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.

```
user$  host kofler.info
kofler.info has address 5.9.22.29
kofler.info has IPv6 address 2a01:4f8:161:107::4
kofler.info mail is handled by 10 mail.kofler.info.
user$  host 5.9.22.29
28.22.9.5.in-addr.arpa domain name pointer kofler.info.
user$  host 2a01:4f8:161:107::4
4.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.7.0.1.0.1.6.1.0.8.f.4.0.1.0.a.2.ip6.arpa
  domain name pointer kofler.info.
user$  host -t txt kofler.info
kofler.info descriptive text "v=spf1 ip4:5.9.22.29 ip6:2a01:4f8:161:107::4 -all"
```

hostname [name]

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] passwortdatei 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` *passwort*

  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 Kurzschreibweise 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

```
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
```

`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-Scripts. 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 Q 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 *ip6adr/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%.md}.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 2a01:4f8:161:107::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` ipadress [netmask]

`ipcalc` aus dem gleichnamigen Paket ermittelt aus einer gegebenen IPv4-Adresse und der Netzmaske 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 Kompatibilitätsschicht 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-Adaptern 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 `wlp`*nsm*.

```
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 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:

`channel`	Frequenzkanäle
`frequency`	Frequenzen
`key`	zulässige Verschlüsselungsverfahren und eingestellte Schlüssel
`rate`	unterstützte Bruttoübertragungsraten des WLAN-Controllers
`scan`	Liste der erreichbaren Netze mit ESSID, Qualität, Frequenz etc.

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 vorgebenen 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.zGFBlF$LdNTE...:1001:1001::/home/peter:/bin/bash
maria:$6$gSJg6.d8$mN.en...:1002:1002::/home/maria:/bin/bash
hans:$6$UinuQqJY$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

```
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
```

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 Signalnamen 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-Zeigegerät, 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 Hostrecher 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 Verbindungsaufbau 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 Verbindungsaufbau 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
```

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
Strg+A	bewegt den Cursor an den Zeilenanfang (wie Pos1).
Strg+C	bricht das laufende Kommando ab.
Strg+E	setzt den Cursor an das Ende der Zeile (wie Ende).
Strg+K	löscht den Rest der Zeile ab der Cursorposition.
Strg+Y	fügt den zuletzt gelöschten Text wieder ein.
Strg+Z	unterbricht das laufende Kommando (Fortsetzung mit `fg` oder `bg`).
⇆	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
Strg+X, Strg+F	lädt eine neue Datei.
Strg+X, Strg+S	speichert die aktuelle Datei.
Strg+X, Strg+W	speichert die Datei unter einem neuen Namen.
Strg+G	bricht die Eingabe eines Kommandos ab.
Strg+X, U	macht die letzte Änderung rückgängig (Undo).
Strg+X, Strg+C	beendet den Emacs (mit Rückfrage zum Speichern).

Tabelle 7 Elementare Emacs-Kommandos

Tastenkürzel	Funktion
Alt+F / Alt+B	bewegt den Cursor ein Wort vor bzw. zurück.
Strg+A / Strg+E	stellt den Cursor an den Zeilenbeginn bzw. dessen Ende.
Alt+< / Alt+⇧+>	bewegt den Cursor an den Beginn bzw. das Ende des Textes.
Alt+G *n* ↵	bewegt den Cursor in Zeile *n*.

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
Strg+Leertaste	setzt einen (unsichtbaren) Markierungspunkt.
Strg+W	löscht den Text zwischen dem Markierungspunkt und der aktuellen Cursorposition.
Strg+Y	fügt den gelöschten Text wieder ein.
Strg+X, Strg+X	vertauscht Cursorposition und Markierungspunkt.
Alt+D	löscht das nächste Wort bzw. das Ende des Wortes ab dem Cursor.
Alt+←	löscht das vorige Wort bzw. den Beginn des Wortes bis zum Cursor.
Strg+K	löscht den Rest der Zeile ab der Cursorposition.
Alt+0, Strg+K	löscht den Zeilenanfang vor der Cursorposition.
Alt+M	löscht den nächsten Absatz.
Alt+Z, *x*	löscht alle Zeichen bis zum nächsten Auftreten von x. Das Zeichen x wird mitgelöscht.
Strg+Y	fügt den zuletzt gelöschten Text an der Cursorposition wieder ein.

Tabelle 9 Text markieren, löschen und wieder einfügen

Tastenkürzel	Funktion
Strg+S	inkrementelle Suche vorwärts
Strg+R	inkrementelle Suche rückwärts
Alt+P	wählt einen früher verwendeten Suchtext aus (Previous).
Alt+N	wählt einen später verwendeten Suchtext aus (Next).
Strg+G	Abbruch der Suche

Tabelle 10 Suchen und Ersetzen

Tastenkürzel	Funktion
Strg+X, Strg+X	vertauscht den Markierungspunkt (Beginn der Suche) und die aktuelle Cursorposition.
Strg+Alt+S	inkrementelle Mustersuche vorwärts
Strg+Alt+R	inkrementelle Mustersuche rückwärts
Alt+%	Suchen und Ersetzen ohne Muster
Alt+X query-replace-r ↵	Suchen und Ersetzen mit Muster

Tabelle 10 Suchen und Ersetzen (Forts.)

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
D	Partition löschen (*delete*)
L	Partitions-ID-Nummer anzeigen (*list*)
M	Online-Hilfe (*menu*)
N	neue Partition anlegen (*new*)
P	Partitionsliste anzeigen (*print*)
Q	Programm beenden (ohne die Partitionstabelle zu verändern; *quit*)
T	Partitionstyp verändern
U	Maßeinheit zwischen Zylindern und Sektoren umschalten (*unit*)
V	Partitionstabelle überprüfen (*verify*)
W	Partitionstabelle ändern (*write*)

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.
⇧+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.
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
Leertaste	scrollt Text nach unten.
←	scrollt Text nach oben.
B, E	springt zum Anfang/Ende der Info-Einheit (*beginning/end*).
↹	bewegt den Cursor zum nächsten Querverweis.
↵	verfolgt einen Querverweis zu einer anderen Info-Einheit.
N	zeigt die nächste Info-Einheit in derselben Hierarchiestufe an (*next*).

Tabelle 14 »info«-Tastenkürzel

Tastenkürzel	Funktion
P	zeigt die vorige Info-Einheit in derselben Hierarchiestufe an (*previous*).
U	springt eine Hierarchieebene nach oben (*up*).
L	springt zurück zum zuletzt angezeigten Text (*last*).
H	zeigt eine ausführliche Bedienungsanleitung an (*help*).
?	zeigt eine Kommandoübersicht an.
Q	beendet info (*quit*).

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
⇧+Strg+A	markiert den Reiter bei Aktivität.
⇧+Strg+C	kopiert den markierten Text in die Zwischenablage.
⇧+Strg+F	sucht einen Text in den Terminalausgaben.
F3	wiederholt die Suche rückwärts.
⇧+F3	wiederholt die Suche vorwärts.
⇧+Strg+I	markiert den Reiter bei längerer Inaktivität.
⇧+Strg+N	öffnet ein neues Terminalfenster.
⇧+Strg+Q	schließt das Fenster.
⇧+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+←/→	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.
Pos1, Ende	springt an den Beginn bzw. das Ende des Textes.
G, ⇧+G	springt an den Beginn bzw. das Ende des Textes.
/ *muster* ↵	sucht vorwärts.

Tabelle 17 »less«-Tastenkürzel

Tastenkürzel	Funktion
? *muster* ↵	sucht rückwärts.
N	wiederholt die Suche vorwärts (*next*).
⇧+N	wiederholt die Suche rückwärts.
V	startet den durch `$EDITOR` oder `$VISUAL` eingestellten Editor.
Q	beendet `less` (*quit*).
H	zeigt einen Hilfetext mit weiteren Tastenkürzeln an.

Tabelle 17 »less«-Tastenkürzel (Forts.)

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 17).

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.
Leertaste	blättert durch den Text einer Mail.
D	löscht die E-Mail.
I	wechselt von der E-Mail-Ansicht zurück in die Inbox.
J	zeigt die nächste E-Mail an.

Tabelle 18 »mutt«-Tastenkürzel

Tastenkürzel	Funktion
M	verfasst eine neue E-Mail mit dem durch $EDITOR oder $VISUAL eingestellten Editor.
R	beantwortet die E-Mail.
?	zeigt einen Hilfetext mit allen Tastenkürzeln an.

Tabelle 18 »mutt«-Tastenkürzel (Forts.)

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
Strg+A	bewegt den Cursor an den Beginn der Zeile.
Strg+D	löscht ein Zeichen.
Strg+E	bewegt den Cursor zum Ende der Zeile.
Strg+H	löscht ein Zeichen rückwärts.
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 (*create*).
Strg+A, D	trennt die aktuelle Session von screen (*detach*).
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
Strg+Alt+Fn	wechselt vom Grafikmodus in die Textkonsole *n*.
Alt+F1	wechselt zurück in den Grafikmodus.
Alt+Fn	wechselt von einer Textkonsole in die Textkonsole *n*.
Alt+→/+←	wechselt in die vorige/nächste Textkonsole.
⇧+Bild↑/Bild↓	scrollt seitenweise vorwärts/rückwärts.
Strg+Alt+Entf	beendet Linux durch shutdown (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
I	aktiviert den Einfügemodus.
A	aktiviert den Einfügemodus. Die Texteingabe beginnt beim nächsten Zeichen.
Esc	aktiviert den Standardmodus bzw. bricht die Kommandoeingabe ab.
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+O *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 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	
Entf, ←	Diese Tasten haben die übliche Bedeutung.
Kommandos im Standardmodus	
X	löscht das Zeichen an der Cursorposition bzw. den markierten Text.
⇧+X	löscht das Zeichen vor dem Cursor.
D, D	löscht die aktuelle Zeile.
D *cursorkommando*	löscht den Text entsprechend dem Kommando zur Cursorbewegung (siehe Tabelle 24). Beispiele: D, $ löscht bis zum Ende der Zeile. D, B löscht das vorige Wort. D, W 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.

Tabelle 24 Tastenkürzel zur Cursorbewegung im Standardmodus

Tastenkürzel	Funktion
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 ()[]{}.

Tabelle 24 Tastenkürzel zur Cursorbewegung im Standardmodus (Forts.)

Tastenkürzel	Funktion
V	(de)aktiviert den Zeichenmarkierungsmodus.
⇧+V	(de)aktiviert den Zeilenmarkierungsmodus.
Strg+V	(de)aktiviert den Blockmarkierungsmodus.
A, W	vergrößert die Markierung um ein Wort.
A, S	vergrößert die Markierung um einen Satz.
A, P	vergrößert die Markierung um einen Absatz.

Tabelle 25 Text markieren

Tastenkürzel	Funktion
A, B	vergrößert die Markierung um eine ()-Ebene.
A, ⇧+B	vergrößert die Markierung um eine {}-Ebene.
G, V	markiert den zuletzt markierten Text nochmals.
O	wechselt die Cursorposition zwischen Markierungsanfang und -ende.

Tabelle 25 Text markieren (Forts.)

Tastenkürzel	Funktion
Y	kopiert den markierten Text in das Kopierregister.
Y, Y	kopiert die aktuelle Zeile in das Kopierregister.
Y *cursorkommando*	kopiert den durch die Cursorbewegung erfassten Text; Beispiel: Y, } kopiert den Text bis zum Ende des Absatzes.

Tabelle 26 Text in das Kopierregister kopieren

Tastenkürzel	Funktion
X	löscht den markierten Text.
Y	kopiert den markierten Text in das Kopierregister.
~	ändert die Groß-/Kleinschreibung.
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.
`*`	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 [:] `set ignorecase` aus (gilt für alle weiteren Suchen).

Mit [:] `set incsearch` aktivieren Sie die inkrementelle Suche: Bereits während der Eingabe des Suchtexts 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 *abc* durch *efg*.
[:] `%s/`*abc*`/`*efg*`/gc`	ersetzt mit Rückfrage alle Vorkommen von *abc* durch *efg*.
[:] `%s/`*abc*`/`*efg*`/gci`	ersetzt ohne Berücksichtigung der Groß- und Kleinschreibung.

Tabelle 29 Suchen und ersetzen