Liste der 50 wichtigsten Linux-Befehle

Befehl	Beschreibung
sudo	Befehle mit Rechten anderer Benutzerinnen bzw. Benutzer ausführen
ls	Verzeichnisinhalte auflisten
cd	Navigation im Verzeichnisbaum
touch	Neue Datei erstellen
mkdir	Neues Verzeichnis erstellen
rm	Datei entfernen
rmdir	Verzeichnis entfernen
mv	Datei oder Verzeichnis verschieben oder umbenennen
cp	Datei oder Verzeichnis kopieren
pwd	Aktuelle Position im Verzeichnisbaum ausgeben
zip	Dateien in Zip-Archiven schreiben
unzip	Dateien aus Zip-Archiven extrahieren
ln	Symbolische Verlinkung erstellen
cat	Datei-Inhalte zusammenführen
grep	Textdateien durchsuchen
diff	Unterschiede zwischen Text-Dateien finden
cmp	Unterschiede zwischen Dateien auf Byte-Ebene finden
tar	Dateien in Tar-Archive schreiben und extrahieren
echo	String auf der Standardangabe ausgeben
clear	Terminal leeren
ssh	Über eine *Secure Shell* mit einem anderen Rechner verbinden
wget	Datei direkt aus dem Internet herunterladen
ping	Server anfragen und Latenz messen
ftp, sftp	Dateien via (S)FTP übertragen
ip	Netzwerkschnittstellen abfragen und konfigurieren
apt/pacman/yum	Software-Pakete herunterladen und verwalten
netstat	Status der Netzwerkschnittstelle ausgeben
traceroute	Datenpakete verfolgen
route	IP-Routing-Tabellen anzeigen und bearbeiten
dig	DNS-Informationen abfragen
mount/unmount	Dateisysteme einbinden (aufsetzen/einhängen/mounten)
dd	Bit-genaues Kopieren von Dateien, Partitionen oder Datenträgern
chmod	Zugriffsrechte verwalten
chown	Besitzrechte verwalten
adduser	Nutzerkonto hinzufügen/modifizieren
passwd	Passwörter für Nutzerkonten anlegen/bearbeiten
groupadd	Benutzergruppen anlegen
chattr	Datei-Attribute verwalten
lsattr	Datei-Attribute anzeigen
chgrp	Gruppenzugehörigkeit von Dateien und Verzeichnissen verwalten
man	Benutzerhandbuch aufrufen
shutdown, reboot	Das System herunterfahren/neu starten
top	Dynamische Prozessübersicht
lscpu	Prozessorinformationen ausgeben
lshw	Hardwareinformationen ausgeben

Befehl	Beschreibung
kill	Prozess via PID stoppen und beenden
killall	Prozesse via Namen stoppen und beenden
nice	Prozessprioritäten definieren
pgrep	PID via Suchbegriff ermitteln
ps	Liste aller laufenden Prozesse anzeigen

sudo-Befehl in Linux

Der Linux sudo-Befehl (*substitute user do*) kann Programmaufrufen vorangestellt werden, um diese mit den Rechten eines anderen Nutzers bzw. einer anderen Nutzerin auszuführen. In der Regel ist dafür eine Passworteingabe notwendig. Der Befehl `sudo` fragt dabei immer nach dem Passwort des aufrufenden Benutzerkontos.

Wird der Befehl ohne Benutzernamen übergeben, wird der Superuser root als Ziel-Benutzerkonto verwendet.

```
sudo -u BENUTZERNAME PROGRAMMAUFRUF
```

bash

ls-Befehl in Linux

Der Linux ls-Befehl steht für *list* und wird verwendet, um den Inhalt eines Verzeichnisses (die Namen aller Dateien und Ordner, die sich im angegebenen Verzeichnis befinden) anzuzeigen. Die Syntax des Befehls lautet:

```
ls [OPTIONEN] VERZEICHNIS
```

bash

Wird `ls` ohne Verzeichnis-Angabe benutzt, listet der Befehl den Inhalt des aktuellen Verzeichnisses auf. Mithilfe zusätzlicher Optionen definieren Sie, welche Informationen angezeigt und wie diese dargestellt werden.

cd-Befehl in Linux

Der Kommandozeilenbefehl Linux cd-Befehl steht für *change directory* und dient der Navigation im Verzeichnisbaum. Die Syntax des Befehls lautet:

```
cd [OPTION] VERZEICHNIS
```

bash

Wird kein Zielverzeichnis angegeben, wechselt `cd` automatisch in das Home-Verzeichnis des Benutzerkontos. Wird `cd` mit einem nachgestellten Minus-Zeichen (-) verwendet, erfolgt ein Wechsel ins vorherige Verzeichnis.

touch-Befehl in Linux

Der Linux touch-Befehl wird verwendet, um Zugriffs- und Änderungs-Zeitstempel von Dateien zu verändern. Wird `touch` auf eine nicht-existente Datei angewendet, wird diese automatisch

angelegt. Der Befehl eignet sich daher auch dazu, leere Dateien zu erstellen. Nutzen Sie `touch` nach diesem Muster:

```
touch [OPTIONEN] DATEI
```

bash

Um den Zeitstempel einer Datei auf ein gewünschtes Datum zu setzten, verwenden Sie die OPTION `-t` inklusive Zeitangabe im Format `[JJ]MMTThhmm[.ss]`.

Beispiel:

```
touch -t 1703231037 datei.txt
```

bash

Zugriffs- und Änderungszeitstempel werden auf den 23. März 2017, 10:37 gesetzt. Die Modifikation lässt sich über die Optionen `-a` und `-m` auf Zugriffs- bzw. Änderungszeitstempel eingrenzen. Wird der Befehl `touch` ohne Option `-t` verwendet, kommt der aktuelle Zeitstempel zum Einsatz.

mkdir-Befehl in Linux

Der Linux mkdir-Befehl steht für *make directory* und ermöglicht es, neue Verzeichnisse anzulegen. Verwenden Sie folgende Syntax, um ein neues Verzeichnis im aktuellen Verzeichnis zu erstellen:

```
mkdir [OPTION] VERZEICHNISNAME
```

bash

Soll ein Verzeichnis in einem bestimmten Zielverzeichnis erstellt werden, geben Sie den absoluten oder relativen Pfad zum Verzeichnis an.

rm-Befehl in Linux

Der Linux rm-Befehl (*remove*) löscht Dateien oder ganze Verzeichnisse unwiederbringlich. Dem Programmaufruf liegt folgendes Schema zugrunde:

```
rm [OPTIONEN] DATEI/VERZEICHNIS
```

bash

Soll ein Verzeichnis inklusive aller Unterverzeichnisse gelöscht werden, verwenden Sie `rm` mit der OPTION `-R` (*–recursive*).

rmdir-Befehl in Linux

Möchten Sie ein bestimmtes Verzeichnis löschen, nutzen Sie den Kommandozeilenbefehl `rmdir` (*remove directory*) gemäß folgender Syntax:

```
rmdir [OPTION] VERZEICHNIS
```

bash

Mit `rmdir` lassen sich lediglich leere Verzeichnis löschen. Um ein Verzeichnis inklusive aller enthaltenen Dateien und Unterordner zu löschen, verwenden Sie den Befehl `rm` (*remove*) mit der Option `-r`.

In anderen Artikel finden Sie weitere Methoden, wie sie eine Linux-Datei löschen oder ein Linux-Verzeichnis löschen.

mv-Befehl in Linux

Der Linux mv-Befehl (*move*) kopiert eine Datei oder ein Verzeichnis und löscht das Ursprungselement. Erfolgt dies innerhalb desselben Verzeichnisses, kann `mv` zum Umbenennen von Dateien verwendet werden.

Dem Programmaufruf liegt folgendes Schema zugrunde:

```
mv [OPTION] QUELLE ZIEL
```

bash

cp-Befehl in Linux

Der Linux cp-Befehl (*copy*) kommt zur Anwendung, um Dateien und Verzeichnisse zu kopieren. Die grundlegende Syntax des Befehls lautet:

```
cp [OPTIONEN] QUELLE ZIEL
```

bash

Bei der QUELLE handelt es sich um das Element, das kopiert werden soll. Als ZIEL des Kopiervorgangs wird entweder eine Datei oder ein Verzeichnis definiert. Definieren Sie eine bereits vorhandene Datei als Zieldatei, wird deren Inhalt mit dem Inhalt der Quelldatei überschrieben. Alternativ haben Sie die Möglichkeit, die Zieldatei als neue Datei mit gewünschtem Namen zu erstellen.

pwd-Befehl in Linux

Nutzen Sie den Linux pwd-Befehl (kurz für *print working directory*), um sich den Namen des aktuellen Arbeitsverzeichnisses auszugeben zu lassen.

Die Syntax des Befehls lautet:

```
pwd [OPTIONEN]
```

bash

zip-Befehl in Linux

Nutzen Sie den `zip`-Befehl, um mehrere Dateien in eine Zip-Archive zu komprimieren. Die Syntax des Befehls lautet:

```
zip ZIEL DATEIEN
```

bash

Beim ZIEL handelt es sich um den Namen bzw. Pfad der entstehenden Zip-Datei. DATEIEN bezeichnet die Dateinamen bzw. Pfade der zu komprimierenden Dateien (durch Leerzeichen getrennt).

unzip-Befehl in Linux

Mit `unzip` können Sie Dateien aus Zip-Archiven extrahieren. Die Syntax lautet:

```
unzip DATEI.zip -d ZIEL
```

bash

Hier bezeichnet DATEI die Zip-Archive, aus der die Dateien zu extrahieren sind. Optional können Sie mit der Option -d ZIEL ein Zielverzeichnis angeben, wo die entstehenden Dateien abgelegt werden. Sonst werden die Dateien im aktuellen Verzeichnis abgelegt.

ln-Befehl in Linux

Der Linux ln-Befehl (kurz für *link*) erzeugt eine Verknüpfung zu einer Datei oder einem Verzeichnis. Dadurch wird ein weiterer Verzeichnis-Eintrag für diese Datei erzeugt, der es Ihnen ermöglicht, über einen weiteren Datei-Pfad auf die entsprechende Datei zuzugreifen. Der Aufruf von `ln` muss immer mindestens den Pfad zur Quelldatei enthalten.

```
ln [OPTIONEN] pfad/zur/quelldatei
```

bash

Die Verlinkung wird in diesem Fall im aktuellen Arbeitsverzeichnis unter demselben Namen erstellt. Alternativ haben Sie die Möglichkeit, einen Zielpfad anzugeben und die Verknüpfung beliebig zu benennen:

```
ln [OPTIONEN] pfad/zur/quelldatei pfad/zur/verknüpfung
```

bash

cat-Befehl in Linux

Der Linux cat-Befehl (kurz für: *concatenate*) wurde als Werkzeug für das Zusammenführen von Datei-Inhalten entwickelt und kann als Pager zum Anzeigen von Datei-Inhalten im Terminal eingesetzt werden.

Rufen Sie `cat` mit folgender Syntax im Terminal auf, um eine Datei einzulesen und auf `stdout` (der Standardausgabe) auszugeben:

```
cat OPTIONEN DATEI
```

bash

Mehrere Dateien werden durch Leerzeichen getrennt:

```
cat OPTIONEN DATEI1 DATEI
```

bash

grep-Befehl in Linux

Mit dem Linux grep-Befehl lassen sich Textdateien durchsuchen. Als Suchmuster können beliebige Zeichenfolgen oder reguläre Ausdrücke zum Einsatz kommen. Nutzen Sie `grep` gemäß folgender Syntax:

```
grep [OPTIONEN] SUCHMUSTER [DATEI(EN)]
```

bash

Stößt `grep` auf einen String, der dem Suchmuster entspricht, wird die Zeilennummer unter Angabe des Dateinamens im Terminal ausgegeben. In der Regel wird `grep` auf alle Dateien im aktuellen Verzeichnis angewendet. Die Option `-r` ermöglicht eine rekursive Suche in Unterverzeichnissen.

diff-Befehl in Linux

Das Kommandozeilenprogramm `diff` dient dem Vergleich zweier Dateien. Alternativ lässt sich via `diff` ermitteln, ob zwei Verzeichnisse gleiche Dateien beinhalten.

Der Aufruf im Terminal Erfolg gemäß folgender Syntax:

```
diff [OPTIONEN] DATEI1 DATEI2
```

bash

cmp-Befehl in Linux

Anders als bei `diff` erfolgt der Abgleich bei `cmp` auf Byte-Ebene und eignet sich somit speziell für Binärdateien. Nutzen Sie `cmp` gemäß folgender Syntax:

```
cmp [OPTIONEN] DATEI1 DATEI2
```

bash

Findet `cmp` Unterschiede, gibt das Kommandozeilenprogramm Byte- und Zeilennummer der ersten Abweichung im Terminal aus.

tar-Befehl in Linux

Der Befehl `tar` ermöglicht es, verschiedene Dateien und Verzeichnisse sequenziell in eine tar-Datei zu schreiben und aus dieser bei Bedarf wiederherzustellen. Anders als bei dem unter Windows gebräuchlichen Zip-Format bleiben dabei alle Benutzerrechte der archivierten Datei auch nach dem Entpacken erhalten. Verwenden Sie folgende Syntax:

```
tar [OPTIONEN] DATEIEN
```

bash

Möchten Sie ein neues Archiv erstellen, verwenden Sie `tar` mit den Optionen `-c` (neues Archiv erzeugen) und `-f` (Archiv in angegebene Datei schreiben oder aus dieser auslesen). Erfahren Sie mehr in unserem Artikel Backup mit tar: So erstellen Sie Archive unter Linux.

echo-Befehl in Linux

Nutzen Sie den Linux echo-Befehl, um sich Zeichenketten zeilenweise auf der Standardausgabe (in der Regel das Terminal) ausgeben zu lassen.

Die allgemeine Syntax des Befehls lautet:

```
echo [OPTIONEN] STRING
```

bash

clear-Befehl in Linux

Nutzen Sie den Kommandozeilenbefehl `clear`, um den Bildschirminhalt zu löschen.

```
clear
```

bash

Sie erhalten ein leeres Terminal mit Eingabeaufforderung (Prompt). Ältere Eingaben bleiben im Scrollback-Buffer erhalten. Statt über diesen Befehl lässt sich das Terminal auch über die Tastenkombination [STRG] + [L] leeren.

ssh-Befehl in Linux

Mit `ssh` können Sie Ihren Rechner mit einem externen Rechner über das SSH-Protokoll verbinden. Sie befinden sich dann in der Shell des anderen Rechners. Die Syntax lautet:

```
ssh BENUTZERNAME@HOSTNAME
```

bash

Hier bezeichnen BENUTZERNAME und HOSTNAME jeweils den Benutzername, unter dem Sie sich anmelden möchten, und die Adresse des externen Rechners.

wget-Befehl in Linux

Mit dem Linux wget-Befehl können Sie Dateien aus dem Internet herunterladen. Nutzen Sie hierfür folgende Syntax:

```
wget [OPTION] LINK
```

bash

Hier bezeichnet LINK die URL, unter der die Datei zu finden ist. Optional können Sie die das optionale Argument `-c` nutzen, um einen unterbrochenen Download fortzusetzen.

ping-Befehl in Linux

Nutzen Sie den Linux ping-Befehl, um die Erreichbarkeit anderer Rechner im Netzwerk zu überprüfen. Dem Befehl liegt folgende Syntax zugrunde:

```
ping [OPTIONEN] ZIEL
```

bash

Zusammen mit der Paketumlaufzeit (*Round trip time*, RTT) – die Zeitspanne zwischen dem Aussenden des Datenpakets und dem Empfang der Antwort – schreibt `ping` auch die IP-Adresse des Zielsystems ins Terminal. Sie können mit optionalen Argumenten einstellen, nach wie vielen Paketen oder Sekunden `ping` sich selbst beendet.

ftp- oder sftp-Befehl in Linux

Dieses bietet Ihnen die Möglichkeit, Dateien zwischen dem lokalen System und einem anderen Rechner im Netzwerk auszutauschen. Dafür wird FTP (File Transfer Protocol) benutzt. Nutzen Sie `ftp` nach dem folgenden Muster, um eine Verbindung zum FTP-Server des Zielrechners aufzubauen.

```
ftp [OPTIONEN] [HOST[PORT]]
```

bash

Die Adressierung erfolgt via Hostname oder IP-Adresse. Die Angabe einer Portnummer ist optional. Verwenden Sie FTP nur in Netzwerken, denen sie vertrauen, denn dieses Protokoll ist nicht gesichert. Aus Sicherheitsgründen ist es fast immer empfehlenswert, SFTP (SSH File Transfer Protocol) zu verwenden. Der Linux-Befehl `sftp` funktioniert nach dem gleichen Prinzip wie `ftp`, aber hier ist die Übertragung verschlüsselt. SFTP nutzt als Standard Secure Shell (SSH), also auch dessen Authentifizierungsmethoden. Wie Sie SSH-Keys für Ihre Netzwerkverbindung nutzen, erklären wir in einem weiteren Artikel.

ip-Befehl in Linux

Das Kommandozeilenprogramm `ip` ist Teil der Programmsammlung `iproute2`, mit der sich Netzwerkschnittstellen über das Terminal abfragen und konfigurieren lassen. Die allgemeine Syntax des Befehls lautet:

```bash
ip [OPTIONEN] OBJEKT [BEFEHL [ARGUMENTE]]
```

bash

Welche Aktion via `ip` ausgeführt wird, definieren Sie mithilfe von Objekten, Subbefehlen und deren Argumenten.

Das Programm unterstützt diverse Objekte wie `address` (IP-Adresse), `link` (Netzwerkschnittstelle), `route` (Eintrag in der Routing-Tabelle) oder `tunnel`, auf die sich Subbefehle wie `add`, `change`, `del`, `list` oder `show` anwenden lassen.

Möchten Sie sich beispielsweise die IP-Adresse einer bestimmten Netzwerkschnittstelle (z.B. `eth0`) abrufen, verwenden Sie den Befehl `ip` in Kombination mit dem Objekt `address`, dem Befehl `show` und dem Argument `dev eth0`:

```bash
ip address show dev eth0
```

bash

Wie Sie sich in Linux eine IP-Adresse anzeigen lassen können, zeigen wir ausführlicher in einem weiteren Artikel.

apt-, pacman- und yum-Befehl in Linux

Jede Linux-Distribution verfügt über einen **Packet Manager**, mit dem Sie Softwarepakete herunterladen und verwalten können. Der Syntax für die App-Installation ist wie folgt:

```bash
apt install [PAKET] # Debian-basierte Distributionen wie Ubuntu
pacman -S [PAKET] # Arch-basierte Distributionen
yum install [PAKET] # Red Hat-basierte Distributionen
```

bash

Hierbei bezeichnet `[PAKET]` den Namen des Pakets bzw. Programms, das Sie installieren möchten. Meistens müssen diese Befehle per `sudo` im root-Modus ausgeführt werden. Bei anderen Distributionen, die andere Paketmanager benutzen, können die Befehle abweichen. Jeder Manager hat zudem auch Befehle, um Pakete zu entfernen, die Paketliste zu aktualisieren und alle installierten Pakete zu aktualisieren, unter anderem. Auf Ubuntu lauten diese Befehle folgendermaßen.

```
apt remove [PAKET] # Paket entfernen
apt update # Paketliste aktualisieren
apt upgrade # Pakete aktualisieren
```

bash

netstat-Befehl in Linux

Das Kommandozeilenprogramm `netstat` dient der Statusabfrage von Netzwerkschnittstellen. Die allgemeine Syntax des Befehls lautet:

```
netstat [OPTIONEN]
```

bash

Nutzen Sie `netstat` ohne Option, um sich alle offenen Sockets im Terminal ausgeben zu lassen. Verwenden Sie alternativ folgende Optionen, um Routingtabellen (`-r`), Schnittstellenstatistiken (`-i`), maskierte Verbindungen (`-M`) oder Netzlink-Nachrichten (`-N`) einzusehen. Erfahren Sie mehr in unserer Einführung in netstat.

traceroute-Befehl in Linux

Um den Transportweg eines IP-Datenpakets zwischen Ihrem System und einem Zielrechner nachzuvollziehen, können Sie mit dem Befehl traceroute den Weg von Datenpaketen verfolgen. Nutzen Sie folgendes Muster:

```
traceroute [OPTIONEN] HOSTNAME
```

bash

Via `traceroute` lässt sich ermitteln, welche Router und Internetknoten ein IP-Paket auf dem Weg zum Zielrechner passiert – beispielsweise, um der Ursache für eine Verzögerung nachzugehen.

route-Befehl in Linux

Mit dem Kommandozeilenprogramm `route` lassen sich IP-Routing-Tabellen des Kernels abfragen und bearbeiten. Dem Befehl liegt folgende Syntax zugrunde:

```
route [OPTIONEN] [add|del] [-net|-host] [ZIEL]
```

bash

Nutzen Sie den Befehl ohne Option, um sich die komplette Routing-Tabelle des Kernels anzeigen zu lassen:

```
route
```

bash

Möchten Sie beispielsweise eine Route zu einem Netzwerk setzen, verwenden Sie den Subbefehl `add`.

```
route add -net 10.0.0.0
```

bash

dig-Befehl in Linux

Bei `dig` handelt es sich um ein Lookup-Tool, mit dem sich Informationen von DNS-Servern abfragen und im Terminal ausgeben lassen. Das Kommandozeilenprogramm wird in der Regel nach diesem Schema verwendet, um die IP-Adresse und andere DNS-Informationen zu einem gegebenen Domainnamen abzufragen:

```
dig [@SERVER] [DOMAIN] [TYP]
```

bash

SERVER ist der DNS-Server, der nach den gewünschten Informationen durchsucht werden soll. Standardmäßig wird der Standard-Server aus der Datei `etc/resolv.conf` verwendet. DOMAIN steht für den Domainnamen, zu dem DNS-Informationen ermittelt werden sollen. Als TYP lässt sich der Abfragetyp festlegen, z. B. ANY (alle Einträge), A (IPv4-Record eines Hosts) oder AAAA (IPv6-Record eines Hosts). Als Standardabfragetyp ist A definiert.

mount- und unmount-Befehl in Linux

Soll ein Dateisystem über das Terminal in die Verzeichnisstruktur des Betriebssystems eingebunden werden, kommt unter Linux das Kommandozeilenprogramm `mount` zum Einsatz. Die allgemeine Syntax des Befehls lautet:

```
mount [OPTIONEN] GERÄT MOUNTPOINT
```

bash

GERÄT = Der Pfad zur Gerätedatei des Speichergeräts, das Sie als Partition einhängen möchten.

MOUNTPOINT = Die Stelle in der Verzeichnisstruktur Ihres Betriebssystems, an der Sie die Partition einbinden möchten. Der Mountpoint wird meist als absoluter Pfad angegeben.

Beispiel:

```
mount /dev/sdd /media/usb
```

bash

Das Gerät `sdd` wird in das Verzeichnis `/media/usb` eingehängt.

dd-Befehl in Linux

Das Kommandozeilenprogramm `dd` ermöglicht einen Kopiervorgang, bei dem Daten Bit für Bit aus einem Input File (`if`) ausgelesen und in ein Output File (`of`) geschrieben werden. Dem Aufruf des Programms liegt folgende Syntax zugrunde:

```
dd if=Quelle of=Ziel [OPTIONEN]
```

bash

Als Quelle und Ziel können dabei sowohl einzelne Dateien als auch ganze Partitionen (z.B. /dev/sda1) oder ein komplettes Speichergerät (z.B. /dev/sda) angegeben werden.

```
dd if=/dev/sda5 of=/dev/sdb1
```

bash

chmod-Befehl in Linux

Das Kommandozeilenprogramm `chmod` (kurz für *change mode*) dient der Rechtevergabe in unixoiden Dateisystemen (z.B. ext2, ext3, ext4, reiser, xfs). Die allgemeine Syntax des Befehls lautet:

```
chmod [OPTIONEN] MODUS DATEI
```

bash

bzw.

```
chmod [OPTIONEN] MODUS VERZEICHNIS
```

bash

Der Platzhalter MODUS steht dabei für die anzuwendende Rechtemaske. Wie Sie eine solche erstellen und was dabei zu beachten ist, erfahren Sie in unserem weiterführenden Artikel zur Vergabe von Zugriffsrechten mit chmod. Mithilfe der Option `-R` lassen sich vergebene Rechte rekursiv auf Unterordner und in einem Verzeichnis enthaltene Dateien ausweiten.

chown-Befehl in Linux

Der Linux chown-Befehl steht für *change owner* und bietet Ihnen die Möglichkeit, die Eigentümereinstellung einer Datei bzw. eines Verzeichnisses nach diesem Schema anzupassen:

```
chown [OPTIONEN] [BENUTZERNAME][:[GRUPPE]] DATEI
```

bash

bzw.

```
chown [OPTIONEN] [BENUTZERNAME][:[GRUPPE]] VERZEICHNIS
```

bash

Um das Eigentümerrecht für Benutzende oder Gruppen zu setzen, stehen Ihnen vier Kombinationsmöglichkeiten zur Verfügung. Besitzer und Gruppe werden entsprechend der Angaben neu gesetzt:

```
chown [OPTIONEN] besitzer_name:gruppen_name datei.txt

# Die Gruppe wird entsprechend der Angabe neu gesetzt, die Benutzerin bzw. der Benutzer bleibt unverändert:

chown [OPTIONEN] :gruppen_name datei.txt

# Der Besitzer wird entsprechend der Angabe neu gesetzt, die Gruppe bleibt unverändert:

chown [OPTIONEN] besitzer_name datei.txt

# Die Benutzerin bzw. der Benutzer wird entsprechend der Angabe neu gesetzt. Die Gruppe wird auf die Standardgruppe des eingeloggten Benutzerkontos gesetzt:

chown [OPTIONEN] besitzer_name: datei.txt

# Die Änderungen lassen sich mithilfe der OPTION `-R` rekursiv auf Unterverzeichnisse ausweiten.
```

bash

adduser-Befehl in Linux

Die einfachste Möglichkeit, einen Benutzer-Account anzulegen, bietet das Kommandozeilenprogramm `adduser`. Dies ist ein Perl-Skript, das auf dem Linux useradd-

Befehl basiert und dieselben Funktionen in nutzerfreundlicher Form bietet. Der Befehl `adduser` erfordert Root-Rechte und wird gemäß folgender Syntax verwendet:

```
adduser [OPTIONEN] BENUTZERNAME
```

bash

Verwenden Sie `adduser` ohne Optionen, werden neben dem neuen Benutzer-Account automatisch eine Benutzer-ID, das Home-Verzeichnis und eine gleichnamige Benutzergruppe angelegt.

```
adduser test
```

bash

Anschließend folgt ein interaktiver Dialog, in dem Sie das Passwort sowie die erweiterten Benutzerinformationen (realer Name, Büronummer, Telefonnummern etc.) definieren können.

passwd-Befehl in Linux

Verwenden Sie den Linux passwd-Befehl, um das Passwort eines Benutzerkontos zu ändern oder Sperr- und Änderungsintervalle zu definieren. Dem Befehl liegt folgende Syntax zugrunde:

```
passwd [OPTIONEN] BENUTZERNAME
```

bash

Möchten Sie das Passwort eines anderen Benutzerkontos ändern, benötigen Sie Root-Rechte. Verwenden Sie den Befehl `passwd` ohne Benutzername, um Ihr eigenes Passwort zu ändern. Soll das Passwort des ausgewählten Benutzerkontos gesperrt werden, verwenden Sie den Befehl `passwd` mit der Option `-l` (*lock*). Weitere Optionen bieten die Möglichkeit, eine Verfallsdauer für Passwörter `(-x)` sowie Warn- `(-w)` und Sperrintervalle `(-i)` zu definieren:

groupadd-Befehl in Linux

Das Kommandozeilenprogramm `groupadd` dient dem Anlegen neuer Benutzergruppen. Verwenden Sie `groupadd` mit Root-Rechten nach diesem Muster:

```
sudo groupadd [OPTIONEN] GRUPPE
```

bash

Jede neu erstelle Gruppe erhält eine einzigartige Gruppen-ID (GID). Gruppen-IDs zwischen 0 und 99 sind für Systemgruppen reserviert. Möchten Sie die GID einer neuen Benutzergruppe selbst definieren, verwenden Sie den Kommandozeilenbefehl `groupadd` mit der Option `-g` (GID). Möchten Sie eine Systemgruppe anlegen, verwenden Sie die Option `-r` (root).

chattr-Befehl in Linux

Das Kommandozeilenprogramm `chattr` (kurz für *change attribute*) ermöglicht es Ihnen, Dateien oder Verzeichnisse mit Attributen zu versehen. Nutzen Sie `chattr` gemäß folgender Syntax, um ein Attribut zu setzen:

```
chattr [OPTIONEN] +ATTRIBUT DATEI
```

bash

Ersetzen Sie das Pluszeichen durch ein Minuszeichen, um Attribute wieder zu entfernen. Setzten Sie beispielsweise das Attribut `-i`, um Änderungen (Löschvorgänge oder Modifikationen) an einer Datei oder einem Verzeichnis zu unterbinden. Weitere Attribute und mögliche Optionen entnehmen Sie der Handbuchseite zum Programm `chattr`.

lsattr-Befehl in Linux

Möchten Sie sich anzeigen lassen, welche Attribute für eine Datei oder ein Verzeichnis gesetzt wurden, verwenden Sie den Kommandozeilenbefehl `lsattr` (kurz für *list attributes*) nach folgendem Schema:

```
lsattr [OPTIONEN] DATEI/VERZEICHNIS
```

bash

chgrp-Befehl in Linux

Der Befehl `chgrp` steht für *change group* und kommt bei der Verwaltung von Gruppenzugehörigkeiten für Dateien und Verzeichnisse zur Anwendung. Um `chgrp` auf eine ausgewählte Datei oder ein Verzeichnis anwenden zu können, müssen Sie Eigentümer- oder Root-Rechte besitzen. Zudem stehen lediglich Gruppen zur Auswahl, denen Sie selbst angehören. `chgrp` kommt gemäß folgender Syntax zur Anwendung:

```
chgrp [OPTIONEN] GRUPPE DATEI
```

bash

oder:

```
chgrp [OPTIONEN] GRUPPE VERZEICHNIS
```

bash

Die Option `-R` bezieht Unterordner und in einem Verzeichnis enthaltene Dateien mit ein.

man-Befehl in Linux

Der Befehl `man` öffnet die Handbuchseiten (*Man-Pages*) Ihrer Linux-Distribution direkt im Terminal. Verwenden Sie folgendes Muster, um eine Handbuchseite aufzurufen:

```
man [OPTION] THEMA
```

bash

Die Linux-Man-Pages sind in 10 Themenbereiche unterteilt: Benutzerkommandos, Systemaufruf, Funktionen der Programmiersprache C, Dateiformate, Konfigurationsdateien, Spiele, Diverses, Kommandos zur Systemadministration, Kernelfunktionen und Neue Kommandos.

shutdown-Befehl in Linux

Der Linux shutdown-Befehl kann von Root-Nutzern verwendet werden, um das System herunterzufahren. Dem Befehl liegt folgende Syntax zugrunde:

```
shutdown [OPTIONEN] [ZEIT] [NACHRICHT]
```

bash

Möchten Sie einen Shutdown veranlassen, haben Sie die Möglichkeit, einen Zeitpunkt zu definieren, wann das System heruntergefahren werden soll. Nutzen Sie dafür entweder eine konkrete Zeitangabe (`hh:mm`) oder einen Countdown (+m). Andere Benutzende auf dem System bekommen eine Shutdown-Meldung. Diese kann bei Bedarf um eine individuelle Nachricht ergänzt werden. Wird der Befehl `shutdown` mit der Option `-r` verwendet, folgt dem Herunterfahren des Systems ein Reboot.

top-Befehl in Linux

Mit dem Befehl `top` rufen Sie eine dynamische Übersicht aller laufenden Prozesse ab. Dem Aufruf liegt folgendes Schema zugrunde:

```
top [OPTIONEN]
```

bash

Die Ausgabe der Prozessinformationen lässt sich mithilfe diverser Optionen anpassen. Zudem unterstützt die `top`-Prozessübersicht (unter anderem) folgende Hotkeys, um die Ausgabe zu sortieren:

- [P] = Sortiert die Ausgabe nach CPU-Last
- [M] = Sortiert die Ausgabe nach Speicherverbrauch
- [N] = Sortiert die Ausgabe nummerisch nach PID
- [A] = Sortiert die Ausgabe nach Alter
- [T] = Sortiert die Ausgabe nach Zeit
- [U BENUTZERNAME oder UID] = Filtert die Ausgabe nach dem jeweiligen Benutzerkonto

Verwenden Sie den Hotkey [H], um sich eine Hilfeseite anzeigen zu lassen und [Q], um die Prozessübersicht zu verlassen.

lscpu-Befehl in Linux

Nutzen Sie `lscpu` (kurz für *list cpu*) nach diesem Muster, um sich Informationen zur CPU-Architektur im Terminal ausgeben zu lassen.

```
lscpu [OPTIONEN]
```

bash

Mögliche Optionen entnehmen Sie den Handbuchseiten Ihres Betriebssystems.

lshw-Befehl in Linux

Der Befehl `lshw` steht für *list hardware*, und gibt Ihnen Informationen zu Hardwarekomponenten im Terminal aus. Nutzen Sie `lshw` nach diesem Schema:

```
lshw [OPTIONEN]
```

bash

Der Befehl unterstützt diverse Optionen, mit denen sich das Ausgabeformat (`-html`, `-xml`, `-short`, `-businfo`) sowie der Umfang der Informationen (z. B. `-sanitize` zum Verstecken sensibler Informationen) anpassen lassen.

kill-Befehl in Linux

Bei `kill` handelt es sich um ein Kommandozeilenprogramm, mit dem sich Prozesse anhalten und beenden lassen. Der Befehl wird nach diesem Schema mit einem gewünschten Signal und der ID des ausgewählten Prozesses übergeben.

```bash
kill [OPTIONEN] [-SIGNAL] PID
```

Gängige Signale sind:

1. `TERM`: Veranlasst einen Prozess, sich selbst zu beenden (Standard)
2. `KILL`: Erzwingt das Beenden eines Prozesses (durch das System)
3. `STOP`: Hält einen Prozess an
4. `CONT`: Lässt einen angehaltenen Prozess weiterlaufen

killall-Befehl in Linux

Verwenden Sie den Linux killall-Befehl in Kombination mit einem bestimmten Suchbegriff, um lediglich die Prozesse zu beenden, deren Namen sich exakt mit diesem decken (es werden die ersten 15 Zeichen abgeglichen).

```bash
killall [OPTIONEN] [-SIGNAL] [PROZESSNAME]
```

Die Option `-e` (*–exact*) ermöglicht es, den Abgleich auf alle Zeichen des Prozessnamens auszuweiten.

nice-Befehl in Linux

Der Kommandozeilenbefehl `nice` weist einem Prozess beim Start einen nice-Wert zwischen -20 und +19 in ganzzahligen Schritten zu, nach der die zu Verfügung stehende Rechenleistung des Systems verteilt wird. Die Spanne -20 bis +19 entspricht den Linux-Prioritätsleveln 100 bis 139. Ein Prozess mit einem nice-Wert von -20 hat somit eine höhere Priorität als ein Prozess mit einem nice-Wert von 19. Die Syntax lautet:

```bash
nice [OPTION] [BEFEHL]
```

Ohne weitere Angabe startet jeder Prozess mit einem nice-Wert von 0. Nutzen Sie die Option -n, um Prozessprioritäten zu definieren. Dabei ist zu beachten, dass negative Prioritäten nur mit Root-Rechten vergeben werden können.

pgrep-Befehl in Linux

Das Kommandozeilenprogramm `pgrep` gleicht die Liste laufender Prozesse mit einem Suchbegriff ab und gibt bei Übereinstimmung die jeweiligen PIDs aus. Die allgemeine Syntax des Aufrufs lautet:

```
pgrep [OPTIONEN] Suchbegriff
```

bash

Standardmäßig gibt `pgrep` die PIDs aller Prozesse aus, die den Suchbegriff enthalten. Soll die Suche auf exakte Übereinstimmungen eingegrenzt werden, verwenden Sie den Befehl mit der Option `-x`. Möchten Sie neben der PID auch den Prozessnamen abrufen, verwenden Sie `pgrep` mit der Option `-l`. `pgrep` unterstützt (ähnlich wie `grep`) Suchbegriffe auf Basis regulärer Ausdrücke.

ps-Befehl in Linux

Der Linux ps-Befehl gibt eine Liste aller laufenden Prozesse im Terminal aus.

```
ps [OPTIONEN]
```

bash

Benötigen Sie eine detaillierte Ausgabe verwenden Sie `ps` mit den Optionen `-f` (detailliert) oder `-F` (sehr detailliert). Weitere mögliche Optionen entnehmen Sie den Handbuchseiten Ihres Betriebssystems.

Weitere Linux-Befehle im Überblick

Grundkommandos

In der Kategorie Grundkommandos finden Sie grundlegende Linux-Befehle, die der **Steuerung des Terminals** dienen. Erfahren Sie, wie Sie den Sichtbereich des Terminals leeren, vorhergehende Terminaleingaben aus dem Verlauf (der History) abrufen oder die Terminal-Sitzung beenden.

1. exit

Der Kommandozeilenbefehl `exit` beendet die aktuelle Sitzung und schließt das Terminal.

```
exit
```

bash

Verwenden Sie alternativ die Tastenkombination [STRG] + [D].

2. help

Nutzen Sie den Befehl help, um sich eine Liste aller integrierten Shell-Befehle (Built-in-Befehle) anzeigen zu lassen. Rufen Sie help in Kombination mit einem Shell-Befehl auf, um sich eine Kurzbeschreibung des jeweiligen Befehls ausgeben zu lassen.

```
help BEFEHL
```

bash

3. history

In der Bash werden die letzten 500 Befehle, die Sie über die Kommandozeile eingegeben haben, in der sogenannten History gespeichert. Diese Funktion dient als Eingabehilfe und ermöglicht Ihnen, die Liste der vorhergehenden Befehle mit den Pfeiltasten zu durchsuchen und erneut auszuführen.

Die History kann über die Tastenkombination [STRG] + [R] nach Stichworten durchsucht werden. Alternativ haben Sie die Möglichkeit, sich die komplette Liste nummeriert im Terminal ausgeben zu lassen. Nutzen Sie dazu den Befehl `history` ohne Optionen und Argumente.

```
history
```

bash

Möchten Sie die Ausgabe filtern, kombinieren Sie `history` via Linux Pipe mit dem Kommandozeilenprogramm `grep` (siehe Suchoptionen) und einem Suchwort.

```
history | grep SUCHWORT
```

bash

Hilfeseiten

Sie wissen nicht weiter? Kein Problem. Unter Linux stehen Ihnen diverse **Hilfs- und Dokumentationsseiten** wie die *Unix-Man-Pages* und die *GNU-Infoseiten* direkt über das Terminal zur Verfügung. Diese beinhalten eine detaillierte Beschreibung aller Kommandozeilenprogramme, Systemaufrufe, Konfigurationsdateien, Dateiformate und Kernelfunktionen. Mit dem Linux whatis-Befehl und `apropos` finden Sie in der Kategorie Hilfsseiten zudem Kommandozeilenprogramme, mit denen sich die Handbuchseiten Ihres Betriebssystems nach Stichworten durchsuchen lassen.

1. apropos

Nutzen Sie `apropos` um die Seitentitel und Beschreibungen der Handbuchseiten Ihres Betriebssystems nach Stichworten zu durchsuchen. Orientieren Sie sich an folgendem Muster:

```
apropos [OPTIONEN] SUCHBEGRIFF
```

bash

Der Befehl unterstützt verschieden Optionen. Grenzen Sie die Suche mit der Option `-e` auf exakte Entsprechungen ein oder nutzen Sie Platzhalter (`-w '\*SUCHBEGRIFF'`) und reguläre Ausdrücke (`-r`).

2. info

Über den Befehl `info` lassen sich GNU-Informationsseiten zu einem bestimmten Thema ausgeben. Diese entsprechen in den meisten Fällen den Handbuchseiten, die sich via `man` aufrufen lassen, weisen im Gegensatz zu diesen jedoch Verlinkungen auf, die Ihnen die Navigatoren im Handbuch erleichtern. Verwenden Sie folgende Syntax:

```
info [OPTION] THEMA
```

bash

Ein Aufruf ohne Option und Thema führt Sie ins Hauptmenü der GNU-Infoseiten.

3. pinfo

Mit `pinfo` steht ihnen eine Variante des Kommandozeilenprogramms `info` zur Verfügung, die sich am Kommandozeilenbrowser Lynx orientiert und Ihnen Infoseiten mit farblich hervorgehobenen Links ausgibt. Nutzen Sie `pinfo` nach demselben Schema wie den `info`-Befehl.

```
pinfo [OPTIONEN] THEMA
```

bash

4. whatis

Das Kommandozeilenprogramm `whatis` dient der Stichwortsuche in den Handbuchseiten. Rufen Sie das Programm mit einem beliebigen Suchbegriff auf, um das Handbuch Ihres Betriebssystems nach exakten Übereinstimmungen zu durchsuchen. Findet es eine Entsprechung, gibt `whatis` eine Kurzbeschreibung im Terminal aus.

```
whatis [OPTIONEN] SUCHBEGRIFF
```

bash

Auch `whatis` (`-w '\*SUCHBEGRIFF'`) unterstützt Platzhalter und reguläre Ausdrücke (`-r`).

Verzeichnisoperationen

Linux-Befehle für Verzeichnisoperationen verwenden Sie, um Verzeichnisse auf Ihrem System über das Terminal zu erstellen, zu löschen und zu verwalten sowie im Verzeichnisbaum zu navigieren. Zu den wichtigsten Kommandozeilenbefehlen aus dieser Kategorie zählen `cd`, `ls`, `mkdir`, `rmdir`.

1. chroot

Der Befehl `chroot` (kurz für: *change root*) wird verwendet, um einen Befehl in einem anderen Wurzelverzeichnis auszuführen. Zur Anwendung kommt `chroot` beispielsweise, um kritische Programme vom übrigen Dateisystem zu isolieren. Der Aufruf des Programms erfordert Root-Rechte und orientiert sich an folgendem Schema:

```
chroot VERZEICHNIS BEFEHL
```

bash

2. mkdirhier

Mit `mkdirhier` lassen sich ganze Verzeichnishierarchien mit einem einzigen Kommandozeilenbefehl erstellen:

```
mkdirhier [OPTION] /home/user/verzeichnis1/verzeichnis2/verzeichnis3
```

bash

Existieren `verzeichnis1` und `verzeichnis2` bereits, erstellt `mkdirhier` lediglich `verzeichnis3`. Andernfalls werden alle drei Verzeichnisse erzeugt.

3. tree

Während `ls` lediglich den Inhalt von Verzeichnissen auflistet, lässt sich mit dem Kommandozeilenprogramm `tree` die gesamte Verzeichnishierarchie rekursiv als Baumstruktur ausgeben. Der Aufruf erfolgt gemäß folgender Syntax:

```
tree [OPTIONEN] [VERZEICHNIS]
```

bash

Dateioperationen

Linux-Commands dieser Rubrik ermöglichen Ihnen diverse Dateioperationen aus dem Terminal heraus. Nutzen Sie grundlegende Linux-Befehle wie `cp`, `mv` und `rm`, um Dateien auf Ihrem System zu kopieren, zu verschieben, umzubenennen oder zu löschen.

1. basetree

Dem Kommandozeilenbefehl `basename` wird ein Dateipfad übergeben; es gibt lediglich den Dateinamen ohne vorangestellten Pfad zurück. Die Syntax des Befehls lautet:

```
basename [OPTIONEN] Pfad/zur/Datei [SUFFIX]
```

bash

Der Befehl kann durch Optionen auf mehrere Dateien ausgeweitet werden.

2. comm

Nutzen Sie das Kommandozeilenprogramm `comm`, um sortierte Dateien (z. B. via `sort`) zeilenweise zu vergleichen. Dem Programmaufruf liegt folgende Syntax zugrunde:

```
comm [OPTIONEN] DATEI1 DATEI2
```

bash

Das Programm unterstützt drei Optionen:

- -1 = einzigartige Zeilen aus `DATEI1` unterdrücken
- -2 = einzigartige Zeilen aus `DATEI2` unterdrücken
- -3 = alle Zeilen unterdrücken, die in beiden Dateien enthalten sind

3. cut

Der Befehl `cut` ermöglicht es Ihnen, Inhalte spaltenweise aus den Textzeilen einer Datei (z. B. Log- oder CSV-Dateien) zu extrahieren. Die Syntax des Befehls lautet:

```
cut [OPTIONEN] DATEI
```

bash

Die genaue Position eines zu extrahierenden Ausschnitts wird über die Optionen `-b` (Byteposition), `-c` (Zeichenposition), `-d` (Trennzeichen) und `-f` (Feld) definiert.

4. dirname

`dirname` stellt das Gegenstück zu `basename` dar. Der Kommandozeilenbefehl ermöglicht es, den Pfadanteil aus einem Dateipfad zu extrahieren und ohne Dateinamen im Terminal auszugeben. Die Syntax des Befehls lautet:

```
dirname [OPTIONEN] Pfad/zur/Datei
```

bash

5. file

Mit dem Kommandozeilenbefehl `file` lassen sich Informationen zum Dateityp einer Datei ausgeben. Dem Aufruf liegt folgendes Muster zugrunde:

```
file [OPTIONEN] DATEI
```

bash

6. lsof

Der Linux lsof-Befehl steht für *list open files*, ein Hilfsprogramm, das Ihnen Informationen über geöffnete Dateien nach PID (Prozess-ID) sortiert im Terminal ausgibt. Der Aufruf über das Terminal erfolgt nach diesem Schema:

```
lsof [OPTIONEN]
```

bash

Da unixoide Systeme wie Linux dem Grundsatz „Everything is a file“ (Alles ist eine Datei) folgen, ist die Liste, die `lsof` ausgibt, entsprechend lang. In der Regel kommen daher Optionen zum Einsatz, um die Ausgabe einzugrenzen.

7. md5sum

Mithilfe des Kommandozeilenbefehls `md5sum` lassen sich MD5-Prüfsummen für Dateien berechnen und überprüfen.

8. paste

Ähnlich wie `cat` ermöglicht auch das Kommandozeilenprogramm `paste` die Ausgabe von Dateiinhalten auf die Standardausgabe. Doch während `cat` Inhalte lediglich aneinanderfügt, verknüpft `paste` diese spaltenweise. Das Grundschema des Befehls lautet:

```
paste [OPTIONEN] DATEI1 DATEI2 …
```

bash

Welches Trennzeichen `paste` verwendet, lässt sich mithilfe der Option `-d` individuell anpassen. Als Standardtrennzeichen kommen Tabs zum Einsatz. Über die Option `-s` (seriell) lässt sich zudem ein zweiter Modus aktivieren. Bei diesem werden alle Zeilen der ersten Eingabedateien in der Ausgabe in ihrer eigenen Zeile übertragen. Jede Zeile der Ausgabe enthält somit lediglich Inhalte einer Eingabedatei.

9. rename

Bei `rename` handelt es sich um ein Kommandozeilenprogramm, das es ermöglicht, Dateien und Ordner mithilfe regulärer Ausdrucke umzubenennen. Anders als `mv` bietet

sich `rename` damit für Dateioperationen an, bei denen die Namen mehrerer Dateien teilweise oder ganz angepasst werden sollen. Verwenden Sie `rename` nach diesem Schema:

```
rename [OPTIONEN] 'REGULÄRER_AUSDRUCK' DATEIEN
```

bash

Reguläre Ausdrücke entsprechen bei Ersetzungen folgender Syntax:

```
s/SUCHMUSTER/ERSETZUNG/MODIFIER
```

bash

10. shred

Bei `shred` handelt es sich um ein Kommandozeilenprogramm, das ein sicheres Löschen von Dateien ermöglicht. Ausgewählte Elemente werden im Rahmen des Löschvorgangs überschrieben und lassen sich somit selbst mit forensischen Mitteln nicht wiederherstellen. Die allgemeine Syntax des Befehls lautet:

```
shred [OPTIONEN] DATEI
```

bash

11. sort

Nutzen Sie den Kommandozeilenbefehl `sort`, um Dateilisten und Programmausgaben zeilenweise numerisch und alphabetisch zu sortieren. Die allgemeine Syntax des Befehls lautet:

```
sort [OPTIONEN] DATEI
```

bash

Die Sortiermethode lässt sich mithilfe von Optionen anpassen: zum Beispiel numerisch (`-n`), zufällig (`-R`) oder in umgekehrter Reihenfolge (`-r`).

12. split

Der Kommandozeilenbefehl `split` wird verwendet, um Dateien aufzuteilen. Die zugrundeliegende Syntax lautet:

```
split [OPTIONEN] [INPUT [PRÄFIX]]
```

bash

Der Platzhalter INPUT entspricht der Datei, die aufgeteilt werden soll. Das *PRÄFIX* fungiert als Vorsilbe für die Namen der Teildateien. Deren Benennung liegt folgendes Muster zugrunde:

```
PRÄFIXaa, PRÄFIXab, PRÄFIXac …
```

bash

Wird kein Präfix definiert, greift `split` auf das Standardpräfix `x` zurück. Mit der Option `-b` (Bytes) lässt sich die Größe der Teildateien festlegen. Die Angabe erfolgt wahlweise in Byte (b), Kilobyte (k) oder Megabyte (m).

Beispiel:

```
split -b 95m archiv.tgz split-archiv.tgz.
```

bash

13. stat

Mit dem Kommandozeilenbefehl `stat` (*status*) lassen sich Zugriffs- und Änderungs-Zeitstempel ausgewählter Dateien und Verzeichnisse ausgeben. Die allgemeine Syntax des Befehls lautet:

```
stat [OPTIONEN] DATEI
```

bash

Das Ausgabeformat lässt sich mithilfe von Optionen anpassen.

14. uniq

Der Kommandozeilenbefehl `uniq` wird in der Regel in Kombination mit `sort` verwendet, um sortierte Dateien von doppelten Zeilen zu bereinigen. In folgendem Beispiel wird der Befehl `sort` durch eine Pipe (`|`) mit dem Befehl `uniq` verknüpft, um eine Datei zunächst zu sortieren und anschließend ohne doppelte Zeilen auszugeben:

```
sort datei.txt | uniq
```

bash

Suchoptionen

Linux bietet diverse Kommandozeilenbefehle, mit denen sich das System direkt aus dem Terminal heraus durchsuchen lässt.

1. find

Der Befehl `find` hilft Ihnen, eine Linux-Datei zu suchen. Dem liegt folgende Syntax zugrunde:

```
find [OPTIONEN] [VERZEICHNIS] [SUCHBEDINGUNG] [AKTIONEN]
```

bash

Das angegebene Verzeichnis gilt als Startverzeichnis der Suche. Es werden somit das Startverzeichnis und dessen Unterverzeichnisse durchsucht. Geben Sie kein Verzeichnis an, startet `find` die Suche vom aktuellen Arbeitsverzeichnis aus.

Optionen ermöglichen es Ihnen, Suchkriterien und Aktionen zu definieren. Als Standardaktion ist `-print` voreingestellt: Die Ausgabe der vollständigen Dateinamen aller Suchergebnisse auf die Standardausgabe (in der Regel das Terminal). Weitere Optionen ermöglichen es, nach Dateinamen, Dateigröße, Zugriffszeitpunkt, usw. zu filtern. Diese sind in der entsprechenden man-Seite aufgelistet.

2. locate

Auch das Kommandozeilenprogramm `locate` ermöglicht eine Suche nach Dateien über das Terminal. Anders als bei `find` wird dabei jedoch nicht das Dateiverzeichnis durchsucht, sondern eine speziell dafür angelegte regelmäßig aktualisierte Datenbank. Dadurch liefert `locate` Suchergebnisse deutlich schneller aus als `find`. Um die Datenbank nach einer bestimmten Datei zu durchsuchen, wird `locate` nach diesem Schema aufgerufen:

```
locate SUCHMUSTER
```

bash

Das Suchmuster kann Metazeichen wie Platzhalter (`*`) beinhalten. Setzen Sie Suchmuster dieser Art in Anführungszeichen, um die Interpretation durch die Shell zu unterbinden.

3. tre-agrep

Auch `tre-agrep` dient der Suche nach Strings in Textdateien auf Basis von Suchmustern. Anders als bei `grep` werden dabei nicht nur exakte Übereinstimmungen ausgegeben, sondern auch Unschärfen wie Buchstabendreher oder fehlende Zeichen zugelassen. Das Programm stützt sich auf die TRE-Bibliothek und macht diese in der Kommandozeile nutzbar. Die Syntax von `tre-agrep` entspricht der des `grep`-Befehls:

```
tre-agrep [OPTIONEN] SUCHMUSTER DATEI(EN)
```

bash

Über Optionen lässt sich eine maximale Fehleranzahl definieren. In folgendem Beispiel wird maximal eine Abweichung vom Suchmuster toleriert.

```
tre-agrep -1 'Linux' test.txt
```

bash

4. updatedb

Eine Suche via `locate` funktioniert nur zuverlässig, wenn die Datei `/var/lib/locatedb` kontinuierlich auf dem neusten Stand gehalten wird. Der Befehl `updatedb` ermöglicht es Ihnen, die Datenbank manuell zu aktualisieren. Beachten Sie, dass Sie für den Aufruf Root-Rechte benötigen:

```
updatedb
```

bash

5. whereis

Mit dem Befehl `whereis` lokalisieren Sie die Binär-, Quellcode- oder Handbuch-Dateien ausgewählter Programme. Die allgemeine Syntax des Befehls lautet:

```
whereis [OPTIONEN] PROGRAMM
```

bash

Über Optionen lässt sich die Suche auf bestimmte Datei-Typen oder Verzeichnisse eingrenzen.

6. which

Möchten Sie die Binärdatei eines Programms ermitteln, verwenden Sie den Befehl `which` nach folgendem Schema, um sich den Pfad im Terminal ausgeben zu lassen.

```
which [OPTIONEN] PROGRAMM
```

bash

Im Standardmodus gibt `which` die erste Datei aus, die gefunden wird. Verwenden Sie die Option `-a`, um sich alle Dateien anzeigen zu lassen, die das Suchkriterium erfüllen.

User-Informationen

Nutzen Sie die Kommandozeilenprogramme der folgenden Kategorie, um **detaillierte Informationen zu den im System registrierten Benutzende** sowie zu deren Gruppen und Prozessen abzurufen.

1. finger

Das Kommandozeilenprogramm `finger` dient dem Abruf von Benutzerinformationen. Dazu wird der gleichnamige Befehl in Kombination mit dem gewünschten Benutzernamen übergeben:

```
finger [Optionen] [BENUTZERNAME]
```

bash

Verwenden Sie `finger` ohne Benutzernamen, um Informationen zu Ihrem eigenen Account abzurufen.

2. groups

Der Befehl `groups` listet die Gruppenzugehörigkeiten eines ausgewählten Benutzer-Accounts auf. Verwenden Sie `groups` ohne Benutzername, um alle Gruppen aufzulisten, denen Ihr Benutzer-Account angehört. Nutzen Sie den Kommandozeilenbefehl nach diesem Muster:

```
groups [OPTIONEN] [BENUTZERNAME]
```

bash

3. id

Der Kommandozeilenbefehl `id` gibt Benutzer- und Gruppenkennungen des ausgewählten Benutzer-Accounts aus. Ihre eigenen IDs ermitteln Sie, indem Sie den Befehl ohne Benutzernamen übergeben.

```
id [OPTIONEN] [BENUTZERNAME]
```

bash

Der Umfang der Ausgabe lässt sich über Optionen einschränken.

4. last

Nutzen Sie den Befehl `last` nach diesem Schema, um sich eine Liste der zuletzt angemeldeten Nutzenden inklusive Login- und Logout-Zeiten ausgeben zu lassen:

```
last [OPTIONEN] [BENUTZERNAME]
```

bash

Die entsprechenden Informationen werden aus der wtmp-Datei unter `/var/log/wtmp` abgerufen. Möchten Sie lediglich die Informationen zu einem bestimmten Account abfragen, übergeben Sie den Kommandozeilenbefehl mit dem gewünschten Benutzernamen.

5. w

Der Befehl `w` gibt eine Liste aller angemeldeten Benutzende aus inklusive aller Prozesse, die von diesen ausgeführt werden. Verwenden Sie `w` in Kombinationen mit einem Benutzernamen, um den Befehl auf diesen Nutzer-Account zu beschränken:

```
w [OPTIONEN] [BENUTZERNAME]
```

bash

Umfang und Format der Ausgabe lassen sich bei Bedarf durch Optionen anpassen.

6. who

Der Befehl `who` gibt Ihnen detaillierte Informationen über die am System angemeldeten Benutzende aus. Die allgemeine Syntax des Aufrufs lautet

```
who [OPTION] [QUELLDATEI]
```

bash

Standardmäßig bezieht `who` Daten über aktuell angemeldete Nutzende aus der Datei `/var/run/utmp`. Optional kann eine der folgenden Dateien als Quelle der Informationen angeben werden.

7. whoami

Nutzen Sie den Befehl `whoami`, um Ihren eigenen Benutzernamen abzurufen.

```
whoami [OPTIONEN]
```

bash

User-Account-Management

Linux bietet Ihnen eine Reihe von Programmen, mit denen Sie Benutzer-Accounts und Gruppen direkt über das Terminal anlegen, löschen und verwalten. Eine Übersicht der wichtigen **Linux-Befehle für das User-Account-Management** haben wir Ihnen im Folgenden zusammengestellt. Darüber hinaus finden Sie in dieser Kategorie Linux-Terminal-Befehle, die es Ihnen ermöglichen, Programme mit den Rechten anderer Benutzenden inklusive des Super-Users Root aufzurufen.

1. chfn

Der Kommandozeilenbefehl `chfn` (kurz für *change finger*) ermöglicht es Ihnen, die erweiterten Informationen eines Benutzer-Accounts wie den realen Namen, die Büronummer sowie private und dienstliche Telefonnummern anzupassen. Die Syntax ist wie folgt:

```
chfn [OPTION "NEUER WERT"] [BENUTZERNAME]
```

bash

Welche Benutzerinformation durch einen neuen Wert ersetzt wird, definieren Sie mithilfe der Optionen `-f` (realer Name), `-r` (Büronummer), `-w` (Dienstnummer) und `-h` (Privatnummer).

2. chsh

Der Kommandozeilenbefehl `chsh` (kurz für *change shell*) ändert die Login-Shell eines ausgewählten Benutzerkontos. Orientieren Sie sich bei der Eingabe an folgendem Muster:

```
chsh [OPTIONEN] BENUTZERNAME
```

bash

Mit der Option `-s` können Sie die Login-Shell eines Benutzerkontos ändern.

3. deluser

Das Kommandozeilenprogramm `deluser` löscht sämtliche Einträge für einen ausgewählten Benutzer-Account aus den System-Account-Dateien. Der Aufruf von `deluser` erfordert Root-Rechte und orientiert sich an folgendem Schema:

```
deluser [OPTIONEN] BENUTZERNAME
```

bash

Wollen Sie das Home-Verzeichnis oder alle Dateien des Nutzerkontos löschen, verwenden Sie jeweils die Optionen `--remove-home` oder `--remove-all-files`. Weitere Optionen finden Sie auf der Handbuchseite des Befehls.

4. delgroup

Der Kommandozeilenbefehl `delgroup` (kurz für *delete group*) löscht eine bestehende Benutzergruppe. Um den Befehl auszuführen zu können, werden Root-Rechte benötigt. Die allgemeine Syntax von `delgroup` lautet:

```
delgroup [OPTIONEN] GRUPPE
```

bash

5. groupmod

Via `groupmod` lassen sich der Name und die Gruppen-ID (GID) einer bestehenden Benutzergruppe anpassen. Der Kommandozeilenbefehl wird mit Root-Rechten nach diesem Muster verwendet:

```
groupmod OPTIONEN GRUPPE
```

bash

Verwenden Sie `groupmod` mit der Option `-g`, um die GID anzupassen. Ein Aufruf mit der Option `-n` überschreibt den Gruppennamen.

6. newgrp

Bei `newgrp` (kurz für *new group*) handelt es sich um einen Befehl, der es angemeldeten Benutzerinnen und Benutzern ermöglicht, ihre aktuelle Gruppen-ID zu wechseln, ohne sich dafür erneut an- und abmelden zu müssen. Die allgemeine Syntax des Befehls lautet:

```
newgrp [-] [GRUPPE]
```

bash

Wird `newgrp` mit dem optionalen Parameter [-] übergeben, führt der Gruppenwechsel zu einem Neustart der Benutzerumgebung – so, als ob sich die Benutzerin bzw. der Benutzer

neu angemeldet hätte. Wird `newgrp` ohne Gruppenangabe verwendet, wird die in `/etc/passwd` angegebene Standardgruppe verwendet.

7. su

Auch der Befehl `su` erlaubt einen temporären Benutzerwechsel, um Programmaufrufe mit den Rechten eines Ziel-Benutzerkontos auszuführen. Anders als bei `sudo` wird das Passwort des Zielnutzers abgefragt, und Befehle werden dabei nicht direkt ausgeführt. Die allgemeine Syntax des Befehls lautet:

```
su [OPTIONEN] [BENUTZERNAME]
```

bash

Ein Aufruf ohne BENUTZERNAME wählt `root` als Ziel-Benutzerkonto aus.

8. usermod

Der Kommandozeilenbefehl `usermod` bietet Ihnen die Möglichkeit, bereits angelegte Benutzer-Accounts zu bearbeiten. Verwenden Sie `usermod` mit Root-Rechten nach diesem Muster:

```
usermod [OPTIONEN] BENUTZERNAME
```

bash

Welche Modifikationen vorgenommen werden sollen, definieren Sie mithilfe von Optionen. Beispielsweise können Sie mit der Option `-l NEUER_NAME` den Benutzername ändern. Weitere Optionen finden Sie in der entsprechenden man-Seite.

Systembefehle

In der Kategorie Systembefehle finden Sie grundlegende Linux-Commands zur Systemsteuerung. Nutzen Sie folgende Befehle, um das System aus dem Terminal heraus neu zu starten und herunterzufahren, auf Wunsch auch zeitgesteuert.

1. logger

Verwenden Sie `logger` nach folgendem Muster:

```
logger "IHRE NACHRICHT"
```

bash

Das System-Log finden Sie unter `/var/log/syslog`.

2. reboot

Bei `reboot` handelt es sich um einen Kommandozeilenbefehl der einen Neustart des Systems herbeiführt. Um einen Neustart einzuleiten, muss der Befehl mit Root-Rechten ausgeführt werden.

```
reboot [OPTIONEN]
```

bash

3. rtcwake

Bei `rtcwake` handelt es sich um einen Kommandozeilenbefehl, der es ermöglicht, das System zeitgesteuert zu starten und herunterzufahren. Dem Befehl liegt folgende Syntax zugrunde

```
rtcwake [OPTIONEN] [MODUS] [Zeit]
```

bash

Wählen Sie einen bestimmten Modus (`-m MODUS`) in den das System für eine bestimmte Zeit in Sekunden (`-s ZEIT IN SEKUNDEN`) versetzt werden soll. Alternativ haben Sie die Möglichkeit, Ihr System zu einem genau definierten Zeitpunkt (`-t UNIXZEIT`) aufzuwecken.

Systeminformationen

In der Kategorie Systeminformationen haben wir Kommandozeilenprogramme zusammengefasst, mit denen Sie **Informationen und Statusmeldungen abrufen** und sich einen umfassenden Überblick über den Zustand Ihres Systems verschaffen.

1. date

Der Befehl `date` gibt Ihnen die Systemzeit inklusive Datum aus.

```
date [OPTIONEN] [AUSGABEFORMAT]
```

bash

Möchten Sie im Rahmen eines Programmaufrufs mit einem bestimmten Datum arbeiten (siehe `rtcwake`), lässt sich dieses mithilfe der Option `-d 'DATUM'` definieren. Darüber hinaus werden diverse Optionen unterstützt, mit denen sich Datums- und Zeitangaben in ein gewünschtes Format übertragen lassen.

2. df

Nutzen Sie den Befehl `df` (*disk free*) nach folgendem Muster:

```
df [OPTIONEN] [DATEI]
```

bash

Wird der Befehl in Kombination mit einer bestimmten Datei verwendet, gibt das System lediglich den freien Speicherplatz der Partition an, auf der sich die Datei befindet. Sonst wird der freie Festplattenspeicher eingehängter Partitionen angezeigt. Die Option `-l` (*lokal*) begrenzt `df` auf lokale Dateisysteme. Zudem werden Optionen unterstützt, mit denen sich das Ausgabeformat anpassen lässt.

3. dmesg

Das Programm `dmesg` (kurz für *display message*) gibt Meldungen des Kernel-Ringpuffers im Terminal aus und ermöglicht es Ihnen so, Hardware- und Treiberfehler zu lokalisieren. Nutzen Sie `dmesg` nach diesem Schema:

```
dmesg [OPTIONEN]
```

bash

Die `dmesg`-Ausgabe enthält sämtliche Meldungen des Bootvorgangs und ist dementsprechend lang. Das Kommandozeilenprogramm wird daher oft in Kombination mit einem Pager wie `more`, `less` oder `tail` verwendet.

4. free

Der Befehl `free` gibt die Auslastung des Arbeitsspeichers aus. Die allgemeine Syntax lautet:

```
free [OPTIONEN]
```

bash

Als Ausgabe erhalten Sie zwei Angaben: `Mem` (*Memory*) und `Swap`. Auch `free` unterstützt die Option `-h`, mit der sich die Speicherauslastung in einem für Menschen lesbaren Format ausgeben lässt.

5. hostname

Nutzen Sie den Befehl `hostname` nach diesem Schema, um sich den DNS-Namen des Systems anzeigen zu lassen.

```
hostname [OPTIONEN]
```

bash

6. uname

Der Kommandozeilenbefehl `uname` steht für *unix name* und wird verwendet, um Systeminformationen zum Kernel abzurufen. Der Befehl unterstützt diverse Optionen, mit denen sich die Ausgabe nach den gewünschten Informationen filtern lässt. Diese finden Sie im entsprechenden man-Eintrag.

```
uname [OPTIONEN]
```

bash

7. uptime

Möchten Sie ermitteln, wie lange Ihr System seit dem letzten Bootvorgang läuft, nutzen Sie den Kommandozeilenbefehl `uptime` nach folgendem Muster:

```
uptime
```

bash

8. vmstat

Mithilfe des Monitoring-Tools `vmstat` lassen sich Informationen zum virtuellen Speicher, zu Schreib- und Lesevorgängen auf der Festplatte sowie zur CPU-Aktivität abrufen. Rufen Sie `vmstat` nach folgendem Schema auf, um sich die durchschnittlichen Werte seit dem letzten Systemstart ausgeben zu lassen.

```
vmstat [OPTIONEN]
```

bash

Alternativ bietet `vmstat` einen fortlaufenden Monitoring-Modus, bei dem die Systemwerte in einem gewünschten Zeitintervall in Sekunden beliebig oft abgerufen werden.

```
vmstat [OPTIONEN] [INTERVALL [WIEDERHOLUNGEN]]
```

bash

Hardware-Informationen

Linux-Befehle dieser Kategorie liefern Ihnen **detaillierte Informationen zur den Hardware-Komponenten**, die Ihrem System zugrunde liegen.

1. lspci

Nutzen Sie `lspci` (kurz für *list pci*) nach diesem Schema, um sich detaillierte Informationen über PCI-Geräte ausgeben zu lassen.

```
lspci [OPTIONEN]
```

bash

Mögliche Optionen entnehmen Sie den Handbuchseiten Ihres Betriebssystems.

2. lsusb

Nutzen Sie `lsusb` (kurz für *list usb*), um sich detaillierte Informationen zu USB-Geräten im Terminal ausgeben zu lassen.

```
lsusb [OPTIONEN]
```

bash

Mögliche Optionen entnehmen Sie den Handbuchseiten Ihres Betriebssystems.

Prozessmanagement

Unter Linux wird die Instanz eines laufenden Programms als Prozess bezeichnet. Folgende Terminalbefehle gehören zum Standardrepertoire des Prozessmanagements und ermöglichen es Ihnen, alle **Prozesse auf Ihrem System bequem aus dem Terminal heraus zu überwachen** und bei Bedarf steuernd einzugreifen.

1. chrt

Bei `chrt` handelt es sich um ein Kommandozeilenprogramm für die fortgeschrittene Prozesskontrolle, das es ermöglicht, die Echtzeitattribute (Scheduling-Regel und Priorität) laufender Prozesse zu ermitteln und anzupassen oder Befehle und deren Argumente mit bestimmten Echtzeitattributen auszuführen. Die allgemeine Syntax des Befehls lautet:

```
chrt [OPTIONEN] [PRIORITÄT] PID/BEFEHL [ARGUMENTE]
```

bash

Nutzen Sie `chrt` ohne Angabe einer Priorität und mit der Option `-p`, um die Echtzeitattribute ausgewählter Prozesse zu ermitteln.

Darüber hinaus bietet `chrt` die Möglichkeit, die Scheduling-Regel laufender oder neu gestarteter Prozesse mithilfe von Optionen anzupassen oder zu definieren. Weitere Informationen hierzu entnehmen Sie aus dem entsprechenden man-Eintrag.

2. ionice

Der Kommandozeilenbefehl `ionice` wird verwendet, um die Priorität eines Prozesses zu beeinflussen, der das I/O-Interface des Kernels verwendet. Die allgemeine Syntax des Befehls lautet:

```
ionice [OPTIONEN] BEFEHL
```
bash

Um ionice ausführen zu können, werden Root-Rechte benötigt. Der Befehl unterscheidet drei Scheduling-Klassen, die über die Option `-cZAHL` übergeben werden. Mögliche Zahlenwerte sind 1, 2 und 3.

- 1 = Real Time: Die I/O-Aktion wird sofort ausgeführt.
- 2 = Best-Effort: Die I/O-Aktion wird so schnell wie möglich durchgeführt.
- 3 = Idle: Die I/O-Aktion wird nur ausgeführt, wenn kein anderer Prozess I/O-Zeit in Anspruch nimmt.

3. nohup

Normalerweise beenden sich alle abhängigen Prozesse einer Benutzerin bzw. eines Benutzers automatisch, sobald dieser die Terminal-Sitzung schließt (z. B. via `exit`). Der Linux nohup-Befehl (kurz für *no hangup*) löst einen Prozess aus der aktuellen Sitzung und ermöglicht es Ihnen, diesen auch dann weiter laufen zu lassen, wenn Sie sich am System abmelden.

```
nohup BEFEHL
```
bash

4. pidof

Das Kommandozeilenprogramm `pidof` gibt Ihnen die Prozessidentifikationsnummern (PIDs) aller Prozesse eines Programms aus. Ermitteln Sie PIDs via `pidof` nach diesem Muster:

```
pidof [OPTIONEN] PROGRAMM
```
bash

Möchten Sie sich lediglich die erste Prozess-ID ausgeben lassen, verwenden Sie `pidof` in Kombination mit der Option `-s` (kurz für Single shot).

5. pidkill

Wie `kill` sendet auch `pkill` ein Signal an einen ausgewählten Prozess. Die Adressierung erfolgt dabei jedoch nicht durch die PID. Stattdessen wird ein Suchbegriff übergeben, der mit den Namen laufender Prozesse abgeglichen wird. Dieser kann als regulärer Ausdruck formuliert werden. Auch `pkill` übergibt das Standardsignal TERM, sofern kein anderes Signal definiert wurde. Die allgemeine Syntax des Befehls lautet:

```
pkill [OPTIONEN] [-SIGNAL] [SUCHBEGRIFF]
```
bash

Über weitere Optionen lässt sich der Befehl auf die Prozesse eines bestimmten Benutzer-Accounts (`-U UID`), auf Kindprozesse eines bestimmten Elternprozess (`-P PID`) oder die neusten (`-n`) bzw. ältesten Prozesse (`-o`) eingrenzen.

6. pstree

Nutzen Sie `pstree`, um sich alle laufenden Prozesse in einer Baumstruktur anzeigen zu lassen. Die allgemeine Syntax des Befehls lautet:

```
pstree [OPTIONEN]
```
bash

Format und Umfang der Ausgabe lassen sich durch diverse Optionen anpassen.

7. renice

Der Kommandozeilenbefehl `renice` ermöglicht es, die Priorität eines laufenden Prozesses anzupassen. Die allgemeine Syntax lautet:

```
renice PRIORITÄT [OPTIONEN]
```
bash

Die Adressierung erfolgt mithilfe von Optionen über die Prozess-ID (`-p PID`), die Gruppen-ID (`-g GID`) oder einen Benutzernamen (`-u BENUTZER`).

8. sleep

Der Linux sleep-Befehl ermöglicht es Ihnen, die aktuelle Terminal-Sitzung für einen angegebenen Zeitraum zu unterbrechen. Die allgemeine Syntax des Befehls lautet:

```
sleep ZAHL[SUFFIX]
```
bash

Verwenden Sie `sleep` ohne Suffix, wird die angegebene Zahl als Zeitspanne in Sekunden (s) interpretiert. Alternativ haben Sie die Möglichkeiten, die Terminal-Sitzung für Minuten (m), Stunden (h) oder Tage (d) zu unterbrechen.

9. taskset

Bei `taskset` handelt es sich um einen Kommandozeilenbefehl für die erweiterte Prozesskontrolle, der bei Mehrprozessorsystemen zum Einsatz kommt, um Prozesse oder Befehle bestimmten Prozessoren zuzuordnen. Der Befehl erfordert Root-Rechte und verwendet eines der folgenden Schemata:

```
taskset [OPTIONEN] MASKE BEFEHL
taskset [OPTIONEN] -p PID
```
bash

Die Zuordnung von Prozess/Befehl zu Prozessor erfolgt mithilfe einer hexadezimalen Bitmaske. Da eine solche Zuordnung via Bitmaske wenig intuitiv ist, wird taskset in der Regel mit der Option -c (–cpu-list) verwendet, die eine numerische Zuordnung der Prozessoren ermöglicht (z.B. 0, 5 7, 9-11).

Pager

Sie möchten auch bei mehrseitigen Dateiinhalten stets den Überblick behalten? Mit einem Kommandozeilenprogramm aus der Kategorie Pager wählen Sie gezielt aus, welche Abschnitte Sie sich im Terminal anzeigen lassen und blättern bei Bedarf im interaktiven Modus durch die Datei.

1. head

Der Linux head-Befehl wird verwendet, um den ersten Teil einer Datei auszugeben. Die allgemeine Syntax des Befehls lautet:

```
head [OPTIONEN] Datei
```

bash

Nutzen Sie Option `-n ANZAHL_ZEILEN`, um zu definieren, wie viele Zeilen beginnend mit der ersten ausgegeben werden sollen.

2. less

Das Kommandozeilenprogramm `less` ermöglicht es, den Inhalt einer Textdatei im Terminal anzuzeigen. Die allgemeine Syntax lautet:

```
less [OPTIONEN] DATEI
```

bash

Die Ausgabe erfolgt automatisch im interaktiven Modus. Dieser ermöglicht es Ihnen, im ausgewählten Dokument zu blättern oder dieses nach Stichworten zu durchsuchen. Die Taste [Q] beendet den interaktiven Lesemodus. Weitere Steuertasten und mögliche Optionen entnehmen Sie der Handbuchseiten zum Programm.

3. tail

Während `head` standardmäßig die ersten 10 Zeilen einer ausgewählten Datei anzeigt, gibt der Linux tail-Befehl die letzten 10 Zeilen aus. Beide Pager werden nach demselben Schema verwendet (siehe `head`).

Editoren

Unter Linux benötigen Sie kein grafisches Textverarbeitungsprogramm, um Konfigurationsdateien anzupassen, Codeabschnitte zu bearbeiten oder kurze Notizen zu verfassen. **Einfache Texteditoren** lassen sich bequem und ohne Zeitverzögerung im Terminal aufrufen. Im Folgenden stellen wir drei Programme vor, die Sie kennen sollten.

1. emacs

Bei *Emacs* handelt es sich um einen plattformunabhängigen Texteditor, der durch eine Programmierschnittstelle beliebig erweiterbar ist. Emacs startet standardmäßig mit grafischer Benutzeroberfläche, kann mit der Option `--no-window-system` jedoch auch im Terminal geöffnet werden.

```
emacs --no-window-system
```

bash

Emacs verfügt über ein integriertes Tutorial, das Sie durch die Tastenkombination [STRG] + [H], [T] aufrufen.

2. nano

Bei *Nano* handelt es sich um einen terminalbasierten Texteditor. Nano bietet zwar einen geringeren Funktionsumfang als vergleichbare Editoren (z. B. *Vim*), zeichnet sich diesen

gegenüber jedoch durch eine besonders nutzerfreundliche Handhabung aus. Die allgemeine Syntax des Programmaufrufs lautet:

```
nano [OPTIONEN] DATEI
```

bash

Das Programm öffnet die angegebene Datei in einem Edit-Fenster im Terminal. Rufen Sie Nano ohne Dateinamen auf, kann eine neue Textdatei erstellt werden, die im aktuell ausgewählten Verzeichnis abgespeichert wird.

3. vim

Beim Editor Vim (kurz für *Vi Improved*) handelt es sich um eine Weiterentwicklung des Texteditors *Vi*. Vim enthält zahlreiche Erweiterungen wie Syntax-Highlighting, ein umfassendes Hilfesystem, natives Scripting, automatische Codevervollständigung und eine visuelle Textauswahl.

Das Open-Source-Programm bietet Ihnen verschiedene Betriebsmodi zur Bearbeitung reiner Textdateien und kann wahlweise im Terminal oder als Standalone-Applikation mit grafischer Benutzeroberfläche (*GVim*) genutzt werden. Ein zentrales Anwendungsgebiet des Programms ist die Bearbeitung von Programmcode.

Starten Sie Vim in der Konsole, erfolgt die Bedienung über die Tastatur. In der Regel wird das Programm zusammen mit einer Textdatei nach folgendem Muster aufgerufen.

```
vim [OPTIONEN] DATEI
```

bash

Eine umfassende Einführung zu Vim bietet das Programm `vimtutor`, das Sie ebenfalls über die Kommandozeile starten. Darüber hinaus bietet unser Grundlagenartikel zum Linux-Editor Vim weitere Informationen zur Installation und den verschiedenen Betriebsmodi des Programms.

Netzwerkmanagement

Auch das **Netzwerkmanagement** erfolgt unter Linux bequem aus dem Terminal heraus. Egal, ob Sie die Verbindung prüfen, DNS-Informationen abfragen, Schnittstellen konfigurieren oder Dateien an einen anderen Rechner im Netzwerk übermitteln möchten, mit folgenden Programmen genügt ein einzelner Befehl, um Ihr Vorhaben in die Tat umzusetzen.

1. arp

Das Kommandozeilenprogramm `arp` ermöglicht es Ihnen, den ARP-Cache Ihres Betriebssystems abzurufen und zu manipulieren. Nutzen Sie `arp` ohne Modifikator, um sich den Inhalt der ARP-Tabelle im Terminal ausgeben zu lassen.

```
arp [OPTION]
```

bash

Alternativ haben Sie die Möglichkeit, die Ausgabe mit Optionen einzugrenzen, oder Einträge zu erstellen bzw. löschen:

- `-a HOSTNAME` = Ausgabe auf Einträge zu bestimmten Hostnamen (alternativ zu einer IP-Adresse) eingrenzen
- `-s HOSTNAME MAC_ADRESSE` = ARP-Eintrag mit angegeben Hostnamen und MAC-Adresse erstellen
- `-d HOSTNAME` = APR-Eintrag löschen

2. iw

Das Kommandozeilenprogramm `iw` dient der Konfiguration von WLAN-Schnittstellen und hat sich als aktuellere Alternative zu `iwconfig` etabliert. Dem Aufruf liegt eine ähnliche Syntax wie dem `ip`-Befehl zugrunde:

```
iw [OPTIONEN] OBJEKT [BEFEHL]
```

bash

Mögliche Objekte sind:

- dev NAME_DER_SCHNITTSTELLE = Netzwerkschnittstelle
- phy NAME_DES_GERÄTS = WLAN-Gerät (via Name)
- phy#INDEX_DES_GERÄTS = WLAN-Gerät (via Index)
- reg = Regulatory Agent zur Konfiguration von Regions- und Ländereinstellungen

Eine Übersicht der möglichen Befehle sowie Optionen finden Sie im entsprechenden man-Eintrag.

3. nslookup

Wie `dig` dient auch nslookup den Namensauflösungen. Das Kommandozeilenprogramm steht Ihnen in zwei Modi zur Verfügung: interaktiv und nicht-interaktiv. Möchten Sie `nslookup` im nicht-interaktiven Modus verwenden, rufen Sie das Programm in Kombination mit einem Hostnamen oder einer IP-Adresse auf.

```
nslookup [OPTIONEN] [HOST/IP]
```

bash

Den interaktiven Modus starten Sie, indem Sie den Befehl `nslookup` ohne Zusatz ins Terminal eingeben und anschließend Hostnamen bzw. IP-Adressen eingeben, um die damit verknüpfte IP-Adressen bzw. Hostnamen ausgeben zu lassen.

Da das Programm **offiziell veraltet** ist, wird Anwendern empfohlen, stattdessen auf `dig` zurückzugreifen.

4. rsync

Das Kommandozeilenprogramm `rsync` ermöglicht es, Dateien lokal oder über ein Netzwerk zu synchronisieren. Dazu werden zunächst die Größe und die Änderungszeit betreffender Dateien abgeglichen. Die Syntax des Aufrufs lautet:

```
rsync [OPTIONEN] QUELLE(N) ZIEL
```

bash

Der Befehl `rsync` wird in der Regel mit der Option `-a` ausgeführt, die dafür sorgt, dass sämtliche Unterverzeichnisse sowie symbolische Links mitkopiert und sämtliche Benutzerrechte übernommen werden.

5. scp

Mit dem Linux scp-Befehl (kurz für *secure copy*) steht Ihnen direkt über das Terminal ein weiteres Programm zur sicheren Datenübertragung im Netzwerk zur Verfügung: `scp` kopiert Daten von einem Rechner zum anderen und nutzt dabei ebenfalls das Netzwerkprotokoll SSH. Das Clientprogramm funktioniert in etwa so wie die Dateioption `cp`, kommt gemäß folgender Syntax jedoch systemübergreifend zum Einsatz:

```
scp [OPTIONEN] DATEI [[user@]remote_host:]PFAD
```

bash

Bei der Angabe zum Pfad des entfernten Rechners werden der Benutzername und der jeweilige Name des Hosts vorangestellt. Lokale Dateien lassen sich mittels relativer oder absoluter Pfade explizit adressieren.

Beispiel:

```
scp /home/max/images/image.jpg max@example.com:/home/max/archiv
```

bash

Weitere Optionen bieten Ihnen die Möglichkeit, Anpassungen am Übertragungsmodus und an den Verschlüsselungseinstellungen zu vorzunehmen.

6. tty

Der Kommandozeilenbefehl `tty` gibt den Dateinamen des Terminals aus, das als Standard-Input definiert ist. Die allgemeine Syntax des Befehls lautet:

```
tty [OPTIONEN]
```

bash

Archivieren und Komprimieren

Linux bietet diverse Technologien, mit denen sich Dateien in Archive verpacken und komprimieren lassen. Dabei ist zu beachten, dass **nicht jede Archivierung eine Kompression beinhaltet**. So wird `tar`, ein Programm zum Archivieren von Dateien, in der Regel mit Komprimierungsprogrammen wie `gzip`, `bzip2` oder `xz` kombiniert.

1. gzip

Bei `gzip` handelt es sich um ein Programm, mit dem sich Dateien bequem über die Kommandozeile komprimieren und dekomprimieren lassen. Die allgemeine Syntax des Befehls lautet:

```
gzip [OPTIONEN] DATEI(EN)
```

bash

Beachten Sie, dass `gzip` die Originaldatei im Rahmen des Packvorgangs standardmäßig löscht. Unterbinden lässt sich dies mithilfe der Option `-k`. Das Programm kann bei Bedarf auf mehrere Dateien gleichzeitig angewendet werden. Dabei wird jede Ausgangsdatei in eine eigenständige gz-Datei überführt. Möchten Sie mehrere Dateien in ein gemeinsames komprimiertes Archiv schreiben, nutzen Sie `gzip` in Kombination mit dem Archivierungsprogramm `tar`.

Möchten Sie eine gz-Datei dekomprimieren, nutzen Sie den Befehl `gzip` mit der Option `-d`.

2. bzip2

Eine beliebte Alternative zu `gzip` stellt das Kommandozeilenprogramm `bzip2` dar. Dieses verwendet dieselbe Syntax wie `gzip`, beruht jedoch auf einem dreistufigen Komprimierungsprozess, der einen deutlich höheren Kompressionsgrad ermöglicht. Dateien, die mit `bzip2` komprimiert wurden, tragen die Dateiendung `.bz2`. Nutzen Sie `bzip` nach diesem Schema, um Dateien zu komprimieren.

```bash
bzip2 [OPTIONEN] DATEI(EN)
```

Auch `bzip2` lässt sich auf `tar`-Archive anwenden. Die Dekompression erfolgt analog zu `gzip` mithilfe der Option `-d`.

3. xz

Das Kommandozeilenprogramm `xz` überführt Dateien in das gleichnamige Datenkompressionsformat `xz`. Der Programmaufruf erfolgt nach demselben Schema wie bei `gzip` und `bzip2`.

```bash
xz [OPTIONEN] DATEI(EN)
```

Dateien, die mit `xz` komprimiert wurden, tragen die Dateiendung `.xz`. Die Dekomprimierung erfolgt wie bei `gzip` und `bzip` mit der Option `-d`. Alternativ kann der Befehl `unxz` verwendet werden.

Wie gz- und bz2-Dateien sind auch xz-Dateien keine Archivdateien. Möchten Sie mehrere Dateien in dieselbe komprimierte xz-Datei schreiben, müssen Sie auch bei diesem Komprimierungsprogramm auf das Archivierungstool `tar` zurückgreifen.

4. cpio

Bei `cpio` (kurz für *copy in, copy out*) handelt es sich um ein Archivierungsprogramm, mit dem Daten in eine Archivdatei (.cpio) geschrieben und aus dieser extrahiert werden können.

Partitionsmanagement

Möchten Sie unter Linux auf ein Dateisystem auf einer anderen Partition zugreifen, muss dieses zunächst in die Verzeichnisstruktur Ihres Betriebssystems eingebunden werden. Man spricht vom „Mounten“ („Einbinden“ oder „Einhängen“) einer Partition. Dies kann bei Bedarf

über die grafische Benutzeroberfläche erfolgen. Alternativ bieten Kommandozeilenprogramme wie `lsblk`, `blkid` und `mount` die Möglichkeit, Informationen zu angeschlossenen Blockspeichergeräten abzufragen und diese bei Bedarf ein- oder auszuhängen.

1. lsblk

Nutzen Sie den Befehl `lsblk` (kurz für *list block devices*), um sich alle angeschlossenen Blockspeichergeräte und Partitionen als Baumstruktur darstellen zu lassen. Diese müssen nicht zwangsläufig eingebunden sein. Dem Aufruf liegt folgende Syntax zugrunde:

```
lsblk [OPTIONEN]
```

bash

Bei Bedarf lässt sich die Ausgabe mithilfe der Option `-o` (–*output*) und einer Liste der gewünschten Attribute individuell anpassen, um zusätzliche Informationen wie die Identifikationsnummer (UUID), das Dateisystem (FSTYPE) oder den Zustand (STATE) abzufragen.

In der Standardeinstellung werden leere Speichergeräte übersprungen. Möchten Sie auch diese in die Übersicht aufnehmen, verwenden Sie `lsblk` in Kombination mit der Option `-a` (–*all*). Möchten Sie lediglich Informationen zu einem bestimmten Gerät abfragen, verwenden Sie `lsblk` nach folgendem Muster:

2. blkid

Ähnlich wie `lsblk` gibt auch `blkid` Informationen zu angeschlossenen Blockspeichergeräten aus. Verwenden Sie `blkid` nach diesem Schema, um die Identifikationsnummer (`UUID`) und den Dateisystemtyp (`TYPE`) aller angeschlossenen Blockspeichergeräte abzufragen.

```
blkid [OPTIONEN]
```

bash

Eine Tabellarische Ausgabe erhalten Sie mithilfe der Option `-o` in Kombination mit dem Wert `list`. Auch `blkid` lässt sich auf ein ausgewähltes Gerät einschränken:

```
blkid [OPTIONEN] GERÄT
```

bash

Verschiedenes

Nachfolgend finden Sie eine Liste weiterer Standardbefehle unter Linux, die sich keiner der oben aufgeführten Kategorien zuordnen lassen.

1. alias

Die Interaktion mit der Shell erfolgt in der Regel über Befehle, mit denen sich gleichnamige **Kommandozeilenprogramme** aufrufen lassen. Für jede Aktion, die Sie über das Terminal ausführen möchten, verwenden Sie einen Programmaufruf. Der Linux alias-Befehl ermöglicht es Ihnen, Kurznamen für Programmaufrufe zu definieren. Verwenden Sie `alias` nach folgendem Schema:

```bash
alias KURZNAME= 'BEFEHL'
```

Ersetzen Sie den Platzhalter BEFEHL durch einen beliebigen Kommandozeilenbefehl inklusive Optionen. Dieser wird mit der für den Platzhalter KURZNAME eingesetzten Zeichenfolge verknüpft.

2. at

Rufen Sie das Kommandozeilenprogramm `at` nach diesem Schema auf, um einen Befehl zeitgesteuert auszuführen.

```bash
at ZEITANGABE
```

Geben Sie anschließend den Befehl ein und schließen Sie den interaktiven Modus mit [STRG] + [D].

3. cal

Nutzen Sie `cal` nach diesem Muster, um sich einen Kalender im Terminal ausgeben zu lassen.

```bash
cal [OPTIONEN] [[MONAT] Jahr]
```

4. pr

Nutzen Sie das Kommandozeilenprogramm `pr`, um Textdateien zum Drucken aufzubereiten. Die allgemeine Syntax des Befehls lautet:

```bash
pr [OPTIONEN] Datei
```

In der Standardeinstellung erzeugt `pr` einen Seiten-Header, der den Dateinamen, das aktuelle Datum und die Seitennummer enthält.

5. script

Das Kommandozeilenprogramm `script` ermöglicht es Ihnen, eine Terminalsitzung in der Datei `typescript` mitzuschreiben. Findet sich in `typescript` bereits ein Mitschnitt einer vorhergehenden Sitzung, wird dieser überschrieben. Die Aufzeichnung startet automatisch mit dem Aufruf des Programms:

```bash
script
```

Verwenden Sie die Tastenkombination [STRG] + [D], um die Aufzeichnung zu beenden. Möchten Sie die Aufzeichnung statt in `typescript` in einer anderen Datei speichern, rufen Sie `script` in Kombination mit einem Dateinamen oder -pfad auf.

6. seq

Nutzen Sie den Befehl `seq`, um sich eine Zahlenreihe auf die Standardausgabe ausgeben zu lassen. Definieren Sie dabei einen Startwert, einen Endwert und wahlweise ein Inkrement.

```
seq [OPTIONEN] STARTWERT INKREMENT ENDWERT
```

bash

7. tasksel

Das Kommandozeilenprogramm `tasksel` dient als Installationshilfe für Standardanwendungen (Mail-Server, DNS-Server, OpenSSH-Server, LAMP-Server etc.). Nutzen Sie das Tool, um alle für eine Aufgabe benötigten Pakete und Programme automatisch in der richtigen Reihenfolge zu installieren. Rufen Sie `tasksel` mit der Option `--list-tasks` auf, um sich eine Liste aller verfügbaren Standardanwendungen ausgeben zu lassen.

```
tasksel --list-tasks
```

bash

Möchten Sie weitere Informationen zu einer Standardanwendung aus dieser Liste abrufen, verwenden Sie `tasksel` mit der Option `--task-desc` und der entsprechenden Aufgabe. Möchten Sie sich alle Pakete, die zur Aufgabe „mail-server“ gehören, auflisten lassen, nutzen Sie `tasksel` in Kombination mit der Option `--task-packages`.

Eine Installation aller Pakete einer Standardanwendung starten Sie mithilfe des Subbefehls `install`. Es werden Root-Rechte benötigt.

8. tee

Beim Linux tee-Befehl handelt es sich um ein Kommandozeilenprogramm, mit dem sich die Ausgabe eines Programms verdoppeln lässt. Dabei wird eine Ausgabe auf die Standardausgabe weitergeleitet und eine andere in eine mit dem tee-Befehl übergebene Datei geschrieben.

```
tee [OPTIONEN] DATEI
```

bash

In der Regel kommt `tee` in Kombination mit dem Umleitungsoperator Pipe (`|`) zum Einsatz.

```
ls | tee beispiel.txt
```

bash

9. time

Nutzen Sie den Befehl `time` nach folgendem Schema, um die Laufzeit von Programmen zu ermitteln, die Sie über das Terminal starten.

```
time [OPTIONEN] Befehl [ARGUMENTE]
```

bash

10. tr

Nutzen Sie `tr`, um eine beliebige Zeichenmenge zu löschen oder durch eine andere zu ersetzen. Dazu liest `tr` den Datenstrom der Standardeingabe (z. B. eine Datei) ein und schreibt diesen nach der gewünschten Modifikation auf die Standardausgabe. Soll eine Zeichenmenge durch eine andere ersetzt werden, kommt `tr` mit zwei Argumenten zum Einsatz.

```
tr OPTION ZEICHENMENGE1 ZEICHENMENGE2
```

bash

Das zweite Argument (ZEICHENMENGE2) ersetzt das erste (ZEICHENMENGE1). Möchten Sie eine Zeichenfolge löschen, verwenden Sie `tr` mit der Option `-d` und übergeben die zu löschende Sequenz als Argument.

```
tr -d ZEICHENMENGE
```

bash

Das Kommandozeilenprogramm kommt in der Regel in Kombination mit Umleitungsoperatoren (< und >) zum Einsatz, um Anpassungen in Dateien vorzunehmen.

```
tr 'a-z' 'A-Z' < beispiel1.txt > beispiel2.txt
```

bash

`tr` liest den Inhalt der Datei `beispiel1.txt` ein, ersetzt die Kleinbuchstaben a bis z durch Großbuchstaben und schreibt den Output in die Datei `beispiel2.txt`.

11. wall

Das Kommandozeilenprogramm `wall` ermöglicht es Ihnen, allen am System angemeldeten Benutzende eine Nachricht zukommen zu lassen. Um eine Mitteilung zu senden, startet Sie das Programm mit folgendem Aufruf:

```
wall
```

bash

Bestätigen Sie den Programmaufruf mit [Enter] und geben Sie Ihre Nachricht ein. Auch diese wird mit [Enter] bestätigt und durch die Tastenkombination [STRG]+[D] abgeschickt. Alle am System angemeldeten Benutzende erhalten Ihre Nachricht als Broadcast-Message im Terminal. Beachten Sie: Um Mitteilungen erhalten zu können, müssen Sie anderen Nutzenden einen Schreibzugriff für Ihr Terminal gewähren. Nutzen Sie dazu den Befehl `mesg`.

Möchten Sie Dateiinhalte an alle angemeldeten Nutzenden versenden, verwenden Sie `wall` in Kombination mit einer Eingabeumleitung und dem jeweiligen Dateinamen:

```
wall < DATEINAME
```

bash

12. watch

Das Kommandozeilenprogramm `watch` ermöglicht es Ihnen, einen Befehl in regelmäßigen Zeitabständen auszuführen. Dem Aufruf des Programms liegt folgende Syntax zugrunde:

```
watch [OPTIONEN] BEFEHL
```

bash

Das Zeitintervall, in dem der mit `watch` übergebene Befehl periodisch aufgerufen wird, definieren Sie mithilfe der Option `-n SEKUNDEN`. Beenden lässt sich `watch` durch die Tastenkombination [STRG] + [C].

13. wc

Der Linux wc-Befehl (kurz für *word count*) gibt Ihnen bei Bedarf die Anzahl der Zeilen, Wörter, Buchstaben, Zeichen und/oder Bytes einer Textdatei aus. Die allgemeine Syntax des Befehls lautet:

```
wc [OPTIONEN] DATEI
```

bash

Wird `wc` ohne Option aufgerufen, entspricht die Ausgabe dem Schema `ZEILEN WÖRTER ZEICHEN DATEI`. Für eine gefilterte Ausgabe unterstützt das Kommandozeilenprogramm die Optionen: `-l` (Zeilen), `-c` (Bytes), `-m` (Zeichen), `-L` (Länge der längsten Zeile) und `-w` (Wörter).

14. xargs

Mit dem Linux xargs-Befehl können Sie einem Befehl die Ausgabe eines vorhergehenden Befehls als Argument übergeben. In der Regel kommt dabei die Pipe (`|`) als Umleitungsoperator zum Einsatz. Nutzen Sie `xargs` gemäß folgender Syntax:

```
BEFEHL1 | xargs [OPTIONEN] BEFEHL2
```

bash

Zum Einsatz kommt `xargs` beispielsweise in Kombination mit dem Befehl `find`. In folgendem Beispiel ermittelt `find` im aktuellen Verzeichnis alle Dateien, die auf das Suchmuster `\*.tmp` passen, und gibt deren Namen auf die Standardausgabe aus. Dort werden die Dateinamen von `xargs` entgegengenommen und als Argumente an den Befehl `rm` übergeben.

```
find . -name '*.tmp' | xargs rm
```

bash

Die hier dargestellte Übersicht erhebt keinen Anspruch auf Vollständigkeit, sondern umfasst lediglich grundlegende Linux-Befehle mit ausgewählten Anwendungsbeispielen für die alltägliche Arbeit mit unixoiden Betriebssystemen. Eine umfassende Beschreibung der hier aufgeführten Kommandozeilenprogramme sowie aller weiteren Befehle finden Sie in den Handbuchseiten Ihres Betriebssystems. Eine Online-Version dieser Hilfe- und Dokumentationsseiten wird im Rahmen des Linux-Man-Pages-Projects von Michael Kerrisk zur Verfügung gestellt.