

Dateiverwaltung

ls [Option(en)] [Datei(en)]

Wird **ls** ohne weitere Angaben und Parameter aufgerufen, listet es den Inhalt des Verzeichnisses, in dem Sie sich gerade befinden, in Kurzform auf.

- l detaillierte Liste,
- a zeigt versteckte Dateien an.

cp [Option(en)] Quelldatei Zieldatei

Erstellt eine Kopie der *Quelldatei* in *Zieldatei*.

- i Wartet ggf. auf Bestätigung, bevor eine existierende *Zieldatei* überschrieben wird.
- r Kopiert rekursiv (mit Unterverzeichnissen)

mv [Option(en)] Quelldatei Zieldatei

Legt eine Kopie der *Quelldatei* in *Zieldatei* an und löscht anschließend die ursprüngliche Datei.

- b Erstellt vor dem Verschieben eine Sicherungskopie der *Quelldatei*
- i Wartet ggf. auf Bestätigung, bevor eine existierende *Zieldatei* überschrieben wird.

rm [Option(en)] Datei(en)

Entfernt die angegebenen *Datei(en)* aus dem Dateisystem. Verzeichnisse werden außer durch explizite Angabe des Parameters **-r** nicht durch **rm** gelöscht.

- r Löscht auch evtl. vorhandene Unterverzeichnisse
- i Wartet vor Löschen jeder angegebenen Datei auf Bestätigung

ln [Option(en)] Quelldatei Zieldatei

Legt in *Zieldatei* einen internen Querverweis („Link“) auf die *Quelldatei* unter anderem Namen an. Normalerweise zeigt dieser Verweis direkt auf die Position der *Quelldatei* innerhalb eines Dateisystems. Wird **ln** hingegen mit der Option **-s** aufgerufen, wird ein so genannter symbolischer Link angelegt, der lediglich auf den Pfad der *Quelldatei* zeigt und deswegen auch über Dateisystemgrenzen hinweg funktioniert.

- s Legt einen symbolischen Link an.

cd [Option(en)] **Verzeichnis**

Wechselt das aktuelle Verzeichnis. Wenn nur **cd** eingegeben wird, wird in das Homeverzeichnis gesprungen.

mkdir [Option(en)] Verzeichnisname

Legt ein neues Verzeichnis an.

rmdir [Option(en)] Verzeichnisname

Löscht das Verzeichnis, allerdings nur, wenn dieses bereits leer ist.

chown [Option(en)] Benutzername.Gruppe **Datei(en)**

Ändert den Besitzer einer Datei auf den angegebenen *Benutzernamen*.

- R Ändern der Dateien und Verzeichnisse in allen Unterverzeichnissen.

chgrp [Option(en)] Gruppenname *Datei(en)*

Ändert den Namen der Gruppe, die eine gegebene *Datei* besitzt auf *Gruppenname*. Der *Datei*-Besitzer darf diesen Wert nur ändern, wenn er sowohl Mitglied in der bisherigen wie auch der neuen Eigentümergruppe ist.

chmod [Optionen] *modus* *Datei(en)*

Ändert Zugriffsrechte. Der Parameter `modus` ist dreiteilig: Gruppe, Zugriff und Zugriffstyp. Für Gruppe sind die Zeichen **u** für den Benutzer User, **g** für die Gruppe group, **o** für alle anderen others. erlaubt. Für Zugriff sind die Zeichen **+** und **-** möglich. Durch das Zeichen **+** kann der Zugriff erlaubt, durch **-** entzogen werden.

Der Zugriffstyp wird durch folgende Optionen gesteuert:

- r** für Lesen read,
- w** für Schreiben write,
- x** Ausführen von Dateien, bzw. Wechseln in das Verzeichnis eXecute.
- s** Setuid-Bit; das Programm wird ausgeführt, als ob es vom Inhaber der Datei gestartet würde.

Alternativ benutzen Sie den Zahlencode. Die vier Stellen dieses Codes setzen sich jeweils aus der Summe der Werte 4, 2, und 1. An der ersten Stelle setzen Sie die set user ID (4), die set group ID (2) und die sticky-Attribute (1). An zweiter Stelle bestimmen Sie die Rechte des Datei-Eigentümers, an dritter Stelle die der Gruppenmitglieder und der letzte Wert setzt die Berechtigungen für alle anderen Benutzer. Das Leserecht wird mit 4, das Schreibrecht mit 2 und das Recht, eine Datei auszuführen, mit 1 kodiert. Der Eigentümer einer Datei wird in der Regel eine 7 bekommen, also die Summer aller Rechte.

gzip [Parameter] Datei(en)

Dieses Programm verkleinert (komprimiert) den Inhalt von Dateien durch komplizierte mathematische Verfahren. Die Namen der reduzierten Dateien enden dann auf `.gz` und müssen vor erneuter Benutzung wieder entpackt werden. Wollen Sie mehrere Dateien oder ganze Verzeichnisse komprimieren, müssen Sie zusätzlich den Befehl `tar` verwenden.

-d Dekomprimiert die gepackten gzip-Dateien, sodass diese ihre ursprüngliche Größe wiedererlangen und normal bearbeitet werden können (entspricht dem Aufruf von `gunzip`).

tar Optionen Archivname Datei(en)

`tar` fasst eine oder (in der Regel) mehrere Dateien, die dann zum Beispiel komprimiert werden können, zu einem so genannten Archiv zusammen. `tar` ist ein sehr komplexer Befehl, der eine Vielzahl von Optionen zur Verfügung stellt. Die gebräuchlichsten stellen wir hier kurz vor.

-f Schreibt die Ausgabe in eine Datei und nicht auf den Bildschirm, wie es standardmäßig vorgesehen ist `file`.

- c** Legt ein neues tar-Archiv an create.
- r** Fügt Dateien einem bestehenden Archiv hinzu.
- t** Gibt den Inhalt eines Archives aus.
- u** Fügt Dateien hinzu, allerdings nur, wenn diese neuer sind als die im Archiv bereits enthaltenen
- x** Packt Dateien aus einem Archiv aus (extrahiert)
- z** Komprimiert das entstandene Archiv mit `gzip`.
- j** Komprimiert das entstandene Archiv mit `bzip2`.
- v** Gibt die Namen aller bearbeiteten Dateien aus.

Die von `tar` erstellten Archivdateien enden mit `.tar`. Wenn das Tar-Archiv noch durch `gzip` komprimiert wurde ist die Endung `.tar.gz`, bei `bzip2` `.tar.bz2`.

locate Muster

Mit `locate` kann man herausfinden, in welchem Verzeichnis sich eine spezifizierte Datei befindet. Zusätzlich können dabei auch Jokerzeichen verwendet werden. Das Programm arbeitet sehr schnell, da es nicht langsam im Dateisystem selbst, sondern in einer eigens dafür erstellten Datenbank sucht. Dies ist auch das Hauptproblem dieses sehr flotten Kommandos, weil darin natürlich keine Dateien gelistet sein können, die nach der letzten Aktualisierung dieser Datenbank erstellt wurden. Unter `root` kann die Datenbank mit `updatedb` erstellt werden.

updatedb [Option(en)]

Mit diesem Befehl ist es möglich, auf einfache Weise die von `locate` benötigte Datenbank auf den aktuellsten Stand zu bringen. Damit möglichst alle Dateien erfasst werden, sollte das Programm als `root` aufgerufen werden. Auch bietet sich an, es durch einen angehängten Ampersand (&) in den Hintergrund zu versetzen, damit gleich weitergearbeitet werden kann (`updatedb &`).

find [Option(en)]

Mit dem Befehl `find` können Sie in einem bestimmten Verzeichnis nach einer Datei suchen. Das erste Argument bezeichnet dabei das Verzeichnis, von dem aus die Suche gestartet werden soll. Die Option `-name` verlangt einen zu suchenden String, in dem auch Wildcards erlaubt sind. `find` sucht im Gegensatz zu `locate` nicht in einer eigenen Datenbank nach Dateien, sondern durchsucht tatsächlich das angegebene Verzeichnis.

Inhaltsbefehle

cat [Option(en)] Datei(en)

gibt den Inhalt einer angegebenen Datei ohne Unterbrechung aus.
`-n` Nummeriert die Ausgabe am linken Rand.

less [Option(en)] Datei(en)

Dieser Befehl ermöglicht es, durch den Inhalt der spezifizierten Datei zu wandern. Beispielsweise können Sie mit [Bild auf] und [Bild ab] um je eine halbe Bildschirmseite vor bzw. zurück springen, mit der Leertaste gar eine ganze nach vorne. Mit `Q` kann dieser Ausgabemodus beendet werden.

grep [Option(en)] Suchwort Datei(en)

`grep` ist dazu gedacht, um ein bestimmtes Suchwort in den angegebenen Datei(en) zu finden. Hat es Erfolg, gibt es die Zeile, in der das Suchwort gefunden wurde, sowie den Namen der Datei aus.

- `-i` Ignoriert Groß-/Kleinschreibung
- `-l` Gibt nur die Namen der jeweiligen Dateien, nicht aber die Textzeilen aus
- `-n` Zeigt zusätzlich Nummern der Zeilen, in denen es fündig wird, an
- `-l` Listet nur Dateien, in denen das Suchwort nicht vorkommt

diff [Option(en)] Datei1 Datei2

`diff` wurde geschaffen, um den Inhalt zweier beliebig gewählter Dateien zu vergleichen und in Form einer Liste von geänderten Zeilen auszugeben. Häufig wird es von Programmierern verwendet, die auf diese Weise nur die Änderungen in Ihren Programmen, nicht aber die gesamten Quelltexte verschicken müssen.

- `-q` Meldet nur, ob sich die beiden angegebenen Dateien überhaupt in ihrem Inhalt unterscheiden.

Dateisysteme

mount [Option(en)] [Gerät] Mountpoint

Mit Hilfe dieses Befehls können beliebige Datenträger in das Dateisystem gemountet werden. Darunter versteht man das Einbinden von Festplatten-, CD-ROM- und anderen Laufwerken in ein Verzeichnis des Linux-Dateisystems.

- `-r` nur lesbar mounten read only.
- `-t dateisystem` Gibt das Dateisystem an. Die gebräuchlichsten sind: `ext2` für Linux-Festplatten, `msdos` für MS-DOS-Medien, `vfat` für das Windows-Dateisystem und `iso9660` für CDs.

Bei Laufwerken, die nicht in der Datei `/etc/fstab` definiert sind, muss auch der Typ des Gerätes angegeben werden. Das Einhängen `mount` kann in diesem Fall nur von `root` vorgenommen werden.

Falls das Dateisystem auch von anderen Benutzern gemountet werden soll, tragen Sie in der entsprechende Zeile der Datei `/etc/fstab` die Option `user` (durch Kommata getrennt) ein und speichern Sie diese. Weitere Informationen entnehmen Sie bitte der `mount`.

umount [Option(en)] Mountpoint

Dieser Befehl entfernt ein gemountetes Laufwerk aus dem Dateisystem. Bevor Sie einen Datenträger aus dem Laufwerk entfernen, rufen Sie bitte diesen Befehl auf. Ansonsten besteht die Gefahr eines Datenverlustes! Sowohl `mount` als auch `umount` können im Regelfall nur von `root` ausgeführt werden. Ausnahme: In der Datei `/etc/fstab` wird für das Laufwerk die Option `user` angegeben.

Information

df [Option(en)] [Verzeichnis]

Der Befehl `df` `disk free`, ohne Optionen aufgerufen, zeigt eine Statistik über den gesamten, den belegten und den verfügbaren Speicherplatz von allen gemounteten Laufwerken an. Wird hingegen ein Verzeichnis angegeben, so wird das Laufwerk, auf dem sich dieses befindet, in der Statistik angezeigt.

-H zeigt die Anzahl der belegten Blöcke in Gigabyte, Megabyte oder Kilobyte an – in menschenlesbarer Form `human readable`.

-t Typ des Laufwerks (`ext2`, `nfs` usw.).

du [Option(en)] [Pfad]

Dieses Kommando ohne Parameter gibt den Gesamtspeicherplatzverbrauch aller im aktuellen Verzeichnis enthaltenen Dateien an; sind Unterverzeichnisse vorhanden, so auch deren Gesamtgröße.

-a Gibt die Größe jeder einzelnen Datei an

-h Ausgabe in „menschenlesbarer“ Form

-s Zeigt nur die errechnete Gesamtgröße an

free [Option(en)]

`Free` zeigt die Summe des gesamten und des genutzten Arbeits- bzw. Swap-Speichers an.

-b Anzeige in Bytes,

-k Anzeige in Kilobytes,

-m Anzeige in Megabytes

date [Option(en)]

Dieses kleine Programm gibt bei Aufruf die aktuelle Systemuhrzeit aus. Darüber hinaus ist es möglich, die Systemzeit als `root` durch diesen Befehl zu verändern. Details hierzu sind nachzulesen in (`date`).

Prozesse

top [Option(en)]

Durch `top` erhält man einen schnellen Überblick über die gegenwärtig laufenden Prozesse. Durch Drücken der Taste `[h]` wird eine Seite mit Erklärungen der wichtigsten Optionen angezeigt, um das Programm seinen individuellen Bedürfnissen anzupassen.

ps [Option(en)] [Prozess-ID]

Ohne Optionen aufgerufen, liefert dieses Kommando eine Tabelle aller eigenen, also von einem selbst gestarteten, Programme bzw. Prozesse zurück. Achtung, bei den Optionen zu diesem Befehl sollte kein Bindestrich vorangestellt werden!

aux Listet detailliert alle Prozesse unabhängig vom Besitzer

kill [Option(en)] Prozess-ID

Manchmal kommt es leider vor, dass sich Programme nicht mehr auf normalem Wege beenden lassen. Mit dem `kill`-Befehl lassen sich nahezu alle Prozess-Leichen anhand Ihrer Prozess-ID (siehe `top` bzw.

`ps`) töten. Dazu sendet es ein so genanntes TERM-Signal, welches das Programm auffordert, sich selbst zu beenden. Hilft dies nicht weiter, gibt es noch einen nützlichen Parameter:

-9 Sendet anstatt eines TERM- ein KILL-Signal, wodurch der Prozess vom Betriebssystem entfernt wird. Dies macht in nahezu allen Fällen den spezifizierten Prozessen den Garaus.

killall [Option(en)] Prozessname

Dieser Befehl funktioniert äquivalent zu `kill`, wobei statt einer Prozess-ID die Angabe des Prozess-Namens ausreicht, um alle sich so nennenden Prozesse zu töten.

Netzwerk

ping [Option(en)] Rechnername|IP-Adresse

`ping` ist das Mittel schlechthin, um TCP-IP-Netzwerke auf ihre grundsätzliche Funktionstüchtigkeit zu überprüfen. Das Tool sendet dabei ein kleines Datenpaket an einen anderen Rechner mit der Aufforderung, dieses sofort wieder zurückzusenden. Klappt dies, zeigt `ping` eine entsprechende Meldung an, wodurch die grundsätzliche Übertragungsfähigkeit des Netzes sichergestellt wurde.

-c Anzahl: Legt die Gesamtzahl der zu versendenden Pakete fest, nach deren Versand sich das Programm beendet. Standardmäßig existiert keine Beschränkung.

-f Flood ping: Sendet so viele Datenpakete wie möglich. Nur von `root` durchführbarer Test, um Netzwerke zu testen.

-i Wert: Legt das Intervall zwischen zwei Datenpaketen in Sekunden fest; standardmäßig eine Sekunde.

nslookup

Zur Umwandlung von Domainnamen in IP-Adressen existiert das so genannte Domain Name System. Mit diesem Tool lassen sich Anfragen an entsprechende Auskunftsdienste (DNS-Server) stellen.

telnet [Option(en)] Rechnername oder IP-Adresse

Telnet ist eigentlich ein Internet-Protokoll, das die Arbeit auf anderen Rechnern (`remote`) über ein Netzwerk ermöglicht. Telnet heißt auch ein Linux-Programm, das genau dieses Protokoll umsetzt und ein Arbeiten an anderen Rechnern ermöglicht, ohne direkt an dem Gerät zu sitzen.

Achtung Benutzen Sie Telnet nicht über Netze, die von Dritten mitgehört werden könnten. Vor allem im Internet sollten verschlüsselte Übertragungsmethoden wie `ssh` benutzt werden, um die Gefahr eines Passwortmissbrauchs auszuschließen (siehe `ssh`).

Sonstiges

passwd [Option(en)] [Benutzername]

Mit diesem Kommando hat jeder Benutzer jederzeit die Möglichkeit, sein eigenes Passwort zu ändern. Der Administrator `root` darf darüber hinaus das Codewort eines jeden Benutzers ändern.

su [Option(en)] [Benutzername]

Durch `su` ist es möglich, das Benutzer-Login während einer Sitzung zu wechseln. Es wird sofort nach dem `root`-Passwort verlangt. Unter Angabe eines Benutzernamens und anschließender Eingabe des zugehörigen Passwortes kann dessen Umgebung genutzt werden. Als `root` muss dieses Passwort nicht eingegeben werden, da mit Administratorrechten die Identität eines jeden Benutzers problemlos angenommen werden kann.

halt [Option(en)]

Um keine Datenverluste zu riskieren, sollten Sie Ihren Rechner immer mit diesem Programm herunterfahren.

reboot [Option(en)]

Funktioniert wie der `halt`-Befehl, allerdings mit sofortigem Neustart.

clear

Wenn Sie den Überblick über die Textzeilen der Konsole verlieren, können Sie die sichtbare Anzeige löschen. Der Befehl hat keine Optionen.

Der Editor vi

Die Bedienung des `vi` ist etwas gewöhnungsbedürftig. Er wird an dieser Stelle anderen Editoren vorgezogen, weil er zum einen auf jedem UNIX-ähnlichen Betriebssystem zur Verfügung steht und bei Linux zum standardmäßigen Installationsumfang gehört; zum anderen, weil seine Bedienung eindeutig ist und dadurch in der Regel keine Missverständnisse auftreten. Außerdem: wenn nichts geht, geht `vi` immer noch. Die nun folgende Kurzanleitung sollte Sie in die Lage versetzen, mit Hilfe des `vi` zum Beispiel diverse Konfigurationsdateien zu editieren.

Der `vi` kennt drei Betriebsarten (Modi):

Im Befehlsmodus (engl. command mode) wird jeder Tastendruck als Teil eines Befehls interpretiert.

Im Einfügemodus (engl. insert mode) werden Tastatureingaben als Text interpretiert.

Im Komplexbefehlsmodus (engl. last line mode) geben Sie komplexere Befehle in der letzten Zeile ein.

Die wichtigsten Befehle des Befehlsmodus sind:

`ESC` wechselt in den komplexen Befehlsmodus.

`i` wechselt in den Eingabemodus (Zeichen werden an der aktuellen Cursorposition eingegeben).

`a` wechselt in den Eingabemodus (Zeichen werden nach der aktuellen Cursorposition

eingegeben).

`A` wechselt in den Eingabemodus (Zeichen werden am Ende der Zeile angehängt).

`R` wechselt in den Eingabemodus (überschreibt den alten Text).

`r` wechselt zum Überschreiben eines einzelnen Zeichens in den Eingabemodus.

`s` wechselt in den Eingabemodus (das Zeichen, auf dem der Cursor steht, wird durch die Eingabe überschrieben).

`C` wechselt in den Eingabemodus (der Rest der Zeile wird durch den neuen Text ersetzt).

`o` wechselt in den Eingabemodus (nach der aktuellen Zeile wird eine neue Zeile eingefügt).

`O` wechselt in den Eingabemodus (vor der aktuellen Zeile wird eine neue Zeile eingefügt).

`x` löscht das aktuelle Zeichen.

`dd` löscht die aktuelle Zeile.

`dw` löscht bis zum Ende des aktuellen Worts.

`cw` wechselt in den Eingabemodus (der Rest des aktuellen Worts wird durch die Eingabe überschrieben).

`u` nimmt den letzten Befehl zurück.

`J` hängt die folgende Zeile an die aktuelle an.

`.` wiederholt den letzten Befehl.

`:` wechselt in den Komplexbefehlsmodus.

Allen Befehlen kann eine Zahl vorangestellt werden, die angibt, auf wie viele Objekte sich der folgende Befehl beziehen soll. So können durch Eingabe von `3dw` drei Wörter auf einmal gelöscht werden. Durch Eingabe von `10x` erreicht man das Löschen von zehn Zeichen ab der Cursorposition, `20dd` löscht 20 Zeilen.

Die wichtigsten Befehle des Komplexbefehlsmodus:

`:q!` verlässt `vi`, ohne Änderungen zu speichern

`:w dateiname` speichert unter `dateiname`

`:x` speichert die geänderte Datei und verlässt den Editor

`:e dateiname` editiert (lädt) `dateiname`

`:u` nimmt den letzten Editierbefehl zurück