien es mir zweckmäßig, die Syntax von *xorg.conf* hier zumindest kurz zusammenzufassen, auch wenn sich das Buch in erster Linie an Linux-Anwender richtet, die im Terminal arbeiten.

Abschnitt	Bedeutung
Monitor	Monitordaten (Auflösung, Bildfrequenz)
Device	Konfiguration der Grafikkarte
Screen	Bildschirmauflösung, Multi-Screen-Setup
Files	Dateinamen (z.B. Font-Verzeichnisse)
Module	Zusatzmodule (z.B. freetype, dri)
ServerFlags	verschiedene Server-Optionen
InputClass	Device-Gruppen (z.B. alle Tastaturen)
InputDevice	einzelne Eingabegeräte (z.B. Tastatur, Maus, Touchpad)

**Tabelle 2** »*xorg.conf*«-Abschnitte

Die Datei */etc/X11/xorg.conf* ist in mehrere Abschnitte gegliedert (siehe [Tabelle 3.6](#)), die mit `Section "name"` eingeleitet und mit `EndSection` abgeschlossen werden. In jedem dieser Abschnitte können einzelne Parameter in der Form `Schlüsselwort "Einstellung"` angegeben werden. Die `Identifier`-Zeile gibt dem Abschnitt einen Namen und ermöglicht Querverweise zwischen den Abschnitten. Beispielsweise verweist der Abschnitt `Screen` auf zumindest eine Grafikkarte (`Device`) und einen Monitor, im

folgenden Beispiel auf `mon0` und `dev0`. Eine Referenz der unzähligen weiteren Parameter gibt `man xorg.conf`.

## Beispiel

Die folgenden Zeilen stellen eine Minimalkonfiguration für Notfälle dar, wenn das Grafiksystem gar nicht funktionieren will. Die hier vorgeschlagene Konfiguration verwendet den VESA-Treiber und bietet damit keine 3D-Unterstützung.

```
Section "Monitor"
    Identifier "mon0"
    HorizSync 31 - 94
    VertRefresh 60
EndSection

Section "Device"
    Identifier "dev0"
    Driver "vesa"
EndSection

Section "Screen"
    Identifier "screen0"
    Monitor "mon0"
    Device "dev0"
    DefaultDepth 24
    SubSection "Display"
        Depth 24
        Modes "1024x768"
    EndSubSection
EndSection
```

## /etc/yum.conf

Die Datei `/etc/yum.conf` ist ein Überbleibsel des Paketverwaltungs-Tools [yum](#). Mittlerweile wurde [yum](#) in aktuellen Versionen von CentOS, Fedora, RHEL etc. durch [dnf](#) ersetzt. Die Konfiguration erfolgt nun durch die Datei `/etc/dnf/dnf.conf`. Bei der noch immer vorhandenen Datei `yum.conf` handelt es sich nur noch um einen Link auf `dnf.conf`.

## Tastenkürzel

Im letzten Abschnitt der *Linux Kommandoreferenz* geht es um die Tastenkürzel der wichtigsten Editoren und anderer Kommandos, die üblicherweise über die Tastatur bedient werden. Dazu zählen z.B. `bash`, `man`, `info`, `less` und `mutt`.

Nahezu alle Programme bieten die Möglichkeit, eigene Tastenkürzel zu definieren. Dieser Abschnitt bezieht sich auf die Defaultkonfiguration, die bei den meisten Linux-Distributionen standardmäßig gilt.

### **bash**

[Tabelle 6](#) fasst zusammen, welche Tastenkürzel Sie innerhalb der Bourne Again Shell (`bash`) bei der Eingabe von Kommandos verwenden können. Die Tastenkürzel stammen eigentlich von der `readline`-Bibliothek.

Die Konfiguration dieser Bibliothek können Sie in `/etc/inputrc` bzw. `.inputrc` verändern.

Tastenkürzel	Funktion
<code>Strg + A</code>	bewegt den Cursor an den Zeilenanfang (wie <code>Pos1</code> ).
<code>Strg + C</code>	bricht das laufende Kommando ab.
<code>Strg + E</code>	setzt den Cursor an das Ende der Zeile (wie <code>Ende</code> ).
<code>Strg + K</code>	löscht den Rest der Zeile ab der Cursorposition.
<code>Strg + Y</code>	fügt den zuletzt gelöschten Text wieder ein.
<code>Strg + Z</code>	unterbricht das laufende Kommando (Fortsetzung mit <code>fg</code> oder <code>bg</code> ).
	vervollständigt Datei- und Kommandonamen.
	blättert durch die bisher ausgeführten Kommandos.

## Tabelle 6 Tastenkürzel zur Kommandoeingabe in der bash

### emacs

Der Emacs zählt zu den funktionsreichsten und komplexesten Editoren, die unter Linux zur Verfügung stehen. Es gibt Hunderte von Tastenkürzeln und Kommandos, von denen hier natürlich nur die allerwichtigsten präsentiert werden.

Generell gibt es drei Möglichkeiten zur Eingabe von Emacs-Kommandos: das Menü, die Verwendung von Tastenkürzeln (zumeist eine Kombination mit **Strg** oder **Alt**) oder die Eingabe des gesamten Kommandonamens. Die dritte Variante wird mit

**Alt**+**X** oder **Esc** eingeleitet, also etwa **Alt**+**X** delete-char **←**. Die Eingabe von Kommandos und anderen Parametern wird durch zwei Mechanismen erleichtert:

- Während der Eingabe können Sie den Namen eines Emacs-Kommandos mit **→** ergänzen. In gleicher Weise können auch Dateinamen ergänzt werden.
- Auf früher mit **Alt**+**X** angegebene Kommandos können Sie nach der Einleitung des neuen Kommandos durch **Alt**+**X** mit **Alt**+**P** (Previous) und **Alt**+**N** (Next) zurückgreifen.

In der Dokumentation zum Emacs werden Tastenkürzel etwas abweichend dargestellt: **DEL** bedeutet nicht **Entf**, sondern **←**! **c** steht für Control (gemeint ist **Strg**) und **M** für **Meta**. Eine direkte Entsprechung der Meta-Taste existiert auf einer Standard-PC-Tastatur nicht. **M-x** kann auf einer PC-Tastatur auf zwei Weisen nachgebildet werden: durch **Esc** und **X** (nacheinander) oder durch **Alt**+**X**.

[Tabelle 7](#) fasst die Grundfunktionen zusammen. Zur Cursorbewegung können Sie die Cursortasten sowie diverse Tastenkürzel verwenden (siehe [Tabelle 8](#)).

Tastenkürzel	Funktion
--------------	----------

Tastenkürzel	Funktion
<code>Strg + X ,</code> <code>Strg + F</code>	lädt eine neue Datei.
<code>Strg + X ,</code> <code>Strg + S</code>	speichert die aktuelle Datei.
<code>Strg + X ,</code> <code>Strg + W</code>	speichert die Datei unter einem neuen Namen.
<code>Strg + G</code>	bricht die Eingabe eines Kommandos ab.
<code>Strg + X , U</code>	macht die letzte Änderung rückgängig (Undo).
<code>Strg + X ,</code> <code>Strg + C</code>	beendet den Emacs (mit Rückfrage zum Speichern).

**Tabelle 7** Elementare Emacs-Kommandos

Tastenkürzel	Funktion
<code>Alt + F / Alt + B</code>	bewegt den Cursor ein Wort vor bzw. zurück.
<code>Strg + A / Strg + E</code>	stellt den Cursor an den Zeilenbeginn bzw. dessen Ende.
<code>Alt + &lt; / Alt + ↑ + &gt;</code>	bewegt den Cursor an den Beginn bzw. das Ende des Textes.
<code>Alt + G n ↵</code>	bewegt den Cursor in Zeile <i>n</i> .

**Tabelle 8** Cursorbewegung

[Tabelle 9](#) gibt an, wie Sie Text markieren, löschen und wieder einfügen, und [Tabelle 10](#) fasst zusammen, wie Sie suchen und ersetzen.

Tastenkürzel	Funktion
<code>Strg + </code>	setzt einen (unsichtbaren) Markierungspunkt.

Tastenkürzel	Funktion
<b>Strg + W</b>	löscht den Text zwischen dem Markierungspunkt und der aktuellen Cursorposition.
<b>Strg + Y</b>	fügt den gelöschten Text wieder ein.
<b>Strg + X</b> , <b>Strg + X</b>	vertauscht Cursorposition und Markierungspunkt.
<b>Alt + D</b>	löscht das nächste Wort bzw. das Ende des Wortes ab dem Cursor.
<b>Alt + ←</b>	löscht das vorige Wort bzw. den Beginn des Wortes bis zum Cursor.
<b>Strg + K</b>	löscht den Rest der Zeile ab der Cursorposition.
<b>Alt + 0</b> , <b>Strg + K</b>	löscht den Zeilenanfang vor der Cursorposition.
<b>Alt + M</b>	löscht den nächsten Absatz.
<b>Alt + Z</b> , x	löscht alle Zeichen bis zum nächsten Auftreten von x. Das Zeichen x wird mitgelöscht.
<b>Strg + Y</b>	fügt den zuletzt gelöschten Text an der Cursorposition wieder ein.

**Tabelle 9** Text markieren, löschen und wieder einfügen

Tastenkürzel	Funktion
<b>Strg + S</b>	inkrementelle Suche vorwärts
<b>Strg + R</b>	inkrementelle Suche rückwärts
<b>Alt + P</b>	wählt einen früher verwendeten Suchtext aus (Previous).
<b>Alt + N</b>	wählt einen später verwendeten Suchtext aus (Next).

Tastenkürzel	Funktion
<b>Strg</b> + <b>G</b>	Abbruch der Suche
<b>Strg</b> + <b>X</b> , <b>Strg</b> + <b>X</b>	vertauscht den Markierungspunkt (Beginn der Suche) und die aktuelle Cursorposition.
<b>Strg</b> + <b>Alt</b> + <b>S</b>	inkrementelle Mustersuche vorwärts
<b>Strg</b> + <b>Alt</b> + <b>R</b>	inkrementelle Mustersuche rückwärts
<b>Alt</b> + <b>%</b>	Suchen und Ersetzen ohne Muster
<b>Alt</b> + <b>X</b> query-replace-r <b>↔</b>	Suchen und Ersetzen mit Muster

**Tabelle 10** Suchen und Ersetzen

### **fdisk**

**fdisk** ist ein interaktives Programm zur Partitionierung von Festplatten. Eine Beschreibung der wichtigsten Optionen sowie ein längeres Anwendungsbeispiel finden Sie in der alphabetischen Kommandoreferenz. [Tabelle 11](#) fasst lediglich die Tastenkürzel zur Bedienung des Programms zusammen.

Tastenkürzel	Bedeutung
<b>D</b>	Partition löschen ( <i>delete</i> )
<b>L</b>	Partitions-ID-Nummer anzeigen ( <i>list</i> )
<b>M</b>	Online-Hilfe ( <i>menu</i> )
<b>N</b>	neue Partition anlegen ( <i>new</i> )
<b>P</b>	Partitionsliste anzeigen ( <i>print</i> )
<b>Q</b>	Programm beenden (ohne die Partitionstabelle zu verändern; <i>quit</i> )
<b>T</b>	Partitionstyp verändern

Tastenkürzel	Bedeutung
[U]	Maßeinheit zwischen Zylindern und Sektoren umschalten ( <i>unit</i> )
[V]	Partitionstabelle überprüfen ( <i>verify</i> )
[W]	Partitionstabelle ändern ( <i>write</i> )

**Tabelle 11** fdisk-Tastenkürzel

#### gnome-terminal

Wenn Sie Shell-Kommandos unter Gnome ausführen, verwenden Sie dazu höchstwahrscheinlich das Programm `gnome-terminal`. Damit Sie die in der bash üblichen Tastenkürzel verwenden können, sollten Sie als Erstes mit BEARBEITEN • EINSTELLUNGEN die Option MENÜKÜRZELBUCHSTABEN zur Steuerung der Menüs durch [Alt]-Tastenkürzel deaktivieren.

Zur Menüsteuerung können Sie dann bei Bedarf immer noch die Taste [F10] verwenden – es sei denn, Sie deaktivieren auch die Verarbeitung dieser Taste im gerade erwähnten Konfigurationsdialog. Einige `gnome-terminal`-spezifische Tastenkürzel bleiben auf jeden Fall verfügbar; sie sind in [Tabelle 12](#) zusammengefasst.

Tastenkürzel	Funktion
[↑] + [Strg] + [C]	kopiert den markierten Text in die Zwischenablage.
[↑] + [Strg] + [F]	sucht einen Text in den Terminalausgaben.
[↑] + [Strg] + [G]	wiederholt die Suche rückwärts.
[↑] + [Strg] + [H]	wiederholt die Suche vorwärts.
[↑] + [Strg] + [N]	öffnet ein neues Terminalfenster.
[↑] + [Strg] + [Q]	schließt das Fenster.

Tastenkürzel	Funktion
⬆ + Strg + T	öffnet einen neuen Terminal-Reiter.
⬆ + Strg + V	fügt den Inhalt der Zwischenablage ein.
⬆ + Strg + W	schließt den Reiter.
⬆ + Strg + +	vergrößert die Schrift.
⬆ + Strg + -	verkleinert die Schrift.
⬆ + Strg + Bild ↑ / Bild ↓	wechselt in den vorigen/nächsten Reiter.
F11	aktiviert bzw. deaktiviert den Vollbildmodus.

**Tabelle 12** Tastenkürzel in »gnome-terminal«

### grub

Im Linux-Bootloader GRUB können Sie mit den Cursortasten ein Betriebssystem bzw. eine Linux-Variante auswählen und diese dann durch ↪ starten. Darüber hinaus bietet GRUB die Möglichkeit, die Parameter eines Menüeintrags interaktiv zu verändern oder eigene Kommandos auszuführen. [Tabelle 13](#) fasst hierfür die wichtigsten Tastenkürzel zusammen. Die Tabelle bezieht sich dabei auf die aktuelle GRUB-Version 2.

Tastenkürzel	Funktion
Esc	beendet den Grafikmodus und aktiviert den Textmodus.
C	startet den Kommandomodus zur interaktiven Ausführung von GRUB-Kommandos. Bei der Kommandoeingabe können Dateinamen wie in der Shell durch ↪ vervollständigt werden.

Tastenkürzel	Funktion
[E]	startet den Editor für den ausgewählten Menüeintrag.
[P]	gibt die interaktiven GRUB-Funktionen durch die Eingabe eines Passworts frei. Das ist nur erforderlich, wenn GRUB durch ein Passwort abgesichert ist.
[Strg] + [X] oder [F10]	startet den zuvor mit [E] veränderten Menüeintrag.

**Tabelle 13** Tastenkürzel zur interaktiven Steuerung von GRUB 2  
**info**

info startet das gleichnamige Online-Hilfesystem. Zur Navigation im Hilfetext verwenden Sie die in [Tabelle 14](#) zusammengefassten Tastenkürzel. info-Texte können Sie mit mehr Komfort auch mit dem Kommando pinfo aus dem gleichnamigen Paket, mit dem Editor Emacs oder in den Hilfesystemen von Gnome und KDE lesen.

Tastenkürzel	Funktion
[↓]	scrollt Text nach unten.
[↑]	scrollt Text nach oben.
[B], [E]	springt zum Anfang/Ende der Info-Einheit ( <i>beginning/end</i> ).
[→]	bewegt den Cursor zum nächsten Querverweis.
[←]	verfolgt einen Querverweis zu einer anderen Info-Einheit.
[N]	zeigt die nächste Info-Einheit in derselben Hierarchiestufe an ( <i>next</i> ).

Tastenkürzel	Funktion
[P]	zeigt die vorige Info-Einheit in derselben Hierarchiestufe an ( <i>previous</i> ).
[U]	springt eine Hierarchieebene nach oben ( <i>up</i> ).
[L]	springt zurück zum zuletzt angezeigten Text ( <i>last</i> ).
[H]	zeigt eine ausführliche Bedienungsanleitung an ( <i>help</i> ).
[?]	zeigt eine Kommandoübersicht an.
[Q]	beendet <code>info</code> ( <i>quit</i> ).

**Tabelle 14** »info«-Tastenkürzel

### joe

joe ist ein einfacher Editor, dessen Tastenkürzel dem Textverarbeitungsprogramm Wordstar nachempfunden sind (siehe [Tabelle 15](#)). Der Editor kann auch unter den Namen `jmacs` oder `jpico` gestartet werden. Es gelten dann andere Tastenkürzel, die zum Emacs bzw. zu Pico kompatibel sind.

Tastenkürzel	Funktion
[Strg] + [K], [H]	blendet das Hilfefenster ein/aus.
[Strg] + [K], [E]	lädt eine neue Datei.
[Strg] + [K], [D]	speichert die Datei (wahlweise unter neuem Namen).
[Strg] + [Y]	löscht eine Zeile.
[Strg] + [↑] + [-]	macht das Löschen rückgängig (Undo).
[Strg] + [C]	beendet joe (mit Rückfrage zum Speichern).

**Tabelle 15** »joe«-Tastenkürzel

### konsole

KDE-Fans führen Shell-Kommandos in der Regel im Programm konsole aus. Die meisten Tastenkürzel werden von diesem Programm direkt an die Shell weitergegeben. Darüber hinaus gibt es aber einige konsole-spezifische Tastenkürzel, die in [Tabelle 16](#) zusammengefasst sind.

Tastenkürzel	Funktion
+  +	markiert den Reiter bei Aktivität.
+  +	kopiert den markierten Text in die Zwischenablage.
+  +	sucht einen Text in den Terminalausgaben.
	wiederholt die Suche rückwärts.
+	wiederholt die Suche vorwärts.
+  +	markiert den Reiter bei längerer Inaktivität.
+  +	öffnet ein neues Terminalfenster.
+  +	schließt das Fenster.
+  +	öffnet einen neuen Terminal-Reiter.
+  +	fügt den Inhalt der Zwischenablage ein.
+  +	schließt den Reiter.
+  +	vergrößert die Schrift.
+  +	verkleinert die Schrift.
+  +  /	wechselt in den vorigen/nächsten Reiter.

**Tabelle 16** Tastenkürzel in »konsole«

### less

Das Kommando less zeigt Texte an. Während das Programm läuft, können Sie mit den Cursortasten durch den Text scrollen, Texte

suchen, den durch die Umgebungsvariable \$EDITOR eingestellten Editor starten etc. (siehe [Tabelle 17](#)). less wird auch zur Anzeige von man-Hilfetexten verwendet.

Tastenkürzel	Funktion
Cursortasten	scrollt den Text nach oben oder unten.
<code>[Pos1], [Ende]</code>	springt an den Beginn bzw. das Ende des Textes.
<code>[G], [↑] + [G]</code>	springt an den Beginn bzw. das Ende des Textes.
<code>/ muster</code> <code>[←]</code>	sucht vorwärts.
<code>? muster</code> <code>[←]</code>	sucht rückwärts.
<code>[N]</code>	wiederholt die Suche vorwärts ( <i>next</i> ).
<code>[↑] + [N]</code>	wiederholt die Suche rückwärts.
<code>[V]</code>	startet den durch \$EDITOR oder \$VISUAL eingestellten Editor.
<code>[Q]</code>	beendet less ( <i>quit</i> ).
<code>[H]</code>	zeigt einen Hilfetext mit weiteren Tastenkürzeln an.

**Tabelle 17** »less«-Tastenkürzel

#### man

Das Kommando `man` zeigt die Dokumentation zu wichtigen Kommandos, Konfigurationsdateien, C-Funktionen etc. an. Um ein einfaches Blättern durch den Hilfetext zu ermöglichen, greift `man` auf das Kommando `less` zurück. Deswegen gelten innerhalb von `man` dieselben Tastenkürzel wie bei `less` (siehe [Tabelle 18](#)).

#### mutt

`mutt` ist ein E-Mail-Client für den Textmodus. Das Programm ist gut dazu geeignet, lokal auf dem Rechner gespeicherte E-Mails zu lesen

und zu beantworten. Hardcore-Linux-Fans bevorzugen `mutt` sogar in grafischen Benutzeroberflächen, weil sich das Programm besonders effizient per Tastatur steuern lässt. [Tabelle 18](#) enthält nur die wichtigsten Tastenkürzel.

Tastenkürzel	Funktion
Cursortasten	bewegt den Cursor durch die Mail-Liste.
	zeigt die ersten Zeilen der ausgewählten E-Mail an.
	blättert durch den Text einer Mail.
	löscht die E-Mail.
	wechselt von der E-Mail-Ansicht zurück in die Inbox.
	zeigt die nächste E-Mail an.
	verfasst eine neue E-Mail mit dem durch <code>\$EDITOR</code> oder <code>\$VISUAL</code> eingestellten Editor.
	beantwortet die E-Mail.
	zeigt einen Hilfetext mit allen Tastenkürzeln an.

**Tabelle 18** »mutt«-Tastenkürzel

### nano

`nano` ist ein minimalistischer Editor, der vor allem für Einsteiger gut geeignet ist. Die wichtigsten Tastenkürzel (siehe auch [Tabelle 19](#)) werden ständig in den zwei untersten Textzeilen eingeblendet.

Tastenkürzel	Funktion
<code>Strg + A</code>	bewegt den Cursor an den Beginn der Zeile.
<code>Strg + D</code>	löscht ein Zeichen.
<code>Strg + E</code>	bewegt den Cursor zum Ende der Zeile.
<code>Strg + H</code>	löscht ein Zeichen rückwärts.

Tastenkürzel	Funktion
Strg + ^	setzt einen Markierungspunkt.
Strg + K	löscht die aktuelle Zeile oder den markierten Text.
Strg + U	fügt den gelöschten Text wieder ein.
Strg + R	fügt eine Textdatei in den Text ein.
Strg + O	speichert die Datei.
Strg + X	beendet den Editor.

**Tabelle 19** »nano«-Tastenkürzel

### screen

Mit dem Kommando screen können Sie mehrere Terminal-Sessions parallel ausführen (*multiplexen*). screen wird durch Tastenkürzel gesteuert, die alle mit Strg + A beginnen.

Tastenkürzel	Funktion
Strg + A , ?	zeigt die Online-Hilfe an.
Strg + A , *	listet alle Sessions auf.
Strg + A , 0 bis 9	wechselt zwischen der ersten und der zehnten Session.
Strg + A , C	erzeugt eine neue Session ( <i>create</i> ).
Strg + A , D	trennt die aktuelle Session von screen ( <i>detach</i> ).
Strg + A , ↑ + H	aktiviert/deaktiviert das Logging.
Strg + A , K	beendet die aktive Session.
Strg + A , N	wechselt in die nächste Session.
Strg + A , \	beendet alle Sessions und screen.

**Tabelle 20** Tastenkürzel in Textkonsolen**Textkonsole**

Wenn Sie direkt in Textkonsolen arbeiten, also nicht unter KDE, Gnome oder einem anderen Desktop-System und auch nicht via SSH, dann gelten dort einige besondere Tastenkürzel. [Tabelle 21](#) fasst die wichtigsten Kürzel zusammen.

Tastenkürzel	Funktion
<code>Strg + Alt + Fn</code>	wechselt vom Grafikmodus in die Textkonsole <i>n</i> .
<code>Alt + F1</code>	wechselt zurück in den Grafikmodus.
<code>Alt + Fn</code>	wechselt von einer Textkonsole in eine andere Textkonsole <i>n</i> .
<code>Alt + → / ←</code>	wechselt in die vorige/nächste Textkonsole.
<code>↑ + Bild ↑ / Bild ↓</code>	scrollt seitenweise vorwärts/rückwärts.
<code>Strg + Alt + Entf</code>	beendet Linux durch <code>shutdown</code> (Vorsicht!).

**Tabelle 21** Tastenkürzel in Textkonsolen**vi/vim**

Der Editor Vi ist ein Urgestein der Unix-Geschichte. Nahezu alle Linux-Distributionen installieren standardmäßig das zum Vi kompatible Programm vim, das Sie sowohl mit vi als auch mit vim starten können.

Tastenkürzel	Funktion
<code>I</code>	aktiviert den Einfügemodus.
<code>A</code>	aktiviert den Einfügemodus. Die Texteingabe beginnt beim nächsten Zeichen.
<code>Esc</code>	aktiviert den Standardmodus bzw. bricht die Kommandoeingabe ab.

Tastenkürzel	Funktion
Kommandos im Standardmodus	
[D], [W]	löscht ein Wort.
[D], [D]	löscht die aktuelle Zeile.
n [D], [D]	löscht n Zeilen.
[P]	fügt den zuletzt gelöschten Text hinter der Cursorposition ein.
[↑] + [P]	fügt den zuletzt gelöschten Text vor der Cursorposition ein.
[.]	wiederholt das letzte Kommando.
[U]	macht die letzte Änderung rückgängig (Undo).
[↑] + [U]	widerruft alle Änderungen in der aktuellen Zeile.
[Strg] + [R]	macht Undo rückgängig (Redo, ab Vim 7).
[:] w	speichert die Datei.
[:] q	beendet vim.
[:] q!	beendet vim auch dann, wenn es nicht gespeicherte Dateien gibt.
Kommandos im Einfügemodus	
[Strg] + [0] kommando	führt das Kommando aus, ohne den Einfügemodus zu verlassen.

**Tabelle 22** Elementare Kommandos

Der Vi bietet ähnlich viele Funktionen wie der Emacs, die Bedienung ist aber noch schwieriger zu erlernen. [Tabelle 22](#) fasst die elementarsten Vi-Kommandos zusammen. Der wichtigste fundamentale Unterschied zu anderen Editoren besteht darin, dass der Vi zwischen verschiedenen Modi unterscheidet:

- **Insert-Modus**: Um Text einzugeben, müssen Sie mit **I** (*insert*) oder **A** (*append*) in den Einfügemodus wechseln. `vim` zeigt nun in der untersten Zeile ganz links den Text -- EINFÜGEN -- an. Im Einfügemodus können Sie Text eingeben, den Cursor bewegen und einzelne Zeichen löschen (**Entf** und **←**). Der Unterschied zwischen **I** und **A** besteht darin, dass die Eingabe bei **I** an der aktuellen Cursorposition beginnt, bei **A** beim Zeichen dahinter.
- **Complex-Command-Modus**: Die Eingabe der meisten Kommandos erfolgt hingegen im Complex-Command-Modus, der mit **:** aktiviert wird. Vorher muss gegebenenfalls der Insert-Modus durch **Esc** verlassen werden.

[Tabelle 23](#) gibt einen Überblick über die wichtigsten Kürzel zum Löschen von Text. [Tabelle 3.24](#) listet nützliche Kommandos zur Cursorbewegung auf. In [Tabelle 25](#), [Tabelle 26](#) und [Tabelle 27](#) geht es darum, wie Sie Text kopieren, markieren und bearbeiten.

### Tastenkürzel im Einfügemodus

<b>Entf</b> , <b>←</b>	Diese Tasten haben die übliche Bedeutung.
------------------------	---

#### Kommandos im Standardmodus

<b>X</b>	löscht das Zeichen an der Cursorposition bzw. den markierten Text.
<b>↑</b> + <b>X</b>	löscht das Zeichen vor dem Cursor.
<b>D</b> , <b>D</b>	löscht die aktuelle Zeile.
<b>D</b> <i>cursorkommando</i>	löscht den Text entsprechend dem Kommando zur Cursorbewegung (siehe <a href="#">Tabelle 3.24</a> ). Beispiele: <b>D</b> , <b>\$</b> löscht bis zum Ende der Zeile. <b>D</b> , <b>B</b> löscht das vorige Wort. <b>D</b> , <b>W</b> löscht das nächste Wort.

**Tabelle 23** Text löschen

Tastenkürzel	Funktion
Cursortasten	Die Cursortasten haben die übliche Bedeutung.
[H] / [L]	bewegt den Cursor nach links/rechts.
[J] / [K]	bewegt den Cursor nach unten/oben.
[↑] + [H] / [↑] + [L]	bewegt den Cursor an den Beginn bzw. das Ende der aktuellen Seite.
[↑] + [M]	bewegt den Cursor in die Mitte der aktuellen Seite.
[B] / [W]	bewegt den Cursor um ein Wort nach links/rechts.
[E]	bewegt den Cursor an das Ende des Worts.
[G], [E]	bewegt den Cursor an den Anfang des Worts.
(, )	bewegt den Cursor an den Beginn des aktuellen/nächsten Satzes.
{, }	bewegt den Cursor an den Beginn des aktuellen/nächsten Absatzes.
[^], [\$]	bewegt den Cursor an den Beginn bzw. das Ende der Zeile.
[↑] + [G]	bewegt den Cursor an das Ende der Datei.
[G], [G]	bewegt den Cursor an den Beginn der Datei.
n [↑] + [G]	bewegt den Cursor in die Zeile n.
n [ ]	bewegt den Cursor in die Spalte n.
[%]	bewegt den Cursor zum korrespondierenden Klammerzeichen ()[]{}.
Tastenkürzel	Funktion
[V]	(de)aktiviert den Zeichenmarkierungsmodus.

Tastenkürzel	Funktion
<b>↑</b> + <b>V</b>	(de)aktiviert den Zeilenmarkierungsmodus.
<b>Strg</b> + <b>V</b>	(de)aktiviert den Blockmarkierungsmodus.
<b>A</b> , <b>W</b>	vergrößert die Markierung um ein Wort.
<b>A</b> , <b>S</b>	vergrößert die Markierung um einen Satz.
<b>A</b> , <b>P</b>	vergrößert die Markierung um einen Absatz.
<b>A</b> , <b>B</b>	vergrößert die Markierung um eine ()-Ebene.
<b>A</b> , <b>↑</b> + <b>B</b>	vergrößert die Markierung um eine {}-Ebene.
<b>G</b> , <b>V</b>	markiert den zuletzt markierten Text nochmals.
<b>O</b>	wechselt die Cursorposition zwischen Markierungsanfang und -ende.

**Tabelle 25** Text markieren

Tastenkürzel	Funktion
<b>Y</b>	kopiert den markierten Text in das Kopierregister.
<b>Y</b> , <b>Y</b>	kopiert die aktuelle Zeile in das Kopierregister.
<b>Y</b> <i>cursorkommando</i>	kopiert den durch die Cursorbewegung erfassten Text; Beispiel: <b>Y</b> , <b>}</b> kopiert den Text bis zum Ende des Absatzes.

**Tabelle 26** Text in das Kopierregister kopieren

Tastenkürzel	Funktion
<b>X</b>	löscht den markierten Text.
<b>Y</b>	kopiert den markierten Text in das Kopierregister.
<b>~</b>	ändert die Groß-/Kleinschreibung.

Tastenkürzel	Funktion
[J]	fügt die markierten Zeilen zu einer langen Zeile zusammen.
[G], [Q]	führt einen Zeilenumbruch durch (für Fließtext).
[>], [<]	rückt den Text um eine Tabulatorposition ein oder aus.
[=]	rückt den Text dem aktuellen indent-Modus entsprechend neu ein.
!sort	sortiert die Zeilen mit dem externen Kommando sort.

**Tabelle 27** Markierten Text bearbeiten

### Suchen und ersetzen

Im Standardmodus bewegt [/] suchtext [←] den Cursor zum gesuchten Text. [N] wiederholt die Suche, [↑]+[N] wiederholt die Suche rückwärts. Um von vornherein rückwärts zu suchen, beginnen Sie die Suche mit [?] suchausdruck. [Tabelle 28](#) fasst die wichtigsten Sonderzeichen zusammen, um nach Mustern zu suchen.

Zeichen	Bedeutung
.	ein beliebiges Zeichen
^ \$	Zeilenanfang/Zeilenende
\< \>	Wortanfang/Wortende
[a-e]	ein Zeichen zwischen a und e
\s, \t	ein Leerzeichen bzw. ein Tabulatorzeichen
\( \)	fasst ein Suchmuster als Gruppe zusammen.
\=	Der Suchausdruck muss 0- oder einmal auftreten.

Zeichen	Bedeutung
*	Der Suchausdruck darf beliebig oft (auch 0-mal) auftreten.
\+	Der Suchausdruck muss mindestens einmal auftreten.

**Tabelle 28** Sonderzeichen im Suchausdruck

Vim unterscheidet bei der Suche zwischen Groß- und Kleinschreibung. Wenn Sie das nicht möchten, leiten Sie das Suchmuster mit /c ein (gilt nur für diese Suche) oder führen [:] ignorecase aus (gilt für alle weiteren Suchen).

Mit [:] set incsearch aktivieren Sie die inkrementelle Suche: Bereits während der Eingabe des Suchtextes durch [/] suchausdruck bewegt Vim den Cursor zum ersten passenden Ort. [←] beendet die Suche, [Esc] bricht sie ab. Nach der Suche bleiben alle Übereinstimmungen im Text markiert, bis Sie eine neue Suche durchführen oder [:] nohlsearch ausführen.  
Um alle Vorkommen des Texts *abc* ohne Rückfrage durch *efg* zu ersetzen, führen Sie im Standardmodus [:] %s/abc/efg/g aus. ['] führt anschließend zurück an den Beginn der Suche. [Tabelle 29](#) fasst einige Varianten des Suchen-und-ersetzen-Kommandos zusammen.

Tastenkürzel	Funktion
[:] %s/abc/efg/g	ersetzt ohne Rückfrage alle Vorkommen von <i>abc</i> durch <i>efg</i> .
[:] %s/abc/efg/gc	ersetzt mit Rückfrage alle Vorkommen von <i>abc</i> durch <i>efg</i> .
[:] %s/abc/efg/gci	ersetzt ohne Berücksichtigung der Groß- und Kleinschreibung.

**Tabelle 29** Suchen und ersetzen

## **Rechtliche Hinweise**

Das vorliegende Werk ist in all seinen Teilen urheberrechtlich geschützt. Weitere Hinweise dazu finden Sie in den Allgemeinen Geschäftsbedingungen des Anbieters, bei dem Sie das Werk erworben haben.

## **Markenschutz**

Die in diesem Werk wiedergegebenen Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. können auch ohne besondere Kennzeichnung Marken sein und als solche den gesetzlichen Bestimmungen unterliegen.

## **Haftungsausschluss**

Ungeachtet der Sorgfalt, die auf die Erstellung von Text, Abbildungen und Programmen verwendet wurde, können weder Verlag noch Autor, Herausgeber, Übersetzer oder Anbieter für mögliche Fehler und deren Folgen eine juristische Verantwortung oder irgendeine Haftung übernehmen.

## Über den Autor



**Michael Kofler** studierte Telematik an der TU Graz. Er ist der erfolgreichsten und vielseitigsten Computerbuchautoren. Zu seinen Themengebieten zählen neben Linux auch Python, Java, Kotlin und die IT-Sicherheit. Viele seiner Bücher wurden übersetzt. **Michael Kofler** arbeitet zudem als Software-Entwickler, Berater sowie als Lehrbeauftragter an zwei Fachhochschulen.

## Dokumentenarchiv

Das Dokumentenarchiv umfasst alle Abbildungen und ggf. Tabellen und Fußnoten dieses E-Books im Überblick.