

Die 20 wichtigsten Linux-Befehle



Ein sicherer Umgang mit dem **Terminal** ist unter Unix-Systemen eine wichtige Fähigkeit: Viele Aufgaben lassen sich schnell und einfach über die Befehlszeile erledigen. Die 20 wichtigsten Befehle, die jeder Unix-Benutzer kennen sollte, finden Sie hier.



Inhaltsverzeichnis

Anmelden und Abmelden

- login - Anmeldung mit einem Benutzer
- logout - Ausloggen des aktiven Benutzers
- whoami - aktiven Benutzer ausgeben
- su - Benutzer wechseln
- exit - Beenden von Sitzungen

Navigation im Unix-Dateisystem

- pwd - aktuelles Verzeichnis anzeigen
- cd - Verzeichnis wechseln
- ls - Verzeichnisinhalt anzeigen

Manipulation von Dateien und Verzeichnissen

- mkdir - Verzeichnis erstellen
- rmdir - Verzeichnis löschen
- touch - Dateien anlegen und Zugriffsdatum aktualisieren
- rm - Dateien und Verzeichnisse löschen
- mv - Verschieben und Umbenennen
- cp - Kopieren
- nano - Texteditor
- cat - Dateiinhalt ausgeben
- clear - Terminal-Bildschirm leeren

System

- shutdown - System herunterfahren
- systemctl - System Kommandos
- systemd - Steuerung von Units

Anmelden und Abmelden

login - Anmeldung mit einem Benutzer

Login wird verwendet um einen Benutzer anzumelden. Falls derzeit ein anderer Benutzer bereits angemeldet ist, wird dieser dann abgemeldet. Normalerweise wird der Befehl login automatisch als Antwort auf die login:-Eingabeaufforderung im Terminal ausgeführt. Die **Syntax** lautet:

```
login [Optionen] [Benutzer]
```

Nach Ausführen des Befehls folgt eine Eingabeaufforderung für die Passwortheingabe.

logout - Ausloggen des aktiven Benutzers

Mit logout wird der aktive Benutzer wieder abgemeldet.

whoami - aktiven Benutzer ausgeben

Vor allem bei häufigem Wechsel zwischen mehreren Benutzern ist es sehr hilfreich, den aktuellen Benutzer abrufen zu können. Mit dem Befehl whoami kann der mit der aktiven Benutzer-ID (UID) verknüpfte Benutzername ausgegeben werden:

```
whoami
```

su - Benutzer wechseln

Mit su ("switch user") ist es möglich, die Identität eines anderen Benutzers anzunehmen. Das Kommando wird grundsätzlich folgendermaßen angewendet:

```
su [Benutzername]
```

Wird kein Benutzername angegeben, dann wird automatisch der **root-Benutzer** verwendet. Nach Ausführen des Befehls wird das Passwort für den jeweiligen Benutzer abgefragt, nach dessen Eingabe Sie mit der Identität des neuen Benutzers fortfahren können.

Vor allem bei Administrativen Aufgaben, die besondere root-Rechte benötigen, wird der Wechsel auf die Identität root-Benutzer oft gebraucht.

exit - Beenden von Sitzungen

Exit wird verwendet um Sitzungen, wie zum Beispiel Root-Sitzungen, die zuvor per su-Befehl gestartet wurden, oder auch **SSH**-Sitzungen wieder zu beenden. Exit kann auch verwendet werden, um den aktiven Benutzer abzumelden - ähnlich wie der logout-Befehl.

```
exit
```



Navigation im Unix-Dateisystem

pwd - aktuelles Verzeichnis anzeigen

Mit dem Befehl pwd ("print working directory") lässt sich der Pfad für das Verzeichnis ausgeben, in dem sich der an dem Terminal angemeldete Nutzer gerade befindet.

```
pwd
```

Beispiel

Eingabe:

```
pwd
```

Ausgabe:

```
/home/hedv
```

cd - Verzeichnis wechseln

Mit cd ("change directory") kann in ein anderes Verzeichnis gewechselt werden. Für die allgemeine Syntax gilt:

```
cd [Verzeichnis]
```

Um beispielsweise in das Verzeichnis `/var/www/html/` zu wechseln, wird folgendes angegeben:

```
cd /var/www/html/
```

Falls der Verzeichnisname Leerzeichen enthält, so muss dieser durch Anführungszeichen eingefasst werden.

Um mit cd in das übergeordnete Verzeichnis zu wechseln, also beispielsweise von `/var/www/html` nach `/var/www/`, wird `..` angegeben:

```
cd ..
```

Wird kein Verzeichnis angegeben, so wechselt cd in das Home-Verzeichnis des jeweiligen Benutzers. Bei dem root-Benutzer ist dieses Verzeichnis das Wurzelverzeichnis.

ls - Verzeichnisinhalt anzeigen

Mit ls ("list") lässt sich der Inhalt eines Verzeichnisses ausgeben. Die Syntax lautet:

```
ls [Optionen] [Verzeichnis]
```

Zwei äußerst wichtige Parameter sind dabei "-l", "-h" und "-a". "-l" gibt die Langform der Datei-Informationen aus, dazu gehören beispielsweise Berechtigungen, die Dateigröße in Bytes und das Änderungsdatum:

```
ls -l [Verzeichnis]
```

Durch Hinzufügen des Parameters "-h" ("human readable") lässt sich die Dateigröße auch in besser lesbarer Form, also in KB, MB, GB etc. ausgeben:

```
ls -lh [Verzeichnis]
```

Der Parameter "-a" listet auch versteckte Dateien:

```
ls -la [Verzeichnis]
```

Wenn kein Verzeichnis angegeben wird, dann wird das aktuelle Verzeichnis verwendet.

Beispiel

Eingabe:

```
ls -la /home/hellberg/daten/
```

Ausgabe:

```
drwxr-xr-x 5 nobody nogroup 4096 Jan 23 13:45 .
-rw-r--r-- 5 nobody nogroup 4096 Jan 23 13:45 ..
-rw-r--r-- 1 hellberg hellberg 4096 Dez 1 18:46 20-linux-befehle.txt
drwxr-xr-x 1 nobody nogroup 73534 Jan 23 13:45 edv-beratung
-rw-r--r-- 1 hellberg hellberg 4096 Dez 1 18:29 hedv.png
```

Manipulation von Dateien und Verzeichnissen

mkdir - Verzeichnis erstellen

Mithilfe von mkdir ("make directory") werden Verzeichnisse erstellt, unter denen weitere Dateien und Verzeichnisse angelegt werden können:

```
mkdir [Optionen] [Pfad]
```

Wenn ein Verzeichnis über mehrere andere Verzeichnisse hinweg angelegt werden soll, die noch nicht existieren, dann muss die Option "-p" übergeben werden. Die fehlenden Verzeichnisse werden dann bei der Ausführung ebenfalls angelegt.

```
mkdir -p [Pfad]
```



rmdir - Verzeichnis löschen

rmdir wird verwendet um Verzeichnisse zu löschen. Das zu löschende Verzeichnis muss dabei jedoch leer sein.

```
rmdir [Optionen] [Pfad]
```

Mit der Option "-p" kann ein Verzeichnis einschließlich aller seiner Parent-Directories gelöscht werden, solange diese ebenfalls leer sind.

```
rmdir -p [Pfad]
```

touch - Dateien anlegen und Zugriffsdatum aktualisieren

Mit dem Befehl touch können neue Dateien mit aktuellen Daten, wie beispielsweise der Zugriffszeit, angelegt werden. Die Syntax von touch lautet:

```
touch [Optionen] [Dateipfad]
```

Eine neue Datei der Länge null wird jedoch lediglich erzeugt, wenn die genannte Datei noch nicht existiert. Deshalb wird touch auch verwendet, wenn das Zugriffsdatum von existierenden Dateien aktualisiert werden soll.

rm - Dateien und Verzeichnisse löschen

Der Befehl rm löscht den Eintrag von einer oder mehrerer Dateien aus einem Verzeichnis.

```
rm [Optionen] [Pfad]
```

Um **rekursiv** auch ganze Verzeichnisse mitsamt enthaltener Dateien und Unterverzeichnisse zu löschen wird die Option "-r" verwendet:

```
rm -r [Pfad]
```

mv - Verschieben und Umbenennen

Mit mv ("move") lassen sich Dateien und Verzeichnisse verschieben und umbenennen. Verschieben lässt sich eine Datei / ein Verzeichnis durch Angabe des alten und neuen Pfades:

```
mv [alter Pfad] [neuer Pfad]
```

Umbenennen lassen sich Dateien und Verzeichnisse auf ähnliche Weise. Der alte- sowie der neue Pfad bleiben dabei jedoch gleich. Lediglich der Name des Verzeichnisses bzw. der Datei wird verändert.

cp - Kopieren

Mit cp kann eine Datei in eine neue Datei mit anderem Namen beziehungsweise Pfad kopiert werden. Außerdem können mehrere Dateien in ein angegebenes Verzeichnis kopiert werden:

```
cp [Quelle] [Ziel]
```

nano - Texteditor

Mit nano kann ein einfacher Texteditor aufgerufen werden. Durch Angabe eines Dateipfades kann die jeweilige genannte Datei mit dem Texteditor bearbeitet werden. Wird kein Pfad angegeben bzw. existiert die angegebene Datei nicht, so können auch neue Dateien erstellt und bearbeitet werden.

```
nano [Dateipfad]
```

cat - Dateiinhalte ausgeben

Mit dem Befehl cat kann der Inhalt beliebiger Dateien direkt im Terminal ausgegeben werden. Der Pfad zur jeweiligen Datei wird dazu einfach direkt hinter dem Befehl angegeben:

```
cat [Dateipfad]
```

Der Befehl eignet sich hervorragend dazu, den Inhalt einer Datei nach Abschluss einer Änderung noch einmal zu überprüfen, da der Dateiinhalt durch cat selbst nicht modifiziert wird.

clear - Terminal-Bildschirm leeren

clear wird verwendet, um den Terminal-Bildschirm, sowie den Scrollback Buffer nach Eingabe von Befehlen wieder zu leeren. Der Befehl schreibt dazu in den Standard Output (stdout). Wenn der Standard Output in eine Datei umgeleitet wird, dann wird der Bildschirm erst nach Ausgabe des Inhalts mit beispielsweise cat geleert.

```
clear
```

Wenn Sie mehr über den Standard Output, sowie Ein- und Ausgabe-Umleitungen erfahren wollen, dann lesen Sie gerne auch [unseren dazugehörigen Beitrag!](#)

System

shutdown - System herunterfahren

Der Befehl shutdown ermöglicht das Herunterfahren des Systems über die Kommandozeile. Der Vorgang kann durch bestimmte Optionen zeitlich gesteuert werden.

Mit der Option "-h now" (halt) wird das System sofort heruntergefahren:

```
shutdown -h now
```

Alternativ können auch beliebige Angaben in Minuten gemacht werden. Durch den folgenden Befehl wird das System nach 10 Minuten automatisch heruntergefahren:

```
shutdown -h 10
```

Schließlich kann ein zeitlich geplantes Herunterfahren auch wieder abgebrochen werden. Dazu wird die Option -c (cancel) verwendet:

```
shutdown -c
```

UPDATE: Aufgrund des vollständigen Wechsels auf **systemd** kann der Befehl shutdown unter Debian 10 Buster nicht mehr ohne Weiteres eingesetzt werden. Stattdessen kann auf systemctl zurückgegriffen werden. Dieser Befehl wird im Folgenden näher behandelt.

systemctl - System Kommandos

Das Kommandozeilenwerkzeug systemctl kann dazu eingesetzt werden, um Befehle an systemd zu senden. So kann beispielsweise das System Heruntergefahren, oder ein Neustart durchgeführt werden. Herunterfahren lässt sich das System, ähnlich wie mit shutdown, mit dem Kommando "poweroff":

```
systemctl poweroff
```

Ein Neustart kann mit dem Kommando "reboot" durchgeführt werden:

```
systemctl reboot
```

systemctl - Steuerung von Units

Systemctl kann auch für die Steuerung von Systemprogrammen, sogenannten Units, verwendet werden. Die wichtigsten Kommandos sind dabei "start", "stop", "restart" und "status". Mit "start" und "stop" können Units gestartet und gestoppt werden:

```
systemctl [start / stop] [Unit]
```

Das Kommando "restart" kann verwendet werden, um eine Unit neu zu starten:

```
systemctl restart [ Unit ]
```

Mit "status" kann der Status einer Unit abgefragt werden - beispielsweise "active" oder "inactive":

```
systemctl status [ Unit ]
```

