

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

6., aktualisierte Auflage

 **Rheinwerk**
Computing

Thematische Kommandoübersicht

Dateien verwalten

cat	verbindet mehrere Dateien zu einer Gesamtdatei	66
cd	wechselt in ein anderes Verzeichnis	68
chgrp	ändert die Gruppenzugehörigkeit einer Datei	72
chmod	ändert die Zugriffsbits einer Datei	73
chown	ändert den Besitzer einer Datei	75
cp	kopiert Dateien	82
dircolors	hilft bei der Konfiguration der ls-Farben	102
erd	kombiniert ls, tree, find und du	122
file	versucht, den Typ einer Datei festzustellen	132
inotifywait	wartet auf die Veränderung von Dateiattributen	191
install	kombiniert cp mit chmod und chown	193
j	wechselt in ein anderes Verzeichnis	205
ln	stellt feste und symbolische Links zu Dateien her	218
ls	zeigt das Inhaltsverzeichnis an	226
mkdir	erzeugt ein neues Verzeichnis	244
mv	verschiebt Dateien bzw. ändert ihre Namen	269
namei	zeigt die Zugriffsrechte aller Verzeichnisse einer Datei	278
rdfind	sucht und beseitigt Doppelgänger	347
rename	gibt mehreren Dateien neue Namen	351
rm	löscht Dateien	357
rmdir	löscht Verzeichnisse	360
rsync	synchronisiert Verzeichnisse	365
stat	liefert detaillierte Informationen zu Dateien	399
tee	dupliziert die Standardeingabe	415
touch	erzeugt eine leere Datei bzw. ändert den Modify-Zeitpunkt	422
tree	stellt die Verzeichnishierarchie dar	425
truncate	verkleinert/vergrößert eine Datei	425
umask	steuert, welche Zugriffsrechte neue Dateien und Verzeichnisse erhalten	429
z	wechselt in ein anderes Verzeichnis	470

Dateien suchen

ack	schnelle grep-Alternative für Programmierer	26
erd	kombiniert ls, tree, find und du	122
find	sucht Dateien nach Name, Datum, Größe etc.	132
fzf	ermöglicht eine interaktive Suche nach Dateien	153
grep	sucht Text in einer Textdatei	171
locate	sucht Dateien in einer dafür vorbereiteten Datenbank ...	221
updatedb	aktualisiert die Suchdatenbank für locate	435
whereis	sucht Dateien in vordefinierten Verzeichnissen	457
which	durchsucht die PATH-Verzeichnisse nach Kommandos	457

Dateien komprimieren und archivieren

bunzip2	dekomprimiert *.bz2-Dateien	63
bzip2	komprimiert Dateien; leistungsfähiger als gzip	64
compress	komprimiert Dateien	79
cpio	überträgt Archivdateien zwischen Dateisystemen	84
gunzip	dekomprimiert *.gz-Dateien	179
gzip	komprimiert Dateien; leistungsfähiger als compress	179
lz4	komprimiert Dateien; deutlich schneller als gzip	238
lzop	komprimiert Dateien; deutlich schneller als gzip	238
tar	bildet ein *.tar-Archiv	409
uncompress	dekomprimiert durch compress komprimierte Dateien ...	431
unxz	dekomprimiert *.xz-Dateien	433
unzip	dekomprimiert *.zip-Dateien	433
xz	komprimiert Dateien; leistungsfähiger als bzip2	469
zip	erzeugt ein Windows-kompatibles ZIP-Archiv	472
zipinfo	zeigt den Inhalt eines ZIP-Archivs an	472

Textdateien verarbeiten

awk	Programmiersprache zur Textauswertung	49
bat	zeigt eine Textdatei mit Syntaxhighlighting an	54
cat	gibt eine Datei aus bzw. vereint mehrere Texte	66
column	formatiert Informationen spaltenweise	79
csplit	zerlegt den Text an vorgegebenen Stellen in Einzeldateien	87
cut	extrahiert Spalten aus jeder Zeile des Textes	90

delta	vergleicht zwei Texte und kennzeichnet die Unterschiede farbig	96
diff	vergleicht zwei Texte	100
expand	ersetzt Tabulator- durch Leerzeichen	127
fold	zerlegt lange Textzeilen in kürzere	144
fx	blättert interaktiv durch JSON-Dateien	153
grep	sucht Texte innerhalb der Datei	171
head	gibt die ersten Zeilen der Datei aus	182
iconv	ändert den Zeichensatz von Textdateien	189
jq	extrahiert Daten aus JSON-Dokumenten	210
less	zeigt Textdateien seitenweise an (auch rückwärts)	217
more	zeigt Textdateien seitenweise an	253
multitail	verfolgt die Änderungen mehrerer Dateien	269
paste	vereint mehrere Texte zeilenweise	315
patch	ändert Textdateien gemäß einer diff-Datei	315
recode	konvertiert zwischen verschiedenen Zeichensätzen	350
sed	Stream-Editor (programmierbarer Editor)	371
sort	sortiert Dateien	391
split	zerlegt eine Datei in Teildateien mit vorgegebener Größe	393
strings	extrahiert Zeichenketten aus Binärdateien	401
tac	gibt Text in umgekehrter Reihenfolge aus, also die letzte Zeile zuerst	408
tail	gibt das Ende einer Datei aus	408
tr	ersetzt vorgegebene Zeichen durch andere Zeichen	423
unexpand	ersetzt Leerzeichen durch Tabulatorzeichen	432
uniq	eliminiert mehrfach auftretende Zeilen in einer Textdatei	432
zcat	gibt eine komprimierte Textdatei aus	471
zless	zeigt eine komprimierte Textdatei an (auch rückwärts)	471
zmore	zeigt eine komprimierte Textdatei seitenweise an	471

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

attr	verwaltet die Zusatzattribute einer Datei	48
chacl	verwaltet die ACLs einer Datei	68
getcap	ermittelt die Capabilities einer Datei	153
getfacl	ermittelt die ACLs einer Datei	154
getfattr	ermittelt die Zusatzattribute einer Datei	155
setcap	verändert die Capabilities einer Datei	374
setfacl	verändert die ACLs einer Datei	375
setfattr	verändert die Zusatzattribute einer Datei	378

Konverter

convert	konvertiert Grafikdateien	80
dvips	wandelt eine DVI-Datei in das PostScript-Format um	117
enscript	konvertiert Textdateien in das PostScript-Format	121
epstopdf	konvertiert EPS-Dateien in PDF-Dateien	122
exiftool	liest bzw. ändert Exif-Daten in JPG-Dateien	125
ffmpeg	konvertiert Videodateien	130
iconv	ändert den Zeichensatz von Textdateien	189
lame	erzeugt MP3-kompatible Audiodateien	215
magick	konvertiert Grafikdateien	239
mogrify	verändert Parameter einer Bilddatei	252
pandoc	erzeugt Dokumente aus Markdown-Dateien	305
paps	konvertiert UTF-8-Textdateien in das PostScript-Format	306
pdf2ps	konvertiert PDF-Dateien in PostScript-Dateien	316
pdfimages	extrahiert Bilder aus PDF-Dateien	317
pdftk	manipuliert PDF-Dateien	318
pdftops	Alternative zu pdf2ps	320
pdftotext	wandelt ein PDF-Dokument in eine reine Textdatei um ...	320
pdfunite	fügt mehrere PDF-Dateien zu einem neuen Dokument zusammen	321
ps2pdf	konvertiert PostScript-Dateien in PDF-Dateien	332
recode	ändert den Zeichensatz von Textdateien	350

Prozesse verwalten

at	führt einen Job zu einem vordefinierten Zeitpunkt aus	46
atq	listet Jobs auf, die später ausgeführt werden sollen	48
atrm	löscht einen Job, der später ausgeführt werden soll	48
batch	führt einen Job aus, sobald das System im Leerlauf läuft	55
bg	setzt einen Prozess im Hintergrund fort	56
chroot	startet eine Shell in einem veränderten Wurzelverzeichnis	76
crontab	hilft bei der Administration eigener Crontab-Einträge	85
disown	löst einen Prozess von der Shell	103
fg	setzt einen Prozess im Vordergrund fort	131
fuser	ermittelt das Programm, das auf eine Datei zugreift	150
glances	moderne Alternative zu top	164
halt	beendet Linux und hält den Rechner an	180

history	zeigt die zuletzt in der bash ausgeführten Kommandos an	182
htop	zeigt alle fünf Sekunden eine Liste aller Prozesse an	184
iftop	beobachtet die Netzwerkaktivität	190
ionice	steuert die I/O-Priorität eines Prozesses	195
iotop	zeigt die Prozesse mit der größten I/O-Aktivität an	196
kill	versendet Signale (meist zum Beenden von Prozessen)	213
killall	wie kill; der Prozess wird mit Namen genannt	213
ldconfig	aktualisiert die Cache-Datei zur Bibliothekssuche	216
ldd	liefert alle erforderlichen Bibliotheken eines Programms	216
lsof	listet offene Dateien und die zugeordneten Prozesse auf	233
nice	startet ein Programm mit verringriger Priorität	291
nohup	startet einen Prozess, der von der Shell losgelöst ist	297
parallel	startet mehrere gleichartige Jobs parallel	307
pidof	ermittelt die Prozessnummer eines Programms	322
powertop	analysiert den Energieverbrauch der laufenden Prozesse	328
ps	zeigt die Liste der laufenden Prozesse an	330
pstree	wie ps; macht die Abhängigkeiten besser sichtbar	334
reboot	beendet Linux und startet den Rechner neu	350
renice	verändert die Priorität eines laufenden Prozesses	352
sudo	führt ein Programm als root aus	402
timeout	limitiert die Ausführungszeit eines Kommandos	421
top	zeigt alle fünf Sekunden eine Liste aller Prozesse an	422
uptime	zeigt an, wie lange der Rechner schon läuft	436
watch	führt ein Kommando periodisch aus und zeigt die Ausgaben an	453

Benutzer und Gruppen verwalten

addgroup	richtet eine neue Gruppe ein (Debian/Ubuntu)	30
adduser	richtet einen neuen Benutzer ein (Debian/Ubuntu)	30
chage	ändert das Ablaufdatum eines Kontos oder Passworts	68
chgrp	ändert die Gruppenzugehörigkeit einer Datei	72
chown	ändert den Besitzer einer Datei	75
chpasswd	verändert ein Benutzerpasswort ohne Interaktion	75
chsh	verändert die Default-Shell eines Benutzers	77
delgroup	löscht eine Gruppe (Debian/Ubuntu)	96
deluser	löscht einen Benutzer (Debian/Ubuntu)	97
groupadd	richtet eine neue Gruppe ein	176
groupdel	löscht eine Gruppe	176
groupmod	verändert Gruppeneigenschaften	176

groups	zeigt die Gruppen des aktuellen Benutzers an	177
gpasswd	ändert Gruppenpasswörter	165
id	zeigt die aktuelle Benutzer- und Gruppen-ID an	189
last	verrät, wer zuletzt auf diesem Rechner eingeloggt war	215
lastb	listet auf, welche Login-Versuche zuletzt gescheitert sind	216
makepasswd	erzeugt ein neues, zufälliges Passwort	240
mkpasswd	erzeugt ein neues, zufälliges Passwort	251
newgrp	ändert die aktive Gruppe eines Benutzers	285
newusers	richtet mehrere neue Benutzer ein	285
passwd	verändert das Passwort eines Benutzers	314
pwgen	generiert leicht merkbare Passwörter	337
useradd	richtet einen neuen Benutzer ein	436
userdel	löscht einen Benutzer	437
usermod	verändert Benutzereigenschaften	437
vigr	bearbeitet /etc/groups	441
vipw	bearbeitet /etc/passwd	441
visudo	bearbeitet /etc/sudoers	441
who	liefert Informationen über die eingeloggten Benutzer	458

Dateisystem administrieren

badblocks	testet, ob Datenträger defekte Sektoren enthalten	53
blkid	liefert die UUID und den Namen eines Dateisystems	56
btrfs	administriert ein Btrfs-Dateisystem	60
cfdisk	partitioniert eine Festplatte	68
cryptsetup	richtet ein verschlüsseltes Device ein	85
dd	kopiert Datenblöcke zwischen Devices	94
df	zeigt den freien Speicher auf der Festplatte an	98
du	ermittelt den Platzbedarf eines Verzeichnisses	115
dumpe2fs	zeigt interne Informationen über ein ext-Dateisystem an	116
e4defrag	defragmentiert Dateien eines ext4-Dateisystems	118
exfatlabel	ändert den Namen eines exFAT-Dateisystems	125
findmnt	liefert eine Liste aller aktiven Dateisysteme	138
fstrim	meldet der SSD alle freien Datenblöcke	147
kpartx	erzeugt oder löscht Device-Dateien für virtuelle Datenträger	213
lsblk	listet alle Block-Devices auf	230
mdadm	verwaltet RAID-Partitionen	242
mkfifo	erzeugt eine FIFO-Datei (eine benannte Pipe)	245
mkfs	richtet ein Dateisystem ein	245

mknod	erstellt Device-Dateien	250
mkswap	richtet eine Datei oder eine Partition als Swap-Bereich ein	251
mount	bindet ein Dateisystem in den Verzeichnisbaum ein	253
ncdu	interaktive, komfortable du-Variante	280
parted	partitioniert eine Festplatte	309
partprobe	informiert den Kernel über die geänderte Partitionierung	312
partx	liest Partitionen bzw. verändert die Partitionstabelle des Kernels	312
resize2fs	verändert die Größe eines ext-Dateisystems	352
sfdisk	partitioniert eine Festplatte mit MBR-Partitionstabelle	381
sgdisk	partitioniert eine Festplatte mit GUID-Partitionstabelle	382
smartctl	steuert die SMART-Funktionen der Festplatte	386
snapper	verwaltet Btrfs-Snapshots (SUSE)	390
swapon	deaktiviert eine Swap-Datei oder -Partition	404
swapon	aktiviert eine Swap-Datei oder -Partition	404
sync	führt alle gepufferten Schreiboperationen aus	404
tune2fs	verändert Systemparameter eines ext-Dateisystems	426
umount	löst ein Dateisystem aus dem Verzeichnisbaum	430
xfs_admin	verändert Parameter eines XFS-Dateisystems	464
xfs_growfs	vergrößert ein XFS-Dateisystem	465
xfs_info	zeigt die Eckdaten eines XFS-Dateisystems	465
xfs_repair	repariert ein defektes XFS-Dateisystem	465
zramctl	steuert den ZRAM-Auslagerungsspeicher	472

Logical Volume Manager (LVM)

lvcreate	richtet ein neues LV (Logical Volume) ein	235
lvdisplay	liefert Detailinformationen zu einem LV	236
lvextend	vergrößert ein LV	236
lvm	LVM-Basiskommando	237
lvreduce	verkleinert ein LV	237
lvremove	löscht ein LV	237
lvrename	gibt dem LV einen neuen Namen	238
lvscan	listet alle LVs auf	238
pvcreate	kennzeichnet eine Partition als PV (Physical Volume)	335
pvdisplay	liefert Detailinformationen zu einem PV	335
pvremove	entfernt die PV-Kennzeichnung eines ungenutzten PVs	336
pvscan	listet alle PVs auf	336
vgchange	ändert die Attribute einer VG (Volume Group)	439
vgcreate	erzeugt eine neue VG aus einem oder mehreren PVs	439

vgdisplay	liefert Detailinformationen zu einer VG	440
vgextend	vergrößert eine VG um ein PV	440
vgmerge	vereint zwei VGs	440
vgreduce	verkleinert eine VG um ein ungenutztes PV	440
vgrename	gibt einer VG einen neuen Namen	440
vgscan	listet alle VGs auf	441

SELinux und AppArmor

aa-complain	protokolliert AppArmor-Regelverstöße, ohne sie zu ahnden	25
aa-disable	deaktiviert ein AppArmor-Regelprofil	25
aa-enforce	stellt die Einhaltung von AppArmor-Regeln sicher	25
aa-status	ermittelt den Zustand des AppArmor-Systems	26
chcon	verändert den SELinux-Kontext von Dateien	70
getenforce	ermittelt den SELinux-Modus (Enforcing oder Permissive)	154
restorecon	stellt den Default-SELinux-Kontext wieder her	354
sealert	hilft bei der Analyse von SELinux-Regelverstößen	370
sestatus	ermittelt den Zustand des SELinux-Systems	373
setenforce	ändert den SELinux-Modus zwischen Enforcing und Permissive	375
setsebool	verändert boolesche Parameter der SELinux-Regeln	379

Paketverwaltung

apk	verwaltet Pakete in Alpine Linux	36
apt	hilft bei der DEB-Paketverwaltung (Debian, Ubuntu)	37
add-apt-repository	richtet eine PPA-Paketquelle ein (Ubuntu)	30
alien	wandelt Pakete zwischen verschiedenen Formaten um ..	32
alternatives	richtet Links in /etc/alternatives ein (Fedora, Red Hat)	34
apt-cache	liefert Informationen über installierte/verfügbare Pakete	39
apt-get	hilft bei der DEB-Paketverwaltung (Debian, Ubuntu)	41
apt-file	sucht nach Dateien auch in nicht installierten Paketen (Debian, Ubuntu)	40
apt-key	richtet einen Schlüssel für eine APT-Paketquelle ein	42
aptitude	hilft bei der DEB-Paketverwaltung	43
cnf	verrät, welches Paket ein Kommando enthält (SUSE)	78
dnf	Alternative zu yum (Fedora)	104

dpkg	(de)installiert oder aktualisiert DEB-Pakete	112
flatpak	verwaltet Flatpak-Pakete	142
pacman	verwaltet Pakete von Arch Linux	302
pip	verwaltet Python-Pakete	324
pkcon	verwaltet Pakete distributionsübergreifend (PackageKit)	326
ppa-purge	deaktiviert eine PPA-Paketquelle (Ubuntu)	329
pro	steuert Ubuntu-Pro-Funktionen	329
rpm	(de)installiert oder aktualisiert RPM-Pakete	362
rpm2archive	wandelt ein Paket in ein TAR-Archiv um	364
rpm2cpio	wandelt ein Paket in ein CPIO-Archiv um	364
snap	verwaltet Snap-Pakete (Ubuntu)	389
tasksel	(de)installiert DEB-Paketgruppen	413
ubuntu-security-status	gibt den Support-Zeitraum der installierten Pakete an	427
update-alternatives	richtet Links in /etc/alternatives ein	434
yay	verwaltet Arch-Linux-Pakete inklusive AUR	470
yum	hilft bei der RPM-Paketverwaltung (Fedora, Red Hat)	470
zypper	hilft bei der RPM-Paketverwaltung (SUSE)	473

Netzwerk administrieren

cadaver	überträgt interaktiv Dateien via WebDAV	64
curl	überträgt Dateien von/zu HTTP-, FTP- und SSH-Servern ...	88
dhclient	führt eine DHCP-Netzwerkkonfiguration durch	99
dig	führt DNS-Abfragen durch	101
etherwake	aktiviert ein Wake-on-LAN-Gerät	124
ethtool	verändert die Parameter eines Ethernet-Adapters	124
exportfs	meldet die NFS-Konfiguration an den NFS-Server	127
firewall-cmd	liest bzw. ändert die Firewall-Konfiguration (RHEL/Fedora)	139
ftp	überträgt interaktiv Dateien via FTP	148
host	löst IP-Adressen bzw. Netzwerknamen auf	183
hostname	liefert bzw. verändert den Namen des lokalen Rechners	183
hostnamectl	verändert den Hostnamen bleibend	184
ifconfig	konfiguriert Netzwerkschnittstellen	190
ifdown	deaktiviert eine Netzwerkschnittstelle	190
ifup	aktiviert eine Netzwerkschnittstelle	190
ip	zeigt Netzwerkeinstellungen an bzw. verändert sie	196
ipcalc	berechnet Netzwerkbereiche und -masken	199
iptables	konfiguriert eine Netfilter-Firewall	199
iptables-restore	liest mehrere Firewall-Regeln ein	203

iptables-save	gibt alle Firewall-Regeln im Textformat aus	203
iptables-xml	gibt alle Firewall-Regeln als XML-Dokument aus	203
iw	steuert WLAN-Schnittstellen	204
mtr	kombiniert ping- und traceroute-Ergebnisse	268
nft	konfiguriert eine nftables-Firewall	286
netplan	steuert andere Netzwerk-Backends (Ubuntu)	281
netstat	analysiert die Netzwerkaktivität des lokalen Rechners	282
networkctl	liefert den Netzwerkstatus (systemd)	283
newaliases	meldet Änderungen in /etc/aliases an den Mail-Server	284
nmap	analysiert die Netzwerkaktivität eines fremden Rechners	292
nmblookup	ermittelt die NetBIOS-Namen von SMB-Servern	295
nmcli	steuert den Network Manager	295
openssl	erzeugt und administriert SSL-Schlüsseldateien	298
ping	überprüft die Netzwerkverbindung zu einem Rechner	324
pnuke	beendet ein Programm parallel auf mehreren Hosts	334
postconf	liest bzw. verändert die Postfix-Konfiguration	327
postqueue	zeigt die Postfix-Warteschlange	327
pscp	kopiert Dateien parallel von/auf mehrere(n) Hosts	334
pssh	führt via SSH Kommandos auf mehreren Hosts aus	334
rdiff-backup	erstellt inkrementelle Backups	348
resolvectl	ermittelt bzw. verändert die DNS-Konfiguration	353
rfkill	(de)aktiviert Bluetooth-, WLAN- und Mobilfunk-Adapter	355
route	verändert bzw. zeigt die IP-Routing-Tabelle	360
rpcinfo	liefert Informationen über RPC- und NFS-Dienste	361
rsync	synchronisiert Netzwerkverzeichnisse	365
scp	überträgt Dateien verschlüsselt via SSH	367
sftp	überträgt Dateien via SFTP	382
showmount	listet NFS-Verzeichnisse auf	384
smbclient	überträgt Dateien aus Windows-Netzwerkverzeichnissen	387
smbpasswd	stellt das Passwort eines Samba-Accounts ein	388
smbstatus	liefert Status-Informationen des lokalen Samba-Servers	389
ss	analysiert die Netzwerkaktivität des lokalen Rechners	395
ssh	ermöglicht Logins auf anderen Rechnern im Netzwerk	396
ssh-keygen	erzeugt und verwaltet SSH-Schlüssel	398
ssh-copy-id	überträgt einen öffentlichen Schlüssel zum SSH-Server	398
swaks	hilft beim Testen und Debuggen von Mail-Servern	404
telnet	kommuniziert interaktiv mit einem Netzwerkdienst	416
traceroute	liefert die Zwischenstationen zu einer Netzwerkadresse	423
ufw	konfiguriert die Firewall (Ubuntu)	427
wakeonlan	aktiviert ein Wake-on-LAN-Gerät (Debian, Ubuntu)	453
wget	lädt Dateien oder Verzeichnisse herunter	454

whois	führt DNS-Abfragen durch	459
wol	aktiviert ein Wake-on-LAN-Gerät (Fedora, Red Hat)	460
wpa_passphrase	hilft bei der WLAN-Konfiguration	460

Hacking und Security

arp	verwaltet den ARP-Cache	45
arp-scan	sendet ARP-Pakete an alle Adressen eines Netzwerks	45
chkrootkit	sucht nach bekannten Rootkits	72
fail2ban-client	administriert Fail2ban	128
hydra	Online-Password-Cracker	186
john	Offline-Password-Cracker	206
nc	Netcat, leitet Netzwerkdaten auf die Standardeingabe bzw. -ausgabe um	278
ngrep	filtert Netzwerk-Streams mit grep (Packet Sniffing)	290
nmap	Netzwerk- und Port-Scanner	292
rkhunter	sucht nach bekannten Rootkits	356
tcpdump	filtert Netzwerk-Streams (Packet Sniffing)	413

Drucker-, Datenbank- und Server-Administration

acme.sh	verwaltet Zertifikate von Let's Encrypt	27
htpasswd	speichert Apache-Login-Daten in einer Passwortdatei	185
lpadmin	richtet neue Drucker ein bzw. löscht sie wieder	223
lpinfo	listet alle Druck-Devices, Druckertreiber etc. auf	224
lpoptions	zeigt die Optionen von Druckern an bzw. verändert sie	224
lpq	zeigt den Inhalt einer Druckerwarteschlange an	225
lpr	druckt eine Datei aus	225
lprm	löscht einen Druckjob aus der Warteschlange	225
lpstat	liefert Informationen über Drucker, Druckjobs etc.	226
mysql	führt SQL-Kommandos auf einem MySQL-Server aus	270
mysqladmin	hilft bei der MySQL-Administration	273
mysqlbinlog	extrahiert Daten aus binären MySQL-Logging-Dateien	274
mysqldump	führt ein MySQL-Backup durch	275
pg_dump	erstellt ein Backup einer PostgreSQL-Datenbank	321
psql	führt SQL-Kommandos auf einem PostgreSQL-Server aus	333
smbpasswd	ändert ein Samba-Passwort	388
sqlite3	führt SQL-Kommandos in SQLite-Datenbanken aus	394

Audiofunktionen und Hardware-Verwaltung

acpi	liefert Informationen über den Batteriezustand	29
alsactl	speichert bzw. lädt alle Parameter des Audiosystems	32
alsamixer	stellt die Audiokanäle interaktiv ein	33
amixer	steuert die Audiokanäle durch Optionen	34
boltctl	steuert Geräte an der Thunderbolt-Schnittstelle	59
free	zeigt den freien Speicherplatz an (RAM/Swap)	145
fwupdmgmt	verwaltet Firmware-Updates	151
hwclock	liest bzw. stellt die Hardware-Uhr	186
inxi	gibt einen Überblick über die Hardware	194
kbdrate	stellt die Tastenwiederholungsrate ein	211
localectl	verändert die Sprach- und Tastatureinstellungen	220
lscpu	liefert Informationen über die CPU	232
lshw	bildet eine hierarchische Liste der Hardware-Komponenten	232
lspci	liefert Informationen über PCI-Komponenten	234
lsscsi	liefert Informationen über angeschlossene SCSI-Geräte	234
lsusb	liefert Informationen über angeschlossene USB-Geräte	235
nproc	liefert die Anzahl der CPU-Cores	298
pactl	steuert den PulseAudio-Server	304
paplay	spielt eine Raw-Datei via PulseAudio ab	306
parecord	führt eine Raw-Audio-Aufnahme via PulseAudio durch	306
powertop	hilft, den Energieverbrauch von Notebooks zu optimieren	328
pw-cat	überträgt Audiodateien von oder zum PipeWire-System	336
pw-cli	steuert das PipeWire-System	336
pw-mon	listet alle PipeWire-Objekte auf	336
pw-top	zeigt die aktiven PipeWire-Audio-Komponenten an	336
speaker-test	testet das Audiosystem und die angeschlossenen Lautsprecher	393
timedatectl	stellt Datum, Uhrzeit und Zeitzone ein	420
vcgencmd	liest bzw. verändert Hardware-Parameter des Raspberry Pi	438

Bluetooth

bluetoothctl	konfiguriert Bluetooth-Geräte	57
hciconfig	(de)aktiviert lokale Bluetooth-Adapter	180
hcitool	verwaltet Bluetooth-Geräte	181

l2ping	sendet Echo-Anfragen an Bluetooth-Geräte	214
rfkill	(de)aktiviert Bluetooth-, WLAN- und Mobilfunk-Adapter	355
sdptool	ermittelt Detailinformationen über Bluetooth-Geräte	370

Kernel

canonical-livepatch	administriert Kernel-Livepatches (Ubuntu)	65
depmod	erzeugt eine Datei mit allen Modulabhängigkeiten	97
dmesg	zeigt die Meldungen des Kernels an	104
dracut	erzeugt eine neue Initrd-Datei (Fedora, RHEL, SUSE)	114
insmod	lädt ein Kernelmodul (Low Level)	193
kexec	aktiviert einen anderen Kernel	211
lsmod	listet alle geladenen Kernelmodule auf	233
modinfo	liefert Informationen zu einem Kernelmodul	252
modprobe	lädt ein Kernelmodul samt Abhängigkeiten	252
uname	zeigt die aktuelle Kernelversion an	431
update-initramfs	erzeugt eine neue Initrd-Datei (Debian, Ubuntu)	435

Systemstart und -stopp, Init-System, Logging und GRUB

efibootmgr	liest bzw. verändert die Tabelle der EFI-Booteinträge	119
grub-install	installiert GRUB in den Bootsektor	177
grub-mkconfig	erzeugt eine neue GRUB-Konfigurationsdatei	178
journalctl	liest Nachrichten aus dem Journal	208
logger	protokolliert eine Nachricht	221
logind	steuert den systemd-Login-Manager	222
needs-restarting	verrät, ob das System oder einzelne Dienste neu gestartet werden müssen	281
shutdown	beendet Linux	384
systemctl	verwaltet systemd-Prozesse	405
systemd-analyze	hilft bei der systemd-Fehlersuche	408
update-grub	aktualisiert die GRUB-Konfiguration (Debian, Ubuntu)	435

Virtualisierung, Container, Cloud

aws	steuert Amazon-Cloud-Dienste	50
docker	verwaltet Container	109

kvm	führt eine virtuelle Maschine aus	339
podman	verwaltet Container	109
qemu	führt eine virtuelle Maschine aus	339
qemu-img	erzeugt bzw. bearbeitet Image-Dateien	344
rclone	synchronisiert ein lokales Verzeichnis mit einem Cloud-Verzeichnis	345
virsh	führt libvirt-Kommandos aus	441
virt-clone	erstellt eine Kopie einer virtuellen Maschine	447
virt-install	richtet eine neue virtuelle Maschine ein	449
virt-viewer	erlaubt die Bedienung einer virtuellen Maschine via VNC	452
wsl	verwaltet Linux-Installationen unter Windows	461

Terminal und Textkonsole

echo	gibt eine Zeile Text aus	118
loadkeys	lädt eine Tastaturtabelle für Textkonsole	219
printf	ermöglicht eine formatierte Ausgabe wie unter C	329
reset	führt einen Reset für das Terminal durch	352
screen	verwaltet mehrere Sessions in einem Terminal	368
setterm	verändert diverse Terminal-einstellungen	379

Onlinehilfe

apropos	sucht Kommandos zu einem Thema	37
help	zeigt die Beschreibung eines Shell-Kommandos an	182
info	startet das info-System	191
man	zeigt die Beschreibung eines Kommandos an	241
tldr	zeigt Beispiele zur Verwendung eines Kommandos an	421
whatis	zeigt eine Kurzbeschreibung eines Kommandos an	457

Grafiksystem und Gnome

chvt	wechselt die aktive Textkonsole bzw. aktiviert das Grafiksystem	77
dconf	verändert die dconf-Datenbank (Low Level)	93
fc-list	listet alle skalierbaren Schriften auf	129

fgconsole	liefert die Nummer der aktiven Konsole	132
gnome-session-quit	initiiert einen Logout bzw. Shutdown	165
grim	erstellt einen Screenshot unter Wayland	175
gsettings	liest bzw. ändert Einstellungen der dconf-Datenbank	178
nvidia-xconfig	hilft bei der Konfiguration des NVIDIA-Grafiktreibers	298
open	öffnet eine Datei mit einem geeigneten Programm	464
scrot	erstellt einen Screenshot unter X	369
slurp	ermöglicht die Auswahl eines Rechtecks unter Wayland	385
wl-copy	fügt Daten in die Wayland-Zwischenablage ein	459
wl-paste	liest Daten aus der Wayland-Zwischenablage	459
wlr-randr	ändert die Auflösung des Grafiksystems (Wayland)	460
xdg-open	öffnet eine Datei mit einem geeigneten Programm	464
xdpyinfo	liefert Informationen zum laufenden X-Server	464
xhost	erlaubt oder sperrt Hosts für den X-Login	466
xinput	konfiguriert Eingabegeräte für X	467
xkill	beendet ein Programm per Mausklick	467
xrandr	ändert die Auflösung des Grafiksystems (X)	467
xset	ändert Benutzereinstellungen des Grafiksystems (X)	469
zenity	zeigt einfache grafische Dialoge an	471

Sonstiges

alias	definiert eine Abkürzung	31
basename	ermittelt den Dateinamen eines Pfads	54
bc	führt einfache Berechnungen durch (wie ein Taschenrechner)	55
cksum	berechnet die CRC-Prüfsumme zu einer Datei	78
date	zeigt Datum und Uhrzeit an	91
dirname	ermittelt das Verzeichnis eines Pfads	103
expr	führt Berechnungen und Mustervergleiche durch	128
git	steuert das Versionsverwaltungssystem Git	157
gpioget	liest den Zustand der Input/Output-Pins des Raspberry Pi	170
gpioset	verändert den Zustand der Input/Output-Pins des Raspberry Pi	170
hash	zeigt die Hash-Tabelle an	180
ldd	zeigt die erforderlichen Libraries für ein Programm an	216
lsb_release	stellt den Namen und die Version der Distribution fest	231
mail	übergibt eine E-Mail an den lokalen Mail-Server	239
md5sum	berechnet eine Prüfsumme zu einer Datei	241
printenv	zeigt nur die Umgebungsvariablen an	329

qalc	ist ein Taschenrechner für das Terminal	338
pinctrl	steuert die Input/Output-Pins des Raspberry Pi	322
rpicam-still	nimmt ein Foto auf (Raspberry Pi OS)	358
rpicam-vid	nimmt ein Video auf (Raspberry Pi OS)	359
seq	liefert eine Zahlensequenz	373
set	zeigt alle der Shell bekannten Variablen an	374
sha512sum	berechnet eine Prüfsumme zu einer Datei	383
sleep	wartet eine vorgegebene Zeit	385
strace	verrät, welche Funktionen ein Programm aufruft	401
time	misst die Ausführzeit eines Kommandos	419
tty	zeigt den Device-Namen des Terminals an	426
type	gibt den Typ eines Kommandos an	427
unalias	löscht eine Abkürzung	431
uname	liefert den Betriebssystemnamen und die Kernelversion	431
xargs	leitet die Standardeingabe an ein Kommando weiter	463

bash-Programmierung

break	beendet eine Schleife vorzeitig	60
case	leitet eine Fallunterscheidung ein	66
continue	überspringt den Schleifenkörper	79
exit	beendet das Shell-Programm	126
for	leitet eine Schleife ein	145
function	definiert eine neue Funktion	149
getopts	verarbeitet an ein Script übergebene Optionen	155
if	leitet eine Verzweigung ein	189
local	definiert lokale Variablen in einer Funktion	219
source	führt die angegebene Shell-Datei aus	392
test	wertet eine Bedingung aus	417
until	leitet eine Schleife ein (Variante 1)	433
while	leitet eine Schleife ein (Variante 2)	458

bash-Variablenverwaltung

alias	definiert eine Abkürzung	31
declare	definiert eine (Umgebungs-)Variable	96
export	definiert eine Umgebungsvariable	127
local	definiert lokale Variablen in einer Funktion	219

read	liest eine Variable ein	349
readonly	zeigt alle schreibgeschützten Variablen an	350
shift	verschiebt die Parameterliste	383
unalias	löscht eine Abkürzung	431
unset	löscht eine Variable	432

Weitere bash-Kommandos und -Sonderzeichen

dirs	zeigt die Liste der gespeicherten Verzeichnisse an	103
disown	löst einen Prozess von der Shell	103
eval	wertet das angegebene Kommando aus	125
popd	wechselt in das letzte gespeicherte Verzeichnis	326
pushd	speichert das Verzeichnis und wechselt in ein anderes	335
trap	führt beim Eintreten eines Signals ein Kommando aus	424
ulimit	kontrolliert die von der Shell beanspruchten Ressourcen	429
wait	wartet auf das Ende eines Hintergrundprozesses	452
#%!	bash-Sonderzeichen	476

Konfigurationsdateien

adduser.conf	Einstellungen für neue Accounts (Debian, Ubuntu)	479
aliases	E-Mail-Weiterleitungen	481
bashrc	Defaulteinstellungen für die bash	482
config.txt	Raspberry-Pi-Hardware-Parameter (Raspberry Pi OS)	482
crontab	Prozesse periodisch ausführen	484
deluser.conf	Einstellungen für deluser und delgroup (Debian, Ubuntu)	487
dnf.conf	Konfiguration der Paketverwaltung dnf (Fedora)	488
fstab	Dateisysteme/Partitionen automatisch einbinden	490
group	Gruppennamen und Gruppenzuordnungen	491
grub	Defaulteinstellungen für GRUB 2	492
grub.cfg	Konfiguration für GRUB 2	495
gshadow	Hash-Codes der Gruppenpasswörter	495
host.conf	Konfiguration der Resolver-Bibliothek	496
hostname	Hostname des Rechners	496
hosts	statische Liste von Hostnamen und IP-Adressen	497
interfaces	Netzwerkkonfiguration (Debian, Ubuntu)	497
journald.conf	Konfiguration des Logging-Diensts Journal	501
locale.conf	Lokalisierungseinstellungen (systemd)	502
login.defs	Optionen für das Anlegen neuer Benutzer und Gruppen	503

mdadm.conf	Software-RAID-Konfiguration	506
modules	Kernelmodule automatisch laden (Debian, Ubuntu)	507
netplan.yaml	Netzwerkeinstellungen (Ubuntu)	507
networkd.network	networkd-Konfiguration (systemd)	508
nsswitch.conf	Konfiguration der Name-Service-Switch-Funktionen	512
os-release	Namen und Versionsnummer der Distribution (systemd)	513
passwd	Liste aller Benutzer und Heimatverzeichnisse	514
profile	Konfiguration systemweiter Umgebungsvariablen	515
rc.local	Script, wird am Ende des Bootprozesses ausgeführt	515
resolv.conf	IP-Adresse des Nameservers	516
rsyslog.conf	Konfiguration des Syslog-Dienstes	517
services	Zuordnung zwischen Netzwerkdiensten und Ports	520
shadow	Hash-Codes der Login-Passwörter	520
sources.list	APT-Paketquellen (Debian, Ubuntu)	522
sudoers	Konfiguration für sudo	523
sysctl.conf	Defaulteinstellungen für Kernelparameter	525
systemd.service	Konfiguration von systemd-Diensten	525
systemd.timer	Konfiguration von periodischen systemd-Jobs	529
vconsole.conf	Tastatureinstellungen (systemd)	531
wpa_supplicant.conf	WLAN-Konfiguration (Raspberry Pi OS)	532

Tastenkürzel

bash	Shell	533
emacs	Editor	533
gnome-terminal	Terminalfenster unter Gnome	536
grub	Bootloader	537
info	Kommando zur Anzeige von Hilfetexten	537
joe	einfacher Editor	538
konsole	Terminalfenster unter KDE	539
less	Kommando zur Anzeige von Textdateien	539
man	Kommando zur Anzeige von Hilfetexten	540
mutt	E-Mail-Client für den Textmodus	540
nano	minimalistischer Editor	541
screen	Terminal-Multiplexer	541
---	Tastenkürzel in Textkonsolen	542
vi/vim	Editor	542

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 dieses Buch mittlerweile die 500-Seiten-Grenze sprengt, 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 Sie für eine bestimmte Aufgabe ein Kommando suchen, seinen Namen aber nicht kennen oder gerade vergessen haben. 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) oder die dazu weitgehend kompatible `zsh`

verwenden. Diese Shells kommen bei den meisten Linux-Distributionen für das interaktive Ausführen von Kommandos zum Einsatz. Ein Beispiel für ein Shell-Kommando ist `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 also 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` anstelle von `-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 dies 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 sich 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`.

Die Hilfetexte von `man` und `info` schießen manchmal über das Ziel hinaus. Wenn Sie lediglich auf der Suche nach ein paar Anwendungsbeispielen für ein Kommando sind, verwenden Sie entweder dieses Buch oder das Kommando `tldr`. Der Kommandoname steht übrigens für: *Too long, didn't read. :-)*

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
```

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 mit telnet interaktiv Netzwerkprotokolle wie HTTP oder SMTP ausprobieren. Mit nc können Sie aber auch Dateien über einen beliebigen Netzwerk-Port 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 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 Verbindungsaufbau (*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 proxyaddr:portnr**
verwendet die angegebene Proxy-Adresse und den dazugehörenden 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 (linke Spalte). Damit überwacht nc den angegebenen Port 1234 und wartet auf einen Verbindungsaufbau.

Auf dem zweiten Rechner wird nc ohne Optionen gestartet, um die Verbindung zum ersten Host herzustellen. Eine sichtbare Bestätigung des Verbindungsaufbaus 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
host2$ nc host1 1234
wie geht's?
wie geht's?
gut
gut
<Strg>+<C>
```

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. Einen Überblick über die wichtigsten Kürzel gibt . Vorsicht, 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 [#####] /Nextcloud
1,0 GiB [#####] /Downloads
528,5 MiB [##] /Bilder
209,4 MiB [#] /Dokumente
...
```

needs-restarting [optionen]

Das unter 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 neu gestartet werden sollte:

```
root# needs-restarting -r
Core libraries or services have been updated since boot-up:
  * glibc
Reboot is required to fully utilize these updates.
More information: https://access.redhat.com/solutions/27943
```

```
root# needs-restarting | sort -n
782 : /usr/lib/polkit-1/polkitd --no-debug
783 : /usr/libexec/power-profiles-daemon
784 : /usr/bin/qemu-ga --method=virtio-serial ...
787 : /usr/libexec/accounts-daemon
...
```

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 `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/slacd
tcp   localhost:mysql          *:*          LISTEN  mysql    5785/mysqld
tcp6  [::]:ssh                [::]:*        LISTEN  root      5559/sshd
udp   *:nfs                  *:*          -        root      -
```

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/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
    State: routable
  Online state: online
    Address: 168.119.33.110 on eth0
              172.17.0.1 on docker0
              2a01:4f8:242:1f88::4 on eth0
              fe80::5054:ff:fe4a:7321 on eth0
  Gateway: 168.119.33.119 on eth0
              2a01:4f8:242:1f88::2 on eth0
    DNS: 213.133.100.100
          213.133.99.99
              2a01:4f8:0:1::add:1010
              2a01:4f8:0:1::add:9999
```

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 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/title/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 als 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 I/O-Belastung eines Kommandos reduzieren möchten, setzen Sie besser ionice ein.

Beispiel

Das folgende Kommando startet das Script `mybackup.sh` mit niedrigerer Priorität:

```
user$ nice -n 10 mybackup.sh
```

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`
scanns 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 auf 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)
```

Die Grundausstattung für Ihr Linux-Bücherregal

Ob Sie erste Schritte auf dem Terminal wagen oder das volle Potenzial von Linux nutzen möchten: Mit diesem Buch haben Sie stets alle wichtigen Kommandos samt Optionen, Erläuterungen und geprüften Beispielen parat. Mit der Tastenkürzelreferenz und den kommentierten Konfigurationsdateien nutzen Sie die Shell sicher und effizient.

»Nützlicher
Helper, der rund-
um gut gefällt!«

LINUX
MAGAZIN

btrfs xkill pinctrl
journalctl dnf
docker chroot eval
aws mount
xfs_admin
mkdir rsync
systemctl logout tac
hydra wget openssl
getfattr vgcreate

Über 500 Kommandos

f39 (Fedora 64bit / Linux 6.7.9-200.fc39.aarch64)						
	CPU		idle		ctx_sw	
	CPU	user	system	idle	irq	inter
8	[]	5.0%	0.0%	1.7%	1.7%	inter
CPU	[]	5.0%	0.0%	1.7%	1.7%	inter
MEM	[43.2%	0.0%	0.0%	0.0%	sw_int
SWAP	[]	10.0%	0.0%	0.0%	0.0%	steal

	Rx/s	Tx/s	TASKS			
	4KB	0B	CPU%	MEM%	VIRT	RES
enp0s1	376B	376B	11.1	7.5	6.07G	217M
lo	0	0	1.9	1.3	486M	38.0M
DISK I/O	R/s	W/s	1.9	1.3	766M	44.2M
sr0	0	0	0.0	0.0	2141	108221
vda	0	0	0.0	0.0	2821	2821
vda1	0	0	0.0	1.3	3302	3092
vda2	0	0	0.0	1.0	37.9M	37.9M
vda3	0	0	0.0	0.0	1	1
zram0	343K	0	0.0	0.0	2812	106205
			0.0	0.7	1.73G	21.2M
FILE SYS	Used	Total	0.0	0.7	1012M	21.1M
/ (vda1)	8.00G	23.4G	0.0	0.7	708M	19.2M
/home (vda3)	8.00G	23.4G	0.0	0.6	851M	18.4M

Befehle richtig einsetzen



Einfach effizienter arbeiten

Befehle schnell und einfach finden

Die Referenz ermöglicht eine rasche alphabetische oder thematische Suche nach Kommandos – distributionsspezifische Besonderheiten inklusive. Bekannte Befehle schlagen Sie schnell nach, unbekannte lernen Sie im thematischen Umfeld.

Kommandos verstehen und anwenden

Viel mehr als eine bloße Funktionsbeschreibung: An zahlreichen Anwendungsbeispielen lernen Sie den Einsatz der Kommandos und Optionen kennen. So administrieren Sie Server sicher über SSH und automatisieren Aufgaben mit Scripts.

Von Praxishilfen profitieren

Was bedeuten die Spalten der /etc/fstab? Und wie beendet man eigentlich den vim? Antworten auf Fragen zur Syntax und Bedeutung zentraler Konfigurationsdateien sowie zu häufig gebrauchten Tastenkürzeln finden Sie hier!



Michael Kofler ist der erfolgreichste deutschsprachige Computerbuchautor. Zu seinen Themen gehören neben Linux auch die IT-Sicherheit, Docker, Git, Python und Java sowie der Raspberry Pi. Er arbeitet als Entwickler, Berater und Lehrbeauftragter der Informatik.

Aus dem Inhalt

Kommando-Sammlung

Dateien verwalten, suchen und bearbeiten

Programm-/Prozess- und Benutzerverwaltung

Dateisystem- und Netzwerk-Administration

SELinux, AppArmor

Drucker-, Datenbank- und Server-Administration

Bluetooth und Thunderbolt

Systemstart/-stopp

Init-System, GRUB

libvirt, KVM, Container

Wayland und Xorg

bash-Programmierung

Konfigurationsdateien

Tastenkürzel im Überblick

