

Kompendium der Linux+ (XK0-005) Befehle

Dieses Nachschlagewerk umfasst alle wichtigen Befehle aus dem offiziellen **Command Reference Guide** für die CompTIA Linux+ Prüfung (XK0-005). Die Befehle sind thematisch gegliedert (Benutzerverwaltung, Netzwerk, Prozesse, Dateisystem, Paketmanagement, Container, Sicherheit etc.), mit Beschreibung, vollständiger Syntax, allen verfügbaren Optionen und ggf. wichtigen Unterbefehlen. Es dient als umfassendes Cheat-Sheet zur Prüfungsvorbereitung.

Grundlegende Linux-Befehle

ls (List Directory Contents)

Beschreibung: Listet den Inhalt eines Verzeichnisses auf. Ohne Parameter zeigt ls die nicht versteckten Einträge im aktuellen Verzeichnis in alphabetischer Reihenfolgewiki.ubuntuusers.deionos.de.

Syntax:

ls [OPTION]... [VERZEICHNIS]

Wenn kein Verzeichnis angegeben wird, wird das aktuelle Verzeichnis aufgelistetwiki.ubuntuusers.de.

Optionen:

Option	Beschreibung
-a, --all	Zeigt alle Einträge an, inklusive versteckter Dateien (auch . und ..) wiki.ubuntuusers.deionos.de .
-A, --almost-all	Wie -a, aber ohne . und .. – listet versteckte Dateien außer den Verzeichnissen ./.. wiki.ubuntuusers.deionos.de .
-l	Langformat: Detaillierte Ausgabe (Rechte, Besitzer, Größe, Datum usw.) je Eintrag wiki.ubuntuusers.deionos.de .
-d, --directory	Listet Verzeichniseintrag selbst statt dessen Inhalt (nützlich, um Infos über Ordner zu sehen) wiki.ubuntuusers.deionos.de .
-h, --human-readable	Im Langformat (-l) Größen in menschenlesbare Einheiten (KB, MB ...) anzeigen wiki.ubuntuusers.deionos.de .
-i	Zeigt die Inode-Nummer jedes Eintrags an wiki.ubuntuusers.deionos.de .
-C	Ausgabe in Spalten nebeneinander (Standardformat) wiki.ubuntuusers.deionos.de .

Option	Beschreibung
-m	Listet Einträge kommasepariert in einer Zeile wiki.ubuntuusers.deionos.at .
-R, --recursive	Rekursiv: Zeigt auch Inhalte von Unterverzeichnissen (durchläuft Verzeichnisbaum) wiki.ubuntuusers.deionos.at .
-t	Sortiert nach Zeitstempel (Änderungszeit, neueste zuerst) anstatt alphabetisch.
-r	Kehrt die Sortierreihenfolge um (Reverse order).
--hyperlink	Macht Dateinamen (in unterstützten Terminals) als klickbare Links ausgeben wiki.ubuntuusers.de .
--help	Zeigt die Hilfe zum Befehl an.
--version	Zeigt Versionsinformationen von ls.

(Weitere Optionen sind in den Manpages dokumentiertwiki.ubuntuusers.de.)

cat (Concatenate and Print Files)

Beschreibung: Gibt den Inhalt von Dateien auf der Standardausgabe aus oder verknüpft (konkatiniert) mehrere Dateien. Oft verwendet, um Textdateien anzuzeigen oder Inhalte zu pipenionos.at. Der Name steht für „concatenate“ (verketten).

Syntax:

`cat [OPTION]... [DATEI]...`

Ohne Optionen schreibt cat den Inhalt der angegebenen Datei(en) in der angegebenen Reihenfolge auf die Standardausgabe. Kann mehrere Dateien hintereinander ausgeben oder mit Umleitungen/ Pipes genutzt werden.

Optionen:

Option Beschreibung

-n	Nummeriert alle ausgegebenen Zeilen fortlaufend ibm.comionos.at .
-b	Nummeriert nur nicht-leere Ausgabelinien (Leerzeilen werden nicht nummeriert) ibm.comionos.at .
-s	Squeeze: Unterdrückt aufeinanderfolgende Leerzeilen (mehrere leere Zeilen werden als eine ausgegeben) ionos.at .

Option Beschreibung

- v "Visible" – Macht nicht druckbare Steuerzeichen sichtbar (außer Tab und Zeilenende)[ionos.at](#).
- e Entspricht -v mit Anzeige des Zeilenende-Zeichens \$ am Zeilenende (kombiniert -v und Kennzeichnung von Zeilenumbrüchen)[ionos.at](#).
- t Entspricht -v mit Darstellung von Tabulatoren als ^I (Tab-Markierung)[ionos.at](#).
- E Zeigt \$ am Ende jeder Zeile (Äquivalent zu -vE, oft identisch zu -e je nach Implementierung).
- T Zeigt Tabulatoren als ^I an (ähnlich -vT, meist wie -t).
- A Kombiniert mehrere der obigen: entspricht oft -vET (alle nicht-druckbaren Zeichen, Tabs und Zeilenenden sichtbar machen).
- u (Ignoriert in GNU cat – historisch für ungepufferte Ausgabe).
- help Zeigt Hilfe-Informationen an.
- version Zeigt Versionsinfo an.

cd (Change Directory)

Beschreibung: Wechselt in ein anderes Verzeichnis. Mit cd kann das aktuelle Arbeitsverzeichnis geändert werden („change directory“)[ionos.at](#). Ohne Argument wechselt cd standardmäßig ins Heimatverzeichnis des Benutzers.

Syntax:

cd [OPTION] [VERZEICHNIS]

Man kann relative Pfade (bezogen auf das aktuelle Verzeichnis) oder absolute Pfade (ausgehend vom Wurzelverzeichnis /) angeben. Erfolgt der Wechsel in ein Zielverzeichnis, so wird dieses das neue aktuelle Arbeitsverzeichnis; bei nicht vorhandenen Zielen oder fehlenden Rechten bleibt das Verzeichnis unverändert[ionos.at](#).

Optionen:

Option Beschreibung

- L Folge symbolischen Links **logisch**. Dies ist die Voreinstellung. (Beim Navigieren durch einen Symlink bleibt man in der logischen Pfadstruktur)[ionos.at](#).
- P Folge symbolischen Links **physisch**. Wechselt in das tatsächliche Ziel im Dateisystem und löst symbolische Links auf[ionos.at](#).

Unterbefehle / spezielle Aufrufe:

- cd ~ wechselt ins Heimatverzeichnis des aktuellen Benutzers[ionos.at](#).
- cd - wechselt in das vorherige Arbeitsverzeichnis zurück (Toggle zwischen zwei Verzeichnissen)[ionos.at](#).
- cd /Pfad/zu/Verz wechselt absolut in ein angegebenes Verzeichnis vom Wurzelpfad aus (hier z.B. nach /Pfad/zu/Verz).
- cd .. wechselt eine Ebene höher (ins Parent-Verzeichnis), cd ../../ entsprechend zwei Ebenen usw.
- cd . bleibt im aktuellen Verzeichnis (. bezeichnet das aktuelle Verzeichnis, meistens kein Effekt).

pwd (Print Working Directory)

Beschreibung: Gibt den vollständigen Pfad des aktuellen Arbeitsverzeichnisses aus[ionos.at](#). Mit pwd kann man sich anzeigen lassen, *in welchem Verzeichnis man sich gerade befindet*.

Syntax:

pwd [OPTION]...

Optionen:

Option Beschreibung

- L** Ausgabe des Pfads mit Berücksichtigung symbolischer Links (logischer Pfad, Standard). Wenn z.B. ein symbolischer Link als aktuelles Verzeichnis dient, wird der Linkpfad angezeigt[ionos.de](#).
- P** Ausgabe des *physischen* Pfads: Symbolische Links werden aufgelöst, der tatsächliche Verzeichnispfad wird angezeigt[ionos.de](#).
- help** Zeigt Hilfe-Informationen zu pwd an.
- version** Zeigt Versionsinformationen an.

Hinweis: pwd ist häufig als Shell-Builtin implementiert. Das externe Programm /bin/pwd unterstützt ggf. andere oder weniger Optionen (bei GNU Coreutils nur --help und --version zusätzlich)[ionos.de](#). In der Regel genügt der Shell-eigene pwd Befehl.

whoami

Beschreibung: Zeigt den Benutzernamen des aktuell angemeldeten Users anfile-4fdk7rvx39azjretinarqb. Praktisch, um schnell zu prüfen, unter welchem Account ein

Prozess läuft.

Syntax:

whoami

Keine spezifischen Optionen. Der Befehl gibt äquivalent echo \$USER aus (den Namen des aktuellen Unix-Accounts).

touch

Beschreibung: Erzeugt eine neue leere Datei oder ändert Zeitstempel (Zugriffs- und Änderungszeit) bestehender Dateien. Wenn die Datei nicht existiert, wird sie (sofern nicht -c verwendet wird) als leere Datei angelegt[man7.org](#). Oft genutzt, um Dateien schnell anzulegen oder deren *timestamp* zu aktualisieren.

Syntax:

touch [OPTION]... DATEI...

Optionen:

Option	Beschreibung
-a	Ändert nur die Zugriffszeit (atime) der Datei man7.org .
-m	Ändert nur die Änderungszeit (mtime) der Datei man7.org .
-c, --no-create	Legt keine neue Datei an. Wenn die Datei nicht existiert, wird kein neues leeres File erstellt (ansonsten Standardverhalten ohne -c) man7.org man7.org .
-d, --date=STRING	Verwendet den angegebenen Datums-String anstelle der aktuellen Zeit. Der String wird geparsst (z.B. "2025-05-01 12:00:00" oder relative Angaben wie "next Friday" sind möglich) man7.org man7.org .
-t [[CC]YY]MMDDhhmm[.ss]	Verwendet einen konkreten Zeitstempel im angegebenen Format (JJMMTThhmm[.ss]) anstatt der aktuellen Zeit man7.org . Beispiel: -t 202312312359.59 setzt 31.12.2023 23:59:59.
-r, --reference=DATEI	Übernimmt die Zeitstempel (Zugriffs- und Änderungszeit) der angegebenen Referenzdatei. Die Zielfile erhält also dieselben Zeiten wie DATEI man7.org .
-h, --no-dereference	Ändert bei symbolischen Links <i>den Link selbst</i> (deren Metadaten), nicht die referenzierte Datei. Ohne diese Option würde touch auf den Ziel-Dateipfad wirken (nicht auf den Symlink). (<i>Diese Option ist nur wirksam, wenn das System</i>

Option	Beschreibung
	<i>das Verändern von Symlink-Zeitstempeln unterstützt.)</i> man7.org
--time=WORD	Bestimmt, welcher Zeitstempel geändert wird. Mögliche Werte: atime, access, use (entspricht -a), oder mtime, modify (entspricht -m) man7.org .
--help, --version	Anzeigen der Hilfe bzw. Versionsinformation zu touch.

man (Manual Page Viewer)

Beschreibung: Zeigt die Handbuchseite (Manual Page) für einen angegebenen Befehl oder ein Thema an. man ist das Offline-Hilfe- und Dokumentationssystem unter Unix/Linux.

Syntax:

man [SECTION] <Thema/Command>

Beispiel: man ls zeigt die Manpage zu **ls**. Manpages sind meist nach Kategorien/Abschnitten (1=Benutzerkommandos, 5=Dateiformate, 8=Administrator-Kommandos etc.) gegliedert – optional kann man mit man 5 passwd z.B. direkt die Manpage zum Dateiformat **passwd** (Abschnitt 5) öffnen.

Optionen:

Option	Beschreibung
-k Suchwort	Durchsucht die Manpage-Kurzzusammenfassungen nach <i>Suchwort</i> (entspricht dem Befehl apropos). Beispiel: man -k network listet alle Manual-Topics auf, deren Beschreibung "network" enthält.
-f Befehl	Findet den Namen und Abschnitt einer Manpage zu <i>Befehl</i> (entspricht whatis). Beispiel: man -f passwd zeigt kurz "passwd (1) - change user password".
-a	Zeigt nacheinander alle Manpages an, die zum angegebenen Namen passen (wenn z.B. ein Name in mehreren Abschnitten existiert).
-P Pager	Legt ein anderes Pager-Programm fest (Standard ist less). Beispiel: man -P cat ls würde die ls-Manpage durch cat leiten.
--help, --version	Zeigt Hilfe bzw. Version von man.
whatis	

Beschreibung: Zeigt eine kurze Ein-Satz-Beschreibung zu einem Befehl an (frägt die Datenbank der *whatis*-Einträge ab). Es ist eine Kurzform von `man -f` und liefert die Synopsis aus der Manpage.

Syntax:

`whatis <Befehl/Topic>`

Optionen:

Option	Beschreibung
(keine üblichen Optionen)	<code>whatis</code> wird meist ohne Optionen verwendet. Es greift auf die gleichen Datenbanken wie <code>apropos/man -k</code> zu. Sollte die Datenbank nicht aktuell sein, kann sie mit <code>mandb</code> erneuert werden.

Benutzer- und Gruppenverwaltung

passwd (Passwort ändern/setzen)

Beschreibung: Ändert das Passwort eines Benutzerkontos oder verwaltet Konto-Sperren. Als normaler Benutzer kann man mit `passwd` sein eigenes Passwort ändern; als `root` kann man mit `passwd <user>` das Passwort eines anderen Accounts setzen oder bestimmte Eigenschaften (Sperren, Ablaufdatum) ändern.

Syntax:

`passwd [OPTION] [BENUTZER]`

Standardaufruf ohne Benutzer wirkt auf den aktuellen Benutzer. Bei Angabe eines Benutzernamens werden dessen Passwortinformationen geändert (Root-Rechte erforderlich).

Optionen:

Option Beschreibung

-l, --lock	Sperrt das Benutzerkonto, indem das Passwort in <code>/etc/shadow</code> mit einem Präfix (meist <code>!</code> oder <code>*</code>) unbrauchbar gemacht wird wiki.ubuntuusers.de . Der Benutzer kann sich danach nicht mehr anmelden, bis das Konto wieder entsperrt wird.
-u, --unlock	Entsperrt ein zuvor gesperrtes Konto (entfernt das Sperr-Präfix im verschlüsselten Passwort) wiki.ubuntuusers.de .
-d, --delete	Löscht das Passwort des Benutzers (leer setzen). Achtung: Damit wird eine passwortlose Anmeldung möglich, sofern nicht andersweitig gesichert wiki.ubuntuusers.de .

Option Beschreibung

- e, -- expire Markiert das Passwort als abgelaufen. Beim nächsten Login muss der Benutzer ein neues Passwort setzen.
- x *TAGE* Setzt die maximale Gültigkeitsdauer eines Passworts auf *TAGE*. Nach Ablauf muss es geändert werden. (Änderung des PASS_MAX_DAYS in /etc/shadow.)
- n *TAGE* (PASS_MIN_DAYS). Setzt die minimale Anzahl an Tagen zwischen zwei Passwortänderungen (PASS_MIN_DAYS). Der Benutzer kann sein Passwort erst nach *TAGE* Tagen erneut ändern.
- w *TAGE* Warnzeit in Tagen vor Passwortablauf (PASS_WARN_AGE). Der Benutzer erhält *TAGE* Tage vor Ablauf eine Warnung beim Login.
- S Zeigt den Passwortstatus an (Gesperrt? nicht gesetzt? Datum der letzten Änderung usw.). passwd -S <user> gibt eine Statuszeile aus.

Interaktives Verhalten: Wird passwd ohne Optionen aufgerufen, fordert es zur Eingabe des aktuellen Passworts (zur Authentifizierung, falls nicht root) und zweimal zur Eingabe des neuen Passworts auf (zwecks Bestätigung). Es verweigert einfache Passwörter gemäß der System-Policy (Pluggable Authentication Modules/pam Richtlinien). Root-Benutzer muss das alte Passwort nicht kennen.

chage (Change Age – Passwortalter und Gültigkeit)

Beschreibung: Verändert Einstellungen zum Passwort-Aging – also Ablaufdatum, Mindestalter, Warnzeit etc. Mit chage kann ein Administrator festlegen, wann ein Benutzer sein Passwort ändern muss bzw. wann der Account deaktiviert wirdfile-4fdk7rvx39azjretinarqb.

Syntax:

chage [OPTION]... BENUTZER

Optionen:

Option Beschreibung

- l Listet die aktuellen Passwort- und Alterseinstellungen für den Benutzer in einem lesbaren Format auf (Letzte Änderung, Min/Max Alter, Warnzeit, Inaktivzeit, Ablaufdatum) – Übersicht.
- m *TAGE* Setzt die minimale Anzahl von Tagen *TAGE*, die ein Passwort gültig sein muss, bevor es geändert werden darf (PASS_MIN_DAYS).
- M *TAGE* Setzt die maximale Gültigkeitsdauer in Tagen (PASS_MAX_DAYS). Nach *TAGE* Tagen läuft das Passwort ab und muss geändert werden.

Option Beschreibung

-W TAGE	Setzt die Anzahl der Tage <i>TAGE</i> Vorwarnzeit vor Passwortablauf (PASS_WARN_AGE).
-I TAGE	Legt fest, wie viele Tage nach Passwortablauf der Account inaktiv wird (PASS_INACTIVE_DAYS). Ein Wert von 0 deaktiviert den Account sofort nach Ablauf des Passworts, -1 schaltet diese Funktion aus wiki.ubuntuusers.de .
-E DATUM	Setzt ein Ablaufdatum für das Benutzerkonto (im Format YYYY-MM-DD). Nach diesem Datum wird der Account deaktiviert wiki.ubuntuusers.de . Das Datum -1 (oder "" leer) entfernt ein gesetztes Ablaufdatum (kein Ablauf).
-d YYYY-MM-DD	Setzt das Datum der letzten Passwortänderung auf das angegebene Datum (wirkt wie Backdating oder um zu triggern, dass ein Passwort abläuft). Kann auf 0 gesetzt werden, um das Passwort als sofort ablaufend zu markieren.
-h, --help	Zeigt Hilfe an.

Hinweis: Viele dieser Einstellungen lassen sich auch mit passwd (Optionen -x, -n, -w, -i, -e) vornehmen. chage bietet eine dedizierte Schnittstelle dafür. Ohne Optionen (als root) startet chage <user> interaktiv einen Dialog, um alle Werte einzustellen.

w

Beschreibung: Zeigt an, welche Benutzer gerade am System angemeldet sind und was sie gerade tun. w fasst Informationen über aktuell eingeloggte User und deren laufende Prozesse zusammen (inkl. Login-TTY, Leerlaufzeit, genutzte CPU-Zeit, aktueller Befehl)`file-4fdk7rvx39azjretinarqb`.

Syntax:

w [OPTION]

Optionen:

Option Beschreibung

-h	Blendet die Kopfzeile in der Ausgabe aus (unterdrückt die Überschriftenzeile).
-s	Kurze Ausgabe. Unterdrückt die Spalten für Login-Zeit, Leerlauf etc. (nicht auf allen Systemen vorhanden).
-f	Zeigt Informationen zu von utmp als <i>geflagt</i> markierten Einträgen. Auf einigen Systemen steuert dies die Anzeige von Remote-Host oder Exit-Informationen.

Option Beschreibung

(Bei neueren procps-Versionen hat -f keine Funktion, da Host immer gezeigt wird.)

-V, --version Zeigt Versionsinfo von w.

Die Ausgabe von w enthält in der ersten Zeile System-Uptime und Load Average. Danach folgen pro eingeloggtem Benutzer: Benutzernname, Terminal, Remote-Host, Login-Zeit, Leerlaufdauer, CPU-Verbrauch und der aktuell laufende Prozess (TTY-Prozess).

who

Beschreibung: Zeigt Informationen über die aktuell angemeldeten Benutzer. Im Gegensatz zu w liefert who primär Login-Zeitpunkte und Terminalsessions, jedoch keine Prozess- oder Idle-Infos.

Syntax:

who [OPTION]... [Datei]

Standard liest who die System-Login-Datenbank /var/run/utmp und gibt für jeden Eintrag Benutzer, Terminal, Datum/Zeit und ggf. Host aus.

Optionen:

Option Beschreibung

-H Zeigt eine Kopfzeile (Überschriften) über den Spalten an.

-q "Quick who": Zeigt nur Benutzernamen und die Anzahl der angemeldeten Nutzer. (Gleichbedeutend mit users-Befehl.)

-b Zeigt die Zeit des letzten Systemstarts (Boot time) an.

-r Zeigt den aktuellen Runlevel des Systems.

-a Zeigt alle Einträge aus utmp, inklusive Einträgen für laufende Prozesse am System (wie getty oder systemd-Spawns). Kombination von mehrere Flags: entspricht oft -b -d --login -p -r -t -T -u je nach Implementierung.

-m Zeigt nur die Einträge für das aktuelle Terminal an (beschränkt auf den Benutzer, der den Befehl aufruft).

-u Zeigt zusätzlich die Leerlaufzeit (Idle) und den PID des Login-Prozesses für jeden Benutzer.

Option	Beschreibung
-T	Zeigt den Terminal-Status (\pm Flag) an: + bedeutet terminal ist bei <i>mesg</i> schreibbar, - bedeutet nicht schreibbar, ? unbekannt. (In vielen who-Implementierungen bereits Standardverhalten ohne Option.)
--help, --version	Anzeigen von Hilfe bzw. Versionsinfo.

useradd (Benutzer anlegen)

Beschreibung: Legt ein neues Benutzerkonto an. useradd ist ein Low-Level-Befehl, der die Systemkontodateien (wie /etc/passwd, /etc/shadow, /etc/group) direkt verändert, um einen neuen User einzurichten[pq.hosting](#). In Debian/Ubuntu-Umgebungen wird oft das Wrapper-Skript adduser verwendet, das auf useradd aufbaut und interaktiv ein paar Dinge abfragt; useradd selbst ist nicht interaktiv und benötigt alle Optionen zum Festlegen der Eigenschaften des neuen Accounts.

Syntax:

useradd [OPTION]... BENUTZERNAME

Optionen: (häufigste Optionen beim Anlegen eines Benutzers)

Option	Beschreibung
-c	Kommentarfeld (GECOS) für den Benutzer, z.B. voller Name oder Hinweis. <i>Kommentar</i> Wird in /etc/passwd eingetragen wiki.ubuntuusers.de .
-d HOME	Pfad des Home-Verzeichnisses. Standard: /home/<Benutzername>. Kann geändert werden; wenn -m verwendet wird, wird dieses Verzeichnis erstellt und mit Standard-Dateien aus /etc/skel befüllt wiki.ubuntuusers.de .
-m	Legt das Home-Verzeichnis an und kopiert Standard-Dateien hinein. Sollte gesetzt sein, außer man möchte <i>kein</i> Home-Verz. (Ohne -m wird das Home nicht automatisch erstellt) wiki.ubuntuusers.de .
-s SHELL	Login-Shell des Benutzers. Standard ist normalerweise /bin/bash (oder was in /etc/default/useradd als SHELL definiert ist). Kann z.B. auf /sbin/nologin gesetzt werden für System-Accounts wiki.ubuntuusers.de .
-u UID	Gewünschte User-ID (numerisch). Darf nicht bereits vergeben sein (außer mit -o). Wenn nicht angegeben, wird die nächste freie UID > 999 gewählt wiki.ubuntuusers.de .
-o	Erlaubt die Verwendung einer bereits vergebenen UID (UID-Doppelungen). Zusammen mit -u benutzt, um zwei Benutzer mit identischer UID

Option	Beschreibung
	anzulegen. (Normalerweise nicht empfohlen, da UID eindeutig sein sollen.)
-g GRP	Primäre Gruppe (Name oder GID), der der Benutzer angehört wiki.ubuntuusers.de . Standard: eine neue Gruppe mit gleichem Namen wie der Benutzer (bei entsprechender Systemkonfiguration) oder eine in /etc/default/useradd definierte GROUP. Die angegebene Gruppe muss existieren.
-G Liste	Liste zusätzlicher Gruppen (sekundäre Gruppen), denen der Benutzer angehören soll wiki.ubuntuusers.de . Gruppennamen durch Komma getrennt angeben (ohne Leerzeichen).
-p Passwort	Das verschlüsselte Passwort für den Account (wie es z.B. von openssl passwd oder crypt(3) erzeugt würde) wiki.ubuntuusers.de . Wichtig: Aus Sicherheitsgründen sollte man hier <i>nicht</i> ein Klartextpasswort übergeben. Bleibt -p leer, wird der Account ohne gesetztes Passwort angelegt (Anmeldung evtl. gesperrt bis passwd gesetzt wird).
-e	(bei useradd nicht üblich – wird bei usermod benutzt für Ablauf setzen. Bei einigen Systemen könnte useradd -e <Datum> ein Ablaufdatum setzen.)
-f TAGE	Anzahl an Tagen nach Passwortablauf, bis der Account deaktiviert wird (Analog zu chage -l). -f 0 würde den Account sofort mit Passwortablauf sperren. Standard -1 (nie sperren).
-N	(Bei einigen Systemen: <i>No user group</i> – verhindert das automatische Anlegen einer gleichnamigen Gruppe.)
-r	Erstellt einen System-Account (UID unter einer gewissen Schwelle, z.B. <1000, kein Expire, kein Mail-Spool). Oft genutzt für Systembenutzer ohne Login.
-M	Kein Home-Verzeichnis anlegen (selbst wenn Default CREATE_HOME aktiviert ist).
-k SKELDIR	Gibt ein alternatives Skeleton-Verzeichnis an (anstatt /etc/skel), von wo aus bei -m die initialen Dateien ins Home kopiert werden.
-D	Zeigt die aktuellen Standardwerte für useradd an (oder mit anderen -D- Optionen können diese verändert werden).

Nach dem Anlegen mit useradd sollte meist ein Passwort gesetzt werden (passwd <user>), außer es handelt sich um einen Account ohne Login.

usermod (Benutzerkonto ändern)

Beschreibung: Modifiziert ein bestehendes Benutzerkonto – z.B. Name, UID, Primärgruppe, zusätzliche Gruppen, Login-Shell, Expire-Datum

usw.wiki.ubuntuusers.dewiki.ubuntuusers.de. Im Grunde die gleichen Optionen wie useradd, aber angewendet auf bestehende Benutzer. Häufig genutzt, um Benutzer umzubenennen, in andere Gruppen aufzunehmen oder Accounts zu sperren.

Syntax:

usermod [OPTION]... LOGIN

Optionen:

Option	Beschreibung
-c Kommentar	Neuen Kommentar (GECOS-Feld) setzen wiki.ubuntuusers.de .
-d Neues_Home	Ändert das Home-Verzeichnis des Benutzers auf <i>Neues_Home</i> . Hinweis: Ohne weiteres passiert das alte Home nicht automatisch mit. Mit zusätzlicher Option -m werden die Inhalte des alten Home-Verz. ins neue Verzeichnis verschoben und das neue Verzeichnis ggf. erstellt wiki.ubuntuusers.de .
-m	(Mit -d verwenden) Verschiebt beim Ändern des Homepfads (-d) die Dateien vom alten ins neue Home und erstellt es falls nötig wiki.ubuntuusers.de .
-l Neuer_Login	Ändert den Login-Namen des Benutzers (Umbenennung des Accounts) wiki.ubuntuusers.dewiki.ubuntuusers.de . Der Benutzername in allen Systemkontodateien wird angepasst, <i>nicht</i> jedoch automatisch der Name des Home-Verzeichnisses oder Mail-Spool (diese müssten manuell umbenannt werden, falls gewünscht).
-u Neue_UID	Ändert die numerische Benutzer-ID auf <i>Neue_UID</i> . Falls die UID bereits vergeben ist, muss -o benutzt werden. Dateien im Home-Verzeichnis mit alter UID werden auf neue UID umgestellt; Dateien außerhalb des Homes muss der Admin selbst anpassen wiki.ubuntuusers.dewiki.ubuntuusers.de .
-o	Erlaubt die Verwendung einer bereits vergebenen UID (nur zusammen mit -u sinnvoll).
-g <i>Neue_PrimGruppe</i>	Ändert die primäre Gruppenzugehörigkeit (Name oder GID) manpages.ubuntu.com . Die neue Gruppe muss existieren. Alle Dateien im Home, die der alten primären Gruppe gehörten, werden

Option	Beschreibung
	auf die neue Gruppe übertragen manpages.ubuntu.com . (Dateien außerhalb des Homes müssen ggf. manuell angepasst werden.)
-G Gruppe(n)	Setzt die Liste der sekundären Gruppen neu für den Benutzer manpages.ubuntu.com . Wichtig: Ohne -a entfernt diese Option den Benutzer aus allen Gruppen, die nicht in der Liste stehen wiki.ubuntuusers.demanpages.ubuntu.com . Um einen Benutzer zusätzlichen Gruppen hinzuzufügen, sollte man -a -G gemeinsam verwenden (siehe -a).
-a	<i>Append</i> – fügt den Benutzer zusätzlichen Gruppen hinzu, anstatt die Gruppenzugehörigkeit zu überschreiben wiki.ubuntuusers.dewiki.ubuntuusers.de . Nur in Kombination mit -G verwenden , um den Benutzer der/die angegebenen Gruppen hinzuzufügen, ohne ihn aus anderen zu entfernen. (Siehe Warnung unten.)
-s Shell	Ändert die Login-Shell des Benutzers wiki.ubuntuusers.de . Z.B. auf /sbin/nologin setzen, um interaktive Logins zu unterbinden.
-e YYYY-MM-DD	Setzt das Ablaufdatum des Accounts (das Datum, an dem das Konto deaktiviert wird) wiki.ubuntuusers.de . Nach diesem Datum kann sich der User nicht mehr anmelden. usermod -e "" user entfernt ein Ablaufdatum (setzt es auf <i>nie</i>).
-f Tage	Setzt die Anzahl an Tagen nach Passwortablauf, bis der Account deaktiviert wird wiki.ubuntuusers.de . -f 0 deaktiviert sofort nach Ablauf; -f -1 entfernt eine evtl. gesetzte Deaktivierung (nie sperren nach Ablauf).
-L	Sperrt das Benutzerkonto (lock). Fügt ein ! dem Passwort-Hash voran (im Shadow-Eintrag) wiki.ubuntuusers.de , was die Anmeldung verhindert. Kann nicht mit -p oder -U kombiniert werden.
-U	Entsperrt ein zuvor gesperrtes Passwort (passwd -l). Entfernt das ! im Shadow-Passwortfeld, sodass der ursprüngliche Hash wieder wirksam wird wiki.ubuntuusers.de . Nicht zusammen mit -L oder -p verwenden.
-p Hash	Setzt das <i>verschlüsselte</i> Passwort (Hash) des Users auf den angegebenen Wert wiki.ubuntuusers.de . Sollte mit Vorsicht genutzt werden – für Passwortänderung ist passwd meist besser geeignet.

Option	Beschreibung
--------	--------------

--help	Hilfe anzeigen.
--------	-----------------

Wichtige Hinweise:

- *Warnung:* Die Kombination von -G ohne -a überschreibt die Gruppenliste vollständig. Das heißt, der Benutzer wird aus allen Gruppen entfernt, die nicht explizit bei -G genannt sind [wiki.ubuntuusers.de](#). Um einen User nur hinzuzufügen, niemals ohne -a verwenden!
- Man kann keinen angemeldeten Benutzer umbenennen oder dessen UID ändern, solange er eingeloggt ist oder laufende Prozesse hat. Zudem keine Änderungen an dem eigenen Account durchführen, während man noch mit diesem angemeldet ist (kann zu Inkonsistenzen führen).
- Beim Ändern des Benutzernamens oder UIDs sollten evtl. Crontab-Dateien und geplante at-Jobs manuell umbenannt/angepasst werden, und in NIS/LDAP-Umgebungen müssen Änderungen dort vorgenommen werden [wiki.ubuntuusers.de](#).

userdel (Benutzer löschen)

Beschreibung: Löscht ein Benutzerkonto aus dem System (entfernt die Einträge in /etc/passwd, /etc/shadow und ggf. /etc/group falls der Benutzer eine eigene Gruppe hatte). Optional können auch das Home-Verzeichnis und die Mail-Spool des Benutzers gelöscht werden.

Syntax:

userdel [OPTION] BENUTZER

Optionen:

Option Beschreibung

-r Entfernt *rekursiv* das Heimatverzeichnis des Benutzers **und** seine Mail-Spool (üblicherweise /var/spool/mail/<user>), sowie alle Dateien darin, beim Löschen des Accounts. Ohne -r bleiben Home und Mails unangetastetfile-4fdk7rvx39azjretinarqb.

-f Force: Erzwingt das Löschen des Accounts, selbst wenn der Benutzer noch angemeldet ist oder laufende Prozesse hat. (Warnung: kann zu inkonsistenten Systemzuständen führen, Prozess bleibt ohne Benutzer etc. Diese Option sollte vermieden werden.)

Option Beschreibung

Ohne Option wird der Account aus den Systemdateien gelöscht, aber das (keine) Home-Verzeichnis, die Dateien und Mails bleiben bestehen. Admin muss sie ggf. manuell entfernen oder anderweitig zuordnen.

Hinweis: userdel bricht ab, wenn der Benutzer gerade angemeldet ist (außer mit -f). Es ist ratsam, vor dem Löschen sicherzustellen, dass keine Prozesse mehr unter der betreffenden UID laufen (z.B. mit pkill -u username beenden). Dateien außerhalb des Home, die dem Benutzer gehören, verbleiben auf dem Dateisystem mit der nun unbenutzten UID – diese müssten manuell gesucht und entfernt oder neu zugeordnet werden (z.B. mit find / -uid <oldUID>).

id

Beschreibung: Zeigt die Benutzer- und Gruppen-Identitäten eines Benutzerkontos an. Standardmäßig gibt id die UID, primäre GID und zugehörigen Gruppennamen des aktuellen Benutzers zurückfile-4fdk7rvx39azjretinarqb. Mit Angabe eines Benutzernamens kann man die IDs für einen anderen Account abrufen.

Syntax:

id [OPTION]... [BENUTZER]

Optionen:

Option	Beschreibung
(ohne)	Ohne Options werden UID, Benutzername, GID, Gruppename und alle zusätzlichen Gruppen des Benutzers aufgelistet (z.B. uid=1000(max) gid=1000(max) Gruppen=1000(max),27(sudo),...).
-u	Gibt nur die effektive User-ID (UID, numerisch) aus. Beispiel: id -u alice könnte 1001 ausgeben.
-g	Gibt nur die effektive Gruppen-ID (GID, numerisch) aus.
-G	Gibt alle Gruppen-IDs (numerisch) aus, in denen der Benutzer Mitglied ist (Primär- und Sekundärgruppen).
-n	Gibt Namen anstelle von numerischen IDs. Kann in Kombination mit -u, -g, -G verwendet werden. Z.B. id -Gn alice listet alle Gruppennamen von alice.
-r	Zeigt die reale UID/GID an (falls z.B. mit sudo aufgerufen, reale vs. effektive ID unterscheiden).
-Z	Zeigt SELinux-Sicherheitskontext, falls SELinux aktiv ist (z.B. unconfined_u:unconfined_r:unconfined_t:s0 für einen unconfined Benutzer).

Option Beschreibung

--help, --version Hilfe bzw. Version anzeigen.

groupadd (Gruppe anlegen)

Beschreibung: Legt eine neue Gruppe im System an. Fügt einen Eintrag in /etc/group (und ggf. /etc/gshadow) hinzu.

Syntax:

groupadd [Optionen] GRUPPENNAME

Optionen:

Option Beschreibung

-g GID Verwendet die angegebene numerische GID für die neue Gruppe. Sollte noch nicht vergeben sein (oder mit -o überschrieben werden).

-o Erlaubt die Verwendung einer bereits existierenden GID (zusammen mit -g zu nutzen, um GID-Doppel zu erzwingen).

-f "Force": falls die Gruppe schon existiert, keinen Fehler melden; falls die angegebene GID bereits vergeben ist, wählt eine andere (ignoriert -o).

-r Erstellt eine Systemgruppe (GID wird aus dem System-Bereich gewählt, z.B. <1000).

-K Schlüssel= Wert Überschreibt einen Eintrag aus /etc/login.defs temporär (z.B. -K GID_MIN=500 um untere GID-Bereiche anzupassen).

-p Passwort Setzt das (verschlüsselte) Gruppenpasswort direkt. (Gruppenpasswörter werden selten genutzt – ermöglicht mit newgrp temporär in eine Gruppe zu wechseln.)

groupmod (Gruppe ändern)

Beschreibung: Ändert Eigenschaften einer bestehenden Gruppe – hauptsächlich Name oder GID.

Syntax:

groupmod [Optionen] GRUPPE

Optionen:

Option	Beschreibung
-n	Ändert den Gruppennamen in <i>NeuerName</i> . Der alte Name in allen <i>NeuerName</i> relevanten Systemdateien wird ersetzt.
-g NeueGID	Ändert die GID der Gruppe. Muss eindeutig sein (oder mit -o doppelt erlaubt). Dateien mit alter GID außerhalb, die nicht automatisch erfasst werden, müssen eventuell manuell angepasst werden.
-o	Erlaubt das Setzen einer bereits vergebenen GID (nur mit -g relevant).
-p Passwort	Ändert das (verschlüsselte) Gruppenpasswort. (Entspricht Editieren von /etc/gshadow).

groupdel (Gruppe löschen)

Beschreibung: Entfernt eine Gruppe aus der Systemgruppe-Datei. Die Gruppe wird aus /etc/group (und /etc/gshadow) getilgt.

Syntax:

groupdel GRUPPENNAME

Hinweis: Dieser Befehl hat keine besonderen Optionen – er löscht die Gruppe, sofern sie nicht als Primärgruppe eines Benutzers in /etc/passwd noch verwendet wird. Man kann keine Gruppe löschen, die noch als Hauptgruppe eines existierenden Benutzerkontos eingetragen ist; in dem Fall muss man ggf. erst den Benutzer anpassen oder löschen. Dateien auf dem System, die dieser Gruppe gehören, behalten die nun ungültige GID bei, bis man sie manuell ändert.

su (Switch User)

Beschreibung: Wechselt den Benutzerkontext im laufenden Terminal. Mit su („substitute user“) kann man einen anderen Benutzeraccount annehmen – standardmäßig root, wenn kein Benutzer angegeben wird. Dies erfordert die Eingabe des Zielbenutzer-Passworts (außer beim Wechsel zu root, dort abhängig von den Systemeinstellungen – oft ist root-Login per Passwort deaktiviert und man nutzt sudo).

Syntax:

su [OPTION]... [BENUTZER [-]]

Wird BENUTZER weggelassen, versucht su auf den Superuser (root) zu wechseln.

Optionen:

Option	Beschreibung
---------------	---------------------

- oder -l	Startet eine <i>Login-Shell</i> der Zielusers, d.h. die Umgebung wird vollständig umgeschaltet als würde sich der Benutzer frisch anmelden. Das Home-
------------------	---

Option Beschreibung

	Verzeichnis, Pfad (\$PATH) etc. werden übernommen. Beispiel: su - alice wechselt zu Benutzer <i>alice</i> mit deren Umgebung (äquivalent zu vollem Logout/Login als <i>alice</i>).
	Führt einen angegebenen <i>Befehl</i> als Zielbenutzer aus und kehrt dann zurück.
-c Befehl	Beispielsweise su -c "systemctl restart apache2" - root führt als root einen Neustart des Apache durch und beendet sich sofort wieder.
-m, -p	Behalten die aktuelle Umgebung (<i>nicht</i> auf Ziel-User umschalten). Der aktuelle \$HOME, \$PATH, usw. bleiben erhalten, nur die Rechte wechseln. Standardverhalten ohne -l. (In vielen su Implementierungen ist -m identisch mit -p: <i>preserve environment</i>).
-s Shell	Startet die angegebene <i>Shell</i> statt der Standard-Login-Shell des Zielbenutzers. Der Pfad zur Shell muss im /etc/shells stehen, sofern nicht root.
-g Gruppe	Wechselt in Kombination mit Benutzer auch die primäre Gruppe auf <i>Gruppe</i> . (Nur verfügbar in neueren util-linux su).
-<Ziffer>	(BSD su) Angabe einer Session-Klasse – unter Linux nicht genutzt.
--help, --version	Anzeigen von Hilfe bzw. Version.

Hinweise:

- su ohne - (also z.B. su root) wechselt zwar die Benutzer-ID, aber behält das aktuelle Verzeichnis und weite Teile der Environment von Ursprungskonto bei (\$LOGNAME, \$HOME bleiben z.B. vom ursprünglichen Nutzer). Das kann zu irritierenden Effekten führen (z.B. Schreibrechte im falschen Home). Daher nutzt man meist su - für einen sauberen Wechsel.
- Viele Distributionen konfigurieren aus Sicherheitsgründen su so, dass ein direkter Wechsel zu root per Passwort entweder deaktiviert ist oder kein root-Passwort gesetzt ist. Stattdessen wird sudo bevorzugt, um Root-Rechte zu erlangen.

sudo (Superuser Do)

Beschreibung: Führt ein Kommando mit erhöhten Rechten aus, typischerweise als *root*, ohne die Benutzeridentität dauerhaft zu wechseln. Benutzer können mit sudo Befehle mit Administratorrechten ausführen, sofern sie in /etc/sudoers entsprechend autorisiert sind. Im Gegensatz zu su wird hier das eigene Passwort abgefragt (nicht das des Zielusers), und nur spezifische Befehle/Optionen sind erlaubt gemäß Konfiguration.

Syntax:

`sudo [OPTION]... [COMMAND]`

Wird ein Befehl angegeben, führt sudo diesen als root (Standard) aus. Ohne Befehl zeigt sudo ggf. die erlaubten Befehle an oder wechselt in eine sudo-Shell (bei -s).

Optionen (Auswahl):

Option Beschreibung

- Listet die erlaubten (und verbotenen) Befehle für den aufrufenden Benutzer
- l** auf (nach sudoers-Regeln). Hilfreich, um zu sehen, was man mit sudo ausführen darf.

- u** Führt das angegebene Kommando als *Benutzer* aus (anstatt als root).
Beispiel: `sudo -u alice whoami` zeigt "alice", wenn man als anderer User ausführt. Standard ist root.

- s** Startet eine Shell als Zielbenutzer (Standard root). Ähnlich wie su, aber mittels sudo-Rechte. Es wird die im \$SHELL oder in passwd eingetragene Shell geöffnet.

- i** Simuliert eine vollständige Login-Shell wie su - (initialisiert Umgebung des Zielusers). Entspricht sudo -s plus Login-Umgebung.

- b** Führt den Befehl im Hintergrund aus (*background*). Sudo verlangt dennoch ggf. ein Passwort, aber detach danach.

- E** Beibehaltung der aktuellen Environment-Variablen (normalerweise bereinigt sudo einige Variablen aus Sicherheitsgründen). Erlaubt das Mitnehmen der eigenen Umgebung. (Ob das erlaubt ist, kann in sudoers eingeschränkt sein.)

- p** Definiert einen eigenen Passwort-Prompt. z.B. `sudo -p "Passwort für %u:"` -v
Prompt (Platzhalter %u = Username).

- v** "Validate": Aktualisiert/überprüft das eigene sudo-Authentifizierungsticket. D.h. fordert ggf. Passwort an, speichert das sudo Token (5min Standard) aber führt sonst nichts aus. (Verlängert die Zeit, in der man kein Passwort erneut eingeben muss.)

- k** Invalidiert das eigene sudo-Ticket sofort (nachfolgende sudo-Aufrufe verlangen wieder Passwort) – (*kill credential*).

- H** Setzt \$HOME in der Environment auf das Home des Zielusers (root). Normalerweise belässt sudo \$HOME vom Aufrufer; mit -H verhält es sich wie su -.

Option Beschreibung

--help, -h Hilfe anzeigen (kurze Übersicht der Optionen).

-K Löscht das Zeitstempel-Ticket komplett (ähnlich -k aber gründlicher, forciert Passwort prompt beim nächsten Mal).

--version Versionsinfo ausgeben (auch Liste der unterstützten Sicherheitsmechanismen etc.).

Hinweise: Bei erfolgreicher Authentifizierung legt sudo einen Zeitstempel (Ticket) an, sodass für nachfolgende Aufrufe (Standard ~5 Minuten) kein erneutes Passwort nötig ist. Dies kann per sudoers Einstellung (timestamp_timeout) angepasst oder mit sudo -k/-K beeinflusst werden.

pkexec (PolicyKit Execute)

Beschreibung: Führt ein Programm als ein anderer Benutzer aus (ähnlich wie sudo, aber Teil von PolKit/PolicyKit). Standardmäßig wird, sofern nicht anders angegeben, das Programm mit root-Rechten gestartetfreedesktop.org. pkexec erfordert, dass der aufrufende Benutzer berechtigt ist, die Aktion auszuführen (PolKit Policies); falls ja, wird eine Authentifizierungsdialog (grafisch oder textbasiert) angezeigt, um ggf. das Passwort einzugeben. Nach erfolgreicher Auth kann das gewünschte Kommando mit erhöhten Rechten ausgeführt werden.

Syntax:

pkexec [--user <Benutzer>] <Kommando> [Arg...]

Optionen:

Option Beschreibung

--user Führt das angegebene Programm als Benutzer aus. Standard ist root
<Benutzer> (Administratorkonto)freedesktop.org.

--disable-internal-agent Unterdrückt den internen Authentifizierungs-Agent von pkexec.

--internal-agent Normalerweise, falls kein grafischer PolKit-Auth-Agent verfügbar ist, startet pkexec einen eigenen textuellen Auth-Dialog im Terminal. Mit dieser Option wird in so einem Fall statt dessen das Programm **nicht** ausgeführt (Fehler, wenn kein Agent aktiv)freedesktop.org.

--help / --version Zeigt Hilfe bzw. Version von pkexec.

Hinweise: pkexec erbt ein Minimal-Umfeld (Environment), um das Einschleusen von Code via Umgebungsvariablen zu verhindernfreedesktop.org. Display- und Xauth-Variablen werden aus Sicherheitsgründen *nicht* an das Zielprogramm weitergegeben,

außer es wurde in den PolKit-Regeln explizit erlaubt (GUI-Programme via pkexec sind daher tricky ohne entsprechende PolKit-Policy). In modernen Distributionen ist pkexec oft als Ersatz für grafische gksu-Aufrufe gedacht, um z.B. aus der Desktop-Umgebung heraus GUI-Administrationsprogramme mit Root-Rechten zu starten (wobei PolKit dann i.d.R. das Benutzer-Passwort abfragt).

Berechtigungen und Zugriffskontrolle

umask (Set Default Permissions Mask)

Beschreibung: Legt eine Berechtigungsschutzmaske fest, die bei neu erstellten Dateien und Verzeichnissen die Standardzugriffsrechte beeinflusst. umask subtrahiert Bits von den vollen Zugriffsrechten. So bestimmt sie, welche Rechte *NICHT* automatisch gesetzt werden dürfen. Dies wirkt pro Prozess/Session. Als Shell-Builtin zeigt umask ohne Parameter die aktuelle Maske an.

Syntax:

umask [Option] [Maske]

Optionen / Verwendung:

- Ohne Argument zeigt umask die aktuelle Mask in Oktalnotation an (z.B. 0022). Mit Option **-S** in symbolischer Form (z.B. u=rwx,g=rx,o=rx entspricht Maske 022).
- Mit Angabe einer neuen Maske (z.B. umask 027) wird diese gesetzt. Die Maske wird meist oktal angegeben: jedes Bit, das in der Maske 1 ist, nimmt der neu erstellten Datei die entsprechende Berechtigung weg. Beispiel: Maske 022 bedeutet, neuen Dateien werden keine Gruppe- und Andere-Schreibrechte gegeben (d.h. max Resultatrechte für Datei typ. 644, für Verzeichnis 755).
- **Symbolische Maske:** Man kann auch wie bei chmod symbolisch angeben, z.B. umask u=rwx,g=rx,o=rx für dasselbe Ergebnis (Maske 022).

Hinweis: Die Umask wirkt auf Systemaufruf-Ebene: Standard-Dateirechte sind 666 (rw-rw-rw) und für Verzeichnisse 777 (rwxrwxrwx); die Maske wird davon abgezogen. Z.B. Umask 002 ergibt Dateien mit 664 (rw-rw-r-) und Verzeichnisse mit 775. Umask 077 würde sehr restriktiv alles nur für den Benutzer selbst erlauben (Datei 600, Dir 700).

chmod (Change Mode - Dateirechte ändern)

Beschreibung: Ändert die Zugriffsberechtigungen einer Datei oder eines Verzeichnisses. Man kann mit chmod sowohl numerisch (Oktalwerte) als auch symbolisch (u/g/o +-/= r,w,x) die Rechte setzen.

Syntax:

chmod [Optionen] <Modus> <Datei/Verzeichnis>...

Optionen:

Option	Beschreibung
-R, --recursive	Rekursive Rechteänderung: traversiert Unterverzeichnisse und wendet chmod auf alle darin befindlichen Dateien/Ordner an. Wichtig bei Verzeichnisbäumen.
-v, --verbose	Gibt für jede Datei eine Meldung aus, deren Rechte erfolgreich geändert wurden (und ggf. was geändert wurde).
-c, --changes	Wie verbose, aber nur Ausgaben für tatsächlich geänderte Rechte (keine Ausgabe, wenn Rechte schon so eingestellt waren).
-f, --silent	Unterdrückt Fehlermeldungen (force).
--reference=<Vorlage>	Setzt die gleichen Berechtigungen wie die Referenz-Datei <Vorlage> hat, auf die angegebenen Dateien. Statt Modus muss man --reference nutzen.

Modus-Angaben:

- **Numerisch (Oktal):** Eine dreistellige (oder vierstellige mit Sonderbit) Oktalzahl. Z.B. chmod 640 datei.txt setzt Rechte auf rw- r-- --- (6=rw,4=r,0=kein Recht). Jede Ziffer entspricht Owner, Group, Others. Beispiel: chmod 755 script.sh => rwx für Besitzer (7), rx für Gruppe (5), rx für andere (5). Eine führende vierte Ziffer kann spezielle Bits setzen: 4 für Setuid, 2 für Setgid, 1 für Sticky-Bit (z.B. 1755 für Sticky).
- **Symbolisch:** Format: [ugoa]*[+-=][rwxXst]* evtl. durch Kommata getrennt für mehrere Operationen in einem Aufruf. Beispiele:
 - chmod u+x,g-w,o= file – dem Besitzer Ausführungsrecht geben, der Gruppe Schreibrecht entziehen, andere bekommen exakt keine Rechte.
 - chmod a+r file – allen (a für all) Leserechte hinzufügen (ohne andere Bits zu ändern).
 - chmod g=u Datei – setzt die Gruppenrechte gleich den Benutzerrechten der Datei (nützliche Kurzform).
 - X: Spezialflag in symbolischer Notation – steht für "execute if directory or if any execute bit is set on any of owner/group/other". Wird meist bei -R verwendet, um rekursiv nur Verzeichnissen das x zu geben.
 - s für setuid/setgid (in u/g Kontext), t für Sticky-Bit (im o-Kontext).
 - Beispiel: chmod u+s script.sh – setzt das setuid-Bit auf die Datei (falls ausführbar).

chown (Change Owner)

Beschreibung: Ändert den Dateibesitzer und/oder die Gruppe einer Datei oder eines Verzeichnisses file-4fdk7rvx39azjretinarqb file-4fdk7rvx39azjretinarqb. chown kann sowohl den User als auch die Gruppe in einem Befehl setzen. Nur root (oder Capability CHOWN) darf Eigentümer von Dateien ändern. Normale Benutzer können in der Regel mittels chgrp die Gruppenzugehörigkeit ändern, wenn sie Mitglied der Zielgruppe sind (und Besitzer der Datei bleiben).

Syntax:

chown [Optionen] <neuerEigentümer>[:<neueGruppe>] DATEI...

Optionen:

Option	Beschreibung
-R	Rekursiv – wendet die Besitzänderung auf alle Dateien und Unterverzeichnisse an (ganzer Baum).
-h	Ändert den Besitzer von Symbolic Links selbst, statt des Ziels. (Auf den meisten Linux ist das Standard mit Kernel 2.6+; teils wird -h ignoriert, da chown auf Symlinks ohnehin Link selbst ändert, außer mit -L Option).
-f	Keine Fehlermeldungen ausgeben (silent/force).
-v	Verbose – listet jede Datei mit geänderten Owner/Group auf.
--	Ändert nur, wenn aktuelle Owner/Group genau wie angegeben ist. (Damit kann man selektiv bestimmte Kombis ersetzen, dient als Sicherheitsprüfung).
from=<old_owner:>old_group>	<i>Follow symlinks:</i> Symbolische Links auflösen (statt Link selbst zu ändern wird das Ziel geändert). Standardverhalten für die meisten chown-Versionen ist bereits, dem Link <i>nicht</i> zu folgen (außer man setzt -L).
--dereference / -L	<i>Don't follow symlinks</i> (explizit Link behandeln, Standard).
-P	Folgen von Symlinks <i>im Kommandozeilen-Argument</i> (Top-Level), aber nicht rekursiv tiefer. (Selten benötigt).
-H	

Verwendung:

- Besitzer ändern: chown alice datei.txt – setzt *alice* als neuen Besitzer der Datei (Gruppe bleibt unverändert).
- Gruppe ändern mit chown: chown :developers projektX – behält Besitzer, setzt Gruppe auf *developers*. Notation: Vor dem Doppelpunkt nichts heißt Besitzer unverändert, aber *Gruppe* ändern.
- Besitzer und Gruppe zugleich: chown alice:staff datei.txt – Besitzer = *alice*, Gruppe = *staff*.
- Nur Gruppe ändern könnte auch mit eigenem Befehl chgrp staff datei.txt gemacht werden.

Besonderheit: Man kann numerische IDs angeben. chown 1001:100 datei setzt Besitzer UID=1001, GID=100. Falls man einen Benutzer- oder Gruppennamen mit Doppelpunkt hat, der numerisch beginnt, kann man -- verwenden um Verwechslung zu vermeiden oder explizit username: schreiben.

chgrp (Change Group)

Beschreibung: Ändert die Gruppenzugehörigkeit einer Datei oder eines Verzeichnisses file-4fdk7rvx39azjretinarqbfile-4fdk7rvx39azjretinarqb. Äquivalent zu chown :<gruppe>. Wird benutzt, um den Gruppeneigentümer anzupassen.

Syntax:

chgrp [Optionen] <neueGruppe> DATEI...

Optionen:

Option	Beschreibung
-R	Rekursive Änderung der Gruppe in Unterverzeichnissen (alle Dateien/Dirs innerhalb werden angepasst).
-f	Keine Fehlermeldungen bei Problemen ausgeben (silent).
-v	Ausführliche Ausgabe pro Datei.
--reference=Vorlage	Übernimmt die Gruppenzugehörigkeit der <i>Vorlage</i> -Datei auf alle angegebenen Dateien. (Ignoriert das Gruppen-Argument und nutzt stattdessen die Gruppe der Referenz.)

Hinweis: Normale Benutzer dürfen chgrp nur verwenden, um eine Datei auf eine Gruppe zu setzen, der sie selbst angehören. Root kann beliebige Zuordnungen vornehmen.

lsattr (List Attributes)

Beschreibung: Listet die erweiterten *Dateiattribute* (Filesystem-Attribute) von ext-Dateisystemen auffile-4fdk7rvx39azjretinarqb. Diese Attribute wie *immutable* oder

append-only steuern besondere Verhalten auf niedriger Ebene (nur auf bestimmten FS wie ext2/3/4, XFS - mit anderen Tools - etc. vorhanden). lsattr zeigt pro Datei eine Zeichenfolge von Attribut-Flags.

Syntax:

lsattr [Optionen] [Datei/Verzeichnis]...

Optionen:

Option Beschreibung

- a** Zeigt auch versteckte Dateien (Dateien, die mit Punkt beginnen) an.
Standardmäßig ignoriert lsattr diese analog zu ls ohne -a.
- d** Listet das Verzeichnis selbst auf, nicht dessen Inhalt. (Bei Angabe eines Verzeichnisses wird normalerweise dessen Inhalt rekursiv aufgelistet; mit -d sieht man nur die Attribute des Verzeichniseintrags selbst)file-4fdk7rvx39azjretinarqb.
- R** Rekursive Ausgabe der Attribute in allen Unterverzeichnissen.
- v** Zeigt die Dateiversionsnummer (Version im ext-Format) mit an, falls vorhanden.
(Das Attribut 'version' wird kaum benutzt, -v gibt eine zusätzliche Spalte mit Versionsnummer).
- V** Ausführliche Informationen über die Version des Programms und detaillierte Fehler ausgeben (verbose).

Beispiel Ausgabe:

```
$ lsattr wichtige_datei  
----i-----e-- wichtige_datei
```

Hier bedeuten die Buchstaben: **i** = immutable (unveränderlich), **e** = extents (standard auf ext4). Bindestriche zeigen nicht gesetzte Flags. Wichtige Flags: **i** (immutable: Datei kann nicht verändert/gelöscht umbenannt werden bis Attribut entfernt), **a** (append-only: nur anhängen erlaubt), **d** (no dump: Datei wird von Dump-Backup ignoriert), **A** (no atime updates), **S** (synchronous updates) etc.

chattr (Change Attributes)

Beschreibung: Ändert die erweiterten Dateisystem-Attribute (wie immutable, append-only etc.) unter Linux auf ext2/ext3/ext4 Dateisystemenfile-4fdk7rvx39azjretinarqb. Nur Root darf die meisten dieser Attribute setzen.

Syntax:

chattr [Optionen] [+--Attribute] Datei...

Optionen:

Option	Beschreibung
-R	Rekursive Attributänderung in Unterverzeichnissenfile-4fdk7rvx39azjretinarqb.
-V	Ausführlich – zeigt an, welche Änderungen für welche Datei vorgenommen wurden.
-f	Unterdrückt Fehlermeldungen (force).
-v <Nummer>	Setzt die Versionsnummer der Datei (falls vom Dateisystem unterstützt). Selten benutzt – ext-Dateisysteme haben optional Dateiversionsnummern.

Attribute setzen (Symbolik): Man gibt nach möglichen Optionen eine Zeichenfolge an, der ein +, - oder = vorangestellt ist:

- + bedeutet diese Attribute *hinzufügen* (zusätzlich zu bestehenden).
- - bedeutet diese Attribute *entfernen*.
- = bedeutet *exakt setzen* (alle aktuellen Attribute durch diese ersetzen).

Die wichtigsten Attribut-Buchstaben auf ext4 sind:

- **i – immutable:** Datei kann nicht modifiziert, umbenannt oder gelöscht werdenfile-5y1edouysrow4zj6phjcqm. Prozesse können sie nicht öffnen im Schreibmodus. Schützt kritische Dateien (z.B. /etc/passwd) vor Änderungen, sogar von root (root muss erst chattr -i machen).
- **a – append only:** Datei kann nur angehängt, aber nicht überschrieben oder gelöscht werden. Typisch für Logfiles, damit niemand alte Einträge löscht.
- **e – extent format:** zeigt an, dass die Datei extents benutzt (automatisch auf ext4, nicht manuell gesetzt).
- **d – no dump:** Datei soll von Backup-Programmen, die dem Dump-Standard folgen, ignoriert werden.
- **A – no atime updates:** beim Lesen der Datei wird der atime-Zeitstempel nicht aktualisiert (Performance-Optimierung).
- **S – synchronous updates:** Änderungen an der Datei werden synchron auf den Datenträger geschrieben (wie bei chattr +S – ähnlich wie Ordner mit Dirsync mount option).
- **t – no tail-merging:** (relevant bei ext-Dateisystem, deaktiviert Tail-Merging bei Reverse-Mapping; selten benutzt).

- **T – top of directory hierarchy:** Verzeichnis wird als Top-Level für Ordnungszwecke in ext behandelt (verhindert bestimmte Ordnungsoptimierungen; selten).
- **u – undeletable:** Dateiinhalt soll beim Löschen inhaltlich gesichert werden, sodass man ihn wiederherstellen könnte. (Wird von ext4 i.d.R. nicht implementiert).

Beispiel:

- chattr +i datei.txt – setzt *immutable*.
- chattr -i datei.txt – entfernt dieses Flag wieder (dann ist Bearbeiten/Löschen wieder möglich).
- chattr =a Datei.log – setzt **nur** append-only und entfernt alle anderen Attribute (falls welche gesetzt waren). Danach kann in Datei.log nur noch angehängt werden.

getfacl (Get File ACL)

Beschreibung: Liest die Access Control Lists (erweiterten Zugriffsrechte) von Dateien oder Verzeichnissen aus und zeigt sie anfile-4fdk7rvx39azjretinarqb. Auf Linux-Systemen mit POSIX-ACL-Unterstützung (z.B. ext4 mit acl-Option eingehängt) kann man pro Datei feinere Berechtigungen für einzelne Benutzer oder Gruppen setzen. getfacl zeigt diese ACL-Einträge zusätzlich zu den Standardrechten.

Syntax:

getfacl [Optionen] Datei...

Optionen:

Option	Beschreibung
-a	Zeigt auch die <i>verschachtelten ACLs</i> (falls vorhanden – nicht üblich bei POSIX ACL).
-e	Zeigt die Ausgabe im <i>effective</i> Modus – Standard. (Jeder Eintrag inkl. "effective permissions", relevant wenn Mask gesetzt ist).
-s	Stripschutz: Zeigt nur die tatsächlichen ACL-Einträge, ohne die Grundeinträge (user/group/other).
-d	Zeigt die <i>Default-ACL</i> eines Verzeichnisses (die Vorlage, die an neu darin erzeugte Dateien vererbt wird). In der Ausgabe mit default: gekennzeichnet.
--omit- header	Unterdrückt den Kommentar-Kopf in der Ausgabe, der Dateinamen und Hinweise beinhaltet.

Option	Beschreibung
-R	Rekursiv über Verzeichnisse, getfacl für jede Datei auflisten.

Ausgabe-Beispiel:

```
$ getfacl projekt/
# file: projekt/
# owner: alice
# group: developers
user::rwx
user:bob:rwx-      # bob hat rw Rechte zusätzlich
group::r-x
mask::rw-          # effektive Maske für ACL = rw
other::---
default:user::rwx # Default ACL für neu angelegte Objekte
...
...
```

Hier sieht man die Standard-Owner/Group-Einträge und eine explizite ACL, die User bob Schreibrechte gibt, sowie eine *mask*, die die maximal wirksamen Rechte auf rw beschränkt.

setfacl (Set File ACL)

Beschreibung: Setzt oder ändert Access Control Lists (ACLs) von Dateien und Verzeichnissenfile-4fdk7rvx39azjretinarqb. Damit kann man zusätzliche Berechtigungen für bestimmte Nutzer oder Gruppen vergeben (oder entziehen), die über die klassischen Besitzer-Gruppe-Andere Rechte hinausgehen.

Syntax:

```
setfacl [Optionen] { -m|-x <ACL-Eintrag> ... | -b | -k } Datei...
```

Optionen:

Option Beschreibung

-m	<i>Modify</i> : Fügt einen ACL-Eintrag hinzu oder aktualisiert ihn. Beispiel: setfacl -m u:bob:rwx file gibt Benutzer <i>bob</i> rw-Rechte auf <i>file</i> . Mehrere -m Einträge oder
ACL	Einträge mit Komma getrennt sind möglich.

Option Beschreibung

- x **ACL** Remove: Entfernt einen spezifischen ACL-Eintrag. Z.B. setfacl -x g:marketing file entfernt den ACL-Eintrag für Gruppe "marketing".
 - b Remove all – Löscht *alle* ACL-Einträge (außer den Basis-Einträgen für Owner/Group/Other). Damit werden alle erweiterten Berechtigungen zurückgesetzt. (Datei hat dann nur noch die klassischen Unix-Rechte.)
 - k Remove Default – Entfernt die Default-ACL eines Verzeichnisses (vererbte ACL-Vorlage). Hat auf Dateien keine Wirkung.
 - R Rekursiv – wendet die ACL-Änderung auf alle Dateien im Verzeichnis (und Unterverzeichnisse) an. Nützlich um z.B. ganze Bäume eine neue ACL zu geben.
 - d Setzt Einträge als *Default-ACL* für Verzeichnisse. Benutzt in Kombination mit -m oder -x. Beispiel: setfacl -m d:g:team:rwx projekt setzt eine Default-ACL, so dass neue Dateien unter *projekt/* der Gruppe *team* rw Rechte erben, und Verzeichnisse rwx.
 - mask Berechnet und setzt die ACL-Maske basierend auf den aktuellen Gruppen-Einträgen neu. (Die Maske begrenzt die maximalen Rechte der ACL-Einträge außer Owner/other). Standardmäßig wird die Maske auto-anangepasst.
 - n Don't recalc mask: Verhindert, dass setfacl die ACL-Maske automatisch anpasst. Normalerweise soll man diese Option nicht verwenden, außer man setzt die Maske manuell mit -m m::....

Beispiel: setfacl -m u:bob:rw,g:marketing:r-- report.txt – gibt Benutzer *bob* Lese- und Schreibrecht und der Gruppe *marketing* nur Leserecht auf *report.txt*.

Hinweis: Nicht alle Dateisysteme unterstützen POSIX-ACL. Bei ext4, XFS etc. muss i.d.R. das Volume mit Option acl eingehängt sein (moderne Systeme haben das meist per Default). Mit getfacl kann man prüfen, ob vorhandene ACLs existieren.

Datei- und Verzeichnisverwaltung

stat (Dateistatus anzeigen)

Beschreibung: Zeigt detaillierte Meta-Informationen zu Dateien oder Dateisystemen an. stat gibt z.B. Größe, Blocks, Zeitstempel (Zugriff, Änderung, Statusänderung), Berechtigungen in Zahlenform, Device-ID, Inode-Nummer usw. in strukturierter Form aus.

Syntax:

stat [Optionen] <Datei>...

Optionen:

Option	Beschreibung
-f, --file-system	Zeigt anstelle der Datei-Details die Informationen zum Dateisystem, auf dem die Datei liegt. Z.B. Statistiken wie Blockgröße, Gesamtblöcke, freie Blöcke, FS-Typ etc.
-c FORMAT , --format=FORMAT	Gibt die Ausgabe gemäß <i>FORMAT</i> formatiert aus. Man kann Platzhalter verwenden, um bestimmte Werte auszugeben. Z.B. stat -c "%A %h %U %G %s %n" datei gibt in einer Zeile: modestr, Linkanzahl, Owner, Group, Size, Name. (Siehe man stat für alle Platzhalter wie %A = Rechte in rwx, %a = oktal, %U = Ownername, %s = Größe etc.)
--printf=FORMAT	Ähnlich wie --format, jedoch ohne automatisch Zeilenumbruch am Ende. Kann benutzt werden, um mehrere stat-Aufrufe zu skripten.
-t, --terse	"Knapp": Gibt die Ausgabe in einer kompakten einzeiligen Form, Felder durch Leerzeichen getrennt, aus (für einfache maschinelle Weiterverarbeitung).
--help, --version	Anzeigen von Hilfe bzw. Version.

Beispieldaten:

```
$ stat /etc/hosts
  File: /etc/hosts
  Size: 258    Blocks: 8      IO Block: 4096 regular file
Device: 802h/2050d Inode: 13422506  Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 0/  root)  Gid: ( 0/  root)
Access: 2025-04-01 10:22:13.000000000 +0200
Modify: 2025-04-01 10:22:13.000000000 +0200
Change: 2025-04-01 10:22:15.000000000 +0200
 Birth: -
```

Dies zeigt z.B., dass /etc/hosts 258 Bytes groß ist, auf einem bestimmten Device, inode etc., Rechte 0644, Eigentümer root, Zeitstempel für Zugriffs-/Änderungs-/Änderungszeit (Birth ist Erstellung, nicht unterstützt).

file (Dateityp bestimmen)

Beschreibung: Untersucht eine Datei und versucht den Typ bzw. Format der Datei zu bestimmen. file nutzt einen umfangreichen Magie-Pattern-Katalog

(/usr/share/file/magic), um anhand der Bytes einer Datei (Magic Numbers, Struktur etc.) menschlich lesbare Dateitypinformationen auszugeben. Praktisch, um z.B. Binärformate, Skripttypen oder Kodierungen zu erkennen.

Syntax:

file [Optionen] <Datei>...

Optionen:

Option Beschreibung

- Zeigt den MIME-Typ der Datei anstatt einer Beschreibung. Z.B. text/plain;
- i** charset=utf-8 statt "UTF-8 Unicode text". Hilfreich für Scripting oder Web-Kontext.
- Brief* – gibt nur die Beschreibung aus, ohne den Dateinamen voranzustellen.
- b** (By default schreibt file: "<Dateiname>: <Beschreibung>"). Mit -b erhält man nur "<Beschreibung>".
- Folgt symbolischen Links und untersucht das Ziel statt den Link selbst.
- L** Standardmäßig wird der Link selbst untersucht und meist als "symbolic link to <target>" ausgegeben.
- (alias --no-dereference) Bewirkt das Gegenteil: behandelt Symlinks wie
- h** normale Dateien (gleichbedeutend mit Standardverhalten, -h ist nützlich falls -L als default kompiliert wäre).
- Untersucht auch Gerätedateien (block/char special) und Auslagerungsdateien.
- s** Standardmäßig werden einige spezielle Dateitypen nicht inhaltlich analysiert (um z.B. nicht eine Device-Datei zu lesen). -s kann riskant sein (kann z.B. /dev/zero "lesen" -> blockiert).
- Durchsucht komprimierte Dateien (gzip-komprimierte) – versucht innerhalb eines .gz die Dateisignatur zu prüfen.
- E** Gibt (Exit)Status 1 zurück falls *irgendeine* der Dateien kein bekannter Typ bestimmt werden konnte. (Nützlich in Skripten).
- help**, **-v** Hilfe bzw. Versionsnummer und unterstützte Magic-Datei-Version ausgeben.

Beispiele:

- file /bin/ls -> **ELF 64-bit LSB executable**, x86-64, dynamically linked ...
- file archive.tar.gz -> **gzip compressed data**, was probably a tar archive (reads inside and erkennt es als tar in gz).

- file picture.jpg -> **JPEG image data**, JFIF standard 1.01, ...

ln (Link erstellen)

Beschreibung: Erstellt Verknüpfungen zu Dateien – entweder **Hard Links** oder **Symbolische Links**. Ein Hardlink ist ein weiterer Directory-Eintrag für dieselbe Inode (nur innerhalb desselben Dateisystems möglich). Ein Symbolischer Link (Symlink) ist eine kleine Spezialdatei, die auf einen Pfad verweist.

Syntax:

`ln [Optionen] <Ziel> [Linkname]`

Standard ohne Optionen erstellt *Hardlink* mit Namen *Linkname*, der auf Datei *Ziel* zeigt. Wenn *Linkname* nicht angegeben, wird im aktuellen Verzeichnis ein Link gleichen Namens wie *Ziel* erstellt. Für Symlinks Option *-s* nutzen.

Optionen:

Option	Beschreibung
-s, --symbolic	Erstellt einen <i>symbolischen Link</i> statt eines Hardlinks. Der Link verweist auf den angegebenen <Ziel>-Pfad. Wenn <Ziel> kein absoluter Pfad ist, wird er relativ vom Link aus interpretiert. Beispiel: <code>ln -s /etc/hosts hostslink</code> erzeugt Symlink <i>hostslink</i> -> <i>/etc/hosts</i> .
-f, --force	Existierende Zielfile (Linkname) ggf. entfernen, falls schon vorhanden. Ohne <i>-f</i> würde <i>ln</i> bei existierendem Linknamen einen Fehler ausgeben. (Bei <i>-s</i> mit <i>-f</i> : überschreibt existierenden Link.)
-n, --no-dereference	Beim Überschreiben eines existierenden symbolischen Links (mit <i>-f</i>) behandle den Link selbst statt dessen Ziel. (Verhindert, dass versehentlich die Datei, auf die ein bestehender Link zeigt, gelöscht wird).
-v, --verbose	Gibt eine Meldung pro verarbeitetem Link aus.
-T, --no-target-directory	Behandelt das letzte Argument immer als <i>genauen Linknamen</i> , selbst wenn es ein existierendes Verzeichnis ist. (Verhindert, dass <i>ln</i> im Zielverzeichnis einen Link mit Standardname erstellt.)
-d, -F, --directory	Erlaubt es root, auch <i>Verzeichnis-Hardlinks</i> zu erstellen. (Normalerweise verboten – Option nur auf speziellen Systemen mit <i>-f</i> kombinierbar, in der Praxis kaum verwendet).
-b, --backup	Erzeugt ein Backup der Zielfile falls <i>-f</i> zum Überschreiben genutzt wird. (Nützlich, um das ersetzte File nicht zu verlieren.) Siehe auch <i>--backup=t</i> etc.

Option	Beschreibung
-S <Suffix>	Gibt ein Suffix für Backup-Dateien an (Standard ~). Bsp: -b -S .bak.

Beispiele:

- Hardlink: ln /path/to/original.txt copy.txt – copy.txt ist nun ein weiterer Verweis auf original.txt (gleiche Inode). Änderungen an einer erscheinen in der anderen, da es dieselbe Datei ist. Löscht man eine, bleibt die andere erhalten (Datei wird erst gelöscht, wenn letzter Link entfernt).
- Symlink: ln -s /etc/apache2 siteconfig – erstellt siteconfig -> /etc/apache2. Aufruf von ls -l siteconfig zeigt dann siteconfig -> /etc/apache2.

cd (Change Directory) – siehe oben bei Grundlegende Befehle

(Im Kontext Datei/Verzeichnis-Verwaltung sei auf den bereits beschriebenen Befehl cd verwiesen, um Verzeichnisse zu wechseln.)

tree (Verzeichnisbaum anzeigen)

Beschreibung: Zeigt die Verzeichnisstruktur grafisch in Form eines Baumes an. tree listet rekursiv alle Unterverzeichnisse und Dateien in hierarchischer eingerückter Darstellung. Es erleichtert das Visualisieren der Verzeichnis-Hierarchie.

Syntax:

tree [Optionen] [Verzeichnis]

Optionen:

Option	Beschreibung
-a	Zeigt <i>alle</i> Dateien an, inkl. versteckter (auch solche, die mit . beginnen). Standardmäßig ignoriert tree versteckte Dateien, außer . und .. immer.
-d	Listet <i>nur Verzeichnisse</i> auf, keine Dateien. Nützlich, um reinen Verzeichnisbaum zu sehen.
-L Level	Beschränkt die Tiefe der Ausgabe auf <i>Level</i> Ebenen. Z.B. tree -L 2 zeigt nur zwei Ebenen tief (akt. Verzeichnis + Unterverzeichnisse, aber keine tieferen).
-f	Zeigt den vollen Pfad vor jedem Eintrag (statt nur Namen relativ).
-I	Ignoriert Einträge, die auf das Muster passen (Groß/Kleinschreibung <i>PATTERN</i>) berücksichtigt, mehrere Patterns mit `
-P	Zeigt nur Einträge an, die dem Pattern entsprechen (Whitelist). Umgekehrt zu <i>PATTERN</i> -I.

Option	Beschreibung
-F	Markiert Verzeichnisse mit abschließendem /, ausführbare Dateien mit *, Symlinks mit @, Pipes mit `
-p	Zeigt UNIX-Dateirechte (Permission bits) vor jedem Eintrag.
-u / -g	Zeigt Benutzer bzw. Gruppenbesitzer an. (Wie ls -l Infos).
-h	<i>Human-readable size</i> : Zeigt Dateigrößen in menschenlesbarem Format (K, M, G) anstatt in Bytes (wirkt mit -s).
-s	Zeigt Dateigrößen in Bytes neben jedem Dateinamen an.
--du	Summiert Verzeichnisgrößen wie du (statt wie normal lediglich die Einzeldateigröße zu zeigen).
-t	Sortiert die Ausgabe nach Zeit (Neueste zuerst) statt alphabetisch.
-o Datei	Gibt die Ausgabe in die angegebene Datei (überschreibt) anstatt auf stdout.
-n	Keine farbliche Hervorhebung (Standard ist bunt, -n erzwingt Monochrom).
--help	Hilfe anzeigen (Übersicht der Optionen).

Beispiel:

```
$ tree -L 2 /etc
```

```
/etc
├── apache2
│   ├── apache2.conf
│   ├── envvars
│   └── mods-available
├── cron.daily
│   ├── apt-compat
│   └── man-db
└── default
    ├── cron
    └── rcS
```

6 directories, 5 files

Hier sieht man /etc mit zwei Ebenen Tiefgang.

mkdir (Make Directory)

Beschreibung: Erstellt ein neues Verzeichnis (oder mehrere) im Dateisystem. Kann rekursiv auch mehrere Ebenen auf einmal anlegen.

Syntax:

`mkdir [Optionen] <Verzeichnisname>...`

Optionen:

Option	Beschreibung
-p, --parents	Erzeugt auch alle notwendigen übergeordneten Verzeichnisse, falls sie nicht existieren. Somit werden keine Fehler ausgegeben, wenn Verzeichnisse in der Pfadangabe fehlen. Beispiel: <code>mkdir -p /tmp/foo/bar</code> legt sowohl <code>/tmp/foo</code> (falls nicht da) als auch <code>/tmp/foo/bar</code> an.
-m, --mode=MODE	Setzt die Zugriffsrechte des neu erstellten Verzeichnisses auf <i>MODE</i> (numerisch oder symbolisch) anstatt der Standardrechte (die Standardrechte sind i.d.R. 0777 abzüglich Umask). Z.B. <code>mkdir -m 750 projDir</code> erstellt <i>projDir</i> mit rwx für Besitzer, rx für Gruppe, nichts für andere.
-v, --verbose	Ausgabe einer Meldung für jedes angelegte Verzeichnis (zeigt Pfad an).
--help, --version	Anzeige von Hilfe bzw. Version.

cp (Copy File)

Beschreibung: Kopiert Dateien oder Verzeichnisse. Bei Angabe mehrerer Quelldateien muss das letzte Argument ein Verzeichnis sein (Zielverzeichnis). cp kann auch Verzeichnisbäume rekursiv kopieren.

Syntax:

`cp [Optionen] <Quelle>... <Ziel>`

Wenn <Ziel> ein Verzeichnis ist, werden alle <Quelle>-Dateien dorthin kopiert (mit gleichem Namen). Ist nur eine Quelle und <Ziel> kein Verzeichnis, wird die Datei dorthin kopiert/umbenannt.

Optionen:

Option	Beschreibung
-r, -R, --recursive	Kopiert Verzeichnisse rekursiv (Inhalt und Unterverzeichnisse). Ohne diese Option kopiert cp standardmäßig keine Verzeichnisse. (GNU cp hat auch -r/-R – beide meist gleichwertig. POSIX fordert -R.)
-a, --archive	Archivmodus: Kopiert <i>rekursiv</i> und bewahrt dabei möglichst alle Attribute. Entspricht -dR --preserve=all (also -R, und -p für timestamps, owner, group, perms, ACLs, xattrs, Links etc.). Dies ist für Vollbackups ideal ionos.at .
-p, --preserve	Bewahrt Eigentümer, Gruppe, Berechtigungen und Zeitstempel der Datei beim Kopieren ionos.at . (Ohne -p übernimmt cp Standard: setzt aktuellen Zeitstempel "jetzt" und erstellt als aktueller Benutzer). Kann Argumente haben wie --preserve=mode,ownership,timestamps,links,xattr,all.
-f, --force	Überschreibt vorhandene Ziel-Dateien ohne Nachfrage, und versucht auch nicht-schreibbare Ziele zu entfernen, um Kopie dennoch zu platzieren. (Im Konflikt mit -n gilt letzter Schalter.)
-i, --interactive	Fragt nach, bevor eine existierende Zielfile überschrieben wird (Confirm Y/N). Gibt dem Benutzer die Möglichkeit, Überschreiben abzulehnen.
-n, --no-clobber	Überschreibt <i>keine</i> existierenden Dateien. D.h. falls Ziel existiert, wird die Kopie für diese Datei ausgelassen. (Verhindert unabsichtliches Überschreiben)
-u, --update	Kopiert nur, wenn <i>Quelle</i> neuer als <i>Ziel</i> ist, oder Ziel noch nicht existiert. Praktisch für inkrementelle Updates. Existiert Ziel mit gleich/neueren Datum, wird es nicht überschrieben.
-d, --no-dereference	Kopiert <i>Symbolic Links</i> als Links (anstatt deren referenzierten Inhalt). Standard: cp folgt symbolischen Links und kopiert die Datei dahinter. Mit -d wird der Link selbst neu erstellt am Ziel (Hardlinks bleiben hardlinks sofern möglich; -d ist in GNU cp gleichbedeutend mit --preserve=links).
-L, --dereference	<i>Folgt allen Symlinks:</i> kopiert stets den referenzierten Inhalt (Standardverhalten bei regulärem cp für Dateien). Bei Angabe von Verzeichnis mit -R: -L würde bewirken, dass cp <i>alle</i> gefundene

Option	Beschreibung
	Symlinks auflöst und deren Ziel kopiert (kann zu Loops führen, daher mit Vorsicht).
-P, --no-dereference --preserve=links	Kopiert Links als Links (das ist Standard mit -a enthalten). (Im GNU cp -P bedeutet No-dereference; beachten: -d und -P ähnlich).
--parents	Erhält die Quellpfadstruktur im Ziel. Bsp: cp --parents dir1/dir2/file.txt /backup legt /backup/dir1/dir2/file.txt an man7.org . Nur sinnvoll, wenn Ziel ein Verzeichnis ist.
-l, --link	Erstellt <i>Hardlinks</i> statt Kopien der Datei, sofern im selben Dateisystem. (Keine Datenkopie, nur neuer Name, ähnlich ln).
-s, --symbolic-link	Erstellt <i>Symlinks</i> statt die Daten zu kopieren. (Zieldatei wird symbolischer Verweis auf Quelldatei).
-H, -L, -P (bei -R)	Steuerung des Symlink-Verhaltens während rekursivem Kopieren: -P (Default) ignoriert Linkinhalt (kopiert Symlinks als Symlinks), -L folgt Symlinks zu Dateien, -H folgt nur Symlinks angegeben auf Befehlzeile. (-a impliziert -P).
-- backup[=Control]	Erzeugt Backup von Zieldateien, die überschrieben würden. (Etwa file.txt~ anlegen). Kann =simple, numbered etc. steuern. Siehe auch VERSION_CONTROL Env.
--help, --version	Hilfe/Version anzeigen.

mv (Move/Rename)

Beschreibung: Verschiebt Dateien oder Verzeichnisse an einen neuen Pfad, bzw. benennt sie um. mv kann auch zum einfachen Umbenennen einer Datei verwendet werden (Quell- und Zielpfad auf selbem FS = schneller Rename, kein Kopieren). Falls Quelle und Ziel auf unterschiedlichen Dateisystemen liegen, arbeitet mv durch Kopieren und anschließendem Löschen der Quelle.

Syntax:

mv [Optionen] <Quelle>... <Ziel>

Optionen:

Option	Beschreibung
-i, --interactive	Fragt nach Bestätigung, falls eine Zielfile durch den Move überschrieben würde (falls im Zielpfad eine Datei gleichen Namens existiert).
-f, --force	Erzwingt das Überschreiben ohne Nachfrage. Ziel-Dateien werden ohne Warnung ersetzt. (Standardverhalten ohne -i).
-n, --no-clobber	Überschreibt keine bestehenden Dateien. Bei Namenskollision bricht mv für diese Datei ab (gibt keinen Fehler, exit code trotzdem 0). In GNU mv hat -n Vorrang vor -i falls beide angegeben.
-u, --update	Bewegt/überschreibt nur, wenn die Quelle neuer ist als die vorhandene Zielfile <i>oder</i> die Zielfile nicht existiert. (So werden neuere Versionen moved, ältere Quellen, die schon am Ziel sind, bleiben unberührt.)
-v, --verbose	Gibt aus, welche Datei wohin verschoben/umbenannt wurde (für jede Operation eine Zeile).
-- backup[=CONTROL]	Wie bei cp: legt Backup der Ziel-Datei an, falls existiert, bevor sie überschrieben wird. CONTROL kann none, numbered, existing, simple sein (siehe Doku oder ENV VERSION_CONTROL).
--strip-trailing-slashes	Entfernt evtl. nachgestellte Schrägstriche bei den Pfadangaben der Quellen (um Verwirrung mit Verzeichnissen zu vermeiden). Standard: neuere mv machen das ohnehin.
--help, --version	Hilfe/Version.

Hinweis: mv kann mehrere Dateien in ein Zielverzeichnis verschieben oder eine einzelne Datei/Ordner umbenennen. Bei Verzeichnissen rekursiv, aber mv muss dafür nicht extra Option haben (im Gegensatz zu cp), da der Move innerhalb desselben FS nur Metadaten ändert. Bei verschiedenen FS kopiert mv im Hintergrund (rename() Syscall schlägt dann fehl, worauf cp+unlink passiert).

rmdir (Remove Directory)

Beschreibung: Löscht leere Verzeichnisse. Im Gegensatz zu rm entfernt rmdir nur Verzeichnisse und auch nur, wenn sie keine Dateien/Unterverzeichnisse mehr enthalten. Es kann optional ganze verschachtelte Verzeichnisbäume entfernen, sofern diese leer sind (Eltern werden dann rekursiv entfernt).

Syntax:

`rmdir [Optionen] <Verzeichnis>...`

Optionen:

Option	Beschreibung
-p, --parents	Entfernt das angegebene Verzeichnis und ggf. dessen übergeordnete Verzeichnisse, falls diese nach jedem Entfernen leer werden. Beispiel: <code>rmdir -p projekt/build/output</code> versucht erst <code>output</code> zu löschen. Ist danach <code>build</code> leer, wird auch <code>build</code> gelöscht. Ist danach <code>projekt</code> leer, wird auch <code>projekt</code> gelöscht.
-v, --verbose	Ausgabe einer Meldung für jedes gelöschte Verzeichnis.
--ignore-fail-on-non-empty	Unterdrückt die normale Fehlermeldung, falls ein Verzeichnis nicht leer ist. (Das Verzeichnis wird nicht gelöscht, aber es kommt kein Fehlerstatus; mit Vorsicht verwenden).

Nützlich: Um verschachtelte leere Ordner zu entfernen. Aber um Verzeichnisse mit Inhalt zu löschen, nutzt man `rm -r`.

rm (Remove Files/Directories)

Beschreibung: Löscht Dateien oder Verzeichnisse. Standardmäßig entfernt `rm` nur Dateien; mit `-r` kann es auch Verzeichnisbäume löschen (inkl. aller enthaltenen Dateien). **Vorsicht:** Gelöschte Dateien werden nicht ohne Weiteres wiederherstellbar gespeichert (kein Papierkorb, außer man nutzt safe-rm Tools).

Syntax:

`rm [Optionen] <Datei/Pfad>...`

Optionen:

Option	Beschreibung
-r, -R, --recursive	Löscht Verzeichnisse <i>rekursiv</i> mitsamt ihrem gesamten Inhalt. Damit können komplette Verzeichnisbäume entfernt werden. Ohne <code>-r</code> verweigert <code>rm</code> das Löschen von Verzeichnissen mit Fehler "is a directory".
-d	Löscht ein <i>leeres</i> Verzeichnis. (Ohne <code>-r</code> löscht <code>rm</code> normalerweise gar keine Verzeichnisse; <code>-d</code> erlaubt es aber für leere, ähnlich <code>rmdir</code> . Nützlich, aber <code>-r</code> deckt diesen Fall auch ab.)
-f, --force	<i>Force/ignore nonexistent:</i> Unterdrückt Rückfragen und Fehlermeldungen. Existiert eine angegebene Datei nicht, wird es ignoriert. Schreibschutz wird ignoriert (d.h. es fragt nicht nach Bestätigung bei schreibgeschützten

Option	Beschreibung
	Dateien). Kombiniert mit -r sehr gefährlich, da es kompromisslos alles löscht. Wird oft in Skripten genutzt.
-i	Interaktiv nachfragen vor <i>jedem</i> Löschvorgang. (Für viele Dateien unpraktisch).
-I	"ein wenig interaktiv": Fragt nur einmal pauschal nach, wenn mehr als 3 Dateien gelöscht werden sollen <i>oder</i> ein rekursiver Lösch auf einem Verzeichnis durchgeführt wird. (GNU rm Feature als Sicherheitsnetz)file-5y1edouysrow4zj6phjcqm.
--	Schützt vor rm -rf /. Diese Option ist in GNU rm standardmäßig aktiv – es
preserve-	verweigert das komplette Löschen des Root-FS. Man kann mit --no-
root	preserve-root überschreiben (nicht empfohlen).
-v, --verbose	Zeigt für jede Datei eine Meldung an, dass sie gelöscht wurde.
--help, --version	Hilfe bzw. Version.

Anwendung:

- **Einzelne Datei löschen:** rm datei.txt.
- **Mehrere Dateien löschen:** rm *.o (löscht alle .o Dateien im aktuellen Verzeichnis).
- **Verzeichnis und Inhalt löschen:** rm -r docs/ – löscht Ordner *docs* mitsamt allen Unterdateien.
- **Ohne Rückfrage alles löschen:** rm -rf tempdir. (Vorsicht!).

head (Dateianfang anzeigen)

Beschreibung: Gibt die ersten Zeilen einer Datei aus (standardmäßig 10 Zeilen). head ist nützlich, um schnell die ersten Zeilen (z.B. Header einer CSV, Beginn eines Logs etc.) anzuschauen, ohne die ganze Datei zu öffnen. Es kann auch binäre Dateien auf Anfang bytes prüfen.

Syntax:

head [Optionen] [Datei]...

Ohne Datei liest head von der Standardeingabe (z.B. in Pipes). Bei mehreren Dateien wird jeweils ein Header mit dem Dateinamen ausgegeben.

Optionen:

Option	Beschreibung
-n K , --lines=K	Gibt die ersten <i>K</i> Zeilen aus. <i>K</i> kann eine Zahl sein oder -K für alle bis auf die letzten <i>K</i> Zeilen (negativer Wert). Standard ist 10. Beispiel: head -n 20 file zeigt 20 Zeilen, head -n -5 file zeigt alle außer der letzten 5 Zeilen.
-c Bytes , *--bytes=B	Gibt die ersten <i>B</i> Bytes aus (statt Zeilen). Kann auch mit K, M, etc. Suffix (Potenzen von 1024) angegeben werden. Z.B. head -c 100K bigfile. Negative Werte analog: -c -10 alles außer letzte 10 Bytes.
-q , --quiet/-silent	Unterdrückt die Überschriften, die vor dem Inhalt stehen, wenn mehrere Dateien angegeben wurden. Normal zeigt head sowas wie "=> filename <==" vor dem Block jeder Datei. Mit -q wird alles aneinandergehängt ohne solche Banner.
-v , --verbose	Zeigt immer die Dateiüberschriften an, auch wenn nur eine Datei angegeben ist. (Normalerweise bei nur einer Datei keine Überschrift).
--help , --version	Hilfe bzw. Version ausgeben.

Beispiel: head -n 3 /etc/passwd zeigt die ersten 3 Zeilen der passwd-Datei (typisch Root, daemon, bin Accounts).

tail (Dateiende anzeigen)

Beschreibung: Gibt die letzten Zeilen einer Datei aus (standardmäßig 10 Zeilen). Sehr nützlich, um z.B. das Ende von Logfiles zu sehen. Kann im *Follow-Modus* (-f) genutzt werden, um fortlaufend neuen Inhalt anzuzeigen (Echtzeit-Log-Überwachung).

Syntax:

tail [Optionen] [Datei]...

Optionen:

Option	Beschreibung
-n K , --lines=K	Zeigt die letzten <i>K</i> Zeilen an. Negative Werte bedeuten ab einer bestimmten Zeile von Anfang an (ab Zeile K). Standard ist 10. Z.B. tail -n 50 /var/log/syslog zeigt die letzten 50 Zeilen.
-c B , *--bytes=B	Zeigt die letzten <i>B</i> Bytes an. Kann auch negative (ab Byte X) sein.
-f , --follow	Bleibt nach dem Anzeigen der letzten Zeilen geöffnet und zeigt fortlaufend neue Zeilen an, falls die Datei wächst (wie tail -f logfile zum Mitlesen).

Option	Beschreibung
-F	Ähnlich wie -f, aber zusätzlich "Wiederöffnen", falls Datei rotiert/neu erstellt wurde. (Follow Name statt Descriptor; praktisch für Logrotation – tail -F /var/log/app.log folgt auch nach Neustart des Logs dem neuen File).
-s Sek , --sleep-interval=Sek	Im Follow-Modus: Warte Sek Sekunden zwischen den Überprüfungen auf neue Daten. Standard 1.0 Sekunde. Kann z.B. auf 0.1 verringert werden, um schneller zu pollen, oder erhöht, um Last zu reduzieren.
--max-unchanged-stats=N	(Bei -F relevant) Wie oft in Folge unveränderte Inode geprüft wird, bevor tail davon ausgeht, dass ggf. Logrotate stattgefunden hat.
-q, --quiet	Unterdrückt Überschriften bei mehreren Dateien (analog head).
-v, --verbose	Erzwingt Überschriften, selbst bei einer Datei.
--help, --version	Hilfe bzw. Version.

Anwendung:

- **Letzte Zeilen einer Datei:** tail -n 20 messages.log (Zeigt die letzten 20 Zeilen).
- **Live-Verfolgung:** tail -f /var/log/syslog (Druckt neu geschriebene Logzeilen fortlaufend). Abbruch mit Ctrl+C.
- **Combine mit grep:** z.B. tail -f server.log \| grep "ERROR" filtert live nur Zeilen mit "ERROR".

Dateien finden und Text filtern

grep (Global Regular Expression Print)

Beschreibung: Durchsucht die Eingabe (Dateien oder Standard Input) nach Zeilen, die einem bestimmten Muster entsprechen, und gibt diese Zeilen ausfile-4fdk7rvx39azjretinarqb. grep arbeitet mit regulären Ausdrücken (Standard *Basic Regex*, erweiterte mit -E). Es ist das klassische Suchwerkzeug in Textstreams. Name stammt von der ed-Editor Anweisung g/re/p (global regex print).

Syntax:

grep [Optionen] 'Muster' [Datei...]

Optionen:

Option	Beschreibung
-i, --ignore-case	Ignoriert Groß-/Kleinschreibung beim Mustervergleichfile-4fdk7rvx39azjretinarqb. "Test" findet auch "test" oder "TEST".
-v, --invert-match	Gibt alle Zeilen aus, die nicht auf das Muster passen (Inverse Treffer)file-4fdk7rvx39azjretinarqb. Nutzt man oft zur Ausschlussfilterung.
-r, -R, --recursive	Durchsucht rekursiv alle Dateien in angegebenen Verzeichnissen (und Unterverz.)file-4fdk7rvx39azjretinarqb. Ohne diese Option behandelt grep Verzeichnisargumente normalerweise als Fehler oder ignoriert sie. (GNU grep: -r folgt Symlinks nicht, -R folgt Symlinks.)
-n, --line-number	Zeigt vor jeder gefundenen Zeile die Zeilennummer in der Datei an ibm.com . Format: Datei:Nr:Zeilentext.
-A NUM	(After) Zeigt <i>NUM</i> Zeilen nach jeder Trefferzeile zusätzlich an (Context) ibm.com .
-B NUM	(Before) Zeigt <i>NUM</i> Zeilen vor jeder Trefferzeile an.
-C NUM	(Context) Zeigt <i>NUM</i> Zeilen <i>vor und nach</i> jedem Treffer (gleichzeitig -A NUM -B NUM). Standard -C 2 beispielsweise gibt 2 Kontextzeilen um jede Fundstelle.
-l, --files-with-matches	Listet nur die Dateinamen auf, die mindestens einen Treffer enthalten. Pro Datei ein Name, Inhalt wird nicht ausgegeben. (Nützlich um zu sehen wo etwas vorkommt)
-L, --files-without-match	Listet nur Dateinamen, die <i>keinen</i> Treffer enthalten (Gegenteil von -l).
-c, --count	Zählt die Treffer pro Datei und gibt nur diese Anzahl aus (zusammen mit Dateinamen, außer mit -h). D.h. "file.txt:3". In Kombination mit -v zählt es Nicht-Treffer-Zeilen.
-o, --only-matching	Gibt nur den Teil der Zeile aus, der dem Suchmuster entspricht, nicht die ganze Zeile. Bei mehreren Treffern pro Zeile werden diese getrennt ausgegeben.
-q, --quiet/--silent	"Quiet": gibt keine Ausgabe; liefert lediglich Exit-Status 0 falls wenigstens ein Match gefunden wurde, 1 falls kein Match. Praktisch in Skripten (nur Rückgabewert interessiert).

Option	Beschreibung
-s, --no-messages	Unterdrückt Fehlermeldungen über nicht lesbare Dateien (z.B. wenn Zugriffsrechte fehlen oder Datei nicht existiert).
-E, --extended-regexp	Interpretiert das Muster als <i>Extended Regular Expression (ERE)</i> ionos.at . Das erlaubt z.B. Nutzung von +, ?, `
-F, --fixed-strings	Interpretiert das Muster als <i>fixen String</i> (kein Regex, jedes Zeichen literal). Entspricht fgrep (die eine schnellere Suche ohne regex ermöglicht). Alle Muster (wenn -e mehrfach) werden als Literalstrings gesucht.
-P, --perl-regexp	Interpretiert das Muster als Perl-Compatible Regex (PCRE), falls grep mit PCRE-Unterstützung kompiliert wurde. Damit stehen mächtigere Regex-Features zur Verfügung.
-h, --no-filename	Unterdrückt die Ausgabe des Dateinamens vor jeder Trefferzeile (wenn mehrere Dateien durchsucht werden). Standardmäßig präfixt grep Datei: bei mehreren Input-Dateien. Mit -h erscheinen nur die Zeilen. (Bei Einzeldatei ist das sowieso Standard.)
-H, --with-filename	Zeigt Dateinamen auch bei einziger Datei an. (Umgekehrt zu -h). Nützlich in Pipelines wo Source unbekannt.
--color[=WHEN]	Hebt das gefundene Pattern farblich hervor in der Ausgabe. WHEN kann auto (Standard, nur wenn Terminal), always oder never sein. Setzt Umgebungsvariable GREP_COLOR/GREP_COLORS ggf. benötigt.
-e Pattern	Gibt das Suchmuster explizit an. Nötig z.B. wenn das Pattern mit - beginnt (damit es nicht als Option interpretiert wird). Kann mehrfach angegeben werden, um mehrere Muster (OR-Verknüpfung) zu suchen.
-f Datei	Liest Suchmuster aus angegebener Datei (ein Pattern pro Zeile). Nützlich bei sehr vielen oder komplexen Patterns.
--exclude=GLOB	Schließt Dateien vom Durchsuchen aus, deren Name auf GLOB (Wildcard-Muster) passt. Z.B. --exclude=*.min.js würde Minimierte JS ignorieren. (Nur wirksam bei rekursiver Suche oder wenn Verzeichnis als Input). Es gibt auch --include analog.

Option	Beschreibung
--exclude-dir=GLOB	Wie --exclude, aber nur auf Verzeichnisnamen angewandt, um ganze Verzeichnisse von der Rekursion auszunehmen (z.B. --exclude-dir=.git).
--help, --version	Hilfe / Version.

Exit-Status: 0 falls mindestens ein Match gefunden, 1 falls kein Match, 2 falls ein Fehler auftrat (z.B. Datei nicht lesbar).

Beispiele:

- grep -i "error" *.log – sucht case-insensitive "error" in allen .log im aktuellen Ordner, gibt Trefferzeilen mit Dateipräfix aus.
- grep -R "TODO" src/ – findet rekursiv alle Vorkommen von "TODO" in Dateien unterhalb von src/ (typisch in Codebasis).
- grep -nE "^Error|Warning):" report.txt – sucht Zeilenbeginn "Error:" oder "Warning:" und gibt mit Zeilennummer aus [pq.hosting](#).
- ps ax | grep "[c]ron" – findet Prozesse mit "cron" im Namen. (Trick: [c] im Pattern matcht 'c', verhindert aber, dass der grep selbst im ps-Listing gefunden wird, da dessen Kommandozeile "grep cron" kein 'cron' substring hat sondern 'cro n' durch char class unterbrochen).

find (Dateien suchen)

Beschreibung: Durchsucht Verzeichnisse nach Dateien/Verzeichnissen, die bestimmten Kriterien entsprechen (Name, Typ, Größe, Datum, Rechte, Eigentümer etc.) file-4fdk7rvx39azjretinarqb. find ist sehr mächtig: man kann logische Bedingungen kombinieren und Aktionen auf gefundene Dateien ausführen (löschen, verschieben, Exec von Kommandos etc.). Es arbeitet rekursiv ab dem angegebenen Pfad.

Syntax:

find [Start-Verzeichnisse] [Optionen/Ausdrücke] [Aktion(en)]

Wichtige Suchkriterien (Ausdrücke):

- **-name Muster** – Dateiname entspricht *Wildcards* (Globbing wie bei Shell, z.B. "*.txt"). Groß/Kleinschreibung beachten.
- **-iname Muster** – wie -name, aber case-*insensitive*.
- **-type c** – Dateityp ist c. Häufig: **f** für regular file, **d** für directory, **l** symlink, **b** blockdevice, **c** chardevice, **p** named pipe, **s** socket. Z.B. -type d nur Verzeichnisse file-4fdk7rvx39azjretinarqb.

- **-user** *name/uid* – Datei gehört dem Benutzer *name* (oder UID numerisch). Analog **-group** *name/gid*.
- **-uid** *+n/-n/n* – UID der Datei ist (*>/</=*) *n*. **-uid +500** UID größer 500. (Entspricht **-user** nur numerisch und mit Vergleichen). Genauso **-gid**.
- **-size** *N*[bckMG]** – Dateigröße-Kriterium. *N* kann mit Vorzeichen angegeben werden (+ größer als, - kleiner als, nichts = genau). Suffix: standardmäßig in 512-Byte Blöcken, oder **c** = Bytes, **k** = KiB, **M** = MiB, **G** = GiB. Beispiel: **-size +100M** Dateien größer 100 MiB[wiki.foxtom.de](#), **-size -10k** kleiner 10 KiB.
- **-mtime** *Tage* – Letzte Änderung (Modify timestamp) liegt *Tage* zurück. *Tage* als ganze Zahl: **+n** = älter als *n* Tage, **-n** = jünger als *n* Tage, **n** = genau *n* Tage (im Bereich ±0.5d). Ähnlich **-atime** (Access), **-ctime** (Statusänderung). Z.B. **-mtime -7** in letzten 7 Tagen modifiziert[wiki.foxtom.de](#).
- **-newer** *Datei* – Datei wurde später geändert als *Datei* (Vergleich mtime). **-newermt** *Datum* (GNU find) erlaubt Vergleich mit einem Zeit-String, z.B. **-newermt "2025-05-01"**.
- **-perm** *mode* – Dateirechte entsprechen *mode*. Mode kann exakt (z.B. 644) oder mit / oder - angegeben werden für Masken. Beispiel: **-perm 644** findet genau 644. **-perm -644** findet Dateien, die mindestens diese Bits haben (also rw- r-- r-- oder mehr Rechte). **-perm /222** findet Dateien, bei denen irgendein Schreibbit gesetzt ist (Owner oder Group oder Other).
- **-maxdepth** *N* – Nur bis Tiefe *N* ins Verzeichnis hinab suchen (0= nur Startdir selbst). **-mindepth** *N* analog minimal tiefe. Z.B. **find . -maxdepth 1 -type f** nur Dateien im aktuellen Verzeichnis, nicht tiefer.
- **-mount** oder **-xdev** – Verlasse nicht das Dateisystem. D.h. keine Verzeichnisse auf anderen gemounteten FS durchsuchen (z.B. ignoriert **find / -xdev** dann andere Partitionen).
- **-prune** – Überspringt (schließt vom Durchsuchen aus) ein Verzeichnis. Bsp: **find . -path "./node_modules" -prune -o -name "*.js" -print** durchsucht alles außer den node_modules Ordner.

Aktionen (wenn Suchkriterien wahr sind):

- **-print** – Gibt gefundene Pfade aus (Standard, wird angenommen, wenn keine andere Aktion).
- **-ls** – Führt ls -dils auf die gefundene Datei aus (anzeige von detaillierten Infos, inkl. Inode, Größe, Rechte etc.).
- **-delete** – Löscht gefundene Dateien unmittelbar. Achtung: **find -delete** entfernt auch Verzeichnisse, wenn leer geworden. Sollte meist mit Kriterien kombiniert

werden (z.B. -type f -delete nur Dateien). **Keine** Aktion mehr nach -delete wird ausgeführt (beendet Durchlauf in diesem Pfad).

- **-exec Befehl {} ;** – Führt einen externen Befehl für jede gefundene Datei aus. {} wird durch den Pfad ersetzt. Die Kommandozeile wird so oft ausgeführt wie Treffer. Beispiel: find . -name "*.bak" -exec rm {} \; löscht alle .bak-Dateien (funktional wie -delete, aber universeller)file-5y1edouysrow4zj6phjcqm. Das \; markiert Ende des -exec Befehls.
- **-exec ... {} +** – Variante, die versucht, mehrere gefundene Dateien in *einen* Aufruf zu packen (fügt die Pfade an Stelle von {} möglichst gesammelt, wie xargs). Effizienter, falls das Programm mehrere auf einmal verarbeiten kann. Z.B. -exec chmod 644 {} + ändert die Rechte aller gefundenen Files, ggf. in batched Calls.
- **-execdir** – Wie -exec, aber führt das Kommando im jeweiligen Verzeichnis der Datei aus (Working Directory wechselt). Erhöht Sicherheit bei {} mit {} containing spaces.

Beispiele:

- find /var/log -type f -name "*.log" -size +1M -mtime +30 – findet Log-Dateien über 1MB, älter als 30 Tage unter /var/log.
- find . -maxdepth 1 -type d ! -name ":" -exec rm -r {} \; – löscht alle Unterverzeichnisse im aktuellen Ordner (maxdepth 1, alles was Type d ist und Name nicht ":"), dann rm -r).
- find /home/user -xtype l -delete – löscht gebrochene Symlinks (xtype l findet Links, deren Ziel nicht existiert).
- find . -perm /u+x -print – findet Dateien, bei denen das User-execute Bit gesetzt ist (also alle ausführbaren Dateien).
- find . -name ".*tmp" -execdir gzip {} \; – packt alle *.tmp im jeweiligen Verzeichnis in eine .tmp.gz (ausgeführt im Ort der Datei).

locate (Datenbank-Dateisuche)

Beschreibung: Durchsucht eine vorgefertigte Index-Datenbank nach Dateinamen (Pfaden), die einem angegebenen Suchstring entsprechenfile-4fdk7rvx39azjretinarqb. locate ist viel schneller als find, da es nicht live das Dateisystem scannt, sondern auf einer regelmäßig aktualisierten Datenbank (typisch via updatedb in Cron) basiert. Nachteil: findet keine erst sehr kürzlich hinzugefügten Dateien (bis zum nächsten updatedb-Lauf) und standardmäßig nur solche, auf die der User zugreifen darf.

Syntax:

locate [Optionen] <Suchmuster>

Das Suchmuster ist im Grunde ein Substring oder einfache Glob. Standardmäßig sucht locate alle Datenbankeinträge, die <Suchmuster> als Teilstring enthalten.

Optionen:

Option	Beschreibung
-i, --ignore-case	Sucht case-insensitive (Groß/Kleinschreibung ignorieren) file-4fdk7rvx39azjretinarqb.
-r, --regex	Interpretiert das angegebene Muster als regulären Ausdruck (POSIX ERE) statt als einfachen Textfile-4fdk7rvx39azjretinarqb.
-w, --wholename	(Standardverhalten) Sucht im gesamten Pfadnamen. (Bei einigen Implementierungen war locate nur substring-suche, wholename explizit wie default, hier nur Info.)
-b, --basename	Sucht nur im Dateinamen selbst, nicht im gesamten Pfad. (Also ignoriert Verzeichnisnamen im Match.)
-c, --count	Zählt nur die Anzahl der Fundstellen und gibt diese Zahl aus, anstatt alle Ergebnisse zu listen.
-l N, --limit=N	Gibt maximal N Ergebnisse aus (dann Abbruch).
--existing	Prüft gefundene Pfade und filtert jene heraus, die aktuell nicht (mehr) existieren. (Die DB könnte veraltete Einträge haben; mit --existing werden nur noch valide Treffer angezeigt, obwohl es Zeit kostet, jeden Pfad zu verifizieren.)
-0, --null	Trennt die Ausgabe mit Nullbytes anstatt neuer Zeile. Nützlich, um in xargs -0 weiterzuverarbeiten.
-e, --nofollow	Ignoriert Einträge, die Symlinks auf nicht-existente Dateien sind (köönnte Implementation-spezifisch).
-S	Zeigt Statistik der Datenbank (Anzahl Einträge etc.) statt zu suchen.
-d DB, --database=DB	Nutzt eine alternative Datenbank-Datei anstelle der Standard-Datenbanken (/var/lib/mlocate.db z.B.). Man kann mehrere via -d db1:db2:... angeben.
--help, --version	Hilfe bzw. Version.

Beispiel: locate -i '\.pdf' listet alle Pfade, die ".pdf" (unabhängig von Großschreibung) enthalten – das würde i.d.R. alle PDF-Dateien auf dem System listen, die zum Zeitpunkt

der letzten Datenbankaktualisierung existierten. Oft benutzt: locate <Programmname> um herauszufinden, wo ein Programm oder dessen Konfigdateien liegen.

updatedb (Update Database for locate)

Beschreibung: Baut die Datei-Datenbank für locate neu auf. Es durchsucht das Dateisystem (oder definierte Pfade) und speichert alle Dateinamen/Pfade in einer komprimierten Datenbank (typischerweise `/var/lib/mlocate/mlocate.db`). updatedb wird meist regelmäßig (täglich) via Cron aus geführt, man kann es aber auch manuell starten, z.B. nach größeren Änderungen im FS, damit locate sofort aktuell ist.

Syntax:

`updatedb [Optionen]`

Optionen: (Häufig werden diese über `/etc/updatedb.conf` voreingestellt)

Option	Beschreibung
--prune-bind-mounts <code>{yes no}</code>	Ob Bind-Mountpoints ausgelassen werden (Default ja – doppelte Scans vermeiden).
--prune- paths='PFADLISTE'	Doppelpunkt-separierte Liste von Verzeichnissen, die <i>nicht</i> indexiert werden sollen file-4fdk7rvx39azjretinarqb. Standardmäßig z.B. <code>/tmp, /proc, /sys, /run, /dev, /var/cache</code> etc., um unnötiges zu sparen. Dies entspricht PRUNEPATHS in <code>/etc/updatedb.conf</code> .
--prune- filesystems='TypList'	Liste von Dateisystemtypen, die nicht durchsucht werden (z.B. nfs,squashfs etc.). Standard meist Netzwerk-FS, Pseudo-FS (proc, sysfs) etc. file-4fdk7rvx39azjretinarqb. Entspricht PRUNEFS.
-l N	<code>--localpaths=LIST</code> – falls man nur lokale Pfade indexieren will (veraltet, use prune).
-U 'PFADLISTE'	Nur diese durch Doppelpunkt getrennten Pfade als Wurzel für Index verwenden (Standard "/" – gesamtes FS). Kann man nutzen um z.B. nur <code>/home</code> zu indexieren.
-o DBFILE	Output: speichere DB in angegebener Datei statt Standard. (So kann man separate DBs erzeugen.)
-f 'FS-Liste'	Nur diese Dateisystemtypen berücksichtigen (Inverse von PRUNEFS).
-v	Ausführliche Ausgabe während des Indizierens.

In der Regel wird updatedb ohne Optionen (ggf. als root über sudo updatedb) ausgeführt und nutzt die System-Voreinstellungen. Danach steht die Datenbank locate zur Verfügung.

which (Welches Kommando - Pfadsuche)

Beschreibung: Zeigt den Pfad zu einem ausführbaren Programm, das in der Umgebungsvariablen \$PATH gefunden wirdfile-4fdk7rvx39azjretinarqb. Damit kann man herausfinden, welches Programm aufgerufen würde, wenn man einen bestimmten Befehl tippt (nützlich bei mehreren Versionen in verschiedenen Verzeichnissen).

Syntax:

which [Optionen] Befehlsname...

Optionen:

Option	Beschreibung
-a	Zeigt <i>alle</i> Vorkommen des angegebenen Befehls im Suchpfad an (nicht nur das erste gefundene)file-4fdk7rvx39azjretinarqb. Z.B. which -a python listet evtl. /usr/bin/python und /usr/local/bin/python falls beide existieren.
--skip-dot , --skip-tilde	Ignoriert . bzw. ~ am Anfang des Pfades (also aktuelles Verzeichnis oder Home in PATH) in der Suche. (Sicherheitsaspekt – Standard bei which).
--show-dot , --show-tilde	Zeigt explizit, wenn gefundene Einträge in . oder ~ liegen, durch Präfix "./" oder "~/".
--tty-only	Zeigt nur etwas an, wenn stdout ein Terminal ist. Unterdrückt z.B. Ausgabe, wenn in Pipe (verhindert sich einzumischen).
--help , --version	Hilfe bzw. Versionsinfo (gilt bei GNU which; einige shells haben eingebautes which mit anderem Verhalten).

Hinweis: which ist oft eine Shell-Funktion/Alias und verhält sich u.U. leicht unterschiedlich. Ähnliche Tools: type (shell-builtin, zeigt auch Aliases, Keywords), command -v.

Beispiel: which gcc -> /usr/bin/gcc (Pfad zum C-Compiler). which -a ls könnte z.B. zeigen /usr/bin/ls und /bin/ls falls beides symlinks aufeinander oder verschieden, etc.

Textverarbeitung und Streams (Shell, Scripting)

awk (Muster/Aktions-Scanner)

Beschreibung: AWK ist ein mächtiges Textverarbeitungswerkzeug und Skriptsprache, mit dem man Textdateien zeilenweise nach Mustern durchsuchen und pro Zeile Aktionen ausführen kannfile-4fdk7rvx39azjretinarqb. Typischerweise wird awk benutzt,

um Spalten auszugeben oder zu filtern, da es Eingabezeilen automatisch in Felder zerlegt. Der Name stammt von den Anfangsbuchstaben seiner Schöpfer (Aho, Weinberger, Kernighan).

Syntax:

awk [Optionen] 'Muster { Aktion }' Datei...

Jede Zeile der Datei wird gelesen; wenn das *Muster* passt, wird die *Aktion* ausgeführt (z.B. etwas ausgeben). AWK hat eigene Variablen (\$1 = erstes Feld, \$0 = ganze Zeile) und eingebaute Rechen- und String-Features.

Optionen:

Option	Beschreibung
-F separator	Legt das Eingabefeld-Trennzeichen fest (Field Separator). Standard sind whitespace (Leerzeichen/Tabs)file-4fdk7rvx39azjretinarqb. Z.B. -F ":" für /etc/passwd mit Doppelpunkten.
-v Var=Wert	Setzt eine AWK-Variable <i>Var</i> auf <i>Wert</i> vor dem Verarbeiten. Z.B. -v ORS="\n\n" um eine Leerzeile nach jeder Ausgabezeile einzufügen (ORS = Output Record Separator).
-f Scriptdatei	Lies das AWK-Programm aus einer Datei anstatt aus der Befehlszeile. Damit kann man größere AWK-Skripte in eigenem File verwalten.
-m, --sandbox	(gawk) AWK in sicheren Modus schalten (kein System(), keine Schreibrechte auf FS etc).
--posix	Schaltet in POSIX-kompatiblen Modus (ggf. einige GNU-Erweiterungen aus).
-W version	Zeigt AWK-Version an (für gawk).
--help, --version	Hilfe / Version.

AWK Sprachkonzepte (Kurz):

- Ein AWK-Programm besteht aus *Muster { Aktion }* Blöcken. Muster ist ein Regex oder Ausdruck, Aktion ist Code in AWK.
- Felder: Eingabezeile wird in Felder zerlegt, zugreifbar als \$1, \$2, ..., \$NF (NF=Anzahl Felder). \$0 ist die ganze Zeile.
- Beispiele:
 - awk '{ print \$1 }' file – gibt die erste Spalte jeder Zeile aus (mit Standard-Trenn whitespace).

- awk '/error/ {print NR ":" \$0}' log.txt – sucht Zeilen, die "error" enthalten (Regex als Muster), und druckt Zeilennummer und kompletten Inhalt.
- awk -F, '\$3 > 100 { sum += \$3 } END { print "Summe:", sum }' data.csv – mit Komma als Separator summiere Wert der 3. Spalte aller Zeilen >100 in Spalte3; am Ende (END-Block) gib Summe aus.
- AWK hat viele eingebaute Funktionen (Mathe, String), assoziative Arrays, Kontrollstrukturen (if, for, while).

sed (Stream Editor)

Beschreibung: sed ist ein nicht-interaktiver Stream-Editor zum Bearbeiten von Texteingaben nach Mustern, insbesondere für Suchen&Ersetzen oder das Extrahieren von Zeilenfile-4fdk7rvx39azjretinarqb. sed liest zeilenweise Eingabe (Dateien oder STDIN), wendet die angegebenen *Editier-Befehle* darauf an und schreibt das Ergebnis auf STDOUT (oder optional in-place zurück in Dateien). Am verbreitetsten ist die *Substitution* mit dem s/alt/neu/ Kommando.

Syntax:

sed [Optionen] 'Script' Datei...

Das *Script* besteht aus einem oder mehreren Editierbefehlen, die auf die Eingabe angewendet werden. Beliebte sed-Befehle: **s** (substitute), **p** (print), **d** (delete), **q** (quit), **{ }** für Gruppen, Adressierungsmodi (Zeilen-Nr oder Regex).

Optionen:

Option	Beschreibung
-n, --quiet/--silent	Suppress normal output. Normal druckt sed jede Eingabezeile (ggf. modifiziert) automatisch. Mit -n wird nichts automatisch gedruckt, nur explizite p (print) Befehle geben ausfile-4fdk7rvx39azjretinarqb. Typisch in Kombination mit p in Script, um nur bestimmte Zeilen auszugeben.
-e Script	Gibt das zu verwendende sed-Skript (Editierbefehle) direkt an. Kann mehrfach genutzt werden für mehrere Befehle in Reihenfolge. Wenn nur ein Skript-Argument ohne -e gegeben ist, kann -e weggelassen werden.
-f Datei	Lies das sed-Skript aus einer Datei. Erlaubt komplexere Edits aus einem File anstatt Kommandozeile.
-i[SUF], --in-place[=SUF]	Bearbeitet Dateien <i>in place</i> , d.h. schreibt Änderungen direkt zurück in die Eingabedateien. Optional kann ein <i>Suffix</i> angegeben werden, um vor Änderungen ein Backup der Originaldatei anzulegen (z.B. -i.bak). Ohne Suffix wird die Datei direkt überschrieben ionos.de .

Option	Beschreibung
-r, -E	Schaltet die Verwendung erweiterter regulärer Ausdrücke ein (so dass in Patterns +, ?, {} etc. ohne Backslash verwendet werden können). Bei GNU sed ist -r (old syntax) und -E (wie bei BSD sed) identisch.
-s	Treat files separately. Normalerweise bei mehreren Dateien behandelt sed den Input als einen langen Stream. Mit -s wird nach jeder Datei der interne Zustand (z.B. Zeilennummer, \$-Regex-Flag) zurückgesetzt.
-z, --null-data	Input wird als Null-terminiert (Zeilen getrennt durch ASCII NUL statt newline) angenommen. Ermöglicht Verarbeitung von Binär oder multiline-chunks. Entspricht dann, dass . nicht auch \n matcht und \$ nur am NUL erkennt. (GNU sed)
--follow-symlinks	Wenn in-place auf Symlinks angewendet, ändert sed standardmäßig den Link selbst (also das Zielfile, aber eben durch den Link?). Mit dieser Option folgt sed dem Symlink und editiert die Ziel-Datei. (GNU sed)
--sandbox	Verhindert, dass sed w oder s///w zum Schreiben verwendet oder ! Shell-Aufrufe macht. (GNU sed 4.4+)
--help, --version	Hilfe / Version.

Häufige sed-Anwendungen:

- **Substitution (Ersetzen):**
Syntax: s/REGEX/REPLACEMENT/[Flags]. Ersetzt das erste Vorkommen von REGEX in jeder Zeile durch REPLACEMENT. Flags: g = global (alle Vorkommen in Zeile ersetzen), i = case-insensitive, p = print (selbst wenn -n), w file = gefundene Zeilen nach file schreiben.
Beispiel: sed -i 's/oldword/newword/g' file.txt – ersetzt *oldword* durch *newword* überall in file.txt, direkt in der Datei file-5y1edouysrow4zj6phjcqm.
Besonderheit: Im Ersatz kann man & verwenden für den gesamten Match oder \1 ... für Capturing Groups aus dem Regex.
- **Zeilen löschen:**
z.B. sed '/^#/ d' config – löscht alle Zeilen, die mit # beginnen (d = delete). Oder sed '5,10d' file löscht Zeilen 5 bis 10.
- **Nur bestimmte Zeilen ausgeben:**
Mit -n und p. Z.B. sed -n '1,5p' file druckt nur Zeilen 1-5 (andere werden durch -n nicht ausgegeben). Oder sed -n '/ERROR/p' log druckt nur Zeilen mit "ERROR".

- **Mehrere Edits:**

z.B. sed -e 's/foo/bar/' -e '/BAZ/d' – erst foo->bar ersetzen, dann Zeilen mit "BAZ" löschen. (Die Reihenfolge ist wichtig).

- **In-Place Backup:**

sed -i.orig 's/^/#/' *.conf – kommentiert alle Zeilen in *.conf aus, speichert vorher Backup mit .orig-Suffix.

Sed kann viel mehr (Insert/Append mit i\ und a\, Transform mit y///, Shell-Befehle mit !, Holding Space etc.), was in Tiefe für Prüfung wohl nicht nötig ist. Wichtig ist Suchen&Ersetzen und Basisfilter.

tee (Output verdoppeln)

Beschreibung: Liest die Standard-Eingabe und schreibt sie *gleichzeitig* auf die Standard-Ausgabe **und** in eine oder mehrere Dateienfile-4fdk7rvx39azjretinarqb. Es "verzweigt" den Datenstrom (daher der Name T-Stück). So kann man die Ausgabe eines Kommandos in eine Datei umleiten, aber trotzdem im Terminal sehen.

Syntax:

... | tee [Optionen] <Datei>...

Optionen:

Option	Beschreibung
-a, --append	Hängt die Ausgabe an bestehende Dateien an, anstatt sie zu überschreibenfile-4fdk7rvx39azjretinarqb. Standard ohne -a überschreibt Ziel-Dateien neu.
-i, --ignore-interrupts	Ignoriert Interrupt-Signale (SIGINT). D.h. Ctrl+C wirkt nicht auf tee selbst (nur auf vorheriges Kommando?), damit tee garantiert die Daten schreibt.
--output-error MODE	Verhaltensweise falls ein Schreibfehler auftritt: warn (Fehler melden, aber weiter versuchen andere outputs zu schreiben), exit (sofort beenden, evtl. ungeschriebene outputs gehen verloren). Standard ist warn.

Anwendung:

- make 2>&1 | tee build.log – Zeigt alle Ausgaben von make im Terminal und speichert sie gleichzeitig in *build.log*.
- ls -l | tee -a listing.txt – Ausgabe von ls -l normal anzeigen und ans Ende von listing.txt anfügen.

- echo "strict mode on" | sudo tee /proc/sys/kernel/yama/ptrace_scope – Durch sudo tee kann man Inhalte in Dateien schreiben, für die man Rechte braucht, während Eingabe evtl. aus einem un-privilegierten Echo kommt.

Tee ist also nützlich bei Logging und bei Verwendung mit sudo zum in Dateien schreiben, da sudo echo >> file wegen Redirection sonst nicht klappt.

xargs (Execute Arguments)

Beschreibung: Liest Eingaben (Standardinput oder Datei) und führt daraus Kommandoaufrufe zusammengebaut aus. Typisch, um die Ausgabe eines Programms als Argumentliste für ein anderes Programm zu verwendenfile-4fdk7rvx39azjretinarqbfile-4fdk7rvx39azjretinarqb. xargs nimmt zeilen- oder spaltengetrennte Wörter und packt möglichst viele davon in einen Aufruf eines angegebenen Kommandos (Standard: echo). Damit überwindet man die Shell-Längenbeschränkung und kann sequentiell Befehle auf Listen anwenden.

Syntax:

xargs [Optionen] [Befehl [Initial-Argumente]]

Wenn kein Befehl angegeben ist, benutzt xargs per Default echo (einfaches Ausgeben).

Optionen:

Option	Beschreibung
-n MaxArgs	Maximal <i>MaxArgs</i> Argumente pro aufgerufenem Kommando verwendenfile-4fdk7rvx39azjretinarqb. Z.B. -n 1 bedeutet, xargs nimmt jeweils 1 Argument und führt das Kommando damit aus (also für jede Eingabezeile einmal). Ohne Angabe bündelt xargs so viele Argumente wie möglich pro Aufruf, bis die Systemlimite (Getconf ARG_MAX) fast erreicht ist.
-L MaxLines	Nimmt maximal <i>MaxLines</i> Input-Zeilen pro Kommando-Aufruf. Z.B. -L 1 bewirkt, dass jede einzelne Zeile als eine xargs-Ausführung behandelt wird (selbst wenn Zeile mehrere Wörter enthält).
-d Delim	Verwendet <i>Delim</i> als Trennzeichen anstelle von Whitespace oder newline. Kann auch mehrere Zeichen (String) sein (GNU). Beispiel: -d ':' um : als Separator zu nehmen.
-0, --null	Erwartet Input-Felder durch Nullbyte getrennt (wie von find -print0 oder grep -Z erzeugt). Dies ermöglicht sichere Übergabe von Dateinamen mit Leer-/Sonderzeichen stackoverlow.com .

Option	Beschreibung
-E EOFSTR	Definiert eine Zeichenfolge, die als <i>Ende der Eingabe</i> interpretiert wird. Wenn xargs diese Zeichenfolge liest, bricht es die weitere Verarbeitung ab (auch wenn nicht EOF).
-e	(veraltet, äquivalent zu -E ohne Parameter) – historically to specify EOF marker, oft nicht mehr benötigt.
-I ReplaceStr	Nutzt <i>ReplaceStr</i> als Platzhalter in dem auszuführenden Kommando, der durch den gesamten Input (pro Aufruf) ersetzt wirdfile-
-J ReplaceStr	4fdk7rvx39azjretinarqb. Z.B. xargs -I {} mv {} /backup nimmt jeden Input (eine Datei) und substituiert anstelle von {} im mv-Aufruf. Bei -I wird implizit -L 1 angenommen (d.h. pro Zeile ein Aufruf).
-t, --verbose	(BSD xargs) Ähnlich -I, aber ersetzt nur einmal in der Kommandozeile (kann mit -n kombinieren). In GNU xargs gibt es -J nicht.
-p, --interactive	Gibt das Kommando, das ausgeführt wird, vorher auf stderr aus (also "echo" what is being done). Zum Debuggen hilfreich.
-r, --no-run-if-empty	Führt das Kommando nicht aus, wenn keine Argumente aus Input entstehen. Standard bei GNU xargs ist, <i>kein Leerlaufaufruf</i> zu machen (sprich diese Option ist implizit an). In POSIX könnte Standard anderes sein.
-0, --null	(siehe oben) Null-terminierte Eingabe verwenden – wichtig in Kombination mit find -print0 um sicher Dateilisten zu pipen ohne Probleme mit Sonderzeichen stackoverflow.com .
--max-chars=K, -s K	Maximal K Zeichen pro Kommandozeilen-Aufruf verwenden (inklusive Kommando und env). Standard an System ARG_MAX orientiert (oft ~131072). Manchmal zu reduzieren bei seltsamen Umgebungen.
--max-procs=N, -P N	Bis zu N Prozesse <i>parallel</i> ausführen statt nacheinander. Z.B. -P 4 führt bis zu 4 Instanzen des Kommandos gleichzeitig aus, um Parallelisierung zu erreichen. N=0 bedeutet unbegrenzt viele parallel (Vorsicht!). Hilfreich z.B. in Multi-Core-Systemen.
--process-slot-var=NAME	Setzt eine Umgebungsvariable \$NAME in den aufgerufenen Prozessen, die eindeutige ID des xargs-Slots enthält (0..N-1) – bei -P parallel. Hilft um Ausgaben zuzuordnen.

Option	Beschreibung
-a Datei	Liest Input aus <i>Datei</i> anstatt STDIN. So kann xargs auch ohne Pipe aus File speisen.
--help, --version	Hilfe/Version.

Beispiele:

- find /tmp -type f -name '*.txt' -print0 | xargs -0 rm -f – Löscht alle .txt-Dateien unter /tmp (sicher gegenüber Sonderzeichen, dank -print0 und -0).
- grep -l "ERROR" *.log | xargs -r zip errors.zip – Sucht in allen .log nach "ERROR", die -l Option listet nur Dateinamen mit Treffer. Diese Liste wird an xargs übergeben, das dann zip errors.zip <files> ausführt. Falls kein Treffer, -r sorgt dafür, dass zip nicht ohne Argument aufgerufen wird.
- cat list.txt | xargs -l {} echo File: {} >> files.txt – Für jede Zeile aus list.txt, führe echo File: <line> aus und append an files.txt. (So fügt man "File: " vor jede Zeile).
- seq 1 1000 | xargs -n 100 > numbers.txt – Nimmt die Zahlen 1 bis 1000 (jede auf neuer Zeile) und packt jeweils 100 Zahlen in eine Zeile, schreibt nach numbers.txt.
- printf "%s\\0" * | xargs -0 -n 1 -P 4 sha256sum – Berechnet parallel (4 at a time) die SHA256 Summen aller Dateien im aktuellen Verzeichnis. printf "%s\\0" * generiert Null-separierte Liste aller Dateien, xargs nimmt je 1 (-n1) und startet bis zu 4 sha256sum gleichzeitig.

Software- und Paketmanagement

tar (Tape ARchiver)

Beschreibung: Archiviert mehrere Dateien zu einer einzigen Datei (tarball). Tar kann Dateien **bündeln** (zu .tar Archiven) und in Zusammenarbeit mit Kompressionsprogrammen (gzip, bzip2, xz etc.) zugleich komprimierenfile-4fdk7rvx39azretinarqb. Oft verwendet, um Verzeichnisse oder viele Dateien als ein Archiv zu speichern oder zu verteilen (.tar.gz, .tgz, .tar.bz2 etc.).

Syntax:

tar [Optionen] -f <Archiv> [Dateien/Verzeichnisse...]

Tar hat **Modi** (Mutual exclusive Optionen): Erstellen, Extrahieren, Listen etc. Üblich ist die kombinierte Kurzoption wie -czf oder -xvf. Wichtig: -f (file) gibt an, welches Archivfile zu benutzen ist, und kommt als *ein* Argument, meist zuletzt bei Short-Options-Kombination.

Haupt-Modi:

- **-c, --create** : Erstelle ein neues Archiv aus den
(Fortsetzung)

tar (Tape Archiver)

Beschreibung: Erstellt oder entpackt Archive aus mehreren Dateien. tar bündelt Dateien in **einem** Archiv (Containerdatei mit Endung `.tarfile-4fdk7rvx39azjretinarqb】` . Oft wird tar mit Kompressionsprogrammen kombiniert (z.B. gzip oder bzip2), um komprimierte Archive wie `.tar.gz` zu erstellefile-4fdk7rvx39azjretinarqb】 .

Syntax:

`tar <Modus-Option> [Optionen] -f <Archivdatei> [Dateien/Verzeichnisse...]`

Tar benötigt **eine** der Hauptoptionen: -c (create), -x (extract) oder -t (list). Dazu wird **-f** angegeben, um den Namen der Archivdatei zu bestimmen. Häufig werden mehrere Kurzoptionen kombiniert (z.B. -czvf für "create, gzip, verbose, file").

Wichtige Optionen:

- **-c – create:** Neues Archiv erstellefile-4fdk7rvx39azjretinarqb】 .
- **-x – extract:** Dateien aus Archiv entpacken.
- **-t – list:** Inhalt des Archivs auflisten (nicht extrahieren).
- **-f Datei** – Gibt den Archivnamen an (Pfad/Name der Tar-Datei). Diese Option **muss** direkt gefolgt von dem Dateinamen kommen (z.B. -f archive.tar).
- **-v – verbose:** Ausführliche Ausgabe (listet während der Verarbeitung die Dateinamen auf).
- **-z – Filter durch gzip:** Bei -c wird das Archiv mit gzip komprimiert (*.tar.gz); bei -x wird erwartet, dass das Archiv gz-komprimiert isfile-4fdk7rvx39azjretinarqb】 . (Äquivalent: --gzip oder mit externem gzip pipen).
- **-j – Filter durch bzip2:** Erstellt/entpackt ein bzip2-komprimiertes Archiv (*.tar.bz2). (Alternativ --bzip2.)
- **-J – Filter durch xz:** Für `.tar.xz`-Archive (--xz).
- **-A – concatenate:** Hängt ein oder mehrere Archive an ein bestehendes Archiv an.
- **-d – diff/compare:** Vergleicht Archiveinträge mit Dateien im Dateisystem (prüft, ob Unterschiede bestehen).
- **-C DIR – Change Directory:** Wechselt ins Verzeichnis `DIR`, bevor Dateien hinzugefügt oder extrahiert werden[freedesktop.org】](http://freedesktop.org) . Z.B. `tar -xvf archive.tar -C /tmp` entpackt nach /tmp.

- **--strip-components=N** – Beim Extrahieren lässt tar N führende Pfadkomponenten weg. Z.B. entpackt /usr/local/share/doc/file als file bei --strip-components=3. Praktisch, um Archiv ohne Top-Level-Ordner auszupacken.
- **--exclude=PATTERN** – Schließt Dateien, die auf PATTERN passen, vom Archivieren aus. Z.B. --exclude="*.o".
- **--wildcards, --anchored** – steuert, wie Muster bei -t oder -x --wildcards -f archive "*.txt" interpretiert werden (glob-Muster an/aus etc.).
- **--numeric-owner** – Beim Entpacken Besitzer/Gruppen als IDs übernehmen (statt Namen auf dem Zielsystem zuzuordnen).

Beispiele:

- Archiv erstellen: tar -cvf archive.tar Verzeichnis/ erzeugt ein unkomprimiertes Archiv *archive.tar* mit allen Dateien aus *Verzeichnis/*.
- Mit Kompression: tar -czvf etc.tar.gz /etc packt ganz /etc als gzip-komprimiertes Archiv.
- Entpacken: tar -xvzf backup.tar.gz -C /restore entpackt *backup.tar.gz* nach */restore*.
- Auflisten: tar -tvf archive.tar zeigt die Dateiliste mit Größen, Timestamp, Rechten etc.

gzip (GNU Zip Compression)

Beschreibung: Komprimiert oder dekomprimiert Dateien im **gzip-Format** (.gz). Es ersetzt beim Komprimieren standardmäßig die Originaldatei durch die komprimierte Version (und fügt .gz anfile-4fdk7rvx39azjretinarqb】). Gzip ist schneller, aber erzeugt etwas größere Dateien als bzip2/xz. Häufig mit tar (siehe oben) eingesetzt.

Syntax:

gzip [Optionen] [Datei...]

gunzip [Optionen] [Datei...]

(gunzip ist äquivalent zu gzip -d.)

Optionen:

- **-c, --stdout**: Schreibt das komprimierte Ergebnis auf die Standardausgabe, anstatt die Datei zu ersetzen. Original bleibt erhalten (kann in Pipeline genutzt werden[ionos.de](#)】).
- **-d, --decompress**: Dekomprimiert die .gz-Datei (auch via gunzip aufrufbar[ionos.de](#)】).

- **-k, --keep:** Behält Originaldatei beim Komprimieren/Entpacken (standardmäßig wird Original gelöscht).
- **-r, --recursive:** Komprimiert alle Dateien in angegebenen Verzeichnissen rekursiv.
- **-1 bis -9:** Einstellen der Kompressionsstufe (1 = schnellste/geringste Kompression, 9 = langsamste/beste). Standard ist 6.
- **-t, --test:** Prüft eine gz-Datei auf Integrität, ohne sie zu entpacken (exit-Status zeigt OK/NOK).
- **-l, --list:** Zeigt Informationen über eine komprimierte Datei: Originalgröße, komprimierte Größe, Kompressionsrate, Zeitstempel.
- **--stdout, -c** (nochmal): Wird oft mit -d kombiniert, um nach stdout zu entpacken (z.B. gzip -dc file.gz).
- **-N, --name:** Bewahrt ursprünglichen Dateinamen und Zeitstempel (in .gz-Header) oder stellt diese beim Entpacken wieder her (ist Standardverhalten bei GNU gzip).
- **--help, --version:** Hilfe bzw. Versionsinfo.

Beispiele:

- gzip bigfile komprimiert *bigfile* zu *bigfile.gz* (und löscht *bigfile*).
- gzip -9 *.txt erstellt maximal komprimierte *.txt.gz Dateien.
- gunzip data.csv.gz entpackt *data.csv.gz* zurück zu *data.csv*.
- gzip -c file.txt > file.txt.gz komprimiert nach stdout und schreibt in separate Datei (Original bleibt).
- In Pipe: tar -cvf - /folder | gzip > folder.tar.gz – tar erzeugt Archiv auf stdout, gzip komprimiert es zu *folder.tar.gz*.

xz (LZMA Compressor)

Beschreibung: Komprimiert/Dekomprimiert Dateien mit dem **xz/LZMA2-Algorithmus**. XZ erreicht hohe Kompressionsraten, aber benötigt mehr CPU und RAM. Dateiendung: *.xz*. Wird z.B. für Kernel-Quellen-Archive genutzt. Syntax und Optionen ähneln gzip.

Syntax:

`xz [Optionen] [Datei...]`

`unxz [Optionen] [Datei...]`

(xz -d entspricht unxz.)

Wichtige Optionen:

- **-z** (Default): Komprimieren (Standardmodus, kann entfallen).
- **-d, --decompress**: Entpacken.
- **-k, --keep**: Originaldatei behalten (nicht löschen).
- **-f, --force**: Vorhandene .xz-Dateien überschreiben, falls vorhanden; oder erzwingt Komprimieren auch bei scheinbar nicht sinnvoll (z.B. sehr kleine Datei).
- **-9** (auch -0 .. -9): Kompressionsgrad (0 = none, 6 = Standard, 9 = max). XZ erlaubt auch -e für extreme Mode (noch etwas bessere Kompression, deutlich langsamere Kodierung).
- **-T N, --threads=N**: Nutzung von N Threads für Kompression (für große Dateien auf Multi-Core). Standard 1. (Für Dekomprimierung wird threads automatisch benutzt).
- **-c, --stdout**: Ergebnis auf stdout schreiben (bzw. von stdin lesen und auf stdout entpacken).
- **-l, --list**: Listet das .xz-Archiv (Inhalt, ratio, etc. – bei Single-File Komprimierung liefert original und comp size).
- **-q, -v**: quiet bzw. verbose Mode.
- **-C, --check=TYPE**: Prüfsummenverfahren wählen (Standard CRC64). Kann z.B. auf CRC32, SHA256 gestellt werden.

Beispiele:

- `xz hugefile` erzeugt `hugefile.xz` (Original wird gelöscht).
- `xz -dk archive.tar.xz` entpackt `archive.tar.xz` nach `archive.tar` und behält auch die .xz.
- `tar -c /usr/bin | xz -9e > bin.tar.xz` erstellt maximal-komprimiertes tar-Archiv.
- `unxz data.csv.xz` oder `xz -d data.csv.xz` entpackt Datei.
- `xzcat file.txt.xz = xz -dc file.txt.xz` entpackt nach stdout (ähnlich zcat).

bzip2 (Burrows–Wheeler Compressor)

Beschreibung: Komprimiert mit dem älteren bzip2-Algorithmus (.bz2). Langsamer als gzip, aber bessere Kompression; heute oft von xz abgelöst. Bedienung ähnlich gzip.

Syntax: `bzip2 [Optionen] [Datei...]` – bunzip2 zum Entpacken.

Optionen (Auswahl): -k (keep file), -d (decompress), -1 .. -9 (Stufe, Standard 9 bei bzip2), -c (stdout), -t (Test), -v (verbose stats).

zip (Archiv mit Kompression, ZIP-Format)

Beschreibung: Erstellt ZIP-Archive (mit Kompression) oder fügt Dateien hinzu. Anders als tar erwartet zip direkt die zu archiviernden Dateien und komprimiert sie einzeln ins Archiv. ZIP-Archive enthalten Verzeichnisstruktur + komprimierte Dateien (beliebtes Format unter Windows). unzip entpackt sie.

Syntax:

zip [Optionen] <Archiv.zip> <Dateien...>

Wichtige Optionen:

- **-r** – Rekursiv Verzeichnisse einschließen. Z.B. zip -r archiv.zip Ordner/.
- **-9** (bzw. -1 ... -9) – Kompressionsgrad (9 = beste, Standard meist 6).
- **-q** – quiet, **-v** – verbose (gibt Kompressionsraten aus).
- **-m** – move: Dateien nach dem Hinzufügen ins Archiv löschen (verschieben).
- **-j** – junk paths: Pfadnamen ignorieren, legt alle Dateien „flach“ im Archiv ab (ohne Verzeichnisstruktur).
- **-x <Pattern>** – bestimmte Dateien ausschließen (z.B. -x "*.o").
- **-F** – Versucht ein beschädigtes Archiv zu reparieren (freshen).
- **-u** – update: nur neuere Dateien (bzgl. Zeitstempel) ins Archiv aktualisieren.
- **-g** – grow: hängt Dateien an ein bestehendes ZIP an (statt neu zu erstellen).
- **-e** – erstellt ein verschlüsseltes Archiv (fordert Passwort-Eingabe, traditionelles ZipCrypto; -P <pw> unsicher ohne Prompt).

unzip Wichtige Optionen:

unzip <file.zip> entpackt im aktuellen Verzeichnis. -d <DIR> entpackt ins DIR. -l listet Inhalte. -t testet. -x schließt beim Entpacken bestimmte aus. -n überschreibt nichts, -o überschreibt ohne Nachfrage.

Paketmanagement-Befehle (Distributionen)

rpm (RPM Package Manager)

Beschreibung: Das Low-Level-Werkzeug zum Installieren, Entfernen und Abfragen von RPM-Paketen (Red Hat, CentOS, Fedora, SUSE etc.). rpm arbeitet auf .rpm-Dateien und der lokalen Paketdatenbank. Häufig nutzt man für höhere Ebene yum oder dnf, aber rpm bleibt wichtig für direkte Paketoperationen.

Syntax (Basis):

rpm [Optionen] <Paketdatei> / rpm [Optionen] <Paketname>

Operationen (Hauptoptionen, nicht kombinierbar):

- **-i – install** ein neues RPM-Paket . Beispiel: rpm -i nginx.rpm. Erfordert Root-Rechte.
- **-U – upgrade** (installiert neu oder aktualisiert älteres Paket). Wird häufig anstelle -i benutzt, da es sowohl installiert als auch updatet.
- **-F – freshen:** wie -U, aber nur wenn schon ältere Version installiert ist (sonst kein Install).
- **-e – erase** (deinstalliert) ein installiertes Paket nach Paketname . Beispiel: rpm -e nginx.
- **-q – query:** Abfrage-Modus (ohne Änderungen) . Kann mit Zusätzen Info liefern. Beispiel: rpm -q nginx prüft, ob Paket installiert ist und welche Version.

Wichtige Optionen/Zusätze:

Install/Upgrade Optionen:

- **-h** – zeigt Hash-Fortschrittsbalken beim Installieren ("#" -Zeichen).
- **-v** – verbose, mehr Ausgabe. Oft kombiniert -Uvh oder -ivh.
- **--force** – erzwingt Installation, selbst wenn z.B. ein Paket bereits installiert ist oder Dateien konfigurieren. (Vorsicht!).
- **--nodeps** – ignoriert Abhängigkeits-Prüfung (installiert trotzdem).
- **--replacefiles** – überschreibt Dateien, die zu anderen Paketen gehören könnten (Konflikte ignorieren).
- **--noshells** – führt keine Pre/Post-Install-Skripte im RPM aus (für spezielle Fälle).

Query Optionen (mit -q):

- **-a – query all** (listet alle installierten Pakete).
- **-i – (mit -q)** zeigt detaillierte Paketinformationen (Name, Version, Summary, Größe, Install-Datum etc. linux-praxis.de) . Z.B. rpm -qi bash.
- **-l – listet alle vom Paket installierten Dateien.** ` rpm -ql <paket>file-5y1edouysrow4zj6phjcqm` .
- **-p – query ein Paketfile** (nicht installiert). Z.B. rpm -qp -l paket.rpm zeigt, welche Dateien im RPM drin sind.
- **-f Dateipfad** – findet heraus, zu welchem installierten Paket eine bestimmte Datei gehört. Z.B. rpm -qf /usr/bin/ls -> coreutils-8.xx.
- **-c – listet nur Konfigurationsdateien des Pakets** (mit -ql kombinierbar: -qlc).
- **--changelog** – zeigt das Changelog des Pakets (sofern im RPM hinterlegt).

- **--provides, --requires** – zeigen welche Provides/Requires (Abhängigkeiten) ein Paket liefert bzw. benötigt. Z.B. rpm -q --requires httpd.
- **-s** – zeigt den Status der Dateien (ob sie verändert wurden seit Installation, (vom user modifiziert = **M**, missing etc.)). rpm -qs <paket>.

Erase (Uninstall) Optionen:

- **--nodeps** beim -e: ignoriert Abhängigkeitsprobleme (entfernt auch wenn andere Pakete darauf angewiesen sind - kann zu kaputten Programmen führen).

Beispiele:

- rpm -Uvh httpd-2.4.rpm – installiert Apache httpd, zeigt Hashtags und Namen.
- rpm -qa | grep kernel – listet alle installierten Pakete, filtert "kernel".
- rpm -qpi package.rpm – zeigt Infos über nicht installiertes RPM.
- rpm -e httpd – entfernt Apache-Paket.

yum (Yellowdog Updater Modified)

Beschreibung: Ein High-Level-Paketmanager für RPM-basierte Distr. (v.a. CentOS/RHEL 7 und älter). yum löst Abhängigkeiten automatisch auf und kann aus Repositories Pakete installieren, entfernen, aktualisieren . (In neueren RHEL8+ abgelöst durch dnf, aber Syntax ähnlich.)

Syntax:

yum [Optionen] <Sub-Befehl> [Paketnamen/...]

Haupt-Subkommandos:

- **install Paket** – Installiert ein Paket (aus den konfigurierten Repos oder einer RPM-Datei). Z.B. yum install httpd. Resolved dependencies automatisch und fragt um Bestätigung.
- **update [Paket]** – Aktualisiert alle installierten Pakete (wenn kein Name angegeben) oder ein bestimmtes. (yum update führt System-Update durch) .
- **remove Paket, erase** – Entfernt angegebene Pakete mit Abhängigkeiten, die dadurch verwiesen.
- **search Stichwort** – Durchsucht die Paketnamen, Beschreibungen etc. nach dem Stichwort (um herauszufinden, welches Paket etwas bietet).
- **info Paket** – Zeigt Infos zu Paket(en) (Name, Version, Summary, Größe, Repo, Abh.) ähnlich rpm -qi aber aus Repos wenn nicht installiert.

- **list [installed|available] <Pattern>** – Listet Pakete (installierte oder verfügbare). Ohne Parameter alle Pakete, optional mit Pattern filtern. Z.B. yum list available 'php*'.
- **provides Datei/Capability** – Findet heraus, welches Paket eine bestimmte Datei oder "Provide" bereitstellt (ähnlich rpm -qf/--whatprovides). Z.B. yum provides /usr/bin/zip.
- **clean <was>** – Bereinigt lokale Caches: z.B. yum clean all (löscht Metadaten- und Paketcache), oder yum clean metadata etc.
- **check-update** – Zeigt verfügbare Updates, ohne sie zu installieren (Exitcode 100 wenn Updates vorhanden, 0 wenn keine).
- **repolist** – Listet eingerichtete Repositories und ob aktiviert.
- **groupinstall / groupremove** – Installiert/entfernt ganze Paketgruppen (z.B. yum groupinstall "Development Tools").

Optionen (Auswahl):

- **-y** – Automatisch "yes" beantworten (nicht nach Bestätigung fragen). Wichtig für Skript】 .
- **-q** – Weniger Ausgabe (quiet).
- **-C** – Nur aus dem Cache arbeiten, keine neuen Metadaten laden (offline).
- **--enablerepo=X, --disablerepo=Y** – Repositories gezielt ein-/ausschalten für diesen Befehl (Overrides config).
- **--nogpgcheck** – Deaktiviert GPG-Signaturprüfung der Pakete (unsicher, nur falls Repo-Key nicht installiert etc.).
- **--downloadonly** – Lädt die Pakete herunter, installiert aber nicht.
- **--security** – (bei update/info/list) berücksichtigt nur sicherheitsrelevante Updates (wenn Repo das unterstützt).

Beispiele:

- yum install nginx php php-mysql – installiert mehrere Pakete in einem Rutsch.
- yum update – bringt System auf neuesten Stand.
- yum search "pdf viewer" – sucht nach Paketen mit "pdf viewer" im Namen/Beschreibung.
- yum remove httpd – deinstalliert Apache (und abhängige Module, wenn keine anderen es benötigen).
- yum provides "*bin/rar" – findet Paket, das rar-Binary liefert.

(Hinweis: Auf neueren Systemen ersetzt dnf den Befehl yum. dnf Syntax und Subcommands sind weitgehend gleich, aber z.B. dnf upgrade statt yum update.)

apt (Advanced Package Tool – Debian/Ubuntu)

Beschreibung: Höher-level Frontend für Debian-Paketverwaltung (dpkg). apt kombiniert Funktionen von apt-get und apt-cache in einem Kommando. Es verwaltet .deb-Pakete: Installation, Updates aus Repositories, Paket-Suche usw】. (Ubuntu 16.04+ und Debian Stretch+ empfehlen apt für interaktive Nutzung, während apt-get skriptfreundlich bleibt.)

Syntax:

apt [Optionen] <Sub-Befehl> [Pakete...]

Wichtige Sub-Befehle:

- **update** – Aktualisiert die Paketquellen-Datenbank (lädt neue Paketlisten von den konfigurierten Repos). Immer vor Install/Upgrade ausführen.
- **upgrade** – Installiert verfügbare Updates für alle **schon installierten** Pakete (führt keine Entfernung durch). Hält Pakete zurück, wenn Abhängigkeitswechsel nötig wären. (vgl. *apt full-upgrade*).
- **full-upgrade** – Führt vollständige Distribution-Upgrades durch, d.h. aktualisiert alle Pakete und darf auch Pakete entfernen oder neue Abhängigkeiten hinzufügen, um ein konsistentes System herzustellen (entspricht apt-get dist-upgrade).
- **install <pkg>...** – Installiert angegebene Pakete (aus Repos oder .deb Dateien via Pfad). Löst Abhängigkeiten auf und fragt bei Bedarf. Akzeptiert Paket=Version oder Paket:arch.
- **remove <pkg>...** – Entfernt Pakete, lässt jedoch veränderte Konfigurationsdateien bestehen (als *.config* in /etc).
- **purge <pkg>...** – Entfernt Pakete **inklusive** Konfigurationsdateien (bereinigt vollständig).
- **search <regex>** – Durchsucht Paketnamen und Beschreibungen nach einem Begriff (regex möglich). Beispiel: apt search editor pdf.
- **show <pkg>** – Zeigt detaillierte Informationen zu einem Paket (Version, Abhängigkeiten, Beschreibung, Maintainer etc.), egal ob installiert oder nur verfügbar】.
- **list** – Listet Pakete. Mit --installed, --upgradeable oder --all-versions. (Ähnlich dpkg -l aber filterbar). Z.B. apt list --installed | grep nginx.

- **autoremove** – Entfernt automatisch Pakete, die als Abhängigkeiten installiert wurden und nun nicht mehr benötigt werden (Waisen).
- **depends <pkg>, rdepends <pkg>** – Zeigt Abhängigkeiten bzw. Reverse-Abhängigkeiten eines Pakets (welche Pakete benötigen es).
- **edit-sources** – Öffnet die sources.list im Editor (um Repos zu bearbeiten).

Optionen:

- **-y, --yes** – Automatisch alle Fragen mit Yes beantworten (nicht interaktiv bestätigen). Praktisch für Skripte.
- **-q** – Weniger Ausgabe (bis zu -qq für gar keine).
- **--no-install-recommends** – Installiert nicht automatisch als "empfohlen" markierte Pakete. Standard apt installiert Recommends mit.
- **--allow-remove-essential, --allow-change-held-packages** – Erlaubt riskante Änderungen (Standard verhindert z.B. Entfernen essentieller Pakete).
- **-s, --simulate** – Simulation: Zeigt an, was würde passieren, aber führt keine Änderung aus (auch Exitcode 0 immer). Gut zum Test.
- **-V, --verbose-versions** – Zeigt bei Auflistungen detailliertere Versionsinfos.
- **--help** – Hilfe-Seite, **--version** – Version von apt.

Beispiele:

- apt update && apt upgrade – Aktualisiert Paketlisten und installiert alle verfügbaren Updates (ohne Dist-Upgrade).
- apt install build-essential – Installiert eine Paketgruppe (hier C/C++ Compiler Toolchain).
- apt remove --purge openssh-server – Entfernt SSH-Server komplett mit Config.
- apt search docker – Sucht alle Pakete mit "docker" im Namen/Beschreibung.
- apt full-upgrade – Aktualisiert System auch über Distribution-Upgrades hinweg (entspricht auf Ubuntu einem Release-Upgrade, sofern Repos angepasst).

(*Hinweis: apt-get und apt-cache sind Vorgänger: z.B. apt-get install, apt-get remove, apt-get dist-upgrade, apt-cache search, apt-cache show. apt vereinfacht den Alltag, aber Skripte nutzen weiterhin oft apt-get -y etc. dpkg siehe unten ist noch eine Stufe tiefer für einzelne .deb ohne Abhängigkeitsauflösung.*)

dpkg (Debian Package Manager - Low-Level)

Beschreibung: dpkg ist das unterliegende Werkzeug auf Debian/Ubuntu, das .deb-Pakete entpackt, installiert, konfiguriert und entfernt. Es löst keine Abhängigkeiten –

dafür sind apt/apt-get zuständig. Man verwendet dpkg meist für manuelle Offline-Installation von .deb-Dateien.

Syntax:

```
dpkg -i <Paket.deb>      # Installieren  
dpkg -r <Paketname>      # Entfernen  
dpkg -P <Paketname>      # Entfernen + Purge  
dpkg -L <Paketname>      # Liste Dateien des installierten Pakets  
dpkg -s <Paketname>      # Statusinfo (ähnlich apt show)
```

Wichtige Optionen/Parameter:

- **-i, --install Datei.deb:** Installiert das angegebene Deb-Paket】 . Gibt Fehler aus, falls Abhängigkeiten fehlen. (Diese müssen vorher mit apt installiert werden).
- **-r, --remove Paketname:** Entfernt ein installiertes Paket (Paketname ohne ".deb", wie in dpkg -l gelistet】 . Konfigdateien unter /etc bleiben bestehen.
- **-P, --purge Paketname:** Entfernt Paket inkl. seiner Konfigurationsdateie】 .
- **-S, --search Dateiname:** Sucht, welches installierte Paket eine bestimmte Datei besitzt (ähnlich rpm -qf). Z.B. dpkg -S /usr/bin/foo.
- **-L, --listfiles Paketname:** Listet alle Dateien, die das installierte Paket auf dem System installiert hatfile-5y1edouysrow4zj6phjcqm】 .
- **-l, --list Muster:** Listet Pakete, optional gefiltert nach Muster. dpkg -l ohne Filter zeigt eine Tabelle aller Pakete und deren Status (ii = installiert, rc = removed config-files left, etc.).
- **-s, --status Paketname:** Zeigt den Status und Informationen eines installierten Pakets (Version, Beschreibung, Maintainer, Abhängigkeiten...).
- **--configure Paketname:** Führt die Konfigurationsschritte (dpkg-configure) für ein installiertes aber noch nicht konfiguriertes Paket aus. (Wird z.B. nach entpacken aufgerufen).
- **-B, --auto-deconfigure:** Beim Entfernen eines Pakets, das Abhängigkeiten bereitstellt, markiert abhängige Pakete als "deconfigured".
- ****--force-**X:** Viele Force-Flags, um dpkg anzuweisen, bestimmte Fehler zu ignorieren (z.B. --force-dependends ignoriert fehlende Abhängigkeiten, --force-confold/--force-confnew bei Config-Datei-Konflikten etc.). Vorsicht: Kann System inkonsistent machen.

- **--print-architecture**: Zeigt die Architektur, für die dpkg Pakete installiert (z.B. amd64).
- **--add-architecture <arch>**: Fügt Multiarch-Unterstützung für andere Arch hinzu (z.B. i386 auf einem amd64 System).

Beispiele:

- dpkg -i google-chrome.deb – Installiert Chrome .deb-Paket (scheitert, wenn Abhängigkeiten fehlen – in dem Fall via apt --fix-broken install auflösbar).
- dpkg -r postfix – Entfernt postfix, belässt aber ggf. geänderte /etc/postfix Dateien.
- dpkg -P postfix – Entfernt postfix vollständig inklusive Config.
- dpkg -L bash – Listet alle vom *bash*-Paket installierten Dateien.
- dpkg -S /bin/ls – Welches Paket stellt /bin/ls bereit? (Antwort: coreutils).

wget (Web GET Utility)

Beschreibung: Ein Kommandozeilen-Downloader für Dateien über HTTP, HTTPS und FTP. wget kann in Skripten genutzt werden, um Dateien aus dem Web herunterzuladen, unterstützt Fortsetzen abgebrochener Downloads, rekursives Herunterladen von Websites etc. Es funktioniert nicht-interaktiv (kein Browser, nur Download).

Syntax:

wget [Optionen] <URL>

Wichtige Optionen:

- **-O Datei** – Speichert die heruntergeladene URL in die angegebene *Datei* (statt den Namen aus URL zu verwenden). Z.B. wget -O ubuntu.iso http://.../ubuntu.iso.
- **-c, --continue** – Setzt einen abgebrochenen Download fort (sofern Server Range Requests unterstützt). Prüft, wieviel schon vorhanden ist und lädt den Rest.
- **-q** – Quiet-Modus (keine Ausgabe). **-nv** (not verbose) eine dazwischenliegende Stufe. **-v** für ausführlicher (Standard).
- **-r, --recursive** – Rekursiver Download einer Website. Erfordert oft zusammen mit -np (no parent), -L (stay on same domain), -k (Links anpassen für Offline). Z.B. wget -r -np -k http://example.com/docs/.
- **-np, --no-parent** – Geht beim rekursiven Download nicht über das Startverzeichnis hinaus.
- **-l <Tiefe>** – Legt die Rekursionstiefe fest (Default 5, -l0 = unendlich).

- **-p, --page-requisites** – Lädt alles mit, was nötig ist, um HTML-Seite offline darzustellen (Bilder, CSS...).
- **-N, --timestamping** – Lädt eine Datei nur herunter, wenn sie neuer ist als die lokale (verwendet Last-Modified). Gut zum Aktualisieren.
- **--limit-rate=<Rate>** – Begrenzt Downloadrate, z.B. --limit-rate=200k (200 KB/s). Hilfreich, um Bandbreite zu drosseln.
- **-U, --user-agent=<Agent>** – Setzt einen bestimmten User-Agent-String. Standard ist "Wget/Version". Kann geändert werden, um z.B. nicht blockiert zu werden.
- **--no-check-certificate** – Ignoriert SSL-Zertifikatfehler (z.B. selbstsigniert).
- **--user=<user> --password=<pw>** – Falls erforderlich, für HTTP-Auth oder FTP-Login. (Achtung: Sichtbar in Prozessliste).
- **-i <Datei>** – Liest eine Liste von URLs aus <Datei> (jede Zeile eine URL) und lädt alle herunter.
- **-b, --background** – Startet im Hintergrund (nur auf Unix, leitet Ausgabe nach wget-log).
- **-P <Verz>** – Speichert Dateien in angegebenem Verzeichnis (statt aktuelles).

Beispiele:

- wget http://example.com/file.zip – Lädt *file.zip* in aktuelles Verzeichnis herunter.
- wget -c http://largefile – setzt Download fort, falls vorhanden.
- wget -r -np -N http://example.com/photos/ – Lädt Verzeichnis "photos" vollständig herunter, aktualisiert nur neue Dateien beim Wiederaufruf.
- wget -O - https://api.example.com/data | jq . – Lädt Daten von API und pipet direkt in jq zur Verarbeitung (hier -O - bedeutet Ausgabe auf stdout statt Datei).

curl (Client URL)

Beschreibung: curl ist ein umfangreiches Kommandozeilen-Tool für Datenübertragungen mit URL-basierten Protokollen (HTTP, HTTPS, FTP, etc.). Im Gegensatz zu wget liegt Fokus auf Flexibilität bei HTTP-Requests (Header setzen, verschiedene HTTP-Methoden, Formulare absenden, APIs ansprechen etc.). Es eignet sich für Web-APIs (GET/POST) und kann ebenfalls Dateien herunterladen oder hochladen.

Syntax:

curl [Optionen] <URL>

Wichtige Optionen:

- **-o Datei** – Speichert die Ausgabe in *Datei*. (Kleinbuchstabe o, analog wget -O).
- **-O** – Speichert unter dem **Originalnamen** aus der URL (Großes O, analog wget ohne -O). Achtung: curl -O http://server/dir/file.txt legt *file.txt* an.
- **-L** – *Location folgen*: folgt HTTP-Redirects (Status 3xx) automatisch [freedesktop.org](#)]. (wget macht das default, curl nicht ohne -L).
- **-C --** Setzt abgebrochenen Download fort (wenn Server unterstützt; '-' heißt automatischen Offset ermitteln).
- **-#** – Fortschrittsleiste im CLI (oder --progress-bar). Standard curl zeigt laufende Stats Zeile, -# macht wie wget progressive Bar.
- **-s** – Silent (kein Progress oder Fehlerausgaben). **-S** kombiniert mit -s zeigt Fehler dennoch an.
- **-v / -vv** – Verbose Mode (zeigt detaillierte Request/Response, Header etc. -vv noch mehr, inkl. Low-level). Gut zum Debuggen.
- **-I** – Führt eine HEAD-Anfrage aus (nur Header vom Server holen, keine Body-Daten). Nützlich um Meta-Informationen (Content-Length, Last-Modified) abzufragen.
- **-X METHOD** – Explizite HTTP-Methode festlegen (z.B. PUT, DELETE). Standard bei Datenvorhandensein oder -d ist POST, sonst GET.
- **-G** – Erzwingt, dass bei Verwendung von -d (Daten) diese als Query-Parameter an URL angehängt werden (statt POST Body).
- **-d Daten** – Sendet die angegebenen *Daten* im HTTP-Request-Body (Standard via POST). Z.B. curl -d "user=test&pass=123" http://.../login. Implizit -X POST wenn nicht -G.
 - **--data-urlencode Daten**: URL-encodet die Daten vor Versand (z.B. wenn Daten Sonderzeichen enthalten).
 - **--data-raw**: ähnlich -d, aber ohne spezielle Form-Codierung (sendet genau so).
- **-H "Header: Wert"** – Fügt einen HTTP-Header hinzu oder überschreibt ihn. Z.B. -H "Authorization: Bearer <token>". Mehrfach nutzbar für mehrere Header.
- **-u Benutzer:Pass** – Nutzt HTTP Basic Auth mit gegebenen Credentials (wird in Header "Authorization: Basic ...Base64" codiert [freedesktop.org](#)]. Vorsicht in Shell-History. Kann auch nur -u user (dann fragt curl interaktiv Passwort ab). Funktioniert auch für FTP.
- **-F Feld@Datei** – HTTP Datei-Upload via *multipart/form-data*. Z.B. curl -F "file=@bild.png" -F "desc=Urlaubsfoto" http://.../upload.

- **-K Konfigdatei** – Liest curl-Parameter aus Datei (kann nützlich sein für komplexe Setups).
- **-b CookieDatei** – Sendet Cookies aus angegebener Datei (. Netscape cookie format oder einfach "name=value" Paare). Kann auch -b "name=value" um Cookies zu setzen.
- **-c CookieAusgabe** – Speichert Cookies, die vom Server kommen, in angegebener Datei (um sie später wieder zu verwenden).
- **--limit-rate <Speed>** – Begrenzt Übertragungsrate (z.B. --limit-rate 100k).
- **--retry N** – Anzahl der automatischen Wiederholungsversuche bei Fehler (z.B. Timeout) oder --retry-all-errors.
- **--cacert Datei / --insecure:** Spezielles SSL: anderes CA-Zertifikat nutzen bzw. bei --insecure SSL-Zertifikat überprüfen ignorieren (wie wget --no-check-certificate).

Beispiele:

- curl -L -o latest.zip https://github.com/user/proj/releases/latest – Lade neueste Release, folge Redirects, speichere als latest.zip.
- curl -u admin:secret -X DELETE http://server/api/item/123 – Sende DELETE-Request mit Basic Auth.
- curl -H "Accept: application/json" 'https://api.example.com/data?id=5' – Hole JSON-Daten von API mit angepasstem Accept-Header.
- curl -d '{"user":"john"}' -H "Content-Type: application/json" -X POST https://api.example.com/users – Sende JSON-Body per POST (hier ein neuer User).
- curl -F 'file=@report.pdf' -F 'desc=Monthly Report' https://upload.example.com – HTTP File-Upload (Multipart).
- curl -I https://example.com – Nur Header einer HTTP-Response anzeigen (z.B. um Last-Modified zu prüfen).

Speicher- und Dateisystem-Verwaltung

lsblk (List Block Devices)

Beschreibung: Zeigt eine Übersicht aller erkannten Blockgeräte (Festplatten, SSDs, Partitionen, LVM-Volumes etc.) in Form einer Baumstruktur. lsblk listet wichtige Attribute wie Gerätename, Größe, Typ (disk/part), Mountpoint usw. Es hilft beim Identifizieren von Device-Namen und Partitionierungsstruktur.

Syntax:

lsblk [Optionen] [Gerät...]

Optionen:

- **-a, --all**: Zeigt auch leere bzw. unverwendete Geräte an (z.B. Geräte ohne Speicher, Loop ohne Backing). Standard lässt einige unwichtige aus.
- **-d, --nodeps**: Zeigt nur "Top-Level" Geräte (Disks) ohne Partitionen/Slaves.
- **-e <LIST>, --exclude <LIST>**: Schließt bestimmte Gerätetypen durch Major-Nummern oder Gerätekategorien aus (z.B. -e 7,11 um Loop und DVD zu ignorieren).
- **-f, --fs**: Zeigt zusätzlich Dateisystem-Informationen (Spalten FSTYPE, LABEL, UUID) . Ähnlich blkid, aber in lsblk integriert.
- **-p, --paths**: Zeigt vollständige /dev/Pfadnamen an statt nur Gerätenamen (z.B. "/dev/sda" statt "sda").
- **-l, --list**: Ausgabe im Listenformat (eine Eintrag pro Zeile, ohne Baum-Struktur).
- **-J, --json**: Ausgabe als JSON-Struktur (maschinenlesbar).
- **-O, --output-all**: Zeigt alle verfügbaren Spalten an. Standardanzeige ist begrenzt. Mit -O sieht man z.B. auch MAJ:MIN, RM (removable), RO (read-only), MODEL, SERIAL etc.
- **-n, --noheadings**: Unterdrückt Überschrift.
- **-b, --bytes**: Größe in Bytes anzeigen (statt lesbar in KiB/MiB etc.).
- **-r, --raw**: Rohformat ohne Ausrichtung/Padding (für Skripting).

Beispielausgabe:

```
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda       8:0   0 100G 0 disk
        ↘-sda1    8:1   0 512M 0 part /boot
        ↘-sda2    8:2   0 99.5G 0 part
            ↘-vg-root 253:0   0 90G 0 lvm /
            ↘-vg-swap 253:1   0 8G 0 lvm [SWAP]
sr0       11:0  1 1024M 0 rom (cdrom)
```

Hier sieht man Disk *sda* mit 100G und zwei Partitionen *sda1*, *sda2*. *sda2* ist ein PV (LVM Physical Volume) mit Volume Group vg -> daraus LVs *vg-root* und *vg-swap*, gemountet entsprechend.

lsscsi (List SCSI Devices)

Beschreibung: Listet erkannte SCSI-Geräte (inkl. SATA, SAS, USB Storage – diese werden SCSI-Subsys emuliert) auf. lsscsi zeigt je Gerät Host/Channel/Target/LUN, dann Gerätetyp, und eventuell Model und Seriennummer je nach Infile-4fdk7rvx39azjretinarqb】. Hilfreich, um z.B. alle angeschlossenen Festplatten und optischen Laufwerke anzuzeigen.

Syntax:

lsscsi [Optionen]

Optionen:

- **-v, --verbose:** Ausführlich – zeigt zusätzliche Infos (z.B. SCSI-Transport oder Zustände).
- **-t, --transport:** Zeigt SCSI-Transportinformationen (z.B. SATA, USB UAS, iSCSI etc.).
- **-L, --lunhex:** Zeigt die LUN in hexadezimaler Notation statt dezimal.
- **-H, --hosts:** Zeigt nur SCSI-Host-Adapter (Controller) an.
- **-c, --classic:** Benutzt "klassisches" Ausgabeformat (alter Stil, ohne Abkürzungen).
- **-g, --generic:** Fügt die entsprechenden sg-Device-Nodes hinzu (SCSI Generic /dev/sgX).
- **-s, --size:** Zeigt Größe (Kapazität) der Block-Geräte an, falls anwendbar (nur für Disk-type).
- **-p, --protection:** Zeigt Protection-Information (DIF/DIX).
- **-i, --scsi_id:** Zeigt die SCSI-IDs im klassischen Sinn (T10 IDs).
- **-B, --usb:** Zeigt Bus-IDs (für USB-Geräte) mit an.

Beispiel:

```
[0:0:0:0] disk ATA Samsung SSD 860 1B6Q /dev/sda 500GB
```

```
[2:0:0:0] cd/dvd TSSTcorp CDDVDW SH-224DB SB00 /dev/sr0
```

```
[3:0:0:0] disk WD MyBook 25DA 4004 /dev/sdb 4TB
```

Format: [H:C:T:L] type vendor model rev /dev/sgX -> /dev/sdX. Hier sieht man SCSI-Gerät auf Host 0 (SATA), Host 2 (DVD), Host 3 (USB-HDD).

fdisk (Partition Table Editor)

Beschreibung: Interaktives Konsolen-Tool zum Anzeigen und Bearbeiten von Partitionstabellen auf Blockgeräten (MBR, GPT). Meistens wird fdisk verwendet, um

schnell die Partitionierung eines Laufwerks anzusehen (fdisk -l) oder bei MBR-Partitionierung Partitionen einzurichten. Für GPT kann man gdisk oder parted benutzen (fdisk unterstützt aber inzwischen auch GPT).

Syntax:

fdisk [Optionen] <Gerät>

Ohne Optionen startet es ein interaktives Menü für das angegebene Gerät (z.B. /dev/sda).

Optionen:

- **-l, --list:** Listet die Partitionstabellen aller verfügbaren Laufwerke auf [file-4fdk7rvx39azjretinarqb]. Oder wenn ein Gerät angegeben, nur diese. Z.B. fdisk -l /dev/sda zeigt Partitionen auf sda.
- **-s Partition:** Gibt die Größe der angegebenen Partition in Sektoren aus (veraltet, blockdev --getsz moderner).
- **-u (veraltet):** Wechselt die Ausgabeeinheit (Sektoren vs Zylinder) in alter fdisk. Heutige fdisk (util-linux) nutzt standard 512-Byte Sektoren in -l.
- **-B, -C, -H, -S:** Kann man Geometrie (Zylinder, Heads, Sectors) angeben (MBR Legacy, heute irrelevant).
- **-t Typ:** Zeigt Partitionen in bestimmtem Format (Dos, GPT, ...).

Interaktiver Modus: Kommandos wie m (Hilfe), p (Print Partitionstabelle), n (neue Partition), d (löschen), t (Typ ändern, z.B. 83 Linux, 82 Swap, in GPT GUIDs), w (write und exit), q (quit ohne save).

Beispiel (List):

Disk /dev/sda: 238.5 GiB, ...

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1	*	2048	534527	532480	260M	7	HPFS/NTFS/exFAT
/dev/sda2		534528	409600000	409065473	195.1G	83	Linux
/dev/sda3		409600001	419430400	9830400	4.7G	82	Linux swap / Solaris

Zeigt z.B. 3 Partitionen (sda1 NTFS, sda2 Linux, sda3 Swap).

parted (Partitionierungstool)

Beschreibung: Moderner Partitionierer, unterstützt GPT und MBR sowie Größe > 2TB. Kann sowohl interaktiv (im eigenen Prompt) als auch direkt per Kommandozeile genutzt werden. Im Gegensatz zu fdisk beherrscht parted auch **Resizing** von Partitionen und Filesystems (begrenzte FS-Typ Unterstützung) und kann in Skripten genutzt werden.

Syntax (non-interactive):

parted [-s] <Gerät> <Befehl> [Parameter...]

Mit -s (script) ohne interaktive Ausgabe.

Häufige parted Befehle: (Nicht alle Optionen, nur exemplarisch)

- **print** – Zeigt Partitionstabelle des Geräts. Z.B. parted /dev/sdb print.
- **mklabel <Typ>** – Erstellt neue Partitionstabelle (Typ: gpt, msdos, etc.). Achtung: löscht alle bestehenden Partitionen.
- **unit <Einheit>** – Setzt Maßeinheit (MB, GB, %, etc.) für folgende Befehle.
- **mkpart Name Typ Anfang Ende** – Erstellt Partition. Z.B.: mkpart primary ext4 1MiB 10000MiB erstellt eine ~10GB Partition vom 1MB Offset. *Typ* hier "primary"/"logical" (bei msdos) oder nur Partitions-TypName (bei GPT optional ein FS-Typ Label ext4 etc.).
- **rm <Nr>** – Löscht Partition mit Nummer.
- **resizepart <Nr> <Ende>** – Passt Größe der Partition Nr an (End-Sektor). Muss danach FS separat resized werden (z.B. mit resize2fs).
- **name <Nr> <Name>** – Setzt den Partitionsnamen (GPT Label).
- **toggle <Nr> boot** – Boot-Flag toggeln (bei MBR Partitionen).
- **align-check Typ Nr** – Überprüft Ausrichtung von Partition Nr (optisch/ Zylinder).

Parted kann auch einfache FS-Aufgaben: mkfs integration, aber meist nutzt man separate Tools (mkfs.*).

Beispiele:

- parted /dev/sdc --script mklabel gpt mkpart primary ext4 0% 50% mkpart primary linux-swap 50% 100% – Initialisiert /dev/sdc als GPT, teilt in zwei: erste Partition bis Mitte ext4, zweite Swap ab Mitte bis Ende.
- Im interaktiven parted (ohne -s) bekommt man (parted) Prompt zum Eingeben der obigen Befehle nacheinander.

mkfs (Make FileSystem)

Beschreibung: Erzeugt ein neues Dateisystem auf einer Partition oder Festplatte. mkfs ist ein Frontend, das je nach -t <FSTYP> das entsprechende Tool (mkfs.ext4, mkfs.xfs, etc.) aufruft. Häufig ruft man direkt das spezifische Formatprogramm auf.

Syntax:

`mkfs -t <FSTYP> [Optionen] <Geräte/Partition>`

Beispiele: `mkfs -t ext4 /dev/sdb1`, `mkfs -t vfat /dev/sdb1`.

Wichtige Optionen (allgemein):

- **-t <typ>** – Dateisystemtyp (ext4, ext3, ext2, xfs, vfat, ntfs, etc.).
- **-V** – zeigt ausführliche Versions-/Debug-Information (welches Tool aufgerufen wird).
- **-q** – quiet (weniger Ausgabe).

Jedes Dateisystem hat eigene spezifische Optionen, wenn man direkt `mkfs.ext4` etc. nutzt:

ext4 (mkfs.ext4): `-L <Label>, -U <UUID>` (benutzerdefiniert), `-m <Proz>` (Reserve-Blöcke Prozent, default 5%), `-O <Feature>` (ein/aus bestimmter FS-Features), `-E stride=` (RAID optimierungen), `-b blocksize`, etc.

vfat (mkfs.vfat): `-n <Label>, -F <32|16>` (FAT type), `-I` (Force if not partition), etc.

xfs (mkfs.xfs): `-L <Label>, -m crc=1` (z.B. CRC enable, default ON neu), etc.

swap (mkswap): `-L <Label>`, (swap selbst ist via mkswap erstellt).

Wichtig: Partition sollte entsprechend markiert/Dimension formatiert sein (z.B. parted type=linux-swap für swap etc.).

Beispiele:

- `mkfs.ext4 /dev/sda3` – Formatiert sda3 mit ext4 Standardparametern.
- `mkfs.vfat -F 32 -n "MYUSB" /dev/sdb1` – Formatiert Partition sdb1 als FAT32 mit Label "MYUSB".
- `mkswap /dev/sda5` – Initialisiert Swap auf sda5 (später via swapon).
- `mkfs.xfs -f -L data /dev/sdc1` – Erzwingt Format auf sdc1 mit XFS, Label "data" (XFS verlangt -f wenn Partition schon ein FS enthält).

mount (Dateisystem einhängen)

Beschreibung: Bindet ein Dateisystem (Partition, Datenträger oder virtuelles FS) in den Verzeichnisbaum ein. `mount` ohne Argumente listet aktuelle Einhängungen. `mount <Gerät> <Verzeichnis>` hängt ein Gerät an den angegebenen Ordner ein (der Ordner muss existieren und leer sein, typischerweise unter /mnt oder /media oder definierter Mountpoint in /etc/fstabfile-4fdk7rvx39azjretinarqb】 .

Syntax:

`mount [Optionen] <Quelle> <Zielverzeichnis>`

Wichtige Optionen:

- **-t <fstype>** – gibt den Dateisystemtyp an (ext4, vfat, nfs, cifs, etc.). Kann oft weggelassen werden, da mount es automatisch erkennt (über blkid).
- **-o <Optionen>** – übergibt eine durch Komma getrennte Liste von Mount-Optionenfile-4fdk7rvx39azjretinarqb】 :

 - **defaults** (rw,suid,dev,exec,auto,nouser,async),
 - **ro/rw** (read-only / read-write),
 - **noexec** (keine Ausführung von Programmen erlauben), **nosuid** (SUID/SGID-Bits ignorieren), **nodev** (keine Gerätedateien interpretieren),
 - **sync** (synchrone IO), **async** (Standard, asynchron gepuffert),
 - **user/nouser** (Allow normal user to mount / nur root),
 - **uid=, gid=** (für FAT/NTFS, setze Besitzer der gemounteten Dateien),
 - **umask=, dmask=, fmask=** (Rechte-Masken für FAT/NTFS),
 - **iocharset=, codepage=** (für vfat),
 - **loop** (Loop-Device mount einer Datei als FS, evtl. -o loop oder auto),
 - **remount** (ein bereits gemountetes FS mit neuen Optionen versehen, z.B. mount -o remount,rw /mnt/point),
 - FS-spezifisch: z.B. **acl**, **user_xattr** (Ext4 Features), **compression**, **encrypt** (btrfs), etc.

- **-L <Label>, -U <UUID>** – anstatt Device-Node, per Label oder UUID identifizieren. mount -U 1234-ABCD /mnt/usb.
- **--bind** – führt einen Bind-Mount aus: mountet ein vorhandenes Verzeichnis an eine zweite Stelle im Verzeichnisbaum. Syntax: mount --bind /pfad/quelle /pfad/ziel. (Nützlich, um z.B. Verzeichnisse in Chroots sichtbar zu machen). Option **--rbind** rekursiv (inkl. Unterpunkte).
- **-a** – Mountet alles was in /etc/fstab als "auto" markiert ist und nicht noauto. (Ausführung typ. beim Boot).
- **-n** – führt Mount ohne Eintrag in /etc/mtab aus (mtab ist heute oft Symlink zu /proc/self/mounts). Selten benötigt.
- **-v** – verbose (gibt Meldungen aus).

Beispiele:

- mount /dev/sdb1 /media/usb -o uid=1000,gid=1000,umask=022 – hängt einen FAT32-Stick so ein, dass Benutzer mit UID/GID 1000 Besitzer ist (typisch eigener User), und Rechte 755 auf Dateien/Ordner (umask 022).

- `mount -t nfs 192.168.1.10:/export/share /mnt/share` – hängt eine NFS-Freigabe ein (NFS FS muss installiert sein).
- `mount -o loop diskimage.iso /mnt/iso` – mountet ein ISO-Image als Loop-Device unter /mnt/iso.
- `mount | grep sda` – sieht, ob /dev/sdX gemountet ist (ohne Parameter listet mount alles).

umount (Unmount FileSystem)

Beschreibung: Hängt ein gemountetes Dateisystem wieder aus. Schreibvorgänge werden abgeschlossen und der Puffer geleert. Syntax kann entweder der Mountpoint oder das Device angegeben werden】 .

Syntax:

`umount [Optionen] <Mountpunkt | Gerät>`

Wichtige Optionen:

- **-f, --force:** Erzwingt das Aushängen eines FS. Sollte nur im Notfall (z.B. NFS-Server down) benutzt werden, da es evtl. Datenverlust verursachen kann.
- **-l, --lazy:** "Lazy unmount". Markiert das FS als ausgeworfen, aber falls noch offene Handles bestehen, wird es erst tatsächlich ausgehängt, wenn diese geschlossen sind. (Hängt unmittelbar virtuell aus, lässt Rest im Hintergrund erledigen). Nützlich bei hängenden Prozessen.
- **-R, --recursive:** Hängt auch alle Unter-Mounts mit ab, die auf dem angegebenen Pfad liegen (rekursiv).
- **-v:** verbose.

Beispiele:

- `umount /media/usb` – hängt das unter /media/usb eingehängte Gerät aus.
- `umount /dev/sdb1` – selbes, angesteuert über Gerät.
- `umount -f -a -t nfs` – erzwingt das Aushängen aller NFS-Mounts (z.B. bei Netzproblemen).

(Tipp: Wenn `umount` meldet "device is busy", kann man mit `lsof` oder `fuser` herausfinden, welcher Prozess noch auf das FS zugreift.)

df (Disk Free)

Beschreibung: Zeigt Übersicht des belegten und freien Speicherplatzes auf eingehängten Dateisystemen

Syntax:

```
df [Optionen] [Dateipfad...]
```

Ohne Argumente zeigt alle gemounteten FS. Mit Pfad ermittelt df, auf welchem FS dieser Pfad liegt, und zeigt nur dessen Werte.

Optionen:

- **-h, --human-readable**: Formatiert Größen in lesbarer Form (KiB, MiB, GiBfile-4fdk7rvx39azjretinarqb】. Sehr gebräuchlich (Standard rohes Blockzahlen unhandlich).
- **-a, --all**: Schließt pseudo- oder 0-Size-Dateisysteme ein (wie proc, sysfs, etc.), die sonst ausgelassen werden.
- **-T, --print-type**: Zeigt Spalte mit Dateisystemtyp (ext4, tmpfs, nfs, etc.).
- **-i, --inodes**: Zeigt Inode-Auslastung statt Blockgrößen (frei/belegt Inodes pro FS). Nützlich, wenn "no space" durch Inode-Erschöpfung.
- **-t <typ>, -x <typ>**: Nur bestimmte FS-Typen anzeigen bzw. ausschließen. Z.B. df -x tmpfs -x devtmpfs um echte Disks anzuzeigen.
- **-B <Größe>**: Ausgabe in Blöcken von angegebener Größe. Z.B. -B MB (MByte) oder -B 1M.
- **--output[=FELDER]**: Benutzerdefinierte Spaltenausgabe. Z.B. df --output=source,fstype,size,used,avail,pcent,target.

Beispielausgabe (df -h):

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda2	50G	30G	18G	63%	/
udev	3.9G	0	3.9G	0%	/dev
tmpfs	798M	1.2M	797M	1%	/run
/dev/sda1	300M	50M	250M	17%	/boot

du (Disk Usage)

Beschreibung: Summiert und zeigt die Größe von Verzeichnissen und Dateien an. du kann rekursiv durch Verzeichnisse gehen und den Platzverbrauch berechnenfile-4fdk7rvx39azjretinarqb】. Standardmäßig zeigt es für jedes Unterverzeichnis den Wert. Oft nutzt man du -sh <dir> um eine schnelle Gesamtgröße zu bekommen.

Syntax:

```
du [Optionen] [Pfad...]
```

Ohne Pfad = aktuelles Verzeichnis.

Optionen:

- **-h, --human-readable**: Größen in menschenlesbare Einheiten (K,M,G) ausgeben
file-4fdk7rvx39azjretinarqb】 .
- **-s, --summarize**: Nur Summen pro angegebenem Argument ausgeben, *nicht* rekursiv für Unterordner-4fdk7rvx39azjretinarqb】 . (Gibt also Gesamtgröße des Pfades).
- **-A, --apparent-size**: Rechnet mit logischen Dateigrößen statt belegtem Blockplatz. (Z.B. sehr sparsame Sparse Files würden sonst weniger anzeigen; --apparent-size zeigt nominelle Größe).
- **-d N, --max-depth=N**: Rekursionstiefe begrenzen. Z.B. du -h --max-depth=1 /var gibt Größe von /var und dessen Unterordner erste Ebene. (--max-depth=0 wäre wie -s).
- **-c, --total**: Am Ende eine Gesamtzeile "total" ausgeben (Summe aller ausgegebenen Einträge). Bei -s von mehreren Pfaden praktisch.
- **-x, --one-file-system**: Geht nicht in andere eingehängte Dateisysteme über. (Beschränkt auf Partition/FS der Startpfade).
- **-L, --dereference**: Folgt symbolischen Links (sonst ignoriert du Symlinks, rechnet sie als 0 Bytes). Vorsicht: Kann zu doppeltem Zählen führen, wenn Links in Kreis oder auf Oberverzeichnis zeigen.
- **--exclude=<PATTERN>**: Schließt Pfade aus, die zum Pattern passen. Z.B. du -sh -exclude=".git" oder mehrere --exclude.

Beispiele:

- du -sh /home/* – Zeigt Summe der Größen aller Benutzerverzeichnisse unter /home (jedes in human-readable).
- du -a /etc | grep -E "mysql.*\.cnf\$" – Listet die Größe jeder einzelnen Datei unter /etc, filtert jene, deren Name auf mysql*.cnf endet. (Option -a listet auch Dateien und nicht nur Verzeichnisse).
- du -d2 /var – Größe von /var, dessen Subdirs und Sub-Subdirs (Tiefe 2).

e2label (ext2/3/4 Volume Label)

Beschreibung: Zeigt oder ändert das Label (Namen) eines ext2/ext3/ext4 Dateisystems. Das *Label* ist eine im Superblock gespeicherte Zeichenkette, die man z.B. in fstab zum Mounten verwenden kann (LABEL=<name>).

Syntax:

e2label <Gerät> [NeuesLabel]

Ohne zweites Argument zeigt es das aktuelle Label des ext-Volumes an. Mit Label schreibt es dieses in den FS.

Beispiel:

- e2label /dev/sda3 -> Ausgabe z.B. UbuntuRoot.
- e2label /dev/sda3 Archive -> setzt das Label von sda3 auf "Archive".

(Für andere FS gibt es analoge Tools: XFS nutzt xfs_admin -L, FAT/NTFS mlabel oder dosfslabel/ntfslabel etc.)

resize2fs (ext2/3/4 Dateisystem verkleinern/vergrößern)

Beschreibung: Passt die Größe eines ext2/3/4-Dateisystems an. Wird verwendet nach dem Ändern der Partitionsgröße (mit fdisk/parted). resize2fs kann im online-Modus vergrößern (wenn gemountet, ext3/4), fürs verkleinern muss das Volume ausgehängt und ein Filesystem-Check gemacht werden.

Syntax:

resize2fs [Optionen] <Gerät> [NeueGröße]

NeueGröße kann in K, M, G oder als Anzahl 4K-Blöcke angegeben werden. Lässt man sie weg, wird auf Maximum (Part.-Ende) vergrößert.

Optionen:

- **-p** – progress anzeigen (Fortschritt).
- **-f** – zwingen, auch wenn FS als nicht sauber markiert (normal will fsck vorher laufen bei Verkleinern).
- **-F** – flush caches vorher (sorgt für up-to-date Infos; Standard normal).
- **-M – minimal**: verkleinert das FS auf die kleinstmögliche Größe (alle Daten gerade passend). Äquivalent zu resize2fs <dev> -M (dauert Berechnung).

Beispiele:

- Nach Vergrößern der Partition: resize2fs /dev/sda2 – erkennt neue Partitionsgröße und wächst das ext4 FS auf vollen Platz.
- Verkleinern: umount /dev/sda2; e2fsck -f /dev/sda2; resize2fs /dev/sda2 20G – erst aushängen, fsck erzwingen, dann auf 20 GiB verkleinern. Anschließend Partition selbst verkleinern mit parted.

tune2fs (Tune ext2/3/4 Filesystem Parameters)

Beschreibung: Ändert diverse Parameter eines bestehenden ext-Dateisystems: Reserveblöcke, Check-Intervalle, Label, Journal einschalten, usw.

Syntax:

tune2fs [Optionen] <Gerät>

Häufige Optionen:

- **-l** – Listet alle Superblock-Parameter (Filesystem-Info) auf freedesktop.org. Zeigt z.B. Inode-Größe, Features, Letzter Mount, Nächster geplanter Check, etc. (Read-only Operation).
- **-c <MaxMounts>** – Setzt die maximale Anzahl Mount-Vorgänge bis zur erzwungenen fsck-Prüfung. Z.B. -c 0 = deaktiviert mount-count-basierten Check.
- **-i <Interval>** – Setzt das Zeit-Intervall für Zwangs-FSCK. z.B. -i 30d = alle 30 Tage. -i 0 = kein zeitbasierter Check.
- **-r <AnzRes>** – Setzt Anzahl der reservierten Blöcke (für root). Standard 5% bei ext4 auf <1TB. Kann man z.B. auf 1% reduzieren auf großen Volumes, z.B. tune2fs -r 0 /dev/sdb1 um Reserven komplett abzuschalten (nicht empfohlen auf /). Alternativ -m <Prozent> (z.B. -m 1).
- **-L <Label>** – Ändert Label (wie e2label).
- **-U <UUID>** – Ändert die UUID (zufällig oder angegeben). (Vorsicht: fstab etc. müssen angepasst).
- **-J** – Journal-Optionen: z.B. -O has_journal um Journal hinzuzufügen (macht aus ext2 -> ext3), oder -O ^has_journal Journal entfernen (ext3->ext2), oder -j Kurzoption (Standardwerte Journal hinzufügen).
- **-O <FeatListe>** – Aktiviert (feature) oder deaktiviert (^feature) Filesystem-Features. Z.B. -O casefold (für ext4 ab 5.2: Case-Insensitive-Folder Support) oder ^huge_file (Feature abschalten – Achtung, Daten verlieren, nur wenn FS leer). Nur bestimmte Features togglebar, meist erfordert offline und vollständiges Backup.
- ****-E <extendopt>**** – Erweiterte Einstellungen: z.B. stripe=<StripeSize>(RAID stripe in Blöcken),lazy_itable_init=1` (schnell format, Inode-Table lazy Clearing).
- **-M <Verzeichnis>** – Markiert FS als gemountet an <Verzeichnis> (setzt den Mountpoint in Superblock – meist rein informativ).

dumpe2fs (Dump ext2/3/4 Superblock)

Beschreibung: Gibt ausführliche Informationen über ein ext2/3/4-Dateisystem aus – Inhalt des Superblocks und der Block Group Deskriptore. Zeigt ähnlich tune2fs -l Informationen, plus Aufteilung der Blöcke in Gruppen, freie Blöcke/Inodes pro Gruppe etc. Hilfreich für Diagnose (z.B. Backup-Superblock Positionen).

Syntax:

dumpe2fs [Optionen] <Gerät>

Optionen:

- **-h** – Zeigt nur den Inhalt des Superblocks und der Gruppen-Deskriptoren (ohne die Liste aller freien Blöcke/Inodes). Das ist die gängigste Verwendung, da ohne -h die Ausgabe sehr lang sein kann (listet freie Blöcke/Inodes aller Groups wiki.ubuntuusers.de).
- **-x** – Zeigt auch die detaillierten Gruppen-Informationen (wie freie Blöcke/Inodes-Bitmap). Standard ohne -h tut das sowieso.
- **-b** – Listet die Nummern der freien Blöcke. (Sehr viel Output auf großem FS.)
- **-i** – Listet die freien Inode-Nummern.

Beispiel (gekürzt dumpe2fs -h /dev/sda2):

Filesystem volume name: <none>

Last mounted on: /

Filesystem UUID: 4e8fea21-...-b6c7

Filesystem magic number: 0xEF53

Filesystem revision #: 1 (dynamic)

Filesystem OS type: Linux

Block size: 4096

Block count: 25600000

Reserved block count: 1280000

Free blocks: 15034567

First Block: 0

...

Journal inode: 8

Journal backup: inode blocks

...

Zeigt Label, UUID, Blocksize, Anzahl Blöcke, Reserve, Anzahl freie etc., und welche Features aktiviert sind.

cryptsetup (LUKS Disk Encryption Setup)

Beschreibung: Tool zur Einrichtung von **verschlüsselten Datenträgern** mit LUKS (Linux Unified Key Setup) sowie zum Öffnen/Schließen solcher Verschlüsselungs-Volume】 . cryptsetup kann ein Gerät mit LUKS initialisieren, Passphrase setzen, und im Betrieb ein verschlüsseltes Volume als Device-Mapper-Gerät bereitstellen, das dann gemountet werden kann.

Typische Aktionen (Subcommands):

- **luksFormat** – Initialisiert ein Blockgerät mit LUKS-Verschlüsselungsheader und fordert eine Passphrase zum Einrichten an. Beispiel: cryptsetup luksFormat /dev/sdb1. *Achtung:* Löscht alle vorhandenen Daten auf dem Gerät.
- **open** oder **luksOpen** – Öffnet ein verschlüsseltes Volume (fragt Passphrase). Syntax: cryptsetup luksOpen /dev/sdb1 geheim_vol. Dies erstellt ein Device /dev/mapper/geheim_vol, das nun entschlüsselt agiert und z.B. mit mkfs formatiert oder gemountet werden kan】 .
- **close** – Schließt ein vorher geöffnetes LUKS-Device: cryptsetup close geheim_vol (dann ist /dev/mapper/geheim_vol weg).
- **luksAddKey** – Fügt einen weiteren Schlüssel (Passphrase) hinzu. LUKS unterstützt mehrere Key-Slots. cryptsetup luksAddKey /dev/sdb1 /pfad/zur/schlüsseldatei oder interaktiv eine neue Passphrase eingeben (nachdem man eine gültige alte eingibt).
- **luksRemoveKey** – Entfernt einen Schlüssel (kann durch Angabe der zu entfernenden Passphrase oder Slot-Nummer erfolgen).
- **luksChangeKey** – Ändert eine bestehende Passphrase (in einem Slot) durch eine neue (kombi aus Remove+Add effizienter).
- **luksHeaderBackup** / **luksHeaderRestore** – Backup bzw. Wiederherstellen des LUKS-Headers (empfohlen, falls Header korrupt -> Daten verloren).
- **status** – Zeigt Informationen über ein offenes Device-Mapper Volume (Name, verschlüsseltes Dev, Cipher, Keylen, etc.): cryptsetup status geheim_vol.

Wichtige Optionen:

- **-y** (im Format-Befehl): fragt neues Passwort zweimal zur Verifizierung ab.
- **-d <Datei>**: verwendet eine Datei als Schlüsseldaten (anstatt interaktiv Passwort). -d - kann genutzt werden, um vom stdin zu lesen.
- **--key-size <bits>**: Schlüsselgröße (in Bits) für die Verschlüsselung (z.B. 256).
- **--cipher <name>**: Zu nutzender Verschlüsselungsalgorithmus und Mode. Standard z.B. aes-xts-plain64 bei LUKS2. Kann z.B. aes-cbc-essiv:sha256 oder anderes angegeben werden.

- **--hash <hash>**: Hash-Algorithmus für Passphrase-Derivation (PBKDF). Standard modern ist z.B. Argon2id oder früher sha256.
- **--iter-time <ms>**: Wie lange (Millisek) auf PBKDF verwendet werden soll (Bestimmt Komplexität).
- **--type luks1/luks2**: LUKS Version festlegen (aktuell Standard LUKS2, kann optional LUKS1 für Kompatibilität gewählt werden).
- **-q**: Quiet (unterdrückt Ausgabe, z.B. Warnung bei Format).

Beispiel Workflow:

1. **Formatieren**: cryptsetup -y luksFormat /dev/sda5 (legt LUKS auf Partition sda5 an, fragt 2x Passwort).
2. **Öffnen**: cryptsetup luksOpen /dev/sda5 geheim (Passwort eintippen, erstellt /dev/mapper/geheim).
3. **Format FS**: mkfs.ext4 /dev/mapper/geheim.
4. **Mount**: mount /dev/mapper/geheim /mnt/secure. Jetzt Daten drauf speichern.
5. **Aushängen+Schließen**: umount /mnt/secure && cryptsetup close geheim.

(LUKS ermöglicht bis zu 8 Passphrases; man kann z.B. mit luksAddKey einen zweiten hinzugeben, sodass entweder Passphrase A oder B das Volume öffnet. Das Gerät /dev/sda5 kann in /etc/crypttab und /etc/fstab eingetragen werden für Boot-Time auto-open via keyfile oder prompt.)

shred (Securely wipe file/device)

Beschreibung: Überschreibt eine Datei oder ein ganzes Gerät mehrfach mit zufälligen Daten, um ein Wiederherstellen gelöschter Inhalte zu erschweren. shred ist nützlich zum sicheren Löschen auf magnetischen HDDs. Bei SSDs mit Wear-Leveling ist Effekt begrenzt – dort lieber ATA Secure Erase oder physical destruction.

Syntax:

shred [Optionen] <Datei1> [Datei2...]

Optionen:

- **-n N** – Anzahl der *zusätzlichen* Überschreib-Pässe (Standard 3). D.h. 3 bedeutet insgesamt 3 mal random + final pass (siehe -z).
- **-z, --zero** – Überschreibt am Ende noch einmal mit Null-Bytes, um das Muster zu verwischen (anstatt dass zuletzt zufälliges Rauschen bleibt). Standard: aktiv (bei neueren shred ist -z default an).

- **-f, --force** – Ändert Schreibschutz falls nötig (chmod u+w), um schreiben zu können.
- **-u, --remove[=how]** – Nach dem Überschreiben die Datei löschen (unlink). Optional how kann unlink (Standard) oder wipe oder wipesync sein: *wipe* überschreibt erst mit Null dann löscht, *wipesync* macht zusätzlich Sync. Standard -u bereits tut *wipesync*.
- **-v, --verbose** – Auskunft über Fortschritt.
- **-x, -X, --exact** – Überschreibt Datei genau Byte-für-Byte anstelle Dateigröße auf nächste Blockgröße aufzurunden (Standard shred nutzt ganze Blocks, kann über Dateiende hinaus schreiben falls Dateigröße nicht mod Blocksize). *Exact* vermeidet das, aber hinterlässt evtl. un-geschredderte Slack am Ende.
- **-s N, --size=N** – Behandelt die Datei als N Byte lang (statt aktuelle Länge). Um auch ungenutzten Platz (z.B. Slack Space/EOF beyond) im Inode zu überschreiben, kann man etwas größer angeben.

Beispiele:

- `shred -u file.txt` – Überschreibt *file.txt* viermal (3 rand + 1 zero) und löscht sie anschließend.
- `shred /dev/sdb` – Überschreibt komplett `/dev/sdb` mit 3 Runden Zufallsdaten (zerstört alle Daten/Partitionen darauf).
- `shred -n 0 -z -u secret.dat` – Einmal mit Nullen überschreiben und entfernen (hier wollte man evtl. nur einfaches Überschreiben und löschen).

Warnung: Shred wirkt nicht zuverlässig auf: **logischen Volumes** in Files (z.B. Datei in FS kann redundante Kopien (Journal, Snapshots) haben), **RAID, SSD, Netzwerkdateisysteme, Dateisysteme mit Copy-on-Write (btrfs, ZFS)** – dort können Daten an mehreren Orten stehen. In solchen Fällen besser Vollverschlüsselung verwenden oder das gesamte physische Medium überschreiben.

iostat (IO Statistics)

Beschreibung: Zeigt Statistik zur Systemauslastung von CPU und Input/Output für Blockgeräte (Datenträger). iostat gehört zum Paket sysstat. Es hilft zu sehen, wie stark Platten ausgelastet sind (%util) und Durchsatz (MB/s), IOPS etc., sowie generelle CPU-Idle/Wait Zeite】 .

Syntax:

`iostat [Optionen] [Intervall [Durchläufe]]`

Ohne Argument gibt einen Durchschnitt seit Boot aus. Mit Intervall (Sekunden) aktualisiert regelmäßig.

Wichtige Optionen:

- **-c** – Zeigt nur CPU-Statistiken.
- **-d** – Zeigt nur Disk (Geräte) Statistiken.
- **-k, -m** – Ausgabe in KiB/s oder MiB/s (anstatt Blöcke/s). (Aktuelle sysstat erlaubt -block-size).
- **-p [Gerät]** – Detaillierte Partitions-Statistiken einbeziehen (normal zeigt iostat nur Whole-Device). Z.B. -p sda zeigt sda und sda1,sda2,...
- **-x** – Ausgabe in erweiterter Form (Pro Gerät: zusätzlich Auslastungsprozentsatz, Average Request Size, Await Times etc.) . Sehr nützlich, da Standard ohne -x recht knapp.
- **-t** – Zeigt Timestamp vor jeder Ausgabe.
- **-y** – Unterdrückt erste Ausgabe (since boot), beginnt erst mit Werten ab erstem Intervall.

Interpretation (bei iostat -x):

Spalten pro Device: rrqm/s wrqm/s (Merge von I/O, veraltet bei modernen Kernel meist 0), r/s w/s (reads/writes per sec), rkB/s wkB/s (Throughput), avgrq-sz (Durchschnittl. Requestgröße in Sektoren oder KB), avgqu-sz (durchschn. Queue length), await (Durchschnittliche Wartezeit pro IO ms), svctm (durchschn. Servicezeit pro IO ms – oft nicht mehr zuverlässig), %util (Prozentuale Zeit, die das Gerät aktiv war innerhalb der Messperiode – 100% bedeutet permanent beschäftigt).

Beispiel:

```
Device  r/s  w/s  rkB/s  wkB/s  avgrq-sz  avgqu-sz  await  svctm  %util
sda    12.5  30.2  500.0  250.0  40.0   0.5    10.0   2.00  85.0
Hier sda ~85% ausgelastet, wartet ~10ms per IO, etwa 42.7 IO/s (r+s) mit ~750kB/s.
```

ioping (I/O Ping)

Beschreibung: Ermittelt Latenz (und ggf. Durchsatz) von Storage in Echtzeit ähnlich dem Prinzip von ping im Netzwerk. Es führt schnelle sequenzielle oder wahlfreie Lesezugriffe durch und misst deren Zeit. Praktisch um Latenz auf Disks oder bestimmten Verzeichnissen (Dateisystem) festzustellen.

Syntax:

```
ioping [Optionen] <Ziel>
```

Ziel kann ein Pfad (Datei oder Verzeichnis) sein für Filesystem IO, oder z.B. ein Blockgerät (mit -D für Direkten Device Zugriff) oder sogar ein bestimmter I/O-Engine (IO depth).

Wichtige Optionen:

- **-c <count>** – Anzahl der Ping-Requests (Standard unendlich, bis abgebrochen).
- **-i <interval>** – Wartezeit zwischen Pings (Sekunden, default 1s). Kann in ms angeben z.B. -i 0.1.
- **-s <bytes>** – Größe der zu lesenden Daten pro Ping (Standard 4KiB).
- **-R** – Ausgabe im Rohformat (Maschinenlesbar).
- **-D** – Direct-Device Mode: Arbeitet auf Blockgeräte direkt (bypasses FS-Cache).
- **-A** – Ausgabe als Aggregatstatistik nur (kein laufendes Ping, nur Summary).
- **-S** – Sync/Sequenziell: testet sequenzielles Lesen statt wahlfrei. (Standard ioping macht random position reads).
- **-W** – führt Schreib-Tests durch statt Lesen (Achtung kann Daten überschreiben zerstören, am besten auf extra Gerät/Datei).
- **-P <N>** – Parallel Requests (Thread count).
- **-q** – keine laufende Ausgabe, nur Summary am Ende.

Beispiele:

- ioping . – Sendet kontinuierlich 4K-Leseanfragen ins aktuelle Verzeichnis (random in FS), zeigt Latenzen, bis Ctrl+C.

```
4096 bytes from . (ext4 /dev/sda2): request=1 time=0.47 ms
```

```
4096 bytes from . (ext4 /dev/sda2): request=2 time=0.44 ms
```

```
... (zeigt typisch ~0.5ms auf SSD, ~10-15ms auf HDD) ...
```

Nach Abbruch gibt es eine **Summary** mit min/avg/max/mdev (Jitter).

- ioping -c 10 /mnt/backup – Misst 10 Zugriffe im FS /mnt/backup (z.B. externer USB), dann stop.
- ioping -R /dev/sdb – Führt 1 Sekunde lang Messungen auf /dev/sdb durch und gibt am Ende aggregated IOPS und Bandbreite.
- ioping -A -S -c 50 /file.iso – Liest sequenziell 50 Anfragen durch große Datei und gibt am Ende Durchschnittszeit.

(Tool muss ggf. installiert werden, gehört nicht standard zu allen Distr.)

Geräte, Prozesse, Speicher und Kernel

hwinfo (Hardware Info)

Beschreibung: Liefert detaillierte Informationen zur Hardware des Systems. hwinfo probiert, alle verfügbaren Hardware-Komponenten aufzulisten: CPU, RAM, Festplatten, USB, PCI-Geräte, Netzwerk, Monitor, etc., inklusive ihrer Eigenschaften und Treiber]. Oft auf openSUSE installiert, aber auch auf anderen Distros verfügbar.

Syntax:

```
hwinfo [Optionen] [Hardwareklasse]
```

Optionen / Klassen:

- Ohne Angabe gibt hwinfo *alles* aus (sehr lang). Besser, man filtert nach Klasse: z.B. hwinfo --cpu, hwinfo --disk, --gfxcard, --netcard, --printer, --usb, --bios, --memory etc.
- **--short:** Kurzfassung ausgeben (ein Eintrag pro Gerät in einer Zeile).
- **--brief:** Noch knapper (ähnlich short).
- **--summary:** Summaries (z.B. nur Zahl der Disks, CPUs etc.).
- **--verbose <level>:** Detaillierungsgrad (0-9). Default 1 (schon sehr ausführlich).
- **--debug N:** Setzt Debug-Level (für Entwickler, welche Module hwinfo benutzt).
- **--hwitem <ID>:** Sucht nach Hardware mit bestimmter ID (z.B. PCI-ID).
- **--save-config <datei> / --load-config <datei>:** Speichert erkannte Hardware in Datei zum späteren Vergleich.

Beispiel: hwinfo --cpu könnte ausgeben:

```
01: None 00.0: 10103 CPU
```

```
[Created at cpu.462]
```

```
Unique ID: rdCR.jX8Un+o2ZT1
```

```
Hardware Class: cpu
```

```
Arch: X86-64
```

```
Vendor: "GenuineIntel"
```

```
Model: 6.158.10 "Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz"
```

```
... (viele weitere Details Frequenzen, Cache etc.) ...
```

hwinfo --short summarisiert z.B. Hauptgeräte und Partitionen.

dmidecode (DMI Table Decode)

Beschreibung: Gibt Informationen aus dem BIOS/UEFI-DMI (SMBIOS) heraus, wie Hersteller, Produktnamen, Seriennummern von System/Mainboard, BIOS-Version, verbauter Speicher, Anschlüsse etc]. Hilfreich für Hardware-Inventarisierung (Seriennummer, RAM-Module, OEM strings).

Syntax:

dmidecode [Optionen]

Optionen:

- **-t <string|Nummer>, --type:** Nur bestimmte DMI-Typen ausgeben. Typ kann Name oder Zahl sein. Z.B.:
 - bios (DMI type 0),
 - system (1),
 - baseboard (2),
 - processor (4),
 - memory (Memory Device, 17),
 - cache (7),
 - connector (8), slot (9), etc.

Man kann -t memory um nur RAM-Infos zu sehen (Größe, Typ, Hersteller, Seriennummer der DIMMs] .
- **-s <keyword>, --string:** Einzelne SMBIOS-Einträge gezielt ausgeben (z.B. bios-vendor, bios-version, system-manufacturer, system-serial-number, baseboard-asset-tag, processor-frequency etc.).
- **-q:** Quiet (unterdrückt weniger wichtige DMI-Eintrags-Labels wo "Not Provided").
- **-u:** Raw dump (hex) der DMI-Daten.
- **--dump-bin file, --from-dump file:** Dump SMBIOS in Datei bzw. lese aus Dump (z.B. offline Analyse).

Beispiel: dmidecode -t system könnte zeigen:

System Information

Manufacturer: Dell Inc.

Product Name: Latitude 5490

Version: None

Serial Number: ABCD123

UUID: 4c4c4544-0034-5810-8055-b6c04f513332

Wake-up Type: Power Switch

SKU Number: 07E6

Family: Latitude

dmidecode -t memory zeigt pro RAM-Riegel z.B. Größe, Form Factor (SODIMM), Type (DDR4), Speed, Manufacturer, Serial, Part Number.

lspci (List PCI Devices)

Beschreibung: Zeigt alle PCI/PCIe-Geräte im System (angeschlossene Hardware am PCI-Bus: z.B. Netzwerkkarten, Sound, USB-Controller, Grafikkarten, SATA-Controller etc.). Es listet Geräte-ID und -Namen anhand einer Datenbank (/usr/share/misc/pci.ids).

Syntax:

`lspci [Optionen]`

Optionen:

- **-v, -vv, -vvv:** Verbose-Ausgabe in ein oder mehreren Stufen (mehr Register-Informationen). -vvv kann *sehr* ausführlich sein (inkl. Capabilities, Konfigspace).
- **-k:** Zeigt Kernel-Treiber und Module, die für jedes Gerät verwendet werden (welcher Treiber zugeordnet ist). (Z.B. Kernel driver in use: e1000e für Intel NIC).
- **-t:** Zeigt die hierarchische Baumstruktur der PCI-Bus-Topologie (wer hängt an welchem Bus/Bridge).
- **-nn:** Zeigt neben dem Gerätenamen auch Vendor:Device ID in eckigen Klammern. -n nur numeric IDs ohne Namen. -nn= beide. Z.B. Ethernet Controller [8086:1533] etc.
- **-d [<vendor>:<device>]:** Filtert nach bestimmten PCI IDs. `lspci -d 8086`: zeigt nur Intel-Geräte, `-d :1533` nur Geräte mit Device 1533.
- **-s [<domain>:<bus>:<slot>.<func>]:** Filtert nach Bus-Adresse (wie in `lspci -t` oder `/sys/bus/pci`). Z.B. `-s 00:1f.2` zeigt nur dieses Gerät.
- **-m:** Maschinenlesbar (gibt in einer CSV-artigen Formatzeile pro Gerät: Domain Bus Slot, Klass, Vendor, Device, SVendor, SDevice).
- **-xxx:** Dump des gesamten PCI-Konfigurationsspeichers (64 Bytes beyond standard) – Vorsicht sensibel. -xx nur ersten 256 Bytes (std config).

- **-H1:** Nutzen von /proc/bus/pci statt direct (historisch, meist nicht nötig).
- **-Q:** Zeigt das Querier-Modell (Kernel vs direct vs /proc).

Beispiel Ausgabe (gekürzt):

```
00:00.0 Host bridge: Intel Corporation Device 5914
00:02.0 VGA compatible controller: Intel Corporation UHD Graphics 620
00:14.0 USB controller: Intel Corporation Sunrise Point-LP USB 3.0 xHCI Controller (rev 21)
00:16.0 Communication controller: Intel Corporation Management Engine Interface (rev 21)
...
03:00.0 Network controller: Intel Corporation Dual Band Wireless-AC 8265 (rev 78)
```

Mit -nn würde man z.B. sehen 00:02.0 VGA controller [0300]: Intel Corporation UHD Graphics 620 [8086:5917]. Mit -k ergänzt: Kernel driver in use: i915.

lsusb (List USB Devices)

Beschreibung: Listet USB-Geräte auf, die am System angeschlossen sind (inkl. interne Hubs etc.). Zeigt Vendor und Produktnamen falls bekannt (Datenbank /usr/share/misc/usb.ids) und ID.

Syntax:

lsusb [Optionen]

Optionen:

- **-v:** Verbose, zeigt detaillierte Descriptor-Infos pro Gerät (ähnlich lsusb -vv noch detaillierter). Das kann sehr lang sein.
- **-t:** Baum-Darstellung der USB-Geräte-Hierarchie (welches Device hängt an welchem Port/Hub) – ähnlich lspci -t.
- **-s <bus>:<dev>:** Zeigt nur spezifisches Gerät (per Bus und Device Nummer). Diese Nummern sieht man in lsusb Standardausgabe vorn.
- **-d vid:pid:** Filtert nach VendorID:ProductID (Hex).
- **-D <device-file>:** Liest Infos aus einer Geräte-Datei (z.B. /dev/bus/usb/001/007) – erfordert Root.
- **-w:** Ausgabe als mit usbdump kompatibler Text (Wire format).
- **-v -v (or -vv):** liefert noch detailliertere Dump (inkl. Hexdump der HID Deskriptoren etc.).

Beispiel Ausgabe:

Bus 002 Device 003: ID 0bda:0316 Realtek Semiconductor Corp. Card Reader

Bus 002 Device 002: ID 8087:0aaa Intel Corp.

Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub

Bus 001 Device 004: ID 0c45:671f Microdia Integrated Webcam

Bus 001 Device 003: ID 8087:07dc Intel Corp. Bluetooth wireless adapter

Bus 001 Device 002: ID 056e:4008 Ecom Co., Ltd

Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

Hier z.B. Card Reader, Webcam, Bluetooth, eine Maus (056e:4008 Ecom), plus Root Hubs.

Mit -t etwa:

/: Bus 02.Port 1: Dev 1, Class=root_hub, ... (bus 2, 3.0)

|__ Port 1: Dev 2, Intel USB 3.0 Hub

|__ Port 2: Dev 3, Realtek Card Reader

...

Zeigt Topologie (Hub etc.).

lscpu (List CPU Info)

Beschreibung: Zeigt Informationen über die CPU(s) des Systems – Anzahl Kerne, Threads, Sockets, Modellname, Frequenz, Virtualisierungsfunktionen uswfile-4fdk7rvx39azjretinarqb】 .

Syntax:

`lscpu [Optionen]`

Optionen:

- **-e, --extended [=list]:** Ausgabe als Tabelle mit Spalten (erweiterte Auflistung jeder logischen CPU mit Eigenschaften). Optional kann man angeben, welche Felder (z.B. CPU, Node, Socket, Core, Online, ...).
- **-p, --parse [=list]:** Ähnlich --extended aber kommagetrennt (parseable). Standardfelder: CPU,Core,Socket,Node.
- **-x, --hex:** Zeigt hex-Werte wo sinnvoll (z.B. CPU bitmasken).
- **-c, --offline / --online:** Zeigt nur offline oder nur online CPUs (bei Hotplug-Systemen).

- **-y, --physical:** Zeigt nur physische IDs (keine virtuellen IDs).
- **-J, --json:** JSON-Ausgabe (maschinenlesbar).
- **-B, --byte:** Einheit Bytes statt kB für Cache-Größen.
- **-C, --cache:** Zeigt Details zu CPU-Caches (Anzahl, Größen).

Beispiel (gekürzt lscpu Standard):

Architecture: x86_64

CPU op-mode(s): 32-bit, 64-bit

Sockets: 1

Cores per socket: 4

Threads per core: 2

Model name: Intel(R) Core(TM) i7-8565U @ 1.80GHz

CPU MHz: 2112.004

Virtualization: VT-x

L1d cache: 32K

L2 cache: 256K

L3 cache: 8192K

Flags: fpu vme ... vmx ...

Zeigt also 1 Sockel, 4 Kerne, 8 Threads (4*2), CPU-Modell und Takt etc.

lsmem (List Memory Blocks)

Beschreibung: Zeigt Speicherblock-Informationen des Systems (insbesondere auf Systemen mit Hotpluggable Memory oder NUMA). Es listet Memory-Block-Geräte (wie in /sys/devices/system/memory) auf. Auf x86-PCs weniger relevant; auf IBM System z oder Power etc. nützlich.

Syntax:

lsmem [Optionen]

Optionen:

- **-n, --online / -o, --offline:** Zeigt nur online bzw. offline Speicherblöcke.
- **-b, --bytes:** Größen in Bytes statt lesbar.
- **-M, --mega:** Größen in MiB. (Standard lsmem gibt in bytes + kMG suffix)

- **-p, --pairs**: Ausgabe im Key:Value Format (für maschinelles Parsing).
- **-J, --json**: JSON-Format.
- **--output <list>**: Spalten auswählen (z.B. --output SIZE,NODE,STATE).
- **--help, --version** analog.

Beispiel lsmem (auf PC ohne Hotplug memory):

RANGE	SIZE	STATE	REMOVABLE	BLOCK
0x0000000000000000-0x000000007fffffff	2G	online	yes	0-15
0x0000000100000000-0x000000027fffffff	6G	online	yes	32-79

Zeigt 2 memory ranges (entsprechend wohl 2 DIMMs). "Removable yes" heißt vom Kernel als Hot-remove möglich (oft 'no' auf Standard-PC). "BLOCK 0-15" referenzieren Memory block device IDs.

Auf NUMA-Systemen würde Spalte NODE relevant sein, um Memory pro NUMA-Node zu sehen.

ps (Process Status)

Beschreibung: Zeigt eine Momentaufnahme der aktuellen Prozesse des Systems. ps kann verschiedene Ausgabeformate haben (BSD-Style, UNIX-style). Häufig nutzt man ps aux oder `ps -ef` .

Syntax:

ps [Optionen]

(Komplexität: ps hat UNIX Optionen (mit - und bündelbar), BSD Optionen (ohne - und können zusammen geschrieben werden) und GNU long Optionen. Z.B. ps aux ist BSD Syntax, ps -ef ist UNIX syntax – beide gängig, liefern ähnliches.)

Gängige Aufrufe:

- **ps aux** – Listet alle Prozesse mit Details (BSD format). Spalten: USER, PID, %CPU, %MEM, VSZ, RSS, TT (TTY), STAT, START, TIME, COMMAND.
- **ps -ef** – Listet alle Prozesse (UNIX format). Spalten: UID, PID, PPID, C (CPU usage), STIME (start time), TTY, TIME (CPU time), CMD.
- **ps -eF** – wie -ef, aber "extra full", zeigt threads als separate Linien und vollständige Kommandozeilen etc.
- **ps -ejH** – zeigt Hierarchie (Prozessbaum) mit Prozess-IDs.

- **ps -eo pid,cmd,%mem,%cpu --sort=-%mem** – Zeigt benutzerdefinierte Spalten (hier pid, Befehl, Speicher- und CPU% sortiert nach Speicherverbrauch absteigend).

Wichtige Optionen:

UNIX-style (präfix mit -, können kombiniert werden):

- **-e** – Alle Prozesse anzeigen (entspricht -A).
- **-f full** Format (mehr Spalten inkl. PPID, STIME, in Kombination z.B. -ef).
- **-F** – Extra full (noch mehr Infos: SZ, RSS, PSR CPU, etc.).
- **-h** – Ohne Header.
- **-H** – Hierarchie-Tree anzeigen (Indentation für Parent-Child).
- **-j** – Jobs format (PGID, SID etc. Spalten).
- **-l** – Long format (mehr spalten inkl. Flags, NI nice, LWP etc.).
- **-u <user>** – Prozesse eines bestimmten Benutzers.
- **-p <pidlist>** – Nur bestimmte PIDs.
- **-G <group>, -U <user>** – nach Gruppe/Benutzer filtern.

BSD-style (ohne -, einzeln hintereinander):

- **a** – Zeigt Prozesse von *allen Terminals* (nicht nur eigene).
- **x** – Zeigt auch prozesse *ohne Terminal* (Daemons). **ax** zusammen = alle Prozesse.
- **u** – Benutzerformat (inkl. Owner, CPU%, MEM%). => **aux** verbreitet.
- **j** – Jobs format (session, pgrp).
- **w** – breites Output (alle columns of command, use -ww to not truncate).

GNU long options (mit --):

- **--pid, --ppid, --user** etc. Alternative Filter.
- **--sort=<key>**: Sortiert Ausgabe nach Spalte(n). Z.B. --sort=-%cpu (absteigend CPU).
- **--no-headers**: Ohne Header (statt -h).
- **--forest**: ASCII-Art Bäumchen (ähnlich -H, aber macht "|_" und Einrückungen). Kombiniert mit e.g. -ef.

STAT Spalte (Prozessstatus):

Codes: R (running), S (sleep), D (uninterruptible sleep, i/o), Z (zombie), T (stopped), t

(gestoppt im Hintergrund), X (dead). Flags dahinter: < (high prio), N (low prio, nice>0), L (pages locked in mem), s (session leader), + (foreground process group in TTY).

sar (System Activity Reporter)

Beschreibung: Teil von sysstat (wie iostat, mpstat, etc.), sammelt und berichtet historische System-Auslastungsdaten: CPU, Speicher, Paging, I/O, Netzwerk etc. sar kann *laufende* Stats anzeigen (wenn mit Interval aufgerufen), oder *vergangene Tageswerte* aus Logs (/var/log/sysstat/...) ausgeben.

Syntax:

```
sar [Optionen] [Interval [Count]]
```

Ohne Parameter, sar liest per default die Tagesstatistiken (cumulated) vom laufenden Tag (falls sar cron job aktiv war). Mit Interval sar 5 5 z.B. 5x alle 5 Sekunden CPU etc.

Hauptoptionen (für Bereiche):

- **-u:** CPU-Auslastung (Default, wenn keine angegeben) – zeigt %user, %nice, %system, %iowait, %steal, %idle.
- **-r:** Memory (free, used, buffers, cache, commit, etc.).
- **-b:** I/O (transfer rates, i/o per sec, etc.).
- **-B:** Paging (pgpgin/s, pgpgout/s, fault, majflt...).
- **-W:** Swapping (swap in/out).
- **-q:** Queue length (runq, i/o queue) and load average.
- **-n DEV:** Network: DEV mode – zeigt Netzwerk-Interface Stats (rxpck/s, txpck/s, rxKB/s, txKB/s, etc.). -n ALL für alle Net-Statistiken.
- **-d:** Block device I/O (similar to iostat per device).
- **-p [DEV]:** CPU per processor (if SMP). e.g. sar -P ALL 1 3 – CPU usage per core.
- **-f file:** Lese Stats aus Log-Datei (z.B. /var/log/sysstat/sarXX).
- **-s HH:MM:SS, -e HH:MM:SS:** Start/End time for reading logs (z.B. 9 bis 18 Uhr Stats).

Beispiel (aktuelle CPU alle 1s):

```
$ sar 1 3
```

Linux 5.4.0 ... (8 CPUs)

```
02:00:00 PM CPU %user %nice %system %iowait %steal %idle
```

```
02:00:01 PM all 5.13 0.00 1.28 0.00 0.00 93.59  
02:00:02 PM all 7.14 0.00 1.02 0.00 0.00 91.84  
02:00:03 PM all 3.06 0.00 0.77 0.00 0.00 96.16  
Average: all 5.11 0.00 1.02 0.00 0.00 93.87
```

nohup (No Hangup)

Beschreibung: Lässt einen Prozess auch nach Logout weiterlaufen. nohup fängt das HUP-Signal (Hangup) ab, das normalerweise gesendet wird, wenn ein Terminal geschlossen wird. Mit nohup startet man ein Kommando so, dass es dieses Signal ignoriert und Standardausgabe ggf. in *nohup.out* umleite】.

Syntax:

```
nohup <Befehl> [Arg...] [&]
```

Funktionsweise:

- nohup sorgt dafür, dass der ausgeführte Prozess das SIGHUP ignoriert. Außerdem werden *stdout* und *stderr* umgeleitet in *nohup.out*, falls diese derzeit ans Terminal gebunden sind. (Wenn man & anhängt – also im Hintergrund – ist es üblich, die Ausgabe nicht am Terminal zu lassen).
- Üblicherweise kombiniert mit & (Hintergrund). Man loggt dann aus, und das Programm läuft weiter.

Beispiele:

- nohup long_running.sh & – Führt *long_running.sh* im Hintergrund aus, Ausgaben gehen in *nohup.out*. Der Befehl überlebt Logout. Die Shell gibt z.B. zurück: appending output to nohup.out.
- nohup python server.py > server.log 2>&1 & – Startet Python-Server im Hintergrund, leitet *stdout* und *stderr* in *server.log* (damit *nohup.out* nicht genutzt).

Hinweis: Wenn man bereits in einer Screen- oder tmux-Session ist, braucht man kein nohup; auch systemd-run oder Cron-Jobs sind Alternativen. Aber nohup ist ein schnelles Mittel.

mkswap/swapon/swapoff (Swap einrichten und verwalten)

- **mkswap** – Initialisiert eine Partition/Datei als Swapbereich. Syntax: mkswap [Optionen] <Gerät/datei>. Wichtige Option: -L <Label> (Swap-Label setzen), -U <UUID>. Beispiel: mkswap /dev/sda3.
- **swapon** – Aktiviert einen Swapbereich für Nutzung durch das System】. Nach mkswap muss Partition via swapon <Device> eingebunden werden (oder in

/etc/fstab eingetragen mit swap Typ). Option -a aktiviert alle in fstab markierten Swaps. --show zeigt aktuellen Swap an (Äquivalent cat /proc/swaps).

- **swapoff** – Deaktiviert Swap. Syntax: swapoff <Device> oder swapoff -a für alle.

Beispiele:

```
# mkswap -L "SWAP" /dev/sda3
```

Setting up swap space version 1, size = 8 GiB (some hex UUID)

```
# swapon /dev/sda3
```

```
# swapon --show
```

NAME	TYPE	SIZE	USED	PRIO
------	------	------	------	------

/dev/sda3	partition	8G	0B	-2
-----------	-----------	----	----	----

swapoff -a würde alle Swapbereiche aus dem Betrieb nehmen (z.B. vor Partition-Vergrößerung oder Kernel-Dump etc.).

free (Speicherübersicht)

Beschreibung: Zeigt eine Übersicht über den aktuellen Speicherverbrauch (RAM und Swap) im System. Es summiert total, used, free, and cached/buffered memory.

Syntax:

```
free [Optionen]
```

Optionen:

- **-h**: Human Readable (z.B. MiB/GiB).
- **-m, -g, -k**: Ausgabe in MiB, GiB, KiB.
- **-b**: in Bytes.
- **-t**: Fügt eine "Total" Zeile (RAM+Swap) hinzu.
- **-s N**: Aktualisiert alle N Sekunden (kontinuierliche Ausgabe wie watch).
- **-c N**: Wenn mit -s genutzt, stop nach N Aktualisierungen.
- **--si**: Benutzt 1000er statt 1024er Basis (KB = 1000 Bytes etc.).

Beispiel free -h:

	total	used	free	shared	buff/cache	available
Mem:	15Gi	5.2Gi	2.3Gi	1.1Gi	7.4Gi	8.8Gi
Swap:	2.0Gi	0.0Gi	2.0Gi			

available ist eine Kernelschätzung, wieviel noch für neue Prozesse verfügbar, unter Berücksichtigung von caches, etc.

vmstat (Virtual Memory and system stats)

Beschreibung: Zeigt verschiedene Systemmetriken: Speicher, Auslagerung, CPU, Interrupts, Context Switches etc. im zeitlichen Verlauf. vmstat ohne Parameter gibt Durchschnitt seit Boot; mit Intervall zeigt periodische Werte (Differenzen pro Intervall).

Syntax:

vmstat [Optionen] [Interval [Count]]

Wichtige Spalten (Standardausgabe):

- **procs:** *r* (Anzahl laufender Prozesse im Runqueue), *b* (blockierte Prozesse).
- **memory:** *swpd* (Swap genutzt), *free* (freier RAM), *buff* (Puffer), *cache* (Page Cache).
- **swap:** *si* (Swap-In KB/s), *so* (Swap-Out KB/s).
- **io:** *bi* (Block In, Blöcke gelesen von Disk/s), *bo* (Block Out, geschrieben/s).
- **system:** *in* (Interrupts/s), *cs* (Context Switches/s).
- **cpu:** *us* (% CPU in Userzeit), *sy* (% Systemzeit), *id* (% Idle), *wa* (% IO-Wait), *st* (% Steal bei virt.).

Optionen:

- **-S <unit>:** Werte in k (1000) oder K (1024) etc.
- **-a:** Zeigt zusätzlich Aktivität der *aktiven/inaktiven* Memory Listen (sin, sout).
- **-d:** Zeigt Disk-Statistiken (like iostat) statt VM stats.
- **-p <dev>:** Zeigt Partition-Statistiken (reads/writes tot und sek) für Device.
- **-s:** Summiert memory stats (eine Liste aller relevanten Zähler, analog /proc/meminfo).
- **-m, -t:** Variation der Ausgabe (meist uninteressant, -m memory in MB etc).

Beispiel (alle 2s):

```
$ vmstat 2 3
```

```
procs -----memory----- swap-- -----io---- system-- -----cpu-----
r b free buff cache si so bi bo in cs us sy id wa st
1 0 230000 120000 760000 0 0 5 10 150 300 5 191 3 0
```

```
0 0 229000 120000 760500 0 0 0 20 140 280 4 1 94 1 0
```

```
1 0 228500 120000 761000 0 0 0 30 160 320 6 1 92 1 0
```

Erste Zeile nach Header ist seit Boot (ohne parameter, oft uninteressant); weitere sind Echtzeit pro Intervall.

modinfo (Module Information)

Beschreibung: Zeigt Informationen über ein Linux-Kernelmodul (Loadable Kernel Module) – Version, Beschreibung, Lizenz, Abhängigkeiten, Parameter etc】 .

Syntax:

```
modinfo [Optionen] <Modulname oder Modul.ko>
```

Optionen:

- **-a, --author:** Nur Autor ausgeben.
- **-d, --description:** Nur Beschreibung.
- **-n, --filename:** Zeigt Pfad der Moduldatei im Filesystem.
- **-p, --parameters:** Listet die Parameter und ihre Typen/Beschreibungen.
- **-F <field>:** Gibt nur das spezifizierte Feld aus (alternativ obige Kurzoptionen). Mögliche Felder: author, description, license, parm, alias, etc.
- **-k <kernelversion>:** Fragt Infos für Kernel-Version (sucht in /lib/modules/<version>) statt laufendem Kernel.

Beispiel: modinfo e1000e (Intel NIC Treiber) gibt etwa:

```
filename: /lib/modules/5.4.0/kernel/drivers/net/ethernet/intel/e1000e/e1000e.ko
```

```
version: 3.2.6-k
```

```
license: GPL
```

```
description: Intel(R) PRO/1000 Network Driver
```

```
author: Intel Corporation, <e1000-devel@lists.sourceforge.net>
```

```
...
```

```
srcversion: 5C6D1A8....
```

```
alias: pci:v00008086d000010D3sv*sd*bc*sc*j*
```

```
...
```

```
parm: IntMode:Interrupt Mode (0=Auto, 1=MSI, 2=MSI-X) (int)
```

```
parm: InterruptThrottleRate:... (array of int)
```

Man sieht Dateipfad, Version, Parameter etc.

insmod/rmmod/modprobe (Kernelmodule laden/entladen)

- **insmod <pfad/modul.ko> [args]**: Lädt ein Modul in den Kernel. Nutzt genau Pfad; löst keine Abhängigkeiten auf.
- **rmmod <modulname>**: Entfernt ein geladenes Modul (wenn kein anderer in Benutzung). Option -f zum Erzwingen (nur falls Kernel CONFIG_MODULE_FORCE_UNLOAD).
- **modprobe [Options] <modulname> [args]**: Empfehlenswerter Befehl zum Laden eines Moduls und seiner Abhängigkeiten. Es durchsucht /lib/modules/<kernel>/modules.dep nach Dependencies. Auch zum Entfernen via modprobe -r <modul> (lädt auch Abhängigkeiten aus, falls unbenutzt).
 - Optionen: --dry-run oder -n testet nur, -v verbose, -q quiet, alias handling etc. modprobe Konfig steht in /etc/modprobe.d.
 - Bsp: modprobe e1000e (lädt Intel NIC modul + ggf. benötigte mii). modprobe -r e1000e entfernt es.

depmod (Dependency modules)

Beschreibung: Generiert die Modulabhängigkeitsdatei *modules.dep* basierend auf Modulen in /lib/modules/<version>. Wird meist beim Kernel-Install aufgerufen. Kann man manuell ausführen nach Hinzufügen eigener Module. Syntax: depmod -a (für alle).

Optionen:

- -a, --all (default): alle Kernel-Versionen verarbeiten.
- -A, --quick: nur verändern, wenn modules.dep älter als Module-Dateien (incremental update).
- -n: Ausgabe nach stdout statt in modules.dep schreiben (Test).
- -v: verbose.

sysctl (Query/Set Kernel Parameters)

Beschreibung: Anzeige oder Änderung von Kernel-Laufzeitparametern (in /proc/sys). sysctl ermöglicht bequemen Zugriff auf z.B. Netzwerktruning, VM, FS Settings zur Laufzeit.

Syntax:

sysctl [Optionen] [<Name>=<Wert> ...] or sysctl -p

Optionen:

- Ohne =: liest Parameter. Z.B. sysctl net.ipv4.ip_forward -> net.ipv4.ip_forward = 0.

- Mit name=value: setzt Parameter. Z.B. sysctl net.ipv4.ip_forward=1 aktiviert IP-Forwarding (sofort).
- **-w**: explizit Schreibmodus (kann man weglassen, war historisch).
- **-p [Datei]**: Lädt Einstellungen aus Datei (Default /etc/sysctl.conf) und wendet a】 . Man ruft sysctl -p typ. beim Boot (init-Skripte) oder manuell nach Bearbeiten der config.
- **-a**: Alle Parameter anzeigen (riesige Liste). sysctl -a | grep ipc etc.
- **-A, --pattern <regex>**: Filtert mit Regex (statt manuell grep).
- **-e**: Ignoriert unbekannte Keys (useful mit -p, sonst bricht ab).
- **-N**: Nur Namen, keine Werte anzeigen (list keys).
- **-q**: Quiet (unterdrückt Bestätigungsmeldungen).

Parameter-Keys entsprechen /proc/sys Pfaden, z.B. /proc/sys/net/ipv4/ip_forward -> key net.ipv4.ip_forward.

Beispiele:

- sysctl vm.swappiness -> zeigt swappiness (z.B. vm.swappiness = 60).
- sysctl -w vm.swappiness=10 -> setzt auf 10 (weniger Swapneigung).
- sysctl -p /etc/sysctl.d/99-sysctl.conf -> läd Parameter aus der angegebenen Datei.

dmesg (Dump kernel message buffer)

Beschreibung: Zeigt die Nachrichten des Kernel-Ringspeichers (Kernellog) a】 . Hier stehen Boot-Meldungen und Hardware-Events (z.B. USB eingesteckt, Treiber Ausgabe). dmesg ohne Optionen druckt alle aktuellen Kernelmeldungen.

Syntax:

dmesg [Optionen]

Optionen:

- **-T**: Zeigt menschenlesbare Zeitstempel (Jahr-Monat-Tag Stunde:Minute:Sekunde) statt der raw Sekunden seit Boo】 . (Achtung: Kann ungenau sein, da Kernel keine Zeitzone kennt; -T rechnet anhand aktueller TZ).
- **-l <level>, --level**: Filtert Nachrichten bestimmter Log-Levels. Level: emerg, alert, crit, err, warn, notice, info, debug. Z.B. dmesg -l err,warn nur Fehler und Warnungen.

- **-k, --kernel:** Zeigt nur Kernel-Messages (Standard). **-u, --userspace:** Nur Benutzerraum (Logeinträge vom printk_user). Kaum genutzt, da meiste aus Kernel.
- **-H, --human:** **Pager-Modus** (lässt scrollen, farbig Highlighting). Interaktiv (Space/PgUp etc.).
- **-w, --follow:** "Tail -f" Modus für dmesg – neue Kernelmeldungen laufend anzeigen.
- **-s N:** Buffer-Größe (Bytes) die gelesen wird (Default könnte begrenzt sein). dmesg -s 1000000.

Beispiel (gekürzt):

```
[ 0.000000] Linux version 5.4.0-80-generic ...
[ 0.345678] PCIe Bus 0000:00: root hub ...
[ 1.234567] usb 1-2: new high-speed USB device number 3 ...
[ 1.345678] ata1: SATA link up ...
[ 10.567890] Bluetooth: RFCOMM ver 1.11
```

dmesg -T zeigt z.B. [Mon Oct 11 14:23:45 2021]

Typisch ruft man dmesg | grep -i error um nach Kernel-Fehlern zu suchen, oder dmesg -w in einem Terminal um z.B. neue USB/Hardware Events sofort zu sehen.

Dienstverwaltung und Systemstart

systemctl (Control systemd)

Beschreibung: Haupt-Werkzeug zur Steuerung des *systemd*-Initsystems (heutzutage auf vielen Linux-Distributionen). systemctl verwaltet **Services** (Start/Stop/Enable), **Daemon-Konfiguration, System-Zustände** (Reboot/Shutdown), **Unit**-Status und Logs, etc】 .

Syntax:

systemctl [Optionen] <Sub-Befehl> [Unit/...]

Units können Services (.service), Mounts (.mount), Timer (.timer), Sockets (.socket), etc. sein. Oft kann man den Suffix weglassen (z.B. "ssh" wird als "ssh.service" erkannt).

Wichtige Sub-Befehle (Service Management):

- **start <service>, stop <service>** – Startet oder stoppt einen Dienst *sofortfile-4fdk7rvx39azjretinarqb】 . Bsp: systemctl start apache2.
- **restart <service>** – Stop + Start (auch wenn nicht lief, wird gestartet).

- **reload <service>** – Lädt Konfiguration neu (sendet i.d.R. SIGHUP). Muss vom Service unterstützt werden (in Unit als ReloadSignal oder ExecReload definiert). Bsp: systemctl reload nginx.
- **status <service>** – Zeigt Status des Dienstes: Läuft er? Letzte Startmeldung, ggf. Auszug aus Journal-Log】 .
- **enable <service>** – Aktiviert den Dienst beim Boot (erstellt Symlinks in passenden WantedBy-Verzeichnissen).
- **disable <service>** – Deaktiviert Autostart.
- **is-enabled, is-active** – Prüfen ob Unit aktiviert ist / gerade aktiv ist (exit 0/1).
- **mask <service>** – "Maskiert" den Service, d.h. verhindert Start vollkommen (setzt Link auf /dev/null). *Unmask* zum Aufheben.
- **daemon-reload** – Falls man eine Unit-Datei geändert oder neu hinzugefügt hat, diese neu einlesen (Konfig neu laden, ohne Dienste neu zu starten).
- **kill <service>** – Sendet ein Signal an alle Prozesse der Service-Unit (z.B. systemctl kill -s SIGUSR1 myapp.service).

System Commands:

- **list-units** – Listet alle geladenen Einheiten (nur aktive standardmäßig). --all um Inaktive auch zu sehen, --type=service filtern auf Services, etc.
- **list-unit-files** – Listet Unit-Dateien und ob sie enabled/disabled/etc. sind (Installationszustand).
- **list-dependencies <unit>** – Zeigt Abhängigkeiten (Requires/Wants) Baum für Unit. --reverse zeigt welche Units diese Unit benötigen.
- **show <unit>** – Gibt alle Properties von Unit aus (oder subset via -p).
- **cat <unit>** – Zeigt den Inhalt der Unit-Datei (und Drop-Ins).
- **edit <unit>** – Öffnet Editor für Drop-In Snippet, um diese Unit zu überschreiben/erweitern.
- **preset <service>** – Setzt enable/disable gemäß Distribution-Vorgabe (in /usr/lib/systemd/system-preset/).

Targets (Runlevels):

- **isolate <target>** – Wechselt in ein target (beendet andere). Z.B. systemctl isolate rescue.target (Einzeluser-Modus ähnlich runlevel 1).
- **set-default <target>** – Legt Default Boot-Target fest (z.B. graphical.target vs multi-user.target).

- **get-default** – zeigt aktuelles Defaulttarget.

Power Management:

- **reboot, poweroff, halt** – führen Neustart bzw. Herunterfahren aus (rufen intern entsprechende target isolation).
- **suspend, hibernate, hybrid-sleep** – starten Energiesparmodus.

Journal/Logging Integration:

- **status** (wie oben) zeigt bereits Log-Auszug.
- **journalctl** (separater Befehl, nicht systemctl subcommand) – zum Betrachten der systemd Logs. Z.B. journalctl -u <service> zeigt Logs zu einem Service, -f zum folgen, --since "2021-10-01" ab bestimmtem Datum etc.

Beispiele:

- systemctl enable --now nginx – Aktiviert **und** startet nginx sofort (Option --now führt gleich start nach enable aus).
- systemctl disable apache2.service – Entfernt Apache2 aus Autostart.
- systemctl restart networking – Dienst neu starten (bei Netz aufpassen).
- systemctl list-units --failed – Zeigt nur fehlgeschlagene Units.
- systemctl mask firewalld; systemctl stop firewalld – Verhindert Start des Firewalld dauerhaft und stoppt ihn sofort (z.B. falls man anderen Firewall nutzt).

service (SysVinit Compatibility)

Beschreibung: Der Befehl service war bei SysVinit/Upstart gebräuchlich, um Start/Stop-Skripte in /etc/init.d aufzurufen. Auf systemd-Systemen ruft service intern systemctl auf. Man kann aber service <name> start/stop/status noch benutzen! `service ssh status`

Syntax:

service <ServiceName> [start|stop|restart|status|...]

Beispiel: service ssh status ruft systemctl status ssh.service auf.

(Auf SysVinit ohne systemd würde es das init.d Script ausführen.)

chkconfig (Runlevel Konfiguration - SysV kompatibel)

Beschreibung: Ein Werkzeug zum Verwalten der Start-/Stopp-Links von SysV-Init-Skripten über verschiedene Runlevels (0-6). Auf systemd-Systemen gegenstandslos (meist vorhanden für Kompatibilität, was auf systemctl enable/disable mappt).

Wichtigste Befehle:

- `chkconfig --list [name]` – Listet alle Dienste und ob sie in runlevel 0-6 on/off sind
file-4fdk7rvx39azjretinarqb】 . Mit Name nur diesen.
- `chkconfig name on` (bzw. off) – Schaltet Service in allen entsprechenden runlevels an oder aus (Standard runlevels 2-5 on für on).
- `chkconfig name --level 35 on` – Feinsteuerung: aktiviert Dienst in Runlevel 3 und 5.
- `chkconfig name reset` – Setzt Standard (wie im Init-Skript Header vorgesehen).

*(Auf systemd delegiert chkconfig intern an systemctl enable/disable. Für Nostalgie, wissen was es tat.)

Zeitplanung (Cron und At)

crontab (Cron Table Management)

Beschreibung: Planen wiederkehrender Aufgaben. crontab bearbeitet die Aufgabenliste des aktuellen Benutzers (oder eines anderen mit -u) im Cron-Daemon. Jede Zeile in crontab -e hat Format "*Min Stunde Tag Monat Wochentag Befehl*". Der Cron-Dienst prüft pro Minute die Tabellen und führt fällige Kommandos auf file-4fdk7rvx39azjretinarqb】 .

Befehle:

- `crontab -e` – Editiert die Crontab des aktuellen Benutzers (öffnet \$EDITOR). Beim Speichern wird die neue Tabelle dem Cron-Daemon übergeben file-4fdk7rvx39azjretinarqb】 .
- `crontab -l` – Listet die aktuelle Crontab auf file-4fdk7rvx39azjretinarqb】 .
- `crontab -r` – Entfernt (löscht) die aktuelle Crontab komplett. (Vorsicht, kein Undo).
- `crontab -u username -e/-l/-r` – Als root: Bearbeitet/liest/löscht die Crontab eines anderen Benutzers.

Cron Format Erinnerung:

Minute(0-59) Hour(0-23) Day(1-31) Month(1-12) Weekday(0-7) Command

0 2 * * * /usr/bin/backup.sh # täglich 02:00

*/5 9-17 * * 1-5 echo "Working..." # Werktags 9-17 Uhr alle 5 min

*=jeder Wert, Komma getrennte Listen, Intervalle mit -, Step mit /.

Spezial @-Strings: @reboot, @hourly, @daily, @yearly etc.

Anacron: Cron führt Jobs aus, wann Maschine an ist. Für nicht ständig laufende Rechner nutzt man anacron (Tagesjobs nachholen).

at (One-time scheduled task)

Beschreibung: Legt einmalige Ausführungen von Befehlen zu einer bestimmten Zeit in der Zukunft fest [4fdk7rvx39azjretinarqb]. at nimmt eine Zeitangabe entgegen und liest dann von STDIN die Befehle, die ausgeführt werden sollen (oder mit -f aus einer Datei). Der atd Daemon muss laufen.

Verwendung:

```
echo "/path/to/script.sh" | at 02:30 # führt Script heute Nacht 2:30 aus
```

Man ruft at <Zeit> auf, landet in einer Stdin-Eingabe (end mit Ctrl+D) oder speist via echo/<<EOF>>.

Zeitformat: Z.B. at 14:00 (heute 14 Uhr, falls schon vorbei dann morgen), at 11pm, at midnight, at now + 5 days, at 08:00 12.12.2025 (8:00 am 12 Dec 2025). Kann englische Wörter: tomorrow, noon, teatime(16:00) etc.

Befehle:

- atq – Zeigt anstehende at-Jobs (Queue) des aktuellen Userfiles [4fdk7rvx39azjretinarqb]. (Listet Job-IDs und geplante Zeit). Root sieht alle oder atq -q <Queue> für bestimmte Batch-Queues.
- atrm <JobID> – Entfernt einen geplanten Job aus der Queue. (JobID sieht man in atq).
- batch – Alias zu at -q b now (führt Jobs aus, wenn Load Average niedrig ist). "Batch" schiebt den Auftrag in Batch-Queue, läuft sobald System nicht ausgelastet (LA<0.8 default).

Beispiel:

```
$ at now + 1 minute
```

```
warning: commands will be executed using /bin/sh
```

```
at> echo "Hello after 1 min" >> /tmp/test.txt
```

```
at> <EOT>
```

```
job 5 at Fri Oct 15 16:41:00 2021
```

Dann innerhalb einer Minute atq zeigt e.g. 5 Fri Oct 15 16:41:00 2021 a user. Nach Ausführung landet Text in Datei.

(Zugriff: /etc/at.allow /etc/at.deny definieren wer at nutzen darf – analog cron.allow/deny.)

Drucken

lpr (Line Print Request)

Beschreibung: Sendet eine Datei (oder STDIN) an den Drucker (Print-Job einreihen). lpr ist Teil des BSD-Linux-Drucksystems (CUPS stellt Kompatibilität bereit). Alternative: lp (System V style).

Syntax:

lpr [Optionen] [Dateien...]

Wichtige Optionen:

- **-P <Druckernname>**: Wählt einen bestimmten Drucker aus (sonst Standarddruckerfile-4fdk7rvx39azjretinarqb】). Druckernamen findet man via lpstat -a oder CUPS Webinterface.
- **-# <Num>**: Anzahl Kopien drucken (z.B. -#3 dreifach).
- **-h**: Unterdrückt Banner-Seite (header page) falls aktiviert.
- **-r**: Löscht die Datei(en) nach dem Versenden an den Spool (rarely used).
- **-m**: Sendet E-Mail Benachrichtigung nach Druck (sofern eingerichtet).
- **-o <Option>=<Wert>**: Drucker-spezifische Optionen übergeben (z.B. -o landscape für Querformat, oder -o media=A4 oder -o sides=two-sided-long-edge für Duplex). Diese Optionen sind je nach Drucker Treiber definierbar (CUPS options).
- **-C <Klasse>, -J <Jobname>, -T <Titel>**: Legen Klassenname, Jobname, Titel für Banner etc. fest.
- **--help** (GNU lpr) usw.

Beispiele:

- lpr report.pdf – Schickt *report.pdf* an Standarddrucker.
- lpr -P HP_LaserJet -#2 invoice.txt – Druckt *invoice.txt* 2-fach auf Drucker "HP_LaserJet".
- echo "Test Page" | lpr – Druckt STDIN (den Text "Test Page").
- lpr -o sides=two-sided-long-edge thesis.pdf – Duplexdruck beidseitig.

(Verwandt: *lpq* zeigt Druckerqueue, *lprm* löscht Jobs aus der Queue.)

Lokale Systemkonfiguration

date (Anzeige/Setzen Datum/Uhrzeit)

Beschreibung: Zeigt das aktuelle Datum und die Uhrzeit an, oder stellt sie ein. Auch Formatierung in beliebigem Muster ist möglich.

Syntax:

date [Optionen] [+Format]

date -s "STRING" (setzen)

Optionen / Verwendung:

- **Ausgabe formatieren:** date "+FORMAT" – Mit Format-Pattern ähnlich strftime:
%Y (Jahr), %m Monat(01), %b Monat (Kurzname), %d Tag, %H Stunde (24h), %I (12h), %M Minute, %S Sek, %A Wochentag, %a (kurz), %Z Zeitzone, %z offset, %T = %H:%M:%S, %F = %Y-%m-%d, etc. Z.B. date "+%d.%m.%Y %H:%M:%S" -> "05.10.2025 14:30:00".
- **Systemzeit stellen:** date -s "2025-10-05 14:30:00" – Setzt Datum. Braucht Root. (Alternative: date --set="...").
- **Hardware-Uhr stellen:** In Linux erfolgt date auf Systemzeit (in Kernel, RTC separiert). Um auch BIOS/RTC Uhr zu setzen, hwclock --systohc (setze Hardware Clock nach System Clock). Umgekehrt hwclock --hctosys.
- **Universale Zeit:** date -u zeigt in UTC. date -u -s "2025-10-05 12:30:00" setzt in UTC.
- **RFC-3339 Format:** date --rfc-3339=ns etc. Standard ISO.
- **Unix Timestamp:** date +%s gibt Sekunden seit 1.1.1970 (UTC).

Beispiele:

- date -> "Tue Oct 5 14:30:00 CEST 2025".
- date +"%Y-%m-%d_%H-%M-%S" -> "2025-10-05_14-30-00".
- sudo date -s "@\$((\$(date +%s) + 3600))" -> Systemzeit um eine Stunde vorstellen (Beispiel mit timestamp math).

timedatectl (Zeiteinstellung via systemd)

Beschreibung: Befehl von systemd zum Anzeigen und Ändern der Systemzeit, Hardware-Uhr, Zeitzone und NTP-Synchronisatiofile-4fdk7rvx39azjretinarqb】 .

Verwendung:

- **timedatectl** (ohne args): Zeigt aktuellen Status: Lokale Zeit, UTC, RTC, Zeitzone, ob NTP aktiv ist etcfile-4fdk7rvx39azjretinarqb】 .
- **timedatectl set-time "YYYY-MM-DD HH:MM:SS"** – Setzt Systemzeit (wenn kein NTP aktiv).
- **timedatectl set-timezone <Zone>** – Ändert Zeitzone. Z.B. timedatectl set-timezone Europe/Berlin. (Verfügbare Zonen unter /usr/share/zoneinfo/, oder timedatectl list-timezones).

- **timedatectl set-ntp true/false** – Aktiviert bzw. deaktiviert die automatische Zeitsynchronisation (via systemd-timesyncd oder anderes NTP). Bei true startet es den Dienst falls vorhanden.
- **timedatectl set-local-rtc 1/0** – Legt fest, ob die Hardware-Uhr (RTC) in **lokaler Zeit** läuft (1) oder in UTC (0, Standard). Windows dual-boot oft RTC=lokal (dann 1), sonst belässt auf 0=UTC um Probleme zu vermeiden.

Beispiele:

\$ timedatectl

Local time: Tue 2025-10-05 14:30:00 CEST

Universal time: Tue 2025-10-05 12:30:00 UTC

RTC time: Tue 2025-10-05 12:30:00

Time zone: Europe/Berlin (CEST, +0200)

System clock synchronized: yes

NTP service: active

RTC in local TZ: no

timedatectl set-time "2025-12-24 18:00:00" stellt Heiligabend 18:00. timedatectl set-timezone UTC wechselt Zeitzone auf UTC.

localectl (Locale and Keyboard Layout)

Beschreibung: Verwaltet System-Locale (Sprache, Gebietsschema) und Tastaturlayout-Einstellungen in systemd-basierten Systemefile-4fdk7rvx39azjretinarqb】 . Es kann die Variablen in /etc/locale.conf sowie Konsolen-Tastaturlayout in /etc/vconsole.conf setzen. (Für X/Wayland Tastatur greift localectl in /etc/X11/xorg.conf.d/ oder mittels setxkbmap separat).

Verwendung:

- **localectl** (ohne args): Zeigt aktuelle Locale(s) und Keymap. Z.B.:
 - System Locale: LANG=de_DE.UTF-8
 - VC Keymap: de
 - X11 Layout: de
 - X11 Model: pc105 etc.
- **localectl list-locales** – Listet alle verfügbaren Locale, die generiert sind (die in /usr/lib/locale oder /etc/locale.gen markiert).
- **localectl set-locale LANG=<locale> [VAR=locale...]** – Setzt System-Locale. Z.B. localectl set-locale LANG=en_US.UTF-8. Optional mehrere, z.B. LC_TIME=en_GB.UTF-8. Wirksam nach Neustart oder neu login shells.

- **localectl list-keymaps** – Listet verfügbare Console-Keymaps (z.B. us, de, ...).
- **localectl set-keymap <map> [<toggle>]** – Setzt Tastaturlayout für Konsole (vconsole) und ggf. X11, falls X11 Layout nicht getrennt definiert. Z.B. localectl set-keymap de.
- **localectl set-x11-keymap <layout> [model] [variant] [option]** – Spezifisch X11 (Graphical) Layout/Modell setzen. Z.B. localectl set-x11-keymap us pc104 "" terminate:ctrl_alt_bksp.

Beispiel:

localectl set-locale LANG=fr_FR.UTF-8 (Systemsprache auf Französisch).

localectl set-keymap us (Konsole auf US-Layout).

localectl set-x11-keymap de pc105 nodeadkeys (Grafische Tastatur auf Deutsch mit Variante "nodeadkeys").

(Die Konfigurationen landen in /etc/locale.conf, /etc/vconsole.conf, /etc/X11/xorg.conf.d/00-keyboard.conf entsprechend.)

Netzwerk-Konfiguration

ip (Netlink Interface Config)

Beschreibung: Moderner Befehl zur Verwaltung von Netzwerkinterfaces, IP-Adressen, Routing, Tunnels etc. (Nachfolger von ifconfig/route). ip hat Unterkommandos für verschiedene Objekte: **link** (Netzwerkinterface), **addr** (IP-Adressen), **route**, **neigh** (ARP/NDP Nachbarn), **rule** (Policy Routing), **maddr** (Multicast), **addrlabel**, **netns** (Net Namespaces), **link set** (Interface-Settings).

Syntax (Grundstruktur):

ip [Options] <Object> <Subcommand> [Arguments...]

z.B. ip link show, ip addr add, ip route list, etc.

Häufige Befehle/Beispiele:

- **Interface anzeigen:** ip link show [dev <iface>] – Listet alle Netzwerkinterfaces (oder ein spezifisches). Ausgabe enthält Index, Name, MAC, MTU, Flags (UP,LOWER_UP=Link OK, MULTICAST), etc.
- **Interface aktivieren/deaktivieren:** ip link set <iface> up / down – entspricht `ifconfig iface up/downfile-4fdk7rvx39azjretinarqb` .
- **Adresse zuweisen:** ip addr add <IP>/<Prefix> [broadcast +] dev <iface> – z.B. ip addr add 192.168.1.100/24 dev eth0. broadcast + lässt broadcast aus Prefix berechnen (optional).
- **Adresse löschen:** ip addr del <IP>/<Prefix> dev <iface>.

- **Adressen anzeigen:** ip addr show [dev <iface>] – Listet IPv4/IPv6 Adressen auf
Interfacefile-4fdk7rvx39azjretinarqb】 . Zeigt auch Scope (global/link), flags (tentative, dynamic).
- **Standardroute setzen:** ip route add default via <Gateway-IP> [dev <iface>] – z.B. ip r add default via 192.168.1.1 dev eth0.
- **Statische Route setzen:** ip route add <Netz>/<Präfix> via <nächsterHop> dev <iface>.
- **Routen auflisten:** ip route show (oder ip r).
- **ARP Cache:** ip neigh show – zeigt ARP/ND-Cache (Neighbors).
- **ARP Eintrag fix setzen:** ip neigh add <IP> lladdr <MAC> dev <iface> nud permanent.
- **Secondary IPs / Aliase:** Einfach mehrfach ip addr add auf selbes dev ausführen (oder use Label).
- **MTU ändern:** ip link set dev <iface> mtu 1400.
- **MAC ändern:** ip link set dev <iface> address <NewMAC> (Down interface first).
- **Promisc Mode:** ip link set dev <iface> promisc on.
- **Routing Policy (advanced):** ip rule und ip route add table <n> for source-based routing etc.
- **Network Namespace mgmt:** ip netns (add, del, exec, etc – advanced).

Options global: -s (stats), -d (details), -br (brief output), -c (color output). Z.B. ip -br addr zeigt Interfaces in einzeiligem Kurzformat.

Beispiele:

- ip -br link ->

```
lo      UNKNOWN    127.0.0.1/8 ::1/128
eth0     UP        192.168.1.10/24 fe80::.../64
```

- ip addr add 10.0.0.1/16 dev eth1 (fügt IP).
- ip route show (z.B. default via 192.168.1.1 dev eth0 proto dhcp metric 100 etc.).
- ip neigh show (z.B. 192.168.1.1 dev eth0 lladdr aa:bb:cc:dd:ee:ff REACHABLE).

(*ifconfig (net-tools) alternative: ip ist ausführlicher und bevorzugt.*)

ifconfig (Interface Config – alt)

Beschreibung: Älteres Werkzeug aus net-tools (weitgehend ersetzt durch ip). ifconfig kann Interfaces anzeigen, IP setzen, MTU ändern, promiscuous Mode etc., aber unterstützt z.B. keine IPv6 out-of-the-box je nach Version.

Syntax:

ifconfig [Interface] [Optionen]

Beispiele:

- ifconfig – zeigt alle aktiven (UP) Interfaces mit IPv4 etc.
- ifconfig -a – zeigt *alle*, inkl. down Interfaces.
- ifconfig eth0 192.168.0.5 netmask 255.255.255.0 broadcast 192.168.0.255 – setzt IP, Maske, Bcast.
- ifconfig eth0 up / down – Interface an/aus.
- ifconfig eth0 mtu 1400 – MTU ändern.
- ifconfig eth0 promisc – Promiscuous Mode an. -promisc aus.
- ifconfig eth0 hw ether 00:11:22:33:44:55 – MAC-Adresse setzen.

(*ifconfig ist nicht mehr standard installiert in neueren Distros; ip commands stattdessen nutzen.*)

iwconfig (Wireless Config – alt)

Beschreibung: Ähnlich ifconfig, aber für Wireless LAN Schnittstellen (Legacy tool aus wireless-tools). Zeigt/ändert ESSID, Mode, Frequenz, Schlüssel etc. Heutzutage ersetzt durch iw (nl80211), oder NetworkManager.

Beispiele (falls vorhanden):

- iwconfig wlan0 – zeigt ESSID, Mode (Managed/Ad-Hoc/Master), Channel/Freq, Bitrate, Tx-Power, Signal Level, Encryption key on/off, etc.
- iwconfig wlan0 essid "WLAN-Name" – mit offener SSID verbinden (sofern AP ohne Sicherheit).
- iwconfig wlan0 key s:password – setzt WEP Key (veraltet).
- iwconfig wlan0 mode Monitor – Interface in Monitor Mode (für Sniffing).

(Iwconfig kann kein WPA2; da nutzt man wpa_supplicant oder NM, etc.)

nmcli (NetworkManager CLI)

Beschreibung: Kommandozeilen-Interface zum NetworkManager-Dienst, um Netzwerkkonfiguration (insb. Desktop-Umfeld) zu betrachten und steuerfile-

4fdk7rvx39azjretinarqb】 . Kann Verbindungen (profiles) verwalten, aktivieren, WLAN scannen, etc.

Verwendung (Struktur):

nmcli [general|networking|radio|connection|device] <aktion> [Parameter]

Beispiele:

- **Status anzeigen:** nmcli general status – Gesamter NM-Status. nmcli connection show – listet gespeicherte Verbindungen (Profiles). nmcli device status – zeigt Geräte und ob connected und welches Profil.
- **WLAN scannen:** nmcli device wifi list – listet verfügbare WLANs.
- **Verbinden WLAN:** nmcli device wifi connect "SSID" password "passwort" – erstellt ggf. Connection-Profil und verbindet.
- **Verbindung aktivieren/deaktivieren:** nmcli connection up <Name> / down <Name>. Oder nmcli device disconnect <iface> (z.B. nmcli device disconnect wlan0).
- **Neue Verbindung anlegen:** nmcli connection add type ethernet ifname eth0 con-name "Wired Manual" ip4 192.168.5.5/24 gw4 192.168.5.1.
- **IPv4 ändern laufend:** nmcli connection modify "Wired connection 1" ipv4.addresses 192.168.1.50/24 ipv4.gateway 192.168.1.1 ipv4.method manual und dann up.
- **Ethernet an/aus:** nmcli networking off (setzt NM generell offline) oder nmcli radio wifi off (schaltet nur WLAN-Funk aus).

Beobachten:

- nmcli connection show --active – nur aktive Conns.
- nmcli -p device status – pretty (ncurses-like) output.

ethtool (NIC Driver Tool)

Beschreibung: Zeigt und ändert Einstellungen von Ethernet-Netzwerkkarten (Treiber-Level): z.B. Geschwindigkeit, Duplex, Auto-Negotiation, Wake-on-LAN, Offload-Funktionen, Statistikzählefile-4fdk7rvx39azjretinarqb】 .

Verwendung:

- ethtool <iface> – Zeigt Standard-Eigenschaften: unterstützt Geschwindigkeiten (Supported link modes), aktuelle Speed/Duplex, Auto-negotiation on/off, Ports (TP, etc.), driver name, firmware-version, bus-info, etc.

- `ethtool -s <iface> speed <Mb/s> duplex <half/full> autoneg <on/off>` – Setzt feste Geschwindigkeit/Duplex. (z.B. `ethtool -s eth0 speed 100 duplex full autoneg off`). Vorsicht: muss vom Link Partner unterstützt werden, sonst Link down.
- `ethtool -k <iface>` – Zeigt Kernel-Offload-Funktionen (tx-checksumming, scatter-gather, TSO etc.) und ob an/aus. -K ... (capital K) schaltet diese: z.B. `ethtool -K eth0 gro off` (GRO ausschalten).
- `ethtool -i <iface>` – Infos zum Treiber (Driver name, version, firmware, bus).
- `ethtool -p <iface> [N]` – "Physically identify": Blinkt die LED am Port für N Sekunden (Standard 15), falls Karte das unterstützt. Hilft um Kabelport zu finden.
- `ethtool -S <iface>` – Ausgabe von Statistikzählern (packets, errors, etc. vom Treiber).
- `ethtool -g <iface>` – Ring Puffer Größen (RX/TX rings). -G zum Setzen.
- `ethtool -a <iface>` – Flow-Control (Pause Frame) Status (Autoneg etc.), -A zum Setzen on/off rx/tx.
- `ethtool -r <iface>` – Veranlasst die Karte, Link neu auszuhandeln (Retrain).
- `ethtool --identify <iface>` – alias für -p (Blinken).

Beispiele:

- `ethtool eth0` -> zeigt z.B. Speed: 1000Mb/s, Duplex: Full, Auto-negotiation: on, etc.
- `ethtool -s eth0 speed 100 duplex half autoneg off` -> zwingt 100Mbit Half-Duplex (nur Test oder Legacy).
- `ethtool -p eth1 10` -> blinkt LED an eth1 für 10 Sekunden.
- `ethtool -K eth0 tso off` -> schaltet TCP Segment Offloading aus (im Troubleshooting falls Problem mit Offload).

hostnamectl (Control hostname in systemd)

Beschreibung: Zeigt oder ändert den System-Hostname und verwandte Settings (wie Chassis Typ, Icon). hostnamectl ist Teil von systemd.

Verwendung:

- **hostnamectl** (ohne args): Zeigt Systeminfo: Static hostname, Transient (via DHCP), Pretty (human name), und Kernel: Operating System, Kernel, Architecture, etc.
- **hostnamectl set-hostname <name> [--pretty|--static|--transient]** – Setzt den Hostnamen. Ohne Flag wird *static* und *transient* gleichermaßen gesetzt. *static* =

in /etc/hostname gespeichert (persistenter Name), *transient* = Laufzeit (kann z.B. DHCP ändern). *pretty* = nur für Anzeigezwecke (z.B. mit Großbuchstaben/Leerzeichen erlaubt).

Beispiel: hostnamectl set-hostname server42 – setzt static+transient auf "server42".

hostnamectl set-hostname "My Home PC" --pretty – setzt Pretty-Name.

- **hostnamectl set-icon-name <icon>** – Setzt Icon (z.B. computer-desktop, rein kosmetisch in einigen GUIs).
- **hostnamectl set-chassis <type>** – Setzt Gehäuse-Typ: desktop, laptop, server, tablet, embedded, etc. (Auch nur informative, kann z.B. in GNOME Systeminfo auftauchen).

Zusatz:

- hostnamectl status (alias ohne args) – Info.
- Traditional hostname Befehl (ohne ctl) zeigt oder setzt nur transient (bis Reboot).

Netzwerk-Fehlerdiagnose und Name-Auflösung

netcat (nc) (TCP/UDP Swiss Army Knife)

Beschreibung: netcat (oder nc) ist ein universelles Netzwerk-Tool, um Verbindungen zu testen, Daten zu senden/empfangen, Ports zu lauschen etcfile-4fdk7rvx39azjretinarqb】. Kann sowohl Client als auch rudimentärer Server sein.

Verwendung:

- **Verbindung als Client:** nc [options] <Host> <Port> – Baut TCP-Verbindung zu Host:Port auf, leitet stdin -> Socket, und Socket -> stdout. So kann man z.B. echo "HEAD / HTTP/1.0" | nc example.com 80 (HTTP-Request senden). Oder nc time.nist.gov 13 (Daytime Protocol). Wenn man interaktiv ausführt, kann man manuell tippen (Telnet-Ersatz).
- **Port Listening (Server):** nc -l -p <Port> [Host] – Horcht auf Port (optional an gegebene IP gebunden). Bsp: nc -l -p 1234 lauscht auf TCP/1234 auf alle Interfaces; was reinkommt wird auf stdout ausgegeben, was man tippt geht an Client.
- **UDP mode:** nc -u – Statt TCP benutzt UDP. Bsp: nc -u 192.168.1.5 53 könnte man UDP Query schicken (z.B. an DNS server).
- **Port Scan:** nc -z -v host startport-endport – "Zero-I/O mode" nur scannen, nicht verbinden. Z.B. nc -z -v 192.168.1.10 1-1024 zeigt offene Ports. (Netcat als Portscanner ist möglich, aber Nmap dafür besser).
- **Timeout:** -w <secs> – Wartezeit bis Timeout für Verbindungen/Eingaben.

- **Keep-open (inetd style):** In GNU nc -k (listen bleibt offen nach Verbindungsende, auf neue).
- **Exec:** -e <prog> – *Unsichere Option* (nicht immer kompiliert, oft disabled aus Sicherheitsgründen): Leitet die I/O eines Programms an den Socket. So kann man z.B. remote shell spawn: nc -l -p 5000 -e /bin/bash. (Equivalent zum berüchtigten rm -rf example). Modern: use socat or ssh for shells.

Beispiele:

- **Chat/Echo Test:** Terminal A: nc -l -p 5000, Terminal B: nc localhost 5000 -> alles was A tippt sieht B und umgekehrt.
- **Datei senden:** Auf Empfänger: nc -l -p 4500 > out.txt, auf Sender: nc destIP 4500 < file.txt. (Netcat baut TCP auf und streamt Datei).
- **HTTP Request:** printf "GET / HTTP/1.1\r\nHost: example.com\r\n\r\n" | nc example.com 80 – zeigt HTML der Homepage (HTTP/1.1 erfordert Host-Header).
- **UDP Ping** (ohne Antwort): echo -n "Hello" | nc -u 192.168.1.5 9000.
- **Port scan:** nc -z -v 192.168.1.20 20-30 -> tries FTP-data, FTP etc, output open/closed.

(*Netcat Implementierungen: Traditional vs OpenBSD netcat vs GNU netcat ncat (part of nmap) – Optionen leicht variierend. Z.B. OpenBSD netcat hat -l und -p zusammen als one option or so, etc.*)

iftop (Interface TOP, Bandwidth monitor)

Beschreibung: Ein curses-basiertes Tool zum Anzeigen der aktuellen Bandbreitennutzung pro Verbindung (ähnlich top, aber für Netzwerk Trafficfile-4fdk7rvx39azjretinarqb】). Es zeigt welche Peers mit dem System kommunizieren und wieviel Bytes/s in/out. Hilfreich zum realtime Debugging wer Bandbreite frisst.

Verwendung:

iftop -i <Interface>

Ohne -i wählt iftop das erste externe Interface.

Optionen:

- **-n** – Keine DNS-Auflösung (zeigt IPs statt aufzulösen, wichtig um Ausgabe schnell zu haltenfile-4fdk7rvx39azjretinarqb】).
- **-N** – Keine Port-Auflösung (zeigt Portnummern statt Services Namen).
- **-P** – Zeigt Ports zusammen mit Host (separat Spalte). Kombiniert man oft: iftop -nP.

- **-B** – Bytes statt Bits in Anzeige (Standard iftop zeigt in bits/s).
- **-F <FilterNet>** – Nur Traffic, der in bestimmten Netzbereich fällt (Filter) betrachten, z.B. -F 192.168.0.0/16 filtert auf diese Range.
- **-p** – Promiscuous Mode (auch Traffic anzeigen, der nicht an lokalem Host endet? Normal iftop Filter host involvement).
- **-t** – Text-Only Modus (keine curses-Interface, nur in stdout Stats; nicht sehr üblich).

Im Interface kann man mit Tasten: h Hilfe, n togg DNS, s sort by source, d sort by dest, t sort by throughput, p toggle port display, b toggle bar graph, B cycles through rate intervals (default 2s,10s,40s windows).

traceroute / tracepath (Route Tracking)

Beschreibung: Sendet eine Serie von Paketen mit steigenden TTL (Time-to-Live) Werten, um den Pfad (Router Hopps) zum Ziel auszumessefile-4fdk7rvx39azjretinarqb】 . Traceroute verwendet standard UDP oder ICMP ECHO, tracepath nutzt UDP with high port (no root needed). Jedes Hop sendet "Time Exceeded" zurück, das traceroute protokolliert. Ergebnis: Liste der Router vom Quell- zum Zielnetz.

Syntax:

traceroute [Optionen] <Host> [PacketSize]

tracepath <Host>

Optionen (traceroute):

- **-I**: ICMP Echo statt UDP verwenden (pakete wie ping).
- **-T**: TCP SYN statt UDP (z.B. um Firewalls zu durchdringen über port 80).
- **-p <Port>**: UDP port Start (Standard 33434).
- **-q <N>**: Anzahl Probes pro TTL (Default 3).
- **-w <Seconds>**: Timeout pro Hop Antwort (default 5s).
- **-n**: Nicht auflösen (IP statt DNSfile-4fdk7rvx39azjretinarqb】 .
- **-m <max_ttl>**: Max Hop Count (Standard 30).
- **-f <first_ttl>**: Start TTL (default 1). Manchmal hilfreich, mittendrin einzusteigen.
- **-4 / -6**: Force IPv4 / IPv6.

Optionen (tracepath) sind wenige: -n no dns, -b show both hostnames and IP, etc.

Beispiel:

```
$ traceroute example.com  
1 192.168.1.1 (192.168.1.1) 1.123 ms 0.846 ms 0.799 ms  
2 100.64.0.1 (100.64.0.1) 10.123 ms 9.876 ms 9.543 ms  
3 203.0.113.5 (203.0.113.5) 15.432 ms 14.876 ms 15.210 ms  
4 ae12.dar02.example.net (198.51.100.1) 30.456 ms 29.999 ms 30.123 ms  
5 93.184.216.34 (93.184.216.34) 35.678 ms 35.900 ms 35.456 ms
```

Zeigt 5 Hops (Heimrouter, CGNAT router, ISP core, example.net router, Ziel).

tracepath example.com ähnlich, oft mit asymmetrischen result such as pmtu info.

mtr (My Traceroute)

Beschreibung: Kombiniert traceroute und ping in einem kontinuierlichen Diagnose-Tool. mtr sendet fortlaufend Pakete über alle Hops und aktualisiert in Echtzeit die Latenz und Packetloss-Statistiken pro Hop. Hilfreich um instabile Verbindungen/Packet Loss auf bestimmten Strecken zu identifizieren.

Verwendung:

mtr [Optionen] <Ziel>

Interaktiv curses UI standard, oder mit -r (report) Modus für einmalige Ausgabe.

Optionen:

- **-4 / -6:** Force IPv4/IPv6.
- **-n:** Nicht auflösen, nur IPs (schneller).
- **-w:** Wide-output (zeigt beide host+IP columns).
- **-c <count>:** Limit auf <count> Probezyklen, dann beenden (für Report).
- **-r:** Report-Modus (nicht curses, druckt Tabelle und endet).
- **-i <sec>:** Sekunden zwischen Probe-Paketen (default 1.0).
- **-p:** Pausiert Ausführung beim Start (Press enter to start).
- **-t:** Text Mode (like curses but in dumb terminal).
- **-u:** UDP mode (Standard mtr uses ICMP Echo, as -l in traceroute terms). -T for TCP mode also available.

Anzeige:

MTR listet Hops fortlaufend wie traceroute, plus Spalten: Loss%, Snt (packets sent), Last (ms last latency), Avg, Best, Wrst (worst), StDev. Es aktualisiert kontinuierlich.

Beispiel-Zeile:

3. 203.0.113.5 0.0% 10 15.4 16.1 14.8 20.3 1.3

ping (ICMP Echo)

Beschreibung: Schickt ICMP Echo-Request-Pakete an eine Adresse und misst die Antwortzeiten (Echo-Reply), um Erreichbarkeit und Round-Trip-Time festzustellen】.

Syntax:

ping

(Fortsetzung)

ping (ICMP Echo Requests)

Beschreibung: `ping` prüft die Erreichbarkeit eines Hosts im Netzwerk mit **ICMP-Echo-Paketen** und misst die Round-Trip-Zeit (RTT) der Antwort. Es sendet fortlaufend Anfragen (standardmäßig, bis abgebrochen) und berechnet Statistik über Verlust und Latenz.

Syntax:

``` bash

ping [Optionen] <Ziel>

Hier kann <Ziel> ein Hostname oder eine IP-Adresse sein.

#### **Wichtige Optionen:**

- **-c <Anzahl>** – Anzahl der ICMP-Pakete festlegen. Z.B. ping -c 4 8.8.8.8 sendet genau 4 Pakete】.
- **-i <Sekunden>** – Interval zwischen den Pings (Standard 1 Sekunde). Man kann z.B. -i 0.2 für fünf Pings pro Sekunde (root-Recht nötig bei <0.2>).
- **-W <Timeout>** – Wartezeit in Sekunden auf eine Antwort, bevor als **timeout** gezählt (Default meist 1 Sekunde, bei -c am Ende kann man höher setzen).
- **-q** – "Quiet": keine per-Ping Ausgabe, nur eine Zusammenfassung am Ende.
- **-p <Pattern>** – Sende individuelles Byte-Muster im Paket (Hexadezimal angegeben). Gut für bestimmte Diagnosen (z.B. bestimmte Bits toggeln sehen).
- **-s <Packetsize>** – Größe des zu sendenden Pakets in Bytes (ohne ICMP-Header). Standard 56 Bytes (plus 8 Byte ICMP-Header = 64 Bytes in Ausgabe)】.

- **-t <TTL>** – Setzt die Time-To-Live (IPv4) oder Hop Limit (IPv6) für die gesendeten Pakete. (Nützlich, um gezielt nur bis zu einem gewissen Hop zu gelangen, ähnlich traceroute Step).
- **-I <Interface>** – Bindet an ein bestimmtes Quell-Interface (oder IP) für Multi-homed hosts. Alternativ **-S <Source\_Address>**.
- **-4, -6** – Erzwingt IPv4 oder IPv6.
- **-f** – "Flood ping": schickt so schnell wie möglich Pings (sehr viele pro Sekunde) und druckt einen Punkt pro empfangenes Echo. (Nur root, zur Stresstest oder schnelle Packetloss Erkennung).
- **-L** (bei einigen ping-Versionen) – Für Echo an Broadcast/Multicast-Adresse (ermöglicht Broadcast-ping, was standardmäßig blockiert ist).
- **-a** – Bei Empfang einen Terminal-Bell ausgeben (akustisches Signal).
- **-D** – Zeitstempel vor jeder Zeile (UNIX time).
- **-U** – Zeitstempel in Us seit Boot vor jeder Zeile (für Messungen).

### **Ausgabe-Interpretation:**

Jede empfangene Antwort erscheint als Zeile:

64 bytes from 8.8.8.8: icmp\_seq=1 ttl=117 time=8.34 ms

Bedeutet: Echo-Reply von 8.8.8.8 erhalten, Größe 64 Bytes, Sequenznummer 1, TTL im Reply 117, RTT 8.34 ms. Wenn Antworten ausbleiben, erscheinen Zeilen wie "Request timeout".

Am Ende kommt eine **Statistik**:

4 packets transmitted, 4 received, 0% packet loss, time 3005ms

rtt min/avg/max/mdev = 8.334/8.512/8.790/0.177 ms

Zeigt Verlust und RTT-Werte.

### **Beispiele:**

- ping 1.1.1.1 – Kontinuierlicher Test zu 1.1.1.1 (Cloudflare DNS). Abbruch mit Strg+C.
- ping -c 5 www.example.com – 5 Pings an example.com, dann stop mit Ausgabe der Statistik.
- ping -I eth1 192.168.2.1 – Ping von Interface eth1 aus (wenn Multi-Interface).
- ping -s 1000 -c 3 host – Sende 3 Pings mit Paketgröße ~1000 Bytes (ICMP Payload).

- ping -6 -c 4 ipv6.google.com – IPv6 Ping zu Google.

### **traceroute (siehe oben)**

(bereits behandelt in vorangehendem Teil, siehe „traceroute“ in Netzwerkdiagnose.)

### **tracepath (siehe oben)**

(siehe „tracepath“ bei Netzwerkdiagnose.)

### **resolvectl (Systemd-resolved Query)**

**Beschreibung:** CLI-Tool zum Abfragen des systemd-resolved-Dienstes (Namensauflösung). resolvectl kann DNS-Lookups durchführen, DNS-Settings anzeigen, LL/MNR Browsing etc. Ersetzt Tools wie systemd-resolve (veraltet).

#### **Verwendung:**

- **resolvectl status** – Zeigt die aktuellen DNS-Einstellungen pro Link (Interface): DNS-Server, Suche-Domains, LLMNR/MulticastDNS Status, DNSSEC, usw.
- **resolvectl query <Hostname>** – Löst einen Hostnamen auf (A/AAAA-Record). Ausgabe enthält v6 und v4 Adressen, CNAME falls, DNSSEC info, Cache info, antwortender Server etc.  
Beispiel: resolvectl query example.com.
- **resolvectl query -t MX <Domain>** – DNS-Abfrage bestimmten Typs (hier MX-Records). Unterstützt -t ANY, A, AAAA, MX, SRV, SOA etc.
- **resolvectl query <IP>** – Reverse Lookup (IP zu Name via PTR).
- **resolvectl flush-caches** – Löscht den DNS-Cache von systemd-resolved.
- **resolvectl dns <Interface> [Server...]** – Setzt DNS-Server für ein bestimmtes Interface (über DBus). resolvectl domain <iface> ... analog. (Normal wird das via NetworkManager/Netplan/etc gemacht).
- **resolvectl revert <Interface>** – Löscht manuelle Overrides und kehrt zu vom Netzwerkdienst gelieferten DNS zurück.

#### **Beispiel:**

```
$ resolvectl query -t MX example.com
example.com IN MX 0 .
-- Information acquired via protocol DNS in 15.1ms.
-- Data is authenticated: no
Bedeutet example.com hat keinen MX (Mail).
```

resolvectl status Ausschnitt:

Global

Protocols: LLMNR=resolve mDNS=no DNSOverTLS=no DNSSEC=no/unsupported  
resolv.conf mode: stub

Link 2 (eth0)

Current Scopes: DNS

Protocols: +DefaultRoute +LLMNR -mDNS DNSOverTLS=opportunistic DNSSEC=...

Current DNS Server: 192.168.1.1

DNS Servers: 192.168.1.1 1.1.1.1

DNS Domain: ~.

Zeigt z.B. that eth0 hat default DNS 192.168.1.1 plus Cloudflare, und ~. (default domain).

### **dig (DNS Lookup Utility)**

**Beschreibung:** dig (domain information groper) führt DNS-Abfragen aus. Es ist Teil von BIND-Tools (bind-utils). Im Gegensatz zu host oder nslookup bietet dig ausführlich formatierbare Antworten. Standard macht eine A-Record-Abfrage und gibt Antwort mit Abschnitt **ANSWER SECTION** aus.

#### **Syntax:**

dig [@DNS-Server] <Name> [<Typ>] [<Klasse>] [Optionen]

<Klasse> normalerweise IN (Internet), selten was anderes.

#### **Wichtige Optionen:**

- **@server** – Spezifischen DNS-Server benutzen statt System-resolver. Z.B. dig @8.8.8.8 example.com.
- **<Typ>** – Abfragetyp: A, AAAA, MX, TXT, SRV, NS, SOA, CNAME, PTR, ANY etc. (Default A).
- **+short** – Ausgabe nur die Antworten knapp (einfacher, scripterfreundlic0】 . Z.B. dig +short example.com A -> "93.184.216.34".
- **+noall +answer** – Unterdrückt alle außer Answer-Section (ähnlich short aber mit TTL und Info).
- **+trace** – Iterativer Trace durch DNS-Hierarchie (Root bis Ziel, ähnlich dig an Root und dann NS referrals folgen). Zeigt jeden Schritt. Hilfreich zum Debug.
- **+tcp** – Erzwinge TCP statt UDP (z.B. bei großen DNS-Records oder EDNS issues).

- **+nssearch** – Findet alle NS des Ziel und fragt diese nach SOA (um Delegation correctness zu checken).
- **-x <IP>** – Reverse lookup (PTR) für IP. (Bequem: dig -x 8.8.8.8).
- **+multi** – Schöne Ausgabe in Spalte falls mehrere TXT oder so.
- **+nodnssec oder +dnssec** – Anfordern oder nicht (um RRSIG etc. zu sehen).
- **+ttlshort** – Output TTL in short mode. (e.g. dig +short +ttl google.com -> "300 172.217.168.14").
- **+stats / +nostats** – Schaltet Statistik-Banner am Ende an/aus.
- **+cm** – Colorize output (if compiled with color support).

#### **Beispiele:**

- dig example.com ANY – Fragt alle Records (die server je nach config liefern).
- dig google.com MX +noall +answer – Zeigt nur MX Antworten.
- dig @1.1.1.1 cloudflare.com A – Abfrage an 1.1.1.1 nach cloudflare's A.
- dig -x 8.8.4.4 +short – Reverse Lookup, kurze Ausgabe ("dns.google").
- dig +trace example.org – Startet vom Root Nameserver und verfolgt Delegation bis example.org resolved ist.

#### **nslookup (Name Server Lookup - veraltet)**

**Beschreibung:** Ein einfaches interaktives DNS-Abfrageprogramm. War Teil von BIND, gilt als veraltet im Vergleich zu dig. Viele Admins kennen es dennoch.

#### **Verwendung:**

nslookup [<Host> [<DNS-Server>]]

Ohne args geht es in interaktiven Modus. Dessen Kommandos: server <DNS>, set q=MX, set type=AAAA, set debug, exit.

#### **Beispiele:**

\$ nslookup example.com 8.8.8.8

Server: 8.8.8.8

Address: 8.8.8.8#53

Non-authoritative answer:

Name: example.com

Address: 93.184.216.34

Reverse: nslookup 93.184.216.34 -> zeigt PTR.

**Warnung:** Neuere doc empfehlen dig oder host statt nslookup.

### **whois (Domain Whois Query)**

**Beschreibung:** Fragt Whois-Datenbanken nach Registrierungsinformationen einer Domain oder 5】. Das Tool verbindet sich zum passenden Whois-Server (verwendet /etc/whois.conf, TLD heuristics) via TCP 43 und gibt den Text aus.

#### **Syntax:**

whois [Optionen] <DomainOrIP>

**Ausgabe:** Je nach TLD und Registry differierend. Enthält Domain-Inhaber (bei Generic oft DSGVO anonymisiert), Registrar, Erstellungs- und Ablaufdatum, Nameserver, Contact-IDs. Bei IP WHOIS (z.B. whois 8.8.8.8) gibt es Informationen über den zugeteilten IP-Block (ISP, RIR wie ARIN/RIPE...).

#### **Optionen:**

- **-H** – Unterdrückt das Banner "Whois Server" in RIPE responses.
- **-p <port>** – Andere Port.
- **--verbose** – Mehr Info über Server selection.
- **--help** – Display help.

#### **Beispiele:**

- whois example.com -> zeigt Registrar (ICANN info), Status, Name der Organisation, etc.
- whois 2620:0:2d0:200::7 -> IPv6 whois (gibt i.d.R. ARIN oder so mit OrgName).
- whois AS15169 -> WHOIS an RADb für Autonomous System 15169 (Google).

(Beachte: Manche TLDs haben separate Tools, z.B. jwhois config. Das Standard whois ist meist in Debian whois package (Marco d'Itri) which covers a lot. GDPR hat Personendaten stark eingeschränkt in Output.)

### **arp (Address Resolution Protocol table) – Hinweis: arp wird mittlerweile durch ip neigh ersetzt.**

**Beschreibung:** Zeigt den ARP-Cache (IPv4) an oder manipuliert Einträge. (Linux modern: use ip neigh aber arp Kommando aus net-tools ist noch gebräuchlich).

#### **Befehle:**

- arp -a – Listet ARP-Einträge (Hostname, IP, MAC, Interface).

- arp -n – Wie -a, aber keine DNS für IP.
- arp -d <IP> – Löscht ARP-Eintrag für IP.
- arp -s <IP> <MAC> – Fügt statischen ARP-Eintrag hinzu (MAC fix für IP) – gleich ip neigh add ... nud permanent.

*Beispiel:* arp -a -> router.lan (192.168.1.1) at aa:bb:cc:dd:ee:ff [ether] on eth0.

## Sicherheit und Firewall

### iptables/iptables-nft (Packet Filter)

**Beschreibung:** Linux Kernel-Paketfilter (Firewall) via Netfilter-Schnittstelle. iptables verwaltet Regeln in verschiedenen Tabellen: **filter**, **nat**, **mangle**, **raw**, **security**, für IPv4. (Für IPv6, ip6tables). Mit Kernel 4.x/5.x viele Distros nutzen "iptables-nft" as a frontend to nftables, aber Syntax bleibt für user gleich.

#### Syntax Grundkommandos:

iptables [-t <table>] <operation> <chain> <matches...> <target>

Operationen: **-A** (append), **-I** (insert), **-D** (delete), **-C** (check exist), **-L** (list), **-F** (flush chain), **-N** (new chain), **-X** (delete user chain).

Standard Tabellen:

- **filter** (Default): Chains **INPUT**, **OUTPUT**, **FORWARD**.
- **nat**: Chains **PREROUTING**, **OUTPUT**, **POSTROUTING** (masquerading, port forwarding etc.).
- **mangle**: For packet alteration (TOS, mark), Chains in multiple points (PREROUTING, OUTPUT, etc.).
- **raw**: For conntrack exemption (NOTRACK), Chains PREROUTING, OUTPUT.
- **security**: SELinux related (rare).

#### Matches / Options:

- **-p <proto>**: Protocol (tcp, udp, icmp, icmpv6, all).
- **-s <IP/CIDR>, -d <IP/CIDR>**: Source or destination address (or network).
- **-i <iface>, -o <iface>**: Inbound or outbound network interface (for PREROUTING/INPUT/forward vs OUTPUT/POSTROUTING).
- **--sport, --dport** (with -p tcp/udp): Source/dest port or port range (e.g. --dport 80 or --dport 1000:2000).

- **-m state --state <states>**: State match (connection tracking states: INVALID, ESTABLISHED, NEW, RELATED). Common: allow ESTABLISHED,RELATED on INPUT for responses.
- **-m conntrack --ctstate**: (newer usage, same as state module now).
- **-m connlimit, -m limit** etc: Many modules. limit --limit 5/minute for rate-limit logging for example. connlimit --connlimit-above N limit parallel connections.
- **-m multiport**: match multiple ports in one rule if not contiguous, e.g. -m multiport --dports 80,443,8080.
- **-m tcp --tcp-flags SYN,RST,ACK SYN**: match specific TCP flags combination (here SYN only). Often used for --syn (alias) which matches syn packets (to identify new connections).
- **\*\*-m comment --comment "text"**: attach comment to rule (for documentation).

#### Targets (Actions):

- **ACCEPT**: Packet erlaubt.
- **DROP**: Verwerfen ohne Antwort.
- **REJECT**: Verwerfen mit Antwort (ICMP port unreachable o.Ä.).
- **LOG**: Loggt Paket (via kernel log), dann Verarbeitung geht weiter (so typ. Kombi LOG dann DROP). Options: --log-prefix "msg", --log-level warning.
- **MASQUERADE** (in nat POSTROUTING): SNAT für dynamische IP (Router Use-case) – ersetzt Source-IP durch Interface-IP.
- **SNAT** (in nat POSTROUTING): Statisches Source-NAT (requires --to-source IP[:port-range]).
- **DNAT** (in nat PREROUTING/OUTPUT): Destination-NAT (Port-Forwarding; --to-destination IP:port).
- **REDIRECT** (nat PREROUTING/OUTPUT): Umleiten an lokalen Host (for transparent proxy etc.).
- **MARK** (mangle): Markiert Paket mit fwmark (für routing decisions via ip rule, etc.).
- **RETURN**: In user-defined chain springt zurück zum Aufrufer-Chain (stop further rules in current chain). In builtin chain, RETURN = apply chain policy.
- **QUEUE**: Übergibt Paket an Userspace (via libnetfilter\_queue).
- **DROP vs. policy**: If chain policy is DROP, falling off end = drop.

#### Beispiele:

- **List Regeln:** iptables -L -n -v (filter table). iptables -t nat -L -n -v NAT rules. -n no DNS, -v verbose (packet counters).
- **Allow incoming SSH:**

② iptables -A INPUT -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT  
 iptables -A OUTPUT -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT  
 (typical stateful allow SSH in).

② **Drop all forward by default:** iptables -P FORWARD DROP.

② **Masquerading (NAT):**

② iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE  
 (for home router with ppp0 outbound).

② **Port Forward:**

Forward ext eth0 port 80 to internal 192.168.1.100 port 8080:

② iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 \  
 -j DNAT --to-destination 192.168.1.100:8080

iptables -A FORWARD -p tcp -d 192.168.1.100 --dport 8080 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

② **Logging Drop:**

- iptables -A INPUT -p tcp --dport 23 -j LOG --log-prefix "Telnet attempt: " --log-level info
- iptables -A INPUT -p tcp --dport 23 -j DROP
- (log then drop telnet).

## **firewall-cmd (firewalld frontend)**

**Beschreibung:** Steuerprogramm für **firewalld**, eine dynamische Firewall (standard in RHEL/CentOS 7+, Fedora, etc.). firewall-cmd ermöglicht das Konfigurieren von Zonen, Regeln (services, ports, masq) zur Laufzeit und in Config. Firewalld nutzt intern iptables/nftables.

### **Syntax:**

firewall-cmd [--zone=<zone>] [--permanent] <aktion> <Parameter>

**Grundkonzepte:** Zonen (trusted, public, home, internal, work, external, dmz, block, drop etc.), Schnittstellen sind Zonen zugewiesen. Dienste (vordefinierte Regelsets) können pro Zone erlaubt werden oder einzelne Ports/Protokolle. Man kann Einstellungen permanent (persistiert in XML config) oder nur runtime vornehmen.

## Wichtige Befehle:

- **--state**: Zeigt an, ob firewalld läuft (running/not running).
- **--get-active-zones**: Zeigt welche Zonen aktiv sind und welche Interfaces zugeordnet (z.B. "public: wlp2s0").
- **--get-zones**: Listet alle bekannten Zonen. **--list-all** [--zone=Z] listet alle Einstellungen einer Zone (services, ports, masquerade etc0) .
- **--zone=<zone> --add-service=<service>**: Erlaubt den vordefinierten service in angegebener Zone (runtime). Bsp: firewall-cmd --zone=public --add-service=http – ermöglicht HTTP (Port 80) in Zone public.
- **--zone=<zone> --add-port=<port>/<proto>**: Erlaubt individuellen Port. Bsp: --add-port=5000/tcp.
- **--remove-service=..., --remove-port=...** analog zum Entfernen.
- **--add-masquerade**: Aktiviert NAT/Masquerading in der Zone (z.B. für Internet-sharing zone).
- **--query-service=<svc>**: Fragt ob Service in zone erlaubt (exit code 0/1).
- **--permanent**: Gibt man diese Option an, wird die Änderung in der persistenten Konfiguration vorgenommen (aber nicht an laufende sofort angewendet, es sei denn man ruft extra an). In firewalld, runtime und permanent config sind getrennt.
- **--reload**: Lädt die permanente Konfiguration neu (in runtime) – im Grunde wendet Config an.
- **--runtime-to-permanent**: Speichert aktuellen Laufzeit-Zustand als neue permanente Konfiguration (was man on-the-fly geändert hat, persistieren).
- **--zone=<zone> --change-interface=<iface>**: Weist ein Interface einer Zone zu (persistente Zuordnung).
- **--new-zone=<name>, --delete-zone=...**: Eigene Zonen erstellen/löschen.
- **--list-services, --get-services**: Listet bekannte Service-Namen (die in /etc/firewalld/services/\*.xml definiert sind).
- **--service=<svc> --get-ports**: Zeigt welche Ports hinter einem Service stecken.
- **--add-rich-rule '<rule>'**: Fügt sog. Rich Rule hinzu (eine erweiterte Regel-Syntax in firewalld, z.B. mit Logging, Zeitplänen). Z.B.:
- `firewall-cmd --add-rich-rule='rule family="ipv4" source address="192.0.2.0/24" port port="22" protocol="tcp" accept'`

- (Erlaubt SSH nur von 192.0.2.0/24).

### **Beispiele:**

- Öffnen von HTTPS permanent:
- `firewall-cmd --add-service=https --permanent`
- `firewall-cmd --reload`
- 
- Port 12345/udp temporär öffnen in Zone public:  
`firewall-cmd --zone=public --add-port=12345/udp` (nach Reboot weg).
- Interface eth0 zur Zone external zuweisen:  
`firewall-cmd --zone=external --change-interface=eth0 --permanent && firewall-cmd --reload.`
- Status: `firewall-cmd --list-all` (wenn default zone, z.B. public, dann deren config).

## **ufw (Uncomplicated Firewall)**

**Beschreibung:** Vereinfachtes Firewall-Frontend (Ubuntu/Debian) für iptables. ufw erlaubt einfache allow/deny-Regeln definieren, typischerweise nach Services oder Ports, optional in/out interface.

### **Syntax:**

`ufw [Optionen] <Aktion>`

### **Grundbefehle:**

- **ufw enable / disable** – Aktiviert oder deaktiviert die Firewall (bei disable: setzt iptables default ACCEPT und löscht Regeln).
- **ufw status [verbose]** – Zeigt aktuelle Regel-Liste. Mit verbose sieht man auch Interfaces, logging status etc.
- **ufw default allow|deny [incoming|outgoing|routed]** – Setzt Standardpolicy. Standard-Einstellung: incoming deny, outgoing allow.
- **ufw allow <Port/Service> [proto tcp/udp] [from <IP> [to <IP>]] [port <P2>]** – Erlaubt eingehenden Traffic. Beispiele:
  - `ufw allow 22` (erlaubt TCP/22 auf allen Interfaces).
  - `ufw allow proto udp from 10.0.0.0/8 to any port 53` (erlaubt UDP DNS von privatem Netz).
  - `ufw allow out 80/tcp` (erlaubt ausgehende HTTP).
  - `ufw allow in on eth0 to any port 443` (erlaubt eingehendes HTTPS auf eth0).

- **ufw deny <Spec>** – Blockiert (mit REJECT) entsprechend.
- **ufw reject <Spec>** – Synonym zu deny (deny und reject meist ähnlich in ufw; auf Debian might differ).
- **ufw limit <Port/Service>** – Wie allow, aber mit Rate-Limit (gegen Bruteforce). Z.B. ufw limit ssh erlaubt SSH, aber blockiert IP temporär bei >6 Verbindungen/30s.
- **ufw delete <RuleSpec>** – Entfernt Regel (man kann Regel genau wie bei allow angeben, mit 'delete' davor, oder Nummer aus ufw status numbered).
- **ufw logging on|off|LEVEL** – Logging ein/aus und Level (off, low, medium, high, full).
- **ufw reset** – Setzt ufw auf Auslieferungszustand (alle Regeln weg, policies reset, und disabled).
- **ufw route allow/deny ...** – Regeln für Forwarded Traffic (Routen). Bsp: ufw route allow in on eth1 out on eth0 to any port 80 for forwarded.
- **ufw insert <num> <rule>** – Fügt Regel an bestimmter Position ein.
- **ufw show raw** – Zeigt resultierende iptables-Regeln.

### **Beispiele:**

- Standard: ufw default deny incoming; ufw default allow outgoing; ufw enable.
- Webserver: ufw allow 'Nginx Full' (in /etc/ufw/applications.d vordefinierte Profile, 'Apache Full' etc., Nginx Full = 80+443).
- Only LAN access to a service: ufw allow from 192.168.0.0/16 to any port 3306 (MySQL offen nur intern).
- Block eine IP: ufw deny from 203.0.113.5.
- Show rules: ufw status verbose -> e.g.
- Status: active
- Logging: on (low)
- Default: deny (incoming), allow (outgoing), deny (routed)
- New profiles: skip
- 
- To              Action    From
- --              -----    -----

- 22/tcp            LIMIT    Anywhere        # SSH
- 80,443/tcp       ALLOW    Anywhere        # WWW
- 22/tcp (v6)      LIMIT    Anywhere (v6)
- 80,443/tcp (v6)   ALLOW    Anywhere (v6)
- 

*(UFW ist beliebt wegen einfacher Syntax, aber fehlender Komplexität bei komplexen Setups. Firewalld und raw iptables/nft sind mächtiger in Feinheiten.)*

### nft (nftables Firewall)

**Beschreibung:** nftables ist der modernere Packet-Filter (ersetzt iptables). Der Befehl nft wird zum Konfigurieren verwendet. Sehr flexibel (Datenstrukturen, one-rule multiple matches).

#### Syntax:

nft [options] <command>

Man kann Kommandos interactive eingeben oder als Skript. Der Kommandostil ähnelt C:  
e.g.

nft add table inet filter

nft add chain inet filter input { type filter hook input priority 0 \; }

nft add rule inet filter input tcp dport 22 accept

(Librally, but given complexity and that the original question references iptables and ufw, likely beyond scope).

### Abschluss

Dieses Kompendium deckt die wesentlichen Befehle für die Linux+ (XK0-005) Prüfung ab, gegliedert nach Themengebieten. Die Befehle umfassen Benutzer- und Rechteverwaltung, Prozess- und System-Tools, Netzwerk-Konfiguration und -Diagnose, Software-Installation, Container, Sicherheit (Firewall, SELinux/AppArmor siehe oben Rich-Rules, etc.), Dienste und Planung.

Durch die strukturierte Darstellung mit Syntax, Beschreibung, Optionen und Beispielen dient es als umfassendes Nachschlagewerk. **Wichtig:** In einer Prüfungssituation sollte man sich auf die korrekte Anwendung der Optionen und das Verständnis der Konzepte dahinter konzentrieren (z.B. was macht chmod 775 oder umask 002 bewirkt).

Mit dieser Referenz sollte die Vorbereitung auf typische Linux-Aufgaben – vom Anlegen eines Benutzers, dem Ändern von Dateirechten, über das Einrichten eines Cronjobs bis

zum Konfigurieren eines Apache-Dienstes oder Setzen einer Firewallregel – bestens unterstützt werden.

## Kompendium der Linux+ (XK0-005) Befehle

Dieses Nachschlagewerk umfasst alle wichtigen Befehle aus dem offiziellen **Command Reference Guide** für die CompTIA Linux+ Prüfung (XK0-005). Die Befehle sind thematisch gegliedert (Benutzerverwaltung, Netzwerk, Prozesse, Dateisystem, Paketmanagement, Container, Sicherheit etc.), mit Beschreibung, vollständiger Syntax, allen verfügbaren Optionen und ggf. wichtigen Unterbefehlen. Es dient als umfassendes Cheat-Sheet zur Prüfungsvorbereitung.

### Grundlegende Linux-Befehle

#### ls (List Directory Contents)

**Beschreibung:** Listet den Inhalt eines Verzeichnisses auf. Ohne Parameter zeigt ls die nicht versteckten Einträge im aktuellen Verzeichnis in alphabetischer Reihenfolge[wiki.ubuntuusers.deionos.de](http://wiki.ubuntuusers.deionos.de).

**Syntax:**

ls [OPTION]... [VERZEICHNIS]

*Wenn kein Verzeichnis angegeben wird, wird das aktuelle Verzeichnis aufgelistet[wiki.ubuntuusers.de](http://wiki.ubuntuusers.de).*

**Optionen:**

| Option               | Beschreibung                                                                                                                                                                  |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a, --all            | Zeigt alle Einträge an, inklusive versteckter Dateien (auch . und ..) <a href="http://wiki.ubuntuusers.deionos.de">wiki.ubuntuusers.deionos.de</a> .                          |
| -A, --almost-all     | Wie -a, aber ohne . und .. – listet versteckte Dateien außer den Verzeichnissen ./.. <a href="http://wiki.ubuntuusers.deionos.de">wiki.ubuntuusers.deionos.de</a> .           |
| -l                   | Langformat: Detaillierte Ausgabe (Rechte, Besitzer, Größe, Datum usw.) je Eintrag <a href="http://wiki.ubuntuusers.deionos.de">wiki.ubuntuusers.deionos.de</a> .              |
| -d, --directory      | Listet Verzeichniseintrag selbst statt dessen Inhalt (nützlich, um Infos über Ordner zu sehen) <a href="http://wiki.ubuntuusers.deionos.de">wiki.ubuntuusers.deionos.de</a> . |
| -h, --human-readable | Im Langformat (-l) Größen in menschenlesbare Einheiten (KB, MB ... ) anzeigen <a href="http://wiki.ubuntuusers.deionos.de">wiki.ubuntuusers.deionos.de</a> .                  |
| -i                   | Zeigt die Inode-Nummer jedes Eintrags an <a href="http://wiki.ubuntuusers.deionos.de">wiki.ubuntuusers.deionos.de</a> .                                                       |

| <b>Option</b>          | <b>Beschreibung</b>                                                                                                                                              |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-C</b>              | Ausgabe in Spalten nebeneinander<br>(Standardformat) <a href="http://wiki.ubuntuusers.de/ionos.de">wiki.ubuntuusers.deionos.de</a> .                             |
| <b>-m</b>              | Listet Einträge kommassepariert in einer Zeile <a href="http://wiki.ubuntuusers.deionos.de">wiki.ubuntuusers.deionos.de</a> .                                    |
| <b>-R, --recursive</b> | Rekursiv: Zeigt auch Inhalte von Unterverzeichnissen (durchläuft Verzeichnisbaum) <a href="http://wiki.ubuntuusers.deionos.de">wiki.ubuntuusers.deionos.de</a> . |
| <b>-t</b>              | Sortiert nach Zeitstempel (Änderungszeit, neueste zuerst) anstatt alphabetisch.                                                                                  |
| <b>-r</b>              | Kehrt die Sortierreihenfolge um (Reverse order).                                                                                                                 |
| <b>--hyperlink</b>     | Macht Dateinamen (in unterstützten Terminals) als klickbare Links ausgebbar <a href="http://wiki.ubuntuusers.de">wiki.ubuntuusers.de</a> .                       |
| <b>--help</b>          | Zeigt die Hilfe zum Befehl an.                                                                                                                                   |
| <b>--version</b>       | Zeigt Versionsinformationen von ls.                                                                                                                              |

(Weitere Optionen sind in den Manpages dokumentiert[wiki.ubuntuusers.de](http://wiki.ubuntuusers.de).)

### **cat (Concatenate and Print Files)**

**Beschreibung:** Gibt den Inhalt von Dateien auf der Standardausgabe aus oder verknüpft (konkatiniert) mehrere Dateien. Oft verwendet, um Textdateien anzuzeigen oder Inhalte zu pipen[ionos.at](http://ionos.at). Der Name steht für „concatenate“ (verketten).

**Syntax:**

`cat [OPTION]... [DATEI]...`

*Ohne Optionen schreibt cat den Inhalt der angegebenen Datei(en) in der angegebenen Reihenfolge auf die Standardausgabe. Kann mehrere Dateien hintereinander ausgeben oder mit Umleitungen/ Pipes genutzt werden.*

**Optionen:**

| <b>Option</b> | <b>Beschreibung</b> |
|---------------|---------------------|
|---------------|---------------------|

|           |                                                                                                                                             |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-n</b> | Nummeriert alle ausgegebenen Zeilen fortlaufend <a href="http://ibm.comionos.at">ibm.comionos.at</a> .                                      |
| <b>-b</b> | Nummeriert nur <b>nicht-leere</b> Ausgabelinien (Leerzeilen werden nicht nummeriert) <a href="http://ibm.comionos.at">ibm.comionos.at</a> . |

## **Option Beschreibung**

- s** Squeeze: Unterdrückt aufeinanderfolgende Leerzeilen (mehrere leere Zeilen werden als eine ausgegeben)[ionos.at](#).
- v** "Visible" – Macht nicht druckbare Steuerzeichen sichtbar (außer Tab und Zeilenende)[ionos.at](#).
- e** Entspricht -v mit Anzeige des Zeilenende-Zeichens \$ am Zeilenende (kombiniert -v und Kennzeichnung von Zeilenumbrüchen)[ionos.at](#).
- t** Entspricht -v mit Darstellung von Tabulatoren als ^I (Tab-Markierung)[ionos.at](#).
- E** Zeigt \$ am Ende jeder Zeile (Äquivalent zu -vE, oft identisch zu -e je nach Implementierung).
- T** Zeigt Tabulatoren als ^I an (ähnlich -vT, meist wie -t).
- A** Kombiniert mehrere der obigen: entspricht oft -vET (alle nicht-druckbaren Zeichen, Tabs und Zeilenenden sichtbar machen).
- u** (Ignoriert in GNU cat – historisch für ungepufferte Ausgabe).
- help** Zeigt Hilfe-Informationen an.
- version** Zeigt Versionsinfo an.

## **cd (Change Directory)**

**Beschreibung:** Wechselt in ein anderes Verzeichnis. Mit cd kann das aktuelle Arbeitsverzeichnis geändert werden („change directory“)[ionos.at](#). Ohne Argument wechselt cd standardmäßig ins Heimatverzeichnis des Benutzers.

### **Syntax:**

`cd [OPTION] [VERZEICHNIS]`

*Man kann relative Pfade (bezogen auf das aktuelle Verzeichnis) oder absolute Pfade (ausgehend vom Wurzelverzeichnis /) angeben. Erfolgt der Wechsel in ein Zielverzeichnis, so wird dieses das neue aktuelle Arbeitsverzeichnis; bei nicht vorhandenen Zielen oder fehlenden Rechten bleibt das Verzeichnis unverändert[ionos.at](#).*

### **Optionen:**

## **Option Beschreibung**

- L** Folge symbolischen Links **logisch**. Dies ist die Voreinstellung. (Beim Navigieren durch einen Symlink bleibt man in der logischen Pfadstruktur)[ionos.at](#).

## Option Beschreibung

- P Folge symbolischen Links **physisch**. Wechselt in das tatsächliche Ziel im Dateisystem und löst symbolische Links auf[ionos.at](#).

## Unterbefehle / spezielle Aufrufe:

- cd ~ wechselt ins Heimatverzeichnis des aktuellen Benutzers[ionos.at](#).
- cd - wechselt in das vorherige Arbeitsverzeichnis zurück (Toggle zwischen zwei Verzeichnissen)[ionos.at](#).
- cd /Pfad/zu/Verz wechselt absolut in ein angegebenes Verzeichnis vom Wurzelpfad aus (hier z.B. nach /Pfad/zu/Verz).
- cd .. wechselt eine Ebene höher (ins Parent-Verzeichnis), cd ../../ entspricht zwei Ebenen usw.
- cd . bleibt im aktuellen Verzeichnis (. bezeichnet das aktuelle Verzeichnis, meistens kein Effekt).

## pwd (Print Working Directory)

**Beschreibung:** Gibt den vollständigen Pfad des aktuellen Arbeitsverzeichnisses aus[ionos.at](#). Mit pwd kann man sich anzeigen lassen, *in welchem Verzeichnis man sich gerade befindet*.

### Syntax:

pwd [OPTION]...

### Optionen:

#### Option Beschreibung

- Ausgabe des Pfads mit Berücksichtigung symbolischer Links (logischer Pfad, Standard). Wenn z.B. ein symbolischer Link als aktuelles Verzeichnis dient, wird der Linkpfad angezeigt[ionos.de](#).
- P Ausgabe des *physischen* Pfads: Symbolische Links werden aufgelöst, der tatsächliche Verzeichnispfad wird angezeigt[ionos.de](#).
- help Zeigt Hilfe-Informationen zu pwd an.
- version Zeigt Versionsinformationen an.

*Hinweis:* pwd ist häufig als Shell-Builtin implementiert. Das externe Programm /bin/pwd unterstützt ggf. andere oder weniger Optionen (bei GNU Coreutils nur --help und --version zusätzlich)[ionos.de](#). In der Regel genügt der Shell-eigene pwd Befehl.

## **whoami**

**Beschreibung:** Zeigt den Benutzernamen des aktuell angemeldeten Users anfile-4fdk7rvx39azjretinarqb. Praktisch, um schnell zu prüfen, unter welchem Account ein Prozess läuft.

### **Syntax:**

whoami

*Keine spezifischen Optionen.* Der Befehl gibt äquivalent echo \$USER aus (den Namen des aktuellen Unix-Accounts).

## **touch**

**Beschreibung:** Erzeugt eine neue leere Datei oder ändert Zeitstempel (Zugriffs- und Änderungszeit) bestehender Dateien. Wenn die Datei nicht existiert, wird sie (sofern nicht -c verwendet wird) als leere Datei angelegt[man7.org](#). Oft genutzt, um Dateien schnell anzulegen oder deren *timestamp* zu aktualisieren.

### **Syntax:**

touch [OPTION]... DATEI...

### **Optionen:**

| <b>Option</b>                | <b>Beschreibung</b>                                                                                                                                                                                                                 |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-a</b>                    | Ändert nur die Zugriffszeit (atime) der Datei <a href="#">man7.org</a> .                                                                                                                                                            |
| <b>-m</b>                    | Ändert nur die Änderungszeit (mtime) der Datei <a href="#">man7.org</a> .                                                                                                                                                           |
| <b>-c, --no-create</b>       | Legt keine neue Datei an. Wenn die Datei nicht existiert, wird kein neues leeres File erstellt (ansonsten Standardverhalten ohne -c) <a href="#">man7.org</a> <a href="#">man7.org</a> .                                            |
| <b>-d, --date=STRING</b>     | Verwendet den angegebenen Datums-String anstelle der aktuellen Zeit. Der String wird geparsst (z.B. "2025-05-01 12:00:00" oder relative Angaben wie "next Friday" sind möglich) <a href="#">man7.org</a> <a href="#">man7.org</a> . |
| <b>-t</b>                    | Verwendet einen konkreten Zeitstempel im angegebenen Format (JJMMTThhmm[.ss]) anstatt der aktuellen [[CC]YY]MMDDhhmm[.ss] Zeit <a href="#">man7.org</a> . Beispiel: -t 202312312359.59 setzt 31.12.2023 23:59:59.                   |
| <b>-r, --reference=DATEI</b> | Übernimmt die Zeitstempel (Zugriffs- und Änderungszeit) der angegebenen Referenzdatei. Die Zielfile erhält also dieselben Zeiten wie DATEI <a href="#">man7.org</a> .                                                               |

| Option                      | Beschreibung                                                                                                                                                                                                                                                                                                                                    |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-h, --no-dereference</b> | Ändert bei symbolischen Links den <i>Link selbst</i> (deren Metadaten), nicht die referenzierte Datei. Ohne diese Option würde touch auf den Ziel-Dateipfad wirken (nicht auf den Symlink). ( <i>Diese Option ist nur wirksam, wenn das System das Verändern von Symlink-Zeitstempeln unterstützt.</i> ) <a href="http://man7.org">man7.org</a> |
| <b>--time=WORD</b>          | Bestimmt, welcher Zeitstempel geändert wird. Mögliche Werte: atime, access, use (entspricht -a), oder mtime, modify (entspricht -m) <a href="http://man7.org">man7.org</a> .                                                                                                                                                                    |
| <b>--help, --version</b>    | Anzeigen der Hilfe bzw. Versionsinformation zu touch.                                                                                                                                                                                                                                                                                           |

## man (Manual Page Viewer)

**Beschreibung:** Zeigt die Handbuchseite (Manual Page) für einen angegebenen Befehl oder ein Thema an. man ist das Offline-Hilfe- und Dokumentationssystem unter Unix/Linux.

### Syntax:

man [SECTION] <Thema/Command>

*Beispiel:* man ls zeigt die Manpage zu **ls**. Manpages sind meist nach Kategorien/Abschnitten (1=Benutzerkommandos, 5=Dateiformate, 8=Administrator-Kommandos etc.) gegliedert – optional kann man mit man 5 passwd z.B. direkt die Manpage zum Dateiformat **passwd** (Abschnitt 5) öffnen.

### Optionen:

| Option             | Beschreibung                                                                                                                                                                                     |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-k Suchwort</b> | Durchsucht die Manpage-Kurzzusammenfassungen nach <i>Suchwort</i> (entspricht dem Befehl apropos). Beispiel: man -k network listet alle Manual-Topics auf, deren Beschreibung "network" enthält. |
| <b>-f Befehl</b>   | Findet den Namen und Abschnitt einer Manpage zu <i>Befehl</i> (entspricht whatis). Beispiel: man -f passwd zeigt kurz "passwd (1) - change user password".                                       |
| <b>-a</b>          | Zeigt nacheinander alle Manpages an, die zum angegebenen Namen passen (wenn z.B. ein Name in mehreren Abschnitten existiert).                                                                    |
| <b>-P Pager</b>    | Legt ein anderes Pager-Programm fest (Standard ist less). Beispiel: man -P cat ls würde die ls-Manpage durch cat leiten.                                                                         |

## **Option      Beschreibung**

**--help, --version** Zeigt Hilfe bzw. Version von man.

### **whatis**

**Beschreibung:** Zeigt eine kurze Ein-Satz-Beschreibung zu einem Befehl an (frägt die Datenbank der *whatis*-Einträge ab). Es ist eine Kurzform von *man -f* und liefert die Synopsis aus der Manpage.

#### **Syntax:**

*whatis <Befehl/Topic>*

#### **Optionen:**

## **Option      Beschreibung**

*(keine üblichen Optionen)* *whatis* wird meist ohne Optionen verwendet. Es greift auf die gleichen Datenbanken wie *apropos/man -k* zu. Sollte die Datenbank nicht aktuell sein, kann sie mit *mandb* erneuert werden.

## **Benutzer- und Gruppenverwaltung**

### **passwd (Passwort ändern/setzen)**

**Beschreibung:** Ändert das Passwort eines Benutzerkontos oder verwaltet Konto-Sperren. Als normaler Benutzer kann man mit *passwd* sein eigenes Passwort ändern; als *root* kann man mit *passwd <user>* das Passwort eines anderen Accounts setzen oder bestimmte Eigenschaften (Sperren, Ablaufdatum) ändern.

#### **Syntax:**

*passwd [OPTION] [BENUTZER]*

*Standardaufruf ohne Benutzer wirkt auf den aktuellen Benutzer. Bei Angabe eines Benutzernamens werden dessen Passwortinformationen geändert (Root-Rechte erforderlich).*

#### **Optionen:**

## **Option   Beschreibung**

**-l, --lock** Sperrt das Benutzerkonto, indem das Passwort in */etc/shadow* mit einem Präfix (meist ! oder \*) unbrauchbar gemacht wird [wiki.ubuntuusers.de](http://wiki.ubuntuusers.de). Der Benutzer kann sich danach nicht mehr anmelden, bis das Konto wieder entsperrt wird.

## Option Beschreibung

- u, -- unlock Entsperrt ein zuvor gesperrtes Konto (entfernt das Sperr-Präfix im verschlüsselten Passwort)[wiki.ubuntuusers.de](http://wiki.ubuntuusers.de).
- d, -- delete Löscht das Passwort des Benutzers (leer setzen). Achtung: Damit wird eine passwortlose Anmeldung möglich, sofern nicht andersweitig gesichert[wiki.ubuntuusers.de](http://wiki.ubuntuusers.de).
- e, -- expire Markiert das Passwort als abgelaufen. Beim nächsten Login muss der Benutzer ein neues Passwort setzen.
- x *TAGE* Setzt die maximale Gültigkeitsdauer eines Passworts auf *TAGE*. Nach Ablauf muss es geändert werden. (Änderung des PASS\_MAX\_DAYS in /etc/shadow.)
- n *TAGE* Setzt die minimale Anzahl an Tagen zwischen zwei Passwortänderungen (PASS\_MIN\_DAYS). Der Benutzer kann sein Passwort erst nach *TAGE* Tagen erneut ändern.
- w *TAGE* Warnzeit in Tagen vor Passwortablauf (PASS\_WARN\_AGE). Der Benutzer erhält *TAGE* Tage vor Ablauf eine Warnung beim Login.
- s Zeigt den Passwortstatus an (Gesperrt? nicht gesetzt? Datum der letzten Änderung usw.). passwd -S <user> gibt eine Statuszeile aus.

*Interaktives Verhalten:* Wird passwd ohne Optionen aufgerufen, fordert es zur Eingabe des aktuellen Passworts (zur Authentifizierung, falls nicht root) und zweimal zur Eingabe des neuen Passworts auf (zwecks Bestätigung). Es verweigert einfache Passwörter gemäß der System-Policy (Pluggable Authentication Modules/pam Richtlinien). Root-Benutzer muss das alte Passwort nicht kennen.

## chage (Change Age – Passwortalter und Gültigkeit)

**Beschreibung:** Verändert Einstellungen zum Passwort-Aging – also Ablaufdatum, Mindestalter, Warnzeit etc. Mit chage kann ein Administrator festlegen, wann ein Benutzer sein Passwort ändern muss bzw. wann der Account deaktiviert wirdfile-4fdk7rvx39azjretinarqb.

### Syntax:

chage [OPTION]... BENUTZER

### Optionen:

## Option Beschreibung

|                      |                                                                                                                                                                                                                                                                                                |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                      | Listet die aktuellen Passwort- und Alterseinstellungen für den Benutzer in einem lesbaren Format auf (Letzte Änderung, Min/Max Alter, Warnzeit, Inaktivzeit, Ablaufdatum) – Übersicht.                                                                                                         |
| <b>-l</b>            |                                                                                                                                                                                                                                                                                                |
| <b>-m TAGE</b>       | Setzt die minimale Anzahl von Tagen <i>TAGE</i> , die ein Passwort gültig sein muss, bevor es geändert werden darf (PASS_MIN_DAYS).                                                                                                                                                            |
| <b>-M TAGE</b>       | Setzt die maximale Gültigkeitsdauer in Tagen (PASS_MAX_DAYS). Nach <i>TAGE</i> Tagen läuft das Passwort ab und muss geändert werden.                                                                                                                                                           |
| <b>-w TAGE</b>       | Setzt die Anzahl der Tage <i>TAGE</i> Vorwarnzeit vor Passwortablauf (PASS_WARN_AGE).                                                                                                                                                                                                          |
| <b>-I TAGE</b>       | Legt fest, wie viele Tage nach Passwortablauf der Account <b>inaktiv</b> wird (PASS_INACTIVE_DAYS). Ein Wert von 0 deaktiviert den Account sofort nach Ablauf des Passworts, -1 schaltet diese Funktion aus <a href="http://wiki.ubuntuusers.de/wiki.ubuntuusers.de">wiki.ubuntuusers.de</a> . |
| <b>-E DATUM</b>      | Setzt ein Ablaufdatum für das Benutzerkonto (im Format YYYY-MM-DD). Nach diesem Datum wird der Account deaktiviert <a href="http://wiki.ubuntuusers.de">wiki.ubuntuusers.de</a> . Das Datum -1 (oder "" leer) entfernt ein gesetztes Ablaufdatum (kein Ablauf).                                |
| <b>-d YYYY-MM-DD</b> | Setzt das Datum der letzten Passwortänderung auf das angegebene Datum (wirkt wie Backdating oder um zu triggern, dass ein Passwort abläuft). Kann auf 0 gesetzt werden, um das Passwort als sofort ablaufend zu markieren.                                                                     |
| <b>-h, --help</b>    | Zeigt Hilfe an.                                                                                                                                                                                                                                                                                |

*Hinweis:* Viele dieser Einstellungen lassen sich auch mit passwd (Optionen -x, -n, -w, -i, -e) vornehmen. chage bietet eine dedizierte Schnittstelle dafür. Ohne Optionen (als root) startet chage <user> interaktiv einen Dialog, um alle Werte einzustellen.

## w

**Beschreibung:** Zeigt an, welche Benutzer gerade am System angemeldet sind und was sie gerade tun. w fasst Informationen über aktuell eingeloggte User und deren laufende Prozesse zusammen (inkl. Login-TTY, Leerlaufzeit, genutzte CPU-Zeit, aktueller Befehl)  
file-4fdk7rvx39azjretinarqb.

## Syntax:

w [OPTION]

## Optionen:

## **Option    Beschreibung**

|                      |                                                                                                                                                                                                                                                   |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-h</b>            | Blendet die Kopfzeile in der Ausgabe aus (unterdrückt die Überschriftenzeile).                                                                                                                                                                    |
| <b>-s</b>            | Kurze Ausgabe. Unterdrückt die Spalten für Login-Zeit, Leerlauf etc. (nicht auf allen Systemen vorhanden).                                                                                                                                        |
| <b>-f</b>            | Zeigt Informationen zu von utmp als <i>geflagt</i> markierten Einträgen. Auf einigen Systemen steuert dies die Anzeige von Remote-Host oder Exit-Informationen. (Bei neueren procps-Versionen hat -f keine Funktion, da Host immer gezeigt wird.) |
| <b>-V, --version</b> | Zeigt Versionsinfo von w.                                                                                                                                                                                                                         |

*Die Ausgabe von w enthält in der ersten Zeile System-Uptime und Load Average. Danach folgen pro eingeloggtem Benutzer: Benutzername, Terminal, Remote-Host, Login-Zeit, Leerlaufdauer, CPU-Verbrauch und der aktuell laufende Prozess (TTY-Prozess).*

## **who**

**Beschreibung:** Zeigt Informationen über die aktuell angemeldeten Benutzer. Im Gegensatz zu w liefert who primär Login-Zeitpunkte und Terminalsessions, jedoch keine Prozess- oder Idle-Infos.

### **Syntax:**

who [OPTION]... [Datei]

*Standard liest who die System-Login-Datenbank /var/run/utmp und gibt für jeden Eintrag Benutzer, Terminal, Datum/Zeit und ggf. Host aus.*

### **Optionen:**

## **Option    Beschreibung**

|           |                                                                                                                  |
|-----------|------------------------------------------------------------------------------------------------------------------|
| <b>-H</b> | Zeigt eine Kopfzeile (Überschriften) über den Spalten an.                                                        |
| <b>-q</b> | "Quick who": Zeigt nur Benutzernamen und die Anzahl der angemeldeten Nutzer. (Gleichbedeutend mit users-Befehl.) |
| <b>-b</b> | Zeigt die Zeit des letzten Systemstarts (Boot time) an.                                                          |
| <b>-r</b> | Zeigt den aktuellen Runlevel des Systems.                                                                        |

| <b>Option</b>            | <b>Beschreibung</b>                                                                                                                                                                                         |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-m</b>                | Zeigt nur die Einträge für das aktuelle Terminal an (beschränkt auf den Benutzer, der den Befehl aufruft).                                                                                                  |
| <b>-u</b>                | Zeigt zusätzlich die Leerlaufzeit (Idle) und den PID des Login-Prozesses für jeden Benutzer.                                                                                                                |
| <b>-T</b>                | Zeigt den Terminal-Status ( $\pm$ Flag) an: + bedeutet terminal ist bei mesg schreibbar, - bedeutet nicht schreibbar, ? unbekannt. (In vielen who-Implementierungen bereits Standardverhalten ohne Option.) |
| <b>--help, --version</b> | Anzeigen von Hilfe bzw. Versionsinfo.                                                                                                                                                                       |

### **useradd (Benutzer anlegen)**

**Beschreibung:** Legt ein neues Benutzerkonto an. useradd ist ein Low-Level-Befehl, der die Systemkontodateien (wie /etc/passwd, /etc/shadow, /etc/group) direkt verändert, um einen neuen User einzurichten [pq.hosting](#). In Debian/Ubuntu-Umgebungen wird oft das Wrapper-Skript adduser verwendet, das auf useradd aufbaut und interaktiv ein paar Dinge abfragt; useradd selbst ist nicht interaktiv und benötigt alle Optionen zum Festlegen der Eigenschaften des neuen Accounts.

#### **Syntax:**

useradd [OPTION]... BENUTZERNAME

**Optionen:** (häufigste Optionen beim Anlegen eines Benutzers)

| <b>Option</b>   | <b>Beschreibung</b>                                                                                                                                                                                                               |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-c</b>       | Kommentarfeld (GECOS) für den Benutzer, z.B. voller Name oder Hinweis.<br><i>Kommentar</i> Wird in /etc/passwd eingetragen <a href="#">wiki.ubuntuusers.de</a> .                                                                  |
| <b>-d HOME</b>  | Pfad des Home-Verzeichnisses. Standard: /home/<Benutzername>. Kann geändert werden; wenn -m verwendet wird, wird dieses Verzeichnis erstellt und mit Standard-Dateien aus /etc/skel befüllt <a href="#">wiki.ubuntuusers.de</a> . |
| <b>-m</b>       | Legt das Home-Verzeichnis an und kopiert Standard-Dateien hinein. Sollte gesetzt sein, außer man möchte <i>kein</i> Home-Verz. (Ohne -m wird das Home nicht automatisch erstellt) <a href="#">wiki.ubuntuusers.de</a> .           |
| <b>-s SHELL</b> | Login-Shell des Benutzers. Standard ist normalerweise /bin/bash (oder was in /etc/default/useradd als SHELL definiert ist). Kann z.B. auf /sbin/nologin gesetzt werden für System-Accounts <a href="#">wiki.ubuntuusers.de</a> .  |

| <b>Option</b>             | <b>Beschreibung</b>                                                                                                                                                                                                                                                                                                                                                      |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-u <i>UID</i></b>      | Gewünschte User-ID (numerisch). Darf nicht bereits vergeben sein (außer mit -o). Wenn nicht angegeben, wird die nächste freie UID > 999 gewählt <a href="#">wiki.ubuntuusers.de</a> .                                                                                                                                                                                    |
| <b>-o</b>                 | Erlaubt die Verwendung einer bereits vergebenen UID (UID-Doppelungen). Zusammen mit -u benutzt, um zwei Benutzer mit identischer UID anzulegen. (Normalerweise nicht empfohlen, da UID eindeutig sein sollen.)                                                                                                                                                           |
| <b>-g <i>GRP</i></b>      | Primäre Gruppe (Name oder GID), der der Benutzer angehört <a href="#">wiki.ubuntuusers.de</a> . Standard: eine neue Gruppe mit gleichem Namen wie der Benutzer (bei entsprechender Systemkonfiguration) oder eine in /etc/default/useradd definierte GROUP. Die angegebene Gruppe muss existieren.                                                                       |
| <b>-G <i>Liste</i></b>    | Liste zusätzlicher Gruppen (sekundäre Gruppen), denen der Benutzer angehören soll <a href="#">wiki.ubuntuusers.de</a> . Gruppennamen durch Komma getrennt angeben (ohne Leerzeichen).                                                                                                                                                                                    |
| <b>-p <i>Passwort</i></b> | Das verschlüsselte Passwort für den Account (wie es z.B. von openssl passwd oder crypt(3) erzeugt würde) <a href="#">wiki.ubuntuusers.de</a> . <b>Wichtig:</b> Aus Sicherheitsgründen sollte man hier <i>nicht</i> ein Klartextpasswort übergeben. Bleibt -p leer, wird der Account ohne gesetztes Passwort angelegt (Anmeldung evtl. gesperrt bis passwd gesetzt wird). |
| <b>-e</b>                 | (bei useradd nicht üblich – wird bei usermod benutzt für Ablauf setzen. Bei einigen Systemen könnte useradd -e <Datum> ein Ablaufdatum setzen.)                                                                                                                                                                                                                          |
| <b>-f <i>TAGE</i></b>     | Anzahl an Tagen nach Passwortablauf, bis der Account deaktiviert wird (Analog zu chage -l). -f 0 würde den Account sofort mit Passwortablauf sperren. Standard -1 (nie sperren).                                                                                                                                                                                         |
| <b>-N</b>                 | (Bei einigen Systemen: <i>No user group</i> – verhindert das automatische Anlegen einer gleichnamigen Gruppe.)                                                                                                                                                                                                                                                           |
| <b>-r</b>                 | Erstellt einen System-Account (UID unter einer gewissen Schwelle, z.B. <1000, kein Expire, kein Mail-Spool). Oft genutzt für Systembenutzer ohne Login.                                                                                                                                                                                                                  |
| <b>-M</b>                 | Kein Home-Verzeichnis anlegen (selbst wenn Default CREATE_HOME aktiviert ist).                                                                                                                                                                                                                                                                                           |

| <b>Option</b>     | <b>Beschreibung</b>                                                                                                                 |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <b>-k SKELDIR</b> | Gibt ein alternatives Skeleton-Verzeichnis an (anstatt /etc/skel), von wo aus bei -m die initialen Dateien ins Home kopiert werden. |
| <b>-D</b>         | Zeigt die aktuellen Standardwerte für useradd an (oder mit anderen -D-Optionen können diese verändert werden).                      |

*Nach dem Anlegen mit useradd sollte meist ein Passwort gesetzt werden (passwd <user>), außer es handelt sich um einen Account ohne Login.*

### **usermod (Benutzerkonto ändern)**

**Beschreibung:** Modifiziert ein bestehendes Benutzerkonto – z.B. Name, UID, Primärgruppe, zusätzliche Gruppen, Login-Shell, Expire-Datum usw.[wiki.ubuntuusers.dewiki.ubuntuusers.de](http://wiki.ubuntuusers.dewiki.ubuntuusers.de). Im Grunde die gleichen Optionen wie useradd, aber angewendet auf bestehende Benutzer. Häufig genutzt, um Benutzer umzubenennen, in andere Gruppen aufzunehmen oder Accounts zu sperren.

#### **Syntax:**

usermod [OPTION]... LOGIN

#### **Optionen:**

| <b>Option</b>         | <b>Beschreibung</b>                                                                                                                                                                                                                                                                                                                                                           |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-c Kommentar</b>   | Neuen Kommentar (GECOS-Feld) setzen <a href="http://wiki.ubuntuusers.dewiki.ubuntuusers.de">wiki.ubuntuusers.dewiki.ubuntuusers.de</a> .                                                                                                                                                                                                                                      |
|                       | Ändert das Home-Verzeichnis des Benutzers auf <i>Neues_Home</i> .                                                                                                                                                                                                                                                                                                             |
| <b>-d Neues_Home</b>  | <b>Hinweis:</b> Ohne weiteres passiert das alte Home nicht automatisch mit. Mit zusätzlicher Option <b>-m</b> werden die Inhalte des alten Home-Verz. ins neue Verzeichnis verschoben und das neue Verzeichnis ggf. erstellt <a href="http://wiki.ubuntuusers.dewiki.ubuntuusers.de">wiki.ubuntuusers.dewiki.ubuntuusers.de</a> .                                             |
| <b>-m</b>             | (Mit -d verwenden) Verschiebt beim Ändern des Homepfads (-d) die Dateien vom alten ins neue Home und erstellt es falls nötig <a href="http://wiki.ubuntuusers.dewiki.ubuntuusers.de">wiki.ubuntuusers.dewiki.ubuntuusers.de</a> .                                                                                                                                             |
| <b>-l Neuer_Login</b> | Ändert den Login-Namen des Benutzers (Umbenennung des Accounts) <a href="http://wiki.ubuntuusers.dewiki.ubuntuusers.de">wiki.ubuntuusers.dewiki.ubuntuusers.de</a> . Der Benutzername in allen Systemkontodateien wird angepasst, <i>nicht</i> jedoch automatisch der Name des Home-Verzeichnisses oder Mail-Spool (diese müssten manuell umbenannt werden, falls gewünscht). |
| <b>-u Neue_UID</b>    | Ändert die numerische Benutzer-ID auf <i>Neue_UID</i> . Falls die UID bereits vergeben ist, muss -o benutzt werden. Dateien im Home-Verzeichnis mit alter UID werden auf neue UID umgestellt; Dateien                                                                                                                                                                         |

| Option                              | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                     | außerhalb des Homes muss der Admin selbst anpassen <a href="#">wiki.ubuntuusers.dewiki.ubuntuusers.de</a> .                                                                                                                                                                                                                                                                                        |
| <b>-o</b>                           | Erlaubt die Verwendung einer bereits vergebenen UID (nur zusammen mit -u sinnvoll).                                                                                                                                                                                                                                                                                                                |
| <b>-g</b><br><i>Neue_PrimGruppe</i> | Ändert die primäre Gruppenzugehörigkeit (Name oder GID) <a href="#">manpages.ubuntu.com</a> . Die neue Gruppe muss existieren. Alle Dateien im Home, die der alten primären Gruppe gehörten, werden auf die neue Gruppe übertragen <a href="#">manpages.ubuntu.com</a> . (Dateien außerhalb des Homes müssen ggf. manuell angepasst werden.)                                                       |
| <b>-G Gruppe(n)</b>                 | Setzt die Liste der sekundären Gruppen neu für den Benutzer <a href="#">manpages.ubuntu.com</a> . <i>Wichtig:</i> Ohne <b>-a</b> entfernt diese Option den Benutzer aus allen Gruppen, die nicht in der Liste stehen <a href="#">wiki.ubuntuusers.demanpages.ubuntu.com</a> . Um einen Benutzer zusätzlichen Gruppen hinzuzufügen, sollte man <b>-a -G</b> gemeinsam verwenden (siehe <b>-a</b> ). |
| <b>-a</b>                           | <i>Append</i> – fügt den Benutzer zusätzlichen Gruppen hinzu, anstatt die Gruppenzugehörigkeit zu überschreiben <a href="#">wiki.ubuntuusers.dewiki.ubuntuusers.de</a> . <b>Nur in Kombination mit -G verwenden</b> , um den Benutzer der/die angegebenen Gruppen hinzuzufügen, ohne ihn aus anderen zu entfernen. (Siehe Warnung unten.)                                                          |
| <b>-s Shell</b>                     | Ändert die Login-Shell des Benutzers <a href="#">wiki.ubuntuusers.de</a> . Z.B. auf /sbin/nologin setzen, um interaktive Logins zu unterbinden.                                                                                                                                                                                                                                                    |
| <b>-e YYYY-MM-DD</b>                | Setzt das Ablaufdatum des Accounts (das Datum, an dem das Konto deaktiviert wird) <a href="#">wiki.ubuntuusers.de</a> . Nach diesem Datum kann sich der User nicht mehr anmelden. usermod -e "" user entfernt ein Ablaufdatum (setzt es auf <i>nie</i> ).                                                                                                                                          |
| <b>-f Tage</b>                      | Setzt die Anzahl an Tagen nach Passwortablauf, bis der Account deaktiviert wird <a href="#">wiki.ubuntuusers.de</a> . -f 0 deaktiviert sofort nach Ablauf; -f -1 entfernt eine evtl. gesetzte Deaktivierung (nie sperren nach Ablauf).                                                                                                                                                             |
| <b>-L</b>                           | Sperrt das Benutzerkonto (lock). Fügt ein ! dem Passwort-Hash voran (im Shadow-Eintrag) <a href="#">wiki.ubuntuusers.de</a> , was die Anmeldung verhindert. Kann nicht mit -p oder -U kombiniert werden.                                                                                                                                                                                           |

| Option         | Beschreibung                                                                                                                                                                                                                 |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-U</b>      | Entsperrt ein zuvor gesperrtes Passwort (passwd -l). Entfernt das ! im Shadow-Passwortfeld, sodass der ursprüngliche Hash wieder wirksam wird <a href="#">wiki.ubuntuusers.de</a> . Nicht zusammen mit -L oder -p verwenden. |
| <b>-p Hash</b> | Setzt das <i>verschlüsselte</i> Passwort (Hash) des Users auf den angegebenen Wert <a href="#">wiki.ubuntuusers.de</a> . Sollte mit Vorsicht genutzt werden – für Passwortänderung ist passwd meist besser geeignet.         |
| <b>--help</b>  | Hilfe anzeigen.                                                                                                                                                                                                              |

#### Wichtige Hinweise:

- **Warnung:** Die Kombination von -G ohne -a überschreibt die Gruppenliste vollständig. Das heißt, der Benutzer wird aus allen Gruppen entfernt, die nicht explizit bei -G genannt sind[wiki.ubuntuusers.de](#). Um einen User nur hinzuzufügen, niemals ohne -a verwenden!
- Man kann keinen angemeldeten Benutzer umbenennen oder dessen UID ändern, solange er eingeloggt ist oder laufende Prozesse hat. Zudem keine Änderungen an dem eigenen Account durchführen, während man noch mit diesem angemeldet ist (kann zu Inkonsistenzen führen).
- Beim Ändern des Benutzernamens oder UIDs sollten evtl. Crontab-Dateien und geplante at-Jobs manuell umbenannt/angepasst werden, und in NIS/LDAP-Umgebungen müssen Änderungen dort vorgenommen werden[wiki.ubuntuusers.de](#).

### userdel (Benutzer löschen)

**Beschreibung:** Löscht ein Benutzerkonto aus dem System (entfernt die Einträge in /etc/passwd, /etc/shadow und ggf. /etc/group falls der Benutzer eine eigene Gruppe hatte). Optional können auch das Home-Verzeichnis und die Mail-Spool des Benutzers gelöscht werden.

#### Syntax:

userdel [OPTION] BENUTZER

#### Optionen:

##### Option Beschreibung

|           |                                                                                                                                                                         |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-r</b> | Entfernt <i>rekursiv</i> das Heimatverzeichnis des Benutzers <b>und</b> seine Mail-Spool (üblicherweise /var/spool/mail/<user>), sowie alle Dateien darin, beim Löschen |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## **Option Beschreibung**

des Accounts. Ohne -r bleiben Home und Mails unangetastet file-4fdk7rvx39azjretinarqb.

**-f** Force: Erzwingt das Löschen des Accounts, selbst wenn der Benutzer noch angemeldet ist oder laufende Prozesse hat. (Warnung: kann zu inkonsistenten Systemzuständen führen, Prozess bleibt ohne Benutzer etc. Diese Option sollte vermieden werden.)

Ohne Option wird der Account aus den Systemdateien gelöscht, aber das (keine) Home-Verzeichnis, die Dateien und Mails bleiben bestehen. Admin muss sie ggf. manuell entfernen oder anderweitig zuordnen.

*Hinweis:* userdel bricht ab, wenn der Benutzer gerade angemeldet ist (außer mit -f). Es ist ratsam, vor dem Löschen sicherzustellen, dass keine Prozesse mehr unter der betreffenden UID laufen (z.B. mit pkill -u username beenden). Dateien außerhalb des Home, die dem Benutzer gehören, verbleiben auf dem Dateisystem mit der nun unbenutzten UID – diese müssten manuell gesucht und entfernt oder neu zugeordnet werden (z.B. mit find / -uid <oldUID>).

## **id**

**Beschreibung:** Zeigt die Benutzer- und Gruppen-Identitäten eines Benutzerkontos an. Standardmäßig gibt id die UID, primäre GID und zugehörigen Gruppennamen des aktuellen Benutzers zurück file-4fdk7rvx39azjretinarqb. Mit Angabe eines Benutzernamens kann man die IDs für einen anderen Account abrufen.

### **Syntax:**

id [OPTION]... [BENUTZER]

### **Optionen:**

#### **Option    Beschreibung**

**(ohne)** Ohne Options werden UID, Benutzername, GID, Gruppenname und alle zusätzlichen Gruppen des Benutzers aufgelistet (z.B. uid=1000(max) gid=1000(max) Gruppen=1000(max),27(sudo),...).

**-u** Gibt nur die effektive User-ID (UID, numerisch) aus. Beispiel: id -u alice könnte 1001 ausgeben.

**-g** Gibt nur die effektive Gruppen-ID (GID, numerisch) aus.

**-G** Gibt alle Gruppen-IDs (numerisch) aus, in denen der Benutzer Mitglied ist (Primär- und Sekundärgruppen).

## **Option      Beschreibung**

|                          |                                                                                                                                                     |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-n</b>                | Gibt Namen anstelle von numerischen IDs. Kann in Kombination mit -u, -g, -G verwendet werden. Z.B. id -Gn alice listet alle Gruppennamen von alice. |
| <b>-r</b>                | Zeigt die reale UID/GID an (falls z.B. mit sudo aufgerufen, reale vs. effektive ID unterscheiden).                                                  |
| <b>-Z</b>                | Zeigt SELinux-Sicherheitskontext, falls SELinux aktiv ist (z.B. unconfined_u:unconfined_r:unconfined_t:s0 für einen unconfined Benutzer).           |
| <b>--help, --version</b> | Hilfe bzw. Version anzeigen.                                                                                                                        |

## **groupadd (Gruppe anlegen)**

**Beschreibung:** Legt eine neue Gruppe im System an. Fügt einen Eintrag in /etc/group (und ggf. /etc/gshadow) hinzu.

**Syntax:**

groupadd [Optionen] GRUPPENNAME

**Optionen:**

## **Option      Beschreibung**

|                           |                                                                                                                                                           |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-g GID</b>             | Verwendet die angegebene numerische GID für die neue Gruppe. Sollte noch nicht vergeben sein (oder mit -o überschrieben werden).                          |
| <b>-o</b>                 | Erlaubt die Verwendung einer bereits existierenden GID (zusammen mit -g zu nutzen, um GID-Doppel zu erzwingen).                                           |
| <b>-f</b>                 | "Force": falls die Gruppe schon existiert, keinen Fehler melden; falls die angegebene GID bereits vergeben ist, wählt eine andere (ignoriert -o).         |
| <b>-r</b>                 | Erstellt eine Systemgruppe (GID wird aus dem System-Bereich gewählt, z.B. <1000).                                                                         |
| <b>-K Schlüssel= Wert</b> | Überschreibt einen Eintrag aus /etc/login.defs temporär (z.B. -K GID_MIN=500 um untere GID-Bereiche anzupassen).                                          |
| <b>-p Passwort</b>        | Setzt das (verschlüsselte) Gruppenpasswort direkt. (Gruppenpasswörter werden selten genutzt – ermöglicht mit newgrp temporär in eine Gruppe zu wechseln.) |

## **groupmod (Gruppe ändern)**

**Beschreibung:** Ändert Eigenschaften einer bestehenden Gruppe – hauptsächlich Name oder GID.

**Syntax:**

groupmod [Optionen] GRUPPE

**Optionen:**

| Option              | Beschreibung                                                                                                                                                                                    |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -n <i>NeuerName</i> | Ändert den Gruppennamen in <i>NeuerName</i> . Der alte Name in allen relevanten Systemdateien wird ersetzt.                                                                                     |
| -g <i>NeueGID</i>   | Ändert die GID der Gruppe. Muss eindeutig sein (oder mit -o doppelt erlaubt). Dateien mit alter GID außerhalb, die nicht automatisch erfasst werden, müssen eventuell manuell angepasst werden. |
| -o                  | Erlaubt das Setzen einer bereits vergebenen GID (nur mit -g relevant).                                                                                                                          |
| -p <i>Passwort</i>  | Ändert das (verschlüsselte) Gruppenpasswort. (Entspricht Editieren von /etc/gshadow).                                                                                                           |

**groupdel (Gruppe löschen)**

**Beschreibung:** Entfernt eine Gruppe aus der Systemgruppe-Datei. Die Gruppe wird aus /etc/group (und /etc/gshadow) getilgt.

**Syntax:**

groupdel GRUPPENNAME

**Hinweis:** Dieser Befehl hat keine besonderen Optionen – er löscht die Gruppe, sofern sie nicht als Primärgruppe eines Benutzers in /etc/passwd noch verwendet wird. Man kann keine Gruppe löschen, die noch als Hauptgruppe eines existierenden Benutzerkontos eingetragen ist; in dem Fall muss man ggf. erst den Benutzer anpassen oder löschen. Dateien auf dem System, die dieser Gruppe gehören, behalten die nun ungültige GID bei, bis man sie manuell ändert.

**su (Switch User)**

**Beschreibung:** Wechselt den Benutzerkontext im laufenden Terminal. Mit su („substitute user“) kann man einen anderen Benutzeraccount annehmen – standardmäßig root, wenn kein Benutzer angegeben wird. Dies erfordert die Eingabe des Zielbenutzer-Passworts (außer beim Wechsel zu root, dort abhängig von den Systemeinstellungen – oft ist root-Login per Passwort deaktiviert und man nutzt sudo).

**Syntax:**

su [OPTION]... [BENUTZER [-]]

*Wird BENUTZER weggelassen, versucht su auf den Superuser (root) zu wechseln.*

## Optionen:

| Option            | Beschreibung                                                                                                                                                                                                                                                                                                                             |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| - oder -l         | Startet eine <i>Login-Shell</i> der Zielusers, d.h. die Umgebung wird vollständig umgeschaltet als würde sich der Benutzer frisch anmelden. Das Home-Verzeichnis, Pfad (\$PATH) etc. werden übernommen. Beispiel: su - alice wechselt zu Benutzer <i>alice</i> mit deren Umgebung (äquivalent zu vollem Logout/Login als <i>alice</i> ). |
| -c Befehl         | Führt einen angegebenen <i>Befehl</i> als Zielbenutzer aus und kehrt dann zurück. Beispielsweise su -c "systemctl restart apache2" - root führt als root einen Neustart des Apache durch und beendet sich sofort wieder.                                                                                                                 |
| -m, -p            | Behalten die aktuelle Umgebung ( <i>nicht</i> auf Ziel-User umschalten). Der aktuelle \$HOME, \$PATH, usw. bleiben erhalten, nur die Rechte wechseln. Standardverhalten ohne -l. (In vielen su Implementierungen ist -m identisch mit -p: <i>preserve environment</i> ).                                                                 |
| -s Shell          | Startet die angegebene <i>Shell</i> statt der Standard-Login-Shell des Zielbenutzers. Der Pfad zur Shell muss im /etc/shells stehen, sofern nicht root.                                                                                                                                                                                  |
| -g Gruppe         | Wechselt in Kombination mit Benutzer auch die primäre Gruppe auf <i>Gruppe</i> . (Nur verfügbar in neueren util-linux su).                                                                                                                                                                                                               |
| -<Ziffer>         | (BSD su) Angabe einer Session-Klasse – unter Linux nicht genutzt.                                                                                                                                                                                                                                                                        |
| --help, --version | Anzeigen von Hilfe bzw. Version.                                                                                                                                                                                                                                                                                                         |

## Hinweise:

- su ohne - (also z.B. su root) wechselt zwar die Benutzer-ID, aber behält das aktuelle Verzeichnis und weite Teile der Environment von Ursprungskonto bei (\$LOGNAME, \$HOME bleiben z.B. vom ursprünglichen Nutzer). Das kann zu irritierenden Effekten führen (z.B. Schreibrechte im falschen Home). Daher nutzt man meist su - für einen *sauberen* Wechsel.
- Viele Distributionen konfigurieren aus Sicherheitsgründen su so, dass ein direkter Wechsel zu *root* per Passwort entweder deaktiviert ist oder kein root-Passwort gesetzt ist. Stattdessen wird sudo bevorzugt, um Root-Rechte zu erlangen.

## sudo (Superuser Do)

**Beschreibung:** Führt ein Kommando mit *erhöhten Rechten* aus, typischerweise als *root*, ohne die Benutzeridentität dauerhaft zu wechseln. Benutzer können mit sudo Befehle

mit Administratorrechten ausführen, sofern sie in /etc/sudoers entsprechend autorisiert sind. Im Gegensatz zu su wird hier das eigene Passwort abgefragt (nicht das des Zielusers), und nur spezifische Befehle/Optionen sind erlaubt gemäß Konfiguration.

### Syntax:

sudo [OPTION]... [COMMAND]

*Wird ein Befehl angegeben, führt sudo diesen als root (Standard) aus. Ohne Befehl zeigt sudo ggf. die erlaubten Befehle an oder wechselt in eine sudo-Shell (bei -s).*

### Optionen (Auswahl):

| Option             | Beschreibung                                                                                                                                                                                                                                            |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -l                 | Listet die erlaubten (und verbotenen) Befehle für den aufrufenden Benutzer auf (nach sudoers-Regeln). Hilfreich, um zu sehen, was man mit sudo ausführen darf.                                                                                          |
| -u <i>Benutzer</i> | Führt das angegebene Kommando als <i>Benutzer</i> aus (anstatt als root). Beispiel: sudo -u alice whoami zeigt "alice", wenn man als anderer User ausführt. Standard ist root.                                                                          |
| -s                 | Startet eine Shell als Zielbenutzer (Standard root). Ähnlich wie su, aber mittels sudo-Rechte. Es wird die im \$SHELL oder in passwd eingetragene Shell geöffnet.                                                                                       |
| -i                 | Simuliert eine vollständige Login-Shell wie su - (initialisiert Umgebung des Zielusers). Entspricht sudo -s plus Login-Umgebung.                                                                                                                        |
| -b                 | Führt den Befehl im Hintergrund aus ( <i>background</i> ). Sudo verlangt dennoch ggf. ein Passwort, aber detach danach.                                                                                                                                 |
| -E                 | Beibehaltung der aktuellen Environment-Variablen (normalerweise bereinigt sudo einige Variablen aus Sicherheitsgründen). Erlaubt das Mitnehmen der eigenen Umgebung. (Ob das erlaubt ist, kann in sudoers eingeschränkt sein.)                          |
| -p <i>Prompt</i>   | Definiert einen eigenen Passwort-Prompt. z.B. sudo -p "Passwort für %u:" -v (Platzhalter %u = Username).                                                                                                                                                |
| -v                 | "Validate": Aktualisiert/überprüft das eigene sudo-Authentifizierungsticket. D.h. fordert ggf. Passwort an, speichert das sudo Token (5min Standard) aber führt sonst nichts aus. (Verlängert die Zeit, in der man kein Passwort erneut eingeben muss.) |
| -k                 | Invalidiert das eigene sudo-Ticket sofort (nachfolgende sudo-Aufrufe verlangen wieder Passwort) – ( <i>kill credential</i> ).                                                                                                                           |

## Option Beschreibung

|                                                                  |                                                                                                                   |
|------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
|                                                                  | Setzt \$HOME in der Environment auf das Home des Zielusers (root).                                                |
| -H                                                               | Normalerweise belässt sudo \$HOME vom Aufrufer; mit -H verhält es sich wie su -.                                  |
| <b>--help, -h</b> Hilfe anzeigen (kurze Übersicht der Optionen). |                                                                                                                   |
| -K                                                               | Löscht das Zeitstempel-Ticket komplett (ähnlich -k aber gründlicher, forciert Passwort prompt beim nächsten Mal). |
| <b>--version</b>                                                 | Versionsinfo ausgeben (auch Liste der unterstützten Sicherheitsmechanismen etc.).                                 |

*Hinweise:* Bei erfolgreicher Authentifizierung legt sudo einen Zeitstempel (Ticket) an, sodass für nachfolgende Aufrufe (Standard ~5 Minuten) kein erneutes Passwort nötig ist. Dies kann per sudoers Einstellung (timestamp\_timeout) angepasst oder mit sudo -k/-K beeinflusst werden.

## pkexec (PolicyKit Execute)

**Beschreibung:** Führt ein Programm als ein anderer Benutzer aus (ähnlich wie sudo, aber Teil von PolKit/PolicyKit). Standardmäßig wird, sofern nicht anders angegeben, das Programm mit root-Rechten gestartet[freedesktop.org](http://freedesktop.org). pkexec erfordert, dass der aufrufende Benutzer berechtigt ist, die Aktion auszuführen (PolKit Policies); falls ja, wird eine Authentifizierungsdialog (grafisch oder textbasiert) angezeigt, um ggf. das Passwort einzugeben. Nach erfolgreicher Auth kann das gewünschte Kommando mit erhöhten Rechten ausgeführt werden.

### Syntax:

pkexec [--user <Benutzer>] <Kommando> [ARG...]

### Optionen:

#### Option Beschreibung

|                          |                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| --user <Benutzer>        | Führt das angegebene Programm als Benutzer aus. Standard ist root (Administratorkonto) <a href="http://freedesktop.org">freedesktop.org</a> .                                                                                                                                                                                                                                         |
| --disable-internal-agent | Unterdrückt den internen Authentifizierungs-Agent von pkexec. Normalerweise, falls kein grafischer PolKit-Auth-Agent verfügbar ist, startet pkexec einen eigenen textuellen Auth-Dialog im Terminal. Mit dieser Option wird in so einem Fall statt dessen das Programm <b>nicht</b> ausgeführt (Fehler, wenn kein Agent aktiv) <a href="http://freedesktop.org">freedesktop.org</a> . |

| Option | Beschreibung |
|--------|--------------|
|--------|--------------|

|                    |                                      |
|--------------------|--------------------------------------|
| --help / --version | Zeigt Hilfe bzw. Version von pkexec. |
|--------------------|--------------------------------------|

*Hinweise:* pkexec erbt ein Minimal-Umfeld (Environment), um das Einschleusen von Code via Umgebungsvariablen zu verhindern [freedesktop.org](http://freedesktop.org). Display- und Xauth-Variablen werden aus Sicherheitsgründen *nicht* an das Zielprogramm weitergegeben, außer es wurde in den PolKit-Regeln explizit erlaubt (GUI-Programme via pkexec sind daher tricky ohne entsprechende PolKit-Policy). In modernen Distributionen ist pkexec oft als Ersatz für grafische gksu-Aufrufe gedacht, um z.B. aus der Desktop-Umgebung heraus GUI-Administrationsprogramme mit Root-Rechten zu starten (wobei PolKit dann i.d.R. das Benutzer-Passwort abfragt).

## Berechtigungen und Zugriffskontrolle

### umask (Set Default Permissions Mask)

**Beschreibung:** Legt eine Berechtigungsschutzmaske fest, die bei neu erstellten Dateien und Verzeichnissen die Standardzugriffsrechte beeinflusst. umask subtrahiert Bits von den vollen Zugriffsrechten. So bestimmt sie, welche Rechte *NICHT* automatisch gesetzt werden dürfen. Dies wirkt pro Prozess/Session. Als Shell-Builtin zeigt umask ohne Parameter die aktuelle Maske an.

#### Syntax:

umask [Option] [Maske]

#### Optionen / Verwendung:

- Ohne Argument zeigt umask die aktuelle Mask in Oktalnotation an (z.B. 0022). Mit Option **-S** in symbolischer Form (z.B. u=rwx,g=rx,o=rx entspricht Maske 022).
- Mit Angabe einer neuen Maske (z.B. umask 027) wird diese gesetzt. Die Maske wird meist oktal angegeben: jedes Bit, das in der Maske 1 ist, nimmt der neu erstellten Datei die entsprechende Berechtigung weg. Beispiel: Maske 022 bedeutet, neuen Dateien werden keine Gruppe- und Andere-Schreibrechte gegeben (d.h. max Resultatrechte für Datei typ. 644, für Verzeichnis 755).
- **Symbolische Maske:** Man kann auch wie bei chmod symbolisch angeben, z.B. umask u=rwx,g=rx,o=rx für dasselbe Ergebnis (Maske 022).

*Hinweis:* Die Umask wirkt auf Systemaufruf-Ebene: Standard-Dateirechte sind 666 (rw-rw-rw) und für Verzeichnisse 777 (rwxrwxrwx); die Maske wird davon abgezogen. Z.B. Umask 002 ergibt Dateien mit 664 (rw-rw-r-) und Verzeichnisse mit 775. Umask 077 würde sehr restriktiv alles nur für den Benutzer selbst erlauben (Datei 600, Dir 700).

### chmod (Change Mode - Dateirechte ändern)

**Beschreibung:** Ändert die Zugriffsberechtigungen einer Datei oder eines Verzeichnisses. Man kann mit chmod sowohl numerisch (Oktalwerte) als auch symbolisch (u/g/o +-/= r,w,x) die Rechte setzen.

**Syntax:**

chmod [Optionen] <Modus> <Datei/Verzeichnis>...

**Optionen:**

| Option                             | Beschreibung                                                                                                                                            |
|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-R, --recursive</b>             | Rekursive Rechteänderung: traversiert Unterverzeichnisse und wendet chmod auf alle darin befindlichen Dateien/Ordner an. Wichtig bei Verzeichnisbäumen. |
| <b>-v, --verbose</b>               | Gibt für jede Datei eine Meldung aus, deren Rechte erfolgreich geändert wurden (und ggf. was geändert wurde).                                           |
| <b>-c, --changes</b>               | Wie verbose, aber nur Ausgaben für tatsächlich geänderte Rechte (keine Ausgabe, wenn Rechte schon so eingestellt waren).                                |
| <b>-f, --silent</b>                | Unterdrückt Fehlermeldungen (force).                                                                                                                    |
| <b>--reference=&lt;Vorlage&gt;</b> | Setzt die gleichen Berechtigungen wie die Referenz-Datei <Vorlage> hat, auf die angegebenen Dateien. Statt Modus muss man --reference nutzen.           |

**Modus-Angaben:**

- **Numerisch (Oktal):** Eine dreistellige (oder vierstellige mit Sonderbit) Oktalzahl. Z.B. chmod 640 datei.txt setzt Rechte auf rw- r-- --- (6=rw, 4=r, 0=kein Recht). Jede Ziffer entspricht Owner, Group, Others. Beispiel: chmod 755 script.sh => rwx für Besitzer (7), rx für Gruppe (5), rx für andere (5). Eine führende vierte Ziffer kann spezielle Bits setzen: 4 für Setuid, 2 für Setgid, 1 für Sticky-Bit (z.B. 1755 für Sticky).
- **Symbolisch:** Format: [ugoa]\*[+-=][rwxXst]\* evtl. durch Kommata getrennt für mehrere Operationen in einem Aufruf. Beispiele:
  - chmod u+x,g-w,o= file – dem Besitzer Ausführungsrecht geben, der Gruppe Schreibrecht entziehen, andere bekommen exakt keine Rechte.
  - chmod a+r file – allen (a für all) Leserechte hinzufügen (ohne andere Bits zu ändern).
  - chmod g=u Datei – setzt die Gruppenrechte gleich den Benutzerrechten der Datei (nützliche Kurzform).

- X: Spezialflag in symbolischer Notation – steht für "execute if directory or if any execute bit is set on any of owner/group/other". Wird meist bei -R verwendet, um rekursiv nur Verzeichnissen das x zu geben.
- s für setuid/setgid (in u/g Kontext), t für Sticky-Bit (im o-Kontext).
- Beispiel: chmod u+s script.sh – setzt das setuid-Bit auf die Datei (falls ausführbar).

## **chown (Change Owner)**

**Beschreibung:** Ändert den Dateibesitzer und/oder die Gruppe einer Datei oder eines Verzeichnissesfile-4fdk7rvx39azjretinarqbfile-4fdk7rvx39azjretinarqb. chown kann sowohl den User als auch die Gruppe in einem Befehl setzen. Nur root (oder Capability CHOWN) darf Eigentümer von Dateien ändern. Normale Benutzer können in der Regel mittels chgrp die Gruppenzugehörigkeit ändern, wenn sie Mitglied der Zielgruppe sind (und Besitzer der Datei bleiben).

### **Syntax:**

chown [Optionen] <neuerEigentümer>[:<neueGruppe>] DATEI...

### **Optionen:**

| <b>Option</b>                                            | <b>Beschreibung</b>                                                                                                                                                                                                           |
|----------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-R</b>                                                | Rekursiv – wendet die Besitzänderung auf alle Dateien und Unterverzeichnisse an (ganzer Baum).                                                                                                                                |
| <b>-h</b>                                                | Ändert den Besitzer von Symbolic Links selbst, statt des Ziels. (Auf den meisten Linux ist das Standard mit Kernel 2.6+; teils wird -h ignoriert, da chown auf Symlinks ohnehin Link selbst ändert, außer mit -L Option).     |
| <b>-f</b>                                                | Keine Fehlermeldungen ausgeben (silent/force).                                                                                                                                                                                |
| <b>-v</b>                                                | Verbose – listet jede Datei mit geänderten Owner/Group auf.                                                                                                                                                                   |
| <b>--</b><br><b>from=&lt;old_owner:&gt;old_group&gt;</b> | Ändert nur, wenn aktuelle Owner/Group genau wie angegeben ist. (Damit kann man selektiv bestimmte Kombis ersetzen, dient als Sicherheitsprüfung).                                                                             |
| <b>--dereference / -L</b>                                | <i>Follow symlinks:</i> Symbolische Links auflösen (statt Link selbst zu ändern wird das Ziel geändert). Standardverhalten für die meisten chown-Versionen ist bereits, dem Link <i>nicht</i> zu folgen (außer man setzt -L). |

| <b>Option</b> | <b>Beschreibung</b>                                                                                               |
|---------------|-------------------------------------------------------------------------------------------------------------------|
| <b>-P</b>     | <i>Don't follow symlinks</i> (explizit Link behandeln, Standard).                                                 |
| <b>-H</b>     | Folgen von Symlinks <i>im Kommandozeilen-Argument</i> (Top-Level), aber nicht rekursiv tiefer. (Selten benötigt). |

### **Verwendung:**

- Besitzer ändern: chown alice datei.txt – setzt *alice* als neuen Besitzer der Datei (Gruppe bleibt unverändert).
- Gruppe ändern mit chown: chown :developers projektX – behält Besitzer, setzt Gruppe auf *developers*. Notation: Vor dem Doppelpunkt nichts heißt Besitzer unverändert, aber *Gruppe* ändern.
- Besitzer und Gruppe zugleich: chown alice:staff datei.txt – Besitzer = *alice*, Gruppe = *staff*.
- Nur Gruppe ändern könnte auch mit eigenem Befehl chgrp staff datei.txt gemacht werden.

**Besonderheit:** Man kann numerische IDs angeben. chown 1001:100 datei setzt Besitzer UID=1001, GID=100. Falls man einen Benutzer- oder Gruppennamen mit Doppelpunkt hat, der numerisch beginnt, kann man -- verwenden um Verwechslung zu vermeiden oder explizit username: schreiben.

### **chgrp (Change Group)**

**Beschreibung:** Ändert die Gruppenzugehörigkeit einer Datei oder eines Verzeichnissesfile-4fdk7rvx39azjretinarqbfile-4fdk7rvx39azjretinarqb. Äquivalent zu chown <gruppe>. Wird benutzt, um den Gruppeneigentümer anzupassen.

### **Syntax:**

chgrp [Optionen] <neueGruppe> DATEI...

### **Optionen:**

| <b>Option</b> | <b>Beschreibung</b>                                                                                  |
|---------------|------------------------------------------------------------------------------------------------------|
| <b>-R</b>     | Rekursive Änderung der Gruppe in Unterverzeichnissen (alle Dateien/Dirs innerhalb werden angepasst). |
| <b>-f</b>     | Keine Fehlermeldungen bei Problemen ausgeben (silent).                                               |
| <b>-v</b>     | Ausführliche Ausgabe pro Datei.                                                                      |

| Option                   | Beschreibung                                                                                                                                                               |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| --                       | Übernimmt die Gruppenzugehörigkeit der <i>Vorlage</i> -Datei auf alle angegebenen Dateien. (Ignoriert das Gruppen-Argument und nutzt stattdessen die Gruppe der Referenz.) |
| <b>reference=Vorlage</b> |                                                                                                                                                                            |

*Hinweis:* Normale Benutzer dürfen chgrp nur verwenden, um eine Datei auf eine Gruppe zu setzen, der sie selbst angehören. Root kann beliebige Zuordnungen vornehmen.

### **lsattr (List Attributes)**

**Beschreibung:** Listet die erweiterten *Dateiattribute* (Filesystem-Attribute) von ext-Dateisystemen auf file-4fdk7rvx39azjretinarqb. Diese Attribute wie *immutable* oder *append-only* steuern besondere Verhalten auf niedriger Ebene (nur auf bestimmten FS wie ext2/3/4, XFS - mit anderen Tools - etc. vorhanden). lsattr zeigt pro Datei eine Zeichenfolge von Attribut-Flags.

#### **Syntax:**

lsattr [Optionen] [Datei/Verzeichnis]...

#### **Optionen:**

##### **Option Beschreibung**

|           |                                                                                                                                                                                                                                                 |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-a</b> | Zeigt auch versteckte Dateien (Dateien, die mit Punkt beginnen) an.<br>Standardmäßig ignoriert lsattr diese analog zu ls ohne -a.                                                                                                               |
| <b>-d</b> | Listet das Verzeichnis selbst auf, nicht dessen Inhalt. (Bei Angabe eines Verzeichnisses wird normalerweise dessen Inhalt rekursiv aufgelistet; mit -d sieht man nur die Attribute des Verzeichniseintrags selbst) file-4fdk7rvx39azjretinarqb. |
| <b>-R</b> | Rekursive Ausgabe der Attribute in allen Unterverzeichnissen.                                                                                                                                                                                   |
| <b>-v</b> | Zeigt die Dateiversionsnummer (Version im ext-Format) mit an, falls vorhanden.<br>(Das Attribut 'version' wird kaum benutzt, -v gibt eine zusätzliche Spalte mit Versionsnummer).                                                               |
| <b>-V</b> | Ausführliche Informationen über die Version des Programms und detaillierte Fehler ausgeben (verbose).                                                                                                                                           |

#### **Beispiel Ausgabe:**

```
$ lsattr wichtige_datei
----i-----e-- wichtige_datei
```

Hier bedeuten die Buchstaben: i = immutable (unveränderlich), e = extents (standard auf ext4). Bindestriche zeigen nicht gesetzte Flags. Wichtige Flags: i (immutable: Datei kann

nicht verändert/gelöscht umbenannt werden bis Attribut entfernt), **a** (append-only: nur anhängen erlaubt), **d** (no dump: Datei wird von Dump-Backup ignoriert), **A** (no atime updates), **S** (synchronous updates) etc.

### chattr (Change Attributes)

**Beschreibung:** Ändert die erweiterten Dateisystem-Attribute (wie immutable, append-only etc.) unter Linux auf ext2/ext3/ext4 Dateisystemenfile-4fdk7rvx39azjretinarqb. Nur Root darf die meisten dieser Attribute setzen.

#### Syntax:

chattr [Optionen] [+-=Attribute] Datei...

#### Optionen:

| Option                | Beschreibung                                                                   |
|-----------------------|--------------------------------------------------------------------------------|
| <b>-R</b>             | Rekursive Attributänderung in Unterverzeichnissenfile-4fdk7rvx39azjretinarqb.  |
| <b>-V</b>             | Ausführlich – zeigt an, welche Änderungen für welche Datei vorgenommen wurden. |
| <b>-f</b>             | Unterdrückt Fehlermeldungen (force).                                           |
| <b>-v</b>             | Setzt die Versionsnummer der Datei (falls vom Dateisystem unterstützt).        |
| <b>&lt;Nummer&gt;</b> | Selten benutzt – ext-Dateisysteme haben optional Dateiversionsnummern.         |

**Attribute setzen (Symbolik):** Man gibt nach möglichen Optionen eine Zeichenfolge an, der ein +, - oder = vorangestellt ist:

- + bedeutet diese Attribute *hinzufügen* (zusätzlich zu bestehenden).
- - bedeutet diese Attribute *entfernen*.
- = bedeutet *exakt setzen* (alle aktuellen Attribute durch diese ersetzen).

Die wichtigsten Attribut-Buchstaben auf ext4 sind:

- **i – immutable:** Datei kann nicht modifiziert, umbenannt oder gelöscht werdenfile-5y1edouysrow4zj6phjcqm. Prozesse können sie nicht öffnen im Schreibmodus. Schützt kritische Dateien (z.B. /etc/passwd) vor Änderungen, sogar von root (root muss erst chattr -i machen).
- **a – append only:** Datei kann nur angehängt, aber nicht überschrieben oder gelöscht werden. Typisch für Logfiles, damit niemand alte Einträge löscht.
- **e – extent format:** zeigt an, dass die Datei extents benutzt (automatisch auf ext4, nicht manuell gesetzt).

- **d – no dump:** Datei soll von Backup-Programmen, die dem Dump-Standard folgen, ignoriert werden.
- **A – no atime updates:** beim Lesen der Datei wird der atime-Zeitstempel nicht aktualisiert (Performance-Optimierung).
- **S – synchronous updates:** Änderungen an der Datei werden synchron auf den Datenträger geschrieben (wie bei chattr +S – ähnlich wie Ordner mit Dirsync mount option).
- **t – no tail-merging:** (relevant bei ext-Dateisystem, deaktiviert Tail-Merging bei Reverse-Mapping; selten benutzt).
- **T – top of directory hierarchy:** Verzeichnis wird als Top-Level für Ordnungszwecke in ext behandelt (verhindert bestimmte Ordnungsoptimierungen; selten).
- **u – undeletable:** Dateiinhalt soll beim Löschen inhaltlich gesichert werden, sodass man ihn wiederherstellen könnte. (Wird von ext4 i.d.R. nicht implementiert).

#### **Beispiel:**

- chattr +i datei.txt – setzt *immutable*.
- chattr -i datei.txt – entfernt dieses Flag wieder (dann ist Bearbeiten/Löschen wieder möglich).
- chattr =a Datei.log – setzt **nur** append-only und entfernt alle anderen Attribute (falls welche gesetzt waren). Danach kann in Datei.log nur noch angehängt werden.

#### **getfacl (Get File ACL)**

**Beschreibung:** Liest die Access Control Lists (erweiterten Zugriffsrechte) von Dateien oder Verzeichnissen aus und zeigt sie anfile-4fdk7rvx39azjretinarqb. Auf Linux-Systemen mit POSIX-ACL-Unterstützung (z.B. ext4 mit acl-Option eingehängt) kann man pro Datei feinere Berechtigungen für einzelne Benutzer oder Gruppen setzen. getfacl zeigt diese ACL-Einträge zusätzlich zu den Standardrechten.

#### **Syntax:**

getfacl [Optionen] Datei...

#### **Optionen:**

| <b>Option</b> | <b>Beschreibung</b> |
|---------------|---------------------|
|---------------|---------------------|

**-a** Zeigt auch die *verschachtelten ACLs* (falls vorhanden – nicht üblich bei POSIX ACL).

| <b>Option</b>        | <b>Beschreibung</b>                                                                                                                                          |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-e</b>            | Zeigt die Ausgabe im <i>effective</i> Modus – Standard. (Jeder Eintrag inkl. "effective permissions", relevant wenn Mask gesetzt ist).                       |
| <b>-s</b>            | Stripschutz: Zeigt nur die tatsächlichen ACL-Einträge, ohne die Grundeinträge (user/group/other).                                                            |
| <b>-d</b>            | Zeigt die <i>Default-ACL</i> eines Verzeichnisses (die Vorlage, die an neu darin erzeugte Dateien vererbt wird). In der Ausgabe mit default: gekennzeichnet. |
| <b>--omit-header</b> | Unterdrückt den Kommentar-Kopf in der Ausgabe, der Dateinamen und Hinweise beinhaltet.                                                                       |
| <b>-R</b>            | Rekursiv über Verzeichnisse, getfacl für jede Datei auflisten.                                                                                               |

### Ausgabe-Beispiel:

```
$ getfacl projekt/
file: projekt/
owner: alice
group: developers
user::rwx
user:bob:rwx # bob hat rw Rechte zusätzlich
group::r-x
mask::rw- # effektive Maske für ACL = rw
other::---
default:user::rwx # Default ACL für neu angelegte Objekte
...
```

Hier sieht man die Standard-Owner/Group-Einträge und eine explizite ACL, die User bob Schreibrechte gibt, sowie eine *mask*, die die maximal wirksamen Rechte auf rw beschränkt.

### setfacl (Set File ACL)

**Beschreibung:** Setzt oder ändert Access Control Lists (ACLs) von Dateien und Verzeichnissen file-4fdk7rvx39azjretinarqb. Damit kann man zusätzliche Berechtigungen für bestimmte Nutzer oder Gruppen vergeben (oder entziehen), die über die klassischen Besitzer-Gruppe-Andere Rechte hinausgehen.

### Syntax:

`setfacl [Optionen] { -m|-x <ACL-Eintrag> ... | -b | -k } Datei...`

### Optionen:

#### Option Beschreibung

- m** *Modify*: Fügt einen ACL-Eintrag hinzu oder aktualisiert ihn. Beispiel: `setfacl -m u:bob:rw file` gibt Benutzer *bob* rw-Rechte auf *file*. Mehrere -m Einträge oder Einträge mit Komma getrennt sind möglich.
- x** *ACL* *Remove*: Entfernt einen spezifischen ACL-Eintrag. Z.B. `setfacl -x g:marketing file` entfernt den ACL-Eintrag für Gruppe "marketing".
- b** *Remove all* – Löscht alle ACL-Einträge (außer den Basis-Einträgen für Owner/Group/Other). Damit werden alle erweiterten Berechtigungen zurückgesetzt. (Datei hat dann nur noch die klassischen Unix-Rechte.)
- k** *Remove Default* – Entfernt die Default-ACL eines Verzeichnisses (vererbte ACL-Vorlage). Hat auf Dateien keine Wirkung.
- R** Rekursiv – wendet die ACL-Änderung auf alle Dateien im Verzeichnis (und Unterverzeichnisse) an. Nützlich um z.B. ganze Bäume eine neue ACL zu geben.
- d** Setzt Einträge als *Default-ACL* für Verzeichnisse. Benutzt in Kombination mit -m oder -x. Beispiel: `setfacl -m d:g:team:rwx projekt` setzt eine Default-ACL, so dass neue Dateien unter *projekt*/ der Gruppe *team* rw Rechte erben, und Verzeichnisse rwx.
- **mask** Berechnet und setzt die ACL-Maske basierend auf den aktuellen Gruppen-Einträgen neu. (Die Maske begrenzt die maximalen Rechte der ACL-Einträge außer Owner/other). Standardmäßig wird die Maske auto-anangepasst.
- n** *Don't recalc mask*: Verhindert, dass setfacl die ACL-Maske automatisch anpasst. Normalerweise soll man diese Option nicht verwenden, außer man setzt die Maske manuell mit -m m::....

**Beispiel:** `setfacl -m u:bob:rw,g:marketing:r-- report.txt` – gibt Benutzer *bob* Lese- und Schreibrecht und der Gruppe *marketing* nur Leserecht auf *report.txt*.

**Hinweis:** Nicht alle Dateisysteme unterstützen POSIX-ACL. Bei ext4, XFS etc. muss i.d.R. das Volume mit Option `acl` eingehängt sein (moderne Systeme haben das meist per Default). Mit `getfacl` kann man prüfen, ob vorhandene ACLs existieren.

### Datei- und Verzeichnisverwaltung

#### stat (Dateistatus anzeigen)

**Beschreibung:** Zeigt detaillierte Meta-Informationen zu Dateien oder Dateisystemen an.  
stat gibt z.B. Größe, Blocks, Zeitstempel (Zugriff, Änderung, Statusänderung),  
Berechtigungen in Zahlenform, Device-ID, Inode-Nummer usw. in strukturierter Form aus.

**Syntax:**

stat [Optionen] <Datei>...

**Optionen:**

| Option                             | Beschreibung                                                                                                                                                                                                                                                                                                                             |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-f, --file-system</b>           | Zeigt anstelle der Datei-Details die Informationen zum Dateisystem, auf dem die Datei liegt. Z.B. Statistiken wie Blockgröße, Gesamtblöcke, freie Blöcke, FS-Typ etc.                                                                                                                                                                    |
| <b>-c FORMAT , --format=FORMAT</b> | Gibt die Ausgabe gemäß <i>FORMAT</i> formatiert aus. Man kann Platzhalter verwenden, um bestimmte Werte auszugeben. Z.B. stat -c "%A %h %U %G %s %n" datei gibt in einer Zeile: modestr, Linkanzahl, Owner, Group, Size, Name. (Siehe man stat für alle Platzhalter wie %A = Rechte in rwx, %a = oktal, %U = Ownername, %s = Größe etc.) |
| <b>--printf=FORMAT</b>             | Ähnlich wie --format, jedoch ohne automatisch Zeilenumbruch am Ende. Kann benutzt werden, um mehrere stat-Aufrufe zu skripten.                                                                                                                                                                                                           |
| <b>-t, --terse</b>                 | "Knapp": Gibt die Ausgabe in einer kompakten einzeiligen Form, Felder durch Leerzeichen getrennt, aus (für einfache maschinelle Weiterverarbeitung).                                                                                                                                                                                     |
| <b>--help, --version</b>           | Anzeigen von Hilfe bzw. Version.                                                                                                                                                                                                                                                                                                         |

*Beispieldaten:*

```
$ stat /etc/hosts
File: /etc/hosts
Size: 258 Blocks: 8 IO Block: 4096 regular file
Device: 802h/2050d Inode: 13422506 Links: 1
Access: (0644/-rw-r--r--) Uid: (0/ root) Gid: (0/ root)
Access: 2025-04-01 10:22:13.000000000 +0200
Modify: 2025-04-01 10:22:13.000000000 +0200
Change: 2025-04-01 10:22:15.000000000 +0200
```

Birth: -

Dies zeigt z.B., dass /etc/hosts 258 Bytes groß ist, auf einem bestimmten Device, inode etc., Rechte 0644, Eigentümer root, Zeitstempel für Zugriffs-/Änderungs-/Änderungszeit (Birth ist Erstellung, nicht unterstützt).

### **file (Dateityp bestimmen)**

**Beschreibung:** Untersucht eine Datei und versucht den Typ bzw. Format der Datei zu bestimmen. file nutzt einen umfangreichen Magie-Pattern-Katalog (/usr/share/file/magic), um anhand der Bytes einer Datei (Magic Numbers, Struktur etc.) menschlich lesbare Dateitypinformationen auszugeben. Praktisch, um z.B. Binärformate, Skripttypen oder Kodierungen zu erkennen.

#### **Syntax:**

file [Optionen] <Datei>...

#### **Optionen:**

##### **Option Beschreibung**

- M** Zeigt den MIME-Typ der Datei anstatt einer Beschreibung. Z.B. text/plain;
- i** charset=utf-8 statt "UTF-8 Unicode text". Hilfreich für Scripting oder Web-Kontext.
- b** *Brief* – gibt nur die Beschreibung aus, ohne den Dateinamen voranzustellen.
- L** (By default schreibt file: "<Dateiname>: <Beschreibung>"). Mit -b erhält man nur "<Beschreibung>".
- F** Folgt symbolischen Links und untersucht das Ziel statt den Link selbst.
- L** Standardmäßig wird der Link selbst untersucht und meist als "symbolic link to <target>" ausgegeben.
- h** (alias --no-dereference) Bewirkt das Gegenteil: behandelt Symlinks wie normale Dateien (gleichbedeutend mit Standardverhalten, -h ist nützlich falls -L als default kompiliert wäre).
- s** Untersucht auch Gerätedateien (block/char special) und Auslagerungsdateien. Standardmäßig werden einige spezielle Dateitypen nicht inhaltlich analysiert (um z.B. nicht eine Device-Datei zu lesen). -s kann riskant sein (kann z.B. /dev/zero "lesen" -> blockiert).
- z** Durchsucht komprimierte Dateien (gzip-komprimierte) – versucht innerhalb eines .gz die Dateisignatur zu prüfen.
- E** Gibt (Exit)Status 1 zurück falls *irgendeine* der Dateien kein bekannter Typ bestimmt werden konnte. (Nützlich in Skripten).

## Option Beschreibung

**--help**, Hilfe bzw. Versionsnummer und unterstützte Magic-Datei-Version ausgeben.  
**-v**

Beispiele:

- file /bin/ls -> **ELF 64-bit LSB executable**, x86-64, dynamically linked ...
- file archive.tar.gz -> **gzip compressed data**, was probably a tar archive (reads inside and erkennt es als tar in gz).
- file picture.jpg -> **JPEG image data**, JFIF standard 1.01, ...

## ln (Link erstellen)

**Beschreibung:** Erstellt Verknüpfungen zu Dateien – entweder **Hard Links** oder **Symbolische Links**. Ein Hardlink ist ein weiterer Directory-Eintrag für dieselbe Inode (nur innerhalb desselben Dateisystems möglich). Ein Symbolischer Link (Symlink) ist eine kleine Spezialdatei, die auf einen Pfad verweist.

**Syntax:**

ln [Optionen] <Ziel> [Linkname]

Standard ohne Optionen erstellt *Hardlink* mit Namen *Linkname*, der auf Datei *Ziel* zeigt. Wenn *Linkname* nicht angegeben, wird im aktuellen Verzeichnis ein Link gleichen Namens wie *Ziel* erstellt. Für Symlinks Option -s nutzen.

**Optionen:**

| Option                      | Beschreibung                                                                                                                                                                                                                                                                                    |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-s, --symbolic</b>       | Erstellt einen <i>symbolischen Link</i> statt eines Hardlinks. Der Link verweist auf den angegebenen <Ziel>-Pfad. Wenn <Ziel> kein absoluter Pfad ist, wird er relativ vom Link aus interpretiert. Beispiel: ln -s /etc/hosts hostslink erzeugt Symlink <i>hostslink</i> -> <i>/etc/hosts</i> . |
| <b>-f, --force</b>          | Existierende Zielfile (Linkname) ggf. entfernen, falls schon vorhanden. Ohne -f würde ln bei existierendem Linknamen einen Fehler ausgeben. (Bei -s mit -f: überschreibt existierenden Link.)                                                                                                   |
| <b>-n, --no-dereference</b> | Beim Überschreiben eines existierenden symbolischen Links (mit -f) behandle den Link selbst statt dessen Ziel. (Verhindert, dass versehentlich die Datei, auf die ein bestehender Link zeigt, gelöscht wird).                                                                                   |
| <b>-v, --verbose</b>        | Gibt eine Meldung pro verarbeitetem Link aus.                                                                                                                                                                                                                                                   |

| <b>Option</b>                    | <b>Beschreibung</b>                                                                                                                                                                                 |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-T, --no-target-directory</b> | Behandelt das letzte Argument immer als <i>genauen Linknamen</i> , selbst wenn es ein existierendes Verzeichnis ist. (Verhindert, dass ln im Zielverzeichnis einen Link mit Standardname erstellt.) |
| <b>-d, -F, --directory</b>       | Erlaubt es root, auch <i>Verzeichnis-Hardlinks</i> zu erstellen.<br>(Normalerweise verboten – Option nur auf speziellen Systemen mit -f kombinierbar, in der Praxis kaum verwendet).                |
| <b>-b, --backup</b>              | Erzeugt ein Backup der Zielfile falls -f zum Überschreiben genutzt wird. (Nützlich, um das ersetzte File nicht zu verlieren.) Siehe auch --backup=t etc.                                            |
| <b>-S &lt;Suffix&gt;</b>         | Gibt ein Suffix für Backup-Dateien an (Standard ~). Bsp: -b -S .bak.                                                                                                                                |

*Beispiele:*

- Hardlink: ln /path/to/original.txt copy.txt – copy.txt ist nun ein weiterer Verweis auf original.txt (gleiche Inode). Änderungen an einer erscheinen in der anderen, da es dieselbe Datei ist. Löscht man eine, bleibt die andere erhalten (Datei wird erst gelöscht, wenn letzter Link entfernt).
- Symlink: ln -s /etc/apache2 siteconfig – erstellt siteconfig -> /etc/apache2. Aufruf von ls -l siteconfig zeigt dann siteconfig -> /etc/apache2.

### **cd (Change Directory) – siehe oben bei Grundlegende Befehle**

*(Im Kontext Datei/Verzeichnis-Verwaltung sei auf den bereits beschriebenen Befehl cd verwiesen, um Verzeichnisse zu wechseln.)*

### **tree (Verzeichnisbaum anzeigen)**

**Beschreibung:** Zeigt die Verzeichnisstruktur grafisch in Form eines Baumes an. tree listet rekursiv alle Unterverzeichnisse und Dateien in hierarchischer eingerückter Darstellung. Es erleichtert das Visualisieren der Verzeichnis-Hierarchie.

#### **Syntax:**

tree [Optionen] [Verzeichnis]

#### **Optionen:**

| <b>Option</b> | <b>Beschreibung</b>                                                                                                                                       |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-a</b>     | Zeigt <i>alle</i> Dateien an, inkl. versteckter (auch solche, die mit . beginnen). Standardmäßig ignoriert tree versteckte Dateien, außer . und .. immer. |

| <b>Option</b>               | <b>Beschreibung</b>                                                                                                                                               |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-d</b>                   | Listet <i>nur Verzeichnisse</i> auf, keine Dateien. Nützlich, um reinen Verzeichnisbaum zu sehen.                                                                 |
| <b>-L Level</b>             | Beschränkt die Tiefe der Ausgabe auf <i>Level</i> Ebenen. Z.B. tree -L 2 zeigt nur zwei Ebenen tief (akt. Verzeichnis + Unterverzeichnisse, aber keine tieferen). |
| <b>-f</b>                   | Zeigt den vollen Pfad vor jedem Eintrag (statt nur Namen relativ).                                                                                                |
| <b>-I</b><br><i>PATTERN</i> | Ignoriert Einträge, die auf das Muster passen (Groß/Kleinschreibung berücksichtigt, mehrere Patterns mit `                                                        |
| <b>-P</b><br><i>PATTERN</i> | Zeigt nur Einträge an, die dem Pattern entsprechen (Whitelist). Umgekehrt zu -I.                                                                                  |
| <b>-F</b>                   | Markiert Verzeichnisse mit abschließendem /, ausführbare Dateien mit *, Symlinks mit @, Pipes mit `                                                               |
| <b>-p</b>                   | Zeigt UNIX-Dateirechte (Permission bits) vor jedem Eintrag.                                                                                                       |
| <b>-u / -g</b>              | Zeigt Benutzer bzw. Gruppenbesitzer an. (Wie ls -l Infos).                                                                                                        |
| <b>-h</b>                   | <i>Human-readable size:</i> Zeigt Dateigrößen in menschenlesbarem Format (K, M, G) anstatt in Bytes (wirkt mit -s).                                               |
| <b>-s</b>                   | Zeigt Dateigrößen in Bytes neben jedem Dateinamen an.                                                                                                             |
| <b>--du</b>                 | Summiert Verzeichnisgrößen wie du (statt wie normal lediglich die Einzeldateigröße zu zeigen).                                                                    |
| <b>-t</b>                   | Sortiert die Ausgabe nach Zeit (Neueste zuerst) statt alphabetisch.                                                                                               |
| <b>-o Datei</b>             | Gibt die Ausgabe in die angegebene Datei (überschreibt) anstatt auf stdout.                                                                                       |
| <b>-n</b>                   | Keine farbliche Hervorhebung (Standard ist bunt, -n erzwingt Monochrom).                                                                                          |
| <b>--help</b>               | Hilfe anzeigen (Übersicht der Optionen).                                                                                                                          |

*Beispiel:*

```
$ tree -L 2 /etc
```

```
/etc
```

```
| └── apache2
```

```
| └── apache2.conf
```

```
| └── envvars
```

```

| └─ mods-available
|
├── cron.daily
|
| └─ apt-compat
|
└─ man-db
|
└─ default
 ├── cron
 └─ rcs

```

6 directories, 5 files

Hier sieht man /etc mit zwei Ebenen Tiefgang.

### **mkdir (Make Directory)**

**Beschreibung:** Erstellt ein neues Verzeichnis (oder mehrere) im Dateisystem. Kann rekursiv auch mehrere Ebenen auf einmal anlegen.

**Syntax:**

`mkdir [Optionen] <Verzeichnisname>...`

**Optionen:**

| <b>Option</b>            | <b>Beschreibung</b>                                                                                                                                                                                                                                                                                               |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-p, --parents</b>     | Erzeugt auch alle notwendigen übergeordneten Verzeichnisse, falls sie nicht existieren. Somit werden keine Fehler ausgegeben, wenn Verzeichnisse in der Pfadangabe fehlen. Beispiel: <code>mkdir -p /tmp/foo/bar</code> legt sowohl <code>/tmp/foo</code> (falls nicht da) als auch <code>/tmp/foo/bar</code> an. |
| <b>-m, --mode=MODE</b>   | Setzt die Zugriffsrechte des neu erstellten Verzeichnisses auf <i>MODE</i> (numerisch oder symbolisch) anstatt der Standardrechte (die Standardrechte sind i.d.R. 0777 abzüglich Umask). Z.B. <code>mkdir -m 750 projDir</code> erstellt <i>projDir</i> mit rwx für Besitzer, rx für Gruppe, nichts für andere.   |
| <b>-v, --verbose</b>     | Ausgabe einer Meldung für jedes angelegte Verzeichnis (zeigt Pfad an).                                                                                                                                                                                                                                            |
| <b>--help, --version</b> | Anzeige von Hilfe bzw. Version.                                                                                                                                                                                                                                                                                   |

### **cp (Copy File)**

**Beschreibung:** Kopiert Dateien oder Verzeichnisse. Bei Angabe mehrerer Quelldateien muss das letzte Argument ein Verzeichnis sein (Zielverzeichnis). cp kann auch Verzeichnisbäume rekursiv kopieren.

**Syntax:**

cp [Optionen] <Quelle>... <Ziel>

*Wenn <Ziel> ein Verzeichnis ist, werden alle <Quelle>-Dateien dorthin kopiert (mit gleichem Namen). Ist nur eine Quelle und <Ziel> kein Verzeichnis, wird die Datei dorthin kopiert/umbenannt.*

**Optionen:**

| Option                     | Beschreibung                                                                                                                                                                                                                                                                                           |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-r, -R, --recursive</b> | Kopiert Verzeichnisse rekursiv (Inhalt und Unterverzeichnisse). Ohne diese Option kopiert cp standardmäßig keine Verzeichnisse. (GNU cp hat auch -r/-R – beide meist gleichwertig. POSIX fordert -R.)                                                                                                  |
| <b>-a, --archive</b>       | Archivmodus: Kopiert rekursiv und bewahrt dabei möglichst alle Attribute. Entspricht -dR --preserve=all (also -R, und -p für timestamps, owner, group, perms, ACLs, xattrs, Links etc.). Dies ist für Vollbackups ideal <a href="#">ionos.at</a> .                                                     |
| <b>-p, --preserve</b>      | Bewahrt Eigentümer, Gruppe, Berechtigungen und Zeitstempel der Datei beim Kopieren <a href="#">ionos.at</a> . (Ohne -p übernimmt cp Standard: setzt aktuellen Zeitstempel "jetzt" und erstellt als aktueller Benutzer). Kann Argumente haben wie --preserve=mode,ownership,timestamps,links,xattr,all. |
| <b>-f, --force</b>         | Überschreibt vorhandene Ziel-Dateien ohne Nachfrage, und versucht auch nicht-schreibbare Ziele zu entfernen, um Kopie dennoch zu platzieren. (Im Konflikt mit -n gilt letzter Schalter.)                                                                                                               |
| <b>-i, --interactive</b>   | Fragt nach, bevor eine existierende Zielfile überschrieben wird (Confirm Y/N). Gibt dem Benutzer die Möglichkeit, Überschreiben abzulehnen.                                                                                                                                                            |
| <b>-n, --no-clobber</b>    | Überschreibt <i>keine</i> existierenden Dateien. D.h. falls Ziel existiert, wird die Kopie für diese Datei ausgelassen. (Verhindert unabsichtliches Überschreiben)                                                                                                                                     |

| Option                                       | Beschreibung                                                                                                                                                                                                                                                                                               |
|----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-u, --update</b>                          | Kopiert nur, wenn <i>Quelle</i> neuer als <i>Ziel</i> ist, oder Ziel noch nicht existiert. Praktisch für inkrementelle Updates. Existiert Ziel mit gleich/neueren Datum, wird es nicht überschrieben.                                                                                                      |
| <b>-d, --no-dereference</b>                  | Kopiert <i>Symbolic Links</i> als Links (anstatt deren referenzierten Inhalt). Standard: cp folgt symbolischen Links und kopiert die Datei dahinter. Mit -d wird der Link selbst neu erstellt am Ziel (Hardlinks bleiben hardlinks sofern möglich; -d ist in GNU cp gleichbedeutend mit --preserve=links). |
| <b>-L, --dereference</b>                     | Folgt allen Symlinks: kopiert stets den referenzierten Inhalt (Standardverhalten bei regulärem cp für Dateien). Bei Angabe von Verzeichnis mit -R: -L würde bewirken, dass cp alle gefundene Symlinks auflöst und deren Ziel kopiert (kann zu Loops führen, daher mit Vorsicht).                           |
| <b>-P, --no-dereference --preserve=links</b> | Kopiert Links als Links (das ist Standard mit -a enthalten). (Im GNU cp -P bedeutet No-dereference; beachten: -d und -P ähnlich).                                                                                                                                                                          |
| <b>--parents</b>                             | Erhält die Quellpfadstruktur im Ziel. Bsp: cp --parents dir1/dir2/file.txt /backup legt /backup/dir1/dir2/file.txt an <a href="#">man7.org</a> . Nur sinnvoll, wenn Ziel ein Verzeichnis ist.                                                                                                              |
| <b>-l, --link</b>                            | Erstellt <i>Hardlinks</i> statt Kopien der Datei, sofern im selben Dateisystem. (Keine Datenkopie, nur neuer Name, ähnlich ln).                                                                                                                                                                            |
| <b>-s, --symbolic-link</b>                   | Erstellt <i>Symlinks</i> statt die Daten zu kopieren. (Zieldatei wird symbolischer Verweis auf Quelldatei).                                                                                                                                                                                                |
| <b>-H, -L, -P (bei -R)</b>                   | Steuerung des Symlink-Verhaltens <b>während</b> rekursivem Kopieren: -P (Default) ignoriert Linkinhalte (kopiert Symlinks als Symlinks), -L folgt Symlinks zu Dateien, -H folgt nur Symlinks angegeben auf Befehlzeile. (-a impliziert -P).                                                                |
| <b>-- backup[=Control]</b>                   | Erzeugt Backup von Zieldateien, die überschrieben würden. (Etwa file.txt~ anlegen). Kann =simple, numbered etc. steuern. Siehe auch VERSION_CONTROL Env.                                                                                                                                                   |
| <b>--help, --version</b>                     | Hilfe/Version anzeigen.                                                                                                                                                                                                                                                                                    |

## mv (Move/Rename)

**Beschreibung:** Verschiebt Dateien oder Verzeichnisse an einen neuen Pfad, bzw. benennt sie um. mv kann auch zum einfachen Umbenennen einer Datei verwendet werden (Quell- und Zielpfad auf selbem FS = schneller Rename, kein Kopieren). Falls Quelle und Ziel auf unterschiedlichen Dateisystemen liegen, arbeitet mv durch Kopieren und anschließendem Löschen der Quelle.

**Syntax:**

`mv [Optionen] <Quelle>... <Ziel>`

**Optionen:**

| Option                               | Beschreibung                                                                                                                                                                                                            |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-i, --interactive</b>             | Fragt nach Bestätigung, falls eine Zielfile durch den Move überschrieben würde (falls im Zielpfad eine Datei gleichen Namens existiert).                                                                                |
| <b>-f, --force</b>                   | Erzwingt das Überschreiben ohne Nachfrage. Ziel-Dateien werden ohne Warnung ersetzt. (Standardverhalten ohne -i).                                                                                                       |
| <b>-n, --no-clobber</b>              | Überschreibt keine bestehenden Dateien. Bei Namenskollision bricht mv für diese Datei ab (gibt keinen Fehler, exit code trotzdem 0). In GNU mv hat -n Vorrang vor -i falls beide angegeben.                             |
| <b>-u, --update</b>                  | Bewegt/überschreibt nur, wenn die Quelle neuer ist als die vorhandene Zielfile <i>oder</i> die Zielfile nicht existiert. (So werden neuere Versionen moved, ältere Quellen, die schon am Ziel sind, bleiben unberührt.) |
| <b>-v, --verbose</b>                 | Gibt aus, welche Datei wohin verschoben/umbenannt wurde (für jede Operation eine Zeile).                                                                                                                                |
| <b>--</b><br><b>backup[=CONTROL]</b> | Wie bei cp: legt Backup der Ziel-Datei an, falls existiert, bevor sie überschrieben wird. CONTROL kann none, numbered, existing, simple sein (siehe Doku oder ENV VERSION_CONTROL).                                     |
| <b>--strip-trailing-slashes</b>      | Entfernt evtl. nachgestellte Schrägstriche bei den Pfadangaben der Quellen (um Verwirrung mit Verzeichnissen zu vermeiden). Standard: neuere mv machen das ohnehin.                                                     |
| <b>--help, --version</b>             | Hilfe/Version.                                                                                                                                                                                                          |

**Hinweis:** mv kann mehrere Dateien in ein Zielverzeichnis verschieben oder eine einzelne Datei/Ordner umbenennen. Bei Verzeichnissen rekursiv, aber mv muss dafür nicht extra Option haben (im Gegensatz zu cp), da der Move innerhalb desselben FS nur Metadaten

ändert. Bei verschiedenen FS kopiert mv im Hintergrund (rename()) Syscall schlägt dann fehl, worauf cp+unlink passiert).

## rmdir (Remove Directory)

**Beschreibung:** Löscht leere Verzeichnisse. Im Gegensatz zu rm entfernt rmdir nur Verzeichnisse und auch nur, wenn sie keine Dateien/Unterverzeichnisse mehr enthalten. Es kann optional ganze verschachtelte Verzeichnisbäume entfernen, sofern diese leer sind (Eltern werden dann rekursiv entfernt).

### Syntax:

rmdir [Optionen] <Verzeichnis>...

### Optionen:

| Option                            | Beschreibung                                                                                                                                                                                                                                                                                                  |
|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-p, --parents</b>              | Entfernt das angegebene Verzeichnis und ggf. dessen übergeordnete Verzeichnisse, falls diese nach jedem Entfernen leer werden. Beispiel: rmdir -p projekt/build/output versucht erst output zu löschen. Ist danach build leer, wird auch build gelöscht. Ist danach projekt leer, wird auch projekt gelöscht. |
| <b>-v, --verbose</b>              | Ausgabe einer Meldung für jedes gelöschte Verzeichnis.                                                                                                                                                                                                                                                        |
| <b>--ignore-fail-on-non-empty</b> | Unterdrückt die normale Fehlermeldung, falls ein Verzeichnis nicht leer ist. (Das Verzeichnis wird nicht gelöscht, aber es kommt kein Fehlerstatus; mit Vorsicht verwenden).                                                                                                                                  |

**Nützlich:** Um verschachtelte leere Ordner zu entfernen. Aber um Verzeichnisse mit Inhalt zu löschen, nutzt man rm -r.

## rm (Remove Files/Directories)

**Beschreibung:** Löscht Dateien oder Verzeichnisse. Standardmäßig entfernt rm nur Dateien; mit -r kann es auch Verzeichnisbäume löschen (inkl. aller enthaltenen Dateien). **Vorsicht:** Gelöschte Dateien werden nicht ohne Weiteres wiederherstellbar gespeichert (kein Papierkorb, außer man nutzt safe-rm Tools).

### Syntax:

rm [Optionen] <Datei/Pfad>...

### Optionen:

| Option                     | Beschreibung                                                                                                                                                                                                                                                                                                                                  |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-r, -R, --recursive</b> | Löscht Verzeichnisse <i>rekursiv</i> mitsamt ihrem gesamten Inhalt. Damit können komplett Verzeichnisbäume entfernt werden. Ohne -r verweigert rm das Löschen von Verzeichnissen mit Fehler "is a directory".                                                                                                                                 |
| <b>-d</b>                  | Löscht ein <i>leeres</i> Verzeichnis. (Ohne -r löscht rm normalerweise gar keine Verzeichnisse; -d erlaubt es aber für leere, ähnlich rmdir. Nützlich, aber -r deckt diesen Fall auch ab.)                                                                                                                                                    |
| <b>-f, --force</b>         | <i>Force/ignore nonexistent:</i> Unterdrückt Rückfragen und Fehlermeldungen. Existiert eine angegebene Datei nicht, wird es ignoriert. Schreibschutz wird ignoriert (d.h. es fragt nicht nach Bestätigung bei schreibgeschützten Dateien). Kombiniert mit -r sehr gefährlich, da es kompromisslos alles löscht. Wird oft in Skripten genutzt. |
| <b>-i</b>                  | Interaktiv nachfragen vor <i>jedem</i> Löschvorgang. (Für viele Dateien unpraktisch).                                                                                                                                                                                                                                                         |
| <b>-I</b>                  | "ein wenig interaktiv": Fragt nur einmal pauschal nach, <b>wenn</b> mehr als 3 Dateien gelöscht werden sollen <i>oder</i> ein rekursiver Lösch auf einem Verzeichnis durchgeführt wird. (GNU rm Feature als Sicherheitsnetz)file-5y1edouysrow4zj6phjcqm.                                                                                      |
| <b>--preserve-root</b>     | Schützt vor rm -rf /. Diese Option ist in GNU rm standardmäßig aktiv – es verweigert das komplette Löschen des Root-FS. Man kann mit --no-preserve-root überschreiben (nicht empfohlen).                                                                                                                                                      |
| <b>-v, --verbose</b>       | Zeigt für jede Datei eine Meldung an, dass sie gelöscht wurde.                                                                                                                                                                                                                                                                                |
| <b>--help, --version</b>   | Hilfe bzw. Version.                                                                                                                                                                                                                                                                                                                           |

Anwendung:

- **Einzelne Datei löschen:** rm datei.txt.
- **Mehrere Dateien löschen:** rm \*.o (löscht alle .o Dateien im aktuellen Verzeichnis).
- **Verzeichnis und Inhalt löschen:** rm -r docs/ – löscht Ordner docs mitsamt allen Unterdateien.
- **Ohne Rückfrage alles löschen:** rm -rf tempdir. (Vorsicht!).

**head (Dateianfang anzeigen)**

**Beschreibung:** Gibt die ersten Zeilen einer Datei aus (standardmäßig 10 Zeilen). head ist nützlich, um schnell die ersten Zeilen (z.B. Header einer CSV, Beginn eines Logs etc.) anzuschauen, ohne die ganze Datei zu öffnen. Es kann auch binäre Dateien auf Anfang bytes prüfen.

**Syntax:**

head [Optionen] [Datei]...

Ohne Datei liest head von der Standardeingabe (z.B. in Pipes). Bei mehreren Dateien wird jeweils ein Header mit dem Dateinamen ausgegeben.

**Optionen:**

| Option                                   | Beschreibung                                                                                                                                                                                                                                                |
|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-n <i>K</i>, --lines=<i>K</i></b>     | Gibt die ersten <i>K</i> Zeilen aus. <i>K</i> kann eine Zahl sein oder -K für alle bis auf die letzten <i>K</i> Zeilen (negativer Wert). Standard ist 10. Beispiel: head -n 20 file zeigt 20 Zeilen, head -n -5 file zeigt alle außer der letzten 5 Zeilen. |
| <b>-c <i>Bytes</i>, --bytes=<i>B</i></b> | Gibt die ersten <i>B</i> Bytes aus (statt Zeilen). Kann auch mit K, M, etc. Suffix (Potenzen von 1024) angegeben werden. Z.B. head -c 100K bigfile. Negative Werte analog: -c -10 alles außer letzte 10 Bytes.                                              |
| <b>-q, --quiet/--silent</b>              | Unterdrückt die Überschriften, die vor dem Inhalt stehen, wenn mehrere Dateien angegeben wurden. Normal zeigt head sowas wie "=> filename <==" vor dem Block jeder Datei. Mit -q wird alles aneinandergehängt ohne solche Banner.                           |
| <b>-v, --verbose</b>                     | Zeigt immer die Dateiüberschriften an, auch wenn nur eine Datei angegeben ist. (Normalerweise bei nur einer Datei keine Überschrift).                                                                                                                       |
| <b>--help, --version</b>                 | Hilfe bzw. Version ausgeben.                                                                                                                                                                                                                                |

*Beispiel:* head -n 3 /etc/passwd zeigt die ersten 3 Zeilen der passwd-Datei (typisch Root, daemon, bin Accounts).

**tail (Dateiende anzeigen)**

**Beschreibung:** Gibt die letzten Zeilen einer Datei aus (standardmäßig 10 Zeilen). Sehr nützlich, um z.B. das Ende von Logfiles zu sehen. Kann im *Follow-Modus* (-f) genutzt werden, um fortlaufend neuen Inhalt anzuzeigen (Echtzeit-Log-Überwachung).

**Syntax:**

tail [Optionen] [Datei]...

**Optionen:**

| Option                               | Beschreibung                                                                                                                                                                                                                   |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-n K , --lines=K</b>              | Zeigt die letzten <i>K</i> Zeilen an. Negative Werte bedeuten ab einer bestimmten Zeile von Anfang an (ab Zeile <i>K</i> ). Standard ist 10. Z.B. tail -n 50 /var/log/syslog zeigt die letzten 50 Zeilen.                      |
| <b>-c B , *--bytes=B</b>             | Zeigt die letzten <i>B</i> Bytes an. Kann auch negative (ab Byte <i>X</i> ) sein.                                                                                                                                              |
| <b>-f, --follow</b>                  | Bleibt nach dem Anzeigen der letzten Zeilen geöffnet und zeigt fortlaufend neue Zeilen an, falls die Datei wächst (wie tail -f logfile zum Mitlesen).                                                                          |
| <b>-F</b>                            | Ähnlich wie -f, aber zusätzlich "Wiederöffnen", falls Datei rotiert/neu erstellt wurde. (Follow Name statt Descriptor; praktisch für Logrotation – tail -F /var/log/app.log folgt auch nach Neustart des Logs dem neuen File). |
| <b>-s Sek , --sleep-interval=Sek</b> | Im Follow-Modus: Warte <i>Sek</i> Sekunden zwischen den Überprüfungen auf neue Daten. Standard 1.0 Sekunde. Kann z.B. auf 0.1 verringert werden, um schneller zu pollen, oder erhöht, um Last zu reduzieren.                   |
| <b>--max-unchanged-stats=N</b>       | (Bei -F relevant) Wie oft in Folge unveränderte Inode geprüft wird, bevor tail davon ausgeht, dass ggf. Logrotate stattgefunden hat.                                                                                           |
| <b>-q, --quiet</b>                   | Unterdrückt Überschriften bei mehreren Dateien (analog head).                                                                                                                                                                  |
| <b>-v, --verbose</b>                 | Erzwingt Überschriften, selbst bei einer Datei.                                                                                                                                                                                |
| <b>--help, --version</b>             | Hilfe bzw. Version.                                                                                                                                                                                                            |

Anwendung:

- **Letzte Zeilen einer Datei:** tail -n 20 messages.log (Zeigt die letzten 20 Zeilen).
- **Live-Verfolgung:** tail -f /var/log/syslog (Druckt neu geschriebene Logzeilen fortlaufend). Abbruch mit Ctrl+C.
- **Combine mit grep:** z.B. tail -f server.log \| grep "ERROR" filtert live nur Zeilen mit "ERROR".

## Dateien finden und Text filtern

### grep (Global Regular Expression Print)

**Beschreibung:** Durchsucht die Eingabe (Dateien oder Standard Input) nach Zeilen, die einem bestimmten Muster entsprechen, und gibt diese Zeilen ausfile-4fdk7rvx39azjretinarqb. grep arbeitet mit regulären Ausdrücken (Standard Basic Regex,

erweiterte mit -E). Es ist das klassische Suchwerkzeug in Textstreams. Name stammt von der ed-Editor Anweisung g/re/p (global regex print).

#### Syntax:

```
grep [Optionen] 'Muster' [Datei...]
```

#### Optionen:

| Option                           | Beschreibung                                                                                                                                                                                                                                                           |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-i, --ignore-case</b>         | Ignoriert Groß-/Kleinschreibung beim Mustervergleichfile-4fdk7rvx39azjretinarqb. "Test" findet auch "test" oder "TEST".                                                                                                                                                |
| <b>-v, --invert-match</b>        | Gibt alle Zeilen <b>aus, die nicht</b> auf das Muster passen (Inverse Treffer)file-4fdk7rvx39azjretinarqb. Nutzt man oft zur Ausschlussfilterung.                                                                                                                      |
| <b>-r, -R, --recursive</b>       | Durchsucht rekursiv alle Dateien in angegebenen Verzeichnissen (und Unterverz.)file-4fdk7rvx39azjretinarqb. Ohne diese Option behandelt grep Verzeichnisargumente normalerweise als Fehler oder ignoriert sie. (GNU grep: -r folgt Symlinks nicht, -R folgt Symlinks.) |
| <b>-n, --line-number</b>         | Zeigt vor jeder gefundenen Zeile die Zeilennummer in der Datei an <a href="#">ibm.com</a> . Format: Datei:Nr:Zeilentext.                                                                                                                                               |
| <b>-A NUM</b>                    | (After) Zeigt <i>NUM</i> Zeilen <b>nach</b> jeder Trefferzeile zusätzlich an (Context) <a href="#">ibm.com</a> .                                                                                                                                                       |
| <b>-B NUM</b>                    | (Before) Zeigt <i>NUM</i> Zeilen <b>vor</b> jeder Trefferzeile an.                                                                                                                                                                                                     |
| <b>-C NUM</b>                    | (Context) Zeigt <i>NUM</i> Zeilen <i>vor und nach</i> jedem Treffer (gleichzeitig -A NUM -B NUM). Standard -C 2 beispielsweise gibt 2 Kontextzeilen um jede Fundstelle.                                                                                                |
| <b>-l, --files-with-matches</b>  | Listet nur die Dateinamen auf, die mindestens einen Treffer enthalten. Pro Datei ein Name, Inhalt wird nicht ausgegeben. (Nützlich um zu sehen wo etwas vorkommt)                                                                                                      |
| <b>-L, --files-without-match</b> | Listet nur Dateinamen, die <i>keinen</i> Treffer enthalten (Gegenteil von -l).                                                                                                                                                                                         |
| <b>-c, --count</b>               | Zählt die Treffer pro Datei und gibt nur diese Anzahl aus (zusammen mit Dateinamen, außer mit -h). D.h. "file.txt:3". In Kombination mit -v zählt es Nicht-Treffer-Zeilen.                                                                                             |

| Option                       | Beschreibung                                                                                                                                                                                                                                          |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-o, --only-matching</b>   | Gibt nur den Teil der Zeile aus, der dem Suchmuster entspricht, nicht die ganze Zeile. Bei mehreren Treffern pro Zeile werden diese getrennt ausgegeben.                                                                                              |
| <b>-q, --quiet/--silent</b>  | "Quiet": gibt keine Ausgabe; liefert lediglich Exit-Status 0 falls wenigstens ein Match gefunden wurde, 1 falls kein Match. Praktisch in Skripten (nur Rückgabewert interessiert).                                                                    |
| <b>-s, --no-messages</b>     | Unterdrückt Fehlermeldungen über nicht lesbare Dateien (z.B. wenn Zugriffsrechte fehlen oder Datei nicht existiert).                                                                                                                                  |
| <b>-E, --extended-regexp</b> | Interpretiert das Muster als <i>Extended Regular Expression (ERE)</i> <a href="#">ionos.at</a> . Das erlaubt z.B. Nutzung von +, ?, `                                                                                                                 |
| <b>-F, --fixed-strings</b>   | Interpretiert das Muster als <i>fixen String</i> (kein Regex, jedes Zeichen literal). Entspricht fgrep (die eine schnellere Suche ohne regex ermöglicht). Alle Muster (wenn -e mehrfach) werden als Literalstrings gesucht.                           |
| <b>-P, --perl-regexp</b>     | Interpretiert das Muster als Perl-Compatible Regex (PCRE), falls grep mit PCRE-Unterstützung kompiliert wurde. Damit stehen mächtigere Regex-Features zur Verfügung.                                                                                  |
| <b>-h, --no-filename</b>     | Unterdrückt die Ausgabe des Dateinamens vor jeder Trefferzeile (wenn mehrere Dateien durchsucht werden). Standardmäßig präfixet grep Datei: bei mehreren Input-Dateien. Mit -h erscheinen nur die Zeilen. (Bei Einzeldatei ist das sowieso Standard.) |
| <b>-H, --with-filename</b>   | Zeigt Dateinamen auch bei einziger Datei an. (Umgekehrt zu -h). Nützlich in Pipelines wo Source unbekannt.                                                                                                                                            |
| <b>--color[=WHEN]</b>        | Hebt das gefundene Pattern farblich hervor in der Ausgabe. WHEN kann auto (Standard, nur wenn Terminal), always oder never sein. Setzt Umgebungsvariable GREP_COLOR/GREP_COLORS ggf. benötigt.                                                        |
| <b>-e Pattern</b>            | Gibt das Suchmuster explizit an. Nötig z.B. wenn das Pattern mit - beginnt (damit es nicht als Option interpretiert wird). Kann mehrfach angegeben werden, um mehrere Muster (OR-Verknüpfung) zu suchen.                                              |
| <b>-f Datei</b>              | Liest Suchmuster aus angegebener Datei (ein Pattern pro Zeile). Nützlich bei sehr vielen oder komplexen Patterns.                                                                                                                                     |

| Option                    | Beschreibung                                                                                                                                                                                                                                  |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>--exclude=GLOB</b>     | Schließt Dateien vom Durchsuchen aus, deren Name auf GLOB (Wildcard-Muster) passt. Z.B. --exclude=*.min.js würde Minimierte JS ignorieren. (Nur wirksam bei rekursiver Suche oder wenn Verzeichnis als Input). Es gibt auch --include analog. |
| <b>--exclude-dir=GLOB</b> | Wie --exclude, aber nur auf Verzeichnisnamen angewandt, um ganze Verzeichnisse von der Rekursion auszunehmen (z.B. --exclude-dir=.git).                                                                                                       |
| <b>--help, --version</b>  | Hilfe / Version.                                                                                                                                                                                                                              |

**Exit-Status:** 0 falls mindestens ein Match gefunden, 1 falls kein Match, 2 falls ein Fehler auftrat (z.B. Datei nicht lesbar).

### Beispiele:

- grep -i "error" \*.log – sucht case-insensitive "error" in allen .log im aktuellen Ordner, gibt Trefferzeilen mit Dateipräfix aus.
- grep -R "TODO" src/ – findet rekursiv alle Vorkommen von "TODO" in Dateien unterhalb von src/ (typisch in Codebasis).
- grep -nE "^Error|Warning):" report.txt – sucht Zeilenbeginn "Error:" oder "Warning:" und gibt mit Zeilennummer aus [pq.hosting](#).
- ps ax | grep "[c]ron" – findet Prozesse mit "cron" im Namen. (Trick: [c] im Pattern matcht 'c', verhindert aber, dass der grep selbst im ps-Listing gefunden wird, da dessen Kommandozeile "grep cron" kein 'cron' substring hat sondern 'cro n' durch char class unterbrochen).

## find (Dateien suchen)

**Beschreibung:** Durchsucht Verzeichnisse nach Dateien/Verzeichnissen, die bestimmten Kriterien entsprechen (Name, Typ, Größe, Datum, Rechte, Eigentümer etc.) [file-4fdk7rvx39azjretinarqb](#). find ist sehr mächtig: man kann logische Bedingungen kombinieren und Aktionen auf gefundene Dateien ausführen (löschen, verschieben, Exec von Kommandos etc.). Es arbeitet rekursiv ab dem angegebenen Pfad.

### Syntax:

find [Start-Verzeichnisse] [Optionen/Ausdrücke] [Aktion(en)]

### Wichtige Suchkriterien (Ausdrücke):

- **-name Muster** – Dateiname entspricht Wildcards (Globbing wie bei Shell, z.B. "\*.txt"). Groß/Kleinschreibung beachten.
- **-iname Muster** – wie -name, aber case-insensitive.

- **-type c** – Dateityp ist c. Häufig: **f** für regular file, **d** für directory, **l** symlink, **b** blockdevice, **c** chardevice, **p** named pipe, **s** socket. Z.B. -type d nur Verzeichnisfile-4fdk7rvx39azjretinarqb.
- **-user name/uid** – Datei gehört dem Benutzer *name* (oder UID numerisch). Analog **-group name/gid**.
- **-uid +n/-n/n** – UID der Datei ist ( $>/</=$ ) n. -uid +500 UID größer 500. (Entspricht -user nur numerisch und mit Vergleichen). Genauso -gid.
- **-size N\*[bckMG]\*** – Dateigröße-Kriterium. *N* kann mit Vorzeichen angegeben werden (+ größer als, - kleiner als, nichts = genau). Suffix: standardmäßig in 512-Byte Blöcken, oder **c** = Bytes, **k** = KiB, **M** = MiB, **G** = GiB. Beispiel: -size +100M Dateien größer 100 MiB[wiki.foxtom.de](#), -size -10k kleiner 10 KiB.
- **-mtime Tage** – Letzte Änderung (Modify timestamp) liegt *Tage* zurück. *Tage* als ganze Zahl: +n = älter als n Tage, -n = jünger als n Tage, n = genau n Tage (im Bereich  $\pm 0.5d$ ). Ähnlich **-atime** (Access), **-ctime** (Statusänderung). Z.B. -mtime -7 in letzten 7 Tagen modifiziert[wiki.foxtom.de](#).
- **-newer Datei** – Datei wurde später geändert als *Datei* (Vergleich mtime). -**newermt Datum** (GNU find) erlaubt Vergleich mit einem Zeit-String, z.B. -newermt "2025-05-01".
- **-perm mode** – Dateirechte entsprechen *mode*. Mode kann exakt (z.B. 644) oder mit / oder - angegeben werden für Masken. Beispiel: -perm 644 findet genau 644. -perm -644 findet Dateien, die mindestens diese Bits haben (also rw- r-- r-- oder mehr Rechte). -perm /222 findet Dateien, bei denen irgendein Schreibbit gesetzt ist (Owner oder Group oder Other).
- **-maxdepth N** – Nur bis Tiefe N ins Verzeichnis hinab suchen (0= nur Startdir selbst). **-mindepth N** analog minimal tiefe. Z.B. find . -maxdepth 1 -type f nur Dateien im aktuellen Verzeichnis, nicht tiefer.
- **-mount** oder **-xdev** – Verlässt nicht das Dateisystem. D.h. keine Verzeichnisse auf anderen gemounteten FS durchsuchen (z.B. ignoriert find / -xdev dann andere Partitionen).
- **-prune** – Überspringt (schließt vom Durchsuchen aus) ein Verzeichnis. Bsp: find . -path "./node\_modules" -prune -o -name "\*.js" -print durchsucht alles außer den node\_modules Ordner.

#### Aktionen (wenn Suchkriterien wahr sind):

- **-print** – Gibt gefundene Pfade aus (Standard, wird angenommen, wenn keine andere Aktion).

- **-ls** – Führt ls -dils auf die gefundene Datei aus (anzeige von detaillierten Infos, inkl. Inode, Größe, Rechte etc.).
- **-delete** – Löscht gefundene Dateien *unmittelbar*. Achtung: find -delete entfernt auch Verzeichnisse, wenn leer geworden. Sollte meist mit Kriterien kombiniert werden (z.B. -type f -delete nur Dateien). **Keine** Aktion mehr nach -delete wird ausgeführt (beendet Durchlauf in diesem Pfad).
- **-exec Befehl {} ;** – Führt einen externen Befehl für jede gefundene Datei aus. {} wird durch den Pfad ersetzt. Die Kommandozeile wird so oft ausgeführt wie Treffer. Beispiel: find . -name "\*.bak" -exec rm {} \; löscht alle .bak-Dateien (funktional wie -delete, aber universeller)file-5y1edouysrow4zj6phjcqm. Das \; markiert Ende des -exec Befehls.
- **-exec ... {} +** – Variante, die versucht, mehrere gefundene Dateien in *einen* Aufruf zu packen (fügt die Pfade an Stelle von {} möglichst gesammelt, wie xargs). Effizienter, falls das Programm mehrere auf einmal verarbeiten kann. Z.B. -exec chmod 644 {} + ändert die Rechte aller gefundenen Files, ggf. in batched Calls.
- **-execdir** – Wie -exec, aber führt das Kommando im jeweiligen Verzeichnis der Datei aus (Working Directory wechselt). Erhöht Sicherheit bei {} mit {} containing spaces.

### **Beispiele:**

- find /var/log -type f -name "\*log" -size +1M -mtime +30 – findet Log-Dateien über 1MB, älter als 30 Tage unter /var/log.
- find . -maxdepth 1 -type d ! -name ":" -exec rm -r {} \; – löscht alle Unterverzeichnisse im aktuellen Ordner (maxdepth 1, alles was Type d ist und Name nicht ":"), dann rm -r).
- find /home/user -xtype l -delete – löscht gebrochene Symlinks (xtype l findet Links, deren Ziel nicht existiert).
- find . -perm /u+x -print – findet Dateien, bei denen das User-execute Bit gesetzt ist (also alle ausführbaren Dateien).
- find . -name ".\*tmp" -execdir gzip {} \; – packt alle \*.tmp im jeweiligen Verzeichnis in eine .tmp.gz (ausgeführt im Ort der Datei).

### **locate (Datenbank-Dateisuche)**

**Beschreibung:** Durchsucht eine vorgefertigte Index-Datenbank nach Dateinamen (Pfaden), die einem angegebenen Suchstring entsprechenfile-4fdk7rvx39azjretinarqb. locate ist viel schneller als find, da es nicht live das Dateisystem scannt, sondern auf einer regelmäßig aktualisierten Datenbank (typisch via updatedb in Cron) basiert. Nachteil: findet keine erst sehr kürzlich hinzugefügten Dateien (bis zum nächsten

updatedb-Lauf) und standardmäßig nur solche, auf die der User zugreifen darf.

### Syntax:

locate [Optionen] <Suchmuster>

Das Suchmuster ist im Grunde ein Substring oder einfache Glob. Standardmäßig sucht locate alle Datenbankeinträge, die <Suchmuster> als Teilstring enthalten.

### Optionen:

| Option                      | Beschreibung                                                                                                                                                                                                                              |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-i, --ignore-case</b>    | Sucht case-insensitive (Groß/Kleinschreibung ignorieren)file-4fdk7rvx39azjretinarqb.                                                                                                                                                      |
| <b>-r, --regex</b>          | Interpretiert das angegebene Muster als regulären Ausdruck (POSIX ERE) statt als einfachen Textfile-4fdk7rvx39azjretinarqb.                                                                                                               |
| <b>-w, --wholename</b>      | (Standardverhalten) Sucht im gesamten Pfadnamen. (Bei einigen Implementierungen war locate nur substring-suche, wholename explizit wie default, hier nur Info.)                                                                           |
| <b>-b, --basename</b>       | Sucht nur im Dateinamen selbst, nicht im gesamten Pfad. (Also ignoriert Verzeichnisnamen im Match.)                                                                                                                                       |
| <b>-c, --count</b>          | Zählt nur die Anzahl der Fundstellen und gibt diese Zahl aus, anstatt alle Ergebnisse zu listen.                                                                                                                                          |
| <b>-l N, --limit=N</b>      | Gibt maximal N Ergebnisse aus (dann Abbruch).                                                                                                                                                                                             |
| <b>--existing</b>           | Prüft gefundene Pfade und filtert jene heraus, die aktuell nicht (mehr) existieren. (Die DB könnte veraltete Einträge haben; mit --existing werden nur noch valide Treffer angezeigt, obwohl es Zeit kostet, jeden Pfad zu verifizieren.) |
| <b>-0, --null</b>           | Trennt die Ausgabe mit Nullbytes anstatt neuer Zeile. Nützlich, um in xargs -0 weiterzuverarbeiten.                                                                                                                                       |
| <b>-e, --nofollow</b>       | Ignoriert Einträge, die Symlinks auf nicht-existente Dateien sind (könnnte Implementation-spezifisch).                                                                                                                                    |
| <b>-S</b>                   | Zeigt Statistik der Datenbank (Anzahl Einträge etc.) statt zu suchen.                                                                                                                                                                     |
| <b>-d DB, --database=DB</b> | Nutzt eine alternative Datenbank-Datei anstelle der Standard-Datenbanken (/var/lib/mlocate.db z.B.). Man kann mehrere via -d db1:db2:... angeben.                                                                                         |

| <b>Option</b> | <b>Beschreibung</b> |
|---------------|---------------------|
|---------------|---------------------|

**--help, --version** Hilfe bzw. Version.

*Beispiel:* locate -i '\.pdf' listet alle Pfade, die ".pdf" (unabhängig von Großschreibung) enthalten – das würde i.d.R. alle PDF-Dateien auf dem System listen, die zum Zeitpunkt der letzten Datenbankaktualisierung existierten. Oft benutzt: locate <Programmname> um herauszufinden, wo ein Programm oder dessen Konfigdateien liegen.

### **updatedb (Update Database for locate)**

**Beschreibung:** Baut die Datei-Datenbank für locate neu auf. Es durchsucht das Dateisystem (oder definierte Pfade) und speichert alle Dateinamen/Pfade in einer komprimierten Datenbank (typischerweise `/var/lib/mlocate/mlocate.db`). updatedb wird meist regelmäßig (täglich) via Cron ausgeführt, man kann es aber auch manuell starten, z.B. nach größeren Änderungen im FS, damit locate sofort aktuell ist.

#### **Syntax:**

`updatedb [Optionen]`

**Optionen:** (Häufig werden diese über `/etc/updatedb.conf` voreingestellt)

| <b>Option</b> | <b>Beschreibung</b> |
|---------------|---------------------|
|---------------|---------------------|

**--prune-bind-mounts {yes|no}** Ob Bind-Mountpoints ausgelassen werden (Default ja – doppelte Scans vermeiden).

**--prune-paths='PFADLISTE'** Doppelpunkt-separierte Liste von Verzeichnissen, die *nicht* indexiert werden sollen file-4fdk7rvx39azjretinarqb. Standardmäßig z.B. `/tmp`, `/proc`, `/sys`, `/run`, `/dev`, `/var/cache` etc., um unnötiges zu sparen. Dies entspricht PRUNEPATHS in `/etc/updatedb.conf`.

**--prune-filesystems='TypList'** Liste von Dateisystemtypen, die nicht durchsucht werden (z.B. nfs,squashfs etc.). Standard meist Netzwerk-FS, Pseudo-FS (proc, sysfs) etc. file-4fdk7rvx39azjretinarqb. Entspricht PRUNEFS.

**-l N** **--localpaths=LIST** – falls man nur lokale Pfade indexieren will (veraltet, use prune).

**-U 'PFADLISTE'** Nur diese durch Doppelpunkt getrennten Pfade als Wurzel für Index verwenden (Standard "/" – gesamtes FS). Kann man nutzen um z.B. nur `/home` zu indexieren.

| Option               | Beschreibung                                                                                   |
|----------------------|------------------------------------------------------------------------------------------------|
| <b>-o DBFILE</b>     | Output: speichere DB in angegebener Datei statt Standard. (So kann man separate DBs erzeugen.) |
| <b>-f 'FS-Liste'</b> | Nur diese Dateisystemtypen berücksichtigen (Inverse von PRUNEFS).                              |
| <b>-v</b>            | Ausführliche Ausgabe während des Indizierens.                                                  |

In der Regel wird updatedb ohne Optionen (ggf. als root über sudo updatedb) ausgeführt und nutzt die System-Voreinstellungen. Danach steht die Datenbank locate zur Verfügung.

### which (Welches Kommando - Pfadsuche)

**Beschreibung:** Zeigt den Pfad zu einem ausführbaren Programm, das in der Umgebungsvariablen \$PATH gefunden wird file-4fdk7rvx39azjretinarqb. Damit kann man herausfinden, welches Programm aufgerufen würde, wenn man einen bestimmten Befehl tippt (nützlich bei mehreren Versionen in verschiedenen Verzeichnissen).

#### Syntax:

which [Optionen] Befehlsname...

#### Optionen:

| Option                                  | Beschreibung                                                                                                                                                                                                                 |
|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-a</b>                               | Zeigt alle Vorkommen des angegebenen Befehls im Suchpfad an (nicht nur das erste gefundene) file-4fdk7rvx39azjretinarqb. Z.B. which -a python listet evtl. /usr/bin/python und /usr/local/bin/python falls beide existieren. |
| <b>--skip-dot</b> , <b>--skip-tilde</b> | Ignoriert . bzw. ~ am Anfang des Pfades (also aktuelles Verzeichnis oder Home in PATH) in der Suche. (Sicherheitsaspekt – Standard bei which).                                                                               |
| <b>--show-dot</b> , <b>--show-tilde</b> | Zeigt explizit, wenn gefundene Einträge in . oder ~ liegen, durch Präfix "./" oder "~/".                                                                                                                                     |
| <b>--tty-only</b>                       | Zeigt nur etwas an, wenn stdout ein Terminal ist. Unterdrückt z.B. Ausgabe, wenn in Pipe (verhindert sich einzumischen).                                                                                                     |
| <b>--help</b> , <b>--version</b>        | Hilfe bzw. Versionsinfo (gilt bei GNU which; einige shells haben eingebautes which mit anderem Verhalten).                                                                                                                   |

*Hinweis:* which ist oft eine Shell-Funktion/Alias und verhält sich u.U. leicht unterschiedlich. Ähnliche Tools: type (shell-builtin, zeigt auch Aliases, Keywords), command -v.

**Beispiel:** which gcc -> /usr/bin/gcc (Pfad zum C-Compiler). which -a ls könnte z.B. zeigen /usr/bin/ls und /bin/ls falls beides symlinks aufeinander oder verschieden, etc.

## Textverarbeitung und Streams (Shell, Scripting)

### awk (Muster/Aktions-Scanner)

**Beschreibung:** AWK ist ein mächtiges Textverarbeitungswerkzeug und Skriptsprache, mit dem man Textdateien zeilenweise nach Mustern durchsuchen und pro Zeile Aktionen ausführen kannfile-4fdk7rvx39azjretinarqb. Typischerweise wird awk benutzt, um Spalten auszugeben oder zu filtern, da es Eingabezeilen automatisch in Felder zerlegt. Der Name stammt von den Anfangsbuchstaben seiner Schöpfer (Aho, Weinberger, Kernighan).

#### Syntax:

awk [Optionen] 'Muster { Aktion }' Datei...

Jede Zeile der Datei wird gelesen; wenn das *Muster* passt, wird die *Aktion* ausgeführt (z.B. etwas ausgeben). AWK hat eigene Variablen (\$1 = erstes Feld, \$0 = ganze Zeile) und eingebaute Rechen- und String-Features.

#### Optionen:

| Option            | Beschreibung                                                                                                                                                                     |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -F separator      | Legt das Eingabefeld-Trennzeichen fest (Field Separator). Standard sind whitespace (Leerzeichen/Tabs)file-4fdk7rvx39azjretinarqb. Z.B. -F ":" für /etc/passwd mit Doppelpunkten. |
| -v Var=Wert       | Setzt eine AWK-Variable <i>Var</i> auf <i>Wert</i> vor dem Verarbeiten. Z.B. -v ORS="\n\n" um eine Leerzeile nach jeder Ausgabezeile einzufügen (ORS = Output Record Separator). |
| -f Scriptdatei    | Lies das AWK-Programm aus einer Datei anstatt aus der Befehlszeile. Damit kann man größere AWK-Skripte in eigenem File verwalten.                                                |
| -m, --sandbox     | (gawk) AWK in sicheren Modus schalten (kein System(), keine Schreibrechte auf FS etc).                                                                                           |
| --posix           | Schaltet in POSIX-kompatiblen Modus (ggf. einige GNU-Erweiterungen aus).                                                                                                         |
| -W version        | Zeigt AWK-Version an (für gawk).                                                                                                                                                 |
| --help, --version | Hilfe / Version.                                                                                                                                                                 |

#### AWK Sprachkonzepte (Kurz):

- Ein AWK-Programm besteht aus *Muster { Aktion }* Blöcken. Muster ist ein Regex oder Ausdruck, Aktion ist Code in AWK.
- Felder: Eingabezeile wird in Felder zerlegt, zugreifbar als \$1, \$2, ..., \$NF (NF=Anzahl Felder). \$0 ist die ganze Zeile.
- Beispiele:
  - awk '{ print \$1 }' file – gibt die erste Spalte jeder Zeile aus (mit Standard-Trenn whitespace).
  - awk '/error/ {print NR ":" \$0}' log.txt – sucht Zeilen, die "error" enthalten (Regex als Muster), und druckt Zeilennummer und kompletten Inhalt.
  - awk -F, '\$3 > 100 { sum += \$3 } END { print "Summe:", sum }' data.csv – mit Komma als Separator summiere Wert der 3. Spalte aller Zeilen >100 in Spalte3; am Ende (END-Block) gib Summe aus.
- AWK hat viele eingebaute Funktionen (Mathe, String), assoziative Arrays, Kontrollstrukturen (if, for, while).

## **sed (Stream Editor)**

**Beschreibung:** sed ist ein nicht-interaktiver Stream-Editor zum Bearbeiten von Texteingaben nach Mustern, insbesondere für Suchen&Ersetzen oder das Extrahieren von Zeilenfile-4fdk7rvx39azjretinarqb. sed liest zeilenweise Eingabe (Dateien oder STDIN), wendet die angegebenen *Editier-Befehle* darauf an und schreibt das Ergebnis auf STDOUT (oder optional in-place zurück in Dateien). Am verbreitetsten ist die *Substitution* mit dem s/alt/neu/ Kommando.

### **Syntax:**

sed [Optionen] 'Script' Datei...

Das *Script* besteht aus einem oder mehreren Editierbefehlen, die auf die Eingabe angewendet werden. Beliebte sed-Befehle: **s** (substitute), **p** (print), **d** (delete), **q** (quit), **{}** für Gruppen, Adressierungsmodi (Zeilen-Nr oder Regex).

### **Optionen:**

| <b>Option</b>              | <b>Beschreibung</b>                                                                                                                                                                                                                                                                    |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-n, --quiet/-silent</b> | Suppress normal output. Normal drückt sed jede Eingabezeile (ggf. modifiziert) automatisch. Mit -n wird nichts automatisch gedruckt, nur explizite p (print) Befehle geben ausfile-4fdk7rvx39azjretinarqb. Typisch in Kombination mit p in Script, um nur bestimmte Zeilen auszugeben. |

| Option                           | Beschreibung                                                                                                                                                                                                                                                                                                          |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-e Script</b>                 | Gibt das zu verwendende sed-Skript (Editierbefehle) direkt an. Kann mehrfach genutzt werden für mehrere Befehle in Reihenfolge. Wenn nur ein Skript-Argument ohne -e gegeben ist, kann -e weggelassen werden.                                                                                                         |
| <b>-f Datei</b>                  | Lies das sed-Skript aus einer Datei. Erlaubt komplexere Edits aus einem File anstatt Kommandozeile.                                                                                                                                                                                                                   |
| <b>-i[SUF], --in-place[=SUF]</b> | Bearbeitet Dateien <i>in place</i> , d.h. schreibt Änderungen direkt zurück in die Eingabedateien. Optional kann ein <i>Suffix</i> angegeben werden, um vor Änderungen ein Backup der Originaldatei anzulegen (z.B. -i.bak). Ohne Suffix wird die Datei direkt überschrieben <a href="http://ionos.de">ionos.de</a> . |
| <b>-r, -E</b>                    | Schaltet die Verwendung <i>erweiterter regulärer Ausdrücke</i> ein (so dass in Patterns +, ?, {}, etc. ohne Backslash verwendet werden können). Bei GNU sed ist -r (old syntax) und -E (wie bei BSD sed) identisch.                                                                                                   |
| <b>-s</b>                        | Treat files separately. Normalerweise bei mehreren Dateien behandelt sed den Input als einen langen Stream. Mit -s wird nach jeder Datei der interne Zustand (z.B. Zeilennummer, \$-Regex-Flag) zurückgesetzt.                                                                                                        |
| <b>-z, --null-data</b>           | Input wird als Null-terminiert (Zeilen getrennt durch ASCII NUL statt newline) angenommen. Ermöglicht Verarbeitung von Binär oder multiline-chunks. Entspricht dann, dass . nicht auch \n matcht und \$ nur am NUL erkennt. (GNU sed)                                                                                 |
| <b>--follow-symlinks</b>         | Wenn in-place auf Symlinks angewendet, ändert sed standardmäßig den Link selbst (also das Zielfile, aber eben durch den Link?). Mit dieser Option folgt sed dem Symlink und editiert die Ziel-Datei. (GNU sed)                                                                                                        |
| <b>--sandbox</b>                 | Verhindert, dass sed w oder s//w zum Schreiben verwendet oder ! Shell-Aufrufe macht. (GNU sed 4.4+)                                                                                                                                                                                                                   |
| <b>--help, --version</b>         | Hilfe / Version.                                                                                                                                                                                                                                                                                                      |

### Häufige sed-Anwendungen:

- **Substitution (Ersetzen):**

Syntax: s/REGEX/REPLACEMENT/[Flags]. Ersetzt das erste Vorkommen von REGEX in jeder Zeile durch REPLACEMENT. Flags: g = global (alle Vorkommen in Zeile ersetzen), i = case-insensitive, p = print (selbst wenn -n), w file = gefundene Zeilen nach file schreiben.

Beispiel: sed -i 's/oldword/newword/g' file.txt – ersetzt *oldword* durch *newword* überall in file.txt, direkt in der Datei `file-5y1edouysrow4zj6phjcqm`.

Besonderheit: Im Ersatz kann man & verwenden für den gesamten Match oder \1 ... für Capturing Groups aus dem Regex.

- **Zeilen löschen:**

z.B. sed '/^#/ d' config – löscht alle Zeilen, die mit # beginnen (d = delete). Oder sed '5,10d' file löscht Zeilen 5 bis 10.

- **Nur bestimmte Zeilen ausgeben:**

Mit -n und p. Z.B. sed -n '1,5p' file druckt nur Zeilen 1-5 (andere werden durch -n nicht ausgegeben). Oder sed -n '/ERROR/p' log druckt nur Zeilen mit "ERROR".

- **Mehrere Edits:**

z.B. sed -e 's/foo/bar/' -e '/BAZ/d' – erst foo->bar ersetzen, dann Zeilen mit "BAZ" löschen. (Die Reihenfolge ist wichtig).

- **In-Place Backup:**

sed -i.orig 's/^#/ #' \*.conf – kommentiert alle Zeilen in \*.conf aus, speichert vorher Backup mit .orig-Suffix.

Sed kann viel mehr (Insert/Append mit i\ und a\, Transform mit y///, Shell-Befehle mit !, Holding Space etc.), was in Tiefe für Prüfung wohl nicht nötig ist. Wichtig ist Suchen&Ersetzen und Basisfilter.

## **tee (Output verdoppeln)**

**Beschreibung:** Liest die Standard-Eingabe und schreibt sie *gleichzeitig* auf die Standard-Ausgabe **und** in eine oder mehrere Dateienfile-4fdk7rvx39azjretinarqb. Es "verzweigt" den Datenstrom (daher der Name T-Stück). So kann man die Ausgabe eines Kommandos in eine Datei umleiten, aber trotzdem im Terminal sehen.

### **Syntax:**

... | tee [Optionen] <Datei>...

### **Optionen:**

| <b>Option</b>                  | <b>Beschreibung</b>                                                                                                                                   |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-a, --append</b>            | Hängt die Ausgabe an bestehende Dateien an, anstatt sie zu überschreibenfile-4fdk7rvx39azjretinarqb. Standard ohne -a überschreibt Ziel-Dateien neu.  |
| <b>-i, --ignore-interrupts</b> | Ignoriert Interrupt-Signale (SIGINT). D.h. Ctrl+C wirkt nicht auf tee selbst (nur auf vorheriges Kommando?), damit tee garantiert die Daten schreibt. |
| <b>--output-error MODE</b>     | Verhaltensweise falls ein Schreibfehler auftritt: warn (Fehler melden, aber weiter versuchen andere outputs zu schreiben), exit (sofort)              |

| <b>Option</b> | <b>Beschreibung</b>                                                       |
|---------------|---------------------------------------------------------------------------|
|               | beenden, evtl. ungeschriebene outputs gehen verloren). Standard ist warn. |

*Anwendung:*

- make 2>&1 | tee build.log – Zeigt alle Ausgaben von make im Terminal und speichert sie gleichzeitig in *build.log*.
- ls -l | tee -a listing.txt – Ausgabe von ls -l normal anzeigen und ans Ende von listing.txt anfügen.
- echo "strict mode on" | sudo tee /proc/sys/kernel/yama/ptrace\_scope – Durch sudo tee kann man Inhalte in Dateien schreiben, für die man Rechte braucht, während Eingabe evtl. aus einem un-privilegierten Echo kommt.

Tee ist also nützlich bei Logging und bei Verwendung mit sudo zum in Dateien schreiben, da sudo echo >> file wegen Redirection sonst nicht klappt.

### **xargs (Execute Arguments)**

**Beschreibung:** Liest Eingaben (Standardinput oder Datei) und führt daraus Kommandoaufrufe zusammengebaut aus. Typisch, um die Ausgabe eines Programms als Argumentliste für ein anderes Programm zu verwendenfile-4fdk7rvx39azjretinarqbfile-4fdk7rvx39azjretinarqb. xargs nimmt zeilen- oder spaltengetrennte Wörter und packt möglichst viele davon in einen Aufruf eines angegebenen Kommandos (Standard: echo). Damit überwindet man die Shell-Längenbeschränkung und kann sequentiell Befehle auf Listen anwenden.

#### **Syntax:**

xargs [Optionen] [Befehl [Initial-Argumente]]

Wenn kein Befehl angegeben ist, benutzt xargs per Default echo (einfaches Ausgeben).

#### **Optionen:**

| <b>Option</b>     | <b>Beschreibung</b>                                                                                                                                                                                                                                                                                                                                      |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-n MaxArgs</b> | Maximal <i>MaxArgs</i> Argumente pro aufgerufenem Kommando verwendenfile-4fdk7rvx39azjretinarqb. Z.B. -n 1 bedeutet, xargs nimmt jeweils 1 Argument und führt das Kommando damit aus (also für jede Eingabezeile einmal). Ohne Angabe bündelt xargs so viele Argumente wie möglich pro Aufruf, bis die Systemlimite (Getconf ARG_MAX) fast erreicht ist. |

| <b>Option</b>                | <b>Beschreibung</b>                                                                                                                                                                                                           |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-L MaxLines</b>           | Nimmt maximal <i>MaxLines</i> Input-Zeilen pro Kommando-Aufruf. Z.B. -L 1 bewirkt, dass jede einzelne Zeile als eine xargs-Ausführung behandelt wird (selbst wenn Zeile mehrere Wörter enthält).                              |
| <b>-d Delim</b>              | Verwendet <i>Delim</i> als Trennzeichen anstelle von Whitespace oder newline. Kann auch mehrere Zeichen (String) sein (GNU). Beispiel: -d ':' um : als Separator zu nehmen.                                                   |
| <b>-0, --null</b>            | Erwartet Input-Felder durch Nullbyte getrennt (wie von find -print0 oder grep -Z erzeugt). Dies ermöglicht sichere Übergabe von Dateinamen mit Leer-/Sonderzeichen <a href="http://stackoverflow.com">stackoverflow.com</a> . |
| <b>-E EOFSTR</b>             | Definiert eine Zeichenfolge, die als <i>Ende der Eingabe</i> interpretiert wird. Wenn xargs diese Zeichenfolge liest, bricht es die weitere Verarbeitung ab (auch wenn nicht EOF).                                            |
| <b>-e</b>                    | (veraltet, äquivalent zu -E ohne Parameter) – historically to specify EOF marker, oft nicht mehr benötigt.                                                                                                                    |
| <b>-I ReplaceStr</b>         | Nutzt <i>ReplaceStr</i> als Platzhalter in dem auszuführenden Kommando, der durch den gesamten Input (pro Aufruf) ersetzt wirdfile-                                                                                           |
| <b>-J ReplaceStr</b>         | 4fdk7rvx39azjretinarqb. Z.B. xargs -I {} mv {} /backup nimmt jeden Input (eine Datei) und substituiert anstelle von {} im mv-Aufruf. Bei -I wird implizit -L 1 angenommen (d.h. pro Zeile ein Aufruf).                        |
| <b>-t, --verbose</b>         | (BSD xargs) Ähnlich -I, aber ersetzt nur einmal in der Kommandozeile (kann mit -n kombinieren). In GNU xargs gibt es -J nicht.                                                                                                |
| <b>-p, --interactive</b>     | Gibt das Kommando, das ausgeführt wird, vorher auf stderr aus (also "echo" what is being done). Zum Debuggen hilfreich.                                                                                                       |
| <b>-r, --no-run-if-empty</b> | Fragt vor jedem Kommando nach Bestätigung (y/n), ähnlich wie -t zeigt es Befehl. Bei 'y' ausführen, 'n' überspringen.                                                                                                         |
| <b>-0, --null</b>            | Führt das Kommando <b>nicht</b> aus, wenn keine Argumente aus Input entstehen. Standard bei GNU xargs ist, <i>kein</i> Leerlaufaufruf zu machen (sprich diese Option ist implizit an). In POSIX könnte Standard anderes sein. |
| <b>(siehe oben)</b>          | Null-terminierte Eingabe verwenden – wichtig in Kombination mit find -print0 um sicher Dateilisten zu pipen ohne Probleme mit Sonderzeichen <a href="http://stackoverflow.com">stackoverflow.com</a> .                        |

| <b>Option</b>                  | <b>Beschreibung</b>                                                                                                                                                                                                                                             |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>--max-chars=K, -s K</b>     | Maximal K Zeichen pro Kommandozeilen-Aufruf verwenden (inklusive Kommando und env). Standard an System ARG_MAX orientiert (oft ~131072). Manchmal zu reduzieren bei seltsamen Umgebungen.                                                                       |
| <b>--max-procs=N, -P N</b>     | Bis zu N Prozesse <i>parallel</i> ausführen statt nacheinander. Z.B. -P 4 führt bis zu 4 Instanzen des Kommandos gleichzeitig aus, um Parallelisierung zu erreichen. N=0 bedeutet unbegrenzt viele parallel (Vorsicht!). Hilfreich z.B. in Multi-Core-Systemen. |
| <b>--process-slot-var=NAME</b> | Setzt eine Umgebungsvariable \$NAME in den aufgerufenen Prozessen, die eindeutige ID des xargs-Slots enthält (0..N-1) – bei -P parallel. Hilft um Ausgaben zuzuordnen.                                                                                          |
| <b>-a Datei</b>                | Liest Input aus <i>Datei</i> anstatt STDIN. So kann xargs auch ohne Pipe aus File speisen.                                                                                                                                                                      |
| <b>--help, --version</b>       | Hilfe/Version.                                                                                                                                                                                                                                                  |

### Beispiele:

- find /tmp -type f -name '\*.txt' -print0 | xargs -0 rm -f – Löscht alle .txt-Dateien unter /tmp (sicher gegenüber Sonderzeichen, dank -print0 und -0).
- grep -l "ERROR" \*.log | xargs -r zip errors.zip – Sucht in allen .log nach "ERROR", die -l Option listet nur Dateinamen mit Treffer. Diese Liste wird an xargs übergeben, das dann zip errors.zip <files> ausführt. Falls kein Treffer, -r sorgt dafür, dass zip nicht ohne Argument aufgerufen wird.
- cat list.txt | xargs -I {} echo File: {} >> files.txt – Für jede Zeile aus list.txt, führe echo File: <line> aus und append an files.txt. (So fügt man "File: " vor jede Zeile).
- seq 1 1000 | xargs -n 100 > numbers.txt – Nimmt die Zahlen 1 bis 1000 (jede auf neuer Zeile) und packt jeweils 100 Zahlen in eine Zeile, schreibt nach numbers.txt.
- printf "%s\0" \* | xargs -0 -n 1 -P 4 sha256sum – Berechnet parallel (4 at a time) die SHA256 Summen aller Dateien im aktuellen Verzeichnis. printf "%s\0" \* generiert Null-separierte Liste aller Dateien, xargs nimmt je 1 (-n1) und startet bis zu 4 sha256sum gleichzeitig.

## Software- und Paketmanagement

### tar (Tape ARchiver)

**Beschreibung:** Archiviert mehrere Dateien zu einer einzigen Datei (tarball). Tar kann Dateien **bündeln** (zu .tar Archiven) und in Zusammenarbeit mit Kompressionsprogrammen (gzip, bzip2, xz etc.) zugleich komprimierenfile-4fdk7rvx39azjretinarqb. Oft verwendet, um Verzeichnisse oder viele Dateien als ein Archiv zu speichern oder zu verteilen (.tar.gz, .tgz, .tar.bz2 etc.).

#### Syntax:

tar [Optionen] -f <Archiv> [Dateien/Verzeichnisse...]

Tar hat **Modi** (Mutual exclusive Optionen): Erstellen, Extrahieren, Listen etc. Üblich ist die kombinierte Kurzoption wie -czf oder -xvf. Wichtig: -f (file) gibt an, welches Archivfile zu benutzen ist, und kommt als *ein* Argument, meist zuletzt bei Short-Options-Kombination.

#### Haupt-Modi:

- **-c, --create** : Erstelle ein neues Archiv aus den  
**(Fortsetzung)**

### tar (Tape Archiver)

**Beschreibung:** Erstellt oder entpackt Archive aus mehreren Dateien. tar bündelt Dateien in **einem** Archiv (Containerdatei mit Endung .tarfile-4fdk7rvx39azjretinarqb】 . Oft wird tar mit Kompressionsprogrammen kombiniert (z.B. gzip oder bzip2), um komprimierte Archive wie .tar.gz zu erstellefile-4fdk7rvx39azjretinarqb】 .

#### Syntax:

tar <Modus-Option> [Optionen] -f <Archivdatei> [Dateien/Verzeichnisse...]

Tar benötigt **eine** der Hauptoptionen: -c (create), -x (extract) oder -t (list). Dazu wird **-f** angegeben, um den Namen der Archivdatei zu bestimmen. Häufig werden mehrere Kurzoptionen kombiniert (z.B. -czvf für "create, gzip, verbose, file").

#### Wichtige Optionen:

- **-c – create:** Neues Archiv erstellefile-4fdk7rvx39azjretinarqb】 .
- **-x – extract:** Dateien aus Archiv entpacken.
- **-t – list:** Inhalt des Archivs auflisten (nicht extrahieren).
- **-f Datei** – Gibt den Archivnamen an (Pfad/Name der Tar-Datei). Diese Option **muss** direkt gefolgt von dem Dateinamen kommen (z.B. -f archive.tar).
- **-v – verbose:** Ausführliche Ausgabe (listet während der Verarbeitung die Dateinamen auf).

- **-z** – Filter durch **gzip**: Bei -c wird das Archiv mit gzip komprimiert (\*.tar.gz); bei -x wird erwartet, dass das Archiv gz-komprimiert ist `file-4fdk7rvx39azjretinarqb`. (Äquivalent: --gzip oder mit externem gzip pipen).
- **-j** – Filter durch **bzip2**: Erstellt/entpackt ein bzip2-komprimiertes Archiv (\*.tar.bz2). (Alternativ --bzip2.)
- **-J** – Filter durch **xz**: Für .tar.xz-Archive (--xz).
- **-A – concatenate**: Hängt ein oder mehrere Archive an ein bestehendes Archiv an.
- **-d – diff/compare**: Vergleicht Archiveinträge mit Dateien im Dateisystem (prüft, ob Unterschiede bestehen).
- **-C DIR – Change Directory**: Wechselt ins Verzeichnis *DIR*, bevor Dateien hinzugefügt oder extrahiert werden [freedesktop.org](#). Z.B. tar -xvf archive.tar -C /tmp entpackt nach /tmp.
- **--strip-components=N** – Beim Extrahieren lässt tar N führende Pfadkomponenten weg. Z.B. entpackt /usr/local/share/doc/file als file bei --strip-components=3. Praktisch, um Archiv ohne Top-Level-Ordner auszupacken.
- **--exclude=PATTERN** – Schließt Dateien, die auf PATTERN passen, vom Archivieren aus. Z.B. --exclude="\*.o".
- **--wildcards, --anchored** – steuert, wie Muster bei -t oder -x --wildcards -f archive "\*.txt" interpretiert werden (glob-Muster an/aus etc.).
- **--numeric-owner** – Beim Entpacken Besitzer/Gruppen als IDs übernehmen (statt Namen auf dem Zielsystem zuzuordnen).

### Beispiele:

- Archiv erstellen: tar -cvf archive.tar Verzeichnis/ erzeugt ein unkomprimiertes Archiv *archive.tar* mit allen Dateien aus *Verzeichnis/*.
- Mit Kompression: tar -czvf etc.tar.gz /etc packt ganz /etc als gzip-komprimiertes Archiv.
- Entpacken: tar -xvzf backup.tar.gz -C /restore entpackt *backup.tar.gz* nach */restore*.
- Auflisten: tar -tvf archive.tar zeigt die Dateiliste mit Größen, Timestamp, Rechten etc.

### gzip (GNU Zip Compression)

**Beschreibung:** Komprimiert oder dekomprimiert Dateien im **gzip-Format** (.gz). Es ersetzt beim Komprimieren standardmäßig die Originaldatei durch die komprimierte

Version (und fügt .gz an file-4fdk7rvx39azjretinarqb】 . Gzip ist schneller, aber erzeugt etwas größere Dateien als bzip2/xz. Häufig mit tar (siehe oben) eingesetzt.

### Syntax:

gzip [Optionen] [Datei...]

gunzip [Optionen] [Datei...]

(gunzip ist äquivalent zu gzip -d.)

### Optionen:

- **-c, --stdout:** Schreibt das komprimierte Ergebnis auf die Standardausgabe, anstatt die Datei zu ersetzen. Original bleibt erhalten (kann in Pipeline genutzt werden[ionos.de](#)】 .
- **-d, --decompress:** Dekomprimiert die .gz-Datei (auch via gunzip aufrufbar[ionos.de](#)】 .
- **-k, --keep:** Behält Originaldatei beim Komprimieren/Entpacken (standardmäßig wird Original gelöscht).
- **-r, --recursive:** Komprimiert alle Dateien in angegebenen Verzeichnissen rekursiv.
- **-1 bis -9:** Einstellen der Kompressionsstufe (1 = schnellste/geringste Kompression, 9 = langsamste/beste). Standard ist 6.
- **-t, --test:** Prüft eine gz-Datei auf Integrität, ohne sie zu entpacken (exit-Status zeigt OK/NOK).
- **-l, --list:** Zeigt Informationen über eine komprimierte Datei: Originalgröße, komprimierte Größe, Kompressionsrate, Zeitstempel.
- **--stdout, -c** (nochmal): Wird oft mit -d kombiniert, um nach stdout zu entpacken (z.B. gzip -dc file.gz).
- **-N, --name:** Bewahrt ursprünglichen Dateinamen und Zeitstempel (in .gz-Header) oder stellt diese beim Entpacken wieder her (ist Standardverhalten bei GNU gzip).
- **--help, --version:** Hilfe bzw. Versionsinfo.

### Beispiele:

- gzip bigfile komprimiert *bigfile* zu *bigfile.gz* (und löscht *bigfile*).
- gzip -9 \*.txt erstellt maximal komprimierte \*.txt.gz Dateien.
- gunzip data.csv.gz entpackt *data.csv.gz* zurück zu *data.csv*.

- gzip -c file.txt > file.txt.gz komprimiert nach stdout und schreibt in separate Datei (Original bleibt).
- In Pipe: tar -cvf - /folder | gzip > folder.tar.gz – tar erzeugt Archiv auf stdout, gzip komprimiert es zu *folder.tar.gz*.

## xz (LZMA Compressor)

**Beschreibung:** Komprimiert/Dekomprimiert Dateien mit dem **xz/LZMA2-Algorithmus**. XZ erreicht hohe Kompressionsraten, aber benötigt mehr CPU und RAM. Dateiendung: .xz. Wird z.B. für Kernel-Quellen-Archive genutzt. Syntax und Optionen ähneln gzip.

### Syntax:

xz [Optionen] [Datei...]

unxz [Optionen] [Datei...]

(xz -d entspricht unxz.)

### Wichtige Optionen:

- **-z** (Default): Komprimieren (Standardmodus, kann entfallen).
- **-d, --decompress**: Entpacken.
- **-k, --keep**: Originaldatei behalten (nicht löschen).
- **-f, --force**: Vorhandene .xz-Dateien überschreiben, falls vorhanden; oder erzwingt Komprimieren auch bei scheinbar nicht sinnvoll (z.B. sehr kleine Datei).
- **-9** (auch -0 .. -9): Kompressionsgrad (0 = none, 6 = Standard, 9 = max). XZ erlaubt auch -e für extreme Mode (noch etwas bessere Kompression, deutlich langsamere Kodierung).
- **-T N, --threads=N**: Nutzung von N Threads für Kompression (für große Dateien auf Multi-Core). Standard 1. (Für Dekomprimierung wird threads automatisch benutzt).
- **-c, --stdout**: Ergebnis auf stdout schreiben (bzw. von stdin lesen und auf stdout entpacken).
- **-l, --list**: Listet das .xz-Archiv (Inhalt, ratio, etc. – bei Single-File Komprimierung liefert original und comp size).
- **-q, -v**: quiet bzw. verbose Mode.
- **-C, --check=TYPE**: Prüfsummenverfahren wählen (Standard CRC64). Kann z.B. auf CRC32, SHA256 gestellt werden.

### Beispiele:

- `xz hugefile` erzeugt `hugefile.xz` (Original wird gelöscht).
- `xz -dk archive.tar.xz` entpackt `archive.tar.xz` nach `archive.tar` und behält auch die `.xz`.
- `tar -c /usr/bin | xz -9e > bin.tar.xz` erstellt maximal-komprimiertes tar-Archiv.
- `unxz data.csv.xz` oder `xz -d data.csv.xz` entpackt Datei.
- `xzcat file.txt.xz = xz -dc file.txt.xz` entpackt nach stdout (ähnlich zcat).

### **bzip2 (Burrows–Wheeler Compressor)**

**Beschreibung:** Komprimiert mit dem älteren bzip2-Algorithmus (.bz2). Langsamer als gzip, aber bessere Kompression; heute oft von xz abgelöst. Bedienung ähnlich gzip.

**Syntax:** `bzip2 [Optionen] [Datei...]` – bunzip2 zum Entpacken.

**Optionen (Auswahl):** `-k` (keep file), `-d` (decompress), `-1 .. -9` (Stufe, Standard 9 bei bzip2), `-c` (stdout), `-t` (Test), `-v` (verbose stats).

### **zip (Archiv mit Kompression, ZIP-Format)**

**Beschreibung:** Erstellt ZIP-Archive (mit Kompression) oder fügt Dateien hinzu. Anders als tar erwartet zip direkt die zu archiviernden Dateien und komprimiert sie einzeln ins Archiv. ZIP-Archive enthalten Verzeichnisstruktur + komprimierte Dateien (beliebtes Format unter Windows). unzip entpackt sie.

**Syntax:**

`zip [Optionen] <Archiv.zip> <Dateien...>`

**Wichtige Optionen:**

- `-r` – Rekursiv Verzeichnisse einschließen. Z.B. `zip -r archiv.zip Ordner/`.
- `-9` (bzw. `-1 .. -9`) – Kompressionsgrad (9 = beste, Standard meist 6).
- `-q` – quiet, `-v` – verbose (gibt Kompressionsraten aus).
- `-m` – move: Dateien nach dem Hinzufügen ins Archiv löschen (verschieben).
- `-j junk paths`: Pfadnamen ignorieren, legt alle Dateien „flach“ im Archiv ab (ohne Verzeichnisstruktur).
- `-x <Pattern>` – bestimmte Dateien ausschließen (z.B. `-x "*.o"`).
- `-F` – Versucht ein beschädigtes Archiv zu reparieren (freshen).
- `-u update`: nur neuere Dateien (bzgl. Zeitstempel) ins Archiv aktualisieren.
- `-g grow`: hängt Dateien an ein bestehendes ZIP an (statt neu zu erstellen).

- **-e** – erstellt ein verschlüsseltes Archiv (fordert Passwort-Eingabe, traditionelles ZipCrypto; **-P <pw>** unsicher ohne Prompt).

### **unzip Wichtige Optionen:**

unzip <file.zip> entpackt im aktuellen Verzeichnis. **-d <DIR>** entpackt ins DIR. **-l** listet Inhalte. **-t** testet. **-x** schließt beim Entpacken bestimmte aus. **-n** überschreibt nichts, **-o** überschreibt ohne Nachfrage.

## **Paketmanagement-Befehle (Distributionen)**

### **rpm (RPM Package Manager)**

**Beschreibung:** Das Low-Level-Werkzeug zum Installieren, Entfernen und Abfragen von RPM-Paketen (Red Hat, CentOS, Fedora, SUSE etc.) . rpm arbeitet auf .rpm-Dateien und der lokalen Paketdatenbank. Häufig nutzt man für höhere Ebene yum oder dnf, aber rpm bleibt wichtig für direkte Paketoperationen.

#### **Syntax (Basis):**

rpm [Optionen] <Paketdatei> / rpm [Optionen] <Paketname>

#### **Operationen (Hauptoptionen, nicht kombinierbar):**

- **-i** – *install* ein neues RPM-Paket . Beispiel: rpm -i nginx.rpm. Erfordert Root-Rechte.
- **-U** – *upgrade* (installiert neu oder aktualisiert älteres Paket). Wird häufig anstelle -i benutzt, da es sowohl installiert als auch updatet.
- **-F** – *freshen*: wie -U, aber nur wenn schon ältere Version installiert ist (sonst kein Install).
- **-e** – *erase* (deinstalliert) ein installiertes Paket nach Paketname . Beispiel: rpm -e nginx.
- **-q** – *query*: Abfrage-Modus (ohne Änderungen) . Kann mit Zusätzen Info liefern. Beispiel: rpm -q nginx prüft, ob Paket installiert ist und welche Version.

#### **Wichtige Optionen/Zusätze:**

##### Install/Upgrade Optionen:

- **-h** – zeigt Hash-Fortschriffs Balken beim Installieren ("#" -Zeichen).
- **-v** – verbose, mehr Ausgabe. Oft kombiniert -Uvh oder -ivh.
- **--force** – erzwingt Installation, selbst wenn z.B. ein Paket bereits installiert ist oder Dateien konfigurieren. (Vorsicht!).
- **--nodeps** – ignoriert Abhängigkeits-Prüfung (installiert trotzdem).

- **--replacefiles** – überschreibt Dateien, die zu anderen Paketen gehören könnten (Konflikte ignorieren).
- **--nossripts** – führt keine Pre/Post-Install-Skripte im RPM aus (für spezielle Fälle).

Query Optionen (mit -q):

- **-a** – query *all* (listet alle installierten Pakete).
- **-i** – (mit -q) zeigt detaillierte Paketinformationen (Name, Version, Summary, Größe, Install-Datum etc.[linux-praxis.de](http://linux-praxis.de)) . Z.B. rpm -qi bash.
- **-l** – listet alle vom Paket installierten Dateien. ` rpm -ql <paket>file-5y1edouysrow4zj6phjcqm` .
- **-p** – query ein *Paketfile* (nicht installiert). Z.B. rpm -qp -l paket.rpm zeigt, welche Dateien im RPM drin sind.
- **-f Dateipfad** – findet heraus, zu welchem installierten Paket eine bestimmte Datei gehört. Z.B. rpm -qf /usr/bin/ls -> coreutils-8.xx.
- **-c** – listet nur Konfigurationsdateien des Pakets (mit -ql kombinierbar: -qlc).
- **--changelog** – zeigt das Changelog des Pakets (sofern im RPM hinterlegt).
- **--provides, --requires** – zeigen welche Provides/Requires (Abhängigkeiten) ein Paket liefert bzw. benötigt. Z.B. rpm -q --requires httpd.
- **-s** – zeigt den Status der Dateien (ob sie verändert wurden seit Installation, (vom user modifiziert = **M**, missing etc.)). rpm -qs <paket>.

Erase (Uninstall) Optionen:

- **--nodeps** beim -e: ignoriert Abhängigkeitsprobleme (entfernt auch wenn andere Pakete darauf angewiesen sind - kann zu kaputten Programmen führen).

### **Beispiele:**

- rpm -Uvh httpd-2.4.rpm – installiert Apache httpd, zeigt Hashtags und Namen.
- rpm -qa | grep kernel – listet alle installierten Pakete, filtert "kernel".
- rpm -qpi package.rpm – zeigt Infos über nicht installiertes RPM.
- rpm -e httpd – entfernt Apache-Paket.

### **yum (Yellowdog Updater Modified)**

**Beschreibung:** Ein High-Level-Paketmanager für RPM-basierte Distr. (v.a. CentOS/RHEL 7 und älter). yum löst Abhängigkeiten automatisch auf und kann aus Repositories Pakete installieren, entfernen, aktualisieren . (In neueren RHEL8+ abgelöst durch dnf, aber Syntax ähnlich.)

## Syntax:

yum [Optionen] <Sub-Befehl> [Paketnamen/...]

## Haupt-Subkommandos:

- **install Paket** – Installiert ein Paket (aus den konfigurierten Repos oder einer RPM-Datei). Z.B. yum install httpd. Resolved dependencies automatisch und fragt um Bestätigung.
- **update [Paket]** – Aktualisiert alle installierten Pakete (wenn kein Name angegeben) oder ein bestimmtes. (yum update führt System-Update durch).
- **remove Paket, erase** – Entfernt angegebene Pakete mit Abhängigkeiten, die dadurch verwiesen.
- **search Stichwort** – Durchsucht die Paketnamen, Beschreibungen etc. nach dem Stichwort (um herauszufinden, welches Paket etwas bietet).
- **info Paket** – Zeigt Infos zu Paket(en) (Name, Version, Summary, Größe, Repo, Abh.) ähnlich rpm -qi aber aus Repos wenn nicht installiert.
- **list [installed|available] <Pattern>** – Listet Pakete (installierte oder verfügbare). Ohne Parameter alle Pakete, optional mit Pattern filtern. Z.B. yum list available 'php\*'.
- **provides Datei/Capability** – Findet heraus, welches Paket eine bestimmte Datei oder "Provide" bereitstellt (ähnlich rpm -qf/--whatprovides). Z.B. yum provides /usr/bin/zip.
- **clean <was>** – Bereinigt lokale Caches: z.B. yum clean all (löscht Metadaten- und Paketcache), oder yum clean metadata etc.
- **check-update** – Zeigt verfügbare Updates, ohne sie zu installieren (Exitcode 100 wenn Updates vorhanden, 0 wenn keine).
- **repolist** – Listet eingerichtete Repositories und ob aktiviert.
- **groupinstall / groupremove** – Installiert/entfernt ganze Paketgruppen (z.B. yum groupinstall "Development Tools").

## Optionen (Auswahl):

- **-y** – Automatisch "yes" beantworten (nicht nach Bestätigung fragen). Wichtig für Skript.
- **-q** – Weniger Ausgabe (quiet).
- **-C** – Nur aus dem Cache arbeiten, keine neuen Metadaten laden (offline).

- **--enablerepo=X, --disablerepo=Y** – Repositories gezielt ein-/ausschalten für diesen Befehl (Overrides config).
- **--nogpgcheck** – Deaktiviert GPG-Signaturprüfung der Pakete (unsicher, nur falls Repo-Key nicht installiert etc.).
- **--downloadonly** – Lädt die Pakete herunter, installiert aber nicht.
- **--security** – (bei update/info/list) berücksichtigt nur sicherheitsrelevante Updates (wenn Repo das unterstützt).

### **Beispiele:**

- yum install nginx php php-mysql – installiert mehrere Pakete in einem Rutsch.
- yum update – bringt System auf neuesten Stand.
- yum search "pdf viewer" – sucht nach Paketen mit "pdf viewer" im Namen/Beschreibung.
- yum remove httpd – deinstalliert Apache (und abhängige Module, wenn keine anderen es benötigen).
- yum provides "\*bin/rar" – findet Paket, das rar-Binary liefert.

(*Hinweis: Auf neueren Systemen ersetzt dnf den Befehl yum. dnf Syntax und Subcommands sind weitgehend gleich, aber z.B. dnf upgrade statt yum update.)*

### **apt (Advanced Package Tool – Debian/Ubuntu)**

**Beschreibung:** Höher-level Frontend für Debian-Paketverwaltung (dpkg). apt kombiniert Funktionen von apt-get und apt-cache in einem Kommando. Es verwaltet .deb-Pakete: Installation, Updates aus Repositories, Paket-Suche usw]. (Ubuntu 16.04+ und Debian Stretch+ empfehlen apt für interaktive Nutzung, während apt-get skriptfreundlich bleibt.)

### **Syntax:**

apt [Optionen] <Sub-Befehl> [Pakete...]

### **Wichtige Sub-Befehle:**

- **update** – Aktualisiert die Paketquellen-Datenbank (lädt neue Paketlisten von den konfigurierten Repos). Immer vor Install/Upgrade ausführen.
- **upgrade** – Installiert verfügbare Updates für alle **schon installierten** Pakete (führt keine Entfernung durch). Hält Pakete zurück, wenn Abhängigkeitswechsel nötig wären. (vgl. *apt full-upgrade*).
- **full-upgrade** – Führt vollständige Distribution-Upgrades durch, d.h. aktualisiert alle Pakete und darf auch Pakete entfernen oder neue Abhängigkeiten

hinzufügen, um ein konsistentes System herzustellen (entspricht apt-get dist-upgrade).

- **install <pkg>...** – Installiert angegebene Pakete (aus Repos oder .deb Dateien via Pfad). Löst Abhängigkeiten auf und fragt bei Bedarf. Akzeptiert Paket=Version oder Paket:arch.
- **remove <pkg>...** – Entfernt Pakete, lässt jedoch veränderte Konfigurationsdateien bestehen (als .config in /etc).
- **purge <pkg>...** – Entfernt Pakete **inklusive** Konfigurationsdateien (bereinigt vollständig).
- **search <regex>** – Durchsucht Paketnamen und Beschreibungen nach einem Begriff (regex möglich). *Beispiel:* apt search editor pdf.
- **show <pkg>** – Zeigt detaillierte Informationen zu einem Paket (Version, Abhängigkeiten, Beschreibung, Maintainer etc.), egal ob installiert oder nur verfügbar.
- **list** – Listet Pakete. Mit --installed, --upgradeable oder --all-versions. (Ähnlich dpkg -l aber filterbar). Z.B. apt list --installed | grep nginx.
- **autoremove** – Entfernt automatisch Pakete, die als Abhängigkeiten installiert wurden und nun nicht mehr benötigt werden (Waisen).
- **depends <pkg>, rdepends <pkg>** – Zeigt Abhängigkeiten bzw. Reverse-Abhängigkeiten eines Pakets (welche Pakete benötigen es).
- **edit-sources** – Öffnet die sources.list im Editor (um Repos zu bearbeiten).

#### Optionen:

- **-y, --yes** – Automatisch alle Fragen mit Yes beantworten (nicht interaktiv bestätigen). Praktisch für Skripte.
- **-q** – Weniger Ausgabe (bis zu -qq für gar keine).
- **--no-install-recommends** – Installiert nicht automatisch als "empfohlen" markierte Pakete. Standard apt installiert Recommends mit.
- **--allow-remove-essential, --allow-change-held-packages** – Erlaubt riskante Änderungen (Standard verhindert z.B. Entfernen essentieller Pakete).
- **-s, --simulate** – Simulation: Zeigt an, was würde passieren, aber führt keine Änderung aus (auch Exitcode 0 immer). Gut zum Test.
- **-V, --verbose-versions** – Zeigt bei Auflistungen detailliertere Versionsinfos.
- **--help** – Hilfe-Seite, **--version** – Version von apt.

## **Beispiele:**

- apt update && apt upgrade – Aktualisiert Paketlisten und installiert alle verfügbaren Updates (ohne Dist-Upgrade).
- apt install build-essential – Installiert eine Paketgruppe (hier C/C++ Compiler Toolchain).
- apt remove --purge openssh-server – Entfernt SSH-Server komplett mit Config.
- apt search docker – Sucht alle Pakete mit "docker" im Namen/Beschreibung.
- apt full-upgrade – Aktualisiert System auch über Distribution-Upgrades hinweg (entspricht auf Ubuntu einem Release-Upgrade, sofern Repos angepasst).

(Hinweis: *apt-get* und *apt-cache* sind Vorgänger: z.B. *apt-get install*, *apt-get remove*, *apt-get dist-upgrade*, *apt-cache search*, *apt-cache show*. *apt* vereinfacht den Alltag, aber Skripte nutzen weiterhin oft *apt-get -y etc. dpkg* siehe unten ist noch eine Stufe tiefer für einzelne .deb ohne Abhängigkeitsauflösung.)

## **dpkg (Debian Package Manager - Low-Level)**

**Beschreibung:** dpkg ist das unterliegende Werkzeug auf Debian/Ubuntu, das .deb-Pakete entpackt, installiert, konfiguriert und entfernt. Es löst keine Abhängigkeiten – dafür sind apt/apt-get zuständig. Man verwendet dpkg meist für manuelle Offline-Installation von .deb-Dateien.

### **Syntax:**

```
dpkg -i <Paket.deb> # Installieren
dpkg -r <Paketname> # Entfernen
dpkg -P <Paketname> # Entfernen + Purge
dpkg -L <Paketname> # Liste Dateien des installierten Pakets
dpkg -s <Paketname> # Statusinfo (ähnlich apt show)
```

### **Wichtige Optionen/Parameter:**

- **-i, --install Datei.deb:** Installiert das angegebene Deb-Paket. Gibt Fehler aus, falls Abhängigkeiten fehlen. (Diese müssen vorher mit apt installiert werden).
- **-r, --remove Paketname:** Entfernt ein installiertes Paket (Paketname ohne ".deb", wie in dpkg -l gelistet). Konfigdateien unter /etc bleiben bestehen.
- **-P, --purge Paketname:** Entfernt Paket inkl. seiner Konfigurationsdateie.
- **-S, --search Dateiname:** Sucht, welches installierte Paket eine bestimmte Datei besitzt (ähnlich rpm -qf). Z.B. dpkg -S /usr/bin/foo.

- **-L, --listfiles Paketname:** Listet alle Dateien, die das installierte Paket auf dem System installiert hat [file-5y1edouysrow4zj6phjcqm] .
- **-l, --list Muster:** Listet Pakete, optional gefiltert nach Muster. dpkg -l ohne Filter zeigt eine Tabelle aller Pakete und deren Status (ii = installiert, rc = removed config-files left, etc.).
- **-s, --status Paketname:** Zeigt den Status und Informationen eines installierten Pakets (Version, Beschreibung, Maintainer, Abhängigkeiten...).
- **--configure Paketname:** Führt die Konfigurationsschritte (dpkg-configure) für ein installiertes aber noch nicht konfiguriertes Paket aus. (Wird z.B. nach entpacken aufgerufen).
- **-B, --auto-deconfigure:** Beim Entfernen eines Pakets, das Abhängigkeiten bereitstellt, markiert abhängige Pakete als "deconfigured".
- **\*\*--force-\*\*X:** Viele Force-Flags, um dpkg anzuweisen, bestimmte Fehler zu ignorieren (z.B. --force-dependency ignoriert fehlende Abhängigkeiten, --force-confold/--force-confnew bei Config-Datei-Konflikten etc.). Vorsicht: Kann System inkonsistent machen.
- **--print-architecture:** Zeigt die Architektur, für die dpkg Pakete installiert (z.B. amd64).
- **--add-architecture <arch>:** Fügt Multiarch-Unterstützung für andere Arch hinzu (z.B. i386 auf einem amd64 System).

### **Beispiele:**

- dpkg -i google-chrome.deb – Installiert Chrome .deb-Paket (scheitert, wenn Abhängigkeiten fehlen – in dem Fall via apt --fix-broken install auflösbar).
- dpkg -r postfix – Entfernt postfix, belässt aber ggf. geänderte /etc/postfix Dateien.
- dpkg -P postfix – Entfernt postfix vollständig inklusive Config.
- dpkg -L bash – Listet alle vom bash-Paket installierten Dateien.
- dpkg -S /bin/ls – Welches Paket stellt /bin/ls bereit? (Antwort: coreutils).

## **wget (Web GET Utility)**

**Beschreibung:** Ein Kommandozeilen-Downloader für Dateien über HTTP, HTTPS und FTP . wget kann in Skripten genutzt werden, um Dateien aus dem Web herunterzuladen, unterstützt Fortsetzen abgebrochener Downloads, rekursives Herunterladen von Websites etc. Es funktioniert nicht-interaktiv (kein Browser, nur Download).

### **Syntax:**

wget [Optionen] <URL>

## Wichtige Optionen:

- **-O Datei** – Speichert die heruntergeladene URL in die angegebene *Datei* (statt den Namen aus URL zu verwenden) . Z.B. wget -O ubuntu.iso http://.../ubuntu.iso.
- **-c, --continue** – Setzt einen abgebrochenen Download fort (sofern Server Range Requests unterstützt). Prüft, wieviel schon vorhanden ist und lädt den Rest von [Resionos.de](#).
- **-q** – Quiet-Modus (keine Ausgabe). **-nv** (not verbose) eine dazwischenliegende Stufe. **-v** für ausführlicher (Standard).
- **-r, --recursive** – Rekursiver Download einer Website. Erfordert oft zusammen mit -np (no parent), -L (stay on same domain), -k (Links anpassen für Offline). Z.B. wget -r -np -k http://example.com/docs/.
- **-np, --no-parent** – Geht beim rekursiven Download nicht über das Startverzeichnis hinaus.
- **-l <Tiefe>** – Legt die Rekursionstiefe fest (Default 5, -l0 = unendlich).
- **-p, --page-requisites** – Lädt alles mit, was nötig ist, um HTML-Seite offline darzustellen (Bilder, CSS...).
- **-N, --timestamping** – Lädt eine Datei nur herunter, wenn sie neuer ist als die lokale (verwendet Last-Modified). Gut zum Aktualisieren.
- **--limit-rate=<Rate>** – Begrenzt Downloadrate, z.B. --limit-rate=200k (200 KB/s). Hilfreich, um Bandbreite zu drosseln.
- **-U, --user-agent=<Agent>** – Setzt einen bestimmten User-Agent-String. Standard ist "Wget/Version". Kann geändert werden, um z.B. nicht blockiert zu werden.
- **--no-check-certificate** – Ignoriert SSL-Zertifikatfehler (z.B. selbstsigniert).
- **--user=<user> --password=<pw>** – Falls erforderlich, für HTTP-Auth oder FTP-Login. (Achtung: Sichtbar in Prozessliste).
- **-i <Datei>** – Liest eine Liste von URLs aus <Datei> (jede Zeile eine URL) und lädt alle herunter.
- **-b, --background** – Startet im Hintergrund (nur auf Unix, leitet Ausgabe nach wget-log).
- **-P <Verz>** – Speichert Dateien in angegebenem Verzeichnis (statt aktuelles).

## Beispiele:

- wget http://example.com/file.zip – Lädt *file.zip* in aktuelles Verzeichnis herunter.
- wget -c http://largefile – setzt Download fort, falls vorhanden.

- wget -r -np -N http://example.com/photos/ – Lädt Verzeichnis "photos" vollständig herunter, aktualisiert nur neue Dateien beim Wiederaufruf.
- wget -O - https://api.example.com/data | jq . – Lädt Daten von API und pipet direkt in jq zur Verarbeitung (hier -O - bedeutet Ausgabe auf stdout statt Datei).

## curl (Client URL)

**Beschreibung:** curl ist ein umfangreiches Kommandozeilen-Tool für Datenübertragungen mit URL-basierten Protokollen (HTTP, HTTPS, FTP, etc.). Im Gegensatz zu wget liegt Fokus auf Flexibilität bei HTTP-Requests (Header setzen, verschiedene HTTP-Methoden, Formulare absenden, APIs ansprechen etc.). Es eignet sich für Web-APIs (GET/POST) und kann ebenfalls Dateien herunterladen oder hochladen.

### Syntax:

curl [Optionen] <URL>

### Wichtige Optionen:

- **-o Datei** – Speichert die Ausgabe in *Datei*. (Kleinbuchstabe o, analog wget -O).
- **-O** – Speichert unter dem **Originalnamen** aus der URL (Großes O, analog wget ohne -O). Achtung: curl -O http://server/dir/file.txt legt *file.txt* an.
- **-L – Location folgen:** folgt HTTP-Redirects (Status 3xx) automatisch [freedesktop.org](https://freedesktop.org) . (wget macht das default, curl nicht ohne -L).
- **-C --** Setzt abgebrochenen Download fort (wenn Server unterstützt; '-' heißt automatischen Offset ermitteln).
- **-#** – Fortschrittsleiste im CLI (oder --progress-bar). Standard curl zeigt laufende Stats Zeile, -# macht wie wget progressive Bar.
- **-s** – Silent (kein Progress oder Fehlerausgaben). **-S** kombiniert mit -s zeigt Fehler dennoch an.
- **-v / -vv** – Verbose Mode (zeigt detaillierte Request/Response, Header etc. -vv noch mehr, inkl. Low-level). Gut zum Debuggen.
- **-I** – Führt eine HEAD-Anfrage aus (nur Header vom Server holen, keine Body-Daten). Nützlich um Meta-Informationen (Content-Length, Last-Modified) abzufragen.
- **-X METHOD** – Explizite HTTP-Methode festlegen (z.B. PUT, DELETE). Standard bei Datenvorhandensein oder -d ist POST, sonst GET.
- **-G** – Erzwingt, dass bei Verwendung von -d (Daten) diese als Query-Parameter an URL angehängt werden (statt POST Body).

- **-d Daten** – Sendet die angegebenen *Daten* im HTTP-Request-Body (Standard via POST). Z.B. curl -d "user=test&pass=123" http://.../login. Implizit -X POST wenn nicht -G.
  - **--data-urlencode Daten**: URL-encodet die Daten vor Versand (z.B. wenn Daten Sonderzeichen enthalten).
  - **--data-raw**: ähnlich -d, aber ohne spezielle Form-Codierung (sendet genau so).
- **-H "Header: Wert"** – Fügt einen HTTP-Header hinzu oder überschreibt ihn. Z.B. -H "Authorization: Bearer <token>". Mehrfach nutzbar für mehrere Header.
- **-u Benutzer:Pass** – Nutzt HTTP Basic Auth mit gegebenen Credentials (wird in Header "Authorization: Basic ...Base64" codiert[freedesktop.org](http://freedesktop.org)). Vorsicht in Shell-History. Kann auch nur -u user (dann fragt curl interaktiv Passwort ab). Funktioniert auch für FTP.
- **-F Feld@Datei** – HTTP Datei-Upload via *multipart/form-data*. Z.B. curl -F "file=@bild.png" -F "desc=Urlaubsfoto" http://.../upload.
- **-K Konigdatei** – Liest curl-Parameter aus Datei (kann nützlich sein für komplexe Setups).
- **-b CookieDatei** – Sendet Cookies aus angegebener Datei (. Netscape cookie format oder einfach "name=value" Paare). Kann auch -b "name=value" um Cookies zu setzen.
- **-c CookieAusgabe** – Speichert Cookies, die vom Server kommen, in angegebener Datei (um sie später wieder zu verwenden).
- **--limit-rate <Speed>** – Begrenzt Übertragungsrate (z.B. --limit-rate 100k).
- **--retry N** – Anzahl der automatischen Wiederholungsversuche bei Fehler (z.B. Timeout) oder --retry-all-errors.
- **--cacert Datei / --insecure**: Spezielles SSL: anderes CA-Zertifikat nutzen bzw. bei --insecure SSL-Zertifikat überprüfen *ignorieren* (wie wget --no-check-certificate).

### Beispiele:

- curl -L -o latest.zip https://github.com/user/proj/releases/latest – Lade neueste Release, folge Redirects, speichere als latest.zip.
- curl -u admin:secret -X DELETE http://server/api/item/123 – Sende DELETE-Request mit Basic Auth.
- curl -H "Accept: application/json" 'https://api.example.com/data?id=5' – Hole JSON-Daten von API mit angepasstem Accept-Header.

- curl -d '{"user":"john"}' -H "Content-Type: application/json" -X POST https://api.example.com/users – Sende JSON-Body per POST (hier ein neuer User).
- curl -F 'file=@report.pdf' -F 'desc=Monthly Report' https://upload.example.com – HTTP File-Upload (Multipart).
- curl -I https://example.com – Nur Header einer HTTP-Response anzeigen (z.B. um Last-Modified zu prüfen).

## Speicher- und Dateisystem-Verwaltung

### lsblk (List Block Devices)

**Beschreibung:** Zeigt eine Übersicht aller erkannten Blockgeräte (Festplatten, SSDs, Partitionen, LVM-Volumes etc.) in Form einer Baumstruktur  
`file-4fdk7rvx39azjretinarqb】 .`  
 lsblk listet wichtige Attribute wie Gerätename, Größe, Typ (disk/part), Mountpoint usw. Es hilft beim Identifizieren von Device-Namen und Partitionierungsstruktur.

#### Syntax:

lsblk [Optionen] [Gerät...]

#### Optionen:

- **-a, --all:** Zeigt auch leere bzw. unverwendete Geräte an (z.B. Geräte ohne Speicher, Loop ohne Backing). Standard lässt einige unwichtige aus.
- **-d, --nodeps:** Zeigt nur "Top-Level" Geräte (Disks) ohne Partitionen/Slaves.
- **-e <LIST>, --exclude <LIST>:** Schließt bestimmte Gerätetypen durch Major-Nummern oder Gerätekategorien aus (z.B. -e 7,11 um Loop und DVD zu ignorieren).
- **-f, --fs:** Zeigt zusätzlich Dateisystem-Informationen (Spalten FSTYPE, LABEL, UUID) . Ähnlich blkid, aber in lsblk integriert.
- **-p, --paths:** Zeigt vollständige /dev/Pfadnamen an statt nur Gerätamen (z.B. "/dev/sda" statt "sda").
- **-l, --list:** Ausgabe im Listenformat (eine Eintrag pro Zeile, ohne Baum-Struktur).
- **-J, --json:** Ausgabe als JSON-Struktur (maschinenlesbar).
- **-O, --output-all:** Zeigt alle verfügbaren Spalten an. Standardanzeige ist begrenzt. Mit -O sieht man z.B. auch MAJ:MIN, RM (removable), RO (read-only), MODEL, SERIAL etc.
- **-n, --noheadings:** Unterdrückt Überschrift.
- **-b, --bytes:** Größe in Bytes anzeigen (statt lesbar in KiB/MiB etc.).

- **-r, --raw:** Rohformat ohne Ausrichtung/Padding (für Skripting).

### **Beispielausgabe:**

```
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda 8:0 0 100G 0 disk
 ↘-sda1 8:1 0 512M 0 part /boot
 ↘-sda2 8:2 0 99.5G 0 part
 ↘-vg-root 253:0 0 90G 0 lvm /
 ↘-vg-swap 253:1 0 8G 0 lvm [SWAP]
sr0 11:0 1 1024M 0 rom (cdrom)
```

Hier sieht man Disk *sda* mit 100G und zwei Partitionen *sda1*, *sda2*. *sda2* ist ein PV (LVM Physical Volume) mit Volume Group vg -> daraus LVs *vg-root* und *vg-swap*, gemountet entsprechend.

### **lsscsi (List SCSI Devices)**

**Beschreibung:** Listet erkannte SCSI-Geräte (inkl. SATA, SAS, USB Storage – diese werden SCSI-Subsys emuliert) auf. **lsscsi** zeigt je Gerät Host/Channel/Target/LUN, dann Gerätetyp, und eventuell Model und Seriennummer je nach Infile-  
4fdk7rvx39azjretinarqb】 . Hilfreich, um z.B. alle angeschlossenen Festplatten und optischen Laufwerke anzuzeigen.

### **Syntax:**

**lsscsi** [Optionen]

### **Optionen:**

- **-v, --verbose:** Ausführlich – zeigt zusätzliche Infos (z.B. SCSI-Transport oder Zustände).
- **-t, --transport:** Zeigt SCSI-Transportinformationen (z.B. SATA, USB UAS, iSCSI etc.).
- **-L, --lunhex:** Zeigt die LUN in hexadezimaler Notation statt dezimal.
- **-H, --hosts:** Zeigt nur SCSI-Host-Adapter (Controller) an.
- **-c, --classic:** Benutzt "klassisches" Ausgabeformat (alter Stil, ohne Abkürzungen).
- **-g, --generic:** Fügt die entsprechenden sg-Device-Nodes hinzu (SCSI Generic /dev/sgX).

- **-s, --size:** Zeigt Größe (Kapazität) der Block-Geräte an, falls anwendbar (nur für Disk-type).
- **-p, --protection:** Zeigt Protection-Information (DIF/DIX).
- **-i, --scsi\_id:** Zeigt die SCSI-IDs im klassischen Sinn (T10 IDs).
- **-B, --usb:** Zeigt Bus-IDs (für USB-Geräte) mit an.

### **Beispiel:**

[0:0:0:0] disk ATA Samsung SSD 860 1B6Q /dev/sda 500GB

[2:0:0:0] cd/dvd TSSTcorp CDDVDW SH-224DB SB00 /dev/sr0

[3:0:0:0] disk WD MyBook 25DA 4004 /dev/sdb 4TB

Format: [H:C:T:L] type vendor model rev /dev/sgX -> /dev/sdX. Hier sieht man SCSI-Gerät auf Host 0 (SATA), Host 2 (DVD), Host 3 (USB-HDD).

### **fdisk (Partition Table Editor)**

**Beschreibung:** Interaktives Konsolen-Tool zum Anzeigen und Bearbeiten von Partitionstabellen auf Blockgeräten (MBR, GPT). Meistens wird fdisk verwendet, um schnell die Partitionierung eines Laufwerks anzusehen (fdisk -l) oder bei MBR-Partitionierung Partitionen einzurichten. Für GPT kann man gdisk oder parted benutzen (fdisk unterstützt aber inzwischen auch GPT).

### **Syntax:**

fdisk [Optionen] <Gerät>

Ohne Optionen startet es ein interaktives Menü für das angegebene Gerät (z.B. /dev/sda).

### **Optionen:**

- **-l, --list:** Listet die Partitionstabellen aller verfügbaren Laufwerke auf [4fdk7rvx39azjretinarqb]. Oder wenn ein Gerät angegeben, nur diese. Z.B. fdisk -l /dev/sda zeigt Partitionen auf sda.
- **-s Partition:** Gibt die Größe der angegebenen Partition in Sektoren aus (veraltet, blockdev --getsz moderner).
- **-u (veraltet):** Wechselt die Ausgabeeinheit (Sektoren vs Zylinder) in alter fdisk. Heutige fdisk (util-linux) nutzt standard 512-Byte Sektoren in -l.
- **-B, -C, -H, -S:** Kann man Geometrie (Zylinder, Heads, Sectors) angeben (MBR Legacy, heute irrelevant).
- **-t Typ:** Zeigt Partitionen in bestimmtem Format (Dos, GPT, ...).

Interaktiver Modus: Kommandos wie m (Hilfe), p (Print Partitionstabelle), n (neue Partition), d (löschen), t (Typ ändern, z.B. 83 Linux, 82 Swap, in GPT GUIDs), w (write und exit), q (quit ohne save).

### Beispiel (List):

Disk /dev/sda: 238.5 GiB, ...

| Device    | Boot | Start     | End       | Sectors   | Size   | Id | Type                 |
|-----------|------|-----------|-----------|-----------|--------|----|----------------------|
| /dev/sda1 | *    | 2048      | 534527    | 532480    | 260M   | 7  | HPFS/NTFS/exFAT      |
| /dev/sda2 |      | 534528    | 409600000 | 409065473 | 195.1G | 83 | Linux                |
| /dev/sda3 |      | 409600001 | 419430400 | 9830400   | 4.7G   | 82 | Linux swap / Solaris |

Zeigt z.B. 3 Partitionen (sda1 NTFS, sda2 Linux, sda3 Swap).

### parted (Partitionierungstool)

**Beschreibung:** Moderner Partitionierer, unterstützt GPT und MBR sowie Größe > 2TB. Kann sowohl interaktiv (im eigenen Prompt) als auch direkt per Kommandozeile genutzt werden. Im Gegensatz zu fdisk beherrscht parted auch **Resizing** von Partitionen und Filesystems (begrenzte FS-Typ Unterstützung) und kann in Skripten genutzt werden.

### Syntax (non-interactive):

parted [-s] <Gerät> <Befehl> [Parameter...]

Mit -s (script) ohne interaktive Ausgabe.

### Häufige parted Befehle: (Nicht alle Optionen, nur exemplarisch)

- **print** – Zeigt Partitionstabelle des Geräts. Z.B. parted /dev/sdb print.
- **mklabel <Typ>** – Erstellt neue Partitionstabelle (Typ: gpt, msdos, etc.). Achtung: löscht alle bestehenden Partitionen.
- **unit <Einheit>** – Setzt Maßeinheit (MB, GB, %, etc.) für folgende Befehle.
- **mkpart Name Typ Anfang Ende** – Erstellt Partition. Z.B.: mkpart primary ext4 1MiB 10000MiB erstellt eine ~10GB Partition vom 1MB Offset. *Typ* hier "primary"/"logical" (bei msdos) oder nur Partitions-TypName (bei GPT optional ein FS-Typ Label ext4 etc.).
- **rm <Nr>** – Löscht Partition mit Nummer.
- **resizepart <Nr> <Ende>** – Passt Größe der Partition Nr an (End-Sektor). Muss danach FS separat resized werden (z.B. mit resize2fs).
- **name <Nr> <Name>** – Setzt den Partitionsnamen (GPT Label).
- **toggle <Nr> boot** – Boot-Flag toggeln (bei MBR Partitionen).

- **align-check** *Typ Nr* – Überprüft Ausrichtung von Partition *Nr* (optisch/ Zylinder).

Parted kann auch einfache FS-Aufgaben: mkfs integration, aber meist nutzt man separate Tools (mkfs.\*).

### **Beispiele:**

- `parted /dev/sdc --script mklabel gpt mkpart primary ext4 0% 50% mkpart primary linux-swap 50% 100%` – Initialisiert /dev/sdc als GPT, teilt in zwei: erste Partition bis Mitte ext4, zweite Swap ab Mitte bis Ende.
- Im interaktiven parted (ohne -s) bekommt man (parted) Prompt zum Eingeben der obigen Befehle nacheinander.

### **mkfs (Make FileSystem)**

**Beschreibung:** Erzeugt ein neues Dateisystem auf einer Partition oder Festplatte. mkfs ist ein Frontend, das je nach -t <FSTYP> das entsprechende Tool (mkfs.ext4, mkfs.xfs, etc.) aufruft [file-4fdk7rvx39azjretinarqb]. Häufig ruft man direkt das spezifische Formatprogramm auf.

### **Syntax:**

`mkfs -t <FSTYP> [Optionen] <Geräte/Partition>`

Beispiele: `mkfs -t ext4 /dev/sdb1`, `mkfs -t vfat /dev/sdb1`.

### **Wichtige Optionen (allgemein):**

- **-t <typ>** – Dateisystemtyp (ext4, ext3, ext2, xfs, vfat, ntfs, etc.).
- **-V** – zeigt ausführliche Versions-/Debug-Information (welches Tool aufgerufen wird).
- **-q** – quiet (weniger Ausgabe).

Jedes Dateisystem hat eigene spezifische Optionen, wenn man direkt mkfs.ext4 etc. nutzt:

**ext4 (mkfs.ext4):** -L <Label>, -U <UUID> (benutzerdefiniert), -m <Proz> (Reserve-Blöcke Prozent, default 5%), -O <Feature> (ein/aus bestimmter FS-Features), -E stride= (RAID optimierungen), -b blocksize, etc.

**vfat (mkfs.vfat):** -n <Label>, -F <32|16> (FAT type), -I (Force if not partition), etc.

**xfs (mkfs.xfs):** -L <Label>, -m crc=1 (z.B. CRC enable, default ON neu), etc.

**swap (mkswap):** -L <Label>, (swap selbst ist via mkswap erstellt).

**Wichtig:** Partition sollte entsprechend markiert/Dimension formatiert sein (z.B. parted type=linux-swap für swap etc.).

### **Beispiele:**

- `mkfs.ext4 /dev/sda3` – Formatiert sda3 mit ext4 Standardparametern.
- `mkfs.vfat -F 32 -n "MYUSB" /dev/sdb1` – Formatiert Partition sdb1 als FAT32 mit Label "MYUSB".
- `mkswap /dev/sda5` – Initialisiert Swap auf sda5 (später via swapon).
- `mkfs.xfs -f -L data /dev/sdc1` – Erzwingt Format auf sdc1 mit XFS, Label "data" (XFS verlangt -f wenn Partition schon ein FS enthält).

## **mount (Dateisystem einhängen)**

**Beschreibung:** Bindet ein Dateisystem (Partition, Datenträger oder virtuelles FS) in den Verzeichnisbaum ein. mount ohne Argumente listet aktuelle Einhängungen. `mount <Gerät> <Verzeichnis>` hängt ein Gerät an den angegebenen Ordner ein (der Ordner muss existieren und leer sein, typischerweise unter /mnt oder /media oder definierter Mountpoint in /etc/fstabfile-4fdk7rvx39azjretinarqb】 .

### **Syntax:**

`mount [Optionen] <Quelle> <Zielverzeichnis>`

### **Wichtige Optionen:**

- **-t <fstype>** – gibt den Dateisystemtyp an (ext4, vfat, nfs, cifs, etc.). Kann oft weggelassen werden, da mount es automatisch erkennt (über blkid).
- **-o <Optionen>** – übergibt eine durch Komma getrennte Liste von Mount- Optionen-4fdk7rvx39azjretinarqb】 :
  - **defaults** (rw,suid,dev,exec,auto,nouser,async),
  - **ro/rw** (read-only / read-write),
  - **noexec** (keine Ausführung von Programmen erlauben), **nosuid** (SUID/SGID-Bits ignorieren), **nodev** (keine Gerätedateien interpretieren),
  - **sync** (synchrone IO), **async** (Standard, asynchron gepuffert),
  - **user/nouser** (Allow normal user to mount / nur root),
  - **uid=, gid=** (für FAT/NTFS, setze Besitzer der gemounteten Dateien),
  - **umask=, dmask=, fmask=** (Rechte-Masken für FAT/NTFS),
  - **iocharset=, codepage=** (für vfat),
  - **loop** (Loop-Device mount einer Datei als FS, evtl. -o loop oder auto),
  - **remount** (ein bereits gemountetes FS mit neuen Optionen versehen, z.B. `mount -o remount,rw /mnt/point`),

- FS-spezifisch: z.B. **acl**, **user\_xattr** (Ext4 Features), **compression**, **encrypt** (btrfs), etc.
- **-L <Label>, -U <UUID>** – anstatt Device-Node, per Label oder UUID identifizieren. `mount -U 1234-ABCD /mnt/usb`.
- **--bind** – führt einen Bind-Mount aus: mountet ein vorhandenes Verzeichnis an eine zweite Stelle im Verzeichnisbaum. Syntax: `mount --bind /pfad/quelle /pfad/ziel`. (Nützlich, um z.B. Verzeichnisse in Chroots sichtbar zu machen). Option **--rbind** rekursiv (inkl. Unterpunkte).
- **-a** – Mountet alles was in /etc/fstab als "auto" markiert ist und nicht noauto. (Ausführung typ. beim Boot).
- **-n** – führt Mount ohne Eintrag in /etc/mtab aus (mtab ist heute oft Symlink zu /proc/self/mounts). Selten benötigt.
- **-v** – verbose (gibt Meldungen aus).

### **Beispiele:**

- `mount /dev/sdb1 /media/usb -o uid=1000,gid=1000,umask=022` – hängt einen FAT32-Stick so ein, dass Benutzer mit UID/GID 1000 Besitzer ist (typisch eigener User), und Rechte 755 auf Dateien/Ordner (umask 022).
- `mount -t nfs 192.168.1.10:/export/share /mnt/share` – hängt eine NFS-Freigabe ein (NFS FS muss installiert sein).
- `mount -o loop diskimage.iso /mnt/iso` – mountet ein ISO-Image als Loop-Device unter /mnt/iso.
- `mount | grep sda` – sieht, ob /dev/sdax gemountet ist (ohne Parameter listet mount alles).

## **umount (Unmount FileSystem)**

**Beschreibung:** Hängt ein gemountetes Dateisystem wieder aus. Schreibvorgänge werden abgeschlossen und der Puffer geleert. Syntax kann entweder der Mountpoint oder das Device angegeben werden】 .

### **Syntax:**

`umount [Optionen] <Mountpunkt | Gerät>`

### **Wichtige Optionen:**

- **-f, --force**: Erzwingt das Aushängen eines FS. Sollte nur im Notfall (z.B. NFS-Server down) benutzt werden, da es evtl. Datenverlust verursachen kann.
- **-l, --lazy**: "Lazy unmount". Markiert das FS als ausgeworfen, aber falls noch offene Handles bestehen, wird es erst tatsächlich ausgehängt, wenn diese

geschlossen sind. (Hängt unmittelbar virtuell aus, lässt Rest im Hintergrund erledigen). Nützlich bei hängenden Prozessen.

- **-R, --recursive:** Hängt auch alle Unter-Mounts mit ab, die auf dem angegebenen Pfad liegen (rekursiv).
- **-v:** verbose.

*Beispiele:*

- `umount /media/usb` – hängt das unter `/media/usb` eingehängte Gerät aus.
- `umount /dev/sdb1` – selbes, angesteuert über Gerät.
- `umount -f -a -t nfs` – erzwingt das Aushängen aller NFS-Mounts (z.B. bei Netzproblemen).

(*Tipp: Wenn umount meldet "device is busy", kann man mit lsof oder fuser herausfinden, welcher Prozess noch auf das FS zugreift.*)

## **df (Disk Free)**

**Beschreibung:** Zeigt Übersicht des belegten und freien Speicherplatzes auf eingehängten Dateisystemen [4fdk7rvx39azjretinarqb]. df listet pro gemountetem FS die Gesamtgröße, verwendeten und freien Bytes sowie den Einhängepunkt.

### **Syntax:**

`df [Optionen] [Dateipfad...]`

Ohne Argumente zeigt alle gemounteten FS. Mit Pfad ermittelt df, auf welchem FS dieser Pfad liegt, und zeigt nur dessen Werte.

### **Optionen:**

- **-h, --human-readable:** Formatiert Größen in lesbarer Form (KiB, MiB, GiB) [4fdk7rvx39azjretinarqb]. Sehr gebräuchlich (Standard rohes Blockzahlen unhandlich).
- **-a, --all:** Schließt pseudo- oder 0-Size-Dateisysteme ein (wie proc, sysfs, etc.), die sonst ausgelassen werden.
- **-T, --print-type:** Zeigt Spalte mit Dateisystemtyp (ext4, tmpfs, nfs, etc.).
- **-i, --inodes:** Zeigt Inode-Auslastung statt Blockgrößen (frei/belegt Inodes pro FS). Nützlich, wenn "no space" durch Inode-Erschöpfung.
- **-t <typ>, -x <typ>:** Nur bestimmte FS-Typen anzeigen bzw. ausschließen. Z.B. `df -x tmpfs -x devtmpfs` um echte Disks anzuzeigen.
- **-B <Größe>:** Ausgabe in Blöcken von angegebener Größe. Z.B. `-B MB` (MByte) oder `-B 1M`.

- **--output[=FELDER]**: Benutzerdefinierte Spaltenausgabe. Z.B. df --output=source,fstype,size,used,avail,pcent,target.

### **Beispielausgabe (df -h):**

```
Filesystem Size Used Avail Use% Mounted on
/dev/sda2 50G 30G 18G 63% /
udev 3.9G 0 3.9G 0% /dev
tmpfs 798M 1.2M 797M 1% /run
/dev/sda1 300M 50M 250M 17% /boot
```

### **du (Disk Usage)**

**Beschreibung:** Summiert und zeigt die Größe von Verzeichnissen und Dateien an. du kann rekursiv durch Verzeichnisse gehen und den Platzverbrauch berechnen. Standardmäßig zeigt es für jedes Unterverzeichnis den Wert. Oft nutzt man du -sh <dir> um eine schnelle Gesamtgröße zu bekommen.

#### **Syntax:**

du [Optionen] [Pfad...]

Ohne Pfad = aktuelles Verzeichnis.

#### **Optionen:**

- **-h, --human-readable**: Größen in menschenlesbare Einheiten (K,M,G) ausgeben.
- **-s, --summarize**: Nur Summen pro angegebenem Argument ausgeben, *nicht* rekursiv für Unterordner. (Gibt also Gesamtgröße des Pfades).
- **-A, --apparent-size**: Rechnet mit logischen Dateigrößen statt belegtem Blockplatz. (Z.B. sehr sparsame Sparse Files würden sonst weniger anzeigen; --apparent-size zeigt nominelle Größe).
- **-d N, --max-depth=N**: Rekursionstiefe begrenzen. Z.B. du -h --max-depth=1 /var gibt Größe von /var und dessen Unterordner erste Ebene. (--max-depth=0 wäre wie -s).
- **-c, --total**: Am Ende eine Gesamtzeile "total" ausgeben (Summe aller ausgegebenen Einträge). Bei -s von mehreren Pfaden praktisch.
- **-x, --one-file-system**: Geht nicht in andere eingehängte Dateisysteme über. (Beschränkt auf Partition/FS der Startpfade).

- **-L, --dereference:** Folgt symbolischen Links (sonst ignoriert du Symlinks, rechnet sie als 0 Bytes). Vorsicht: Kann zu doppeltem Zählen führen, wenn Links in Kreis oder auf Oberverzeichnis zeigen.
- **--exclude=<PATTERN>:** Schließt Pfade aus, die zum Pattern passen. Z.B. du -sh -exclude="\*.git" oder mehrere --exclude.

### **Beispiele:**

- du -sh /home/\* – Zeigt Summe der Größen aller Benutzerverzeichnisse unter /home (jedes in human-readable).
- du -a /etc | grep -E "mysql.\*\.cnf\$" – Listet die Größe jeder einzelnen Datei unter /etc, filtert jene, deren Name auf mysql\*.cnf endet. (Option -a listet auch Dateien und nicht nur Verzeichnisse).
- du -d2 /var – Größe von /var, dessen Subdirs und Sub-Subdirs (Tiefe 2).

### **e2label (ext2/3/4 Volume Label)**

**Beschreibung:** Zeigt oder ändert das Label (Namen) eines ext2/ext3/ext4 Dateisystems. Das *Label* ist eine im Superblock gespeicherte Zeichenkette, die man z.B. in fstab zum Mounten verwenden kann (LABEL=<name>).

#### **Syntax:**

e2label <Gerät> [NeuesLabel]

Ohne zweites Argument zeigt es das aktuelle Label des ext-Volumes an. Mit Label schreibt es dieses in den FS.

#### **Beispiel:**

- e2label /dev/sda3 -> Ausgabe z.B. UbuntuRoot.
- e2label /dev/sda3 Archive -> setzt das Label von sda3 auf "Archive".

(Für andere FS gibt es analoge Tools: XFS nutzt xfs\_admin -L, FAT/NTFS mlabel oder dosfslabel/ntfslabel etc.)

### **resize2fs (ext2/3/4 Dateisystem verkleinern/vergrößern)**

**Beschreibung:** Passt die Größe eines ext2/3/4-Dateisystems an. Wird verwendet nach dem Ändern der Partitionsgröße (mit fdisk/parted). resize2fs kann im online-Modus vergrößern (wenn gemountet, ext3/4), fürs verkleinern muss das Volume ausgehängt und ein Filesystem-Check gemacht werden.

#### **Syntax:**

resize2fs [Optionen] <Gerät> [NeueGröße]

Neue Größe kann in K, M, G oder als Anzahl 4K-Blöcke angegeben werden. Lässt man sie weg, wird auf Maximum (Part.-Ende) vergrößert.

#### Optionen:

- **-p** – progress anzeigen (Fortschritt).
- **-f** – zwingen, auch wenn FS als nicht sauber markiert (normal will fsck vorher laufen bei Verkleinern).
- **-F** – flush caches vorher (sorgt für up-to-date Infos; Standard normal).
- **-M – minimal**: verkleinert das FS auf die kleinstmögliche Größe (alle Daten gerade passend). Äquivalent zu resize2fs <dev> -M (dauert Berechnung).

#### Beispiele:

- Nach Vergrößern der Partition: resize2fs /dev/sda2 – erkennt neue Partitionsgröße und wächst das ext4 FS auf vollen Platz.
- Verkleinern: umount /dev/sda2; e2fsck -f /dev/sda2; resize2fs /dev/sda2 20G – erst aushängen, fsck erzwingen, dann auf 20 GiB verkleinern. Anschließend Partition selbst verkleinern mit parted.

### tune2fs (Tune ext2/3/4 Filesystem Parameters)

**Beschreibung:** Ändert diverse Parameter eines bestehenden ext-Dateisystems: Reserveblöcke, Check-Intervalle, Label, Journal einschalten, usw.

#### Syntax:

tune2fs [Optionen] <Gerät>

#### Häufige Optionen:

- **-l** – Listet alle Superblock-Parameter (Filesystem-Info) auf [freedesktop.org](http://freedesktop.org). Zeigt z.B. Inode-Größe, Features, Letzter Mount, Nächster geplanter Check, etc. (Read-only Operation).
- **-c <MaxMounts>** – Setzt die maximale Anzahl Mount-Vorgänge bis zur erzwungenen fsck-Prüfung. Z.B. -c 0 = deaktiviert mount-count-basierten Check.
- **-i <Interval>** – Setzt das Zeit-Intervall für Zwangs-FSCK. z.B. -i 30d = alle 30 Tage. -i 0 = kein zeitbasierter Check.
- **-r <AnzRes>** – Setzt Anzahl der reservierten Blöcke (für root). Standard 5% bei ext4 auf <1TB. Kann man z.B. auf 1% reduzieren auf großen Volumes, z.B. tune2fs -r 0 /dev/sdb1 um Reserven komplett abzuschalten (nicht empfohlen auf /). Alternativ -m <Prozent> (z.B. -m 1).
- **-L <Label>** – Ändert Label (wie e2label).

- **-U <UUID>** – Ändert die UUID (zufällig oder angegeben). (Vorsicht: fstab etc. müssen angepasst).
- **-J** – Journal-Optionen: z.B. -O has\_journal um Journal hinzuzufügen (macht aus ext2 -> ext3), oder -O ^has\_journal Journal entfernen (ext3->ext2), oder -j Kurzoption (Standardwerte Journal hinzufügen).
- **-O <FeatListe>** – Aktiviert (feature) oder deaktiviert (^feature) Filesystem-Features. Z.B. -O casefold (für ext4 ab 5.2: Case-Insensitive-Folder Support) oder ^huge\_file (Feature abschalten – Achtung, Daten verlieren, nur wenn FS leer). Nur bestimmte Features togglebar, meist erfordert offline und vollständiges Backup.
- **\*\*-E <extendopt>\*\*** – Erweiterte Einstellungen: z.B. stripe=<StripeSize>(RAID stripe in Blöcken),lazy\_itable\_init=1` (schnell format, Inode-Table lazy Clearing).
- **-M <Verzeichnis>** – Markiert FS als gemountet an <Verzeichnis> (setzt den Mountpoint in Superblock – meist rein informativ).

### **dumpe2fs (Dump ext2/3/4 Superblock)**

**Beschreibung:** Gibt ausführliche Informationen über ein ext2/3/4-Dateisystem aus – Inhalt des Superblocks und der Block Group Deskriptoren . Zeigt ähnlich tune2fs -l Informationen, plus Aufteilung der Blöcke in Gruppen, freie Blöcke/Inodes pro Gruppe etc. Hilfreich für Diagnose (z.B. Backup-Superblock Positionen).

#### **Syntax:**

dumpe2fs [Optionen] <Gerät>

#### **Optionen:**

- **-h** – Zeigt nur den Inhalt des Superblocks und der Gruppen-Deskriptoren (ohne die Liste aller freien Blöcke/Inodes). Das ist die gängigste Verwendung, da ohne -h die Ausgabe sehr lang sein kann (listet freie Blöcke/Inodes aller Groups[wiki.ubuntuusers.de](http://wiki.ubuntuusers.de)).
- **-x** – Zeigt auch die detaillierten Gruppen-Informationen (wie freie Blöcke/Inodes-Bitmap). Standard ohne -h tut das sowieso.
- **-b** – Listet die Nummern der freien Blöcke. (Sehr viel Output auf großem FS.)
- **-i** – Listet die freien Inode-Nummern.

#### **Beispiel (gekürzt dumpe2fs -h /dev/sda2):**

Filesystem volume name: <none>

Last mounted on: /

Filesystem UUID: 4e8fea21-...-b6c7

Filesystem magic number: 0xEF53

Filesystem revision #: 1 (dynamic)

Filesystem OS type: Linux

Block size: 4096

Block count: 25600000

Reserved block count: 1280000

Free blocks: 15034567

First Block: 0

...

Journal inode: 8

Journal backup: inode blocks

...

Zeigt Label, UUID, Blocksize, Anzahl Blöcke, Reserve, Anzahl freie etc., und welche Features aktiviert sind.

### **cryptsetup (LUKS Disk Encryption Setup)**

**Beschreibung:** Tool zur Einrichtung von **verschlüsselten Datenträgern** mit LUKS (Linux Unified Key Setup) sowie zum Öffnen/Schließen solcher Verschlüsselungs-Volume】 . cryptsetup kann ein Gerät mit LUKS initialisieren, Passphrase setzen, und im Betrieb ein verschlüsseltes Volume als Device-Mapper-Gerät bereitstellen, das dann gemountet werden kann.

#### **Typische Aktionen (Subcommands):**

- **luksFormat** – Initialisiert ein Blockgerät mit LUKS-Verschlüsselungsheader und fordert eine Passphrase zum Einrichten an. Beispiel: cryptsetup luksFormat /dev/sdb1. *Achtung:* Löscht alle vorhandenen Daten auf dem Gerät.
- **open** oder **luksOpen** – Öffnet ein verschlüsseltes Volume (fragt Passphrase). Syntax: cryptsetup luksOpen /dev/sdb1 geheim\_vol. Dies erstellt ein Device /dev/mapper/geheim\_vol, das nun entschlüsselt agiert und z.B. mit mkfs formatiert oder gemountet werden kan】 .
- **close** – Schließt ein vorher geöffnetes LUKS-Device: cryptsetup close geheim\_vol (dann ist /dev/mapper/geheim\_vol weg).
- **luksAddKey** – Fügt einen weiteren Schlüssel (Passphrase) hinzu. LUKS unterstützt mehrere Key-Slots. cryptsetup luksAddKey /dev/sdb1

/pfad/zur/schlüsseldatei oder interaktiv eine neue Passphrase eingeben (nachdem man eine gültige alte eingibt).

- **luksRemoveKey** – Entfernt einen Schlüssel (kann durch Angabe der zu entfernenden Passphrase oder Slot-Nummer erfolgen).
- **luksChangeKey** – Ändert eine bestehende Passphrase (in einem Slot) durch eine neue (kombi aus Remove+Add effizienter).
- **luksHeaderBackup / luksHeaderRestore** – Backup bzw. Wiederherstellen des LUKS-Headers (empfohlen, falls Header korrupt -> Daten verloren).
- **status** – Zeigt Informationen über ein offenes Device-Mapper Volume (Name, verschlüsseltes Dev, Cipher, Keylen, etc.): cryptsetup status geheim\_vol.

### Wichtige Optionen:

- **-y** (im Format-Befehl): fragt neues Passwort zweimal zur Verifizierung ab.
- **-d <Datei>**: verwendet eine Datei als Schlüsseldaten (anstatt interaktiv Passwort). -d kann genutzt werden, um vom stdin zu lesen.
- **--key-size <bits>**: Schlüsselgröße (in Bits) für die Verschlüsselung (z.B. 256).
- **--cipher <name>**: Zu nutzender Verschlüsselungsalgorithmus und Mode. Standard z.B. aes-xts-plain64 bei LUKS2. Kann z.B. aes-cbc-essiv:sha256 oder anderes angegeben werden.
- **--hash <hash>**: Hash-Algorithmus für Passphrase-Derivation (PBKDF). Standard modern ist z.B. Argon2id oder früher sha256.
- **--iter-time <ms>**: Wie lange (Millisek) auf PBKDF verwendet werden soll (Bestimmt Komplexität).
- **--type luks1/luks2**: LUKS Version festlegen (aktuell Standard LUKS2, kann optional LUKS1 für Kompatibilität gewählt werden).
- **-q**: Quiet (unterdrückt Ausgabe, z.B. Warnung bei Format).

### Beispiel Workflow:

1. **Formatieren**: cryptsetup -y luksFormat /dev/sda5 (legt LUKS auf Partition sda5 an, fragt 2x Passwort).
2. **Öffnen**: cryptsetup luksOpen /dev/sda5 geheim (Passwort eintippen, erstellt /dev/mapper/geheim).
3. **Format FS**: mkfs.ext4 /dev/mapper/geheim.
4. **Mount**: mount /dev/mapper/geheim /mnt/secure. Jetzt Daten drauf speichern.
5. **Aushängen+Schließen**: umount /mnt/secure && cryptsetup close geheim.

(LUKS ermöglicht bis zu 8 Passphrases; man kann z.B. mit luksAddKey einen zweiten hinzugeben, sodass entweder Passphrase A oder B das Volume öffnet. Das Gerät /dev/sda5 kann in /etc/crypttab und /etc/fstab eingetragen werden für Boot-Time auto-open via keyfile oder prompt.)

## shred (Securely wipe file/device)

**Beschreibung:** Überschreibt eine Datei oder ein ganzes Gerät mehrfach mit zufälligen Daten, um ein Wiederherstellen gelöschter Inhalte zu erschweren. shred ist nützlich zum sicheren Löschen auf magnetischen HDDs. Bei SSDs mit Wear-Leveling ist Effekt begrenzt – dort lieber ATA Secure Erase oder physical destruction.

### Syntax:

```
shred [Optionen] <Datei1> [Datei2...]
```

### Optionen:

- **-n N** – Anzahl der *zusätzlichen* Überschreib-Pässe (Standard 3). D.h. 3 bedeutet insgesamt 3 mal random + final pass (siehe -z).
- **-z, --zero** – Überschreibt am Ende noch einmal mit Null-Bytes, um das Muster zu verwischen (anstatt dass zuletzt zufälliges Rauschen bleibt). Standard: *aktiv* (bei neueren shred ist -z default an).
- **-f, --force** – Ändert Schreibschutz falls nötig (chmod u+w), um schreiben zu können.
- **-u, --remove[=how]** – Nach dem Überschreiben die Datei löschen (unlink). Optional how kann unlink (Standard) oder wipe oder wipesync sein: *wipe* überschreibt erst mit Null dann löscht, *wipesync* macht zusätzlich Sync. Standard -u bereits tut wipesync.
- **-v, --verbose** – Auskunft über Fortschritt.
- **-x, -X, --exact** – Überschreibt Datei genau Byte-für-Byte anstelle Dateigröße auf nächste Blockgröße aufzurunden (Standard shred nutzt ganze Blocks, kann über Dateiende hinaus schreiben falls Dateigröße nicht mod Blocksize). *Exact* vermeidet das, aber hinterlässt evtl. un-geschredderte Slack am Ende.
- **-s N, --size=N** – Behandelt die Datei als N Byte lang (statt aktuelle Länge). Um auch ungenutzten Platz (z.B. Slack Space/EOF beyond) im Inode zu überschreiben, kann man etwas größer angeben.

### Beispiele:

- shred -u file.txt – Überschreibt *file.txt* viermal (3 rand + 1 zero) und löscht sie anschließend.

- shred /dev/sdb – Überschreibt komplett /dev/sdb mit 3 Runden Zufallsdaten (zerstört alle Daten/Partitionen darauf).
- shred -n 0 -z -u secret.dat – Einmal mit Nullen überschreiben und entfernen (hier wollte man evtl. nur einfaches Überschreiben und löschen).

**Warnung:** Shred wirkt nicht zuverlässig auf: **logischen Volumes** in Files (z.B. Datei in FS kann redundante Kopien (Journal, Snapshots) haben), **RAID, SSD, Netzwerkdateisysteme, Dateisysteme mit Copy-on-Write (btrfs, ZFS)** – dort können Daten an mehreren Orten stehen. In solchen Fällen besser Vollverschlüsselung verwenden oder das gesamte physische Medium überschreiben.

### iostat (IO Statistics)

**Beschreibung:** Zeigt Statistik zur Systemauslastung von CPU und Input/Output für Blockgeräte (Datenträger). iostat gehört zum Paket sysstat. Es hilft zu sehen, wie stark Platten ausgelastet sind (%util) und Durchsatz (MB/s), IOPS etc., sowie generelle CPU-Idle/Wait Zeite .

#### Syntax:

iostat [Optionen] [Intervall [Durchläufe]]

Ohne Argument gibt einen Durchschnitt seit Boot aus. Mit Intervall (Sekunden) aktualisiert regelmäßig.

#### Wichtige Optionen:

- **-c** – Zeigt nur CPU-Statistiken.
- **-d** – Zeigt nur Disk (Geräte) Statistiken.
- **-k, -m** – Ausgabe in KiB/s oder MiB/s (anstatt Blöcke/s). (Aktuelle sysstat erlaubt -block-size).
- **-p [Gerät]** – Detaillierte Partitions-Statistiken einbeziehen (normal zeigt iostat nur Whole-Device). Z.B. -p sda zeigt sda und sda1,sda2,...
- **-x** – Ausgabe in erweiterter Form (Pro Gerät: zusätzlich Auslastungsprozentsatz, Average Request Size, Await Times etc.) . Sehr nützlich, da Standard ohne -x recht knapp.
- **-t** – Zeigt Timestamp vor jeder Ausgabe.
- **-y** – Unterdrückt erste Ausgabe (since boot), beginnt erst mit Werten ab erstem Intervall.

#### Interpretation (bei iostat -x):

Spalten pro Device: rrqm/s wrqm/s (Merge von I/O, veraltet bei modernen Kernel meist 0), r/s w/s (reads/writes per sec), rkB/s wkB/s (Throughput), avgrq-sz (Durchschnitl.

Requestgröße in Sektoren oder KB), avgqu-sz (durchschn. Queue length), await (Durchschnittliche Wartezeit pro IO ms), svctm (durchschn. Servicezeit pro IO ms – oft nicht mehr zuverlässig), %util (Prozentuale Zeit, die das Gerät aktiv war innerhalb der Messperiode – 100% bedeutet permanent beschäftigt).

### Beispiel:

```
Device r/s w/s rkB/s wkB/s avgrq-sz avgqu-sz await svctm %util
sda 12.5 30.2 500.0 250.0 40.0 0.5 10.0 2.00 85.0
```

Hier sda ~85% ausgelastet, wartet ~10ms per IO, etwa 42.7 IO/s (r+s) mit ~750kB/s.

### ioping (I/O Ping)

**Beschreibung:** Ermittelt Latenz (und ggf. Durchsatz) von Storage in Echtzeit ähnlich dem Prinzip von ping im Netzwerk. Es führt schnelle sequenzielle oder wahlfreie Lesezugriffe durch und misst deren Zeit. Praktisch um Latenz auf Disks oder bestimmten Verzeichnissen (Dateisystem) festzustellen.

### Syntax:

```
ioping [Optionen] <Ziel>
```

Ziel kann ein Pfad (Datei oder Verzeichnis) sein für Filesystem IO, oder z.B. ein Blockgerät (mit -D für Direkten Device Zugriff) oder sogar ein bestimmter I/O-Engine (IO depth).

### Wichtige Optionen:

- **-c <count>** – Anzahl der Ping-Requests (Standard unendlich, bis abgebrochen).
- **-i <interval>** – Wartezeit zwischen Pings (Sekunden, default 1s). Kann in ms angeben z.B. -i 0.1.
- **-s <bytes>** – Größe der zu lesenden Daten pro Ping (Standard 4KiB).
- **-R** – Ausgabe im Rohformat (Maschinenlesbar).
- **-D** – Direct-Device Mode: Arbeitet auf Blockgeräte direkt (bypasses FS-Cache).
- **-A** – Ausgabe als Aggregatstatistik nur (kein laufendes Ping, nur Summary).
- **-S** – Sync/Sequenziell: testet sequenzielles Lesen statt wahlfrei. (Standard ioping macht random position reads).
- **-W** – führt Schreib-Tests durch statt Lesen (Achtung kann Daten überschreiben zerstören, am besten auf extra Gerät/Datei).
- **-P <N>** – Parallel Requests (Thread count).
- **-q** – keine laufende Ausgabe, nur Summary am Ende.

### **Beispiele:**

- ioping . – Sendet kontinuierlich 4K-Leseanfragen ins aktuelle Verzeichnis (random in FS), zeigt Latenzen, bis Ctrl+C.

4096 bytes from . (ext4 /dev/sda2): request=1 time=0.47 ms

4096 bytes from . (ext4 /dev/sda2): request=2 time=0.44 ms

... (zeigt typisch ~0.5ms auf SSD, ~10-15ms auf HDD) ...

Nach Abbruch gibt es eine **Summary** mit min/avg/max/mdev (Jitter).

- ioping -c 10 /mnt/backup – Misst 10 Zugriffe im FS /mnt/backup (z.B. externer USB), dann stop.
- ioping -R /dev/sdb – Führt 1 Sekunde lang Messungen auf /dev/sdb durch und gibt am Ende aggregated IOPS und Bandbreite.
- ioping -A -S -c 50 /file.iso – Liest sequenziell 50 Anfragen durch große Datei und gibt am Ende Durchschnittszeit.

(Tool muss ggf. installiert werden, gehört nicht standard zu allen Distr.)

## **Geräte, Prozesse, Speicher und Kernel**

### **hwinfo (Hardware Info)**

**Beschreibung:** Liefert detaillierte Informationen zur Hardware des Systems. hwinfo probiert, alle verfügbaren Hardware-Komponenten aufzulisten: CPU, RAM, Festplatten, USB, PCI-Geräte, Netzwerk, Monitor, etc., inklusive ihrer Eigenschaften und Treiber]. Oft auf openSUSE installiert, aber auch auf anderen Distros verfügbar.

#### **Syntax:**

hwinfo [Optionen] [Hardwareklasse]

#### **Optionen / Klassen:**

- Ohne Angabe gibt hwinfo *alles* aus (sehr lang). Besser, man filtert nach Klasse: z.B. hwinfo --cpu, hwinfo --disk, --gfxcard, --netcard, --printer, --usb, --bios, --memory etc.
- **--short:** Kurzfassung ausgeben (ein Eintrag pro Gerät in einer Zeile).
- **--brief:** Noch knapper (ähnlich short).
- **--summary:** Summaries (z.B. nur Zahl der Disks, CPUs etc.).
- **--verbose <level>:** Detaillierungsgrad (0-9). Default 1 (schon sehr ausführlich).
- **--debug N:** Setzt Debug-Level (für Entwickler, welche Module hwinfo benutzt).

- **--hwitem <ID>**: Sucht nach Hardware mit bestimmter ID (z.B. PCI-ID).
- **--save-config <datei> / --load-config <datei>**: Speichert erkannte Hardware in Datei zum späteren Vergleich.

*Beispiel:* hwinfo --cpu könnte ausgeben:

```
01: None 00.0: 10103 CPU
[Created at cpu.462]
Unique ID: rdCR.jX8Un+o2ZT1
Hardware Class: cpu
Arch: X86-64
Vendor: "GenuineIntel"
Model: 6.158.10 "Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz"
... (viele weitere Details Frequenzen, Cache etc.) ...
```

hwinfo --short summarisiert z.B. Hauptgeräte und Partitionen.

### **dmidecode (DMI Table Decode)**

**Beschreibung:** Gibt Informationen aus dem BIOS/UEFI-DMI (SMBIOS) heraus, wie Hersteller, Produktnamen, Seriennummern von System/Mainboard, BIOS-Version, verbauter Speicher, Anschlüsse etc]. Hilfreich für Hardware-Inventarisierung (Seriennummer, RAM-Module, OEM strings).

#### **Syntax:**

dmidecode [Optionen]

#### **Optionen:**

- **-t <string|Nummer>, --type**: Nur bestimmte DMI-Typen ausgeben. Typ kann Name oder Zahl sein. Z.B.:
  - bios (DMI type 0),
  - system (1),
  - baseboard (2),
  - processor (4),
  - memory (Memory Device, 17),
  - cache (7),

- connector (8), slot (9), etc.  
Man kann -t memory um nur RAM-Infos zu sehen (Größe, Typ, Hersteller, Seriennummer der DIMMs) .
- **-s <keyword>, --string:** Einzelne SMBIOS-Einträge gezielt ausgeben (z.B. bios-vendor, bios-version, system-manufacturer, system-serial-number, baseboard-asset-tag, processor-frequency etc.).
- **-q:** Quiet (unterdrückt weniger wichtige DMI-Eintrags-Labels wo "Not Provided").
- **-u:** Raw dump (hex) der DMI-Daten.
- **--dump-bin file, --from-dump file:** Dump SMBIOS in Datei bzw. lese aus Dump (z.B. offline Analyse).

*Beispiel:* dmidecode -t system könnte zeigen:

System Information

Manufacturer: Dell Inc.

Product Name: Latitude 5490

Version: None

Serial Number: ABCD123

UUID: 4c4c4544-0034-5810-8055-b6c04f513332

Wake-up Type: Power Switch

SKU Number: 07E6

Family: Latitude

dmidecode -t memory zeigt pro RAM-Riegel z.B. Größe, Form Factor (SODIMM), Type (DDR4), Speed, Manufacturer, Serial, Part Number.

### **lspci (List PCI Devices)**

**Beschreibung:** Zeigt alle PCI/PCIe-Geräte im System (angeschlossene Hardware am PCI-Bus: z.B. Netzwerkkarten, Sound, USB-Controller, Grafikkarten, SATA-Controller etc.file-4fdk7rvx39azjretinarqb) . Es listet Geräte-ID und -Namen anhand einer Datenbank (/usr/share/misc/pci.ids).

### **Syntax:**

**lspci [Optionen]**

### **Optionen:**

- **-v, -vv, -vvv:** Verbose-Ausgabe in ein oder mehreren Stufen (mehr Register-Informationen). -vvv kann sehr ausführlich sein (inkl. Capabilities, Konfigspace).

- **-k:** Zeigt Kernel-Treiber und Module, die für jedes Gerät verwendet werden (welcher Treiber zugeordnet ist file-4fdk7rvx39azjretinarqb]. (Z.B. Kernel driver in use: e1000e für Intel NIC).
- **-t:** Zeigt die hierarchische Baumstruktur der PCI-Bus-Topologie (wer hängt an welchem Bus/Bridge).
- **-nn:** Zeigt neben dem Gerätenamen auch Vendor:Device ID in eckigen Klammern. -n nur numeric IDs ohne Namen. -nn= beide. Z.B. Ethernet Controller [8086:1533] etc.
- **-d [<vendor>:<device>]:** Filtert nach bestimmten PCI IDs. lspci -d 8086: zeigt nur Intel-Geräte, -d :1533 nur Geräte mit Device 1533.
- **-s [[<domain>]:]<bus>:<slot>[.<func>]:** Filtert nach Bus-Adresse (wie in lspci -t oder /sys/bus/pci). Z.B. -s 00:1f.2 zeigt nur dieses Gerät.
- **-m:** Maschinenlesbar (gibt in einer CSV-artigen Formatzeile pro Gerät: Domain Bus Slot, Klass, Vendor, Device, SVendor, SDevice).
- **-xxx:** Dump des gesamten PCI-Konfigurationsspeichers (64 Bytes beyond standard) – Vorsicht sensibel. -xx nur ersten 256 Bytes (std config).
- **-H1:** Nutzen von /proc/bus/pci statt direct (historisch, meist nicht nötig).
- **-Q:** Zeigt das Querier-Modell (Kernel vs direct vs /proc).

#### **Beispiel Ausgabe (gekürzt):**

00:00.0 Host bridge: Intel Corporation Device 5914

00:02.0 VGA compatible controller: Intel Corporation UHD Graphics 620

00:14.0 USB controller: Intel Corporation Sunrise Point-LP USB 3.0 xHCI Controller (rev 21)

00:16.0 Communication controller: Intel Corporation Management Engine Interface (rev 21)

...

03:00.0 Network controller: Intel Corporation Dual Band Wireless-AC 8265 (rev 78)

Mit -nn würde man z.B. sehen 00:02.0 VGA controller [0300]: Intel Corporation UHD Graphics 620 [8086:5917]. Mit -k ergänzt: Kernel driver in use: i915.

#### **lsusb (List USB Devices)**

**Beschreibung:** Listet USB-Geräte auf, die am System angeschlossen sind (inkl. interne Hubs etc.). Zeigt Vendor und Produktnamen falls bekannt (Datenbank /usr/share/misc/usb.ids) und ID.

## Syntax:

lsusb [Optionen]

## Optionen:

- **-v**: Verbose, zeigt detaillierte Descriptor-Infos pro Gerät (ähnlich lsusb -vv noch detaillierter). Das kann sehr lang sein.
- **-t**: Baum-Darstellung der USB-Geräte-Hierarchie (welches Device hängt an welchem Port/Hub) – ähnlich lspci -t.
- **-s <bus>:<dev>**: Zeigt nur spezifisches Gerät (per Bus und Device Nummer). Diese Nummern sieht man in lsusb Standardausgabe vorn.
- **-d vid:pid**: Filtert nach VendorID:ProductID (Hex).
- **-D <device-file>**: Liest Infos aus einer Geräte-Datei (z.B. /dev/bus/usb/001/007) – erfordert Root.
- **-w**: Ausgabe als mit usbdump kompatibler Text (Wire format).
- **-v -v (or -vv)** liefert noch detailliertere Dump (inkl. Hexdump der HID Deskriptoren etc.).

## Beispiel Ausgabe:

Bus 002 Device 003: ID 0bda:0316 Realtek Semiconductor Corp. Card Reader

Bus 002 Device 002: ID 8087:0aaa Intel Corp.

Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub

Bus 001 Device 004: ID 0c45:671f Microdia Integrated Webcam

Bus 001 Device 003: ID 8087:07dc Intel Corp. Bluetooth wireless adapter

Bus 001 Device 002: ID 056e:4008 Elecom Co., Ltd

Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

Hier z.B. Card Reader, Webcam, Bluetooth, eine Maus (056e:4008 Elecom), plus Root Hubs.

Mit -t etwa:

/: Bus 02.Port 1: Dev 1, Class=root\_hub, ... (bus 2, 3.0)

  |\_\_ Port 1: Dev 2, Intel USB 3.0 Hub

  |\_\_ Port 2: Dev 3, Realtek Card Reader

...

Zeigt Topologie (Hub etc.).

### **lscpu (List CPU Info)**

**Beschreibung:** Zeigt Informationen über die CPU(s) des Systems – Anzahl Kerne, Threads, Sockets, Modellname, Frequenz, Virtualisierungsfunktionen uswfile-4fdk7rvx39azjretinarqb】.

#### **Syntax:**

`lscpu [Optionen]`

#### **Optionen:**

- **-e, --extended [=list]:** Ausgabe als Tabelle mit Spalten (erweiterte Auflistung jeder logischen CPU mit Eigenschaften). Optional kann man angeben, welche Felder (z.B. CPU, Node, Socket, Core, Online, ...).
- **-p, --parse [=list]:** Ähnlich --extended aber kommagetrennt (parseable). Standardfelder: CPU,Core,Socket,Node.
- **-x, --hex:** Zeigt hex-Werte wo sinnvoll (z.B. CPU bitmasken).
- **-c, --offline / --online:** Zeigt nur offline oder nur online CPUs (bei Hotplug-Systemen).
- **-y, --physical:** Zeigt nur physische IDs (keine virtuellen IDs).
- **-J, --json:** JSON-Ausgabe (maschinenlesbar).
- **-B, --byte:** Einheit Bytes statt kB für Cache-Größen.
- **-C, --cache:** Zeigt Details zu CPU-Caches (Anzahl, Größen).

*Beispiel (gekürzt lscpu Standard):*

Architecture: x86\_64

CPU op-mode(s): 32-bit, 64-bit

Sockets: 1

Cores per socket: 4

Threads per core: 2

Model name: Intel(R) Core(TM) i7-8565U @ 1.80GHz

CPU MHz: 2112.004

Virtualization: VT-x

L1d cache: 32K

L2 cache: 256K

L3 cache: 8192K

Flags: fpu vme ... vmx ...

Zeigt also 1 Sockel, 4 Kerne, 8 Threads (4\*2), CPU-Modell und Takt etc.

### **lsmem (List Memory Blocks)**

**Beschreibung:** Zeigt Speicherblock-Informationen des Systems (insbesondere auf Systemen mit Hotpluggable Memory oder NUMA). Es listet Memory-Block-Geräte (wie in /sys/devices/system/memory) auf. Auf x86-PCs weniger relevant; auf IBM System z oder Power etc. nützlich.

#### **Syntax:**

lsmem [Optionen]

#### **Optionen:**

- **-n, --online / -o, --offline:** Zeigt nur online bzw. offline Speicherblöcke.
- **-b, --bytes:** Größen in Bytes statt lesbar.
- **-M, --mega:** Größen in MiB. (Standard lsmem gibt in bytes + kMG suffix)
- **-p, --pairs:** Ausgabe im Key:Value Format (für maschinelles Parsing).
- **-J, --json:** JSON-Format.
- **--output <list>:** Spalten auswählen (z.B. --output SIZE,NODE,STATE).
- **--help, --version:** analog.

*Beispiel lsmem (auf PC ohne Hotplug memory):*

| RANGE                                 | SIZE | STATE  | REMOVABLE | BLOCK |
|---------------------------------------|------|--------|-----------|-------|
| 0x0000000000000000-0x000000007fffffff | 2G   | online | yes       | 0-15  |
| 0x0000000100000000-0x000000027fffffff | 6G   | online | yes       | 32-79 |

Zeigt 2 memory ranges (entsprechend wohl 2 DIMMs). "Removable yes" heißt vom Kernel als Hot-remove möglich (oft 'no' auf Standard-PC). "BLOCK 0-15" referenzieren Memory block device IDs.

Auf NUMA-Systemen würde Spalte NODE relevant sein, um Memory pro NUMA-Node zu sehen.

### **ps (Process Status)**

**Beschreibung:** Zeigt eine Momentaufnahme der aktuellen Prozesse des Systems. ps kann verschiedene Ausgabeformate haben (BSD-Style, UNIX-style). Häufig nutzt man ps aux oder `ps -effile-4fdk7rvx39azjretinarqb` .

### Syntax:

ps [Optionen]

(Komplexität: ps hat UNIX Optionen (mit - und bündelbar), BSD Optionen (ohne - und können zusammen geschrieben werden) und GNU long Optionen. Z.B. ps aux ist BSD Syntax, ps -ef ist UNIX syntax – beide gängig, liefern ähnliches.)

### Gängige Aufrufe:

- **ps aux** – Listet alle Prozesse mit Details (BSD format). Spalten: USER, PID, %CPU, %MEM, VSZ, RSS, TT (TTY), STAT, START, TIME, COMMAND.
- **ps -ef** – Listet alle Prozesse (UNIX format). Spalten: UID, PID, PPID, C (CPU usage), STIME (start time), TTY, TIME (CPU time), CMD.
- **ps -eF** – wie -ef, aber "extra full", zeigt threads als separate Linien und vollständige Kommandozeilen etc.
- **ps -ejH** – zeigt Hierarchie (Prozessbaum) mit Prozess-IDs.
- **ps -eo pid,cmd,%mem,%cpu --sort=-%mem** – Zeigt benutzerdefinierte Spalten (hier pid, Befehl, Speicher- und CPU% sortiert nach Speicherverbrauch absteigend).

### Wichtige Optionen:

UNIX-style (präfix mit -, können kombiniert werden):

- **-e** – Alle Prozesse anzeigen (entspricht -A).
- **-f full** Format (mehr Spalten inkl. PPID, STIME, in Kombination z.B. -ef).
- **-F** – Extra full (noch mehr Infos: SZ, RSS, PSR CPU, etc.).
- **-h** – Ohne Header.
- **-H** – Hierarchie-Tree anzeigen (Indentation für Parent-Child).
- **-j** – Jobs format (PGID, SID etc. Spalten).
- **-l** – Long format (mehr spalten inkl. Flags, NI nice, LWP etc.).
- **-u <user>** – Prozesse eines bestimmten Benutzers.
- **-p <pidlist>** – Nur bestimmte PIDs.
- **-G <group>, -U <user>** – nach Gruppe/Benutzer filtern.

BSD-style (ohne -, einzeln hintereinander):

- **a** – Zeigt Prozesse von *allen Terminals* (nicht nur eigene).
- **x** – Zeigt auch prozesse *ohne Terminal* (Daemons). **ax** zusammen = alle Prozesse.
- **u** – Benutzerformat (inkl. Owner, CPU%, MEM%). => **aux** verbreitet.
- **j** – Jobs format (session, pgrp).
- **w** – breites Output (alle columns of command, use -ww to not truncate).

GNU long options (mit --):

- **--pid, --ppid, --user** etc. Alternative Filter.
- **--sort=<key>**: Sortiert Ausgabe nach Spalte(n). Z.B. --sort=-%cpu (absteigend CPU).
- **--no-headers**: Ohne Header (statt -h).
- **--forest**: ASCII-Art Bäumchen (ähnlich -H, aber macht "|\_" und Einrückungen). Kombiniert mit e.g. -ef.

### **STAT Spalte (Prozessstatus):**

Codes: R (running), S (sleep), D (uninterruptible sleep, i/o), Z (zombie), T (stopped), t (gestoppt im Hintergrund), X (dead). Flags dahinter: < (high prio), N (low prio, nice>0), L (pages locked in mem), s (session leader), + (foreground process group in TTY).

## **sar (System Activity Reporter)**

**Beschreibung:** Teil von sysstat (wie iostat, mpstat, etc.), sammelt und berichtet historische System-Auslastungsdaten: CPU, Speicher, Paging, I/O, Netzwerk etc. sar kann *laufende* Stats anzeigen (wenn mit Interval aufgerufen), oder *vergangene Tageswerte* aus Logs (/var/log/sysstat/...) ausgeben.

### **Syntax:**

`sar [Optionen] [Interval [Count]]`

Ohne Parameter, sar liest per default die Tagesstatistiken (cumulated) vom laufenden Tag (falls sar cron job aktiv war). Mit Interval sar 5 5 z.B. 5x alle 5 Sekunden CPU etc.

### **Hauptoptionen (für Bereiche):**

- **-u**: CPU-Auslastung (Default, wenn keine angegeben) – zeigt %user, %nice, %system, %iowait, %steal, %idle.
- **-r**: Memory (free, used, buffers, cache, commit, etc.).
- **-b**: I/O (transfer rates, i/o per sec, etc.).
- **-B**: Paging (pgpgin/s, pgpgout/s, fault, majflt...).

- **-W**: Swapping (swap in/out).
- **-q**: Queue length (runq, i/o queue) and load average.
- **-n DEV**: Network: DEV mode – zeigt Netzwerk-Interface Stats (rxpck/s, txpck/s, rxKB/s, txKB/s, etc.). -n ALL für alle Net-Statistiken.
- **-d**: Block device I/O (similar to iostat per device).
- **-p [DEV]**: CPU per processor (if SMP). e.g. sar -P ALL 1 3 – CPU usage per core.
- **-f file**: Lese Stats aus Log-Datei (z.B. /var/log/sysstat/sarXX).
- **-s HH:MM:SS, -e HH:MM:SS**: Start/End time for reading logs (z.B. 9 bis 18 Uhr Stats).

#### **Beispiel (aktuelle CPU alle 1s):**

```
$ sar 1 3
```

Linux 5.4.0 ... (8 CPUs)

```
02:00:00 PM CPU %user %nice %system %iowait %steal %idle
02:00:01 PM all 5.13 0.00 1.28 0.00 0.00 93.59
02:00:02 PM all 7.14 0.00 1.02 0.00 0.00 91.84
02:00:03 PM all 3.06 0.00 0.77 0.00 0.00 96.16
Average: all 5.11 0.00 1.02 0.00 0.00 93.87
```

#### **nohup (No Hangup)**

**Beschreibung:** Lässt einen Prozess auch nach Logout weiterlaufen. nohup fängt das HUP-Signal (Hangup) ab, das normalerweise gesendet wird, wenn ein Terminal geschlossen wird. Mit nohup startet man ein Kommando so, dass es dieses Signal ignoriert und Standardausgabe ggf. in *nohup.out* umleite】 .

#### **Syntax:**

```
nohup <Befehl> [Arg...] [&]
```

#### **Funktionsweise:**

- nohup sorgt dafür, dass der ausgeführte Prozess das SIGHUP ignoriert. Außerdem werden *stdout* und *stderr* umgeleitet in *nohup.out*, falls diese derzeit ans Terminal gebunden sind. (Wenn man & anhängt – also im Hintergrund – ist es üblich, die Ausgabe nicht am Terminal zu lassen).

- Üblicherweise kombiniert mit & (Hintergrund). Man loggt dann aus, und das Programm läuft weiter.

### **Beispiele:**

- `nohup long_running.sh &` – Führt *long\_running.sh* im Hintergrund aus, Ausgaben gehen in *nohup.out*. Der Befehl überlebt Logout. Die Shell gibt z.B. zurück: appending output to nohup.out.
- `nohup python server.py > server.log 2>&1 &` – Startet Python-Server im Hintergrund, leitet stdout und stderr in *server.log* (damit *nohup.out* nicht genutzt).

*Hinweis:* Wenn man bereits in einer Screen- oder tmux-Session ist, braucht man kein nohup; auch systemd-run oder Cron-Jobs sind Alternativen. Aber nohup ist ein schnelles Mittel.

### **`mkswap/swapon/swapoff` (Swap einrichten und verwalten)**

- **mkswap** – Initialisiert eine Partition/Datei als Swapbereich. Syntax: `mkswap [Optionen] <Gerät/datei>`. Wichtige Option: -L <Label> (Swap-Label setzen), -U <UUID>. Beispiel: `mkswap /dev/sda3`.
- **swapon** – Aktiviert einen Swapbereich für Nutzung durch das System. Nach mkswap muss Partition via swapon <Device> eingebunden werden (oder in /etc/fstab eingetragen mit swap Typ). Option -a aktiviert alle in fstab markierten Swaps. --show zeigt aktuellen Swap an (Äquivalent cat /proc/swaps).
- **swapoff** – Deaktiviert Swap. Syntax: `swapoff <Device>` oder `swapoff -a` für alle.

### **Beispiele:**

```
mkswap -L "SWAP" /dev/sda3
```

Setting up swapspace version 1, size = 8 GiB (some hex UUID)

```
swapon /dev/sda3
```

```
swapon --show
```

| NAME      | TYPE      | SIZE | USED | PRIOR |
|-----------|-----------|------|------|-------|
| /dev/sda3 | partition | 8G   | 0B   | -2    |

`swapoff -a` würde alle Swapbereiche aus dem Betrieb nehmen (z.B. vor Partition-Vergrößerung oder Kernel-Dump etc.).

### **`free` (Speicherübersicht)**

**Beschreibung:** Zeigt eine Übersicht über den aktuellen Speicherverbrauch (RAM und Swap) im System. Es summiert total, used, free, and cached/buffered memory.

## Syntax:

free [Optionen]

## Optionen:

- **-h**: Human Readable (z.B. MiB/GiB) .
- **-m, -g, -k**: Ausgabe in MiB, GiB, KiB.
- **-b**: in Bytes.
- **-t**: Fügt eine "Total" Zeile (RAM+Swap) hinzu.
- **-s N**: Aktualisiert alle N Sekunden (kontinuierliche Ausgabe wie watch).
- **-c N**: Wenn mit -s genutzt, stop nach N Aktualisierungen.
- **--si**: Benutzt 1000er statt 1024er Basis (KB = 1000 Bytes etc.).

## Beispiel free -h:

|       | total | used  | free  | shared | buff/cache | available |
|-------|-------|-------|-------|--------|------------|-----------|
| Mem:  | 15Gi  | 5.2Gi | 2.3Gi | 1.1Gi  | 7.4Gi      | 8.8Gi     |
| Swap: | 2.0Gi | 0.0Gi | 2.0Gi |        |            |           |

available ist eine Kernelschätzung, wieviel noch für neue Prozesse verfügbar, unter Berücksichtigung von caches, etc.

## vmstat (Virtual Memory and system stats)

**Beschreibung:** Zeigt verschiedene Systemmetriken: Speicher, Auslagerung, CPU, Interrupts, Context Switches etc. im zeitlichen Verlauf. vmstat ohne Parameter gibt Durchschnitt seit Boot; mit Intervall zeigt periodische Werte (Differenzen pro Intervall).

## Syntax:

vmstat [Optionen] [Interval [Count]]

## Wichtige Spalten (Standardausgabe):

- **procs**: *r* (Anzahl laufender Prozesse im Runqueue), *b* (blockierte Prozesse).
- **memory**: *swpd* (Swap genutzt), *free* (freier RAM), *buff* (Puffer), *cache* (Page Cache).
- **swap**: *si* (Swap-In KB/s), *so* (Swap-Out KB/s).
- **io**: *bi* (Block In, Blöcke gelesen von Disk/s), *bo* (Block Out, geschrieben/s).
- **system**: *in* (Interrupts/s), *cs* (Context Switches/s).

- **cpu:** us (% CPU in Userzeit), sy (% Systemzeit), id (% Idle), wa (% IO-Wait), st (% Steal bei virt.).

#### **Optionen:**

- **-S <unit>:** Werte in k (1000) oder K (1024) etc.
- **-a:** Zeigt zusätzlich Aktivität der *aktiven/inaktiven* Memory Listen (sin, sout).
- **-d:** Zeigt Disk-Statistiken (like iostat) statt VM stats.
- **-p <dev>:** Zeigt Partition-Statistiken (reads/writes tot und sek) für Device.
- **-s:** Summiert memory stats (eine Liste aller relevanten Zähler, analog /proc/meminfo).
- **-m, -t:** Variation der Ausgabe (meist uninteressant, -m memory in MB etc).

#### **Beispiel (alle 2s):**

```
$ vmstat 2 3
procs -----memory----- swap-- -----io---- -system-- -----cpu-----
r b free buff cache si so bi bo in cs us sy id wa st
1 0 230000 120000 760000 0 0 5 10 150 300 5 191 3 0
0 0 229000 120000 760500 0 0 0 20 140 280 4 194 1 0
1 0 228500 120000 761000 0 0 0 30 160 320 6 192 1 0
```

Erste Zeile nach Header ist seit Boot (ohne parameter, oft uninteressant); weitere sind Echtzeit pro Intervall.

#### **modinfo (Module Information)**

**Beschreibung:** Zeigt Informationen über ein Linux-Kernelmodul (Loadable Kernel Module) – Version, Beschreibung, Lizenz, Abhängigkeiten, Parameter etc】 .

#### **Syntax:**

```
modinfo [Optionen] <Modulname oder Modul.ko>
```

#### **Optionen:**

- **-a, --author:** Nur Autor ausgeben.
- **-d, --description:** Nur Beschreibung.
- **-n, --filename:** Zeigt Pfad der Moduldatei im Filesystem.
- **-p, --parameters:** Listet die Parameter und ihre Typen/Beschreibungen.

- **-F <field>**: Gibt nur das spezifizierte Feld aus (alternativ obige Kurzoptionen). Mögliche Felder: author, description, license, parm, alias, etc.
- **-k <kernelversion>**: Fragt Infos für Kernel-Version (sucht in /lib/modules/<version>) statt laufendem Kernel.

**Beispiel:** modinfo e1000e (Intel NIC Treiber) gibt etwa:

```
filename: /lib/modules/5.4.0/kernel/drivers/net/ethernet/intel/e1000e/e1000e.ko
version: 3.2.6-k
license: GPL
description: Intel(R) PRO/1000 Network Driver
author: Intel Corporation, <e1000-devel@lists.sourceforge.net>
...
srcversion: 5C6D1A8....
alias: pci:v00008086d000010D3sv*sd*bc*sc*i*
...
parm: IntMode:Interrupt Mode (0=Auto, 1=MSI, 2=MSI-X) (int)
parm: InterruptThrottleRate:... (array of int)
```

Man sieht Dateipfad, Version, Parameter etc.

### **insmod/rmmod/modprobe (Kernelmodule laden/entladen)**

- **insmod <pfad/modul.ko> [args]**: Lädt ein Modul in den Kernel. Nutzt genau Pfad; löst keine Abhängigkeiten auf.
- **rmmod <modulname>**: Entfernt ein geladenes Modul (wenn kein anderer in Benutzung). Option -f zum Erzwingen (nur falls Kernel CONFIG\_MODULE\_FORCE\_UNLOAD).
- **modprobe [Options] <modulname> [args]**: Empfehlenswerter Befehl zum Laden eines Moduls und seiner Abhängigkeiten. Es durchsucht /lib/modules/<kernel>/modules.dep nach Dependencies. Auch zum Entfernen via modprobe -r <modul> (lädt auch Abhängigkeiten aus, falls unbenutzt).
  - Optionen: --dry-run oder -n testet nur, -v verbose, -q quiet, alias handling etc. modprobe Konfig steht in /etc/modprobe.d.
  - Bsp: modprobe e1000e (lädt Intel NIC modul + ggf. benötigte mii). modprobe -r e1000e entfernt es.

### **depmod (Dependency modules)**

**Beschreibung:** Generiert die Modulabhängigkeitsdatei *modules.dep* basierend auf Modulen in */lib/modules/<version>*. Wird meist beim Kernel-Install aufgerufen. Kann man manuell ausführen nach Hinzufügen eigener Module. Syntax: depmod -a (für alle).

### Optionen:

- -a, --all (default): alle Kernel-Versionen verarbeiten.
- -A, --quick: nur verändern, wenn *modules.dep* älter als Module-Dateien (incremental update).
- -n: Ausgabe nach *stdout* statt in *modules.dep* schreiben (Test).
- -v: verbose.

## sysctl (Query/Set Kernel Parameters)

**Beschreibung:** Anzeige oder Änderung von Kernel-Laufzeitparametern (in */proc/sys*). sysctl ermöglicht bequemen Zugriff auf z.B. Netzwerktruning, VM, FS Settings zur Laufzeit.

### Syntax:

sysctl [Optionen] [<Name>=<Wert> ...] or sysctl -p

### Optionen:

- Ohne =: liest Parameter. Z.B. `sysctl net.ipv4.ip_forward` -> `net.ipv4.ip_forward = 0`.
- Mit name=value: setzt Parameter. Z.B. `sysctl net.ipv4.ip_forward=1` aktiviert IP-Forwarding (sofort).
- -w: explizit Schreibmodus (kann man weglassen, war historisch).
- -p [Datei]: Lädt Einstellungen aus Datei (Default */etc/sysctl.conf*) und wendet an. Man ruft `sysctl -p` typ. beim Boot (init-Skripte) oder manuell nach Bearbeiten der config.
- -a: Alle Parameter anzeigen (riesige Liste). `sysctl -a | grep ipc etc`.
- -A, --pattern <regex>: Filtert mit Regex (statt manuell grep).
- -e: Ignoriert unbekannte Keys (useful mit -p, sonst bricht ab).
- -N: Nur Namen, keine Werte anzeigen (list keys).
- -q: Quiet (unterdrückt Bestätigungsmeldungen).

Parameter-Keys entsprechen */proc/sys* Pfaden, z.B. */proc/sys/net/ipv4/ip\_forward* -> key `net.ipv4.ip_forward`.

### Beispiele:

- sysctl vm.swappiness -> zeigt swappiness (z.B. vm.swappiness = 60).
- sysctl -w vm.swappiness=10 -> setzt auf 10 (weniger Swapneigung).
- sysctl -p /etc/sysctl.d/99-sysctl.conf -> lädt Parameter aus der angegebenen Datei.

## dmesg (Dump kernel message buffer)

**Beschreibung:** Zeigt die Nachrichten des Kernel-Ringspeichers (Kernellog) an. Hier stehen Boot-Meldungen und Hardware-Events (z.B. USB eingesteckt, Treiber Ausgabe). dmesg ohne Optionen druckt alle aktuellen Kernelmeldungen.

### Syntax:

dmesg [Optionen]

### Optionen:

- **-T**: Zeigt menschenlesbare Zeitstempel (Jahr-Monat-Tag Stunde:Minute:Sekunde) statt der raw Sekunden seit Boot. (Achtung: Kann ungenau sein, da Kernel keine Zeitzone kennt; -T rechnet anhand aktueller TZ).
- **-l <level>, --level**: Filtert Nachrichten bestimmter Log-Levels. Level: emerg, alert, crit, err, warn, notice, info, debug. Z.B. dmesg -l err, warn nur Fehler und Warnungen.
- **-k, --kernel**: Zeigt nur Kernel-Messages (Standard). **-u, --userspace**: Nur Benutzerraum (Logeinträge vom printk\_user). Kaum genutzt, da meiste aus Kernel.
- **-H, --human: Pager-Modus** (lässt scrollen, farbig Highlighting). Interaktiv (Space/PgUp etc.).
- **-w, --follow**: "Tail -f" Modus für dmesg – neue Kernelmeldungen laufend anzeigen.
- **-s N**: Buffer-Größe (Bytes) die gelesen wird (Default könnte begrenzt sein). dmesg -s 1000000.

### Beispiel (gekürzt):

```
[0.000000] Linux version 5.4.0-80-generic ...
[0.345678] PCIe Bus 0000:00: root hub ...
[1.234567] usb 1-2: new high-speed USB device number 3 ...
[1.345678] ata1: SATA link up ...
[10.567890] Bluetooth: RFCOMM ver 1.11
```

dmesg -T zeigt z.B. [Mon Oct 11 14:23:45 2021] ....

Typisch ruft man dmesg | grep -i error um nach Kernel-Fehlern zu suchen, oder dmesg -w in einem Terminal um z.B. neue USB/Hardware Events sofort zu sehen.

## Dienstverwaltung und Systemstart

### systemctl (Control systemd)

**Beschreibung:** Haupt-Werkzeug zur Steuerung des *systemd*-Initsystems (heutzutage auf vielen Linux-Distributionen). systemctl verwaltet **Services** (Start/Stop/Enable), **Daemon-Konfiguration, System-Zustände** (Reboot/Shutdown), **Unit**-Status und Logs, etc】 .

#### Syntax:

systemctl [Optionen] <Sub-Befehl> [Unit/...]

Units können Services (.service), Mounts (.mount), Timer (.timer), Sockets (.socket), etc. sein. Oft kann man den Suffix weglassen (z.B. "ssh" wird als "ssh.service" erkannt).

#### Wichtige Sub-Befehle (Service Management):

- **start <service>, stop <service>** – Startet oder stoppt einen Dienst \*sofortfile-4fdk7rvx39azjretinarqb】 . Bsp: systemctl start apache2.
- **restart <service>** – Stop + Start (auch wenn nicht lief, wird gestartet).
- **reload <service>** – Lädt Konfiguration neu (sendet i.d.R. SIGHUP). Muss vom Service unterstützt werden (in Unit als ReloadSignal oder ExecReload definiert). Bsp: systemctl reload nginx.
- **status <service>** – Zeigt Status des Dienstes: Läuft er? Letzte Startmeldung, ggf. Auszug aus Journal-Log】 .
- **enable <service>** – Aktiviert den Dienst beim Boot (erstellt Symlinks in passenden WantedBy-Verzeichnissen).
- **disable <service>** – Deaktiviert Autostart.
- **is-enabled, is-active** – Prüfen ob Unit aktiviert ist / gerade aktiv ist (exit 0/1).
- **mask <service>** – "Maskiert" den Service, d.h. verhindert Start vollkommen (setzt Link auf /dev/null). *Unmask* zum Aufheben.
- **daemon-reload** – Falls man eine Unit-Datei geändert oder neu hinzugefügt hat, diese neu einlesen (Konfig neu laden, ohne Dienste neu zu starten).
- **kill <service>** – Sendet ein Signal an alle Prozesse der Service-Unit (z.B. systemctl kill -s SIGUSR1 myapp.service).

## System Commands:

- **list-units** – Listet alle geladenen Einheiten (nur aktive standardmäßig). --all um Inaktive auch zu sehen, --type=service filtern auf Services, etc.
- **list-unit-files** – Listet Unit-Dateien und ob sie enabled/disabled/etc. sind (Installationszustand).
- **list-dependencies <unit>** – Zeigt Abhängigkeiten (Requires/Wants) Baum für Unit. --reverse zeigt welche Units diese Unit benötigen.
- **show <unit>** – Gibt alle Properties von Unit aus (oder subset via -p).
- **cat <unit>** – Zeigt den Inhalt der Unit-Datei (und Drop-Ins).
- **edit <unit>** – Öffnet Editor für Drop-In Snippet, um diese Unit zu überschreiben/erweitern.
- **preset <service>** – Setzt enable/disable gemäß Distribution-Vorgabe (in /usr/lib/systemd/system-preset/).

### Targets (Runlevels):

- **isolate <target>** – Wechselt in ein target (beendet andere). Z.B. systemctl isolate rescue.target (Einzeluser-Modus ähnlich runlevel 1).
- **set-default <target>** – Legt Default Boot-Target fest (z.B. graphical.target vs multi-user.target).
- **get-default** – zeigt aktuelles Defaulttarget.

### Power Management:

- **reboot, poweroff, halt** – führen Neustart bzw. Herunterfahren aus (rufen intern entsprechende target isolation).
- **suspend, hibernate, hybrid-sleep** – starten Energiesparmodus.

### Journal/Logging Integration:

- **status** (wie oben) zeigt bereits Log-Auszug.
- **journalctl** (separater Befehl, nicht systemctl subcommand) – zum Betrachten der systemd Logs. Z.B. journalctl -u <service> zeigt Logs zu einem Service, -f zum folgen, --since "2021-10-01" ab bestimmtem Datum etc.

### Beispiele:

- systemctl enable --now nginx – Aktiviert **und** startet nginx sofort (Option --now führt gleich start nach enable aus).
- systemctl disable apache2.service – Entfernt Apache2 aus Autostart.
- systemctl restart networking – Dienst neu starten (bei Netz aufpassen).

- `systemctl list-units --failed` – Zeigt nur fehlgeschlagene Units.
- `systemctl mask firewalld; systemctl stop firewalld` – Verhindert Start des Firewall dauerhaft und stoppt ihn sofort (z.B. falls man anderen Firewall nutzt).

### **service (SysVinit Compatibility)**

**Beschreibung:** Der Befehl `service` war bei SysVinit/Upstart gebräuchlich, um Start/Stop-Skripte in `/etc/init.d` aufzurufen. Auf systemd-Systemen ruft `service` intern `systemctl` auf. Man kann aber `service <name> start/stop/status` noch benutzen.  
`4fdk7rvx39azjretinarqb】`. Moderne Systeme – besser `systemctl` direkt.

#### **Syntax:**

`service <ServiceName> [start|stop|restart|status|...]`

**Beispiel:** `service ssh status` ruft `systemctl status ssh.service` auf.

*(Auf SysVinit ohne systemd würde es das `init.d` Script ausführen.)*

### **chkconfig (Runlevel Konfiguration - SysV kompatibel)**

**Beschreibung:** Ein Werkzeug zum Verwalten der Start-/Stopp-Links von SysV-Init-Skripten über verschiedene Runlevels (0-6file-4fdk7rvx39azjretinarqb】). Auf systemd-Systemen gegenstandslos (meist vorhanden für Kompatibilität, was auf `systemctl enable/disable` mappt).

#### **Wichtigste Befehle:**

- `chkconfig --list [name]` – Listet alle Dienste und ob sie in runlevel 0-6 on/off sind.  
`4fdk7rvx39azjretinarqb】`. Mit Name nur diesen.
- `chkconfig name on` (bzw. `off`) – Schaltet Service in allen entsprechenden runlevels an oder aus (Standard runlevels 2-5 on für on).
- `chkconfig name --level 35 on` – Feinsteuerung: aktiviert Dienst in Runlevel 3 und 5.
- `chkconfig name reset` – Setzt Standard (wie im Init-Skript Header vorgesehen).

*\*(Auf systemd delegiert `chkconfig` intern an `systemctl enable/disable`. Für Nostalgie, wissen was es tat.)*

### **Zeitplanung (Cron und At)**

#### **crontab (Cron Table Management)**

**Beschreibung:** Planen wiederkehrender Aufgaben. `crontab` bearbeitet die Aufgabenliste des aktuellen Benutzers (oder eines anderen mit `-u`) im Cron-Daemon. Jede Zeile in `crontab -e` hat Format "Min Stunde Tag Monat Wochentag Befehl". Der Cron-Dienst prüft pro Minute die Tabellen und führt fällige Kommandos aus.  
`4fdk7rvx39azjretinarqb】`.

### **Befehle:**

- crontab -e – Editiert die Crontab des aktuellen Benutzers (öffnet \$EDITOR). Beim Speichern wird die neue Tabelle dem Cron-Daemon übergeben [4fdk7rvx39azjretinarqb].
- crontab -l – Listet die aktuelle Crontab auf [file-4fdk7rvx39azjretinarqb].
- crontab -r – Entfernt (löscht) die aktuelle Crontab komplett. (Vorsicht, kein Undo).
- crontab -u username -e/-l/-r – Als root: Bearbeitet/liest/löscht die Crontab eines anderen Benutzers.

### **Cron Format Erinnerung:**

# Minute(0-59) Hour(0-23) Day(1-31) Month(1-12) Weekday(0-7) Command

0 2 \* \* \* /usr/bin/backup.sh # täglich 02:00

\*/5 9-17 \* \* 1-5 echo "Working..." # Werktags 9-17 Uhr alle 5 min

\*=jeder Wert, Komma getrennte Listen, Intervalle mit -, Step mit /.

Spezial @-Strings: @reboot, @hourly, @daily, @yearly etc.

**Anacron:** Cron führt Jobs aus, wann Maschine an ist. Für nicht ständig laufende Rechner nutzt man anacron (Tagesjobs nachholen).

### **at (One-time scheduled task)**

**Beschreibung:** Legt einmalige Ausführungen von Befehlen zu einer bestimmten Zeit in der Zukunft fest [file-4fdk7rvx39azjretinarqb]. at nimmt eine Zeitangabe entgegen und liest dann von STDIN die Befehle, die ausgeführt werden sollen (oder mit -f aus einer Datei). Der atd Daemon muss laufen.

### **Verwendung:**

echo "/path/to/script.sh" | at 02:30 # führt Script heute Nacht 2:30 aus

Man ruft at <Zeit> auf, landet in einer Stdin-Eingabe (end mit Ctrl+D) oder speist via echo/<<EOF>>.

**Zeitformat:** Z.B. at 14:00 (heute 14 Uhr, falls schon vorbei dann morgen), at 11pm, at midnight, at now + 5 days, at 08:00 12.12.2025 (8:00 am 12 Dec 2025). Kann englische Wörter: tomorrow, noon, teatime(16:00) etc.

### **Befehle:**

- atq – Zeigt anstehende at-Jobs (Queue) des aktuellen Userfiles [file-4fdk7rvx39azjretinarqb]. (Listet Job-IDs und geplante Zeit). Root sieht alle oder atq -q <Queue> für bestimmte Batch-Queues.

- **atrm <JobID>** – Entfernt einen geplanten Job aus der Queue. (JobID sieht man in atq).
- **batch** – Alias zu at -q b now (führt Jobs aus, wenn Load Average niedrig ist). "Batch" schiebt den Auftrag in Batch-Queue, läuft sobald System nicht ausgelastet (LA<0.8 default).

### **Beispiel:**

```
$ at now + 1 minute
warning: commands will be executed using /bin/sh
at> echo "Hello after 1 min" >> /tmp/test.txt
at> <EOT>
job 5 at Fri Oct 15 16:41:00 2021
```

Dann innerhalb einer Minute atq zeigt e.g. 5 Fri Oct 15 16:41:00 2021 a user. Nach Ausführung landet Text in Datei.

(Zugriff: /etc/at.allow /etc/at.deny definieren wer at nutzen darf – analog cron.allow/deny.)

## **Drucken**

### **lpr (Line Print Request)**

**Beschreibung:** Sendet eine Datei (oder STDIN) an den Drucker (Print-Job einreihen). lpr ist Teil des BSD-Linux-Drucksystems (CUPS stellt Kompatibilität bereit). Alternative: lp (System V style).

### **Syntax:**

lpr [Optionen] [Dateien...]

### **Wichtige Optionen:**

- **-P <Druckernname>**: Wählt einen bestimmten Drucker aus (sonst Standarddruckerfile-4fdk7rvx39azjretinarqb】). Druckernamen findet man via lpstat -a oder CUPS Webinterface.
- **-# <Num>**: Anzahl Kopien drucken (z.B. -#3 dreifach).
- **-h**: Unterdrückt Banner-Seite (header page) falls aktiviert.
- **-r**: Löscht die Datei(en) nach dem Versenden an den Spool (rarely used).
- **-m**: Sendet E-Mail Benachrichtigung nach Druck (sofern eingerichtet).
- **-o <Option>=<Wert>**: Drucker-spezifische Optionen übergeben (z.B. -o landscape für Querformat, oder -o media=A4 oder -o sides=two-sided-long-edge

für Duplex). Diese Optionen sind je nach Drucker Treiber definierbar (CUPS options).

- **-C <Klasse>, -J <Jobname>, -T <Titel>**: Legen Klassenname, Jobname, Titel für Banner etc. fest.
- **--help** (GNU lpr) usw.

### Beispiele:

- lpr report.pdf – Schickt *report.pdf* an Standarddrucker.
- lpr -P HP\_LaserJet -#2 invoice.txt – Druckt *invoice.txt* 2-fach auf Drucker "HP\_LaserJet".
- echo "Test Page" | lpr – Druckt STDIN (den Text "Test Page").
- lpr -o sides=two-sided-long-edge thesis.pdf – Duplexdruck beidseitig.

(Verwandt: *lpq* zeigt Druckerqueue, *lprm* löscht Jobs aus der Queue.)

## Lokale Systemkonfiguration

### date (Anzeige/Setzen Datum/Uhrzeit)

**Beschreibung:** Zeigt das aktuelle Datum und die Uhrzeit an, oder stellt sie ein. Auch Formatierung in beliebigem Muster ist möglich.

#### Syntax:

date [Optionen] [+Format]

date -s "STRING" (setzen)

#### Optionen / Verwendung:

- **Ausgabe formatieren:** date "+FORMAT" – Mit Format-Pattern ähnlich strftime: %Y (Jahr), %m Monat(01), %b Monat (Kurzname), %d Tag, %H Stunde (24h), %I (12h), %M Minute, %S Sek, %A Wochentag, %a (kurz), %Z Zeitzone, %z offset, %T = %H:%M:%S, %F = %Y-%m-%d, etc. Z.B. date "+%d.%m.%Y %H:%M:%S" -> "05.10.2025 14:30:00".
- **Systemzeit stellen:** date -s "2025-10-05 14:30:00" – Setzt Datum. Braucht Root. (Alternative: date --set="...").
- **Hardware-Uhr stellen:** In Linux erfolgt date auf Systemzeit (in Kernel, RTC separiert). Um auch BIOS/RTC Uhr zu setzen, hwclock --systohc (setze Hardware Clock nach System Clock). Umgekehrt hwclock --hctosys.
- **Universale Zeit:** date -u zeigt in UTC. date -u -s "2025-10-05 12:30:00" setzt in UTC.

- **RFC-3339 Format:** date --rfc-3339=ns etc. Standard ISO.
- **Unix Timestamp:** date +%s gibt Sekunden seit 1.1.1970 (UTC).

### **Beispiele:**

- date -> "Tue Oct 5 14:30:00 CEST 2025".
- date +"%Y-%m-%d\_%H-%M-%S" -> "2025-10-05\_14-30-00".
- sudo date -s "@\$(( \$(date +%s) + 3600 ))" -> Systemzeit um eine Stunde vorstellen  
(Beispiel mit timestamp math).

### **timedatectl (Zeiteinstellung via systemd)**

**Beschreibung:** Befehl von systemd zum Anzeigen und Ändern der Systemzeit, Hardware-Uhr, Zeitzone und NTP-Synchronisatiofile-4fdk7rvx39azjretinarqb】 .

### **Verwendung:**

- **timedatectl** (ohne args): Zeigt aktuellen Status: Lokale Zeit, UTC, RTC, Zeitzone, ob NTP aktiv ist etcfile-4fdk7rvx39azjretinarqb】 .
- **timedatectl set-time "YYYY-MM-DD HH:MM:SS"** – Setzt Systemzeit (wenn kein NTP aktiv).
- **timedatectl set-timezone <Zone>** – Ändert Zeitzone. Z.B. timedatectl set-timezone Europe/Berlin. (Verfügbare Zonen unter /usr/share/zoneinfo/, oder timedatectl list-timezones).
- **timedatectl set-ntp true/false** – Aktiviert bzw. deaktiviert die automatische Zeitsynchronisation (via systemd-timesyncd oder anderes NTP). Bei true startet es den Dienst falls vorhanden.
- **timedatectl set-local-rtc 1/0** – Legt fest, ob die Hardware-Uhr (RTC) in **lokaler Zeit** läuft (1) oder in UTC (0, Standard). Windows dual-boot oft RTC=lokal (dann 1), sonst belässt auf 0=UTF um Probleme zu vermeiden.

### **Beispiele:**

```
$ timedatectl
```

Local time: Tue 2025-10-05 14:30:00 CEST

Universal time: Tue 2025-10-05 12:30:00 UTC

RTC time: Tue 2025-10-05 12:30:00

Time zone: Europe/Berlin (CEST, +0200)

System clock synchronized: yes

NTP service: active

RTC in local TZ: no

timedatectl set-time "2025-12-24 18:00:00" stellt Heiligabend 18:00. timedatectl set-timezone UTC wechselt Zeitzone auf UTC.

## **localectl (Locale and Keyboard Layout)**

**Beschreibung:** Verwaltet System-Locale (Sprache, Gebietsschema) und Tastaturlayout-Einstellungen in systemd-basierten Systemefile-4fdk7rvx39azjretinarqb】. Es kann die Variablen in /etc/locale.conf sowie Konsolen-Tastaturlayout in /etc/vconsole.conf setzen. (Für X/Wayland Tastatur greift localectl in /etc/X11/xorg.conf.d/ oder mittels setxkbmap separat).

### **Verwendung:**

- **localectl** (ohne args): Zeigt aktuelle Locale(s) und Keymap. Z.B.:  
System Locale: LANG=de\_DE.UTF-8  
VC Keymap: de  
X11 Layout: de  
X11 Model: pc105 etc.
- **localectl list-locales** – Listet alle verfügbaren Locale, die generiert sind (die in /usr/lib/locale oder /etc/locale.gen markiert).
- **localectl set-locale LANG=<locale> [VAR=locale...]** – Setzt System-Locale. Z.B. localectl set-locale LANG=en\_US.UTF-8. Optional mehrere, z.B. LC\_TIME=en\_GB.UTF-8. Wirksam nach Neustart oder neu login shells.
- **localectl list-keymaps** – Listet verfügbare Console-Keymaps (z.B. us, de, ...).
- **localectl set-keymap <map> [<toggle>]** – Setzt Tastaturlayout für Konsole (vconsole) und ggf. X11, falls X11 Layout nicht getrennt definiert. Z.B. localectl set-keymap de.
- **localectl set-x11-keymap <layout> [model] [variant] [option]** – Spezifisch X11 (Graphical) Layout/Modell setzen. Z.B. localectl set-x11-keymap us pc104 "" terminate:ctrl\_alt\_bksp.

### *Beispiel:*

localectl set-locale LANG=fr\_FR.UTF-8 (Systemsprache auf Französisch).

localectl set-keymap us (Konsole auf US-Layout).

localectl set-x11-keymap de pc105 nodeadkeys (Grafische Tastatur auf Deutsch mit Variante "nodeadkeys").

(Die Konfigurationen landen in /etc/locale.conf, /etc/vconsole.conf, /etc/X11/xorg.conf.d/00-keyboard.conf entsprechend.)

## **Netzwerk-Konfiguration**

## ip (Netlink Interface Config)

**Beschreibung:** Moderner Befehl zur Verwaltung von Netzwerkinterfaces, IP-Adressen, Routing, Tunnels etc. (Nachfolger von ifconfig/route). ip hat Unterkommandos für verschiedene Objekte: **link** (Netzwerkinterface), **addr** (IP-Adressen), **route**, **neigh** (ARP/NDP Nachbarn), **rule** (Policy Routing), **maddr** (Multicast), **addrlabel**, **netns** (Net Namespaces), **link set** (Interface-Settings).

### Syntax (Grundstruktur):

ip [Options] <Object> <Subcommand> [Arguments...]

z.B. ip link show, ip addr add, ip route list, etc.

### Häufige Befehle/Beispiele:

- **Interface anzeigen:** ip link show [dev <iface>] – Listet alle Netzwerkinterfaces (oder ein spezifisches). Ausgabe enthält Index, Name, MAC, MTU, Flags (UP,LOWER\_UP=Link OK, MULTICAST), etc.
- **Interface aktivieren/deaktivieren:** ip link set <iface> up / down – entspricht `ifconfig iface up/downfile-4fdk7rvx39azjretinarqb】 .
- **Adresse zuweisen:** ip addr add <IP>/<Prefix> [broadcast +] dev <iface> – z.B. ip addr add 192.168.1.100/24 dev eth0. broadcast + lässt broadcast aus Prefix berechnen (optional).
- **Adresse löschen:** ip addr del <IP>/<Prefix> dev <iface>.
- **Adressen anzeigen:** ip addr show [dev <iface>] – Listet IPv4/IPv6 Adressen auf Interfacefile-4fdk7rvx39azjretinarqb】 . Zeigt auch Scope (global/link), flags (tentative, dynamic).
- **Standardroute setzen:** ip route add default via <Gateway-IP> [dev <iface>] – z.B. ip r add default via 192.168.1.1 dev eth0.
- **Statische Route setzen:** ip route add <Netz>/<Präfix> via <nächsterHop> dev <iface>.
- **Routen auflisten:** ip route show (oder ip r).
- **ARP Cache:** ip neigh show – zeigt ARP/ND-Cache (Neighbors).
- **ARP Eintrag fix setzen:** ip neigh add <IP> lladdr <MAC> dev <iface> nud permanent.
- **Secondary IPs / Aliase:** Einfach mehrfach ip addr add auf selbes dev ausführen (oder use Label).
- **MTU ändern:** ip link set dev <iface> mtu 1400.

- **MAC ändern:** ip link set dev <iface> address <NewMAC> (Down interface first).
- **Promisc Mode:** ip link set dev <iface> promisc on.
- **Routing Policy (advanced):** ip rule und ip route add table <n> for source-based routing etc.
- **Network Namespace mgmt:** ip netns (add, del, exec, etc – advanced).

**Options global:** -s (stats), -d (details), -br (brief output), -c (color output). Z.B. ip -br addr zeigt Interfaces in einzeiligem Kurzformat.

### Beispiele:

- ip -br link ->

```
lo UNKNOWN 127.0.0.1/8 ::1/128
eth0 UP 192.168.1.10/24 fe80::.../64
```

- ip addr add 10.0.0.1/16 dev eth1 (fügt IP).
- ip route show (z.B. default via 192.168.1.1 dev eth0 proto dhcp metric 100 etc.).
- ip neigh show (z.B. 192.168.1.1 dev eth0 lladdr aa:bb:cc:dd:ee:ff REACHABLE).

(ifconfig (net-tools) alternative: ip ist ausführlicher und bevorzugt.)

### ifconfig (Interface Config – alt)

**Beschreibung:** Älteres Werkzeug aus net-tools (weitgehend ersetzt durch ip). ifconfig kann Interfaces anzeigen, IP setzen, MTU ändern, promiscuous Mode etc., aber unterstützt z.B. keine IPv6 out-of-the-box je nach Version.

### Syntax:

ifconfig [Interface] [Optionen]

### Beispiele:

- ifconfig – zeigt alle aktiven (UP) Interfaces mit IPv4 etc.
- ifconfig -a – zeigt *alle*, inkl. down Interfaces.
- ifconfig eth0 192.168.0.5 netmask 255.255.255.0 broadcast 192.168.0.255 – setzt IP, Maske, Bcast.
- ifconfig eth0 up / down – Interface an/aus.
- ifconfig eth0 mtu 1400 – MTU ändern.
- ifconfig eth0 promisc – Promiscuous Mode an. -promisc aus.
- ifconfig eth0 hw ether 00:11:22:33:44:55 – MAC-Adresse setzen.

(ifconfig ist nicht mehr standard installiert in neueren Distros; ip commands stattdessen nutzen.)

### **iwconfig (Wireless Config – alt)**

**Beschreibung:** Ähnlich ifconfig, aber für Wireless LAN Schnittstellen (Legacy tool aus wireless-tools). Zeigt/ändert ESSID, Mode, Frequenz, Schlüssel etc. Heutzutage ersetzt durch iw (nl80211), oder NetworkManager.

#### **Beispiele (falls vorhanden):**

- iwconfig wlan0 – zeigt ESSID, Mode (Managed/Ad-Hoc/Master), Channel/Freq, Bitrate, Tx-Power, Signal Level, Encryption key on/off, etc.
- iwconfig wlan0 essid "WLAN-Name" – mit offener SSID verbinden (sofern AP ohne Sicherheit).
- iwconfig wlan0 key s:password – setzt WEP Key (veraltet).
- iwconfig wlan0 mode Monitor – Interface in Monitor Mode (für Sniffing).

(Iwconfig kann kein WPA2; da nutzt man wpa\_supplicant oder NM, etc.)

### **nmcli (NetworkManager CLI)**

**Beschreibung:** Kommandozeilen-Interface zum NetworkManager-Dienst, um Netzwerkkonfiguration (insb. Desktop-Umfeld) zu betrachten und steuerfile-4fdk7rvx39azjretinarqb】 . Kann Verbindungen (profiles) verwalten, aktivieren, WLAN scannen, etc.

#### **Verwendung (Struktur):**

nmcli [general|networking|radio|connection|device] <aktion> [Parameter]

#### **Beispiele:**

- **Status anzeigen:** nmcli general status – Gesamter NM-Status. nmcli connection show – listet gespeicherte Verbindungen (Profiles). nmcli device status – zeigt Geräte und ob connected und welches Profil.
- **WLAN scannen:** nmcli device wifi list – listet verfügbare WLANs.
- **Verbinden WLAN:** nmcli device wifi connect "SSID" password "passwort" – erstellt ggf. Connection-Profil und verbindet.
- **Verbindung aktivieren/deaktivieren:** nmcli connection up <Name> / down <Name>. Oder nmcli device disconnect <iface> (z.B. nmcli device disconnect wlan0).
- **Neue Verbindung anlegen:** nmcli connection add type ethernet ifname eth0 con-name "Wired Manual" ip4 192.168.5.5/24 gw4 192.168.5.1.

- **IPv4 ändern laufend:** nmcli connection modify "Wired connection 1" ipv4.addresses 192.168.1.50/24 ipv4.gateway 192.168.1.1 ipv4.method manual und dann up.
- **Ethernet an/aus:** nmcli networking off (setzt NM generell offline) oder nmcli radio wifi off (schaltet nur WLAN-Funk aus).

#### **Beobachten:**

- nmcli connection show --active – nur aktive Conns.
- nmcli -p device status – pretty (ncurses-like) output.

#### **ethtool (NIC Driver Tool)**

**Beschreibung:** Zeigt und ändert Einstellungen von Ethernet-Netzwerkkarten (Treiber-Level): z.B. Geschwindigkeit, Duplex, Auto-Negotiation, Wake-on-LAN, Offload-Funktionen, Statistikzählefile-4fdk7rvx39azjretinarqb】 .

#### **Verwendung:**

- ethtool <iface> – Zeigt Standard-Eigenschaften: unterstützt Geschwindigkeiten (Supported link modes), aktuelle Speed/Duplex, Auto-negotiation on/off, Ports (TP, etc.), driver name, firmware-version, bus-info, etc.
- ethtool -s <iface> speed <Mb/s> duplex <half/full> autoneg <on/off> – Setzt feste Geschwindigkeit/Duplex. (z.B. ethtool -s eth0 speed 100 duplex full autoneg off). Vorsicht: muss vom Link Partner unterstützt werden, sonst Link down.
- ethtool -k <iface> – Zeigt Kernel-Offload-Funktionen (tx-checksumming, scatter-gather, TSO etc.) und ob an/aus. -K ... (capital K) schaltet diese: z.B. ethtool -K eth0 gro off (GRO ausschalten).
- ethtool -i <iface> – Infos zum Treiber (Driver name, version, firmware, bus).
- ethtool -p <iface> [N] – "Physically identify": Blinkt die LED am Port für N Sekunden (Standard 15), falls Karte das unterstützt. Hilft um Kabelport zu finden.
- ethtool -S <iface> – Ausgabe von Statistikzählern (packets, errors, etc. vom Treiber).
- ethtool -g <iface> – Ring Puffer Größen (RX/TX rings). -G zum Setzen.
- ethtool -a <iface> – Flow-Control (Pause Frame) Status (Autoneg etc.), -A zum Setzen on/off rx/tx.
- ethtool -r <iface> – Veranlasst die Karte, Link neu auszuhandeln (Retrain).
- ethtool --identify <iface> – alias für -p (Blinken).

#### **Beispiele:**

- ethtool eth0 -> zeigt z.B. Speed: 1000Mb/s, Duplex: Full, Auto-negotiation: on, etc.
- ethtool -s eth0 speed 100 duplex half autoneg off -> zwingt 100Mbit Half-Duplex (nur Test oder Legacy).
- ethtool -p eth1 10 -> blinkt LED an eth1 für 10 Sekunden.
- ethtool -K eth0 tso off -> schaltet TCP Segment Offloading aus (im Troubleshooting falls Problem mit Offload).

### **hostnamectl (Control hostname in systemd)**

**Beschreibung:** Zeigt oder ändert den System-Hostname und verwandte Settings (wie Chassis Typ, Icon). hostnamectl ist Teil von systemd.

#### **Verwendung:**

- **hostnamectl** (ohne args): Zeigt Systeminfo: Static hostname, Transient (via DHCP), Pretty (human name), und Kernel: Operating System, Kernel, Architecture, etc.
- **hostnamectl set-hostname <name> [--pretty|--static|--transient]** – Setzt den Hostnamen. Ohne Flag wird *static* und *transient* gleichermaßen gesetzt. *static* = in /etc/hostname gespeichert (persistenter Name), *transient* = Laufzeit (kann z.B. DHCP ändern). *pretty* = nur für Anzeigezwecke (z.B. mit Großbuchstaben/Leerzeichen erlaubt).  
Beispiel: hostnamectl set-hostname server42 – setzt static+transient auf "server42".  
hostnamectl set-hostname "My Home PC" --pretty – setzt Pretty-Name.
- **hostnamectl set-icon-name <icon>** – Setzt Icon (z.B. computer-desktop, rein kosmetisch in einigen GUIs).
- **hostnamectl set-chassis <type>** – Setzt Gehäuse-Typ: desktop, laptop, server, tablet, embedded, etc. (Auch nur informative, kann z.B. in GNOME Systeminfo auftauchen).

#### **Zusatz:**

- hostnamectl status (alias ohne args) – Info.
- Traditional hostname Befehl (ohne ctl) zeigt oder setzt nur transient (bis Reboot).

### **Netzwerk-Fehlerdiagnose und Name-Auflösung**

#### **netcat (nc) (TCP/UDP Swiss Army Knife)**

**Beschreibung:** netcat (oder nc) ist ein universelles Netzwerk-Tool, um Verbindungen zu testen, Daten zu senden/empfangen, Ports zu lauschen etcfile-4fdk7rvx39azjretinarqb】. Kann sowohl Client als auch rudimentärer Server sein.

### Verwendung:

- **Verbindung als Client:** nc [options] <Host> <Port> – Baut TCP-Verbindung zu Host:Port auf, leitet stdin -> Socket, und Socket -> stdout. So kann man z.B. echo "HEAD / HTTP/1.0" | nc example.com 80 (HTTP-Request senden). Oder nc time.nist.gov 13 (Daytime Protocol). Wenn man interaktiv ausführt, kann man manuell tippen (Telnet-Ersatz).
- **Port Listening (Server):** nc -l -p <Port> [Host] – Horcht auf Port (optional an gegebene IP gebunden). Bsp: nc -l -p 1234 lauscht auf TCP/1234 auf alle Interfaces; was reinkommt wird auf stdout ausgegeben, was man tippt geht an Client.
- **UDP mode:** nc -u – Statt TCP benutzt UDP. Bsp: nc -u 192.168.1.5 53 könnte man UDP Query schicken (z.B. an DNS server).
- **Port Scan:** nc -z -v host startport-endport – "Zero-I/O mode" nur scannen, nicht verbinden. Z.B. nc -z -v 192.168.1.10 1-1024 zeigt offene Ports. (Netcat als Portscanner ist möglich, aber Nmap dafür besser).
- **Timeout:** -w <secs> – Wartezeit bis Timeout für Verbindungen/Eingaben.
- **Keep-open (inetd style):** In GNU nc -k (listen bleibt offen nach Verbindungsende, auf neue).
- **Exec:** -e <prog> – *Unsichere* Option (nicht immer kompiliert, oft disabled aus Sicherheitsgründen): Leitet die I/O eines Programms an den Socket. So kann man z.B. remote shell spawn: nc -l -p 5000 -e /bin/bash. (Equivalent zum berüchtigten rm -rf example). Modern: use socat or ssh for shells.

### Beispiele:

- **Chat/Echo Test:** Terminal A: nc -l -p 5000, Terminal B: nc localhost 5000 -> alles was A tippt sieht B und umgekehrt.
- **Datei senden:** Auf Empfänger: nc -l -p 4500 > out.txt, auf Sender: nc destIP 4500 < file.txt. (Netcat baut TCP auf und streamt Datei).
- **HTTP Request:** printf "GET / HTTP/1.1\r\nHost: example.com\r\n\r\n" | nc example.com 80 – zeigt HTML der Homepage (HTTP/1.1 erfordert Host-Header).
- **UDP Ping (ohne Antwort):** echo -n "Hello" | nc -u 192.168.1.5 9000.
- **Port scan:** nc -z -v 192.168.1.20 20-30 -> tries FTP-data, FTP etc, output open/closed.

(Netcat Implementierungen: Traditional vs OpenBSD netcat vs GNU netcat ncat (part of nmap) – Optionen leicht variierend. Z.B. OpenBSD netcat hat -l und -p zusammen als one option or so, etc.)

### **iftop (Interface TOP, Bandwidth monitor)**

**Beschreibung:** Ein curses-basiertes Tool zum Anzeigen der aktuellen Bandbreitennutzung pro Verbindung (ähnlich top, aber für Netzwerk Trafficfile-4fdk7rvx39azjretinarqb】). Es zeigt welche Peers mit dem System kommunizieren und wieviel Bytes/s in/out. Hilfreich zum realtime Debugging wer Bandbreite frisst.

#### **Verwendung:**

iftop -i <Interface>

Ohne -i wählt iftop das erste externe Interface.

#### **Optionen:**

- **-n** – Keine DNS-Auflösung (zeigt IPs statt aufzulösen, wichtig um Ausgabe schnell zu haltenfile-4fdk7rvx39azjretinarqb】).
- **-N** – Keine Port-Auflösung (zeigt Portnummern statt Services Namen).
- **-P** – Zeigt Ports zusammen mit Host (separat Spalte). Kombiniert man oft: iftop -nP.
- **-B** – Bytes statt Bits in Anzeige (Standard iftop zeigt in bits/s).
- **-F <FilterNet>** – Nur Traffic, der in bestimmten Netzbereich fällt (Filter) betrachten, z.B. -F 192.168.0.0/16 filtert auf diese Range.
- **-p** – Promiscuous Mode (auch Traffic anzeigen, der nicht an lokalem Host endet? Normal iftop Filter host involvement).
- **-t** – Text-Only Modus (keine curses-Interface, nur in stdout Stats; nicht sehr üblich).

Im Interface kann man mit Tasten: h Hilfe, n togg DNS, s sort by source, d sort by dest, t sort by throughput, p toggle port display, b toggle bar graph, B cycles through rate intervals (default 2s,10s,40s windows).

### **traceroute / tracepath (Route Tracking)**

**Beschreibung:** Sendet eine Serie von Paketen mit steigenden TTL (Time-to-Live) Werten, um den Pfad (Router Hopps) zum Ziel auszumessefile-4fdk7rvx39azjretinarqb】. Traceroute verwendet standard UDP oder ICMP ECHO, tracepath nutzt UDP with high port (no root needed). Jedes Hop sendet "Time Exceeded" zurück, das traceroute protokolliert. Ergebnis: Liste der Router vom Quell- zum Zielnetz.

#### **Syntax:**

traceroute [Optionen] <Host> [PacketSize]

tracepath <Host>

#### Optionen (traceroute):

- **-I:** ICMP Echo statt UDP verwenden (pakete wie ping).
- **-T:** TCP SYN statt UDP (z.B. um Firewalls zu durchdringen über port 80).
- **-p <Port>:** UDP port Start (Standard 33434).
- **-q <N>:** Anzahl Probes pro TTL (Default 3).
- **-w <Seconds>:** Timeout pro Hop Antwort (default 5s).
- **-n:** Nicht auflösen (IP statt DNSfile-4fdk7rvx39azjretinarqb】 .
- **-m <max\_ttl>:** Max Hop Count (Standard 30).
- **-f <first\_ttl>:** Start TTL (default 1). Manchmal hilfreich, mittendrin einzusteigen.
- **-4 / -6:** Force IPv4 / IPv6.

Optionen (tracepath) sind wenige: -n no dns, -b show both hostnames and IP, etc.

#### Beispiel:

```
$ traceroute example.com
```

```
1 192.168.1.1 (192.168.1.1) 1.123 ms 0.846 ms 0.799 ms
2 100.64.0.1 (100.64.0.1) 10.123 ms 9.876 ms 9.543 ms
3 203.0.113.5 (203.0.113.5) 15.432 ms 14.876 ms 15.210 ms
4 ae12.dar02.example.net (198.51.100.1) 30.456 ms 29.999 ms 30.123 ms
5 93.184.216.34 (93.184.216.34) 35.678 ms 35.900 ms 35.456 ms
```

Zeigt 5 Hops (Heimrouter, CGNAT router, ISP core, example.net router, Ziel).

tracepath example.com ähnlich, oft mit asymmetrischen result such as pmtu info.

#### mtr (My Traceroute)

**Beschreibung:** Kombiniert traceroute und ping in einem kontinuierlichen Diagnose-Tool. mtr sendet fortlaufend Pakete über alle Hops und aktualisiert in Echtzeit die Latenz und Packetloss-Statistiken pro Ho】 . Hilfreich um instabile Verbindungen/Packet Loss auf bestimmten Strecken zu identifizieren.

#### Verwendung:

mtr [Optionen] <Ziel>

Interaktiv curses UI standard, oder mit -r (report) Modus für einmalige Ausgabe.

#### Optionen:

- **-4 / -6**: Force IPv4/IPv6.
- **-n**: Nicht auflösen, nur IPs (schneller).
- **-w**: Wide-output (zeigt beide host+IP columns).
- **-c <count>**: Limit auf <count> Probezyklen, dann beenden (für Report).
- **-r**: Report-Modus (nicht curses, druckt Tabelle und endet).
- **-i <sec>**: Sekunden zwischen Probe-Paketen (default 1.0).
- **-p**: Pausiert Ausführung beim Start (Press enter to start).
- **-t**: Text Mode (like curses but in dumb terminal).
- **-u**: UDP mode (Standard mtr uses ICMP Echo, as -l in traceroute terms). -T for TCP mode also available.

#### Anzeige:

MTR listet Hops fortlaufend wie traceroute, plus Spalten: Loss%, Snt (packets sent), Last (ms last latency), Avg, Best, Wrst (worst), StDev. Es aktualisiert kontinuierlich.

Beispiel-Zeile:

```
3. 203.0.113.5 0.0% 10 15.4 16.1 14.8 20.3 1.3
```

### ping (ICMP Echo)

**Beschreibung:** Schickt ICMP Echo-Request-Pakete an eine Adresse und misst die Antwortzeiten (Echo-Reply), um Erreichbarkeit und Round-Trip-Time festzustellen .

#### Syntax:

```
ping
```

\*\*(Fortsetzung)\*\*

```
ping (ICMP Echo Requests)
```

\*\*Beschreibung:\*\* `ping` prüft die Erreichbarkeit eines Hosts im Netzwerk mit \*\*ICMP-Echo-Paketen\*\* und misst die Round-Trip-Zeit (RTT) der Antwort . Es sendet fortlaufend Anfragen (standardmäßig, bis abgebrochen) und berechnet Statistik über Verlust und Latenz.

\*\*Syntax:\*\*

```
``` bash
```

```
ping [Optionen] <Ziel>
```

Hier kann <Ziel> ein Hostname oder eine IP-Adresse sein.

Wichtige Optionen:

- **-c <Anzahl>** – Anzahl der ICMP-Pakete festlegen. Z.B. ping -c 4 8.8.8.8 sendet genau 4 Pakete .
- **-i <Sekunden>** – Interval zwischen den Pings (Standard 1 Sekunde). Man kann z.B. -i 0.2 für fünf Pings pro Sekunde (root-Recht nötig bei <0.2).
- **-W <Timeout>** – Wartezeit in Sekunden auf eine Antwort, bevor als **timeout** gezählt (Default meist 1 Sekunde, bei -c am Ende kann man höher setzen).
- **-q** – "Quiet": keine per-Ping Ausgabe, nur eine Zusammenfassung am Ende.
- **-p <Pattern>** – Sende individuelles Byte-Muster im Paket (Hexadezimal angegeben). Gut für bestimmte Diagnosen (z.B. bestimmte Bits toggeln sehen).
- **-s <Packetsize>** – Größe des zu sendenden Pakets in Bytes (ohne ICMP-Header). Standard 56 Bytes (plus 8 Byte ICMP-Header = 64 Bytes in Ausgabe) .
- **-t <TTL>** – Setzt die Time-To-Live (IPv4) oder Hop Limit (IPv6) für die gesendeten Pakete. (Nützlich, um gezielt nur bis zu einem gewissen Hop zu gelangen, ähnlich traceroute Step).
- **-I <Interface>** – Bindet an ein bestimmtes Quell-Interface (oder IP) für Multi-homed hosts. Alternativ **-S <Source_Address>**.
- **-4, -6** – Erzwingt IPv4 oder IPv6.
- **-f** – "Flood ping": schickt so schnell wie möglich Pings (sehr viele pro Sekunde) und druckt einen Punkt pro empfangenes Echo. (Nur root, zur Stresstest oder schnelle Packetloss Erkennung).
- **-L** (bei einigen ping-Versionen) – Für Echo an Broadcast/Multicast-Adresse (ermöglicht Broadcast-ping, was standardmäßig blockiert ist).
- **-a** – Bei Empfang einen Terminal-Bell ausgeben (akustisches Signal).
- **-D** – Zeitstempel vor jeder Zeile (UNIX time).
- **-U** – Zeitstempel in Us seit Boot vor jeder Zeile (für Messungen).

Ausgabe-Interpretation:

Jede empfangene Antwort erscheint als Zeile:

```
64 bytes from 8.8.8.8: icmp_seq=1 ttl=117 time=8.34 ms
```

Bedeutet: Echo-Reply von 8.8.8.8 erhalten, Größe 64 Bytes, Sequenznummer 1, TTL im Reply 117, RTT 8.34 ms. Wenn Antworten ausbleiben, erscheinen Zeilen wie "Request timeout".

Am Ende kommt eine **Statistik**:

4 packets transmitted, 4 received, 0% packet loss, time 3005ms

rtt min/avg/max/mdev = 8.334/8.512/8.790/0.177 ms

Zeigt Verlust und RTT-Werte.

Beispiele:

- ping 1.1.1.1 – Kontinuierlicher Test zu 1.1.1.1 (Cloudflare DNS). Abbruch mit Strg+C.
- ping -c 5 www.example.com – 5 Pings an example.com, dann stop mit Ausgabe der Statistik.
- ping -l eth1 192.168.2.1 – Ping von Interface eth1 aus (wenn Multi-Interface).
- ping -s 1000 -c 3 host – Sende 3 Pings mit Paketgröße ~1000 Bytes (ICMP Payload).
- ping -6 -c 4 ipv6.google.com – IPv6 Ping zu Google.

traceroute (siehe oben)

(bereits behandelt in vorgehendem Teil, siehe „traceroute“ in Netzwerkdagnostik.)

tracepath (siehe oben)

(siehe „tracepath“ bei Netzwerkdagnostik.)

resolvectl (Systemd-resolved Query)

Beschreibung: CLI-Tool zum Abfragen des systemd-resolved-Dienstes (Namensauflösung). resolvectl kann DNS-Lookups durchführen, DNS-Settings anzeigen, LL/MNR Browsing etc. Ersetzt Tools wie systemd-resolve (veraltet).

Verwendung:

- **resolvectl status** – Zeigt die aktuellen DNS-Einstellungen pro Link (Interface): DNS-Server, Suche-Domains, LLMNR/MulticastDNS Status, DNSSEC, usw.
- **resolvectl query <Hostname>** – Löst einen Hostnamen auf (A/AAAA-Record). Ausgabe enthält v6 und v4 Adressen, CNAME falls, DNSSEC info, Cache info, antwortender Server etc.
Beispiel: resolvectl query example.com.

- **resolvectl query -t MX <Domain>** – DNS-Abfrage bestimmten Typs (hier MX-Records). Unterstützt -t ANY, A, AAAA, MX, SRV, SOA etc.
- **resolvectl query <IP>** – Reverse Lookup (IP zu Name via PTR).
- **resolvectl flush-caches** – Löscht den DNS-Cache von systemd-resolved.
- **resolvectl dns <Interface> [Server...]** – Setzt DNS-Server für ein bestimmtes Interface (über DBus). resolvectl domain <iface> ... analog. (Normal wird das via NetworkManager/Netplan/etc gemacht).
- **resolvectl revert <Interface>** – Löscht manuelle Overrides und kehrt zu vom Netzwerkdienst gelieferten DNS zurück.

Beispiel:

```
$ resolvectl query -t MX example.com
example.com IN MX 0 .
-- Information acquired via protocol DNS in 15.1ms.
```

-- Data is authenticated: no

Bedeutet example.com hat keinen MX (Mail).

resolvectl status Ausschnitt:

Global

Protocols: LLMNR=resolve mDNS=no DNSOverTLS=no DNSSEC=no/unsupported

resolv.conf mode: stub

Link 2 (eth0)

Current Scopes: DNS

Protocols: +DefaultRoute +LLMNR -mDNS DNSOverTLS=opportunistic DNSSEC=...

Current DNS Server: 192.168.1.1

DNS Servers: 192.168.1.1 1.1.1.1

DNS Domain: ~.

Zeigt z.B. that eth0 hat default DNS 192.168.1.1 plus Cloudflare, und ~. (default domain).

dig (DNS Lookup Utility)

Beschreibung: dig (domain information groper) führt DNS-Abfragen aus. Es ist Teil von BIND-Tools (bind-utils). Im Gegensatz zu host oder nslookup bietet dig ausführlich

formatierbare Antworten. Standard macht eine A-Record-Abfrage und gibt Antwort mit Abschnitt **ANSWER SECTION** aus.

Syntax:

dig [@DNS-Server] <Name> [<Typ>] [<Klasse>] [Optionen]

<Klasse> normalerweise IN (Internet), selten was anderes.

Wichtige Optionen:

- **@server** – Spezifischen DNS-Server benutzen statt System-resolver. Z.B. dig @8.8.8.8 example.com.
- **<Typ>** – Abfragetyp: A, AAAA, MX, TXT, SRV, NS, SOA, CNAME, PTR, ANY etc. (Default A).
- **+short** – Ausgabe nur die Antworten knapp (einfacher, scripterfreundlich) . Z.B. dig +short example.com A -> "93.184.216.34".
- **+noall +answer** – Unterdrückt alle außer Answer-Section (ähnlich short aber mit TTL und Info).
- **+trace** – Iterativer Trace durch DNS-Hierarchie (Root bis Ziel, ähnlich dig an Root und dann NS referrals folgen). Zeigt jeden Schritt. Hilfreich zum Debug.
- **+tcp** – Erzwinge TCP statt UDP (z.B. bei großen DNS-Records oder EDNS issues).
- **+nssearch** – Findet alle NS des Ziel und fragt diese nach SOA (um Delegation correctness zu checken).
- **-x <IP>** – Reverse lookup (PTR) für IP. (Bequem: dig -x 8.8.8.8).
- **+multi** – Schöne Ausgabe in Spalte falls mehrere TXT oder so.
- **+nodnssec oder +dnssec** – Anfordern oder nicht (um RRSIG etc. zu sehen).
- **+ttlshort** – Output TTL in short mode. (e.g. dig +short +ttl google.com -> "300 172.217.168.14").
- **+stats / +nostats** – Schaltet Statistik-Banner am Ende an/aus.
- **+cm** – Colorize output (if compiled with color support).

Beispiele:

- dig example.com ANY – Fragt alle Records (die server je nach config liefern).
- dig google.com MX +noall +answer – Zeigt nur MX Antworten.
- dig @1.1.1.1 cloudflare.com A – Abfrage an 1.1.1.1 nach cloudflare's A.
- dig -x 8.8.4.4 +short – Reverse Lookup, kurze Ausgabe ("dns.google.").

- dig +trace example.org – Startet vom Root Nameserver und verfolgt Delegation bis example.org resolved ist.

nslookup (Name Server Lookup - veraltet)

Beschreibung: Ein einfaches interaktives DNS-Abfrageprogramm. War Teil von BIND, gilt als veraltet im Vergleich zu dig. Viele Admins kennen es dennoch.

Verwendung:

```
nslookup [<Host> [<DNS-Server>]]
```

Ohne args geht es in interaktiven Modus. Dessen Kommandos: server <DNS>, set q=MX, set type=AAAA, set debug, exit.

Beispiele:

```
$ nslookup example.com 8.8.8.8
```

```
Server: 8.8.8.8
```

```
Address: 8.8.8.8#53
```

Non-authoritative answer:

```
Name: example.com
```

```
Address: 93.184.216.34
```

Reverse: nslookup 93.184.216.34 -> zeigt PTR.

Warnung: Neuere doc empfehlen dig oder host statt nslookup.

whois (Domain Whois Query)

Beschreibung: Fragt Whois-Datenbanken nach Registrierungsinformationen einer Domain oder 5】. Das Tool verbindet sich zum passenden Whois-Server (verwendet /etc/whois.conf, TLD heuristics) via TCP 43 und gibt den Text aus.

Syntax:

```
whois [Optionen] <DomainOrIP>
```

Ausgabe: Je nach TLD und Registry differierend. Enthält Domain-Inhaber (bei Generic oft DSGVO anonymisiert), Registrar, Erstellungs- und Ablaufdatum, Nameserver, Contact-IDs. Bei IP WHOIS (z.B. whois 8.8.8.8) gibt es Informationen über den zugeteilten IP-Block (ISP, RIR wie ARIN/RIPE...).

Optionen:

- **-H** – Unterdrückt das Banner "Whois Server" in RIPE responses.

- **-p <port>** – Andere Port.
- **--verbose** – Mehr Info über Server selection.
- **--help** – Display help.

Beispiele:

- whois example.com -> zeigt Registrar (ICANN info), Status, Name der Organisation, etc.
- whois 2620:0:2d0:200::7 -> IPv6 whois (gibt i.d.R. ARIN oder so mit OrgName).
- whois AS15169 -> WHOIS an RADb für Autonomous System 15169 (Google).

(*Beachte: Manche TLDs haben separate Tools, z.B. jwhois config. Das Standard whois ist meist in Debian whois package (Marco d'Itri) which covers a lot. GDPR hat Personendaten stark eingeschränkt in Output.*)

arp (Address Resolution Protocol table) – Hinweis: arp wird mittlerweile durch ip neigh ersetzt.

Beschreibung: Zeigt den ARP-Cache (IPv4) an oder manipuliert Einträge. (Linux modern: use ip neigh aber arp Kommando aus net-tools ist noch gebräuchlich).

Befehle:

- arp -a – Listet ARP-Einträge (Hostname, IP, MAC, Interface).
- arp -n – Wie -a, aber keine DNS für IP.
- arp -d <IP> – Löscht ARP-Eintrag für IP.
- arp -s <IP> <MAC> – Fügt statischen ARP-Eintrag hinzu (MAC fix für IP) – gleich ip neigh add ... nud permanent.

Beispiel: arp -a -> router.lan (192.168.1.1) at aa:bb:cc:dd:ee:ff [ether] on eth0.

Sicherheit und Firewall

iptables/iptables-nft (Packet Filter)

Beschreibung: Linux Kernel-Paketfilter (Firewall) via Netfilter-Schnittstelle. iptables verwaltet Regeln in verschiedenen Tabellen: **filter, nat, mangle, raw, security**, für IPv4. (Für IPv6, ip6tables). Mit Kernel 4.x/5.x viele Distros nutzen "iptables-nft" as a frontend to nftables, aber Syntax bleibt für user gleich.

Syntax Grundkommandos:

iptables [-t <table>] <operation> <chain> <matches...> <target>

Operationen: **-A** (append), **-I** (insert), **-D** (delete), **-C** (check exist), **-L** (list), **-F** (flush chain), **-N** (new chain), **-X** (delete user chain).

Standard Tabellen:

- **filter** (Default): Chains **INPUT, OUTPUT, FORWARD**.
- **nat**: Chains **PREROUTING, OUTPUT, POSTROUTING** (masquerading, port forwarding etc.).
- **mangle**: For packet alteration (TOS, mark), Chains in multiple points (PREROUTING, OUTPUT, etc.).
- **raw**: For conntrack exemption (NOTRACK), Chains PREROUTING, OUTPUT.
- **security**: SELinux related (rare).

Matches / Options:

- **-p <proto>**: Protocol (tcp, udp, icmp, icmpv6, all).
- **-s <IP/CIDR>, -d <IP/CIDR>**: Source or destination address (or network).
- **-i <iface>, -o <iface>**: Inbound or outbound network interface (for PREROUTING/INPUT/forward vs OUTPUT/POSTROUTING).
- **--sport, --dport** (with -p tcp/udp): Source/dest port or port range (e.g. --dport 80 or --dport 1000:2000).
- **-m state --state <states>**: State match (connection tracking states: INVALID, ESTABLISHED, NEW, RELATED). Common: allow ESTABLISHED,RELATED on INPUT for responses.
- **-m conntrack --ctstate**: (newer usage, same as state module now).
- **-m connlimit, -m limit** etc: Many modules. limit --limit 5/minute for rate-limit logging for example. connlimit --connlimit-above N limit parallel connections.
- **-m multiport**: match multiple ports in one rule if not contiguous, e.g. -m multiport --dports 80,443,8080.
- **-m tcp --tcp-flags SYN,RST,ACK SYN**: match specific TCP flags combination (here SYN only). Often used for --syn (alias) which matches syn packets (to identify new connections).
- **--comment "text"**: attach comment to rule (for documentation).

Targets (Actions):

- **ACCEPT**: Packet erlaubt.
- **DROP**: Verwerfen ohne Antwort.
- **REJECT**: Verwerfen mit Antwort (ICMP port unreachable o.Ä.).

- **LOG**: Loggt Paket (via kernel log), dann Verarbeitung geht weiter (so typ. Kombi LOG dann DROP). Options: --log-prefix "msg", --log-level warning.
- **MASQUERADE** (in nat POSTROUTING): SNAT für dynamische IP (Router Use-case) – ersetzt Source-IP durch Interface-IP.
- **SNAT** (in nat POSTROUTING): Statisches Source-NAT (requires --to-source IP[:port-range]).
- **DNAT** (in nat PREROUTING/OUTPUT): Destination-NAT (Port-Forwarding; --to-destination IP:port).
- **REDIRECT** (nat PREROUTING/OUTPUT): Umleiten an lokalen Host (for transparent proxy etc.).
- **MARK** (mangle): Markiert Paket mit fwmark (für routing decisions via ip rule, etc.).
- **RETURN**: In user-defined chain springt zurück zum Aufrufer-Chain (stop further rules in current chain). In builtin chain, RETURN = apply chain policy.
- **QUEUE**: Übergibt Paket an Userspace (via libnetfilter_queue).
- **DROP** vs. policy: If chain policy is DROP, falling off end = drop.

Beispiele:

- **List Regeln:** iptables -L -n -v (filter table). iptables -t nat -L -n -v NAT rules. -n no DNS, -v verbose (packet counters).
- **Allow incoming SSH:**

② iptables -A INPUT -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
 iptables -A OUTPUT -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
 (typical stateful allow SSH in).

② **Drop all forward by default:** iptables -P FORWARD DROP.

② **Masquerading (NAT):**

② iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
 (for home router with ppp0 outbound).

② **Port Forward:**

Forward ext eth0 port 80 to internal 192.168.1.100 port 8080:

② iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 \
 -j DNAT --to-destination 192.168.1.100:8080

```
iptables -A FORWARD -p tcp -d 192.168.1.100 --dport 8080 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
```

Logging Drop:

- iptables -A INPUT -p tcp --dport 23 -j LOG --log-prefix "Telnet attempt: " --log-level info
- iptables -A INPUT -p tcp --dport 23 -j DROP
- (log then drop telnet).

firewall-cmd (firewalld frontend)

Beschreibung: Steuerprogramm für **firewalld**, eine dynamische Firewall (standard in RHEL/CentOS 7+, Fedora, etc.). firewall-cmd ermöglicht das Konfigurieren von Zonen, Regeln (services, ports, masq) zur Laufzeit und in Config. Firewalld nutzt intern iptables/nftables.

Syntax:

```
firewall-cmd [--zone=<zone>] [--permanent] <aktion> <Parameter>
```

Grundkonzepte: Zonen (trusted, public, home, internal, work, external, dmz, block, drop etc.), Schnittstellen sind Zonen zugewiesen. Dienste (vordefinierte Regelsets) können pro Zone erlaubt werden oder einzelne Ports/Protokolle. Man kann Einstellungen permanent (persistiert in XML config) oder nur runtime vornehmen.

Wichtige Befehle:

- **--state:** Zeigt an, ob firewalld läuft (running/not running).
- **--get-active-zones:** Zeigt welche Zonen aktiv sind und welche Interfaces zugeordnet (z.B. "public: wlp2s0").
- **--get-zones:** Listet alle bekannten Zonen. **--list-all** [--zone=Z] listet alle Einstellungen einer Zone (services, ports, masquerade etc0】 .
- **--zone=<zone> --add-service=<service>:** Erlaubt den vordefinierten service in angegebener Zone (runtime). Bsp: firewall-cmd --zone=public --add-service=http – ermöglicht HTTP (Port 80) in Zone *public*.
- **--zone=<zone> --add-port=<port>/<proto>:** Erlaubt individuellen Port. Bsp: --add-port=5000/tcp.
- **--remove-service=..., --remove-port=...** analog zum Entfernen.
- **--add-masquerade:** Aktiviert NAT/Masquerading in der Zone (z.B. für Internet-sharing zone).
- **--query-service=<svc>:** Fragt ob Service in zone erlaubt (exit code 0/1).

- **--permanent:** Gibt man diese Option an, wird die Änderung in der persistenten Konfiguration vorgenommen (aber nicht an laufende sofort angewendet, es sei denn man ruft extra an). In firewalld, runtime und permanent config sind getrennt.
- **--reload:** Lädt die permanente Konfiguration neu (in runtime) – im Grunde wendet Config an.
- **--runtime-to-permanent:** Speichert aktuellen Laufzeit-Zustand als neue permanente Konfiguration (was man on-the-fly geändert hat, persistieren).
- **--zone=<zone> --change-interface=<iface>:** Weist ein Interface einer Zone zu (persistente Zuordnung).
- **--new-zone=<name>, --delete-zone=...:** Eigene Zonen erstellen/löschen.
- **--list-services, --get-services:** Listet bekannte Service-Namen (die in /etc/firewalld/services/*.xml definiert sind).
- **--service=<svc> --get-ports:** Zeigt welche Ports hinter einem Service stecken.
- **--add-rich-rule '<rule>':** Fügt sog. Rich Rule hinzu (eine erweiterte Regel-Syntax in firewalld, z.B. mit Logging, Zeitplänen). Z.B.:
 - `firewall-cmd --add-rich-rule='rule family="ipv4" source address="192.0.2.0/24" port port="22" protocol="tcp" accept'`
 - (Erlaubt SSH nur von 192.0.2.0/24).

Beispiele:

- Öffnen von HTTPS permanent:
 - `firewall-cmd --add-service=https --permanent`
 - `firewall-cmd --reload`
 -
- Port 12345/udp temporär öffnen in Zone public:
 - `firewall-cmd --zone=public --add-port=12345/udp` (nach Reboot weg).
- Interface eth0 zur Zone external zuweisen:
 - `firewall-cmd --zone=external --change-interface=eth0 --permanent && firewall-cmd --reload`.
- Status: `firewall-cmd --list-all` (wenn default zone, z.B. public, dann deren config).

ufw (Uncomplicated Firewall)

Beschreibung: Vereinfachtes Firewall-Frontend (Ubuntu/Debian) für iptables. ufw erlaubt einfache allow/deny-Regeln definieren, typischerweise nach Services oder Ports, optional in/out interface.

Syntax:

ufw [Optionen] <Aktion>

Grundbefehle:

- **ufw enable / disable** – Aktiviert oder deaktiviert die Firewall (bei disable: setzt iptables default ACCEPT und löscht Regeln).
- **ufw status [verbose]** – Zeigt aktuelle Regel-Liste. Mit verbose sieht man auch Interfaces, logging status etc.
- **ufw default allow|deny [incoming|outgoing|routed]** – Setzt Standardpolicy. Standard-Einstellung: incoming deny, outgoing allow.
- **ufw allow <Port/Service> [proto tcp/udp] [from <IP> [to <IP>]] [port <P2>]** – Erlaubt eingehenden Traffic. Beispiele:
 - ufw allow 22 (erlaubt TCP/22 auf allen Interfaces).
 - ufw allow proto udp from 10.0.0.0/8 to any port 53 (erlaubt UDP DNS von privatem Netz).
 - ufw allow out 80/tcp (erlaubt ausgehende HTTP).
 - ufw allow in on eth0 to any port 443 (erlaubt eingehendes HTTPS auf eth0).
- **ufw deny <Spec>** – Blockiert (mit REJECT) entsprechend.
- **ufw reject <Spec>** – Synonym zu deny (deny und reject meist ähnlich in ufw; auf Debian might differ).
- **ufw limit <Port/Service>** – Wie allow, aber mit Rate-Limit (gegen Bruteforce). Z.B. ufw limit ssh erlaubt SSH, aber blockiert IP temporär bei >6 Verbindungen/30s.
- **ufw delete <RuleSpec>** – Entfernt Regel (man kann Regel genau wie bei allow angeben, mit 'delete' davor, oder Nummer aus ufw status numbered).
- **ufw logging on|off|LEVEL** – Logging ein/aus und Level (off, low, medium, high, full).
- **ufw reset** – Setzt ufw auf Auslieferungszustand (alle Regeln weg, policies reset, und disabled).
- **ufw route allow/deny ...** – Regeln für Forwarded Traffic (Routen). Bsp: ufw route allow in on eth1 out on eth0 to any port 80 for forwarded.

- **ufw insert <num> <rule>** – Fügt Regel an bestimmter Position ein.
- **ufw show raw** – Zeigt resultierende iptables-Regeln.

Beispiele:

- Standard: ufw default deny incoming; ufw default allow outgoing; ufw enable.
- Webserver: ufw allow 'Nginx Full' (in /etc/ufw/applications.d vordefinierte Profile, 'Apache Full' etc., Nginx Full = 80+443).
- Only LAN access to a service: ufw allow from 192.168.0.0/16 to any port 3306 (MySQL offen nur intern).
- Block eine IP: ufw deny from 203.0.113.5.
- Show rules: ufw status verbose -> e.g.
- Status: active
- Logging: on (low)
- Default: deny (incoming), allow (outgoing), deny (routed)
- New profiles: skip
-
- To Action From
- -- ----- -----
- 22/tcp LIMIT Anywhere # SSH
- 80,443/tcp ALLOW Anywhere # WWW
- 22/tcp (v6) LIMIT Anywhere (v6)
- 80,443/tcp (v6) ALLOW Anywhere (v6)
-

(UFW ist beliebt wegen einfacher Syntax, aber fehlender Komplexität bei komplexen Setups. Firewalld und raw iptables/nft sind mächtiger in Feinheiten.)

nft (nftables Firewall)

Beschreibung: nftables ist der modernere Packet-Filter (ersetzt iptables). Der Befehl nft wird zum Konfigurieren verwendet. Sehr flexibel (Datenstrukturen, one-rule multiple matches).

Syntax:

nft [options] <command>

Man kann Kommandos interactive eingeben oder als Skript. Der Kommandostil ähnelt C:
e.g.

```
nft add table inet filter
```

```
nft add chain inet filter input { type filter hook input priority 0 \; }
```

```
nft add rule inet filter input tcp dport 22 accept
```

(Librally, but given complexity and that the original question references iptables and ufw, likely beyond scope).

Abschluss

Dieses Kompendium deckt die wesentlichen Befehle für die Linux+ (XK0-005) Prüfung ab, gegliedert nach Themengebieten. Die Befehle umfassen Benutzer- und Rechteverwaltung, Prozess- und System-Tools, Netzwerk-Konfiguration und -Diagnose, Software-Installation, Container, Sicherheit (Firewall, SELinux/AppArmor siehe oben Rich-Rules, etc.), Dienste und Planung.

Durch die strukturierte Darstellung mit Syntax, Beschreibung, Optionen und Beispielen dient es als umfassendes Nachschlagewerk. **Wichtig:** In einer Prüfungssituation sollte man sich auf die korrekte Anwendung der Optionen und das Verständnis der Konzepte dahinter konzentrieren (z.B. was macht chmod 775 oder umask 002 bewirkt).

Mit dieser Referenz sollte die Vorbereitung auf typische Linux-Aufgaben – vom Anlegen eines Benutzers, dem Ändern von Dateirechten, über das Einrichten eines Cronjobs bis zum Konfigurieren eines Apache-Dienstes oder Setzen einer Firewallregel – bestens unterstützt werden.

Kompendium der Linux+ (XK0-005) Befehle

Dieses Nachschlagewerk umfasst alle wichtigen Befehle aus dem offiziellen **Command Reference Guide** für die CompTIA Linux+ Prüfung (XK0-005). Die Befehle sind thematisch gegliedert (Benutzerverwaltung, Netzwerk, Prozesse, Dateisystem, Paketmanagement, Container, Sicherheit etc.), mit Beschreibung, vollständiger Syntax, allen verfügbaren Optionen und ggf. wichtigen Unterbefehlen. Es dient als umfassendes Cheat-Sheet zur Prüfungsvorbereitung.

Grundlegende Linux-Befehle

ls (List Directory Contents)

Beschreibung: Listet den Inhalt eines Verzeichnisses auf. Ohne Parameter zeigt ls die nicht versteckten Einträge im aktuellen Verzeichnis in alphabetischer Reihenfolge wiki.ubuntuusers.de/ionos.de.

Syntax:

```
ls [OPTION]... [VERZEICHNIS]
```

Wenn kein Verzeichnis angegeben wird, wird das aktuelle Verzeichnis aufgelistetwiki.ubuntuusers.de.

Optionen:

Option	Beschreibung
-a, --all	Zeigt alle Einträge an, inklusive versteckter Dateien (auch . und ..) wiki.ubuntuusers.deionos.de .
-A, --almost-all	Wie -a, aber ohne . und .. – listet versteckte Dateien außer den Verzeichnissen ./.. wiki.ubuntuusers.deionos.de .
-l	Langformat: Detaillierte Ausgabe (Rechte, Besitzer, Größe, Datum usw.) je Eintrag wiki.ubuntuusers.deionos.de .
-d, --directory	Listet Verzeichniseintrag selbst statt dessen Inhalt (nützlich, um Infos über Ordner zu sehen) wiki.ubuntuusers.deionos.de .
-h, --human-readable	Im Langformat (-l) Größen in menschenlesbare Einheiten (KB, MB ...) anzeigen wiki.ubuntuusers.deionos.de .
-i	Zeigt die Inode-Nummer jedes Eintrags an wiki.ubuntuusers.deionos.de .
-C	Ausgabe in Spalten nebeneinander (Standardformat) wiki.ubuntuusers.deionos.de .
-m	Listet Einträge kommasepariert in einer Zeile wiki.ubuntuusers.deionos.de .
-R, --recursive	Rekursiv: Zeigt auch Inhalte von Unterverzeichnissen (durchläuft Verzeichnisbaum) wiki.ubuntuusers.deionos.de .
-t	Sortiert nach Zeitstempel (Änderungszeit, neueste zuerst) anstatt alphabetisch.
-r	Kehrt die Sortierreihenfolge um (Reverse order).
--hyperlink	Macht Dateinamen (in unterstützten Terminals) als klickbare Links ausgebbar wiki.ubuntuusers.de .
--help	Zeigt die Hilfe zum Befehl an.
--version	Zeigt Versionsinformationen von ls.

(Weitere Optionen sind in den Manpages dokumentiert)wiki.ubuntuusers.de.

cat (Concatenate and Print Files)

Beschreibung: Gibt den Inhalt von Dateien auf der Standardausgabe aus oder verknüpft (konkateniert) mehrere Dateien. Oft verwendet, um Textdateien anzuzeigen oder Inhalte zu pipen[ionos.at](#). Der Name steht für „concatenate“ (verketten).

Syntax:

cat [OPTION]... [DATEI]...

Ohne Optionen schreibt cat den Inhalt der angegebenen Datei(en) in der angegebenen Reihenfolge auf die Standardausgabe. Kann mehrere Dateien hintereinander ausgeben oder mit Umleitungen/ Pipes genutzt werden.

Optionen:

Option Beschreibung

- n** Nummeriert *alle* ausgegebenen Zeilen fortlaufend[ibm.comionos.at](#).
- b** Nummeriert nur **nicht-leere** Ausgabelinien (Leerzeilen werden nicht nummeriert)[ibm.comionos.at](#).
- s** Squeeze: Unterdrückt aufeinanderfolgende Leerzeilen (mehrere leere Zeilen werden als eine ausgegeben)[ionos.at](#).
- v** “Visible” – Macht nicht druckbare Steuerzeichen sichtbar (außer Tab und Zeilenende)[ionos.at](#).
- e** Entspricht -v mit Anzeige des Zeilenende-Zeichens \$ am Zeilenende (kombiniert -v und Kennzeichnung von Zeilenumbrüchen)[ionos.at](#).
- t** Entspricht -v mit Darstellung von Tabulatoren als ^I (Tab-Markierung)[ionos.at](#).
- E** Zeigt \$ am Ende jeder Zeile (Äquivalent zu -vE, oft identisch zu -e je nach Implementierung).
- T** Zeigt Tabulatoren als ^I an (ähnlich -vT, meist wie -t).
- A** Kombiniert mehrere der obigen: entspricht oft -vET (alle nicht-druckbaren Zeichen, Tabs und Zeilenenden sichtbar machen).
- u** (Ignoriert in GNU cat – historisch für ungepufferte Ausgabe).
- help** Zeigt Hilfe-Informationen an.
- version** Zeigt Versionsinfo an.

cd (Change Directory)

Beschreibung: Wechselt in ein anderes Verzeichnis. Mit cd kann das aktuelle Arbeitsverzeichnis geändert werden („change directory“)[ionos.at](#). Ohne Argument wechselt cd standardmäßig ins Heimatverzeichnis des Benutzers.

Syntax:

cd [OPTION] [VERZEICHNIS]

Man kann relative Pfade (bezogen auf das aktuelle Verzeichnis) oder absolute Pfade (ausgehend vom Wurzelverzeichnis /) angeben. Erfolgt der Wechsel in ein Zielverzeichnis, so wird dieses das neue aktuelle Arbeitsverzeichnis; bei nicht vorhandenen Zielen oder fehlenden Rechten bleibt das Verzeichnis unverändert[ionos.at](#).

Optionen:

Option Beschreibung

- L Folge symbolischen Links **logisch**. Dies ist die Voreinstellung. (Beim Navigieren durch einen Symlink bleibt man in der logischen Pfadstruktur)[ionos.at](#).
- P Folge symbolischen Links **physisch**. Wechselt in das tatsächliche Ziel im Dateisystem und löst symbolische Links auf[ionos.at](#).

Unterbefehle / spezielle Aufrufe:

- cd ~ wechselt ins Heimatverzeichnis des aktuellen Benutzers[ionos.at](#).
- cd - wechselt in das vorherige Arbeitsverzeichnis zurück (Toggle zwischen zwei Verzeichnissen)[ionos.at](#).
- cd /Pfad/zu/Verz wechselt absolut in ein angegebenes Verzeichnis vom Wurzelpfad aus (hier z.B. nach /Pfad/zu/Verz).
- cd .. wechselt eine Ebene höher (ins Parent-Verzeichnis), cd ../../ entspricht zwei Ebenen usw.
- cd . bleibt im aktuellen Verzeichnis (. bezeichnet das aktuelle Verzeichnis, meistens kein Effekt).

pwd (Print Working Directory)

Beschreibung: Gibt den vollständigen Pfad des aktuellen Arbeitsverzeichnisses aus[ionos.at](#). Mit pwd kann man sich anzeigen lassen, *in welchem Verzeichnis man sich gerade befindet*.

Syntax:

pwd [OPTION]...

Optionen:

Option **Beschreibung**

- Ausgabe des Pfads mit Berücksichtigung symbolischer Links (logischer Pfad, Standard). Wenn z.B. ein symbolischer Link als aktuelles Verzeichnis dient, wird der Linkpfad angezeigt[ionos.de](#).
- P** Ausgabe des *physischen* Pfads: Symbolische Links werden aufgelöst, der tatsächliche Verzeichnispfad wird angezeigt[ionos.de](#).
- help** Zeigt Hilfe-Informationen zu pwd an.
- version** Zeigt Versionsinformationen an.

Hinweis: pwd ist häufig als Shell-Builtin implementiert. Das externe Programm /bin/pwd unterstützt ggf. andere oder weniger Optionen (bei GNU Coreutils nur --help und --version zusätzlich)[ionos.de](#). In der Regel genügt der Shell-eigene pwd Befehl.

whoami

Beschreibung: Zeigt den Benutzernamen des aktuell angemeldeten Users anfile-4fdk7rvx39azjretinarqb. Praktisch, um schnell zu prüfen, unter welchem Account ein Prozess läuft.

Syntax:

whoami

Keine spezifischen Optionen. Der Befehl gibt äquivalent echo \$USER aus (den Namen des aktuellen Unix-Accounts).

touch

Beschreibung: Erzeugt eine neue leere Datei oder ändert Zeitstempel (Zugriffs- und Änderungszeit) bestehender Dateien. Wenn die Datei nicht existiert, wird sie (sofern nicht -c verwendet wird) als leere Datei angelegt[man7.org](#). Oft genutzt, um Dateien schnell anzulegen oder deren *timestamp* zu aktualisieren.

Syntax:

touch [OPTION]... DATEI...

Optionen:

Option	Beschreibung
-a	Ändert nur die Zugriffszeit (atime) der Datei man7.org .
-m	Ändert nur die Änderungszeit (mtime) der Datei man7.org .

Option	Beschreibung
-c, --no-create	Legt keine neue Datei an. Wenn die Datei nicht existiert, wird kein neues leeres File erstellt (ansonsten Standardverhalten ohne -c) man7.org .
-d, --date=STRING	Verwendet den angegebenen Datums-String anstelle der aktuellen Zeit. Der String wird geparsst (z.B. "2025-05-01 12:00:00" oder relative Angaben wie "next Friday" sind möglich) man7.org .
-t [[CC]YY]MMDDhhmm[.ss]	Verwendet einen konkreten Zeitstempel im angegebenen Format (JJMMTThhmm[.ss]) anstatt der aktuellen Zeit man7.org . Beispiel: -t 202312312359.59 setzt 31.12.2023 23:59:59.
-r, --reference=DATEI	Übernimmt die Zeitstempel (Zugriffs- und Änderungszeit) der angegebenen Referenzdatei. Die Zielfile erhält also dieselben Zeiten wie DATEI man7.org .
-h, --no-dereference	Ändert bei symbolischen Links <i>den Link selbst</i> (deren Metadaten), nicht die referenzierte Datei. Ohne diese Option würde touch auf den Ziel-Dateipfad wirken (nicht auf den Symlink). (<i>Diese Option ist nur wirksam, wenn das System das Verändern von Symlink-Zeitstempeln unterstützt.</i>) man7.org
--time=WORD	Bestimmt, welcher Zeitstempel geändert wird. Mögliche Werte: atime, access, use (entspricht -a), oder mtime, modify (entspricht -m) man7.org .
--help, --version	Anzeigen der Hilfe bzw. Versionsinformation zu touch.

man (Manual Page Viewer)

Beschreibung: Zeigt die Handbuchseite (Manual Page) für einen angegebenen Befehl oder ein Thema an. man ist das Offline-Hilfe- und Dokumentationssystem unter Unix/Linux.

Syntax:

man [SECTION] <Thema/Command>

Beispiel: man ls zeigt die Manpage zu **ls**. Manpages sind meist nach Kategorien/Abschnitten (1=Benutzerkommandos, 5=Dateiformate, 8=Administrator-Kommandos etc.) gegliedert – optional kann man mit man 5 passwd z.B. direkt die Manpage zum Dateiformat **passwd** (Abschnitt 5) öffnen.

Optionen:

Option	Beschreibung
-k Suchwort	Durchsucht die Manpage-Kurzzusammenfassungen nach <i>Suchwort</i> (entspricht dem Befehl apropos). Beispiel: man -k network listet alle Manual-Topics auf, deren Beschreibung "network" enthält.
-f Befehl	Findet den Namen und Abschnitt einer Manpage zu <i>Befehl</i> (entspricht whatis). Beispiel: man -f passwd zeigt kurz "passwd (1) - change user password".
-a	Zeigt nacheinander alle Manpages an, die zum angegebenen Namen passen (wenn z.B. ein Name in mehreren Abschnitten existiert).
-P Pager	Legt ein anderes Pager-Programm fest (Standard ist less). Beispiel: man -P cat ls würde die ls-Manpage durch cat leiten.
--help, --version	Zeigt Hilfe bzw. Version von man.

whatis

Beschreibung: Zeigt eine kurze Ein-Satz-Beschreibung zu einem Befehl an (frägt die Datenbank der whatis-Einträge ab). Es ist eine Kurzform von man -f und liefert die Synopsis aus der Manpage.

Syntax:

whatis <Befehl/Topic>

Optionen:

Option	Beschreibung
(keine üblichen Optionen)	whatis wird meist ohne Optionen verwendet. Es greift auf die gleichen Datenbanken wie apropos/man -k zu. Sollte die Datenbank nicht aktuell sein, kann sie mit mandb erneuert werden.

Benutzer- und Gruppenverwaltung

passwd (Passwort ändern/setzen)

Beschreibung: Ändert das Passwort eines Benutzerkontos oder verwaltet Konto-Sperren. Als normaler Benutzer kann man mit passwd sein eigenes Passwort ändern; als root kann man mit passwd <user> das Passwort eines anderen Accounts setzen oder bestimmte Eigenschaften (Sperren, Ablaufdatum) ändern.

Syntax:

`passwd [OPTION] [BENUTZER]`

Standardaufruf ohne Benutzer wirkt auf den aktuellen Benutzer. Bei Angabe eines Benutzernamens werden dessen Passwortinformationen geändert (Root-Rechte erforderlich).

Optionen:

Option Beschreibung

- Sperrt das Benutzerkonto, indem das Passwort in /etc/shadow mit einem **-l, --lock** Präfix (meist ! oder *) unbrauchbar gemacht wirdwiki.ubuntuusers.de. Der Benutzer kann sich danach nicht mehr anmelden, bis das Konto wieder entsperrt wird.
- Entsperrt ein zuvor gesperrtes Konto (entfernt das Sperr-Präfix im **-u, --unlock** verschlüsselten Passwort)wiki.ubuntuusers.de.
- Löscht das Passwort des Benutzers (leer setzen). Achtung: Damit wird eine **-d, --delete** passwortlose Anmeldung möglich, sofern nicht andersweitig gesichertwiki.ubuntuusers.de.
- Markiert das Passwort als abgelaufen. Beim nächsten Login muss der **-e, --expire** Benutzer ein neues Passwort setzen.
- Setzt die maximale Gültigkeitsdauer eines Passworts auf *TAGE*. Nach Ablauf **-x TAGE** muss es geändert werden. (Änderung des PASS_MAX_DAYS in /etc/shadow.)
- Setzt die minimale Anzahl an Tagen zwischen zwei Passwortänderungen **-n TAGE** (PASS_MIN_DAYS). Der Benutzer kann sein Passwort erst nach *TAGE* Tagen erneut ändern.
- Warnzeit in Tagen vor Passwortablauf (PASS_WARN_AGE). Der Benutzer erhält *TAGE* Tage vor Ablauf eine Warnung beim Login.
- Zeigt den Passwortstatus an (Gesperrt? nicht gesetzt? Datum der letzten Änderung usw.). `passwd -S <user>` gibt eine Statuszeile aus.

Interaktives Verhalten: Wird `passwd` ohne Optionen aufgerufen, fordert es zur Eingabe des aktuellen Passworts (zur Authentifizierung, falls nicht root) und zweimal zur Eingabe des neuen Passworts auf (zwecks Bestätigung). Es verweigert einfache Passwörter gemäß der System-Policy (Pluggable Authentication Modules/pam Richtlinien). Root-Benutzer muss das alte Passwort nicht kennen.

chage (Change Age – Passwortalter und Gültigkeit)

Beschreibung: Verändert Einstellungen zum Passwort-Aging – also Ablaufdatum, Mindestalter, Warnzeit etc. Mit chage kann ein Administrator festlegen, wann ein Benutzer sein Passwort ändern muss bzw. wann der Account deaktiviert wirdfile-4fdk7rvx39azretinarqb.

Syntax:

chage [OPTION]... BENUTZER

Optionen:

Option Beschreibung

-l Listet die aktuellen Passwort- und Alterseinstellungen für den Benutzer in einem lesbaren Format auf (Letzte Änderung, Min/Max Alter, Warnzeit, Inaktivzeit, Ablaufdatum) – Übersicht.

-m TAGE Setzt die minimale Anzahl von Tagen *TAGE*, die ein Passwort gültig sein muss, bevor es geändert werden darf (PASS_MIN_DAYS).

-M TAGE Setzt die maximale Gültigkeitsdauer in Tagen (PASS_MAX_DAYS). Nach *TAGE* Tagen läuft das Passwort ab und muss geändert werden.

-w TAGE Setzt die Anzahl der Tage *TAGE* Vorwarnzeit vor Passwortablauf (PASS_WARN_AGE).

-I TAGE Legt fest, wie viele Tage nach Passwortablauf der Account **inaktiv** wird (PASS_INACTIVE_DAYS). Ein Wert von 0 deaktiviert den Account sofort nach Ablauf des Passworts, -1 schaltet diese Funktion aus[wiki.ubuntuusers.de](http://wiki.ubuntuusers.de/wiki.ubuntuusers.de).

-E DATUM Setzt ein Ablaufdatum für das Benutzerkonto (im Format YYYY-MM-DD). Nach diesem Datum wird der Account deaktiviertwiki.ubuntuusers.de. Das Datum -1 (oder "" leer) entfernt ein gesetztes Ablaufdatum (kein Ablauf).

-d YYYY-MM-DD Setzt das Datum der letzten Passwortänderung auf das angegebene Datum (wirkt wie Backdating oder um zu triggern, dass ein Passwort abläuft). Kann auf 0 gesetzt werden, um das Passwort als sofort ablaufend zu markieren.

-h, --help Zeigt Hilfe an.

Hinweis: Viele dieser Einstellungen lassen sich auch mit passwd (Optionen -x, -n, -w, -i, -e) vornehmen. chage bietet eine dedizierte Schnittstelle dafür. Ohne Optionen (als root) startet chage <user> interaktiv einen Dialog, um alle Werte einzustellen.

w

Beschreibung: Zeigt an, welche Benutzer gerade am System angemeldet sind und was sie gerade tun. w fasst Informationen über aktuell eingeloggte User und deren laufende Prozesse zusammen (inkl. Login-TTY, Leerlaufzeit, genutzte CPU-Zeit, aktueller Befehl)file-4fdk7rvx39azjretinarqb.

Syntax:

w [OPTION]

Optionen:

Option Beschreibung

- h Blendet die Kopfzeile in der Ausgabe aus (unterdrückt die Überschriftenzeile).
- s Kurze Ausgabe. Unterdrückt die Spalten für Login-Zeit, Leerlauf etc. (nicht auf allen Systemen vorhanden).
- f Zeigt Informationen zu von utmp als *geflagt* markierten Einträgen. Auf einigen Systemen steuert dies die Anzeige von Remote-Host oder Exit-Informationen. (Bei neueren procps-Versionen hat -f keine Funktion, da Host immer gezeigt wird.)
- V, --version Zeigt Versionsinfo von w.

Die Ausgabe von w enthält in der ersten Zeile System-Uptime und Load Average. Danach folgen pro eingeloggtem Benutzer: Benutzername, Terminal, Remote-Host, Login-Zeit, Leerlaufdauer, CPU-Verbrauch und der aktuell laufende Prozess (TTY-Prozess).

who

Beschreibung: Zeigt Informationen über die aktuell angemeldeten Benutzer. Im Gegensatz zu w liefert who primär Login-Zeitpunkte und Terminalsessions, jedoch keine Prozess- oder Idle-Infos.

Syntax:

who [OPTION]... [Datei]

Standard liest who die System-Login-Datenbank /var/run/utmp und gibt für jeden Eintrag Benutzer, Terminal, Datum/Zeit und ggf. Host aus.

Optionen:

Option Beschreibung

- H Zeigt eine Kopfzeile (Überschriften) über den Spalten an.

Option	Beschreibung
-q	"Quick who": Zeigt nur Benutzernamen und die Anzahl der angemeldeten Nutzer. (Gleichbedeutend mit users-Befehl.)
-b	Zeigt die Zeit des letzten Systemstarts (Boot time) an.
-r	Zeigt den aktuellen Runlevel des Systems.
-a	Zeigt <i>alle</i> Einträge aus utmp, inklusive Einträgen für laufende Prozesse am System (wie getty oder systemd-Spawns). Kombination von mehrere Flags: entspricht oft -b -d --login -p -r -t -T -u je nach Implementierung.
-m	Zeigt nur die Einträge für das aktuelle Terminal an (beschränkt auf den Benutzer, der den Befehl aufruft).
-u	Zeigt zusätzlich die Leerlaufzeit (Idle) und den PID des Login-Prozesses für jeden Benutzer.
-T	Zeigt den Terminal-Status (\pm Flag) an: + bedeutet terminal ist bei mesg schreibbar, - bedeutet nicht schreibbar, ? unbekannt. (In vielen who-Implementierungen bereits Standardverhalten ohne Option.)
--help, --version	Anzeigen von Hilfe bzw. Versionsinfo.

useradd (Benutzer anlegen)

Beschreibung: Legt ein neues Benutzerkonto an. useradd ist ein Low-Level-Befehl, der die Systemkontodateien (wie /etc/passwd, /etc/shadow, /etc/group) direkt verändert, um einen neuen User einzurichten [pq.hosting](#). In Debian/Ubuntu-Umgebungen wird oft das Wrapper-Skript adduser verwendet, das auf useradd aufbaut und interaktiv ein paar Dinge abfragt; useradd selbst ist nicht interaktiv und benötigt alle Optionen zum Festlegen der Eigenschaften des neuen Accounts.

Syntax:

useradd [OPTION]... BENUTZERNAME

Optionen: (häufigste Optionen beim Anlegen eines Benutzers)

Option	Beschreibung
-c	Kommentarfeld (GECOS) für den Benutzer, z.B. voller Name oder Hinweis. Kommentar Wird in /etc/passwd eingetragen wiki.ubuntuusers.de .

Option	Beschreibung
-d HOME	Pfad des Home-Verzeichnisses. Standard: /home/<Benutzername>. Kann geändert werden; wenn -m verwendet wird, wird dieses Verzeichnis erstellt und mit Standard-Dateien aus /etc/skel befüllt wiki.ubuntuusers.de .
-m	Legt das Home-Verzeichnis an und kopiert Standard-Dateien hinein. Sollte gesetzt sein, außer man möchte <i>kein</i> Home-Verz. (Ohne -m wird das Home nicht automatisch erstellt) wiki.ubuntuusers.de .
-s SHELL	Login-Shell des Benutzers. Standard ist normalerweise /bin/bash (oder was in /etc/default/useradd als SHELL definiert ist). Kann z.B. auf /sbin/nologin gesetzt werden für System-Accounts wiki.ubuntuusers.de .
-u UID	Gewünschte User-ID (numerisch). Darf nicht bereits vergeben sein (außer mit -o). Wenn nicht angegeben, wird die nächste freie UID > 999 gewählt wiki.ubuntuusers.de .
-o	Erlaubt die Verwendung einer bereits vergebenen UID (UID-Doppelungen). Zusammen mit -u benutzt, um zwei Benutzer mit identischer UID anzulegen. (Normalerweise nicht empfohlen, da UID eindeutig sein sollen.)
-g GRP	Primäre Gruppe (Name oder GID), der der Benutzer angehört wiki.ubuntuusers.de . Standard: eine neue Gruppe mit gleichem Namen wie der Benutzer (bei entsprechender Systemkonfiguration) oder eine in /etc/default/useradd definierte GROUP. Die angegebene Gruppe muss existieren.
-G Liste	Liste zusätzlicher Gruppen (sekundäre Gruppen), denen der Benutzer angehören soll wiki.ubuntuusers.de . Gruppennamen durch Komma getrennt angeben (ohne Leerzeichen).
-p Passwort	Das verschlüsselte Passwort für den Account (wie es z.B. von openssl passwd oder crypt(3) erzeugt würde) wiki.ubuntuusers.de . Wichtig: Aus Sicherheitsgründen sollte man hier <i>nicht</i> ein Klartextpasswort übergeben. Bleibt -p leer, wird der Account ohne gesetztes Passwort angelegt (Anmeldung evtl. gesperrt bis passwd gesetzt wird).
-e	(bei useradd nicht üblich – wird bei usermod benutzt für Ablauf setzen. Bei einigen Systemen könnte useradd -e <Datum> ein Ablaufdatum setzen.)

Option	Beschreibung
-f TAGE	Anzahl an Tagen nach Passwortablauf, bis der Account deaktiviert wird (Analog zu chage -l). -f 0 würde den Account sofort mit Passwortablauf sperren. Standard -1 (nie sperren).
-N	(Bei einigen Systemen: <i>No user group</i> – verhindert das automatische Anlegen einer gleichnamigen Gruppe.)
-r	Erstellt einen System-Account (UID unter einer gewissen Schwelle, z.B. <1000, kein Expire, kein Mail-Spool). Oft genutzt für Systembenutzer ohne Login.
-M	Kein Home-Verzeichnis anlegen (selbst wenn Default CREATE_HOME aktiviert ist).
-k SKELDIR	Gibt ein alternatives Skeleton-Verzeichnis an (anstatt /etc/skel), von wo aus bei -m die initialen Dateien ins Home kopiert werden.
-D	Zeigt die aktuellen Standardwerte für useradd an (oder mit anderen -D Optionen können diese verändert werden).

Nach dem Anlegen mit useradd sollte meist ein Passwort gesetzt werden (passwd <user>), außer es handelt sich um einen Account ohne Login.

usermod (Benutzerkonto ändern)

Beschreibung: Modifiziert ein bestehendes Benutzerkonto – z.B. Name, UID, Primärgruppe, zusätzliche Gruppen, Login-Shell, Expire-Datum usw.[wiki.ubuntuusers.de](http://wiki.ubuntuusers.de/wiki.ubuntuusers.de). Im Grunde die gleichen Optionen wie useradd, aber angewendet auf bestehende Benutzer. Häufig genutzt, um Benutzer umzubenennen, in andere Gruppen aufzunehmen oder Accounts zu sperren.

Syntax:

usermod [OPTION]... LOGIN

Optionen:

Option	Beschreibung
-c Kommentar	Neuen Kommentar (GECOS-Feld) setzen wiki.ubuntuusers.de .
-d Neues_Home	Ändert das Home-Verzeichnis des Benutzers auf <i>Neues_Home</i> . Hinweis: Ohne weiteres passiert das alte Home nicht automatisch mit. Mit zusätzlicher Option -m werden die Inhalte des alten Home-Verz. ins neue Verzeichnis verschoben und das neue Verzeichnis ggf. erstellt wiki.ubuntuusers.de .

Option	Beschreibung
-m	(Mit -d verwenden) Verschiebt beim Ändern des Homepfads (-d) die Dateien vom alten ins neue Home und erstellt es falls nötig wiki.ubuntuusers.de .
-l Neuer_Login	Ändert den Login-Namen des Benutzers (Umbenennung des Accounts) wiki.ubuntuusers.de . Der Benutzername in allen Systemkontodateien wird angepasst, <i>nicht</i> jedoch automatisch der Name des Home-Verzeichnisses oder Mail-Spool (diese müssten manuell umbenannt werden, falls gewünscht).
-u Neue_UID	Ändert die numerische Benutzer-ID auf <i>Neue_UID</i> . Falls die UID bereits vergeben ist, muss -o benutzt werden. Dateien im Home-Verzeichnis mit alter UID werden auf neue UID umgestellt; Dateien außerhalb des Homes muss der Admin selbst anpassen wiki.ubuntuusers.dewiki.ubuntuusers.de .
-o	Erlaubt die Verwendung einer bereits vergebenen UID (nur zusammen mit -u sinnvoll).
-g Neue_PrimGruppe	Ändert die primäre Gruppenzugehörigkeit (Name oder GID) manpages.ubuntu.com . Die neue Gruppe muss existieren. Alle Dateien im Home, die der alten primären Gruppe gehörten, werden auf die neue Gruppe übertragen manpages.ubuntu.com . (Dateien außerhalb des Homes müssen ggf. manuell angepasst werden.)
-G Gruppe(n)	Setzt die Liste der sekundären Gruppen neu für den Benutzer manpages.ubuntu.com . <i>Wichtig:</i> Ohne -a entfernt diese Option den Benutzer aus allen Gruppen, die nicht in der Liste stehen wiki.ubuntuusers.demanpages.ubuntu.com . Um einen Benutzer zusätzlichen Gruppen hinzuzufügen, sollte man -a -G gemeinsam verwenden (siehe -a).
-a	<i>Append</i> – fügt den Benutzer zusätzlichen Gruppen hinzu, anstatt die Gruppenzugehörigkeit zu überschreiben wiki.ubuntuusers.dewiki.ubuntuusers.de . Nur in Kombination mit -G verwenden , um den Benutzer der/die angegebenen Gruppen hinzuzufügen, ohne ihn aus anderen zu entfernen. (Siehe Warnung unten.)
-s Shell	Ändert die Login-Shell des Benutzers wiki.ubuntuusers.de . Z.B. auf /sbin/nologin setzen, um interaktive Logins zu unterbinden.

Option	Beschreibung
-e YYYY-MM-DD	Setzt das Ablaufdatum des Accounts (das Datum, an dem das Konto deaktiviert wird) wiki.ubuntuusers.de . Nach diesem Datum kann sich der User nicht mehr anmelden. usermod -e "" user entfernt ein Ablaufdatum (setzt es auf <i>nie</i>).
-f Tage	Setzt die Anzahl an Tagen nach Passwortablauf, bis der Account deaktiviert wird wiki.ubuntuusers.de . -f 0 deaktiviert sofort nach Ablauf; -f -1 entfernt eine evtl. gesetzte Deaktivierung (nie sperren nach Ablauf).
-L	Sperrt das Benutzerkonto (lock). Fügt ein ! dem Passwort-Hash voran (im Shadow-Eintrag) wiki.ubuntuusers.de , was die Anmeldung verhindert. Kann nicht mit -p oder -U kombiniert werden.
-U	Entsperrt ein zuvor gesperrtes Passwort (passwd -l). Entfernt das ! im Shadow-Passwortfeld, sodass der ursprüngliche Hash wieder wirksam wird wiki.ubuntuusers.de . Nicht zusammen mit -L oder -p verwenden.
-p Hash	Setzt das <i>verschlüsselte</i> Passwort (Hash) des Users auf den angegebenen Wert wiki.ubuntuusers.de . Sollte mit Vorsicht genutzt werden – für Passwortänderung ist passwd meist besser geeignet.
--help	Hilfe anzeigen.

Wichtige Hinweise:

- *Warnung:* Die Kombination von -G ohne -a überschreibt die Gruppenliste vollständig. Das heißt, der Benutzer wird aus allen Gruppen entfernt, die nicht explizit bei -G genannt sind[wiki.ubuntuusers.de](#). Um einen User nur hinzuzufügen, niemals ohne -a verwenden!
- Man kann keinen angemeldeten Benutzer umbenennen oder dessen UID ändern, solange er eingeloggt ist oder laufende Prozesse hat. Zudem keine Änderungen an dem eigenen Account durchführen, während man noch mit diesem angemeldet ist (kann zu Inkonsistenzen führen).
- Beim Ändern des Benutzernamens oder UIDs sollten evtl. Crontab-Dateien und geplante at-Jobs manuell umbenannt/angepasst werden, und in NIS/LDAP-Umgebungen müssen Änderungen dort vorgenommen werden[wiki.ubuntuusers.de](#).

userdel (Benutzer löschen)

Beschreibung: Löscht ein Benutzerkonto aus dem System (entfernt die Einträge in /etc/passwd, /etc/shadow und ggf. /etc/group falls der Benutzer eine eigene Gruppe hatte). Optional können auch das Home-Verzeichnis und die Mail-Spool des Benutzers gelöscht werden.

Syntax:

userdel [OPTION] BENUTZER

Optionen:

Option Beschreibung

-r Entfernt *rekursiv* das Heimatverzeichnis des Benutzers **und** seine Mail-Spool (üblicherweise /var/spool/mail/<user>), sowie alle Dateien darin, beim Löschen des Accounts. Ohne -r bleiben Home und Mails unangetastetfile-4fdk7rvx39azjretinarqb.

-f Force: Erzwingt das Löschen des Accounts, selbst wenn der Benutzer noch angemeldet ist oder laufende Prozesse hat. (Warnung: kann zu inkonsistenten Systemzuständen führen, Prozess bleibt ohne Benutzer etc. Diese Option sollte vermieden werden.)

Ohne Option wird der Account aus den Systemdateien gelöscht, aber das (*keine*) Home-Verzeichnis, die Dateien und Mails bleiben bestehen. Admin muss sie ggf. manuell entfernen oder anderweitig zuordnen.

Hinweis: userdel bricht ab, wenn der Benutzer gerade angemeldet ist (außer mit -f). Es ist ratsam, vor dem Löschen sicherzustellen, dass keine Prozesse mehr unter der betreffenden UID laufen (z.B. mit pkill -u username beenden). Dateien außerhalb des Home, die dem Benutzer gehören, verbleiben auf dem Dateisystem mit der nun unbenutzten UID – diese müssten manuell gesucht und entfernt oder neu zugeordnet werden (z.B. mit find / -uid <oldUID>).

id

Beschreibung: Zeigt die Benutzer- und Gruppen-Identitäten eines Benutzerkontos an. Standardmäßig gibt id die UID, primäre GID und zugehörigen Gruppennamen des aktuellen Benutzers zurückfile-4fdk7rvx39azjretinarqb. Mit Angabe eines Benutzernamens kann man die IDs für einen anderen Account abrufen.

Syntax:

id [OPTION]... [BENUTZER]

Optionen:

Option	Beschreibung
(ohne)	Ohne Options werden UID, Benutzername, GID, Gruppenname und alle zusätzlichen Gruppen des Benutzers aufgelistet (z.B. uid=1000(max) gid=1000(max) Gruppen=1000(max),27(sudo),...).
-u	Gibt nur die effektive User-ID (UID, numerisch) aus. Beispiel: id -u alice könnte 1001 ausgeben.
-g	Gibt nur die effektive Gruppen-ID (GID, numerisch) aus.
-G	Gibt alle Gruppen-IDs (numerisch) aus, in denen der Benutzer Mitglied ist (Primär- und Sekundärgruppen).
-n	Gibt Namen anstelle von numerischen IDs. Kann in Kombination mit -u, -g, -G verwendet werden. Z.B. id -Gn alice listet alle Gruppennamen von alice.
-r	Zeigt die reale UID/GID an (falls z.B. mit sudo aufgerufen, reale vs. effektive ID unterscheiden).
-Z	Zeigt SELinux-Sicherheitskontext, falls SELinux aktiv ist (z.B. unconfined_u:unconfined_r:unconfined_t:s0 für einen unconfined Benutzer).
--help, --version	Hilfe bzw. Version anzeigen.

groupadd (Gruppe anlegen)

Beschreibung: Legt eine neue Gruppe im System an. Fügt einen Eintrag in /etc/group (und ggf. /etc/gshadow) hinzu.

Syntax:

groupadd [Optionen] GRUPPENNAME

Optionen:

Option	Beschreibung
-g GID	Verwendet die angegebene numerische GID für die neue Gruppe. Sollte noch nicht vergeben sein (oder mit -o überschrieben werden).
-o	Erlaubt die Verwendung einer bereits existierenden GID (zusammen mit -g zu nutzen, um GID-Doppel zu erzwingen).
-f	"Force": falls die Gruppe schon existiert, keinen Fehler melden; falls die angegebene GID bereits vergeben ist, wählt eine andere (ignoriert -o).

Option	Beschreibung
-r	Erstellt eine Systemgruppe (GID wird aus dem System-Bereich gewählt, z.B. <1000).
-K Schlüssel= Wert	Überschreibt einen Eintrag aus /etc/login.defs temporär (z.B. -K GID_MIN=500 um untere GID-Bereiche anzupassen).
-p Passwort	Setzt das (verschlüsselte) Gruppenpasswort direkt. (Gruppenpasswörter werden selten genutzt – ermöglicht mit newgrp temporär in eine Gruppe zu wechseln.)

groupmod (Gruppe ändern)

Beschreibung: Ändert Eigenschaften einer bestehenden Gruppe – hauptsächlich Name oder GID.

Syntax:

groupmod [Optionen] GRUPPE

Optionen:

Option	Beschreibung
-n NeuerName	Ändert den Gruppennamen in <i>NeuerName</i> . Der alte Name in allen relevanten Systemdateien wird ersetzt.
-g NeueGID	Ändert die GID der Gruppe. Muss eindeutig sein (oder mit -o doppelt erlaubt). Dateien mit alter GID außerhalb, die nicht automatisch erfasst werden, müssen eventuell manuell angepasst werden.
-o	Erlaubt das Setzen einer bereits vergebenen GID (nur mit -g relevant).
-p Passwort	Ändert das (verschlüsselte) Gruppenpasswort. (Entspricht Editieren von /etc/gshadow).

groupdel (Gruppe löschen)

Beschreibung: Entfernt eine Gruppe aus der Systemgruppe-Datei. Die Gruppe wird aus /etc/group (und /etc/gshadow) getilgt.

Syntax:

groupdel GRUPPENNAME

Hinweis: Dieser Befehl hat keine besonderen Optionen – er löscht die Gruppe, sofern sie nicht als Primärgruppe eines Benutzers in /etc/passwd noch verwendet wird. Man kann keine Gruppe löschen, die noch als Hauptgruppe eines existierenden Benutzerkontos eingetragen ist; in dem Fall muss man ggf. erst den Benutzer anpassen

oder löschen. Dateien auf dem System, die dieser Gruppe gehören, behalten die nun ungültige GID bei, bis man sie manuell ändert.

su (Switch User)

Beschreibung: Wechselt den Benutzerkontext im laufenden Terminal. Mit su („substitute user“) kann man einen anderen Benutzeraccount annehmen – standardmäßig root, wenn kein Benutzer angegeben wird. Dies erfordert die Eingabe des Zielbenutzer-Passworts (außer beim Wechsel zu root, dort abhängig von den Systemeinstellungen – oft ist root-Login per Passwort deaktiviert und man nutzt sudo).

Syntax:

`su [OPTION]... [BENUTZER [-]]`

Wird BENUTZER weggelassen, versucht su auf den Superuser (root) zu wechseln.

Optionen:

Option Beschreibung

-l Startet eine *Login-Shell* der Zielusers, d.h. die Umgebung wird vollständig umgeschaltet als würde sich der Benutzer frisch anmelden. Das Home-Verzeichnis, Pfad (\$PATH) etc. werden übernommen. Beispiel: `su - alice` wechselt zu Benutzer *alice* mit deren Umgebung (äquivalent zu vollem Logout/Login als *alice*).

-c Befehl Führt einen angegebenen *Befehl* als Zielbenutzer aus und kehrt dann zurück. Beispielsweise `su -c "systemctl restart apache2"` - root führt als root einen Neustart des Apache durch und beendet sich sofort wieder.

-m, -p Behalten die aktuelle Umgebung (*nicht* auf Ziel-User umschalten). Der aktuelle \$HOME, \$PATH, usw. bleiben erhalten, nur die Rechte wechseln. Standardverhalten ohne -l. (In vielen su Implementierungen ist -m identisch mit -p: *preserve environment*).

-s Shell Startet die angegebene *Shell* statt der Standard-Login-Shell des Zielbenutzers. Der Pfad zur Shell muss im /etc/shells stehen, sofern nicht root.

-g Gruppe Wechselt in Kombination mit Benutzer auch die primäre Gruppe auf *Gruppe*. (Nur verfügbar in neueren util-linux su).

-<Ziffer> (BSD su) Angabe einer Session-Klasse – unter Linux nicht genutzt.

--help, --version Anzeigen von Hilfe bzw. Version.

Hinweise:

- su ohne - (also z.B. su root) wechselt zwar die Benutzer-ID, aber behält das aktuelle Verzeichnis und weite Teile der Environment von Ursprungskonto bei (\$LOGNAME, \$HOME bleiben z.B. vom ursprünglichen Nutzer). Das kann zu irritierenden Effekten führen (z.B. Schreibrechte im falschen Home). Daher nutzt man meist su - für einen *sauberen* Wechsel.
- Viele Distributionen konfigurieren aus Sicherheitsgründen su so, dass ein direkter Wechsel zu *root* per Passwort entweder deaktiviert ist oder kein *root*-Passwort gesetzt ist. Stattdessen wird sudo bevorzugt, um Root-Rechte zu erlangen.

sudo (Superuser Do)

Beschreibung: Führt ein Kommando mit *erhöhten Rechten* aus, typischerweise als *root*, ohne die Benutzeridentität dauerhaft zu wechseln. Benutzer können mit sudo Befehle mit Administratorrechten ausführen, sofern sie in /etc/sudoers entsprechend autorisiert sind. Im Gegensatz zu su wird hier das eigene Passwort abgefragt (nicht das des Zielusers), und nur spezifische Befehle/Optionen sind erlaubt gemäß Konfiguration.

Syntax:

sudo [OPTION]... [COMMAND]

Wird ein Befehl angegeben, führt sudo diesen als root (Standard) aus. Ohne Befehl zeigt sudo ggf. die erlaubten Befehle an oder wechselt in eine sudo-Shell (bei -s).

Optionen (Auswahl):

Option	Beschreibung
---------------	---------------------

- | | | | | | |
|------------------------------|--|--|--|---|---|
| -l
Benutzer | Listet die erlaubten (und verbotenen) Befehle für den aufrufenden Benutzer auf (nach sudoers-Regeln). Hilfreich, um zu sehen, was man mit sudo ausführen darf.

-u
Benutzer | Führt das angegebene Kommando als Benutzer aus (anstatt als root). Beispiel: sudo -u alice whoami zeigt "alice", wenn man als anderer User ausführt. Standard ist root.

-s | Startet eine Shell als Zielbenutzer (Standard root). Ähnlich wie su, aber mittels sudo-Rechte. Es wird die im \$SHELL oder in passwd eingetragene Shell geöffnet.

-i | Simuliert eine vollständige Login-Shell wie su - (initialisiert Umgebung des Zielusers). Entspricht sudo -s plus Login-Umgebung.

-b | Führt den Befehl im Hintergrund aus (<i>background</i>). Sudo verlangt dennoch ggf. ein Passwort, aber detach danach. |
|------------------------------|--|--|--|---|---|

Option Beschreibung

	Beibehaltung der aktuellen Environment-Variablen (normalerweise bereinigt
-E	sudo einige Variablen aus Sicherheitsgründen). Erlaubt das Mitnehmen der eigenen Umgebung. (Ob das erlaubt ist, kann in sudoers eingeschränkt sein.)
-p	Definiert einen eigenen Passwort-Prompt. z.B. sudo -p "Passwort für %u:" -v
<i>Prompt</i>	(Platzhalter %u = Username).
	"Validate": Aktualisiert/überprüft das eigene sudo-Authentifizierungsticket.
-v	D.h. fordert ggf. Passwort an, speichert das sudo Token (5min Standard) aber führt sonst nichts aus. (Verlängert die Zeit, in der man kein Passwort erneut eingeben muss.)
-k	Invalidiert das eigene sudo-Ticket sofort (nachfolgende sudo-Aufrufe verlangen wieder Passwort) – (<i>kill credential</i>).
	Setzt \$HOME in der Environment auf das Home des Zielusers (root).
-H	Normalerweise belässt sudo \$HOME vom Aufrufer; mit -H verhält es sich wie su -.
--help, -h	Hilfe anzeigen (kurze Übersicht der Optionen).
-K	Löscht das Zeitstempel-Ticket komplett (ähnlich -k aber gründlicher, forciert Passwort prompt beim nächsten Mal).
--version	Versionsinfo ausgeben (auch Liste der unterstützten Sicherheitsmechanismen etc.).

Hinweise: Bei erfolgreicher Authentifizierung legt sudo einen Zeitstempel (Ticket) an, sodass für nachfolgende Aufrufe (Standard ~5 Minuten) kein erneutes Passwort nötig ist. Dies kann per sudoers Einstellung (timestamp_timeout) angepasst oder mit sudo -k/-K beeinflusst werden.

pkexec (PolicyKit Execute)

Beschreibung: Führt ein Programm als ein anderer Benutzer aus (ähnlich wie sudo, aber Teil von PolKit/PolicyKit). Standardmäßig wird, sofern nicht anders angegeben, das Programm mit root-Rechten gestartetfreedesktop.org. pkexec erfordert, dass der aufrufende Benutzer berechtigt ist, die Aktion auszuführen (PolKit Policies); falls ja, wird eine Authentifizierungsdialog (grafisch oder textbasiert) angezeigt, um ggf. das Passwort einzugeben. Nach erfolgreicher Auth kann das gewünschte Kommando mit erhöhten Rechten ausgeführt werden.

Syntax:

```
pkexec [--user <BENUTZER>] <KOMMANDO> [ARG...]
```

Optionen:

Option	Beschreibung
--user <Benutzer>	Führt das angegebene Programm als <i>Benutzer</i> aus. Standard ist root (Administratorkonto) freedesktop.org .
--disable-internal-agent	Unterdrückt den internen Authentifizierungs-Agent von pkexec. Normalerweise, falls kein grafischer PolKit-Auth-Agent verfügbar ist, startet pkexec einen eigenen textuellen Auth-Dialog im Terminal. Mit dieser Option wird in so einem Fall statt dessen das Programm nicht ausgeführt (Fehler, wenn kein Agent aktiv) freedesktop.org .
--help / --version	Zeigt Hilfe bzw. Version von pkexec.

Hinweise: pkexec erbt ein Minimal-Umfeld (Environment), um das Einschleusen von Code via Umgebungsvariablen zu verhindernfreedesktop.org. Display- und Xauth-Variablen werden aus Sicherheitsgründen *nicht* an das Zielprogramm weitergegeben, außer es wurde in den PolKit-Regeln explizit erlaubt (GUI-Programme via pkexec sind daher tricky ohne entsprechende PolKit-Policy). In modernen Distributionen ist pkexec oft als Ersatz für grafische gksu-Aufrufe gedacht, um z.B. aus der Desktop-Umgebung heraus GUI-Administrationsprogramme mit Root-Rechten zu starten (wobei PolKit dann i.d.R. das Benutzer-Passwort abfragt).

Berechtigungen und Zugriffskontrolle

umask (Set Default Permissions Mask)

Beschreibung: Legt eine Berechtigungsschutzmaske fest, die bei neu erstellten Dateien und Verzeichnissen die Standardzugriffsrechte beeinflusst. umask subtrahiert Bits von den vollen Zugriffsrechten. So bestimmt sie, welche Rechte *NICHT* automatisch gesetzt werden dürfen. Dies wirkt pro Prozess/Session. Als Shell-Builtin zeigt umask ohne Parameter die aktuelle Maske an.

Syntax:

umask [Option] [Maske]

Optionen / Verwendung:

- Ohne Argument zeigt umask die aktuelle Mask in Oktalnotation an (z.B. 0022). Mit Option **-S** in symbolischer Form (z.B. u=rwx,g=rx,o=rx entspricht Maske 022).
- Mit Angabe einer neuen Maske (z.B. umask 027) wird diese gesetzt. Die Maske wird meist oktal angegeben: jedes Bit, das in der Maske 1 ist, nimmt der neu erstellten Datei die entsprechende Berechtigung weg. Beispiel: Maske 022

bedeutet, neuen Dateien werden keine Gruppe- und Andere-Schreibrechte gegeben (d.h. max Resultatrechte für Datei typ. 644, für Verzeichnis 755).

- **Symbolische Maske:** Man kann auch wie bei chmod symbolisch angeben, z.B. umask u=rwx,g=rx,o=rx für dasselbe Ergebnis (Maske 022).

Hinweis: Die Umask wirkt auf Systemaufruf-Ebene: Standard-Dateirechte sind 666 (rw-rw-rw) und für Verzeichnisse 777 (rwxrwxrwx); die Maske wird davon abgezogen. Z.B. Umask 002 ergibt Dateien mit 664 (rw-rw-r-) und Verzeichnisse mit 775. Umask 077 würde sehr restriktiv alles nur für den Benutzer selbst erlauben (Datei 600, Dir 700).

chmod (Change Mode - Dateirechte ändern)

Beschreibung: Ändert die Zugriffsberechtigungen einer Datei oder eines Verzeichnisses. Man kann mit chmod sowohl numerisch (Oktalwerte) als auch symbolisch (u/g/o +-/=r,w,x) die Rechte setzen.

Syntax:

chmod [Optionen] <Modus> <Datei/Verzeichnis>...

Optionen:

Option	Beschreibung
-R, --recursive	Rekursive Rechteänderung: traversiert Unterverzeichnisse und wendet chmod auf alle darin befindlichen Dateien/Ordner an. Wichtig bei Verzeichnisbäumen.
-v, --verbose	Gibt für jede Datei eine Meldung aus, deren Rechte erfolgreich geändert wurden (und ggf. was geändert wurde).
-c, --changes	Wie verbose, aber nur Ausgaben für tatsächlich geänderte Rechte (keine Ausgabe, wenn Rechte schon so eingestellt waren).
-f, --silent	Unterdrückt Fehlermeldungen (force).
--reference=<Vorlage>	Setzt die gleichen Berechtigungen wie die Referenz-Datei <Vorlage> hat, auf die angegebenen Dateien. Statt Modus muss man --reference nutzen.

Modus-Angaben:

- **Numerisch (Oktal):** Eine dreistellige (oder vierstellige mit Sonderbit) Oktalzahl. Z.B. chmod 640 datei.txt setzt Rechte auf rw- r-- --- (6=rw,4=r,0=kein Recht). Jede Ziffer entspricht Owner, Group, Others. Beispiel: chmod 755 script.sh => rwx für Besitzer (7), rx für Gruppe (5), rx für andere (5). Eine führende vierte Ziffer kann

spezielle Bits setzen: 4 für Setuid, 2 für Setgid, 1 für Sticky-Bit (z.B. 1755 für Sticky).

- **Symbolisch:** Format: [ugoa]*[+-=][rwxXst]* evtl. durch Kommata getrennt für mehrere Operationen in einem Aufruf. Beispiele:
 - chmod u+x,g-w,o= file – dem Besitzer Ausführungsrecht geben, der Gruppe Schreibrecht entziehen, andere bekommen exakt keine Rechte.
 - chmod a+r file – allen (a für all) Leserechte hinzufügen (ohne andere Bits zu ändern).
 - chmod g=u Datei – setzt die Gruppenrechte gleich den Benutzerrechten der Datei (nützliche Kurzform).
 - X: Spezialflag in symbolischer Notation – steht für "execute if directory or if any execute bit is set on any of owner/group/other". Wird meist bei -R verwendet, um rekursiv nur Verzeichnissen das x zu geben.
 - s für setuid/setgid (in u/g Kontext), t für Sticky-Bit (im o-Kontext).
 - Beispiel: chmod u+s script.sh – setzt das setuid-Bit auf die Datei (falls ausführbar).

chown (Change Owner)

Beschreibung: Ändert den Dateibesitzer und/oder die Gruppe einer Datei oder eines Verzeichnisses

file-4fdk7rvx39azjretinarqbfile-4fdk7rvx39azjretinarqb. chown kann sowohl den User als auch die Gruppe in einem Befehl setzen. Nur root (oder Capability CHOWN) darf Eigentümer von Dateien ändern. Normale Benutzer können in der Regel mittels chgrp die Gruppenzugehörigkeit ändern, wenn sie Mitglied der Zielgruppe sind (und Besitzer der Datei bleiben).

Syntax:

chown [Optionen] <neuerEigentümer>[:<neueGruppe>] DATEI...

Optionen:

Option	Beschreibung
-R	Rekursiv – wendet die Besitzänderung auf alle Dateien und Unterverzeichnisse an (ganzer Baum).
-h	Ändert den Besitzer von Symbolic Links selbst, statt des Ziels. (Auf den meisten Linux ist das Standard mit Kernel 2.6+; teils wird -h ignoriert, da chown auf Symlinks ohnehin Link selbst ändert, außer mit -L Option).

Option	Beschreibung
-f	Keine Fehlermeldungen ausgeben (silent/force).
-v	Verbose – listet jede Datei mit geänderten Owner/Group auf.
-- from=<old_owner:old_group>	Ändert nur, wenn aktuelle Owner/Group genau wie angegeben ist. (Damit kann man selektiv bestimmte Kombis ersetzen, dient als Sicherheitsprüfung).
--dereference / -L	<i>Follow symlinks:</i> Symbolische Links auflösen (statt Link selbst zu ändern wird das Ziel geändert). Standardverhalten für die meisten chown-Versionen ist bereits, dem Link <i>nicht</i> zu folgen (außer man setzt -L).
-P	<i>Don't follow symlinks</i> (explizit Link behandeln, Standard).
-H	Folgen von Symlinks im Kommandozeilen-Argument (Top-Level), aber nicht rekursiv tiefer. (Selten benötigt).

Verwendung:

- Besitzer ändern: chown alice datei.txt – setzt *alice* als neuen Besitzer der Datei (Gruppe bleibt unverändert).
- Gruppe ändern mit chown: chown :developers projektX – behält Besitzer, setzt Gruppe auf *developers*. Notation: Vor dem Doppelpunkt nichts heißt Besitzer unverändert, aber *Gruppe* ändern.
- Besitzer und Gruppe zugleich: chown alice:staff datei.txt – Besitzer = *alice*, Gruppe = *staff*.
- Nur Gruppe ändern könnte auch mit eigenem Befehl chgrp staff datei.txt gemacht werden.

Besonderheit: Man kann numerische IDs angeben. chown 1001:100 datei setzt Besitzer UID=1001, GID=100. Falls man einen Benutzer- oder Gruppennamen mit Doppelpunkt hat, der numerisch beginnt, kann man -- verwenden um Verwechslung zu vermeiden oder explizit username: schreiben.

chgrp (Change Group)

Beschreibung: Ändert die Gruppenzugehörigkeit einer Datei oder eines Verzeichnissesfile-4fdk7rvx39azjretinarqbfile-4fdk7rvx39azjretinarqb. Äquivalent zu chown :<gruppe>. Wird benutzt, um den Gruppeneigentümer anzupassen.

Syntax:

chgrp [Optionen] <neueGruppe> DATEI...

Optionen:

Option	Beschreibung
-R	Rekursive Änderung der Gruppe in Unterverzeichnissen (alle Dateien/Dirs innerhalb werden angepasst).
-f	Keine Fehlermeldungen bei Problemen ausgeben (silent).
-v	Ausführliche Ausgabe pro Datei.
--reference=Vorlage	Übernimmt die Gruppenzugehörigkeit der <i>Vorlage</i> -Datei auf alle angegebenen Dateien. (Ignoriert das Gruppen-Argument und nutzt stattdessen die Gruppe der Referenz.)

Hinweis: Normale Benutzer dürfen chgrp nur verwenden, um eine Datei auf eine Gruppe zu setzen, der sie selbst angehören. Root kann beliebige Zuordnungen vornehmen.

lsattr (List Attributes)

Beschreibung: Listet die *erweiterten Dateiattribute* (Filesystem-Attribute) von ext-Dateisystemen auffile-4fdk7rvx39azjretinarqb. Diese Attribute wie *immutable* oder *append-only* steuern besondere Verhalten auf niedriger Ebene (nur auf bestimmten FS wie ext2/3/4, XFS - mit anderen Tools - etc. vorhanden). lsattr zeigt pro Datei eine Zeichenfolge von Attribut-Flags.

Syntax:

lsattr [Optionen] [Datei/Verzeichnis]...

Optionen:

Option Beschreibung

-a	Zeigt auch versteckte Dateien (Dateien, die mit Punkt beginnen) an. Standardmäßig ignoriert lsattr diese analog zu ls ohne -a.
-d	Listet das Verzeichnis selbst auf, nicht dessen Inhalt. (Bei Angabe eines Verzeichnisses wird normalerweise dessen Inhalt rekursiv aufgelistet; mit -d sieht man nur die Attribute des Verzeichniseintrags selbst)file-4fdk7rvx39azjretinarqb.
-R	Rekursive Ausgabe der Attribute in allen Unterverzeichnissen.
-v	Zeigt die Dateiversionsnummer (Version im ext-Format) mit an, falls vorhanden. (Das Attribut 'version' wird kaum benutzt, -v gibt eine zusätzliche Spalte mit Versionsnummer).

Option Beschreibung

-V Ausführliche Informationen über die Version des Programms und detaillierte Fehler ausgeben (verbose).

Beispiel Ausgabe:

```
$ lsattr wichtige_datei  
----i-----e-- wichtige_datei
```

Hier bedeuten die Buchstaben: **i** = immutable (unveränderlich), **e** = extents (standard auf ext4). Bindestriche zeigen nicht gesetzte Flags. Wichtige Flags: **i** (immutable: Datei kann nicht verändert/gelöscht umbenannt werden bis Attribut entfernt), **a** (append-only: nur anhängen erlaubt), **d** (no dump: Datei wird von Dump-Backup ignoriert), **A** (no atime updates), **S** (synchronous updates) etc.

chattr (Change Attributes)

Beschreibung: Ändert die erweiterten Dateisystem-Attribute (wie immutable, append-only etc.) unter Linux auf ext2/ext3/ext4 Dateisystemenfile-4fdk7rvx39azjretinarqb. Nur Root darf die meisten dieser Attribute setzen.

Syntax:

```
chattr [Optionen] [+--Attribute] Datei...
```

Optionen:

Option	Beschreibung
-R	Rekursive Attributänderung in Unterverzeichnissenfile-4fdk7rvx39azjretinarqb.
-V	Ausführlich – zeigt an, welche Änderungen für welche Datei vorgenommen wurden.
-f	Unterdrückt Fehlermeldungen (force).
-v <Nummer>	Setzt die Versionsnummer der Datei (falls vom Dateisystem unterstützt). Selten benutzt – ext-Dateisysteme haben optional Dateiversionsnummern.

Attribute setzen (Symbolik): Man gibt nach möglichen Optionen eine Zeichenfolge an, der ein +, - oder = vorangestellt ist:

- + bedeutet diese Attribute *hinzufügen* (zusätzlich zu bestehenden).
- - bedeutet diese Attribute *entfernen*.
- = bedeutet *exakt setzen* (alle aktuellen Attribute durch diese ersetzen).

Die wichtigsten Attribut-Buchstaben auf ext4 sind:

- **i – immutable**: Datei kann nicht modifiziert, umbenannt oder gelöscht werden file-5y1edouysrow4zj6phjcqm. Prozesse können sie nicht öffnen im Schreibmodus. Schützt kritische Dateien (z.B. /etc/passwd) vor Änderungen, sogar von root (root muss erst chattr -i machen).
- **a – append only**: Datei kann nur angehängt, aber nicht überschrieben oder gelöscht werden. Typisch für Logfiles, damit niemand alte Einträge löscht.
- **e – extent format**: zeigt an, dass die Datei extents benutzt (automatisch auf ext4, nicht manuell gesetzt).
- **d – no dump**: Datei soll von Backup-Programmen, die dem Dump-Standard folgen, ignoriert werden.
- **A – no atime updates**: beim Lesen der Datei wird der atime-Zeitstempel nicht aktualisiert (Performance-Optimierung).
- **S – synchronous updates**: Änderungen an der Datei werden synchron auf den Datenträger geschrieben (wie bei chattr +S – ähnlich wie Ordner mit Dirsync mount option).
- **t – no tail-merging**: (relevant bei ext-Dateisystem, deaktiviert Tail-Merging bei Reverse-Mapping; selten benutzt).
- **T – top of directory hierarchy**: Verzeichnis wird als Top-Level für Ordnungszwecke in ext behandelt (verhindert bestimmte Ordnungsoptimierungen; selten).
- **u – undeletable**: Dateiinhalt soll beim Löschen inhaltlich gesichert werden, sodass man ihn wiederherstellen könnte. (Wird von ext4 i.d.R. nicht implementiert).

Beispiel:

- chattr +i datei.txt – setzt *immutable*.
- chattr -i datei.txt – entfernt dieses Flag wieder (dann ist Bearbeiten/Löschen wieder möglich).
- chattr =a Datei.log – setzt **nur** append-only und entfernt alle anderen Attribute (falls welche gesetzt waren). Danach kann in Datei.log nur noch angehängt werden.

getfacl (Get File ACL)

Beschreibung: Liest die Access Control Lists (erweiterten Zugriffsrechte) von Dateien oder Verzeichnissen aus und zeigt sie an file-4fdk7rvx39azjretinarqb. Auf Linux-Systemen mit POSIX-ACL-Unterstützung (z.B. ext4 mit acl-Option eingehängt) kann man pro Datei feinere Berechtigungen für einzelne Benutzer oder Gruppen setzen. getfacl zeigt diese

ACL-Einträge zusätzlich zu den Standardrechten.

Syntax:

getfacl [Optionen] Datei...

Optionen:

Option	Beschreibung
-a	Zeigt auch die <i>verschachtelten ACLs</i> (falls vorhanden – nicht üblich bei POSIX ACL).
-e	Zeigt die Ausgabe im <i>effective Modus</i> – Standard. (Jeder Eintrag inkl. "effective permissions", relevant wenn Mask gesetzt ist).
-s	Stripschutz: Zeigt nur die tatsächlichen ACL-Einträge, ohne die Grundeinträge (user/group/other).
-d	Zeigt die <i>Default-ACL</i> eines Verzeichnisses (die Vorlage, die an neu darin erzeugte Dateien vererbt wird). In der Ausgabe mit default: gekennzeichnet.
--omit-header	Unterdrückt den Kommentar-Kopf in der Ausgabe, der Dateinamen und Hinweise beinhaltet.
-R	Rekursiv über Verzeichnisse, getfacl für jede Datei auflisten.

Ausgabe-Beispiel:

```
$ getfacl projekt/
# file: projekt/
# owner: alice
# group: developers
user::rwx
user:bob:rwx-      # bob hat rw Rechte zusätzlich
group::r-x
mask::rw-          # effektive Maske für ACL = rw
other::---
default:user::rwx  # Default ACL für neu angelegte Objekte
...
```

Hier sieht man die Standard-Owner/Group-Einträge und eine explizite ACL, die User bob Schreibrechte gibt, sowie eine *mask*, die die maximal wirksamen Rechte auf rw beschränkt.

setfacl (Set File ACL)

Beschreibung: Setzt oder ändert Access Control Lists (ACLs) von Dateien und Verzeichnissen file-4fdk7rvx39azjretinarqb. Damit kann man zusätzliche Berechtigungen für bestimmte Nutzer oder Gruppen vergeben (oder entziehen), die über die klassischen Besitzer-Gruppe-Andere Rechte hinausgehen.

Syntax:

```
setfacl [Optionen] { -m|-x <ACL-Eintrag> ... | -b | -k } Datei...
```

Optionen:

Option Beschreibung

- m** *Modify*: Fügt einen ACL-Eintrag hinzu oder aktualisiert ihn. Beispiel: setfacl -m u:bob:rw file gibt Benutzer *bob* rw-Rechte auf *file*. Mehrere -m Einträge oder Einträge mit Komma getrennt sind möglich.
- x** *ACL* *Remove*: Entfernt einen spezifischen ACL-Eintrag. Z.B. setfacl -x g:marketing file entfernt den ACL-Eintrag für Gruppe "marketing".

Remove all – Löscht alle ACL-Einträge (außer den Basis-Einträgen für
- b** Owner/Group/Other). Damit werden alle erweiterten Berechtigungen zurückgesetzt. (Datei hat dann nur noch die klassischen Unix-Rechte.)
- k** *Remove Default* – Entfernt die Default-ACL eines Verzeichnisses (vererbte ACL-Vorlage). Hat auf Dateien keine Wirkung.
- R** Rekursiv – wendet die ACL-Änderung auf alle Dateien im Verzeichnis (und Unterverzeichnisse) an. Nützlich um z.B. ganze Bäume eine neue ACL zu geben.
- d** Setzt Einträge als *Default-ACL* für Verzeichnisse. Benutzt in Kombination mit -m oder -x. Beispiel: setfacl -m d:g:team:rwx projekt setzt eine Default-ACL, so dass neue Dateien unter *projekt*/ der Gruppe *team* rw Rechte erben, und Verzeichnisse rwx.
- **mask** Berechnet und setzt die ACL-Maske basierend auf den aktuellen Gruppen-Einträgen neu. (Die Maske begrenzt die maximalen Rechte der ACL-Einträge außer Owner/other). Standardmäßig wird die Maske auto-anangepasst.

Option Beschreibung

- Don't recalc mask:* Verhindert, dass setfacl die ACL-Maske automatisch anpasst. Normalerweise soll man diese Option nicht verwenden, außer man setzt die Maske manuell mit -m m::....

Beispiel: setfacl -m u:bob:rw,g:marketing:r-- report.txt – gibt Benutzer *bob* Lese- und Schreibrecht und der Gruppe *marketing* nur Leserecht auf *report.txt*.

Hinweis: Nicht alle Dateisysteme unterstützen POSIX-ACL. Bei ext4, XFS etc. muss i.d.R. das Volume mit Option acl eingehängt sein (moderne Systeme haben das meist per Default). Mit getfacl kann man prüfen, ob vorhandene ACLs existieren.

Datei- und Verzeichnisverwaltung

stat (Dateistatus anzeigen)

Beschreibung: Zeigt detaillierte Meta-Informationen zu Dateien oder Dateisystemen an. stat gibt z.B. Größe, Blocks, Zeitstempel (Zugriff, Änderung, Statusänderung), Berechtigungen in Zahlenform, Device-ID, Inode-Nummer usw. in strukturierter Form aus.

Syntax:

stat [Optionen] <Datei>...

Optionen:

Option	Beschreibung
-f, --file-system	Zeigt anstelle der Datei-Details die Informationen zum Dateisystem, auf dem die Datei liegt. Z.B. Statistiken wie Blockgröße, Gesamtblöcke, freie Blöcke, FS-Typ etc.
-c FORMAT, --format=FORMAT	Gibt die Ausgabe gemäß <i>FORMAT</i> formatiert aus. Man kann Platzhalter verwenden, um bestimmte Werte auszugeben. Z.B. stat -c "%A %h %U %G %s %n" datei gibt in einer Zeile: modestr, Linkanzahl, Owner, Group, Size, Name. (Siehe man stat für alle Platzhalter wie %A = Rechte in rwx, %a = oktal, %U = Ownername, %s = Größe etc.)
--printf=FORMAT	Ähnlich wie --format, jedoch ohne automatisch Zeilenumbruch am Ende. Kann benutzt werden, um mehrere stat-Aufrufe zu skripten.
-t, --terse	"Knapp": Gibt die Ausgabe in einer kompakten einzeiligen Form, Felder durch Leerzeichen getrennt, aus (für einfache maschinelle Weiterverarbeitung).

Option	Beschreibung
--help, --version	Anzeigen von Hilfe bzw. Version.

Beispielausgabe:

```
$ stat /etc/hosts
  File: /etc/hosts
  Size: 258    Blocks: 8      IO Block: 4096  regular file
Device: 802h/2050d Inode: 13422506  Links: 1
Access: (0644/-rw-r--r--) Uid: ( 0/  root)  Gid: ( 0/  root)
Access: 2025-04-01 10:22:13.000000000 +0200
Modify: 2025-04-01 10:22:13.000000000 +0200
Change: 2025-04-01 10:22:15.000000000 +0200
 Birth: -
```

Dies zeigt z.B., dass /etc/hosts 258 Bytes groß ist, auf einem bestimmten Device, inode etc., Rechte 0644, Eigentümer root, Zeitstempel für Zugriffs-/Änderungs-/Änderungszeit (Birth ist Erstellung, nicht unterstützt).

file (Dateityp bestimmen)

Beschreibung: Untersucht eine Datei und versucht den Typ bzw. Format der Datei zu bestimmen. file nutzt einen umfangreichen Magie-Pattern-Katalog (/usr/share/file/magic), um anhand der Bytes einer Datei (Magic Numbers, Struktur etc.) menschlich lesbare Dateitypinformationen auszugeben. Praktisch, um z.B. Binärformate, Skripttypen oder Kodierungen zu erkennen.

Syntax:

file [Optionen] <Datei>...

Optionen:

Option Beschreibung

- Zeigt den MIME-Typ der Datei anstatt einer Beschreibung. Z.B. text/plain;
- i** charset=utf-8 statt "UTF-8 Unicode text". Hilfreich für Scripting oder Web-Kontext.
- Brief* – gibt nur die Beschreibung aus, ohne den Dateinamen voranzustellen.
- b** (By default schreibt file: "<Dateiname>: <Beschreibung>"). Mit -b erhält man nur "<Beschreibung>".

Option Beschreibung

- Folgt symbolischen Links und untersucht das Ziel statt den Link selbst.
- L Standardmäßig wird der Link selbst untersucht und meist als "symbolic link to <target>" ausgegeben.
- (alias --no-dereference) Bewirkt das Gegenteil: behandelt Symlinks wie
- h normale Dateien (gleichbedeutend mit Standardverhalten, -h ist nützlich falls -L als default kompiliert wäre).
- Untersucht auch Gerätedateien (block/char special) und Auslagerungsdateien.
- s Standardmäßig werden einige spezielle Dateitypen nicht inhaltlich analysiert
(um z.B. nicht eine Device-Datei zu lesen). -s kann riskant sein (kann z.B. /dev/zero "lesen" -> blockiert).
- z Durchsucht komprimierte Dateien (gzip-komprimierte) – versucht innerhalb eines .gz die Dateisignatur zu prüfen.
- E Gibt (Exit)Status 1 zurück falls *irgendeine* der Dateien kein bekannter Typ bestimmt werden konnte. (Nützlich in Skripten).
- help, Hilfe bzw. Versionsnummer und unterstützte Magic-Datei-Version ausgeben.
-v

Beispiele:

- file /bin/ls -> **ELF 64-bit LSB executable**, x86-64, dynamically linked ...
- file archive.tar.gz -> **gzip compressed data**, was probably a tar archive (reads inside and erkennt es als tar in gz).
- file picture.jpg -> **JPEG image data**, JFIF standard 1.01, ...

ln (Link erstellen)

Beschreibung: Erstellt Verknüpfungen zu Dateien – entweder **Hard Links** oder **Symbolische Links**. Ein Hardlink ist ein weiterer Directory-Eintrag für dieselbe Inode (nur innerhalb desselben Dateisystems möglich). Ein Symbolischer Link (Symlink) ist eine kleine Spezialdatei, die auf einen Pfad verweist.

Syntax:

ln [Optionen] <Ziel> [Linkname]

Standard ohne Optionen erstellt *Hardlink* mit Namen *Linkname*, der auf Datei *Ziel* zeigt. Wenn *Linkname* nicht angegeben, wird im aktuellen Verzeichnis ein Link gleichen Namens wie *Ziel* erstellt. Für Symlinks Option -s nutzen.

Optionen:

Option	Beschreibung
-s, --symbolic	Erstellt einen <i>symbolischen Link</i> statt eines Hardlinks. Der Link verweist auf den angegebenen <Ziel>-Pfad. Wenn <Ziel> kein absoluter Pfad ist, wird er relativ vom Link aus interpretiert. Beispiel: ln -s /etc/hosts hostslink erzeugt Symlink <i>hostslink</i> -> /etc/hosts.
-f, --force	Existierende Zielfile (Linkname) ggf. entfernen, falls schon vorhanden. Ohne -f würde ln bei existierendem Linknamen einen Fehler ausgeben. (Bei -s mit -f: überschreibt existierenden Link.)
-n, --no-dereference	Beim Überschreiben eines existierenden symbolischen Links (mit -f) behandle den Link selbst statt dessen Ziel. (Verhindert, dass versehentlich die Datei, auf die ein bestehender Link zeigt, gelöscht wird).
-v, --verbose	Gibt eine Meldung pro verarbeitetem Link aus.
-T, --no-target-directory	Behandelt das letzte Argument immer als <i>genauen Linknamen</i> , selbst wenn es ein existierendes Verzeichnis ist. (Verhindert, dass ln im Zielverzeichnis einen Link mit Standardname erstellt.)
-d, -F, --directory	Erlaubt es root, auch <i>Verzeichnis-Hardlinks</i> zu erstellen. (Normalerweise verboten – Option nur auf speziellen Systemen mit -f kombinierbar, in der Praxis kaum verwendet).
-b, --backup	Erzeugt ein Backup der Zielfile falls -f zum Überschreiben genutzt wird. (Nützlich, um das ersetzte File nicht zu verlieren.) Siehe auch -- backup=t etc.
-S <Suffix>	Gibt ein Suffix für Backup-Dateien an (Standard ~). Bsp: -b -S .bak.

Beispiele:

- Hardlink: ln /path/to/original.txt copy.txt – copy.txt ist nun ein weiterer Verweis auf *original.txt* (gleiche Inode). Änderungen an einer erscheinen in der anderen, da es dieselbe Datei ist. Löscht man eine, bleibt die andere erhalten (Datei wird erst gelöscht, wenn letzter Link entfernt).
- Symlink: ln -s /etc/apache2 siteconfig – erstellt *siteconfig* -> /etc/apache2. Aufruf von ls -l siteconfig zeigt dann siteconfig -> /etc/apache2.

cd (Change Directory) – siehe oben bei Grundlegende Befehle

(Im Kontext Datei/Verzeichnis-Verwaltung sei auf den bereits beschriebenen Befehl cd verwiesen, um Verzeichnisse zu wechseln.)

tree (Verzeichnisbaum anzeigen)

Beschreibung: Zeigt die Verzeichnisstruktur grafisch in Form eines Baumes an. tree listet rekursiv alle Unterverzeichnisse und Dateien in hierarchischer eingerückter Darstellung. Es erleichtert das Visualisieren der Verzeichnis-Hierarchie.

Syntax:

`tree [Optionen] [Verzeichnis]`

Optionen:

Option	Beschreibung
-a	Zeigt <i>alle</i> Dateien an, inkl. versteckter (auch solche, die mit . beginnen). Standardmäßig ignoriert tree versteckte Dateien, außer . und .. immer.
-d	Listet <i>nur Verzeichnisse</i> auf, keine Dateien. Nützlich, um reinen Verzeichnisbaum zu sehen.
-L Level	Beschränkt die Tiefe der Ausgabe auf <i>Level</i> Ebenen. Z.B. tree -L 2 zeigt nur zwei Ebenen tief (akt. Verzeichnis + Unterverzeichnisse, aber keine tieferen).
-f	Zeigt den vollen Pfad vor jedem Eintrag (statt nur Namen relativ).
-I	Ignoriert Einträge, die auf das Muster passen (Groß/Kleinschreibung
<i>PATTERN</i>	berücksichtigt, mehrere Patterns mit `
-P	Zeigt nur Einträge an, die dem Pattern entsprechen (Whitelist). Umgekehrt zu
<i>PATTERN</i>	-I.
-F	Markiert Verzeichnisse mit abschließendem /, ausführbare Dateien mit *, Symlinks mit @, Pipes mit `
-p	Zeigt UNIX-Dateirechte (Permission bits) vor jedem Eintrag.
-u / -g	Zeigt Benutzer bzw. Gruppenbesitzer an. (Wie ls -l Infos).
-h	<i>Human-readable size:</i> Zeigt Dateigrößen in menschenlesbarem Format (K, M, G) anstatt in Bytes (wirkt mit -s).
-s	Zeigt Dateigrößen in Bytes neben jedem Dateinamen an.
--du	Summiert Verzeichnisgrößen wie du (statt wie normal lediglich die Einzeldateigröße zu zeigen).
-t	Sortiert die Ausgabe nach Zeit (Neueste zuerst) statt alphabetisch.
-o Datei	Gibt die Ausgabe in die angegebene Datei (überschreibt) anstatt auf stdout.

Option	Beschreibung
-n	Keine farbliche Hervorhebung (Standard ist bunt, -n erzwingt Monochrom).
--help	Hilfe anzeigen (Übersicht der Optionen).

Beispiel:

```
$ tree -L 2 /etc
```

/etc

```
    └── apache2
        ├── apache2.conf
        ├── envvars
        └── mods-available

    └── cron.daily
        ├── apt-compat
        └── man-db

    └── default
        ├── cron
        └── rcS
```

6 directories, 5 files

Hier sieht man /etc mit zwei Ebenen Tiefgang.

mkdir (Make Directory)

Beschreibung: Erstellt ein neues Verzeichnis (oder mehrere) im Dateisystem. Kann rekursiv auch mehrere Ebenen auf einmal anlegen.

Syntax:

```
mkdir [Optionen] <Verzeichnisname>...
```

Optionen:

Option	Beschreibung
-p, --parents	Erzeugt auch alle notwendigen übergeordneten Verzeichnisse, falls sie nicht existieren. Somit werden keine Fehler ausgegeben, wenn

Option	Beschreibung
	Verzeichnisse in der Pfadangabe fehlen. Beispiel: mkdir -p /tmp/foo/bar legt sowohl /tmp/foo (falls nicht da) als auch /tmp/foo/bar an.
-m, --mode=MODE	Setzt die Zugriffsrechte des neu erstellten Verzeichnisses auf <i>MODE</i> (numerisch oder symbolisch) anstatt der Standardrechte (die Standardrechte sind i.d.R. 0777 abzüglich Umask). Z.B. mkdir -m 750 projDir erstellt projDir mit rwx für Besitzer, rx für Gruppe, nichts für andere.
-v, --verbose	Ausgabe einer Meldung für jedes angelegte Verzeichnis (zeigt Pfad an).
--help, --version	Anzeige von Hilfe bzw. Version.

cp (Copy File)

Beschreibung: Kopiert Dateien oder Verzeichnisse. Bei Angabe mehrerer Quelldateien muss das letzte Argument ein Verzeichnis sein (Zielverzeichnis). cp kann auch Verzeichnisbäume rekursiv kopieren.

Syntax:

cp [Optionen] <Quelle>... <Ziel>

Wenn <Ziel> ein Verzeichnis ist, werden alle <Quelle>-Dateien dorthin kopiert (mit gleichem Namen). Ist nur eine Quelle und <Ziel> kein Verzeichnis, wird die Datei dorthin kopiert/umbenannt.

Optionen:

Option	Beschreibung
-r, -R, --recursive	Kopiert Verzeichnisse rekursiv (Inhalt und Unterverzeichnisse). Ohne diese Option kopiert cp standardmäßig keine Verzeichnisse. (GNU cp hat auch -r/-R – beide meist gleichwertig. POSIX fordert -R.)
-a, --archive	Archivmodus: Kopiert rekursiv und bewahrt dabei möglichst alle Attribute. Entspricht -dR --preserve=all (also -R, und -p für timestamps, owner, group, perms, ACLs, xattrs, Links etc.). Dies ist für Vollbackups ideal ionos.at .
-p, --preserve	Bewahrt Eigentümer, Gruppe, Berechtigungen und Zeitstempel der Datei beim Kopieren ionos.at . (Ohne -p übernimmt cp Standard: setzt aktuellen Zeitstempel "jetzt" und erstellt als aktueller

Option	Beschreibung
	Benutzer). Kann Argumente haben wie --preserve=mode,ownership,timestamps,links,xattr,all.
-f, --force	Überschreibt vorhandene Ziel-Dateien ohne Nachfrage, und versucht auch nicht-schreibbare Ziele zu entfernen, um Kopie dennoch zu platzieren. (Im Konflikt mit -n gilt letzter Schalter.)
-i, --interactive	Fragt nach, bevor eine existierende Zielfile überschrieben wird (Confirm Y/N). Gibt dem Benutzer die Möglichkeit, Überschreiben abzulehnen.
-n, --no-clobber	Überschreibt <i>keine</i> existierenden Dateien. D.h. falls Ziel existiert, wird die Kopie für diese Datei ausgelassen. (Verhindert unabsichtliches Überschreiben)
-u, --update	Kopiert nur, wenn <i>Quelle</i> neuer als <i>Ziel</i> ist, oder Ziel noch nicht existiert. Praktisch für inkrementelle Updates. Existiert Ziel mit gleich/neueren Datum, wird es nicht überschrieben.
-d, --no-dereference	Kopiert <i>Symbolic Links</i> als Links (anstatt deren referenzierten Inhalt). Standard: cp folgt symbolischen Links und kopiert die Datei dahinter. Mit -d wird der Link selbst neu erstellt am Ziel (Hardlinks bleiben hardlinks sofern möglich; -d ist in GNU cp gleichbedeutend mit --preserve=links).
-L, --dereference	<i>Folgt allen Symlinks:</i> kopiert stets den referenzierten Inhalt (Standardverhalten bei regulärem cp für Dateien). Bei Angabe von Verzeichnis mit -R: -L würde bewirken, dass cp <i>alle</i> gefundene Symlinks auflöst und deren Ziel kopiert (kann zu Loops führen, daher mit Vorsicht).
-P, --no-dereference --preserve=links	Kopiert Links als Links (das ist Standard mit -a enthalten). (Im GNU cp -P bedeutet No-dereference; beachten: -d und -P ähnlich).
--parents	Erhält die Quellpfadstruktur im Ziel. Bsp: cp --parents dir1/dir2/file.txt /backup legt /backup/dir1/dir2/file.txt an man7.org . Nur sinnvoll, wenn Ziel ein Verzeichnis ist.
-l, --link	Erstellt <i>Hardlinks</i> statt Kopien der Datei, sofern im selben Dateisystem. (Keine Datenkopie, nur neuer Name, ähnlich ln).

Option	Beschreibung
-s, --symbolic-link	Erstellt <i>Symlinks</i> statt die Daten zu kopieren. (Zieldatei wird symbolischer Verweis auf Quelldatei).
-H, -L, -P (bei -R)	Steuerung des Symlink-Verhaltens während rekursivem Kopieren: -P (Default) ignoriert Linkinhalte (kopiert Symlinks als Symlinks), -L folgt Symlinks zu Dateien, -H folgt nur Symlinks angegeben auf Befehlzeile. (-a impliziert -P).
-- backup[=Control]	Erzeugt Backup von Zieldateien, die überschrieben würden. (Etwa file.txt~ anlegen). Kann =simple, numbered etc. steuern. Siehe auch VERSION_CONTROL Env.
--help, --version	Hilfe/Version anzeigen.

mv (Move/Rename)

Beschreibung: Verschiebt Dateien oder Verzeichnisse an einen neuen Pfad, bzw. benennt sie um. mv kann auch zum einfachen Umbenennen einer Datei verwendet werden (Quell- und Zielpfad auf selbem FS = schneller Rename, kein Kopieren). Falls Quelle und Ziel auf unterschiedlichen Dateisystemen liegen, arbeitet mv durch Kopieren und anschließendem Löschen der Quelle.

Syntax:

mv [Optionen] <Quelle>... <Ziel>

Optionen:

Option	Beschreibung
-i, --interactive	Fragt nach Bestätigung, falls eine Zieldatei durch den Move überschrieben würde (falls im Zielpfad eine Datei gleichen Namens existiert).
-f, --force	Erzwingt das Überschreiben ohne Nachfrage. Ziel-Dateien werden ohne Warnung ersetzt. (Standardverhalten ohne -i).
-n, --no-clobber	Überschreibt keine bestehenden Dateien. Bei Namenskollision bricht mv für diese Datei ab (gibt keinen Fehler, exit code trotzdem 0). In GNU mv hat -n Vorrang vor -i falls beide angegeben.
-u, --update	Bewegt/überschreibt nur, wenn die Quelle neuer ist als die vorhandene Zieldatei <i>oder</i> die Zieldatei nicht existiert. (So

Option	Beschreibung
	werden neuere Versionen moved, ältere Quellen, die schon am Ziel sind, bleiben unberührt.)
-v, --verbose	Gibt aus, welche Datei wohin verschoben/umbenannt wurde (für jede Operation eine Zeile).
--	Wie bei cp: legt Backup der Ziel-Datei an, falls existiert, bevor sie überschrieben wird. CONTROL kann none, numbered, existing, simple sein (siehe Doku oder ENV VERSION_CONTROL).
backup[=CONTROL]	
--strip-trailing-slashes	Entfernt evtl. nachgestellte Schrägstriche bei den Pfadangaben der Quellen (um Verwirrung mit Verzeichnissen zu vermeiden). Standard: neuere mv machen das ohnehin.
--help, --version	Hilfe/Version.

Hinweis: mv kann mehrere Dateien in ein Zielverzeichnis verschieben oder eine einzelne Datei/Ordner umbenennen. Bei Verzeichnissen rekursiv, aber mv muss dafür nicht extra Option haben (im Gegensatz zu cp), da der Move innerhalb desselben FS nur Metadaten ändert. Bei verschiedenen FS kopiert mv im Hintergrund (rename() Syscall schlägt dann fehl, worauf cp+unlink passiert).

rmdir (Remove Directory)

Beschreibung: Löscht leere Verzeichnisse. Im Gegensatz zu rm entfernt rmdir nur Verzeichnisse und auch nur, wenn sie keine Dateien/Unterverzeichnisse mehr enthalten. Es kann optional ganze verschachtelte Verzeichnisbäume entfernen, sofern diese leer sind (Eltern werden dann rekursiv entfernt).

Syntax:

rmdir [Optionen] <Verzeichnis>...

Optionen:

Option	Beschreibung
-p, --parents	Entfernt das angegebene Verzeichnis und ggf. dessen übergeordnete Verzeichnisse, falls diese nach jedem Entfernen leer werden. Beispiel: rmdir -p projekt/build/output versucht erst output zu löschen. Ist danach build leer, wird auch build gelöscht. Ist danach projekt leer, wird auch projekt gelöscht.
-v, --verbose	Ausgabe einer Meldung für jedes gelöschte Verzeichnis.

Option	Beschreibung
--ignore-fail-on-non-empty	Unterdrückt die normale Fehlermeldung, falls ein Verzeichnis nicht leer ist. (Das Verzeichnis wird nicht gelöscht, aber es kommt kein Fehlerstatus; mit Vorsicht verwenden).
-r, -R, --recursive	Löscht Verzeichnisse <i>rekursiv</i> mitsamt ihrem gesamten Inhalt. Damit können komplette Verzeichnisbäume entfernt werden. Ohne -r verweigert rm das Löschen von Verzeichnissen mit Fehler "is a directory".
-d	Löscht ein <i>leeres</i> Verzeichnis. (Ohne -r löscht rm normalerweise gar keine Verzeichnisse; -d erlaubt es aber für leere, ähnlich rmdir. Nützlich, aber -r deckt diesen Fall auch ab.)

Nützlich: Um verschachtelte leere Ordner zu entfernen. Aber um Verzeichnisse mit Inhalt zu löschen, nutzt man rm -r.

rm (Remove Files/Directories)

Beschreibung: Löscht Dateien oder Verzeichnisse. Standardmäßig entfernt rm nur Dateien; mit -r kann es auch Verzeichnisbäume löschen (inkl. aller enthaltenen Dateien). **Vorsicht:** Gelöschte Dateien werden nicht ohne Weiteres wiederherstellbar gespeichert (kein Papierkorb, außer man nutzt safe-rm Tools).

Syntax:

rm [Optionen] <Datei/Pfad>...

Optionen:

Option	Beschreibung
-r, -R, --recursive	Löscht Verzeichnisse <i>rekursiv</i> mitsamt ihrem gesamten Inhalt. Damit können komplette Verzeichnisbäume entfernt werden. Ohne -r verweigert rm das Löschen von Verzeichnissen mit Fehler "is a directory".
-d	Löscht ein <i>leeres</i> Verzeichnis. (Ohne -r löscht rm normalerweise gar keine Verzeichnisse; -d erlaubt es aber für leere, ähnlich rmdir. Nützlich, aber -r deckt diesen Fall auch ab.)
-f, --force	<i>Force/ignore nonexistent:</i> Unterdrückt Rückfragen und Fehlermeldungen. Existiert eine angegebene Datei nicht, wird es ignoriert. Schreibschutz wird ignoriert (d.h. es fragt nicht nach Bestätigung bei schreibgeschützten Dateien). Kombiniert mit -r sehr gefährlich, da es kompromisslos alles löscht. Wird oft in Skripten genutzt.
-i	Interaktiv nachfragen vor <i>jedem</i> Löschvorgang. (Für viele Dateien unpraktisch).
-I	"ein wenig interaktiv": Fragt nur einmal pauschal nach, wenn mehr als 3 Dateien gelöscht werden sollen <i>oder</i> ein rekursiver Lösch auf einem Verzeichnis durchgeführt wird. (GNU rm Feature als Sicherheitsnetz)file-5y1edouysrow4zj6phjcqm.

Option	Beschreibung
--preserve-root	Schützt vor rm -rf /. Diese Option ist in GNU rm standardmäßig aktiv – es verweigert das komplette Löschen des Root-FS. Man kann mit --no-preserve-root überschreiben (nicht empfohlen).
-v, --verbose	Zeigt für jede Datei eine Meldung an, dass sie gelöscht wurde.
--help, --version	Hilfe bzw. Version.

Anwendung:

- **Einzelne Datei löschen:** rm datei.txt.
- **Mehrere Dateien löschen:** rm *.o (löscht alle .o Dateien im aktuellen Verzeichnis).
- **Verzeichnis und Inhalt löschen:** rm -r docs/ – löscht Ordner docs mitsamt allen Unterdateien.
- **Ohne Rückfrage alles löschen:** rm -rf tempdir. (Vorsicht!).

head (Dateianfang anzeigen)

Beschreibung: Gibt die ersten Zeilen einer Datei aus (standardmäßig 10 Zeilen). head ist nützlich, um schnell die ersten Zeilen (z.B. Header einer CSV, Beginn eines Logs etc.) anzuschauen, ohne die ganze Datei zu öffnen. Es kann auch binäre Dateien auf Anfang bytes prüfen.

Syntax:

head [Optionen] [Datei]...

Ohne Datei liest head von der Standardeingabe (z.B. in Pipes). Bei mehreren Dateien wird jeweils ein Header mit dem Dateinamen ausgegeben.

Optionen:

Option	Beschreibung
-n K, --lines=K	Gibt die ersten <i>K</i> Zeilen aus. <i>K</i> kann eine Zahl sein oder -K für alle bis auf die letzten <i>K</i> Zeilen (negativer Wert). Standard ist 10. Beispiel: head -n 20 file zeigt 20 Zeilen, head -n -5 file zeigt alle außer der letzten 5 Zeilen.
-c Bytes, --bytes=B	Gibt die ersten <i>B</i> Bytes aus (statt Zeilen). Kann auch mit K, M, etc. Suffix (Potenzen von 1024) angegeben werden. Z.B. head -c 100K bigfile. Negative Werte analog: -c -10 alles außer letzte 10 Bytes.

Option	Beschreibung
-q, --quiet/-silent	Unterdrückt die Überschriften, die vor dem Inhalt stehen, wenn mehrere Dateien angegeben wurden. Normal zeigt head sowas wie "=> filename <==" vor dem Block jeder Datei. Mit -q wird alles aneinandergehängt ohne solche Banner.
-v, --verbose	Zeigt immer die Dateiüberschriften an, auch wenn nur eine Datei angegeben ist. (Normalerweise bei nur einer Datei keine Überschrift).
--help, --version	Hilfe bzw. Version ausgeben.

Beispiel: head -n 3 /etc/passwd zeigt die ersten 3 Zeilen der passwd-Datei (typisch Root, daemon, bin Accounts).

tail (Dateiende anzeigen)

Beschreibung: Gibt die letzten Zeilen einer Datei aus (standardmäßig 10 Zeilen). Sehr nützlich, um z.B. das Ende von Logfiles zu sehen. Kann im *Follow-Modus* (-f) genutzt werden, um fortlaufend neuen Inhalt anzuzeigen (Echtzeit-Log-Überwachung).

Syntax:

tail [Optionen] [Datei]...

Optionen:

Option	Beschreibung
-n K, --lines=K	Zeigt die letzten <i>K</i> Zeilen an. Negative Werte bedeuten ab einer bestimmten Zeile von Anfang an (<i>ab</i> Zeile <i>K</i>). Standard ist 10. Z.B. tail -n 50 /var/log/syslog zeigt die letzten 50 Zeilen.
-c B, --bytes=B	Zeigt die letzten <i>B</i> Bytes an. Kann auch negative (<i>ab</i> Byte <i>X</i>) sein.
-f, --follow	Bleibt nach dem Anzeigen der letzten Zeilen geöffnet und zeigt fortlaufend neue Zeilen an, falls die Datei wächst (wie tail -f logfile zum Mitlesen).
-F	Ähnlich wie -f, aber zusätzlich "Wiederöffnen", falls Datei rotiert/neu erstellt wurde. (Follow Name statt Descriptor; praktisch für Logrotation – tail -F /var/log/app.log folgt auch nach Neustart des Logs dem neuen File).
-s Sek, --sleep-interval=Sek	Im Follow-Modus: Warte <i>Sek</i> Sekunden zwischen den Überprüfungen auf neue Daten. Standard 1.0 Sekunde. Kann z.B. auf 0.1 verringert werden, um schneller zu pollen, oder erhöht, um Last zu reduzieren.

Option	Beschreibung
--max- unchanged- stats=N	(Bei -F relevant) Wie oft in Folge unveränderte Inode geprüft wird, bevor tail davon ausgeht, dass ggf. Logrotate stattgefunden hat.
-q, --quiet	Unterdrückt Überschriften bei mehreren Dateien (analog head).
-v, --verbose	Erzwingt Überschriften, selbst bei einer Datei.
--help, --version	Hilfe bzw. Version.

Anwendung:

- **Letzte Zeilen einer Datei:** tail -n 20 messages.log (Zeigt die letzten 20 Zeilen).
- **Live-Verfolgung:** tail -f /var/log/syslog (Druckt neu geschriebene Logzeilen fortlaufend). Abbruch mit Ctrl+C.
- **Combine mit grep:** z.B. tail -f server.log \| grep "ERROR" filtert live nur Zeilen mit "ERROR".

Dateien finden und Text filtern

grep (Global Regular Expression Print)

Beschreibung: Durchsucht die Eingabe (Dateien oder Standard Input) nach Zeilen, die einem bestimmten Muster entsprechen, und gibt diese Zeilen ausfile-4fdk7rvx39azjretinarqb. grep arbeitet mit regulären Ausdrücken (Standard *Basic Regex*, erweiterte mit -E). Es ist das klassische Suchwerkzeug in Textstreams. Name stammt von der ed-Editor Anweisung g/re/p (global regex print).

Syntax:

grep [Optionen] 'Muster' [Datei...]

Optionen:

Option	Beschreibung
-i, --ignore-case	Ignoriert Groß-/Kleinschreibung beim Mustervergleichfile-4fdk7rvx39azjretinarqb. "Test" findet auch "test" oder "TEST".
-v, --invert- match	Gibt alle Zeilen aus, die nicht auf das Muster passen (Inverse Treffer)file-4fdk7rvx39azjretinarqb. Nutzt man oft zur Ausschlussfilterung.
-r, -R, -- recursive	Durchsucht rekursiv alle Dateien in angegebenen Verzeichnissen (und Unterverz.)file-4fdk7rvx39azjretinarqb. Ohne diese Option

Option	Beschreibung
	behandelt grep Verzeichnisargumente normalerweise als Fehler oder ignoriert sie. (GNU grep: -r folgt Symlinks nicht, -R folgt Symlinks.)
-n, --line-number	Zeigt vor jeder gefundenen Zeile die Zeilennummer in der Datei an ibm.com . Format: Datei:Nr:Zeilentext.
-A NUM	(After) Zeigt <i>NUM</i> Zeilen nach jeder Trefferzeile zusätzlich an (Context) ibm.com .
-B NUM	(Before) Zeigt <i>NUM</i> Zeilen vor jeder Trefferzeile an.
-C NUM	(Context) Zeigt <i>NUM</i> Zeilen <i>vor und nach</i> jedem Treffer (gleichzeitig -A NUM -B NUM). Standard -C 2 beispielsweise gibt 2 Kontextzeilen um jede Fundstelle.
-l, --files-with-matches	Listet nur die Dateinamen auf, die mindestens einen Treffer enthalten. Pro Datei ein Name, Inhalt wird nicht ausgegeben. (Nützlich um zu sehen wo etwas vorkommt)
-L, --files-without-match	Listet nur Dateinamen, die <i>keinen</i> Treffer enthalten (Gegenteil von -l).
-c, --count	Zählt die Treffer pro Datei und gibt nur diese Anzahl aus (zusammen mit Dateinamen, außer mit -h). D.h. "file.txt:3". In Kombination mit -v zählt es Nicht-Treffer-Zeilen.
-o, --only-matching	Gibt nur den Teil der Zeile aus, der dem Suchmuster entspricht, nicht die ganze Zeile. Bei mehreren Treffern pro Zeile werden diese getrennt ausgegeben.
-q, --quiet/-silent	"Quiet": gibt keine Ausgabe; liefert lediglich Exit-Status 0 falls wenigstens ein Match gefunden wurde, 1 falls kein Match. Praktisch in Skripten (nur Rückgabewert interessiert).
-s, --no-messages	Unterdrückt Fehlermeldungen über nicht lesbare Dateien (z.B. wenn Zugriffsrechte fehlen oder Datei nicht existiert).
-E, --extended-regexp	Interpretiert das Muster als <i>Extended Regular Expression (ERE)</i> ionos.at . Das erlaubt z.B. Nutzung von +, ?, `
-F, --fixed-strings	Interpretiert das Muster als <i>fixen String</i> (kein Regex, jedes Zeichen literal). Entspricht fgrep (das eine schnellere Suche ohne regex

Option	Beschreibung
	ermöglicht). Alle Muster (wenn -e mehrfach) werden als Literalstrings gesucht.
-P, --perl-regexp	Interpretiert das Muster als Perl-Compatible Regex (PCRE), falls grep mit PCRE-Unterstützung kompiliert wurde. Damit stehen mächtigere Regex-Features zur Verfügung.
-h, --no-filename	Unterdrückt die Ausgabe des Dateinamens vor jeder Trefferzeile (wenn mehrere Dateien durchsucht werden). Standardmäßig präfixt grep Datei: bei mehreren Input-Dateien. Mit -h erscheinen nur die Zeilen. (Bei Einzeldatei ist das sowieso Standard.)
-H, --with-filename	Zeigt Dateinamen auch bei einziger Datei an. (Umgekehrt zu -h). Nützlich in Pipelines wo Source unbekannt.
--color[=WHEN]	Hebt das gefundene Pattern farblich hervor in der Ausgabe. WHEN kann auto (Standard, nur wenn Terminal), always oder never sein. Setzt Umgebungsvariable GREP_COLOR/GREP_COLORS ggf. benötigt.
-e Pattern	Gibt das Suchmuster explizit an. Nötig z.B. wenn das Pattern mit - beginnt (damit es nicht als Option interpretiert wird). Kann mehrfach angegeben werden, um mehrere Muster (OR-Verknüpfung) zu suchen.
-f Datei	Liest Suchmuster aus angegebener Datei (ein Pattern pro Zeile). Nützlich bei sehr vielen oder komplexen Patterns.
--exclude=GLOB	Schließt Dateien vom Durchsuchen aus, deren Name auf GLOB (Wildcard-Muster) passt. Z.B. --exclude=*.min.js würde Minimierte JS ignorieren. (Nur wirksam bei rekursiver Suche oder wenn Verzeichnis als Input). Es gibt auch --include analog.
--exclude-dir=GLOB	Wie --exclude, aber nur auf Verzeichnisnamen angewandt, um ganze Verzeichnisse von der Rekursion auszunehmen (z.B. --exclude-dir=.git).
--help, --version	Hilfe / Version.

Exit-Status: 0 falls mindestens ein Match gefunden, 1 falls kein Match, 2 falls ein Fehler auftrat (z.B. Datei nicht lesbar).

Beispiele:

- grep -i "error" *.log – sucht case-insensitive "error" in allen .log im aktuellen Ordner, gibt Trefferzeilen mit Dateipräfix aus.
- grep -R "TODO" src/ – findet rekursiv alle Vorkommen von "TODO" in Dateien unterhalb von src/ (typisch in Codebasis).
- grep -nE "^Error|Warning:" report.txt – sucht Zeilenbeginn "Error:" oder "Warning:" und gibt mit Zeilennummer aus [pq.hosting](#).
- ps ax | grep "[c]ron" – findet Prozesse mit "cron" im Namen. (Trick: [c] im Pattern matcht 'c', verhindert aber, dass der grep selbst im ps-Listing gefunden wird, da dessen Kommandozeile "grep cron" kein 'cron' substring hat sondern 'cro n' durch char class unterbrochen).

find (Dateien suchen)

Beschreibung: Durchsucht Verzeichnisse nach Dateien/Verzeichnissen, die bestimmten Kriterien entsprechen (Name, Typ, Größe, Datum, Rechte, Eigentümer etc.)
file-4fdk7rvx39azjretinarqb. find ist sehr mächtig: man kann logische Bedingungen kombinieren und Aktionen auf gefundene Dateien ausführen (löschen, verschieben, Exec von Kommandos etc.). Es arbeitet rekursiv ab dem angegebenen Pfad.

Syntax:

find [Start-Verzeichnis] [Optionen/Ausdrücke] [Aktion(en)]

Wichtige Suchkriterien (Ausdrücke):

- **-name Muster** – Dateiname entspricht *Wildcards* (Globbing wie bei Shell, z.B. "*.txt"). Groß/Kleinschreibung beachten.
- **-iname Muster** – wie -name, aber case-*insensitive*.
- **-type c** – Dateityp ist c. Häufig: **f** für regular file, **d** für directory, **l** symlink, **b** blockdevice, **c** chardevice, **p** named pipe, **s** socket. Z.B. -type d nur Verzeichnisse
file-4fdk7rvx39azjretinarqb.
- **-user name/uid** – Datei gehört dem Benutzer *name* (oder UID numerisch). Analog **-group name/gid**.
- **-uid +n/-n/n** – UID der Datei ist ($>/</=$) n. -uid +500 UID größer 500. (Entspricht -user nur numerisch und mit Vergleichen). Genauso -gid.
- **-size N*[bckMG]*** – Dateigröße-Kriterium. *N* kann mit Vorzeichen angegeben werden (+ größer als, - kleiner als, nichts = genau). Suffix: standardmäßig in 512-Byte Blöcken, oder **c** = Bytes, **k** = KiB, **M** = MiB, **G** = GiB. Beispiel: -size +100M Dateien größer 100 MiB [wiki.foxtom.de](#), -size -10k kleiner 10 KiB.
- **-mtime Tage** – Letzte Änderung (Modify timestamp) liegt *Tage* zurück. *Tage* als ganze Zahl: +n = älter als n Tage, -n = jünger als n Tage, n = genau n Tage (im

Bereich ±0.5d). Ähnlich **-atime** (Access), **-ctime** (Statusänderung). Z.B. -mtime -7 in letzten 7 Tagen modifiziert wiki.foxtom.de.

- **-newer Datei** – Datei wurde später geändert als *Datei* (Vergleich mtime). **-newermt Datum** (GNU find) erlaubt Vergleich mit einem Zeit-String, z.B. -newermt "2025-05-01".
- **-perm mode** – Dateirechte entsprechen *mode*. Mode kann exakt (z.B. 644) oder mit / oder - angegeben werden für Masken. Beispiel: -perm 644 findet genau 644. -perm -644 findet Dateien, die mindestens diese Bits haben (also rw- r-- r-- oder mehr Rechte). -perm /222 findet Dateien, bei denen irgendein Schreibbit gesetzt ist (Owner oder Group oder Other).
- **-maxdepth N** – Nur bis Tiefe N ins Verzeichnis hinab suchen (0= nur Startdir selbst). **-mindepth N** analog minimal tiefe. Z.B. find . -maxdepth 1 -type f nur Dateien im aktuellen Verzeichnis, nicht tiefer.
- **-mount** oder **-xdev** – Verlasse nicht das Dateisystem. D.h. keine Verzeichnisse auf anderen gemounteten FS durchsuchen (z.B. ignoriert find / -xdev dann andere Partitionen).
- **-prune** – Überspringt (schließt vom Durchsuchen aus) ein Verzeichnis. Bsp: find . -path "./node_modules" -prune -o -name "*.js" -print durchsucht alles außer den node_modules Ordner.

Aktionen (wenn Suchkriterien wahr sind):

- **-print** – Gibt gefundene Pfade aus (Standard, wird angenommen, wenn keine andere Aktion).
- **-ls** – Führt ls -dils auf die gefundene Datei aus (anzeige von detaillierten Infos, inkl. Inode, Größe, Rechte etc.).
- **-delete** – Löscht gefundene Dateien *unmittelbar*. Achtung: find -delete entfernt auch Verzeichnisse, wenn leer geworden. Sollte meist mit Kriterien kombiniert werden (z.B. -type f -delete nur Dateien). **Keine** Aktion mehr nach -delete wird ausgeführt (beendet Durchlauf in diesem Pfad).
- **-exec Befehl {} ;** – Führt einen externen Befehl für jede gefundene Datei aus. {} wird durch den Pfad ersetzt. Die Kommandozeile wird so oft ausgeführt wie Treffer. Beispiel: find . -name "*.bak" -exec rm {} \; löscht alle .bak-Dateien (funktional wie -delete, aber universeller)file-5y1edouysrow4zj6phjcqm. Das \; markiert Ende des -exec Befehls.
- **-exec ... {} +** – Variante, die versucht, mehrere gefundene Dateien in *einen* Aufruf zu packen (fügt die Pfade an Stelle von {} möglichst gesammelt, wie xargs). Effizienter, falls das Programm mehrere auf einmal verarbeiten kann. Z.B. -exec chmod 644 {} + ändert die Rechte aller gefundenen Files, ggf. in batched Calls.

- **-execdir** – Wie -exec, aber führt das Kommando im jeweiligen Verzeichnis der Datei aus (Working Directory wechselt). Erhöht Sicherheit bei {} mit {} containing spaces.

Beispiele:

- find /var/log -type f -name "*.log" -size +1M -mtime +30 – findet Log-Dateien über 1MB, älter als 30 Tage unter /var/log.
- find . -maxdepth 1 -type d ! -name ":" -exec rm -r {} \; – löscht alle Unterverzeichnisse im aktuellen Ordner (maxdepth 1, alles was Type d ist und Name nicht ":"), dann rm -r).
- find /home/user -xtype l -delete – löscht gebrochene Symlinks (xtype l findet Links, deren Ziel nicht existiert).
- find . -perm /u+x -print – findet Dateien, bei denen das User-execute Bit gesetzt ist (also alle ausführbaren Dateien).
- find . -name ".*.tmp" -execdir gzip {} \; – packt alle *.tmp im jeweiligen Verzeichnis in eine .tmp.gz (ausgeführt im Ort der Datei).

locate (Datenbank-Dateisuche)

Beschreibung: Durchsucht eine vorgefertigte Index-Datenbank nach Dateinamen (Pfaden), die einem angegebenen Suchstring entsprechenfile-4fdk7rvx39azjretinarqb. locate ist viel schneller als find, da es nicht live das Dateisystem scannt, sondern auf einer regelmäßig aktualisierten Datenbank (typisch via updatedb in Cron) basiert. Nachteil: findet keine erst sehr kürzlich hinzugefügten Dateien (bis zum nächsten updatedb-Lauf) und standardmäßig nur solche, auf die der User zugreifen darf.

Syntax:

locate [Optionen] <Suchmuster>

Das Suchmuster ist im Grunde ein Substring oder einfache Glob. Standardmäßig sucht locate alle Datenbankeinträge, die <Suchmuster> als Teilstring enthalten.

Optionen:

Option	Beschreibung
-i, --ignore-case	Sucht case-insensitive (Groß/Kleinschreibung ignorieren)file-4fdk7rvx39azjretinarqb.
-r, --regex	Interpretiert das angegebene Muster als regulären Ausdruck (POSIX ERE) statt als einfachen Textfile-4fdk7rvx39azjretinarqb.

Option	Beschreibung
-w, --wholename	(Standardverhalten) Sucht im gesamten Pfadnamen. (Bei einigen Implementierungen war locate nur substring-suche, wholename explizit wie default, hier nur Info.)
-b, --basename	Sucht nur im Dateinamen selbst, nicht im gesamten Pfad. (Also ignoriert Verzeichnisnamen im Match.)
-c, --count	Zählt nur die Anzahl der Fundstellen und gibt diese Zahl aus, anstatt alle Ergebnisse zu listen.
-l N, --limit=N	Gibt maximal N Ergebnisse aus (dann Abbruch).
--existing	Prüft gefundene Pfade und filtert jene heraus, die aktuell nicht (mehr) existieren. (Die DB könnte veraltete Einträge haben; mit --existing werden nur noch valide Treffer angezeigt, obwohl es Zeit kostet, jeden Pfad zu verifizieren.)
-0, --null	Trennt die Ausgabe mit Nullbytes anstatt neuer Zeile. Nützlich, um in xargs -0 weiterzuverarbeiten.
-e, --nofollow	Ignoriert Einträge, die Symlinks auf nicht-existente Dateien sind (könnte Implementation-spezifisch).
-S	Zeigt Statistik der Datenbank (Anzahl Einträge etc.) statt zu suchen.
-d DB, --database=DB	Nutzt eine alternative Datenbank-Datei anstelle der Standard-Datenbanken (/var/lib/mlocate.db z.B.). Man kann mehrere via -d db1:db2:... angeben.
--help, --version	Hilfe bzw. Version.

Beispiel: locate -i '\.pdf' listet alle Pfade, die ".pdf" (unabhängig von Großschreibung) enthalten – das würde i.d.R. alle PDF-Dateien auf dem System listen, die zum Zeitpunkt der letzten Datenbankaktualisierung existierten. Oft benutzt: locate <Programmname> um herauszufinden, wo ein Programm oder dessen Konfigdateien liegen.

updatedb (Update Database for locate)

Beschreibung: Baut die Datei-Datenbank für locate neu auf. Es durchsucht das Dateisystem (oder definierte Pfade) und speichert alle Dateinamen/Pfade in einer komprimierten Datenbank (typischerweise /var/lib/mlocate/mlocate.db). updatedb wird meist regelmäßig (täglich) via Cron ausgeführt, man kann es aber auch manuell starten, z.B. nach größeren Änderungen im FS, damit locate sofort aktuell ist.

Syntax:

updatedb [Optionen]

Optionen: (Häufig werden diese über /etc/updatedb.conf voreingestellt)

Option	Beschreibung
--prune-bind-mounts {yes no}	Ob Bind-Mountpoints ausgelassen werden (Default ja – doppelte Scans vermeiden).
--prune- paths='PFADLISTE'	Doppelpunkt-separierte Liste von Verzeichnissen, die <i>nicht</i> indexiert werden sollenfile-4fdk7rvx39azjretinarqb. Standardmäßig z.B. /tmp, /proc, /sys, /run, /dev, /var/cache etc., um unnötiges zu sparen. Dies entspricht PRUNEPATHS in /etc/updatedb.conf.
--prune- filesystems='TypList'	Liste von Dateisystemtypen, die nicht durchsucht werden (z.B. nfs,squashfs etc.). Standard meist Netzwerk-FS, Pseudo-FS (proc, sysfs) etc.file-4fdk7rvx39azjretinarqb. Entspricht PRUNEFS.
-l N	--localpaths=LIST – falls man nur lokale Pfade indexieren will (veraltet, use prune).
-U 'PFADLISTE'	Nur diese durch Doppelpunkt getrennten Pfade als Wurzel für Index verwenden (Standard "/" – gesamtes FS). Kann man nutzen um z.B. nur /home zu indexieren.
-o DBFILE	Output: speichere DB in angegebener Datei statt Standard. (So kann man separate DBs erzeugen.)
-f 'FS-Liste'	Nur diese Dateisystemtypen berücksichtigen (Inverse von PRUNEFS).
-v	Ausführliche Ausgabe während des Indizierens.

In der Regel wird updatedb ohne Optionen (ggf. als root über sudo updatedb) ausgeführt und nutzt die System-Voreinstellungen. Danach steht die Datenbank locate zur Verfügung.

which (Welches Kommando - Pfadsuche)

Beschreibung: Zeigt den Pfad zu einem ausführbaren Programm, das in der Umgebungsvariablen \$PATH gefunden wirdfile-4fdk7rvx39azjretinarqb. Damit kann man herausfinden, welches Programm aufgerufen würde, wenn man einen bestimmten Befehl tippt (nützlich bei mehreren Versionen in verschiedenen Verzeichnissen).

Syntax:

which [Optionen] Befehlsname...

Optionen:

Option	Beschreibung
-a	Zeigt alle Vorkommen des angegebenen Befehls im Suchpfad an (nicht nur das erste gefundene) file-4fdk7rvx39azjretinarqb. Z.B. which -a python listet evtl. /usr/bin/python und /usr/local/bin/python falls beide existieren.
--skip-dot , --skip-tilde	Ignoriert . bzw. ~ am Anfang des Pfades (also aktuelles Verzeichnis oder Home in PATH) in der Suche. (Sicherheitsaspekt – Standard bei which).
--show-dot , --show-tilde	Zeigt explizit, wenn gefundene Einträge in . oder ~ liegen, durch Präfix "./" oder "~/".
--tty-only	Zeigt nur etwas an, wenn stdout ein Terminal ist. Unterdrückt z.B. Ausgabe, wenn in Pipe (verhindert sich einzumischen).
--help , --version	Hilfe bzw. Versionsinfo (gilt bei GNU which; einige shells haben eingebautes which mit anderem Verhalten).

Hinweis: which ist oft eine Shell-Funktion/Alias und verhält sich u.U. leicht unterschiedlich. Ähnliche Tools: type (shell-builtin, zeigt auch Aliases, Keywords), command -v.

Beispiel: which gcc -> /usr/bin/gcc (Pfad zum C-Compiler). which -a ls könnte z.B. zeigen /usr/bin/ls und /bin/ls falls beides symlinks aufeinander oder verschieden, etc.

Textverarbeitung und Streams (Shell, Scripting)

awk (Muster/Aktions-Scanner)

Beschreibung: AWK ist ein mächtiges Textverarbeitungswerkzeug und Skriptsprache, mit dem man Textdateien zeilenweise nach Mustern durchsuchen und pro Zeile Aktionen ausführen kann file-4fdk7rvx39azjretinarqb. Typischerweise wird awk benutzt, um Spalten auszugeben oder zu filtern, da es Eingabezeilen automatisch in Felder zerlegt. Der Name stammt von den Anfangsbuchstaben seiner Schöpfer (Aho, Weinberger, Kernighan).

Syntax:

awk [Optionen] 'Muster { Aktion }' Datei...

Jede Zeile der Datei wird gelesen; wenn das *Muster* passt, wird die *Aktion* ausgeführt (z.B. etwas ausgeben). AWK hat eigene Variablen (\$1 = erstes Feld, \$0 = ganze Zeile) und eingebaute Rechen- und String-Features.

Optionen:

Option	Beschreibung
-F separator	Legt das Eingabefeld-Trennzeichen fest (Field Separator). Standard sind whitespace (Leerzeichen/Tabs)file-4fdk7rvx39azjretinarqb. Z.B. -F ":" für /etc/passwd mit Doppelpunkten.
-v Var=Wert	Setzt eine AWK-Variable <i>Var</i> auf <i>Wert</i> vor dem Verarbeiten. Z.B. -v ORS="\n\n" um eine Leerzeile nach jeder Ausgabezeile einzufügen (ORS = Output Record Separator).
-f Scriptdatei	Lies das AWK-Programm aus einer Datei anstatt aus der Befehlszeile. Damit kann man größere AWK-Skripte in eigenem File verwalten.
-m, --sandbox	(gawk) AWK in sicheren Modus schalten (kein System(), keine Schreibrechte auf FS etc).
--posix	Schaltet in POSIX-kompatiblen Modus (ggf. einige GNU-Erweiterungen aus).
-W version	Zeigt AWK-Version an (für gawk).
--help, --version	Hilfe / Version.

AWK Sprachkonzepte (Kurz):

- Ein AWK-Programm besteht aus *Muster { Aktion }* Blöcken. Muster ist ein Regex oder Ausdruck, Aktion ist Code in AWK.
- Felder: Eingabezeile wird in Felder zerlegt, zugreifbar als \$1, \$2, ..., \$NF (NF=Anzahl Felder). \$0 ist die ganze Zeile.
- Beispiele:
 - awk '{ print \$1 }' file – gibt die erste Spalte jeder Zeile aus (mit Standard-Trenn whitespace).
 - awk '/error/ {print NR ":" "\$0}' log.txt – sucht Zeilen, die "error" enthalten (Regex als Muster), und druckt Zeilennummer und kompletten Inhalt.
 - awk -F ',' '\$3 > 100 { sum += \$3 } END { print "Summe:", sum }' data.csv – mit Komma als Separator summiere Wert der 3. Spalte aller Zeilen >100 in Spalte3; am Ende (END-Block) gib Summe aus.
- AWK hat viele eingebaute Funktionen (Mathe, String), assoziative Arrays, Kontrollstrukturen (if, for, while).

sed (Stream Editor)

Beschreibung: sed ist ein nicht-interaktiver Stream-Editor zum Bearbeiten von Texteingaben nach Mustern, insbesondere für Suchen&Ersetzen oder das Extrahieren von Zeilenfile-4fdk7rvx39azjretinarqb. sed liest zeilenweise Eingabe (Dateien oder STDIN), wendet die angegebenen *Editier-Befehle* darauf an und schreibt das Ergebnis auf STDOUT (oder optional in-place zurück in Dateien). Am verbreitetsten ist die *Substitution* mit dem s/alt/neu/ Kommando.

Syntax:

sed [Optionen] 'Script' Datei...

Das *Script* besteht aus einem oder mehreren Editierbefehlen, die auf die Eingabe angewendet werden. Beliebte sed-Befehle: **s** (substitute), **p** (print), **d** (delete), **q** (quit), **{}** für Gruppen, Adressierungsmodi (Zeilen-Nr oder Regex).

Optionen:

Option	Beschreibung
-n, --quiet/--silent	Suppress normal output. Normal druckt sed jede Eingabezeile (ggf. modifiziert) automatisch. Mit -n wird nichts automatisch gedruckt, nur explizite p (print) Befehle geben ausfile-4fdk7rvx39azjretinarqb. Typisch in Kombination mit p in Script, um nur bestimmte Zeilen auszugeben.
-e Script	Gibt das zu verwendende sed-Skript (Editierbefehle) direkt an. Kann mehrfach genutzt werden für mehrere Befehle in Reihenfolge. Wenn nur ein Skript-Argument ohne -e gegeben ist, kann -e weggelassen werden.
-f Datei	Lies das sed-Skript aus einer Datei. Erlaubt komplexere Edits aus einem File anstatt Kommandozeile.
-i[SUF], --in-place[=SUF]	Bearbeitet Dateien <i>in place</i> , d.h. schreibt Änderungen direkt zurück in die Eingabedateien. Optional kann ein <i>Suffix</i> angegeben werden, um vor Änderungen ein Backup der Originaldatei anzulegen (z.B. -i.bak). Ohne Suffix wird die Datei direkt überschrieben ionos.de .
-r, -E	Schaltet die Verwendung erweiterter regulärer Ausdrücke ein (so dass in Patterns +, ?, {} etc. ohne Backslash verwendet werden können). Bei GNU sed ist -r (old syntax) und -E (wie bei BSD sed) identisch.
-s	Treat files separately. Normalerweise bei mehreren Dateien behandelt sed den Input als einen langen Stream. Mit -s wird nach jeder Datei der interne Zustand (z.B. Zeilennummer, \$-Regex-Flag) zurückgesetzt.
-z, --null-data	Input wird als Null-terminiert (Zeilen getrennt durch ASCII NUL statt newline) angenommen. Ermöglicht Verarbeitung von Binär oder

Option	Beschreibung
	multiline-chunks. Entspricht dann, dass . nicht auch \n matcht und \$ nur am NUL erkennt. (GNU sed)
--follow-symlinks	Wenn in-place auf Symlinks angewendet, ändert sed standardmäßig den Link selbst (also das Zielfile, aber eben durch den Link?). Mit dieser Option folgt sed dem Symlink und editiert die Ziel-Datei. (GNU sed)
--sandbox	Verhindert, dass sed w oder s///w zum Schreiben verwendet oder ! Shell-Aufrufe macht. (GNU sed 4.4+)
--help, --version	Hilfe / Version.

Häufige sed-Anwendungen:

- **Substitution (Ersetzen):**
Syntax: s/REGEX/REPLACEMENT/[Flags]. Ersetzt das erste Vorkommen von REGEX in jeder Zeile durch REPLACEMENT. Flags: g = global (alle Vorkommen in Zeile ersetzen), i = case-insensitive, p = print (selbst wenn -n), w file = gefundene Zeilen nach file schreiben.
Beispiel: sed -i 's/oldword/newword/g' file.txt – ersetzt *oldword* durch *newword* überall in file.txt, direkt in der Datei file-5y1edouysrow4zj6phjcqm.
Besonderheit: Im Ersatz kann man & verwenden für den gesamten Match oder \1 ... für Capturing Groups aus dem Regex.
- **Zeilen löschen:**
z.B. sed '/^#/ d' config – löscht alle Zeilen, die mit # beginnen (d = delete). Oder sed '5,10d' file löscht Zeilen 5 bis 10.
- **Nur bestimmte Zeilen ausgeben:**
Mit -n und p. Z.B. sed -n '1,5p' file druckt nur Zeilen 1-5 (andere werden durch -n nicht ausgegeben). Oder sed -n '/ERROR/p' log druckt nur Zeilen mit "ERROR".
- **Mehrere Edits:**
z.B. sed -e 's/foo/bar/' -e '/BAZ/d' – erst foo->bar ersetzen, dann Zeilen mit "BAZ" löschen. (Die Reihenfolge ist wichtig).
- **In-Place Backup:**
sed -i.orig 's/^#/.*conf' – kommentiert alle Zeilen in *.conf aus, speichert vorher Backup mit .orig-Suffix.

Sed kann viel mehr (Insert/Append mit i\ und a\, Transform mit y///, Shell-Befehle mit !, Holding Space etc.), was in Tiefe für Prüfung wohl nicht nötig ist. Wichtig ist Suchen&Ersetzen und Basisfilter.

tee (Output verdoppeln)

Beschreibung: Liest die Standard-Eingabe und schreibt sie *gleichzeitig* auf die Standard-Ausgabe **und** in eine oder mehrere Dateienfile-4fdk7rvx39azjretinarqb. Es "verzweigt" den Datenstrom (daher der Name T-Stück). So kann man die Ausgabe eines Kommandos in eine Datei umleiten, aber trotzdem im Terminal sehen.

Syntax:

... | tee [Optionen] <Datei>...

Optionen:

Option	Beschreibung
-a, --append	Hängt die Ausgabe an bestehende Dateien an, anstatt sie zu überschreibenfile-4fdk7rvx39azjretinarqb. Standard ohne -a überschreibt Ziel-Dateien neu.
-i, --ignore-interrupts	Ignoriert Interrupt-Signale (SIGINT). D.h. Ctrl+C wirkt nicht auf tee selbst (nur auf vorheriges Kommando?), damit tee garantiert die Daten schreibt.
--output-error MODE	Verhaltensweise falls ein Schreibfehler auftritt: warn (Fehler melden, aber weiter versuchen andere outputs zu schreiben), exit (sofort beenden, evtl. ungeschriebene outputs gehen verloren). Standard ist warn.

Anwendung:

- make 2>&1 | tee build.log – Zeigt alle Ausgaben von make im Terminal und speichert sie gleichzeitig in *build.log*.
- ls -l | tee -a listing.txt – Ausgabe von ls -l normal anzeigen und ans Ende von listing.txt anfügen.
- echo "strict mode on" | sudo tee /proc/sys/kernel/yama/ptrace_scope – Durch sudo tee kann man Inhalte in Dateien schreiben, für die man Rechte braucht, während Eingabe evtl. aus einem un-privilegierten Echo kommt.

Tee ist also nützlich bei Logging und bei Verwendung mit sudo zum in Dateien schreiben, da sudo echo >> file wegen Redirection sonst nicht klappt.

xargs (Execute Arguments)

Beschreibung: Liest Eingaben (Standardinput oder Datei) und führt daraus Kommandoaufrufe zusammengebaut aus. Typisch, um die Ausgabe eines Programms als Argumentliste für ein anderes Programm zu verwendenfile-4fdk7rvx39azjretinarqbfile-4fdk7rvx39azjretinarqb. xargs nimmt zeilen- oder spaltengetrennte Wörter und packt möglichst viele davon in einen Aufruf eines

angegebenen Kommandos (Standard: echo). Damit überwindet man die Shell-Längenbeschränkung und kann sequentiell Befehle auf Listen anwenden.

Syntax:

xargs [Optionen] [Befehl [Initial-Argumente]]

Wenn kein Befehl angegeben ist, benutzt xargs per Default echo (einfaches Ausgeben).

Optionen:

Option	Beschreibung
-n MaxArgs	Maximal <i>MaxArgs</i> Argumente pro aufgerufenem Kommando verwenden. Z.B. -n 1 bedeutet, xargs nimmt jeweils 1 Argument und führt das Kommando damit aus (also für jede Eingabezeile einmal). Ohne Angabe bündelt xargs so viele Argumente wie möglich pro Aufruf, bis die Systemlimite (Getconf ARG_MAX) fast erreicht ist.
-L MaxLines	Nimmt maximal <i>MaxLines</i> Input-Zeilen pro Kommando-Aufruf. Z.B. -L 1 bewirkt, dass jede einzelne Zeile als eine xargs-Ausführung behandelt wird (selbst wenn Zeile mehrere Wörter enthält).
-d Delim	Verwendet <i>Delim</i> als Trennzeichen anstelle von Whitespace oder newline. Kann auch mehrere Zeichen (String) sein (GNU). Beispiel: -d ':' um : als Separator zu nehmen.
-0, --null	Erwartet Input-Felder durch Nullbyte getrennt (wie von find -print0 oder grep -Z erzeugt). Dies ermöglicht sichere Übergabe von Dateinamen mit Leer-/Sonderzeichen stackoverflow.com .
-E EOFSTR	Definiert eine Zeichenfolge, die als Ende der Eingabe interpretiert wird. Wenn xargs diese Zeichenfolge liest, bricht es die weitere Verarbeitung ab (auch wenn nicht EOF).
-e	(veraltet, äquivalent zu -E ohne Parameter) – historically to specify EOF marker, oft nicht mehr benötigt.
-I ReplaceStr	Nutzt <i>ReplaceStr</i> als Platzhalter in dem auszuführenden Kommando, der durch den gesamten Input (pro Aufruf) ersetzt wird. file-4fdk7rvx39azjretinarqb. Z.B. xargs -I {} mv {} /backup nimmt jeden Input (eine Datei) und substituiert anstelle von {} im mv-Aufruf. Bei -I wird implizit -L 1 angenommen (d.h. pro Zeile ein Aufruf).
-J ReplaceStr	(BSD xargs) Ähnlich -I, aber ersetzt nur einmal in der Kommandozeile (kann mit -n kombinieren). In GNU xargs gibt es -J nicht.

Option	Beschreibung
-t, --verbose	Gibt das Kommando, das ausgeführt wird, vorher auf stderr aus (also "echo" what is being done). Zum Debuggen hilfreich.
-p, --interactive	Fragt vor <i>jedem</i> Kommando nach Bestätigung (y/n), ähnlich wie -t zeigt es Befehl. Bei 'y' ausführen, 'n' überspringen.
-r, --no-run-if-empty	Führt das Kommando nicht aus, wenn keine Argumente aus Input entstehen. Standard bei GNU xargs ist, <i>kein</i> Leerlaufaufruf zu machen (sprich diese Option ist implizit an). In POSIX könnte Standard anderes sein.
-0, --null	(siehe oben) Null-terminierte Eingabe verwenden – wichtig in Kombination mit find -print0 um sicher Dateilisten zu pipen ohne Probleme mit Sonderzeichen stackoverflow.com .
--max-chars=K, -s K	Maximal K Zeichen pro Kommandozeilen-Aufruf verwenden (inklusive Kommando und env). Standard an System ARG_MAX orientiert (oft ~131072). Manchmal zu reduzieren bei seltsamen Umgebungen.
--max-procs=N, -P N	Bis zu N Prozesse <i>parallel</i> ausführen statt nacheinander. Z.B. -P 4 führt bis zu 4 Instanzen des Kommandos gleichzeitig aus, um Parallelisierung zu erreichen. N=0 bedeutet unbegrenzt viele parallel (Vorsicht!). Hilfreich z.B. in Multi-Core-Systemen.
--process-slot-var=NAME	Setzt eine Umgebungsvariable \$NAME in den aufgerufenen Prozessen, die eindeutige ID des xargs-Slots enthält (0..N-1) – bei -P parallel. Hilft um Ausgaben zuzuordnen.
-a Datei	Liest Input aus <i>Datei</i> anstatt STDIN. So kann xargs auch ohne Pipe aus File speisen.
--help, --version	Hilfe/Version.

Beispiele:

- find /tmp -type f -name '*.txt' -print0 | xargs -0 rm -f – Löscht alle .txt-Dateien unter /tmp (sicher gegenüber Sonderzeichen, dank -print0 und -0).
- grep -l "ERROR" *.log | xargs -r zip errors.zip – Sucht in allen .log nach "ERROR", die -l Option listet nur Dateinamen mit Treffer. Diese Liste wird an xargs übergeben, das dann zip errors.zip <files> ausführt. Falls kein Treffer, -r sorgt dafür, dass zip nicht ohne Argument aufgerufen wird.

- `cat list.txt | xargs -I {} echo File: {} >> files.txt` – Für jede Zeile aus list.txt, führe `echo File: <line>` aus und append an files.txt. (So fügt man "File: " vor jede Zeile).
- `seq 1 1000 | xargs -n 100 > numbers.txt` – Nimmt die Zahlen 1 bis 1000 (jede auf neuer Zeile) und packt jeweils 100 Zahlen in eine Zeile, schreibt nach numbers.txt.
- `printf "%s\n" * | xargs -0 -n 1 -P 4 sha256sum` – Berechnet parallel (4 at a time) die SHA256 Summen aller Dateien im aktuellen Verzeichnis. `printf "%s\n" *` generiert Null-separierte Liste aller Dateien, xargs nimmt je 1 (-n1) und startet bis zu 4 sha256sum gleichzeitig.

Software- und Paketmanagement

tar (Tape ARchiver)

Beschreibung: Archiviert mehrere Dateien zu einer einzigen Datei (tarball). Tar kann Dateien **bündeln** (zu .tar Archiven) und in Zusammenarbeit mit Kompressionsprogrammen (gzip, bzip2, xz etc.) zugleich komprimierenfile-4fdk7rvx39azjretinarqb. Oft verwendet, um Verzeichnisse oder viele Dateien als ein Archiv zu speichern oder zu verteilen (.tar.gz, .tgz, .tar.bz2 etc.).

Syntax:

`tar [Optionen] -f <Archiv> [Dateien/Verzeichnisse...]`

Tar hat **Modi** (Mutual exclusive Optionen): Erstellen, Extrahieren, Listen etc. Üblich ist die kombinierte Kurzoption wie -czf oder -xvf. Wichtig: -f (file) gibt an, welches Archivfile zu benutzen ist, und kommt als *ein* Argument, meist zuletzt bei Short-Options-Kombination.

Haupt-Modi:

- **-c, --create** : Erstelle ein neues Archiv aus den **(Fortsetzung)**

tar (Tape Archiver)

Beschreibung: Erstellt oder entpackt Archive aus mehreren Dateien. tar bündelt Dateien in **einem** Archiv (Containerdatei mit Endung .tarfile-4fdk7rvx39azjretinarqb】. Oft wird tar mit Kompressionsprogrammen kombiniert (z.B. gzip oder bzip2), um komprimierte Archive wie .tar.gz zu erstellefile-4fdk7rvx39azjretinarqb】.

Syntax:

`tar <Modus-Option> [Optionen] -f <Archivdatei> [Dateien/Verzeichnisse...]`

Tar benötigt **eine** der Hauptoptionen: -c (create), -x (extract) oder -t (list). Dazu wird **-f** angegeben, um den Namen der Archivdatei zu bestimmen. Häufig werden mehrere Kurzoptionen kombiniert (z.B. -czvf für "create, gzip, verbose, file").

Wichtige Optionen:

- **-c – create:** Neues Archiv erstellefile-4fdk7rvx39azjretinarqb】 .
- **-x – extract:** Dateien aus Archiv entpacken.
- **-t – list:** Inhalt des Archivs auflisten (nicht extrahieren).
- **-f Datei** – Gibt den Archivnamen an (Pfad/Name der Tar-Datei). Diese Option **muss** direkt gefolgt von dem Dateinamen kommen (z.B. -f archive.tar).
- **-v – verbose:** Ausführliche Ausgabe (listet während der Verarbeitung die Dateinamen auf).
- **-z – Filter durch gzip:** Bei -c wird das Archiv mit gzip komprimiert (*.tar.gz); bei -x wird erwartet, dass das Archiv gz-komprimiert isfile-4fdk7rvx39azjretinarqb】 . (Äquivalent: --gzip oder mit externem gzip pipen).
- **-j – Filter durch bzip2:** Erstellt/entpackt ein bzip2-komprimiertes Archiv (*.tar.bz2). (Alternativ --bzip2.)
- **-J – Filter durch xz:** Für .tar.xz-Archive (--xz).
- **-A – concatenate:** Hängt ein oder mehrere Archive an ein bestehendes Archiv an.
- **-d – diff/compare:** Vergleicht Archiveinträge mit Dateien im Dateisystem (prüft, ob Unterschiede bestehen).
- **-C DIR – Change Directory:** Wechselt ins Verzeichnis *DIR*, bevor Dateien hinzugefügt oder extrahiert werden [freedesktop.org](#)】 . Z.B. tar -xvf archive.tar -C /tmp entpackt nach /tmp.
- **--strip-components=N** – Beim Extrahieren lässt tar N führende Pfadkomponenten weg. Z.B. entpackt /usr/local/share/doc/file als file bei --strip-components=3. Praktisch, um Archiv ohne Top-Level-Ordner auszupacken.
- **--exclude=PATTERN** – Schließt Dateien, die auf PATTERN passen, vom Archivieren aus. Z.B. --exclude="*.o".
- **--wildcards, --anchored** – steuert, wie Muster bei -t oder -x --wildcards -f archive ".*.txt" interpretiert werden (glob-Muster an/aus etc.).
- **--numeric-owner** – Beim Entpacken Besitzer/Gruppen als IDs übernehmen (statt Namen auf dem Zielsystem zuzuordnen).

Beispiele:

- Archiv erstellen: tar -cvf archive.tar Verzeichnis/ erzeugt ein unkomprimiertes Archiv archive.tar mit allen Dateien aus Verzeichnis/.

- Mit Kompression: tar -czvf etc.tar.gz /etc packt ganz /etc als gzip-komprimiertes Archiv.
- Entpacken: tar -xvzf backup.tar.gz -C /restore entpackt *backup.tar.gz* nach */restore*.
- Auflisten: tar -tvf archive.tar zeigt die Dateiliste mit Größen, Timestamp, Rechten etc.

gzip (GNU Zip Compression)

Beschreibung: Komprimiert oder dekomprimiert Dateien im **gzip-Format** (.gz). Es ersetzt beim Komprimieren standardmäßig die Originaldatei durch die komprimierte Version (und fügt .gz anfile-4fdk7rvx39azjretinarqb】). Gzip ist schneller, aber erzeugt etwas größere Dateien als bzip2/xz. Häufig mit tar (siehe oben) eingesetzt.

Syntax:

gzip [Optionen] [Datei...]

gunzip [Optionen] [Datei...]

(gunzip ist äquivalent zu gzip -d.)

Optionen:

- **-c, --stdout:** Schreibt das komprimierte Ergebnis auf die Standardausgabe, anstatt die Datei zu ersetzen. Original bleibt erhalten (kann in Pipeline genutzt werden[ionos.de](#)】).
- **-d, --decompress:** Dekomprimiert die .gz-Datei (auch via gunzip aufrufbar[ionos.de](#)】).
- **-k, --keep:** Behält Originaldatei beim Komprimieren/Entpacken (standardmäßig wird Original gelöscht).
- **-r, --recursive:** Komprimiert alle Dateien in angegebenen Verzeichnissen rekursiv.
- **-1 bis -9:** Einstellen der Kompressionsstufe (1 = schnellste/geringste Kompression, 9 = langsamste/beste). Standard ist 6.
- **-t, --test:** Prüft eine gz-Datei auf Integrität, ohne sie zu entpacken (exit-Status zeigt OK/NOK).
- **-l, --list:** Zeigt Informationen über eine komprimierte Datei: Originalgröße, komprimierte Größe, Kompressionsrate, Zeitstempel.
- **--stdout, -c (nochmal):** Wird oft mit -d kombiniert, um nach stdout zu entpacken (z.B. gzip -dc file.gz).

- **-N, --name:** Bewahrt ursprünglichen Dateinamen und Zeitstempel (in .gz-Header) oder stellt diese beim Entpacken wieder her (ist Standardverhalten bei GNU gzip).
- **--help, --version:** Hilfe bzw. Versionsinfo.

Beispiele:

- gzip bigfile komprimiert *bigfile* zu *bigfile.gz* (und löscht *bigfile*).
- gzip -9 *.txt erstellt maximal komprimierte *.txt.gz Dateien.
- gunzip data.csv.gz entpackt *data.csv.gz* zurück zu *data.csv*.
- gzip -c file.txt > file.txt.gz komprimiert nach stdout und schreibt in separate Datei (Original bleibt).
- In Pipe: tar -cvf - /folder | gzip > folder.tar.gz – tar erzeugt Archiv auf stdout, gzip komprimiert es zu *folder.tar.gz*.

xz (LZMA Compressor)

Beschreibung: Komprimiert/Dekomprimiert Dateien mit dem **xz/LZMA2-Algorithmus**. XZ erreicht hohe Kompressionsraten, aber benötigt mehr CPU und RAM. Dateiendung: *.xz*. Wird z.B. für Kernel-Quellen-Archive genutzt. Syntax und Optionen ähneln gzip.

Syntax:

`xz [Optionen] [Datei...]`

`unxz [Optionen] [Datei...]`

(`xz -d` entspricht `unxz`.)

Wichtige Optionen:

- **-z (Default):** Komprimieren (Standardmodus, kann entfallen).
- **-d, --decompress:** Entpacken.
- **-k, --keep:** Originaldatei behalten (nicht löschen).
- **-f, --force:** Vorhandene .xz-Dateien überschreiben, falls vorhanden; oder erzwingt Komprimieren auch bei scheinbar nicht sinnvoll (z.B. sehr kleine Datei).
- **-9 (auch -0 .. -9):** Kompressionsgrad (0 = none, 6 = Standard, 9 = max). XZ erlaubt auch -e für extreme Mode (noch etwas bessere Kompression, deutlich langsamere Kodierung).
- **-T N, --threads=N:** Nutzung von N Threads für Kompression (für große Dateien auf Multi-Core). Standard 1. (Für Dekomprimierung wird threads automatisch benutzt).

- **-c, --stdout:** Ergebnis auf stdout schreiben (bzw. von stdin lesen und auf stdout entpacken).
- **-l, --list:** Listet das .xz-Archiv (Inhalt, ratio, etc. – bei Single-File Komprimierung liefert original und comp size).
- **-q, -v:** quiet bzw. verbose Mode.
- **-C, --check=TYPE:** Prüfsummenverfahren wählen (Standard CRC64). Kann z.B. auf CRC32, SHA256 gestellt werden.

Beispiele:

- xz hugefile erzeugt *hugefile.xz* (Original wird gelöscht).
- xz -dk archive.tar.xz entpackt *archive.tar.xz* nach *archive.tar* und behält auch die .xz.
- tar -c /usr/bin | xz -9e > bin.tar.xz erstellt maximal-komprimiertes tar-Archiv.
- unxz data.csv.xz oder xz -d data.csv.xz entpackt Datei.
- xzcat file.txt.xz = xz -dc file.txt.xz entpackt nach stdout (ähnlich zcat).

bzip2 (Burrows–Wheeler Compressor)

Beschreibung: Komprimiert mit dem älteren bzip2-Algorithmus (.bz2). Langsamer als gzip, aber bessere Kompression; heute oft von xz abgelöst. Bedienung ähnlich gzip.

Syntax: bzip2 [Optionen] [Datei...] – bunzip2 zum Entpacken.

Optionen (Auswahl): -k (keep file), -d (decompress), -1 .. -9 (Stufe, Standard 9 bei bzip2), -c (stdout), -t (Test), -v (verbose stats).

zip (Archiv mit Kompression, ZIP-Format)

Beschreibung: Erstellt ZIP-Archive (mit Kompression) oder fügt Dateien hinzu. Anders als tar erwartet zip direkt die zu archiviernden Dateien und komprimiert sie einzeln ins Archiv. ZIP-Archive enthalten Verzeichnisstruktur + komprimierte Dateien (beliebtes Format unter Windows). unzip entpackt sie.

Syntax:

zip [Optionen] <Archiv.zip> <Dateien...>

Wichtige Optionen:

- **-r** – Rekursiv Verzeichnisse einschließen. Z.B. zip -r archiv.zip Ordner/.
- **-9** (bzw. -1 .. -9) – Kompressionsgrad (9 = beste, Standard meist 6).
- **-q** – quiet, **-v** – verbose (gibt Kompressionsraten aus).

- **-m – move:** Dateien nach dem Hinzufügen ins Archiv löschen (verschieben).
- **-j – junk paths:** Pfadnamen ignorieren, legt alle Dateien „flach“ im Archiv ab (ohne Verzeichnisstruktur).
- **-x <Pattern>** – bestimmte Dateien ausschließen (z.B. -x "*.o").
- **-F** – Versucht ein beschädigtes Archiv zu reparieren (freshen).
- **-u – update:** nur neuere Dateien (bzgl. Zeitstempel) ins Archiv aktualisieren.
- **-g – grow:** hängt Dateien an ein bestehendes ZIP an (statt neu zu erstellen).
- **-e** – erstellt ein verschlüsseltes Archiv (fordert Passwort-Eingabe, traditionelles ZipCrypto; -P <pw> unsicher ohne Prompt).

unzip Wichtige Optionen:

unzip <file.zip> entpackt im aktuellen Verzeichnis. -d <DIR> entpackt ins DIR. -l listet Inhalte. -t testet. -x schließt beim Entpacken bestimmte aus. -n überschreibt nichts, -o überschreibt ohne Nachfrage.

Paketmanagement-Befehle (Distributionen)

rpm (RPM Package Manager)

Beschreibung: Das Low-Level-Werkzeug zum Installieren, Entfernen und Abfragen von RPM-Paketen (Red Hat, CentOS, Fedora, SUSE etc.) . rpm arbeitet auf .rpm-Dateien und der lokalen Paketdatenbank. Häufig nutzt man für höhere Ebene yum oder dnf, aber rpm bleibt wichtig für direkte Paketoperationen.

Syntax (Basis):

rpm [Optionen] <Paketdatei> / rpm [Optionen] <Paketname>

Operationen (Hauptoptionen, nicht kombinierbar):

- **-i – install** ein neues RPM-Paket . Beispiel: rpm -i nginx.rpm. Erfordert Root-Rechte.
- **-U – upgrade** (installiert neu oder aktualisiert älteres Paket). Wird häufig anstelle -i benutzt, da es sowohl installiert als auch updatet.
- **-F – freshen:** wie -U, aber nur wenn schon ältere Version installiert ist (sonst kein Install).
- **-e – erase** (deinstalliert) ein installiertes Paket nach Paketname . Beispiel: rpm -e nginx.
- **-q – query:** Abfrage-Modus (ohne Änderungen) . Kann mit Zusätzen Info liefern. Beispiel: rpm -q nginx prüft, ob Paket installiert ist und welche Version.

Wichtige Optionen/Zusätze:

Install/Upgrade Optionen:

- **-h** – zeigt Hash-Fortschriffs Balken beim Installieren ("#" -Zeichen).
- **-v** – verbose, mehr Ausgabe. Oft kombiniert -Uvh oder -ivh.
- **--force** – erzwingt Installation, selbst wenn z.B. ein Paket bereits installiert ist oder Dateien konfigurieren. (Vorsicht!).
- **--nodeps** – ignoriert Abhängigkeits-Prüfung (installiert trotzdem).
- **--replacefiles** – überschreibt Dateien, die zu anderen Paketen gehören könnten (Konflikte ignorieren).
- **--noscripts** – führt keine Pre/Post-Install-Skripte im RPM aus (für spezielle Fälle).

Query Optionen (mit -q):

- **-a** – query all (listet alle installierten Pakete).
- **-i** – (mit -q) zeigt detaillierte Paketinformationen (Name, Version, Summary, Größe, Install-Datum etc. linux-praxis.de) . Z.B. rpm -qi bash.
- **-l** – listet alle vom Paket installierten Dateien. ` rpm -ql <paket> file-5y1edouysrow4zj6phjcqm` .
- **-p** – query ein *Paketfile* (nicht installiert). Z.B. rpm -qp -l paket.rpm zeigt, welche Dateien im RPM drin sind.
- **-f Dateipfad** – findet heraus, zu welchem installierten Paket eine bestimmte Datei gehört. Z.B. rpm -qf /usr/bin/ls -> coreutils-8.xx.
- **-c** – listet nur Konfigurationsdateien des Pakets (mit -ql kombinierbar: -qlc).
- **--changelog** – zeigt das Changelog des Pakets (sofern im RPM hinterlegt).
- **--provides, --requires** – zeigen welche Provides/Requires (Abhängigkeiten) ein Paket liefert bzw. benötigt. Z.B. rpm -q --requires httpd.
- **-s** – zeigt den Status der Dateien (ob sie verändert wurden seit Installation, (vom user modifiziert = **M**, missing etc.)). rpm -qs <paket>.

Erase (Uninstall) Optionen:

- **--nodeps** beim -e: ignoriert Abhängigkeitsprobleme (entfernt auch wenn andere Pakete darauf angewiesen sind - kann zu kaputten Programmen führen).

Beispiele:

- rpm -Uvh httpd-2.4.rpm – installiert Apache httpd, zeigt Hashtags und Namen.
- rpm -qa | grep kernel – listet alle installierten Pakete, filtert "kernel".

- rpm -qpi package.rpm – zeigt Infos über nicht installiertes RPM.
- rpm -e httpd – entfernt Apache-Paket.

yum (Yellowdog Updater Modified)

Beschreibung: Ein High-Level-Paketmanager für RPM-basierte Distr. (v.a. CentOS/RHEL 7 und älter). yum löst Abhängigkeiten automatisch auf und kann aus Repositories Pakete installieren, entfernen, aktualisieren . (In neueren RHEL8+ abgelöst durch dnf, aber Syntax ähnlich.)

Syntax:

yum [Optionen] <Sub-Befehl> [Paketnamen/...]

Haupt-Subkommandos:

- **install Paket** – Installiert ein Paket (aus den konfigurierten Repos oder einer RPM-Datei). Z.B. yum install httpd. Resolved dependencies automatisch und fragt um Bestätigung.
- **update [Paket]** – Aktualisiert alle installierten Pakete (wenn kein Name angegeben) oder ein bestimmtes. (yum update führt System-Update durch) .
- **remove Paket, erase** – Entfernt angegebene Pakete mit Abhängigkeiten, die dadurch verwiesen.
- **search Stichwort** – Durchsucht die Paketnamen, Beschreibungen etc. nach dem Stichwort (um herauszufinden, welches Paket etwas bietet).
- **info Paket** – Zeigt Infos zu Paket(en) (Name, Version, Summary, Größe, Repo, Abh.) ähnlich rpm -qi aber aus Repos wenn nicht installiert.
- **list [installed|available] <Pattern>** – Listet Pakete (installierte oder verfügbare). Ohne Parameter alle Pakete, optional mit Pattern filtern. Z.B. yum list available 'php*'.
- **provides Datei/Capability** – Findet heraus, welches Paket eine bestimmte Datei oder "Provide" bereitstellt (ähnlich rpm -qf/--whatprovides). Z.B. yum provides /usr/bin/zip.
- **clean <was>** – Bereinigt lokale Caches: z.B. yum clean all (löscht Metadaten- und Paketcache), oder yum clean metadata etc.
- **check-update** – Zeigt verfügbare Updates, ohne sie zu installieren (Exitcode 100 wenn Updates vorhanden, 0 wenn keine).
- **repolist** – Listet eingerichtete Repositories und ob aktiviert.
- **groupinstall / groupremove** – Installiert/entfernt ganze Paketgruppen (z.B. yum groupinstall "Development Tools").

Optionen (Auswahl):

- **-y** – Automatisch "yes" beantworten (nicht nach Bestätigung fragen). Wichtig für Skript】.
- **-q** – Weniger Ausgabe (quiet).
- **-C** – Nur aus dem Cache arbeiten, keine neuen Metadaten laden (offline).
- **--enablerepo=X, --disablerepo=Y** – Repositories gezielt ein-/ausschalten für diesen Befehl (Overrides config).
- **--nogpgcheck** – Deaktiviert GPG-Signaturprüfung der Pakete (unsicher, nur falls Repo-Key nicht installiert etc.).
- **--downloadonly** – Lädt die Pakete herunter, installiert aber nicht.
- **--security** – (bei update/info/list) berücksichtigt nur sicherheitsrelevante Updates (wenn Repo das unterstützt).

Beispiele:

- yum install nginx php php-mysql – installiert mehrere Pakete in einem Rutsch.
- yum update – bringt System auf neuesten Stand.
- yum search "pdf viewer" – sucht nach Paketen mit "pdf viewer" im Namen/Beschreibung.
- yum remove httpd – deinstalliert Apache (und abhängige Module, wenn keine anderen es benötigen).
- yum provides "*bin/rar" – findet Paket, das rar-Binary liefert.

(*Hinweis: Auf neueren Systemen ersetzt dnf den Befehl yum. dnf Syntax und Subcommands sind weitgehend gleich, aber z.B. dnf upgrade statt yum update.)*

apt (Advanced Package Tool – Debian/Ubuntu)

Beschreibung: Höher-level Frontend für Debian-Paketverwaltung (dpkg). apt kombiniert Funktionen von apt-get und apt-cache in einem Kommando. Es verwaltet .deb-Pakete: Installation, Updates aus Repositories, Paket-Suche usw】. (Ubuntu 16.04+ und Debian Stretch+ empfehlen apt für interaktive Nutzung, während apt-get skriptfreundlich bleibt.)

Syntax:

apt [Optionen] <Sub-Befehl> [Pakete...]

Wichtige Sub-Befehle:

- **update** – Aktualisiert die Paketquellen-Datenbank (lädt neue Paketlisten von den konfigurierten Repos). Immer vor Install/Upgrade ausführen.
- **upgrade** – Installiert verfügbare Updates für alle **schon installierten** Pakete (führt keine Entfernung durch). Hält Pakete zurück, wenn Abhängigkeitswechsel nötig wären. (*vgl. apt full-upgrade*).
- **full-upgrade** – Führt vollständige Distribution-Upgrades durch, d.h. aktualisiert alle Pakete und darf auch Pakete entfernen oder neue Abhängigkeiten hinzufügen, um ein konsistentes System herzustellen (entspricht apt-get dist-upgrade).
- **install <pkg>...** – Installiert angegebene Pakete (aus Repos oder .deb Dateien via Pfad). Löst Abhängigkeiten auf und fragt bei Bedarf. Akzeptiert Paket=Version oder Paket:arch.
- **remove <pkg>...** – Entfernt Pakete, lässt jedoch veränderte Konfigurationsdateien bestehen (als *.config* in /etc).
- **purge <pkg>...** – Entfernt Pakete **inklusive** Konfigurationsdateien (bereinigt vollständig).
- **search <regex>** – Durchsucht Paketnamen und Beschreibungen nach einem Begriff (regex möglich). *Beispiel:* apt search editor pdf.
- **show <pkg>** – Zeigt detaillierte Informationen zu einem Paket (Version, Abhängigkeiten, Beschreibung, Maintainer etc.), egal ob installiert oder nur verfügbar】.
- **list** – Listet Pakete. Mit --installed, --upgradeable oder --all-versions. (Ähnlich dpkg -l aber filterbar). Z.B. apt list --installed | grep nginx.
- **autoremove** – Entfernt automatisch Pakete, die als Abhängigkeiten installiert wurden und nun nicht mehr benötigt werden (Waisen).
- **depends <pkg>, rdepends <pkg>** – Zeigt Abhängigkeiten bzw. Reverse-Abhängigkeiten eines Pakets (welche Pakete benötigen es).
- **edit-sources** – Öffnet die sources.list im Editor (um Repos zu bearbeiten).

Optionen:

- **-y, --yes** – Automatisch alle Fragen mit Yes beantworten (nicht interaktiv bestätigen). Praktisch für Skripte.
- **-q** – Weniger Ausgabe (bis zu -qq für gar keine).
- **--no-install-recommends** – Installiert nicht automatisch als "empfohlen" markierte Pakete. Standard apt installiert Recommends mit.

- **--allow-remove-essential, --allow-change-held-packages** – Erlaubt riskante Änderungen (Standard verhindert z.B. Entfernen essentieller Pakete).
- **-s, --simulate** – Simulation: Zeigt an, was würde passieren, aber führt keine Änderung aus (auch Exitcode 0 immer). Gut zum Test.
- **-V, --verbose-versions** – Zeigt bei Auflistungen detailliertere Versionsinfos.
- **--help** – Hilfe-Seite, **--version** – Version von apt.

Beispiele:

- apt update && apt upgrade – Aktualisiert Paketlisten und installiert alle verfügbaren Updates (ohne Dist-Upgrade).
- apt install build-essential – Installiert eine Paketgruppe (hier C/C++ Compiler Toolchain).
- apt remove --purge openssh-server – Entfernt SSH-Server komplett mit Config.
- apt search docker – Sucht alle Pakete mit "docker" im Namen/Beschreibung.
- apt full-upgrade – Aktualisiert System auch über Distribution-Upgrades hinweg (entspricht auf Ubuntu einem Release-Upgrade, sofern Repos angepasst).

(Hinweis: apt-get und apt-cache sind Vorgänger: z.B. apt-get install, apt-get remove, apt-get dist-upgrade, apt-cache search, apt-cache show. apt vereinfacht den Alltag, aber Skripte nutzen weiterhin oft apt-get -y etc. dpkg siehe unten ist noch eine Stufe tiefer für einzelne .deb ohne Abhängigkeitsauflösung.)

dpkg (Debian Package Manager - Low-Level)

Beschreibung: dpkg ist das unterliegende Werkzeug auf Debian/Ubuntu, das .deb-Pakete entpackt, installiert, konfiguriert und entfernt. Es löst keine Abhängigkeiten – dafür sind apt/apt-get zuständig. Man verwendet dpkg meist für manuelle Offline-Installation von .deb-Dateien.

Syntax:

```
dpkg -i <Paket.deb>      # Installieren
dpkg -r <Paketname>      # Entfernen
dpkg -P <Paketname>      # Entfernen + Purge
dpkg -L <Paketname>      # Liste Dateien des installierten Pakets
dpkg -s <Paketname>      # Statusinfo (ähnlich apt show)
```

Wichtige Optionen/Parameter:

- **-i, --install Datei.deb**: Installiert das angegebene Deb-Paket. Gibt Fehler aus, falls Abhängigkeiten fehlen. (Diese müssen vorher mit apt installiert werden).
- **-r, --remove Paketname**: Entfernt ein installiertes Paket (Paketname ohne ".deb", wie in dpkg -l gelistet). Konfigdateien unter /etc bleiben bestehen.
- **-P, --purge Paketname**: Entfernt Paket inkl. seiner Konfigurationsdateien.
- **-S, --search Dateiname**: Sucht, welches installierte Paket eine bestimmte Datei besitzt (ähnlich rpm -qf). Z.B. dpkg -S /usr/bin/foo.
- **-L, --listfiles Paketname**: Listet alle Dateien, die das installierte Paket auf dem System installiert hat [file-5y1edouysrow4zj6phjcqm].
- **-l, --list Muster**: Listet Pakete, optional gefiltert nach Muster. dpkg -l ohne Filter zeigt eine Tabelle aller Pakete und deren Status (ii = installiert, rc = removed config-files left, etc.).
- **-s, --status Paketname**: Zeigt den Status und Informationen eines installierten Pakets (Version, Beschreibung, Maintainer, Abhängigkeiten...).
- **--configure Paketname**: Führt die Konfigurationsschritte (dpkg-configure) für ein installiertes aber noch nicht konfiguriertes Paket aus. (Wird z.B. nach entpacken aufgerufen).
- **-B, --auto-deconfigure**: Beim Entfernen eines Pakets, das Abhängigkeiten bereitstellt, markiert abhängige Pakete als "deconfigured".
- ****--force-**X**: Viele Force-Flags, um dpkg anzuweisen, bestimmte Fehler zu ignorieren (z.B. --force-dependencies ignoriert fehlende Abhängigkeiten, --force-confold/--force-confnew bei Config-Datei-Konflikten etc.). Vorsicht: Kann System inkonsistent machen.
- **--print-architecture**: Zeigt die Architektur, für die dpkg Pakete installiert (z.B. amd64).
- **--add-architecture <arch>**: Fügt Multiarch-Unterstützung für andere Arch hinzu (z.B. i386 auf einem amd64 System).

Beispiele:

- dpkg -i google-chrome.deb – Installiert Chrome .deb-Paket (scheitert, wenn Abhängigkeiten fehlen – in dem Fall via apt --fix-broken install auflösbar).
- dpkg -r postfix – Entfernt postfix, belässt aber ggf. geänderte /etc/postfix Dateien.
- dpkg -P postfix – Entfernt postfix vollständig inklusive Config.
- dpkg -L bash – Listet alle vom *bash*-Paket installierten Dateien.

- `dpkg -S /bin/ls` – Welches Paket stellt `/bin/ls` bereit? (Antwort: coreutils).

wget (Web GET Utility)

Beschreibung: Ein Kommandozeilen-Downloader für Dateien über HTTP, HTTPS und FTP. wget kann in Skripten genutzt werden, um Dateien aus dem Web herunterzuladen, unterstützt Fortsetzen abgebrochener Downloads, rekursives Herunterladen von Websites etc. Es funktioniert nicht-interaktiv (kein Browser, nur Download).

Syntax:

`wget [Optionen] <URL>`

Wichtige Optionen:

- **-O Datei** – Speichert die heruntergeladene URL in die angegebene *Datei* (statt den Namen aus URL zu verwenden). Z.B. `wget -O ubuntu.iso http://.../ubuntu.iso`.
- **-c, --continue** – Setzt einen abgebrochenen Download fort (sofern Server Range Requests unterstützt). Prüft, wieviel schon vorhanden ist und lädt den Rest.
- **-q** – Quiet-Modus (keine Ausgabe). **-nv** (not verbose) eine dazwischenliegende Stufe. **-v** für ausführlicher (Standard).
- **-r, --recursive** – Rekursiver Download einer Website. Erfordert oft zusammen mit **-np** (no parent), **-L** (stay on same domain), **-k** (Links anpassen für Offline). Z.B. `wget -r -np -k http://example.com/docs/`.
- **-np, --no-parent** – Geht beim rekursiven Download nicht über das Startverzeichnis hinaus.
- **-l <Tiefe>** – Legt die Rekursionstiefe fest (Default 5, **-l0** = unendlich).
- **-p, --page-requisites** – Lädt alles mit, was nötig ist, um HTML-Seite offline darzustellen (Bilder, CSS...).
- **-N, --timestamping** – Lädt eine Datei nur herunter, wenn sie neuer ist als die lokale (verwendet Last-Modified). Gut zum Aktualisieren.
- **--limit-rate=<Rate>** – Begrenzt Downloadrate, z.B. `--limit-rate=200k` (200 KB/s). Hilfreich, um Bandbreite zu drosseln.
- **-U, --user-agent=<Agent>** – Setzt einen bestimmten User-Agent-String. Standard ist "Wget/Version". Kann geändert werden, um z.B. nicht blockiert zu werden.
- **--no-check-certificate** – Ignoriert SSL-Zertifikatfehler (z.B. selbstsigniert).
- **--user=<user> --password=<pw>** – Falls erforderlich, für HTTP-Auth oder FTP-Login. (Achtung: Sichtbar in Prozessliste).

- **-i <Datei>** – Liest eine Liste von URLs aus <Datei> (jede Zeile eine URL) und lädt alle herunter.
- **-b, --background** – Startet im Hintergrund (nur auf Unix, leitet Ausgabe nach wget-log).
- **-P <Verz>** – Speichert Dateien in angegebenem Verzeichnis (statt aktuelles).

Beispiele:

- wget http://example.com/file.zip – Lädt file.zip in aktuelles Verzeichnis herunter.
- wget -c http://largefile – setzt Download fort, falls vorhanden.
- wget -r -np -N http://example.com/photos/ – Lädt Verzeichnis "photos" vollständig herunter, aktualisiert nur neue Dateien beim Wiederaufruf.
- wget -O - https://api.example.com/data | jq . – Lädt Daten von API und pipet direkt in jq zur Verarbeitung (hier -O - bedeutet Ausgabe auf stdout statt Datei).

curl (Client URL)

Beschreibung: curl ist ein umfangreiches Kommandozeilen-Tool für Datenübertragungen mit URL-basierten Protokollen (HTTP, HTTPS, FTP, etc.). Im Gegensatz zu wget liegt Fokus auf Flexibilität bei HTTP-Requests (Header setzen, verschiedene HTTP-Methoden, Formulare absenden, APIs ansprechen etc.). Es eignet sich für Web-APIs (GET/POST) und kann ebenfalls Dateien herunterladen oder hochladen.

Syntax:

curl [Optionen] <URL>

Wichtige Optionen:

- **-o Datei** – Speichert die Ausgabe in Datei. (Kleinbuchstabe o, analog wget -O).
- **-O** – Speichert unter dem **Originalnamen** aus der URL (Großes O, analog wget ohne -O). Achtung: curl -O http://server/dir/file.txt legt file.txt an.
- **-L – Location folgen:** folgt HTTP-Redirects (Status 3xx) automatisch freedesktop.org. (wget macht das default, curl nicht ohne -L).
- **-C –** Setzt abgebrochenen Download fort (wenn Server unterstützt; '-' heißt automatischen Offset ermitteln).
- **-# –** Fortschrittsleiste im CLI (oder --progress-bar). Standard curl zeigt laufende Stats Zeile, -# macht wie wget progressive Bar.
- **-s –** Silent (kein Progress oder Fehlerausgaben). **-S** kombiniert mit -s zeigt Fehler dennoch an.

- **-v / -vv** – Verbose Mode (zeigt detaillierte Request/Response, Header etc. -vv noch mehr, inkl. Low-level). Gut zum Debuggen.
- **-I** – Führt eine HEAD-Anfrage aus (nur Header vom Server holen, keine Body-Daten). Nützlich um Meta-Informationen (Content-Length, Last-Modified) abzufragen.
- **-X METHOD** – Explizite HTTP-Methode festlegen (z.B. PUT, DELETE). Standard bei Datenvorhandensein oder -d ist POST, sonst GET.
- **-G** – Erzwingt, dass bei Verwendung von -d (Daten) diese als Query-Parameter an URL angehängt werden (statt POST Body).
- **-d Daten** – Sendet die angegebenen *Daten* im HTTP-Request-Body (Standard via POST). Z.B. curl -d "user=test&pass=123" http://.../login. Implizit -X POST wenn nicht -G.
 - **--data-urlencode Daten**: URL-encodet die Daten vor Versand (z.B. wenn Daten Sonderzeichen enthalten).
 - **--data-raw**: ähnlich -d, aber ohne spezielle Form-Codierung (sendet genau so).
- **-H "Header: Wert"** – Fügt einen HTTP-Header hinzu oder überschreibt ihn. Z.B. -H "Authorization: Bearer <token>". Mehrfach nutzbar für mehrere Header.
- **-u Benutzer:Pass** – Nutzt HTTP Basic Auth mit gegebenen Credentials (wird in Header "Authorization: Basic ...Base64" codiertfreedesktop.org). Vorsicht in Shell-History. Kann auch nur -u user (dann fragt curl interaktiv Passwort ab). Funktioniert auch für FTP.
- **-F Feld@Datei** – HTTP Datei-Upload via *multipart/form-data*. Z.B. curl -F "file=@bild.png" -F "desc=Urlaubsfoto" http://.../upload.
- **-K Konfigdatei** – Liest curl-Parameter aus Datei (kann nützlich sein für komplexe Setups).
- **-b CookieDatei** – Sendet Cookies aus angegebener Datei (. Netscape cookie format oder einfach "name=value" Paare). Kann auch -b "name=value" um Cookies zu setzen.
- **-c CookieAusgabe** – Speichert Cookies, die vom Server kommen, in angegebener Datei (um sie später wieder zu verwenden).
- **--limit-rate <Speed>** – Begrenzt Übertragungsrate (z.B. --limit-rate 100k).
- **--retry N** – Anzahl der automatischen Wiederholungsversuche bei Fehler (z.B. Timeout) oder --retry-all-errors.

- **--cacert Datei / --insecure:** Spezielles SSL: anderes CA-Zertifikat nutzen bzw. bei --insecure SSL-Zertifikat überprüfen *ignorieren* (wie wget --no-check-certificate).

Beispiele:

- curl -L -o latest.zip https://github.com/user/proj/releases/latest – Lade neueste Release, folge Redirects, speichere als latest.zip.
- curl -u admin:secret -X DELETE http://server/api/item/123 – Sende DELETE-Request mit Basic Auth.
- curl -H "Accept: application/json" 'https://api.example.com/data?id=5' – Hole JSON-Daten von API mit angepasstem Accept-Header.
- curl -d '{"user":"john"}' -H "Content-Type: application/json" -X POST https://api.example.com/users – Sende JSON-Body per POST (hier ein neuer User).
- curl -F 'file=@report.pdf' -F 'desc=Monthly Report' https://upload.example.com – HTTP File-Upload (Multipart).
- curl -I https://example.com – Nur Header einer HTTP-Response anzeigen (z.B. um Last-Modified zu prüfen).

Speicher- und Dateisystem-Verwaltung

lsblk (List Block Devices)

Beschreibung: Zeigt eine Übersicht aller erkannten Blockgeräte (Festplatten, SSDs, Partitionen, LVM-Volumes etc.) in Form einer Baumstrukturfile-4fdk7rvx39azjretinarqb】 . lsblk listet wichtige Attribute wie Gerätename, Größe, Typ (disk/part), Mountpoint usw. Es hilft beim Identifizieren von Device-Namen und Partitionierungsstruktur.

Syntax:

lsblk [Optionen] [Gerät...]

Optionen:

- **-a, --all:** Zeigt auch leere bzw. unverwendete Geräte an (z.B. Geräte ohne Speicher, Loop ohne Backing). Standard lässt einige unwichtige aus.
- **-d, --nodeps:** Zeigt nur "Top-Level" Geräte (Disks) ohne Partitionen/Slaves.
- **-e <LIST>, --exclude <LIST>:** Schließt bestimmte Gerätetypen durch Major-Nummern oder Gerätekategorien aus (z.B. -e 7,11 um Loop und DVD zu ignorieren).
- **-f, --fs:** Zeigt zusätzlich Dateisystem-Informationen (Spalten FSTYPE, LABEL, UUID】 . Ähnlich blkid, aber in lsblk integriert.

- **-p, --paths:** Zeigt vollständige /dev/Pfadnamen an statt nur Gerätenamen (z.B. "/dev/sda" statt "sda").
- **-l, --list:** Ausgabe im Listenformat (eine Eintrag pro Zeile, ohne Baum-Struktur).
- **-J, --json:** Ausgabe als JSON-Struktur (maschinenlesbar).
- **-O, --output-all:** Zeigt alle verfügbaren Spalten an. Standardanzeige ist begrenzt. Mit -O sieht man z.B. auch MAJ:MIN, RM (removable), RO (read-only), MODEL, SERIAL etc.
- **-n, --noheadings:** Unterdrückt Überschrift.
- **-b, --bytes:** Größe in Bytes anzeigen (statt lesbar in KiB/MiB etc.).
- **-r, --raw:** Rohformat ohne Ausrichtung/Padding (für Skripting).

Beispielausgabe:

```
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda       8:0   0 100G 0 disk
        ↳ sda1    8:1   0 512M 0 part /boot
        ↳ sda2    8:2   0 99.5G 0 part
          ↳ vg-root 253:0  0 90G 0 lvm /
          ↳ vg-swap 253:1  0 8G 0 lvm [SWAP]
sr0      11:0  1 1024M 0 rom (cdrom)
```

Hier sieht man Disk *sda* mit 100G und zwei Partitionen *sda1*, *sda2*. *sda2* ist ein PV (Physical Volume) mit Volume Group *vg* -> daraus LVs *vg-root* und *vg-swap*, gemountet entsprechend.

lsscsi (List SCSI Devices)

Beschreibung: Listet erkannte SCSI-Geräte (inkl. SATA, SAS, USB Storage – diese werden SCSI-Subsys emuliert) auf. **lsscsi** zeigt je Gerät Host/Channel/Target/LUN, dann Gerätetyp, und eventuell Model und Seriennummer je nach Infile-
4fdk7rvx39azjretinarqb】 . Hilfreich, um z.B. alle angeschlossenen Festplatten und optischen Laufwerke anzuzeigen.

Syntax:

lsscsi [Optionen]

Optionen:

- **-v, --verbose:** Ausführlich – zeigt zusätzliche Infos (z.B. SCSI-Transport oder Zustände).

- **-t, --transport:** Zeigt SCSI-Transportinformationen (z.B. SATA, USB UAS, iSCSI etc.).
- **-L, --lunhex:** Zeigt die LUN in hexadezimaler Notation statt dezimal.
- **-H, --hosts:** Zeigt nur SCSI-Host-Adapter (Controller) an.
- **-c, --classic:** Benutzt "klassisches" Ausgabeformat (alter Stil, ohne Abkürzungen).
- **-g, --generic:** Fügt die entsprechenden sg-Device-Nodes hinzu (SCSI Generic /dev/sgX).
- **-s, --size:** Zeigt Größe (Kapazität) der Block-Geräte an, falls anwendbar (nur für Disk-type).
- **-p, --protection:** Zeigt Protection-Information (DIF/DIX).
- **-i, --scsi_id:** Zeigt die SCSI-IDs im klassischen Sinn (T10 IDs).
- **-B, --usb:** Zeigt Bus-IDs (für USB-Geräte) mit an.

Beispiel:

[0:0:0:0] disk ATA Samsung SSD 860 1B6Q /dev/sda 500GB

[2:0:0:0] cd/dvd TSSTcorp CDDVDW SH-224DB SB00 /dev/sr0

[3:0:0:0] disk WD MyBook 25DA 4004 /dev/sdb 4TB

Format: [H:C:T:L] type vendor model rev /dev/sgX -> /dev/sdX. Hier sieht man SCSI-Gerät auf Host 0 (SATA), Host 2 (DVD), Host 3 (USB-HDD).

fdisk (Partition Table Editor)

Beschreibung: Interaktives Konsolen-Tool zum Anzeigen und Bearbeiten von Partitionstabellen auf Blockgeräten (MBR, GPT). Meistens wird fdisk verwendet, um schnell die Partitionierung eines Laufwerks anzusehen (fdisk -l) oder bei MBR-Partitionierung Partitionen einzurichten. Für GPT kann man gdisk oder parted benutzen (fdisk unterstützt aber inzwischen auch GPT).

Syntax:

fdisk [Optionen] <Gerät>

Ohne Optionen startet es ein interaktives Menü für das angegebene Gerät (z.B. /dev/sda).

Optionen:

- **-l, --list:** Listet die Partitionstabellen aller verfügbaren Laufwerke auf [file-4fdk7rvx39azjretinarqb]. Oder wenn ein Gerät angegeben, nur diese. Z.B. fdisk -l /dev/sda zeigt Partitionen auf sda.
- **-s Partition:** Gibt die Größe der angegebenen Partition in Sektoren aus (veraltet, blockdev --getsz moderner).
- **-u (veraltet):** Wechselt die Ausgabeeinheit (Sektoren vs Zylinder) in alter fdisk. Heutige fdisk (util-linux) nutzt standard 512-Byte Sektoren in -l.
- **-B, -C, -H, -S:** Kann man Geometrie (Zylinder, Heads, Sectors) angeben (MBR Legacy, heute irrelevant).
- **-t Typ:** Zeigt Partitionen in bestimmtem Format (Dos, GPT, ...).

Interaktiver Modus: Kommandos wie m (Hilfe), p (Print Partitionstabelle), n (neue Partition), d (löschen), t (Typ ändern, z.B. 83 Linux, 82 Swap, in GPT GUIDs), w (write und exit), q (quit ohne save).

Beispiel (List):

Disk /dev/sda: 238.5 GiB, ...

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1	*	2048	534527	532480	260M	7	HPFS/NTFS/exFAT
/dev/sda2		534528	409600000	409065473	195.1G	83	Linux
/dev/sda3		409600001	419430400	9830400	4.7G	82	Linux swap / Solaris

Zeigt z.B. 3 Partitionen (sda1 NTFS, sda2 Linux, sda3 Swap).

parted (Partitionierungstool)

Beschreibung: Moderner Partitionierer, unterstützt GPT und MBR sowie Größe > 2TB. Kann sowohl interaktiv (im eigenen Prompt) als auch direkt per Kommandozeile genutzt werden. Im Gegensatz zu fdisk beherrscht parted auch **Resizing** von Partitionen und Filesystems (begrenzte FS-Typ Unterstützung) und kann in Skripten genutzt werden.

Syntax (non-interactive):

parted [-s] <Gerät> <Befehl> [Parameter...]

Mit -s (script) ohne interaktive Ausgabe.

Häufige parted Befehle: (Nicht alle Optionen, nur exemplarisch)

- **print** – Zeigt Partitionstabelle des Geräts. Z.B. parted /dev/sdb print.
- **mklabel <Typ>** – Erstellt neue Partitionstabelle (Typ: gpt, msdos, etc.). Achtung: löscht alle bestehenden Partitionen.

- **unit <Einheit>** – Setzt Maßeinheit (MB, GB, %, etc.) für folgende Befehle.
- **mkpart Name Typ Anfang Ende** – Erstellt Partition. Z.B.: mkpart primary ext4 1MiB 10000MiB erstellt eine ~10GB Partition vom 1MB Offset. **Typ** hier "primary"/"logical" (bei msdos) oder nur Partitions-TypName (bei GPT optional ein FS-Typ Label ext4 etc.).
- **rm <Nr>** – Löscht Partition mit Nummer.
- **resizepart <Nr> <Ende>** – Passt Größe der Partition Nr an (End-Sektor). Muss danach FS separat resized werden (z.B. mit resize2fs).
- **name <Nr> <Name>** – Setzt den Partitionsnamen (GPT Label).
- **toggle <Nr> boot** – Boot-Flag toggeln (bei MBR Partitionen).
- **align-check Typ Nr** – Überprüft Ausrichtung von Partition Nr (optisch/ Zylinder).

Parted kann auch einfache FS-Aufgaben: mkfs integration, aber meist nutzt man separate Tools (mkfs.*).

Beispiele:

- parted /dev/sdc --script mklabel gpt mkpart primary ext4 0% 50% mkpart primary linux-swap 50% 100% – Initialisiert /dev/sdc als GPT, teilt in zwei: erste Partition bis Mitte ext4, zweite Swap ab Mitte bis Ende.
- Im interaktiven parted (ohne -s) bekommt man (parted) Prompt zum Eingeben der obigen Befehle nacheinander.

mkfs (Make FileSystem)

Beschreibung: Erzeugt ein neues Dateisystem auf einer Partition oder Festplatte. mkfs ist ein Frontend, das je nach -t <FSTYP> das entsprechende Tool (mkfs.ext4, mkfs.xfs, etc.) aufruft. Häufig ruft man direkt das spezifische Formatprogramm auf.

Syntax:

mkfs -t <FSTYP> [Optionen] <Geräte/Partition>

Beispiele: mkfs -t ext4 /dev/sdb1, mkfs -t vfat /dev/sdb1.

Wichtige Optionen (allgemein):

- **-t <typ>** – Dateisystemtyp (ext4, ext3, ext2, xfs, vfat, ntfs, etc.).
- **-V** – zeigt ausführliche Versions-/Debug-Information (welches Tool aufgerufen wird).
- **-q** – quiet (weniger Ausgabe).

Jedes Dateisystem hat eigene spezifische Optionen, wenn man direkt mkfs.ext4 etc. nutzt:

ext4 (mkfs.ext4): -L <Label>, -U <UUID> (benutzerdefiniert), -m <Proz> (Reserve-Blöcke Prozent, default 5%), -O <Feature> (ein/aus bestimmter FS-Features), -E stride= (RAID optimierungen), -b blocksize, etc.

vfat (mkfs.vfat): -n <Label>, -F <32|16> (FAT type), -I (Force if not partition), etc.

xfs (mkfs.xfs): -L <Label>, -m crc=1 (z.B. CRC enable, default ON neu), etc.

swap (mkswap): -L <Label>, (swap selbst ist via mkswap erstellt).

Wichtig: Partition sollte entsprechend markiert/Dimension formatiert sein (z.B. parted type=linux-swap für swap etc.).

Beispiele:

- mkfs.ext4 /dev/sda3 – Formatiert sda3 mit ext4 Standardparametern.
- mkfs.vfat -F 32 -n "MYUSB" /dev/sdb1 – Formatiert Partition sdb1 als FAT32 mit Label "MYUSB".
- mkswap /dev/sda5 – Initialisiert Swap auf sda5 (später via swapon).
- mkfs.xfs -f -L data /dev/sdc1 – Erzwingt Format auf sdc1 mit XFS, Label "data" (XFS verlangt -f wenn Partition schon ein FS enthält).

mount (Dateisystem einhängen)

Beschreibung: Bindet ein Dateisystem (Partition, Datenträger oder virtuelles FS) in den Verzeichnisbaum ein. mount ohne Argumente listet aktuelle Einhängungen. mount <Gerät> <Verzeichnis> hängt ein Gerät an den angegebenen Ordner ein (der Ordner muss existieren und leer sein, typischerweise unter /mnt oder /media oder definierter Mountpoint in /etc/fstabfile-4fdk7rvx39azjretinarqb】 .

Syntax:

mount [Optionen] <Quelle> <Zielverzeichnis>

Wichtige Optionen:

- **-t <fstype>** – gibt den Dateisystemtyp an (ext4, vfat, nfs, cifs, etc.). Kann oft weggelassen werden, da mount es automatisch erkennt (über blkid).
- **-o <Optionen>** – übergibt eine durch Komma getrennte Liste von Mount- Optionenfile-4fdk7rvx39azjretinarqb】 :
 - **defaults** (rw,suid,dev,exec,auto,nouser,async),
 - **ro/rw** (read-only / read-write),
 - **noexec** (keine Ausführung von Programmen erlauben), **nosuid** (SUID/SGID-Bits ignorieren), **nodev** (keine Gerätedateien interpretieren),

- **sync** (synchrone IO), **async** (Standard, asynchron gepuffert),
 - **user/nouser** (Allow normal user to mount / nur root),
 - **uid=, gid=** (für FAT/NTFS, setze Besitzer der gemounteten Dateien),
 - **umask=, dmask=, fmask=** (Rechte-Masken für FAT/NTFS),
 - **iocharset=, codepage=** (für vfat),
 - **loop** (Loop-Device mount einer Datei als FS, evtl. -o loop oder auto),
 - **remount** (ein bereits gemountetes FS mit neuen Optionen versehen, z.B. mount -o remount,rw /mnt/point),
 - FS-spezifisch: z.B. **acl**, **user_xattr** (Ext4 Features), **compression**, **encrypt** (btrfs), etc.
- **-L <Label>, -U <UUID>** – anstatt Device-Node, per Label oder UUID identifizieren. mount -U 1234-ABCD /mnt/usb.
- **--bind** – führt einen Bind-Mount aus: mountet ein vorhandenes Verzeichnis an eine zweite Stelle im Verzeichnisbaum. Syntax: mount --bind /pfad/quelle /pfad/ziel. (Nützlich, um z.B. Verzeichnisse in Chroots sichtbar zu machen). Option **--rbind** rekursiv (inkl. Unterpunkte).
- **-a** – Mountet alles was in /etc/fstab als "auto" markiert ist und nicht noauto. (Ausführung typ. beim Boot).
- **-n** – führt Mount ohne Eintrag in /etc/mtab aus (mtab ist heute oft Symlink zu /proc/self/mounts). Selten benötigt.
- **-v** – verbose (gibt Meldungen aus).

Beispiele:

- mount /dev/sdb1 /media/usb -o uid=1000,gid=1000,umask=022 – hängt einen FAT32-Stick so ein, dass Benutzer mit UID/GID 1000 Besitzer ist (typisch eigener User), und Rechte 755 auf Dateien/Ordner (umask 022).
- mount -t nfs 192.168.1.10:/export/share /mnt/share – hängt eine NFS-Freigabe ein (NFS FS muss installiert sein).
- mount -o loop diskimage.iso /mnt/iso – mountet ein ISO-Image als Loop-Device unter /mnt/iso.
- mount | grep sda – sieht, ob /dev/sdaxX gemountet ist (ohne Parameter listet mount alles).

umount (Unmount FileSystem)

Beschreibung: Hängt ein gemountetes Dateisystem wieder aus. Schreibvorgänge werden abgeschlossen und der Puffer geleert. Syntax kann entweder der Mountpoint oder das Device angegeben werden】 .

Syntax:

umount [Optionen] <Mountpunkt | Gerät>

Wichtige Optionen:

- **-f, --force:** Erzwingt das Aushängen eines FS. Sollte nur im Notfall (z.B. NFS-Server down) benutzt werden, da es evtl. Datenverlust verursachen kann.
- **-l, --lazy:** "Lazy unmount". Markiert das FS als ausgeworfen, aber falls noch offene Handles bestehen, wird es erst tatsächlich ausgehängt, wenn diese geschlossen sind. (Hängt unmittelbar virtuell aus, lässt Rest im Hintergrund erledigen). Nützlich bei hängenden Prozessen.
- **-R, --recursive:** Hängt auch alle Unter-Mounts mit ab, die auf dem angegebenen Pfad liegen (rekursiv).
- **-v:** verbose.

Beispiele:

- umount /media/usb – hängt das unter /media/usb eingehängte Gerät aus.
- umount /dev/sdb1 – selbes, angesteuert über Gerät.
- umount -f -a -t nfs – erzwingt das Aushängen aller NFS-Mounts (z.B. bei Netzproblemen).

(Tipp: Wenn umount meldet "device is busy", kann man mit lsof oder fuser herausfinden, welcher Prozess noch auf das FS zugreift.)

df (Disk Free)

Beschreibung: Zeigt Übersicht des belegten und freien Speicherplatzes auf eingehängten Dateisystemen】 . df listet pro gemountetem FS die Gesamtgröße, verwendeten und freien Bytes sowie den Einhängepunkt.

Syntax:

df [Optionen] [Dateipfad...]

Ohne Argumente zeigt alle gemounteten FS. Mit Pfad ermittelt df, auf welchem FS dieser Pfad liegt, und zeigt nur dessen Werte.

Optionen:

- **-h, --human-readable:** Formatiert Größen in lesbarer Form (KiB, MiB, GiBfile-4fdk7rvx39azjretinarqb】 . Sehr gebräuchlich (Standard rohes Blockzahlen unhandlich).
- **-a, --all:** Schließt pseudo- oder 0-Size-Dateisysteme ein (wie proc, sysfs, etc.), die sonst ausgelassen werden.
- **-T, --print-type:** Zeigt Spalte mit Dateisystemtyp (ext4, tmpfs, nfs, etc.).
- **-i, --inodes:** Zeigt Inode-Auslastung statt Blockgrößen (frei/belegt Inodes pro FS). Nützlich, wenn "no space" durch Inode-Erschöpfung.
- **-t <typ>, -x <typ>:** Nur bestimmte FS-Typen anzeigen bzw. ausschließen. Z.B. df -x tmpfs -x devtmpfs um echte Disks anzuzeigen.
- **-B <Größe>:** Ausgabe in Blöcken von angegebener Größe. Z.B. -B MB (MByte) oder -B 1M.
- **--output[=FELDER]:** Benutzerdefinierte Spaltenausgabe. Z.B. df --output=source,fstype,size,used,avail,pcent,target.

Beispielausgabe (df -h):

```
Filesystem  Size Used Avail Use% Mounted on
/dev/sda2    50G  30G  18G  63% /
udev        3.9G   0  3.9G  0% /dev
tmpfs       798M 1.2M 797M  1% /run
/dev/sda1    300M  50M 250M 17% /boot
```

du (Disk Usage)

Beschreibung: Summiert und zeigt die Größe von Verzeichnissen und Dateien an. du kann rekursiv durch Verzeichnisse gehen und den Platzverbrauch berechnenfile-4fdk7rvx39azjretinarqb】 . Standardmäßig zeigt es für jedes Unterverzeichnis den Wert. Oft nutzt man du -sh <dir> um eine schnelle Gesamtgröße zu bekommen.

Syntax:

du [Optionen] [Pfad...]

Ohne Pfad = aktuelles Verzeichnis.

Optionen:

- **-h, --human-readable:** Größen in menschenlesbare Einheiten (K,M,G) ausgebene-4fdk7rvx39azjretinarqb】 .

- **-s, --summarize:** Nur Summen pro angegebenem Argument ausgeben, *nicht* rekursiv für Unterordnerfile-4fdk7rvx39azjretinarqb】 . (Gibt also Gesamtgröße des Pfades).
- **-A, --apparent-size:** Rechnet mit logischen Dateigrößen statt belegtem Blockplatz. (Z.B. sehr sparsame Sparse Files würden sonst weniger anzeigen; --apparent-size zeigt nominelle Größe).
- **-d N, --max-depth=N:** Rekursionstiefe begrenzen. Z.B. du -h --max-depth=1 /var gibt Größe von /var und dessen Unterordner erste Ebene. (--max-depth=0 wäre wie -s).
- **-c, --total:** Am Ende eine Gesamtzeile "total" ausgeben (Summe aller ausgegebenen Einträge). Bei -s von mehreren Pfaden praktisch.
- **-x, --one-file-system:** Geht nicht in andere eingehängte Dateisysteme über. (Beschränkt auf Partition/FS der Startpfade).
- **-L, --dereference:** Folgt symbolischen Links (sonst ignoriert du Symlinks, rechnet sie als 0 Bytes). Vorsicht: Kann zu doppeltem Zählen führen, wenn Links in Kreis oder auf Oberverzeichnis zeigen.
- **--exclude=<PATTERN>:** Schließt Pfade aus, die zum Pattern passen. Z.B. du -sh -exclude="*.git" oder mehrere --exclude.

Beispiele:

- du -sh /home/* – Zeigt Summe der Größen aller Benuterverzeichnisse unter /home (jedes in human-readable).
- du -a /etc | grep -E "mysql.*\..cnf\$" – Listet die Größe jeder einzelnen Datei unter /etc, filtert jene, deren Name auf mysql*.cnf endet. (Option -a listet auch Dateien und nicht nur Verzeichnisse).
- du -d2 /var – Größe von /var, dessen Subdirs und Sub-Subdirs (Tiefe 2).

e2label (ext2/3/4 Volume Label)

Beschreibung: Zeigt oder ändert das Label (Namen) eines ext2/ext3/ext4 Dateisystems. Das *Label* ist eine im Superblock gespeicherte Zeichenkette, die man z.B. in fstab zum Mounten verwenden kann (LABEL=<name>).

Syntax:

e2label <Gerät> [NeuesLabel]

Ohne zweites Argument zeigt es das aktuelle Label des ext-Volumes an. Mit Label schreibt es dieses in den FS.

Beispiel:

- e2label /dev/sda3 -> Ausgabe z.B. UbuntuRoot.
- e2label /dev/sda3 Archive -> setzt das Label von sda3 auf "Archive".

(Für andere FS gibt es analoge Tools: XFS nutzt xfs_admin -L, FAT/NTFS mlabel oder dosfslabel/ntfslabel etc.)

resize2fs (ext2/3/4 Dateisystem verkleinern/vergrößern)

Beschreibung: Passt die Größe eines ext2/3/4-Dateisystems an. Wird verwendet nach dem Ändern der Partitionsgröße (mit fdisk/parted). resize2fs kann im online-Modus vergrößern (wenn gemountet, ext3/4), fürs verkleinern muss das Volume ausgehängt und ein Filesystem-Check gemacht werden.

Syntax:

resize2fs [Optionen] <Gerät> [NeueGröße]

NeueGröße kann in K, M, G oder als Anzahl 4K-Blöcke angegeben werden. Lässt man sie weg, wird auf Maximum (Part.-Ende) vergrößert.

Optionen:

- **-p** – progress anzeigen (Fortschritt).
- **-f** – zwingen, auch wenn FS als nicht sauber markiert (normal will fsck vorher laufen bei Verkleinern).
- **-F** – flush caches vorher (sorgt für up-to-date Infos; Standard normal).
- **-M – minimal**: verkleinert das FS auf die kleinstmögliche Größe (alle Daten gerade passend). Äquivalent zu resize2fs <dev> -M (dauert Berechnung).

Beispiele:

- Nach Vergrößern der Partition: resize2fs /dev/sda2 – erkennt neue Partitionsgröße und wächst das ext4 FS auf vollen Platz.
- Verkleinern: umount /dev/sda2; e2fsck -f /dev/sda2; resize2fs /dev/sda2 20G – erst aushängen, fsck erzwingen, dann auf 20 GiB verkleinern. Anschließend Partition selbst verkleinern mit parted.

tune2fs (Tune ext2/3/4 Filesystem Parameters)

Beschreibung: Ändert diverse Parameter eines bestehenden ext-Dateisystems: Reserveblöcke, Check-Intervalle, Label, Journal einschalten, usw.

Syntax:

tune2fs [Optionen] <Gerät>

Häufige Optionen:

- **-l** – Listet alle Superblock-Parameter (Filesystem-Info) auf freedesktop.org . Zeigt z.B. Inode-Größe, Features, Letzter Mount, Nächster geplanter Check, etc. (Read-only Operation).
- **-c <MaxMounts>** – Setzt die maximale Anzahl Mount-Vorgänge bis zur erzwungenen fsck-Prüfung. Z.B. -c 0 = deaktiviert mount-count-basierten Check.
- **-i <Interval>** – Setzt das Zeit-Intervall für Zwangs-FSCK. z.B. -i 30d = alle 30 Tage. -i 0 = kein zeitbasierter Check.
- **-r <AnzRes>** – Setzt Anzahl der reservierten Blöcke (für root). Standard 5% bei ext4 auf <1TB. Kann man z.B. auf 1% reduzieren auf großen Volumes, z.B. tune2fs -r 0 /dev/sdb1 um Reserven komplett abzuschalten (nicht empfohlen auf /). Alternativ -m <Prozent> (z.B. -m 1).
- **-L <Label>** – Ändert Label (wie e2label).
- **-U <UUID>** – Ändert die UUID (zufällig oder angegeben). (Vorsicht: fstab etc. müssen angepasst).
- **-J** – Journal-Optionen: z.B. -O has_journal um Journal hinzuzufügen (macht aus ext2 -> ext3), oder -O ^has_journal Journal entfernen (ext3->ext2), oder -j Kurzoption (Standardwerte Journal hinzufügen).
- **-O <FeatListe>** – Aktiviert (feature) oder deaktiviert (^feature) Filesystem-Features. Z.B. -O casefold (für ext4 ab 5.2: Case-Insensitive-Folder Support) oder ^huge_file (Feature abschalten – Achtung, Daten verlieren, nur wenn FS leer). Nur bestimmte Features togglebar, meist erfordert offline und vollständiges Backup.
- ****-E <extendopt>**** – Erweiterte Einstellungen: z.B. stripe=<StripeSize>(RAID stripe in Blöcken),lazy_itable_init=1` (schnell format, Inode-Table lazy Clearing).
- **-M <Verzeichnis>** – Markiert FS als gemountet an <Verzeichnis> (setzt den Mountpoint in Superblock – meist rein informativ).

dumpe2fs (Dump ext2/3/4 Superblock)

Beschreibung: Gibt ausführliche Informationen über ein ext2/3/4-Dateisystem aus – Inhalt des Superblocks und der Block Group Deskriptoren . Zeigt ähnlich tune2fs -l Informationen, plus Aufteilung der Blöcke in Gruppen, freie Blöcke/Inodes pro Gruppe etc. Hilfreich für Diagnose (z.B. Backup-Superblock Positionen).

Syntax:

dumpe2fs [Optionen] <Gerät>

Optionen:

- **-h** – Zeigt nur den Inhalt des Superblocks und der Gruppen-Deskriptoren (ohne die Liste aller freien Blöcke/Inodes). Das ist die gängigste Verwendung, da ohne -

h die Ausgabe sehr lang sein kann (listet freie Blöcke/Inodes aller Groupswiki.ubuntuusers.de) .

- **-x** – Zeigt auch die detaillierten Gruppen-Informationen (wie freie Blöcke/Inodes-Bitmap). Standard ohne -h tut das sowieso.
- **-b** – Listet die Nummern der freien Blöcke. (Sehr viel Output auf großem FS.)
- **-i** – Listet die freien Inode-Nummern.

Beispiel (gekürzt dumpe2fs -h /dev/sda2):

Filesystem volume name: <none>

Last mounted on: /

Filesystem UUID: 4e8fea21-...-b6c7

Filesystem magic number: 0xEF53

Filesystem revision #: 1 (dynamic)

Filesystem OS type: Linux

Block size: 4096

Block count: 25600000

Reserved block count: 1280000

Free blocks: 15034567

First Block: 0

...

Journal inode: 8

Journal backup: inode blocks

...

Zeigt Label, UUID, Blocksize, Anzahl Blöcke, Reserve, Anzahl freie etc., und welche Features aktiviert sind.

cryptsetup (LUKS Disk Encryption Setup)

Beschreibung: Tool zur Einrichtung von **verschlüsselten Datenträgern** mit LUKS (Linux Unified Key Setup) sowie zum Öffnen/Schließen solcher Verschlüsselungs-Volume . cryptsetup kann ein Gerät mit LUKS initialisieren, Passphrase setzen, und im Betrieb ein verschlüsseltes Volume als Device-Mapper-Gerät bereitstellen, das dann gemountet werden kann.

Typische Aktionen (Subcommands):

- **luksFormat** – Initialisiert ein Blockgerät mit LUKS-Verschlüsselungsheader und fordert eine Passphrase zum Einrichten an. Beispiel: cryptsetup luksFormat /dev/sdb1. Achtung: Löscht alle vorhandenen Daten auf dem Gerät.
- **open** oder **luksOpen** – Öffnet ein verschlüsseltes Volume (fragt Passphrase). Syntax: cryptsetup luksOpen /dev/sdb1 geheim_vol. Dies erstellt ein Device /dev/mapper/geheim_vol, das nun entschlüsselt agiert und z.B. mit mkfs formatiert oder gemountet werden kann】 .
- **close** – Schließt ein vorher geöffnetes LUKS-Device: cryptsetup close geheim_vol (dann ist /dev/mapper/geheim_vol weg).
- **luksAddKey** – Fügt einen weiteren Schlüssel (Passphrase) hinzu. LUKS unterstützt mehrere Key-Slots. cryptsetup luksAddKey /dev/sdb1 /pfad/zur/schlüsseldatei oder interaktiv eine neue Passphrase eingeben (nachdem man eine gültige alte eingibt).
- **luksRemoveKey** – Entfernt einen Schlüssel (kann durch Angabe der zu entfernenden Passphrase oder Slot-Nummer erfolgen).
- **luksChangeKey** – Ändert eine bestehende Passphrase (in einem Slot) durch eine neue (kombi aus Remove+Add effizienter).
- **luksHeaderBackup** / **luksHeaderRestore** – Backup bzw. Wiederherstellen des LUKS-Headers (empfohlen, falls Header korrupt -> Daten verloren).
- **status** – Zeigt Informationen über ein offenes Device-Mapper Volume (Name, verschlüsseltes Dev, Cipher, Keylen, etc.): cryptsetup status geheim_vol.

Wichtige Optionen:

- **-y** (im Format-Befehl): fragt neues Passwort zweimal zur Verifizierung ab.
- **-d <Datei>**: verwendet eine Datei als Schlüsseldaten (anstatt interaktiv Passwort). -d - kann genutzt werden, um vom stdin zu lesen.
- **--key-size <bits>**: Schlüsselgröße (in Bits) für die Verschlüsselung (z.B. 256).
- **--cipher <name>**: Zu nutzender Verschlüsselungsalgorimus und Mode. Standard z.B. aes-xts-plain64 bei LUKS2. Kann z.B. aes-cbc-essiv:sha256 oder anderes angegeben werden.
- **--hash <hash>**: Hash-Algorithmus für Passphrase-Derivation (PBKDF). Standard modern ist z.B. Argon2id oder früher sha256.
- **--iter-time <ms>**: Wie lange (Millisek) auf PBKDF verwendet werden soll (Bestimmt Komplexität).
- **--type luks1/luks2**: LUKS Version festlegen (aktuell Standard LUKS2, kann optional LUKS1 für Kompatibilität gewählt werden).

- **-q:** Quiet (unterdrückt Ausgabe, z.B. Warnung bei Format).

Beispiel Workflow:

1. **Formatieren:** cryptsetup -y luksFormat /dev/sda5 (legt LUKS auf Partition sda5 an, fragt 2x Passwort).
2. **Öffnen:** cryptsetup luksOpen /dev/sda5 geheim (Passwort eintippen, erstellt /dev/mapper/geheim).
3. **Format FS:** mkfs.ext4 /dev/mapper/geheim.
4. **Mount:** mount /dev/mapper/geheim /mnt/secure. Jetzt Daten drauf speichern.
5. **Aushängen+Schließen:** umount /mnt/secure && cryptsetup close geheim.

(LUKS ermöglicht bis zu 8 Passphrases; man kann z.B. mit luksAddKey einen zweiten hinzugeben, sodass entweder Passphrase A oder B das Volume öffnet. Das Gerät /dev/sda5 kann in /etc/crypttab und /etc/fstab eingetragen werden für Boot-Time auto-open via keyfile oder prompt.)

shred (Securely wipe file/device)

Beschreibung: Überschreibt eine Datei oder ein ganzes Gerät mehrfach mit zufälligen Daten, um ein Wiederherstellen gelöschter Inhalte zu erschweren. shred ist nützlich zum sicheren Löschen auf magnetischen HDDs. Bei SSDs mit Wear-Leveling ist Effekt begrenzt – dort lieber ATA Secure Erase oder physical destruction.

Syntax:

shred [Optionen] <Datei1> [Datei2...]

Optionen:

- **-n N** – Anzahl der *zusätzlichen* Überschreib-Pässe (Standard 3). D.h. 3 bedeutet insgesamt 3 mal random + final pass (siehe -z).
- **-z, --zero** – Überschreibt am Ende noch einmal mit Null-Bytes, um das Muster zu verwischen (anstatt dass zuletzt zufälliges Rauschen bleibt). Standard: aktiv (bei neueren shred ist -z default an).
- **-f, --force** – Ändert Schreibschutz falls nötig (chmod u+w), um schreiben zu können.
- **-u, --remove[=how]** – Nach dem Überschreiben die Datei löschen (unlink). Optional how kann unlink (Standard) oder wipe oder wipesync sein: *wipe* überschreibt erst mit Null dann löscht, *wipesync* macht zusätzlich Sync. Standard -u bereits tut wipesync.
- **-v, --verbose** – Auskunft über Fortschritt.

- **-x, -X, --exact** – Überschreibt Datei genau Byte-für-Byte anstelle Dateigröße auf nächste Blockgröße aufzurunden (Standard shred nutzt ganze Blocks, kann über Dateiende hinaus schreiben falls Dateigröße nicht mod Blocksize). *Exact* vermeidet das, aber hinterlässt evtl. un-geschredderte Slack am Ende.
- **-s N, --size=N** – Behandelt die Datei als N Byte lang (statt aktuelle Länge). Um auch ungenutzten Platz (z.B. Slack Space/EOF beyond) im Inode zu überschreiben, kann man etwas größer angeben.

Beispiele:

- `shred -u file.txt` – Überschreibt *file.txt* viermal (3 rand + 1 zero) und löscht sie anschließend.
- `shred /dev/sdb` – Überschreibt komplett `/dev/sdb` mit 3 Runden Zufallsdaten (zerstört alle Daten/Partitionen darauf).
- `shred -n 0 -z -u secret.dat` – Einmal mit Nullen überschreiben und entfernen (hier wollte man evtl. nur einfaches Überschreiben und löschen).

Warnung: Shred wirkt nicht zuverlässig auf: **logischen Volumes** in Files (z.B. Datei in FS kann redundante Kopien (Journal, Snapshots) haben), **RAID, SSD**,

Netzwerkdateisysteme, Dateisysteme mit Copy-on-Write (btrfs, ZFS) – dort können Daten an mehreren Orten stehen. In solchen Fällen besser Vollverschlüsselung verwenden oder das gesamte physische Medium überschreiben.

iostat (IO Statistics)

Beschreibung: Zeigt Statistik zur Systemauslastung von CPU und Input/Output für Blockgeräte (Datenträger). iostat gehört zum Paket sysstat. Es hilft zu sehen, wie stark Platten ausgelastet sind (%util) und Durchsatz (MB/s), IOPS etc., sowie generelle CPU-Idle/Wait Zeite .

Syntax:

`iostat [Optionen] [Intervall [Durchläufe]]`

Ohne Argument gibt einen Durchschnitt seit Boot aus. Mit Intervall (Sekunden) aktualisiert regelmäßig.

Wichtige Optionen:

- **-c** – Zeigt nur CPU-Statistiken.
- **-d** – Zeigt nur Disk (Geräte) Statistiken.
- **-k, -m** – Ausgabe in KiB/s oder MiB/s (anstatt Blöcke/s). (Aktuelle sysstat erlaubt -block-size).

- **-p [Gerät]** – Detaillierte Partitions-Statistiken einbeziehen (normal zeigt iostat nur Whole-Device). Z.B. -p sda zeigt sda und sda1,sda2,...
- **-x** – Ausgabe in erweiterter Form (Pro Gerät: zusätzlich Auslastungsprozentsatz, Average Request Size, Await Times etc.) . Sehr nützlich, da Standard ohne -x recht knapp.
- **-t** – Zeigt Timestamp vor jeder Ausgabe.
- **-y** – Unterdrückt erste Ausgabe (since boot), beginnt erst mit Werten ab erstem Intervall.

Interpretation (bei iostat -x):

Spalten pro Device: rrqm/s wrqm/s (Merge von I/O, veraltet bei modernen Kernel meist 0), r/s w/s (reads/writes per sec), rkB/s wkB/s (Throughput), avgrq-sz (Durchschnittl. Requestgröße in Sektoren oder KB), avgqu-sz (durchschn. Queue length), await (Durchschnittliche Wartezeit pro IO ms), svctm (durchschn. Servicezeit pro IO ms – oft nicht mehr zuverlässig), %util (Prozentuale Zeit, die das Gerät aktiv war innerhalb der Messperiode – 100% bedeutet permanent beschäftigt).

Beispiel:

```
Device r/s w/s rkB/s wkB/s avgrq-sz avgqu-sz await svctm %util
sda 12.5 30.2 500.0 250.0 40.0 0.5 10.0 2.00 85.0
```

Hier sda ~85% ausgelastet, wartet ~10ms per IO, etwa 42.7 IO/s (r+s) mit ~750kB/s.

ioping (I/O Ping)

Beschreibung: Ermittelt Latenz (und ggf. Durchsatz) von Storage in Echtzeit ähnlich dem Prinzip von ping im Netzwerk. Es führt schnelle sequenzielle oder wahlfreie Lesezugriffe durch und misst deren Zeit. Praktisch um Latenz auf Disks oder bestimmten Verzeichnissen (Dateisystem) festzustellen.

Syntax:

```
ioping [Optionen] <Ziel>
```

Ziel kann ein Pfad (Datei oder Verzeichnis) sein für Filesystem IO, oder z.B. ein Blockgerät (mit -D für Direkten Device Zugriff) oder sogar ein bestimmter I/O-Engine (IO depth).

Wichtige Optionen:

- **-c <count>** – Anzahl der Ping-Requests (Standard unendlich, bis abgebrochen).
- **-i <interval>** – Wartezeit zwischen Pings (Sekunden, default 1s). Kann in ms angeben z.B. -i 0.1.

- **-s <bytes>** – Größe der zu lesenden Daten pro Ping (Standard 4KiB).
- **-R** – Ausgabe im Rohformat (Maschinenlesbar).
- **-D** – Direct-Device Mode: Arbeitet auf Blockgeräte direkt (bypasses FS-Cache).
- **-A** – Ausgabe als Aggregatstatistik nur (kein laufendes Ping, nur Summary).
- **-S** – Sync/Sequenziell: testet sequenzielles Lesen statt wahlfrei. (Standard ioping macht random position reads).
- **-W** – führt Schreib-Tests durch statt Lesen (Achtung kann Daten überschreiben zerstören, am besten auf extra Gerät/Datei).
- **-P <N>** – Parallel Requests (Thread count).
- **-q** – keine laufende Ausgabe, nur Summary am Ende.

Beispiele:

- ioping . – Sendet kontinuierlich 4K-Leseanfragen ins aktuelle Verzeichnis (random in FS), zeigt Latenzen, bis Ctrl+C.

4096 bytes from . (ext4 /dev/sda2): request=1 time=0.47 ms

4096 bytes from . (ext4 /dev/sda2): request=2 time=0.44 ms

... (zeigt typisch ~0.5ms auf SSD, ~10-15ms auf HDD) ...

Nach Abbruch gibt es eine **Summary** mit min/avg/max/mdev (Jitter).

- ioping -c 10 /mnt/backup – Misst 10 Zugriffe im FS /mnt/backup (z.B. externer USB), dann stop.
- ioping -R /dev/sdb – Führt 1 Sekunde lang Messungen auf /dev/sdb durch und gibt am Ende aggregated IOPS und Bandbreite.
- ioping -A -S -c 50 /file.iso – Liest sequenziell 50 Anfragen durch große Datei und gibt am Ende Durchschnittszeit.

(Tool muss ggf. installiert werden, gehört nicht standard zu allen Distr.)

Geräte, Prozesse, Speicher und Kernel

hwinfo (Hardware Info)

Beschreibung: Liefert detaillierte Informationen zur Hardware des Systems. hwinfo probiert, alle verfügbaren Hardware-Komponenten aufzulisten: CPU, RAM, Festplatten, USB, PCI-Geräte, Netzwerk, Monitor, etc., inklusive ihrer Eigenschaften und Treiber. Oft auf openSUSE installiert, aber auch auf anderen Distros verfügbar.

Syntax:

`hwinfo [Optionen] [Hardwareklasse]`

Optionen / Klassen:

- Ohne Angabe gibt `hwinfo` *alles* aus (sehr lang). Besser, man filtert nach Klasse: z.B. `hwinfo --cpu`, `hwinfo --disk`, `--gfxcard`, `--netcard`, `--printer`, `--usb`, `--bios`, `--memory` etc.
- **--short:** Kurzfassung ausgeben (ein Eintrag pro Gerät in einer Zeile).
- **--brief:** Noch knapper (ähnlich short).
- **--summary:** Summaries (z.B. nur Zahl der Disks, CPUs etc.).
- **--verbose <level>:** Detaillierungsgrad (0-9). Default 1 (schon sehr ausführlich).
- **--debug N:** Setzt Debug-Level (für Entwickler, welche Module `hwinfo` benutzt).
- **--hwitem <ID>:** Sucht nach Hardware mit bestimmter ID (z.B. PCI-ID).
- **--save-config <datei> / --load-config <datei>:** Speichert erkannte Hardware in Datei zum späteren Vergleich.

Beispiel: `hwinfo --cpu` könnte ausgeben:

01: None 00.0: 10103 CPU

[Created at cpu.462]

Unique ID: rdCR.jX8Un+o2ZT1

Hardware Class: cpu

Arch: X86-64

Vendor: "GenuineIntel"

Model: 6.158.10 "Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz"

... (viele weitere Details Frequenzen, Cache etc.) ...

`hwinfo --short` summarisiert z.B. Hauptgeräte und Partitionen.

dmidecode (DMI Table Decode)

Beschreibung: Gibt Informationen aus dem BIOS/UEFI-DMI (SMBIOS) heraus, wie Hersteller, Produktnamen, Seriennummern von System/Mainboard, BIOS-Version, verbauter Speicher, Anschlüsse etc. . Hilfreich für Hardware-Inventarisierung (Seriennummer, RAM-Module, OEM strings).

Syntax:

`dmidecode [Optionen]`

Optionen:

- **-t <string|Nummer>, --type**: Nur bestimmte DMI-Typen ausgeben. Typ kann Name oder Zahl sein. Z.B.:
 - bios (DMI type 0),
 - system (1),
 - baseboard (2),
 - processor (4),
 - memory (Memory Device, 17),
 - cache (7),
 - connector (8), slot (9), etc.
Man kann -t memory um nur RAM-Infos zu sehen (Größe, Typ, Hersteller, Seriennummer der DIMMs) .
- **-s <keyword>, --string**: Einzelne SMBIOS-Einträge gezielt ausgeben (z.B. bios-vendor, bios-version, system-manufacturer, system-serial-number, baseboard-asset-tag, processor-frequency etc.).
- **-q**: Quiet (unterdrückt weniger wichtige DMI-Eintrags-Labels wo "Not Provided").
- **-u**: Raw dump (hex) der DMI-Daten.
- **--dump-bin file, --from-dump file**: Dump SMBIOS in Datei bzw. lese aus Dump (z.B. offline Analyse).

Beispiel: dmidecode -t system könnte zeigen:

System Information

Manufacturer: Dell Inc.

Product Name: Latitude 5490

Version: None

Serial Number: ABCD123

UUID: 4c4c4544-0034-5810-8055-b6c04f513332

Wake-up Type: Power Switch

SKU Number: 07E6

Family: Latitude

dmidecode -t memory zeigt pro RAM-Riegel z.B. Größe, Form Factor (SODIMM), Type (DDR4), Speed, Manufacturer, Serial, Part Number.

lspci (List PCI Devices)

Beschreibung: Zeigt alle PCI/PCIe-Geräte im System (angeschlossene Hardware am PCI-Bus: z.B. Netzwerkkarten, Sound, USB-Controller, Grafikkarten, SATA-Controller etc.). Es listet Geräte-ID und -Namen anhand einer Datenbank (/usr/share/misc/pci.ids).

Syntax:

`lspci [Optionen]`

Optionen:

- **-v, -vv, -vvv:** Verbose-Ausgabe in ein oder mehreren Stufen (mehr Register-Informationen). -vvv kann *sehr* ausführlich sein (inkl. Capabilities, Konfigspace).
- **-k:** Zeigt Kernel-Treiber und Module, die für jedes Gerät verwendet werden (welcher Treiber zugeordnet ist).
- **-t:** Zeigt die hierarchische Baumstruktur der PCI-Bus-Topologie (wer hängt an welchem Bus/Bridge).
- **-nn:** Zeigt neben dem Gerätenamen auch Vendor:Device ID in eckigen Klammern. -n nur numeric IDs ohne Namen. -nn= beide. Z.B. Ethernet Controller [8086:1533] etc.
- **-d [<vendor>]:[<device>]:** Filtert nach bestimmten PCI IDs. `lspci -d 8086`: zeigt nur Intel-Geräte, `-d :1533` nur Geräte mit Device 1533.
- **-s [<domain>]::<bus>:<slot>[.<func>]:** Filtert nach Bus-Adresse (wie in `lspci -t` oder `/sys/bus/pci`). Z.B. `-s 00:1f.2` zeigt nur dieses Gerät.
- **-m:** Maschinenlesbar (gibt in einer CSV-artigen Formatzeile pro Gerät: Domain, Bus Slot, Klass, Vendor, Device, SVendor, SDevice).
- **-xxx:** Dump des gesamten PCI-Konfigurationsspeichers (64 Bytes beyond standard) – Vorsicht sensibel. -xx nur ersten 256 Bytes (std config).
- **-H1:** Nutzen von `/proc/bus/pci` statt direct (historisch, meist nicht nötig).
- **-Q:** Zeigt das Querier-Modell (Kernel vs direct vs `/proc`).

Beispiel Ausgabe (gekürzt):

00:00.0 Host bridge: Intel Corporation Device 5914

00:02.0 VGA compatible controller: Intel Corporation UHD Graphics 620

00:14.0 USB controller: Intel Corporation Sunrise Point-LP USB 3.0 xHCI Controller (rev 21)

00:16.0 Communication controller: Intel Corporation Management Engine Interface (rev 21)

...

03:00.0 Network controller: Intel Corporation Dual Band Wireless-AC 8265 (rev 78)

Mit -nn würde man z.B. sehen 00:02.0 VGA controller [0300]: Intel Corporation UHD Graphics 620 [8086:5917]. Mit -k ergänzt: Kernel driver in use: i915.

lsusb (List USB Devices)

Beschreibung: Listet USB-Geräte auf, die am System angeschlossen sind (inkl. interne Hubs etc.). Zeigt Vendor und Produktnamen falls bekannt (Datenbank /usr/share/misc/usb.ids) und ID.

Syntax:

lsusb [Optionen]

Optionen:

- **-v:** Verbose, zeigt detaillierte Descriptor-Infos pro Gerät (ähnlich lsusb -vv noch detaillierter). Das kann sehr lang sein.
- **-t:** Baum-Darstellung der USB-Geräte-Hierarchie (welches Device hängt an welchem Port/Hub) – ähnlich lspci -t.
- **-s <bus>:<dev>:** Zeigt nur spezifisches Gerät (per Bus und Device Nummer). Diese Nummern sieht man in lsusb Standardausgabe vorn.
- **-d vid:pid:** Filtert nach VendorID:ProductID (Hex).
- **-D <device-file>:** Liest Infos aus einer Geräte-Datei (z.B. /dev/bus/usb/001/007) – erfordert Root.
- **-w:** Ausgabe als mit usbdump kompatibler Text (Wire format).
- **-v -v (or -vv):** liefert noch detailliertere Dump (inkl. Hexdump der HID Deskriptoren etc.).

Beispiel Ausgabe:

Bus 002 Device 003: ID 0bda:0316 Realtek Semiconductor Corp. Card Reader

Bus 002 Device 002: ID 8087:0aaa Intel Corp.

Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub

Bus 001 Device 004: ID 0c45:671f Microdia Integrated Webcam

Bus 001 Device 003: ID 8087:07dc Intel Corp. Bluetooth wireless adapter

Bus 001 Device 002: ID 056e:4008 Elecom Co., Ltd

Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

Hier z.B. Card Reader, Webcam, Bluetooth, eine Maus (056e:4008 Elecom), plus Root Hubs.

Mit -t etwa:

/: Bus 02.Port 1: Dev 1, Class=root_hub, ... (bus 2, 3.0)

|__ Port 1: Dev 2, Intel USB 3.0 Hub

 |__ Port 2: Dev 3, Realtek Card Reader

...

Zeigt Topologie (Hub etc.).

lscpu (List CPU Info)

Beschreibung: Zeigt Informationen über die CPU(s) des Systems – Anzahl Kerne, Threads, Sockets, Modellname, Frequenz, Virtualisierungsfunktionen uswfile-4fdk7rvx39azjretinarqb】 .

Syntax:

`lscpu [Optionen]`

Optionen:

- **-e, --extended [=list]:** Ausgabe als Tabelle mit Spalten (erweiterte Auflistung jeder logischen CPU mit Eigenschaften). Optional kann man angeben, welche Felder (z.B. CPU, Node, Socket, Core, Online, ...).
- **-p, --parse [=list]:** Ähnlich --extended aber kommagetrennt (parseable). Standardfelder: CPU,Core,Socket,Node.
- **-x, --hex:** Zeigt hex-Werte wo sinnvoll (z.B. CPU bitmasken).
- **-c, --offline / --online:** Zeigt nur offline oder nur online CPUs (bei Hotplug-Systemen).
- **-y, --physical:** Zeigt nur physische IDs (keine virtuellen IDs).
- **-J, --json:** JSON-Ausgabe (maschinenlesbar).
- **-B, --byte:** Einheit Bytes statt kB für Cache-Größen.
- **-C, --cache:** Zeigt Details zu CPU-Caches (Anzahl, Größen).

Beispiel (gekürzt lscpu Standard):

Architecture: x86_64

CPU op-mode(s): 32-bit, 64-bit
Sockets: 1
Cores per socket: 4
Threads per core: 2
Model name: Intel(R) Core(TM) i7-8565U @ 1.80GHz
CPU MHz: 2112.004
Virtualization: VT-x
L1d cache: 32K
L2 cache: 256K
L3 cache: 8192K
Flags: fpu vme ... vmx ...

Zeigt also 1 Sockel, 4 Kerne, 8 Threads (4*2), CPU-Modell und Takt etc.

lsmem (List Memory Blocks)

Beschreibung: Zeigt Speicherblock-Informationen des Systems (insbesondere auf Systemen mit Hotpluggable Memory oder NUMA). Es listet Memory-Block-Geräte (wie in /sys/devices/system/memory) auf. Auf x86-PCs weniger relevant; auf IBM System z oder Power etc. nützlich.

Syntax:

lsmem [Optionen]

Optionen:

- **-n, --online / -o, --offline:** Zeigt nur online bzw. offline Speicherblöcke.
- **-b, --bytes:** Größen in Bytes statt lesbar.
- **-M, --mega:** Größen in MiB. (Standard lsmem gibt in bytes + kMG suffix)
- **-p, --pairs:** Ausgabe im Key:Value Format (für maschinelles Parsing).
- **-J, --json:** JSON-Format.
- **--output <list>:** Spalten auswählen (z.B. --output SIZE,NODE,STATE).
- **--help, --version:** analog.

Beispiel lsmem (auf PC ohne Hotplug memory):

RANGE	SIZE STATE REMOVABLE BLOCK
-------	----------------------------

```
0x0000000000000000-0x000000007fffffff 2G online yes 0-15
```

```
0x0000000100000000-0x000000027fffffff 6G online yes 32-79
```

Zeigt 2 memory ranges (entsprechend wohl 2 DIMMs). "Removable yes" heißt vom Kernel als Hot-remove möglich (oft 'no' auf Standard-PC). "BLOCK 0-15" referenzieren Memory block device IDs.

Auf NUMA-Systemen würde Spalte NODE relevant sein, um Memory pro NUMA-Node zu sehen.

ps (Process Status)

Beschreibung: Zeigt eine Momentaufnahme der aktuellen Prozesse des Systems. ps kann verschiedene Ausgabeformate haben (BSD-Style, UNIX-style). Häufig nutzt man ps aux oder `ps -effile-4fdk7rvx39azjretinarqb】 .

Syntax:

ps [Optionen]

(Komplexität: ps hat UNIX Optionen (mit - und bündelbar), BSD Optionen (ohne - und können zusammen geschrieben werden) und GNU long Optionen. Z.B. ps aux ist BSD Syntax, ps -ef ist UNIX syntax – beide gängig, liefern ähnliches.)

Gängige Aufrufe:

- **ps aux** – Listet alle Prozesse mit Details (BSD format). Spalten: USER, PID, %CPU, %MEM, VSZ, RSS, TT (TTY), STAT, START, TIME, COMMAND.
- **ps -ef** – Listet alle Prozesse (UNIX format). Spalten: UID, PID, PPID, C (CPU usage), STIME (start time), TTY, TIME (CPU time), CMD.
- **ps -eF** – wie -ef, aber "extra full", zeigt threads als separate Linien und vollständige Kommandozeilen etc.
- **ps -ejH** – zeigt Hierarchie (Prozessbaum) mit Prozess-IDs.
- **ps -eo pid,cmd,%mem,%cpu --sort=-%mem** – Zeigt benutzerdefinierte Spalten (hier pid, Befehl, Speicher- und CPU% sortiert nach Speicherverbrauch absteigend).

Wichtige Optionen:

UNIX-style (präfix mit -, können kombiniert werden):

- **-e** – Alle Prozesse anzeigen (entspricht -A).
- **-f – full** Format (mehr Spalten inkl. PPID, STIME, in Kombination z.B. -ef).
- **-F – Extra full** (noch mehr Infos: SZ, RSS, PSR CPU, etc.).

- **-h** – Ohne Header.
- **-H** – Hierarchie-Tree anzeigen (Indentation für Parent-Child).
- **-j** – Jobs format (PGID, SID etc. Spalten).
- **-l** – Long format (mehr spalten inkl. Flags, NI nice, LWP etc.).
- **-u <user>** – Prozesse eines bestimmten Benutzers.
- **-p <pidlist>** – Nur bestimmte PIDs.
- **-G <group>, -U <user>** – nach Gruppe/Benutzer filtern.

BSD-style (ohne -, einzeln hintereinander):

- **a** – Zeigt Prozesse von *allen Terminals* (nicht nur eigene).
- **x** – Zeigt auch prozesse *ohne Terminal* (Daemons). **ax** zusammen = alle Prozesse.
- **u** – Benutzerformat (inkl. Owner, CPU%, MEM%). => **aux** verbreitet.
- **j** – Jobs format (session, pgrp).
- **w** – breites Output (alle columns of command, use -ww to not truncate).

GNU long options (mit --):

- **--pid, --ppid, --user** etc. Alternative Filter.
- **--sort=<key>**: Sortiert Ausgabe nach Spalte(n). Z.B. --sort=-%cpu (absteigend CPU).
- **--no-headers**: Ohne Header (statt -h).
- **--forest**: ASCII-Art Bäumchen (ähnlich -H, aber macht "|_" und Einrückungen). Kombiniert mit e.g. -ef.

STAT Spalte (Prozessstatus):

Codes: R (running), S (sleep), D (uninterruptible sleep, i/o), Z (zombie), T (stopped), t (gestoppt im Hintergrund), X (dead). Flags dahinter: < (high prio), N (low prio, nice>0), L (pages locked in mem), s (session leader), + (foreground process group in TTY).

sar (System Activity Reporter)

Beschreibung: Teil von sysstat (wie iostat, mpstat, etc.), sammelt und berichtet historische System-Auslastungsdaten: CPU, Speicher, Paging, I/O, Netzwerk etc. sar kann *laufende* Stats anzeigen (wenn mit Interval aufgerufen), oder *vergangene Tageswerte* aus Logs (/var/log/sysstat/...) ausgeben.

Syntax:

`sar [Optionen] [Interval [Count]]`

Ohne Parameter, sar liest per default die Tagesstatistiken (cumulated) vom laufenden Tag (falls sar cron job aktiv war). Mit Interval sar 5 5 z.B. 5x alle 5 Sekunden CPU etc.

Hauptoptionen (für Bereiche):

- **-u:** CPU-Auslastung (Default, wenn keine angegeben) – zeigt %user, %nice, %system, %iowait, %steal, %idle.
- **-r:** Memory (free, used, buffers, cache, commit, etc.).
- **-b:** I/O (transfer rates, i/o per sec, etc.).
- **-B:** Paging (pgpgin/s, pgpgout/s, fault, majflt...).
- **-W:** Swapping (swap in/out).
- **-q:** Queue length (runq, i/o queue) and load average.
- **-n DEV:** Network: DEV mode – zeigt Netzwerk-Interface Stats (rxpck/s, txpck/s, rxKB/s, txKB/s, etc.). -n ALL für alle Net-Statistiken.
- **-d:** Block device I/O (similar to iostat per device).
- **-p [DEV]:** CPU per processor (if SMP). e.g. sar -P ALL 1 3 – CPU usage per core.
- **-f file:** Lese Stats aus Log-Datei (z.B. /var/log/sysstat/sarXX).
- **-s HH:MM:SS, -e HH:MM:SS:** Start/End time for reading logs (z.B. 9 bis 18 Uhr Stats).

Beispiel (aktuelle CPU alle 1s):

```
$ sar 1 3
```

Linux 5.4.0 ... (8 CPUs)

```
02:00:00 PM  CPU  %user  %nice %system %iowait %steal  %idle
02:00:01 PM  all   5.13  0.00  1.28  0.00  0.00  93.59
02:00:02 PM  all   7.14  0.00  1.02  0.00  0.00  91.84
02:00:03 PM  all   3.06  0.00  0.77  0.00  0.00  96.16
Average:    all   5.11  0.00  1.02  0.00  0.00  93.87
```

nohup (No Hangup)

Beschreibung: Lässt einen Prozess auch nach Logout weiterlaufen. nohup fängt das HUP-Signal (Hangup) ab, das normalerweise gesendet wird, wenn ein Terminal geschlossen wird. Mit nohup startet man ein Kommando so, dass es dieses Signal ignoriert und Standardausgabe ggf. in *nohup.out* umleite】 .

Syntax:

```
nohup <Befehl> [Arg...] [&]
```

Funktionsweise:

- nohup sorgt dafür, dass der ausgeführte Prozess das SIGHUP ignoriert. Außerdem werden *stdout* und *stderr* umgeleitet in *nohup.out*, falls diese derzeit ans Terminal gebunden sind. (Wenn man & anhängt – also im Hintergrund – ist es üblich, die Ausgabe nicht am Terminal zu lassen).
- Üblicherweise kombiniert mit & (Hintergrund). Man loggt dann aus, und das Programm läuft weiter.

Beispiele:

- nohup long_running.sh & – Führt *long_running.sh* im Hintergrund aus, Ausgaben gehen in *nohup.out*. Der Befehl überlebt Logout. Die Shell gibt z.B. zurück: appending output to nohup.out.
- nohup python server.py > server.log 2>&1 & – Startet Python-Server im Hintergrund, leitet *stdout* und *stderr* in *server.log* (damit *nohup.out* nicht genutzt).

Hinweis: Wenn man bereits in einer Screen- oder tmux-Session ist, braucht man kein nohup; auch systemd-run oder Cron-Jobs sind Alternativen. Aber nohup ist ein schnelles Mittel.

mkswap/swapon/swapoff (Swap einrichten und verwalten)

- **mkswap** – Initialisiert eine Partition/Datei als Swapbereich. Syntax: mkswap [Optionen] <Gerät/datei>. Wichtige Option: -L <Label> (Swap-Label setzen), -U <UUID>. Beispiel: mkswap /dev/sda3.
- **swapon** – Aktiviert einen Swapbereich für Nutzung durch das System. Nach mkswap muss Partition via swapon <Device> eingebunden werden (oder in /etc/fstab eingetragen mit swap Typ). Option -a aktiviert alle in fstab markierten Swaps. --show zeigt aktuellen Swap an (Äquivalent cat /proc/swaps).
- **swapoff** – Deaktiviert Swap. Syntax: swapoff <Device> oder swapoff -a für alle.

Beispiele:

```
# mkswap -L "SWAP" /dev/sda3
```

```
Setting up swapspace version 1, size = 8 GiB (some hex UUID)
```

```
# swapon /dev/sda3
```

```
# swapon --show
```

```
NAME  TYPE SIZE USED PRIO
```

```
/dev/sda3 partition 8G 0B -2
```

swapoff -a würde alle Swapbereiche aus dem Betrieb nehmen (z.B. vor Partition-Vergrößerung oder Kernel-Dump etc.).

free (Speicherübersicht)

Beschreibung: Zeigt eine Übersicht über den aktuellen Speicherverbrauch (RAM und Swap) im System. Es summiert total, used, free, and cached/buffered memory.

Syntax:

```
free [Optionen]
```

Optionen:

- **-h:** Human Readable (z.B. MiB/GiB).
- **-m, -g, -k:** Ausgabe in MiB, GiB, KiB.
- **-b:** in Bytes.
- **-t:** Fügt eine "Total" Zeile (RAM+Swap) hinzu.
- **-s N:** Aktualisiert alle N Sekunden (kontinuierliche Ausgabe wie watch).
- **-c N:** Wenn mit -s genutzt, stop nach N Aktualisierungen.
- **--si:** Benutzt 1000er statt 1024er Basis (KB = 1000 Bytes etc.).

Beispiel free -h:

	total	used	free	shared	buff/cache	available
Mem:	15Gi	5.2Gi	2.3Gi	1.1Gi	7.4Gi	8.8Gi
Swap:	2.0Gi	0.0Gi	2.0Gi			

available ist eine Kernelschätzung, wieviel noch für neue Prozesse verfügbar, unter Berücksichtigung von caches, etc.

vmstat (Virtual Memory and system stats)

Beschreibung: Zeigt verschiedene Systemmetriken: Speicher, Auslagerung, CPU, Interrupts, Context Switches etc. im zeitlichen Verlauf. vmstat ohne Parameter gibt Durchschnitt seit Boot; mit Intervall zeigt periodische Werte (Differenzen pro Intervall).

Syntax:

```
vmstat [Optionen] [Interval [Count]]
```

Wichtige Spalten (Standardausgabe):

- **procs:** *r* (Anzahl laufender Prozesse im Runqueue), *b* (blockierte Prozesse).
- **memory:** *swpd* (Swap genutzt), *free* (freier RAM), *buff* (Puffer), *cache* (Page Cache).
- **swap:** *si* (Swap-In KB/s), *so* (Swap-Out KB/s).
- **io:** *bi* (Block In, Blöcke gelesen von Disk/s), *bo* (Block Out, geschrieben/s).
- **system:** *in* (Interrupts/s), *cs* (Context Switches/s).
- **cpu:** *us* (% CPU in Userzeit), *sy* (% Systemzeit), *id* (% Idle), *wa* (% IO-Wait), *st* (% Steal bei virt.).

Optionen:

- **-S <unit>**: Werte in k (1000) oder K (1024) etc.
- **-a**: Zeigt zusätzlich Aktivität der *aktiven/inaktiven* Memory Listen (sin, sout).
- **-d**: Zeigt Disk-Statistiken (like iostat) statt VM stats.
- **-p <dev>**: Zeigt Partition-Statistiken (reads/writes tot und sek) für Device.
- **-s**: Summiert memory stats (eine Liste aller relevanten Zähler, analog /proc/meminfo).
- **-m, -t**: Variation der Ausgabe (meist uninteressant, -m memory in MB etc).

Beispiel (alle 2s):

```
$ vmstat 2 3

procs -----memory----- swap-- -----io---- -system-- -----cpu-----
r b free buff cache si so bi bo in cs us sy id wa st
1 0 230000 120000 760000 0 0 5 10 150 300 5 191 3 0
0 0 229000 120000 760500 0 0 0 20 140 280 4 194 1 0
1 0 228500 120000 761000 0 0 0 30 160 320 6 192 1 0
```

Erste Zeile nach Header ist seit Boot (ohne parameter, oft uninteressant); weitere sind Echtzeit pro Intervall.

modinfo (Module Information)

Beschreibung: Zeigt Informationen über ein Linux-Kernelmodul (Loadable Kernel Module) – Version, Beschreibung, Lizenz, Abhängigkeiten, Parameter etc】 .

Syntax:

modinfo [Optionen] <Modulname oder Modul.ko>

Optionen:

- **-a, --author:** Nur Autor ausgeben.
- **-d, --description:** Nur Beschreibung.
- **-n, --filename:** Zeigt Pfad der Moduldatei im Filesystem.
- **-p, --parameters:** Listet die Parameter und ihre Typen/Beschreibungen.
- **-F <field>:** Gibt nur das spezifizierte Feld aus (alternativ obige Kurzoptionen).
Mögliche Felder: author, description, license, parm, alias, etc.
- **-k <kernelversion>:** Fragt Infos für Kernel-Version (sucht in
`/lib/modules/<version>`) statt laufendem Kernel.

Beispiel: modinfo e1000e (Intel NIC Treiber) gibt etwa:

```
filename: /lib/modules/5.4.0/kernel/drivers/net/ethernet/intel/e1000e/e1000e.ko
version: 3.2.6-k
license: GPL
description: Intel(R) PRO/1000 Network Driver
author: Intel Corporation, <e1000-devel@lists.sourceforge.net>
...
srcversion: 5C6D1A8....
alias: pci:v00008086d000010D3sv*sd*bc*sc*i*
...
parm: IntMode:Interrupt Mode (0=Auto, 1=MSI, 2=MSI-X) (int)
parm: InterruptThrottleRate:... (array of int)
```

Man sieht Dateipfad, Version, Parameter etc.

insmod/rmmod/modprobe (Kernelmodule laden/entladen)

- **insmod <pfad/modul.ko> [args]:** Lädt ein Modul in den Kernel. Nutzt genau Pfad; löst keine Abhängigkeiten auf.
- **rmmod <modulname>:** Entfernt ein geladenes Modul (wenn kein anderer in Benutzung). Option -f zum Erzwingen (nur falls Kernel CONFIG_MODULE_FORCE_UNLOAD).
- **modprobe [Options] <modulname> [args]:** Empfehlenswerter Befehl zum Laden eines Moduls und seiner Abhängigkeiten. Es durchsucht

/lib/modules/<kernel>/modules.dep nach Dependencies. Auch zum Entfernen via modprobe -r <modul> (lädt auch Abhängigkeiten aus, falls unbenutzt).

- Optionen: --dry-run oder -n testet nur, -v verbose, -q quiet, alias handling etc. modprobe Konfig steht in /etc/modprobe.d.
- Bsp: modprobe e1000e (lädt Intel NIC modul + ggf. benötigte mii). modprobe -r e1000e entfernt es.

depmod (Dependency modules)

Beschreibung: Generiert die Modulabhängigkeitsdatei *modules.dep* basierend auf Modulen in /lib/modules/<version>. Wird meist beim Kernel-Install aufgerufen. Kann man manuell ausführen nach Hinzufügen eigener Module. Syntax: depmod -a (für alle).

Optionen:

- -a, --all (default): alle Kernel-Versionen verarbeiten.
- -A, --quick: nur verändern, wenn *modules.dep* älter als Module-Dateien (incremental update).
- -n: Ausgabe nach stdout statt in *modules.dep* schreiben (Test).
- -v: verbose.

sysctl (Query/Set Kernel Parameters)

Beschreibung: Anzeige oder Änderung von Kernel-Laufzeitparametern (in /proc/sys). sysctl ermöglicht bequemen Zugriff auf z.B. Netzwerktruning, VM, FS Settings zur Laufzeit.

Syntax:

sysctl [Optionen] [<Name>=<Wert> ...] or sysctl -p

Optionen:

- Ohne =: liest Parameter. Z.B. sysctl net.ipv4.ip_forward -> net.ipv4.ip_forward = 0.
- Mit name=value: setzt Parameter. Z.B. sysctl net.ipv4.ip_forward=1 aktiviert IP-Forwarding (sofort).
- -w: explizit Schreibmodus (kann man weglassen, war historisch).
- -p [Datei]: Lädt Einstellungen aus Datei (Default /etc/sysctl.conf) und wendet an. Man ruft sysctl -p typ. beim Boot (init-Skripte) oder manuell nach Bearbeiten der config.
- -a: Alle Parameter anzeigen (riesige Liste). sysctl -a | grep ipc etc.
- -A, --pattern <regex>: Filtert mit Regex (statt manuell grep).

- **-e:** Ignoriert unbekannte Keys (useful mit -p, sonst bricht ab).
- **-N:** Nur Namen, keine Werte anzeigen (list keys).
- **-q:** Quiet (unterdrückt Bestätigungsmeldungen).

Parameter-Keys entsprechen /proc/sys Pfaden, z.B. /proc/sys/net/ipv4/ip_forward -> key net.ipv4.ip_forward.

Beispiele:

- `sysctl vm.swappiness` -> zeigt swappiness (z.B. `vm.swappiness = 60`).
- `sysctl -w vm.swappiness=10` -> setzt auf 10 (weniger Swapneigung).
- `sysctl -p /etc/sysctl.d/99-sysctl.conf` -> lädt Parameter aus der angegebenen Datei.

dmesg (Dump kernel message buffer)

Beschreibung: Zeigt die Nachrichten des Kernel-Ringspeichers (Kernellog) an. Hier stehen Boot-Meldungen und Hardware-Events (z.B. USB eingesteckt, Treiber Ausgabe). dmesg ohne Optionen druckt alle aktuellen Kernelmeldungen.

Syntax:

`dmesg [Optionen]`

Optionen:

- **-T:** Zeigt menschenlesbare Zeitstempel (Jahr-Monat-Tag Stunde:Minute:Sekunde) statt der raw Sekunden seit Boo. (Achtung: Kann ungenau sein, da Kernel keine Zeitzone kennt; -T rechnet anhand aktueller TZ).
- **-l <level>, --level:** Filtert Nachrichten bestimmter Log-Levels. Level: emerg, alert, crit, err, warn, notice, info, debug. Z.B. `dmesg -l err, warn` nur Fehler und Warnungen.
- **-k, --kernel:** Zeigt nur Kernel-Messages (Standard). **-u, --userspace:** Nur Benutzerraum (Logeinträge vom `printk_user`). Kaum genutzt, da meiste aus Kernel.
- **-H, --human:** **Pager-Modus** (lässt scrollen, farbig Highlighting). Interaktiv (Space/PgUp etc.).
- **-w, --follow:** "Tail -f" Modus für dmesg – neue Kernelmeldungen laufend anzeigen.
- **-s N:** Buffer-Größe (Bytes) die gelesen wird (Default könnte begrenzt sein). `dmesg -s 1000000`.

Beispiel (gekürzt):

```
[ 0.000000] Linux version 5.4.0-80-generic ...
[ 0.345678] PCIe Bus 0000:00: root hub ...
[ 1.234567] usb 1-2: new high-speed USB device number 3 ...
[ 1.345678] ata1: SATA link up ...
[ 10.567890] Bluetooth: RFCOMM ver 1.11
```

dmesg -T zeigt z.B. [Mon Oct 11 14:23:45 2021]

Typisch ruft man dmesg | grep -i error um nach Kernel-Fehlern zu suchen, oder dmesg -w in einem Terminal um z.B. neue USB/Hardware Events sofort zu sehen.

Dienstverwaltung und Systemstart

systemctl (Control systemd)

Beschreibung: Haupt-Werkzeug zur Steuerung des *systemd*-Initsystems (heutzutage auf vielen Linux-Distributionen). systemctl verwaltet **Services** (Start/Stop/Enable), **Daemon-Konfiguration, System-Zustände** (Reboot/Shutdown), **Unit**-Status und Logs, etc】 .

Syntax:

systemctl [Optionen] <Sub-Befehl> [Unit/...]

Units können Services (.service), Mounts (.mount), Timer (.timer), Sockets (.socket), etc. sein. Oft kann man den Suffix weglassen (z.B. "ssh" wird als "ssh.service" erkannt).

Wichtige Sub-Befehle (Service Management):

- **start <service>, stop <service>** – Startet oder stoppt einen Dienst *sofortfile-4fdk7rvx39azjretinarqb】 . Bsp: systemctl start apache2.
- **restart <service>** – Stop + Start (auch wenn nicht lief, wird gestartet).
- **reload <service>** – Lädt Konfiguration neu (sendet i.d.R. SIGHUP). Muss vom Service unterstützt werden (in Unit als ReloadSignal oder ExecReload definiert). Bsp: systemctl reload nginx.
- **status <service>** – Zeigt Status des Dienstes: Läuft er? Letzte Startmeldung, ggf. Auszug aus Journal-Log】 .
- **enable <service>** – Aktiviert den Dienst beim Boot (erstellt Symlinks in passenden WantedBy-Verzeichnissen).
- **disable <service>** – Deaktiviert Autostart.
- **is-enabled, is-active** – Prüfen ob Unit aktiviert ist / gerade aktiv ist (exit 0/1).

- **mask <service>** – "Maskiert" den Service, d.h. verhindert Start vollkommen (setzt Link auf /dev/null). *Unmask* zum Aufheben.
- **daemon-reload** – Falls man eine Unit-Datei geändert oder neu hinzugefügt hat, diese neu einlesen (Konfig neu laden, ohne Dienste neu zu starten).
- **kill <service>** – Sendet ein Signal an alle Prozesse der Service-Unit (z.B. systemctl kill -s SIGUSR1 myapp.service).

System Commands:

- **list-units** – Listet alle geladenen Einheiten (nur aktive standardmäßig). --all um Inaktive auch zu sehen, --type=service filtern auf Services, etc.
- **list-unit-files** – Listet Unit-Dateien und ob sie enabled/disabled/etc. sind (Installationszustand).
- **list-dependencies <unit>** – Zeigt Abhängigkeiten (Requires/Wants) Baum für Unit. --reverse zeigt welche Units diese Unit benötigen.
- **show <unit>** – Gibt alle Properties von Unit aus (oder subset via -p).
- **cat <unit>** – Zeigt den Inhalt der Unit-Datei (und Drop-Ins).
- **edit <unit>** – Öffnet Editor für Drop-In Snippet, um diese Unit zu überschreiben/erweitern.
- **preset <service>** – Setzt enable/disable gemäß Distribution-Vorgabe (in /usr/lib/systemd/system-preset/).

Targets (Runlevels):

- **isolate <target>** – Wechselt in ein target (beendet andere). Z.B. systemctl isolate rescue.target (Einzeluser-Modus ähnlich runlevel 1).
- **set-default <target>** – Legt Default Boot-Target fest (z.B. graphical.target vs multi-user.target).
- **get-default** – zeigt aktuelles Defaulttarget.

Power Management:

- **reboot, poweroff, halt** – führen Neustart bzw. Herunterfahren aus (rufen intern entsprechende target isolation).
- **suspend, hibernate, hybrid-sleep** – starten Energiesparmodus.

Journal/Logging Integration:

- **status** (wie oben) zeigt bereits Log-Auszug.

- **journalctl** (separater Befehl, nicht systemctl subcommand) – zum Betrachten der systemd Logs. Z.B. journalctl -u <service> zeigt Logs zu einem Service, -f zum folgen, --since "2021-10-01" ab bestimmtem Datum etc.

Beispiele:

- systemctl enable --now nginx – Aktiviert **und** startet nginx sofort (Option --now führt gleich start nach enable aus).
- systemctl disable apache2.service – Entfernt Apache2 aus Autostart.
- systemctl restart networking – Dienst neu starten (bei Netz aufpassen).
- systemctl list-units --failed – Zeigt nur fehlgeschlagene Units.
- systemctl mask firewalld; systemctl stop firewalld – Verhindert Start des Firewalld dauerhaft und stoppt ihn sofort (z.B. falls man anderen Firewall nutzt).

service (SysVinit Compatibility)

Beschreibung: Der Befehl service war bei SysVinit/Upstart gebräuchlich, um Start/Stop-Skripte in /etc/init.d aufzurufen. Auf systemd-Systemen ruft service intern systemctl auf. Man kann aber service <name> start/stop/status noch benutzen.
4fdk7rvx39azjretinarqb】 . Moderne Systeme – besser systemctl direkt.

Syntax:

service <ServiceName> [start|stop|restart|status|...]

Beispiel: service ssh status ruft systemctl status ssh.service auf.

(Auf SysVinit ohne systemd würde es das init.d Script ausführen.)

chkconfig (Runlevel Konfiguration - SysV kompatibel)

Beschreibung: Ein Werkzeug zum Verwalten der Start-/Stopp-Links von SysV-Init-Skripten über verschiedene Runlevels (0-6file-4fdk7rvx39azjretinarqb】 . Auf systemd-Systemen gegenstandslos (meist vorhanden für Kompatibilität, was auf systemctl enable/disable mappt).

Wichtigste Befehle:

- chkconfig --list [name] – Listet alle Dienste und ob sie in runlevel 0-6 on/off sind
4fdk7rvx39azjretinarqb】 . Mit Name nur diesen.
- chkconfig name on (bzw. off) – Schaltet Service in allen entsprechenden runlevels an oder aus (Standard runlevels 2-5 on für on).
- chkconfig name --level 35 on – Feinsteuerung: aktiviert Dienst in Runlevel 3 und 5.
- chkconfig name reset – Setzt Standard (wie im Init-Skript Header vorgesehen).

*(Auf systemd delegiert chkconfig intern an systemctl enable/disable. Für Nostalgie, wissen was es tat.)

Zeitplanung (Cron und At)

crontab (Cron Table Management)

Beschreibung: Planen wiederkehrender Aufgaben. crontab bearbeitet die Aufgabenliste des aktuellen Benutzers (oder eines anderen mit -u) im Cron-Daemon. Jede Zeile in crontab -e hat Format "*Min Stunde Tag Monat Wochentag Befehl*". Der Cron-Dienst prüft pro Minute die Tabellen und führt fällige Kommandos auf.

Befehle:

- crontab -e – Ediert die Crontab des aktuellen Benutzers (öffnet \$EDITOR). Beim Speichern wird die neue Tabelle dem Cron-Daemon übergeben.
- crontab -l – Listet die aktuelle Crontab auf.
- crontab -r – Entfernt (löscht) die aktuelle Crontab komplett. (Vorsicht, kein Undo).
- crontab -u username -e/-l/-r – Als root: Bearbeitet/liest/löscht die Crontab eines anderen Benutzers.

Cron Format Erinnerung:

Minute(0-59) Hour(0-23) Day(1-31) Month(1-12) Weekday(0-7) Command

0 2 * * * /usr/bin/backup.sh # täglich 02:00

*/5 9-17 * * 1-5 echo "Working..." # Werktags 9-17 Uhr alle 5 min

*=jeder Wert, Komma getrennte Listen, Intervalle mit -, Step mit /.

Spezial @-Strings: @reboot, @hourly, @daily, @yearly etc.

Anacron: Cron führt Jobs aus, wann Maschine an ist. Für nicht ständig laufende Rechner nutzt man anacron (Tagesjobs nachholen).

at (One-time scheduled task)

Beschreibung: Legt einmalige Ausführungen von Befehlen zu einer bestimmten Zeit in der Zukunft fest. at nimmt eine Zeitangabe entgegen und liest dann von STDIN die Befehle, die ausgeführt werden sollen (oder mit -f aus einer Datei). Der atd Daemon muss laufen.

Verwendung:

```
echo "/path/to/script.sh" | at 02:30 # führt Script heute Nacht 2:30 aus
```

Man ruft at <Zeit> auf, landet in einer Stdin-Eingabe (end mit Ctrl+D) oder speist via echo/<<EOF.

Zeitformat: Z.B. at 14:00 (heute 14 Uhr, falls schon vorbei dann morgen), at 11pm, at midnight, at now + 5 days, at 08:00 12.12.2025 (8:00 am 12 Dec 2025). Kann englische Wörter: tomorrow, noon, teatime(16:00) etc.

Befehle:

- atq – Zeigt anstehende at-Jobs (Queue) des aktuellen Userfile-4fdk7rvx39azjretinarqb】 . (Listet Job-IDs und geplante Zeit). Root sieht alle oder atq -q <Queue> für bestimmte Batch-Queues.
- atrm <JobID> – Entfernt einen geplanten Job aus der Queue. (JobID sieht man in atq).
- batch – Alias zu at -q b now (führt Jobs aus, wenn Load Average niedrig ist). "Batch" schiebt den Auftrag in Batch-Queue, läuft sobald System nicht ausgelastet (LA<0.8 default).

Beispiel:

```
$ at now + 1 minute
```

```
warning: commands will be executed using /bin/sh
```

```
at> echo "Hello after 1 min" >> /tmp/test.txt
```

```
at> <EOT>
```

```
job 5 at Fri Oct 15 16:41:00 2021
```

Dann innerhalb einer Minute atq zeigt e.g. 5 Fri Oct 15 16:41:00 2021 a user. Nach Ausführung landet Text in Datei.

(Zugriff: /etc/at.allow /etc/at.deny definieren wer at nutzen darf – analog cron.allow/deny.)

Drucken

lpr (Line Print Request)

Beschreibung: Sendet eine Datei (oder STDIN) an den Drucker (Print-Job einreihen). lpr ist Teil des BSD-Linux-Drucksystems (CUPS stellt Kompatibilität bereit). Alternative: lp (System V style).

Syntax:

```
lpr [Optionen] [Dateien...]
```

Wichtige Optionen:

- **-P <Druckernname>**: Wählt einen bestimmten Drucker aus (sonst Standarddruckerfile-4fdk7rvx39azjretinarqb】 . Druckernamen findet man via lpstat -a oder CUPS Webinterface.

- **-# <Num>**: Anzahl Kopien drucken (z.B. -#3 dreifach).
- **-h**: Unterdrückt Banner-Seite (header page) falls aktiviert.
- **-r**: Löscht die Datei(en) nach dem Versenden an den Spool (rarely used).
- **-m**: Sendet E-Mail Benachrichtigung nach Druck (sofern eingerichtet).
- **-o <Option>=<Wert>**: Drucker-spezifische Optionen übergeben (z.B. -o landscape für Querformat, oder -o media=A4 oder -o sides=two-sided-long-edge für Duplex). Diese Optionen sind je nach Drucker Treiber definierbar (CUPS options).
- **-C <Klasse>, -J <Jobname>, -T <Titel>**: Legen Klassenname, Jobname, Titel für Banner etc. fest.
- **--help** (GNU lpr) usw.

Beispiele:

- lpr report.pdf – Schickt *report.pdf* an Standarddrucker.
- lpr -P HP_LaserJet -#2 invoice.txt – Druckt *invoice.txt* 2-fach auf Drucker "HP_LaserJet".
- echo "Test Page" | lpr – Druckt STDIN (den Text "Test Page").
- lpr -o sides=two-sided-long-edge thesis.pdf – Duplexdruck beidseitig.

(Verwandt: *lpq* zeigt Druckerqueue, *lprm* löscht Jobs aus der Queue.)

Lokale Systemkonfiguration

date (Anzeige/Setzen Datum/Uhrzeit)

Beschreibung: Zeigt das aktuelle Datum und die Uhrzeit an, oder stellt sie ein. Auch Formatierung in beliebigem Muster ist möglich.

Syntax:

date [Optionen] [+Format]

date -s "STRING" (setzen)

Optionen / Verwendung:

- **Ausgabe formatieren:** date "+FORMAT" – Mit Format-Pattern ähnlich strftime:
%Y (Jahr), %m Monat(01), %b Monat (Kurzname), %d Tag, %H Stunde (24h), %I (12h), %M Minute, %S Sek, %A Wochentag, %a (kurz), %Z Zeitzone, %z offset, %T = %H:%M:%S, %F = %Y-%m-%d, etc. Z.B. date "+%d.%m.%Y %H:%M:%S" -> "05.10.2025 14:30:00".

- **Systemzeit stellen:** date -s "2025-10-05 14:30:00" – Setzt Datum. Braucht Root. (Alternative: date --set="...").
- **Hardware-Uhr stellen:** In Linux erfolgt date auf Systemzeit (in Kernel, RTC separiert). Um auch BIOS/RTC Uhr zu setzen, hwclock --systohc (setze Hardware Clock nach System Clock). Umgekehrt hwclock --hctosys.
- **Universale Zeit:** date -u zeigt in UTC. date -u -s "2025-10-05 12:30:00" setzt in UTC.
- **RFC-3339 Format:** date --rfc-3339=ns etc. Standard ISO.
- **Unix Timestamp:** date +%s gibt Sekunden seit 1.1.1970 (UTC).

Beispiele:

- date -> "Tue Oct 5 14:30:00 CEST 2025".
- date +"%Y-%m-%d_%H-%M-%S" -> "2025-10-05_14-30-00".
- sudo date -s "@\$((\$(date +%s) + 3600))" -> Systemzeit um eine Stunde vorstellen (Beispiel mit timestamp math).

timedatectl (Zeiteinstellung via systemd)

Beschreibung: Befehl von systemd zum Anzeigen und Ändern der Systemzeit, Hardware-Uhr, Zeitzone und NTP-Synchronisatiofile-4fdk7rvx39azjretinarqb】 .

Verwendung:

- **timedatectl** (ohne args): Zeigt aktuellen Status: Lokale Zeit, UTC, RTC, Zeitzone, ob NTP aktiv ist etcfile-4fdk7rvx39azjretinarqb】 .
- **timedatectl set-time "YYYY-MM-DD HH:MM:SS"** – Setzt Systemzeit (wenn kein NTP aktiv).
- **timedatectl set-timezone <Zone>** – Ändert Zeitzone. Z.B. timedatectl set-timezone Europe/Berlin. (Verfügbare Zonen unter /usr/share/zoneinfo/, oder timedatectl list-timezones).
- **timedatectl set-ntp true/false** – Aktiviert bzw. deaktiviert die automatische Zeitsynchronisation (via systemd-timesyncd oder anderes NTP). Bei true startet es den Dienst falls vorhanden.
- **timedatectl set-local-rtc 1/0** – Legt fest, ob die Hardware-Uhr (RTC) in **lokaler Zeit** läuft (1) oder in UTC (0, Standard). Windows dual-boot oft RTC=lokal (dann 1), sonst belässt auf 0=UTC um Probleme zu vermeiden.

Beispiele:

\$ timedatectl

Local time: Tue 2025-10-05 14:30:00 CEST

Universal time: Tue 2025-10-05 12:30:00 UTC

RTC time: Tue 2025-10-05 12:30:00

Time zone: Europe/Berlin (CEST, +0200)

System clock synchronized: yes

NTP service: active

RTC in local TZ: no

timedatectl set-time "2025-12-24 18:00:00" stellt Heiligabend 18:00. timedatectl set-timezone UTC wechselt Zeitzone auf UTC.

localectl (Locale and Keyboard Layout)

Beschreibung: Verwaltet System-Locale (Sprache, Gebietsschema) und Tastaturlayout-Einstellungen in systemd-basierten Systemefile-4fdk7rvx39azjretinarqb】. Es kann die Variablen in /etc/locale.conf sowie Konsolen-Tastaturlayout in /etc/vconsole.conf setzen. (Für X/Wayland Tastatur greift localectl in /etc/X11/xorg.conf.d/ oder mittels setxkbmap separat).

Verwendung:

- **localectl** (ohne args): Zeigt aktuelle Locale(s) und Keymap. Z.B.:
System Locale: LANG=de_DE.UTF-8
VC Keymap: de
X11 Layout: de
X11 Model: pc105 etc.
- **localectl list-locales** – Listet alle verfügbaren Locale, die generiert sind (die in /usr/lib/locale oder /etc/locale.gen markiert).
- **localectl set-locale LANG=<locale> [VAR=locale...]** – Setzt System-Locale. Z.B. localectl set-locale LANG=en_US.UTF-8. Optional mehrere, z.B. LC_TIME=en_GB.UTF-8. Wirksam nach Neustart oder neu login shells.
- **localectl list-keymaps** – Listet verfügbare Console-Keymaps (z.B. us, de, ...).
- **localectl set-keymap <map> [<toggle>]** – Setzt Tastaturlayout für Konsole (vconsole) und ggf. X11, falls X11 Layout nicht getrennt definiert. Z.B. localectl set-keymap de.
- **localectl set-x11-keymap <layout> [model] [variant] [option]** – Spezifisch X11 (Graphical) Layout/Modell setzen. Z.B. localectl set-x11-keymap us pc104 "" terminate:ctrl_alt_bksp.

Beispiel:

localectl set-locale LANG=fr_FR.UTF-8 (Systemssprache auf Französisch).

localectl set-keymap us (Konsole auf US-Layout).

localectl set-x11-keymap de pc105 nodeadkeys (Grafische Tastatur auf Deutsch mit Variante "nodeadkeys").

(Die Konfigurationen landen in /etc/locale.conf, /etc/vconsole.conf, /etc/X11/xorg.conf.d/00-keyboard.conf entsprechend.)

Netzwerk-Konfiguration

ip (Netlink Interface Config)

Beschreibung: Moderner Befehl zur Verwaltung von Netzwerkinterfaces, IP-Adressen, Routing, Tunnels etc. (Nachfolger von ifconfig/route). ip hat Unterkommandos für verschiedene Objekte: **link** (Netzwerkinterface), **addr** (IP-Adressen), **route**, **neigh** (ARP/NDP Nachbarn), **rule** (Policy Routing), **maddr** (Multicast), **addrlabel**, **netns** (Net Namespaces), **link set** (Interface-Settings).

Syntax (Grundstruktur):

ip [Options] <Object> <Subcommand> [Arguments...]

z.B. ip link show, ip addr add, ip route list, etc.

Häufige Befehle/Beispiele:

- **Interface anzeigen:** ip link show [dev <iface>] – Listet alle Netzwerkinterfaces (oder ein spezifisches). Ausgabe enthält Index, Name, MAC, MTU, Flags (UP,LOWER_UP=Link OK, MULTICAST), etc.
- **Interface aktivieren/deaktivieren:** ip link set <iface> up / down – entspricht `ifconfig iface up/downfile-4fdk7rvx39azjretinarqb】 .
- **Adresse zuweisen:** ip addr add <IP>/<Prefix> [broadcast +] dev <iface> – z.B. ip addr add 192.168.1.100/24 dev eth0. broadcast + lässt broadcast aus Prefix berechnen (optional).
- **Adresse löschen:** ip addr del <IP>/<Prefix> dev <iface>.
- **Adressen anzeigen:** ip addr show [dev <iface>] – Listet IPv4/IPv6 Adressen auf Interfacefile-4fdk7rvx39azjretinarqb】 . Zeigt auch Scope (global/link), flags (tentative, dynamic).
- **Standardroute setzen:** ip route add default via <Gateway-IP> [dev <iface>] – z.B. ip r add default via 192.168.1.1 dev eth0.
- **Statische Route setzen:** ip route add <Netz>/<Präfix> via <nächsterHop> dev <iface>.

- **Routen auflisten:** ip route show (oder ip r).
- **ARP Cache:** ip neigh show – zeigt ARP/ND-Cache (Neighbors).
- **ARP Eintrag fix setzen:** ip neigh add <IP> lladdr <MAC> dev <iface> nud permanent.
- **Secondary IPs / Aliase:** Einfach mehrfach ip addr add auf selbes dev ausführen (oder use Label).
- **MTU ändern:** ip link set dev <iface> mtu 1400.
- **MAC ändern:** ip link set dev <iface> address <NewMAC> (Down interface first).
- **Promisc Mode:** ip link set dev <iface> promisc on.
- **Routing Policy (advanced):** ip rule und ip route add table <n> for source-based routing etc.
- **Network Namespace mgmt:** ip netns (add, del, exec, etc – advanced).

Options global: -s (stats), -d (details), -br (brief output), -c (color output). Z.B. ip -br addr zeigt Interfaces in einzeiligem Kurzformat.

Beispiele:

- ip -br link ->

```
lo      UNKNOWN    127.0.0.1/8 ::1/128
eth0     UP        192.168.1.10/24 fe80::.../64
```

- ip addr add 10.0.0.1/16 dev eth1 (fügt IP).
- ip route show (z.B. default via 192.168.1.1 dev eth0 proto dhcp metric 100 etc.).
- ip neigh show (z.B. 192.168.1.1 dev eth0 lladdr aa:bb:cc:dd:ee:ff REACHABLE).

(ifconfig (net-tools) alternative: ip ist ausführlicher und bevorzugt.)

ifconfig (Interface Config – alt)

Beschreibung: Älteres Werkzeug aus net-tools (weitgehend ersetzt durch ip). ifconfig kann Interfaces anzeigen, IP setzen, MTU ändern, promiscuous Mode etc., aber unterstützt z.B. keine IPv6 out-of-the-box je nach Version.

Syntax:

ifconfig [Interface] [Optionen]

Beispiele:

- ifconfig – zeigt alle aktiven (UP) Interfaces mit IPv4 etc.

- ifconfig -a – zeigt alle, inkl. down Interfaces.
- ifconfig eth0 192.168.0.5 netmask 255.255.255.0 broadcast 192.168.0.255 – setzt IP, Maske, Bcast.
- ifconfig eth0 up / down – Interface an/aus.
- ifconfig eth0 mtu 1400 – MTU ändern.
- ifconfig eth0 promisc – Promiscuous Mode an. -promisc aus.
- ifconfig eth0 hw ether 00:11:22:33:44:55 – MAC-Adresse setzen.

(ifconfig ist nicht mehr standard installiert in neueren Distros; ip commands stattdessen nutzen.)

iwconfig (Wireless Config – alt)

Beschreibung: Ähnlich ifconfig, aber für Wireless LAN Schnittstellen (Legacy tool aus wireless-tools). Zeigt/ändert ESSID, Mode, Frequenz, Schlüssel etc. Heutzutage ersetzt durch iw (nl80211), oder NetworkManager.

Beispiele (falls vorhanden):

- iwconfig wlan0 – zeigt ESSID, Mode (Managed/Ad-Hoc/Master), Channel/Freq, Bitrate, Tx-Power, Signal Level, Encryption key on/off, etc.
- iwconfig wlan0 essid "WLAN-Name" – mit offener SSID verbinden (sofern AP ohne Sicherheit).
- iwconfig wlan0 key s:password – setzt WEP Key (veraltet).
- iwconfig wlan0 mode Monitor – Interface in Monitor Mode (für Sniffing).

(Iwconfig kann kein WPA2; da nutzt man wpa_supplicant oder NM, etc.)

nmcli (NetworkManager CLI)

Beschreibung: Kommandozeilen-Interface zum NetworkManager-Dienst, um Netzwerkkonfiguration (insb. Desktop-Umfeld) zu betrachten und steuerfile-4fdk7rvx39azjretinarqb】 . Kann Verbindungen (profiles) verwalten, aktivieren, WLAN scannen, etc.

Verwendung (Struktur):

nmcli [general|networking|radio|connection|device] < Aktion > [Parameter]

Beispiele:

- **Status anzeigen:** nmcli general status – Gesamter NM-Status. nmcli connection show – listet gespeicherte Verbindungen (Profiles). nmcli device status – zeigt Geräte und ob connected und welches Profil.

- **WLAN scannen:** nmcli device wifi list – listet verfügbare WLANs.
- **Verbinden WLAN:** nmcli device wifi connect "SSID" password "passwort" – erstellt ggf. Connection-Profil und verbindet.
- **Verbindung aktivieren/deaktivieren:** nmcli connection up <Name> / down <Name>. Oder nmcli device disconnect <iface> (z.B. nmcli device disconnect wlan0).
- **Neue Verbindung anlegen:** nmcli connection add type ethernet ifname eth0 con-name "Wired Manual" ip4 192.168.5.5/24 gw4 192.168.5.1.
- **IPv4 ändern laufend:** nmcli connection modify "Wired connection 1" ipv4.addresses 192.168.1.50/24 ipv4.gateway 192.168.1.1 ipv4.method manual und dann up.
- **Ethernet an/aus:** nmcli networking off (setzt NM generell offline) oder nmcli radio wifi off (schaltet nur WLAN-Funk aus).

Beobachten:

- nmcli connection show --active – nur aktive Conns.
- nmcli -p device status – pretty (ncurses-like) output.

ethtool (NIC Driver Tool)

Beschreibung: Zeigt und ändert Einstellungen von Ethernet-Netzwerkkarten (Treiber-Level): z.B. Geschwindigkeit, Duplex, Auto-Negotiation, Wake-on-LAN, Offload-Funktionen, Statistikzählefile-4fdk7rvx39azjretinarqb】.

Verwendung:

- ethtool <iface> – Zeigt Standard-Eigenschaften: unterstützt Geschwindigkeiten (Supported link modes), aktuelle Speed/Duplex, Auto-negotiation on/off, Ports (TP, etc.), driver name, firmware-version, bus-info, etc.
- ethtool -s <iface> speed <Mb/s> duplex <half/full> autoneg <on/off> – Setzt feste Geschwindigkeit/Duplex. (z.B. ethtool -s eth0 speed 100 duplex full autoneg off). Vorsicht: muss vom Link Partner unterstützt werden, sonst Link down.
- ethtool -k <iface> – Zeigt Kernel-Offload-Funktionen (tx-checksumming, scatter-gather, TSO etc.) und ob an/aus. -K ... (capital K) schaltet diese: z.B. ethtool -K eth0 gro off (GRO ausschalten).
- ethtool -i <iface> – Infos zum Treiber (Driver name, version, firmware, bus).
- ethtool -p <iface> [N] – "Physically identify": Blinkt die LED am Port für N Sekunden (Standard 15), falls Karte das unterstützt. Hilft um Kabelport zu finden.

- ethtool -S <iface> – Ausgabe von Statistikzählern (packets, errors, etc. vom Treiber).
- ethtool -g <iface> – Ring Puffer Größen (RX/TX rings). -G zum Setzen.
- ethtool -a <iface> – Flow-Control (Pause Frame) Status (Autoneg etc.), -A zum Setzen on/off rx/tx.
- ethtool -r <iface> – Veranlasst die Karte, Link neu auszuhandeln (Retrain).
- ethtool --identify <iface> – alias für -p (Blinken).

Beispiele:

- ethtool eth0 -> zeigt z.B. Speed: 1000Mb/s, Duplex: Full, Auto-negotiation: on, etc.
- ethtool -s eth0 speed 100 duplex half autoneg off -> zwingt 100Mbit Half-Duplex (nur Test oder Legacy).
- ethtool -p eth1 10 -> blinkt LED an eth1 für 10 Sekunden.
- ethtool -K eth0 tso off -> schaltet TCP Segment Offloading aus (im Troubleshooting falls Problem mit Offload).

hostnamectl (Control hostname in systemd)

Beschreibung: Zeigt oder ändert den System-Hostname und verwandte Settings (wie Chassis Typ, Icon). hostnamectl ist Teil von systemd.

Verwendung:

- **hostnamectl** (ohne args): Zeigt Systeminfo: Static hostname, Transient (via DHCP), Pretty (human name), und Kernel: Operating System, Kernel, Architecture, etc.
- **hostnamectl set-hostname <name> [--pretty|--static|--transient]** – Setzt den Hostnamen. Ohne Flag wird *static* und *transient* gleichermaßen gesetzt. *static* = in /etc/hostname gespeichert (persistenter Name), *transient* = Laufzeit (kann z.B. DHCP ändern). *pretty* = nur für Anzeigezwecke (z.B. mit Großbuchstaben/Leerzeichen erlaubt).
Beispiel: hostnamectl set-hostname server42 – setzt static+transient auf "server42".
hostnamectl set-hostname "My Home PC" --pretty – setzt Pretty-Name.
- **hostnamectl set-icon-name <icon>** – Setzt Icon (z.B. computer-desktop, rein kosmetisch in einigen GUIs).

- **hostnamectl set-chassis <type>** – Setzt Gehäuse-Typ: desktop, laptop, server, tablet, embedded, etc. (Auch nur informative, kann z.B. in GNOME Systeminfo auftauchen).

Zusatz:

- hostnamectl status (alias ohne args) – Info.
- Traditional hostname Befehl (ohne ctl) zeigt oder setzt nur transient (bis Reboot).

Netzwerk-Fehlerdiagnose und Name-Auflösung

netcat (nc) (TCP/UDP Swiss Army Knife)

Beschreibung: netcat (oder nc) ist ein universelles Netzwerk-Tool, um Verbindungen zu testen, Daten zu senden/empfangen, Ports zu lauschen etcfile-4fdk7rvx39azjretinarqb】. Kann sowohl Client als auch rudimentärer Server sein.

Verwendung:

- **Verbindung als Client:** nc [options] <Host> <Port> – Baut TCP-Verbindung zu Host:Port auf, leitet stdin -> Socket, und Socket -> stdout. So kann man z.B. echo "HEAD / HTTP/1.0" | nc example.com 80 (HTTP-Request senden). Oder nc time.nist.gov 13 (Daytime Protocol). Wenn man interaktiv ausführt, kann man manuell tippen (Telnet-Ersatz).
- **Port Listening (Server):** nc -l -p <Port> [Host] – Horcht auf Port (optional an gegebene IP gebunden). Bsp: nc -l -p 1234 lauscht auf TCP/1234 auf alle Interfaces; was reinkommt wird auf stdout ausgegeben, was man tippt geht an Client.
- **UDP mode:** nc -u – Statt TCP benutzt UDP. Bsp: nc -u 192.168.1.5 53 könnte man UDP Query schicken (z.B. an DNS server).
- **Port Scan:** nc -z -v host startport-endport – "Zero-I/O mode" nur scannen, nicht verbinden. Z.B. nc -z -v 192.168.1.10 1-1024 zeigt offene Ports. (Netcat als Portscanner ist möglich, aber Nmap dafür besser).
- **Timeout:** -w <secs> – Wartezeit bis Timeout für Verbindungen/Eingaben.
- **Keep-open (inetd style):** In GNU nc -k (listen bleibt offen nach Verbindungsende, auf neue).
- **Exec:** -e <prog> – *Unsichere Option* (nicht immer kompiliert, oft disabled aus Sicherheitsgründen): Leitet die I/O eines Programms an den Socket. So kann man z.B. remote shell spawn: nc -l -p 5000 -e /bin/bash. (Equivalent zum berüchtigten rm -rf example). Modern: use socat or ssh for shells.

Beispiele:

- **Chat/Echo Test:** Terminal A: nc -l -p 5000, Terminal B: nc localhost 5000 -> alles was A tippt sieht B und umgekehrt.
- **Datei senden:** Auf Empfänger: nc -l -p 4500 > out.txt, auf Sender: nc destIP 4500 < file.txt. (Netcat baut TCP auf und streamt Datei).
- **HTTP Request:** printf "GET / HTTP/1.1\r\nHost: example.com\r\n\r\n" | nc example.com 80 – zeigt HTML der Homepage (HTTP/1.1 erfordert Host-Header).
- **UDP Ping** (ohne Antwort): echo -n "Hello" | nc -u 192.168.1.5 9000.
- **Port scan:** nc -z -v 192.168.1.20 20-30 -> tries FTP-data, FTP etc, output open/closed.

(Netcat Implementierungen: Traditional vs OpenBSD netcat vs GNU netcat ncat (part of nmap) – Optionen leicht variierend. Z.B. OpenBSD netcat hat -l und -p zusammen als one option or so, etc.)

iftop (Interface TOP, Bandwidth monitor)

Beschreibung: Ein curses-basiertes Tool zum Anzeigen der aktuellen Bandbreitennutzung pro Verbindung (ähnlich top, aber für Netzwerk Trafficfile-4fdk7rvx39azjretinarqb】). Es zeigt welche Peers mit dem System kommunizieren und wieviel Bytes/s in/out. Hilfreich zum realtime Debugging wer Bandbreite frisst.

Verwendung:

iftop -i <Interface>

Ohne -i wählt iftop das erste externe Interface.

Optionen:

- **-n** – Keine DNS-Auflösung (zeigt IPs statt aufzulösen, wichtig um Ausgabe schnell zu haltenfile-4fdk7rvx39azjretinarqb】).
- **-N** – Keine Port-Auflösung (zeigt Portnummern statt Services Namen).
- **-P** – Zeigt Ports zusammen mit Host (separat Spalte). Kombiniert man oft: iftop -nP.
- **-B** – Bytes statt Bits in Anzeige (Standard iftop zeigt in bits/s).
- **-F <FilterNet>** – Nur Traffic, der in bestimmten Netzbereich fällt (Filter) betrachten, z.B. -F 192.168.0.0/16 filtert auf diese Range.
- **-p** – Promiscuous Mode (auch Traffic anzeigen, der nicht an lokalem Host endet? Normal iftop Filter host involvement).
- **-t** – Text-Only Modus (keine curses-Interface, nur in stdout Stats; nicht sehr üblich).

Im Interface kann man mit Tasten: h Hilfe, n togg DNS, s sort by source, d sort by dest, t sort by throughput, p toggle port display, b toggle bar graph, B cycles through rate intervals (default 2s,10s,40s windows).

traceroute / tracepath (Route Tracking)

Beschreibung: Sendet eine Serie von Paketen mit steigenden TTL (Time-to-Live) Werten, um den Pfad (Router Hopps) zum Ziel auszumessefile-4fdk7rvx39azjretinarqb】.

Traceroute verwendet standard UDP oder ICMP ECHO, tracepath nutzt UDP with high port (no root needed). Jedes Hop sendet "Time Exceeded" zurück, das traceroute protokolliert. Ergebnis: Liste der Router vom Quell- zum Zielnetz.

Syntax:

```
traceroute [Optionen] <Host> [PacketSize]
```

```
tracepath <Host>
```

Optionen (traceroute):

- **-I:** ICMP Echo statt UDP verwenden (pakete wie ping).
- **-T:** TCP SYN statt UDP (z.B. um Firewalls zu durchdringen über port 80).
- **-p <Port>:** UDP port Start (Standard 33434).
- **-q <N>:** Anzahl Probes pro TTL (Default 3).
- **-w <Seconds>:** Timeout pro Hop Antwort (default 5s).
- **-n:** Nicht auflösen (IP statt DNSfile-4fdk7rvx39azjretinarqb】).
- **-m <max_ttl>:** Max Hop Count (Standard 30).
- **-f <first_ttl>:** Start TTL (default 1). Manchmal hilfreich, mittendrin einzusteigen.
- **-4 / -6:** Force IPv4 / IPv6.

Optionen (tracepath) sind wenige: -n no dns, -b show both hostnames and IP, etc.

Beispiel:

```
$ traceroute example.com
```

```
1 192.168.1.1 (192.168.1.1) 1.123 ms 0.846 ms 0.799 ms
2 100.64.0.1 (100.64.0.1) 10.123 ms 9.876 ms 9.543 ms
3 203.0.113.5 (203.0.113.5) 15.432 ms 14.876 ms 15.210 ms
4 ae12.dar02.example.net (198.51.100.1) 30.456 ms 29.999 ms 30.123 ms
5 93.184.216.34 (93.184.216.34) 35.678 ms 35.900 ms 35.456 ms
```

Zeigt 5 Hops (Heimrouter, CGNAT router, ISP core, example.net router, Ziel).

tracepath example.com ähnlich, oft mit asymmetrischen result such as pmtu info.

mtr (My Traceroute)

Beschreibung: Kombiniert traceroute und ping in einem kontinuierlichen Diagnose-Tool. mtr sendet fortlaufend Pakete über alle Hops und aktualisiert in Echtzeit die Latenz und Packetloss-Statistiken pro Hop. Hilfreich um instabile Verbindungen/Packet Loss auf bestimmten Strecken zu identifizieren.

Verwendung:

mtr [Optionen] <Ziel>

Interaktiv curses UI standard, oder mit -r (report) Modus für einmalige Ausgabe.

Optionen:

- **-4 / -6:** Force IPv4/IPv6.
- **-n:** Nicht auflösen, nur IPs (schneller).
- **-w:** Wide-output (zeigt beide host+IP columns).
- **-c <count>:** Limit auf <count> Probezyklen, dann beenden (für Report).
- **-r:** Report-Modus (nicht curses, druckt Tabelle und endet).
- **-i <sec>:** Sekunden zwischen Probe-Paketen (default 1.0).
- **-p:** Pausiert Ausführung beim Start (Press enter to start).
- **-t:** Text Mode (like curses but in dumb terminal).
- **-u:** UDP mode (Standard mtr uses ICMP Echo, as -l in traceroute terms). -T for TCP mode also available.

Anzeige:

MTR listet Hops fortlaufend wie traceroute, plus Spalten: Loss%, Snt (packets sent), Last (ms last latency), Avg, Best, Wrst (worst), StDev. Es aktualisiert kontinuierlich.

Beispiel-Zeile:

3. 203.0.113.5 0.0% 10 15.4 16.1 14.8 20.3 1.3

ping (ICMP Echo)

Beschreibung: Schickt ICMP Echo-Request-Pakete an eine Adresse und misst die Antwortzeiten (Echo-Reply), um Erreichbarkeit und Round-Trip-Time festzustellen.

Syntax:

ping

(Fortsetzung)

ping (ICMP Echo Requests)

Beschreibung: `ping` prüft die Erreichbarkeit eines Hosts im Netzwerk mit **ICMP-Echo-Paketen** und misst die Round-Trip-Zeit (RTT) der Antwort. Es sendet fortlaufend Anfragen (standardmäßig, bis abgebrochen) und berechnet Statistik über Verlust und Latenz.

Syntax:

``` bash

ping [Optionen] <Ziel>

Hier kann <Ziel> ein Hostname oder eine IP-Adresse sein.

#### Wichtige Optionen:

- **-c <Anzahl>** – Anzahl der ICMP-Pakete festlegen. Z.B. ping -c 4 8.8.8.8 sendet genau 4 Pakete.
- **-i <Sekunden>** – Interval zwischen den Pings (Standard 1 Sekunde). Man kann z.B. -i 0.2 für fünf Pings pro Sekunde (root-Recht nötig bei <0.2).
- **-W <Timeout>** – Wartezeit in Sekunden auf eine Antwort, bevor als **timeout** gezählt (Default meist 1 Sekunde, bei -c am Ende kann man höher setzen).
- **-q** – "Quiet": keine per-Ping Ausgabe, nur eine Zusammenfassung am Ende.
- **-p <Pattern>** – Sende individuelles Byte-Muster im Paket (Hexadezimal angegeben). Gut für bestimmte Diagnosen (z.B. bestimmte Bits toggeln sehen).
- **-s <Packetsize>** – Größe des zu sendenden Pakets in Bytes (ohne ICMP-Header). Standard 56 Bytes (plus 8 Byte ICMP-Header = 64 Bytes in Ausgabe).
- **-t <TTL>** – Setzt die Time-To-Live (IPv4) oder Hop Limit (IPv6) für die gesendeten Pakete. (Nützlich, um gezielt nur bis zu einem gewissen Hop zu gelangen, ähnlich traceroute Step).
- **-I <Interface>** – Bindet an ein bestimmtes Quell-Interface (oder IP) für Multi-homed hosts. Alternativ **-S <Source\_Address>**.
- **-4, -6** – Erzwingt IPv4 oder IPv6.
- **-f** – "Flood ping": schickt so schnell wie möglich Pings (sehr viele pro Sekunde) und druckt einen Punkt pro empfangenes Echo. (Nur root, zur Stresstest oder schnelle Packetloss Erkennung).

- **-L** (bei einigen ping-Versionen) – Für Echo an Broadcast/Multicast-Adresse (ermöglicht Broadcast-ping, was standardmäßig blockiert ist).
- **-a** – Bei Empfang einen Terminal-Bell ausgeben (akustisches Signal).
- **-D** – Zeitstempel vor jeder Zeile (UNIX time).
- **-U** – Zeitstempel in Us seit Boot vor jeder Zeile (für Messungen).

### **Ausgabe-Interpretation:**

Jede empfangene Antwort erscheint als Zeile:

64 bytes from 8.8.8.8: icmp\_seq=1 ttl=117 time=8.34 ms

Bedeutet: Echo-Reply von 8.8.8.8 erhalten, Größe 64 Bytes, Sequenznummer 1, TTL im Reply 117, RTT 8.34 ms. Wenn Antworten ausbleiben, erscheinen Zeilen wie "Request timeout".

Am Ende kommt eine **Statistik**:

4 packets transmitted, 4 received, 0% packet loss, time 3005ms

rtt min/avg/max/mdev = 8.334/8.512/8.790/0.177 ms

Zeigt Verlust und RTT-Werte.

### **Beispiele:**

- ping 1.1.1.1 – Kontinuierlicher Test zu 1.1.1.1 (Cloudflare DNS). Abbruch mit Strg+C.
- ping -c 5 www.example.com – 5 Pings an example.com, dann stop mit Ausgabe der Statistik.
- ping -l eth1 192.168.2.1 – Ping von Interface eth1 aus (wenn Multi-Interface).
- ping -s 1000 -c 3 host – Sende 3 Pings mit Paketgröße ~1000 Bytes (ICMP Payload).
- ping -6 -c 4 ipv6.google.com – IPv6 Ping zu Google.

### **traceroute (siehe oben)**

(bereits behandelt in vorangehendem Teil, siehe „traceroute“ in Netzwerkdagnostik.)

### **tracepath (siehe oben)**

(siehe „tracepath“ bei Netzwerkdagnostik.)

### **resolvectl (Systemd-resolved Query)**

**Beschreibung:** CLI-Tool zum Abfragen des systemd-resolved-Dienstes (Namensauflösung). resolvectl kann DNS-Lookups durchführen, DNS-Settings anzeigen, LL/MNR Browsing etc. Ersetzt Tools wie systemd-resolve (veraltet).

### Verwendung:

- **resolvectl status** – Zeigt die aktuellen DNS-Einstellungen pro Link (Interface): DNS-Server, Suche-Domains, LLMNR/MulticastDNS Status, DNSSEC, usw.
- **resolvectl query <Hostname>** – Löst einen Hostnamen auf (A/AAAA-Record). Ausgabe enthält v6 und v4 Adressen, CNAME falls, DNSSEC info, Cache info, antwortender Server etc.  
Beispiel: resolvectl query example.com.
- **resolvectl query -t MX <Domain>** – DNS-Abfrage bestimmten Typs (hier MX-Records). Unterstützt -t ANY, A, AAAA, MX, SRV, SOA etc.
- **resolvectl query <IP>** – Reverse Lookup (IP zu Name via PTR).
- **resolvectl flush-caches** – Löscht den DNS-Cache von systemd-resolved.
- **resolvectl dns <Interface> [Server...]** – Setzt DNS-Server für ein bestimmtes Interface (über DBus). resolvectl domain <iface> ... analog. (Normal wird das via NetworkManager/Netplan/etc gemacht).
- **resolvectl revert <Interface>** – Löscht manuelle Overrides und kehrt zu vom Netzwerkdienst gelieferten DNS zurück.

### Beispiel:

```
$ resolvectl query -t MX example.com
example.com IN MX 0 .
-- Information acquired via protocol DNS in 15.1ms.
```

-- Data is authenticated: no

Bedeutet example.com hat keinen MX (Mail).

resolvectl status Ausschnitt:

Global

Protocols: LLNR=resolve mDNS=no DNSOverTLS=no DNSSEC=no/unsupported

resolv.conf mode: stub

Link 2 (eth0)

Current Scopes: DNS

Protocols: +DefaultRoute +LLNR -mDNS DNSOverTLS=opportunistic DNSSEC=...

Current DNS Server: 192.168.1.1

DNS Servers: 192.168.1.1 1.1.1.1

DNS Domain: ~.

Zeigt z.B. that eth0 hat default DNS 192.168.1.1 plus Cloudflare, und ~. (default domain).

## **dig (DNS Lookup Utility)**

**Beschreibung:** dig (domain information groper) führt DNS-Abfragen aus. Es ist Teil von BIND-Tools (bind-utils). Im Gegensatz zu host oder nslookup bietet dig ausführlich formatierbare Antworten. Standard macht eine A-Record-Abfrage und gibt Antwort mit Abschnitt **ANSWER SECTION** aus.

### **Syntax:**

dig [@DNS-Server] <Name> [<Typ>] [<Klasse>] [Optionen]

<Klasse> normalerweise IN (Internet), selten was anderes.

### **Wichtige Optionen:**

- **@server** – Spezifischen DNS-Server benutzen statt System-resolver. Z.B. dig @8.8.8.8 example.com.
- **<Typ>** – Abfragetyp: A, AAAA, MX, TXT, SRV, NS, SOA, CNAME, PTR, ANY etc. (Default A).
- **+short** – Ausgabe nur die Antworten knapp (einfacher, scripterfreundlic0】 . Z.B. dig +short example.com A -> "93.184.216.34".
- **+noall +answer** – Unterdrückt alle außer Answer-Section (ähnlich short aber mit TTL und Info).
- **+trace** – Iterativer Trace durch DNS-Hierarchie (Root bis Ziel, ähnlich dig an Root und dann NS referrals folgen). Zeigt jeden Schritt. Hilfreich zum Debug.
- **+tcp** – Erzwinge TCP statt UDP (z.B. bei großen DNS-Records oder EDNS issues).
- **+nssearch** – Findet alle NS des Ziel und fragt diese nach SOA (um Delegation correctness zu checken).
- **-x <IP>** – Reverse lookup (PTR) für IP. (Bequem: dig -x 8.8.8.8).
- **+multi** – Schöne Ausgabe in Spalte falls mehrere TXT oder so.
- **+nodnssec oder +dnssec** – Anfordern oder nicht (um RRSIG etc. zu sehen).
- **+ttlshort** – Output TTL in short mode. (e.g. dig +short +ttl google.com -> "300 172.217.168.14").

- **+stats / +nostats** – Schaltet Statistik-Banner am Ende an/aus.
- **+cm** – Colorize output (if compiled with color support).

### **Beispiele:**

- dig example.com ANY – Fragt alle Records (die server je nach config liefern).
- dig google.com MX +noall +answer – Zeigt nur MX Antworten.
- dig @1.1.1.1 cloudflare.com A – Abfrage an 1.1.1.1 nach cloudflare's A.
- dig -x 8.8.4.4 +short – Reverse Lookup, kurze Ausgabe ("dns.google").
- dig +trace example.org – Startet vom Root Nameserver und verfolgt Delegation bis example.org resolved ist.

### **nslookup (Name Server Lookup - veraltet)**

**Beschreibung:** Ein einfaches interaktives DNS-Abfrageprogramm. War Teil von BIND, gilt als veraltet im Vergleich zu dig. Viele Admins kennen es dennoch.

### **Verwendung:**

nslookup [<Host> [<DNS-Server>]]

Ohne args geht es in interaktiven Modus. Dessen Kommandos: server <DNS>, set q=MX, set type=AAAA, set debug, exit.

### **Beispiele:**

```
$ nslookup example.com 8.8.8.8
```

Server: 8.8.8.8

Address: 8.8.8.8#53

Non-authoritative answer:

Name: example.com

Address: 93.184.216.34

Reverse: nslookup 93.184.216.34 -> zeigt PTR.

**Warnung:** Neuere doc empfehlen dig oder host statt nslookup.

### **whois (Domain Whois Query)**

**Beschreibung:** Fragt Whois-Datenbanken nach Registrierungsinformationen einer Domain oder 5】. Das Tool verbindet sich zum passenden Whois-Server (verwendet /etc/whois.conf, TLD heuristics) via TCP 43 und gibt den Text aus.

## Syntax:

whois [Optionen] <DomainOrIP>

**Ausgabe:** Je nach TLD und Registry differierend. Enthält Domain-Inhaber (bei Generic oft DSGVO anonymisiert), Registrar, Erstellungs- und Ablaufdatum, Nameserver, Contact-IDs. Bei IP WHOIS (z.B. whois 8.8.8.8) gibt es Informationen über den zugeteilten IP-Block (ISP, RIR wie ARIN/RIPE...).

## Optionen:

- **-H** – Unterdrückt das Banner "Whois Server" in RIPE responses.
- **-p <port>** – Andere Port.
- **--verbose** – Mehr Info über Server selection.
- **--help** – Display help.

## Beispiele:

- whois example.com -> zeigt Registrar (ICANN info), Status, Name der Organisation, etc.
- whois 2620:0:2d0:200::7 -> IPv6 whois (gibt i.d.R. ARIN oder so mit OrgName).
- whois AS15169 -> WHOIS an RADb für Autonomous System 15169 (Google).

(*Beachte: Manche TLDs haben separate Tools, z.B. jwhois config. Das Standard whois ist meist in Debian whois package (Marco d'Itri) which covers a lot. GDPR hat Personendaten stark eingeschränkt in Output.*)

**arp (Address Resolution Protocol table) – Hinweis: arp wird mittlerweile durch ip neigh ersetzt.**

**Beschreibung:** Zeigt den ARP-Cache (IPv4) an oder manipuliert Einträge. (Linux modern: use ip neigh aber arp Kommando aus net-tools ist noch gebräuchlich).

## Befehle:

- arp -a – Listet ARP-Einträge (Hostname, IP, MAC, Interface).
- arp -n – Wie -a, aber keine DNS für IP.
- arp -d <IP> – Löscht ARP-Eintrag für IP.
- arp -s <IP> <MAC> – Fügt statischen ARP-Eintrag hinzu (MAC fix für IP) – gleich ip neigh add ... nud permanent.

*Beispiel:* arp -a -> router.lan (192.168.1.1) at aa:bb:cc:dd:ee:ff [ether] on eth0.

## Sicherheit und Firewall

## **iptables/iptables-nft (Packet Filter)**

**Beschreibung:** Linux Kernel-Paketfilter (Firewall) via Netfilter-Schnittstelle. iptables verwaltet Regeln in verschiedenen Tabellen: **filter**, **nat**, **mangle**, **raw**, **security**, für IPv4. (Für IPv6, ip6tables). Mit Kernel 4.x/5.x viele Distros nutzen "iptables-nft" as a frontend to nftables, aber Syntax bleibt für user gleich.

### **Syntax Grundkommandos:**

`iptables [-t <table>] <operation> <chain> <matches...> <target>`

Operationen: **-A** (append), **-I** (insert), **-D** (delete), **-C** (check exist), **-L** (list), **-F** (flush chain), **-N** (new chain), **-X** (delete user chain).

Standard Tabellen:

- **filter** (Default): Chains **INPUT**, **OUTPUT**, **FORWARD**.
- **nat**: Chains **PREROUTING**, **OUTPUT**, **POSTROUTING** (masquerading, port forwarding etc.).
- **mangle**: For packet alteration (TOS, mark), Chains in multiple points (PREROUTING, OUTPUT, etc.).
- **raw**: For conntrack exemption (NOTRACK), Chains PREROUTING, OUTPUT.
- **security**: SELinux related (rare).

### **Matches / Options:**

- **-p <proto>**: Protocol (tcp, udp, icmp, icmpv6, all).
- **-s <IP/CIDR>, -d <IP/CIDR>**: Source or destination address (or network).
- **-i <iface>, -o <iface>**: Inbound or outbound network interface (for PREROUTING/INPUT/forward vs OUTPUT/POSTROUTING).
- **--sport, --dport** (with -p tcp/udp): Source/dest port or port range (e.g. --dport 80 or --dport 1000:2000).
- **-m state --state <states>**: State match (connection tracking states: INVALID, ESTABLISHED, NEW, RELATED). Common: allow ESTABLISHED,RELATED on INPUT for responses.
- **-m conntrack --ctstate**: (newer usage, same as state module now).
- **-m connlimit, -m limit** etc: Many modules. limit --limit 5/minute for rate-limit logging for example. connlimit --connlimit-above N limit parallel connections.
- **-m multiport**: match multiple ports in one rule if not contiguous, e.g. -m multiport --dports 80,443,8080.

- **-m tcp --tcp-flags SYN,RST,ACK SYN**: match specific TCP flags combination (here SYN only). Often used for --syn (alias) which matches syn packets (to identify new connections).
- **\*\*-m comment --comment "text"**: attach comment to rule (for documentation).

### Targets (Actions):

- **ACCEPT**: Packet erlaubt.
- **DROP**: Verwerfen ohne Antwort.
- **REJECT**: Verwerfen mit Antwort (ICMP port unreachable o.Ä.).
- **LOG**: Loggt Paket (via kernel log), dann Verarbeitung geht weiter (so typ. Kombi LOG dann DROP). Options: --log-prefix "msg", --log-level warning.
- **MASQUERADE** (in nat POSTROUTING): SNAT für dynamische IP (Router Use-case) – ersetzt Source-IP durch Interface-IP.
- **SNAT** (in nat POSTROUTING): Statisches Source-NAT (requires --to-source IP[:port-range]).
- **DNAT** (in nat PREROUTING/OUTPUT): Destination-NAT (Port-Forwarding; --to-destination IP:port).
- **REDIRECT** (nat PREROUTING/OUTPUT): Umleiten an lokalen Host (for transparent proxy etc.).
- **MARK** (mangle): Markiert Paket mit fwmark (für routing decisions via ip rule, etc.).
- **RETURN**: In user-defined chain springt zurück zum Aufrufer-Chain (stop further rules in current chain). In builtin chain, RETURN = apply chain policy.
- **QUEUE**: Übergibt Paket an Userspace (via libnetfilter\_queue).
- **DROP** vs. policy: If chain policy is DROP, falling off end = drop.

### Beispiele:

- **List Regeln**: iptables -L -n -v (filter table). iptables -t nat -L -n -v NAT rules. -n no DNS, -v verbose (packet counters).
- **Allow incoming SSH**:

```
② iptables -A INPUT -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
(typical stateful allow SSH in).
```

② **Drop all forward by default**: iptables -P FORWARD DROP.

## ② **Masquerading (NAT):**

② `iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE`

(for home router with ppp0 outbound).

## ② **Port Forward:**

Forward ext eth0 port 80 to internal 192.168.1.100 port 8080:

```
② iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 \
-j DNAT --to-destination 192.168.1.100:8080
```

```
iptables -A FORWARD -p tcp -d 192.168.1.100 --dport 8080 -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT
```

## ② **Logging Drop:**

- `iptables -A INPUT -p tcp --dport 23 -j LOG --log-prefix "Telnet attempt: " --log-level info`
- `iptables -A INPUT -p tcp --dport 23 -j DROP`
- (log then drop telnet).

## **firewall-cmd (firewalld frontend)**

**Beschreibung:** Steuerprogramm für **firewalld**, eine dynamische Firewall (standard in RHEL/CentOS 7+, Fedora, etc.). firewall-cmd ermöglicht das Konfigurieren von Zonen, Regeln (services, ports, masq) zur Laufzeit und in Config. Firewalld nutzt intern iptables/nftables.

### **Syntax:**

```
firewall-cmd [--zone=<zone>] [--permanent] <aktion> <Parameter>
```

**Grundkonzepte:** Zonen (trusted, public, home, internal, work, external, dmz, block, drop etc.), Schnittstellen sind Zonen zugewiesen. Dienste (vordefinierte Regelsets) können pro Zone erlaubt werden oder einzelne Ports/Protokolle. Man kann Einstellungen permanent (persistiert in XML config) oder nur runtime vornehmen.

### **Wichtige Befehle:**

- **--state:** Zeigt an, ob firewalld läuft (running/not running).
- **--get-active-zones:** Zeigt welche Zonen aktiv sind und welche Interfaces zugeordnet (z.B. "public: wlp2s0").
- **--get-zones:** Listet alle bekannten Zonen. **--list-all** [--zone=Z] listet alle Einstellungen einer Zone (services, ports, masquerade etc0) .

- **--zone=<zone> --add-service=<service>**: Erlaubt den vordefinierten service in angegebener Zone (runtime). Bsp: firewall-cmd --zone=public --add-service=http – ermöglicht HTTP (Port 80) in Zone *public*.
- **--zone=<zone> --add-port=<port>/<proto>**: Erlaubt individuellen Port. Bsp: --add-port=5000/tcp.
- **--remove-service=..., --remove-port=...** analog zum Entfernen.
- **--add-masquerade**: Aktiviert NAT/Masquerading in der Zone (z.B. für Internet-sharing zone).
- **--query-service=<svc>**: Fragt ob Service in zone erlaubt (exit code 0/1).
- **--permanent**: Gibt man diese Option an, wird die Änderung in der persistenten Konfiguration vorgenommen (aber nicht an laufende sofort angewendet, es sei denn man ruft extra an). In firewalld, runtime und permanent config sind getrennt.
- **--reload**: Lädt die permanente Konfiguration neu (in runtime) – im Grunde wendet Config an.
- **--runtime-to-permanent**: Speichert aktuellen Laufzeit-Zustand als neue permanente Konfiguration (was man on-the-fly geändert hat, persistieren).
- **--zone=<zone> --change-interface=<iface>**: Weist ein Interface einer Zone zu (persistente Zuordnung).
- **--new-zone=<name>, --delete-zone=...**: Eigene Zonen erstellen/löschen.
- **--list-services, --get-services**: Listet bekannte Service-Namen (die in /etc/firewalld/services/\*.xml definiert sind).
- **--service=<svc> --get-ports**: Zeigt welche Ports hinter einem Service stecken.
- **--add-rich-rule '<rule>'**: Fügt sog. Rich Rule hinzu (eine erweiterte Regel-Syntax in firewalld, z.B. mit Logging, Zeitplänen). Z.B.:
  - firewall-cmd --add-rich-rule='rule family="ipv4" source address="192.0.2.0/24" port port="22" protocol="tcp" accept'
  - (Erlaubt SSH nur von 192.0.2.0/24).

### **Beispiele:**

- Öffnen von HTTPS permanent:
- `firewall-cmd --add-service=https --permanent`
- `firewall-cmd --reload`
-

- Port 12345/udp temporär öffnen in Zone public:  
firewall-cmd --zone=public --add-port=12345/udp (nach Reboot weg).
- Interface eth0 zur Zone external zuweisen:  
firewall-cmd --zone=external --change-interface=eth0 --permanent && firewall-cmd --reload.
- Status: firewall-cmd --list-all (wenn default zone, z.B. public, dann deren config).

## **ufw (Uncomplicated Firewall)**

**Beschreibung:** Vereinfachtes Firewall-Frontend (Ubuntu/Debian) für iptables. ufw erlaubt einfache allow/deny-Regeln definieren, typischerweise nach Services oder Ports, optional in/out interface.

### **Syntax:**

ufw [Optionen] <Aktion>

### **Grundbefehle:**

- **ufw enable / disable** – Aktiviert oder deaktiviert die Firewall (bei disable: setzt iptables default ACCEPT und löscht Regeln).
- **ufw status [verbose]** – Zeigt aktuelle Regel-Liste. Mit verbose sieht man auch Interfaces, logging status etc.
- **ufw default allow|deny [incoming|outgoing|routed]** – Setzt Standardpolicy. Standard-Einstellung: incoming deny, outgoing allow.
- **ufw allow <Port/Service> [proto tcp/udp] [from <IP> [to <IP>]] [port <P2>]** – Erlaubt eingehenden Traffic. Beispiele:
  - ufw allow 22 (erlaubt TCP/22 auf allen Interfaces).
  - ufw allow proto udp from 10.0.0.0/8 to any port 53 (erlaubt UDP DNS von privatem Netz).
  - ufw allow out 80/tcp (erlaubt ausgehende HTTP).
  - ufw allow in on eth0 to any port 443 (erlaubt eingehendes HTTPS auf eth0).
- **ufw deny <Spec>** – Blockiert (mit REJECT) entsprechend.
- **ufw reject <Spec>** – Synonym zu deny (deny und reject meist ähnlich in ufw; auf Debian might differ).
- **ufw limit <Port/Service>** – Wie allow, aber mit Rate-Limit (gegen Bruteforce). Z.B. ufw limit ssh erlaubt SSH, aber blockiert IP temporär bei >6 Verbindungen/30s.

- **ufw delete <RuleSpec>** – Entfernt Regel (man kann Regel genau wie bei allow angeben, mit 'delete' davor, oder Nummer aus ufw status numbered).
- **ufw logging on|off|LEVEL** – Logging ein/aus und Level (off, low, medium, high, full).
- **ufw reset** – Setzt ufw auf Auslieferungszustand (alle Regeln weg, policies reset, und disabled).
- **ufw route allow/deny ...** – Regeln für Forwarded Traffic (Routen). Bsp: ufw route allow in on eth1 out on eth0 to any port 80 for forwarded.
- **ufw insert <num> <rule>** – Fügt Regel an bestimmter Position ein.
- **ufw show raw** – Zeigt resultierende iptables-Regeln.

### Beispiele:

- Standard: ufw default deny incoming; ufw default allow outgoing; ufw enable.
- Webserver: ufw allow 'Nginx Full' (in /etc/ufw/applications.d vordefinierte Profile, 'Apache Full' etc., Nginx Full = 80+443).
- Only LAN access to a service: ufw allow from 192.168.0.0/16 to any port 3306 (MySQL offen nur intern).
- Block eine IP: ufw deny from 203.0.113.5.
- Show rules: ufw status verbose -> e.g.
- Status: active
- Logging: on (low)
- Default: deny (incoming), allow (outgoing), deny (routed)
- New profiles: skip
- 
- To Action From
- -- ----- -----
- 22/tcp LIMIT Anywhere # SSH
- 80,443/tcp ALLOW Anywhere # WWW
- 22/tcp (v6) LIMIT Anywhere (v6)
- 80,443/tcp (v6) ALLOW Anywhere (v6)
-

*(UFW ist beliebt wegen einfacher Syntax, aber fehlender Komplexität bei komplexen Setups. Firewalld und raw iptables/nft sind mächtiger in Feinheiten.)*

### **nft (nftables Firewall)**

**Beschreibung:** nftables ist der modernere Packet-Filter (ersetzt iptables). Der Befehl nft wird zum Konfigurieren verwendet. Sehr flexibel (Datenstrukturen, one-rule multiple matches).

#### **Syntax:**

```
nft [options] <command>
```

Man kann Kommandos interactive eingeben oder als Skript. Der Kommandostil ähnelt C:  
e.g.

```
nft add table inet filter
```

```
nft add chain inet filter input { type filter hook input priority 0 \; }
```

```
nft add rule inet filter input tcp dport 22 accept
```

(Librally, but given complexity and that the original question references iptables and ufw, likely beyond scope).

### **Abschluss**

Dieses Kompendium deckt die wesentlichen Befehle für die Linux+ (XK0-005) Prüfung ab, gegliedert nach Themengebieten. Die Befehle umfassen Benutzer- und Rechteverwaltung, Prozess- und System-Tools, Netzwerk-Konfiguration und -Diagnose, Software-Installation, Container, Sicherheit (Firewall, SELinux/AppArmor siehe oben Rich-Rules, etc.), Dienste und Planung.

Durch die strukturierte Darstellung mit Syntax, Beschreibung, Optionen und Beispielen dient es als umfassendes Nachschlagewerk. **Wichtig:** In einer Prüfungssituation sollte man sich auf die korrekte Anwendung der Optionen und das Verständnis der Konzepte dahinter konzentrieren (z.B. was macht chmod 775 oder umask 002 bewirkt).

Mit dieser Referenz sollte die Vorbereitung auf typische Linux-Aufgaben – vom Anlegen eines Benutzers, dem Ändern von Dateirechten, über das Einrichten eines Cronjobs bis zum Konfigurieren eines Apache-Dienstes oder Setzen einer Firewallregel – bestens unterstützt werden.

ChatGPT kann Fehler machen. Überprüfe wichtige Informationen. Siehe Cookie-Voreinstellungen.

### **Vorschau**

## **Kompendium der Linux+ (XK0-005) Befehle**

Dieses Nachschlagewerk umfasst alle wichtigen Befehle aus dem offiziellen **Command Reference Guide** für die CompTIA Linux+ Prüfung (XK0-005). Die Befehle sind thematisch gegliedert (Benutzerverwaltung, Netzwerk, Prozesse, Dateisystem, Paketmanagement, Container, Sicherheit etc.), mit Beschreibung, vollständiger Syntax, allen verfügbaren Optionen und ggf. wichtigen Unterbefehlen. Es dient als umfassendes Cheat-Sheet zur Prüfungsvorbereitung.

## Grundlegende Linux-Befehle

### ls (List Directory Contents)

**Beschreibung:** Listet den Inhalt eines Verzeichnisses auf. Ohne Parameter zeigt ls die nicht versteckten Einträge im aktuellen Verzeichnis in alphabetischer Reihenfolge[wiki.ubuntuusers.deionos.de](#).

#### Syntax:

ls [OPTION]... [VERZEICHNIS]

*Wenn kein Verzeichnis angegeben wird, wird das aktuelle Verzeichnis aufgelistet[wiki.ubuntuusers.de](#).*

#### Optionen:

| Option                      | Beschreibung                                                                                                                                 |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-a, --all</b>            | Zeigt <i>alle</i> Einträge an, <b>inklusive</b> versteckter Dateien (auch . und ..) <a href="#">wiki.ubuntuusers.deionos.de</a> .            |
| <b>-A, --almost-all</b>     | Wie -a, aber ohne . und .. – listet versteckte Dateien außer den Verzeichnissen ./.. <a href="#">wiki.ubuntuusers.deionos.de</a> .           |
| <b>-l</b>                   | Langformat: Detaillierte Ausgabe (Rechte, Besitzer, Größe, Datum usw.) je Eintrag <a href="#">wiki.ubuntuusers.deionos.de</a> .              |
| <b>-d, --directory</b>      | Listet Verzeichniseintrag selbst statt dessen Inhalt (nützlich, um Infos über Ordner zu sehen) <a href="#">wiki.ubuntuusers.deionos.de</a> . |
| <b>-h, --human-readable</b> | Im Langformat (-l) Größen in menschenlesbare Einheiten (KB, MB ...) anzeigen <a href="#">wiki.ubuntuusers.deionos.de</a> .                   |
| <b>-i</b>                   | Zeigt die Inode-Nummer jedes Eintrags an <a href="#">wiki.ubuntuusers.deionos.de</a> .                                                       |
| <b>-C</b>                   | Ausgabe in Spalten nebeneinander<br>(Standardformat) <a href="#">wiki.ubuntuusers.deionos.de</a> .                                           |
| <b>-m</b>                   | Listet Einträge kommasepariert in einer Zeile <a href="#">wiki.ubuntuusers.deionos.de</a> .                                                  |

| <b>Option</b>          | <b>Beschreibung</b>                                                                                                                                              |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-R, --recursive</b> | Rekursiv: Zeigt auch Inhalte von Unterverzeichnissen (durchläuft Verzeichnisbaum) <a href="http://wiki.ubuntuusers.de/onos.de">wiki.ubuntuusers.deionos.de</a> . |
| <b>-t</b>              | Sortiert nach Zeitstempel (Änderungszeit, neueste zuerst) anstatt alphabetisch.                                                                                  |
| <b>-r</b>              | Kehrt die Sortierreihenfolge um (Reverse order).                                                                                                                 |
| <b>--hyperlink</b>     | Macht Dateinamen (in unterstützten Terminals) als klickbare Links ausgebbar <a href="http://wiki.ubuntuusers.de">wiki.ubuntuusers.de</a> .                       |
| <b>--help</b>          | Zeigt die Hilfe zum Befehl an.                                                                                                                                   |
| <b>--version</b>       | Zeigt Versionsinformationen von ls.                                                                                                                              |

(Weitere Optionen sind in den Manpages dokumentiert[wiki.ubuntuusers.de](http://wiki.ubuntuusers.de).)

### **cat (Concatenate and Print Files)**

**Beschreibung:** Gibt den Inhalt von Dateien auf der Standardausgabe aus oder verknüpft (konkateniert) mehrere Dateien. Oft verwendet, um Textdateien anzuzeigen oder Inhalte zu pipen[ionos.at](http://ionos.at). Der Name steht für „concatenate“ (verketten).

#### **Syntax:**

cat [OPTION]... [DATEI]...

*Ohne Optionen schreibt cat den Inhalt der angegebenen Datei(en) in der angegebenen Reihenfolge auf die Standardausgabe. Kann mehrere Dateien hintereinander ausgeben oder mit Umleitungen/ Pipes genutzt werden.*

#### **Optionen:**

### **Option Beschreibung**

|           |                                                                                                                                                |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-n</b> | Nummeriert <i>alle</i> ausgegebenen Zeilen fortlaufend <a href="http://ibm.comionos.at">ibm.comionos.at</a> .                                  |
| <b>-b</b> | Nummeriert nur <b>nicht-leere</b> Ausgabelinien (Leerzeilen werden nicht nummeriert) <a href="http://ibm.comionos.at">ibm.comionos.at</a> .    |
| <b>-s</b> | Squeeze: Unterdrückt aufeinanderfolgende Leerzeilen (mehrere leere Zeilen werden als eine ausgegeben) <a href="http://ionos.at">ionos.at</a> . |
| <b>-v</b> | “Visible” – Macht nicht druckbare Steuerzeichen sichtbar (außer Tab und Zeilenende) <a href="http://ionos.at">ionos.at</a> .                   |

## Option Beschreibung

- e      Entspricht -v mit Anzeige des Zeilenende-Zeichens \$ am Zeilenende (kombiniert -v und Kennzeichnung von Zeilenumbrüchen)[ionos.at](#).
- t      Entspricht -v mit Darstellung von Tabulatoren als ^I (Tab-Markierung)[ionos.at](#).
- E      Zeigt \$ am Ende jeder Zeile (Äquivalent zu -vE, oft identisch zu -e je nach Implementierung).
- T      Zeigt Tabulatoren als ^I an (ähnlich -vT, meist wie -t).
- A      Kombiniert mehrere der obigen: entspricht oft -vET (alle nicht-druckbaren Zeichen, Tabs und Zeilenenden sichtbar machen).
- u      (Ignoriert in GNU cat – historisch für ungepufferte Ausgabe).
- help    Zeigt Hilfe-Informationen an.
- version    Zeigt Versionsinfo an.

## cd (Change Directory)

**Beschreibung:** Wechselt in ein anderes Verzeichnis. Mit cd kann das aktuelle Arbeitsverzeichnis geändert werden („change directory“)[ionos.at](#). Ohne Argument wechselt cd standardmäßig ins Heimatverzeichnis des Benutzers.

**Syntax:**

cd [OPTION] [VERZEICHNIS]

*Man kann relative Pfade (bezogen auf das aktuelle Verzeichnis) oder absolute Pfade (ausgehend vom Wurzelverzeichnis /) angeben. Erfolgt der Wechsel in ein Zielverzeichnis, so wird dieses das neue aktuelle Arbeitsverzeichnis; bei nicht vorhandenen Zielen oder fehlenden Rechten bleibt das Verzeichnis unverändert[ionos.at](#).*

**Optionen:**

## Option Beschreibung

- L      Folge symbolischen Links **logisch**. Dies ist die Voreinstellung. (Beim Navigieren durch einen Symlink bleibt man in der logischen Pfadstruktur)[ionos.at](#).
- P      Folge symbolischen Links **physisch**. Wechselt in das tatsächliche Ziel im Dateisystem und löst symbolische Links auf[ionos.at](#).

**Unterbefehle / spezielle Aufrufe:**

- cd ~ wechselt ins Heimatverzeichnis des aktuellen Benutzers[ionos.at](#).

- cd - wechselt in das vorherige Arbeitsverzeichnis zurück (Toggle zwischen zwei Verzeichnissen)[ionos.at](#).
- cd /Pfad/zu/Verz wechselt absolut in ein angegebenes Verzeichnis vom Wurzelpfad aus (hier z.B. nach /Pfad/zu/Verz).
- cd .. wechselt eine Ebene höher (ins Parent-Verzeichnis), cd ../../ entsprechend zwei Ebenen usw.
- cd . bleibt im aktuellen Verzeichnis (. bezeichnet das aktuelle Verzeichnis, meistens kein Effekt).

## **pwd (Print Working Directory)**

**Beschreibung:** Gibt den vollständigen Pfad des aktuellen Arbeitsverzeichnisses aus[ionos.at](#). Mit pwd kann man sich anzeigen lassen, *in welchem Verzeichnis man sich gerade befindet*.

### **Syntax:**

pwd [OPTION]...

### **Optionen:**

#### **Option Beschreibung**

**-L** Ausgabe des Pfads mit Berücksichtigung symbolischer Links (logischer Pfad, Standard). Wenn z.B. ein symbolischer Link als aktuelles Verzeichnis dient, wird der Linkpfad angezeigt[ionos.de](#).

**-P** Ausgabe des *physischen* Pfads: Symbolische Links werden aufgelöst, der tatsächliche Verzeichnispfad wird angezeigt[ionos.de](#).

**--help** Zeigt Hilfe-Informationen zu pwd an.

**--version** Zeigt Versionsinformationen an.

*Hinweis:* pwd ist häufig als Shell-Builtin implementiert. Das externe Programm /bin/pwd unterstützt ggf. andere oder weniger Optionen (bei GNU Coreutils nur --help und --version zusätzlich)[ionos.de](#). In der Regel genügt der Shell-eigene pwd Befehl.

## **whoami**

**Beschreibung:** Zeigt den Benutzernamen des aktuell angemeldeten Users anfile-4fdk7rvx39azjretinarqb. Praktisch, um schnell zu prüfen, unter welchem Account ein Prozess läuft.

### **Syntax:**

whoami

*Keine spezifischen Optionen.* Der Befehl gibt äquivalent echo \$USER aus (den Namen des aktuellen Unix-Accounts).

## touch

**Beschreibung:** Erzeugt eine neue leere Datei oder ändert Zeitstempel (Zugriffs- und Änderungszeit) bestehender Dateien. Wenn die Datei nicht existiert, wird sie (sofern nicht -c verwendet wird) als leere Datei angelegt[man7.org](#). Oft genutzt, um Dateien schnell anzulegen oder deren *timestamp* zu aktualisieren.

### Syntax:

touch [OPTION]... DATEI...

### Optionen:

| Option                      | Beschreibung                                                                                                                                                                                                                                                                                                                      |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a                          | Ändert nur die Zugriffszeit (atime) der Datei <a href="#">man7.org</a> .                                                                                                                                                                                                                                                          |
| -m                          | Ändert nur die Änderungszeit (mtime) der Datei <a href="#">man7.org</a> .                                                                                                                                                                                                                                                         |
| -c, --no-create             | Legt keine neue Datei an. Wenn die Datei nicht existiert, wird kein neues leeres File erstellt (ansonsten Standardverhalten ohne -c) <a href="#">man7.org</a> <a href="#">man7.org</a> .                                                                                                                                          |
| -d, --date=STRING           | Verwendet den angegebenen Datums-String anstelle der aktuellen Zeit. Der String wird geparsst (z.B. "2025-05-01 12:00:00" oder relative Angaben wie "next Friday" sind möglich) <a href="#">man7.org</a> <a href="#">man7.org</a> .                                                                                               |
| -t<br>[[CC]YY]MMDDhhmm[.ss] | Verwendet einen konkreten Zeitstempel im angegebenen Format (JJMMTThhmm[.ss]) anstatt der aktuellen Zeit <a href="#">man7.org</a> . Beispiel: -t 202312312359.59 setzt 31.12.2023 23:59:59.                                                                                                                                       |
| -r, --reference=DATEI       | Übernimmt die Zeitstempel (Zugriffs- und Änderungszeit) der angegebenen Referenzdatei. Die Zielfile erhält also dieselben Zeiten wie DATEI <a href="#">man7.org</a> .                                                                                                                                                             |
| -h, --no-dereference        | Ändert bei symbolischen Links den <i>Link selbst</i> (deren Metadaten), nicht die referenzierte Datei. Ohne diese Option würde touch auf den Ziel-Dateipfad wirken (nicht auf den Symlink). ( <i>Diese Option ist nur wirksam, wenn das System das Verändern von Symlink-Zeitstempeln unterstützt.</i> ) <a href="#">man7.org</a> |

| Option            | Beschreibung                                                                                                                                                                 |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| --time=WORD       | Bestimmt, welcher Zeitstempel geändert wird. Mögliche Werte: atime, access, use (entspricht -a), oder mtime, modify (entspricht -m) <a href="http://man7.org">man7.org</a> . |
| --help, --version | Anzeigen der Hilfe bzw. Versionsinformation zu touch.                                                                                                                        |

### man (Manual Page Viewer)

**Beschreibung:** Zeigt die Handbuchseite (Manual Page) für einen angegebenen Befehl oder ein Thema an. man ist das Offline-Hilfe- und Dokumentationssystem unter Unix/Linux.

#### Syntax:

man [SECTION] <Thema/Command>

*Beispiel:* man ls zeigt die Manpage zu **ls**. Manpages sind meist nach Kategorien/Abschnitten (1=Benutzerkommandos, 5=Dateiformate, 8=Administrator-Kommandos etc.) gegliedert – optional kann man mit man 5 passwd z.B. direkt die Manpage zum Dateiformat **passwd** (Abschnitt 5) öffnen.

#### Optionen:

| Option             | Beschreibung                                                                                                                                                                                     |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -k <i>Suchwort</i> | Durchsucht die Manpage-Kurzzusammenfassungen nach <i>Suchwort</i> (entspricht dem Befehl apropos). Beispiel: man -k network listet alle Manual-Topics auf, deren Beschreibung "network" enthält. |
| -f <i>Befehl</i>   | Findet den Namen und Abschnitt einer Manpage zu <i>Befehl</i> (entspricht whatis). Beispiel: man -f passwd zeigt kurz "passwd (1) - change user password".                                       |
| -a                 | Zeigt nacheinander alle Manpages an, die zum angegebenen Namen passen (wenn z.B. ein Name in mehreren Abschnitten existiert).                                                                    |
| -P <i>Pager</i>    | Legt ein anderes Pager-Programm fest (Standard ist less). Beispiel: man -P cat ls würde die ls-Manpage durch cat leiten.                                                                         |
| --help, --version  | Zeigt Hilfe bzw. Version von man.                                                                                                                                                                |

### whatis

**Beschreibung:** Zeigt eine kurze Ein-Satz-Beschreibung zu einem Befehl an (frägt die Datenbank der whatis-Einträge ab). Es ist eine Kurzform von man -f und liefert die

Synopsis aus der Manpage.

### Syntax:

whatis <Befehl/Topic>

### Optionen:

| Option                    | Beschreibung                                                                                                                                                                          |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (keine üblichen Optionen) | whatis wird meist ohne Optionen verwendet. Es greift auf die gleichen Datenbanken wie apropos/man -k zu. Sollte die Datenbank nicht aktuell sein, kann sie mit mandb erneuert werden. |

## Benutzer- und Gruppenverwaltung

### passwd (Passwort ändern/setzen)

**Beschreibung:** Ändert das Passwort eines Benutzerkontos oder verwaltet Konto-Sperren. Als normaler Benutzer kann man mit passwd sein eigenes Passwort ändern; als root kann man mit passwd <user> das Passwort eines anderen Accounts setzen oder bestimmte Eigenschaften (Sperren, Ablaufdatum) ändern. file-4fdk7rvx39azjretinarqbfile-4fdk7rvx39azjretinarqb.

### Syntax:

passwd [OPTION] [BENUTZER]

*Standardaufruf ohne Benutzer wirkt auf den aktuellen Benutzer. Bei Angabe eines Benutzernamens werden dessen Passwortinformationen geändert (Root-Rechte erforderlich).*

### Optionen:

#### Option Beschreibung

|              |                                                                                                                                                                                                                                                      |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -l, --lock   | Sperrt das Benutzerkonto, indem das Passwort in /etc/shadow mit einem Präfix (meist ! oder *) unbrauchbar gemacht wird <a href="#">wiki.ubuntuusers.de</a> . Der Benutzer kann sich danach nicht mehr anmelden, bis das Konto wieder entsperrt wird. |
| -u, --unlock | Entsperrt ein zuvor gesperrtes Konto (entfernt das Sperr-Präfix im verschlüsselten Passwort) <a href="#">wiki.ubuntuusers.de</a> .                                                                                                                   |
| -d, --delete | Löscht das Passwort des Benutzers (leer setzen). Achtung: Damit wird eine passwortlose Anmeldung möglich, sofern nicht andersweitig gesichert <a href="#">wiki.ubuntuusers.de</a> .                                                                  |
| -e, --expire | Markiert das Passwort als abgelaufen. Beim nächsten Login muss der Benutzer ein neues Passwort setzen.                                                                                                                                               |

## Option Beschreibung

**-x TAGE** Setzt die maximale Gültigkeitsdauer eines Passworts auf *TAGE*. Nach Ablauf muss es geändert werden. (Änderung des PASS\_MAX\_DAYS in /etc/shadow.)

**-n TAGE** Setzt die minimale Anzahl an Tagen zwischen zwei Passwortänderungen (PASS\_MIN\_DAYS). Der Benutzer kann sein Passwort erst nach *TAGE* Tagen erneut ändern.

**-w TAGE** Warnzeit in Tagen vor Passwortablauf (PASS\_WARN\_AGE). Der Benutzer erhält *TAGE* Tage vor Ablauf eine Warnung beim Login.

**-S** Zeigt den Passwortstatus an (Gesperrt? nicht gesetzt? Datum der letzten Änderung usw.). passwd -S <user> gibt eine Statuszeile aus.

**Interaktives Verhalten:** Wird passwd ohne Optionen aufgerufen, fordert es zur Eingabe des aktuellen Passworts (zur Authentifizierung, falls nicht root) und zweimal zur Eingabe des neuen Passworts auf (zwecks Bestätigung). Es verweigert einfache Passwörter gemäß der System-Policy (Pluggable Authentication Modules/pam Richtlinien). Root-Benutzer muss das alte Passwort nicht kennen.

## chage (Change Age – Passwortalter und Gültigkeit)

**Beschreibung:** Verändert Einstellungen zum Passwort-Aging – also Ablaufdatum, Mindestalter, Warnzeit etc. Mit chage kann ein Administrator festlegen, wann ein Benutzer sein Passwort ändern muss bzw. wann der Account deaktiviert wirdfile-4fdk7rvx39azjretinarqb.

### Syntax:

chage [OPTION]... BENUTZER

### Optionen:

## Option Beschreibung

**-l** Listet die aktuellen Passwort- und Alterseinstellungen für den Benutzer in einem lesbaren Format auf (Letzte Änderung, Min/Max Alter, Warnzeit, Inaktivzeit, Ablaufdatum) – Übersicht.

**-m TAGE** Setzt die minimale Anzahl von Tagen *TAGE*, die ein Passwort gültig sein muss, bevor es geändert werden darf (PASS\_MIN\_DAYS).

**-M TAGE** Setzt die maximale Gültigkeitsdauer in Tagen (PASS\_MAX\_DAYS). Nach *TAGE* Tagen läuft das Passwort ab und muss geändert werden.

**-w TAGE** Setzt die Anzahl der Tage *TAGE* Vorwarnzeit vor Passwortablauf (PASS\_WARN\_AGE).

## Option Beschreibung

|                      |                                                                                                                                                                                                                                                                                                |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                      | Legt fest, wie viele Tage nach Passwortablauf der Account <b>inaktiv</b> wird (PASS_INACTIVE_DAYS). Ein Wert von 0 deaktiviert den Account sofort nach Ablauf des Passworts, -1 schaltet diese Funktion aus <a href="http://wiki.ubuntuusers.de/wiki.ubuntuusers.de">wiki.ubuntuusers.de</a> . |
| <b>-I TAGE</b>       |                                                                                                                                                                                                                                                                                                |
| <b>-E DATUM</b>      | Setzt ein Ablaufdatum für das Benutzerkonto (im Format YYYY-MM-DD). Nach diesem Datum wird der Account deaktiviert <a href="http://wiki.ubuntuusers.de/wiki.ubuntuusers.de">wiki.ubuntuusers.de</a> . Das Datum -1 (oder "" leer) entfernt ein gesetztes Ablaufdatum (kein Ablauf).            |
| <b>-d YYYY-MM-DD</b> | Setzt das Datum der letzten Passwortänderung auf das angegebene Datum (wirkt wie Backdating oder um zu triggern, dass ein Passwort abläuft). Kann auf 0 gesetzt werden, um das Passwort als sofort ablaufend zu markieren.                                                                     |
| <b>-h, --help</b>    | Zeigt Hilfe an.                                                                                                                                                                                                                                                                                |

*Hinweis:* Viele dieser Einstellungen lassen sich auch mit passwd (Optionen -x, -n, -w, -i, -e) vornehmen. chage bietet eine dedizierte Schnittstelle dafür. Ohne Optionen (als root) startet chage <user> interaktiv einen Dialog, um alle Werte einzustellen.

## w

**Beschreibung:** Zeigt an, welche Benutzer gerade am System angemeldet sind und was sie gerade tun. w fasst Informationen über aktuell eingeloggte User und deren laufende Prozesse zusammen (inkl. Login-TTY, Leerlaufzeit, genutzte CPU-Zeit, aktueller Befehl)`file-4fdk7rvx39azjretinarqb`.

### Syntax:

`w [OPTION]`

### Optionen:

## Option Beschreibung

|           |                                                                                                                                                                                                                                                   |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-h</b> | Blendet die Kopfzeile in der Ausgabe aus (unterdrückt die Überschriftenzeile).                                                                                                                                                                    |
| <b>-s</b> | Kurze Ausgabe. Unterdrückt die Spalten für Login-Zeit, Leerlauf etc. (nicht auf allen Systemen vorhanden).                                                                                                                                        |
| <b>-f</b> | Zeigt Informationen zu von utmp als <i>geflagt</i> markierten Einträgen. Auf einigen Systemen steuert dies die Anzeige von Remote-Host oder Exit-Informationen. (Bei neueren procps-Versionen hat -f keine Funktion, da Host immer gezeigt wird.) |

## Option Beschreibung

**-V, --version** Zeigt Versionsinfo von w.

Die Ausgabe von w enthält in der ersten Zeile System-Uptime und Load Average. Danach folgen pro eingeloggtem Benutzer: Benutzername, Terminal, Remote-Host, Login-Zeit, Leerlaufdauer, CPU-Verbrauch und der aktuell laufende Prozess (TTY-Prozess).

## who

**Beschreibung:** Zeigt Informationen über die aktuell angemeldeten Benutzer. Im Gegensatz zu w liefert who primär Login-Zeitpunkte und Terminalsessions, jedoch keine Prozess- oder Idle-Infos.

### Syntax:

who [OPTION]... [Datei]

Standard liest who die System-Login-Datenbank /var/run/utmp und gibt für jeden Eintrag Benutzer, Terminal, Datum/Zeit und ggf. Host aus.

### Optionen:

## Option Beschreibung

**-H** Zeigt eine Kopfzeile (Überschriften) über den Spalten an.

**-q** "Quick who": Zeigt nur Benutzernamen und die Anzahl der angemeldeten Nutzer. (Gleichbedeutend mit users-Befehl.)

**-b** Zeigt die Zeit des letzten Systemstarts (Boot time) an.

**-r** Zeigt den aktuellen Runlevel des Systems.

**-a** Zeigt alle Einträge aus utmp, inklusive Einträgen für laufende Prozesse am System (wie getty oder systemd-Spawns). Kombination von mehrere Flags: entspricht oft -b -d --login -p -r -t -T -u je nach Implementierung.

**-m** Zeigt nur die Einträge für das aktuelle Terminal an (beschränkt auf den Benutzer, der den Befehl aufruft).

**-u** Zeigt zusätzlich die Leerlaufzeit (Idle) und den PID des Login-Prozesses für jeden Benutzer.

**-T** Zeigt den Terminal-Status (± Flag) an: + bedeutet terminal ist bei mesg schreibbar, - bedeutet nicht schreibbar, ? unbekannt. (In vielen who-Implementierungen bereits Standardverhalten ohne Option.)

| Option | Beschreibung |
|--------|--------------|
|--------|--------------|

|                   |                                       |
|-------------------|---------------------------------------|
| --help, --version | Anzeigen von Hilfe bzw. Versionsinfo. |
|-------------------|---------------------------------------|

### useradd (Benutzer anlegen)

**Beschreibung:** Legt ein neues Benutzerkonto an. useradd ist ein Low-Level-Befehl, der die Systemkontodateien (wie /etc/passwd, /etc/shadow, /etc/group) direkt verändert, um einen neuen User einzurichten [pq.hosting](#). In Debian/Ubuntu-Umgebungen wird oft das Wrapper-Skript adduser verwendet, das auf useradd aufbaut und interaktiv ein paar Dinge abfragt; useradd selbst ist nicht interaktiv und benötigt alle Optionen zum Festlegen der Eigenschaften des neuen Accounts.

**Syntax:**

useradd [OPTION]... BENUTZERNAME

**Optionen:** (häufigste Optionen beim Anlegen eines Benutzers)

| Option | Beschreibung |
|--------|--------------|
|--------|--------------|

|    |                                                                                                                                                                  |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -c | Kommentarfeld (GECOS) für den Benutzer, z.B. voller Name oder Hinweis.<br><i>Kommentar</i> Wird in /etc/passwd eingetragen <a href="#">wiki.ubuntuusers.de</a> . |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|         |                                                                                                                                                                                                                                   |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -d HOME | Pfad des Home-Verzeichnisses. Standard: /home/<Benutzername>. Kann geändert werden; wenn -m verwendet wird, wird dieses Verzeichnis erstellt und mit Standard-Dateien aus /etc/skel befüllt <a href="#">wiki.ubuntuusers.de</a> . |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|    |                                                                                                                                                                                                                         |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -m | Legt das Home-Verzeichnis an und kopiert Standard-Dateien hinein. Sollte gesetzt sein, außer man möchte <i>kein</i> Home-Verz. (Ohne -m wird das Home nicht automatisch erstellt) <a href="#">wiki.ubuntuusers.de</a> . |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|          |                                                                                                                                                                                                                                  |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -s SHELL | Login-Shell des Benutzers. Standard ist normalerweise /bin/bash (oder was in /etc/default/useradd als SHELL definiert ist). Kann z.B. auf /sbin/nologin gesetzt werden für System-Accounts <a href="#">wiki.ubuntuusers.de</a> . |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|        |                                                                                                                                                                                       |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -u UID | Gewünschte User-ID (numerisch). Darf nicht bereits vergeben sein (außer mit -o). Wenn nicht angegeben, wird die nächste freie UID > 999 gewählt <a href="#">wiki.ubuntuusers.de</a> . |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|    |                                                                                                                                                                                                                |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -o | Erlaubt die Verwendung einer bereits vergebenen UID (UID-Doppelungen). Zusammen mit -u benutzt, um zwei Benutzer mit identischer UID anzulegen. (Normalerweise nicht empfohlen, da UID eindeutig sein sollen.) |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|        |                                                                                                                                         |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------|
| -g GRP | Primäre Gruppe (Name oder GID), der der Benutzer angehört <a href="#">wiki.ubuntuusers.de</a> . Standard: eine neue Gruppe mit gleichem |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------|

| <b>Option</b>      | <b>Beschreibung</b>                                                                                                                                                                                                                                                                                                                                                      |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    | Namen wie der Benutzer (bei entsprechender Systemkonfiguration) oder eine in /etc/default/useradd definierte GROUP. Die angegebene Gruppe muss existieren.                                                                                                                                                                                                               |
| <b>-G Liste</b>    | Liste zusätzlicher Gruppen (sekundäre Gruppen), denen der Benutzer angehören soll <a href="#">wiki.ubuntuusers.de</a> . Gruppennamen durch Komma getrennt angeben (ohne Leerzeichen).                                                                                                                                                                                    |
| <b>-p Passwort</b> | Das verschlüsselte Passwort für den Account (wie es z.B. von openssl passwd oder crypt(3) erzeugt würde) <a href="#">wiki.ubuntuusers.de</a> . <b>Wichtig:</b> Aus Sicherheitsgründen sollte man hier <i>nicht</i> ein Klartextpasswort übergeben. Bleibt -p leer, wird der Account ohne gesetztes Passwort angelegt (Anmeldung evtl. gesperrt bis passwd gesetzt wird). |
| <b>-e</b>          | (bei useradd nicht üblich – wird bei usermod benutzt für Ablauf setzen. Bei einigen Systemen könnte useradd -e <Datum> ein Ablaufdatum setzen.)                                                                                                                                                                                                                          |
| <b>-f TAGE</b>     | Anzahl an Tagen nach Passwortablauf, bis der Account deaktiviert wird (Analog zu chage -l). -f 0 würde den Account sofort mit Passwortablauf sperren. Standard -1 (nie sperren).                                                                                                                                                                                         |
| <b>-N</b>          | (Bei einigen Systemen: <i>No user group</i> – verhindert das automatische Anlegen einer gleichnamigen Gruppe.)                                                                                                                                                                                                                                                           |
| <b>-r</b>          | Erstellt einen System-Account (UID unter einer gewissen Schwelle, z.B. <1000, kein Expire, kein Mail-Spool). Oft genutzt für Systembenutzer ohne Login.                                                                                                                                                                                                                  |
| <b>-M</b>          | Kein Home-Verzeichnis anlegen (selbst wenn Default CREATE_HOME aktiviert ist).                                                                                                                                                                                                                                                                                           |
| <b>-k SKELDIR</b>  | Gibt ein alternatives Skeleton-Verzeichnis an (anstatt /etc/skel), von wo aus bei -m die initialen Dateien ins Home kopiert werden.                                                                                                                                                                                                                                      |
| <b>-D</b>          | Zeigt die aktuellen Standardwerte für useradd an (oder mit anderen -D- Optionen können diese verändert werden).                                                                                                                                                                                                                                                          |

*Nach dem Anlegen mit useradd sollte meist ein Passwort gesetzt werden (passwd <user>), außer es handelt sich um einen Account ohne Login.*

### **usermod (Benutzerkonto ändern)**

**Beschreibung:** Modifiziert ein bestehendes Benutzerkonto – z.B. Name, UID, Primärgruppe, zusätzliche Gruppen, Login-Shell, Expire-Datum usw.[wiki.ubuntuusers.dewiki.ubuntuusers.de](#). Im Grunde die gleichen Optionen wie

useradd, aber angewendet auf bestehende Benutzer. Häufig genutzt, um Benutzer umzubenennen, in andere Gruppen aufzunehmen oder Accounts zu sperren.

**Syntax:**

usermod [OPTION]... LOGIN

**Optionen:**

| Option             | Beschreibung                                                                                                                                                                                                                                                                                                                                 |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -c Kommentar       | Neuen Kommentar (GECOS-Feld) setzen <a href="#">wiki.ubuntuusers.de</a> .                                                                                                                                                                                                                                                                    |
|                    | Ändert das Home-Verzeichnis des Benutzers auf <i>Neues_Home</i> .<br><b>Hinweis:</b> Ohne weiteres passiert das alte Home nicht automatisch mit. Mit zusätzlicher Option -m werden die Inhalte des alten Home-Verz. ins neue Verzeichnis verschoben und das neue Verzeichnis ggf. erstellt <a href="#">wiki.ubuntuusers.de</a> .             |
| -d Neues_Home      | (Mit -d verwenden) Verschiebt beim Ändern des Homepfads (-d) die Dateien vom alten ins neue Home und erstellt es falls nötig <a href="#">wiki.ubuntuusers.de</a> .                                                                                                                                                                           |
| -m                 | Ändert den Login-Namen des Benutzers (Umbenennung des Accounts) <a href="#">wiki.ubuntuusers.de</a> . Der Benutzername in allen Systemkontodateien wird angepasst, <i>nicht</i> jedoch automatisch der Name des Home-Verzeichnisses oder Mail-Spool (diese müssten manuell umbenannt werden, falls gewünscht).                               |
| -l Neuer_Login     | Ändert die numerische Benutzer-ID auf <i>Neue_UID</i> . Falls die UID bereits vergeben ist, muss -o benutzt werden. Dateien im Home-Verzeichnis mit alter UID werden auf neue UID umgestellt; Dateien außerhalb des Homes muss der Admin selbst anpassen <a href="#">wiki.ubuntuusers.de</a>                                                 |
| -u Neue_UID        | Erlaubt die Verwendung einer bereits vergebenen UID (nur zusammen mit -u sinnvoll).                                                                                                                                                                                                                                                          |
| -o                 | Ändert die primäre Gruppenzugehörigkeit (Name oder GID) <a href="#">manpages.ubuntu.com</a> . Die neue Gruppe muss existieren. Alle Dateien im Home, die der alten primären Gruppe gehörten, werden auf die neue Gruppe übertragen <a href="#">manpages.ubuntu.com</a> . (Dateien außerhalb des Homes müssen ggf. manuell angepasst werden.) |
| -g Neue_PrimGruppe | Setzt die Liste der sekundären Gruppen neu für den Benutzer <a href="#">manpages.ubuntu.com</a> . <b>Wichtig:</b> Ohne -a entfernt diese Option den Benutzer aus allen Gruppen, die nicht in der Liste                                                                                                                                       |
| -G Gruppe(n)       |                                                                                                                                                                                                                                                                                                                                              |

| Option               | Beschreibung                                                                                                                                                                                                                                                                                                                              |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                      | stehen <a href="#">wiki.ubuntuusers.demanpages.ubuntu.com</a> . Um einen Benutzer zusätzlichen Gruppen hinzuzufügen, sollte man <b>-a -G</b> gemeinsam verwenden (siehe -a).                                                                                                                                                              |
| <b>-a</b>            | <i>Append</i> – fügt den Benutzer zusätzlichen Gruppen hinzu, anstatt die Gruppenzugehörigkeit zu überschreiben <a href="#">wiki.ubuntuusers.dewiki.ubuntuusers.de</a> . <b>Nur in Kombination mit -G verwenden</b> , um den Benutzer der/die angegebenen Gruppen hinzuzufügen, ohne ihn aus anderen zu entfernen. (Siehe Warnung unten.) |
| <b>-s Shell</b>      | Ändert die Login-Shell des Benutzers <a href="#">wiki.ubuntuusers.de</a> . Z.B. auf /sbin/nologin setzen, um interaktive Logins zu unterbinden.                                                                                                                                                                                           |
| <b>-e YYYY-MM-DD</b> | Setzt das Ablaufdatum des Accounts (das Datum, an dem das Konto deaktiviert wird) <a href="#">wiki.ubuntuusers.de</a> . Nach diesem Datum kann sich der User nicht mehr anmelden. usermod -e "" user entfernt ein Ablaufdatum (setzt es auf <i>nie</i> ).                                                                                 |
| <b>-f Tage</b>       | Setzt die Anzahl an Tagen nach Passwortablauf, bis der Account deaktiviert wird <a href="#">wiki.ubuntuusers.de</a> . -f 0 deaktiviert sofort nach Ablauf; -f -1 entfernt eine evtl. gesetzte Deaktivierung (nie sperren nach Ablauf).                                                                                                    |
| <b>-L</b>            | Sperrt das Benutzerkonto (lock). Fügt ein ! dem Passwort-Hash voran (im Shadow-Eintrag) <a href="#">wiki.ubuntuusers.de</a> , was die Anmeldung verhindert. Kann nicht mit -p oder -U kombiniert werden.                                                                                                                                  |
| <b>-U</b>            | Entsperrt ein zuvor gesperrtes Passwort (passwd -l). Entfernt das ! im Shadow-Passwortfeld, sodass der ursprüngliche Hash wieder wirksam wird <a href="#">wiki.ubuntuusers.de</a> . Nicht zusammen mit -L oder -p verwenden.                                                                                                              |
| <b>-p Hash</b>       | Setzt das <i>verschlüsselte</i> Passwort (Hash) des Users auf den angegebenen Wert <a href="#">wiki.ubuntuusers.de</a> . Sollte mit Vorsicht genutzt werden – für Passwortänderung ist passwd meist besser geeignet.                                                                                                                      |
| <b>--help</b>        | Hilfe anzeigen.                                                                                                                                                                                                                                                                                                                           |

### Wichtige Hinweise:

- *Warnung:* Die Kombination von -G ohne -a überschreibt die Gruppenliste vollständig. Das heißt, der Benutzer wird aus allen Gruppen entfernt, die nicht

explizit bei -G genannt sind [wiki.ubuntuusers.de](http://wiki.ubuntuusers.de). Um einen User nur hinzuzufügen, niemals ohne -a verwenden!

- Man kann keinen angemeldeten Benutzer umbenennen oder dessen UID ändern, solange er eingeloggt ist oder laufende Prozesse hat. Zudem keine Änderungen an dem eigenen Account durchführen, während man noch mit diesem angemeldet ist (kann zu Inkonsistenzen führen).
- Beim Ändern des Benutzernamens oder UIDs sollten evtl. Crontab-Dateien und geplante at-Jobs manuell umbenannt/angepasst werden, und in NIS/LDAP-Umgebungen müssen Änderungen dort vorgenommen werden [wiki.ubuntuusers.de](http://wiki.ubuntuusers.de).

## **userdel (Benutzer löschen)**

**Beschreibung:** Löscht ein Benutzerkonto aus dem System (entfernt die Einträge in /etc/passwd, /etc/shadow und ggf. /etc/group falls der Benutzer eine eigene Gruppe hatte). Optional können auch das Home-Verzeichnis und die Mail-Spool des Benutzers gelöscht werden.

**Syntax:**

userdel [OPTION] BENUTZER

**Optionen:**

### **Option Beschreibung**

**-r** Entfernt *rekursiv* das Heimatverzeichnis des Benutzers **und** seine Mail-Spool (üblicherweise /var/spool/mail/<user>), sowie alle Dateien darin, beim Löschen des Accounts. Ohne -r bleiben Home und Mails unangetastetfile-4fdk7rvx39azjretinarqb.

**-f** Force: Erzwingt das Löschen des Accounts, selbst wenn der Benutzer noch angemeldet ist oder laufende Prozesse hat. (Warnung: kann zu inkonsistenten Systemzuständen führen, Prozess bleibt ohne Benutzer etc. Diese Option sollte vermieden werden.)

Ohne Option wird der Account aus den Systemdateien gelöscht, aber das (*keine*) Home-Verzeichnis, die Dateien und Mails bleiben bestehen. Admin muss sie ggf. manuell entfernen oder anderweitig zuordnen.

*Hinweis:* userdel bricht ab, wenn der Benutzer gerade angemeldet ist (außer mit -f). Es ist ratsam, vor dem Löschen sicherzustellen, dass keine Prozesse mehr unter der betreffenden UID laufen (z.B. mit pkill -u username beenden). Dateien außerhalb des Home, die dem Benutzer gehören, verbleiben auf dem Dateisystem mit der nun unbenutzten UID – diese müssten manuell gesucht und entfernt oder neu zugeordnet werden (z.B. mit find / -uid <oldUID>).

## **id**

**Beschreibung:** Zeigt die Benutzer- und Gruppen-Identitäten eines Benutzerkontos an. Standardmäßig gibt id die UID, primäre GID und zugehörigen Gruppennamen des aktuellen Benutzers zurückfile-4fdk7rvx39azjretinarqb. Mit Angabe eines Benutzernamens kann man die IDs für einen anderen Account abrufen.

**Syntax:**

```
id [OPTION]... [BENUTZER]
```

**Optionen:**

| <b>Option</b>            | <b>Beschreibung</b>                                                                                                                                                                |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (ohne)                   | Ohne Options werden UID, Benutzername, GID, Gruppenname und alle zusätzlichen Gruppen des Benutzers aufgelistet (z.B. uid=1000(max) gid=1000(max) Gruppen=1000(max),27(sudo),...). |
| <b>-u</b>                | Gibt nur die effektive User-ID (UID, numerisch) aus. Beispiel: id -u alice könnte 1001 ausgeben.                                                                                   |
| <b>-g</b>                | Gibt nur die effektive Gruppen-ID (GID, numerisch) aus.                                                                                                                            |
| <b>-G</b>                | Gibt alle Gruppen-IDs (numerisch) aus, in denen der Benutzer Mitglied ist (Primär- und Sekundärgruppen).                                                                           |
| <b>-n</b>                | Gibt Namen anstelle von numerischen IDs. Kann in Kombination mit -u, -g, -G verwendet werden. Z.B. id -Gn alice listet alle Gruppennamen von alice.                                |
| <b>-r</b>                | Zeigt die reale UID/GID an (falls z.B. mit sudo aufgerufen, reale vs. effektive ID unterscheiden).                                                                                 |
| <b>-Z</b>                | Zeigt SELinux-Sicherheitskontext, falls SELinux aktiv ist (z.B. unconfined_u:unconfined_r:unconfined_t:s0 für einen unconfined Benutzer).                                          |
| <b>--help, --version</b> | Hilfe bzw. Version anzeigen.                                                                                                                                                       |

## **groupadd (Gruppe anlegen)**

**Beschreibung:** Legt eine neue Gruppe im System an. Fügt einen Eintrag in /etc/group (und ggf. /etc/gshadow) hinzu.

**Syntax:**

```
groupadd [Optionen] GRUPPENNAME
```

**Optionen:**

| <b>Option</b>             | <b>Beschreibung</b>                                                                                                                                       |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-g GID</b>             | Verwendet die angegebene numerische GID für die neue Gruppe. Sollte noch nicht vergeben sein (oder mit -o überschrieben werden).                          |
| <b>-o</b>                 | Erlaubt die Verwendung einer bereits existierenden GID (zusammen mit -g zu nutzen, um GID-Doppel zu erzwingen).                                           |
| <b>-f</b>                 | "Force": falls die Gruppe schon existiert, keinen Fehler melden; falls die angegebene GID bereits vergeben ist, wählt eine andere (ignoriert -o).         |
| <b>-r</b>                 | Erstellt eine Systemgruppe (GID wird aus dem System-Bereich gewählt, z.B. <1000).                                                                         |
| <b>-K Schlüssel= Wert</b> | Überschreibt einen Eintrag aus /etc/login.defs temporär (z.B. -K GID_MIN=500 um untere GID-Bereiche anzupassen).                                          |
| <b>-p Passwort</b>        | Setzt das (verschlüsselte) Gruppenpasswort direkt. (Gruppenpasswörter werden selten genutzt – ermöglicht mit newgrp temporär in eine Gruppe zu wechseln.) |

### **groupmod (Gruppe ändern)**

**Beschreibung:** Ändert Eigenschaften einer bestehenden Gruppe – hauptsächlich Name oder GID.

#### **Syntax:**

groupmod [Optionen] GRUPPE

#### **Optionen:**

| <b>Option</b>       | <b>Beschreibung</b>                                                                                                                                                                             |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-n NeuerName</b> | Ändert den Gruppennamen in <i>NeuerName</i> . Der alte Name in allen relevanten Systemdateien wird ersetzt.                                                                                     |
| <b>-g NeueGID</b>   | Ändert die GID der Gruppe. Muss eindeutig sein (oder mit -o doppelt erlaubt). Dateien mit alter GID außerhalb, die nicht automatisch erfasst werden, müssen eventuell manuell angepasst werden. |
| <b>-o</b>           | Erlaubt das Setzen einer bereits vergebenen GID (nur mit -g relevant).                                                                                                                          |
| <b>-p Passwort</b>  | Ändert das (verschlüsselte) Gruppenpasswort. (Entspricht Editieren von /etc/gshadow).                                                                                                           |

### **groupdel (Gruppe löschen)**

**Beschreibung:** Entfernt eine Gruppe aus der Systemgruppe-Datei. Die Gruppe wird aus /etc/group (und /etc/gshadow) getilgt.

**Syntax:**

groupdel GRUPPENNAME

**Hinweis:** Dieser Befehl hat keine besonderen Optionen – er löscht die Gruppe, sofern sie nicht als Primärgruppe eines Benutzers in /etc/passwd noch verwendet wird. Man kann keine Gruppe löschen, die noch als Hauptgruppe eines existierenden Benutzerkontos eingetragen ist; in dem Fall muss man ggf. erst den Benutzer anpassen oder löschen. Dateien auf dem System, die dieser Gruppe gehören, behalten die nun ungültige GID bei, bis man sie manuell ändert.

**su (Switch User)**

**Beschreibung:** Wechselt den Benutzerkontext im laufenden Terminal. Mit su („substitute user“) kann man einen anderen Benutzeraccount annehmen – standardmäßig root, wenn kein Benutzer angegeben wird. Dies erfordert die Eingabe des Zielbenutzer-Passworts (außer beim Wechsel zu root, dort abhängig von den Systemeinstellungen – oft ist root-Login per Passwort deaktiviert und man nutzt sudo).

**Syntax:**

su [OPTION]... [BENUTZER [-]]

Wird BENUTZER weggelassen, versucht su auf den Superuser (root) zu wechseln.

**Optionen:**

**Option      Beschreibung**

- Startet eine *Login-Shell* der Zielusers, d.h. die Umgebung wird vollständig umgeschaltet als würde sich der Benutzer frisch anmelden. Das Home-  
- oder **-l** Verzeichnis, Pfad (\$PATH) etc. werden übernommen. Beispiel: su - alice wechselt zu Benutzer *alice* mit deren Umgebung (äquivalent zu vollem Logout/Login als *alice*).

Führt einen angegebenen *Befehl* als Zielbenutzer aus und kehrt dann zurück.

- c Befehl** Beispielsweise su -c "systemctl restart apache2" - root führt als root einen Neustart des Apache durch und beendet sich sofort wieder.

- m, -p** Behalten die aktuelle Umgebung (*nicht* auf Ziel-User umschalten). Der aktuelle \$HOME, \$PATH, usw. bleiben erhalten, nur die Rechte wechseln. Standardverhalten ohne -l. (In vielen su Implementierungen ist -m identisch mit -p: *preserve environment*).

| <b>Option</b>            | <b>Beschreibung</b>                                                                                                                                     |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-s Shell</b>          | Startet die angegebene <i>Shell</i> statt der Standard-Login-Shell des Zielbenutzers. Der Pfad zur Shell muss im /etc/shells stehen, sofern nicht root. |
| <b>-g Gruppe</b>         | Wechselt in Kombination mit Benutzer auch die primäre Gruppe auf <i>Gruppe</i> . (Nur verfügbar in neueren util-linux su).                              |
| <b>-&lt;Ziffer&gt;</b>   | (BSD su) Angabe einer Session-Klasse – unter Linux nicht genutzt.                                                                                       |
| <b>--help, --version</b> | Anzeigen von Hilfe bzw. Version.                                                                                                                        |

*Hinweise:*

- su ohne - (also z.B. su root) wechselt zwar die Benutzer-ID, aber behält das aktuelle Verzeichnis und weite Teile der Environment von Ursprungskonto bei (\$LOGNAME, \$HOME bleiben z.B. vom ursprünglichen Nutzer). Das kann zu irritierenden Effekten führen (z.B. Schreibrechte im falschen Home). Daher nutzt man meist su - für einen *sauberen* Wechsel.
- Viele Distributionen konfigurieren aus Sicherheitsgründen su so, dass ein direkter Wechsel zu root per Passwort entweder deaktiviert ist oder kein root-Passwort gesetzt ist. Stattdessen wird sudo bevorzugt, um Root-Rechte zu erlangen.

### **sudo (Superuser Do)**

**Beschreibung:** Führt ein Kommando mit *erhöhten Rechten* aus, typischerweise als *root*, ohne die Benutzeridentität dauerhaft zu wechseln. Benutzer können mit sudo Befehle mit Administratorrechten ausführen, sofern sie in /etc/sudoers entsprechend autorisiert sind. Im Gegensatz zu su wird hier das eigene Passwort abgefragt (nicht das des Zielusers), und nur spezifische Befehle/Optionen sind erlaubt gemäß Konfiguration.

#### **Syntax:**

sudo [OPTION]... [COMMAND]

*Wird ein Befehl angegeben, führt sudo diesen als root (Standard) aus. Ohne Befehl zeigt sudo ggf. die erlaubten Befehle an oder wechselt in eine sudo-Shell (bei -s).*

#### **Optionen (Auswahl):**

| <b>Option</b> | <b>Beschreibung</b>                                                                                                                                            |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-l</b>     | Listet die erlaubten (und verbotenen) Befehle für den aufrufenden Benutzer auf (nach sudoers-Regeln). Hilfreich, um zu sehen, was man mit sudo ausführen darf. |

| Option                     | Beschreibung                                                                                                                                                                                                                                            |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-u</b>                  | Führt das angegebene Kommando als <i>Benutzer</i> aus (anstatt als root).                                                                                                                                                                               |
| <i>Benutzer</i>            | Beispiel: sudo -u alice whoami zeigt "alice", wenn man als anderer User ausführt. Standard ist root.                                                                                                                                                    |
| <b>-s</b>                  | Startet eine Shell als Zielbenutzer (Standard root). Ähnlich wie su, aber mittels sudo-Rechte. Es wird die im \$SHELL oder in passwd eingetragene Shell geöffnet.                                                                                       |
| <b>-i</b>                  | Simuliert eine vollständige Login-Shell wie su - (initialisiert Umgebung des Zielusers). Entspricht sudo -s plus Login-Umgebung.                                                                                                                        |
| <b>-b</b>                  | Führt den Befehl im Hintergrund aus ( <i>background</i> ). Sudo verlangt dennoch ggf. ein Passwort, aber detach danach.                                                                                                                                 |
| <b>-E</b>                  | Beibehaltung der aktuellen Environment-Variablen (normalerweise bereinigt sudo einige Variablen aus Sicherheitsgründen). Erlaubt das Mitnehmen der eigenen Umgebung. (Ob das erlaubt ist, kann in sudoers eingeschränkt sein.)                          |
| <b>-p</b><br><i>Prompt</i> | Definiert einen eigenen Passwort-Prompt. z.B. sudo -p "Passwort für %u:" -v (Platzhalter %u = Username).                                                                                                                                                |
| <b>-v</b>                  | "Validate": Aktualisiert/überprüft das eigene sudo-Authentifizierungsticket. D.h. fordert ggf. Passwort an, speichert das sudo Token (5min Standard) aber führt sonst nichts aus. (Verlängert die Zeit, in der man kein Passwort erneut eingeben muss.) |
| <b>-k</b>                  | Invalidiert das eigene sudo-Ticket sofort (nachfolgende sudo-Aufrufe verlangen wieder Passwort) – ( <i>kill credential</i> ).                                                                                                                           |
| <b>-H</b>                  | Setzt \$HOME in der Environment auf das Home des Zielusers (root). Normalerweise belässt sudo \$HOME vom Aufrufer; mit -H verhält es sich wie su -.                                                                                                     |
| <b>--help, -h</b>          | Hilfe anzeigen (kurze Übersicht der Optionen).                                                                                                                                                                                                          |
| <b>-K</b>                  | Löscht das Zeitstempel-Ticket komplett (ähnlich -k aber gründlicher, forciert Passwort prompt beim nächsten Mal).                                                                                                                                       |
| <b>--version</b>           | Versionsinfo ausgeben (auch Liste der unterstützten Sicherheitsmechanismen etc.).                                                                                                                                                                       |

*Hinweise:* Bei erfolgreicher Authentifizierung legt sudo einen Zeitstempel (Ticket) an, sodass für nachfolgende Aufrufe (Standard ~5 Minuten) kein erneutes Passwort nötig ist.

Dies kann per sudoers Einstellung (timestamp\_timeout) angepasst oder mit sudo -k/-K beeinflusst werden.

### **pkexec (PolicyKit Execute)**

**Beschreibung:** Führt ein Programm als ein anderer Benutzer aus (ähnlich wie sudo, aber Teil von PolKit/PolicyKit). Standardmäßig wird, sofern nicht anders angegeben, das Programm mit root-Rechten gestartet[freedesktop.org](http://freedesktop.org). pkexec erfordert, dass der aufrufende Benutzer berechtigt ist, die Aktion auszuführen (PolKit Policies); falls ja, wird eine Authentifizierungsdialog (grafisch oder textbasiert) angezeigt, um ggf. das Passwort einzugeben. Nach erfolgreicher Auth kann das gewünschte Kommando mit erhöhten Rechten ausgeführt werden.

#### **Syntax:**

```
pkexec [--user <BENUTZER>] <KOMMANDO> [ARG...]
```

#### **Optionen:**

| <b>Option</b>                    | <b>Beschreibung</b>                                                                                                                                                                                                                                                                                                                                                                   |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| --user<br><Benutzer>             | Führt das angegebene Programm als <i>Benutzer</i> aus. Standard ist root (Administratorkonto) <a href="http://freedesktop.org">freedesktop.org</a> .                                                                                                                                                                                                                                  |
| --disable-<br>internal-<br>agent | Unterdrückt den internen Authentifizierungs-Agent von pkexec. Normalerweise, falls kein grafischer PolKit-Auth-Agent verfügbar ist, startet pkexec einen eigenen textuellen Auth-Dialog im Terminal. Mit dieser Option wird in so einem Fall statt dessen das Programm <b>nicht</b> ausgeführt (Fehler, wenn kein Agent aktiv) <a href="http://freedesktop.org">freedesktop.org</a> . |
| --help / --<br>version           | Zeigt Hilfe bzw. Version von pkexec.                                                                                                                                                                                                                                                                                                                                                  |

*Hinweise:* pkexec erbt ein Minimal-Umfeld (Environment), um das Einschleusen von Code via Umgebungsvariablen zu verhindern[freedesktop.org](http://freedesktop.org). Display- und Xauth-Variablen werden aus Sicherheitsgründen *nicht* an das Zielprogramm weitergegeben, außer es wurde in den PolKit-Regeln explizit erlaubt (GUI-Programme via pkexec sind daher tricky ohne entsprechende PolKit-Policy). In modernen Distributionen ist pkexec oft als Ersatz für grafische gksu-Aufrufe gedacht, um z.B. aus der Desktop-Umgebung heraus GUI-Administrationsprogramme mit Root-Rechten zu starten (wobei PolKit dann i.d.R. das Benutzer-Passwort abfragt).

### **Berechtigungen und Zugriffskontrolle**

#### **umask (Set Default Permissions Mask)**

**Beschreibung:** Legt eine Berechtigungsschutzmaske fest, die bei neu erstellten Dateien und Verzeichnissen die Standardzugriffsrechte beeinflusst. umask subtrahiert Bits von

den vollen Zugriffsrechten. So bestimmt sie, welche Rechte *NICHT* automatisch gesetzt werden dürfen. Dies wirkt pro Prozess/Session. Als Shell-Builtin zeigt umask ohne Parameter die aktuelle Maske an.

#### Syntax:

umask [Option] [Maske]

#### Optionen / Verwendung:

- Ohne Argument zeigt umask die aktuelle Mask in Oktalnotation an (z.B. 0022). Mit Option **-S** in symbolischer Form (z.B. u=rwx,g=rx,o=rx entspricht Maske 022).
- Mit Angabe einer neuen Maske (z.B. umask 027) wird diese gesetzt. Die Maske wird meist oktal angegeben: jedes Bit, das in der Maske 1 ist, nimmt der neu erstellten Datei die entsprechende Berechtigung weg. Beispiel: Maske 022 bedeutet, neuen Dateien werden keine Gruppe- und Andere-Schreibrechte gegeben (d.h. max Resultatrechte für Datei typ. 644, für Verzeichnis 755).
- **Symbolische Maske:** Man kann auch wie bei chmod symbolisch angeben, z.B. umask u=rwx,g=rx,o=rx für dasselbe Ergebnis (Maske 022).

*Hinweis:* Die Umask wirkt auf Systemaufruf-Ebene: Standard-Dateirechte sind 666 (rw-rw-rw) und für Verzeichnisse 777 (rwxrwxrwx); die Maske wird davon abgezogen. Z.B. Umask 002 ergibt Dateien mit 664 (rw-rw-r-) und Verzeichnisse mit 775. Umask 077 würde sehr restriktiv alles nur für den Benutzer selbst erlauben (Datei 600, Dir 700).

#### chmod (Change Mode - Dateirechte ändern)

**Beschreibung:** Ändert die Zugriffsberechtigungen einer Datei oder eines Verzeichnisses. Man kann mit chmod sowohl numerisch (Oktalwerte) als auch symbolisch (u/g/o +-/= r,w,x) die Rechte setzen.

#### Syntax:

chmod [Optionen] <Modus> <Datei/Verzeichnis>...

#### Optionen:

| Option                 | Beschreibung                                                                                                                                            |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-R, --recursive</b> | Rekursive Rechteänderung: traversiert Unterverzeichnisse und wendet chmod auf alle darin befindlichen Dateien/Ordner an. Wichtig bei Verzeichnisbäumen. |
| <b>-v, --verbose</b>   | Gibt für jede Datei eine Meldung aus, deren Rechte erfolgreich geändert wurden (und ggf. was geändert wurde).                                           |

| Option                             | Beschreibung                                                                                                                                  |
|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-c, --changes</b>               | Wie verbose, aber nur Ausgaben für tatsächlich geänderte Rechte (keine Ausgabe, wenn Rechte schon so eingestellt waren).                      |
| <b>-f, --silent</b>                | Unterdrückt Fehlermeldungen (force).                                                                                                          |
| <b>--reference=&lt;Vorlage&gt;</b> | Setzt die gleichen Berechtigungen wie die Referenz-Datei <Vorlage> hat, auf die angegebenen Dateien. Statt Modus muss man --reference nutzen. |

### Modus-Angaben:

- **Numerisch (Oktal):** Eine dreistellige (oder vierstellige mit Sonderbit) Oktalzahl. Z.B. chmod 640 datei.txt setzt Rechte auf rw- r-- --- (6=rw,4=r,0=kein Recht). Jede Ziffer entspricht Owner, Group, Others. Beispiel: chmod 755 script.sh => rwx für Besitzer (7), rx für Gruppe (5), rx für andere (5). Eine führende vierte Ziffer kann spezielle Bits setzen: 4 für Setuid, 2 für Setgid, 1 für Sticky-Bit (z.B. 1755 für Sticky).
- **Symbolisch:** Format: [ugoa]\*[+-=][rwxXst]\* evtl. durch Kommata getrennt für mehrere Operationen in einem Aufruf. Beispiele:
  - chmod u+x,g-w,o= file – dem Besitzer Ausführungsrecht geben, der Gruppe Schreibrecht entziehen, andere bekommen exakt keine Rechte.
  - chmod a+r file – allen (a für all) Leserechte hinzufügen (ohne andere Bits zu ändern).
  - chmod g=u Datei – setzt die Gruppenrechte gleich den Benutzerrechten der Datei (nützliche Kurzform).
  - X: Spezialflag in symbolischer Notation – steht für "execute if directory or if any execute bit is set on any of owner/group/other". Wird meist bei -R verwendet, um rekursiv nur Verzeichnissen das x zu geben.
  - s für setuid/setgid (in u/g Kontext), t für Sticky-Bit (im o-Kontext).
  - Beispiel: chmod u+s script.sh – setzt das setuid-Bit auf die Datei (falls ausführbar).

### chown (Change Owner)

**Beschreibung:** Ändert den Dateibesitzer und/oder die Gruppe einer Datei oder eines Verzeichnisses

file-4fdk7rvx39azjretinarqbfile-4fdk7rvx39azjretinarqb. chown kann sowohl den User als auch die Gruppe in einem Befehl setzen. Nur root (oder Capability CHOWN) darf Eigentümer von Dateien ändern. Normale Benutzer können in der Regel

mittels chgrp die Gruppenzugehörigkeit ändern, wenn sie Mitglied der Zielgruppe sind (und Besitzer der Datei bleiben).

#### Syntax:

```
chown [Optionen] <neuerEigentümer>[:<neueGruppe>] DATEI...
```

#### Optionen:

| Option                                      | Beschreibung                                                                                                                                                                                                                   |
|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-R</b>                                   | Rekursiv – wendet die Besitzänderung auf alle Dateien und Unterverzeichnisse an (ganzer Baum).                                                                                                                                 |
| <b>-h</b>                                   | Ändert den Besitzer von Symbolic Links selbst, statt des Ziels. (Auf den meisten Linux ist das Standard mit Kernel 2.6+; teils wird -h ignoriert, da chown auf Symlinks ohnehin Link selbst ändert, außer mit -L Option).      |
| <b>-f</b>                                   | Keine Fehlermeldungen ausgeben (silent/force).                                                                                                                                                                                 |
| <b>-v</b>                                   | Verbose – listet jede Datei mit geänderten Owner/Group auf.                                                                                                                                                                    |
| <b>--</b>                                   | Ändert nur, wenn aktuelle Owner/Group genau wie angegeben ist. (Damit kann man selektiv bestimmte Kombis ersetzen, dient als Sicherheitsprüfung).                                                                              |
| <b>from=&lt;old_owner:&gt;old_group&gt;</b> | <i>Follow symlinks</i> : Symbolische Links auflösen (statt Link selbst zu ändern wird das Ziel geändert). Standardverhalten für die meisten chown-Versionen ist bereits, dem Link <i>nicht</i> zu folgen (außer man setzt -L). |
| <b>--dereference / -L</b>                   | <i>Don't follow symlinks</i> (explizit Link behandeln, Standard).                                                                                                                                                              |
| <b>-P</b>                                   | Folgen von Symlinks <i>im Kommandozeilen-Argument</i> (Top-Level), aber nicht rekursiv tiefer. (Selten benötigt).                                                                                                              |
| <b>-H</b>                                   |                                                                                                                                                                                                                                |

#### Verwendung:

- Besitzer ändern: chown alice datei.txt – setzt *alice* als neuen Besitzer der Datei (Gruppe bleibt unverändert).
- Gruppe ändern mit chown: chown :developers projektX – behält Besitzer, setzt Gruppe auf *developers*. Notation: Vor dem Doppelpunkt nichts heißt Besitzer unverändert, aber Gruppe ändern.

- Besitzer und Gruppe zugleich: chown alice:staff datei.txt – Besitzer = alice, Gruppe = staff.
- Nur Gruppe ändern könnte auch mit eigenem Befehl chgrp staff datei.txt gemacht werden.

**Besonderheit:** Man kann numerische IDs angeben. chown 1001:100 datei setzt Besitzer UID=1001, GID=100. Falls man einen Benutzer- oder Gruppennamen mit Doppelpunkt hat, der numerisch beginnt, kann man -- verwenden um Verwechslung zu vermeiden oder explizit username: schreiben.

### **chgrp (Change Group)**

**Beschreibung:** Ändert die Gruppenzugehörigkeit einer Datei oder eines Verzeichnissesfile-4fdk7rvx39azjretinarqbfile-4fdk7rvx39azjretinarqb. Äquivalent zu chown :<gruppe>. Wird benutzt, um den Gruppeneigentümer anzupassen.

#### **Syntax:**

chgrp [Optionen] <neueGruppe> DATEI...

#### **Optionen:**

| Option              | Beschreibung                                                                                                                                                               |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -R                  | Rekursive Änderung der Gruppe in Unterverzeichnissen (alle Dateien/Dirs innerhalb werden angepasst).                                                                       |
| -f                  | Keine Fehlermeldungen bei Problemen ausgeben (silent).                                                                                                                     |
| -v                  | Ausführliche Ausgabe pro Datei.                                                                                                                                            |
| --reference=Vorlage | Übernimmt die Gruppenzugehörigkeit der <i>Vorlage</i> -Datei auf alle angegebenen Dateien. (Ignoriert das Gruppen-Argument und nutzt stattdessen die Gruppe der Referenz.) |

**Hinweis:** Normale Benutzer dürfen chgrp nur verwenden, um eine Datei auf eine Gruppe zu setzen, der sie selbst angehören. Root kann beliebige Zuordnungen vornehmen.

### **lsattr (List Attributes)**

**Beschreibung:** Listet die *erweiterten Dateiattribute* (Filesystem-Attribute) von ext-Dateisystemen auffile-4fdk7rvx39azjretinarqb. Diese Attribute wie *immutable* oder *append-only* steuern besondere Verhalten auf niedriger Ebene (nur auf bestimmten FS wie ext2/3/4, XFS - mit anderen Tools - etc. vorhanden). lsattr zeigt pro Datei eine Zeichenfolge von Attribut-Flags.

#### **Syntax:**

lsattr [Optionen] [Datei/Verzeichnis]...

## Optionen:

### Option Beschreibung

- a** Zeigt auch versteckte Dateien (Dateien, die mit Punkt beginnen) an.  
Standardmäßig ignoriert lsattr diese analog zu ls ohne -a.
- d** Listet das Verzeichnis selbst auf, nicht dessen Inhalt. (Bei Angabe eines Verzeichnisses wird normalerweise dessen Inhalt rekursiv aufgelistet; mit -d sieht man nur die Attribute des Verzeichniseintrags selbst)file-4fdk7rvx39azjretinarqb.
- R** Rekursive Ausgabe der Attribute in allen Unterverzeichnissen.
- v** Zeigt die Dateiversionsnummer (Version im ext-Format) mit an, falls vorhanden.  
(Das Attribut 'version' wird kaum benutzt, -v gibt eine zusätzliche Spalte mit Versionsnummer).
- V** Ausführliche Informationen über die Version des Programms und detaillierte Fehler ausgeben (verbose).

## Beispiel Ausgabe:

```
$ lsattr wichtige_datei
----i-----e-- wichtige_datei
```

Hier bedeuten die Buchstaben: **i** = immutable (unveränderlich), **e** = extents (standard auf ext4). Bindestriche zeigen nicht gesetzte Flags. Wichtige Flags: **i** (immutable: Datei kann nicht verändert/gelöscht umbenannt werden bis Attribut entfernt), **a** (append-only: nur anhängen erlaubt), **d** (no dump: Datei wird von Dump-Backup ignoriert), **A** (no atime updates), **S** (synchronous updates) etc.

## chattr (Change Attributes)

**Beschreibung:** Ändert die erweiterten Dateisystem-Attribute (wie immutable, append-only etc.) unter Linux auf ext2/ext3/ext4 Dateisystemenfile-4fdk7rvx39azjretinarqb. Nur Root darf die meisten dieser Attribute setzen.

## Syntax:

```
chattr [Optionen] [+--Attribute] Datei...
```

## Optionen:

### Option      Beschreibung

- R** Rekursive Attributänderung in Unterverzeichnissenfile-4fdk7rvx39azjretinarqb.

| Option                | Beschreibung                                                                   |
|-----------------------|--------------------------------------------------------------------------------|
| <b>-v</b>             | Ausführlich – zeigt an, welche Änderungen für welche Datei vorgenommen wurden. |
| <b>-f</b>             | Unterdrückt Fehlermeldungen (force).                                           |
| <b>-v</b>             | Setzt die Versionsnummer der Datei (falls vom Dateisystem unterstützt).        |
| <b>&lt;Nummer&gt;</b> | Selten benutzt – ext-Dateisysteme haben optional Dateiversionsnummern.         |

**Attribute setzen (Symbolik):** Man gibt nach möglichen Optionen eine Zeichenfolge an, der ein +, - oder = vorangestellt ist:

- + bedeutet diese Attribute *hinzufügen* (zusätzlich zu bestehenden).
- - bedeutet diese Attribute *entfernen*.
- = bedeutet *exakt setzen* (alle aktuellen Attribute durch diese ersetzen).

Die wichtigsten Attribut-Buchstaben auf ext4 sind:

- **i – immutable:** Datei kann nicht modifiziert, umbenannt oder gelöscht werden file-5y1edouysrow4zj6phjcqm. Prozesse können sie nicht öffnen im Schreibmodus. Schützt kritische Dateien (z.B. /etc/passwd) vor Änderungen, sogar von root (root muss erst chattr -i machen).
- **a – append only:** Datei kann nur angehängt, aber nicht überschrieben oder gelöscht werden. Typisch für Logfiles, damit niemand alte Einträge löscht.
- **e – extent format:** zeigt an, dass die Datei extents benutzt (automatisch auf ext4, nicht manuell gesetzt).
- **d – no dump:** Datei soll von Backup-Programmen, die dem Dump-Standard folgen, ignoriert werden.
- **A – no atime updates:** beim Lesen der Datei wird der atime-Zeitstempel nicht aktualisiert (Performance-Optimierung).
- **S – synchronous updates:** Änderungen an der Datei werden synchron auf den Datenträger geschrieben (wie bei chattr +S – ähnlich wie Ordner mit Dirsync mount option).
- **t – no tail-merging:** (relevant bei ext-Dateisystem, deaktiviert Tail-Merging bei Reverse-Mapping; selten benutzt).
- **T – top of directory hierarchy:** Verzeichnis wird als Top-Level für Ordnungszwecke in ext behandelt (verhindert bestimmte Ordnungsoptimierungen; selten).

- **u – undeletable**: Dateiinhalt soll beim Löschen inhaltlich gesichert werden, sodass man ihn wiederherstellen könnte. (Wird von ext4 i.d.R. nicht implementiert).

#### **Beispiel:**

- chattr +i datei.txt – setzt *immutable*.
- chattr -i datei.txt – entfernt dieses Flag wieder (dann ist Bearbeiten/Löschen wieder möglich).
- chattr =a Datei.log – setzt **nur** append-only und entfernt alle anderen Attribute (falls welche gesetzt waren). Danach kann in Datei.log nur noch angehängt werden.

### **getfacl (Get File ACL)**

**Beschreibung:** Liest die Access Control Lists (erweiterten Zugriffsrechte) von Dateien oder Verzeichnissen aus und zeigt sie anfile-4fdk7rvx39azjretinarqb. Auf Linux-Systemen mit POSIX-ACL-Unterstützung (z.B. ext4 mit acl-Option eingehängt) kann man pro Datei feinere Berechtigungen für einzelne Benutzer oder Gruppen setzen. getfacl zeigt diese ACL-Einträge zusätzlich zu den Standardrechten.

#### **Syntax:**

getfacl [Optionen] Datei...

#### **Optionen:**

| <b>Option</b>                   | <b>Beschreibung</b>                                                                                                                                          |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-a</b>                       | Zeigt auch die <i>verschachtelten ACLs</i> (falls vorhanden – nicht üblich bei POSIX ACL).                                                                   |
| <b>-e</b>                       | Zeigt die Ausgabe im <i>effective</i> Modus – Standard. (Jeder Eintrag inkl. "effective permissions", relevant wenn Mask gesetzt ist).                       |
| <b>-s</b>                       | Stripschutz: Zeigt nur die tatsächlichen ACL-Einträge, ohne die Grundeinträge (user/group/other).                                                            |
| <b>-d</b>                       | Zeigt die <i>Default-ACL</i> eines Verzeichnisses (die Vorlage, die an neu darin erzeugte Dateien vererbt wird). In der Ausgabe mit default: gekennzeichnet. |
| <b>--omit-</b><br><b>header</b> | Unterdrückt den Kommentar-Kopf in der Ausgabe, der Dateinamen und Hinweise beinhaltet.                                                                       |
| <b>-R</b>                       | Rekursiv über Verzeichnisse, getfacl für jede Datei auflisten.                                                                                               |

#### **Ausgabe-Beispiel:**

```

$ getfacl projekt/
file: projekt/
owner: alice
group: developers
user::rwx
user:bob:rwx- # bob hat rw Rechte zusätzlich
group::r-x
mask::rw- # effektive Maske für ACL = rw
other::---
default:user::rwx # Default ACL für neu angelegte Objekte
...

```

Hier sieht man die Standard-Owner/Group-Einträge und eine explizite ACL, die User bob Schreibrechte gibt, sowie eine *mask*, die die maximal wirksamen Rechte auf rw beschränkt.

### **setfacl (Set File ACL)**

**Beschreibung:** Setzt oder ändert Access Control Lists (ACLs) von Dateien und Verzeichnissen file-4fdk7rvx39azjretinarqb. Damit kann man zusätzliche Berechtigungen für bestimmte Nutzer oder Gruppen vergeben (oder entziehen), die über die klassischen Besitzer-Gruppe-Andere Rechte hinausgehen.

#### **Syntax:**

```
setfacl [Optionen] { -m|-x <ACL-Eintrag> ... | -b | -k } Datei...
```

#### **Optionen:**

##### **Option Beschreibung**

**-m ACL** *Modify:* Fügt einen ACL-Eintrag hinzu oder aktualisiert ihn. Beispiel: setfacl -m u:bob:rwx file gibt Benutzer *bob* rw-Rechte auf *file*. Mehrere -m Einträge oder Einträge mit Komma getrennt sind möglich.

**-x ACL** *Remove:* Entfernt einen spezifischen ACL-Eintrag. Z.B. setfacl -x g:marketing file entfernt den ACL-Eintrag für Gruppe "marketing".

**-b** *Remove all – Löscht alle* ACL-Einträge (außer den Basis-Einträgen für Owner/Group/Other). Damit werden alle erweiterten Berechtigungen zurückgesetzt. (Datei hat dann nur noch die klassischen Unix-Rechte.)

## Option Beschreibung

- k      Remove Default – Entfernt die Default-ACL eines Verzeichnisses (vererbte ACL-Vorlage). Hat auf Dateien keine Wirkung.
- R      Rekursiv – wendet die ACL-Änderung auf alle Dateien im Verzeichnis (und Unterverzeichnisse) an. Nützlich um z.B. ganze Bäume eine neue ACL zu geben.
- d      Setzt Einträge als *Default-ACL* für Verzeichnisse. Benutzt in Kombination mit -m oder -x. Beispiel: setfacl -m d:g:team:rwx projekt setzt eine Default-ACL, so dass neue Dateien unter *projekt*/ der Gruppe *team* rw Rechte erben, und Verzeichnisse rwx.
- mask      Berechnet und setzt die ACL-Maske basierend auf den aktuellen Gruppen-Einträgen neu. (Die Maske begrenzt die maximalen Rechte der ACL-Einträge außer Owner/other). Standardmäßig wird die Maske auto-anangepasst.
- n      *Don't recalc mask*: Verhindert, dass setfacl die ACL-Maske automatisch anpasst. Normalerweise soll man diese Option nicht verwenden, außer man setzt die Maske manuell mit -m m::....

**Beispiel:** setfacl -m u:bob:rw,g:marketing:r-- report.txt – gibt Benutzer *bob* Lese- und Schreibrecht und der Gruppe *marketing* nur Leserecht auf *report.txt*.

*Hinweis:* Nicht alle Dateisysteme unterstützen POSIX-ACL. Bei ext4, XFS etc. muss i.d.R. das Volume mit Option acl eingehängt sein (moderne Systeme haben das meist per Default). Mit getfacl kann man prüfen, ob vorhandene ACLs existieren.

## Datei- und Verzeichnisverwaltung

### stat (Dateistatus anzeigen)

**Beschreibung:** Zeigt detaillierte Meta-Informationen zu Dateien oder Dateisystemen an. stat gibt z.B. Größe, Blocks, Zeitstempel (Zugriff, Änderung, Statusänderung), Berechtigungen in Zahlenform, Device-ID, Inode-Nummer usw. in strukturierter Form aus.

#### Syntax:

stat [Optionen] <Datei>...

#### Optionen:

| Option            | Beschreibung                                                                                                                                                          |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -f, --file-system | Zeigt anstelle der Datei-Details die Informationen zum Dateisystem, auf dem die Datei liegt. Z.B. Statistiken wie Blockgröße, Gesamtblöcke, freie Blöcke, FS-Typ etc. |

| Option                             | Beschreibung                                                                                                                                                                                                                                                                                                                          |
|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-c FORMAT , --format=FORMAT</b> | Gibt die Ausgabe gemäß <i>FORMAT</i> formatiert aus. Man kann Platzhalter verwenden, um bestimmte Werte auszugeben. Z.B. stat -c "%A %h %U %G %s %n" datei gibt in einer Zeile: mode, Linkanzahl, Owner, Group, Size, Name. (Siehe man stat für alle Platzhalter wie %A = Rechte in rwx, %a = oktal, %U = Ownername, %s = Größe etc.) |
| <b>--printf=FORMAT</b>             | Ähnlich wie --format, jedoch ohne automatisch Zeilenumbruch am Ende. Kann benutzt werden, um mehrere stat-Aufrufe zu skripten.                                                                                                                                                                                                        |
| <b>-t, --terse</b>                 | "Knapp": Gibt die Ausgabe in einer kompakten einzeiligen Form, Felder durch Leerzeichen getrennt, aus (für einfache maschinelle Weiterverarbeitung).                                                                                                                                                                                  |
| <b>--help, --version</b>           | Anzeigen von Hilfe bzw. Version.                                                                                                                                                                                                                                                                                                      |

#### Beispielausgabe:

```
$ stat /etc/hosts
 File: /etc/hosts
 Size: 258 Blocks: 8 IO Block: 4096 regular file
Device: 802h/2050d Inode: 13422506 Links: 1
Access: (0644/-rw-r--r--) Uid: (0/ root) Gid: (0/ root)
Access: 2025-04-01 10:22:13.000000000 +0200
Modify: 2025-04-01 10:22:13.000000000 +0200
Change: 2025-04-01 10:22:15.000000000 +0200
 Birth: -
```

Dies zeigt z.B., dass /etc/hosts 258 Bytes groß ist, auf einem bestimmten Device, inode etc., Rechte 0644, Eigentümer root, Zeitstempel für Zugriffs-/Änderungs-/Änderungszeit (Birth ist Erstellung, nicht unterstützt).

### file (Dateityp bestimmen)

**Beschreibung:** Untersucht eine Datei und versucht den Typ bzw. Format der Datei zu bestimmen. file nutzt einen umfangreichen Magie-Pattern-Katalog (/usr/share/file/magic), um anhand der Bytes einer Datei (Magic Numbers, Struktur etc.) menschlich lesbare Dateitypinformationen auszugeben. Praktisch, um z.B. Binärformate, Skripttypen oder Kodierungen zu erkennen.

#### Syntax:

file [Optionen] <Datei>...

### Optionen:

#### Option Beschreibung

- Zeigt den MIME-Typ der Datei anstatt einer Beschreibung. Z.B. text/plain;
- i** charset=utf-8 statt "UTF-8 Unicode text". Hilfreich für Scripting oder Web-Kontext.
- Brief* – gibt nur die Beschreibung aus, ohne den Dateinamen voranzustellen.
- b** (By default schreibt file: "<Dateiname>: <Beschreibung>"). Mit -b erhält man nur "<Beschreibung>".
- Folgt symbolischen Links und untersucht das Ziel statt den Link selbst.
- L** Standardmäßig wird der Link selbst untersucht und meist als "symbolic link to <target>" ausgegeben.
- (alias --no-dereference) Bewirkt das Gegenteil: behandelt Symlinks wie
- h** normale Dateien (gleichbedeutend mit Standardverhalten, -h ist nützlich falls -L als default kompiliert wäre).
- Untersucht auch Gerätedateien (block/char special) und Auslagerungsdateien.
- s** Standardmäßig werden einige spezielle Dateitypen nicht inhaltlich analysiert (um z.B. nicht eine Device-Datei zu lesen). -s kann riskant sein (kann z.B. /dev/zero "lesen" -> blockiert).
- z** Durchsucht komprimierte Dateien (gzip-komprimierte) – versucht innerhalb eines .gz die Dateisignatur zu prüfen.
- E** Gibt (Exit)Status 1 zurück falls *irgendeine* der Dateien kein bekannter Typ bestimmt werden konnte. (Nützlich in Skripten).
- help,** Hilfe bzw. Versionsnummer und unterstützte Magic-Datei-Version ausgeben.
- v**

### Beispiele:

- file /bin/ls -> **ELF 64-bit LSB executable**, x86-64, dynamically linked ...
- file archive.tar.gz -> **gzip compressed data**, was probably a tar archive (reads inside and erkennt es als tar in gz).
- file picture.jpg -> **JPEG image data**, JFIF standard 1.01, ...

### In (Link erstellen)

**Beschreibung:** Erstellt Verknüpfungen zu Dateien – entweder **Hard Links** oder **Symbolische Links**. Ein Hardlink ist ein weiterer Directory-Eintrag für dieselbe Inode

(nur innerhalb desselben Dateisystems möglich). Ein Symbolischer Link (Symlink) ist eine kleine Spezialdatei, die auf einen Pfad verweist.

#### Syntax:

`ln [Optionen] <Ziel> [Linkname]`

Standard ohne Optionen erstellt *Hardlink* mit Namen *Linkname*, der auf Datei *Ziel* zeigt. Wenn *Linkname* nicht angegeben, wird im aktuellen Verzeichnis ein Link gleichen Namens wie *Ziel* erstellt. Für Symlinks Option -s nutzen.

#### Optionen:

| Option                           | Beschreibung                                                                                                                                                                                                                                                                                                 |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-s, --symbolic</b>            | Erstellt einen <i>symbolischen Link</i> statt eines Hardlinks. Der Link verweist auf den angegebenen <Ziel>-Pfad. Wenn <Ziel> kein absoluter Pfad ist, wird er relativ vom Link aus interpretiert. Beispiel: <code>ln -s /etc/hosts hostslink</code> erzeugt Symlink <i>hostslink</i> -> <i>/etc/hosts</i> . |
| <b>-f, --force</b>               | Existierende Zielfile (Linkname) ggf. entfernen, falls schon vorhanden. Ohne -f würde ln bei existierendem Linknamen einen Fehler ausgeben. (Bei -s mit -f: überschreibt existierenden Link.)                                                                                                                |
| <b>-n, --no-dereference</b>      | Beim Überschreiben eines existierenden symbolischen Links (mit -f) behandle den Link selbst statt dessen Ziel. (Verhindert, dass versehentlich die Datei, auf die ein bestehender Link zeigt, gelöscht wird).                                                                                                |
| <b>-v, --verbose</b>             | Gibt eine Meldung pro verarbeitetem Link aus.                                                                                                                                                                                                                                                                |
| <b>-T, --no-target-directory</b> | Behandelt das letzte Argument immer als <i>genauen Linknamen</i> , selbst wenn es ein existierendes Verzeichnis ist. (Verhindert, dass ln im Zielverzeichnis einen Link mit Standardname erstellt.)                                                                                                          |
| <b>-d, -F, --directory</b>       | Erlaubt es root, auch <i>Verzeichnis-Hardlinks</i> zu erstellen. (Normalerweise verboten – Option nur auf speziellen Systemen mit -f kombinierbar, in der Praxis kaum verwendet).                                                                                                                            |
| <b>-b, --backup</b>              | Erzeugt ein Backup der Zielfile falls -f zum Überschreiben genutzt wird. (Nützlich, um das ersetzte File nicht zu verlieren.) Siehe auch -- backup=t etc.                                                                                                                                                    |
| <b>-S &lt;Suffix&gt;</b>         | Gibt ein Suffix für Backup-Dateien an (Standard ~). Bsp: -b -S .bak.                                                                                                                                                                                                                                         |

#### Beispiele:

- Hardlink: `ln /path/to/original.txt copy.txt` – `copy.txt` ist nun ein weiterer Verweis auf `original.txt` (gleiche Inode). Änderungen an einer erscheinen in der anderen, da es dieselbe Datei ist. Löscht man eine, bleibt die andere erhalten (Datei wird erst gelöscht, wenn letzter Link entfernt).
- Symlink: `ln -s /etc/apache2 siteconfig` – erstellt `siteconfig` → `/etc/apache2`. Aufruf von `ls -l siteconfig` zeigt dann `siteconfig` → `/etc/apache2`.

## **cd (Change Directory) – siehe oben bei Grundlegende Befehle**

(Im Kontext Datei/Verzeichnis-Verwaltung sei auf den bereits beschriebenen Befehl `cd` verwiesen, um Verzeichnisse zu wechseln.)

## **tree (Verzeichnisbaum anzeigen)**

**Beschreibung:** Zeigt die Verzeichnisstruktur grafisch in Form eines Baumes an. `tree` listet rekursiv alle Unterverzeichnisse und Dateien in hierarchischer eingerückter Darstellung. Es erleichtert das Visualisieren der Verzeichnis-Hierarchie.

### **Syntax:**

`tree [Optionen] [Verzeichnis]`

### **Optionen:**

| <b>Option</b> | <b>Beschreibung</b> |
|---------------|---------------------|
|---------------|---------------------|

- |                 |                                                                                                                                                                                                               |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-a</b>       | Zeigt <i>alle</i> Dateien an, inkl. versteckter (auch solche, die mit <code>.</code> beginnen). Standardmäßig ignoriert <code>tree</code> versteckte Dateien, außer <code>.</code> und <code>..</code> immer. |
| <b>-d</b>       | Listet <i>nur Verzeichnisse</i> auf, keine Dateien. Nützlich, um reinen Verzeichnisbaum zu sehen.                                                                                                             |
| <b>-L Level</b> | Beschränkt die Tiefe der Ausgabe auf <i>Level</i> Ebenen. Z.B. <code>tree -L 2</code> zeigt nur zwei Ebenen tief (akt. Verzeichnis + Unterverzeichnisse, aber keine tieferen).                                |
| <b>-f</b>       | Zeigt den vollen Pfad vor jedem Eintrag (statt nur Namen relativ).                                                                                                                                            |
| <b>-I</b>       | Ignoriert Einträge, die auf das Muster passen (Groß/Kleinschreibung <i>PATTERN</i> ) berücksichtigt, mehrere Patterns mit `                                                                                   |
| <b>-P</b>       | Zeigt nur Einträge an, die dem Pattern entsprechen (Whitelist). Umgekehrt zu <i>PATTERN</i> -I.                                                                                                               |
| <b>-F</b>       | Markiert Verzeichnisse mit abschließendem <code>/</code> , ausführbare Dateien mit <code>*</code> , Symlinks mit <code>@</code> , Pipes mit <code>`</code>                                                    |
| <b>-p</b>       | Zeigt UNIX-Dateirechte (Permission bits) vor jedem Eintrag.                                                                                                                                                   |

| <b>Option</b>   | <b>Beschreibung</b>                                                                                                  |
|-----------------|----------------------------------------------------------------------------------------------------------------------|
| <b>-u / -g</b>  | Zeigt Benutzer bzw. Gruppenbesitzer an. (Wie ls -l Infos).                                                           |
| <b>-h</b>       | <i>Human-readable size</i> : Zeigt Dateigrößen in menschenlesbarem Format (K, M, G) anstatt in Bytes (wirkt mit -s). |
| <b>-s</b>       | Zeigt Dateigrößen in Bytes neben jedem Dateinamen an.                                                                |
| <b>--du</b>     | Summiert Verzeichnisgrößen wie du (statt wie normal lediglich die Einzeldateigröße zu zeigen).                       |
| <b>-t</b>       | Sortiert die Ausgabe nach Zeit (Neueste zuerst) statt alphabetisch.                                                  |
| <b>-o Datei</b> | Gibt die Ausgabe in die angegebene Datei (überschreibt) anstatt auf stdout.                                          |
| <b>-n</b>       | Keine farbliche Hervorhebung (Standard ist bunt, -n erzwingt Monochrom).                                             |
| <b>--help</b>   | Hilfe anzeigen (Übersicht der Optionen).                                                                             |

*Beispiel:*

```
$ tree -L 2 /etc
```

```
/etc
├── apache2
│ ├── apache2.conf
│ ├── envvars
│ └── mods-available
├── cron.daily
│ ├── apt-compat
│ └── man-db
└── default
 ├── cron
 └── rcS
```

6 directories, 5 files

Hier sieht man /etc mit zwei Ebenen Tiefgang.

### **mkdir (Make Directory)**

**Beschreibung:** Erstellt ein neues Verzeichnis (oder mehrere) im Dateisystem. Kann rekursiv auch mehrere Ebenen auf einmal anlegen.

**Syntax:**

`mkdir [Optionen] <Verzeichnisname>...`

**Optionen:**

| Option                   | Beschreibung                                                                                                                                                                                                                                                                                                      |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-p, --parents</b>     | Erzeugt auch alle notwendigen übergeordneten Verzeichnisse, falls sie nicht existieren. Somit werden keine Fehler ausgegeben, wenn Verzeichnisse in der Pfadangabe fehlen. Beispiel: <code>mkdir -p /tmp/foo/bar</code> legt sowohl <code>/tmp/foo</code> (falls nicht da) als auch <code>/tmp/foo/bar</code> an. |
| <b>-m, --mode=MODE</b>   | Setzt die Zugriffsrechte des neu erstellten Verzeichnisses auf <i>MODE</i> (numerisch oder symbolisch) anstatt der Standardrechte (die Standardrechte sind i.d.R. 0777 abzüglich Umask). Z.B. <code>mkdir -m 750 projDir</code> erstellt <i>projDir</i> mit rwx für Besitzer, rx für Gruppe, nichts für andere.   |
| <b>-v, --verbose</b>     | Ausgabe einer Meldung für jedes angelegte Verzeichnis (zeigt Pfad an).                                                                                                                                                                                                                                            |
| <b>--help, --version</b> | Anzeige von Hilfe bzw. Version.                                                                                                                                                                                                                                                                                   |

**cp (Copy File)**

**Beschreibung:** Kopiert Dateien oder Verzeichnisse. Bei Angabe mehrerer Quelldateien muss das letzte Argument ein Verzeichnis sein (Zielverzeichnis). cp kann auch Verzeichnisbäume rekursiv kopieren.

**Syntax:**

`cp [Optionen] <Quelle>... <Ziel>`

*Wenn <Ziel> ein Verzeichnis ist, werden alle <Quelle>-Dateien dorthin kopiert (mit gleichem Namen). Ist nur eine Quelle und <Ziel> kein Verzeichnis, wird die Datei dorthin kopiert/umbenannt.*

**Optionen:**

| Option                     | Beschreibung                                                                                                                                                                                          |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-r, -R, --recursive</b> | Kopiert Verzeichnisse rekursiv (Inhalt und Unterverzeichnisse). Ohne diese Option kopiert cp standardmäßig keine Verzeichnisse. (GNU cp hat auch -r/-R – beide meist gleichwertig. POSIX fordert -R.) |

| Option                                       | Beschreibung                                                                                                                                                                                                                                                                                               |
|----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-a, --archive</b>                         | Archivmodus: Kopiert <i>rekursiv</i> und bewahrt dabei möglichst alle Attribute. Entspricht -dR --preserve=all (also -R, und -p für timestamps, owner, group, perms, ACLs, xattrs, Links etc.). Dies ist für Vollbackups ideal <a href="#">ionos.at</a> .                                                  |
| <b>-p, --preserve</b>                        | Bewahrt Eigentümer, Gruppe, Berechtigungen und Zeitstempel der Datei beim Kopieren <a href="#">ionos.at</a> . (Ohne -p übernimmt cp Standard: setzt aktuellen Zeitstempel "jetzt" und erstellt als aktueller Benutzer). Kann Argumente haben wie --preserve=mode,ownership,timestamps,links,xattr,all.     |
| <b>-f, --force</b>                           | Überschreibt vorhandene Ziel-Dateien ohne Nachfrage, und versucht auch nicht-schreibbare Ziele zu entfernen, um Kopie dennoch zu platzieren. (Im Konflikt mit -n gilt letzter Schalter.)                                                                                                                   |
| <b>-i, --interactive</b>                     | Fragt nach, bevor eine existierende Zielfile überschrieben wird (Confirm Y/N). Gibt dem Benutzer die Möglichkeit, Überschreiben abzulehnen.                                                                                                                                                                |
| <b>-n, --no-clobber</b>                      | Überschreibt <i>keine</i> existierenden Dateien. D.h. falls Ziel existiert, wird die Kopie für diese Datei ausgelassen. (Verhindert unabsichtliches Überschreiben)                                                                                                                                         |
| <b>-u, --update</b>                          | Kopiert nur, wenn <i>Quelle</i> neuer als <i>Ziel</i> ist, oder Ziel noch nicht existiert. Praktisch für inkrementelle Updates. Existiert Ziel mit gleich/neueren Datum, wird es nicht überschrieben.                                                                                                      |
| <b>-d, --no-dereference</b>                  | Kopiert <i>Symbolic Links</i> als Links (anstatt deren referenzierten Inhalt). Standard: cp folgt symbolischen Links und kopiert die Datei dahinter. Mit -d wird der Link selbst neu erstellt am Ziel (Hardlinks bleiben hardlinks sofern möglich; -d ist in GNU cp gleichbedeutend mit --preserve=links). |
| <b>-L, --dereference</b>                     | <i>Folgt allen Symlinks:</i> kopiert stets den referenzierten Inhalt (Standardverhalten bei regulärem cp für Dateien). Bei Angabe von Verzeichnis mit -R: -L würde bewirken, dass cp <i>alle</i> gefundene Symlinks auflöst und deren Ziel kopiert (kann zu Loops führen, daher mit Vorsicht).             |
| <b>-P, --no-dereference --preserve=links</b> | Kopiert Links als Links (das ist Standard mit -a enthalten). (Im GNU cp -P bedeutet No-dereference; beachten: -d und -P ähnlich).                                                                                                                                                                          |

| <b>Option</b>                        | <b>Beschreibung</b>                                                                                                                                                                                                                        |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>--parents</b>                     | Erhält die Quellpfadstruktur im Ziel. Bsp: cp --parents dir1/dir2/file.txt /backup legt /backup/dir1/dir2/file.txt an <a href="#">man7.org</a> . Nur sinnvoll, wenn Ziel ein Verzeichnis ist.                                              |
| <b>-l, --link</b>                    | Erstellt <i>Hardlinks</i> statt Kopien der Datei, sofern im selben Dateisystem. (Keine Datenkopie, nur neuer Name, ähnlich ln).                                                                                                            |
| <b>-s, --symbolic-link</b>           | Erstellt <i>Symlinks</i> statt die Daten zu kopieren. (Zieldatei wird symbolischer Verweis auf Quelldatei).                                                                                                                                |
| <b>-H, -L, -P (bei -R)</b>           | Steuerung des Symlink-Verhaltens <b>während</b> rekursivem Kopieren: -P (Default) ignoriert Linkinhalt (kopiert Symlinks als Symlinks), -L folgt Symlinks zu Dateien, -H folgt nur Symlinks angegeben auf Befehlzeile. (-a impliziert -P). |
| <b>--</b><br><b>backup[=Control]</b> | Erzeugt Backup von Zieldateien, die überschrieben würden. (Etwa file.txt~ anlegen). Kann =simple, numbered etc. steuern. Siehe auch VERSION_CONTROL Env.                                                                                   |
| <b>--help, --version</b>             | Hilfe/Version anzeigen.                                                                                                                                                                                                                    |

## **mv (Move/Rename)**

**Beschreibung:** Verschiebt Dateien oder Verzeichnisse an einen neuen Pfad, bzw. benennt sie um. mv kann auch zum einfachen Umbenennen einer Datei verwendet werden (Quell- und Zielpfad auf selbem FS = schneller Rename, kein Kopieren). Falls Quelle und Ziel auf unterschiedlichen Dateisystemen liegen, arbeitet mv durch Kopieren und anschließendem Löschen der Quelle.

### **Syntax:**

mv [Optionen] <Quelle>... <Ziel>

### **Optionen:**

| <b>Option</b>            | <b>Beschreibung</b>                                                                                                                       |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-i, --interactive</b> | Fragt nach Bestätigung, falls eine Zieldatei durch den Move überschrieben würde (falls im Zielpfad eine Datei gleichen Namens existiert). |
| <b>-f, --force</b>       | Erzwingt das Überschreiben ohne Nachfrage. Ziel-Dateien werden ohne Warnung ersetzt. (Standardverhalten ohne -i).                         |
| <b>-n, --no-clobber</b>  | Überschreibt keine bestehenden Dateien. Bei Namenskollision bricht mv für diese Datei ab (gibt keinen Fehler, exit code                   |

| <b>Option</b>                   | <b>Beschreibung</b>                                                                                                                                                                                                     |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                 | trotzdem 0). In GNU mv hat -n Vorrang vor -i falls beide angegeben.                                                                                                                                                     |
| <b>-u, --update</b>             | Bewegt/überschreibt nur, wenn die Quelle neuer ist als die vorhandene Zielfile <i>oder</i> die Zielfile nicht existiert. (So werden neuere Versionen moved, ältere Quellen, die schon am Ziel sind, bleiben unberührt.) |
| <b>-v, --verbose</b>            | Gibt aus, welche Datei wohin verschoben/umbenannt wurde (für jede Operation eine Zeile).                                                                                                                                |
| <b>--</b>                       | Wie bei cp: legt Backup der Ziel-Datei an, falls existiert, bevor sie überschrieben wird. CONTROL kann none, numbered, existing, simple sein (siehe Doku oder ENV VERSION_CONTROL).                                     |
| <b>backup[=CONTROL]</b>         |                                                                                                                                                                                                                         |
| <b>--strip-trailing-slashes</b> | Entfernt evtl. nachgestellte Schrägstriche bei den Pfadangaben der Quellen (um Verwirrung mit Verzeichnissen zu vermeiden). Standard: neuere mv machen das ohnehin.                                                     |
| <b>--help, --version</b>        | Hilfe/Version.                                                                                                                                                                                                          |

*Hinweis:* mv kann mehrere Dateien in ein Zielverzeichnis verschieben oder eine einzelne Datei/Ordner umbenennen. Bei Verzeichnissen rekursiv, aber mv muss dafür nicht extra Option haben (im Gegensatz zu cp), da der Move innerhalb desselben FS nur Metadaten ändert. Bei verschiedenen FS kopiert mv im Hintergrund (rename() Syscall schlägt dann fehl, worauf cp+unlink passiert).

### rmdir (Remove Directory)

**Beschreibung:** Löscht *leere* Verzeichnisse. Im Gegensatz zu rm entfernt rmdir nur Verzeichnisse und auch nur, wenn sie keine Dateien/Unterverzeichnisse mehr enthalten. Es kann optional ganze verschachtelte Verzeichnisbäume entfernen, *sofern* diese leer sind (Eltern werden dann rekursiv entfernt).

#### Syntax:

rmdir [Optionen] <Verzeichnis>...

#### Optionen:

| <b>Option</b>        | <b>Beschreibung</b>                                                                                                                                                                                                       |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-p, --parents</b> | Entfernt das angegebene Verzeichnis und <i>ggf.</i> dessen übergeordnete Verzeichnisse, falls diese nach jedem Entfernen leer werden. Beispiel: rmdir -p projekt/build/output versucht erst output zu löschen. Ist danach |

| Option                            | Beschreibung                                                                                                                                                                 |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                   | build leer, wird auch build gelöscht. Ist danach projekt leer, wird auch projekt gelöscht.                                                                                   |
| <b>-v, --verbose</b>              | Ausgabe einer Meldung für jedes gelöschte Verzeichnis.                                                                                                                       |
| <b>--ignore-fail-on-non-empty</b> | Unterdrückt die normale Fehlermeldung, falls ein Verzeichnis nicht leer ist. (Das Verzeichnis wird nicht gelöscht, aber es kommt kein Fehlerstatus; mit Vorsicht verwenden). |

*Nützlich:* Um verschachtelte leere Ordner zu entfernen. Aber um Verzeichnisse mit Inhalt zu löschen, nutzt man rm -r.

### rm (Remove Files/Directories)

**Beschreibung:** Löscht Dateien oder Verzeichnisse. Standardmäßig entfernt rm nur Dateien; mit -r kann es auch Verzeichnisbäume löschen (inkl. aller enthaltenen Dateien). **Vorsicht:** Gelöschte Dateien werden nicht ohne Weiteres wiederherstellbar gespeichert (kein Papierkorb, außer man nutzt safe-rm Tools).

#### Syntax:

rm [Optionen] <Datei/Pfad>...

#### Optionen:

| Option                     | Beschreibung                                                                                                                                                                                                                                                                                                                                  |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-r, -R, --recursive</b> | Löscht Verzeichnisse <i>rekursiv</i> mitsamt ihrem gesamten Inhalt. Damit können komplette Verzeichnisbäume entfernt werden. Ohne -r verweigert rm das Löschen von Verzeichnissen mit Fehler "is a directory".                                                                                                                                |
| <b>-d</b>                  | Löscht ein <i>leeres</i> Verzeichnis. (Ohne -r löscht rm normalerweise gar keine Verzeichnisse; -d erlaubt es aber für leere, ähnlich rmdir. Nützlich, aber -r deckt diesen Fall auch ab.)                                                                                                                                                    |
| <b>-f, --force</b>         | <i>Force/ignore nonexistent:</i> Unterdrückt Rückfragen und Fehlermeldungen. Existiert eine angegebene Datei nicht, wird es ignoriert. Schreibschutz wird ignoriert (d.h. es fragt nicht nach Bestätigung bei schreibgeschützten Dateien). Kombiniert mit -r sehr gefährlich, da es kompromisslos alles löscht. Wird oft in Skripten genutzt. |
| <b>-i</b>                  | Interaktiv nachfragen vor <i>jedem</i> Löschvorgang. (Für viele Dateien unpraktisch).                                                                                                                                                                                                                                                         |

| Option                   | Beschreibung                                                                                                                                                                                                                                             |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-I</b>                | "ein wenig interaktiv": Fragt nur einmal pauschal nach, <b>wenn</b> mehr als 3 Dateien gelöscht werden sollen <i>oder</i> ein rekursiver Lösch auf einem Verzeichnis durchgeführt wird. (GNU rm Feature als Sicherheitsnetz)file-5y1edouysrow4zj6phjcqm. |
| <b>--preserve-root</b>   | Schützt vor rm -rf /. Diese Option ist in GNU rm standardmäßig aktiv – es verweigert das komplette Löschen des Root-FS. Man kann mit --no-preserve-root überschreiben (nicht empfohlen).                                                                 |
| <b>-v, --verbose</b>     | Zeigt für jede Datei eine Meldung an, dass sie gelöscht wurde.                                                                                                                                                                                           |
| <b>--help, --version</b> | Hilfe bzw. Version.                                                                                                                                                                                                                                      |

*Anwendung:*

- **Einzelne Datei löschen:** rm datei.txt.
- **Mehrere Dateien löschen:** rm \*.o (löscht alle .o Dateien im aktuellen Verzeichnis).
- **Verzeichnis und Inhalt löschen:** rm -r docs/ – löscht Ordner docs mitsamt allen Unterdateien.
- **Ohne Rückfrage alles löschen:** rm -rf tempdir. (Vorsicht!).

### **head (Dateianfang anzeigen)**

**Beschreibung:** Gibt die ersten Zeilen einer Datei aus (standardmäßig 10 Zeilen). head ist nützlich, um schnell die ersten Zeilen (z.B. Header einer CSV, Beginn eines Logs etc.) anzuschauen, ohne die ganze Datei zu öffnen. Es kann auch binäre Dateien auf Anfang bytes prüfen.

#### **Syntax:**

head [Optionen] [Datei]...

Ohne Datei liest head von der Standardeingabe (z.B. in Pipes). Bei mehreren Dateien wird jeweils ein Header mit dem Dateinamen ausgegeben.

#### **Optionen:**

| <b>Option</b>                | <b>Beschreibung</b>                                                                                                                                                                                                                                         |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-n K , --lines=K</b>      | Gibt die ersten <i>K</i> Zeilen aus. <i>K</i> kann eine Zahl sein oder -K für alle bis auf die letzten <i>K</i> Zeilen (negativer Wert). Standard ist 10. Beispiel: head -n 20 file zeigt 20 Zeilen, head -n -5 file zeigt alle außer der letzten 5 Zeilen. |
| <b>-c Bytes , *--bytes=B</b> | Gibt die ersten <i>B</i> Bytes aus (statt Zeilen). Kann auch mit K, M, etc. Suffix (Potenzen von 1024) angegeben werden. Z.B. head -c 100K bigfile. Negative Werte analog: -c -10 alles außer letzte 10 Bytes.                                              |
| <b>-q , --quiet/-silent</b>  | Unterdrückt die Überschriften, die vor dem Inhalt stehen, wenn mehrere Dateien angegeben wurden. Normal zeigt head sowas wie "=> filename <==" vor dem Block jeder Datei. Mit -q wird alles aneinandergehängt ohne solche Banner.                           |
| <b>-v , --verbose</b>        | Zeigt immer die Dateiüberschriften an, auch wenn nur eine Datei angegeben ist. (Normalerweise bei nur einer Datei keine Überschrift).                                                                                                                       |
| <b>--help , --version</b>    | Hilfe bzw. Version ausgeben.                                                                                                                                                                                                                                |

*Beispiel:* head -n 3 /etc/passwd zeigt die ersten 3 Zeilen der passwd-Datei (typisch Root, daemon, bin Accounts).

### **tail (Dateiende anzeigen)**

**Beschreibung:** Gibt die letzten Zeilen einer Datei aus (standardmäßig 10 Zeilen). Sehr nützlich, um z.B. das Ende von Logfiles zu sehen. Kann im *Follow-Modus* (-f) genutzt werden, um fortlaufend neuen Inhalt anzuzeigen (Echtzeit-Log-Überwachung).

#### **Syntax:**

tail [Optionen] [Datei]...

#### **Optionen:**

| <b>Option</b>            | <b>Beschreibung</b>                                                                                                                                                                               |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-n K , --lines=K</b>  | Zeigt die letzten <i>K</i> Zeilen an. Negative Werte bedeuten ab einer bestimmten Zeile von Anfang an (ab Zeile K). Standard ist 10. Z.B. tail -n 50 /var/log/syslog zeigt die letzten 50 Zeilen. |
| <b>-c B , *--bytes=B</b> | Zeigt die letzten <i>B</i> Bytes an. Kann auch negative (ab Byte X) sein.                                                                                                                         |
| <b>-f , --follow</b>     | Bleibt nach dem Anzeigen der letzten Zeilen geöffnet und zeigt fortlaufend neue Zeilen an, falls die Datei wächst (wie tail -f logfile zum Mitlesen).                                             |

| Option                               | Beschreibung                                                                                                                                                                                                                   |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-F</b>                            | Ähnlich wie -f, aber zusätzlich "Wiederöffnen", falls Datei rotiert/neu erstellt wurde. (Follow Name statt Descriptor; praktisch für Logrotation – tail -F /var/log/app.log folgt auch nach Neustart des Logs dem neuen File). |
| <b>-s Sek , --sleep-interval=Sek</b> | Im Follow-Modus: Warte Sek Sekunden zwischen den Überprüfungen auf neue Daten. Standard 1.0 Sekunde. Kann z.B. auf 0.1 verringert werden, um schneller zu pollen, oder erhöht, um Last zu reduzieren.                          |
| <b>--max-unchanged-stats=N</b>       | (Bei -F relevant) Wie oft in Folge unveränderte Inode geprüft wird, bevor tail davon ausgeht, dass ggf. Logrotate stattgefunden hat.                                                                                           |
| <b>-q, --quiet</b>                   | Unterdrückt Überschriften bei mehreren Dateien (analog head).                                                                                                                                                                  |
| <b>-v, --verbose</b>                 | Erzwingt Überschriften, selbst bei einer Datei.                                                                                                                                                                                |
| <b>--help, --version</b>             | Hilfe bzw. Version.                                                                                                                                                                                                            |

**Anwendung:**

- **Letzte Zeilen einer Datei:** tail -n 20 messages.log (Zeigt die letzten 20 Zeilen).
- **Live-Verfolgung:** tail -f /var/log/syslog (Druckt neu geschriebene Logzeilen fortlaufend). Abbruch mit Ctrl+C.
- **Combine mit grep:** z.B. tail -f server.log \| grep "ERROR" filtert live nur Zeilen mit "ERROR".

## Dateien finden und Text filtern

### grep (Global Regular Expression Print)

**Beschreibung:** Durchsucht die Eingabe (Dateien oder Standard Input) nach Zeilen, die einem bestimmten Muster entsprechen, und gibt diese Zeilen ausfile-4fdk7rvx39azjretinarqb. grep arbeitet mit regulären Ausdrücken (Standard *Basic Regex*, erweiterte mit -E). Es ist das klassische Suchwerkzeug in Textstreams. Name stammt von der ed-Editor Anweisung g/re/p (global regex print).

#### Syntax:

grep [Optionen] 'Muster' [Datei...]

#### Optionen:

| Option                           | Beschreibung                                                                                                                                                                                                                                                           |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-i, --ignore-case</b>         | Ignoriert Groß-/Kleinschreibung beim Mustervergleichfile-4fdk7rvx39azjretinarqb. "Test" findet auch "test" oder "TEST".                                                                                                                                                |
| <b>-v, --invert-match</b>        | Gibt alle Zeilen <b>aus, die nicht</b> auf das Muster passen (Inverse Treffer)file-4fdk7rvx39azjretinarqb. Nutzt man oft zur Ausschlussfilterung.                                                                                                                      |
| <b>-r, -R, --recursive</b>       | Durchsucht rekursiv alle Dateien in angegebenen Verzeichnissen (und Unterverz.)file-4fdk7rvx39azjretinarqb. Ohne diese Option behandelt grep Verzeichnisargumente normalerweise als Fehler oder ignoriert sie. (GNU grep: -r folgt Symlinks nicht, -R folgt Symlinks.) |
| <b>-n, --line-number</b>         | Zeigt vor jeder gefundenen Zeile die Zeilennummer in der Datei an <a href="#">ibm.com</a> . Format: Datei:Nr:Zeilentext.                                                                                                                                               |
| <b>-A NUM</b>                    | (After) Zeigt <i>NUM</i> Zeilen <b>nach</b> jeder Trefferzeile zusätzlich an<br>(Context) <a href="#">ibm.com</a> .                                                                                                                                                    |
| <b>-B NUM</b>                    | (Before) Zeigt <i>NUM</i> Zeilen <b>vor</b> jeder Trefferzeile an.                                                                                                                                                                                                     |
| <b>-C NUM</b>                    | (Context) Zeigt <i>NUM</i> Zeilen <i>vor und nach</i> jedem Treffer (gleichzeitig -A NUM -B NUM). Standard -C 2 beispielsweise gibt 2 Kontextzeilen um jede Fundstelle.                                                                                                |
| <b>-l, --files-with-matches</b>  | Listet nur die Dateinamen auf, die mindestens einen Treffer enthalten. Pro Datei ein Name, Inhalt wird nicht ausgegeben.<br>(Nützlich um zu sehen wo etwas vorkommt)                                                                                                   |
| <b>-L, --files-without-match</b> | Listet nur Dateinamen, die <i>keinen</i> Treffer enthalten (Gegenteil von -l).                                                                                                                                                                                         |
| <b>-c, --count</b>               | Zählt die Treffer pro Datei und gibt nur diese Anzahl aus (zusammen mit Dateinamen, außer mit -h). D.h. "file.txt:3". In Kombination mit -v zählt es Nicht-Treffer-Zeilen.                                                                                             |
| <b>-o, --only-matching</b>       | Gibt nur den Teil der Zeile aus, der dem Suchmuster entspricht, nicht die ganze Zeile. Bei mehreren Treffern pro Zeile werden diese getrennt ausgegeben.                                                                                                               |
| <b>-q, --quiet/--silent</b>      | "Quiet": gibt keine Ausgabe; liefert lediglich Exit-Status 0 falls wenigstens ein Match gefunden wurde, 1 falls kein Match. Praktisch in Skripten (nur Rückgabewert interessiert).                                                                                     |

| Option                       | Beschreibung                                                                                                                                                                                                                                         |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-s, --no-messages</b>     | Unterdrückt Fehlermeldungen über nicht lesbare Dateien (z.B. wenn Zugriffsrechte fehlen oder Datei nicht existiert).                                                                                                                                 |
| <b>-E, --extended-regexp</b> | Interpretiert das Muster als <i>Extended Regular Expression (ERE)</i> <a href="#">ionos.at</a> . Das erlaubt z.B. Nutzung von +, ?, `                                                                                                                |
| <b>-F, --fixed-strings</b>   | Interpretiert das Muster als <i>fixen String</i> (kein Regex, jedes Zeichen literal). Entspricht fgrep (die eine schnellere Suche ohne regex ermöglicht). Alle Muster (wenn -e mehrfach) werden als Literalstrings gesucht.                          |
| <b>-P, --perl-regexp</b>     | Interpretiert das Muster als Perl-Compatible Regex (PCRE), falls grep mit PCRE-Unterstützung kompiliert wurde. Damit stehen mächtigere Regex-Features zur Verfügung.                                                                                 |
| <b>-h, --no-filename</b>     | Unterdrückt die Ausgabe des Dateinamens vor jeder Trefferzeile (wenn mehrere Dateien durchsucht werden). Standardmäßig präfixt grep Datei: bei mehreren Input-Dateien. Mit -h erscheinen nur die Zeilen. (Bei Einzeldatei ist das sowieso Standard.) |
| <b>-H, --with-filename</b>   | Zeigt Dateinamen auch bei einziger Datei an. (Umgekehrt zu -h). Nützlich in Pipelines wo Source unbekannt.                                                                                                                                           |
| <b>--color[=WHEN]</b>        | Hebt das gefundene Pattern farblich hervor in der Ausgabe. WHEN kann auto (Standard, nur wenn Terminal), always oder never sein. Setzt Umgebungsvariable GREP_COLOR/GREP_COLORS ggf. benötigt.                                                       |
| <b>-e Pattern</b>            | Gibt das Suchmuster explizit an. Nötig z.B. wenn das Pattern mit - beginnt (damit es nicht als Option interpretiert wird). Kann mehrfach angegeben werden, um mehrere Muster (OR-Verknüpfung) zu suchen.                                             |
| <b>-f Datei</b>              | Liest Suchmuster aus angegebener Datei (ein Pattern pro Zeile). Nützlich bei sehr vielen oder komplexen Patterns.                                                                                                                                    |
| <b>--exclude=GLOB</b>        | Schließt Dateien vom Durchsuchen aus, deren Name auf GLOB (Wildcard-Muster) passt. Z.B. --exclude=*.min.js würde Minimierte JS ignorieren. (Nur wirksam bei rekursiver Suche oder wenn Verzeichnis als Input). Es gibt auch --include analog.        |

| Option             | Beschreibung                                                                                                                            |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| --exclude-dir=GLOB | Wie --exclude, aber nur auf Verzeichnisnamen angewandt, um ganze Verzeichnisse von der Rekursion auszunehmen (z.B. --exclude-dir=.git). |
| --help, --version  | Hilfe / Version.                                                                                                                        |

**Exit-Status:** 0 falls mindestens ein Match gefunden, 1 falls kein Match, 2 falls ein Fehler auftrat (z.B. Datei nicht lesbar).

### Beispiele:

- grep -i "error" \*.log – sucht case-insensitive "error" in allen .log im aktuellen Ordner, gibt Trefferzeilen mit Dateipräfix aus.
- grep -R "TODO" src/ – findet rekursiv alle Vorkommen von "TODO" in Dateien unterhalb von src/ (typisch in Codebasis).
- grep -nE "^Error|Warning):" report.txt – sucht Zeilenbeginn "Error:" oder "Warning:" und gibt mit Zeilennummer aus [pq.hosting](#).
- ps ax | grep "[c]ron" – findet Prozesse mit "cron" im Namen. (Trick: [c] im Pattern matcht 'c', verhindert aber, dass der grep selbst im ps-Listing gefunden wird, da dessen Kommandozeile "grep cron" kein 'cron' substring hat sondern 'cro n' durch char class unterbrochen).

## find (Dateien suchen)

**Beschreibung:** Durchsucht Verzeichnisse nach Dateien/Verzeichnissen, die bestimmten Kriterien entsprechen (Name, Typ, Größe, Datum, Rechte, Eigentümer etc.) file-4fdk7rvx39azjretinarqb. find ist sehr mächtig: man kann logische Bedingungen kombinieren und Aktionen auf gefundene Dateien ausführen (löschen, verschieben, Exec von Kommandos etc.). Es arbeitet rekursiv ab dem angegebenen Pfad.

### Syntax:

find [Start-Verzeichnisse] [Optionen/Ausdrücke] [Aktion(en)]

### Wichtige Suchkriterien (Ausdrücke):

- **-name Muster** – Dateiname entspricht *Wildcards* (Globbing wie bei Shell, z.B. "\*.txt"). Groß/Kleinschreibung beachten.
- **-iname Muster** – wie -name, aber case-*insensitive*.
- **-type c** – Dateityp ist c. Häufig: **f** für regular file, **d** für directory, **l** symlink, **b** blockdevice, **c** chardevice, **p** named pipe, **s** socket. Z.B. -type d nur Verzeichnisse file-4fdk7rvx39azjretinarqb.

- **-user** *name/uid* – Datei gehört dem Benutzer *name* (oder UID numerisch). Analog **-group** *name/gid*.
- **-uid** *+n/-n/n* – UID der Datei ist (*>/</=*) *n*. **-uid +500** UID größer 500. (Entspricht **-user** nur numerisch und mit Vergleichen). Genauso **-gid**.
- **-size** *N\*[bckMG]\** – Dateigröße-Kriterium. *N* kann mit Vorzeichen angegeben werden (+ größer als, - kleiner als, nichts = genau). Suffix: standardmäßig in 512-Byte Blöcken, oder **c** = Bytes, **k** = KiB, **M** = MiB, **G** = GiB. Beispiel: **-size +100M** Dateien größer 100 MiB[wiki.foxtom.de](#), **-size -10k** kleiner 10 KiB.
- **-mtime** *Tage* – Letzte Änderung (Modify timestamp) liegt *Tage* zurück. *Tage* als ganze Zahl: **+n** = älter als *n* Tage, **-n** = jünger als *n* Tage, **n** = genau *n* Tage (im Bereich ±0.5d). Ähnlich **-atime** (Access), **-ctime** (Statusänderung). Z.B. **-mtime -7** in letzten 7 Tagen modifiziert[wiki.foxtom.de](#).
- **-newer** *Datei* – Datei wurde später geändert als *Datei* (Vergleich mtime). **-newermt** *Datum* (GNU find) erlaubt Vergleich mit einem Zeit-String, z.B. **-newermt "2025-05-01"**.
- **-perm** *mode* – Dateirechte entsprechen *mode*. Mode kann exakt (z.B. 644) oder mit / oder - angegeben werden für Masken. Beispiel: **-perm 644** findet genau 644. **-perm -644** findet Dateien, die mindestens diese Bits haben (also rw- r-- r-- oder mehr Rechte). **-perm /222** findet Dateien, bei denen irgendein Schreibbit gesetzt ist (Owner oder Group oder Other).
- **-maxdepth** *N* – Nur bis Tiefe *N* ins Verzeichnis hinab suchen (0= nur Startdir selbst). **-mindepth** *N* analog minimal tiefe. Z.B. **find . -maxdepth 1 -type f** nur Dateien im aktuellen Verzeichnis, nicht tiefer.
- **-mount** oder **-xdev** – Verlasse nicht das Dateisystem. D.h. keine Verzeichnisse auf anderen gemounteten FS durchsuchen (z.B. ignoriert **find / -xdev** dann andere Partitionen).
- **-prune** – Überspringt (schließt vom Durchsuchen aus) ein Verzeichnis. Bsp: **find . -path "./node\_modules" -prune -o -name "\*.js" -print** durchsucht alles außer den node\_modules Ordner.

#### Aktionen (wenn Suchkriterien wahr sind):

- **-print** – Gibt gefundene Pfade aus (Standard, wird angenommen, wenn keine andere Aktion).
- **-ls** – Führt ls -dils auf die gefundene Datei aus (anzeige von detaillierten Infos, inkl. Inode, Größe, Rechte etc.).
- **-delete** – Löscht gefundene Dateien *unmittelbar*. Achtung: **find -delete** entfernt auch Verzeichnisse, wenn leer geworden. Sollte meist mit Kriterien kombiniert

werden (z.B. -type f -delete nur Dateien). **Keine** Aktion mehr nach -delete wird ausgeführt (beendet Durchlauf in diesem Pfad).

- **-exec Befehl {} ;** – Führt einen externen Befehl für jede gefundene Datei aus. {} wird durch den Pfad ersetzt. Die Kommandozeile wird so oft ausgeführt wie Treffer. Beispiel: find . -name "\*.bak" -exec rm {} \; löscht alle .bak-Dateien (funktional wie -delete, aber universeller)file-5y1edouysrow4zj6phjcqm. Das \; markiert Ende des -exec Befehls.
- **-exec ... {} +** – Variante, die versucht, mehrere gefundene Dateien in *einen* Aufruf zu packen (fügt die Pfade an Stelle von {} möglichst gesammelt, wie xargs). Effizienter, falls das Programm mehrere auf einmal verarbeiten kann. Z.B. -exec chmod 644 {} + ändert die Rechte aller gefundenen Files, ggf. in batched Calls.
- **-execdir** – Wie -exec, aber führt das Kommando im jeweiligen Verzeichnis der Datei aus (Working Directory wechselt). Erhöht Sicherheit bei {} mit {} containing spaces.

### Beispiele:

- find /var/log -type f -name "\*.log" -size +1M -mtime +30 – findet Log-Dateien über 1MB, älter als 30 Tage unter /var/log.
- find . -maxdepth 1 -type d ! -name ":" -exec rm -r {} \; – löscht alle Unterverzeichnisse im aktuellen Ordner (maxdepth 1, alles was Type d ist und Name nicht ":"), dann rm -r).
- find /home/user -xtype l -delete – löscht gebrochene Symlinks (xtype l findet Links, deren Ziel nicht existiert).
- find . -perm /u+x -print – findet Dateien, bei denen das User-execute Bit gesetzt ist (also alle ausführbaren Dateien).
- find . -name ".\*tmp" -execdir gzip {} \; – packt alle \*.tmp im jeweiligen Verzeichnis in eine .tmp.gz (ausgeführt im Ort der Datei).

### locate (Datenbank-Dateisuche)

**Beschreibung:** Durchsucht eine vorgefertigte Index-Datenbank nach Dateinamen (Pfaden), die einem angegebenen Suchstring entsprechenfile-4fdk7rvx39azjretinarqb. locate ist viel schneller als find, da es nicht live das Dateisystem scannt, sondern auf einer regelmäßig aktualisierten Datenbank (typisch via updatedb in Cron) basiert. Nachteil: findet keine erst sehr kürzlich hinzugefügten Dateien (bis zum nächsten updatedb-Lauf) und standardmäßig nur solche, auf die der User zugreifen darf.

### Syntax:

locate [Optionen] <Suchmuster>

*Das Suchmuster ist im Grunde ein Substring oder einfache Glob. Standardmäßig sucht locate alle Datenbankeinträge, die <Suchmuster> als Teilstring enthalten.*

### Optionen:

| Option                      | Beschreibung                                                                                                                                                                                                                              |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-i, --ignore-case</b>    | Sucht case-insensitive (Groß/Kleinschreibung ignorieren)file-4fdk7rvx39azjretinarqb.                                                                                                                                                      |
| <b>-r, --regex</b>          | Interpretiert das angegebene Muster als regulären Ausdruck (POSIX ERE) statt als einfachen Textfile-4fdk7rvx39azjretinarqb.                                                                                                               |
| <b>-w, --wholename</b>      | (Standardverhalten) Sucht im gesamten Pfadnamen. (Bei einigen Implementierungen war locate nur substring-suche, wholename explizit wie default, hier nur Info.)                                                                           |
| <b>-b, --basename</b>       | Sucht nur im Dateinamen selbst, nicht im gesamten Pfad. (Also ignoriert Verzeichnisnamen im Match.)                                                                                                                                       |
| <b>-c, --count</b>          | Zählt nur die Anzahl der Fundstellen und gibt diese Zahl aus, anstatt alle Ergebnisse zu listen.                                                                                                                                          |
| <b>-l N, --limit=N</b>      | Gibt maximal N Ergebnisse aus (dann Abbruch).                                                                                                                                                                                             |
| <b>--existing</b>           | Prüft gefundene Pfade und filtert jene heraus, die aktuell nicht (mehr) existieren. (Die DB könnte veraltete Einträge haben; mit --existing werden nur noch valide Treffer angezeigt, obwohl es Zeit kostet, jeden Pfad zu verifizieren.) |
| <b>-0, --null</b>           | Trennt die Ausgabe mit Nullbytes anstatt neuer Zeile. Nützlich, um in xargs -0 weiterzuverarbeiten.                                                                                                                                       |
| <b>-e, --nofollow</b>       | Ignoriert Einträge, die Symlinks auf nicht-existente Dateien sind (köönnte Implementation-spezifisch).                                                                                                                                    |
| <b>-S</b>                   | Zeigt Statistik der Datenbank (Anzahl Einträge etc.) statt zu suchen.                                                                                                                                                                     |
| <b>-d DB, --database=DB</b> | Nutzt eine alternative Datenbank-Datei anstelle der Standard-Datenbanken (/var/lib/mlocate.db z.B.). Man kann mehrere via -d db1:db2:... angeben.                                                                                         |
| <b>--help, --version</b>    | Hilfe bzw. Version.                                                                                                                                                                                                                       |

*Beispiel:* locate -i '\.pdf' listet alle Pfade, die ".pdf" (unabhängig von Großschreibung) enthalten – das würde i.d.R. alle PDF-Dateien auf dem System listen, die zum Zeitpunkt

der letzten Datenbankaktualisierung existierten. Oft benutzt: locate <Programmname> um herauszufinden, wo ein Programm oder dessen Konfigdateien liegen.

### **updatedb (Update Database for locate)**

**Beschreibung:** Baut die Datei-Datenbank für locate neu auf. Es durchsucht das Dateisystem (oder definierte Pfade) und speichert alle Dateinamen/Pfade in einer komprimierten Datenbank (typischerweise `/var/lib/mlocate/mlocate.db`). updatedb wird meist regelmäßig (täglich) via Cron aus geführt, man kann es aber auch manuell starten, z.B. nach größeren Änderungen im FS, damit locate sofort aktuell ist.

#### **Syntax:**

`updatedb [Optionen]`

**Optionen:** (Häufig werden diese über `/etc/updatedb.conf` voreingestellt)

| <b>Option</b>                                       | <b>Beschreibung</b>                                                                                                                                                                                                                                                                            |
|-----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>--prune-bind-mounts</b><br><code>{yes no}</code> | Ob Bind-Mountpoints ausgelassen werden (Default ja – doppelte Scans vermeiden).                                                                                                                                                                                                                |
| <b>--prune-</b><br><b>paths='PFADLISTE'</b>         | Doppelpunkt-separierte Liste von Verzeichnissen, die <i>nicht</i> indexiert werden sollen file-4fdk7rvx39azjretinarqb. Standardmäßig z.B. <code>/tmp, /proc, /sys, /run, /dev, /var/cache</code> etc., um unnötiges zu sparen. Dies entspricht PRUNEPATHS in <code>/etc/updatedb.conf</code> . |
| <b>--prune-</b><br><b>filesystems='TypList'</b>     | Liste von Dateisystemtypen, die nicht durchsucht werden (z.B. nfs,squashfs etc.). Standard meist Netzwerk-FS, Pseudo-FS (proc, sysfs) etc. file-4fdk7rvx39azjretinarqb. Entspricht PRUNEFS.                                                                                                    |
| <b>-l N</b>                                         | <code>--localpaths=LIST</code> – falls man nur lokale Pfade indexieren will (veraltet, use prune).                                                                                                                                                                                             |
| <b>-U 'PFADLISTE'</b>                               | Nur diese durch Doppelpunkt getrennten Pfade als Wurzel für Index verwenden (Standard "/" – gesamtes FS). Kann man nutzen um z.B. nur <code>/home</code> zu indexieren.                                                                                                                        |
| <b>-o DBFILE</b>                                    | Output: speichere DB in angegebener Datei statt Standard. (So kann man separate DBs erzeugen.)                                                                                                                                                                                                 |
| <b>-f 'FS-Liste'</b>                                | Nur diese Dateisystemtypen berücksichtigen (Inverse von PRUNEFS).                                                                                                                                                                                                                              |
| <b>-v</b>                                           | Ausführliche Ausgabe während des Indizierens.                                                                                                                                                                                                                                                  |

*In der Regel* wird updatedb ohne Optionen (ggf. als root über sudo updatedb) ausgeführt und nutzt die System-Voreinstellungen. Danach steht die Datenbank locate zur Verfügung.

### **which (Welches Kommando - Pfadsuche)**

**Beschreibung:** Zeigt den Pfad zu einem ausführbaren Programm, das in der Umgebungsvariablen \$PATH gefunden wirdfile-4fdk7rvx39azjretinarqb. Damit kann man herausfinden, welches Programm aufgerufen würde, wenn man einen bestimmten Befehl tippt (nützlich bei mehreren Versionen in verschiedenen Verzeichnissen).

#### **Syntax:**

which [Optionen] Befehlsname...

#### **Optionen:**

| <b>Option</b>                           | <b>Beschreibung</b>                                                                                                                                                                                                                |
|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-a</b>                               | Zeigt <i>alle</i> Vorkommen des angegebenen Befehls im Suchpfad an (nicht nur das erste gefundene)file-4fdk7rvx39azjretinarqb. Z.B. which -a python listet evtl. /usr/bin/python und /usr/local/bin/python falls beide existieren. |
| <b>--skip-dot</b> , <b>--skip-tilde</b> | Ignoriert . bzw. ~ am Anfang des Pfades (also aktuelles Verzeichnis oder Home in PATH) in der Suche. (Sicherheitsaspekt – Standard bei which).                                                                                     |
| <b>--show-dot</b> , <b>--show-tilde</b> | Zeigt explizit, wenn gefundene Einträge in . oder ~ liegen, durch Präfix "./" oder "~/".                                                                                                                                           |
| <b>--tty-only</b>                       | Zeigt nur etwas an, wenn stdout ein Terminal ist. Unterdrückt z.B. Ausgabe, wenn in Pipe (verhindert sich einzumischen).                                                                                                           |
| <b>--help</b> , <b>--version</b>        | Hilfe bzw. Versionsinfo (gilt bei GNU which; einige shells haben eingebautes which mit anderem Verhalten).                                                                                                                         |

*Hinweis:* which ist oft eine Shell-Funktion/Alias und verhält sich u.U. leicht unterschiedlich. Ähnliche Tools: type (shell-builtin, zeigt auch Aliases, Keywords), command -v.

**Beispiel:** which gcc -> /usr/bin/gcc (Pfad zum C-Compiler). which -a ls könnte z.B. zeigen /usr/bin/ls und /bin/ls falls beides symlinks aufeinander oder verschieden, etc.

### **Textverarbeitung und Streams (Shell, Scripting)**

#### **awk (Muster/Aktions-Scanner)**

**Beschreibung:** AWK ist ein mächtiges Textverarbeitungswerkzeug und Skriptsprache, mit dem man Textdateien zeilenweise nach Mustern durchsuchen und pro Zeile Aktionen ausführen kannfile-4fdk7rvx39azjretinarqb. Typischerweise wird awk benutzt,

um Spalten auszugeben oder zu filtern, da es Eingabezeilen automatisch in Felder zerlegt. Der Name stammt von den Anfangsbuchstaben seiner Schöpfer (Aho, Weinberger, Kernighan).

### Syntax:

awk [Optionen] 'Muster { Aktion }' Datei...

Jede Zeile der Datei wird gelesen; wenn das *Muster* passt, wird die *Aktion* ausgeführt (z.B. etwas ausgeben). AWK hat eigene Variablen (\$1 = erstes Feld, \$0 = ganze Zeile) und eingebaute Rechen- und String-Features.

### Optionen:

| Option                   | Beschreibung                                                                                                                                                                     |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-F separator</b>      | Legt das Eingabefeld-Trennzeichen fest (Field Separator). Standard sind whitespace (Leerzeichen/Tabs)file-4fdk7rvx39azjretinarqb. Z.B. -F ":" für /etc/passwd mit Doppelpunkten. |
| <b>-v Var=Wert</b>       | Setzt eine AWK-Variable <i>Var</i> auf <i>Wert</i> vor dem Verarbeiten. Z.B. -v ORS="\n\n" um eine Leerzeile nach jeder Ausgabezeile einzufügen (ORS = Output Record Separator). |
| <b>-f Scriptdatei</b>    | Lies das AWK-Programm aus einer Datei anstatt aus der Befehlszeile.<br>Damit kann man größere AWK-Skripte in eigenem File verwalten.                                             |
| <b>-m, --sandbox</b>     | (gawk) AWK in sicheren Modus schalten (kein System(), keine Schreibrechte auf FS etc).                                                                                           |
| <b>--posix</b>           | Schaltet in POSIX-kompatiblen Modus (ggf. einige GNU-Erweiterungen aus).                                                                                                         |
| <b>-W version</b>        | Zeigt AWK-Version an (für gawk).                                                                                                                                                 |
| <b>--help, --version</b> | Hilfe / Version.                                                                                                                                                                 |

### AWK Sprachkonzepte (Kurz):

- Ein AWK-Programm besteht aus *Muster { Aktion }* Blöcken. Muster ist ein Regex oder Ausdruck, Aktion ist Code in AWK.
- Felder: Eingabezeile wird in Felder zerlegt, zugreifbar als \$1, \$2, ..., \$NF (NF=Anzahl Felder). \$0 ist die ganze Zeile.
- Beispiele:
  - awk '{ print \$1 }' file – gibt die erste Spalte jeder Zeile aus (mit Standard-Trenn whitespace).

- awk '/error/ {print NR ":" \$0}' log.txt – sucht Zeilen, die "error" enthalten (Regex als Muster), und druckt Zeilennummer und kompletten Inhalt.
- awk -F, '\$3 > 100 { sum += \$3 } END { print "Summe:", sum }' data.csv – mit Komma als Separator summiere Wert der 3. Spalte aller Zeilen >100 in Spalte3; am Ende (END-Block) gib Summe aus.
- AWK hat viele eingebaute Funktionen (Mathe, String), assoziative Arrays, Kontrollstrukturen (if, for, while).

## **sed (Stream Editor)**

**Beschreibung:** sed ist ein nicht-interaktiver Stream-Editor zum Bearbeiten von Texteingaben nach Mustern, insbesondere für Suchen&Ersetzen oder das Extrahieren von Zeilenfile-4fdk7rvx39azjretinarqb. sed liest zeilenweise Eingabe (Dateien oder STDIN), wendet die angegebenen *Editier-Befehle* darauf an und schreibt das Ergebnis auf STDOUT (oder optional in-place zurück in Dateien). Am verbreitetsten ist die *Substitution* mit dem s/alt/neu/ Kommando.

### **Syntax:**

sed [Optionen] 'Script' Datei...

Das *Script* besteht aus einem oder mehreren Editierbefehlen, die auf die Eingabe angewendet werden. Beliebte sed-Befehle: **s** (substitute), **p** (print), **d** (delete), **q** (quit), **{ }** für Gruppen, Adressierungsmodi (Zeilen-Nr oder Regex).

### **Optionen:**

| <b>Option</b>                    | <b>Beschreibung</b>                                                                                                                                                                                                                                                                                                   |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-n, --quiet/--silent</b>      | Suppress normal output. Normal druckt sed jede Eingabezeile (ggf. modifiziert) automatisch. Mit -n wird nichts automatisch gedruckt, nur explizite p (print) Befehle geben ausfile-4fdk7rvx39azjretinarqb. Typisch in Kombination mit p in Script, um nur bestimmte Zeilen auszugeben.                                |
| <b>-e Script</b>                 | Gibt das zu verwendende sed-Skript (Editierbefehle) direkt an. Kann mehrfach genutzt werden für mehrere Befehle in Reihenfolge. Wenn nur ein Skript-Argument ohne -e gegeben ist, kann -e weggelassen werden.                                                                                                         |
| <b>-f Datei</b>                  | Lies das sed-Skript aus einer Datei. Erlaubt komplexere Edits aus einem File anstatt Kommandozeile.                                                                                                                                                                                                                   |
| <b>-i[SUF], --in-place[=SUF]</b> | Bearbeitet Dateien <i>in place</i> , d.h. schreibt Änderungen direkt zurück in die Eingabedateien. Optional kann ein <i>Suffix</i> angegeben werden, um vor Änderungen ein Backup der Originaldatei anzulegen (z.B. -i.bak). Ohne Suffix wird die Datei direkt überschrieben <a href="http://ionos.de">ionos.de</a> . |

| Option                   | Beschreibung                                                                                                                                                                                                                          |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-r, -E</b>            | Schaltet die Verwendung erweiterter regulärer Ausdrücke ein (so dass in Patterns +, ?, {} etc. ohne Backslash verwendet werden können). Bei GNU sed ist -r (old syntax) und -E (wie bei BSD sed) identisch.                           |
| <b>-s</b>                | Treat files separately. Normalerweise bei mehreren Dateien behandelt sed den Input als einen langen Stream. Mit -s wird nach jeder Datei der interne Zustand (z.B. Zeilennummer, \$-Regex-Flag) zurückgesetzt.                        |
| <b>-z, --null-data</b>   | Input wird als Null-terminiert (Zeilen getrennt durch ASCII NUL statt newline) angenommen. Ermöglicht Verarbeitung von Binär oder multiline-chunks. Entspricht dann, dass . nicht auch \n matcht und \$ nur am NUL erkennt. (GNU sed) |
| <b>--follow-symlinks</b> | Wenn in-place auf Symlinks angewendet, ändert sed standardmäßig den Link selbst (also das Zielfile, aber eben durch den Link?). Mit dieser Option folgt sed dem Symlink und editiert die Ziel-Datei. (GNU sed)                        |
| <b>--sandbox</b>         | Verhindert, dass sed w oder s///w zum Schreiben verwendet oder ! Shell-Aufrufe macht. (GNU sed 4.4+)                                                                                                                                  |
| <b>--help, --version</b> | Hilfe / Version.                                                                                                                                                                                                                      |

### Häufige sed-Anwendungen:

- **Substitution (Ersetzen):**  
Syntax: s/REGEX/REPLACEMENT/[Flags]. Ersetzt das erste Vorkommen von REGEX in jeder Zeile durch REPLACEMENT. Flags: g = global (alle Vorkommen in Zeile ersetzen), i = case-insensitive, p = print (selbst wenn -n), w file = gefundene Zeilen nach file schreiben.  
Beispiel: sed -i 's/oldword/newword/g' file.txt – ersetzt *oldword* durch *newword* überall in file.txt, direkt in der Datei file-5y1edouysrow4zj6phjcqm.  
Besonderheit: Im Ersatz kann man & verwenden für den gesamten Match oder \1 ... für Capturing Groups aus dem Regex.
- **Zeilen löschen:**  
z.B. sed '/^#/ d' config – löscht alle Zeilen, die mit # beginnen (d = delete). Oder sed '5,10d' file löscht Zeilen 5 bis 10.
- **Nur bestimmte Zeilen ausgeben:**  
Mit -n und p. Z.B. sed -n '1,5p' file druckt nur Zeilen 1-5 (andere werden durch -n nicht ausgegeben). Oder sed -n '/ERROR/p' log druckt nur Zeilen mit "ERROR".

- **Mehrere Edits:**

z.B. sed -e 's/foo/bar/' -e '/BAZ/d' – erst foo->bar ersetzen, dann Zeilen mit "BAZ" löschen. (Die Reihenfolge ist wichtig).

- **In-Place Backup:**

sed -i.orig 's/^/#/' \*.conf – kommentiert alle Zeilen in \*.conf aus, speichert vorher Backup mit .orig-Suffix.

Sed kann viel mehr (Insert/Append mit i\ und a\, Transform mit y///, Shell-Befehle mit !, Holding Space etc.), was in Tiefe für Prüfung wohl nicht nötig ist. Wichtig ist Suchen&Ersetzen und Basisfilter.

### **tee (Output verdoppeln)**

**Beschreibung:** Liest die Standard-Eingabe und schreibt sie *gleichzeitig* auf die Standard-Ausgabe **und** in eine oder mehrere Dateienfile-4fdk7rvx39azjretinarqb. Es "verzweigt" den Datenstrom (daher der Name T-Stück). So kann man die Ausgabe eines Kommandos in eine Datei umleiten, aber trotzdem im Terminal sehen.

**Syntax:**

... | tee [Optionen] <Datei>...

**Optionen:**

| <b>Option</b>           | <b>Beschreibung</b>                                                                                                                                                                                               |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a, --append            | Hängt die Ausgabe an bestehende Dateien an, anstatt sie zu überschreibenfile-4fdk7rvx39azjretinarqb. Standard ohne -a überschreibt Ziel-Dateien neu.                                                              |
| -i, --ignore-interrupts | Ignoriert Interrupt-Signale (SIGINT). D.h. Ctrl+C wirkt nicht auf tee selbst (nur auf vorheriges Kommando?), damit tee garantiert die Daten schreibt.                                                             |
| --output-error MODE     | Verhaltensweise falls ein Schreibfehler auftritt: warn (Fehler melden, aber weiter versuchen andere outputs zu schreiben), exit (sofort beenden, evtl. ungeschriebene outputs gehen verloren). Standard ist warn. |

**Anwendung:**

- make 2>&1 | tee build.log – Zeigt alle Ausgaben von make im Terminal und speichert sie gleichzeitig in *build.log*.
- ls -l | tee -a listing.txt – Ausgabe von ls -l normal anzeigen und ans Ende von listing.txt anfügen.

- echo "strict mode on" | sudo tee /proc/sys/kernel/yama/ptrace\_scope – Durch sudo tee kann man Inhalte in Dateien schreiben, für die man Rechte braucht, während Eingabe evtl. aus einem un-privilegierten Echo kommt.

Tee ist also nützlich bei Logging und bei Verwendung mit sudo zum in Dateien schreiben, da sudo echo >> file wegen Redirection sonst nicht klappt.

## xargs (Execute Arguments)

**Beschreibung:** Liest Eingaben (Standardinput oder Datei) und führt daraus Kommandoaufrufe zusammengebaut aus. Typisch, um die Ausgabe eines Programms als Argumentliste für ein anderes Programm zu verwendenfile-4fdk7rvx39azjretinarqbfile-4fdk7rvx39azjretinarqb. xargs nimmt zeilen- oder spaltengetrennte Wörter und packt möglichst viele davon in einen Aufruf eines angegebenen Kommandos (Standard: echo). Damit überwindet man die Shell-Längenbeschränkung und kann sequentiell Befehle auf Listen anwenden.

### Syntax:

xargs [Optionen] [Befehl [Initial-Argumente]]

Wenn kein Befehl angegeben ist, benutzt xargs per Default echo (einfaches Ausgeben).

### Optionen:

| Option      | Beschreibung                                                                                                                                                                                                                                                                                                                                             |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -n MaxArgs  | Maximal <i>MaxArgs</i> Argumente pro aufgerufenem Kommando verwendenfile-4fdk7rvx39azjretinarqb. Z.B. -n 1 bedeutet, xargs nimmt jeweils 1 Argument und führt das Kommando damit aus (also für jede Eingabezeile einmal). Ohne Angabe bündelt xargs so viele Argumente wie möglich pro Aufruf, bis die Systemlimite (Getconf ARG_MAX) fast erreicht ist. |
| -L MaxLines | Nimmt maximal <i>MaxLines</i> Input-Zeilen pro Kommando-Aufruf. Z.B. -L 1 bewirkt, dass jede einzelne Zeile als eine xargs-Ausführung behandelt wird (selbst wenn Zeile mehrere Wörter enthält).                                                                                                                                                         |
| -d Delim    | Verwendet <i>Delim</i> als Trennzeichen anstelle von Whitespace oder newline. Kann auch mehrere Zeichen (String) sein (GNU). Beispiel: -d ':' um : als Separator zu nehmen.                                                                                                                                                                              |
| -0, --null  | Erwartet Input-Felder durch Nullbyte getrennt (wie von find -print0 oder grep -Z erzeugt). Dies ermöglicht sichere Übergabe von Dateinamen mit Leer-/Sonderzeichen <a href="http://stackoverflow.com">stackoverlow.com</a> .                                                                                                                             |

| <b>Option</b>                  | <b>Beschreibung</b>                                                                                                                                                                                                                                             |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-E EOFSTR</b>               | Definiert eine Zeichenfolge, die als <i>Ende der Eingabe</i> interpretiert wird.<br>Wenn xargs diese Zeichenfolge liest, bricht es die weitere Verarbeitung ab (auch wenn nicht EOF).                                                                           |
| <b>-e</b>                      | (veraltet, äquivalent zu -E ohne Parameter) – historically to specify EOF marker, oft nicht mehr benötigt.                                                                                                                                                      |
| <b>-I ReplaceStr</b>           | Nutzt <i>ReplaceStr</i> als Platzhalter in dem auszuführenden Kommando, der durch den gesamten Input (pro Aufruf) ersetzt wirdfile-                                                                                                                             |
| <b>-J ReplaceStr</b>           | 4fdk7rvx39azjretinarqb. Z.B. xargs -I {} mv {} /backup nimmt jeden Input (eine Datei) und substituiert anstelle von {} im mv-Aufruf. Bei -I wird implizit -L 1 angenommen (d.h. pro Zeile ein Aufruf).                                                          |
| <b>-t, --verbose</b>           | (BSD xargs) Ähnlich -I, aber ersetzt nur einmal in der Kommandozeile (kann mit -n kombinieren). In GNU xargs gibt es -J nicht.                                                                                                                                  |
| <b>-p, --interactive</b>       | Gibt das Kommando, das ausgeführt wird, vorher auf stderr aus (also "echo" what is being done). Zum Debuggen hilfreich.                                                                                                                                         |
| <b>-r, --no-run-if-empty</b>   | Führt das Kommando <b>nicht</b> aus, wenn keine Argumente aus Input entstehen. Standard bei GNU xargs ist, <i>kein Leerlaufaufruf</i> zu machen (sprich diese Option ist implizit an). In POSIX könnte Standard anderes sein.                                   |
| <b>-0, --null</b>              | (siehe oben) Null-terminierte Eingabe verwenden – wichtig in Kombination mit find -print0 um sicher Dateilisten zu pipen ohne Probleme mit Sonderzeichen <a href="http://stackoverflow.com">stackoverflow.com</a> .                                             |
| <b>--max-chars=K, -s K</b>     | Maximal K Zeichen pro Kommandozeilen-Aufruf verwenden (inklusive Kommando und env). Standard an System ARG_MAX orientiert (oft ~131072). Manchmal zu reduzieren bei seltsamen Umgebungen.                                                                       |
| <b>--max-procs=N, -P N</b>     | Bis zu N Prozesse <i>parallel</i> ausführen statt nacheinander. Z.B. -P 4 führt bis zu 4 Instanzen des Kommandos gleichzeitig aus, um Parallelisierung zu erreichen. N=0 bedeutet unbegrenzt viele parallel (Vorsicht!). Hilfreich z.B. in Multi-Core-Systemen. |
| <b>--process-slot-var=NAME</b> | Setzt eine Umgebungsvariable \$NAME in den aufgerufenen Prozessen, die eindeutige ID des xargs-Slots enthält (0..N-1) – bei -P parallel. Hilft um Ausgaben zuzuordnen.                                                                                          |

| Option                   | Beschreibung                                                                               |
|--------------------------|--------------------------------------------------------------------------------------------|
| <b>-a Datei</b>          | Liest Input aus <i>Datei</i> anstatt STDIN. So kann xargs auch ohne Pipe aus File speisen. |
| <b>--help, --version</b> | Hilfe/Version.                                                                             |

### Beispiele:

- find /tmp -type f -name '\*.txt' -print0 | xargs -0 rm -f – Löscht alle .txt-Dateien unter /tmp (sicher gegenüber Sonderzeichen, dank -print0 und -0).
- grep -l "ERROR" \*.log | xargs -r zip errors.zip – Sucht in allen .log nach "ERROR", die -l Option listet nur Dateinamen mit Treffer. Diese Liste wird an xargs übergeben, das dann zip errors.zip <files> ausführt. Falls kein Treffer, -r sorgt dafür, dass zip nicht ohne Argument aufgerufen wird.
- cat list.txt | xargs -l {} echo File: {} >> files.txt – Für jede Zeile aus list.txt, führe echo File: <line> aus und append an files.txt. (So fügt man "File: " vor jede Zeile).
- seq 1 1000 | xargs -n 100 > numbers.txt – Nimmt die Zahlen 1 bis 1000 (jede auf neuer Zeile) und packt jeweils 100 Zahlen in eine Zeile, schreibt nach numbers.txt.
- printf "%s\\0" \* | xargs -0 -n 1 -P 4 sha256sum – Berechnet parallel (4 at a time) die SHA256 Summen aller Dateien im aktuellen Verzeichnis. printf "%s\\0" \* generiert Null-separierte Liste aller Dateien, xargs nimmt je 1 (-n1) und startet bis zu 4 sha256sum gleichzeitig.

## Software- und Paketmanagement

### tar (Tape ARchiver)

**Beschreibung:** Archiviert mehrere Dateien zu einer einzigen Datei (tarball). Tar kann Dateien **bündeln** (zu .tar Archiven) und in Zusammenarbeit mit Kompressionsprogrammen (gzip, bzip2, xz etc.) zugleich komprimierenfile-4fdk7rvx39azretinarqb. Oft verwendet, um Verzeichnisse oder viele Dateien als ein Archiv zu speichern oder zu verteilen (.tar.gz, .tgz, .tar.bz2 etc.).

### Syntax:

tar [Optionen] -f <Archiv> [Dateien/Verzeichnisse...]

Tar hat **Modi** (Mutual exclusive Optionen): Erstellen, Extrahieren, Listen etc. Üblich ist die kombinierte Kurzoption wie -czf oder -xvf. Wichtig: -f (file) gibt an, welches Archivfile zu benutzen ist, und kommt als *ein* Argument, meist zuletzt bei Short-Options-Kombination.

### Haupt-Modi:

- **-c, --create** : Erstelle ein neues Archiv aus den  
**(Fortsetzung)**

## **tar (Tape Archiver)**

**Beschreibung:** Erstellt oder entpackt Archive aus mehreren Dateien. tar bündelt Dateien in **einem** Archiv (Containerdatei mit Endung `.tarfile-4fdk7rvx39azjretinarqb】` . Oft wird tar mit Kompressionsprogrammen kombiniert (z.B. gzip oder bzip2), um komprimierte Archive wie `.tar.gz` zu erstellefile-4fdk7rvx39azjretinarqb】 .

### **Syntax:**

`tar <Modus-Option> [Optionen] -f <Archivdatei> [Dateien/Verzeichnisse...]`

Tar benötigt **eine** der Hauptoptionen: -c (create), -x (extract) oder -t (list). Dazu wird **-f** angegeben, um den Namen der Archivdatei zu bestimmen. Häufig werden mehrere Kurzoptionen kombiniert (z.B. -czvf für "create, gzip, verbose, file").

### **Wichtige Optionen:**

- **-c – create:** Neues Archiv erstellefile-4fdk7rvx39azjretinarqb】 .
- **-x – extract:** Dateien aus Archiv entpacken.
- **-t – list:** Inhalt des Archivs auflisten (nicht extrahieren).
- **-f Datei** – Gibt den Archivnamen an (Pfad/Name der Tar-Datei). Diese Option **muss** direkt gefolgt von dem Dateinamen kommen (z.B. -f archive.tar).
- **-v – verbose:** Ausführliche Ausgabe (listet während der Verarbeitung die Dateinamen auf).
- **-z – Filter durch gzip:** Bei -c wird das Archiv mit gzip komprimiert (\*.tar.gz); bei -x wird erwartet, dass das Archiv gz-komprimiert isfile-4fdk7rvx39azjretinarqb】 . (Äquivalent: --gzip oder mit externem gzip pipen).
- **-j – Filter durch bzip2:** Erstellt/entpackt ein bzip2-komprimiertes Archiv (\*.tar.bz2). (Alternativ --bzip2.)
- **-J – Filter durch xz:** Für `.tar.xz`-Archive (--xz).
- **-A – concatenate:** Hängt ein oder mehrere Archive an ein bestehendes Archiv an.
- **-d – diff/compare:** Vergleicht Archiveinträge mit Dateien im Dateisystem (prüft, ob Unterschiede bestehen).
- **-C DIR – Change Directory:** Wechselt ins Verzeichnis `DIR`, bevor Dateien hinzugefügt oder extrahiert werden[freedesktop.org】](http://freedesktop.org) . Z.B. `tar -xvf archive.tar -C /tmp` entpackt nach `/tmp`.

- **--strip-components=N** – Beim Extrahieren lässt tar N führende Pfadkomponenten weg. Z.B. entpackt /usr/local/share/doc/file als file bei --strip-components=3. Praktisch, um Archiv ohne Top-Level-Ordner auszupacken.
- **--exclude=PATTERN** – Schließt Dateien, die auf PATTERN passen, vom Archivieren aus. Z.B. --exclude="\*.o".
- **--wildcards, --anchored** – steuert, wie Muster bei -t oder -x --wildcards -f archive "\*.txt" interpretiert werden (glob-Muster an/aus etc.).
- **--numeric-owner** – Beim Entpacken Besitzer/Gruppen als IDs übernehmen (statt Namen auf dem Zielsystem zuzuordnen).

### **Beispiele:**

- Archiv erstellen: tar -cvf archive.tar Verzeichnis/ erzeugt ein unkomprimiertes Archiv *archive.tar* mit allen Dateien aus *Verzeichnis/*.
- Mit Kompression: tar -czvf etc.tar.gz /etc packt ganz /etc als gzip-komprimiertes Archiv.
- Entpacken: tar -xvzf backup.tar.gz -C /restore entpackt *backup.tar.gz* nach */restore*.
- Auflisten: tar -tvf archive.tar zeigt die Dateiliste mit Größen, Timestamp, Rechten etc.

## **gzip (GNU Zip Compression)**

**Beschreibung:** Komprimiert oder dekomprimiert Dateien im **gzip-Format** (.gz). Es ersetzt beim Komprimieren standardmäßig die Originaldatei durch die komprimierte Version (und fügt .gz anfile-4fdk7rvx39azjretinarqb】). Gzip ist schneller, aber erzeugt etwas größere Dateien als bzip2/xz. Häufig mit tar (siehe oben) eingesetzt.

### **Syntax:**

gzip [Optionen] [Datei...]

gunzip [Optionen] [Datei...]

(gunzip ist äquivalent zu gzip -d.)

### **Optionen:**

- **-c, --stdout**: Schreibt das komprimierte Ergebnis auf die Standardausgabe, anstatt die Datei zu ersetzen. Original bleibt erhalten (kann in Pipeline genutzt werden[ionos.de](#)】).
- **-d, --decompress**: Dekomprimiert die .gz-Datei (auch via gunzip aufrufbar[ionos.de](#)】).

- **-k, --keep:** Behält Originaldatei beim Komprimieren/Entpacken (standardmäßig wird Original gelöscht).
- **-r, --recursive:** Komprimiert alle Dateien in angegebenen Verzeichnissen rekursiv.
- **-1 bis -9:** Einstellen der Kompressionsstufe (1 = schnellste/geringste Kompression, 9 = langsamste/beste). Standard ist 6.
- **-t, --test:** Prüft eine gz-Datei auf Integrität, ohne sie zu entpacken (exit-Status zeigt OK/NOK).
- **-l, --list:** Zeigt Informationen über eine komprimierte Datei: Originalgröße, komprimierte Größe, Kompressionsrate, Zeitstempel.
- **--stdout, -c** (nochmal): Wird oft mit -d kombiniert, um nach stdout zu entpacken (z.B. gzip -dc file.gz).
- **-N, --name:** Bewahrt ursprünglichen Dateinamen und Zeitstempel (in .gz-Header) oder stellt diese beim Entpacken wieder her (ist Standardverhalten bei GNU gzip).
- **--help, --version:** Hilfe bzw. Versionsinfo.

### **Beispiele:**

- gzip bigfile komprimiert *bigfile* zu *bigfile.gz* (und löscht *bigfile*).
- gzip -9 \*.txt erstellt maximal komprimierte \*.txt.gz Dateien.
- gunzip data.csv.gz entpackt *data.csv.gz* zurück zu *data.csv*.
- gzip -c file.txt > file.txt.gz komprimiert nach stdout und schreibt in separate Datei (Original bleibt).
- In Pipe: tar -cvf - /folder | gzip > folder.tar.gz – tar erzeugt Archiv auf stdout, gzip komprimiert es zu *folder.tar.gz*.

### **xz (LZMA Compressor)**

**Beschreibung:** Komprimiert/Dekomprimiert Dateien mit dem **xz/LZMA2-Algorithmus**. XZ erreicht hohe Kompressionsraten, aber benötigt mehr CPU und RAM. Dateiendung: *.xz*. Wird z.B. für Kernel-Quellen-Archive genutzt. Syntax und Optionen ähneln gzip.

### **Syntax:**

`xz [Optionen] [Datei...]`

`unxz [Optionen] [Datei...]`

(xz -d entspricht unxz.)

### **Wichtige Optionen:**

- **-z** (Default): Komprimieren (Standardmodus, kann entfallen).
- **-d, --decompress**: Entpacken.
- **-k, --keep**: Originaldatei behalten (nicht löschen).
- **-f, --force**: Vorhandene .xz-Dateien überschreiben, falls vorhanden; oder erzwingt Komprimieren auch bei scheinbar nicht sinnvoll (z.B. sehr kleine Datei).
- **-9** (auch -0 .. -9): Kompressionsgrad (0 = none, 6 = Standard, 9 = max). XZ erlaubt auch -e für extreme Mode (noch etwas bessere Kompression, deutlich langsamere Kodierung).
- **-T N, --threads=N**: Nutzung von N Threads für Kompression (für große Dateien auf Multi-Core). Standard 1. (Für Dekomprimierung wird threads automatisch benutzt).
- **-c, --stdout**: Ergebnis auf stdout schreiben (bzw. von stdin lesen und auf stdout entpacken).
- **-l, --list**: Listet das .xz-Archiv (Inhalt, ratio, etc. – bei Single-File Komprimierung liefert original und comp size).
- **-q, -v**: quiet bzw. verbose Mode.
- **-C, --check=TYPE**: Prüfsummenverfahren wählen (Standard CRC64). Kann z.B. auf CRC32, SHA256 gestellt werden.

### **Beispiele:**

- `xz hugefile` erzeugt `hugefile.xz` (Original wird gelöscht).
- `xz -dk archive.tar.xz` entpackt `archive.tar.xz` nach `archive.tar` und behält auch die .xz.
- `tar -c /usr/bin | xz -9e > bin.tar.xz` erstellt maximal-komprimiertes tar-Archiv.
- `unxz data.csv.xz` oder `xz -d data.csv.xz` entpackt Datei.
- `xzcat file.txt.xz = xz -dc file.txt.xz` entpackt nach stdout (ähnlich zcat).

### **bzip2 (Burrows–Wheeler Compressor)**

**Beschreibung:** Komprimiert mit dem älteren bzip2-Algorithmus (.bz2). Langsamer als gzip, aber bessere Kompression; heute oft von xz abgelöst. Bedienung ähnlich gzip.

**Syntax:** bzip2 [Optionen] [Datei...] – bunzip2 zum Entpacken.

**Optionen (Auswahl):** -k (keep file), -d (decompress), -1 .. -9 (Stufe, Standard 9 bei bzip2), -c (stdout), -t (Test), -v (verbose stats).

### **zip (Archiv mit Kompression, ZIP-Format)**

**Beschreibung:** Erstellt ZIP-Archive (mit Kompression) oder fügt Dateien hinzu. Anders als tar erwartet zip direkt die zu archiviernden Dateien und komprimiert sie einzeln ins Archiv. ZIP-Archive enthalten Verzeichnisstruktur + komprimierte Dateien (beliebtes Format unter Windows). unzip entpackt sie.

#### Syntax:

zip [Optionen] <Archiv.zip> <Dateien...>

#### Wichtige Optionen:

- **-r** – Rekursiv Verzeichnisse einschließen. Z.B. zip -r archiv.zip Ordner/.
- **-9** (bzw. -1 ... -9) – Kompressionsgrad (9 = beste, Standard meist 6).
- **-q** – quiet, **-v** – verbose (gibt Kompressionsraten aus).
- **-m** – move: Dateien nach dem Hinzufügen ins Archiv löschen (verschieben).
- **-j** – junk paths: Pfadnamen ignorieren, legt alle Dateien „flach“ im Archiv ab (ohne Verzeichnisstruktur).
- **-x <Pattern>** – bestimmte Dateien ausschließen (z.B. -x "\*.o").
- **-F** – Versucht ein beschädigtes Archiv zu reparieren (freshen).
- **-u** – update: nur neuere Dateien (bzgl. Zeitstempel) ins Archiv aktualisieren.
- **-g** – grow: hängt Dateien an ein bestehendes ZIP an (statt neu zu erstellen).
- **-e** – erstellt ein verschlüsseltes Archiv (fordert Passwort-Eingabe, traditionelles ZipCrypto; -P <pw> unsicher ohne Prompt).

#### unzip Wichtige Optionen:

unzip <file.zip> entpackt im aktuellen Verzeichnis. -d <DIR> entpackt ins DIR. -l listet Inhalte. -t testet. -x schließt beim Entpacken bestimmte aus. -n überschreibt nichts, -o überschreibt ohne Nachfrage.

### Paketmanagement-Befehle (Distributionen)

#### rpm (RPM Package Manager)

**Beschreibung:** Das Low-Level-Werkzeug zum Installieren, Entfernen und Abfragen von RPM-Paketen (Red Hat, CentOS, Fedora, SUSE etc.). rpm arbeitet auf .rpm-Dateien und der lokalen Paketdatenbank. Häufig nutzt man für höhere Ebene yum oder dnf, aber rpm bleibt wichtig für direkte Paketoperationen.

#### Syntax (Basis):

rpm [Optionen] <Paketdatei> / rpm [Optionen] <Paketname>

#### Operationen (Hauptoptionen, nicht kombinierbar):

- **-i – install** ein neues RPM-Paket . Beispiel: rpm -i nginx.rpm. Erfordert Root-Rechte.
- **-U – upgrade** (installiert neu oder aktualisiert älteres Paket). Wird häufig anstelle -i benutzt, da es sowohl installiert als auch updatet.
- **-F – freshen:** wie -U, aber nur wenn schon ältere Version installiert ist (sonst kein Install).
- **-e – erase** (deinstalliert) ein installiertes Paket nach Paketname . Beispiel: rpm -e nginx.
- **-q – query:** Abfrage-Modus (ohne Änderungen) . Kann mit Zusätzen Info liefern. Beispiel: rpm -q nginx prüft, ob Paket installiert ist und welche Version.

### **Wichtige Optionen/Zusätze:**

Install/Upgrade Optionen:

- **-h** – zeigt Hash-Fortschrittsbalken beim Installieren ("#" -Zeichen).
- **-v** – verbose, mehr Ausgabe. Oft kombiniert -Uvh oder -ivh.
- **--force** – erzwingt Installation, selbst wenn z.B. ein Paket bereits installiert ist oder Dateien konfigurieren. (Vorsicht!).
- **--nodeps** – ignoriert Abhängigkeits-Prüfung (installiert trotzdem).
- **--replacefiles** – überschreibt Dateien, die zu anderen Paketen gehören könnten (Konflikte ignorieren).
- **--noshells** – führt keine Pre/Post-Install-Skripte im RPM aus (für spezielle Fälle).

Query Optionen (mit -q):

- **-a – query all** (listet alle installierten Pakete).
- **-i – (mit -q)** zeigt detaillierte Paketinformationen (Name, Version, Summary, Größe, Install-Datum etc. [linux-praxis.de](http://linux-praxis.de)) . Z.B. rpm -qi bash.
- **-l – listet alle vom Paket installierten Dateien.** ` rpm -ql <paket>file-5y1edouysrow4zj6phjcqm` .
- **-p – query ein Paketfile** (nicht installiert). Z.B. rpm -qp -l paket.rpm zeigt, welche Dateien im RPM drin sind.
- **-f Dateipfad** – findet heraus, zu welchem installierten Paket eine bestimmte Datei gehört. Z.B. rpm -qf /usr/bin/ls -> coreutils-8.xx.
- **-c – listet nur Konfigurationsdateien des Pakets** (mit -ql kombinierbar: -qlc).
- **--changelog** – zeigt das Changelog des Pakets (sofern im RPM hinterlegt).

- **--provides, --requires** – zeigen welche Provides/Requires (Abhängigkeiten) ein Paket liefert bzw. benötigt. Z.B. rpm -q --requires httpd.
- **-s** – zeigt den Status der Dateien (ob sie verändert wurden seit Installation, (vom user modifiziert = **M**, missing etc.)). rpm -qs <paket>.

Erase (Uninstall) Optionen:

- **--nodeps** beim -e: ignoriert Abhängigkeitsprobleme (entfernt auch wenn andere Pakete darauf angewiesen sind - kann zu kaputten Programmen führen).

### **Beispiele:**

- rpm -Uvh httpd-2.4.rpm – installiert Apache httpd, zeigt Hashtags und Namen.
- rpm -qa | grep kernel – listet alle installierten Pakete, filtert "kernel".
- rpm -qpi package.rpm – zeigt Infos über nicht installiertes RPM.
- rpm -e httpd – entfernt Apache-Paket.

## **yum (Yellowdog Updater Modified)**

**Beschreibung:** Ein High-Level-Paketmanager für RPM-basierte Distr. (v.a. CentOS/RHEL 7 und älter). yum löst Abhängigkeiten automatisch auf und kann aus Repositories Pakete installieren, entfernen, aktualisieren . (In neueren RHEL8+ abgelöst durch dnf, aber Syntax ähnlich.)

### **Syntax:**

yum [Optionen] <Sub-Befehl> [Paketnamen/...]

### **Haupt-Subkommandos:**

- **install Paket** – Installiert ein Paket (aus den konfigurierten Repos oder einer RPM-Datei). Z.B. yum install httpd. Resolved dependencies automatisch und fragt um Bestätigung.
- **update [Paket]** – Aktualisiert alle installierten Pakete (wenn kein Name angegeben) oder ein bestimmtes. (yum update führt System-Update durch) .
- **remove Paket, erase** – Entfernt angegebene Pakete mit Abhängigkeiten, die dadurch verwiesen.
- **search Stichwort** – Durchsucht die Paketnamen, Beschreibungen etc. nach dem Stichwort (um herauszufinden, welches Paket etwas bietet).
- **info Paket** – Zeigt Infos zu Paket(en) (Name, Version, Summary, Größe, Repo, Abh.) ähnlich rpm -qi aber aus Repos wenn nicht installiert.

- **list [installed|available] <Pattern>** – Listet Pakete (installierte oder verfügbare). Ohne Parameter alle Pakete, optional mit Pattern filtern. Z.B. yum list available 'php\*'.
- **provides Datei/Capability** – Findet heraus, welches Paket eine bestimmte Datei oder "Provide" bereitstellt (ähnlich rpm -qf/--whatprovides). Z.B. yum provides /usr/bin/zip.
- **clean <was>** – Bereinigt lokale Caches: z.B. yum clean all (löscht Metadaten- und Paketcache), oder yum clean metadata etc.
- **check-update** – Zeigt verfügbare Updates, ohne sie zu installieren (Exitcode 100 wenn Updates vorhanden, 0 wenn keine).
- **repolist** – Listet eingerichtete Repositories und ob aktiviert.
- **groupinstall / groupremove** – Installiert/entfernt ganze Paketgruppen (z.B. yum groupinstall "Development Tools").

#### Optionen (Auswahl):

- **-y** – Automatisch "yes" beantworten (nicht nach Bestätigung fragen). Wichtig für Skript】 .
- **-q** – Weniger Ausgabe (quiet).
- **-C** – Nur aus dem Cache arbeiten, keine neuen Metadaten laden (offline).
- **--enablerepo=X, --disablerepo=Y** – Repositories gezielt ein-/ausschalten für diesen Befehl (Overrides config).
- **--nogpgcheck** – Deaktiviert GPG-Signaturprüfung der Pakete (unsicher, nur falls Repo-Key nicht installiert etc.).
- **--downloadonly** – Lädt die Pakete herunter, installiert aber nicht.
- **--security** – (bei update/info/list) berücksichtigt nur sicherheitsrelevante Updates (wenn Repo das unterstützt).

#### Beispiele:

- yum install nginx php php-mysql – installiert mehrere Pakete in einem Rutsch.
- yum update – bringt System auf neuesten Stand.
- yum search "pdf viewer" – sucht nach Paketen mit "pdf viewer" im Namen/Beschreibung.
- yum remove httpd – deinstalliert Apache (und abhängige Module, wenn keine anderen es benötigen).
- yum provides "\*bin/rar" – findet Paket, das rar-Binary liefert.

(Hinweis: Auf neueren Systemen ersetzt dnf den Befehl yum. dnf Syntax und Subcommands sind weitgehend gleich, aber z.B. dnf upgrade statt yum update.)

## apt (Advanced Package Tool – Debian/Ubuntu)

**Beschreibung:** Höher-level Frontend für Debian-Paketverwaltung (dpkg). apt kombiniert Funktionen von apt-get und apt-cache in einem Kommando. Es verwaltet .deb-Pakete: Installation, Updates aus Repositories, Paket-Suche usw】. (Ubuntu 16.04+ und Debian Stretch+ empfehlen apt für interaktive Nutzung, während apt-get skriptfreundlich bleibt.)

### Syntax:

apt [Optionen] <Sub-Befehl> [Pakete...]

### Wichtige Sub-Befehle:

- **update** – Aktualisiert die Paketquellen-Datenbank (lädt neue Paketlisten von den konfigurierten Repos). Immer vor Install/Upgrade ausführen.
- **upgrade** – Installiert verfügbare Updates für alle **schon installierten** Pakete (führt keine Entfernung durch). Hält Pakete zurück, wenn Abhängigkeitswechsel nötig wären. (vgl. *apt full-upgrade*).
- **full-upgrade** – Führt vollständige Distribution-Upgrades durch, d.h. aktualisiert alle Pakete und darf auch Pakete entfernen oder neue Abhängigkeiten hinzufügen, um ein konsistentes System herzustellen (entspricht apt-get dist-upgrade).
- **install <pkg>...** – Installiert angegebene Pakete (aus Repos oder .deb Dateien via Pfad). Löst Abhängigkeiten auf und fragt bei Bedarf. Akzeptiert Paket=Version oder Paket:arch.
- **remove <pkg>...** – Entfernt Pakete, lässt jedoch veränderte Konfigurationsdateien bestehen (als *.config* in /etc).
- **purge <pkg>...** – Entfernt Pakete **inklusive** Konfigurationsdateien (bereinigt vollständig).
- **search <regex>** – Durchsucht Paketnamen und Beschreibungen nach einem Begriff (regex möglich). Beispiel: apt search editor pdf.
- **show <pkg>** – Zeigt detaillierte Informationen zu einem Paket (Version, Abhängigkeiten, Beschreibung, Maintainer etc.), egal ob installiert oder nur verfügbar】.
- **list** – Listet Pakete. Mit --installed, --upgradeable oder --all-versions. (Ähnlich dpkg -l aber filterbar). Z.B. apt list --installed | grep nginx.

- **autoremove** – Entfernt automatisch Pakete, die als Abhängigkeiten installiert wurden und nun nicht mehr benötigt werden (Waisen).
- **depends <pkg>, rdepends <pkg>** – Zeigt Abhängigkeiten bzw. Reverse-Abhängigkeiten eines Pakets (welche Pakete benötigen es).
- **edit-sources** – Öffnet die sources.list im Editor (um Repos zu bearbeiten).

#### **Optionen:**

- **-y, --yes** – Automatisch alle Fragen mit Yes beantworten (nicht interaktiv bestätigen). Praktisch für Skripte.
- **-q** – Weniger Ausgabe (bis zu -qq für gar keine).
- **--no-install-recommends** – Installiert nicht automatisch als "empfohlen" markierte Pakete. Standard apt installiert Recommends mit.
- **--allow-remove-essential, --allow-change-held-packages** – Erlaubt riskante Änderungen (Standard verhindert z.B. Entfernen essentieller Pakete).
- **-s, --simulate** – Simulation: Zeigt an, was würde passieren, aber führt keine Änderung aus (auch Exitcode 0 immer). Gut zum Test.
- **-V, --verbose-versions** – Zeigt bei Auflistungen detailliertere Versionsinfos.
- **--help** – Hilfe-Seite, **--version** – Version von apt.

#### **Beispiele:**

- apt update && apt upgrade – Aktualisiert Paketlisten und installiert alle verfügbaren Updates (ohne Dist-Upgrade).
- apt install build-essential – Installiert eine Paketgruppe (hier C/C++ Compiler Toolchain).
- apt remove --purge openssh-server – Entfernt SSH-Server komplett mit Config.
- apt search docker – Sucht alle Pakete mit "docker" im Namen/Beschreibung.
- apt full-upgrade – Aktualisiert System auch über Distribution-Upgrades hinweg (entspricht auf Ubuntu einem Release-Upgrade, sofern Repos angepasst).

(Hinweis: *apt-get* und *apt-cache* sind Vorgänger: z.B. *apt-get install*, *apt-get remove*, *apt-get dist-upgrade*, *apt-cache search*, *apt-cache show*. *apt* vereinfacht den Alltag, aber Skripte nutzen weiterhin oft *apt-get -y etc. dpkg* siehe unten ist noch eine Stufe tiefer für einzelne .deb ohne Abhängigkeitsauflösung.)

#### **dpkg (Debian Package Manager - Low-Level)**

**Beschreibung:** dpkg ist das unterliegende Werkzeug auf Debian/Ubuntu, das .deb-Pakete entpackt, installiert, konfiguriert und entfernt. Es löst keine Abhängigkeiten –

dafür sind apt/apt-get zuständig. Man verwendet dpkg meist für manuelle Offline-Installation von .deb-Dateien.

### Syntax:

```
dpkg -i <Paket.deb> # Installieren
dpkg -r <Paketname> # Entfernen
dpkg -P <Paketname> # Entfernen + Purge
dpkg -L <Paketname> # Liste Dateien des installierten Pakets
dpkg -s <Paketname> # Statusinfo (ähnlich apt show)
```

### Wichtige Optionen/Parameter:

- **-i, --install Datei.deb:** Installiert das angegebene Deb-Paket】 . Gibt Fehler aus, falls Abhängigkeiten fehlen. (Diese müssen vorher mit apt installiert werden).
- **-r, --remove Paketname:** Entfernt ein installiertes Paket (Paketname ohne ".deb", wie in dpkg -l gelistet】 . Konfigdateien unter /etc bleiben bestehen.
- **-P, --purge Paketname:** Entfernt Paket inkl. seiner Konfigurationsdateie】 .
- **-S, --search Dateiname:** Sucht, welches installierte Paket eine bestimmte Datei besitzt (ähnlich rpm -qf). Z.B. dpkg -S /usr/bin/foo.
- **-L, --listfiles Paketname:** Listet alle Dateien, die das installierte Paket auf dem System installiert hatfile-5y1edouysrow4zj6phjcqm】 .
- **-l, --list Muster:** Listet Pakete, optional gefiltert nach Muster. dpkg -l ohne Filter zeigt eine Tabelle aller Pakete und deren Status (ii = installiert, rc = removed config-files left, etc.).
- **-s, --status Paketname:** Zeigt den Status und Informationen eines installierten Pakets (Version, Beschreibung, Maintainer, Abhängigkeiten...).
- **--configure Paketname:** Führt die Konfigurationsschritte (dpkg-configure) für ein installiertes aber noch nicht konfiguriertes Paket aus. (Wird z.B. nach entpacken aufgerufen).
- **-B, --auto-deconfigure:** Beim Entfernen eines Pakets, das Abhängigkeiten bereitstellt, markiert abhängige Pakete als "deconfigured".
- **\*\*--force-\*\*X:** Viele Force-Flags, um dpkg anzuweisen, bestimmte Fehler zu ignorieren (z.B. --force-dependends ignoriert fehlende Abhängigkeiten, --force-confold/--force-confnew bei Config-Datei-Konflikten etc.). Vorsicht: Kann System inkonsistent machen.

- **--print-architecture**: Zeigt die Architektur, für die dpkg Pakete installiert (z.B. amd64).
- **--add-architecture <arch>**: Fügt Multiarch-Unterstützung für andere Arch hinzu (z.B. i386 auf einem amd64 System).

### **Beispiele:**

- dpkg -i google-chrome.deb – Installiert Chrome .deb-Paket (scheitert, wenn Abhängigkeiten fehlen – in dem Fall via apt --fix-broken install auflösbar).
- dpkg -r postfix – Entfernt postfix, belässt aber ggf. geänderte /etc/postfix Dateien.
- dpkg -P postfix – Entfernt postfix vollständig inklusive Config.
- dpkg -L bash – Listet alle vom *bash*-Paket installierten Dateien.
- dpkg -S /bin/ls – Welches Paket stellt /bin/ls bereit? (Antwort: coreutils).

## **wget (Web GET Utility)**

**Beschreibung:** Ein Kommandozeilen-Downloader für Dateien über HTTP, HTTPS und FTP. wget kann in Skripten genutzt werden, um Dateien aus dem Web herunterzuladen, unterstützt Fortsetzen abgebrochener Downloads, rekursives Herunterladen von Websites etc. Es funktioniert nicht-interaktiv (kein Browser, nur Download).

### **Syntax:**

wget [Optionen] <URL>

### **Wichtige Optionen:**

- **-O Datei** – Speichert die heruntergeladene URL in die angegebene *Datei* (statt den Namen aus URL zu verwenden). Z.B. wget -O ubuntu.iso http://.../ubuntu.iso.
- **-c, --continue** – Setzt einen abgebrochenen Download fort (sofern Server Range Requests unterstützt). Prüft, wieviel schon vorhanden ist und lädt den Rest.
- **-q** – Quiet-Modus (keine Ausgabe). **-nv** (not verbose) eine dazwischenliegende Stufe. **-v** für ausführlicher (Standard).
- **-r, --recursive** – Rekursiver Download einer Website. Erfordert oft zusammen mit -np (no parent), -L (stay on same domain), -k (Links anpassen für Offline). Z.B. wget -r -np -k http://example.com/docs/.
- **-np, --no-parent** – Geht beim rekursiven Download nicht über das Startverzeichnis hinaus.
- **-l <Tiefe>** – Legt die Rekursionstiefe fest (Default 5, -l0 = unendlich).

- **-p, --page-requisites** – Lädt alles mit, was nötig ist, um HTML-Seite offline darzustellen (Bilder, CSS...).
- **-N, --timestamping** – Lädt eine Datei nur herunter, wenn sie neuer ist als die lokale (verwendet Last-Modified). Gut zum Aktualisieren.
- **--limit-rate=<Rate>** – Begrenzt Downloadrate, z.B. --limit-rate=200k (200 KB/s). Hilfreich, um Bandbreite zu drosseln.
- **-U, --user-agent=<Agent>** – Setzt einen bestimmten User-Agent-String. Standard ist "Wget/Version". Kann geändert werden, um z.B. nicht blockiert zu werden.
- **--no-check-certificate** – Ignoriert SSL-Zertifikatfehler (z.B. selbstsigniert).
- **--user=<user> --password=<pw>** – Falls erforderlich, für HTTP-Auth oder FTP-Login. (Achtung: Sichtbar in Prozessliste).
- **-i <Datei>** – Liest eine Liste von URLs aus <Datei> (jede Zeile eine URL) und lädt alle herunter.
- **-b, --background** – Startet im Hintergrund (nur auf Unix, leitet Ausgabe nach wget-log).
- **-P <Verz>** – Speichert Dateien in angegebenem Verzeichnis (statt aktuelles).

### **Beispiele:**

- wget http://example.com/file.zip – Lädt *file.zip* in aktuelles Verzeichnis herunter.
- wget -c http://largefile – setzt Download fort, falls vorhanden.
- wget -r -np -N http://example.com/photos/ – Lädt Verzeichnis "photos" vollständig herunter, aktualisiert nur neue Dateien beim Wiederaufruf.
- wget -O - https://api.example.com/data | jq . – Lädt Daten von API und pipet direkt in jq zur Verarbeitung (hier -O - bedeutet Ausgabe auf stdout statt Datei).

### **curl (Client URL)**

**Beschreibung:** curl ist ein umfangreiches Kommandozeilen-Tool für Datenübertragungen mit URL-basierten Protokollen (HTTP, HTTPS, FTP, etc.). Im Gegensatz zu wget liegt Fokus auf Flexibilität bei HTTP-Requests (Header setzen, verschiedene HTTP-Methoden, Formulare absenden, APIs ansprechen etc.). Es eignet sich für Web-APIs (GET/POST) und kann ebenfalls Dateien herunterladen oder hochladen.

### **Syntax:**

curl [Optionen] <URL>

### **Wichtige Optionen:**

- **-o Datei** – Speichert die Ausgabe in *Datei*. (Kleinbuchstabe o, analog wget -O).
- **-O** – Speichert unter dem **Originalnamen** aus der URL (Großes O, analog wget ohne -O). Achtung: curl -O http://server/dir/file.txt legt *file.txt* an.
- **-L** – *Location folgen*: folgt HTTP-Redirects (Status 3xx) automatisch [freedesktop.org](#) ]. (wget macht das default, curl nicht ohne -L).
- **-C --** Setzt abgebrochenen Download fort (wenn Server unterstützt; '-' heißt automatischen Offset ermitteln).
- **-#** – Fortschrittsleiste im CLI (oder --progress-bar). Standard curl zeigt laufende Stats Zeile, -# macht wie wget progressive Bar.
- **-s** – Silent (kein Progress oder Fehlerausgaben). **-S** kombiniert mit -s zeigt Fehler dennoch an.
- **-v / -vv** – Verbose Mode (zeigt detaillierte Request/Response, Header etc. -vv noch mehr, inkl. Low-level). Gut zum Debuggen.
- **-I** – Führt eine HEAD-Anfrage aus (nur Header vom Server holen, keine Body-Daten). Nützlich um Meta-Informationen (Content-Length, Last-Modified) abzufragen.
- **-X METHOD** – Explizite HTTP-Methode festlegen (z.B. PUT, DELETE). Standard bei Datenvorhandensein oder -d ist POST, sonst GET.
- **-G** – Erzwingt, dass bei Verwendung von -d (Daten) diese als Query-Parameter an URL angehängt werden (statt POST Body).
- **-d Daten** – Sendet die angegebenen *Daten* im HTTP-Request-Body (Standard via POST). Z.B. curl -d "user=test&pass=123" http://.../login. Implizit -X POST wenn nicht -G.
  - **--data-urlencode Daten**: URL-encodet die Daten vor Versand (z.B. wenn Daten Sonderzeichen enthalten).
  - **--data-raw**: ähnlich -d, aber ohne spezielle Form-Codierung (sendet genau so).
- **-H "Header: Wert"** – Fügt einen HTTP-Header hinzu oder überschreibt ihn. Z.B. -H "Authorization: Bearer <token>". Mehrfach nutzbar für mehrere Header.
- **-u Benutzer:Pass** – Nutzt HTTP Basic Auth mit gegebenen Credentials (wird in Header "Authorization: Basic ...Base64" codiert [freedesktop.org](#) ]. Vorsicht in Shell-History. Kann auch nur -u user (dann fragt curl interaktiv Passwort ab). Funktioniert auch für FTP.
- **-F Feld@Datei** – HTTP Datei-Upload via *multipart/form-data*. Z.B. curl -F "file=@bild.png" -F "desc=Urlaubsfoto" http://.../upload.

- **-K Konfigdatei** – Liest curl-Parameter aus Datei (kann nützlich sein für komplexe Setups).
- **-b CookieDatei** – Sendet Cookies aus angegebener Datei (. Netscape cookie format oder einfach "name=value" Paare). Kann auch -b "name=value" um Cookies zu setzen.
- **-c CookieAusgabe** – Speichert Cookies, die vom Server kommen, in angegebener Datei (um sie später wieder zu verwenden).
- **--limit-rate <Speed>** – Begrenzt Übertragungsrate (z.B. --limit-rate 100k).
- **--retry N** – Anzahl der automatischen Wiederholungsversuche bei Fehler (z.B. Timeout) oder --retry-all-errors.
- **--cacert Datei / --insecure:** Spezielles SSL: anderes CA-Zertifikat nutzen bzw. bei --insecure SSL-Zertifikat überprüfen ignorieren (wie wget --no-check-certificate).

### **Beispiele:**

- curl -L -o latest.zip https://github.com/user/proj/releases/latest – Lade neueste Release, folge Redirects, speichere als latest.zip.
- curl -u admin:secret -X DELETE http://server/api/item/123 – Sende DELETE-Request mit Basic Auth.
- curl -H "Accept: application/json" 'https://api.example.com/data?id=5' – Hole JSON-Daten von API mit angepasstem Accept-Header.
- curl -d '{"user":"john"}' -H "Content-Type: application/json" -X POST https://api.example.com/users – Sende JSON-Body per POST (hier ein neuer User).
- curl -F 'file=@report.pdf' -F 'desc=Monthly Report' https://upload.example.com – HTTP File-Upload (Multipart).
- curl -I https://example.com – Nur Header einer HTTP-Response anzeigen (z.B. um Last-Modified zu prüfen).

## **Speicher- und Dateisystem-Verwaltung**

### **lsblk (List Block Devices)**

**Beschreibung:** Zeigt eine Übersicht aller erkannten Blockgeräte (Festplatten, SSDs, Partitionen, LVM-Volumes etc.) in Form einer Baumstruktur. lsblk listet wichtige Attribute wie Gerätename, Größe, Typ (disk/part), Mountpoint usw. Es hilft beim Identifizieren von Device-Namen und Partitionierungsstruktur.

### **Syntax:**

lsblk [Optionen] [Gerät...]

## Optionen:

- **-a, --all:** Zeigt auch leere bzw. unverwendete Geräte an (z.B. Geräte ohne Speicher, Loop ohne Backing). Standard lässt einige unwichtige aus.
- **-d, --nodeps:** Zeigt nur "Top-Level" Geräte (Disks) ohne Partitionen/Slaves.
- **-e <LIST>, --exclude <LIST>:** Schließt bestimmte Gerätetypen durch Major-Nummern oder Gerätekategorien aus (z.B. -e 7,11 um Loop und DVD zu ignorieren).
- **-f, --fs:** Zeigt zusätzlich Dateisystem-Informationen (Spalten FSTYPE, LABEL, UUID) . Ähnlich blkid, aber in lsblk integriert.
- **-p, --paths:** Zeigt vollständige /dev/Pfadnamen an statt nur Gerätenamen (z.B. "/dev/sda" statt "sda").
- **-l, --list:** Ausgabe im Listenformat (eine Eintrag pro Zeile, ohne Baum-Struktur).
- **-J, --json:** Ausgabe als JSON-Struktur (maschinenlesbar).
- **-O, --output-all:** Zeigt alle verfügbaren Spalten an. Standardanzeige ist begrenzt. Mit -O sieht man z.B. auch MAJ:MIN, RM (removable), RO (read-only), MODEL, SERIAL etc.
- **-n, --noheadings:** Unterdrückt Überschrift.
- **-b, --bytes:** Größe in Bytes anzeigen (statt lesbar in KiB/MiB etc.).
- **-r, --raw:** Rohformat ohne Ausrichtung/Padding (für Skripting).

## Beispielausgabe:

```
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda 8:0 0 100G 0 disk
 ↘-sda1 8:1 0 512M 0 part /boot
 ↘-sda2 8:2 0 99.5G 0 part
 ↘-vg-root 253:0 0 90G 0 lvm /
 ↘-vg-swap 253:1 0 8G 0 lvm [SWAP]
sr0 11:0 1 1024M 0 rom (cdrom)
```

Hier sieht man Disk *sda* mit 100G und zwei Partitionen *sda1*, *sda2*. *sda2* ist ein PV (LVM Physical Volume) mit Volume Group vg -> daraus LVs *vg-root* und *vg-swap*, gemountet entsprechend.

## **lsscsi (List SCSI Devices)**

**Beschreibung:** Listet erkannte SCSI-Geräte (inkl. SATA, SAS, USB Storage – diese werden SCSI-Subsys emuliert) auf. lsscsi zeigt je Gerät Host/Channel/Target/LUN, dann Gerätetyp, und eventuell Model und Seriennummer je nach Infile-4fdk7rvx39azjretinarqb】. Hilfreich, um z.B. alle angeschlossenen Festplatten und optischen Laufwerke anzuzeigen.

#### Syntax:

lsscsi [Optionen]

#### Optionen:

- **-v, --verbose:** Ausführlich – zeigt zusätzliche Infos (z.B. SCSI-Transport oder Zustände).
- **-t, --transport:** Zeigt SCSI-Transportinformationen (z.B. SATA, USB UAS, iSCSI etc.).
- **-L, --lunhex:** Zeigt die LUN in hexadezimaler Notation statt dezimal.
- **-H, --hosts:** Zeigt nur SCSI-Host-Adapter (Controller) an.
- **-c, --classic:** Benutzt "klassisches" Ausgabeformat (alter Stil, ohne Abkürzungen).
- **-g, --generic:** Fügt die entsprechenden sg-Device-Nodes hinzu (SCSI Generic /dev/sgX).
- **-s, --size:** Zeigt Größe (Kapazität) der Block-Geräte an, falls anwendbar (nur für Disk-type).
- **-p, --protection:** Zeigt Protection-Information (DIF/DIX).
- **-i, --scsi\_id:** Zeigt die SCSI-IDs im klassischen Sinn (T10 IDs).
- **-B, --usb:** Zeigt Bus-IDs (für USB-Geräte) mit an.

#### Beispiel:

```
[0:0:0:0] disk ATA Samsung SSD 860 1B6Q /dev/sda 500GB
```

```
[2:0:0:0] cd/dvd TSSTcorp CDDVDW SH-224DB SB00 /dev/sr0
```

```
[3:0:0:0] disk WD MyBook 25DA 4004 /dev/sdb 4TB
```

Format: [H:C:T:L] type vendor model rev /dev/sgX -> /dev/sdX. Hier sieht man SCSI-Gerät auf Host 0 (SATA), Host 2 (DVD), Host 3 (USB-HDD).

#### fdisk (Partition Table Editor)

**Beschreibung:** Interaktives Konsolen-Tool zum Anzeigen und Bearbeiten von Partitionstabellen auf Blockgeräten (MBR, GPT). Meistens wird fdisk verwendet, um

schnell die Partitionierung eines Laufwerks anzusehen (fdisk -l) oder bei MBR-Partitionierung Partitionen einzurichten. Für GPT kann man gdisk oder parted benutzen (fdisk unterstützt aber inzwischen auch GPT).

### Syntax:

fdisk [Optionen] <Gerät>

Ohne Optionen startet es ein interaktives Menü für das angegebene Gerät (z.B. /dev/sda).

### Optionen:

- **-l, --list:** Listet die Partitionstabellen aller verfügbaren Laufwerke auf [file-4fdk7rvx39azjretinarqb]. Oder wenn ein Gerät angegeben, nur diese. Z.B. fdisk -l /dev/sda zeigt Partitionen auf sda.
- **-s Partition:** Gibt die Größe der angegebenen Partition in Sektoren aus (veraltet, blockdev --getsz moderner).
- **-u (veraltet):** Wechselt die Ausgabeeinheit (Sektoren vs Zylinder) in alter fdisk. Heutige fdisk (util-linux) nutzt standard 512-Byte Sektoren in -l.
- **-B, -C, -H, -S:** Kann man Geometrie (Zylinder, Heads, Sectors) angeben (MBR Legacy, heute irrelevant).
- **-t Typ:** Zeigt Partitionen in bestimmtem Format (Dos, GPT, ...).

Interaktiver Modus: Kommandos wie m (Hilfe), p (Print Partitionstabelle), n (neue Partition), d (löschen), t (Typ ändern, z.B. 83 Linux, 82 Swap, in GPT GUIDs), w (write und exit), q (quit ohne save).

### Beispiel (List):

Disk /dev/sda: 238.5 GiB, ...

| Device    | Boot | Start     | End       | Sectors   | Size   | Id | Type                 |
|-----------|------|-----------|-----------|-----------|--------|----|----------------------|
| /dev/sda1 | *    | 2048      | 534527    | 532480    | 260M   | 7  | HPFS/NTFS/exFAT      |
| /dev/sda2 |      | 534528    | 409600000 | 409065473 | 195.1G | 83 | Linux                |
| /dev/sda3 |      | 409600001 | 419430400 | 9830400   | 4.7G   | 82 | Linux swap / Solaris |

Zeigt z.B. 3 Partitionen (sda1 NTFS, sda2 Linux, sda3 Swap).

### parted (Partitionierungstool)

**Beschreibung:** Moderner Partitionierer, unterstützt GPT und MBR sowie Größe > 2TB. Kann sowohl interaktiv (im eigenen Prompt) als auch direkt per Kommandozeile genutzt werden. Im Gegensatz zu fdisk beherrscht parted auch **Resizing** von Partitionen und Filesystems (begrenzte FS-Typ Unterstützung) und kann in Skripten genutzt werden.

### Syntax (non-interactive):

parted [-s] <Gerät> <Befehl> [Parameter...]

Mit -s (script) ohne interaktive Ausgabe.

### Häufige parted Befehle: (Nicht alle Optionen, nur exemplarisch)

- **print** – Zeigt Partitionstabelle des Geräts. Z.B. parted /dev/sdb print.
- **mklabel <Typ>** – Erstellt neue Partitionstabelle (Typ: gpt, msdos, etc.). Achtung: löscht alle bestehenden Partitionen.
- **unit <Einheit>** – Setzt Maßeinheit (MB, GB, %, etc.) für folgende Befehle.
- **mkpart Name Typ Anfang Ende** – Erstellt Partition. Z.B.: mkpart primary ext4 1MiB 10000MiB erstellt eine ~10GB Partition vom 1MB Offset. *Typ* hier "primary"/"logical" (bei msdos) oder nur Partitions-TypName (bei GPT optional ein FS-Typ Label ext4 etc.).
- **rm <Nr>** – Löscht Partition mit Nummer.
- **resizepart <Nr> <Ende>** – Passt Größe der Partition Nr an (End-Sektor). Muss danach FS separat resized werden (z.B. mit resize2fs).
- **name <Nr> <Name>** – Setzt den Partitionsnamen (GPT Label).
- **toggle <Nr> boot** – Boot-Flag toggeln (bei MBR Partitionen).
- **align-check Typ Nr** – Überprüft Ausrichtung von Partition Nr (optisch/ Zylinder).

Parted kann auch einfache FS-Aufgaben: mkfs integration, aber meist nutzt man separate Tools (mkfs.\*).

### Beispiele:

- parted /dev/sdc --script mklabel gpt mkpart primary ext4 0% 50% mkpart primary linux-swap 50% 100% – Initialisiert /dev/sdc als GPT, teilt in zwei: erste Partition bis Mitte ext4, zweite Swap ab Mitte bis Ende.
- Im interaktiven parted (ohne -s) bekommt man (parted) Prompt zum Eingeben der obigen Befehle nacheinander.

### mkfs (Make FileSystem)

**Beschreibung:** Erzeugt ein neues Dateisystem auf einer Partition oder Festplatte. mkfs ist ein Frontend, das je nach -t <FSTYP> das entsprechende Tool (mkfs.ext4, mkfs.xfs, etc.) aufruft. Häufig ruft man direkt das spezifische Formatprogramm auf.

### Syntax:

`mkfs -t <FSTYP> [Optionen] <Geräte/Partition>`

Beispiele: `mkfs -t ext4 /dev/sdb1`, `mkfs -t vfat /dev/sdb1`.

### **Wichtige Optionen (allgemein):**

- **-t <typ>** – Dateisystemtyp (ext4, ext3, ext2, xfs, vfat, ntfs, etc.).
- **-V** – zeigt ausführliche Versions-/Debug-Information (welches Tool aufgerufen wird).
- **-q** – quiet (weniger Ausgabe).

Jedes Dateisystem hat eigene spezifische Optionen, wenn man direkt `mkfs.ext4` etc. nutzt:

**ext4 (mkfs.ext4):** `-L <Label>, -U <UUID>` (benutzerdefiniert), `-m <Proz>` (Reserve-Blöcke Prozent, default 5%), `-O <Feature>` (ein/aus bestimmter FS-Features), `-E stride=` (RAID optimierungen), `-b blocksize`, etc.

**vfat (mkfs.vfat):** `-n <Label>, -F <32|16>` (FAT type), `-I` (Force if not partition), etc.

**xfs (mkfs.xfs):** `-L <Label>, -m crc=1` (z.B. CRC enable, default ON neu), etc.

**swap (mkswap):** `-L <Label>`, (swap selbst ist via mkswap erstellt).

*Wichtig:* Partition sollte entsprechend markiert/Dimension formatiert sein (z.B. parted type=linux-swap für swap etc.).

### **Beispiele:**

- `mkfs.ext4 /dev/sda3` – Formatiert sda3 mit ext4 Standardparametern.
- `mkfs.vfat -F 32 -n "MYUSB" /dev/sdb1` – Formatiert Partition sdb1 als FAT32 mit Label "MYUSB".
- `mkswap /dev/sda5` – Initialisiert Swap auf sda5 (später via swapon).
- `mkfs.xfs -f -L data /dev/sdc1` – Erzwingt Format auf sdc1 mit XFS, Label "data" (XFS verlangt -f wenn Partition schon ein FS enthält).

### **mount (Dateisystem einhängen)**

**Beschreibung:** Bindet ein Dateisystem (Partition, Datenträger oder virtuelles FS) in den Verzeichnisbaum ein. `mount` ohne Argumente listet aktuelle Einhängungen. `mount <Gerät> <Verzeichnis>` hängt ein Gerät an den angegebenen Ordner ein (der Ordner muss existieren und leer sein, typischerweise unter /mnt oder /media oder definierter Mountpoint in /etc/fstabfile-4fdk7rvx39azjretinarqb】 .

### **Syntax:**

`mount [Optionen] <Quelle> <Zielverzeichnis>`

### **Wichtige Optionen:**

- **-t <fstype>** – gibt den Dateisystemtyp an (ext4, vfat, nfs, cifs, etc.). Kann oft weggelassen werden, da mount es automatisch erkennt (über blkid).
- **-o <Optionen>** – übergibt eine durch Komma getrennte Liste von Mount-Optionenfile-4fdk7rvx39azjretinarqb】 :

  - **defaults** (rw,suid,dev,exec,auto,nouser,async),
  - **ro/rw** (read-only / read-write),
  - **noexec** (keine Ausführung von Programmen erlauben), **nosuid** (SUID/SGID-Bits ignorieren), **nodev** (keine Gerätedateien interpretieren),
  - **sync** (synchrone IO), **async** (Standard, asynchron gepuffert),
  - **user/nouser** (Allow normal user to mount / nur root),
  - **uid=, gid=** (für FAT/NTFS, setze Besitzer der gemounteten Dateien),
  - **umask=, dmask=, fmask=** (Rechte-Masken für FAT/NTFS),
  - **iocharset=, codepage=** (für vfat),
  - **loop** (Loop-Device mount einer Datei als FS, evtl. -o loop oder auto),
  - **remount** (ein bereits gemountetes FS mit neuen Optionen versehen, z.B. mount -o remount,rw /mnt/point),
  - FS-spezifisch: z.B. **acl**, **user\_xattr** (Ext4 Features), **compression**, **encrypt** (btrfs), etc.

- **-L <Label>, -U <UUID>** – anstatt Device-Node, per Label oder UUID identifizieren. mount -U 1234-ABCD /mnt/usb.
- **--bind** – führt einen Bind-Mount aus: mountet ein vorhandenes Verzeichnis an eine zweite Stelle im Verzeichnisbaum. Syntax: mount --bind /pfad/quelle /pfad/ziel. (Nützlich, um z.B. Verzeichnisse in Chroots sichtbar zu machen). Option **--rbind** rekursiv (inkl. Unterpunkte).
- **-a** – Mountet alles was in /etc/fstab als "auto" markiert ist und nicht noauto. (Ausführung typ. beim Boot).
- **-n** – führt Mount ohne Eintrag in /etc/mtab aus (mtab ist heute oft Symlink zu /proc/self/mounts). Selten benötigt.
- **-v** – verbose (gibt Meldungen aus).

### **Beispiele:**

- mount /dev/sdb1 /media/usb -o uid=1000,gid=1000,umask=022 – hängt einen FAT32-Stick so ein, dass Benutzer mit UID/GID 1000 Besitzer ist (typisch eigener User), und Rechte 755 auf Dateien/Ordner (umask 022).

- `mount -t nfs 192.168.1.10:/export/share /mnt/share` – hängt eine NFS-Freigabe ein (NFS FS muss installiert sein).
- `mount -o loop diskimage.iso /mnt/iso` – mountet ein ISO-Image als Loop-Device unter /mnt/iso.
- `mount | grep sda` – sieht, ob /dev/sdX gemountet ist (ohne Parameter listet mount alles).

## **umount (Unmount FileSystem)**

**Beschreibung:** Hängt ein gemountetes Dateisystem wieder aus. Schreibvorgänge werden abgeschlossen und der Puffer geleert. Syntax kann entweder der Mountpoint oder das Device angegeben werden】 .

### **Syntax:**

`umount [Optionen] <Mountpunkt | Gerät>`

### **Wichtige Optionen:**

- **-f, --force:** Erzwingt das Aushängen eines FS. Sollte nur im Notfall (z.B. NFS-Server down) benutzt werden, da es evtl. Datenverlust verursachen kann.
- **-l, --lazy:** "Lazy unmount". Markiert das FS als ausgeworfen, aber falls noch offene Handles bestehen, wird es erst tatsächlich ausgehängt, wenn diese geschlossen sind. (Hängt unmittelbar virtuell aus, lässt Rest im Hintergrund erledigen). Nützlich bei hängenden Prozessen.
- **-R, --recursive:** Hängt auch alle Unter-Mounts mit ab, die auf dem angegebenen Pfad liegen (rekursiv).
- **-v:** verbose.

### **Beispiele:**

- `umount /media/usb` – hängt das unter /media/usb eingehängte Gerät aus.
- `umount /dev/sdb1` – selbes, angesteuert über Gerät.
- `umount -f -a -t nfs` – erzwingt das Aushängen aller NFS-Mounts (z.B. bei Netzproblemen).

(Tipp: Wenn `umount` meldet "device is busy", kann man mit `lsof` oder `fuser` herausfinden, welcher Prozess noch auf das FS zugreift.)

## **df (Disk Free)**

**Beschreibung:** Zeigt Übersicht des belegten und freien Speicherplatzes auf eingehängten Dateisystemen

## Syntax:

```
df [Optionen] [Dateipfad...]
```

Ohne Argumente zeigt alle gemounteten FS. Mit Pfad ermittelt df, auf welchem FS dieser Pfad liegt, und zeigt nur dessen Werte.

## Optionen:

- **-h, --human-readable**: Formatiert Größen in lesbarer Form (KiB, MiB, GiBfile-4fdk7rvx39azjretinarqb】. Sehr gebräuchlich (Standard rohes Blockzahlen unhandlich).
- **-a, --all**: Schließt pseudo- oder 0-Size-Dateisysteme ein (wie proc, sysfs, etc.), die sonst ausgelassen werden.
- **-T, --print-type**: Zeigt Spalte mit Dateisystemtyp (ext4, tmpfs, nfs, etc.).
- **-i, --inodes**: Zeigt Inode-Auslastung statt Blockgrößen (frei/belegt Inodes pro FS). Nützlich, wenn "no space" durch Inode-Erschöpfung.
- **-t <typ>, -x <typ>**: Nur bestimmte FS-Typen anzeigen bzw. ausschließen. Z.B. df -x tmpfs -x devtmpfs um echte Disks anzuzeigen.
- **-B <Größe>**: Ausgabe in Blöcken von angegebener Größe. Z.B. -B MB (MByte) oder -B 1M.
- **--output[=FELDER]**: Benutzerdefinierte Spaltenausgabe. Z.B. df --output=source,fstype,size,used,avail,pcent,target.

## Beispielausgabe (df -h):

| Filesystem | Size | Used | Avail | Use% | Mounted on |
|------------|------|------|-------|------|------------|
| /dev/sda2  | 50G  | 30G  | 18G   | 63%  | /          |
| udev       | 3.9G | 0    | 3.9G  | 0%   | /dev       |
| tmpfs      | 798M | 1.2M | 797M  | 1%   | /run       |
| /dev/sda1  | 300M | 50M  | 250M  | 17%  | /boot      |

## du (Disk Usage)

**Beschreibung:** Summiert und zeigt die Größe von Verzeichnissen und Dateien an. du kann rekursiv durch Verzeichnisse gehen und den Platzverbrauch berechnenfile-4fdk7rvx39azjretinarqb】. Standardmäßig zeigt es für jedes Unterverzeichnis den Wert. Oft nutzt man du -sh <dir> um eine schnelle Gesamtgröße zu bekommen.

## Syntax:

```
du [Optionen] [Pfad...]
```

Ohne Pfad = aktuelles Verzeichnis.

### Optionen:

- **-h, --human-readable**: Größen in menschenlesbare Einheiten (K,M,G) ausgeben  
file-4fdk7rvx39azjretinarqb】 .
- **-s, --summarize**: Nur Summen pro angegebenem Argument ausgeben, *nicht* rekursiv für Unterordner-4fdk7rvx39azjretinarqb】 . (Gibt also Gesamtgröße des Pfades).
- **-A, --apparent-size**: Rechnet mit logischen Dateigrößen statt belegtem Blockplatz. (Z.B. sehr sparsame Sparse Files würden sonst weniger anzeigen; --apparent-size zeigt nominelle Größe).
- **-d N, --max-depth=N**: Rekursionstiefe begrenzen. Z.B. du -h --max-depth=1 /var gibt Größe von /var und dessen Unterordner erste Ebene. (--max-depth=0 wäre wie -s).
- **-c, --total**: Am Ende eine Gesamtzeile "total" ausgeben (Summe aller ausgegebenen Einträge). Bei -s von mehreren Pfaden praktisch.
- **-x, --one-file-system**: Geht nicht in andere eingehängte Dateisysteme über. (Beschränkt auf Partition/FS der Startpfade).
- **-L, --dereference**: Folgt symbolischen Links (sonst ignoriert du Symlinks, rechnet sie als 0 Bytes). Vorsicht: Kann zu doppeltem Zählen führen, wenn Links in Kreis oder auf Oberverzeichnis zeigen.
- **--exclude=<PATTERN>**: Schließt Pfade aus, die zum Pattern passen. Z.B. du -sh -exclude=".git" oder mehrere --exclude.

### Beispiele:

- du -sh /home/\* – Zeigt Summe der Größen aller Benutzerverzeichnisse unter /home (jedes in human-readable).
- du -a /etc | grep -E "mysql.\*\.cnf\$" – Listet die Größe jeder einzelnen Datei unter /etc, filtert jene, deren Name auf mysql\*.cnf endet. (Option -a listet auch Dateien und nicht nur Verzeichnisse).
- du -d2 /var – Größe von /var, dessen Subdirs und Sub-Subdirs (Tiefe 2).

### e2label (ext2/3/4 Volume Label)

**Beschreibung:** Zeigt oder ändert das Label (Namen) eines ext2/ext3/ext4 Dateisystems. Das *Label* ist eine im Superblock gespeicherte Zeichenkette, die man z.B. in fstab zum Mounten verwenden kann (LABEL=<name>).

### Syntax:

## e2label <Gerät> [NeuesLabel]

Ohne zweites Argument zeigt es das aktuelle Label des ext-Volumes an. Mit Label schreibt es dieses in den FS.

### Beispiel:

- e2label /dev/sda3 -> Ausgabe z.B. UbuntuRoot.
- e2label /dev/sda3 Archive -> setzt das Label von sda3 auf "Archive".

(Für andere FS gibt es analoge Tools: XFS nutzt xfs\_admin -L, FAT/NTFS mlabel oder dosfslabel/ntfslabel etc.)

## resize2fs (ext2/3/4 Dateisystem verkleinern/vergrößern)

**Beschreibung:** Passt die Größe eines ext2/3/4-Dateisystems an. Wird verwendet nach dem Ändern der Partitionsgröße (mit fdisk/parted). resize2fs kann im online-Modus vergrößern (wenn gemountet, ext3/4), fürs verkleinern muss das Volume ausgehängt und ein Filesystem-Check gemacht werden.

### Syntax:

resize2fs [Optionen] <Gerät> [NeueGröße]

NeueGröße kann in K, M, G oder als Anzahl 4K-Blöcke angegeben werden. Lässt man sie weg, wird auf Maximum (Part.-Ende) vergrößert.

### Optionen:

- **-p** – progress anzeigen (Fortschritt).
- **-f** – zwingen, auch wenn FS als nicht sauber markiert (normal will fsck vorher laufen bei Verkleinern).
- **-F** – flush caches vorher (sorgt für up-to-date Infos; Standard normal).
- **-M – minimal**: verkleinert das FS auf die kleinstmögliche Größe (alle Daten gerade passend). Äquivalent zu resize2fs <dev> -M (dauert Berechnung).

### Beispiele:

- Nach Vergrößern der Partition: resize2fs /dev/sda2 – erkennt neue Partitionsgröße und wächst das ext4 FS auf vollen Platz.
- Verkleinern: umount /dev/sda2; e2fsck -f /dev/sda2; resize2fs /dev/sda2 20G – erst aushängen, fsck erzwingen, dann auf 20 GiB verkleinern. Anschließend Partition selbst verkleinern mit parted.

## tune2fs (Tune ext2/3/4 Filesystem Parameters)

**Beschreibung:** Ändert diverse Parameter eines bestehenden ext-Dateisystems: Reserveblöcke, Check-Intervalle, Label, Journal einschalten, usw.

### Syntax:

tune2fs [Optionen] <Gerät>

### Häufige Optionen:

- **-l** – Listet alle Superblock-Parameter (Filesystem-Info) auf [freedesktop.org](https://freedesktop.org). Zeigt z.B. Inode-Größe, Features, Letzter Mount, Nächster geplanter Check, etc. (Read-only Operation).
- **-c <MaxMounts>** – Setzt die maximale Anzahl Mount-Vorgänge bis zur erzwungenen fsck-Prüfung. Z.B. -c 0 = deaktiviert mount-count-basierten Check.
- **-i <Interval>** – Setzt das Zeit-Intervall für Zwangs-FSCK. z.B. -i 30d = alle 30 Tage. -i 0 = kein zeitbasierter Check.
- **-r <AnzRes>** – Setzt Anzahl der reservierten Blöcke (für root). Standard 5% bei ext4 auf <1TB. Kann man z.B. auf 1% reduzieren auf großen Volumes, z.B. tune2fs -r 0 /dev/sdb1 um Reserven komplett abzuschalten (nicht empfohlen auf /). Alternativ -m <Prozent> (z.B. -m 1).
- **-L <Label>** – Ändert Label (wie e2label).
- **-U <UUID>** – Ändert die UUID (zufällig oder angegeben). (Vorsicht: fstab etc. müssen angepasst).
- **-J** – Journal-Optionen: z.B. -O has\_journal um Journal hinzuzufügen (macht aus ext2 -> ext3), oder -O ^has\_journal Journal entfernen (ext3->ext2), oder -j Kurzoption (Standardwerte Journal hinzufügen).
- **-O <FeatListe>** – Aktiviert (feature) oder deaktiviert (^feature) Filesystem-Features. Z.B. -O casefold (für ext4 ab 5.2: Case-Insensitive-Folder Support) oder ^huge\_file (Feature abschalten – Achtung, Daten verlieren, nur wenn FS leer). Nur bestimmte Features togglebar, meist erfordert offline und vollständiges Backup.
- **\*\*-E <extendopt>\*\*** – Erweiterte Einstellungen: z.B. stripe=<StripeSize>(RAID stripe in Blöcken),lazy\_itable\_init=1` (schnell format, Inode-Table lazy Clearing).
- **-M <Verzeichnis>** – Markiert FS als gemountet an <Verzeichnis> (setzt den Mountpoint in Superblock – meist rein informativ).

### dumpe2fs (Dump ext2/3/4 Superblock)

**Beschreibung:** Gibt ausführliche Informationen über ein ext2/3/4-Dateisystem aus – Inhalt des Superblocks und der Block Group Deskriptore. Zeigt ähnlich tune2fs -l Informationen, plus Aufteilung der Blöcke in Gruppen, freie Blöcke/Inodes pro Gruppe etc. Hilfreich für Diagnose (z.B. Backup-Superblock Positionen).

## Syntax:

dumpe2fs [Optionen] <Gerät>

## Optionen:

- **-h** – Zeigt nur den Inhalt des Superblocks und der Gruppen-Deskriptoren (ohne die Liste aller freien Blöcke/Inodes). Das ist die gängigste Verwendung, da ohne -h die Ausgabe sehr lang sein kann (listet freie Blöcke/Inodes aller Groups [wiki.ubuntuusers.de](http://wiki.ubuntuusers.de)).
- **-x** – Zeigt auch die detaillierten Gruppen-Informationen (wie freie Blöcke/Inodes-Bitmap). Standard ohne -h tut das sowieso.
- **-b** – Listet die Nummern der freien Blöcke. (Sehr viel Output auf großem FS.)
- **-i** – Listet die freien Inode-Nummern.

## Beispiel (gekürzt dumpe2fs -h /dev/sda2):

Filesystem volume name: <none>

Last mounted on: /

Filesystem UUID: 4e8fea21-...-b6c7

Filesystem magic number: 0xEF53

Filesystem revision #: 1 (dynamic)

Filesystem OS type: Linux

Block size: 4096

Block count: 25600000

Reserved block count: 1280000

Free blocks: 15034567

First Block: 0

...

Journal inode: 8

Journal backup: inode blocks

...

Zeigt Label, UUID, Blocksize, Anzahl Blöcke, Reserve, Anzahl freie etc., und welche Features aktiviert sind.

## cryptsetup (LUKS Disk Encryption Setup)

**Beschreibung:** Tool zur Einrichtung von **verschlüsselten Datenträgern** mit LUKS (Linux Unified Key Setup) sowie zum Öffnen/Schließen solcher Verschlüsselungs-Volume】 . cryptsetup kann ein Gerät mit LUKS initialisieren, Passphrase setzen, und im Betrieb ein verschlüsseltes Volume als Device-Mapper-Gerät bereitstellen, das dann gemountet werden kann.

### Typische Aktionen (Subcommands):

- **luksFormat** – Initialisiert ein Blockgerät mit LUKS-Verschlüsselungsheader und fordert eine Passphrase zum Einrichten an. Beispiel: cryptsetup luksFormat /dev/sdb1. *Achtung:* Löscht alle vorhandenen Daten auf dem Gerät.
- **open** oder **luksOpen** – Öffnet ein verschlüsseltes Volume (fragt Passphrase). Syntax: cryptsetup luksOpen /dev/sdb1 geheim\_vol. Dies erstellt ein Device /dev/mapper/geheim\_vol, das nun entschlüsselt agiert und z.B. mit mkfs formatiert oder gemountet werden kan】 .
- **close** – Schließt ein vorher geöffnetes LUKS-Device: cryptsetup close geheim\_vol (dann ist /dev/mapper/geheim\_vol weg).
- **luksAddKey** – Fügt einen weiteren Schlüssel (Passphrase) hinzu. LUKS unterstützt mehrere Key-Slots. cryptsetup luksAddKey /dev/sdb1 /pfad/zur/schlüsseldatei oder interaktiv eine neue Passphrase eingeben (nachdem man eine gültige alte eingibt).
- **luksRemoveKey** – Entfernt einen Schlüssel (kann durch Angabe der zu entfernenden Passphrase oder Slot-Nummer erfolgen).
- **luksChangeKey** – Ändert eine bestehende Passphrase (in einem Slot) durch eine neue (kombi aus Remove+Add effizienter).
- **luksHeaderBackup** / **luksHeaderRestore** – Backup bzw. Wiederherstellen des LUKS-Headers (empfohlen, falls Header korrupt -> Daten verloren).
- **status** – Zeigt Informationen über ein offenes Device-Mapper Volume (Name, verschlüsseltes Dev, Cipher, Keylen, etc.): cryptsetup status geheim\_vol.

### Wichtige Optionen:

- **-y** (im Format-Befehl): fragt neues Passwort zweimal zur Verifizierung ab.
- **-d <Datei>**: verwendet eine Datei als Schlüsseldaten (anstatt interaktiv Passwort). -d - kann genutzt werden, um vom stdin zu lesen.
- **--key-size <bits>**: Schlüsselgröße (in Bits) für die Verschlüsselung (z.B. 256).
- **--cipher <name>**: Zu nutzender Verschlüsselungsalgorithmus und Mode. Standard z.B. aes-xts-plain64 bei LUKS2. Kann z.B. aes-cbc-essiv:sha256 oder anderes angegeben werden.

- **--hash <hash>**: Hash-Algorithmus für Passphrase-Derivation (PBKDF). Standard modern ist z.B. Argon2id oder früher sha256.
- **--iter-time <ms>**: Wie lange (Millisek) auf PBKDF verwendet werden soll (Bestimmt Komplexität).
- **--type luks1/luks2**: LUKS Version festlegen (aktuell Standard LUKS2, kann optional LUKS1 für Kompatibilität gewählt werden).
- **-q**: Quiet (unterdrückt Ausgabe, z.B. Warnung bei Format).

### **Beispiel Workflow:**

1. **Formatieren**: cryptsetup -y luksFormat /dev/sda5 (legt LUKS auf Partition sda5 an, fragt 2x Passwort).
2. **Öffnen**: cryptsetup luksOpen /dev/sda5 geheim (Passwort eintippen, erstellt /dev/mapper/geheim).
3. **Format FS**: mkfs.ext4 /dev/mapper/geheim.
4. **Mount**: mount /dev/mapper/geheim /mnt/secure. Jetzt Daten drauf speichern.
5. **Aushängen+Schließen**: umount /mnt/secure && cryptsetup close geheim.

(LUKS ermöglicht bis zu 8 Passphrases; man kann z.B. mit luksAddKey einen zweiten hinzugeben, sodass entweder Passphrase A oder B das Volume öffnet. Das Gerät /dev/sda5 kann in /etc/crypttab und /etc/fstab eingetragen werden für Boot-Time auto-open via keyfile oder prompt.)

### **shred (Securely wipe file/device)**

**Beschreibung:** Überschreibt eine Datei oder ein ganzes Gerät mehrfach mit zufälligen Daten, um ein Wiederherstellen gelöschter Inhalte zu erschweren. shred ist nützlich zum sicheren Löschen auf magnetischen HDDs. Bei SSDs mit Wear-Leveling ist Effekt begrenzt – dort lieber ATA Secure Erase oder physical destruction.

### **Syntax:**

shred [Optionen] <Datei1> [Datei2...]

### **Optionen:**

- **-n N** – Anzahl der *zusätzlichen* Überschreib-Pässe (Standard 3). D.h. 3 bedeutet insgesamt 3 mal random + final pass (siehe -z).
- **-z, --zero** – Überschreibt am Ende noch einmal mit Null-Bytes, um das Muster zu verwischen (anstatt dass zuletzt zufälliges Rauschen bleibt). Standard: aktiv (bei neueren shred ist -z default an).

- **-f, --force** – Ändert Schreibschutz falls nötig (chmod u+w), um schreiben zu können.
- **-u, --remove[=how]** – Nach dem Überschreiben die Datei löschen (unlink). Optional how kann unlink (Standard) oder wipe oder wipesync sein: *wipe* überschreibt erst mit Null dann löscht, *wipesync* macht zusätzlich Sync. Standard -u bereits tut *wipesync*.
- **-v, --verbose** – Auskunft über Fortschritt.
- **-x, -X, --exact** – Überschreibt Datei genau Byte-für-Byte anstelle Dateigröße auf nächste Blockgröße aufzurunden (Standard shred nutzt ganze Blocks, kann über Dateiende hinaus schreiben falls Dateigröße nicht mod Blocksize). *Exact* vermeidet das, aber hinterlässt evtl. un-geschredderte Slack am Ende.
- **-s N, --size=N** – Behandelt die Datei als N Byte lang (statt aktuelle Länge). Um auch ungenutzten Platz (z.B. Slack Space/EOF beyond) im Inode zu überschreiben, kann man etwas größer angeben.

### Beispiele:

- `shred -u file.txt` – Überschreibt *file.txt* viermal (3 rand + 1 zero) und löscht sie anschließend.
- `shred /dev/sdb` – Überschreibt komplett `/dev/sdb` mit 3 Runden Zufallsdaten (zerstört alle Daten/Partitionen darauf).
- `shred -n 0 -z -u secret.dat` – Einmal mit Nullen überschreiben und entfernen (hier wollte man evtl. nur einfaches Überschreiben und löschen).

**Warnung:** Shred wirkt nicht zuverlässig auf: **logischen Volumes** in Files (z.B. Datei in FS kann redundante Kopien (Journal, Snapshots) haben), **RAID, SSD, Netzwerkdateisysteme, Dateisysteme mit Copy-on-Write (btrfs, ZFS)** – dort können Daten an mehreren Orten stehen. In solchen Fällen besser Vollverschlüsselung verwenden oder das gesamte physische Medium überschreiben.

### iostat (IO Statistics)

**Beschreibung:** Zeigt Statistik zur Systemauslastung von CPU und Input/Output für Blockgeräte (Datenträger). iostat gehört zum Paket sysstat. Es hilft zu sehen, wie stark Platten ausgelastet sind (%util) und Durchsatz (MB/s), IOPS etc., sowie generelle CPU-Idle/Wait Zeite】 .

### Syntax:

`iostat [Optionen] [Intervall [Durchläufe]]`

Ohne Argument gibt einen Durchschnitt seit Boot aus. Mit Intervall (Sekunden) aktualisiert regelmäßig.

## **Wichtige Optionen:**

- **-c** – Zeigt nur CPU-Statistiken.
- **-d** – Zeigt nur Disk (Geräte) Statistiken.
- **-k, -m** – Ausgabe in KiB/s oder MiB/s (anstatt Blöcke/s). (Aktuelle sysstat erlaubt -block-size).
- **-p [Gerät]** – Detaillierte Partitions-Statistiken einbeziehen (normal zeigt iostat nur Whole-Device). Z.B. -p sda zeigt sda und sda1,sda2,...
- **-x** – Ausgabe in erweiterter Form (Pro Gerät: zusätzlich Auslastungsprozentsatz, Average Request Size, Await Times etc.) . Sehr nützlich, da Standard ohne -x recht knapp.
- **-t** – Zeigt Timestamp vor jeder Ausgabe.
- **-y** – Unterdrückt erste Ausgabe (since boot), beginnt erst mit Werten ab erstem Intervall.

## **Interpretation (bei iostat -x):**

Spalten pro Device: rrqm/s wrqm/s (Merge von I/O, veraltet bei modernen Kernel meist 0), r/s w/s (reads/writes per sec), rkB/s wkB/s (Throughput), avgrq-sz (Durchschnittl. Requestgröße in Sektoren oder KB), avgqu-sz (durchschn. Queue length), await (Durchschnittliche Wartezeit pro IO ms), svctm (durchschn. Servicezeit pro IO ms – oft nicht mehr zuverlässig), %util (Prozentuale Zeit, die das Gerät aktiv war innerhalb der Messperiode – 100% bedeutet permanent beschäftigt).

## **Beispiel:**

```
Device r/s w/s rkB/s wkB/s avgrq-sz avgqu-sz await svctm %util
sda 12.5 30.2 500.0 250.0 40.0 0.5 10.0 2.00 85.0
```

Hier sda ~85% ausgelastet, wartet ~10ms per IO, etwa 42.7 IO/s (r+s) mit ~750kB/s.

## **ioping (I/O Ping)**

**Beschreibung:** Ermittelt Latenz (und ggf. Durchsatz) von Storage in Echtzeit ähnlich dem Prinzip von ping im Netzwerk. Es führt schnelle sequenzielle oder wahlfreie Lesezugriffe durch und misst deren Zeit. Praktisch um Latenz auf Disks oder bestimmten Verzeichnissen (Dateisystem) festzustellen.

## **Syntax:**

```
ioping [Optionen] <Ziel>
```

Ziel kann ein Pfad (Datei oder Verzeichnis) sein für Filesystem IO, oder z.B. ein Blockgerät (mit -D für Direkten Device Zugriff) oder sogar ein bestimmter I/O-Engine (IO depth).

### Wichtige Optionen:

- **-c <count>** – Anzahl der Ping-Requests (Standard unendlich, bis abgebrochen).
- **-i <interval>** – Wartezeit zwischen Pings (Sekunden, default 1s). Kann in ms angeben z.B. -i 0.1.
- **-s <bytes>** – Größe der zu lesenden Daten pro Ping (Standard 4KiB).
- **-R** – Ausgabe im Rohformat (Maschinenlesbar).
- **-D** – Direct-Device Mode: Arbeitet auf Blockgeräte direkt (bypasses FS-Cache).
- **-A** – Ausgabe als Aggregatstatistik nur (kein laufendes Ping, nur Summary).
- **-S** – Sync/Sequenziell: testet sequenzielles Lesen statt wahlfrei. (Standard ioping macht random position reads).
- **-W** – führt Schreib-Tests durch statt Lesen (Achtung kann Daten überschreiben zerstören, am besten auf extra Gerät/Datei).
- **-P <N>** – Parallel Requests (Thread count).
- **-q** – keine laufende Ausgabe, nur Summary am Ende.

### Beispiele:

- ioping . – Sendet kontinuierlich 4K-Leseanfragen ins aktuelle Verzeichnis (random in FS), zeigt Latenzen, bis Ctrl+C.

```
4096 bytes from . (ext4 /dev/sda2): request=1 time=0.47 ms
```

```
4096 bytes from . (ext4 /dev/sda2): request=2 time=0.44 ms
```

```
... (zeigt typisch ~0.5ms auf SSD, ~10-15ms auf HDD) ...
```

Nach Abbruch gibt es eine **Summary** mit min/avg/max/mdev (Jitter).

- ioping -c 10 /mnt/backup – Misst 10 Zugriffe im FS /mnt/backup (z.B. externer USB), dann stop.
- ioping -R /dev/sdb – Führt 1 Sekunde lang Messungen auf /dev/sdb durch und gibt am Ende aggregated IOPS und Bandbreite.
- ioping -A -S -c 50 /file.iso – Liest sequenziell 50 Anfragen durch große Datei und gibt am Ende Durchschnittszeit.

(Tool muss ggf. installiert werden, gehört nicht standard zu allen Distr.)

## Geräte, Prozesse, Speicher und Kernel

### hwinfo (Hardware Info)

**Beschreibung:** Liefert detaillierte Informationen zur Hardware des Systems. hwinfo probiert, alle verfügbaren Hardware-Komponenten aufzulisten: CPU, RAM, Festplatten, USB, PCI-Geräte, Netzwerk, Monitor, etc., inklusive ihrer Eigenschaften und Treiber]. Oft auf openSUSE installiert, aber auch auf anderen Distros verfügbar.

#### Syntax:

```
hwinfo [Optionen] [Hardwareklasse]
```

#### Optionen / Klassen:

- Ohne Angabe gibt hwinfo *alles* aus (sehr lang). Besser, man filtert nach Klasse: z.B. hwinfo --cpu, hwinfo --disk, --gfxcard, --netcard, --printer, --usb, --bios, --memory etc.
- **--short:** Kurzfassung ausgeben (ein Eintrag pro Gerät in einer Zeile).
- **--brief:** Noch knapper (ähnlich short).
- **--summary:** Summaries (z.B. nur Zahl der Disks, CPUs etc.).
- **--verbose <level>:** Detaillierungsgrad (0-9). Default 1 (schon sehr ausführlich).
- **--debug N:** Setzt Debug-Level (für Entwickler, welche Module hwinfo benutzt).
- **--hwitem <ID>:** Sucht nach Hardware mit bestimmter ID (z.B. PCI-ID).
- **--save-config <datei> / --load-config <datei>:** Speichert erkannte Hardware in Datei zum späteren Vergleich.

*Beispiel:* hwinfo --cpu könnte ausgeben:

```
01: None 00.0: 10103 CPU
```

```
[Created at cpu.462]
```

```
Unique ID: rdCR.jX8Un+o2ZT1
```

```
Hardware Class: cpu
```

```
Arch: X86-64
```

```
Vendor: "GenuineIntel"
```

```
Model: 6.158.10 "Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz"
```

```
... (viele weitere Details Frequenzen, Cache etc.) ...
```

hwinfo --short summarisiert z.B. Hauptgeräte und Partitionen.

## **dmidecode (DMI Table Decode)**

**Beschreibung:** Gibt Informationen aus dem BIOS/UEFI-DMI (SMBIOS) heraus, wie Hersteller, Produktnamen, Seriennummern von System/Mainboard, BIOS-Version, verbauter Speicher, Anschlüsse etc]. Hilfreich für Hardware-Inventarisierung (Seriennummer, RAM-Module, OEM strings).

### **Syntax:**

dmidecode [Optionen]

### **Optionen:**

- **-t <string|Nummer>, --type:** Nur bestimmte DMI-Typen ausgeben. Typ kann Name oder Zahl sein. Z.B.:
  - bios (DMI type 0),
  - system (1),
  - baseboard (2),
  - processor (4),
  - memory (Memory Device, 17),
  - cache (7),
  - connector (8), slot (9), etc.

Man kann -t memory um nur RAM-Infos zu sehen (Größe, Typ, Hersteller, Seriennummer der DIMMs] .
- **-s <keyword>, --string:** Einzelne SMBIOS-Einträge gezielt ausgeben (z.B. bios-vendor, bios-version, system-manufacturer, system-serial-number, baseboard-asset-tag, processor-frequency etc.).
- **-q:** Quiet (unterdrückt weniger wichtige DMI-Eintrags-Labels wo "Not Provided").
- **-u:** Raw dump (hex) der DMI-Daten.
- **--dump-bin file, --from-dump file:** Dump SMBIOS in Datei bzw. lese aus Dump (z.B. offline Analyse).

*Beispiel:* dmidecode -t system könnte zeigen:

System Information

Manufacturer: Dell Inc.

Product Name: Latitude 5490

Version: None

Serial Number: ABCD123

UUID: 4c4c4544-0034-5810-8055-b6c04f513332

Wake-up Type: Power Switch

SKU Number: 07E6

Family: Latitude

dmidecode -t memory zeigt pro RAM-Riegel z.B. Größe, Form Factor (SODIMM), Type (DDR4), Speed, Manufacturer, Serial, Part Number.

### **lspci (List PCI Devices)**

**Beschreibung:** Zeigt alle PCI/PCIe-Geräte im System (angeschlossene Hardware am PCI-Bus: z.B. Netzwerkkarten, Sound, USB-Controller, Grafikkarten, SATA-Controller etc.). Es listet Geräte-ID und -Namen anhand einer Datenbank (/usr/share/misc/pci.ids).

#### **Syntax:**

`lspci [Optionen]`

#### **Optionen:**

- **-v, -vv, -vvv:** Verbose-Ausgabe in ein oder mehreren Stufen (mehr Register-Informationen). -vvv kann *sehr* ausführlich sein (inkl. Capabilities, Konfigspace).
- **-k:** Zeigt Kernel-Treiber und Module, die für jedes Gerät verwendet werden (welcher Treiber zugeordnet ist). (Z.B. Kernel driver in use: e1000e für Intel NIC).
- **-t:** Zeigt die hierarchische Baumstruktur der PCI-Bus-Topologie (wer hängt an welchem Bus/Bridge).
- **-nn:** Zeigt neben dem Gerätenamen auch Vendor:Device ID in eckigen Klammern. -n nur numeric IDs ohne Namen. -nn= beide. Z.B. Ethernet Controller [8086:1533] etc.
- **-d [<vendor>:<device>]:** Filtert nach bestimmten PCI IDs. `lspci -d 8086`: zeigt nur Intel-Geräte, `-d :1533` nur Geräte mit Device 1533.
- **-s [<domain>:<bus>:<slot>.<func>]:** Filtert nach Bus-Adresse (wie in `lspci -t` oder `/sys/bus/pci`). Z.B. `-s 00:1f.2` zeigt nur dieses Gerät.
- **-m:** Maschinenlesbar (gibt in einer CSV-artigen Formatzeile pro Gerät: Domain Bus Slot, Klass, Vendor, Device, SVendor, SDevice).
- **-xxx:** Dump des gesamten PCI-Konfigurationsspeichers (64 Bytes beyond standard) – Vorsicht sensibel. -xx nur ersten 256 Bytes (std config).

- **-H1:** Nutzen von /proc/bus/pci statt direct (historisch, meist nicht nötig).
- **-Q:** Zeigt das Querier-Modell (Kernel vs direct vs /proc).

### **Beispiel Ausgabe (gekürzt):**

```
00:00.0 Host bridge: Intel Corporation Device 5914
00:02.0 VGA compatible controller: Intel Corporation UHD Graphics 620
00:14.0 USB controller: Intel Corporation Sunrise Point-LP USB 3.0 xHCI Controller (rev 21)
00:16.0 Communication controller: Intel Corporation Management Engine Interface (rev 21)
...
03:00.0 Network controller: Intel Corporation Dual Band Wireless-AC 8265 (rev 78)
```

Mit -nn würde man z.B. sehen 00:02.0 VGA controller [0300]: Intel Corporation UHD Graphics 620 [8086:5917]. Mit -k ergänzt: Kernel driver in use: i915.

## **lsusb (List USB Devices)**

**Beschreibung:** Listet USB-Geräte auf, die am System angeschlossen sind (inkl. interne Hubs etc.). Zeigt Vendor und Produktnamen falls bekannt (Datenbank /usr/share/misc/usb.ids) und ID.

### **Syntax:**

lsusb [Optionen]

### **Optionen:**

- **-v:** Verbose, zeigt detaillierte Descriptor-Infos pro Gerät (ähnlich lsusb -vv noch detaillierter). Das kann sehr lang sein.
- **-t:** Baum-Darstellung der USB-Geräte-Hierarchie (welches Device hängt an welchem Port/Hub) – ähnlich lspci -t.
- **-s <bus>:<dev>:** Zeigt nur spezifisches Gerät (per Bus und Device Nummer). Diese Nummern sieht man in lsusb Standardausgabe vorn.
- **-d vid:pid:** Filtert nach VendorID:ProductID (Hex).
- **-D <device-file>:** Liest Infos aus einer Geräte-Datei (z.B. /dev/bus/usb/001/007) – erfordert Root.
- **-w:** Ausgabe als mit usbdump kompatibler Text (Wire format).
- **-v -v (or -vv):** liefert noch detailliertere Dump (inkl. Hexdump der HID Deskriptoren etc.).

### **Beispiel Ausgabe:**

Bus 002 Device 003: ID 0bda:0316 Realtek Semiconductor Corp. Card Reader

Bus 002 Device 002: ID 8087:0aaa Intel Corp.

Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub

Bus 001 Device 004: ID 0c45:671f Microdia Integrated Webcam

Bus 001 Device 003: ID 8087:07dc Intel Corp. Bluetooth wireless adapter

Bus 001 Device 002: ID 056e:4008 ELECOM Co., Ltd

Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

Hier z.B. Card Reader, Webcam, Bluetooth, eine Maus (056e:4008 ELECOM), plus Root Hubs.

Mit -t etwa:

/: Bus 02.Port 1: Dev 1, Class=root\_hub, ... (bus 2, 3.0)

  |\_\_ Port 1: Dev 2, Intel USB 3.0 Hub

    |\_\_ Port 2: Dev 3, Realtek Card Reader

...

Zeigt Topologie (Hub etc.).

### **lscpu (List CPU Info)**

**Beschreibung:** Zeigt Informationen über die CPU(s) des Systems – Anzahl Kerne, Threads, Sockets, Modellname, Frequenz, Virtualisierungsfunktionen uswfile-4fdk7rvx39azjretinarqb】 .

### **Syntax:**

`lscpu [Optionen]`

### **Optionen:**

- **-e, --extended [=list]:** Ausgabe als Tabelle mit Spalten (erweiterte Auflistung jeder logischen CPU mit Eigenschaften). Optional kann man angeben, welche Felder (z.B. CPU, Node, Socket, Core, Online, ...).
- **-p, --parse [=list]:** Ähnlich --extended aber kommagetrennt (parseable). Standardfelder: CPU,Core,Socket,Node.
- **-x, --hex:** Zeigt hex-Werte wo sinnvoll (z.B. CPU bitmasken).
- **-c, --offline / --online:** Zeigt nur offline oder nur online CPUs (bei Hotplug-Systemen).

- **-y, --physical:** Zeigt nur physische IDs (keine virtuellen IDs).
- **-J, --json:** JSON-Ausgabe (maschinenlesbar).
- **-B, --byte:** Einheit Bytes statt kB für Cache-Größen.
- **-C, --cache:** Zeigt Details zu CPU-Caches (Anzahl, Größen).

*Beispiel (gekürzt lscpu Standard):*

Architecture: x86\_64

CPU op-mode(s): 32-bit, 64-bit

Sockets: 1

Cores per socket: 4

Threads per core: 2

Model name: Intel(R) Core(TM) i7-8565U @ 1.80GHz

CPU MHz: 2112.004

Virtualization: VT-x

L1d cache: 32K

L2 cache: 256K

L3 cache: 8192K

Flags: fpu vme ... vmx ...

Zeigt also 1 Sockel, 4 Kerne, 8 Threads (4\*2), CPU-Modell und Takt etc.

### **lsmem (List Memory Blocks)**

**Beschreibung:** Zeigt Speicherblock-Informationen des Systems (insbesondere auf Systemen mit Hotpluggable Memory oder NUMA). Es listet Memory-Block-Geräte (wie in /sys/devices/system/memory) auf. Auf x86-PCs weniger relevant; auf IBM System z oder Power etc. nützlich.

### **Syntax:**

`lsmem [Optionen]`

### **Optionen:**

- **-n, --online / -o, --offline:** Zeigt nur online bzw. offline Speicherblöcke.
- **-b, --bytes:** Größen in Bytes statt lesbar.
- **-M, --mega:** Größen in MiB. (Standard lsmem gibt in bytes + kMG suffix)

- **-p, --pairs**: Ausgabe im Key:Value Format (für maschinelles Parsing).
- **-J, --json**: JSON-Format.
- **--output <list>**: Spalten auswählen (z.B. --output SIZE,NODE,STATE).
- **--help, --version** analog.

*Beispiel lsmem (auf PC ohne Hotplug memory):*

| RANGE                                 | SIZE | STATE  | REMOVABLE | BLOCK |
|---------------------------------------|------|--------|-----------|-------|
| 0x0000000000000000-0x000000007fffffff | 2G   | online | yes       | 0-15  |
| 0x0000000100000000-0x000000027fffffff | 6G   | online | yes       | 32-79 |

Zeigt 2 memory ranges (entsprechend wohl 2 DIMMs). "Removable yes" heißt vom Kernel als Hot-remove möglich (oft 'no' auf Standard-PC). "BLOCK 0-15" referenzieren Memory block device IDs.

Auf NUMA-Systemen würde Spalte NODE relevant sein, um Memory pro NUMA-Node zu sehen.

### **ps (Process Status)**

**Beschreibung:** Zeigt eine Momentaufnahme der aktuellen Prozesse des Systems. ps kann verschiedene Ausgabeformate haben (BSD-Style, UNIX-style). Häufig nutzt man ps aux oder `ps -ef` .

#### **Syntax:**

ps [Optionen]

(Komplexität: ps hat UNIX Optionen (mit - und bündelbar), BSD Optionen (ohne - und können zusammen geschrieben werden) und GNU long Optionen. Z.B. ps aux ist BSD Syntax, ps -ef ist UNIX syntax – beide gängig, liefern ähnliches.)

#### **Gängige Aufrufe:**

- **ps aux** – Listet alle Prozesse mit Details (BSD format). Spalten: USER, PID, %CPU, %MEM, VSZ, RSS, TT (TTY), STAT, START, TIME, COMMAND.
- **ps -ef** – Listet alle Prozesse (UNIX format). Spalten: UID, PID, PPID, C (CPU usage), STIME (start time), TTY, TIME (CPU time), CMD.
- **ps -eF** – wie -ef, aber "extra full", zeigt threads als separate Linien und vollständige Kommandozeilen etc.
- **ps -ejH** – zeigt Hierarchie (Prozessbaum) mit Prozess-IDs.

- **ps -eo pid,cmd,%mem,%cpu --sort=-%mem** – Zeigt benutzerdefinierte Spalten (hier pid, Befehl, Speicher- und CPU% sortiert nach Speicherverbrauch absteigend).

### **Wichtige Optionen:**

UNIX-style (präfix mit -, können kombiniert werden):

- **-e** – Alle Prozesse anzeigen (entspricht -A).
- **-f full** Format (mehr Spalten inkl. PPID, STIME, in Kombination z.B. -ef).
- **-F** – Extra full (noch mehr Infos: SZ, RSS, PSR CPU, etc.).
- **-h** – Ohne Header.
- **-H** – Hierarchie-Tree anzeigen (Indentation für Parent-Child).
- **-j** – Jobs format (PGID, SID etc. Spalten).
- **-l** – Long format (mehr spalten inkl. Flags, NI nice, LWP etc.).
- **-u <user>** – Prozesse eines bestimmten Benutzers.
- **-p <pidlist>** – Nur bestimmte PIDs.
- **-G <group>, -U <user>** – nach Gruppe/Benutzer filtern.

BSD-style (ohne -, einzeln hintereinander):

- **a** – Zeigt Prozesse von *allen Terminals* (nicht nur eigene).
- **x** – Zeigt auch prozesse *ohne Terminal* (Daemons). **ax** zusammen = alle Prozesse.
- **u** – Benutzerformat (inkl. Owner, CPU%, MEM%). => **aux** verbreitet.
- **j** – Jobs format (session, pgrp).
- **w** – breites Output (alle columns of command, use -ww to not truncate).

GNU long options (mit --):

- **--pid, --ppid, --user** etc. Alternative Filter.
- **--sort=<key>**: Sortiert Ausgabe nach Spalte(n). Z.B. --sort=-%cpu (absteigend CPU).
- **--no-headers**: Ohne Header (statt -h).
- **--forest**: ASCII-Art Bäumchen (ähnlich -H, aber macht "|\_" und Einrückungen). Kombiniert mit e.g. -ef.

### **STAT Spalte (Prozessstatus):**

Codes: R (running), S (sleep), D (uninterruptible sleep, i/o), Z (zombie), T (stopped), t

(gestoppt im Hintergrund), X (dead). Flags dahinter: < (high prio), N (low prio, nice>0), L (pages locked in mem), s (session leader), + (foreground process group in TTY).

## **sar (System Activity Reporter)**

**Beschreibung:** Teil von sysstat (wie iostat, mpstat, etc.), sammelt und berichtet historische System-Auslastungsdaten: CPU, Speicher, Paging, I/O, Netzwerk etc. sar kann *laufende* Stats anzeigen (wenn mit Interval aufgerufen), oder *vergangene Tageswerte* aus Logs (/var/log/sysstat/...) ausgeben.

### **Syntax:**

```
sar [Optionen] [Interval [Count]]
```

Ohne Parameter, sar liest per default die Tagesstatistiken (cumulated) vom laufenden Tag (falls sar cron job aktiv war). Mit Interval sar 5 5 z.B. 5x alle 5 Sekunden CPU etc.

### **Hauptoptionen (für Bereiche):**

- **-u:** CPU-Auslastung (Default, wenn keine angegeben) – zeigt %user, %nice, %system, %iowait, %steal, %idle.
- **-r:** Memory (free, used, buffers, cache, commit, etc.).
- **-b:** I/O (transfer rates, i/o per sec, etc.).
- **-B:** Paging (pgpgin/s, pgpgout/s, fault, majflt...).
- **-W:** Swapping (swap in/out).
- **-q:** Queue length (runq, i/o queue) and load average.
- **-n DEV:** Network: DEV mode – zeigt Netzwerk-Interface Stats (rxpck/s, txpck/s, rxKB/s, txKB/s, etc.). -n ALL für alle Net-Statistiken.
- **-d:** Block device I/O (similar to iostat per device).
- **-p [DEV]:** CPU per processor (if SMP). e.g. sar -P ALL 1 3 – CPU usage per core.
- **-f file:** Lese Stats aus Log-Datei (z.B. /var/log/sysstat/sarXX).
- **-s HH:MM:SS, -e HH:MM:SS:** Start/End time for reading logs (z.B. 9 bis 18 Uhr Stats).

### **Beispiel (aktuelle CPU alle 1s):**

```
$ sar 1 3
```

Linux 5.4.0 ... (8 CPUs)

```
02:00:00 PM CPU %user %nice %system %iowait %steal %idle
```

```
02:00:01 PM all 5.13 0.00 1.28 0.00 0.00 93.59
02:00:02 PM all 7.14 0.00 1.02 0.00 0.00 91.84
02:00:03 PM all 3.06 0.00 0.77 0.00 0.00 96.16
Average: all 5.11 0.00 1.02 0.00 0.00 93.87
```

### **nohup (No Hangup)**

**Beschreibung:** Lässt einen Prozess auch nach Logout weiterlaufen. nohup fängt das HUP-Signal (Hangup) ab, das normalerweise gesendet wird, wenn ein Terminal geschlossen wird. Mit nohup startet man ein Kommando so, dass es dieses Signal ignoriert und Standardausgabe ggf. in *nohup.out* umleite】.

#### **Syntax:**

```
nohup <Befehl> [Arg...] [&]
```

#### **Funktionsweise:**

- nohup sorgt dafür, dass der ausgeführte Prozess das SIGHUP ignoriert. Außerdem werden *stdout* und *stderr* umgeleitet in *nohup.out*, falls diese derzeit ans Terminal gebunden sind. (Wenn man & anhängt – also im Hintergrund – ist es üblich, die Ausgabe nicht am Terminal zu lassen).
- Üblicherweise kombiniert mit & (Hintergrund). Man loggt dann aus, und das Programm läuft weiter.

#### **Beispiele:**

- nohup long\_running.sh & – Führt *long\_running.sh* im Hintergrund aus, Ausgaben gehen in *nohup.out*. Der Befehl überlebt Logout. Die Shell gibt z.B. zurück: appending output to nohup.out.
- nohup python server.py > server.log 2>&1 & – Startet Python-Server im Hintergrund, leitet *stdout* und *stderr* in *server.log* (damit *nohup.out* nicht genutzt).

*Hinweis:* Wenn man bereits in einer Screen- oder tmux-Session ist, braucht man kein nohup; auch systemd-run oder Cron-Jobs sind Alternativen. Aber nohup ist ein schnelles Mittel.

### **mkswap/swapon/swapoff (Swap einrichten und verwalten)**

- **mkswap** – Initialisiert eine Partition/Datei als Swapbereich. Syntax: mkswap [Optionen] <Gerät/datei>. Wichtige Option: -L <Label> (Swap-Label setzen), -U <UUID>. Beispiel: mkswap /dev/sda3.
- **swapon** – Aktiviert einen Swapbereich für Nutzung durch das System】. Nach mkswap muss Partition via swapon <Device> eingebunden werden (oder in

/etc/fstab eingetragen mit swap Typ). Option -a aktiviert alle in fstab markierten Swaps. --show zeigt aktuellen Swap an (Äquivalent cat /proc/swaps).

- **swapoff** – Deaktiviert Swap. Syntax: swapoff <Device> oder swapoff -a für alle.

### Beispiele:

```
mkswap -L "SWAP" /dev/sda3
```

Setting up swap space version 1, size = 8 GiB (some hex UUID)

```
swapon /dev/sda3
```

```
swapon --show
```

| NAME | TYPE | SIZE | USED | PRIO |
|------|------|------|------|------|
|------|------|------|------|------|

|           |           |    |    |    |
|-----------|-----------|----|----|----|
| /dev/sda3 | partition | 8G | 0B | -2 |
|-----------|-----------|----|----|----|

swapoff -a würde alle Swapbereiche aus dem Betrieb nehmen (z.B. vor Partition-Vergrößerung oder Kernel-Dump etc.).

### free (Speicherübersicht)

**Beschreibung:** Zeigt eine Übersicht über den aktuellen Speicherverbrauch (RAM und Swap) im System. Es summiert total, used, free, and cached/buffered memory.

### Syntax:

```
free [Optionen]
```

### Optionen:

- **-h**: Human Readable (z.B. MiB/GiB).
- **-m, -g, -k**: Ausgabe in MiB, GiB, KiB.
- **-b**: in Bytes.
- **-t**: Fügt eine "Total" Zeile (RAM+Swap) hinzu.
- **-s N**: Aktualisiert alle N Sekunden (kontinuierliche Ausgabe wie watch).
- **-c N**: Wenn mit -s genutzt, stop nach N Aktualisierungen.
- **--si**: Benutzt 1000er statt 1024er Basis (KB = 1000 Bytes etc.).

### Beispiel free -h:

|       | total | used  | free  | shared | buff/cache | available |
|-------|-------|-------|-------|--------|------------|-----------|
| Mem:  | 15Gi  | 5.2Gi | 2.3Gi | 1.1Gi  | 7.4Gi      | 8.8Gi     |
| Swap: | 2.0Gi | 0.0Gi | 2.0Gi |        |            |           |

available ist eine Kernelschätzung, wieviel noch für neue Prozesse verfügbar, unter Berücksichtigung von caches, etc.

### **vmstat (Virtual Memory and system stats)**

**Beschreibung:** Zeigt verschiedene Systemmetriken: Speicher, Auslagerung, CPU, Interrupts, Context Switches etc. im zeitlichen Verlauf. vmstat ohne Parameter gibt Durchschnitt seit Boot; mit Intervall zeigt periodische Werte (Differenzen pro Intervall).

#### **Syntax:**

```
vmstat [Optionen] [Interval [Count]]
```

#### **Wichtige Spalten (Standardausgabe):**

- **procs:** *r* (Anzahl laufender Prozesse im Runqueue), *b* (blockierte Prozesse).
- **memory:** *swpd* (Swap genutzt), *free* (freier RAM), *buff* (Puffer), *cache* (Page Cache).
- **swap:** *si* (Swap-In KB/s), *so* (Swap-Out KB/s).
- **io:** *bi* (Block In, Blöcke gelesen von Disk/s), *bo* (Block Out, geschrieben/s).
- **system:** *in* (Interrupts/s), *cs* (Context Switches/s).
- **cpu:** *us* (% CPU in Userzeit), *sy* (% Systemzeit), *id* (% Idle), *wa* (% IO-Wait), *st* (% Steal bei virt.).

#### **Optionen:**

- **-S <unit>:** Werte in k (1000) oder K (1024) etc.
- **-a:** Zeigt zusätzlich Aktivität der *aktiven/inaktiven* Memory Listen (sin, sout).
- **-d:** Zeigt Disk-Statistiken (like iostat) statt VM stats.
- **-p <dev>:** Zeigt Partition-Statistiken (reads/writes tot und sek) für Device.
- **-s:** Summiert memory stats (eine Liste aller relevanten Zähler, analog /proc/meminfo).
- **-m, -t:** Variation der Ausgabe (meist uninteressant, -m memory in MB etc).

#### **Beispiel (alle 2s):**

```
$ vmstat 2 3
```

```
procs -----memory----- swap-- -----io---- system-- -----cpu-----
r b free buff cache si so bi bo in cs us sy id wa st
1 0 230000 120000 760000 0 0 5 10 150 300 5 191 3 0
```

```
0 0 229000 120000 760500 0 0 0 20 140 280 4 1 94 1 0
```

```
1 0 228500 120000 761000 0 0 0 30 160 320 6 1 92 1 0
```

Erste Zeile nach Header ist seit Boot (ohne parameter, oft uninteressant); weitere sind Echtzeit pro Intervall.

### **modinfo (Module Information)**

**Beschreibung:** Zeigt Informationen über ein Linux-Kernelmodul (Loadable Kernel Module) – Version, Beschreibung, Lizenz, Abhängigkeiten, Parameter etc】 .

#### **Syntax:**

```
modinfo [Optionen] <Modulname oder Modul.ko>
```

#### **Optionen:**

- **-a, --author:** Nur Autor ausgeben.
- **-d, --description:** Nur Beschreibung.
- **-n, --filename:** Zeigt Pfad der Moduldatei im Filesystem.
- **-p, --parameters:** Listet die Parameter und ihre Typen/Beschreibungen.
- **-F <field>:** Gibt nur das spezifizierte Feld aus (alternativ obige Kurzoptionen). Mögliche Felder: author, description, license, parm, alias, etc.
- **-k <kernelversion>:** Fragt Infos für Kernel-Version (sucht in /lib/modules/<version>) statt laufendem Kernel.

**Beispiel:** modinfo e1000e (Intel NIC Treiber) gibt etwa:

```
filename: /lib/modules/5.4.0/kernel/drivers/net/ethernet/intel/e1000e/e1000e.ko
```

```
version: 3.2.6-k
```

```
license: GPL
```

```
description: Intel(R) PRO/1000 Network Driver
```

```
author: Intel Corporation, <e1000-devel@lists.sourceforge.net>
```

```
...
```

```
srcversion: 5C6D1A8....
```

```
alias: pci:v00008086d000010D3sv*sd*bc*sc*j*
```

```
...
```

```
parm: IntMode:Interrupt Mode (0=Auto, 1=MSI, 2=MSI-X) (int)
```

```
parm: InterruptThrottleRate:... (array of int)
```

Man sieht Dateipfad, Version, Parameter etc.

### **insmod/rmmod/modprobe (Kernelmodule laden/entladen)**

- **insmod <pfad/modul.ko> [args]**: Lädt ein Modul in den Kernel. Nutzt genau Pfad; löst keine Abhängigkeiten auf.
- **rmmod <modulname>**: Entfernt ein geladenes Modul (wenn kein anderer in Benutzung). Option -f zum Erzwingen (nur falls Kernel CONFIG\_MODULE\_FORCE\_UNLOAD).
- **modprobe [Options] <modulname> [args]**: Empfehlenswerter Befehl zum Laden eines Moduls und seiner Abhängigkeiten. Es durchsucht /lib/modules/<kernel>/modules.dep nach Dependencies. Auch zum Entfernen via modprobe -r <modul> (lädt auch Abhängigkeiten aus, falls unbenutzt).
  - Optionen: --dry-run oder -n testet nur, -v verbose, -q quiet, alias handling etc. modprobe Konfig steht in /etc/modprobe.d.
  - Bsp: modprobe e1000e (lädt Intel NIC modul + ggf. benötigte mii). modprobe -r e1000e entfernt es.

### **depmod (Dependency modules)**

**Beschreibung:** Generiert die Modulabhängigkeitsdatei *modules.dep* basierend auf Modulen in /lib/modules/<version>. Wird meist beim Kernel-Install aufgerufen. Kann man manuell ausführen nach Hinzufügen eigener Module. Syntax: depmod -a (für alle).

#### **Optionen:**

- -a, --all (default): alle Kernel-Versionen verarbeiten.
- -A, --quick: nur verändern, wenn modules.dep älter als Module-Dateien (incremental update).
- -n: Ausgabe nach stdout statt in modules.dep schreiben (Test).
- -v: verbose.

### **sysctl (Query/Set Kernel Parameters)**

**Beschreibung:** Anzeige oder Änderung von Kernel-Laufzeitparametern (in /proc/sys). sysctl ermöglicht bequemen Zugriff auf z.B. Netzwerktruning, VM, FS Settings zur Laufzeit.

#### **Syntax:**

sysctl [Optionen] [<Name>=<Wert> ...] or sysctl -p

#### **Optionen:**

- Ohne =: liest Parameter. Z.B. sysctl net.ipv4.ip\_forward -> net.ipv4.ip\_forward = 0.

- Mit name=value: setzt Parameter. Z.B. sysctl net.ipv4.ip\_forward=1 aktiviert IP-Forwarding (sofort).
- **-w**: explizit Schreibmodus (kann man weglassen, war historisch).
- **-p [Datei]**: Lädt Einstellungen aus Datei (Default /etc/sysctl.conf) und wendet a】 . Man ruft sysctl -p typ. beim Boot (init-Skripte) oder manuell nach Bearbeiten der config.
- **-a**: Alle Parameter anzeigen (riesige Liste). sysctl -a | grep ipc etc.
- **-A, --pattern <regex>**: Filtert mit Regex (statt manuell grep).
- **-e**: Ignoriert unbekannte Keys (useful mit -p, sonst bricht ab).
- **-N**: Nur Namen, keine Werte anzeigen (list keys).
- **-q**: Quiet (unterdrückt Bestätigungsmeldungen).

Parameter-Keys entsprechen /proc/sys Pfaden, z.B. /proc/sys/net/ipv4/ip\_forward -> key net.ipv4.ip\_forward.

### **Beispiele:**

- sysctl vm.swappiness -> zeigt swappiness (z.B. vm.swappiness = 60).
- sysctl -w vm.swappiness=10 -> setzt auf 10 (weniger Swapneigung).
- sysctl -p /etc/sysctl.d/99-sysctl.conf -> läd Parameter aus der angegebenen Datei.

## **dmesg (Dump kernel message buffer)**

**Beschreibung:** Zeigt die Nachrichten des Kernel-Ringspeichers (Kernellog) a】 . Hier stehen Boot-Meldungen und Hardware-Events (z.B. USB eingesteckt, Treiber Ausgabe). dmesg ohne Optionen druckt alle aktuellen Kernelmeldungen.

### **Syntax:**

dmesg [Optionen]

### **Optionen:**

- **-T**: Zeigt menschenlesbare Zeitstempel (Jahr-Monat-Tag Stunde:Minute:Sekunde) statt der raw Sekunden seit Boo】 . (Achtung: Kann ungenau sein, da Kernel keine Zeitzone kennt; -T rechnet anhand aktueller TZ).
- **-l <level>, --level**: Filtert Nachrichten bestimmter Log-Levels. Level: emerg, alert, crit, err, warn, notice, info, debug. Z.B. dmesg -l err,warn nur Fehler und Warnungen.

- **-k, --kernel:** Zeigt nur Kernel-Messages (Standard). **-u, --userspace:** Nur Benutzerraum (Logeinträge vom printk\_user). Kaum genutzt, da meiste aus Kernel.
- **-H, --human:** **Pager-Modus** (lässt scrollen, farbig Highlighting). Interaktiv (Space/PgUp etc.).
- **-w, --follow:** "Tail -f" Modus für dmesg – neue Kernelmeldungen laufend anzeigen.
- **-s N:** Buffer-Größe (Bytes) die gelesen wird (Default könnte begrenzt sein). dmesg -s 1000000.

### **Beispiel (gekürzt):**

```
[0.000000] Linux version 5.4.0-80-generic ...
[0.345678] PCIe Bus 0000:00: root hub ...
[1.234567] usb 1-2: new high-speed USB device number 3 ...
[1.345678] ata1: SATA link up ...
[10.567890] Bluetooth: RFCOMM ver 1.11
```

dmesg -T zeigt z.B. [Mon Oct 11 14:23:45 2021] ....

Typisch ruft man dmesg | grep -i error um nach Kernel-Fehlern zu suchen, oder dmesg -w in einem Terminal um z.B. neue USB/Hardware Events sofort zu sehen.

## **Dienstverwaltung und Systemstart**

### **systemctl (Control systemd)**

**Beschreibung:** Haupt-Werkzeug zur Steuerung des *systemd*-Initsystems (heutzutage auf vielen Linux-Distributionen). systemctl verwaltet **Services** (Start/Stop/Enable), **Daemon-Konfiguration, System-Zustände** (Reboot/Shutdown), **Unit**-Status und Logs, etc】 .

### **Syntax:**

systemctl [Optionen] <Sub-Befehl> [Unit/...]

Units können Services (.service), Mounts (.mount), Timer (.timer), Sockets (.socket), etc. sein. Oft kann man den Suffix weglassen (z.B. "ssh" wird als "ssh.service" erkannt).

### **Wichtige Sub-Befehle (Service Management):**

- **start <service>, stop <service>** – Startet oder stoppt einen Dienst \*sofortfile-4fdk7rvx39azjretinarqb】 . Bsp: systemctl start apache2.
- **restart <service>** – Stop + Start (auch wenn nicht lief, wird gestartet).

- **reload <service>** – Lädt Konfiguration neu (sendet i.d.R. SIGHUP). Muss vom Service unterstützt werden (in Unit als ReloadSignal oder ExecReload definiert). Bsp: systemctl reload nginx.
- **status <service>** – Zeigt Status des Dienstes: Läuft er? Letzte Startmeldung, ggf. Auszug aus Journal-Log】 .
- **enable <service>** – Aktiviert den Dienst beim Boot (erstellt Symlinks in passenden WantedBy-Verzeichnissen).
- **disable <service>** – Deaktiviert Autostart.
- **is-enabled, is-active** – Prüfen ob Unit aktiviert ist / gerade aktiv ist (exit 0/1).
- **mask <service>** – "Maskiert" den Service, d.h. verhindert Start vollkommen (setzt Link auf /dev/null). *Unmask* zum Aufheben.
- **daemon-reload** – Falls man eine Unit-Datei geändert oder neu hinzugefügt hat, diese neu einlesen (Konfig neu laden, ohne Dienste neu zu starten).
- **kill <service>** – Sendet ein Signal an alle Prozesse der Service-Unit (z.B. systemctl kill -s SIGUSR1 myapp.service).

### System Commands:

- **list-units** – Listet alle geladenen Einheiten (nur aktive standardmäßig). --all um Inaktive auch zu sehen, --type=service filtern auf Services, etc.
- **list-unit-files** – Listet Unit-Dateien und ob sie enabled/disabled/etc. sind (Installationszustand).
- **list-dependencies <unit>** – Zeigt Abhängigkeiten (Requires/Wants) Baum für Unit. --reverse zeigt welche Units diese Unit benötigen.
- **show <unit>** – Gibt alle Properties von Unit aus (oder subset via -p).
- **cat <unit>** – Zeigt den Inhalt der Unit-Datei (und Drop-Ins).
- **edit <unit>** – Öffnet Editor für Drop-In Snippet, um diese Unit zu überschreiben/erweitern.
- **preset <service>** – Setzt enable/disable gemäß Distribution-Vorgabe (in /usr/lib/systemd/system-preset/).

### Targets (Runlevels):

- **isolate <target>** – Wechselt in ein target (beendet andere). Z.B. systemctl isolate rescue.target (Einzeluser-Modus ähnlich runlevel 1).
- **set-default <target>** – Legt Default Boot-Target fest (z.B. graphical.target vs multi-user.target).

- **get-default** – zeigt aktuelles Defaulttarget.

#### **Power Management:**

- **reboot, poweroff, halt** – führen Neustart bzw. Herunterfahren aus (rufen intern entsprechende target isolation).
- **suspend, hibernate, hybrid-sleep** – starten Energiesparmodus.

#### **Journal/Logging Integration:**

- **status** (wie oben) zeigt bereits Log-Auszug.
- **journalctl** (separater Befehl, nicht systemctl subcommand) – zum Betrachten der systemd Logs. Z.B. journalctl -u <service> zeigt Logs zu einem Service, -f zum folgen, --since "2021-10-01" ab bestimmtem Datum etc.

#### **Beispiele:**

- systemctl enable --now nginx – Aktiviert **und** startet nginx sofort (Option --now führt gleich start nach enable aus).
- systemctl disable apache2.service – Entfernt Apache2 aus Autostart.
- systemctl restart networking – Dienst neu starten (bei Netz aufpassen).
- systemctl list-units --failed – Zeigt nur fehlgeschlagene Units.
- systemctl mask firewalld; systemctl stop firewalld – Verhindert Start des Firewalld dauerhaft und stoppt ihn sofort (z.B. falls man anderen Firewall nutzt).

#### **service (SysVinit Compatibility)**

**Beschreibung:** Der Befehl service war bei SysVinit/Upstart gebräuchlich, um Start/Stop-Skripte in /etc/init.d aufzurufen. Auf systemd-Systemen ruft service intern systemctl auf. Man kann aber service <name> start/stop/status noch benutzen! `service ssh status`

#### **Syntax:**

service <ServiceName> [start|stop|restart|status|...]

**Beispiel:** service ssh status ruft systemctl status ssh.service auf.

(Auf SysVinit ohne systemd würde es das init.d Script ausführen.)

#### **chkconfig (Runlevel Konfiguration - SysV kompatibel)**

**Beschreibung:** Ein Werkzeug zum Verwalten der Start-/Stopp-Links von SysV-Init-Skripten über verschiedene Runlevels (0-6). Auf systemd-Systemen gegenstandslos (meist vorhanden für Kompatibilität, was auf systemctl enable/disable mappt).

## **Wichtigste Befehle:**

- `chkconfig --list [name]` – Listet alle Dienste und ob sie in runlevel 0-6 on/off sind  
file-4fdk7rvx39azjretinarqb】 . Mit Name nur diesen.
- `chkconfig name on` (bzw. off) – Schaltet Service in allen entsprechenden runlevels an oder aus (Standard runlevels 2-5 on für on).
- `chkconfig name --level 35 on` – Feinsteuerung: aktiviert Dienst in Runlevel 3 und 5.
- `chkconfig name reset` – Setzt Standard (wie im Init-Skript Header vorgesehen).

\*(Auf systemd delegiert chkconfig intern an systemctl enable/disable. Für Nostalgie, wissen was es tat.)

## **Zeitplanung (Cron und At)**

### **crontab (Cron Table Management)**

**Beschreibung:** Planen wiederkehrender Aufgaben. crontab bearbeitet die Aufgabenliste des aktuellen Benutzers (oder eines anderen mit -u) im Cron-Daemon. Jede Zeile in crontab -e hat Format "*Min Stunde Tag Monat Wochentag Befehl*". Der Cron-Dienst prüft pro Minute die Tabellen und führt fällige Kommandos auf file-4fdk7rvx39azjretinarqb】 .

#### **Befehle:**

- `crontab -e` – Editiert die Crontab des aktuellen Benutzers (öffnet \$EDITOR). Beim Speichern wird die neue Tabelle dem Cron-Daemon übergeben file-4fdk7rvx39azjretinarqb】 .
- `crontab -l` – Listet die aktuelle Crontab auf file-4fdk7rvx39azjretinarqb】 .
- `crontab -r` – Entfernt (löscht) die aktuelle Crontab komplett. (Vorsicht, kein Undo).
- `crontab -u username -e/-l/-r` – Als root: Bearbeitet/liest/löscht die Crontab eines anderen Benutzers.

#### **Cron Format Erinnerung:**

# Minute(0-59) Hour(0-23) Day(1-31) Month(1-12) Weekday(0-7) Command

0 2 \* \* \* /usr/bin/backup.sh # täglich 02:00

\*/5 9-17 \* \* 1-5 echo "Working..." # Werktags 9-17 Uhr alle 5 min

\*=jeder Wert, Komma getrennte Listen, Intervalle mit -, Step mit /.

Spezial @-Strings: @reboot, @hourly, @daily, @yearly etc.

**Anacron:** Cron führt Jobs aus, wann Maschine an ist. Für nicht ständig laufende Rechner nutzt man anacron (Tagesjobs nachholen).

## **at (One-time scheduled task)**

**Beschreibung:** Legt einmalige Ausführungen von Befehlen zu einer bestimmten Zeit in der Zukunft fest [4fdk7rvx39azjretinarqb]. at nimmt eine Zeitangabe entgegen und liest dann von STDIN die Befehle, die ausgeführt werden sollen (oder mit -f aus einer Datei). Der atd Daemon muss laufen.

### **Verwendung:**

```
echo "/path/to/script.sh" | at 02:30 # führt Script heute Nacht 2:30 aus
```

Man ruft at <Zeit> auf, landet in einer Stdin-Eingabe (end mit Ctrl+D) oder speist via echo/<<EOF>>.

**Zeitformat:** Z.B. at 14:00 (heute 14 Uhr, falls schon vorbei dann morgen), at 11pm, at midnight, at now + 5 days, at 08:00 12.12.2025 (8:00 am 12 Dec 2025). Kann englische Wörter: tomorrow, noon, teatime(16:00) etc.

### **Befehle:**

- atq – Zeigt anstehende at-Jobs (Queue) des aktuellen Userfiles [4fdk7rvx39azjretinarqb]. (Listet Job-IDs und geplante Zeit). Root sieht alle oder atq -q <Queue> für bestimmte Batch-Queues.
- atrm <JobID> – Entfernt einen geplanten Job aus der Queue. (JobID sieht man in atq).
- batch – Alias zu at -q b now (führt Jobs aus, wenn Load Average niedrig ist). "Batch" schiebt den Auftrag in Batch-Queue, läuft sobald System nicht ausgelastet (LA<0.8 default).

### **Beispiel:**

```
$ at now + 1 minute
```

```
warning: commands will be executed using /bin/sh
```

```
at> echo "Hello after 1 min" >> /tmp/test.txt
```

```
at> <EOT>
```

```
job 5 at Fri Oct 15 16:41:00 2021
```

Dann innerhalb einer Minute atq zeigt e.g. 5 Fri Oct 15 16:41:00 2021 a user. Nach Ausführung landet Text in Datei.

(Zugriff: /etc/at.allow /etc/at.deny definieren wer at nutzen darf – analog cron.allow/deny.)

### **Drucken**

## **lpr (Line Print Request)**

**Beschreibung:** Sendet eine Datei (oder STDIN) an den Drucker (Print-Job einreihen). lpr ist Teil des BSD-Linux-Drucksystems (CUPS stellt Kompatibilität bereit). Alternative: lp (System V style).

### Syntax:

lpr [Optionen] [Dateien...]

### Wichtige Optionen:

- **-P <Druckernname>**: Wählt einen bestimmten Drucker aus (sonst Standarddruckerfile-4fdk7rvx39azjretinarqb】). Druckernamen findet man via lpstat -a oder CUPS Webinterface.
- **-# <Num>**: Anzahl Kopien drucken (z.B. -#3 dreifach).
- **-h**: Unterdrückt Banner-Seite (header page) falls aktiviert.
- **-r**: Löscht die Datei(en) nach dem Versenden an den Spool (rarely used).
- **-m**: Sendet E-Mail Benachrichtigung nach Druck (sofern eingerichtet).
- **-o <Option>=<Wert>**: Drucker-spezifische Optionen übergeben (z.B. -o landscape für Querformat, oder -o media=A4 oder -o sides=two-sided-long-edge für Duplex). Diese Optionen sind je nach Drucker Treiber definierbar (CUPS options).
- **-C <Klasse>, -J <Jobname>, -T <Titel>**: Legen Klassenname, Jobname, Titel für Banner etc. fest.
- **--help** (GNU lpr) usw.

### Beispiele:

- lpr report.pdf – Schickt *report.pdf* an Standarddrucker.
- lpr -P HP\_LaserJet -#2 invoice.txt – Druckt *invoice.txt* 2-fach auf Drucker "HP\_LaserJet".
- echo "Test Page" | lpr – Druckt STDIN (den Text "Test Page").
- lpr -o sides=two-sided-long-edge thesis.pdf – Duplexdruck beidseitig.

(Verwandt: *lpq* zeigt Druckerqueue, *lprm* löscht Jobs aus der Queue.)

## Lokale Systemkonfiguration

### date (Anzeige/Setzen Datum/Uhrzeit)

**Beschreibung:** Zeigt das aktuelle Datum und die Uhrzeit an, oder stellt sie ein. Auch Formatierung in beliebigem Muster ist möglich.

### Syntax:

date [Optionen] [+Format]

date -s "STRING" (setzen)

### Optionen / Verwendung:

- **Ausgabe formatieren:** date "+FORMAT" – Mit Format-Pattern ähnlich strftime:  
%Y (Jahr), %m Monat(01), %b Monat (Kurzname), %d Tag, %H Stunde (24h), %I (12h), %M Minute, %S Sek, %A Wochentag, %a (kurz), %Z Zeitzone, %z offset, %T = %H:%M:%S, %F = %Y-%m-%d, etc. Z.B. date "+%d.%m.%Y %H:%M:%S" -> "05.10.2025 14:30:00".
- **Systemzeit stellen:** date -s "2025-10-05 14:30:00" – Setzt Datum. Braucht Root. (Alternative: date --set="...").
- **Hardware-Uhr stellen:** In Linux erfolgt date auf Systemzeit (in Kernel, RTC separiert). Um auch BIOS/RTC Uhr zu setzen, hwclock --systohc (setze Hardware Clock nach System Clock). Umgekehrt hwclock --hctosys.
- **Universale Zeit:** date -u zeigt in UTC. date -u -s "2025-10-05 12:30:00" setzt in UTC.
- **RFC-3339 Format:** date --rfc-3339=ns etc. Standard ISO.
- **Unix Timestamp:** date +%s gibt Sekunden seit 1.1.1970 (UTC).

### Beispiele:

- date -> "Tue Oct 5 14:30:00 CEST 2025".
- date +"%Y-%m-%d\_%H-%M-%S" -> "2025-10-05\_14-30-00".
- sudo date -s "@\$(( \$(date +%s) + 3600 ))" -> Systemzeit um eine Stunde vorstellen (Beispiel mit timestamp math).

### timedatectl (Zeiteinstellung via systemd)

**Beschreibung:** Befehl von systemd zum Anzeigen und Ändern der Systemzeit, Hardware-Uhr, Zeitzone und NTP-Synchronisatiofile-4fdk7rvx39azjretinarqb】 .

### Verwendung:

- **timedatectl** (ohne args): Zeigt aktuellen Status: Lokale Zeit, UTC, RTC, Zeitzone, ob NTP aktiv ist etcfile-4fdk7rvx39azjretinarqb】 .
- **timedatectl set-time "YYYY-MM-DD HH:MM:SS"** – Setzt Systemzeit (wenn kein NTP aktiv).
- **timedatectl set-timezone <Zone>** – Ändert Zeitzone. Z.B. timedatectl set-timezone Europe/Berlin. (Verfügbare Zonen unter /usr/share/zoneinfo/, oder timedatectl list-timezones).

- **timedatectl set-ntp true/false** – Aktiviert bzw. deaktiviert die automatische Zeitsynchronisation (via systemd-timesyncd oder anderes NTP). Bei true startet es den Dienst falls vorhanden.
- **timedatectl set-local-rtc 1/0** – Legt fest, ob die Hardware-Uhr (RTC) in **lokaler Zeit** läuft (1) oder in UTC (0, Standard). Windows dual-boot oft RTC=lokal (dann 1), sonst belässt auf 0=UTC um Probleme zu vermeiden.

### **Beispiele:**

\$ timedatectl

Local time: Tue 2025-10-05 14:30:00 CEST

Universal time: Tue 2025-10-05 12:30:00 UTC

RTC time: Tue 2025-10-05 12:30:00

Time zone: Europe/Berlin (CEST, +0200)

System clock synchronized: yes

NTP service: active

RTC in local TZ: no

timedatectl set-time "2025-12-24 18:00:00" stellt Heiligabend 18:00. timedatectl set-timezone UTC wechselt Zeitzone auf UTC.

### **localectl (Locale and Keyboard Layout)**

**Beschreibung:** Verwaltet System-Locale (Sprache, Gebietsschema) und Tastaturlayout-Einstellungen in systemd-basierten Systemefile-4fdk7rvx39azjretinarqb】 . Es kann die Variablen in /etc/locale.conf sowie Konsolen-Tastaturlayout in /etc/vconsole.conf setzen. (Für X/Wayland Tastatur greift localectl in /etc/X11/xorg.conf.d/ oder mittels setxkbmap separat).

### **Verwendung:**

- **localectl** (ohne args): Zeigt aktuelle Locale(s) und Keymap. Z.B.:
  - System Locale: LANG=de\_DE.UTF-8
  - VC Keymap: de
  - X11 Layout: de
  - X11 Model: pc105 etc.
- **localectl list-locales** – Listet alle verfügbaren Locale, die generiert sind (die in /usr/lib/locale oder /etc/locale.gen markiert).
- **localectl set-locale LANG=<locale> [VAR=locale...]** – Setzt System-Locale. Z.B. localectl set-locale LANG=en\_US.UTF-8. Optional mehrere, z.B. LC\_TIME=en\_GB.UTF-8. Wirksam nach Neustart oder neu login shells.

- **localectl list-keymaps** – Listet verfügbare Console-Keymaps (z.B. us, de, ...).
- **localectl set-keymap <map> [<toggle>]** – Setzt Tastaturlayout für Konsole (vconsole) und ggf. X11, falls X11 Layout nicht getrennt definiert. Z.B. localectl set-keymap de.
- **localectl set-x11-keymap <layout> [model] [variant] [option]** – Spezifisch X11 (Graphical) Layout/Modell setzen. Z.B. localectl set-x11-keymap us pc104 "" terminate:ctrl\_alt\_bksp.

*Beispiel:*

localectl set-locale LANG=fr\_FR.UTF-8 (Systemsprache auf Französisch).

localectl set-keymap us (Konsole auf US-Layout).

localectl set-x11-keymap de pc105 nodeadkeys (Grafische Tastatur auf Deutsch mit Variante "nodeadkeys").

(Die Konfigurationen landen in /etc/locale.conf, /etc/vconsole.conf, /etc/X11/xorg.conf.d/00-keyboard.conf entsprechend.)

## Netzwerk-Konfiguration

### ip (Netlink Interface Config)

**Beschreibung:** Moderner Befehl zur Verwaltung von Netzwerkinterfaces, IP-Adressen, Routing, Tunnels etc. (Nachfolger von ifconfig/route). ip hat Unterkommandos für verschiedene Objekte: **link** (Netzwerkinterface), **addr** (IP-Adressen), **route**, **neigh** (ARP/NDP Nachbarn), **rule** (Policy Routing), **maddr** (Multicast), **addrlabel**, **netns** (Net Namespaces), **link set** (Interface-Settings).

#### Syntax (Grundstruktur):

ip [Options] <Object> <Subcommand> [Arguments...]

z.B. ip link show, ip addr add, ip route list, etc.

#### Häufige Befehle/Beispiele:

- **Interface anzeigen:** ip link show [dev <iface>] – Listet alle Netzwerkinterfaces (oder ein spezifisches). Ausgabe enthält Index, Name, MAC, MTU, Flags (UP,LOWER\_UP=Link OK, MULTICAST), etc.
- **Interface aktivieren/deaktivieren:** ip link set <iface> up / down – entspricht `ifconfig iface up/downfile-4fdk7rvx39azjretinarqb` .
- **Adresse zuweisen:** ip addr add <IP>/<Prefix> [broadcast +] dev <iface> – z.B. ip addr add 192.168.1.100/24 dev eth0. broadcast + lässt broadcast aus Prefix berechnen (optional).
- **Adresse löschen:** ip addr del <IP>/<Prefix> dev <iface>.

- **Adressen anzeigen:** ip addr show [dev <iface>] – Listet IPv4/IPv6 Adressen auf  
Interfacefile-4fdk7rvx39azjretinarqb】 . Zeigt auch Scope (global/link), flags (tentative, dynamic).
- **Standardroute setzen:** ip route add default via <Gateway-IP> [dev <iface>] – z.B. ip r add default via 192.168.1.1 dev eth0.
- **Statische Route setzen:** ip route add <Netz>/<Präfix> via <nächsterHop> dev <iface>.
- **Routen auflisten:** ip route show (oder ip r).
- **ARP Cache:** ip neigh show – zeigt ARP/ND-Cache (Neighbors).
- **ARP Eintrag fix setzen:** ip neigh add <IP> lladdr <MAC> dev <iface> nud permanent.
- **Secondary IPs / Aliase:** Einfach mehrfach ip addr add auf selbes dev ausführen (oder use Label).
- **MTU ändern:** ip link set dev <iface> mtu 1400.
- **MAC ändern:** ip link set dev <iface> address <NewMAC> (Down interface first).
- **Promisc Mode:** ip link set dev <iface> promisc on.
- **Routing Policy (advanced):** ip rule und ip route add table <n> for source-based routing etc.
- **Network Namespace mgmt:** ip netns (add, del, exec, etc – advanced).

**Options global:** -s (stats), -d (details), -br (brief output), -c (color output). Z.B. ip -br addr zeigt Interfaces in einzeiligem Kurzformat.

### Beispiele:

- ip -br link ->

```
lo UNKNOWN 127.0.0.1/8 ::1/128
eth0 UP 192.168.1.10/24 fe80::.../64
```

- ip addr add 10.0.0.1/16 dev eth1 (fügt IP).
- ip route show (z.B. default via 192.168.1.1 dev eth0 proto dhcp metric 100 etc.).
- ip neigh show (z.B. 192.168.1.1 dev eth0 lladdr aa:bb:cc:dd:ee:ff REACHABLE).

(*ifconfig (net-tools) alternative: ip ist ausführlicher und bevorzugt.*)

### ifconfig (Interface Config – alt)

**Beschreibung:** Älteres Werkzeug aus net-tools (weitgehend ersetzt durch ip). ifconfig kann Interfaces anzeigen, IP setzen, MTU ändern, promiscuous Mode etc., aber unterstützt z.B. keine IPv6 out-of-the-box je nach Version.

### Syntax:

ifconfig [Interface] [Optionen]

### Beispiele:

- ifconfig – zeigt alle aktiven (UP) Interfaces mit IPv4 etc.
- ifconfig -a – zeigt *alle*, inkl. down Interfaces.
- ifconfig eth0 192.168.0.5 netmask 255.255.255.0 broadcast 192.168.0.255 – setzt IP, Maske, Bcast.
- ifconfig eth0 up / down – Interface an/aus.
- ifconfig eth0 mtu 1400 – MTU ändern.
- ifconfig eth0 promisc – Promiscuous Mode an. -promisc aus.
- ifconfig eth0 hw ether 00:11:22:33:44:55 – MAC-Adresse setzen.

(*ifconfig ist nicht mehr standard installiert in neueren Distros; ip commands stattdessen nutzen.*)

### iwconfig (Wireless Config – alt)

**Beschreibung:** Ähnlich ifconfig, aber für Wireless LAN Schnittstellen (Legacy tool aus wireless-tools). Zeigt/ändert ESSID, Mode, Frequenz, Schlüssel etc. Heutzutage ersetzt durch iw (nl80211), oder NetworkManager.

### Beispiele (falls vorhanden):

- iwconfig wlan0 – zeigt ESSID, Mode (Managed/Ad-Hoc/Master), Channel/Freq, Bitrate, Tx-Power, Signal Level, Encryption key on/off, etc.
- iwconfig wlan0 essid "WLAN-Name" – mit offener SSID verbinden (sofern AP ohne Sicherheit).
- iwconfig wlan0 key s:password – setzt WEP Key (veraltet).
- iwconfig wlan0 mode Monitor – Interface in Monitor Mode (für Sniffing).

(Iwconfig kann kein WPA2; da nutzt man wpa\_supplicant oder NM, etc.)

### nmcli (NetworkManager CLI)

**Beschreibung:** Kommandozeilen-Interface zum NetworkManager-Dienst, um Netzwerkkonfiguration (insb. Desktop-Umfeld) zu betrachten und steuerfile-

4fdk7rvx39azjretinarqb】 . Kann Verbindungen (profiles) verwalten, aktivieren, WLAN scannen, etc.

### **Verwendung (Struktur):**

nmcli [general|networking|radio|connection|device] <aktion> [Parameter]

### **Beispiele:**

- **Status anzeigen:** nmcli general status – Gesamter NM-Status. nmcli connection show – listet gespeicherte Verbindungen (Profiles). nmcli device status – zeigt Geräte und ob connected und welches Profil.
- **WLAN scannen:** nmcli device wifi list – listet verfügbare WLANs.
- **Verbinden WLAN:** nmcli device wifi connect "SSID" password "passwort" – erstellt ggf. Connection-Profil und verbindet.
- **Verbindung aktivieren/deaktivieren:** nmcli connection up <Name> / down <Name>. Oder nmcli device disconnect <iface> (z.B. nmcli device disconnect wlan0).
- **Neue Verbindung anlegen:** nmcli connection add type ethernet ifname eth0 con-name "Wired Manual" ip4 192.168.5.5/24 gw4 192.168.5.1.
- **IPv4 ändern laufend:** nmcli connection modify "Wired connection 1" ipv4.addresses 192.168.1.50/24 ipv4.gateway 192.168.1.1 ipv4.method manual und dann up.
- **Ethernet an/aus:** nmcli networking off (setzt NM generell offline) oder nmcli radio wifi off (schaltet nur WLAN-Funk aus).

### **Beobachten:**

- nmcli connection show --active – nur aktive Conns.
- nmcli -p device status – pretty (ncurses-like) output.

### **ethtool (NIC Driver Tool)**

**Beschreibung:** Zeigt und ändert Einstellungen von Ethernet-Netzwerkkarten (Treiber-Level): z.B. Geschwindigkeit, Duplex, Auto-Negotiation, Wake-on-LAN, Offload-Funktionen, Statistikzählefile-4fdk7rvx39azjretinarqb】 .

### **Verwendung:**

- ethtool <iface> – Zeigt Standard-Eigenschaften: unterstützt Geschwindigkeiten (Supported link modes), aktuelle Speed/Duplex, Auto-negotiation on/off, Ports (TP, etc.), driver name, firmware-version, bus-info, etc.

- `ethtool -s <iface> speed <Mb/s> duplex <half/full> autoneg <on/off>` – Setzt feste Geschwindigkeit/Duplex. (z.B. `ethtool -s eth0 speed 100 duplex full autoneg off`). Vorsicht: muss vom Link Partner unterstützt werden, sonst Link down.
- `ethtool -k <iface>` – Zeigt Kernel-Offload-Funktionen (tx-checksumming, scatter-gather, TSO etc.) und ob an/aus. -K ... (capital K) schaltet diese: z.B. `ethtool -K eth0 gro off` (GRO ausschalten).
- `ethtool -i <iface>` – Infos zum Treiber (Driver name, version, firmware, bus).
- `ethtool -p <iface> [N]` – "Physically identify": Blinkt die LED am Port für N Sekunden (Standard 15), falls Karte das unterstützt. Hilft um Kabelport zu finden.
- `ethtool -S <iface>` – Ausgabe von Statistikzählern (packets, errors, etc. vom Treiber).
- `ethtool -g <iface>` – Ring Puffer Größen (RX/TX rings). -G zum Setzen.
- `ethtool -a <iface>` – Flow-Control (Pause Frame) Status (Autoneg etc.), -A zum Setzen on/off rx/tx.
- `ethtool -r <iface>` – Veranlasst die Karte, Link neu auszuhandeln (Retrain).
- `ethtool --identify <iface>` – alias für -p (Blinken).

### **Beispiele:**

- `ethtool eth0` -> zeigt z.B. Speed: 1000Mb/s, Duplex: Full, Auto-negotiation: on, etc.
- `ethtool -s eth0 speed 100 duplex half autoneg off` -> zwingt 100Mbit Half-Duplex (nur Test oder Legacy).
- `ethtool -p eth1 10` -> blinkt LED an eth1 für 10 Sekunden.
- `ethtool -K eth0 tso off` -> schaltet TCP Segment Offloading aus (im Troubleshooting falls Problem mit Offload).

### **hostnamectl (Control hostname in systemd)**

**Beschreibung:** Zeigt oder ändert den System-Hostname und verwandte Settings (wie Chassis Typ, Icon). hostnamectl ist Teil von systemd.

### **Verwendung:**

- **hostnamectl** (ohne args): Zeigt Systeminfo: Static hostname, Transient (via DHCP), Pretty (human name), und Kernel: Operating System, Kernel, Architecture, etc.
- **hostnamectl set-hostname <name> [--pretty|--static|--transient]** – Setzt den Hostnamen. Ohne Flag wird *static* und *transient* gleichermaßen gesetzt. *static* =

in /etc/hostname gespeichert (persistenter Name), *transient* = Laufzeit (kann z.B. DHCP ändern). *pretty* = nur für Anzeigezwecke (z.B. mit Großbuchstaben/Leerzeichen erlaubt).

Beispiel: hostnamectl set-hostname server42 – setzt static+transient auf "server42".

hostnamectl set-hostname "My Home PC" --pretty – setzt Pretty-Name.

- **hostnamectl set-icon-name <icon>** – Setzt Icon (z.B. computer-desktop, rein kosmetisch in einigen GUIs).
- **hostnamectl set-chassis <type>** – Setzt Gehäuse-Typ: desktop, laptop, server, tablet, embedded, etc. (Auch nur informative, kann z.B. in GNOME Systeminfo auftauchen).

#### Zusatz:

- hostnamectl status (alias ohne args) – Info.
- Traditional hostname Befehl (ohne ctl) zeigt oder setzt nur transient (bis Reboot).

### Netzwerk-Fehlerdiagnose und Name-Auflösung

#### netcat (nc) (TCP/UDP Swiss Army Knife)

**Beschreibung:** netcat (oder nc) ist ein universelles Netzwerk-Tool, um Verbindungen zu testen, Daten zu senden/empfangen, Ports zu lauschen etcfile-4fdk7rvx39azjretinarqb】. Kann sowohl Client als auch rudimentärer Server sein.

#### Verwendung:

- **Verbindung als Client:** nc [options] <Host> <Port> – Baut TCP-Verbindung zu Host:Port auf, leitet stdin -> Socket, und Socket -> stdout. So kann man z.B. echo "HEAD / HTTP/1.0" | nc example.com 80 (HTTP-Request senden). Oder nc time.nist.gov 13 (Daytime Protocol). Wenn man interaktiv ausführt, kann man manuell tippen (Telnet-Ersatz).
- **Port Listening (Server):** nc -l -p <Port> [Host] – Horcht auf Port (optional an gegebene IP gebunden). Bsp: nc -l -p 1234 lauscht auf TCP/1234 auf alle Interfaces; was reinkommt wird auf stdout ausgegeben, was man tippt geht an Client.
- **UDP mode:** nc -u – Statt TCP benutzt UDP. Bsp: nc -u 192.168.1.5 53 könnte man UDP Query schicken (z.B. an DNS server).
- **Port Scan:** nc -z -v host startport-endport – "Zero-I/O mode" nur scannen, nicht verbinden. Z.B. nc -z -v 192.168.1.10 1-1024 zeigt offene Ports. (Netcat als Portscanner ist möglich, aber Nmap dafür besser).
- **Timeout:** -w <secs> – Wartezeit bis Timeout für Verbindungen/Eingaben.

- **Keep-open (inetd style):** In GNU nc -k (listen bleibt offen nach Verbindungsende, auf neue).
- **Exec:** -e <prog> – *Unsichere Option* (nicht immer kompiliert, oft disabled aus Sicherheitsgründen): Leitet die I/O eines Programms an den Socket. So kann man z.B. remote shell spawn: nc -l -p 5000 -e /bin/bash. (Equivalent zum berüchtigten rm -rf example). Modern: use socat or ssh for shells.

### **Beispiele:**

- **Chat/Echo Test:** Terminal A: nc -l -p 5000, Terminal B: nc localhost 5000 -> alles was A tippt sieht B und umgekehrt.
- **Datei senden:** Auf Empfänger: nc -l -p 4500 > out.txt, auf Sender: nc destIP 4500 < file.txt. (Netcat baut TCP auf und streamt Datei).
- **HTTP Request:** printf "GET / HTTP/1.1\r\nHost: example.com\r\n\r\n" | nc example.com 80 – zeigt HTML der Homepage (HTTP/1.1 erfordert Host-Header).
- **UDP Ping** (ohne Antwort): echo -n "Hello" | nc -u 192.168.1.5 9000.
- **Port scan:** nc -z -v 192.168.1.20 20-30 -> tries FTP-data, FTP etc, output open/closed.

(*Netcat Implementierungen: Traditional vs OpenBSD netcat vs GNU netcat ncat (part of nmap) – Optionen leicht variierend. Z.B. OpenBSD netcat hat -l und -p zusammen als one option or so, etc.*)

### **iftop (Interface TOP, Bandwidth monitor)**

**Beschreibung:** Ein curses-basiertes Tool zum Anzeigen der aktuellen Bandbreitennutzung pro Verbindung (ähnlich top, aber für Netzwerk Trafficfile-4fdk7rvx39azjretinarqb】. Es zeigt welche Peers mit dem System kommunizieren und wieviel Bytes/s in/out. Hilfreich zum realtime Debugging wer Bandbreite frisst.

### **Verwendung:**

iftop -i <Interface>

Ohne -i wählt iftop das erste externe Interface.

### **Optionen:**

- **-n** – Keine DNS-Auflösung (zeigt IPs statt aufzulösen, wichtig um Ausgabe schnell zu haltenfile-4fdk7rvx39azjretinarqb】.
- **-N** – Keine Port-Auflösung (zeigt Portnummern statt Services Namen).
- **-P** – Zeigt Ports zusammen mit Host (separat Spalte). Kombiniert man oft: iftop -nP.

- **-B** – Bytes statt Bits in Anzeige (Standard iftop zeigt in bits/s).
- **-F <FilterNet>** – Nur Traffic, der in bestimmten Netzbereich fällt (Filter) betrachten, z.B. -F 192.168.0.0/16 filtert auf diese Range.
- **-p** – Promiscuous Mode (auch Traffic anzeigen, der nicht an lokalem Host endet? Normal iftop Filter host involvement).
- **-t** – Text-Only Modus (keine curses-Interface, nur in stdout Stats; nicht sehr üblich).

Im Interface kann man mit Tasten: h Hilfe, n togg DNS, s sort by source, d sort by dest, t sort by throughput, p toggle port display, b toggle bar graph, B cycles through rate intervals (default 2s,10s,40s windows).

### **traceroute / tracepath (Route Tracking)**

**Beschreibung:** Sendet eine Serie von Paketen mit steigenden TTL (Time-to-Live) Werten, um den Pfad (Router Hopps) zum Ziel auszumessefile-4fdk7rvx39azjretinarqb】 . Traceroute verwendet standard UDP oder ICMP ECHO, tracepath nutzt UDP with high port (no root needed). Jedes Hop sendet "Time Exceeded" zurück, das traceroute protokolliert. Ergebnis: Liste der Router vom Quell- zum Zielnetz.

#### **Syntax:**

traceroute [Optionen] <Host> [PacketSize]

tracepath <Host>

#### **Optionen (traceroute):**

- **-I**: ICMP Echo statt UDP verwenden (pakete wie ping).
- **-T**: TCP SYN statt UDP (z.B. um Firewalls zu durchdringen über port 80).
- **-p <Port>**: UDP port Start (Standard 33434).
- **-q <N>**: Anzahl Probes pro TTL (Default 3).
- **-w <Seconds>**: Timeout pro Hop Antwort (default 5s).
- **-n**: Nicht auflösen (IP statt DNSfile-4fdk7rvx39azjretinarqb】 .
- **-m <max\_ttl>**: Max Hop Count (Standard 30).
- **-f <first\_ttl>**: Start TTL (default 1). Manchmal hilfreich, mittendrin einzusteigen.
- **-4 / -6**: Force IPv4 / IPv6.

**Optionen (tracepath)** sind wenige: -n no dns, -b show both hostnames and IP, etc.

#### **Beispiel:**

```
$ traceroute example.com
1 192.168.1.1 (192.168.1.1) 1.123 ms 0.846 ms 0.799 ms
2 100.64.0.1 (100.64.0.1) 10.123 ms 9.876 ms 9.543 ms
3 203.0.113.5 (203.0.113.5) 15.432 ms 14.876 ms 15.210 ms
4 ae12.dar02.example.net (198.51.100.1) 30.456 ms 29.999 ms 30.123 ms
5 93.184.216.34 (93.184.216.34) 35.678 ms 35.900 ms 35.456 ms
```

Zeigt 5 Hops (Heimrouter, CGNAT router, ISP core, example.net router, Ziel).

tracepath example.com ähnlich, oft mit asymmetrischen result such as pmtu info.

### **mtr (My Traceroute)**

**Beschreibung:** Kombiniert traceroute und ping in einem kontinuierlichen Diagnose-Tool. mtr sendet fortlaufend Pakete über alle Hops und aktualisiert in Echtzeit die Latenz und Packetloss-Statistiken pro Hop. Hilfreich um instabile Verbindungen/Packet Loss auf bestimmten Strecken zu identifizieren.

#### **Verwendung:**

mtr [Optionen] <Ziel>

Interaktiv curses UI standard, oder mit -r (report) Modus für einmalige Ausgabe.

#### **Optionen:**

- **-4 / -6:** Force IPv4/IPv6.
- **-n:** Nicht auflösen, nur IPs (schneller).
- **-w:** Wide-output (zeigt beide host+IP columns).
- **-c <count>:** Limit auf <count> Probezyklen, dann beenden (für Report).
- **-r:** Report-Modus (nicht curses, druckt Tabelle und endet).
- **-i <sec>:** Sekunden zwischen Probe-Paketen (default 1.0).
- **-p:** Pausiert Ausführung beim Start (Press enter to start).
- **-t:** Text Mode (like curses but in dumb terminal).
- **-u:** UDP mode (Standard mtr uses ICMP Echo, as -l in traceroute terms). -T for TCP mode also available.

#### **Anzeige:**

MTR listet Hops fortlaufend wie traceroute, plus Spalten: Loss%, Snt (packets sent), Last (ms last latency), Avg, Best, Wrst (worst), StDev. Es aktualisiert kontinuierlich.

Beispiel-Zeile:

3. 203.0.113.5 0.0% 10 15.4 16.1 14.8 20.3 1.3

### **ping (ICMP Echo)**

**Beschreibung:** Schickt ICMP Echo-Request-Pakete an eine Adresse und misst die Antwortzeiten (Echo-Reply), um Erreichbarkeit und Round-Trip-Time festzustellen】.

#### **Syntax:**

ping

\*\*(Fortsetzung)\*\*

### ping (ICMP Echo Requests)

\*\*Beschreibung:\*\* `ping` prüft die Erreichbarkeit eines Hosts im Netzwerk mit \*\*ICMP-Echo-Paketen\*\* und misst die Round-Trip-Zeit (RTT) der Antwort. Es sendet fortlaufend Anfragen (standardmäßig, bis abgebrochen) und berechnet Statistik über Verlust und Latenz.

\*\*Syntax:\*\*

``` bash

ping [Optionen] <Ziel>

Hier kann <Ziel> ein Hostname oder eine IP-Adresse sein.

Wichtige Optionen:

- **-c <Anzahl>** – Anzahl der ICMP-Pakete festlegen. Z.B. ping -c 4 8.8.8.8 sendet genau 4 Pakete】.
- **-i <Sekunden>** – Interval zwischen den Pings (Standard 1 Sekunde). Man kann z.B. -i 0.2 für fünf Pings pro Sekunde (root-Recht nötig bei <0.2>).
- **-W <Timeout>** – Wartezeit in Sekunden auf eine Antwort, bevor als **timeout** gezählt (Default meist 1 Sekunde, bei -c am Ende kann man höher setzen).
- **-q** – "Quiet": keine per-Ping Ausgabe, nur eine Zusammenfassung am Ende.
- **-p <Pattern>** – Sende individuelles Byte-Muster im Paket (Hexadezimal angegeben). Gut für bestimmte Diagnosen (z.B. bestimmte Bits toggeln sehen).
- **-s <Packetsize>** – Größe des zu sendenden Pakets in Bytes (ohne ICMP-Header). Standard 56 Bytes (plus 8 Byte ICMP-Header = 64 Bytes in Ausgabe)】.

- **-t <TTL>** – Setzt die Time-To-Live (IPv4) oder Hop Limit (IPv6) für die gesendeten Pakete. (Nützlich, um gezielt nur bis zu einem gewissen Hop zu gelangen, ähnlich traceroute Step).
- **-I <Interface>** – Bindet an ein bestimmtes Quell-Interface (oder IP) für Multi-homed hosts. Alternativ **-S <Source_Address>**.
- **-4, -6** – Erzwingt IPv4 oder IPv6.
- **-f** – "Flood ping": schickt so schnell wie möglich Pings (sehr viele pro Sekunde) und druckt einen Punkt pro empfangenes Echo. (Nur root, zur Stresstest oder schnelle Packetloss Erkennung).
- **-L** (bei einigen ping-Versionen) – Für Echo an Broadcast/Multicast-Adresse (ermöglicht Broadcast-ping, was standardmäßig blockiert ist).
- **-a** – Bei Empfang einen Terminal-Bell ausgeben (akustisches Signal).
- **-D** – Zeitstempel vor jeder Zeile (UNIX time).
- **-U** – Zeitstempel in Us seit Boot vor jeder Zeile (für Messungen).

Ausgabe-Interpretation:

Jede empfangene Antwort erscheint als Zeile:

64 bytes from 8.8.8.8: icmp_seq=1 ttl=117 time=8.34 ms

Bedeutet: Echo-Reply von 8.8.8.8 erhalten, Größe 64 Bytes, Sequenznummer 1, TTL im Reply 117, RTT 8.34 ms. Wenn Antworten ausbleiben, erscheinen Zeilen wie "Request timeout".

Am Ende kommt eine **Statistik**:

4 packets transmitted, 4 received, 0% packet loss, time 3005ms

rtt min/avg/max/mdev = 8.334/8.512/8.790/0.177 ms

Zeigt Verlust und RTT-Werte.

Beispiele:

- ping 1.1.1.1 – Kontinuierlicher Test zu 1.1.1.1 (Cloudflare DNS). Abbruch mit Strg+C.
- ping -c 5 www.example.com – 5 Pings an example.com, dann stop mit Ausgabe der Statistik.
- ping -I eth1 192.168.2.1 – Ping von Interface eth1 aus (wenn Multi-Interface).
- ping -s 1000 -c 3 host – Sende 3 Pings mit Paketgröße ~1000 Bytes (ICMP Payload).

- ping -6 -c 4 ipv6.google.com – IPv6 Ping zu Google.

traceroute (siehe oben)

(bereits behandelt in vorangehendem Teil, siehe „traceroute“ in Netzwerkdiagnose.)

tracepath (siehe oben)

(siehe „tracepath“ bei Netzwerkdiagnose.)

resolvectl (Systemd-resolved Query)

Beschreibung: CLI-Tool zum Abfragen des systemd-resolved-Dienstes (Namensauflösung). resolvectl kann DNS-Lookups durchführen, DNS-Settings anzeigen, LL/MNR Browsing etc. Ersetzt Tools wie systemd-resolve (veraltet).

Verwendung:

- **resolvectl status** – Zeigt die aktuellen DNS-Einstellungen pro Link (Interface): DNS-Server, Suche-Domains, LLMNR/MulticastDNS Status, DNSSEC, usw.
- **resolvectl query <Hostname>** – Löst einen Hostnamen auf (A/AAAA-Record). Ausgabe enthält v6 und v4 Adressen, CNAME falls, DNSSEC info, Cache info, antwortender Server etc.
Beispiel: resolvectl query example.com.
- **resolvectl query -t MX <Domain>** – DNS-Abfrage bestimmten Typs (hier MX-Records). Unterstützt -t ANY, A, AAAA, MX, SRV, SOA etc.
- **resolvectl query <IP>** – Reverse Lookup (IP zu Name via PTR).
- **resolvectl flush-caches** – Löscht den DNS-Cache von systemd-resolved.
- **resolvectl dns <Interface> [Server...]** – Setzt DNS-Server für ein bestimmtes Interface (über DBus). resolvectl domain <iface> ... analog. (Normal wird das via NetworkManager/Netplan/etc gemacht).
- **resolvectl revert <Interface>** – Löscht manuelle Overrides und kehrt zu vom Netzwerkdienst gelieferten DNS zurück.

Beispiel:

```
$ resolvectl query -t MX example.com
example.com IN MX 0 .
-- Information acquired via protocol DNS in 15.1ms.
-- Data is authenticated: no
Bedeutet example.com hat keinen MX (Mail).
```

resolvectl status Ausschnitt:

Global

Protocols: LLMNR=resolve mDNS=no DNSOverTLS=no DNSSEC=no/unsupported
resolv.conf mode: stub

Link 2 (eth0)

Current Scopes: DNS

Protocols: +DefaultRoute +LLMNR -mDNS DNSOverTLS=opportunistic DNSSEC=...

Current DNS Server: 192.168.1.1

DNS Servers: 192.168.1.1 1.1.1.1

DNS Domain: ~.

Zeigt z.B. that eth0 hat default DNS 192.168.1.1 plus Cloudflare, und ~. (default domain).

dig (DNS Lookup Utility)

Beschreibung: dig (domain information groper) führt DNS-Abfragen aus. Es ist Teil von BIND-Tools (bind-utils). Im Gegensatz zu host oder nslookup bietet dig ausführlich formatierbare Antworten. Standard macht eine A-Record-Abfrage und gibt Antwort mit Abschnitt **ANSWER SECTION** aus.

Syntax:

dig [@DNS-Server] <Name> [<Typ>] [<Klasse>] [Optionen]

<Klasse> normalerweise IN (Internet), selten was anderes.

Wichtige Optionen:

- **@server** – Spezifischen DNS-Server benutzen statt System-resolver. Z.B. dig @8.8.8.8 example.com.
- **<Typ>** – Abfragetyp: A, AAAA, MX, TXT, SRV, NS, SOA, CNAME, PTR, ANY etc. (Default A).
- **+short** – Ausgabe nur die Antworten knapp (einfacher, scripterfreundlic0】 . Z.B. dig +short example.com A -> "93.184.216.34".
- **+noall +answer** – Unterdrückt alle außer Answer-Section (ähnlich short aber mit TTL und Info).
- **+trace** – Iterativer Trace durch DNS-Hierarchie (Root bis Ziel, ähnlich dig an Root und dann NS referrals folgen). Zeigt jeden Schritt. Hilfreich zum Debug.
- **+tcp** – Erzwinge TCP statt UDP (z.B. bei großen DNS-Records oder EDNS issues).

- **+nssearch** – Findet alle NS des Ziel und fragt diese nach SOA (um Delegation correctness zu checken).
- **-x <IP>** – Reverse lookup (PTR) für IP. (Bequem: dig -x 8.8.8.8).
- **+multi** – Schöne Ausgabe in Spalte falls mehrere TXT oder so.
- **+nodnssec oder +dnssec** – Anfordern oder nicht (um RRSIG etc. zu sehen).
- **+ttlshort** – Output TTL in short mode. (e.g. dig +short +ttl google.com -> "300 172.217.168.14").
- **+stats / +nostats** – Schaltet Statistik-Banner am Ende an/aus.
- **+cm** – Colorize output (if compiled with color support).

Beispiele:

- dig example.com ANY – Fragt alle Records (die server je nach config liefern).
- dig google.com MX +noall +answer – Zeigt nur MX Antworten.
- dig @1.1.1.1 cloudflare.com A – Abfrage an 1.1.1.1 nach cloudflare's A.
- dig -x 8.8.4.4 +short – Reverse Lookup, kurze Ausgabe ("dns.google").
- dig +trace example.org – Startet vom Root Nameserver und verfolgt Delegation bis example.org resolved ist.

nslookup (Name Server Lookup - veraltet)

Beschreibung: Ein einfaches interaktives DNS-Abfrageprogramm. War Teil von BIND, gilt als veraltet im Vergleich zu dig. Viele Admins kennen es dennoch.

Verwendung:

nslookup [<Host> [<DNS-Server>]]

Ohne args geht es in interaktiven Modus. Dessen Kommandos: server <DNS>, set q=MX, set type=AAAA, set debug, exit.

Beispiele:

```
$ nslookup example.com 8.8.8.8
```

Server: 8.8.8.8

Address: 8.8.8.8#53

Non-authoritative answer:

Name: example.com

Address: 93.184.216.34

Reverse: nslookup 93.184.216.34 -> zeigt PTR.

Warnung: Neuere doc empfehlen dig oder host statt nslookup.

whois (Domain Whois Query)

Beschreibung: Fragt Whois-Datenbanken nach Registrierungsinformationen einer Domain oder 5】. Das Tool verbindet sich zum passenden Whois-Server (verwendet /etc/whois.conf, TLD heuristics) via TCP 43 und gibt den Text aus.

Syntax:

whois [Optionen] <DomainOrIP>

Ausgabe: Je nach TLD und Registry differierend. Enthält Domain-Inhaber (bei Generic oft DSGVO anonymisiert), Registrar, Erstellungs- und Ablaufdatum, Nameserver, Contact-IDs. Bei IP WHOIS (z.B. whois 8.8.8.8) gibt es Informationen über den zugeteilten IP-Block (ISP, RIR wie ARIN/RIPE...).

Optionen:

- **-H** – Unterdrückt das Banner "Whois Server" in RIPE responses.
- **-p <port>** – Andere Port.
- **--verbose** – Mehr Info über Server selection.
- **--help** – Display help.

Beispiele:

- whois example.com -> zeigt Registrar (ICANN info), Status, Name der Organisation, etc.
- whois 2620:0:2d0:200::7 -> IPv6 whois (gibt i.d.R. ARIN oder so mit OrgName).
- whois AS15169 -> WHOIS an RADb für Autonomous System 15169 (Google).

(Beachte: Manche TLDs haben separate Tools, z.B. jwhois config. Das Standard whois ist meist in Debian whois package (Marco d'Itri) which covers a lot. GDPR hat Personendaten stark eingeschränkt in Output.)

arp (Address Resolution Protocol table) – Hinweis: arp wird mittlerweile durch ip neigh ersetzt.

Beschreibung: Zeigt den ARP-Cache (IPv4) an oder manipuliert Einträge. (Linux modern: use ip neigh aber arp Kommando aus net-tools ist noch gebräuchlich).

Befehle:

- arp -a – Listet ARP-Einträge (Hostname, IP, MAC, Interface).

- arp -n – Wie -a, aber keine DNS für IP.
- arp -d <IP> – Löscht ARP-Eintrag für IP.
- arp -s <IP> <MAC> – Fügt statischen ARP-Eintrag hinzu (MAC fix für IP) – gleich ip neigh add ... nud permanent.

Beispiel: arp -a -> router.lan (192.168.1.1) at aa:bb:cc:dd:ee:ff [ether] on eth0.

Sicherheit und Firewall

iptables/iptables-nft (Packet Filter)

Beschreibung: Linux Kernel-Paketfilter (Firewall) via Netfilter-Schnittstelle. iptables verwaltet Regeln in verschiedenen Tabellen: **filter**, **nat**, **mangle**, **raw**, **security**, für IPv4. (Für IPv6, ip6tables). Mit Kernel 4.x/5.x viele Distros nutzen "iptables-nft" as a frontend to nftables, aber Syntax bleibt für user gleich.

Syntax Grundkommandos:

iptables [-t <table>] <operation> <chain> <matches...> <target>

Operationen: **-A** (append), **-I** (insert), **-D** (delete), **-C** (check exist), **-L** (list), **-F** (flush chain), **-N** (new chain), **-X** (delete user chain).

Standard Tabellen:

- **filter** (Default): Chains **INPUT**, **OUTPUT**, **FORWARD**.
- **nat**: Chains **PREROUTING**, **OUTPUT**, **POSTROUTING** (masquerading, port forwarding etc.).
- **mangle**: For packet alteration (TOS, mark), Chains in multiple points (PREROUTING, OUTPUT, etc.).
- **raw**: For conntrack exemption (NOTRACK), Chains PREROUTING, OUTPUT.
- **security**: SELinux related (rare).

Matches / Options:

- **-p <proto>**: Protocol (tcp, udp, icmp, icmpv6, all).
- **-s <IP/CIDR>, -d <IP/CIDR>**: Source or destination address (or network).
- **-i <iface>, -o <iface>**: Inbound or outbound network interface (for PREROUTING/INPUT/forward vs OUTPUT/POSTROUTING).
- **--sport, --dport** (with -p tcp/udp): Source/dest port or port range (e.g. --dport 80 or --dport 1000:2000).

- **-m state --state <states>**: State match (connection tracking states: INVALID, ESTABLISHED, NEW, RELATED). Common: allow ESTABLISHED,RELATED on INPUT for responses.
- **-m conntrack --ctstate**: (newer usage, same as state module now).
- **-m connlimit, -m limit** etc: Many modules. limit --limit 5/minute for rate-limit logging for example. connlimit --connlimit-above N limit parallel connections.
- **-m multiport**: match multiple ports in one rule if not contiguous, e.g. -m multiport --dports 80,443,8080.
- **-m tcp --tcp-flags SYN,RST,ACK SYN**: match specific TCP flags combination (here SYN only). Often used for --syn (alias) which matches syn packets (to identify new connections).
- ****-m comment --comment "text"**: attach comment to rule (for documentation).

Targets (Actions):

- **ACCEPT**: Packet erlaubt.
- **DROP**: Verwerfen ohne Antwort.
- **REJECT**: Verwerfen mit Antwort (ICMP port unreachable o.Ä.).
- **LOG**: Loggt Paket (via kernel log), dann Verarbeitung geht weiter (so typ. Kombi LOG dann DROP). Options: --log-prefix "msg", --log-level warning.
- **MASQUERADE** (in nat POSTROUTING): SNAT für dynamische IP (Router Use-case) – ersetzt Source-IP durch Interface-IP.
- **SNAT** (in nat POSTROUTING): Statisches Source-NAT (requires --to-source IP[:port-range]).
- **DNAT** (in nat PREROUTING/OUTPUT): Destination-NAT (Port-Forwarding; --to-destination IP:port).
- **REDIRECT** (nat PREROUTING/OUTPUT): Umleiten an lokalen Host (for transparent proxy etc.).
- **MARK** (mangle): Markiert Paket mit fwmark (für routing decisions via ip rule, etc.).
- **RETURN**: In user-defined chain springt zurück zum Aufrufer-Chain (stop further rules in current chain). In builtin chain, RETURN = apply chain policy.
- **QUEUE**: Übergibt Paket an Userspace (via libnetfilter_queue).
- **DROP vs. policy**: If chain policy is DROP, falling off end = drop.

Beispiele:

- **List Regeln:** iptables -L -n -v (filter table). iptables -t nat -L -n -v NAT rules. -n no DNS, -v verbose (packet counters).
- **Allow incoming SSH:**

② iptables -A INPUT -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
 iptables -A OUTPUT -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
 (typical stateful allow SSH in).

② **Drop all forward by default:** iptables -P FORWARD DROP.

② **Masquerading (NAT):**

② iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
 (for home router with ppp0 outbound).

② **Port Forward:**

Forward ext eth0 port 80 to internal 192.168.1.100 port 8080:

② iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 \
 -j DNAT --to-destination 192.168.1.100:8080

iptables -A FORWARD -p tcp -d 192.168.1.100 --dport 8080 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

② **Logging Drop:**

- iptables -A INPUT -p tcp --dport 23 -j LOG --log-prefix "Telnet attempt: " --log-level info
- iptables -A INPUT -p tcp --dport 23 -j DROP
- (log then drop telnet).

firewall-cmd (firewalld frontend)

Beschreibung: Steuerprogramm für **firewalld**, eine dynamische Firewall (standard in RHEL/CentOS 7+, Fedora, etc.). firewall-cmd ermöglicht das Konfigurieren von Zonen, Regeln (services, ports, masq) zur Laufzeit und in Config. Firewalld nutzt intern iptables/nftables.

Syntax:

firewall-cmd [--zone=<zone>] [--permanent] <aktion> <Parameter>

Grundkonzepte: Zonen (trusted, public, home, internal, work, external, dmz, block, drop etc.), Schnittstellen sind Zonen zugewiesen. Dienste (vordefinierte Regelsets) können pro Zone erlaubt werden oder einzelne Ports/Protokolle. Man kann Einstellungen permanent (persistiert in XML config) oder nur runtime vornehmen.

Wichtige Befehle:

- **--state**: Zeigt an, ob firewalld läuft (running/not running).
- **--get-active-zones**: Zeigt welche Zonen aktiv sind und welche Interfaces zugeordnet (z.B. "public: wlp2s0").
- **--get-zones**: Listet alle bekannten Zonen. **--list-all** [--zone=Z] listet alle Einstellungen einer Zone (services, ports, masquerade etc0) .
- **--zone=<zone> --add-service=<service>**: Erlaubt den vordefinierten service in angegebener Zone (runtime). Bsp: firewall-cmd --zone=public --add-service=http – ermöglicht HTTP (Port 80) in Zone public.
- **--zone=<zone> --add-port=<port>/<proto>**: Erlaubt individuellen Port. Bsp: --add-port=5000/tcp.
- **--remove-service=..., --remove-port=...** analog zum Entfernen.
- **--add-masquerade**: Aktiviert NAT/Masquerading in der Zone (z.B. für Internet-sharing zone).
- **--query-service=<svc>**: Fragt ob Service in zone erlaubt (exit code 0/1).
- **--permanent**: Gibt man diese Option an, wird die Änderung in der persistenten Konfiguration vorgenommen (aber nicht an laufende sofort angewendet, es sei denn man ruft extra an). In firewalld, runtime und permanent config sind getrennt.
- **--reload**: Lädt die permanente Konfiguration neu (in runtime) – im Grunde wendet Config an.
- **--runtime-to-permanent**: Speichert aktuellen Laufzeit-Zustand als neue permanente Konfiguration (was man on-the-fly geändert hat, persistieren).
- **--zone=<zone> --change-interface=<iface>**: Weist ein Interface einer Zone zu (persistente Zuordnung).
- **--new-zone=<name>, --delete-zone=...**: Eigene Zonen erstellen/löschen.
- **--list-services, --get-services**: Listet bekannte Service-Namen (die in /etc/firewalld/services/*.xml definiert sind).
- **--service=<svc> --get-ports**: Zeigt welche Ports hinter einem Service stecken.
- **--add-rich-rule '<rule>'**: Fügt sog. Rich Rule hinzu (eine erweiterte Regel-Syntax in firewalld, z.B. mit Logging, Zeitplänen). Z.B.:
- `firewall-cmd --add-rich-rule='rule family="ipv4" source address="192.0.2.0/24" port port="22" protocol="tcp" accept'`

- (Erlaubt SSH nur von 192.0.2.0/24).

Beispiele:

- Öffnen von HTTPS permanent:
- `firewall-cmd --add-service=https --permanent`
- `firewall-cmd --reload`
-
- Port 12345/udp temporär öffnen in Zone public:
`firewall-cmd --zone=public --add-port=12345/udp` (nach Reboot weg).
- Interface eth0 zur Zone external zuweisen:
`firewall-cmd --zone=external --change-interface=eth0 --permanent && firewall-cmd --reload.`
- Status: `firewall-cmd --list-all` (wenn default zone, z.B. public, dann deren config).

ufw (Uncomplicated Firewall)

Beschreibung: Vereinfachtes Firewall-Frontend (Ubuntu/Debian) für iptables. ufw erlaubt einfache allow/deny-Regeln definieren, typischerweise nach Services oder Ports, optional in/out interface.

Syntax:

`ufw [Optionen] <Aktion>`

Grundbefehle:

- **ufw enable / disable** – Aktiviert oder deaktiviert die Firewall (bei disable: setzt iptables default ACCEPT und löscht Regeln).
- **ufw status [verbose]** – Zeigt aktuelle Regel-Liste. Mit verbose sieht man auch Interfaces, logging status etc.
- **ufw default allow|deny [incoming|outgoing|routed]** – Setzt Standardpolicy. Standard-Einstellung: incoming deny, outgoing allow.
- **ufw allow <Port/Service> [proto tcp/udp] [from <IP> [to <IP>]] [port <P2>]** – Erlaubt eingehenden Traffic. Beispiele:
 - `ufw allow 22` (erlaubt TCP/22 auf allen Interfaces).
 - `ufw allow proto udp from 10.0.0.0/8 to any port 53` (erlaubt UDP DNS von privatem Netz).
 - `ufw allow out 80/tcp` (erlaubt ausgehende HTTP).
 - `ufw allow in on eth0 to any port 443` (erlaubt eingehendes HTTPS auf eth0).

- **ufw deny <Spec>** – Blockiert (mit REJECT) entsprechend.
- **ufw reject <Spec>** – Synonym zu deny (deny und reject meist ähnlich in ufw; auf Debian might differ).
- **ufw limit <Port/Service>** – Wie allow, aber mit Rate-Limit (gegen Bruteforce). Z.B. ufw limit ssh erlaubt SSH, aber blockiert IP temporär bei >6 Verbindungen/30s.
- **ufw delete <RuleSpec>** – Entfernt Regel (man kann Regel genau wie bei allow angeben, mit 'delete' davor, oder Nummer aus ufw status numbered).
- **ufw logging on|off|LEVEL** – Logging ein/aus und Level (off, low, medium, high, full).
- **ufw reset** – Setzt ufw auf Auslieferungszustand (alle Regeln weg, policies reset, und disabled).
- **ufw route allow/deny ...** – Regeln für Forwarded Traffic (Routen). Bsp: ufw route allow in on eth1 out on eth0 to any port 80 for forwarded.
- **ufw insert <num> <rule>** – Fügt Regel an bestimmter Position ein.
- **ufw show raw** – Zeigt resultierende iptables-Regeln.

Beispiele:

- Standard: ufw default deny incoming; ufw default allow outgoing; ufw enable.
- Webserver: ufw allow 'Nginx Full' (in /etc/ufw/applications.d vordefinierte Profile, 'Apache Full' etc., Nginx Full = 80+443).
- Only LAN access to a service: ufw allow from 192.168.0.0/16 to any port 3306 (MySQL offen nur intern).
- Block eine IP: ufw deny from 203.0.113.5.
- Show rules: ufw status verbose -> e.g.
- Status: active
- Logging: on (low)
- Default: deny (incoming), allow (outgoing), deny (routed)
- New profiles: skip
-
- To Action From
- -- ----- -----

- 22/tcp LIMIT Anywhere # SSH
- 80,443/tcp ALLOW Anywhere # WWW
- 22/tcp (v6) LIMIT Anywhere (v6)
- 80,443/tcp (v6) ALLOW Anywhere (v6)
-

(UFW ist beliebt wegen einfacher Syntax, aber fehlender Komplexität bei komplexen Setups. Firewalld und raw iptables/nft sind mächtiger in Feinheiten.)

nft (nftables Firewall)

Beschreibung: nftables ist der modernere Packet-Filter (ersetzt iptables). Der Befehl nft wird zum Konfigurieren verwendet. Sehr flexibel (Datenstrukturen, one-rule multiple matches).

Syntax:

nft [options] <command>

Man kann Kommandos interactive eingeben oder als Skript. Der Kommandostil ähnelt C:
e.g.

nft add table inet filter

nft add chain inet filter input { type filter hook input priority 0 \; }

nft add rule inet filter input tcp dport 22 accept

(Librally, but given complexity and that the original question references iptables and ufw, likely beyond scope).

Abschluss

Dieses Kompendium deckt die wesentlichen Befehle für die Linux+ (XK0-005) Prüfung ab, gegliedert nach Themengebieten. Die Befehle umfassen Benutzer- und Rechteverwaltung, Prozess- und System-Tools, Netzwerk-Konfiguration und -Diagnose, Software-Installation, Container, Sicherheit (Firewall, SELinux/AppArmor siehe oben Rich-Rules, etc.), Dienste und Planung.

Durch die strukturierte Darstellung mit Syntax, Beschreibung, Optionen und Beispielen dient es als umfassendes Nachschlagewerk. **Wichtig:** In einer Prüfungssituation sollte man sich auf die korrekte Anwendung der Optionen und das Verständnis der Konzepte dahinter konzentrieren (z.B. was macht chmod 775 oder umask 002 bewirkt).

Mit dieser Referenz sollte die Vorbereitung auf typische Linux-Aufgaben – vom Anlegen eines Benutzers, dem Ändern von Dateirechten, über das Einrichten eines Cronjobs bis

zum Konfigurieren eines Apache-Dienstes oder Setzen einer Firewallregel – bestens unterstützt werden.

Kompendium der Linux+ (XK0-005) Befehle

Dieses Nachschlagewerk umfasst alle wichtigen Befehle aus dem offiziellen **Command Reference Guide** für die CompTIA Linux+ Prüfung (XK0-005). Die Befehle sind thematisch gegliedert (Benutzerverwaltung, Netzwerk, Prozesse, Dateisystem, Paketmanagement, Container, Sicherheit etc.), mit Beschreibung, vollständiger Syntax, allen verfügbaren Optionen und ggf. wichtigen Unterbefehlen. Es dient als umfassendes Cheat-Sheet zur Prüfungsvorbereitung.

Grundlegende Linux-Befehle

ls (List Directory Contents)

Beschreibung: Listet den Inhalt eines Verzeichnisses auf. Ohne Parameter zeigt ls die nicht versteckten Einträge im aktuellen Verzeichnis in alphabetischer Reihenfolgewiki.ubuntuusers.deionos.de.

Syntax:

ls [OPTION]... [VERZEICHNIS]

Wenn kein Verzeichnis angegeben wird, wird das aktuelle Verzeichnis aufgelistetwiki.ubuntuusers.de.

Optionen:

| Option | Beschreibung |
|----------------------|---|
| -a, --all | Zeigt alle Einträge an, inklusive versteckter Dateien (auch . und ..) wiki.ubuntuusers.deionos.de . |
| -A, --almost-all | Wie -a, aber ohne . und .. – listet versteckte Dateien außer den Verzeichnissen ./.. wiki.ubuntuusers.deionos.de . |
| -l | Langformat: Detaillierte Ausgabe (Rechte, Besitzer, Größe, Datum usw.) je Eintrag wiki.ubuntuusers.deionos.de . |
| -d, --directory | Listet Verzeichniseintrag selbst statt dessen Inhalt (nützlich, um Infos über Ordner zu sehen) wiki.ubuntuusers.deionos.de . |
| -h, --human-readable | Im Langformat (-l) Größen in menschenlesbare Einheiten (KB, MB ...) anzeigen wiki.ubuntuusers.deionos.de . |
| -i | Zeigt die Inode-Nummer jedes Eintrags an wiki.ubuntuusers.deionos.de . |

| Option | Beschreibung |
|------------------------|--|
| -C | Ausgabe in Spalten nebeneinander
(Standardformat) wiki.ubuntuusers.deionos.de . |
| -m | Listet Einträge kommassepariert in <i>einer</i> Zeile wiki.ubuntuusers.deionos.de . |
| -R, --recursive | Rekursiv: Zeigt auch Inhalte von Unterverzeichnissen (durchläuft Verzeichnisbaum) wiki.ubuntuusers.deionos.de . |
| -t | Sortiert nach Zeitstempel (Änderungszeit, neueste zuerst) anstatt alphabetisch. |
| -r | Kehrt die Sortierreihenfolge um (Reverse order). |
| --hyperlink | Macht Dateinamen (in unterstützten Terminals) als klickbare Links ausgebbar wiki.ubuntuusers.de . |
| --help | Zeigt die Hilfe zum Befehl an. |
| --version | Zeigt Versionsinformationen von ls. |

(Weitere Optionen sind in den Manpages dokumentiertwiki.ubuntuusers.de.)

cat (Concatenate and Print Files)

Beschreibung: Gibt den Inhalt von Dateien auf der Standardausgabe aus oder verknüpft (konkatiniert) mehrere Dateien. Oft verwendet, um Textdateien anzuzeigen oder Inhalte zu pipenionos.at. Der Name steht für „concatenate“ (verketten).

Syntax:

cat [OPTION]... [DATEI]...

Ohne Optionen schreibt cat den Inhalt der angegebenen Datei(en) in der angegebenen Reihenfolge auf die Standardausgabe. Kann mehrere Dateien hintereinander ausgeben oder mit Umleitungen/ Pipes genutzt werden.

Optionen:

| Option | Beschreibung |
|---------------|---------------------|
|---------------|---------------------|

| | |
|-----------|---|
| -n | Nummeriert <i>alle</i> ausgegebenen Zeilen fortlaufend ibm.comionos.at . |
| -b | Nummeriert nur nicht-leere Ausgabelinien (Leerzeilen werden nicht nummeriert) ibm.comionos.at . |

Option Beschreibung

- s** Squeeze: Unterdrückt aufeinanderfolgende Leerzeilen (mehrere leere Zeilen werden als eine ausgegeben)[ionos.at](#).
- v** "Visible" – Macht nicht druckbare Steuerzeichen sichtbar (außer Tab und Zeilenende)[ionos.at](#).
- e** Entspricht -v mit Anzeige des Zeilenende-Zeichens \$ am Zeilenende (kombiniert -v und Kennzeichnung von Zeilenumbrüchen)[ionos.at](#).
- t** Entspricht -v mit Darstellung von Tabulatoren als ^I (Tab-Markierung)[ionos.at](#).
- E** Zeigt \$ am Ende jeder Zeile (Äquivalent zu -vE, oft identisch zu -e je nach Implementierung).
- T** Zeigt Tabulatoren als ^I an (ähnlich -vT, meist wie -t).
- A** Kombiniert mehrere der obigen: entspricht oft -vET (alle nicht-druckbaren Zeichen, Tabs und Zeilenenden sichtbar machen).
- u** (Ignoriert in GNU cat – historisch für ungepufferte Ausgabe).
- help** Zeigt Hilfe-Informationen an.
- version** Zeigt Versionsinfo an.

cd (Change Directory)

Beschreibung: Wechselt in ein anderes Verzeichnis. Mit cd kann das aktuelle Arbeitsverzeichnis geändert werden („change directory“)[ionos.at](#). Ohne Argument wechselt cd standardmäßig ins Heimatverzeichnis des Benutzers.

Syntax:

`cd [OPTION] [VERZEICHNIS]`

Man kann relative Pfade (bezogen auf das aktuelle Verzeichnis) oder absolute Pfade (ausgehend vom Wurzelverzeichnis /) angeben. Erfolgt der Wechsel in ein Zielverzeichnis, so wird dieses das neue aktuelle Arbeitsverzeichnis; bei nicht vorhandenen Zielen oder fehlenden Rechten bleibt das Verzeichnis unverändert[ionos.at](#).

Optionen:

Option Beschreibung

- L** Folge symbolischen Links **logisch**. Dies ist die Voreinstellung. (Beim Navigieren durch einen Symlink bleibt man in der logischen Pfadstruktur)[ionos.at](#).

Option Beschreibung

- P Folge symbolischen Links **physisch**. Wechselt in das tatsächliche Ziel im Dateisystem und löst symbolische Links auf[ionos.at](#).

Unterbefehle / spezielle Aufrufe:

- cd ~ wechselt ins Heimatverzeichnis des aktuellen Benutzers[ionos.at](#).
- cd - wechselt in das vorherige Arbeitsverzeichnis zurück (Toggle zwischen zwei Verzeichnissen)[ionos.at](#).
- cd /Pfad/zu/Verz wechselt absolut in ein angegebenes Verzeichnis vom Wurzelpfad aus (hier z.B. nach /Pfad/zu/Verz).
- cd .. wechselt eine Ebene höher (ins Parent-Verzeichnis), cd ../../ entspricht zwei Ebenen usw.
- cd . bleibt im aktuellen Verzeichnis (. bezeichnet das aktuelle Verzeichnis, meistens kein Effekt).

pwd (Print Working Directory)

Beschreibung: Gibt den vollständigen Pfad des aktuellen Arbeitsverzeichnisses aus[ionos.at](#). Mit pwd kann man sich anzeigen lassen, *in welchem Verzeichnis man sich gerade befindet*.

Syntax:

pwd [OPTION]...

Optionen:

Option Beschreibung

- Ausgabe des Pfads mit Berücksichtigung symbolischer Links (logischer Pfad, Standard). Wenn z.B. ein symbolischer Link als aktuelles Verzeichnis dient, wird der Linkpfad angezeigt[ionos.de](#).
- P Ausgabe des *physischen* Pfads: Symbolische Links werden aufgelöst, der tatsächliche Verzeichnispfad wird angezeigt[ionos.de](#).
- help Zeigt Hilfe-Informationen zu pwd an.
- version Zeigt Versionsinformationen an.

Hinweis: pwd ist häufig als Shell-Builtin implementiert. Das externe Programm /bin/pwd unterstützt ggf. andere oder weniger Optionen (bei GNU Coreutils nur --help und --version zusätzlich)[ionos.de](#). In der Regel genügt der Shell-eigene pwd Befehl.

whoami

Beschreibung: Zeigt den Benutzernamen des aktuell angemeldeten Users anfile-4fdk7rvx39azjretinarqb. Praktisch, um schnell zu prüfen, unter welchem Account ein Prozess läuft.

Syntax:

whoami

Keine spezifischen Optionen. Der Befehl gibt äquivalent echo \$USER aus (den Namen des aktuellen Unix-Accounts).

touch

Beschreibung: Erzeugt eine neue leere Datei oder ändert Zeitstempel (Zugriffs- und Änderungszeit) bestehender Dateien. Wenn die Datei nicht existiert, wird sie (sofern nicht -c verwendet wird) als leere Datei angelegt[man7.org](#). Oft genutzt, um Dateien schnell anzulegen oder deren *timestamp* zu aktualisieren.

Syntax:

touch [OPTION]... DATEI...

Optionen:

| Option | Beschreibung |
|------------------------------|---|
| -a | Ändert nur die Zugriffszeit (atime) der Datei man7.org . |
| -m | Ändert nur die Änderungszeit (mtime) der Datei man7.org . |
| -c, --no-create | Legt keine neue Datei an. Wenn die Datei nicht existiert, wird kein neues leeres File erstellt (ansonsten Standardverhalten ohne -c) man7.org man7.org . |
| -d, --date=STRING | Verwendet den angegebenen Datums-String anstelle der aktuellen Zeit. Der String wird geparsst (z.B. "2025-05-01 12:00:00" oder relative Angaben wie "next Friday" sind möglich) man7.org man7.org . |
| -t | Verwendet einen konkreten Zeitstempel im angegebenen Format (JJMMTThhmm[.ss]) anstatt der aktuellen [[CC]YY]MMDDhhmm[.ss] Zeit man7.org . Beispiel: -t 202312312359.59 setzt 31.12.2023 23:59:59. |
| -r, --reference=DATEI | Übernimmt die Zeitstempel (Zugriffs- und Änderungszeit) der angegebenen Referenzdatei. Die Zielfile erhält also dieselben Zeiten wie DATEI man7.org . |

| Option | Beschreibung |
|-----------------------------|---|
| -h, --no-dereference | Ändert bei symbolischen Links den <i>Link selbst</i> (deren Metadaten), nicht die referenzierte Datei. Ohne diese Option würde touch auf den Ziel-Dateipfad wirken (nicht auf den Symlink). (<i>Diese Option ist nur wirksam, wenn das System das Verändern von Symlink-Zeitstempeln unterstützt.</i>) man7.org |
| --time=WORD | Bestimmt, welcher Zeitstempel geändert wird. Mögliche Werte: atime, access, use (entspricht -a), oder mtime, modify (entspricht -m) man7.org . |
| --help, --version | Anzeigen der Hilfe bzw. Versionsinformation zu touch. |

man (Manual Page Viewer)

Beschreibung: Zeigt die Handbuchseite (Manual Page) für einen angegebenen Befehl oder ein Thema an. man ist das Offline-Hilfe- und Dokumentationssystem unter Unix/Linux.

Syntax:

man [SECTION] <Thema/Command>

Beispiel: man ls zeigt die Manpage zu **ls**. Manpages sind meist nach Kategorien/Abschnitten (1=Benutzerkommandos, 5=Dateiformate, 8=Administrator-Kommandos etc.) gegliedert – optional kann man mit man 5 passwd z.B. direkt die Manpage zum Dateiformat **passwd** (Abschnitt 5) öffnen.

Optionen:

| Option | Beschreibung |
|--------------------|--|
| -k Suchwort | Durchsucht die Manpage-Kurzzusammenfassungen nach <i>Suchwort</i> (entspricht dem Befehl apropos). Beispiel: man -k network listet alle Manual-Topics auf, deren Beschreibung "network" enthält. |
| -f Befehl | Findet den Namen und Abschnitt einer Manpage zu <i>Befehl</i> (entspricht whatis). Beispiel: man -f passwd zeigt kurz "passwd (1) - change user password". |
| -a | Zeigt nacheinander alle Manpages an, die zum angegebenen Namen passen (wenn z.B. ein Name in mehreren Abschnitten existiert). |
| -P Pager | Legt ein anderes Pager-Programm fest (Standard ist less). Beispiel: man -P cat ls würde die ls-Manpage durch cat leiten. |

Option Beschreibung

--help, --version Zeigt Hilfe bzw. Version von man.

whatis

Beschreibung: Zeigt eine kurze Ein-Satz-Beschreibung zu einem Befehl an (frägt die Datenbank der *whatis*-Einträge ab). Es ist eine Kurzform von *man -f* und liefert die Synopsis aus der Manpage.

Syntax:

whatis <Befehl/Topic>

Optionen:

Option Beschreibung

(keine üblichen Optionen) *whatis* wird meist ohne Optionen verwendet. Es greift auf die gleichen Datenbanken wie *apropos/man -k* zu. Sollte die Datenbank nicht aktuell sein, kann sie mit *mandb* erneuert werden.

Benutzer- und Gruppenverwaltung

passwd (Passwort ändern/setzen)

Beschreibung: Ändert das Passwort eines Benutzerkontos oder verwaltet Konto-Sperren. Als normaler Benutzer kann man mit *passwd* sein eigenes Passwort ändern; als *root* kann man mit *passwd <user>* das Passwort eines anderen Accounts setzen oder bestimmte Eigenschaften (Sperren, Ablaufdatum) ändern.

Syntax:

passwd [OPTION] [BENUTZER]

Standardaufruf ohne Benutzer wirkt auf den aktuellen Benutzer. Bei Angabe eines Benutzernamens werden dessen Passwortinformationen geändert (Root-Rechte erforderlich).

Optionen:

Option Beschreibung

-l, --lock Sperrt das Benutzerkonto, indem das Passwort in */etc/shadow* mit einem Präfix (meist ! oder *) unbrauchbar gemacht wird wiki.ubuntuusers.de. Der Benutzer kann sich danach nicht mehr anmelden, bis das Konto wieder entsperrt wird.

Option Beschreibung

- u, -- unlock Entsperrt ein zuvor gesperrtes Konto (entfernt das Sperr-Präfix im verschlüsselten Passwort)wiki.ubuntuusers.de.
- d, -- delete Löscht das Passwort des Benutzers (leer setzen). Achtung: Damit wird eine passwortlose Anmeldung möglich, sofern nicht andersweitig gesichertwiki.ubuntuusers.de.
- e, -- expire Markiert das Passwort als abgelaufen. Beim nächsten Login muss der Benutzer ein neues Passwort setzen.
- x *TAGE* Setzt die maximale Gültigkeitsdauer eines Passworts auf *TAGE*. Nach Ablauf muss es geändert werden. (Änderung des PASS_MAX_DAYS in /etc/shadow.)
- n *TAGE* Setzt die minimale Anzahl an Tagen zwischen zwei Passwortänderungen (PASS_MIN_DAYS). Der Benutzer kann sein Passwort erst nach *TAGE* Tagen erneut ändern.
- w *TAGE* Warnzeit in Tagen vor Passwortablauf (PASS_WARN_AGE). Der Benutzer erhält *TAGE* Tage vor Ablauf eine Warnung beim Login.
- s Zeigt den Passwortstatus an (Gesperrt? nicht gesetzt? Datum der letzten Änderung usw.). passwd -S <user> gibt eine Statuszeile aus.

Interaktives Verhalten: Wird passwd ohne Optionen aufgerufen, fordert es zur Eingabe des aktuellen Passworts (zur Authentifizierung, falls nicht root) und zweimal zur Eingabe des neuen Passworts auf (zwecks Bestätigung). Es verweigert einfache Passwörter gemäß der System-Policy (Pluggable Authentication Modules/pam Richtlinien). Root-Benutzer muss das alte Passwort nicht kennen.

chage (Change Age – Passwortalter und Gültigkeit)

Beschreibung: Verändert Einstellungen zum Passwort-Aging – also Ablaufdatum, Mindestalter, Warnzeit etc. Mit chage kann ein Administrator festlegen, wann ein Benutzer sein Passwort ändern muss bzw. wann der Account deaktiviert wirdfile-4fdk7rvx39azjretinarqb.

Syntax:

chage [OPTION]... BENUTZER

Optionen:

Option Beschreibung

| | |
|----------------------|--|
| | Listet die aktuellen Passwort- und Alterseinstellungen für den Benutzer in einem lesbaren Format auf (Letzte Änderung, Min/Max Alter, Warnzeit, Inaktivzeit, Ablaufdatum) – Übersicht. |
| -l | |
| -m TAGE | Setzt die minimale Anzahl von Tagen <i>TAGE</i> , die ein Passwort gültig sein muss, bevor es geändert werden darf (PASS_MIN_DAYS). |
| -M TAGE | Setzt die maximale Gültigkeitsdauer in Tagen (PASS_MAX_DAYS). Nach <i>TAGE</i> Tagen läuft das Passwort ab und muss geändert werden. |
| -w TAGE | Setzt die Anzahl der Tage <i>TAGE</i> Vorwarnzeit vor Passwortablauf (PASS_WARN_AGE). |
| -I TAGE | Legt fest, wie viele Tage nach Passwortablauf der Account inaktiv wird (PASS_INACTIVE_DAYS). Ein Wert von 0 deaktiviert den Account sofort nach Ablauf des Passworts, -1 schaltet diese Funktion aus wiki.ubuntuusers.de . |
| -E DATUM | Setzt ein Ablaufdatum für das Benutzerkonto (im Format YYYY-MM-DD). Nach diesem Datum wird der Account deaktiviert wiki.ubuntuusers.de . Das Datum -1 (oder "" leer) entfernt ein gesetztes Ablaufdatum (kein Ablauf). |
| -d YYYY-MM-DD | Setzt das Datum der letzten Passwortänderung auf das angegebene Datum (wirkt wie Backdating oder um zu triggern, dass ein Passwort abläuft). Kann auf 0 gesetzt werden, um das Passwort als sofort ablaufend zu markieren. |
| -h, --help | Zeigt Hilfe an. |

Hinweis: Viele dieser Einstellungen lassen sich auch mit passwd (Optionen -x, -n, -w, -i, -e) vornehmen. chage bietet eine dedizierte Schnittstelle dafür. Ohne Optionen (als root) startet chage <user> interaktiv einen Dialog, um alle Werte einzustellen.

w

Beschreibung: Zeigt an, welche Benutzer gerade am System angemeldet sind und was sie gerade tun. w fasst Informationen über aktuell eingeloggte User und deren laufende Prozesse zusammen (inkl. Login-TTY, Leerlaufzeit, genutzte CPU-Zeit, aktueller Befehl)
file-4fdk7rvx39azjretinarqb.

Syntax:

w [OPTION]

Optionen:

Option Beschreibung

| | |
|----------------------|---|
| -h | Blendet die Kopfzeile in der Ausgabe aus (unterdrückt die Überschriftenzeile). |
| -s | Kurze Ausgabe. Unterdrückt die Spalten für Login-Zeit, Leerlauf etc. (nicht auf allen Systemen vorhanden). |
| -f | Zeigt Informationen zu von utmp als <i>geflagt</i> markierten Einträgen. Auf einigen Systemen steuert dies die Anzeige von Remote-Host oder Exit-Informationen. (Bei neueren procps-Versionen hat -f keine Funktion, da Host immer gezeigt wird.) |
| -V, --version | Zeigt Versionsinfo von w. |

Die Ausgabe von w enthält in der ersten Zeile System-Uptime und Load Average. Danach folgen pro eingeloggtem Benutzer: Benutzername, Terminal, Remote-Host, Login-Zeit, Leerlaufdauer, CPU-Verbrauch und der aktuell laufende Prozess (TTY-Prozess).

who

Beschreibung: Zeigt Informationen über die aktuell angemeldeten Benutzer. Im Gegensatz zu w liefert who primär Login-Zeitpunkte und Terminalsessions, jedoch keine Prozess- oder Idle-Infos.

Syntax:

who [OPTION]... [Datei]

Standard liest who die System-Login-Datenbank /var/run/utmp und gibt für jeden Eintrag Benutzer, Terminal, Datum/Zeit und ggf. Host aus.

Optionen:

Option Beschreibung

| | |
|-----------|--|
| -H | Zeigt eine Kopfzeile (Überschriften) über den Spalten an. |
| -q | "Quick who": Zeigt nur Benutzernamen und die Anzahl der angemeldeten Nutzer. (Gleichbedeutend mit users-Befehl.) |
| -b | Zeigt die Zeit des letzten Systemstarts (Boot time) an. |
| -r | Zeigt den aktuellen Runlevel des Systems. |

| Option | Beschreibung |
|--------------------------|---|
| -m | Zeigt nur die Einträge für das aktuelle Terminal an (beschränkt auf den Benutzer, der den Befehl aufruft). |
| -u | Zeigt zusätzlich die Leerlaufzeit (Idle) und den PID des Login-Prozesses für jeden Benutzer. |
| -T | Zeigt den Terminal-Status (\pm Flag) an: + bedeutet terminal ist bei mesg schreibbar, - bedeutet nicht schreibbar, ? unbekannt. (In vielen who-Implementierungen bereits Standardverhalten ohne Option.) |
| --help, --version | Anzeigen von Hilfe bzw. Versionsinfo. |

useradd (Benutzer anlegen)

Beschreibung: Legt ein neues Benutzerkonto an. useradd ist ein Low-Level-Befehl, der die Systemkontodateien (wie /etc/passwd, /etc/shadow, /etc/group) direkt verändert, um einen neuen User einzurichten [pq.hosting](#). In Debian/Ubuntu-Umgebungen wird oft das Wrapper-Skript adduser verwendet, das auf useradd aufbaut und interaktiv ein paar Dinge abfragt; useradd selbst ist nicht interaktiv und benötigt alle Optionen zum Festlegen der Eigenschaften des neuen Accounts.

Syntax:

useradd [OPTION]... BENUTZERNAME

Optionen: (häufigste Optionen beim Anlegen eines Benutzers)

| Option | Beschreibung |
|-----------------|---|
| -c | Kommentarfeld (GECOS) für den Benutzer, z.B. voller Name oder Hinweis.
<i>Kommentar</i> Wird in /etc/passwd eingetragen wiki.ubuntuusers.de . |
| -d HOME | Pfad des Home-Verzeichnisses. Standard: /home/<Benutzername>. Kann geändert werden; wenn -m verwendet wird, wird dieses Verzeichnis erstellt und mit Standard-Dateien aus /etc/skel befüllt wiki.ubuntuusers.de . |
| -m | Legt das Home-Verzeichnis an und kopiert Standard-Dateien hinein. Sollte gesetzt sein, außer man möchte <i>kein</i> Home-Verz. (Ohne -m wird das Home nicht automatisch erstellt) wiki.ubuntuusers.de . |
| -s SHELL | Login-Shell des Benutzers. Standard ist normalerweise /bin/bash (oder was in /etc/default/useradd als SHELL definiert ist). Kann z.B. auf /sbin/nologin gesetzt werden für System-Accounts wiki.ubuntuusers.de . |

| Option | Beschreibung |
|---------------------------|--|
| -u <i>UID</i> | Gewünschte User-ID (numerisch). Darf nicht bereits vergeben sein (außer mit -o). Wenn nicht angegeben, wird die nächste freie UID > 999 gewählt wiki.ubuntuusers.de . |
| -o | Erlaubt die Verwendung einer bereits vergebenen UID (UID-Doppelungen). Zusammen mit -u benutzt, um zwei Benutzer mit identischer UID anzulegen. (Normalerweise nicht empfohlen, da UID eindeutig sein sollen.) |
| -g <i>GRP</i> | Primäre Gruppe (Name oder GID), der der Benutzer angehört wiki.ubuntuusers.de . Standard: eine neue Gruppe mit gleichem Namen wie der Benutzer (bei entsprechender Systemkonfiguration) oder eine in /etc/default/useradd definierte GROUP. Die angegebene Gruppe muss existieren. |
| -G <i>Liste</i> | Liste zusätzlicher Gruppen (sekundäre Gruppen), denen der Benutzer angehören soll wiki.ubuntuusers.de . Gruppennamen durch Komma getrennt angeben (ohne Leerzeichen). |
| -p <i>Passwort</i> | Das verschlüsselte Passwort für den Account (wie es z.B. von openssl passwd oder crypt(3) erzeugt würde) wiki.ubuntuusers.de . Wichtig: Aus Sicherheitsgründen sollte man hier <i>nicht</i> ein Klartextpasswort übergeben. Bleibt -p leer, wird der Account ohne gesetztes Passwort angelegt (Anmeldung evtl. gesperrt bis passwd gesetzt wird). |
| -e | (bei useradd nicht üblich – wird bei usermod benutzt für Ablauf setzen. Bei einigen Systemen könnte useradd -e <Datum> ein Ablaufdatum setzen.) |
| -f <i>TAGE</i> | Anzahl an Tagen nach Passwortablauf, bis der Account deaktiviert wird (Analog zu chage -l). -f 0 würde den Account sofort mit Passwortablauf sperren. Standard -1 (nie sperren). |
| -N | (Bei einigen Systemen: <i>No user group</i> – verhindert das automatische Anlegen einer gleichnamigen Gruppe.) |
| -r | Erstellt einen System-Account (UID unter einer gewissen Schwelle, z.B. <1000, kein Expire, kein Mail-Spool). Oft genutzt für Systembenutzer ohne Login. |
| -M | Kein Home-Verzeichnis anlegen (selbst wenn Default CREATE_HOME aktiviert ist). |

| Option | Beschreibung |
|-------------------|---|
| -k SKELDIR | Gibt ein alternatives Skeleton-Verzeichnis an (anstatt /etc/skel), von wo aus bei -m die initialen Dateien ins Home kopiert werden. |
| -D | Zeigt die aktuellen Standardwerte für useradd an (oder mit anderen -D-Optionen können diese verändert werden). |

Nach dem Anlegen mit useradd sollte meist ein Passwort gesetzt werden (passwd <user>), außer es handelt sich um einen Account ohne Login.

usermod (Benutzerkonto ändern)

Beschreibung: Modifiziert ein bestehendes Benutzerkonto – z.B. Name, UID, Primärgruppe, zusätzliche Gruppen, Login-Shell, Expire-Datum usw.wiki.ubuntuusers.dewiki.ubuntuusers.de. Im Grunde die gleichen Optionen wie useradd, aber angewendet auf bestehende Benutzer. Häufig genutzt, um Benutzer umzubenennen, in andere Gruppen aufzunehmen oder Accounts zu sperren.

Syntax:

usermod [OPTION]... LOGIN

Optionen:

| Option | Beschreibung |
|-----------------------|---|
| -c Kommentar | Neuen Kommentar (GECOS-Feld) setzen wiki.ubuntuusers.dewiki.ubuntuusers.de . |
| | Ändert das Home-Verzeichnis des Benutzers auf <i>Neues_Home</i> . |
| -d Neues_Home | Hinweis: Ohne weiteres passiert das alte Home nicht automatisch mit. Mit zusätzlicher Option -m werden die Inhalte des alten Home-Verz. ins neue Verzeichnis verschoben und das neue Verzeichnis ggf. erstellt wiki.ubuntuusers.dewiki.ubuntuusers.de . |
| -m | (Mit -d verwenden) Verschiebt beim Ändern des Homepfads (-d) die Dateien vom alten ins neue Home und erstellt es falls nötig wiki.ubuntuusers.dewiki.ubuntuusers.de . |
| -l Neuer_Login | Ändert den Login-Namen des Benutzers (Umbenennung des Accounts) wiki.ubuntuusers.dewiki.ubuntuusers.de . Der Benutzername in allen Systemkontodateien wird angepasst, <i>nicht</i> jedoch automatisch der Name des Home-Verzeichnisses oder Mail-Spool (diese müssten manuell umbenannt werden, falls gewünscht). |
| -u Neue_UID | Ändert die numerische Benutzer-ID auf <i>Neue_UID</i> . Falls die UID bereits vergeben ist, muss -o benutzt werden. Dateien im Home-Verzeichnis mit alter UID werden auf neue UID umgestellt; Dateien |

| Option | Beschreibung |
|-------------------------------------|--|
| | außerhalb des Homes muss der Admin selbst anpassen wiki.ubuntuusers.dewiki.ubuntuusers.de . |
| -o | Erlaubt die Verwendung einer bereits vergebenen UID (nur zusammen mit -u sinnvoll). |
| -g
<i>Neue_PrimGruppe</i> | Ändert die primäre Gruppenzugehörigkeit (Name oder GID) manpages.ubuntu.com . Die neue Gruppe muss existieren. Alle Dateien im Home, die der alten primären Gruppe gehörten, werden auf die neue Gruppe übertragen manpages.ubuntu.com . (Dateien außerhalb des Homes müssen ggf. manuell angepasst werden.) |
| -G Gruppe(n) | Setzt die Liste der sekundären Gruppen neu für den Benutzer manpages.ubuntu.com . <i>Wichtig:</i> Ohne -a entfernt diese Option den Benutzer aus allen Gruppen, die nicht in der Liste stehen wiki.ubuntuusers.demanpages.ubuntu.com . Um einen Benutzer zusätzlichen Gruppen hinzuzufügen, sollte man -a -G gemeinsam verwenden (siehe -a). |
| -a | Append – fügt den Benutzer zusätzlichen Gruppen hinzu, anstatt die Gruppenzugehörigkeit zu überschreiben wiki.ubuntuusers.dewiki.ubuntuusers.de . Nur in Kombination mit -G verwenden , um den Benutzer der/die angegebenen Gruppen hinzuzufügen, ohne ihn aus anderen zu entfernen. (Siehe Warnung unten.) |
| -s Shell | Ändert die Login-Shell des Benutzers wiki.ubuntuusers.de . Z.B. auf /sbin/nologin setzen, um interaktive Logins zu unterbinden. |
| -e YYYY-MM-DD | Setzt das Ablaufdatum des Accounts (das Datum, an dem das Konto deaktiviert wird) wiki.ubuntuusers.de . Nach diesem Datum kann sich der User nicht mehr anmelden. usermod -e "" user entfernt ein Ablaufdatum (setzt es auf <i>nie</i>). |
| -f Tage | Setzt die Anzahl an Tagen nach Passwortablauf, bis der Account deaktiviert wird wiki.ubuntuusers.de . -f 0 deaktiviert sofort nach Ablauf; -f -1 entfernt eine evtl. gesetzte Deaktivierung (nie sperren nach Ablauf). |
| -L | Sperrt das Benutzerkonto (lock). Fügt ein ! dem Passwort-Hash voran (im Shadow-Eintrag) wiki.ubuntuusers.de , was die Anmeldung verhindert. Kann nicht mit -p oder -U kombiniert werden. |

| Option | Beschreibung |
|----------------|--|
| -U | Entsperrt ein zuvor gesperrtes Passwort (passwd -l). Entfernt das ! im Shadow-Passwortfeld, sodass der ursprüngliche Hash wieder wirksam wird wiki.ubuntuusers.de . Nicht zusammen mit -L oder -p verwenden. |
| -p Hash | Setzt das <i>verschlüsselte</i> Passwort (Hash) des Users auf den angegebenen Wert wiki.ubuntuusers.de . Sollte mit Vorsicht genutzt werden – für Passwortänderung ist passwd meist besser geeignet. |
| --help | Hilfe anzeigen. |

Wichtige Hinweise:

- **Warnung:** Die Kombination von -G ohne -a überschreibt die Gruppenliste vollständig. Das heißt, der Benutzer wird aus allen Gruppen entfernt, die nicht explizit bei -G genannt sind[wiki.ubuntuusers.de](#). Um einen User nur hinzuzufügen, niemals ohne -a verwenden!
- Man kann keinen angemeldeten Benutzer umbenennen oder dessen UID ändern, solange er eingeloggt ist oder laufende Prozesse hat. Zudem keine Änderungen an dem eigenen Account durchführen, während man noch mit diesem angemeldet ist (kann zu Inkonsistenzen führen).
- Beim Ändern des Benutzernamens oder UIDs sollten evtl. Crontab-Dateien und geplante at-Jobs manuell umbenannt/angepasst werden, und in NIS/LDAP-Umgebungen müssen Änderungen dort vorgenommen werden[wiki.ubuntuusers.de](#).

userdel (Benutzer löschen)

Beschreibung: Löscht ein Benutzerkonto aus dem System (entfernt die Einträge in /etc/passwd, /etc/shadow und ggf. /etc/group falls der Benutzer eine eigene Gruppe hatte). Optional können auch das Home-Verzeichnis und die Mail-Spool des Benutzers gelöscht werden.

Syntax:

userdel [OPTION] BENUTZER

Optionen:

Option Beschreibung

| | |
|-----------|---|
| -r | Entfernt <i>rekursiv</i> das Heimatverzeichnis des Benutzers und seine Mail-Spool (üblicherweise /var/spool/mail/<user>), sowie alle Dateien darin, beim Löschen |
|-----------|---|

Option Beschreibung

des Accounts. Ohne -r bleiben Home und Mails unangetastet file-4fdk7rvx39azjretinarqb.

-f *Force:* Erzwingt das Löschen des Accounts, selbst wenn der Benutzer noch angemeldet ist oder laufende Prozesse hat. (Warnung: kann zu inkonsistenten Systemzuständen führen, Prozess bleibt ohne Benutzer etc. Diese Option sollte vermieden werden.)

Ohne Option wird der Account aus den Systemdateien gelöscht, aber das (keine) Home-Verzeichnis, die Dateien und Mails bleiben bestehen. Admin muss sie ggf. manuell entfernen oder anderweitig zuordnen.

Hinweis: userdel bricht ab, wenn der Benutzer gerade angemeldet ist (außer mit -f). Es ist ratsam, vor dem Löschen sicherzustellen, dass keine Prozesse mehr unter der betreffenden UID laufen (z.B. mit pkill -u username beenden). Dateien außerhalb des Home, die dem Benutzer gehören, verbleiben auf dem Dateisystem mit der nun unbenutzten UID – diese müssten manuell gesucht und entfernt oder neu zugeordnet werden (z.B. mit find / -uid <oldUID>).

id

Beschreibung: Zeigt die Benutzer- und Gruppen-Identitäten eines Benutzerkontos an. Standardmäßig gibt id die UID, primäre GID und zugehörigen Gruppennamen des aktuellen Benutzers zurück file-4fdk7rvx39azjretinarqb. Mit Angabe eines Benutzernamens kann man die IDs für einen anderen Account abrufen.

Syntax:

id [OPTION]... [BENUTZER]

Optionen:

Option Beschreibung

(ohne) Ohne Options werden UID, Benutzername, GID, Gruppenname und alle zusätzlichen Gruppen des Benutzers aufgelistet (z.B. uid=1000(max) gid=1000(max) Gruppen=1000(max),27(sudo),...).

-u Gibt nur die effektive User-ID (UID, numerisch) aus. Beispiel: id -u alice könnte 1001 ausgeben.

-g Gibt nur die effektive Gruppen-ID (GID, numerisch) aus.

-G Gibt alle Gruppen-IDs (numerisch) aus, in denen der Benutzer Mitglied ist (Primär- und Sekundärgruppen).

Option Beschreibung

| | |
|--------------------------|---|
| -n | Gibt Namen anstelle von numerischen IDs. Kann in Kombination mit -u, -g, -G verwendet werden. Z.B. id -Gn alice listet alle Gruppennamen von alice. |
| -r | Zeigt die reale UID/GID an (falls z.B. mit sudo aufgerufen, reale vs. effektive ID unterscheiden). |
| -Z | Zeigt SELinux-Sicherheitskontext, falls SELinux aktiv ist (z.B. unconfined_u:unconfined_r:unconfined_t:s0 für einen unconfined Benutzer). |
| --help, --version | Hilfe bzw. Version anzeigen. |

groupadd (Gruppe anlegen)

Beschreibung: Legt eine neue Gruppe im System an. Fügt einen Eintrag in /etc/group (und ggf. /etc/gshadow) hinzu.

Syntax:

groupadd [Optionen] GRUPPENNAME

Optionen:

Option Beschreibung

| | |
|---------------------------|---|
| -g GID | Verwendet die angegebene numerische GID für die neue Gruppe. Sollte noch nicht vergeben sein (oder mit -o überschrieben werden). |
| -o | Erlaubt die Verwendung einer bereits existierenden GID (zusammen mit -g zu nutzen, um GID-Doppel zu erzwingen). |
| -f | "Force": falls die Gruppe schon existiert, keinen Fehler melden; falls die angegebene GID bereits vergeben ist, wählt eine andere (ignoriert -o). |
| -r | Erstellt eine Systemgruppe (GID wird aus dem System-Bereich gewählt, z.B. <1000). |
| -K Schlüssel= Wert | Überschreibt einen Eintrag aus /etc/login.defs temporär (z.B. -K GID_MIN=500 um untere GID-Bereiche anzupassen). |
| -p Passwort | Setzt das (verschlüsselte) Gruppenpasswort direkt. (Gruppenpasswörter werden selten genutzt – ermöglicht mit newgrp temporär in eine Gruppe zu wechseln.) |

groupmod (Gruppe ändern)

Beschreibung: Ändert Eigenschaften einer bestehenden Gruppe – hauptsächlich Name oder GID.

Syntax:

groupmod [Optionen] GRUPPE

Optionen:

| Option | Beschreibung |
|---------------------|---|
| -n <i>NeuerName</i> | Ändert den Gruppennamen in <i>NeuerName</i> . Der alte Name in allen relevanten Systemdateien wird ersetzt. |
| -g <i>NeueGID</i> | Ändert die GID der Gruppe. Muss eindeutig sein (oder mit -o doppelt erlaubt). Dateien mit alter GID außerhalb, die nicht automatisch erfasst werden, müssen eventuell manuell angepasst werden. |
| -o | Erlaubt das Setzen einer bereits vergebenen GID (nur mit -g relevant). |
| -p <i>Passwort</i> | Ändert das (verschlüsselte) Gruppenpasswort. (Entspricht Editieren von /etc/gshadow). |

groupdel (Gruppe löschen)

Beschreibung: Entfernt eine Gruppe aus der Systemgruppe-Datei. Die Gruppe wird aus /etc/group (und /etc/gshadow) getilgt.

Syntax:

groupdel GRUPPENNAME

Hinweis: Dieser Befehl hat keine besonderen Optionen – er löscht die Gruppe, sofern sie nicht als Primärgruppe eines Benutzers in /etc/passwd noch verwendet wird. Man kann keine Gruppe löschen, die noch als Hauptgruppe eines existierenden Benutzerkontos eingetragen ist; in dem Fall muss man ggf. erst den Benutzer anpassen oder löschen. Dateien auf dem System, die dieser Gruppe gehören, behalten die nun ungültige GID bei, bis man sie manuell ändert.

su (Switch User)

Beschreibung: Wechselt den Benutzerkontext im laufenden Terminal. Mit su („substitute user“) kann man einen anderen Benutzeraccount annehmen – standardmäßig root, wenn kein Benutzer angegeben wird. Dies erfordert die Eingabe des Zielbenutzer-Passworts (außer beim Wechsel zu root, dort abhängig von den Systemeinstellungen – oft ist root-Login per Passwort deaktiviert und man nutzt sudo).

Syntax:

su [OPTION]... [BENUTZER [-]]

Wird BENUTZER weggelassen, versucht su auf den Superuser (root) zu wechseln.

Optionen:

| Option | Beschreibung |
|-------------------|--|
| - oder -l | Startet eine <i>Login-Shell</i> der Zielusers, d.h. die Umgebung wird vollständig umgeschaltet als würde sich der Benutzer frisch anmelden. Das Home-Verzeichnis, Pfad (\$PATH) etc. werden übernommen. Beispiel: su - alice wechselt zu Benutzer <i>alice</i> mit deren Umgebung (äquivalent zu vollem Logout/Login als <i>alice</i>). |
| -c Befehl | Führt einen angegebenen <i>Befehl</i> als Zielbenutzer aus und kehrt dann zurück. Beispielsweise su -c "systemctl restart apache2" - root führt als root einen Neustart des Apache durch und beendet sich sofort wieder. |
| -m, -p | Behalten die aktuelle Umgebung (<i>nicht</i> auf Ziel-User umschalten). Der aktuelle \$HOME, \$PATH, usw. bleiben erhalten, nur die Rechte wechseln. Standardverhalten ohne -l. (In vielen su Implementierungen ist -m identisch mit -p: <i>preserve environment</i>). |
| -s Shell | Startet die angegebene <i>Shell</i> statt der Standard-Login-Shell des Zielbenutzers. Der Pfad zur Shell muss im /etc/shells stehen, sofern nicht root. |
| -g Gruppe | Wechselt in Kombination mit Benutzer auch die primäre Gruppe auf <i>Gruppe</i> . (Nur verfügbar in neueren util-linux su). |
| -<Ziffer> | (BSD su) Angabe einer Session-Klasse – unter Linux nicht genutzt. |
| --help, --version | Anzeigen von Hilfe bzw. Version. |

Hinweise:

- su ohne - (also z.B. su root) wechselt zwar die Benutzer-ID, aber behält das aktuelle Verzeichnis und weite Teile der Environment von Ursprungskonto bei (\$LOGNAME, \$HOME bleiben z.B. vom ursprünglichen Nutzer). Das kann zu irritierenden Effekten führen (z.B. Schreibrechte im falschen Home). Daher nutzt man meist su - für einen *sauberen* Wechsel.
- Viele Distributionen konfigurieren aus Sicherheitsgründen su so, dass ein direkter Wechsel zu *root* per Passwort entweder deaktiviert ist oder kein root-Passwort gesetzt ist. Stattdessen wird sudo bevorzugt, um Root-Rechte zu erlangen.

sudo (Superuser Do)

Beschreibung: Führt ein Kommando mit *erhöhten Rechten* aus, typischerweise als *root*, ohne die Benutzeridentität dauerhaft zu wechseln. Benutzer können mit sudo Befehle

mit Administratorrechten ausführen, sofern sie in /etc/sudoers entsprechend autorisiert sind. Im Gegensatz zu su wird hier das eigene Passwort abgefragt (nicht das des Zielusers), und nur spezifische Befehle/Optionen sind erlaubt gemäß Konfiguration.

Syntax:

sudo [OPTION]... [COMMAND]

Wird ein Befehl angegeben, führt sudo diesen als root (Standard) aus. Ohne Befehl zeigt sudo ggf. die erlaubten Befehle an oder wechselt in eine sudo-Shell (bei -s).

Optionen (Auswahl):

| Option | Beschreibung |
|--------------------|---|
| -l | Listet die erlaubten (und verbotenen) Befehle für den aufrufenden Benutzer auf (nach sudoers-Regeln). Hilfreich, um zu sehen, was man mit sudo ausführen darf. |
| -u <i>Benutzer</i> | Führt das angegebene Kommando als <i>Benutzer</i> aus (anstatt als root). Beispiel: sudo -u alice whoami zeigt "alice", wenn man als anderer User ausführt. Standard ist root. |
| -s | Startet eine Shell als Zielbenutzer (Standard root). Ähnlich wie su, aber mittels sudo-Rechte. Es wird die im \$SHELL oder in passwd eingetragene Shell geöffnet. |
| -i | Simuliert eine vollständige Login-Shell wie su - (initialisiert Umgebung des Zielusers). Entspricht sudo -s plus Login-Umgebung. |
| -b | Führt den Befehl im Hintergrund aus (<i>background</i>). Sudo verlangt dennoch ggf. ein Passwort, aber detach danach. |
| -E | Beibehaltung der aktuellen Environment-Variablen (normalerweise bereinigt sudo einige Variablen aus Sicherheitsgründen). Erlaubt das Mitnehmen der eigenen Umgebung. (Ob das erlaubt ist, kann in sudoers eingeschränkt sein.) |
| -p <i>Prompt</i> | Definiert einen eigenen Passwort-Prompt. z.B. sudo -p "Passwort für %u:" -v (Platzhalter %u = Username). |
| -v | "Validate": Aktualisiert/überprüft das eigene sudo-Authentifizierungsticket. D.h. fordert ggf. Passwort an, speichert das sudo Token (5min Standard) aber führt sonst nichts aus. (Verlängert die Zeit, in der man kein Passwort erneut eingeben muss.) |
| -k | Invalidiert das eigene sudo-Ticket sofort (nachfolgende sudo-Aufrufe verlangen wieder Passwort) – (<i>kill credential</i>). |

Option Beschreibung

| | |
|--|---|
| | Setzt \$HOME in der Environment auf das Home des Zielusers (root). |
| -H | Normalerweise belässt sudo \$HOME vom Aufrufer; mit -H verhält es sich wie su -. |
| --help, -h Hilfe anzeigen (kurze Übersicht der Optionen). | |
| -K | Löscht das Zeitstempel-Ticket komplett (ähnlich -k aber gründlicher, forciert Passwort prompt beim nächsten Mal). |
| --version | Versionsinfo ausgeben (auch Liste der unterstützten Sicherheitsmechanismen etc.). |

Hinweise: Bei erfolgreicher Authentifizierung legt sudo einen Zeitstempel (Ticket) an, sodass für nachfolgende Aufrufe (Standard ~5 Minuten) kein erneutes Passwort nötig ist. Dies kann per sudoers Einstellung (timestamp_timeout) angepasst oder mit sudo -k/-K beeinflusst werden.

pkexec (PolicyKit Execute)

Beschreibung: Führt ein Programm als ein anderer Benutzer aus (ähnlich wie sudo, aber Teil von PolKit/PolicyKit). Standardmäßig wird, sofern nicht anders angegeben, das Programm mit root-Rechten gestartetfreedesktop.org. pkexec erfordert, dass der aufrufende Benutzer berechtigt ist, die Aktion auszuführen (PolKit Policies); falls ja, wird eine Authentifizierungsdialog (grafisch oder textbasiert) angezeigt, um ggf. das Passwort einzugeben. Nach erfolgreicher Auth kann das gewünschte Kommando mit erhöhten Rechten ausgeführt werden.

Syntax:

pkexec [--user <Benutzer>] <Kommando> [ARG...]

Optionen:

Option Beschreibung

| | |
|--------------------------|---|
| --user
<Benutzer> | Führt das angegebene Programm als Benutzer aus. Standard ist root (Administratorkonto) freedesktop.org . |
| --disable-internal-agent | Unterdrückt den internen Authentifizierungs-Agent von pkexec. Normalerweise, falls kein grafischer PolKit-Auth-Agent verfügbar ist, startet pkexec einen eigenen textuellen Auth-Dialog im Terminal. Mit dieser Option wird in so einem Fall statt dessen das Programm nicht ausgeführt (Fehler, wenn kein Agent aktiv) freedesktop.org . |

| Option | Beschreibung |
|--------|--------------|
|--------|--------------|

| | |
|--------------------|--------------------------------------|
| --help / --version | Zeigt Hilfe bzw. Version von pkexec. |
|--------------------|--------------------------------------|

Hinweise: pkexec erbt ein Minimal-Umfeld (Environment), um das Einschleusen von Code via Umgebungsvariablen zu verhindern freedesktop.org. Display- und Xauth-Variablen werden aus Sicherheitsgründen *nicht* an das Zielprogramm weitergegeben, außer es wurde in den PolKit-Regeln explizit erlaubt (GUI-Programme via pkexec sind daher tricky ohne entsprechende PolKit-Policy). In modernen Distributionen ist pkexec oft als Ersatz für grafische gksu-Aufrufe gedacht, um z.B. aus der Desktop-Umgebung heraus GUI-Administrationsprogramme mit Root-Rechten zu starten (wobei PolKit dann i.d.R. das Benutzer-Passwort abfragt).

Berechtigungen und Zugriffskontrolle

umask (Set Default Permissions Mask)

Beschreibung: Legt eine Berechtigungsschutzmaske fest, die bei neu erstellten Dateien und Verzeichnissen die Standardzugriffsrechte beeinflusst. umask subtrahiert Bits von den vollen Zugriffsrechten. So bestimmt sie, welche Rechte *NICHT* automatisch gesetzt werden dürfen. Dies wirkt pro Prozess/Session. Als Shell-Builtin zeigt umask ohne Parameter die aktuelle Maske an.

Syntax:

umask [Option] [Maske]

Optionen / Verwendung:

- Ohne Argument zeigt umask die aktuelle Mask in Oktalnotation an (z.B. 0022). Mit Option **-S** in symbolischer Form (z.B. u=rwx,g=rx,o=rx entspricht Maske 022).
- Mit Angabe einer neuen Maske (z.B. umask 027) wird diese gesetzt. Die Maske wird meist oktal angegeben: jedes Bit, das in der Maske 1 ist, nimmt der neu erstellten Datei die entsprechende Berechtigung weg. Beispiel: Maske 022 bedeutet, neuen Dateien werden keine Gruppe- und Andere-Schreibrechte gegeben (d.h. max Resultatrechte für Datei typ. 644, für Verzeichnis 755).
- **Symbolische Maske:** Man kann auch wie bei chmod symbolisch angeben, z.B. umask u=rwx,g=rx,o=rx für dasselbe Ergebnis (Maske 022).

Hinweis: Die Umask wirkt auf Systemaufruf-Ebene: Standard-Dateirechte sind 666 (rw-rw-rw) und für Verzeichnisse 777 (rwxrwxrwx); die Maske wird davon abgezogen. Z.B. Umask 002 ergibt Dateien mit 664 (rw-rw-r-) und Verzeichnisse mit 775. Umask 077 würde sehr restriktiv alles nur für den Benutzer selbst erlauben (Datei 600, Dir 700).

chmod (Change Mode - Dateirechte ändern)

Beschreibung: Ändert die Zugriffsberechtigungen einer Datei oder eines Verzeichnisses. Man kann mit chmod sowohl numerisch (Oktalwerte) als auch symbolisch (u/g/o +-/= r,w,x) die Rechte setzen.

Syntax:

chmod [Optionen] <Modus> <Datei/Verzeichnis>...

Optionen:

| Option | Beschreibung |
|------------------------------------|---|
| -R, --recursive | Rekursive Rechteänderung: traversiert Unterverzeichnisse und wendet chmod auf alle darin befindlichen Dateien/Ordner an. Wichtig bei Verzeichnisbäumen. |
| -v, --verbose | Gibt für jede Datei eine Meldung aus, deren Rechte erfolgreich geändert wurden (und ggf. was geändert wurde). |
| -c, --changes | Wie verbose, aber nur Ausgaben für tatsächlich geänderte Rechte (keine Ausgabe, wenn Rechte schon so eingestellt waren). |
| -f, --silent | Unterdrückt Fehlermeldungen (force). |
| --reference=<Vorlage> | Setzt die gleichen Berechtigungen wie die Referenz-Datei <Vorlage> hat, auf die angegebenen Dateien. Statt Modus muss man --reference nutzen. |

Modus-Angaben:

- **Numerisch (Oktal):** Eine dreistellige (oder vierstellige mit Sonderbit) Oktalzahl. Z.B. chmod 640 datei.txt setzt Rechte auf rw- r-- --- (6=rw, 4=r, 0=kein Recht). Jede Ziffer entspricht Owner, Group, Others. Beispiel: chmod 755 script.sh => rwx für Besitzer (7), rx für Gruppe (5), rx für andere (5). Eine führende vierte Ziffer kann spezielle Bits setzen: 4 für Setuid, 2 für Setgid, 1 für Sticky-Bit (z.B. 1755 für Sticky).
- **Symbolisch:** Format: [ugoa]*[+-=][rwxXst]* evtl. durch Kommata getrennt für mehrere Operationen in einem Aufruf. Beispiele:
 - chmod u+x,g-w,o= file – dem Besitzer Ausführungsrecht geben, der Gruppe Schreibrecht entziehen, andere bekommen exakt keine Rechte.
 - chmod a+r file – allen (a für all) Leserechte hinzufügen (ohne andere Bits zu ändern).
 - chmod g=u Datei – setzt die Gruppenrechte gleich den Benutzerrechten der Datei (nützliche Kurzform).

- X: Spezialflag in symbolischer Notation – steht für "execute if directory or if any execute bit is set on any of owner/group/other". Wird meist bei -R verwendet, um rekursiv nur Verzeichnissen das x zu geben.
- s für setuid/setgid (in u/g Kontext), t für Sticky-Bit (im o-Kontext).
- Beispiel: chmod u+s script.sh – setzt das setuid-Bit auf die Datei (falls ausführbar).

chown (Change Owner)

Beschreibung: Ändert den Dateibesitzer und/oder die Gruppe einer Datei oder eines Verzeichnissesfile-4fdk7rvx39azjretinarqbfile-4fdk7rvx39azjretinarqb. chown kann sowohl den User als auch die Gruppe in einem Befehl setzen. Nur root (oder Capability CHOWN) darf Eigentümer von Dateien ändern. Normale Benutzer können in der Regel mittels chgrp die Gruppenzugehörigkeit ändern, wenn sie Mitglied der Zielgruppe sind (und Besitzer der Datei bleiben).

Syntax:

chown [Optionen] <neuerEigentümer>[:<neueGruppe>] DATEI...

Optionen:

| Option | Beschreibung |
|--|---|
| -R | Rekursiv – wendet die Besitzänderung auf alle Dateien und Unterverzeichnisse an (ganzer Baum). |
| -h | Ändert den Besitzer von Symbolic Links selbst, statt des Ziels. (Auf den meisten Linux ist das Standard mit Kernel 2.6+; teils wird -h ignoriert, da chown auf Symlinks ohnehin Link selbst ändert, außer mit -L Option). |
| -f | Keine Fehlermeldungen ausgeben (silent/force). |
| -v | Verbose – listet jede Datei mit geänderten Owner/Group auf. |
| --
from=<old_owner:>old_group> | Ändert nur, wenn aktuelle Owner/Group genau wie angegeben ist. (Damit kann man selektiv bestimmte Kombis ersetzen, dient als Sicherheitsprüfung). |
| --dereference / -L | <i>Follow symlinks:</i> Symbolische Links auflösen (statt Link selbst zu ändern wird das Ziel geändert). Standardverhalten für die meisten chown-Versionen ist bereits, dem Link <i>nicht</i> zu folgen (außer man setzt -L). |

| Option | Beschreibung |
|---------------|---|
| -P | <i>Don't follow symlinks</i> (explizit Link behandeln, Standard). |
| -H | Folgen von Symlinks <i>im Kommandozeilen-Argument</i> (Top-Level), aber nicht rekursiv tiefer. (Selten benötigt). |

Verwendung:

- Besitzer ändern: chown alice datei.txt – setzt *alice* als neuen Besitzer der Datei (Gruppe bleibt unverändert).
- Gruppe ändern mit chown: chown :developers projektX – behält Besitzer, setzt Gruppe auf *developers*. Notation: Vor dem Doppelpunkt nichts heißt Besitzer unverändert, aber *Gruppe* ändern.
- Besitzer und Gruppe zugleich: chown alice:staff datei.txt – Besitzer = *alice*, Gruppe = *staff*.
- Nur Gruppe ändern könnte auch mit eigenem Befehl chgrp staff datei.txt gemacht werden.

Besonderheit: Man kann numerische IDs angeben. chown 1001:100 datei setzt Besitzer UID=1001, GID=100. Falls man einen Benutzer- oder Gruppennamen mit Doppelpunkt hat, der numerisch beginnt, kann man -- verwenden um Verwechslung zu vermeiden oder explizit username: schreiben.

chgrp (Change Group)

Beschreibung: Ändert die Gruppenzugehörigkeit einer Datei oder eines Verzeichnissesfile-4fdk7rvx39azjretinarqbfile-4fdk7rvx39azjretinarqb. Äquivalent zu chown <gruppe>. Wird benutzt, um den Gruppeneigentümer anzupassen.

Syntax:

chgrp [Optionen] <neueGruppe> DATEI...

Optionen:

| Option | Beschreibung |
|---------------|--|
| -R | Rekursive Änderung der Gruppe in Unterverzeichnissen (alle Dateien/Dirs innerhalb werden angepasst). |
| -f | Keine Fehlermeldungen bei Problemen ausgeben (silent). |
| -v | Ausführliche Ausgabe pro Datei. |

| Option | Beschreibung |
|--------------------------|--|
| -- | Übernimmt die Gruppenzugehörigkeit der <i>Vorlage</i> -Datei auf alle angegebenen Dateien. (Ignoriert das Gruppen-Argument und nutzt stattdessen die Gruppe der Referenz.) |
| reference=Vorlage | |

Hinweis: Normale Benutzer dürfen chgrp nur verwenden, um eine Datei auf eine Gruppe zu setzen, der sie selbst angehören. Root kann beliebige Zuordnungen vornehmen.

lsattr (List Attributes)

Beschreibung: Listet die erweiterten *Dateiattribute* (Filesystem-Attribute) von ext-Dateisystemen auf file-4fdk7rvx39azjretinarqb. Diese Attribute wie *immutable* oder *append-only* steuern besondere Verhalten auf niedriger Ebene (nur auf bestimmten FS wie ext2/3/4, XFS - mit anderen Tools - etc. vorhanden). lsattr zeigt pro Datei eine Zeichenfolge von Attribut-Flags.

Syntax:

lsattr [Optionen] [Datei/Verzeichnis]...

Optionen:

Option Beschreibung

| | |
|-----------|---|
| -a | Zeigt auch versteckte Dateien (Dateien, die mit Punkt beginnen) an.
Standardmäßig ignoriert lsattr diese analog zu ls ohne -a. |
| -d | Listet das Verzeichnis selbst auf, nicht dessen Inhalt. (Bei Angabe eines Verzeichnisses wird normalerweise dessen Inhalt rekursiv aufgelistet; mit -d sieht man nur die Attribute des Verzeichniseintrags selbst) file-4fdk7rvx39azjretinarqb. |
| -R | Rekursive Ausgabe der Attribute in allen Unterverzeichnissen. |
| -v | Zeigt die Dateiversionsnummer (Version im ext-Format) mit an, falls vorhanden.
(Das Attribut 'version' wird kaum benutzt, -v gibt eine zusätzliche Spalte mit Versionsnummer). |
| -V | Ausführliche Informationen über die Version des Programms und detaillierte Fehler ausgeben (verbose). |

Beispiel Ausgabe:

```
$ lsattr wichtige_datei
----i-----e-- wichtige_datei
```

Hier bedeuten die Buchstaben: i = immutable (unveränderlich), e = extents (standard auf ext4). Bindestriche zeigen nicht gesetzte Flags. Wichtige Flags: i (immutable: Datei kann

nicht verändert/gelöscht umbenannt werden bis Attribut entfernt), **a** (append-only: nur anhängen erlaubt), **d** (no dump: Datei wird von Dump-Backup ignoriert), **A** (no atime updates), **S** (synchronous updates) etc.

chattr (Change Attributes)

Beschreibung: Ändert die erweiterten Dateisystem-Attribute (wie immutable, append-only etc.) unter Linux auf ext2/ext3/ext4 Dateisystemenfile-4fdk7rvx39azjretinarqb. Nur Root darf die meisten dieser Attribute setzen.

Syntax:

chattr [Optionen] [+-=Attribute] Datei...

Optionen:

| Option | Beschreibung |
|-----------------------|--|
| -R | Rekursive Attributänderung in Unterverzeichnissenfile-4fdk7rvx39azjretinarqb. |
| -V | Ausführlich – zeigt an, welche Änderungen für welche Datei vorgenommen wurden. |
| -f | Unterdrückt Fehlermeldungen (force). |
| -v | Setzt die Versionsnummer der Datei (falls vom Dateisystem unterstützt). |
| <Nummer> | Selten benutzt – ext-Dateisysteme haben optional Dateiversionsnummern. |

Attribute setzen (Symbolik): Man gibt nach möglichen Optionen eine Zeichenfolge an, der ein +, - oder = vorangestellt ist:

- + bedeutet diese Attribute *hinzufügen* (zusätzlich zu bestehenden).
- - bedeutet diese Attribute *entfernen*.
- = bedeutet *exakt setzen* (alle aktuellen Attribute durch diese ersetzen).

Die wichtigsten Attribut-Buchstaben auf ext4 sind:

- **i – immutable:** Datei kann nicht modifiziert, umbenannt oder gelöscht werdenfile-5y1edouysrow4zj6phjcqm. Prozesse können sie nicht öffnen im Schreibmodus. Schützt kritische Dateien (z.B. /etc/passwd) vor Änderungen, sogar von root (root muss erst chattr -i machen).
- **a – append only:** Datei kann nur angehängt, aber nicht überschrieben oder gelöscht werden. Typisch für Logfiles, damit niemand alte Einträge löscht.
- **e – extent format:** zeigt an, dass die Datei extents benutzt (automatisch auf ext4, nicht manuell gesetzt).

- **d – no dump:** Datei soll von Backup-Programmen, die dem Dump-Standard folgen, ignoriert werden.
- **A – no atime updates:** beim Lesen der Datei wird der atime-Zeitstempel nicht aktualisiert (Performance-Optimierung).
- **S – synchronous updates:** Änderungen an der Datei werden synchron auf den Datenträger geschrieben (wie bei chattr +S – ähnlich wie Ordner mit Dirsync mount option).
- **t – no tail-merging:** (relevant bei ext-Dateisystem, deaktiviert Tail-Merging bei Reverse-Mapping; selten benutzt).
- **T – top of directory hierarchy:** Verzeichnis wird als Top-Level für Ordnungszwecke in ext behandelt (verhindert bestimmte Ordnungsoptimierungen; selten).
- **u – undeletable:** Dateiinhalt soll beim Löschen inhaltlich gesichert werden, sodass man ihn wiederherstellen könnte. (Wird von ext4 i.d.R. nicht implementiert).

Beispiel:

- chattr +i datei.txt – setzt *immutable*.
- chattr -i datei.txt – entfernt dieses Flag wieder (dann ist Bearbeiten/Löschen wieder möglich).
- chattr =a Datei.log – setzt **nur** append-only und entfernt alle anderen Attribute (falls welche gesetzt waren). Danach kann in Datei.log nur noch angehängt werden.

getfacl (Get File ACL)

Beschreibung: Liest die Access Control Lists (erweiterten Zugriffsrechte) von Dateien oder Verzeichnissen aus und zeigt sie anfile-4fdk7rvx39azjretinarqb. Auf Linux-Systemen mit POSIX-ACL-Unterstützung (z.B. ext4 mit acl-Option eingehängt) kann man pro Datei feinere Berechtigungen für einzelne Benutzer oder Gruppen setzen. getfacl zeigt diese ACL-Einträge zusätzlich zu den Standardrechten.

Syntax:

getfacl [Optionen] Datei...

Optionen:

| Option | Beschreibung |
|---------------|---------------------|
|---------------|---------------------|

- a** Zeigt auch die *verschachtelten ACLs* (falls vorhanden – nicht üblich bei POSIX ACL).

| Option | Beschreibung |
|----------------------|--|
| -e | Zeigt die Ausgabe im <i>effective</i> Modus – Standard. (Jeder Eintrag inkl. "effective permissions", relevant wenn Mask gesetzt ist). |
| -s | Stripschutz: Zeigt nur die tatsächlichen ACL-Einträge, ohne die Grundeinträge (user/group/other). |
| -d | Zeigt die <i>Default-ACL</i> eines Verzeichnisses (die Vorlage, die an neu darin erzeugte Dateien vererbt wird). In der Ausgabe mit default: gekennzeichnet. |
| --omit-header | Unterdrückt den Kommentar-Kopf in der Ausgabe, der Dateinamen und Hinweise beinhaltet. |
| -R | Rekursiv über Verzeichnisse, getfacl für jede Datei auflisten. |

Ausgabe-Beispiel:

```
$ getfacl projekt/
# file: projekt/
# owner: alice
# group: developers
user::rwx
user:bob:rwx      # bob hat rw Rechte zusätzlich
group::r-x
mask::rw-        # effektive Maske für ACL = rw
other::---
default:user::rwx # Default ACL für neu angelegte Objekte
...

```

Hier sieht man die Standard-Owner/Group-Einträge und eine explizite ACL, die User bob Schreibrechte gibt, sowie eine *mask*, die die maximal wirksamen Rechte auf rw beschränkt.

setfacl (Set File ACL)

Beschreibung: Setzt oder ändert Access Control Lists (ACLs) von Dateien und Verzeichnissen file-4fdk7rvx39azjretinarqb. Damit kann man zusätzliche Berechtigungen für bestimmte Nutzer oder Gruppen vergeben (oder entziehen), die über die klassischen Besitzer-Gruppe-Andere Rechte hinausgehen.

Syntax:

`setfacl [Optionen] { -m|-x <ACL-Eintrag> ... | -b | -k } Datei...`

Optionen:

Option Beschreibung

- m** *Modify*: Fügt einen ACL-Eintrag hinzu oder aktualisiert ihn. Beispiel: `setfacl -m u:bob:rw file` gibt Benutzer *bob* rw-Rechte auf *file*. Mehrere -m Einträge oder Einträge mit Komma getrennt sind möglich.
- x** *ACL* *Remove*: Entfernt einen spezifischen ACL-Eintrag. Z.B. `setfacl -x g:marketing file` entfernt den ACL-Eintrag für Gruppe "marketing".
- b** *Remove all* – Löscht alle ACL-Einträge (außer den Basis-Einträgen für Owner/Group/Other). Damit werden alle erweiterten Berechtigungen zurückgesetzt. (Datei hat dann nur noch die klassischen Unix-Rechte.)
- k** *Remove Default* – Entfernt die Default-ACL eines Verzeichnisses (vererbte ACL-Vorlage). Hat auf Dateien keine Wirkung.
- R** Rekursiv – wendet die ACL-Änderung auf alle Dateien im Verzeichnis (und Unterverzeichnisse) an. Nützlich um z.B. ganze Bäume eine neue ACL zu geben.
- d** Setzt Einträge als *Default-ACL* für Verzeichnisse. Benutzt in Kombination mit -m oder -x. Beispiel: `setfacl -m d:g:team:rwx projekt` setzt eine Default-ACL, so dass neue Dateien unter *projekt*/ der Gruppe *team* rw Rechte erben, und Verzeichnisse rwx.
- **mask** Berechnet und setzt die ACL-Maske basierend auf den aktuellen Gruppen-Einträgen neu. (Die Maske begrenzt die maximalen Rechte der ACL-Einträge außer Owner/other). Standardmäßig wird die Maske auto-anangepasst.
- n** *Don't recalc mask*: Verhindert, dass setfacl die ACL-Maske automatisch anpasst. Normalerweise soll man diese Option nicht verwenden, außer man setzt die Maske manuell mit -m m::....

Beispiel: `setfacl -m u:bob:rw,g:marketing:r-- report.txt` – gibt Benutzer *bob* Lese- und Schreibrecht und der Gruppe *marketing* nur Leserecht auf *report.txt*.

Hinweis: Nicht alle Dateisysteme unterstützen POSIX-ACL. Bei ext4, XFS etc. muss i.d.R. das Volume mit Option `acl` eingehängt sein (moderne Systeme haben das meist per Default). Mit `getfacl` kann man prüfen, ob vorhandene ACLs existieren.

Datei- und Verzeichnisverwaltung

stat (Dateistatus anzeigen)

Beschreibung: Zeigt detaillierte Meta-Informationen zu Dateien oder Dateisystemen an.
stat gibt z.B. Größe, Blocks, Zeitstempel (Zugriff, Änderung, Statusänderung),
Berechtigungen in Zahlenform, Device-ID, Inode-Nummer usw. in strukturierter Form aus.

Syntax:

stat [Optionen] <Datei>...

Optionen:

| Option | Beschreibung |
|------------------------------------|--|
| -f, --file-system | Zeigt anstelle der Datei-Details die Informationen zum Dateisystem, auf dem die Datei liegt. Z.B. Statistiken wie Blockgröße, Gesamtblöcke, freie Blöcke, FS-Typ etc. |
| -c FORMAT , --format=FORMAT | Gibt die Ausgabe gemäß <i>FORMAT</i> formatiert aus. Man kann Platzhalter verwenden, um bestimmte Werte auszugeben. Z.B. stat -c "%A %h %U %G %s %n" datei gibt in einer Zeile: modestr, Linkanzahl, Owner, Group, Size, Name. (Siehe man stat für alle Platzhalter wie %A = Rechte in rwx, %a = oktal, %U = Ownername, %s = Größe etc.) |
| --printf=FORMAT | Ähnlich wie --format, jedoch ohne automatisch Zeilenumbruch am Ende. Kann benutzt werden, um mehrere stat-Aufrufe zu skripten. |
| -t, --terse | "Knapp": Gibt die Ausgabe in einer kompakten einzeiligen Form, Felder durch Leerzeichen getrennt, aus (für einfache maschinelle Weiterverarbeitung). |
| --help, --version | Anzeigen von Hilfe bzw. Version. |

Beispieldaten:

```
$ stat /etc/hosts
File: /etc/hosts
Size: 258    Blocks: 8      IO Block: 4096  regular file
Device: 802h/2050d Inode: 13422506  Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 0/  root)  Gid: ( 0/  root)
Access: 2025-04-01 10:22:13.000000000 +0200
Modify: 2025-04-01 10:22:13.000000000 +0200
Change: 2025-04-01 10:22:15.000000000 +0200
```

Birth: -

Dies zeigt z.B., dass /etc/hosts 258 Bytes groß ist, auf einem bestimmten Device, inode etc., Rechte 0644, Eigentümer root, Zeitstempel für Zugriffs-/Änderungs-/Änderungszeit (Birth ist Erstellung, nicht unterstützt).

file (Dateityp bestimmen)

Beschreibung: Untersucht eine Datei und versucht den Typ bzw. Format der Datei zu bestimmen. file nutzt einen umfangreichen Magie-Pattern-Katalog (/usr/share/file/magic), um anhand der Bytes einer Datei (Magic Numbers, Struktur etc.) menschlich lesbare Dateitypinformationen auszugeben. Praktisch, um z.B. Binärformate, Skripttypen oder Kodierungen zu erkennen.

Syntax:

file [Optionen] <Datei>...

Optionen:

Option Beschreibung

- M** Zeigt den MIME-Typ der Datei anstatt einer Beschreibung. Z.B. text/plain;
- i** charset=utf-8 statt "UTF-8 Unicode text". Hilfreich für Scripting oder Web-Kontext.
- b** *Brief* – gibt nur die Beschreibung aus, ohne den Dateinamen voranzustellen.
- L** (By default schreibt file: "<Dateiname>: <Beschreibung>"). Mit -b erhält man nur "<Beschreibung>".
- F** Folgt symbolischen Links und untersucht das Ziel statt den Link selbst.
- L** Standardmäßig wird der Link selbst untersucht und meist als "symbolic link to <target>" ausgegeben.
- h** (alias --no-dereference) Bewirkt das Gegenteil: behandelt Symlinks wie normale Dateien (gleichbedeutend mit Standardverhalten, -h ist nützlich falls -L als default kompiliert wäre).
- s** Untersucht auch Gerätedateien (block/char special) und Auslagerungsdateien. Standardmäßig werden einige spezielle Dateitypen nicht inhaltlich analysiert (um z.B. nicht eine Device-Datei zu lesen). -s kann riskant sein (kann z.B. /dev/zero "lesen" -> blockiert).
- z** Durchsucht komprimierte Dateien (gzip-komprimierte) – versucht innerhalb eines .gz die Dateisignatur zu prüfen.
- E** Gibt (Exit)Status 1 zurück falls *irgendeine* der Dateien kein bekannter Typ bestimmt werden konnte. (Nützlich in Skripten).

Option Beschreibung

--help, Hilfe bzw. Versionsnummer und unterstützte Magic-Datei-Version ausgeben.
-v

Beispiele:

- file /bin/ls -> **ELF 64-bit LSB executable**, x86-64, dynamically linked ...
- file archive.tar.gz -> **gzip compressed data**, was probably a tar archive (reads inside and erkennt es als tar in gz).
- file picture.jpg -> **JPEG image data**, JFIF standard 1.01, ...

ln (Link erstellen)

Beschreibung: Erstellt Verknüpfungen zu Dateien – entweder **Hard Links** oder **Symbolische Links**. Ein Hardlink ist ein weiterer Directory-Eintrag für dieselbe Inode (nur innerhalb desselben Dateisystems möglich). Ein Symbolischer Link (Symlink) ist eine kleine Spezialdatei, die auf einen Pfad verweist.

Syntax:

ln [Optionen] <Ziel> [Linkname]

Standard ohne Optionen erstellt *Hardlink* mit Namen *Linkname*, der auf Datei *Ziel* zeigt. Wenn *Linkname* nicht angegeben, wird im aktuellen Verzeichnis ein Link gleichen Namens wie *Ziel* erstellt. Für Symlinks Option -s nutzen.

Optionen:

| Option | Beschreibung |
|-----------------------------|---|
| -s, --symbolic | Erstellt einen <i>symbolischen Link</i> statt eines Hardlinks. Der Link verweist auf den angegebenen <Ziel>-Pfad. Wenn <Ziel> kein absoluter Pfad ist, wird er relativ vom Link aus interpretiert. Beispiel: ln -s /etc/hosts hostslink erzeugt Symlink <i>hostslink</i> -> <i>/etc/hosts</i> . |
| -f, --force | Existierende Zielfile (Linkname) ggf. entfernen, falls schon vorhanden. Ohne -f würde ln bei existierendem Linknamen einen Fehler ausgeben. (Bei -s mit -f: überschreibt existierenden Link.) |
| -n, --no-dereference | Beim Überschreiben eines existierenden symbolischen Links (mit -f) behandle den Link selbst statt dessen Ziel. (Verhindert, dass versehentlich die Datei, auf die ein bestehender Link zeigt, gelöscht wird). |
| -v, --verbose | Gibt eine Meldung pro verarbeitetem Link aus. |

| Option | Beschreibung |
|----------------------------------|---|
| -T, --no-target-directory | Behandelt das letzte Argument immer als <i>genauen Linknamen</i> , selbst wenn es ein existierendes Verzeichnis ist. (Verhindert, dass ln im Zielverzeichnis einen Link mit Standardname erstellt.) |
| -d, -F, --directory | Erlaubt es root, auch <i>Verzeichnis-Hardlinks</i> zu erstellen.
(Normalerweise verboten – Option nur auf speziellen Systemen mit -f kombinierbar, in der Praxis kaum verwendet). |
| -b, --backup | Erzeugt ein Backup der Zielfile falls -f zum Überschreiben genutzt wird. (Nützlich, um das ersetzte File nicht zu verlieren.) Siehe auch --backup=t etc. |
| -S <Suffix> | Gibt ein Suffix für Backup-Dateien an (Standard ~). Bsp: -b -S .bak. |

Beispiele:

- Hardlink: ln /path/to/original.txt copy.txt – copy.txt ist nun ein weiterer Verweis auf original.txt (gleiche Inode). Änderungen an einer erscheinen in der anderen, da es dieselbe Datei ist. Löscht man eine, bleibt die andere erhalten (Datei wird erst gelöscht, wenn letzter Link entfernt).
- Symlink: ln -s /etc/apache2 siteconfig – erstellt siteconfig -> /etc/apache2. Aufruf von ls -l siteconfig zeigt dann siteconfig -> /etc/apache2.

cd (Change Directory) – siehe oben bei Grundlegende Befehle

(Im Kontext Datei/Verzeichnis-Verwaltung sei auf den bereits beschriebenen Befehl cd verwiesen, um Verzeichnisse zu wechseln.)

tree (Verzeichnisbaum anzeigen)

Beschreibung: Zeigt die Verzeichnisstruktur grafisch in Form eines Baumes an. tree listet rekursiv alle Unterverzeichnisse und Dateien in hierarchischer eingerückter Darstellung. Es erleichtert das Visualisieren der Verzeichnis-Hierarchie.

Syntax:

tree [Optionen] [Verzeichnis]

Optionen:

| Option | Beschreibung |
|---------------|---|
| -a | Zeigt <i>alle</i> Dateien an, inkl. versteckter (auch solche, die mit . beginnen). Standardmäßig ignoriert tree versteckte Dateien, außer . und .. immer. |

| Option | Beschreibung |
|-----------------------------|---|
| -d | Listet <i>nur Verzeichnisse</i> auf, keine Dateien. Nützlich, um reinen Verzeichnisbaum zu sehen. |
| -L Level | Beschränkt die Tiefe der Ausgabe auf <i>Level</i> Ebenen. Z.B. tree -L 2 zeigt nur zwei Ebenen tief (akt. Verzeichnis + Unterverzeichnisse, aber keine tieferen). |
| -f | Zeigt den vollen Pfad vor jedem Eintrag (statt nur Namen relativ). |
| -I
<i>PATTERN</i> | Ignoriert Einträge, die auf das Muster passen (Groß/Kleinschreibung berücksichtigt, mehrere Patterns mit ` |
| -P
<i>PATTERN</i> | Zeigt nur Einträge an, die dem Pattern entsprechen (Whitelist). Umgekehrt zu -I. |
| -F | Markiert Verzeichnisse mit abschließendem /, ausführbare Dateien mit *, Symlinks mit @, Pipes mit ` |
| -p | Zeigt UNIX-Dateirechte (Permission bits) vor jedem Eintrag. |
| -u / -g | Zeigt Benutzer bzw. Gruppenbesitzer an. (Wie ls -l Infos). |
| -h | <i>Human-readable size:</i> Zeigt Dateigrößen in menschenlesbarem Format (K, M, G) anstatt in Bytes (wirkt mit -s). |
| -s | Zeigt Dateigrößen in Bytes neben jedem Dateinamen an. |
| --du | Summiert Verzeichnisgrößen wie du (statt wie normal lediglich die Einzeldateigröße zu zeigen). |
| -t | Sortiert die Ausgabe nach Zeit (Neueste zuerst) statt alphabetisch. |
| -o Datei | Gibt die Ausgabe in die angegebene Datei (überschreibt) anstatt auf stdout. |
| -n | Keine farbliche Hervorhebung (Standard ist bunt, -n erzwingt Monochrom). |
| --help | Hilfe anzeigen (Übersicht der Optionen). |

Beispiel:

```
$ tree -L 2 /etc
```

```
/etc
```

```
|   └── apache2
```

```
|       └── apache2.conf
```

```
|       └── envvars
```

```

|   └── mods-available
|
├── cron.daily
|
|   ├── apt-compat
|
|   └── man-db
|
└── default
    ├── cron
    └── rcS

```

6 directories, 5 files

Hier sieht man /etc mit zwei Ebenen Tiefgang.

mkdir (Make Directory)

Beschreibung: Erstellt ein neues Verzeichnis (oder mehrere) im Dateisystem. Kann rekursiv auch mehrere Ebenen auf einmal anlegen.

Syntax:

`mkdir [Optionen] <Verzeichnisname>...`

Optionen:

| Option | Beschreibung |
|--------------------------|---|
| -p, --parents | Erzeugt auch alle notwendigen übergeordneten Verzeichnisse, falls sie nicht existieren. Somit werden keine Fehler ausgegeben, wenn Verzeichnisse in der Pfadangabe fehlen. Beispiel: <code>mkdir -p /tmp/foo/bar</code> legt sowohl <code>/tmp/foo</code> (falls nicht da) als auch <code>/tmp/foo/bar</code> an. |
| -m, --mode=MODE | Setzt die Zugriffsrechte des neu erstellten Verzeichnisses auf <i>MODE</i> (numerisch oder symbolisch) anstatt der Standardrechte (die Standardrechte sind i.d.R. 0777 abzüglich Umask). Z.B. <code>mkdir -m 750 projDir</code> erstellt <i>projDir</i> mit rwx für Besitzer, rx für Gruppe, nichts für andere. |
| -v, --verbose | Ausgabe einer Meldung für jedes angelegte Verzeichnis (zeigt Pfad an). |
| --help, --version | Anzeige von Hilfe bzw. Version. |

cp (Copy File)

Beschreibung: Kopiert Dateien oder Verzeichnisse. Bei Angabe mehrerer Quelldateien muss das letzte Argument ein Verzeichnis sein (Zielverzeichnis). cp kann auch Verzeichnisbäume rekursiv kopieren.

Syntax:

cp [Optionen] <Quelle>... <Ziel>

Wenn <Ziel> ein Verzeichnis ist, werden alle <Quelle>-Dateien dorthin kopiert (mit gleichem Namen). Ist nur eine Quelle und <Ziel> kein Verzeichnis, wird die Datei dorthin kopiert/umbenannt.

Optionen:

| Option | Beschreibung |
|----------------------------|--|
| -r, -R, --recursive | Kopiert Verzeichnisse rekursiv (Inhalt und Unterverzeichnisse). Ohne diese Option kopiert cp standardmäßig keine Verzeichnisse. (GNU cp hat auch -r/-R – beide meist gleichwertig. POSIX fordert -R.) |
| -a, --archive | Archivmodus: Kopiert rekursiv und bewahrt dabei möglichst alle Attribute. Entspricht -dR --preserve=all (also -R, und -p für timestamps, owner, group, perms, ACLs, xattrs, Links etc.). Dies ist für Vollbackups ideal ionos.at . |
| -p, --preserve | Bewahrt Eigentümer, Gruppe, Berechtigungen und Zeitstempel der Datei beim Kopieren ionos.at . (Ohne -p übernimmt cp Standard: setzt aktuellen Zeitstempel "jetzt" und erstellt als aktueller Benutzer). Kann Argumente haben wie --preserve=mode,ownership,timestamps,links,xattr,all. |
| -f, --force | Überschreibt vorhandene Ziel-Dateien ohne Nachfrage, und versucht auch nicht-schreibbare Ziele zu entfernen, um Kopie dennoch zu platzieren. (Im Konflikt mit -n gilt letzter Schalter.) |
| -i, --interactive | Fragt nach, bevor eine existierende Zielfile überschrieben wird (Confirm Y/N). Gibt dem Benutzer die Möglichkeit, Überschreiben abzulehnen. |
| -n, --no-clobber | Überschreibt <i>keine</i> existierenden Dateien. D.h. falls Ziel existiert, wird die Kopie für diese Datei ausgelassen. (Verhindert unabsichtliches Überschreiben) |

| Option | Beschreibung |
|--|--|
| -u, --update | Kopiert nur, wenn <i>Quelle</i> neuer als <i>Ziel</i> ist, oder Ziel noch nicht existiert. Praktisch für inkrementelle Updates. Existiert Ziel mit gleich/neueren Datum, wird es nicht überschrieben. |
| -d, --no-dereference | Kopiert <i>Symbolic Links</i> als Links (anstatt deren referenzierten Inhalt). Standard: cp folgt symbolischen Links und kopiert die Datei dahinter. Mit -d wird der Link selbst neu erstellt am Ziel (Hardlinks bleiben hardlinks sofern möglich; -d ist in GNU cp gleichbedeutend mit --preserve=links). |
| -L, --dereference | Folgt allen Symlinks: kopiert stets den referenzierten Inhalt (Standardverhalten bei regulärem cp für Dateien). Bei Angabe von Verzeichnis mit -R: -L würde bewirken, dass cp alle gefundene Symlinks auflöst und deren Ziel kopiert (kann zu Loops führen, daher mit Vorsicht). |
| -P, --no-dereference --preserve=links | Kopiert Links als Links (das ist Standard mit -a enthalten). (Im GNU cp -P bedeutet No-dereference; beachten: -d und -P ähnlich). |
| --parents | Erhält die Quellpfadstruktur im Ziel. Bsp: cp --parents dir1/dir2/file.txt /backup legt /backup/dir1/dir2/file.txt an man7.org . Nur sinnvoll, wenn Ziel ein Verzeichnis ist. |
| -l, --link | Erstellt <i>Hardlinks</i> statt Kopien der Datei, sofern im selben Dateisystem. (Keine Datenkopie, nur neuer Name, ähnlich ln). |
| -s, --symbolic-link | Erstellt <i>Symlinks</i> statt die Daten zu kopieren. (Zieldatei wird symbolischer Verweis auf Quelldatei). |
| -H, -L, -P (bei -R) | Steuerung des Symlink-Verhaltens während rekursivem Kopieren: -P (Default) ignoriert Linkinhalte (kopiert Symlinks als Symlinks), -L folgt Symlinks zu Dateien, -H folgt nur Symlinks angegeben auf Befehlzeile. (-a impliziert -P). |
| -- backup[=Control] | Erzeugt Backup von Zieldateien, die überschrieben würden. (Etwa file.txt~ anlegen). Kann =simple, numbered etc. steuern. Siehe auch VERSION_CONTROL Env. |
| --help, --version | Hilfe/Version anzeigen. |

mv (Move/Rename)

Beschreibung: Verschiebt Dateien oder Verzeichnisse an einen neuen Pfad, bzw. benennt sie um. mv kann auch zum einfachen Umbenennen einer Datei verwendet werden (Quell- und Zielpfad auf selbem FS = schneller Rename, kein Kopieren). Falls Quelle und Ziel auf unterschiedlichen Dateisystemen liegen, arbeitet mv durch Kopieren und anschließendem Löschen der Quelle.

Syntax:

`mv [Optionen] <Quelle>... <Ziel>`

Optionen:

| Option | Beschreibung |
|--------------------------------------|---|
| -i, --interactive | Fragt nach Bestätigung, falls eine Zielfile durch den Move überschrieben würde (falls im Zielpfad eine Datei gleichen Namens existiert). |
| -f, --force | Erzwingt das Überschreiben ohne Nachfrage. Ziel-Dateien werden ohne Warnung ersetzt. (Standardverhalten ohne -i). |
| -n, --no-clobber | Überschreibt keine bestehenden Dateien. Bei Namenskollision bricht mv für diese Datei ab (gibt keinen Fehler, exit code trotzdem 0). In GNU mv hat -n Vorrang vor -i falls beide angegeben. |
| -u, --update | Bewegt/überschreibt nur, wenn die Quelle neuer ist als die vorhandene Zielfile <i>oder</i> die Zielfile nicht existiert. (So werden neuere Versionen moved, ältere Quellen, die schon am Ziel sind, bleiben unberührt.) |
| -v, --verbose | Gibt aus, welche Datei wohin verschoben/umbenannt wurde (für jede Operation eine Zeile). |
| --
backup[=CONTROL] | Wie bei cp: legt Backup der Ziel-Datei an, falls existiert, bevor sie überschrieben wird. CONTROL kann none, numbered, existing, simple sein (siehe Doku oder ENV VERSION_CONTROL). |
| --strip-trailing-slashes | Entfernt evtl. nachgestellte Schrägstriche bei den Pfadangaben der Quellen (um Verwirrung mit Verzeichnissen zu vermeiden). Standard: neuere mv machen das ohnehin. |
| --help, --version | Hilfe/Version. |

Hinweis: mv kann mehrere Dateien in ein Zielverzeichnis verschieben oder eine einzelne Datei/Ordner umbenennen. Bei Verzeichnissen rekursiv, aber mv muss dafür nicht extra Option haben (im Gegensatz zu cp), da der Move innerhalb desselben FS nur Metadaten

ändert. Bei verschiedenen FS kopiert mv im Hintergrund (rename()) Syscall schlägt dann fehl, worauf cp+unlink passiert).

rmdir (Remove Directory)

Beschreibung: Löscht leere Verzeichnisse. Im Gegensatz zu rm entfernt rmdir nur Verzeichnisse und auch nur, wenn sie keine Dateien/Unterverzeichnisse mehr enthalten. Es kann optional ganze verschachtelte Verzeichnisbäume entfernen, sofern diese leer sind (Eltern werden dann rekursiv entfernt).

Syntax:

rmdir [Optionen] <Verzeichnis>...

Optionen:

| Option | Beschreibung |
|-----------------------------------|---|
| -p, --parents | Entfernt das angegebene Verzeichnis und ggf. dessen übergeordnete Verzeichnisse, falls diese nach jedem Entfernen leer werden. Beispiel: rmdir -p projekt/build/output versucht erst output zu löschen. Ist danach build leer, wird auch build gelöscht. Ist danach projekt leer, wird auch projekt gelöscht. |
| -v, --verbose | Ausgabe einer Meldung für jedes gelöschte Verzeichnis. |
| --ignore-fail-on-non-empty | Unterdrückt die normale Fehlermeldung, falls ein Verzeichnis nicht leer ist. (Das Verzeichnis wird nicht gelöscht, aber es kommt kein Fehlerstatus; mit Vorsicht verwenden). |

Nützlich: Um verschachtelte leere Ordner zu entfernen. Aber um Verzeichnisse mit Inhalt zu löschen, nutzt man rm -r.

rm (Remove Files/Directories)

Beschreibung: Löscht Dateien oder Verzeichnisse. Standardmäßig entfernt rm nur Dateien; mit -r kann es auch Verzeichnisbäume löschen (inkl. aller enthaltenen Dateien). **Vorsicht:** Gelöschte Dateien werden nicht ohne Weiteres wiederherstellbar gespeichert (kein Papierkorb, außer man nutzt safe-rm Tools).

Syntax:

rm [Optionen] <Datei/Pfad>...

Optionen:

| Option | Beschreibung |
|----------------------------|---|
| -r, -R, --recursive | Löscht Verzeichnisse <i>rekursiv</i> mitsamt ihrem gesamten Inhalt. Damit können komplett Verzeichnisbäume entfernt werden. Ohne -r verweigert rm das Löschen von Verzeichnissen mit Fehler "is a directory". |
| -d | Löscht ein <i>leeres</i> Verzeichnis. (Ohne -r löscht rm normalerweise gar keine Verzeichnisse; -d erlaubt es aber für leere, ähnlich rmdir. Nützlich, aber -r deckt diesen Fall auch ab.) |
| -f, --force | <i>Force/ignore nonexistent:</i> Unterdrückt Rückfragen und Fehlermeldungen. Existiert eine angegebene Datei nicht, wird es ignoriert. Schreibschutz wird ignoriert (d.h. es fragt nicht nach Bestätigung bei schreibgeschützten Dateien). Kombiniert mit -r sehr gefährlich, da es kompromisslos alles löscht. Wird oft in Skripten genutzt. |
| -i | Interaktiv nachfragen vor <i>jedem</i> Löschvorgang. (Für viele Dateien unpraktisch). |
| -I | "ein wenig interaktiv": Fragt nur einmal pauschal nach, wenn mehr als 3 Dateien gelöscht werden sollen <i>oder</i> ein rekursiver Lösch auf einem Verzeichnis durchgeführt wird. (GNU rm Feature als Sicherheitsnetz)file-5y1edouysrow4zj6phjcqm. |
| --preserve-root | Schützt vor rm -rf /. Diese Option ist in GNU rm standardmäßig aktiv – es verweigert das komplette Löschen des Root-FS. Man kann mit --no-preserve-root überschreiben (nicht empfohlen). |
| -v, --verbose | Zeigt für jede Datei eine Meldung an, dass sie gelöscht wurde. |
| --help, --version | Hilfe bzw. Version. |

Anwendung:

- **Einzelne Datei löschen:** rm datei.txt.
- **Mehrere Dateien löschen:** rm *.o (löscht alle .o Dateien im aktuellen Verzeichnis).
- **Verzeichnis und Inhalt löschen:** rm -r docs/ – löscht Ordner docs mitsamt allen Unterdateien.
- **Ohne Rückfrage alles löschen:** rm -rf tempdir. (Vorsicht!).

head (Dateianfang anzeigen)

Beschreibung: Gibt die ersten Zeilen einer Datei aus (standardmäßig 10 Zeilen). head ist nützlich, um schnell die ersten Zeilen (z.B. Header einer CSV, Beginn eines Logs etc.) anzuschauen, ohne die ganze Datei zu öffnen. Es kann auch binäre Dateien auf Anfang bytes prüfen.

Syntax:

head [Optionen] [Datei]...

Ohne Datei liest head von der Standardeingabe (z.B. in Pipes). Bei mehreren Dateien wird jeweils ein Header mit dem Dateinamen ausgegeben.

Optionen:

| Option | Beschreibung |
|--|---|
| -n <i>K</i>, --lines=<i>K</i> | Gibt die ersten <i>K</i> Zeilen aus. <i>K</i> kann eine Zahl sein oder -K für alle bis auf die letzten <i>K</i> Zeilen (negativer Wert). Standard ist 10. Beispiel: head -n 20 file zeigt 20 Zeilen, head -n -5 file zeigt alle außer der letzten 5 Zeilen. |
| -c <i>Bytes</i>, --bytes=<i>B</i> | Gibt die ersten <i>B</i> Bytes aus (statt Zeilen). Kann auch mit K, M, etc. Suffix (Potenzen von 1024) angegeben werden. Z.B. head -c 100K bigfile. Negative Werte analog: -c -10 alles außer letzte 10 Bytes. |
| -q, --quiet/--silent | Unterdrückt die Überschriften, die vor dem Inhalt stehen, wenn mehrere Dateien angegeben wurden. Normal zeigt head sowas wie "=> filename <==" vor dem Block jeder Datei. Mit -q wird alles aneinandergehängt ohne solche Banner. |
| -v, --verbose | Zeigt immer die Dateiüberschriften an, auch wenn nur eine Datei angegeben ist. (Normalerweise bei nur einer Datei keine Überschrift). |
| --help, --version | Hilfe bzw. Version ausgeben. |

Beispiel: head -n 3 /etc/passwd zeigt die ersten 3 Zeilen der passwd-Datei (typisch Root, daemon, bin Accounts).

tail (Dateiende anzeigen)

Beschreibung: Gibt die letzten Zeilen einer Datei aus (standardmäßig 10 Zeilen). Sehr nützlich, um z.B. das Ende von Logfiles zu sehen. Kann im *Follow-Modus* (-f) genutzt werden, um fortlaufend neuen Inhalt anzuzeigen (Echtzeit-Log-Überwachung).

Syntax:

tail [Optionen] [Datei]...

Optionen:

| Option | Beschreibung |
|--------------------------------------|--|
| -n K , --lines=K | Zeigt die letzten <i>K</i> Zeilen an. Negative Werte bedeuten ab einer bestimmten Zeile von Anfang an (ab Zeile <i>K</i>). Standard ist 10. Z.B. tail -n 50 /var/log/syslog zeigt die letzten 50 Zeilen. |
| -c B , *--bytes=B | Zeigt die letzten <i>B</i> Bytes an. Kann auch negative (ab Byte <i>X</i>) sein. |
| -f, --follow | Bleibt nach dem Anzeigen der letzten Zeilen geöffnet und zeigt fortlaufend neue Zeilen an, falls die Datei wächst (wie tail -f logfile zum Mitlesen). |
| -F | Ähnlich wie -f, aber zusätzlich "Wiederöffnen", falls Datei rotiert/neu erstellt wurde. (Follow Name statt Descriptor; praktisch für Logrotation – tail -F /var/log/app.log folgt auch nach Neustart des Logs dem neuen File). |
| -s Sek , --sleep-interval=Sek | Im Follow-Modus: Warte <i>Sek</i> Sekunden zwischen den Überprüfungen auf neue Daten. Standard 1.0 Sekunde. Kann z.B. auf 0.1 verringert werden, um schneller zu pollen, oder erhöht, um Last zu reduzieren. |
| --max-unchanged-stats=N | (Bei -F relevant) Wie oft in Folge unveränderte Inode geprüft wird, bevor tail davon ausgeht, dass ggf. Logrotate stattgefunden hat. |
| -q, --quiet | Unterdrückt Überschriften bei mehreren Dateien (analog head). |
| -v, --verbose | Erzwingt Überschriften, selbst bei einer Datei. |
| --help, --version | Hilfe bzw. Version. |

Anwendung:

- **Letzte Zeilen einer Datei:** tail -n 20 messages.log (Zeigt die letzten 20 Zeilen).
- **Live-Verfolgung:** tail -f /var/log/syslog (Druckt neu geschriebene Logzeilen fortlaufend). Abbruch mit Ctrl+C.
- **Combine mit grep:** z.B. tail -f server.log \| grep "ERROR" filtert live nur Zeilen mit "ERROR".

Dateien finden und Text filtern

grep (Global Regular Expression Print)

Beschreibung: Durchsucht die Eingabe (Dateien oder Standard Input) nach Zeilen, die einem bestimmten Muster entsprechen, und gibt diese Zeilen ausfile-4fdk7rvx39azjretinarqb. grep arbeitet mit regulären Ausdrücken (Standard Basic Regex,

erweiterte mit -E). Es ist das klassische Suchwerkzeug in Textstreams. Name stammt von der ed-Editor Anweisung g/re/p (global regex print).

Syntax:

```
grep [Optionen] 'Muster' [Datei...]
```

Optionen:

| Option | Beschreibung |
|----------------------------------|--|
| -i, --ignore-case | Ignoriert Groß-/Kleinschreibung beim Mustervergleichfile-4fdk7rvx39azjretinarqb. "Test" findet auch "test" oder "TEST". |
| -v, --invert-match | Gibt alle Zeilen aus, die nicht auf das Muster passen (Inverse Treffer)file-4fdk7rvx39azjretinarqb. Nutzt man oft zur Ausschlussfilterung. |
| -r, -R, --recursive | Durchsucht rekursiv alle Dateien in angegebenen Verzeichnissen (und Unterverz.)file-4fdk7rvx39azjretinarqb. Ohne diese Option behandelt grep Verzeichnisargumente normalerweise als Fehler oder ignoriert sie. (GNU grep: -r folgt Symlinks nicht, -R folgt Symlinks.) |
| -n, --line-number | Zeigt vor jeder gefundenen Zeile die Zeilennummer in der Datei an ibm.com . Format: Datei:Nr:Zeilentext. |
| -A NUM | (After) Zeigt <i>NUM</i> Zeilen nach jeder Trefferzeile zusätzlich an (Context) ibm.com . |
| -B NUM | (Before) Zeigt <i>NUM</i> Zeilen vor jeder Trefferzeile an. |
| -C NUM | (Context) Zeigt <i>NUM</i> Zeilen <i>vor und nach</i> jedem Treffer (gleichzeitig -A NUM -B NUM). Standard -C 2 beispielsweise gibt 2 Kontextzeilen um jede Fundstelle. |
| -l, --files-with-matches | Listet nur die Dateinamen auf, die mindestens einen Treffer enthalten. Pro Datei ein Name, Inhalt wird nicht ausgegeben. (Nützlich um zu sehen wo etwas vorkommt) |
| -L, --files-without-match | Listet nur Dateinamen, die <i>keinen</i> Treffer enthalten (Gegenteil von -l). |
| -c, --count | Zählt die Treffer pro Datei und gibt nur diese Anzahl aus (zusammen mit Dateinamen, außer mit -h). D.h. "file.txt:3". In Kombination mit -v zählt es Nicht-Treffer-Zeilen. |

| Option | Beschreibung |
|------------------------------|---|
| -o, --only-matching | Gibt nur den Teil der Zeile aus, der dem Suchmuster entspricht, nicht die ganze Zeile. Bei mehreren Treffern pro Zeile werden diese getrennt ausgegeben. |
| -q, --quiet/--silent | "Quiet": gibt keine Ausgabe; liefert lediglich Exit-Status 0 falls wenigstens ein Match gefunden wurde, 1 falls kein Match. Praktisch in Skripten (nur Rückgabewert interessiert). |
| -s, --no-messages | Unterdrückt Fehlermeldungen über nicht lesbare Dateien (z.B. wenn Zugriffsrechte fehlen oder Datei nicht existiert). |
| -E, --extended-regexp | Interpretiert das Muster als <i>Extended Regular Expression (ERE)</i> ionos.at . Das erlaubt z.B. Nutzung von +, ?, ` |
| -F, --fixed-strings | Interpretiert das Muster als <i>fixen String</i> (kein Regex, jedes Zeichen literal). Entspricht fgrep (die eine schnellere Suche ohne regex ermöglicht). Alle Muster (wenn -e mehrfach) werden als Literalstrings gesucht. |
| -P, --perl-regexp | Interpretiert das Muster als Perl-Compatible Regex (PCRE), falls grep mit PCRE-Unterstützung kompiliert wurde. Damit stehen mächtigere Regex-Features zur Verfügung. |
| -h, --no-filename | Unterdrückt die Ausgabe des Dateinamens vor jeder Trefferzeile (wenn mehrere Dateien durchsucht werden). Standardmäßig präfixet grep Datei: bei mehreren Input-Dateien. Mit -h erscheinen nur die Zeilen. (Bei Einzeldatei ist das sowieso Standard.) |
| -H, --with-filename | Zeigt Dateinamen auch bei einziger Datei an. (Umgekehrt zu -h). Nützlich in Pipelines wo Source unbekannt. |
| --color[=WHEN] | Hebt das gefundene Pattern farblich hervor in der Ausgabe. WHEN kann auto (Standard, nur wenn Terminal), always oder never sein. Setzt Umgebungsvariable GREP_COLOR/GREP_COLORS ggf. benötigt. |
| -e Pattern | Gibt das Suchmuster explizit an. Nötig z.B. wenn das Pattern mit - beginnt (damit es nicht als Option interpretiert wird). Kann mehrfach angegeben werden, um mehrere Muster (OR-Verknüpfung) zu suchen. |
| -f Datei | Liest Suchmuster aus angegebener Datei (ein Pattern pro Zeile). Nützlich bei sehr vielen oder komplexen Patterns. |

| Option | Beschreibung |
|---------------------------|---|
| --exclude=GLOB | Schließt Dateien vom Durchsuchen aus, deren Name auf GLOB (Wildcard-Muster) passt. Z.B. --exclude=*.min.js würde Minimierte JS ignorieren. (Nur wirksam bei rekursiver Suche oder wenn Verzeichnis als Input). Es gibt auch --include analog. |
| --exclude-dir=GLOB | Wie --exclude, aber nur auf Verzeichnisnamen angewandt, um ganze Verzeichnisse von der Rekursion auszunehmen (z.B. --exclude-dir=.git). |
| --help, --version | Hilfe / Version. |

Exit-Status: 0 falls mindestens ein Match gefunden, 1 falls kein Match, 2 falls ein Fehler auftrat (z.B. Datei nicht lesbar).

Beispiele:

- grep -i "error" *.log – sucht case-insensitive "error" in allen .log im aktuellen Ordner, gibt Trefferzeilen mit Dateipräfix aus.
- grep -R "TODO" src/ – findet rekursiv alle Vorkommen von "TODO" in Dateien unterhalb von src/ (typisch in Codebasis).
- grep -nE "^Error|Warning):" report.txt – sucht Zeilenbeginn "Error:" oder "Warning:" und gibt mit Zeilennummer aus [pq.hosting](#).
- ps ax | grep "[c]ron" – findet Prozesse mit "cron" im Namen. (Trick: [c] im Pattern matcht 'c', verhindert aber, dass der grep selbst im ps-Listing gefunden wird, da dessen Kommandozeile "grep cron" kein 'cron' substring hat sondern 'cro n' durch char class unterbrochen).

find (Dateien suchen)

Beschreibung: Durchsucht Verzeichnisse nach Dateien/Verzeichnissen, die bestimmten Kriterien entsprechen (Name, Typ, Größe, Datum, Rechte, Eigentümer etc.) [file-4fdk7rvx39azjretinarqb](#). find ist sehr mächtig: man kann logische Bedingungen kombinieren und Aktionen auf gefundene Dateien ausführen (löschen, verschieben, Exec von Kommandos etc.). Es arbeitet rekursiv ab dem angegebenen Pfad.

Syntax:

find [Start-Verzeichnisse] [Optionen/Ausdrücke] [Aktion(en)]

Wichtige Suchkriterien (Ausdrücke):

- **-name Muster** – Dateiname entspricht Wildcards (Globbing wie bei Shell, z.B. "*.txt"). Groß/Kleinschreibung beachten.
- **-iname Muster** – wie -name, aber case-insensitive.

- **-type c** – Dateityp ist c. Häufig: **f** für regular file, **d** für directory, **l** symlink, **b** blockdevice, **c** chardevice, **p** named pipe, **s** socket. Z.B. -type d nur Verzeichnissefile-4fdk7rvx39azjretinarqb.
- **-user name/uid** – Datei gehört dem Benutzer *name* (oder UID numerisch). Analog **-group name/gid**.
- **-uid +n/-n/n** – UID der Datei ist ($>/</=$) n. -uid +500 UID größer 500. (Entspricht -user nur numerisch und mit Vergleichen). Genauso -gid.
- **-size N*[bckMG]*** – Dateigröße-Kriterium. *N* kann mit Vorzeichen angegeben werden (+ größer als, - kleiner als, nichts = genau). Suffix: standardmäßig in 512-Byte Blöcken, oder **c** = Bytes, **k** = KiB, **M** = MiB, **G** = GiB. Beispiel: -size +100M Dateien größer 100 MiB[wiki.foxtom.de](#), -size -10k kleiner 10 KiB.
- **-mtime Tage** – Letzte Änderung (Modify timestamp) liegt *Tage* zurück. *Tage* als ganze Zahl: +n = älter als n Tage, -n = jünger als n Tage, n = genau n Tage (im Bereich $\pm 0.5d$). Ähnlich **-atime** (Access), **-ctime** (Statusänderung). Z.B. -mtime -7 in letzten 7 Tagen modifiziert[wiki.foxtom.de](#).
- **-newer Datei** – Datei wurde später geändert als *Datei* (Vergleich mtime). -**newermt Datum** (GNU find) erlaubt Vergleich mit einem Zeit-String, z.B. -newermt "2025-05-01".
- **-perm mode** – Dateirechte entsprechen *mode*. Mode kann exakt (z.B. 644) oder mit / oder - angegeben werden für Masken. Beispiel: -perm 644 findet genau 644. -perm -644 findet Dateien, die mindestens diese Bits haben (also rw- r-- r-- oder mehr Rechte). -perm /222 findet Dateien, bei denen irgendein Schreibbit gesetzt ist (Owner oder Group oder Other).
- **-maxdepth N** – Nur bis Tiefe N ins Verzeichnis hinab suchen (0= nur Startdir selbst). **-mindepth N** analog minimal tiefe. Z.B. find . -maxdepth 1 -type f nur Dateien im aktuellen Verzeichnis, nicht tiefer.
- **-mount** oder **-xdev** – Verlässt nicht das Dateisystem. D.h. keine Verzeichnisse auf anderen gemounteten FS durchsuchen (z.B. ignoriert find / -xdev dann andere Partitionen).
- **-prune** – Überspringt (schließt vom Durchsuchen aus) ein Verzeichnis. Bsp: find . -path "./node_modules" -prune -o -name "*.js" -print durchsucht alles außer den node_modules Ordner.

Aktionen (wenn Suchkriterien wahr sind):

- **-print** – Gibt gefundene Pfade aus (Standard, wird angenommen, wenn keine andere Aktion).

- **-ls** – Führt ls -dils auf die gefundene Datei aus (anzeige von detaillierten Infos, inkl. Inode, Größe, Rechte etc.).
- **-delete** – Löscht gefundene Dateien *unmittelbar*. Achtung: find -delete entfernt auch Verzeichnisse, wenn leer geworden. Sollte meist mit Kriterien kombiniert werden (z.B. -type f -delete nur Dateien). **Keine** Aktion mehr nach -delete wird ausgeführt (beendet Durchlauf in diesem Pfad).
- **-exec Befehl {} ;** – Führt einen externen Befehl für jede gefundene Datei aus. {} wird durch den Pfad ersetzt. Die Kommandozeile wird so oft ausgeführt wie Treffer. Beispiel: find . -name "*.bak" -exec rm {} \; löscht alle .bak-Dateien (funktional wie -delete, aber universeller)file-5y1edouysrow4zj6phjcqm. Das \; markiert Ende des -exec Befehls.
- **-exec ... {} +** – Variante, die versucht, mehrere gefundene Dateien in *einen* Aufruf zu packen (fügt die Pfade an Stelle von {} möglichst gesammelt, wie xargs). Effizienter, falls das Programm mehrere auf einmal verarbeiten kann. Z.B. -exec chmod 644 {} + ändert die Rechte aller gefundenen Files, ggf. in batched Calls.
- **-execdir** – Wie -exec, aber führt das Kommando im jeweiligen Verzeichnis der Datei aus (Working Directory wechselt). Erhöht Sicherheit bei {} mit {} containing spaces.

Beispiele:

- find /var/log -type f -name "*log" -size +1M -mtime +30 – findet Log-Dateien über 1MB, älter als 30 Tage unter /var/log.
- find . -maxdepth 1 -type d ! -name ":" -exec rm -r {} \; – löscht alle Unterverzeichnisse im aktuellen Ordner (maxdepth 1, alles was Type d ist und Name nicht ":"), dann rm -r).
- find /home/user -xtype l -delete – löscht gebrochene Symlinks (xtype l findet Links, deren Ziel nicht existiert).
- find . -perm /u+x -print – findet Dateien, bei denen das User-execute Bit gesetzt ist (also alle ausführbaren Dateien).
- find . -name ".*tmp" -execdir gzip {} \; – packt alle *.tmp im jeweiligen Verzeichnis in eine .tmp.gz (ausgeführt im Ort der Datei).

locate (Datenbank-Dateisuche)

Beschreibung: Durchsucht eine vorgefertigte Index-Datenbank nach Dateinamen (Pfaden), die einem angegebenen Suchstring entsprechenfile-4fdk7rvx39azjretinarqb. locate ist viel schneller als find, da es nicht live das Dateisystem scannt, sondern auf einer regelmäßig aktualisierten Datenbank (typisch via updatedb in Cron) basiert. Nachteil: findet keine erst sehr kürzlich hinzugefügten Dateien (bis zum nächsten

updatedb-Lauf) und standardmäßig nur solche, auf die der User zugreifen darf.

Syntax:

locate [Optionen] <Suchmuster>

Das Suchmuster ist im Grunde ein Substring oder einfache Glob. Standardmäßig sucht locate alle Datenbankeinträge, die <Suchmuster> als Teilstring enthalten.

Optionen:

| Option | Beschreibung |
|-----------------------------|---|
| -i, --ignore-case | Sucht case-insensitive (Groß/Kleinschreibung ignorieren)file-4fdk7rvx39azjretinarqb. |
| -r, --regex | Interpretiert das angegebene Muster als regulären Ausdruck (POSIX ERE) statt als einfachen Textfile-4fdk7rvx39azjretinarqb. |
| -w, --wholename | (Standardverhalten) Sucht im gesamten Pfadnamen. (Bei einigen Implementierungen war locate nur substring-suche, wholename explizit wie default, hier nur Info.) |
| -b, --basename | Sucht nur im Dateinamen selbst, nicht im gesamten Pfad. (Also ignoriert Verzeichnisnamen im Match.) |
| -c, --count | Zählt nur die Anzahl der Fundstellen und gibt diese Zahl aus, anstatt alle Ergebnisse zu listen. |
| -l N, --limit=N | Gibt maximal N Ergebnisse aus (dann Abbruch). |
| --existing | Prüft gefundene Pfade und filtert jene heraus, die aktuell nicht (mehr) existieren. (Die DB könnte veraltete Einträge haben; mit --existing werden nur noch valide Treffer angezeigt, obwohl es Zeit kostet, jeden Pfad zu verifizieren.) |
| -0, --null | Trennt die Ausgabe mit Nullbytes anstatt neuer Zeile. Nützlich, um in xargs -0 weiterzuverarbeiten. |
| -e, --nofollow | Ignoriert Einträge, die Symlinks auf nicht-existente Dateien sind (könnnte Implementation-spezifisch). |
| -S | Zeigt Statistik der Datenbank (Anzahl Einträge etc.) statt zu suchen. |
| -d DB, --database=DB | Nutzt eine alternative Datenbank-Datei anstelle der Standard-Datenbanken (/var/lib/mlocate.db z.B.). Man kann mehrere via -d db1:db2:... angeben. |

| Option | Beschreibung |
|---------------|---------------------|
|---------------|---------------------|

--help, --version Hilfe bzw. Version.

Beispiel: locate -i '\.pdf' listet alle Pfade, die ".pdf" (unabhängig von Großschreibung) enthalten – das würde i.d.R. alle PDF-Dateien auf dem System listen, die zum Zeitpunkt der letzten Datenbankaktualisierung existierten. Oft benutzt: locate <Programmname> um herauszufinden, wo ein Programm oder dessen Konfigdateien liegen.

updatedb (Update Database for locate)

Beschreibung: Baut die Datei-Datenbank für locate neu auf. Es durchsucht das Dateisystem (oder definierte Pfade) und speichert alle Dateinamen/Pfade in einer komprimierten Datenbank (typischerweise `/var/lib/mlocate/mlocate.db`). updatedb wird meist regelmäßig (täglich) via Cron ausgeführt, man kann es aber auch manuell starten, z.B. nach größeren Änderungen im FS, damit locate sofort aktuell ist.

Syntax:

`updatedb [Optionen]`

Optionen: (Häufig werden diese über `/etc/updatedb.conf` voreingestellt)

| Option | Beschreibung |
|---------------|---------------------|
|---------------|---------------------|

--prune-bind-mounts {yes|no} Ob Bind-Mountpoints ausgelassen werden (Default ja – doppelte Scans vermeiden).

--prune-paths='PFADLISTE' Doppelpunkt-separierte Liste von Verzeichnissen, die *nicht* indexiert werden sollen file-4fdk7rvx39azjretinarqb. Standardmäßig z.B. `/tmp, /proc, /sys, /run, /dev, /var/cache` etc., um unnötiges zu sparen. Dies entspricht PRUNEPATHS in `/etc/updatedb.conf`.

--prune-filenames='TypList' Liste von Dateisystemtypen, die nicht durchsucht werden (z.B. nfs,squashfs etc.). Standard meist Netzwerk-FS, Pseudo-FS (proc, sysfs) etc. file-4fdk7rvx39azjretinarqb. Entspricht PRUNEFS.

-l N **--localpaths=LIST** – falls man nur lokale Pfade indexieren will (veraltet, use prune).

-U 'PFADLISTE' Nur diese durch Doppelpunkt getrennten Pfade als Wurzel für Index verwenden (Standard "/" – gesamtes FS). Kann man nutzen um z.B. nur `/home` zu indexieren.

| Option | Beschreibung |
|----------------------|--|
| -o DBFILE | Output: speichere DB in angegebener Datei statt Standard. (So kann man separate DBs erzeugen.) |
| -f 'FS-Liste' | Nur diese Dateisystemtypen berücksichtigen (Inverse von PRUNEFS). |
| -v | Ausführliche Ausgabe während des Indizierens. |

In der Regel wird updatedb ohne Optionen (ggf. als root über sudo updatedb) ausgeführt und nutzt die System-Voreinstellungen. Danach steht die Datenbank locate zur Verfügung.

which (Welches Kommando - Pfadsuche)

Beschreibung: Zeigt den Pfad zu einem ausführbaren Programm, das in der Umgebungsvariablen \$PATH gefunden wird file-4fdk7rvx39azjretinarqb. Damit kann man herausfinden, welches Programm aufgerufen würde, wenn man einen bestimmten Befehl tippt (nützlich bei mehreren Versionen in verschiedenen Verzeichnissen).

Syntax:

which [Optionen] Befehlsname...

Optionen:

| Option | Beschreibung |
|---|--|
| -a | Zeigt alle Vorkommen des angegebenen Befehls im Suchpfad an (nicht nur das erste gefundene) file-4fdk7rvx39azjretinarqb. Z.B. which -a python listet evtl. /usr/bin/python und /usr/local/bin/python falls beide existieren. |
| --skip-dot , --skip-tilde | Ignoriert . bzw. ~ am Anfang des Pfades (also aktuelles Verzeichnis oder Home in PATH) in der Suche. (Sicherheitsaspekt – Standard bei which). |
| --show-dot , --show-tilde | Zeigt explizit, wenn gefundene Einträge in . oder ~ liegen, durch Präfix "./" oder "~/". |
| --tty-only | Zeigt nur etwas an, wenn stdout ein Terminal ist. Unterdrückt z.B. Ausgabe, wenn in Pipe (verhindert sich einzumischen). |
| --help , --version | Hilfe bzw. Versionsinfo (gilt bei GNU which; einige shells haben eingebautes which mit anderem Verhalten). |

Hinweis: which ist oft eine Shell-Funktion/Alias und verhält sich u.U. leicht unterschiedlich. Ähnliche Tools: type (shell-builtin, zeigt auch Aliases, Keywords), command -v.

Beispiel: which gcc -> /usr/bin/gcc (Pfad zum C-Compiler). which -a ls könnte z.B. zeigen /usr/bin/ls und /bin/ls falls beides symlinks aufeinander oder verschieden, etc.

Textverarbeitung und Streams (Shell, Scripting)

awk (Muster/Aktions-Scanner)

Beschreibung: AWK ist ein mächtiges Textverarbeitungswerkzeug und Skriptsprache, mit dem man Textdateien zeilenweise nach Mustern durchsuchen und pro Zeile Aktionen ausführen kannfile-4fdk7rvx39azjretinarqb. Typischerweise wird awk benutzt, um Spalten auszugeben oder zu filtern, da es Eingabezeilen automatisch in Felder zerlegt. Der Name stammt von den Anfangsbuchstaben seiner Schöpfer (Aho, Weinberger, Kernighan).

Syntax:

awk [Optionen] 'Muster { Aktion }' Datei...

Jede Zeile der Datei wird gelesen; wenn das *Muster* passt, wird die *Aktion* ausgeführt (z.B. etwas ausgeben). AWK hat eigene Variablen (\$1 = erstes Feld, \$0 = ganze Zeile) und eingebaute Rechen- und String-Features.

Optionen:

| Option | Beschreibung |
|-------------------|--|
| -F separator | Legt das Eingabefeld-Trennzeichen fest (Field Separator). Standard sind whitespace (Leerzeichen/Tabs)file-4fdk7rvx39azjretinarqb. Z.B. -F ":" für /etc/passwd mit Doppelpunkten. |
| -v Var=Wert | Setzt eine AWK-Variable <i>Var</i> auf <i>Wert</i> vor dem Verarbeiten. Z.B. -v ORS="\n\n" um eine Leerzeile nach jeder Ausgabezeile einzufügen (ORS = Output Record Separator). |
| -f Scriptdatei | Lies das AWK-Programm aus einer Datei anstatt aus der Befehlszeile. Damit kann man größere AWK-Skripte in eigenem File verwalten. |
| -m, --sandbox | (gawk) AWK in sicheren Modus schalten (kein System(), keine Schreibrechte auf FS etc). |
| --posix | Schaltet in POSIX-kompatiblen Modus (ggf. einige GNU-Erweiterungen aus). |
| -W version | Zeigt AWK-Version an (für gawk). |
| --help, --version | Hilfe / Version. |

AWK Sprachkonzepte (Kurz):

- Ein AWK-Programm besteht aus *Muster { Aktion }* Blöcken. Muster ist ein Regex oder Ausdruck, Aktion ist Code in AWK.
- Felder: Eingabezeile wird in Felder zerlegt, zugreifbar als \$1, \$2, ..., \$NF (NF=Anzahl Felder). \$0 ist die ganze Zeile.
- Beispiele:
 - awk '{ print \$1 }' file – gibt die erste Spalte jeder Zeile aus (mit Standard-Trenn whitespace).
 - awk '/error/ {print NR ":" \$0}' log.txt – sucht Zeilen, die "error" enthalten (Regex als Muster), und druckt Zeilennummer und kompletten Inhalt.
 - awk -F, '\$3 > 100 { sum += \$3 } END { print "Summe:", sum }' data.csv – mit Komma als Separator summiere Wert der 3. Spalte aller Zeilen >100 in Spalte3; am Ende (END-Block) gib Summe aus.
- AWK hat viele eingebaute Funktionen (Mathe, String), assoziative Arrays, Kontrollstrukturen (if, for, while).

sed (Stream Editor)

Beschreibung: sed ist ein nicht-interaktiver Stream-Editor zum Bearbeiten von Texteingaben nach Mustern, insbesondere für Suchen&Ersetzen oder das Extrahieren von Zeilenfile-4fdk7rvx39azjretinarqb. sed liest zeilenweise Eingabe (Dateien oder STDIN), wendet die angegebenen *Editier-Befehle* darauf an und schreibt das Ergebnis auf STDOUT (oder optional in-place zurück in Dateien). Am verbreitetsten ist die *Substitution* mit dem s/alt/neu/ Kommando.

Syntax:

sed [Optionen] 'Script' Datei...

Das *Script* besteht aus einem oder mehreren Editierbefehlen, die auf die Eingabe angewendet werden. Beliebte sed-Befehle: **s** (substitute), **p** (print), **d** (delete), **q** (quit), **{}** für Gruppen, Adressierungsmodi (Zeilen-Nr oder Regex).

Optionen:

| Option | Beschreibung |
|----------------------------|--|
| -n, --quiet/-silent | Suppress normal output. Normal drückt sed jede Eingabezeile (ggf. modifiziert) automatisch. Mit -n wird nichts automatisch gedruckt, nur explizite p (print) Befehle geben ausfile-4fdk7rvx39azjretinarqb. Typisch in Kombination mit p in Script, um nur bestimmte Zeilen auszugeben. |

| Option | Beschreibung |
|----------------------------------|---|
| -e Script | Gibt das zu verwendende sed-Skript (Editierbefehle) direkt an. Kann mehrfach genutzt werden für mehrere Befehle in Reihenfolge. Wenn nur ein Skript-Argument ohne -e gegeben ist, kann -e weggelassen werden. |
| -f Datei | Lies das sed-Skript aus einer Datei. Erlaubt komplexere Edits aus einem File anstatt Kommandozeile. |
| -i[SUF], --in-place[=SUF] | Bearbeitet Dateien <i>in place</i> , d.h. schreibt Änderungen direkt zurück in die Eingabedateien. Optional kann ein <i>Suffix</i> angegeben werden, um vor Änderungen ein Backup der Originaldatei anzulegen (z.B. -i.bak). Ohne Suffix wird die Datei direkt überschrieben ionos.de . |
| -r, -E | Schaltet die Verwendung <i>erweiterter regulärer Ausdrücke</i> ein (so dass in Patterns +, ?, {}, etc. ohne Backslash verwendet werden können). Bei GNU sed ist -r (old syntax) und -E (wie bei BSD sed) identisch. |
| -s | Treat files separately. Normalerweise bei mehreren Dateien behandelt sed den Input als einen langen Stream. Mit -s wird nach jeder Datei der interne Zustand (z.B. Zeilennummer, \$-Regex-Flag) zurückgesetzt. |
| -z, --null-data | Input wird als Null-terminiert (Zeilen getrennt durch ASCII NUL statt newline) angenommen. Ermöglicht Verarbeitung von Binär oder multiline-chunks. Entspricht dann, dass . nicht auch \n matcht und \$ nur am NUL erkennt. (GNU sed) |
| --follow-symlinks | Wenn in-place auf Symlinks angewendet, ändert sed standardmäßig den Link selbst (also das Zielfile, aber eben durch den Link?). Mit dieser Option folgt sed dem Symlink und editiert die Ziel-Datei. (GNU sed) |
| --sandbox | Verhindert, dass sed w oder s//w zum Schreiben verwendet oder ! Shell-Aufrufe macht. (GNU sed 4.4+) |
| --help, --version | Hilfe / Version. |

Häufige sed-Anwendungen:

- **Substitution (Ersetzen):**

Syntax: s/REGEX/REPLACEMENT/[Flags]. Ersetzt das erste Vorkommen von REGEX in jeder Zeile durch REPLACEMENT. Flags: g = global (alle Vorkommen in Zeile ersetzen), i = case-insensitive, p = print (selbst wenn -n), w file = gefundene Zeilen nach file schreiben.

Beispiel: sed -i 's/oldword/newword/g' file.txt – ersetzt *oldword* durch *newword* überall in file.txt, direkt in der Datei `file-5y1edouysrow4zj6phjcqm`.

Besonderheit: Im Ersatz kann man & verwenden für den gesamten Match oder \1 ... für Capturing Groups aus dem Regex.

- **Zeilen löschen:**

z.B. sed '/^#/ d' config – löscht alle Zeilen, die mit # beginnen (d = delete). Oder sed '5,10d' file löscht Zeilen 5 bis 10.

- **Nur bestimmte Zeilen ausgeben:**

Mit -n und p. Z.B. sed -n '1,5p' file druckt nur Zeilen 1-5 (andere werden durch -n nicht ausgegeben). Oder sed -n '/ERROR/p' log druckt nur Zeilen mit "ERROR".

- **Mehrere Edits:**

z.B. sed -e 's/foo/bar/' -e '/BAZ/d' – erst foo->bar ersetzen, dann Zeilen mit "BAZ" löschen. (Die Reihenfolge ist wichtig).

- **In-Place Backup:**

sed -i.orig 's/^#/ #' *.conf – kommentiert alle Zeilen in *.conf aus, speichert vorher Backup mit .orig-Suffix.

Sed kann viel mehr (Insert/Append mit i\ und a\, Transform mit y///, Shell-Befehle mit !, Holding Space etc.), was in Tiefe für Prüfung wohl nicht nötig ist. Wichtig ist Suchen&Ersetzen und Basisfilter.

tee (Output verdoppeln)

Beschreibung: Liest die Standard-Eingabe und schreibt sie *gleichzeitig* auf die Standard-Ausgabe **und** in eine oder mehrere Dateienfile-4fdk7rvx39azjretinarqb. Es "verzweigt" den Datenstrom (daher der Name T-Stück). So kann man die Ausgabe eines Kommandos in eine Datei umleiten, aber trotzdem im Terminal sehen.

Syntax:

... | tee [Optionen] <Datei>...

Optionen:

| Option | Beschreibung |
|--------------------------------|---|
| -a, --append | Hängt die Ausgabe an bestehende Dateien an, anstatt sie zu überschreibenfile-4fdk7rvx39azjretinarqb. Standard ohne -a überschreibt Ziel-Dateien neu. |
| -i, --ignore-interrupts | Ignoriert Interrupt-Signale (SIGINT). D.h. Ctrl+C wirkt nicht auf tee selbst (nur auf vorheriges Kommando?), damit tee garantiert die Daten schreibt. |
| --output-error MODE | Verhaltensweise falls ein Schreibfehler auftritt: warn (Fehler melden, aber weiter versuchen andere outputs zu schreiben), exit (sofort) |

| Option | Beschreibung |
|---------------|---|
| | beenden, evtl. ungeschriebene outputs gehen verloren). Standard ist warn. |

Anwendung:

- make 2>&1 | tee build.log – Zeigt alle Ausgaben von make im Terminal und speichert sie gleichzeitig in *build.log*.
- ls -l | tee -a listing.txt – Ausgabe von ls -l normal anzeigen und ans Ende von listing.txt anfügen.
- echo "strict mode on" | sudo tee /proc/sys/kernel/yama/ptrace_scope – Durch sudo tee kann man Inhalte in Dateien schreiben, für die man Rechte braucht, während Eingabe evtl. aus einem un-privilegierten Echo kommt.

Tee ist also nützlich bei Logging und bei Verwendung mit sudo zum in Dateien schreiben, da sudo echo >> file wegen Redirection sonst nicht klappt.

xargs (Execute Arguments)

Beschreibung: Liest Eingaben (Standardinput oder Datei) und führt daraus Kommandoaufrufe zusammengebaut aus. Typisch, um die Ausgabe eines Programms als Argumentliste für ein anderes Programm zu verwendenfile-4fdk7rvx39azjretinarqbfile-4fdk7rvx39azjretinarqb. xargs nimmt zeilen- oder spaltengetrennte Wörter und packt möglichst viele davon in einen Aufruf eines angegebenen Kommandos (Standard: echo). Damit überwindet man die Shell-Längenbeschränkung und kann sequentiell Befehle auf Listen anwenden.

Syntax:

xargs [Optionen] [Befehl [Initial-Argumente]]

Wenn kein Befehl angegeben ist, benutzt xargs per Default echo (einfaches Ausgeben).

Optionen:

| Option | Beschreibung |
|-------------------|--|
| -n MaxArgs | Maximal <i>MaxArgs</i> Argumente pro aufgerufenem Kommando verwendenfile-4fdk7rvx39azjretinarqb. Z.B. -n 1 bedeutet, xargs nimmt jeweils 1 Argument und führt das Kommando damit aus (also für jede Eingabezeile einmal). Ohne Angabe bündelt xargs so viele Argumente wie möglich pro Aufruf, bis die Systemlimite (Getconf ARG_MAX) fast erreicht ist. |

| Option | Beschreibung |
|------------------------------|---|
| -L MaxLines | Nimmt maximal <i>MaxLines</i> Input-Zeilen pro Kommando-Aufruf. Z.B. -L 1 bewirkt, dass jede einzelne Zeile als eine xargs-Ausführung behandelt wird (selbst wenn Zeile mehrere Wörter enthält). |
| -d Delim | Verwendet <i>Delim</i> als Trennzeichen anstelle von Whitespace oder newline. Kann auch mehrere Zeichen (String) sein (GNU). Beispiel: -d ':' um : als Separator zu nehmen. |
| -0, --null | Erwartet Input-Felder durch Nullbyte getrennt (wie von find -print0 oder grep -Z erzeugt). Dies ermöglicht sichere Übergabe von Dateinamen mit Leer-/Sonderzeichen stackoverflow.com . |
| -E EOFSTR | Definiert eine Zeichenfolge, die als <i>Ende der Eingabe</i> interpretiert wird. Wenn xargs diese Zeichenfolge liest, bricht es die weitere Verarbeitung ab (auch wenn nicht EOF). |
| -e | (veraltet, äquivalent zu -E ohne Parameter) – historically to specify EOF marker, oft nicht mehr benötigt. |
| -I ReplaceStr | Nutzt <i>ReplaceStr</i> als Platzhalter in dem auszuführenden Kommando, der durch den gesamten Input (pro Aufruf) ersetzt wirdfile- |
| -J ReplaceStr | 4fdk7rvx39azjretinarqb. Z.B. xargs -I {} mv {} /backup nimmt jeden Input (eine Datei) und substituiert anstelle von {} im mv-Aufruf. Bei -I wird implizit -L 1 angenommen (d.h. pro Zeile ein Aufruf). |
| -t, --verbose | (BSD xargs) Ähnlich -I, aber ersetzt nur einmal in der Kommandozeile (kann mit -n kombinieren). In GNU xargs gibt es -J nicht. |
| -p, --interactive | Gibt das Kommando, das ausgeführt wird, vorher auf stderr aus (also "echo" what is being done). Zum Debuggen hilfreich. |
| -r, --no-run-if-empty | Fragt vor jedem Kommando nach Bestätigung (y/n), ähnlich wie -t zeigt es Befehl. Bei 'y' ausführen, 'n' überspringen. |
| -0, --null | Führt das Kommando nicht aus, wenn keine Argumente aus Input entstehen. Standard bei GNU xargs ist, <i>kein</i> Leerlaufaufruf zu machen (sprich diese Option ist implizit an). In POSIX könnte Standard anderes sein. |
| | (siehe oben) Null-terminierte Eingabe verwenden – wichtig in Kombination mit find -print0 um sicher Dateilisten zu pipen ohne Probleme mit Sonderzeichen stackoverflow.com . |

| Option | Beschreibung |
|--------------------------------|---|
| --max-chars=K, -s K | Maximal K Zeichen pro Kommandozeilen-Aufruf verwenden (inklusive Kommando und env). Standard an System ARG_MAX orientiert (oft ~131072). Manchmal zu reduzieren bei seltsamen Umgebungen. |
| --max-procs=N, -P N | Bis zu N Prozesse <i>parallel</i> ausführen statt nacheinander. Z.B. -P 4 führt bis zu 4 Instanzen des Kommandos gleichzeitig aus, um Parallelisierung zu erreichen. N=0 bedeutet unbegrenzt viele parallel (Vorsicht!). Hilfreich z.B. in Multi-Core-Systemen. |
| --process-slot-var=NAME | Setzt eine Umgebungsvariable \$NAME in den aufgerufenen Prozessen, die eindeutige ID des xargs-Slots enthält (0..N-1) – bei -P parallel. Hilft um Ausgaben zuzuordnen. |
| -a Datei | Liest Input aus <i>Datei</i> anstatt STDIN. So kann xargs auch ohne Pipe aus File speisen. |
| --help, --version | Hilfe/Version. |

Beispiele:

- find /tmp -type f -name '*.txt' -print0 | xargs -0 rm -f – Löscht alle .txt-Dateien unter /tmp (sicher gegenüber Sonderzeichen, dank -print0 und -0).
- grep -l "ERROR" *.log | xargs -r zip errors.zip – Sucht in allen .log nach "ERROR", die -l Option listet nur Dateinamen mit Treffer. Diese Liste wird an xargs übergeben, das dann zip errors.zip <files> ausführt. Falls kein Treffer, -r sorgt dafür, dass zip nicht ohne Argument aufgerufen wird.
- cat list.txt | xargs -I {} echo File: {} >> files.txt – Für jede Zeile aus list.txt, führe echo File: <line> aus und append an files.txt. (So fügt man "File: " vor jede Zeile).
- seq 1 1000 | xargs -n 100 > numbers.txt – Nimmt die Zahlen 1 bis 1000 (jede auf neuer Zeile) und packt jeweils 100 Zahlen in eine Zeile, schreibt nach numbers.txt.
- printf "%s\0" * | xargs -0 -n 1 -P 4 sha256sum – Berechnet parallel (4 at a time) die SHA256 Summen aller Dateien im aktuellen Verzeichnis. printf "%s\0" * generiert Null-separierte Liste aller Dateien, xargs nimmt je 1 (-n1) und startet bis zu 4 sha256sum gleichzeitig.

Software- und Paketmanagement

tar (Tape ARchiver)

Beschreibung: Archiviert mehrere Dateien zu einer einzigen Datei (tarball). Tar kann Dateien **bündeln** (zu .tar Archiven) und in Zusammenarbeit mit Kompressionsprogrammen (gzip, bzip2, xz etc.) zugleich komprimierenfile-4fdk7rvx39azjretinarqb. Oft verwendet, um Verzeichnisse oder viele Dateien als ein Archiv zu speichern oder zu verteilen (.tar.gz, .tgz, .tar.bz2 etc.).

Syntax:

tar [Optionen] -f <Archiv> [Dateien/Verzeichnisse...]

Tar hat **Modi** (Mutual exclusive Optionen): Erstellen, Extrahieren, Listen etc. Üblich ist die kombinierte Kurzoption wie -czf oder -xvf. Wichtig: -f (file) gibt an, welches Archivfile zu benutzen ist, und kommt als *ein* Argument, meist zuletzt bei Short-Options-Kombination.

Haupt-Modi:

- **-c, --create** : Erstelle ein neues Archiv aus den
(Fortsetzung)

tar (Tape Archiver)

Beschreibung: Erstellt oder entpackt Archive aus mehreren Dateien. tar bündelt Dateien in **einem** Archiv (Containerdatei mit Endung .tarfile-4fdk7rvx39azjretinarqb】 . Oft wird tar mit Kompressionsprogrammen kombiniert (z.B. gzip oder bzip2), um komprimierte Archive wie .tar.gz zu erstellefile-4fdk7rvx39azjretinarqb】 .

Syntax:

tar <Modus-Option> [Optionen] -f <Archivdatei> [Dateien/Verzeichnisse...]

Tar benötigt **eine** der Hauptoptionen: -c (create), -x (extract) oder -t (list). Dazu wird **-f** angegeben, um den Namen der Archivdatei zu bestimmen. Häufig werden mehrere Kurzoptionen kombiniert (z.B. -czvf für "create, gzip, verbose, file").

Wichtige Optionen:

- **-c – create:** Neues Archiv erstellefile-4fdk7rvx39azjretinarqb】 .
- **-x – extract:** Dateien aus Archiv entpacken.
- **-t – list:** Inhalt des Archivs auflisten (nicht extrahieren).
- **-f Datei** – Gibt den Archivnamen an (Pfad/Name der Tar-Datei). Diese Option **muss** direkt gefolgt von dem Dateinamen kommen (z.B. -f archive.tar).
- **-v – verbose:** Ausführliche Ausgabe (listet während der Verarbeitung die Dateinamen auf).

- **-z** – Filter durch **gzip**: Bei -c wird das Archiv mit gzip komprimiert (*.tar.gz); bei -x wird erwartet, dass das Archiv gz-komprimiert ist `file-4fdk7rvx39azjretinarqb`. (Äquivalent: --gzip oder mit externem gzip pipen).
- **-j** – Filter durch **bzip2**: Erstellt/entpackt ein bzip2-komprimiertes Archiv (*.tar.bz2). (Alternativ --bzip2.)
- **-J** – Filter durch **xz**: Für .tar.xz-Archive (--xz).
- **-A – concatenate**: Hängt ein oder mehrere Archive an ein bestehendes Archiv an.
- **-d – diff/compare**: Vergleicht Archiveinträge mit Dateien im Dateisystem (prüft, ob Unterschiede bestehen).
- **-C DIR – Change Directory**: Wechselt ins Verzeichnis *DIR*, bevor Dateien hinzugefügt oder extrahiert werden. Z.B. tar -xvf archive.tar -C /tmp entpackt nach /tmp.
- **--strip-components=N** – Beim Extrahieren lässt tar N führende Pfadkomponenten weg. Z.B. entpackt /usr/local/share/doc/file als file bei --strip-components=3. Praktisch, um Archiv ohne Top-Level-Ordner auszupacken.
- **--exclude=PATTERN** – Schließt Dateien, die auf PATTERN passen, vom Archivieren aus. Z.B. --exclude="*.o".
- **--wildcards, --anchored** – steuert, wie Muster bei -t oder -x --wildcards -f archive "*.txt" interpretiert werden (glob-Muster an/aus etc.).
- **--numeric-owner** – Beim Entpacken Besitzer/Gruppen als IDs übernehmen (statt Namen auf dem Zielsystem zuzuordnen).

Beispiele:

- Archiv erstellen: tar -cvf archive.tar Verzeichnis/ erzeugt ein unkomprimiertes Archiv *archive.tar* mit allen Dateien aus *Verzeichnis/*.
- Mit Kompression: tar -czvf etc.tar.gz /etc packt ganz /etc als gzip-komprimiertes Archiv.
- Entpacken: tar -xvzf backup.tar.gz -C /restore entpackt *backup.tar.gz* nach */restore*.
- Auflisten: tar -tvf archive.tar zeigt die Dateiliste mit Größen, Timestamp, Rechten etc.

gzip (GNU Zip Compression)

Beschreibung: Komprimiert oder dekomprimiert Dateien im **gzip-Format** (.gz). Es ersetzt beim Komprimieren standardmäßig die Originaldatei durch die komprimierte

Version (und fügt .gz an file-4fdk7rvx39azjretinarqb】 . Gzip ist schneller, aber erzeugt etwas größere Dateien als bzip2/xz. Häufig mit tar (siehe oben) eingesetzt.

Syntax:

gzip [Optionen] [Datei...]

gunzip [Optionen] [Datei...]

(gunzip ist äquivalent zu gzip -d.)

Optionen:

- **-c, --stdout:** Schreibt das komprimierte Ergebnis auf die Standardausgabe, anstatt die Datei zu ersetzen. Original bleibt erhalten (kann in Pipeline genutzt werden[ionos.de](#)】 .
- **-d, --decompress:** Dekomprimiert die .gz-Datei (auch via gunzip aufrufbar[ionos.de](#)】 .
- **-k, --keep:** Behält Originaldatei beim Komprimieren/Entpacken (standardmäßig wird Original gelöscht).
- **-r, --recursive:** Komprimiert alle Dateien in angegebenen Verzeichnissen rekursiv.
- **-1 bis -9:** Einstellen der Kompressionsstufe (1 = schnellste/geringste Kompression, 9 = langsamste/beste). Standard ist 6.
- **-t, --test:** Prüft eine gz-Datei auf Integrität, ohne sie zu entpacken (exit-Status zeigt OK/NOK).
- **-l, --list:** Zeigt Informationen über eine komprimierte Datei: Originalgröße, komprimierte Größe, Kompressionsrate, Zeitstempel.
- **--stdout, -c** (nochmal): Wird oft mit -d kombiniert, um nach stdout zu entpacken (z.B. gzip -dc file.gz).
- **-N, --name:** Bewahrt ursprünglichen Dateinamen und Zeitstempel (in .gz-Header) oder stellt diese beim Entpacken wieder her (ist Standardverhalten bei GNU gzip).
- **--help, --version:** Hilfe bzw. Versionsinfo.

Beispiele:

- gzip bigfile komprimiert *bigfile* zu *bigfile.gz* (und löscht *bigfile*).
- gzip -9 *.txt erstellt maximal komprimierte *.txt.gz Dateien.
- gunzip data.csv.gz entpackt *data.csv.gz* zurück zu *data.csv*.

- gzip -c file.txt > file.txt.gz komprimiert nach stdout und schreibt in separate Datei (Original bleibt).
- In Pipe: tar -cvf - /folder | gzip > folder.tar.gz – tar erzeugt Archiv auf stdout, gzip komprimiert es zu *folder.tar.gz*.

xz (LZMA Compressor)

Beschreibung: Komprimiert/Dekomprimiert Dateien mit dem **xz/LZMA2-Algorithmus**. XZ erreicht hohe Kompressionsraten, aber benötigt mehr CPU und RAM. Dateiendung: .xz. Wird z.B. für Kernel-Quellen-Archive genutzt. Syntax und Optionen ähneln gzip.

Syntax:

xz [Optionen] [Datei...]

unxz [Optionen] [Datei...]

(xz -d entspricht unxz.)

Wichtige Optionen:

- **-z** (Default): Komprimieren (Standardmodus, kann entfallen).
- **-d, --decompress**: Entpacken.
- **-k, --keep**: Originaldatei behalten (nicht löschen).
- **-f, --force**: Vorhandene .xz-Dateien überschreiben, falls vorhanden; oder erzwingt Komprimieren auch bei scheinbar nicht sinnvoll (z.B. sehr kleine Datei).
- **-9** (auch -0 .. -9): Kompressionsgrad (0 = none, 6 = Standard, 9 = max). XZ erlaubt auch -e für extreme Mode (noch etwas bessere Kompression, deutlich langsamere Kodierung).
- **-T N, --threads=N**: Nutzung von N Threads für Kompression (für große Dateien auf Multi-Core). Standard 1. (Für Dekomprimierung wird threads automatisch benutzt).
- **-c, --stdout**: Ergebnis auf stdout schreiben (bzw. von stdin lesen und auf stdout entpacken).
- **-l, --list**: Listet das .xz-Archiv (Inhalt, ratio, etc. – bei Single-File Komprimierung liefert original und comp size).
- **-q, -v**: quiet bzw. verbose Mode.
- **-C, --check=TYPE**: Prüfsummenverfahren wählen (Standard CRC64). Kann z.B. auf CRC32, SHA256 gestellt werden.

Beispiele:

- `xz hugefile` erzeugt `hugefile.xz` (Original wird gelöscht).
- `xz -dk archive.tar.xz` entpackt `archive.tar.xz` nach `archive.tar` und behält auch die `.xz`.
- `tar -c /usr/bin | xz -9e > bin.tar.xz` erstellt maximal-komprimiertes tar-Archiv.
- `unxz data.csv.xz` oder `xz -d data.csv.xz` entpackt Datei.
- `xzcat file.txt.xz = xz -dc file.txt.xz` entpackt nach stdout (ähnlich zcat).

bzip2 (Burrows–Wheeler Compressor)

Beschreibung: Komprimiert mit dem älteren bzip2-Algorithmus (.bz2). Langsamer als gzip, aber bessere Kompression; heute oft von xz abgelöst. Bedienung ähnlich gzip.

Syntax: `bzip2 [Optionen] [Datei...]` – bunzip2 zum Entpacken.

Optionen (Auswahl): `-k` (keep file), `-d` (decompress), `-1 .. -9` (Stufe, Standard 9 bei bzip2), `-c` (stdout), `-t` (Test), `-v` (verbose stats).

zip (Archiv mit Kompression, ZIP-Format)

Beschreibung: Erstellt ZIP-Archive (mit Kompression) oder fügt Dateien hinzu. Anders als tar erwartet zip direkt die zu archiviernden Dateien und komprimiert sie einzeln ins Archiv. ZIP-Archive enthalten Verzeichnisstruktur + komprimierte Dateien (beliebtes Format unter Windows). unzip entpackt sie.

Syntax:

`zip [Optionen] <Archiv.zip> <Dateien...>`

Wichtige Optionen:

- `-r` – Rekursiv Verzeichnisse einschließen. Z.B. `zip -r archiv.zip Ordner/`.
- `-9` (bzw. `-1 .. -9`) – Kompressionsgrad (9 = beste, Standard meist 6).
- `-q` – quiet, `-v` – verbose (gibt Kompressionsraten aus).
- `-m` – move: Dateien nach dem Hinzufügen ins Archiv löschen (verschieben).
- `-j junk paths`: Pfadnamen ignorieren, legt alle Dateien „flach“ im Archiv ab (ohne Verzeichnisstruktur).
- `-x <Pattern>` – bestimmte Dateien ausschließen (z.B. `-x "*.o"`).
- `-F` – Versucht ein beschädigtes Archiv zu reparieren (freshen).
- `-u update`: nur neuere Dateien (bzgl. Zeitstempel) ins Archiv aktualisieren.
- `-g grow`: hängt Dateien an ein bestehendes ZIP an (statt neu zu erstellen).

- **-e** – erstellt ein verschlüsseltes Archiv (fordert Passwort-Eingabe, traditionelles ZipCrypto; **-P <pw>** unsicher ohne Prompt).

unzip Wichtige Optionen:

unzip <file.zip> entpackt im aktuellen Verzeichnis. **-d <DIR>** entpackt ins DIR. **-l** listet Inhalte. **-t** testet. **-x** schließt beim Entpacken bestimmte aus. **-n** überschreibt nichts, **-o** überschreibt ohne Nachfrage.

Paketmanagement-Befehle (Distributionen)

rpm (RPM Package Manager)

Beschreibung: Das Low-Level-Werkzeug zum Installieren, Entfernen und Abfragen von RPM-Paketen (Red Hat, CentOS, Fedora, SUSE etc.) . rpm arbeitet auf .rpm-Dateien und der lokalen Paketdatenbank. Häufig nutzt man für höhere Ebene yum oder dnf, aber rpm bleibt wichtig für direkte Paketoperationen.

Syntax (Basis):

rpm [Optionen] <Paketdatei> / rpm [Optionen] <Paketname>

Operationen (Hauptoptionen, nicht kombinierbar):

- **-i – install** ein neues RPM-Paket . Beispiel: rpm -i nginx.rpm. Erfordert Root-Rechte.
- **-U – upgrade** (installiert neu oder aktualisiert älteres Paket). Wird häufig anstelle -i benutzt, da es sowohl installiert als auch updatet.
- **-F – freshen**: wie -U, aber nur wenn schon ältere Version installiert ist (sonst kein Install).
- **-e – erase** (deinstalliert) ein installiertes Paket nach Paketname . Beispiel: rpm -e nginx.
- **-q – query**: Abfrage-Modus (ohne Änderungen) . Kann mit Zusätzen Info liefern. Beispiel: rpm -q nginx prüft, ob Paket installiert ist und welche Version.

Wichtige Optionen/Zusätze:

Install/Upgrade Optionen:

- **-h** – zeigt Hash-Fortschriffs Balken beim Installieren ("#" -Zeichen).
- **-v** – verbose, mehr Ausgabe. Oft kombiniert -Uvh oder -ivh.
- **--force** – erzwingt Installation, selbst wenn z.B. ein Paket bereits installiert ist oder Dateien konfigurieren. (Vorsicht!).
- **--nodeps** – ignoriert Abhängigkeits-Prüfung (installiert trotzdem).

- **--replacefiles** – überschreibt Dateien, die zu anderen Paketen gehören könnten (Konflikte ignorieren).
- **--nossripts** – führt keine Pre/Post-Install-Skripte im RPM aus (für spezielle Fälle).

Query Optionen (mit -q):

- **-a** – query *all* (listet alle installierten Pakete).
- **-i** – (mit -q) zeigt detaillierte Paketinformationen (Name, Version, Summary, Größe, Install-Datum etc.linux-praxis.de) . Z.B. rpm -qi bash.
- **-l** – listet alle vom Paket installierten Dateien. ` rpm -ql <paket>file-5y1edouysrow4zj6phjcqm` .
- **-p** – query ein *Paketfile* (nicht installiert). Z.B. rpm -qp -l paket.rpm zeigt, welche Dateien im RPM drin sind.
- **-f Dateipfad** – findet heraus, zu welchem installierten Paket eine bestimmte Datei gehört. Z.B. rpm -qf /usr/bin/ls -> coreutils-8.xx.
- **-c** – listet nur Konfigurationsdateien des Pakets (mit -ql kombinierbar: -qlc).
- **--changelog** – zeigt das Changelog des Pakets (sofern im RPM hinterlegt).
- **--provides, --requires** – zeigen welche Provides/Requires (Abhängigkeiten) ein Paket liefert bzw. benötigt. Z.B. rpm -q --requires httpd.
- **-s** – zeigt den Status der Dateien (ob sie verändert wurden seit Installation, (vom user modifiziert = **M**, missing etc.)). rpm -qs <paket>.

Erase (Uninstall) Optionen:

- **--nodeps** beim -e: ignoriert Abhängigkeitsprobleme (entfernt auch wenn andere Pakete darauf angewiesen sind - kann zu kaputten Programmen führen).

Beispiele:

- rpm -Uvh httpd-2.4.rpm – installiert Apache httpd, zeigt Hashtags und Namen.
- rpm -qa | grep kernel – listet alle installierten Pakete, filtert "kernel".
- rpm -qpi package.rpm – zeigt Infos über nicht installiertes RPM.
- rpm -e httpd – entfernt Apache-Paket.

yum (Yellowdog Updater Modified)

Beschreibung: Ein High-Level-Paketmanager für RPM-basierte Distr. (v.a. CentOS/RHEL 7 und älter). yum löst Abhängigkeiten automatisch auf und kann aus Repositories Pakete installieren, entfernen, aktualisieren . (In neueren RHEL8+ abgelöst durch dnf, aber Syntax ähnlich.)

Syntax:

yum [Optionen] <Sub-Befehl> [Paketnamen/...]

Haupt-Subkommandos:

- **install Paket** – Installiert ein Paket (aus den konfigurierten Repos oder einer RPM-Datei). Z.B. yum install httpd. Resolved dependencies automatisch und fragt um Bestätigung.
- **update [Paket]** – Aktualisiert alle installierten Pakete (wenn kein Name angegeben) oder ein bestimmtes. (yum update führt System-Update durch).
- **remove Paket, erase** – Entfernt angegebene Pakete mit Abhängigkeiten, die dadurch verwiesen.
- **search Stichwort** – Durchsucht die Paketnamen, Beschreibungen etc. nach dem Stichwort (um herauszufinden, welches Paket etwas bietet).
- **info Paket** – Zeigt Infos zu Paket(en) (Name, Version, Summary, Größe, Repo, Abh.) ähnlich rpm -qi aber aus Repos wenn nicht installiert.
- **list [installed|available] <Pattern>** – Listet Pakete (installierte oder verfügbare). Ohne Parameter alle Pakete, optional mit Pattern filtern. Z.B. yum list available 'php*'.
- **provides Datei/Capability** – Findet heraus, welches Paket eine bestimmte Datei oder "Provide" bereitstellt (ähnlich rpm -qf/--whatprovides). Z.B. yum provides /usr/bin/zip.
- **clean <was>** – Bereinigt lokale Caches: z.B. yum clean all (löscht Metadaten- und Paketcache), oder yum clean metadata etc.
- **check-update** – Zeigt verfügbare Updates, ohne sie zu installieren (Exitcode 100 wenn Updates vorhanden, 0 wenn keine).
- **repolist** – Listet eingerichtete Repositories und ob aktiviert.
- **groupinstall / groupremove** – Installiert/entfernt ganze Paketgruppen (z.B. yum groupinstall "Development Tools").

Optionen (Auswahl):

- **-y** – Automatisch "yes" beantworten (nicht nach Bestätigung fragen). Wichtig für Skript.
- **-q** – Weniger Ausgabe (quiet).
- **-C** – Nur aus dem Cache arbeiten, keine neuen Metadaten laden (offline).

- **--enablerepo=X, --disablerepo=Y** – Repositories gezielt ein-/ausschalten für diesen Befehl (Overrides config).
- **--nogpgcheck** – Deaktiviert GPG-Signaturprüfung der Pakete (unsicher, nur falls Repo-Key nicht installiert etc.).
- **--downloadonly** – Lädt die Pakete herunter, installiert aber nicht.
- **--security** – (bei update/info/list) berücksichtigt nur sicherheitsrelevante Updates (wenn Repo das unterstützt).

Beispiele:

- yum install nginx php php-mysql – installiert mehrere Pakete in einem Rutsch.
- yum update – bringt System auf neuesten Stand.
- yum search "pdf viewer" – sucht nach Paketen mit "pdf viewer" im Namen/Beschreibung.
- yum remove httpd – deinstalliert Apache (und abhängige Module, wenn keine anderen es benötigen).
- yum provides "*bin/rar" – findet Paket, das rar-Binary liefert.

(*Hinweis: Auf neueren Systemen ersetzt dnf den Befehl yum. dnf Syntax und Subcommands sind weitgehend gleich, aber z.B. dnf upgrade statt yum update.)*

apt (Advanced Package Tool – Debian/Ubuntu)

Beschreibung: Höher-level Frontend für Debian-Paketverwaltung (dpkg). apt kombiniert Funktionen von apt-get und apt-cache in einem Kommando. Es verwaltet .deb-Pakete: Installation, Updates aus Repositories, Paket-Suche usw]. (Ubuntu 16.04+ und Debian Stretch+ empfehlen apt für interaktive Nutzung, während apt-get skriptfreundlich bleibt.)

Syntax:

apt [Optionen] <Sub-Befehl> [Pakete...]

Wichtige Sub-Befehle:

- **update** – Aktualisiert die Paketquellen-Datenbank (lädt neue Paketlisten von den konfigurierten Repos). Immer vor Install/Upgrade ausführen.
- **upgrade** – Installiert verfügbare Updates für alle **schon installierten** Pakete (führt keine Entfernung durch). Hält Pakete zurück, wenn Abhängigkeitswechsel nötig wären. (vgl. *apt full-upgrade*).
- **full-upgrade** – Führt vollständige Distribution-Upgrades durch, d.h. aktualisiert alle Pakete und darf auch Pakete entfernen oder neue Abhängigkeiten

hinzufügen, um ein konsistentes System herzustellen (entspricht apt-get dist-upgrade).

- **install <pkg>...** – Installiert angegebene Pakete (aus Repos oder .deb Dateien via Pfad). Löst Abhängigkeiten auf und fragt bei Bedarf. Akzeptiert Paket=Version oder Paket:arch.
- **remove <pkg>...** – Entfernt Pakete, lässt jedoch veränderte Konfigurationsdateien bestehen (als .config in /etc).
- **purge <pkg>...** – Entfernt Pakete **inklusive** Konfigurationsdateien (bereinigt vollständig).
- **search <regex>** – Durchsucht Paketnamen und Beschreibungen nach einem Begriff (regex möglich). *Beispiel:* apt search editor pdf.
- **show <pkg>** – Zeigt detaillierte Informationen zu einem Paket (Version, Abhängigkeiten, Beschreibung, Maintainer etc.), egal ob installiert oder nur verfügbar.
- **list** – Listet Pakete. Mit --installed, --upgradeable oder --all-versions. (Ähnlich dpkg -l aber filterbar). Z.B. apt list --installed | grep nginx.
- **autoremove** – Entfernt automatisch Pakete, die als Abhängigkeiten installiert wurden und nun nicht mehr benötigt werden (Waisen).
- **depends <pkg>, rdepends <pkg>** – Zeigt Abhängigkeiten bzw. Reverse-Abhängigkeiten eines Pakets (welche Pakete benötigen es).
- **edit-sources** – Öffnet die sources.list im Editor (um Repos zu bearbeiten).

Optionen:

- **-y, --yes** – Automatisch alle Fragen mit Yes beantworten (nicht interaktiv bestätigen). Praktisch für Skripte.
- **-q** – Weniger Ausgabe (bis zu -qq für gar keine).
- **--no-install-recommends** – Installiert nicht automatisch als "empfohlen" markierte Pakete. Standard apt installiert Recommends mit.
- **--allow-remove-essential, --allow-change-held-packages** – Erlaubt riskante Änderungen (Standard verhindert z.B. Entfernen essentieller Pakete).
- **-s, --simulate** – Simulation: Zeigt an, was würde passieren, aber führt keine Änderung aus (auch Exitcode 0 immer). Gut zum Test.
- **-V, --verbose-versions** – Zeigt bei Auflistungen detailliertere Versionsinfos.
- **--help** – Hilfe-Seite, **--version** – Version von apt.

Beispiele:

- apt update && apt upgrade – Aktualisiert Paketlisten und installiert alle verfügbaren Updates (ohne Dist-Upgrade).
- apt install build-essential – Installiert eine Paketgruppe (hier C/C++ Compiler Toolchain).
- apt remove --purge openssh-server – Entfernt SSH-Server komplett mit Config.
- apt search docker – Sucht alle Pakete mit "docker" im Namen/Beschreibung.
- apt full-upgrade – Aktualisiert System auch über Distribution-Upgrades hinweg (entspricht auf Ubuntu einem Release-Upgrade, sofern Repos angepasst).

(Hinweis: *apt-get* und *apt-cache* sind Vorgänger: z.B. *apt-get install*, *apt-get remove*, *apt-get dist-upgrade*, *apt-cache search*, *apt-cache show*. *apt* vereinfacht den Alltag, aber Skripte nutzen weiterhin oft *apt-get -y etc. dpkg* siehe unten ist noch eine Stufe tiefer für einzelne .deb ohne Abhängigkeitsauflösung.)

dpkg (Debian Package Manager - Low-Level)

Beschreibung: dpkg ist das unterliegende Werkzeug auf Debian/Ubuntu, das .deb-Pakete entpackt, installiert, konfiguriert und entfernt. Es löst keine Abhängigkeiten – dafür sind apt/apt-get zuständig. Man verwendet dpkg meist für manuelle Offline-Installation von .deb-Dateien.

Syntax:

```
dpkg -i <Paket.deb>      # Installieren  
dpkg -r <Paketname>      # Entfernen  
dpkg -P <Paketname>      # Entfernen + Purge  
dpkg -L <Paketname>      # Liste Dateien des installierten Pakets  
dpkg -s <Paketname>      # Statusinfo (ähnlich apt show)
```

Wichtige Optionen/Parameter:

- **-i, --install Datei.deb:** Installiert das angegebene Deb-Paket. Gibt Fehler aus, falls Abhängigkeiten fehlen. (Diese müssen vorher mit apt installiert werden).
- **-r, --remove Paketname:** Entfernt ein installiertes Paket (Paketname ohne ".deb", wie in dpkg -l gelistet). Konfigdateien unter /etc bleiben bestehen.
- **-P, --purge Paketname:** Entfernt Paket inkl. seiner Konfigurationsdateie.
- **-S, --search Dateiname:** Sucht, welches installierte Paket eine bestimmte Datei besitzt (ähnlich rpm -qf). Z.B. dpkg -S /usr/bin/foo.

- **-L, --listfiles Paketname:** Listet alle Dateien, die das installierte Paket auf dem System installiert hat [file-5y1edouysrow4zj6phjcqm] .
- **-l, --list Muster:** Listet Pakete, optional gefiltert nach Muster. dpkg -l ohne Filter zeigt eine Tabelle aller Pakete und deren Status (ii = installiert, rc = removed config-files left, etc.).
- **-s, --status Paketname:** Zeigt den Status und Informationen eines installierten Pakets (Version, Beschreibung, Maintainer, Abhängigkeiten...).
- **--configure Paketname:** Führt die Konfigurationsschritte (dpkg-configure) für ein installiertes aber noch nicht konfiguriertes Paket aus. (Wird z.B. nach entpacken aufgerufen).
- **-B, --auto-deconfigure:** Beim Entfernen eines Pakets, das Abhängigkeiten bereitstellt, markiert abhängige Pakete als "deconfigured".
- ****--force-**X:** Viele Force-Flags, um dpkg anzuweisen, bestimmte Fehler zu ignorieren (z.B. --force-dependency ignoriert fehlende Abhängigkeiten, --force-confold/--force-confnew bei Config-Datei-Konflikten etc.). Vorsicht: Kann System inkonsistent machen.
- **--print-architecture:** Zeigt die Architektur, für die dpkg Pakete installiert (z.B. amd64).
- **--add-architecture <arch>:** Fügt Multiarch-Unterstützung für andere Arch hinzu (z.B. i386 auf einem amd64 System).

Beispiele:

- dpkg -i google-chrome.deb – Installiert Chrome .deb-Paket (scheitert, wenn Abhängigkeiten fehlen – in dem Fall via apt --fix-broken install auflösbar).
- dpkg -r postfix – Entfernt postfix, belässt aber ggf. geänderte /etc/postfix Dateien.
- dpkg -P postfix – Entfernt postfix vollständig inklusive Config.
- dpkg -L bash – Listet alle vom bash-Paket installierten Dateien.
- dpkg -S /bin/ls – Welches Paket stellt /bin/ls bereit? (Antwort: coreutils).

wget (Web GET Utility)

Beschreibung: Ein Kommandozeilen-Downloader für Dateien über HTTP, HTTPS und FTP . wget kann in Skripten genutzt werden, um Dateien aus dem Web herunterzuladen, unterstützt Fortsetzen abgebrochener Downloads, rekursives Herunterladen von Websites etc. Es funktioniert nicht-interaktiv (kein Browser, nur Download).

Syntax:

wget [Optionen] <URL>

Wichtige Optionen:

- **-O Datei** – Speichert die heruntergeladene URL in die angegebene *Datei* (statt den Namen aus URL zu verwenden) . Z.B. wget -O ubuntu.iso http://.../ubuntu.iso.
- **-c, --continue** – Setzt einen abgebrochenen Download fort (sofern Server Range Requests unterstützt). Prüft, wieviel schon vorhanden ist und lädt den Rest von [Resionos.de](#).
- **-q** – Quiet-Modus (keine Ausgabe). **-nv** (not verbose) eine dazwischenliegende Stufe. **-v** für ausführlicher (Standard).
- **-r, --recursive** – Rekursiver Download einer Website. Erfordert oft zusammen mit -np (no parent), -L (stay on same domain), -k (Links anpassen für Offline). Z.B. wget -r -np -k http://example.com/docs/.
- **-np, --no-parent** – Geht beim rekursiven Download nicht über das Startverzeichnis hinaus.
- **-l <Tiefe>** – Legt die Rekursionstiefe fest (Default 5, -l0 = unendlich).
- **-p, --page-requisites** – Lädt alles mit, was nötig ist, um HTML-Seite offline darzustellen (Bilder, CSS...).
- **-N, --timestamping** – Lädt eine Datei nur herunter, wenn sie neuer ist als die lokale (verwendet Last-Modified). Gut zum Aktualisieren.
- **--limit-rate=<Rate>** – Begrenzt Downloadrate, z.B. --limit-rate=200k (200 KB/s). Hilfreich, um Bandbreite zu drosseln.
- **-U, --user-agent=<Agent>** – Setzt einen bestimmten User-Agent-String. Standard ist "Wget/Version". Kann geändert werden, um z.B. nicht blockiert zu werden.
- **--no-check-certificate** – Ignoriert SSL-Zertifikatfehler (z.B. selbstsigniert).
- **--user=<user> --password=<pw>** – Falls erforderlich, für HTTP-Auth oder FTP-Login. (Achtung: Sichtbar in Prozessliste).
- **-i <Datei>** – Liest eine Liste von URLs aus <Datei> (jede Zeile eine URL) und lädt alle herunter.
- **-b, --background** – Startet im Hintergrund (nur auf Unix, leitet Ausgabe nach wget-log).
- **-P <Verz>** – Speichert Dateien in angegebenem Verzeichnis (statt aktuelles).

Beispiele:

- wget http://example.com/file.zip – Lädt *file.zip* in aktuelles Verzeichnis herunter.
- wget -c http://largefile – setzt Download fort, falls vorhanden.

- wget -r -np -N http://example.com/photos/ – Lädt Verzeichnis "photos" vollständig herunter, aktualisiert nur neue Dateien beim Wiederaufruf.
- wget -O - https://api.example.com/data | jq . – Lädt Daten von API und pipet direkt in jq zur Verarbeitung (hier -O - bedeutet Ausgabe auf stdout statt Datei).

curl (Client URL)

Beschreibung: curl ist ein umfangreiches Kommandozeilen-Tool für Datenübertragungen mit URL-basierten Protokollen (HTTP, HTTPS, FTP, etc.). Im Gegensatz zu wget liegt Fokus auf Flexibilität bei HTTP-Requests (Header setzen, verschiedene HTTP-Methoden, Formulare absenden, APIs ansprechen etc.). Es eignet sich für Web-APIs (GET/POST) und kann ebenfalls Dateien herunterladen oder hochladen.

Syntax:

curl [Optionen] <URL>

Wichtige Optionen:

- **-o Datei** – Speichert die Ausgabe in *Datei*. (Kleinbuchstabe o, analog wget -O).
- **-O** – Speichert unter dem **Originalnamen** aus der URL (Großes O, analog wget ohne -O). Achtung: curl -O http://server/dir/file.txt legt *file.txt* an.
- **-L – Location folgen:** folgt HTTP-Redirects (Status 3xx) automatisch freedesktop.org . (wget macht das default, curl nicht ohne -L).
- **-C --** Setzt abgebrochenen Download fort (wenn Server unterstützt; '-' heißt automatischen Offset ermitteln).
- **-#** – Fortschrittsleiste im CLI (oder --progress-bar). Standard curl zeigt laufende Stats Zeile, -# macht wie wget progressive Bar.
- **-s** – Silent (kein Progress oder Fehlerausgaben). **-S** kombiniert mit -s zeigt Fehler dennoch an.
- **-v / -vv** – Verbose Mode (zeigt detaillierte Request/Response, Header etc. -vv noch mehr, inkl. Low-level). Gut zum Debuggen.
- **-I** – Führt eine HEAD-Anfrage aus (nur Header vom Server holen, keine Body-Daten). Nützlich um Meta-Informationen (Content-Length, Last-Modified) abzufragen.
- **-X METHOD** – Explizite HTTP-Methode festlegen (z.B. PUT, DELETE). Standard bei Datenvorhandensein oder -d ist POST, sonst GET.
- **-G** – Erzwingt, dass bei Verwendung von -d (Daten) diese als Query-Parameter an URL angehängt werden (statt POST Body).

- **-d Daten** – Sendet die angegebenen *Daten* im HTTP-Request-Body (Standard via POST). Z.B. curl -d "user=test&pass=123" http://.../login. Implizit -X POST wenn nicht -G.
 - **--data-urlencode Daten**: URL-encodet die Daten vor Versand (z.B. wenn Daten Sonderzeichen enthalten).
 - **--data-raw**: ähnlich -d, aber ohne spezielle Form-Codierung (sendet genau so).
- **-H "Header: Wert"** – Fügt einen HTTP-Header hinzu oder überschreibt ihn. Z.B. -H "Authorization: Bearer <token>". Mehrfach nutzbar für mehrere Header.
- **-u Benutzer:Pass** – Nutzt HTTP Basic Auth mit gegebenen Credentials (wird in Header "Authorization: Basic ...Base64" codiertfreedesktop.org). Vorsicht in Shell-History. Kann auch nur -u user (dann fragt curl interaktiv Passwort ab). Funktioniert auch für FTP.
- **-F Feld@Datei** – HTTP Datei-Upload via *multipart/form-data*. Z.B. curl -F "file=@bild.png" -F "desc=Urlaubsfoto" http://.../upload.
- **-K Konigdatei** – Liest curl-Parameter aus Datei (kann nützlich sein für komplexe Setups).
- **-b CookieDatei** – Sendet Cookies aus angegebener Datei (. Netscape cookie format oder einfach "name=value" Paare). Kann auch -b "name=value" um Cookies zu setzen.
- **-c CookieAusgabe** – Speichert Cookies, die vom Server kommen, in angegebener Datei (um sie später wieder zu verwenden).
- **--limit-rate <Speed>** – Begrenzt Übertragungsrate (z.B. --limit-rate 100k).
- **--retry N** – Anzahl der automatischen Wiederholungsversuche bei Fehler (z.B. Timeout) oder --retry-all-errors.
- **--cacert Datei / --insecure**: Spezielles SSL: anderes CA-Zertifikat nutzen bzw. bei --insecure SSL-Zertifikat überprüfen *ignorieren* (wie wget --no-check-certificate).

Beispiele:

- curl -L -o latest.zip https://github.com/user/proj/releases/latest – Lade neueste Release, folge Redirects, speichere als latest.zip.
- curl -u admin:secret -X DELETE http://server/api/item/123 – Sende DELETE-Request mit Basic Auth.
- curl -H "Accept: application/json" 'https://api.example.com/data?id=5' – Hole JSON-Daten von API mit angepasstem Accept-Header.

- curl -d '{"user":"john"}' -H "Content-Type: application/json" -X POST https://api.example.com/users – Sende JSON-Body per POST (hier ein neuer User).
- curl -F 'file=@report.pdf' -F 'desc=Monthly Report' https://upload.example.com – HTTP File-Upload (Multipart).
- curl -I https://example.com – Nur Header einer HTTP-Response anzeigen (z.B. um Last-Modified zu prüfen).

Speicher- und Dateisystem-Verwaltung

lsblk (List Block Devices)

Beschreibung: Zeigt eine Übersicht aller erkannten Blockgeräte (Festplatten, SSDs, Partitionen, LVM-Volumes etc.) in Form einer Baumstruktur
`file-4fdk7rvx39azjretinarqb】 .`
 lsblk listet wichtige Attribute wie Gerätename, Größe, Typ (disk/part), Mountpoint usw. Es hilft beim Identifizieren von Device-Namen und Partitionierungsstruktur.

Syntax:

lsblk [Optionen] [Gerät...]

Optionen:

- **-a, --all:** Zeigt auch leere bzw. unverwendete Geräte an (z.B. Geräte ohne Speicher, Loop ohne Backing). Standard lässt einige unwichtige aus.
- **-d, --nodeps:** Zeigt nur "Top-Level" Geräte (Disks) ohne Partitionen/Slaves.
- **-e <LIST>, --exclude <LIST>:** Schließt bestimmte Gerätetypen durch Major-Nummern oder Gerätekategorien aus (z.B. -e 7,11 um Loop und DVD zu ignorieren).
- **-f, --fs:** Zeigt zusätzlich Dateisystem-Informationen (Spalten FSTYPE, LABEL, UUID) . Ähnlich blkid, aber in lsblk integriert.
- **-p, --paths:** Zeigt vollständige /dev/Pfadnamen an statt nur Gerätamen (z.B. "/dev/sda" statt "sda").
- **-l, --list:** Ausgabe im Listenformat (eine Eintrag pro Zeile, ohne Baum-Struktur).
- **-J, --json:** Ausgabe als JSON-Struktur (maschinenlesbar).
- **-O, --output-all:** Zeigt alle verfügbaren Spalten an. Standardanzeige ist begrenzt. Mit -O sieht man z.B. auch MAJ:MIN, RM (removable), RO (read-only), MODEL, SERIAL etc.
- **-n, --noheadings:** Unterdrückt Überschrift.
- **-b, --bytes:** Größe in Bytes anzeigen (statt lesbar in KiB/MiB etc.).

- **-r, --raw:** Rohformat ohne Ausrichtung/Padding (für Skripting).

Beispielausgabe:

```
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda       8:0   0 100G 0 disk
        ↘-sda1    8:1   0 512M 0 part /boot
        ↘-sda2    8:2   0 99.5G 0 part
            ↘-vg-root 253:0  0 90G 0 lvm /
            ↘-vg-swap 253:1  0 8G 0 lvm [SWAP]
sr0       11:0  1 1024M 0 rom (cdrom)
```

Hier sieht man Disk *sda* mit 100G und zwei Partitionen *sda1*, *sda2*. *sda2* ist ein PV (LVM Physical Volume) mit Volume Group vg -> daraus LVs *vg-root* und *vg-swap*, gemountet entsprechend.

lsscsi (List SCSI Devices)

Beschreibung: Listet erkannte SCSI-Geräte (inkl. SATA, SAS, USB Storage – diese werden SCSI-Subsys emuliert) auf. **lsscsi** zeigt je Gerät Host/Channel/Target/LUN, dann Gerätetyp, und eventuell Model und Seriennummer je nach Infile-
4fdk7rvx39azjretinarqb】 . Hilfreich, um z.B. alle angeschlossenen Festplatten und optischen Laufwerke anzuzeigen.

Syntax:

lsscsi [Optionen]

Optionen:

- **-v, --verbose:** Ausführlich – zeigt zusätzliche Infos (z.B. SCSI-Transport oder Zustände).
- **-t, --transport:** Zeigt SCSI-Transportinformationen (z.B. SATA, USB UAS, iSCSI etc.).
- **-L, --lunhex:** Zeigt die LUN in hexadezimaler Notation statt dezimal.
- **-H, --hosts:** Zeigt nur SCSI-Host-Adapter (Controller) an.
- **-c, --classic:** Benutzt "klassisches" Ausgabeformat (alter Stil, ohne Abkürzungen).
- **-g, --generic:** Fügt die entsprechenden sg-Device-Nodes hinzu (SCSI Generic /dev/sgX).

- **-s, --size:** Zeigt Größe (Kapazität) der Block-Geräte an, falls anwendbar (nur für Disk-type).
- **-p, --protection:** Zeigt Protection-Information (DIF/DIX).
- **-i, --scsi_id:** Zeigt die SCSI-IDs im klassischen Sinn (T10 IDs).
- **-B, --usb:** Zeigt Bus-IDs (für USB-Geräte) mit an.

Beispiel:

[0:0:0:0] disk ATA Samsung SSD 860 1B6Q /dev/sda 500GB

[2:0:0:0] cd/dvd TSSTcorp CDDVDW SH-224DB SB00 /dev/sr0

[3:0:0:0] disk WD MyBook 25DA 4004 /dev/sdb 4TB

Format: [H:C:T:L] type vendor model rev /dev/sgX -> /dev/sdX. Hier sieht man SCSI-Gerät auf Host 0 (SATA), Host 2 (DVD), Host 3 (USB-HDD).

fdisk (Partition Table Editor)

Beschreibung: Interaktives Konsolen-Tool zum Anzeigen und Bearbeiten von Partitionstabellen auf Blockgeräten (MBR, GPT). Meistens wird fdisk verwendet, um schnell die Partitionierung eines Laufwerks anzusehen (fdisk -l) oder bei MBR-Partitionierung Partitionen einzurichten. Für GPT kann man gdisk oder parted benutzen (fdisk unterstützt aber inzwischen auch GPT).

Syntax:

fdisk [Optionen] <Gerät>

Ohne Optionen startet es ein interaktives Menü für das angegebene Gerät (z.B. /dev/sda).

Optionen:

- **-l, --list:** Listet die Partitionstabellen aller verfügbaren Laufwerke auf [4fdk7rvx39azjretinarqb]. Oder wenn ein Gerät angegeben, nur diese. Z.B. fdisk -l /dev/sda zeigt Partitionen auf sda.
- **-s Partition:** Gibt die Größe der angegebenen Partition in Sektoren aus (veraltet, blockdev --getsz moderner).
- **-u (veraltet):** Wechselt die Ausgabeeinheit (Sektoren vs Zylinder) in alter fdisk. Heutige fdisk (util-linux) nutzt standard 512-Byte Sektoren in -l.
- **-B, -C, -H, -S:** Kann man Geometrie (Zylinder, Heads, Sectors) angeben (MBR Legacy, heute irrelevant).
- **-t Typ:** Zeigt Partitionen in bestimmtem Format (Dos, GPT, ...).

Interaktiver Modus: Kommandos wie m (Hilfe), p (Print Partitionstabelle), n (neue Partition), d (löschen), t (Typ ändern, z.B. 83 Linux, 82 Swap, in GPT GUIDs), w (write und exit), q (quit ohne save).

Beispiel (List):

Disk /dev/sda: 238.5 GiB, ...

| Device | Boot | Start | End | Sectors | Size | Id | Type |
|-----------|------|-----------|-----------|-----------|--------|----|----------------------|
| /dev/sda1 | * | 2048 | 534527 | 532480 | 260M | 7 | HPFS/NTFS/exFAT |
| /dev/sda2 | | 534528 | 409600000 | 409065473 | 195.1G | 83 | Linux |
| /dev/sda3 | | 409600001 | 419430400 | 9830400 | 4.7G | 82 | Linux swap / Solaris |

Zeigt z.B. 3 Partitionen (sda1 NTFS, sda2 Linux, sda3 Swap).

parted (Partitionierungstool)

Beschreibung: Moderner Partitionierer, unterstützt GPT und MBR sowie Größe > 2TB. Kann sowohl interaktiv (im eigenen Prompt) als auch direkt per Kommandozeile genutzt werden. Im Gegensatz zu fdisk beherrscht parted auch **Resizing** von Partitionen und Filesystems (begrenzte FS-Typ Unterstützung) und kann in Skripten genutzt werden.

Syntax (non-interactive):

parted [-s] <Gerät> <Befehl> [Parameter...]

Mit -s (script) ohne interaktive Ausgabe.

Häufige parted Befehle: (Nicht alle Optionen, nur exemplarisch)

- **print** – Zeigt Partitionstabelle des Geräts. Z.B. parted /dev/sdb print.
- **mklabel <Typ>** – Erstellt neue Partitionstabelle (Typ: gpt, msdos, etc.). Achtung: löscht alle bestehenden Partitionen.
- **unit <Einheit>** – Setzt Maßeinheit (MB, GB, %, etc.) für folgende Befehle.
- **mkpart Name Typ Anfang Ende** – Erstellt Partition. Z.B.: mkpart primary ext4 1MiB 10000MiB erstellt eine ~10GB Partition vom 1MB Offset. *Typ* hier "primary"/"logical" (bei msdos) oder nur Partitions-TypName (bei GPT optional ein FS-Typ Label ext4 etc.).
- **rm <Nr>** – Löscht Partition mit Nummer.
- **resizepart <Nr> <Ende>** – Passt Größe der Partition Nr an (End-Sektor). Muss danach FS separat resized werden (z.B. mit resize2fs).
- **name <Nr> <Name>** – Setzt den Partitionsnamen (GPT Label).
- **toggle <Nr> boot** – Boot-Flag toggeln (bei MBR Partitionen).

- **align-check** *Typ Nr* – Überprüft Ausrichtung von Partition *Nr* (optisch/ Zylinder).

Parted kann auch einfache FS-Aufgaben: mkfs integration, aber meist nutzt man separate Tools (mkfs.*).

Beispiele:

- `parted /dev/sdc --script mklabel gpt mkpart primary ext4 0% 50% mkpart primary linux-swap 50% 100%` – Initialisiert /dev/sdc als GPT, teilt in zwei: erste Partition bis Mitte ext4, zweite Swap ab Mitte bis Ende.
- Im interaktiven parted (ohne -s) bekommt man (parted) Prompt zum Eingeben der obigen Befehle nacheinander.

mkfs (Make FileSystem)

Beschreibung: Erzeugt ein neues Dateisystem auf einer Partition oder Festplatte. mkfs ist ein Frontend, das je nach -t <FSTYP> das entsprechende Tool (mkfs.ext4, mkfs.xfs, etc.) aufruft [file-4fdk7rvx39azjretinarqb]. Häufig ruft man direkt das spezifische Formatprogramm auf.

Syntax:

`mkfs -t <FSTYP> [Optionen] <Geräte/Partition>`

Beispiele: `mkfs -t ext4 /dev/sdb1`, `mkfs -t vfat /dev/sdb1`.

Wichtige Optionen (allgemein):

- **-t <typ>** – Dateisystemtyp (ext4, ext3, ext2, xfs, vfat, ntfs, etc.).
- **-V** – zeigt ausführliche Versions-/Debug-Information (welches Tool aufgerufen wird).
- **-q** – quiet (weniger Ausgabe).

Jedes Dateisystem hat eigene spezifische Optionen, wenn man direkt mkfs.ext4 etc. nutzt:

ext4 (mkfs.ext4): -L <Label>, -U <UUID> (benutzerdefiniert), -m <Proz> (Reserve-Blöcke Prozent, default 5%), -O <Feature> (ein/aus bestimmter FS-Features), -E stride= (RAID optimierungen), -b blocksize, etc.

vfat (mkfs.vfat): -n <Label>, -F <32|16> (FAT type), -I (Force if not partition), etc.

xfs (mkfs.xfs): -L <Label>, -m crc=1 (z.B. CRC enable, default ON neu), etc.

swap (mkswap): -L <Label>, (swap selbst ist via mkswap erstellt).

Wichtig: Partition sollte entsprechend markiert/Dimension formatiert sein (z.B. parted type=linux-swap für swap etc.).

Beispiele:

- `mkfs.ext4 /dev/sda3` – Formatiert sda3 mit ext4 Standardparametern.
- `mkfs.vfat -F 32 -n "MYUSB" /dev/sdb1` – Formatiert Partition sdb1 als FAT32 mit Label "MYUSB".
- `mkswap /dev/sda5` – Initialisiert Swap auf sda5 (später via swapon).
- `mkfs.xfs -f -L data /dev/sdc1` – Erzwingt Format auf sdc1 mit XFS, Label "data" (XFS verlangt -f wenn Partition schon ein FS enthält).

mount (Dateisystem einhängen)

Beschreibung: Bindet ein Dateisystem (Partition, Datenträger oder virtuelles FS) in den Verzeichnisbaum ein. mount ohne Argumente listet aktuelle Einhängungen. `mount <Gerät> <Verzeichnis>` hängt ein Gerät an den angegebenen Ordner ein (der Ordner muss existieren und leer sein, typischerweise unter /mnt oder /media oder definierter Mountpoint in /etc/fstabfile-4fdk7rvx39azjretinarqb】 .

Syntax:

`mount [Optionen] <Quelle> <Zielverzeichnis>`

Wichtige Optionen:

- **-t <fstype>** – gibt den Dateisystemtyp an (ext4, vfat, nfs, cifs, etc.). Kann oft weggelassen werden, da mount es automatisch erkennt (über blkid).
- **-o <Optionen>** – übergibt eine durch Komma getrennte Liste von Mount- Optionen-4fdk7rvx39azjretinarqb】 :
 - **defaults** (rw,suid,dev,exec,auto,nouser,async),
 - **ro/rw** (read-only / read-write),
 - **noexec** (keine Ausführung von Programmen erlauben), **nosuid** (SUID/SGID-Bits ignorieren), **nodev** (keine Gerätedateien interpretieren),
 - **sync** (synchrone IO), **async** (Standard, asynchron gepuffert),
 - **user/nouser** (Allow normal user to mount / nur root),
 - **uid=, gid=** (für FAT/NTFS, setze Besitzer der gemounteten Dateien),
 - **umask=, dmask=, fmask=** (Rechte-Masken für FAT/NTFS),
 - **iocharset=, codepage=** (für vfat),
 - **loop** (Loop-Device mount einer Datei als FS, evtl. -o loop oder auto),
 - **remount** (ein bereits gemountetes FS mit neuen Optionen versehen, z.B. `mount -o remount,rw /mnt/point`),

- FS-spezifisch: z.B. **acl**, **user_xattr** (Ext4 Features), **compression**, **encrypt** (btrfs), etc.
- **-L <Label>, -U <UUID>** – anstatt Device-Node, per Label oder UUID identifizieren. `mount -U 1234-ABCD /mnt/usb`.
- **--bind** – führt einen Bind-Mount aus: mountet ein vorhandenes Verzeichnis an eine zweite Stelle im Verzeichnisbaum. Syntax: `mount --bind /pfad/quelle /pfad/ziel`. (Nützlich, um z.B. Verzeichnisse in Chroots sichtbar zu machen). Option **--rbind** rekursiv (inkl. Unterpunkte).
- **-a** – Mountet alles was in /etc/fstab als "auto" markiert ist und nicht noauto. (Ausführung typ. beim Boot).
- **-n** – führt Mount ohne Eintrag in /etc/mtab aus (mtab ist heute oft Symlink zu /proc/self/mounts). Selten benötigt.
- **-v** – verbose (gibt Meldungen aus).

Beispiele:

- `mount /dev/sdb1 /media/usb -o uid=1000,gid=1000,umask=022` – hängt einen FAT32-Stick so ein, dass Benutzer mit UID/GID 1000 Besitzer ist (typisch eigener User), und Rechte 755 auf Dateien/Ordner (umask 022).
- `mount -t nfs 192.168.1.10:/export/share /mnt/share` – hängt eine NFS-Freigabe ein (NFS FS muss installiert sein).
- `mount -o loop diskimage.iso /mnt/iso` – mountet ein ISO-Image als Loop-Device unter /mnt/iso.
- `mount | grep sda` – sieht, ob /dev/sdax gemountet ist (ohne Parameter listet mount alles).

umount (Unmount FileSystem)

Beschreibung: Hängt ein gemountetes Dateisystem wieder aus. Schreibvorgänge werden abgeschlossen und der Puffer geleert. Syntax kann entweder der Mountpoint oder das Device angegeben werden】 .

Syntax:

`umount [Optionen] <Mountpunkt | Gerät>`

Wichtige Optionen:

- **-f, --force**: Erzwingt das Aushängen eines FS. Sollte nur im Notfall (z.B. NFS-Server down) benutzt werden, da es evtl. Datenverlust verursachen kann.
- **-l, --lazy**: "Lazy unmount". Markiert das FS als ausgeworfen, aber falls noch offene Handles bestehen, wird es erst tatsächlich ausgehängt, wenn diese

geschlossen sind. (Hängt unmittelbar virtuell aus, lässt Rest im Hintergrund erledigen). Nützlich bei hängenden Prozessen.

- **-R, --recursive:** Hängt auch alle Unter-Mounts mit ab, die auf dem angegebenen Pfad liegen (rekursiv).
- **-v:** verbose.

Beispiele:

- `umount /media/usb` – hängt das unter `/media/usb` eingehängte Gerät aus.
- `umount /dev/sdb1` – selbes, angesteuert über Gerät.
- `umount -f -a -t nfs` – erzwingt das Aushängen aller NFS-Mounts (z.B. bei Netzproblemen).

(*Tipp: Wenn umount meldet "device is busy", kann man mit lsof oder fuser herausfinden, welcher Prozess noch auf das FS zugreift.*)

df (Disk Free)

Beschreibung: Zeigt Übersicht des belegten und freien Speicherplatzes auf eingehängten Dateisystemen [4fdk7rvx39azjretinarqb]. df listet pro gemountetem FS die Gesamtgröße, verwendeten und freien Bytes sowie den Einhängepunkt.

Syntax:

`df [Optionen] [Dateipfad...]`

Ohne Argumente zeigt alle gemounteten FS. Mit Pfad ermittelt df, auf welchem FS dieser Pfad liegt, und zeigt nur dessen Werte.

Optionen:

- **-h, --human-readable:** Formatiert Größen in lesbarer Form (KiB, MiB, GiB) [4fdk7rvx39azjretinarqb]. Sehr gebräuchlich (Standard rohes Blockzahlen unhandlich).
- **-a, --all:** Schließt pseudo- oder 0-Size-Dateisysteme ein (wie proc, sysfs, etc.), die sonst ausgelassen werden.
- **-T, --print-type:** Zeigt Spalte mit Dateisystemtyp (ext4, tmpfs, nfs, etc.).
- **-i, --inodes:** Zeigt Inode-Auslastung statt Blockgrößen (frei/belegt Inodes pro FS). Nützlich, wenn "no space" durch Inode-Erschöpfung.
- **-t <typ>, -x <typ>:** Nur bestimmte FS-Typen anzeigen bzw. ausschließen. Z.B. `df -x tmpfs -x devtmpfs` um echte Disks anzuzeigen.
- **-B <Größe>:** Ausgabe in Blöcken von angegebener Größe. Z.B. `-B MB` (MByte) oder `-B 1M`.

- **--output[=FELDER]**: Benutzerdefinierte Spaltenausgabe. Z.B. df --output=source,stype,size,used,avail,pcent,target.

Beispielausgabe (df -h):

```
Filesystem  Size Used Avail Use% Mounted on
/dev/sda2    50G  30G  18G  63% /
udev        3.9G   0  3.9G  0% /dev
tmpfs       798M 1.2M 797M  1% /run
/dev/sda1    300M  50M 250M 17% /boot
```

du (Disk Usage)

Beschreibung: Summiert und zeigt die Größe von Verzeichnissen und Dateien an. du kann rekursiv durch Verzeichnisse gehen und den Platzverbrauch berechnen. Standardmäßig zeigt es für jedes Unterverzeichnis den Wert. Oft nutzt man du -sh <dir> um eine schnelle Gesamtgröße zu bekommen.

Syntax:

du [Optionen] [Pfad...]

Ohne Pfad = aktuelles Verzeichnis.

Optionen:

- **-h, --human-readable**: Größen in menschenlesbare Einheiten (K,M,G) ausgeben.
- **-s, --summarize**: Nur Summen pro angegebenem Argument ausgeben, *nicht* rekursiv für Unterordner. (Gibt also Gesamtgröße des Pfades).
- **-A, --apparent-size**: Rechnet mit logischen Dateigrößen statt belegtem Blockplatz. (Z.B. sehr sparsame Sparse Files würden sonst weniger anzeigen; --apparent-size zeigt nominelle Größe).
- **-d N, --max-depth=N**: Rekursionstiefe begrenzen. Z.B. du -h --max-depth=1 /var gibt Größe von /var und dessen Unterordner erste Ebene. (--max-depth=0 wäre wie -s).
- **-c, --total**: Am Ende eine Gesamtzeile "total" ausgeben (Summe aller ausgegebenen Einträge). Bei -s von mehreren Pfaden praktisch.
- **-x, --one-file-system**: Geht nicht in andere eingehängte Dateisysteme über. (Beschränkt auf Partition/FS der Startpfade).

- **-L, --dereference:** Folgt symbolischen Links (sonst ignoriert du Symlinks, rechnet sie als 0 Bytes). Vorsicht: Kann zu doppeltem Zählen führen, wenn Links in Kreis oder auf Oberverzeichnis zeigen.
- **--exclude=<PATTERN>:** Schließt Pfade aus, die zum Pattern passen. Z.B. du -sh -exclude="*.git" oder mehrere --exclude.

Beispiele:

- du -sh /home/* – Zeigt Summe der Größen aller Benutzerverzeichnisse unter /home (jedes in human-readable).
- du -a /etc | grep -E "mysql.*\.cnf\$" – Listet die Größe jeder einzelnen Datei unter /etc, filtert jene, deren Name auf mysql*.cnf endet. (Option -a listet auch Dateien und nicht nur Verzeichnisse).
- du -d2 /var – Größe von /var, dessen Subdirs und Sub-Subdirs (Tiefe 2).

e2label (ext2/3/4 Volume Label)

Beschreibung: Zeigt oder ändert das Label (Namen) eines ext2/ext3/ext4 Dateisystems. Das *Label* ist eine im Superblock gespeicherte Zeichenkette, die man z.B. in fstab zum Mounten verwenden kann (LABEL=<name>).

Syntax:

e2label <Gerät> [NeuesLabel]

Ohne zweites Argument zeigt es das aktuelle Label des ext-Volumes an. Mit Label schreibt es dieses in den FS.

Beispiel:

- e2label /dev/sda3 -> Ausgabe z.B. UbuntuRoot.
- e2label /dev/sda3 Archive -> setzt das Label von sda3 auf "Archive".

(Für andere FS gibt es analoge Tools: XFS nutzt xfs_admin -L, FAT/NTFS mlabel oder dosfslabel/ntfslabel etc.)

resize2fs (ext2/3/4 Dateisystem verkleinern/vergrößern)

Beschreibung: Passt die Größe eines ext2/3/4-Dateisystems an. Wird verwendet nach dem Ändern der Partitionsgröße (mit fdisk/parted). resize2fs kann im online-Modus vergrößern (wenn gemountet, ext3/4), fürs verkleinern muss das Volume ausgehängt und ein Filesystem-Check gemacht werden.

Syntax:

resize2fs [Optionen] <Gerät> [NeueGröße]

Neue Größe kann in K, M, G oder als Anzahl 4K-Blöcke angegeben werden. Lässt man sie weg, wird auf Maximum (Part.-Ende) vergrößert.

Optionen:

- **-p** – progress anzeigen (Fortschritt).
- **-f** – zwingen, auch wenn FS als nicht sauber markiert (normal will fsck vorher laufen bei Verkleinern).
- **-F** – flush caches vorher (sorgt für up-to-date Infos; Standard normal).
- **-M – minimal**: verkleinert das FS auf die kleinstmögliche Größe (alle Daten gerade passend). Äquivalent zu resize2fs <dev> -M (dauert Berechnung).

Beispiele:

- Nach Vergrößern der Partition: resize2fs /dev/sda2 – erkennt neue Partitionsgröße und wächst das ext4 FS auf vollen Platz.
- Verkleinern: umount /dev/sda2; e2fsck -f /dev/sda2; resize2fs /dev/sda2 20G – erst aushängen, fsck erzwingen, dann auf 20 GiB verkleinern. Anschließend Partition selbst verkleinern mit parted.

tune2fs (Tune ext2/3/4 Filesystem Parameters)

Beschreibung: Ändert diverse Parameter eines bestehenden ext-Dateisystems: Reserveblöcke, Check-Intervalle, Label, Journal einschalten, usw.

Syntax:

tune2fs [Optionen] <Gerät>

Häufige Optionen:

- **-l** – Listet alle Superblock-Parameter (Filesystem-Info) auf freedesktop.org. Zeigt z.B. Inode-Größe, Features, Letzter Mount, Nächster geplanter Check, etc. (Read-only Operation).
- **-c <MaxMounts>** – Setzt die maximale Anzahl Mount-Vorgänge bis zur erzwungenen fsck-Prüfung. Z.B. -c 0 = deaktiviert mount-count-basierten Check.
- **-i <Interval>** – Setzt das Zeit-Intervall für Zwangs-FSCK. z.B. -i 30d = alle 30 Tage. -i 0 = kein zeitbasierter Check.
- **-r <AnzRes>** – Setzt Anzahl der reservierten Blöcke (für root). Standard 5% bei ext4 auf <1TB. Kann man z.B. auf 1% reduzieren auf großen Volumes, z.B. tune2fs -r 0 /dev/sdb1 um Reserven komplett abzuschalten (nicht empfohlen auf /). Alternativ -m <Prozent> (z.B. -m 1).
- **-L <Label>** – Ändert Label (wie e2label).

- **-U <UUID>** – Ändert die UUID (zufällig oder angegeben). (Vorsicht: fstab etc. müssen angepasst).
- **-J** – Journal-Optionen: z.B. -O has_journal um Journal hinzuzufügen (macht aus ext2 -> ext3), oder -O ^has_journal Journal entfernen (ext3->ext2), oder -j Kurzoption (Standardwerte Journal hinzufügen).
- **-O <FeatListe>** – Aktiviert (feature) oder deaktiviert (^feature) Filesystem-Features. Z.B. -O casefold (für ext4 ab 5.2: Case-Insensitive-Folder Support) oder ^huge_file (Feature abschalten – Achtung, Daten verlieren, nur wenn FS leer). Nur bestimmte Features togglebar, meist erfordert offline und vollständiges Backup.
- ****-E <extendopt>**** – Erweiterte Einstellungen: z.B. stripe=<StripeSize>(RAID stripe in Blöcken),lazy_itable_init=1` (schnell format, Inode-Table lazy Clearing).
- **-M <Verzeichnis>** – Markiert FS als gemountet an <Verzeichnis> (setzt den Mountpoint in Superblock – meist rein informativ).

dumpe2fs (Dump ext2/3/4 Superblock)

Beschreibung: Gibt ausführliche Informationen über ein ext2/3/4-Dateisystem aus – Inhalt des Superblocks und der Block Group Deskriptoren . Zeigt ähnlich tune2fs -l Informationen, plus Aufteilung der Blöcke in Gruppen, freie Blöcke/Inodes pro Gruppe etc. Hilfreich für Diagnose (z.B. Backup-Superblock Positionen).

Syntax:

dumpe2fs [Optionen] <Gerät>

Optionen:

- **-h** – Zeigt nur den Inhalt des Superblocks und der Gruppen-Deskriptoren (ohne die Liste aller freien Blöcke/Inodes). Das ist die gängigste Verwendung, da ohne -h die Ausgabe sehr lang sein kann (listet freie Blöcke/Inodes aller Groupswiki.ubuntuusers.de).
- **-x** – Zeigt auch die detaillierten Gruppen-Informationen (wie freie Blöcke/Inodes-Bitmap). Standard ohne -h tut das sowieso.
- **-b** – Listet die Nummern der freien Blöcke. (Sehr viel Output auf großem FS.)
- **-i** – Listet die freien Inode-Nummern.

Beispiel (gekürzt dumpe2fs -h /dev/sda2):

Filesystem volume name: <none>

Last mounted on: /

Filesystem UUID: 4e8fea21-...-b6c7

Filesystem magic number: 0xEF53

Filesystem revision #: 1 (dynamic)

Filesystem OS type: Linux

Block size: 4096

Block count: 25600000

Reserved block count: 1280000

Free blocks: 15034567

First Block: 0

...

Journal inode: 8

Journal backup: inode blocks

...

Zeigt Label, UUID, Blocksize, Anzahl Blöcke, Reserve, Anzahl freie etc., und welche Features aktiviert sind.

cryptsetup (LUKS Disk Encryption Setup)

Beschreibung: Tool zur Einrichtung von **verschlüsselten Datenträgern** mit LUKS (Linux Unified Key Setup) sowie zum Öffnen/Schließen solcher Verschlüsselungs-Volume】 . cryptsetup kann ein Gerät mit LUKS initialisieren, Passphrase setzen, und im Betrieb ein verschlüsseltes Volume als Device-Mapper-Gerät bereitstellen, das dann gemountet werden kann.

Typische Aktionen (Subcommands):

- **luksFormat** – Initialisiert ein Blockgerät mit LUKS-Verschlüsselungsheader und fordert eine Passphrase zum Einrichten an. Beispiel: cryptsetup luksFormat /dev/sdb1. *Achtung:* Löscht alle vorhandenen Daten auf dem Gerät.
- **open** oder **luksOpen** – Öffnet ein verschlüsseltes Volume (fragt Passphrase). Syntax: cryptsetup luksOpen /dev/sdb1 geheim_vol. Dies erstellt ein Device /dev/mapper/geheim_vol, das nun entschlüsselt agiert und z.B. mit mkfs formatiert oder gemountet werden kan】 .
- **close** – Schließt ein vorher geöffnetes LUKS-Device: cryptsetup close geheim_vol (dann ist /dev/mapper/geheim_vol weg).
- **luksAddKey** – Fügt einen weiteren Schlüssel (Passphrase) hinzu. LUKS unterstützt mehrere Key-Slots. cryptsetup luksAddKey /dev/sdb1

/pfad/zur/schlüsseldatei oder interaktiv eine neue Passphrase eingeben (nachdem man eine gültige alte eingibt).

- **luksRemoveKey** – Entfernt einen Schlüssel (kann durch Angabe der zu entfernenden Passphrase oder Slot-Nummer erfolgen).
- **luksChangeKey** – Ändert eine bestehende Passphrase (in einem Slot) durch eine neue (kombi aus Remove+Add effizienter).
- **luksHeaderBackup / luksHeaderRestore** – Backup bzw. Wiederherstellen des LUKS-Headers (empfohlen, falls Header korrupt -> Daten verloren).
- **status** – Zeigt Informationen über ein offenes Device-Mapper Volume (Name, verschlüsseltes Dev, Cipher, Keylen, etc.): cryptsetup status geheim_vol.

Wichtige Optionen:

- **-y** (im Format-Befehl): fragt neues Passwort zweimal zur Verifizierung ab.
- **-d <Datei>**: verwendet eine Datei als Schlüsseldaten (anstatt interaktiv Passwort). -d kann genutzt werden, um vom stdin zu lesen.
- **--key-size <bits>**: Schlüsselgröße (in Bits) für die Verschlüsselung (z.B. 256).
- **--cipher <name>**: Zu nutzender Verschlüsselungsalgorithmus und Mode. Standard z.B. aes-xts-plain64 bei LUKS2. Kann z.B. aes-cbc-essiv:sha256 oder anderes angegeben werden.
- **--hash <hash>**: Hash-Algorithmus für Passphrase-Derivation (PBKDF). Standard modern ist z.B. Argon2id oder früher sha256.
- **--iter-time <ms>**: Wie lange (Millisek) auf PBKDF verwendet werden soll (Bestimmt Komplexität).
- **--type luks1/luks2**: LUKS Version festlegen (aktuell Standard LUKS2, kann optional LUKS1 für Kompatibilität gewählt werden).
- **-q**: Quiet (unterdrückt Ausgabe, z.B. Warnung bei Format).

Beispiel Workflow:

1. **Formatieren**: cryptsetup -y luksFormat /dev/sda5 (legt LUKS auf Partition sda5 an, fragt 2x Passwort).
2. **Öffnen**: cryptsetup luksOpen /dev/sda5 geheim (Passwort eintippen, erstellt /dev/mapper/geheim).
3. **Format FS**: mkfs.ext4 /dev/mapper/geheim.
4. **Mount**: mount /dev/mapper/geheim /mnt/secure. Jetzt Daten drauf speichern.
5. **Aushängen+Schließen**: umount /mnt/secure && cryptsetup close geheim.

(LUKS ermöglicht bis zu 8 Passphrases; man kann z.B. mit luksAddKey einen zweiten hinzugeben, sodass entweder Passphrase A oder B das Volume öffnet. Das Gerät /dev/sda5 kann in /etc/crypttab und /etc/fstab eingetragen werden für Boot-Time auto-open via keyfile oder prompt.)

shred (Securely wipe file/device)

Beschreibung: Überschreibt eine Datei oder ein ganzes Gerät mehrfach mit zufälligen Daten, um ein Wiederherstellen gelöschter Inhalte zu erschweren. shred ist nützlich zum sicheren Löschen auf magnetischen HDDs. Bei SSDs mit Wear-Leveling ist Effekt begrenzt – dort lieber ATA Secure Erase oder physical destruction.

Syntax:

```
shred [Optionen] <Datei1> [Datei2...]
```

Optionen:

- **-n N** – Anzahl der *zusätzlichen* Überschreib-Pässe (Standard 3). D.h. 3 bedeutet insgesamt 3 mal random + final pass (siehe -z).
- **-z, --zero** – Überschreibt am Ende noch einmal mit Null-Bytes, um das Muster zu verwischen (anstatt dass zuletzt zufälliges Rauschen bleibt). Standard: *aktiv* (bei neueren shred ist -z default an).
- **-f, --force** – Ändert Schreibschutz falls nötig (chmod u+w), um schreiben zu können.
- **-u, --remove[=how]** – Nach dem Überschreiben die Datei löschen (unlink). Optional how kann unlink (Standard) oder wipe oder wipesync sein: *wipe* überschreibt erst mit Null dann löscht, *wipesync* macht zusätzlich Sync. Standard -u bereits tut wipesync.
- **-v, --verbose** – Auskunft über Fortschritt.
- **-x, -X, --exact** – Überschreibt Datei genau Byte-für-Byte anstelle Dateigröße auf nächste Blockgröße aufzurunden (Standard shred nutzt ganze Blocks, kann über Dateiende hinaus schreiben falls Dateigröße nicht mod Blocksize). *Exact* vermeidet das, aber hinterlässt evtl. un-geschredderte Slack am Ende.
- **-s N, --size=N** – Behandelt die Datei als N Byte lang (statt aktuelle Länge). Um auch ungenutzten Platz (z.B. Slack Space/EOF beyond) im Inode zu überschreiben, kann man etwas größer angeben.

Beispiele:

- shred -u file.txt – Überschreibt *file.txt* viermal (3 rand + 1 zero) und löscht sie anschließend.

- shred /dev/sdb – Überschreibt komplett /dev/sdb mit 3 Runden Zufallsdaten (zerstört alle Daten/Partitionen darauf).
- shred -n 0 -z -u secret.dat – Einmal mit Nullen überschreiben und entfernen (hier wollte man evtl. nur einfaches Überschreiben und löschen).

Warnung: Shred wirkt nicht zuverlässig auf: **logischen Volumes** in Files (z.B. Datei in FS kann redundante Kopien (Journal, Snapshots) haben), **RAID, SSD, Netzwerkdateisysteme, Dateisysteme mit Copy-on-Write (btrfs, ZFS)** – dort können Daten an mehreren Orten stehen. In solchen Fällen besser Vollverschlüsselung verwenden oder das gesamte physische Medium überschreiben.

iostat (IO Statistics)

Beschreibung: Zeigt Statistik zur Systemauslastung von CPU und Input/Output für Blockgeräte (Datenträger). iostat gehört zum Paket sysstat. Es hilft zu sehen, wie stark Platten ausgelastet sind (%util) und Durchsatz (MB/s), IOPS etc., sowie generelle CPU-Idle/Wait Zeite .

Syntax:

iostat [Optionen] [Intervall [Durchläufe]]

Ohne Argument gibt einen Durchschnitt seit Boot aus. Mit Intervall (Sekunden) aktualisiert regelmäßig.

Wichtige Optionen:

- **-c** – Zeigt nur CPU-Statistiken.
- **-d** – Zeigt nur Disk (Geräte) Statistiken.
- **-k, -m** – Ausgabe in KiB/s oder MiB/s (anstatt Blöcke/s). (Aktuelle sysstat erlaubt -block-size).
- **-p [Gerät]** – Detaillierte Partitions-Statistiken einbeziehen (normal zeigt iostat nur Whole-Device). Z.B. -p sda zeigt sda und sda1,sda2,...
- **-x** – Ausgabe in erweiterter Form (Pro Gerät: zusätzlich Auslastungsprozentsatz, Average Request Size, Await Times etc.) . Sehr nützlich, da Standard ohne -x recht knapp.
- **-t** – Zeigt Timestamp vor jeder Ausgabe.
- **-y** – Unterdrückt erste Ausgabe (since boot), beginnt erst mit Werten ab erstem Intervall.

Interpretation (bei iostat -x):

Spalten pro Device: rrqm/s wrqm/s (Merge von I/O, veraltet bei modernen Kernel meist 0), r/s w/s (reads/writes per sec), rkB/s wkB/s (Throughput), avgrq-sz (Durchschnitl.

Requestgröße in Sektoren oder KB), avgqu-sz (durchschn. Queue length), await (Durchschnittliche Wartezeit pro IO ms), svctm (durchschn. Servicezeit pro IO ms – oft nicht mehr zuverlässig), %util (Prozentuale Zeit, die das Gerät aktiv war innerhalb der Messperiode – 100% bedeutet permanent beschäftigt).

Beispiel:

```
Device r/s w/s rkB/s wkB/s avgrq-sz avgqu-sz await svctm %util
sda 12.5 30.2 500.0 250.0 40.0 0.5 10.0 2.00 85.0
```

Hier sda ~85% ausgelastet, wartet ~10ms per IO, etwa 42.7 IO/s (r+s) mit ~750kB/s.

ioping (I/O Ping)

Beschreibung: Ermittelt Latenz (und ggf. Durchsatz) von Storage in Echtzeit ähnlich dem Prinzip von ping im Netzwerk. Es führt schnelle sequenzielle oder wahlfreie Lesezugriffe durch und misst deren Zeit. Praktisch um Latenz auf Disks oder bestimmten Verzeichnissen (Dateisystem) festzustellen.

Syntax:

```
ioping [Optionen] <Ziel>
```

Ziel kann ein Pfad (Datei oder Verzeichnis) sein für Filesystem IO, oder z.B. ein Blockgerät (mit -D für Direkten Device Zugriff) oder sogar ein bestimmter I/O-Engine (IO depth).

Wichtige Optionen:

- **-c <count>** – Anzahl der Ping-Requests (Standard unendlich, bis abgebrochen).
- **-i <interval>** – Wartezeit zwischen Pings (Sekunden, default 1s). Kann in ms angeben z.B. -i 0.1.
- **-s <bytes>** – Größe der zu lesenden Daten pro Ping (Standard 4KiB).
- **-R** – Ausgabe im Rohformat (Maschinenlesbar).
- **-D** – Direct-Device Mode: Arbeitet auf Blockgeräte direkt (bypasses FS-Cache).
- **-A** – Ausgabe als Aggregatstatistik nur (kein laufendes Ping, nur Summary).
- **-S** – Sync/Sequenziell: testet sequenzielles Lesen statt wahlfrei. (Standard ioping macht random position reads).
- **-W** – führt Schreib-Tests durch statt Lesen (Achtung kann Daten überschreiben zerstören, am besten auf extra Gerät/Datei).
- **-P <N>** – Parallel Requests (Thread count).
- **-q** – keine laufende Ausgabe, nur Summary am Ende.

Beispiele:

- ioping . – Sendet kontinuierlich 4K-Leseanfragen ins aktuelle Verzeichnis (random in FS), zeigt Latenzen, bis Ctrl+C.

4096 bytes from . (ext4 /dev/sda2): request=1 time=0.47 ms

4096 bytes from . (ext4 /dev/sda2): request=2 time=0.44 ms

... (zeigt typisch ~0.5ms auf SSD, ~10-15ms auf HDD) ...

Nach Abbruch gibt es eine **Summary** mit min/avg/max/mdev (Jitter).

- ioping -c 10 /mnt/backup – Misst 10 Zugriffe im FS /mnt/backup (z.B. externer USB), dann stop.
- ioping -R /dev/sdb – Führt 1 Sekunde lang Messungen auf /dev/sdb durch und gibt am Ende aggregated IOPS und Bandbreite.
- ioping -A -S -c 50 /file.iso – Liest sequenziell 50 Anfragen durch große Datei und gibt am Ende Durchschnittszeit.

(Tool muss ggf. installiert werden, gehört nicht standard zu allen Distr.)

Geräte, Prozesse, Speicher und Kernel

hwinfo (Hardware Info)

Beschreibung: Liefert detaillierte Informationen zur Hardware des Systems. hwinfo probiert, alle verfügbaren Hardware-Komponenten aufzulisten: CPU, RAM, Festplatten, USB, PCI-Geräte, Netzwerk, Monitor, etc., inklusive ihrer Eigenschaften und Treiber]. Oft auf openSUSE installiert, aber auch auf anderen Distros verfügbar.

Syntax:

hwinfo [Optionen] [Hardwareklasse]

Optionen / Klassen:

- Ohne Angabe gibt hwinfo *alles* aus (sehr lang). Besser, man filtert nach Klasse: z.B. hwinfo --cpu, hwinfo --disk, --gfxcard, --netcard, --printer, --usb, --bios, --memory etc.
- **--short:** Kurzfassung ausgeben (ein Eintrag pro Gerät in einer Zeile).
- **--brief:** Noch knapper (ähnlich short).
- **--summary:** Summaries (z.B. nur Zahl der Disks, CPUs etc.).
- **--verbose <level>:** Detaillierungsgrad (0-9). Default 1 (schon sehr ausführlich).
- **--debug N:** Setzt Debug-Level (für Entwickler, welche Module hwinfo benutzt).

- **--hwitem <ID>**: Sucht nach Hardware mit bestimmter ID (z.B. PCI-ID).
- **--save-config <datei> / --load-config <datei>**: Speichert erkannte Hardware in Datei zum späteren Vergleich.

Beispiel: hwinfo --cpu könnte ausgeben:

```
01: None 00.0: 10103 CPU
[Created at cpu.462]
Unique ID: rdCR.jX8Un+o2ZT1
Hardware Class: cpu
Arch: X86-64
Vendor: "GenuineIntel"
Model: 6.158.10 "Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz"
... (viele weitere Details Frequenzen, Cache etc.) ...
```

hwinfo --short summarisiert z.B. Hauptgeräte und Partitionen.

dmidecode (DMI Table Decode)

Beschreibung: Gibt Informationen aus dem BIOS/UEFI-DMI (SMBIOS) heraus, wie Hersteller, Produktnamen, Seriennummern von System/Mainboard, BIOS-Version, verbauter Speicher, Anschlüsse etc]. Hilfreich für Hardware-Inventarisierung (Seriennummer, RAM-Module, OEM strings).

Syntax:

dmidecode [Optionen]

Optionen:

- **-t <string|Nummer>, --type**: Nur bestimmte DMI-Typen ausgeben. Typ kann Name oder Zahl sein. Z.B.:
 - bios (DMI type 0),
 - system (1),
 - baseboard (2),
 - processor (4),
 - memory (Memory Device, 17),
 - cache (7),

- connector (8), slot (9), etc.
Man kann -t memory um nur RAM-Infos zu sehen (Größe, Typ, Hersteller, Seriennummer der DIMMs) .
- **-s <keyword>, --string:** Einzelne SMBIOS-Einträge gezielt ausgeben (z.B. bios-vendor, bios-version, system-manufacturer, system-serial-number, baseboard-asset-tag, processor-frequency etc.).
- **-q:** Quiet (unterdrückt weniger wichtige DMI-Eintrags-Labels wo "Not Provided").
- **-u:** Raw dump (hex) der DMI-Daten.
- **--dump-bin file, --from-dump file:** Dump SMBIOS in Datei bzw. lese aus Dump (z.B. offline Analyse).

Beispiel: dmidecode -t system könnte zeigen:

System Information

Manufacturer: Dell Inc.

Product Name: Latitude 5490

Version: None

Serial Number: ABCD123

UUID: 4c4c4544-0034-5810-8055-b6c04f513332

Wake-up Type: Power Switch

SKU Number: 07E6

Family: Latitude

dmidecode -t memory zeigt pro RAM-Riegel z.B. Größe, Form Factor (SODIMM), Type (DDR4), Speed, Manufacturer, Serial, Part Number.

lspci (List PCI Devices)

Beschreibung: Zeigt alle PCI/PCIe-Geräte im System (angeschlossene Hardware am PCI-Bus: z.B. Netzwerkkarten, Sound, USB-Controller, Grafikkarten, SATA-Controller etc.file-4fdk7rvx39azjretinarqb) . Es listet Geräte-ID und -Namen anhand einer Datenbank (/usr/share/misc/pci.ids).

Syntax:

lspci [Optionen]

Optionen:

- **-v, -vv, -vvv:** Verbose-Ausgabe in ein oder mehreren Stufen (mehr Register-Informationen). -vvv kann sehr ausführlich sein (inkl. Capabilities, Konfigspace).

- **-k:** Zeigt Kernel-Treiber und Module, die für jedes Gerät verwendet werden (welcher Treiber zugeordnet ist file-4fdk7rvx39azjretinarqb]. (Z.B. Kernel driver in use: e1000e für Intel NIC).
- **-t:** Zeigt die hierarchische Baumstruktur der PCI-Bus-Topologie (wer hängt an welchem Bus/Bridge).
- **-nn:** Zeigt neben dem Gerätenamen auch Vendor:Device ID in eckigen Klammern. -n nur numeric IDs ohne Namen. -nn= beide. Z.B. Ethernet Controller [8086:1533] etc.
- **-d [<vendor>:<device>]:** Filtert nach bestimmten PCI IDs. lspci -d 8086: zeigt nur Intel-Geräte, -d :1533 nur Geräte mit Device 1533.
- **-s [[<domain>]:]<bus>:<slot>[.<func>]:** Filtert nach Bus-Adresse (wie in lspci -t oder /sys/bus/pci). Z.B. -s 00:1f.2 zeigt nur dieses Gerät.
- **-m:** Maschinenlesbar (gibt in einer CSV-artigen Formatzeile pro Gerät: Domain Bus Slot, Klass, Vendor, Device, SVendor, SDevice).
- **-xxx:** Dump des gesamten PCI-Konfigurationsspeichers (64 Bytes beyond standard) – Vorsicht sensibel. -xx nur ersten 256 Bytes (std config).
- **-H1:** Nutzen von /proc/bus/pci statt direct (historisch, meist nicht nötig).
- **-Q:** Zeigt das Querier-Modell (Kernel vs direct vs /proc).

Beispiel Ausgabe (gekürzt):

00:00.0 Host bridge: Intel Corporation Device 5914

00:02.0 VGA compatible controller: Intel Corporation UHD Graphics 620

00:14.0 USB controller: Intel Corporation Sunrise Point-LP USB 3.0 xHCI Controller (rev 21)

00:16.0 Communication controller: Intel Corporation Management Engine Interface (rev 21)

...

03:00.0 Network controller: Intel Corporation Dual Band Wireless-AC 8265 (rev 78)

Mit -nn würde man z.B. sehen 00:02.0 VGA controller [0300]: Intel Corporation UHD Graphics 620 [8086:5917]. Mit -k ergänzt: Kernel driver in use: i915.

lsusb (List USB Devices)

Beschreibung: Listet USB-Geräte auf, die am System angeschlossen sind (inkl. interne Hubs etc.). Zeigt Vendor und Produktnamen falls bekannt (Datenbank /usr/share/misc/usb.ids) und ID.

Syntax:

lsusb [Optionen]

Optionen:

- **-v**: Verbose, zeigt detaillierte Descriptor-Infos pro Gerät (ähnlich lsusb -vv noch detaillierter). Das kann sehr lang sein.
- **-t**: Baum-Darstellung der USB-Geräte-Hierarchie (welches Device hängt an welchem Port/Hub) – ähnlich lspci -t.
- **-s <bus>:<dev>**: Zeigt nur spezifisches Gerät (per Bus und Device Nummer). Diese Nummern sieht man in lsusb Standardausgabe vorn.
- **-d vid:pid**: Filtert nach VendorID:ProductID (Hex).
- **-D <device-file>**: Liest Infos aus einer Geräte-Datei (z.B. /dev/bus/usb/001/007) – erfordert Root.
- **-w**: Ausgabe als mit usbdump kompatibler Text (Wire format).
- **-v -v (or -vv)** liefert noch detailliertere Dump (inkl. Hexdump der HID Deskriptoren etc.).

Beispiel Ausgabe:

Bus 002 Device 003: ID 0bda:0316 Realtek Semiconductor Corp. Card Reader

Bus 002 Device 002: ID 8087:0aaa Intel Corp.

Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub

Bus 001 Device 004: ID 0c45:671f Microdia Integrated Webcam

Bus 001 Device 003: ID 8087:07dc Intel Corp. Bluetooth wireless adapter

Bus 001 Device 002: ID 056e:4008 Elecom Co., Ltd

Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

Hier z.B. Card Reader, Webcam, Bluetooth, eine Maus (056e:4008 Elecom), plus Root Hubs.

Mit -t etwa:

/: Bus 02.Port 1: Dev 1, Class=root_hub, ... (bus 2, 3.0)

|__ Port 1: Dev 2, Intel USB 3.0 Hub

|__ Port 2: Dev 3, Realtek Card Reader

...

Zeigt Topologie (Hub etc.).

lscpu (List CPU Info)

Beschreibung: Zeigt Informationen über die CPU(s) des Systems – Anzahl Kerne, Threads, Sockets, Modellname, Frequenz, Virtualisierungsfunktionen uswfile-4fdk7rvx39azjretinarqb】.

Syntax:

`lscpu [Optionen]`

Optionen:

- **-e, --extended [=list]:** Ausgabe als Tabelle mit Spalten (erweiterte Auflistung jeder logischen CPU mit Eigenschaften). Optional kann man angeben, welche Felder (z.B. CPU, Node, Socket, Core, Online, ...).
- **-p, --parse [=list]:** Ähnlich --extended aber kommagetrennt (parseable). Standardfelder: CPU,Core,Socket,Node.
- **-x, --hex:** Zeigt hex-Werte wo sinnvoll (z.B. CPU bitmasken).
- **-c, --offline / --online:** Zeigt nur offline oder nur online CPUs (bei Hotplug-Systemen).
- **-y, --physical:** Zeigt nur physische IDs (keine virtuellen IDs).
- **-J, --json:** JSON-Ausgabe (maschinenlesbar).
- **-B, --byte:** Einheit Bytes statt kB für Cache-Größen.
- **-C, --cache:** Zeigt Details zu CPU-Caches (Anzahl, Größen).

Beispiel (gekürzt lscpu Standard):

Architecture: x86_64

CPU op-mode(s): 32-bit, 64-bit

Sockets: 1

Cores per socket: 4

Threads per core: 2

Model name: Intel(R) Core(TM) i7-8565U @ 1.80GHz

CPU MHz: 2112.004

Virtualization: VT-x

L1d cache: 32K

L2 cache: 256K

L3 cache: 8192K

Flags: fpu vme ... vmx ...

Zeigt also 1 Sockel, 4 Kerne, 8 Threads (4*2), CPU-Modell und Takt etc.

lsmem (List Memory Blocks)

Beschreibung: Zeigt Speicherblock-Informationen des Systems (insbesondere auf Systemen mit Hotpluggable Memory oder NUMA). Es listet Memory-Block-Geräte (wie in /sys/devices/system/memory) auf. Auf x86-PCs weniger relevant; auf IBM System z oder Power etc. nützlich.

Syntax:

lsmem [Optionen]

Optionen:

- **-n, --online / -o, --offline:** Zeigt nur online bzw. offline Speicherblöcke.
- **-b, --bytes:** Größen in Bytes statt lesbar.
- **-M, --mega:** Größen in MiB. (Standard lsmem gibt in bytes + kMG suffix)
- **-p, --pairs:** Ausgabe im Key:Value Format (für maschinelles Parsing).
- **-J, --json:** JSON-Format.
- **--output <list>:** Spalten auswählen (z.B. --output SIZE,NODE,STATE).
- **--help, --version:** analog.

Beispiel lsmem (auf PC ohne Hotplug memory):

| RANGE | SIZE | STATE | REMOVABLE | BLOCK |
|---------------------------------------|------|--------|-----------|-------|
| 0x0000000000000000-0x000000007fffffff | 2G | online | yes | 0-15 |
| 0x0000000100000000-0x000000027fffffff | 6G | online | yes | 32-79 |

Zeigt 2 memory ranges (entsprechend wohl 2 DIMMs). "Removable yes" heißt vom Kernel als Hot-remove möglich (oft 'no' auf Standard-PC). "BLOCK 0-15" referenzieren Memory block device IDs.

Auf NUMA-Systemen würde Spalte NODE relevant sein, um Memory pro NUMA-Node zu sehen.

ps (Process Status)

Beschreibung: Zeigt eine Momentaufnahme der aktuellen Prozesse des Systems. ps kann verschiedene Ausgabeformate haben (BSD-Style, UNIX-style). Häufig nutzt man ps aux oder `ps -effile-4fdk7rvx39azjretinarqb` .

Syntax:

ps [Optionen]

(Komplexität: ps hat UNIX Optionen (mit - und bündelbar), BSD Optionen (ohne - und können zusammen geschrieben werden) und GNU long Optionen. Z.B. ps aux ist BSD Syntax, ps -ef ist UNIX syntax – beide gängig, liefern ähnliches.)

Gängige Aufrufe:

- **ps aux** – Listet alle Prozesse mit Details (BSD format). Spalten: USER, PID, %CPU, %MEM, VSZ, RSS, TT (TTY), STAT, START, TIME, COMMAND.
- **ps -ef** – Listet alle Prozesse (UNIX format). Spalten: UID, PID, PPID, C (CPU usage), STIME (start time), TTY, TIME (CPU time), CMD.
- **ps -eF** – wie -ef, aber "extra full", zeigt threads als separate Linien und vollständige Kommandozeilen etc.
- **ps -ejH** – zeigt Hierarchie (Prozessbaum) mit Prozess-IDs.
- **ps -eo pid,cmd,%mem,%cpu --sort=-%mem** – Zeigt benutzerdefinierte Spalten (hier pid, Befehl, Speicher- und CPU% sortiert nach Speicherverbrauch absteigend).

Wichtige Optionen:

UNIX-style (präfix mit -, können kombiniert werden):

- **-e** – Alle Prozesse anzeigen (entspricht -A).
- **-f full** Format (mehr Spalten inkl. PPID, STIME, in Kombination z.B. -ef).
- **-F** – Extra full (noch mehr Infos: SZ, RSS, PSR CPU, etc.).
- **-h** – Ohne Header.
- **-H** – Hierarchie-Tree anzeigen (Indentation für Parent-Child).
- **-j** – Jobs format (PGID, SID etc. Spalten).
- **-l** – Long format (mehr spalten inkl. Flags, NI nice, LWP etc.).
- **-u <user>** – Prozesse eines bestimmten Benutzers.
- **-p <pidlist>** – Nur bestimmte PIDs.
- **-G <group>, -U <user>** – nach Gruppe/Benutzer filtern.

BSD-style (ohne -, einzeln hintereinander):

- **a** – Zeigt Prozesse von *allen Terminals* (nicht nur eigene).
- **x** – Zeigt auch prozesse *ohne Terminal* (Daemons). **ax** zusammen = alle Prozesse.
- **u** – Benutzerformat (inkl. Owner, CPU%, MEM%). => **aux** verbreitet.
- **j** – Jobs format (session, pgrp).
- **w** – breites Output (alle columns of command, use -ww to not truncate).

GNU long options (mit --):

- **--pid, --ppid, --user** etc. Alternative Filter.
- **--sort=<key>**: Sortiert Ausgabe nach Spalte(n). Z.B. --sort=-%cpu (absteigend CPU).
- **--no-headers**: Ohne Header (statt -h).
- **--forest**: ASCII-Art Bäumchen (ähnlich -H, aber macht "|_" und Einrückungen). Kombiniert mit e.g. -ef.

STAT Spalte (Prozessstatus):

Codes: R (running), S (sleep), D (uninterruptible sleep, i/o), Z (zombie), T (stopped), t (gestoppt im Hintergrund), X (dead). Flags dahinter: < (high prio), N (low prio, nice>0), L (pages locked in mem), s (session leader), + (foreground process group in TTY).

sar (System Activity Reporter)

Beschreibung: Teil von sysstat (wie iostat, mpstat, etc.), sammelt und berichtet historische System-Auslastungsdaten: CPU, Speicher, Paging, I/O, Netzwerk etc. sar kann *laufende* Stats anzeigen (wenn mit Interval aufgerufen), oder *vergangene Tageswerte* aus Logs (/var/log/sysstat/...) ausgeben.

Syntax:

sar [Optionen] [Interval [Count]]

Ohne Parameter, sar liest per default die Tagesstatistiken (cumulated) vom laufenden Tag (falls sar cron job aktiv war). Mit Interval sar 5 5 z.B. 5x alle 5 Sekunden CPU etc.

Hauptoptionen (für Bereiche):

- **-u**: CPU-Auslastung (Default, wenn keine angegeben) – zeigt %user, %nice, %system, %iowait, %steal, %idle.
- **-r**: Memory (free, used, buffers, cache, commit, etc.).
- **-b**: I/O (transfer rates, i/o per sec, etc.).
- **-B**: Paging (pgpgin/s, pgpgout/s, fault, majflt...).

- **-W**: Swapping (swap in/out).
- **-q**: Queue length (runq, i/o queue) and load average.
- **-n DEV**: Network: DEV mode – zeigt Netzwerk-Interface Stats (rpxck/s, txpck/s, rxKB/s, txKB/s, etc.). -n ALL für alle Net-Statistiken.
- **-d**: Block device I/O (similar to iostat per device).
- **-p [DEV]**: CPU per processor (if SMP). e.g. sar -P ALL 1 3 – CPU usage per core.
- **-f file**: Lese Stats aus Log-Datei (z.B. /var/log/sysstat/sarXX).
- **-s HH:MM:SS, -e HH:MM:SS**: Start/End time for reading logs (z.B. 9 bis 18 Uhr Stats).

Beispiel (aktuelle CPU alle 1s):

```
$ sar 1 3
```

Linux 5.4.0 ... (8 CPUs)

```
02:00:00 PM  CPU  %user  %nice %system %iowait %steal  %idle
02:00:01 PM  all   5.13  0.00  1.28  0.00  0.00  93.59
02:00:02 PM  all   7.14  0.00  1.02  0.00  0.00  91.84
02:00:03 PM  all   3.06  0.00  0.77  0.00  0.00  96.16
Average:    all   5.11  0.00  1.02  0.00  0.00  93.87
```

nohup (No Hangup)

Beschreibung: Lässt einen Prozess auch nach Logout weiterlaufen. nohup fängt das HUP-Signal (Hangup) ab, das normalerweise gesendet wird, wenn ein Terminal geschlossen wird. Mit nohup startet man ein Kommando so, dass es dieses Signal ignoriert und Standardausgabe ggf. in *nohup.out* umleite】 .

Syntax:

```
nohup <Befehl> [Arg...] [&]
```

Funktionsweise:

- nohup sorgt dafür, dass der ausgeführte Prozess das SIGHUP ignoriert. Außerdem werden *stdout* und *stderr* umgeleitet in *nohup.out*, falls diese derzeit ans Terminal gebunden sind. (Wenn man & anhängt – also im Hintergrund – ist es üblich, die Ausgabe nicht am Terminal zu lassen).

- Üblicherweise kombiniert mit & (Hintergrund). Man loggt dann aus, und das Programm läuft weiter.

Beispiele:

- `nohup long_running.sh &` – Führt *long_running.sh* im Hintergrund aus, Ausgaben gehen in *nohup.out*. Der Befehl überlebt Logout. Die Shell gibt z.B. zurück: appending output to nohup.out.
- `nohup python server.py > server.log 2>&1 &` – Startet Python-Server im Hintergrund, leitet stdout und stderr in *server.log* (damit *nohup.out* nicht genutzt).

Hinweis: Wenn man bereits in einer Screen- oder tmux-Session ist, braucht man kein nohup; auch systemd-run oder Cron-Jobs sind Alternativen. Aber nohup ist ein schnelles Mittel.

`mkswap/swapon/swapoff` (Swap einrichten und verwalten)

- **mkswap** – Initialisiert eine Partition/Datei als Swapbereich. Syntax: `mkswap [Optionen] <Gerät/datei>`. Wichtige Option: -L <Label> (Swap-Label setzen), -U <UUID>. Beispiel: `mkswap /dev/sda3`.
- **swapon** – Aktiviert einen Swapbereich für Nutzung durch das System. Nach mkswap muss Partition via swapon <Device> eingebunden werden (oder in /etc/fstab eingetragen mit swap Typ). Option -a aktiviert alle in fstab markierten Swaps. --show zeigt aktuellen Swap an (Äquivalent cat /proc/swaps).
- **swapoff** – Deaktiviert Swap. Syntax: `swapoff <Device>` oder `swapoff -a` für alle.

Beispiele:

```
# mkswap -L "SWAP" /dev/sda3
```

Setting up swapspace version 1, size = 8 GiB (some hex UUID)

```
# swapon /dev/sda3
```

```
# swapon --show
```

| NAME | TYPE | SIZE | USED | PRIOR |
|-----------|-----------|------|------|-------|
| /dev/sda3 | partition | 8G | 0B | -2 |

`swapoff -a` würde alle Swapbereiche aus dem Betrieb nehmen (z.B. vor Partition-Vergrößerung oder Kernel-Dump etc.).

`free` (Speicherübersicht)

Beschreibung: Zeigt eine Übersicht über den aktuellen Speicherverbrauch (RAM und Swap) im System. Es summiert total, used, free, and cached/buffered memory.

Syntax:

free [Optionen]

Optionen:

- **-h**: Human Readable (z.B. MiB/GiB) .
- **-m, -g, -k**: Ausgabe in MiB, GiB, KiB.
- **-b**: in Bytes.
- **-t**: Fügt eine "Total" Zeile (RAM+Swap) hinzu.
- **-s N**: Aktualisiert alle N Sekunden (kontinuierliche Ausgabe wie watch).
- **-c N**: Wenn mit -s genutzt, stop nach N Aktualisierungen.
- **--si**: Benutzt 1000er statt 1024er Basis (KB = 1000 Bytes etc.).

Beispiel free -h:

| | total | used | free | shared | buff/cache | available |
|-------|-------|-------|-------|--------|------------|-----------|
| Mem: | 15Gi | 5.2Gi | 2.3Gi | 1.1Gi | 7.4Gi | 8.8Gi |
| Swap: | 2.0Gi | 0.0Gi | 2.0Gi | | | |

available ist eine Kernelschätzung, wieviel noch für neue Prozesse verfügbar, unter Berücksichtigung von caches, etc.

vmstat (Virtual Memory and system stats)

Beschreibung: Zeigt verschiedene Systemmetriken: Speicher, Auslagerung, CPU, Interrupts, Context Switches etc. im zeitlichen Verlauf. vmstat ohne Parameter gibt Durchschnitt seit Boot; mit Intervall zeigt periodische Werte (Differenzen pro Intervall).

Syntax:

vmstat [Optionen] [Interval [Count]]

Wichtige Spalten (Standardausgabe):

- **procs**: *r* (Anzahl laufender Prozesse im Runqueue), *b* (blockierte Prozesse).
- **memory**: *swpd* (Swap genutzt), *free* (freier RAM), *buff* (Puffer), *cache* (Page Cache).
- **swap**: *si* (Swap-In KB/s), *so* (Swap-Out KB/s).
- **io**: *bi* (Block In, Blöcke gelesen von Disk/s), *bo* (Block Out, geschrieben/s).
- **system**: *in* (Interrupts/s), *cs* (Context Switches/s).

- **cpu:** *us* (% CPU in Userzeit), *sy* (% Systemzeit), *id* (% Idle), *wa* (% IO-Wait), *st* (% Steal bei virt.).

Optionen:

- **-S <unit>:** Werte in k (1000) oder K (1024) etc.
- **-a:** Zeigt zusätzlich Aktivität der *aktiven/inaktiven* Memory Listen (sin, sout).
- **-d:** Zeigt Disk-Statistiken (like iostat) statt VM stats.
- **-p <dev>:** Zeigt Partition-Statistiken (reads/writes tot und sek) für Device.
- **-s:** Summiert memory stats (eine Liste aller relevanten Zähler, analog /proc/meminfo).
- **-m, -t:** Variation der Ausgabe (meist uninteressant, -m memory in MB etc).

Beispiel (alle 2s):

```
$ vmstat 2 3
procs -----memory----- swap-- -----io---- -system-- -----cpu-----
r b free buff cache si so bi bo in cs us sy id wa st
1 0 230000 120000 760000 0 0 5 10 150 300 5 191 3 0
0 0 229000 120000 760500 0 0 0 20 140 280 4 194 1 0
1 0 228500 120000 761000 0 0 0 30 160 320 6 192 1 0
```

Erste Zeile nach Header ist seit Boot (ohne parameter, oft uninteressant); weitere sind Echtzeit pro Intervall.

modinfo (Module Information)

Beschreibung: Zeigt Informationen über ein Linux-Kernelmodul (Loadable Kernel Module) – Version, Beschreibung, Lizenz, Abhängigkeiten, Parameter etc】 .

Syntax:

```
modinfo [Optionen] <Modulname oder Modul.ko>
```

Optionen:

- **-a, --author:** Nur Autor ausgeben.
- **-d, --description:** Nur Beschreibung.
- **-n, --filename:** Zeigt Pfad der Moduldatei im Filesystem.
- **-p, --parameters:** Listet die Parameter und ihre Typen/Beschreibungen.

- **-F <field>**: Gibt nur das spezifizierte Feld aus (alternativ obige Kurzoptionen). Mögliche Felder: author, description, license, parm, alias, etc.
- **-k <kernelversion>**: Fragt Infos für Kernel-Version (sucht in /lib/modules/<version>) statt laufendem Kernel.

Beispiel: modinfo e1000e (Intel NIC Treiber) gibt etwa:

```
filename: /lib/modules/5.4.0/kernel/drivers/net/ethernet/intel/e1000e/e1000e.ko
version: 3.2.6-k
license: GPL
description: Intel(R) PRO/1000 Network Driver
author: Intel Corporation, <e1000-devel@lists.sourceforge.net>
...
srcversion: 5C6D1A8....
alias: pci:v00008086d000010D3sv*sd*bc*sc*i*
...
parm: IntMode:Interrupt Mode (0=Auto, 1=MSI, 2=MSI-X) (int)
parm: InterruptThrottleRate:... (array of int)
```

Man sieht Dateipfad, Version, Parameter etc.

insmod/rmmod/modprobe (Kernelmodule laden/entladen)

- **insmod <pfad/modul.ko> [args]**: Lädt ein Modul in den Kernel. Nutzt genau Pfad; löst keine Abhängigkeiten auf.
- **rmmod <modulname>**: Entfernt ein geladenes Modul (wenn kein anderer in Benutzung). Option -f zum Erzwingen (nur falls Kernel CONFIG_MODULE_FORCE_UNLOAD).
- **modprobe [Options] <modulname> [args]**: Empfehlenswerter Befehl zum Laden eines Moduls und seiner Abhängigkeiten. Es durchsucht /lib/modules/<kernel>/modules.dep nach Dependencies. Auch zum Entfernen via modprobe -r <modul> (lädt auch Abhängigkeiten aus, falls unbenutzt).
 - Optionen: --dry-run oder -n testet nur, -v verbose, -q quiet, alias handling etc. modprobe Konfig steht in /etc/modprobe.d.
 - Bsp: modprobe e1000e (lädt Intel NIC modul + ggf. benötigte mii). modprobe -r e1000e entfernt es.

depmod (Dependency modules)

Beschreibung: Generiert die Modulabhängigkeitsdatei *modules.dep* basierend auf Modulen in */lib/modules/<version>*. Wird meist beim Kernel-Install aufgerufen. Kann man manuell ausführen nach Hinzufügen eigener Module. Syntax: depmod -a (für alle).

Optionen:

- -a, --all (default): alle Kernel-Versionen verarbeiten.
- -A, --quick: nur verändern, wenn *modules.dep* älter als Module-Dateien (incremental update).
- -n: Ausgabe nach *stdout* statt in *modules.dep* schreiben (Test).
- -v: verbose.

sysctl (Query/Set Kernel Parameters)

Beschreibung: Anzeige oder Änderung von Kernel-Laufzeitparametern (in */proc/sys*). sysctl ermöglicht bequemen Zugriff auf z.B. Netzwerktruning, VM, FS Settings zur Laufzeit.

Syntax:

sysctl [Optionen] [<Name>=<Wert> ...] or sysctl -p

Optionen:

- Ohne =: liest Parameter. Z.B. `sysctl net.ipv4.ip_forward` -> `net.ipv4.ip_forward = 0`.
- Mit name=value: setzt Parameter. Z.B. `sysctl net.ipv4.ip_forward=1` aktiviert IP-Forwarding (sofort).
- -w: explizit Schreibmodus (kann man weglassen, war historisch).
- -p [Datei]: Lädt Einstellungen aus Datei (Default */etc/sysctl.conf*) und wendet an. Man ruft `sysctl -p` typ. beim Boot (init-Skripte) oder manuell nach Bearbeiten der config.
- -a: Alle Parameter anzeigen (riesige Liste). `sysctl -a | grep ipc etc`.
- -A, --pattern <regex>: Filtert mit Regex (statt manuell grep).
- -e: Ignoriert unbekannte Keys (useful mit -p, sonst bricht ab).
- -N: Nur Namen, keine Werte anzeigen (list keys).
- -q: Quiet (unterdrückt Bestätigungsmeldungen).

Parameter-Keys entsprechen */proc/sys* Pfaden, z.B. */proc/sys/net/ipv4/ip_forward* -> key `net.ipv4.ip_forward`.

Beispiele:

- sysctl vm.swappiness -> zeigt swappiness (z.B. vm.swappiness = 60).
- sysctl -w vm.swappiness=10 -> setzt auf 10 (weniger Swapneigung).
- sysctl -p /etc/sysctl.d/99-sysctl.conf -> lädt Parameter aus der angegebenen Datei.

dmesg (Dump kernel message buffer)

Beschreibung: Zeigt die Nachrichten des Kernel-Ringspeichers (Kernellog) an. Hier stehen Boot-Meldungen und Hardware-Events (z.B. USB eingesteckt, Treiber Ausgabe). dmesg ohne Optionen druckt alle aktuellen Kernelmeldungen.

Syntax:

dmesg [Optionen]

Optionen:

- **-T**: Zeigt menschenlesbare Zeitstempel (Jahr-Monat-Tag Stunde:Minute:Sekunde) statt der raw Sekunden seit Boot. (Achtung: Kann ungenau sein, da Kernel keine Zeitzone kennt; -T rechnet anhand aktueller TZ).
- **-l <level>, --level**: Filtert Nachrichten bestimmter Log-Levels. Level: emerg, alert, crit, err, warn, notice, info, debug. Z.B. dmesg -l err, warn nur Fehler und Warnungen.
- **-k, --kernel**: Zeigt nur Kernel-Messages (Standard). **-u, --userspace**: Nur Benutzerraum (Logeinträge vom printk_user). Kaum genutzt, da meiste aus Kernel.
- **-H, --human: Pager-Modus** (lässt scrollen, farbig Highlighting). Interaktiv (Space/PgUp etc.).
- **-w, --follow**: "Tail -f" Modus für dmesg – neue Kernelmeldungen laufend anzeigen.
- **-s N**: Buffer-Größe (Bytes) die gelesen wird (Default könnte begrenzt sein). dmesg -s 1000000.

Beispiel (gekürzt):

```
[ 0.000000] Linux version 5.4.0-80-generic ...
[ 0.345678] PCIe Bus 0000:00: root hub ...
[ 1.234567] usb 1-2: new high-speed USB device number 3 ...
[ 1.345678] ata1: SATA link up ...
[ 10.567890] Bluetooth: RFCOMM ver 1.11
```

dmesg -T zeigt z.B. [Mon Oct 11 14:23:45 2021]

Typisch ruft man dmesg | grep -i error um nach Kernel-Fehlern zu suchen, oder dmesg -w in einem Terminal um z.B. neue USB/Hardware Events sofort zu sehen.

Dienstverwaltung und Systemstart

systemctl (Control systemd)

Beschreibung: Haupt-Werkzeug zur Steuerung des *systemd*-Initsystems (heutzutage auf vielen Linux-Distributionen). systemctl verwaltet **Services** (Start/Stop/Enable), **Daemon-Konfiguration, System-Zustände** (Reboot/Shutdown), **Unit**-Status und Logs, etc】 .

Syntax:

systemctl [Optionen] <Sub-Befehl> [Unit/...]

Units können Services (.service), Mounts (.mount), Timer (.timer), Sockets (.socket), etc. sein. Oft kann man den Suffix weglassen (z.B. "ssh" wird als "ssh.service" erkannt).

Wichtige Sub-Befehle (Service Management):

- **start <service>, stop <service>** – Startet oder stoppt einen Dienst *sofortfile-4fdk7rvx39azjretinarqb】 . Bsp: systemctl start apache2.
- **restart <service>** – Stop + Start (auch wenn nicht lief, wird gestartet).
- **reload <service>** – Lädt Konfiguration neu (sendet i.d.R. SIGHUP). Muss vom Service unterstützt werden (in Unit als ReloadSignal oder ExecReload definiert). Bsp: systemctl reload nginx.
- **status <service>** – Zeigt Status des Dienstes: Läuft er? Letzte Startmeldung, ggf. Auszug aus Journal-Log】 .
- **enable <service>** – Aktiviert den Dienst beim Boot (erstellt Symlinks in passenden WantedBy-Verzeichnissen).
- **disable <service>** – Deaktiviert Autostart.
- **is-enabled, is-active** – Prüfen ob Unit aktiviert ist / gerade aktiv ist (exit 0/1).
- **mask <service>** – "Maskiert" den Service, d.h. verhindert Start vollkommen (setzt Link auf /dev/null). *Unmask* zum Aufheben.
- **daemon-reload** – Falls man eine Unit-Datei geändert oder neu hinzugefügt hat, diese neu einlesen (Konfig neu laden, ohne Dienste neu zu starten).
- **kill <service>** – Sendet ein Signal an alle Prozesse der Service-Unit (z.B. systemctl kill -s SIGUSR1 myapp.service).

System Commands:

- **list-units** – Listet alle geladenen Einheiten (nur aktive standardmäßig). --all um Inaktive auch zu sehen, --type=service filtern auf Services, etc.
- **list-unit-files** – Listet Unit-Dateien und ob sie enabled/disabled/etc. sind (Installationszustand).
- **list-dependencies <unit>** – Zeigt Abhängigkeiten (Requires/Wants) Baum für Unit. --reverse zeigt welche Units diese Unit benötigen.
- **show <unit>** – Gibt alle Properties von Unit aus (oder subset via -p).
- **cat <unit>** – Zeigt den Inhalt der Unit-Datei (und Drop-Ins).
- **edit <unit>** – Öffnet Editor für Drop-In Snippet, um diese Unit zu überschreiben/erweitern.
- **preset <service>** – Setzt enable/disable gemäß Distribution-Vorgabe (in /usr/lib/systemd/system-preset/).

Targets (Runlevels):

- **isolate <target>** – Wechselt in ein target (beendet andere). Z.B. systemctl isolate rescue.target (Einzeluser-Modus ähnlich runlevel 1).
- **set-default <target>** – Legt Default Boot-Target fest (z.B. graphical.target vs multi-user.target).
- **get-default** – zeigt aktuelles Defaulttarget.

Power Management:

- **reboot, poweroff, halt** – führen Neustart bzw. Herunterfahren aus (rufen intern entsprechende target isolation).
- **suspend, hibernate, hybrid-sleep** – starten Energiesparmodus.

Journal/Logging Integration:

- **status** (wie oben) zeigt bereits Log-Auszug.
- **journalctl** (separater Befehl, nicht systemctl subcommand) – zum Betrachten der systemd Logs. Z.B. journalctl -u <service> zeigt Logs zu einem Service, -f zum folgen, --since "2021-10-01" ab bestimmtem Datum etc.

Beispiele:

- systemctl enable --now nginx – Aktiviert **und** startet nginx sofort (Option --now führt gleich start nach enable aus).
- systemctl disable apache2.service – Entfernt Apache2 aus Autostart.
- systemctl restart networking – Dienst neu starten (bei Netz aufpassen).

- `systemctl list-units --failed` – Zeigt nur fehlgeschlagene Units.
- `systemctl mask firewalld; systemctl stop firewalld` – Verhindert Start des Firewall dauerhaft und stoppt ihn sofort (z.B. falls man anderen Firewall nutzt).

service (SysVinit Compatibility)

Beschreibung: Der Befehl `service` war bei SysVinit/Upstart gebräuchlich, um Start/Stop-Skripte in `/etc/init.d` aufzurufen. Auf systemd-Systemen ruft `service` intern `systemctl` auf. Man kann aber `service <name> start/stop/status` noch benutzen.
`4fdk7rvx39azjretinarqb】`. Moderne Systeme – besser `systemctl` direkt.

Syntax:

`service <ServiceName> [start|stop|restart|status|...]`

Beispiel: `service ssh status` ruft `systemctl status ssh.service` auf.

(Auf SysVinit ohne systemd würde es das `init.d` Script ausführen.)

chkconfig (Runlevel Konfiguration - SysV kompatibel)

Beschreibung: Ein Werkzeug zum Verwalten der Start-/Stopp-Links von SysV-Init-Skripten über verschiedene Runlevels (0-6file-4fdk7rvx39azjretinarqb】). Auf systemd-Systemen gegenstandslos (meist vorhanden für Kompatibilität, was auf `systemctl enable/disable` mappt).

Wichtigste Befehle:

- `chkconfig --list [name]` – Listet alle Dienste und ob sie in runlevel 0-6 on/off sind.
`4fdk7rvx39azjretinarqb】`. Mit Name nur diesen.
- `chkconfig name on` (bzw. `off`) – Schaltet Service in allen entsprechenden runlevels an oder aus (Standard runlevels 2-5 on für on).
- `chkconfig name --level 35 on` – Feinsteuerung: aktiviert Dienst in Runlevel 3 und 5.
- `chkconfig name reset` – Setzt Standard (wie im Init-Skript Header vorgesehen).

**(Auf systemd delegiert `chkconfig` intern an `systemctl enable/disable`. Für Nostalgie, wissen was es tat.)*

Zeitplanung (Cron und At)

crontab (Cron Table Management)

Beschreibung: Planen wiederkehrender Aufgaben. `crontab` bearbeitet die Aufgabenliste des aktuellen Benutzers (oder eines anderen mit `-u`) im Cron-Daemon. Jede Zeile in `crontab -e` hat Format "Min Stunde Tag Monat Wochentag Befehl". Der Cron-Dienst prüft pro Minute die Tabellen und führt fällige Kommandos aus.
`4fdk7rvx39azjretinarqb】`.

Befehle:

- crontab -e – Editiert die Crontab des aktuellen Benutzers (öffnet \$EDITOR). Beim Speichern wird die neue Tabelle dem Cron-Daemon übergeben [4fdk7rvx39azjretinarqb].
- crontab -l – Listet die aktuelle Crontab auf [file-4fdk7rvx39azjretinarqb].
- crontab -r – Entfernt (löscht) die aktuelle Crontab komplett. (Vorsicht, kein Undo).
- crontab -u username -e/-l/-r – Als root: Bearbeitet/liest/löscht die Crontab eines anderen Benutzers.

Cron Format Erinnerung:

Minute(0-59) Hour(0-23) Day(1-31) Month(1-12) Weekday(0-7) Command

0 2 * * * /usr/bin/backup.sh # täglich 02:00

*/5 9-17 * * 1-5 echo "Working..." # Werktags 9-17 Uhr alle 5 min

*=jeder Wert, Komma getrennte Listen, Intervalle mit -, Step mit /.

Spezial @-Strings: @reboot, @hourly, @daily, @yearly etc.

Anacron: Cron führt Jobs aus, wann Maschine an ist. Für nicht ständig laufende Rechner nutzt man anacron (Tagesjobs nachholen).

at (One-time scheduled task)

Beschreibung: Legt einmalige Ausführungen von Befehlen zu einer bestimmten Zeit in der Zukunft fest [file-4fdk7rvx39azjretinarqb]. at nimmt eine Zeitangabe entgegen und liest dann von STDIN die Befehle, die ausgeführt werden sollen (oder mit -f aus einer Datei). Der atd Daemon muss laufen.

Verwendung:

echo "/path/to/script.sh" | at 02:30 # führt Script heute Nacht 2:30 aus

Man ruft at <Zeit> auf, landet in einer Stdin-Eingabe (end mit Ctrl+D) oder speist via echo/<<EOF>>.

Zeitformat: Z.B. at 14:00 (heute 14 Uhr, falls schon vorbei dann morgen), at 11pm, at midnight, at now + 5 days, at 08:00 12.12.2025 (8:00 am 12 Dec 2025). Kann englische Wörter: tomorrow, noon, teatime(16:00) etc.

Befehle:

- atq – Zeigt anstehende at-Jobs (Queue) des aktuellen Userfiles [file-4fdk7rvx39azjretinarqb]. (Listet Job-IDs und geplante Zeit). Root sieht alle oder atq -q <Queue> für bestimmte Batch-Queues.

- **atrm <JobID>** – Entfernt einen geplanten Job aus der Queue. (JobID sieht man in atq).
- **batch** – Alias zu at -q b now (führt Jobs aus, wenn Load Average niedrig ist). "Batch" schiebt den Auftrag in Batch-Queue, läuft sobald System nicht ausgelastet (LA<0.8 default).

Beispiel:

```
$ at now + 1 minute
warning: commands will be executed using /bin/sh
at> echo "Hello after 1 min" >> /tmp/test.txt
at> <EOT>
job 5 at Fri Oct 15 16:41:00 2021
```

Dann innerhalb einer Minute atq zeigt e.g. 5 Fri Oct 15 16:41:00 2021 a user. Nach Ausführung landet Text in Datei.

(Zugriff: /etc/at.allow /etc/at.deny definieren wer at nutzen darf – analog cron.allow/deny.)

Drucken

lpr (Line Print Request)

Beschreibung: Sendet eine Datei (oder STDIN) an den Drucker (Print-Job einreihen). lpr ist Teil des BSD-Linux-Drucksystems (CUPS stellt Kompatibilität bereit). Alternative: lp (System V style).

Syntax:

lpr [Optionen] [Dateien...]

Wichtige Optionen:

- **-P <Druckernname>**: Wählt einen bestimmten Drucker aus (sonst Standarddruckerfile-4fdk7rvx39azjretinarqb】). Druckernamen findet man via lpstat -a oder CUPS Webinterface.
- **-# <Num>**: Anzahl Kopien drucken (z.B. -#3 dreifach).
- **-h**: Unterdrückt Banner-Seite (header page) falls aktiviert.
- **-r**: Löscht die Datei(en) nach dem Versenden an den Spool (rarely used).
- **-m**: Sendet E-Mail Benachrichtigung nach Druck (sofern eingerichtet).
- **-o <Option>=<Wert>**: Drucker-spezifische Optionen übergeben (z.B. -o landscape für Querformat, oder -o media=A4 oder -o sides=two-sided-long-edge

für Duplex). Diese Optionen sind je nach Drucker Treiber definierbar (CUPS options).

- **-C <Klasse>, -J <Jobname>, -T <Titel>**: Legen Klassenname, Jobname, Titel für Banner etc. fest.
- **--help** (GNU lpr) usw.

Beispiele:

- lpr report.pdf – Schickt *report.pdf* an Standarddrucker.
- lpr -P HP_LaserJet -#2 invoice.txt – Druckt *invoice.txt* 2-fach auf Drucker "HP_LaserJet".
- echo "Test Page" | lpr – Druckt STDIN (den Text "Test Page").
- lpr -o sides=two-sided-long-edge thesis.pdf – Duplexdruck beidseitig.

(Verwandt: *lpq* zeigt Druckerqueue, *lprm* löscht Jobs aus der Queue.)

Lokale Systemkonfiguration

date (Anzeige/Setzen Datum/Uhrzeit)

Beschreibung: Zeigt das aktuelle Datum und die Uhrzeit an, oder stellt sie ein. Auch Formatierung in beliebigem Muster ist möglich.

Syntax:

date [Optionen] [+Format]

date -s "STRING" (setzen)

Optionen / Verwendung:

- **Ausgabe formatieren:** date "+FORMAT" – Mit Format-Pattern ähnlich strftime: %Y (Jahr), %m Monat(01), %b Monat (Kurzname), %d Tag, %H Stunde (24h), %I (12h), %M Minute, %S Sek, %A Wochentag, %a (kurz), %Z Zeitzone, %z offset, %T = %H:%M:%S, %F = %Y-%m-%d, etc. Z.B. date "+%d.%m.%Y %H:%M:%S" -> "05.10.2025 14:30:00".
- **Systemzeit stellen:** date -s "2025-10-05 14:30:00" – Setzt Datum. Braucht Root. (Alternative: date --set="...").
- **Hardware-Uhr stellen:** In Linux erfolgt date auf Systemzeit (in Kernel, RTC separiert). Um auch BIOS/RTC Uhr zu setzen, hwclock --systohc (setze Hardware Clock nach System Clock). Umgekehrt hwclock --hctosys.
- **Universale Zeit:** date -u zeigt in UTC. date -u -s "2025-10-05 12:30:00" setzt in UTC.

- **RFC-3339 Format:** date --rfc-3339=ns etc. Standard ISO.
- **Unix Timestamp:** date +%s gibt Sekunden seit 1.1.1970 (UTC).

Beispiele:

- date -> "Tue Oct 5 14:30:00 CEST 2025".
- date +"%Y-%m-%d_%H-%M-%S" -> "2025-10-05_14-30-00".
- sudo date -s "@\$((\$(date +%s) + 3600))" -> Systemzeit um eine Stunde vorstellen
(Beispiel mit timestamp math).

timedatectl (Zeiteinstellung via systemd)

Beschreibung: Befehl von systemd zum Anzeigen und Ändern der Systemzeit, Hardware-Uhr, Zeitzone und NTP-Synchronisatiofile-4fdk7rvx39azjretinarqb】 .

Verwendung:

- **timedatectl** (ohne args): Zeigt aktuellen Status: Lokale Zeit, UTC, RTC, Zeitzone, ob NTP aktiv ist etcfile-4fdk7rvx39azjretinarqb】 .
- **timedatectl set-time "YYYY-MM-DD HH:MM:SS"** – Setzt Systemzeit (wenn kein NTP aktiv).
- **timedatectl set-timezone <Zone>** – Ändert Zeitzone. Z.B. timedatectl set-timezone Europe/Berlin. (Verfügbare Zonen unter /usr/share/zoneinfo/, oder timedatectl list-timezones).
- **timedatectl set-ntp true/false** – Aktiviert bzw. deaktiviert die automatische Zeitsynchronisation (via systemd-timesyncd oder anderes NTP). Bei true startet es den Dienst falls vorhanden.
- **timedatectl set-local-rtc 1/0** – Legt fest, ob die Hardware-Uhr (RTC) in **lokaler Zeit** läuft (1) oder in UTC (0, Standard). Windows dual-boot oft RTC=lokal (dann 1), sonst belässt auf 0=UTF um Probleme zu vermeiden.

Beispiele:

```
$ timedatectl
```

Local time: Tue 2025-10-05 14:30:00 CEST

Universal time: Tue 2025-10-05 12:30:00 UTC

RTC time: Tue 2025-10-05 12:30:00

Time zone: Europe/Berlin (CEST, +0200)

System clock synchronized: yes

NTP service: active

RTC in local TZ: no

timedatectl set-time "2025-12-24 18:00:00" stellt Heiligabend 18:00. timedatectl set-timezone UTC wechselt Zeitzone auf UTC.

localectl (Locale and Keyboard Layout)

Beschreibung: Verwaltet System-Locale (Sprache, Gebietsschema) und Tastaturlayout-Einstellungen in systemd-basierten Systemefile-4fdk7rvx39azjretinarqb】. Es kann die Variablen in /etc/locale.conf sowie Konsolen-Tastaturlayout in /etc/vconsole.conf setzen. (Für X/Wayland Tastatur greift localectl in /etc/X11/xorg.conf.d/ oder mittels setxkbmap separat).

Verwendung:

- **localectl** (ohne args): Zeigt aktuelle Locale(s) und Keymap. Z.B.:
System Locale: LANG=de_DE.UTF-8
VC Keymap: de
X11 Layout: de
X11 Model: pc105 etc.
- **localectl list-locales** – Listet alle verfügbaren Locale, die generiert sind (die in /usr/lib/locale oder /etc/locale.gen markiert).
- **localectl set-locale LANG=<locale> [VAR=locale...]** – Setzt System-Locale. Z.B. localectl set-locale LANG=en_US.UTF-8. Optional mehrere, z.B. LC_TIME=en_GB.UTF-8. Wirksam nach Neustart oder neu login shells.
- **localectl list-keymaps** – Listet verfügbare Console-Keymaps (z.B. us, de, ...).
- **localectl set-keymap <map> [<toggle>]** – Setzt Tastaturlayout für Konsole (vconsole) und ggf. X11, falls X11 Layout nicht getrennt definiert. Z.B. localectl set-keymap de.
- **localectl set-x11-keymap <layout> [model] [variant] [option]** – Spezifisch X11 (Graphical) Layout/Modell setzen. Z.B. localectl set-x11-keymap us pc104 "" terminate:ctrl_alt_bksp.

Beispiel:

localectl set-locale LANG=fr_FR.UTF-8 (Systemsprache auf Französisch).

localectl set-keymap us (Konsole auf US-Layout).

localectl set-x11-keymap de pc105 nodeadkeys (Grafische Tastatur auf Deutsch mit Variante "nodeadkeys").

(Die Konfigurationen landen in /etc/locale.conf, /etc/vconsole.conf, /etc/X11/xorg.conf.d/00-keyboard.conf entsprechend.)

Netzwerk-Konfiguration

ip (Netlink Interface Config)

Beschreibung: Moderner Befehl zur Verwaltung von Netzwerkinterfaces, IP-Adressen, Routing, Tunnels etc. (Nachfolger von ifconfig/route). ip hat Unterkommandos für verschiedene Objekte: **link** (Netzwerkinterface), **addr** (IP-Adressen), **route**, **neigh** (ARP/NDP Nachbarn), **rule** (Policy Routing), **maddr** (Multicast), **addrlabel**, **netns** (Net Namespaces), **link set** (Interface-Settings).

Syntax (Grundstruktur):

```
ip [Options] <Object> <Subcommand> [Arguments...]
```

z.B. ip link show, ip addr add, ip route list, etc.

Häufige Befehle/Beispiele:

- **Interface anzeigen:** ip link show [dev <iface>] – Listet alle Netzwerkinterfaces (oder ein spezifisches). Ausgabe enthält Index, Name, MAC, MTU, Flags (UP,LOWER_UP=Link OK, MULTICAST), etc.
- **Interface aktivieren/deaktivieren:** ip link set <iface> up / down – entspricht `ifconfig iface up/downfile-4fdk7rvx39azjretinarqb】 .
- **Adresse zuweisen:** ip addr add <IP>/<Prefix> [broadcast +] dev <iface> – z.B. ip addr add 192.168.1.100/24 dev eth0. broadcast + lässt broadcast aus Prefix berechnen (optional).
- **Adresse löschen:** ip addr del <IP>/<Prefix> dev <iface>.
- **Adressen anzeigen:** ip addr show [dev <iface>] – Listet IPv4/IPv6 Adressen auf Interfacefile-4fdk7rvx39azjretinarqb】 . Zeigt auch Scope (global/link), flags (tentative, dynamic).
- **Standardroute setzen:** ip route add default via <Gateway-IP> [dev <iface>] – z.B. ip r add default via 192.168.1.1 dev eth0.
- **Statische Route setzen:** ip route add <Netz>/<Präfix> via <nächsterHop> dev <iface>.
- **Routen auflisten:** ip route show (oder ip r).
- **ARP Cache:** ip neigh show – zeigt ARP/ND-Cache (Neighbors).
- **ARP Eintrag fix setzen:** ip neigh add <IP> lladdr <MAC> dev <iface> nud permanent.
- **Secondary IPs / Aliase:** Einfach mehrfach ip addr add auf selbes dev ausführen (oder use Label).
- **MTU ändern:** ip link set dev <iface> mtu 1400.

- **MAC ändern:** ip link set dev <iface> address <NewMAC> (Down interface first).
- **Promisc Mode:** ip link set dev <iface> promisc on.
- **Routing Policy (advanced):** ip rule und ip route add table <n> for source-based routing etc.
- **Network Namespace mgmt:** ip netns (add, del, exec, etc – advanced).

Options global: -s (stats), -d (details), -br (brief output), -c (color output). Z.B. ip -br addr zeigt Interfaces in einzeiligem Kurzformat.

Beispiele:

- ip -br link ->

```
lo      UNKNOWN    127.0.0.1/8 ::1/128
eth0     UP        192.168.1.10/24 fe80::.../64
```

- ip addr add 10.0.0.1/16 dev eth1 (fügt IP).
- ip route show (z.B. default via 192.168.1.1 dev eth0 proto dhcp metric 100 etc.).
- ip neigh show (z.B. 192.168.1.1 dev eth0 lladdr aa:bb:cc:dd:ee:ff REACHABLE).

(ifconfig (net-tools) alternative: ip ist ausführlicher und bevorzugt.)

ifconfig (Interface Config – alt)

Beschreibung: Älteres Werkzeug aus net-tools (weitgehend ersetzt durch ip). ifconfig kann Interfaces anzeigen, IP setzen, MTU ändern, promiscuous Mode etc., aber unterstützt z.B. keine IPv6 out-of-the-box je nach Version.

Syntax:

ifconfig [Interface] [Optionen]

Beispiele:

- ifconfig – zeigt alle aktiven (UP) Interfaces mit IPv4 etc.
- ifconfig -a – zeigt *alle*, inkl. down Interfaces.
- ifconfig eth0 192.168.0.5 netmask 255.255.255.0 broadcast 192.168.0.255 – setzt IP, Maske, Bcast.
- ifconfig eth0 up / down – Interface an/aus.
- ifconfig eth0 mtu 1400 – MTU ändern.
- ifconfig eth0 promisc – Promiscuous Mode an. -promisc aus.
- ifconfig eth0 hw ether 00:11:22:33:44:55 – MAC-Adresse setzen.

(ifconfig ist nicht mehr standard installiert in neueren Distros; ip commands stattdessen nutzen.)

iwconfig (Wireless Config – alt)

Beschreibung: Ähnlich ifconfig, aber für Wireless LAN Schnittstellen (Legacy tool aus wireless-tools). Zeigt/ändert ESSID, Mode, Frequenz, Schlüssel etc. Heutzutage ersetzt durch iw (nl80211), oder NetworkManager.

Beispiele (falls vorhanden):

- iwconfig wlan0 – zeigt ESSID, Mode (Managed/Ad-Hoc/Master), Channel/Freq, Bitrate, Tx-Power, Signal Level, Encryption key on/off, etc.
- iwconfig wlan0 essid "WLAN-Name" – mit offener SSID verbinden (sofern AP ohne Sicherheit).
- iwconfig wlan0 key s:password – setzt WEP Key (veraltet).
- iwconfig wlan0 mode Monitor – Interface in Monitor Mode (für Sniffing).

(Iwconfig kann kein WPA2; da nutzt man wpa_supplicant oder NM, etc.)

nmcli (NetworkManager CLI)

Beschreibung: Kommandozeilen-Interface zum NetworkManager-Dienst, um Netzwerkkonfiguration (insb. Desktop-Umfeld) zu betrachten und steuerfile-4fdk7rvx39azjretinarqb】 . Kann Verbindungen (profiles) verwalten, aktivieren, WLAN scannen, etc.

Verwendung (Struktur):

nmcli [general|networking|radio|connection|device] <aktion> [Parameter]

Beispiele:

- **Status anzeigen:** nmcli general status – Gesamter NM-Status. nmcli connection show – listet gespeicherte Verbindungen (Profiles). nmcli device status – zeigt Geräte und ob connected und welches Profil.
- **WLAN scannen:** nmcli device wifi list – listet verfügbare WLANs.
- **Verbinden WLAN:** nmcli device wifi connect "SSID" password "passwort" – erstellt ggf. Connection-Profil und verbindet.
- **Verbindung aktivieren/deaktivieren:** nmcli connection up <Name> / down <Name>. Oder nmcli device disconnect <iface> (z.B. nmcli device disconnect wlan0).
- **Neue Verbindung anlegen:** nmcli connection add type ethernet ifname eth0 con-name "Wired Manual" ip4 192.168.5.5/24 gw4 192.168.5.1.

- **IPv4 ändern laufend:** nmcli connection modify "Wired connection 1" ipv4.addresses 192.168.1.50/24 ipv4.gateway 192.168.1.1 ipv4.method manual und dann up.
- **Ethernet an/aus:** nmcli networking off (setzt NM generell offline) oder nmcli radio wifi off (schaltet nur WLAN-Funk aus).

Beobachten:

- nmcli connection show --active – nur aktive Conns.
- nmcli -p device status – pretty (ncurses-like) output.

ethtool (NIC Driver Tool)

Beschreibung: Zeigt und ändert Einstellungen von Ethernet-Netzwerkkarten (Treiber-Level): z.B. Geschwindigkeit, Duplex, Auto-Negotiation, Wake-on-LAN, Offload-Funktionen, Statistikzählefile-4fdk7rvx39azjretinarqb】 .

Verwendung:

- ethtool <iface> – Zeigt Standard-Eigenschaften: unterstützt Geschwindigkeiten (Supported link modes), aktuelle Speed/Duplex, Auto-negotiation on/off, Ports (TP, etc.), driver name, firmware-version, bus-info, etc.
- ethtool -s <iface> speed <Mb/s> duplex <half/full> autoneg <on/off> – Setzt feste Geschwindigkeit/Duplex. (z.B. ethtool -s eth0 speed 100 duplex full autoneg off). Vorsicht: muss vom Link Partner unterstützt werden, sonst Link down.
- ethtool -k <iface> – Zeigt Kernel-Offload-Funktionen (tx-checksumming, scatter-gather, TSO etc.) und ob an/aus. -K ... (capital K) schaltet diese: z.B. ethtool -K eth0 gro off (GRO ausschalten).
- ethtool -i <iface> – Infos zum Treiber (Driver name, version, firmware, bus).
- ethtool -p <iface> [N] – "Physically identify": Blinkt die LED am Port für N Sekunden (Standard 15), falls Karte das unterstützt. Hilft um Kabelport zu finden.
- ethtool -S <iface> – Ausgabe von Statistikzählern (packets, errors, etc. vom Treiber).
- ethtool -g <iface> – Ring Puffer Größen (RX/TX rings). -G zum Setzen.
- ethtool -a <iface> – Flow-Control (Pause Frame) Status (Autoneg etc.), -A zum Setzen on/off rx/tx.
- ethtool -r <iface> – Veranlasst die Karte, Link neu auszuhandeln (Retrain).
- ethtool --identify <iface> – alias für -p (Blinken).

Beispiele:

- ethtool eth0 -> zeigt z.B. Speed: 1000Mb/s, Duplex: Full, Auto-negotiation: on, etc.
- ethtool -s eth0 speed 100 duplex half autoneg off -> zwingt 100Mbit Half-Duplex (nur Test oder Legacy).
- ethtool -p eth1 10 -> blinkt LED an eth1 für 10 Sekunden.
- ethtool -K eth0 tso off -> schaltet TCP Segment Offloading aus (im Troubleshooting falls Problem mit Offload).

hostnamectl (Control hostname in systemd)

Beschreibung: Zeigt oder ändert den System-Hostname und verwandte Settings (wie Chassis Typ, Icon). hostnamectl ist Teil von systemd.

Verwendung:

- **hostnamectl** (ohne args): Zeigt Systeminfo: Static hostname, Transient (via DHCP), Pretty (human name), und Kernel: Operating System, Kernel, Architecture, etc.
- **hostnamectl set-hostname <name> [--pretty|--static|--transient]** – Setzt den Hostnamen. Ohne Flag wird *static* und *transient* gleichermaßen gesetzt. *static* = in /etc/hostname gespeichert (persistenter Name), *transient* = Laufzeit (kann z.B. DHCP ändern). *pretty* = nur für Anzeigezwecke (z.B. mit Großbuchstaben/Leerzeichen erlaubt).
Beispiel: hostnamectl set-hostname server42 – setzt static+transient auf "server42".
hostnamectl set-hostname "My Home PC" --pretty – setzt Pretty-Name.
- **hostnamectl set-icon-name <icon>** – Setzt Icon (z.B. computer-desktop, rein kosmetisch in einigen GUIs).
- **hostnamectl set-chassis <type>** – Setzt Gehäuse-Typ: desktop, laptop, server, tablet, embedded, etc. (Auch nur informative, kann z.B. in GNOME Systeminfo auftauchen).

Zusatz:

- hostnamectl status (alias ohne args) – Info.
- Traditional hostname Befehl (ohne ctl) zeigt oder setzt nur transient (bis Reboot).

Netzwerk-Fehlerdiagnose und Name-Auflösung

netcat (nc) (TCP/UDP Swiss Army Knife)

Beschreibung: netcat (oder nc) ist ein universelles Netzwerk-Tool, um Verbindungen zu testen, Daten zu senden/empfangen, Ports zu lauschen etcfile-4fdk7rvx39azjretinarqb】. Kann sowohl Client als auch rudimentärer Server sein.

Verwendung:

- **Verbindung als Client:** nc [options] <Host> <Port> – Baut TCP-Verbindung zu Host:Port auf, leitet stdin -> Socket, und Socket -> stdout. So kann man z.B. echo "HEAD / HTTP/1.0" | nc example.com 80 (HTTP-Request senden). Oder nc time.nist.gov 13 (Daytime Protocol). Wenn man interaktiv ausführt, kann man manuell tippen (Telnet-Ersatz).
- **Port Listening (Server):** nc -l -p <Port> [Host] – Horcht auf Port (optional an gegebene IP gebunden). Bsp: nc -l -p 1234 lauscht auf TCP/1234 auf alle Interfaces; was reinkommt wird auf stdout ausgegeben, was man tippt geht an Client.
- **UDP mode:** nc -u – Statt TCP benutzt UDP. Bsp: nc -u 192.168.1.5 53 könnte man UDP Query schicken (z.B. an DNS server).
- **Port Scan:** nc -z -v host startport-endport – "Zero-I/O mode" nur scannen, nicht verbinden. Z.B. nc -z -v 192.168.1.10 1-1024 zeigt offene Ports. (Netcat als Portscanner ist möglich, aber Nmap dafür besser).
- **Timeout:** -w <secs> – Wartezeit bis Timeout für Verbindungen/Eingaben.
- **Keep-open (inetd style):** In GNU nc -k (listen bleibt offen nach Verbindungsende, auf neue).
- **Exec:** -e <prog> – *Unsichere* Option (nicht immer kompiliert, oft disabled aus Sicherheitsgründen): Leitet die I/O eines Programms an den Socket. So kann man z.B. remote shell spawn: nc -l -p 5000 -e /bin/bash. (Equivalent zum berüchtigten rm -rf example). Modern: use socat or ssh for shells.

Beispiele:

- **Chat/Echo Test:** Terminal A: nc -l -p 5000, Terminal B: nc localhost 5000 -> alles was A tippt sieht B und umgekehrt.
- **Datei senden:** Auf Empfänger: nc -l -p 4500 > out.txt, auf Sender: nc destIP 4500 < file.txt. (Netcat baut TCP auf und streamt Datei).
- **HTTP Request:** printf "GET / HTTP/1.1\r\nHost: example.com\r\n\r\n" | nc example.com 80 – zeigt HTML der Homepage (HTTP/1.1 erfordert Host-Header).
- **UDP Ping (ohne Antwort):** echo -n "Hello" | nc -u 192.168.1.5 9000.
- **Port scan:** nc -z -v 192.168.1.20 20-30 -> tries FTP-data, FTP etc, output open/closed.

(Netcat Implementierungen: Traditional vs OpenBSD netcat vs GNU netcat ncat (part of nmap) – Optionen leicht variierend. Z.B. OpenBSD netcat hat -l und -p zusammen als one option or so, etc.)

iftop (Interface TOP, Bandwidth monitor)

Beschreibung: Ein curses-basiertes Tool zum Anzeigen der aktuellen Bandbreitennutzung pro Verbindung (ähnlich top, aber für Netzwerk Trafficfile-4fdk7rvx39azjretinarqb】). Es zeigt welche Peers mit dem System kommunizieren und wieviel Bytes/s in/out. Hilfreich zum realtime Debugging wer Bandbreite frisst.

Verwendung:

iftop -i <Interface>

Ohne -i wählt iftop das erste externe Interface.

Optionen:

- **-n** – Keine DNS-Auflösung (zeigt IPs statt aufzulösen, wichtig um Ausgabe schnell zu haltenfile-4fdk7rvx39azjretinarqb】).
- **-N** – Keine Port-Auflösung (zeigt Portnummern statt Services Namen).
- **-P** – Zeigt Ports zusammen mit Host (separat Spalte). Kombiniert man oft: iftop -nP.
- **-B** – Bytes statt Bits in Anzeige (Standard iftop zeigt in bits/s).
- **-F <FilterNet>** – Nur Traffic, der in bestimmten Netzbereich fällt (Filter) betrachten, z.B. -F 192.168.0.0/16 filtert auf diese Range.
- **-p** – Promiscuous Mode (auch Traffic anzeigen, der nicht an lokalem Host endet? Normal iftop Filter host involvement).
- **-t** – Text-Only Modus (keine curses-Interface, nur in stdout Stats; nicht sehr üblich).

Im Interface kann man mit Tasten: h Hilfe, n togg DNS, s sort by source, d sort by dest, t sort by throughput, p toggle port display, b toggle bar graph, B cycles through rate intervals (default 2s,10s,40s windows).

traceroute / tracepath (Route Tracking)

Beschreibung: Sendet eine Serie von Paketen mit steigenden TTL (Time-to-Live) Werten, um den Pfad (Router Hopps) zum Ziel auszumessefile-4fdk7rvx39azjretinarqb】. Traceroute verwendet standard UDP oder ICMP ECHO, tracepath nutzt UDP with high port (no root needed). Jedes Hop sendet "Time Exceeded" zurück, das traceroute protokolliert. Ergebnis: Liste der Router vom Quell- zum Zielnetz.

Syntax:

traceroute [Optionen] <Host> [PacketSize]

tracepath <Host>

Optionen (traceroute):

- **-I:** ICMP Echo statt UDP verwenden (pakete wie ping).
- **-T:** TCP SYN statt UDP (z.B. um Firewalls zu durchdringen über port 80).
- **-p <Port>:** UDP port Start (Standard 33434).
- **-q <N>:** Anzahl Probes pro TTL (Default 3).
- **-w <Seconds>:** Timeout pro Hop Antwort (default 5s).
- **-n:** Nicht auflösen (IP statt DNSfile-4fdk7rvx39azjretinarqb】 .
- **-m <max_ttl>:** Max Hop Count (Standard 30).
- **-f <first_ttl>:** Start TTL (default 1). Manchmal hilfreich, mittendrin einzusteigen.
- **-4 / -6:** Force IPv4 / IPv6.

Optionen (tracepath) sind wenige: -n no dns, -b show both hostnames and IP, etc.

Beispiel:

\$ traceroute example.com

```
1 192.168.1.1 (192.168.1.1) 1.123 ms 0.846 ms 0.799 ms
2 100.64.0.1 (100.64.0.1) 10.123 ms 9.876 ms 9.543 ms
3 203.0.113.5 (203.0.113.5) 15.432 ms 14.876 ms 15.210 ms
4 ae12.dar02.example.net (198.51.100.1) 30.456 ms 29.999 ms 30.123 ms
5 93.184.216.34 (93.184.216.34) 35.678 ms 35.900 ms 35.456 ms
```

Zeigt 5 Hops (Heimrouter, CGNAT router, ISP core, example.net router, Ziel).

tracepath example.com ähnlich, oft mit asymmetrischen result such as pmtu info.

mtr (My Traceroute)

Beschreibung: Kombiniert traceroute und ping in einem kontinuierlichen Diagnose-Tool. mtr sendet fortlaufend Pakete über alle Hops und aktualisiert in Echtzeit die Latenz und Packetloss-Statistiken pro Ho】 . Hilfreich um instabile Verbindungen/Packet Loss auf bestimmten Strecken zu identifizieren.

Verwendung:

mtr [Optionen] <Ziel>

Interaktiv curses UI standard, oder mit -r (report) Modus für einmalige Ausgabe.

Optionen:

- **-4 / -6**: Force IPv4/IPv6.
- **-n**: Nicht auflösen, nur IPs (schneller).
- **-w**: Wide-output (zeigt beide host+IP columns).
- **-c <count>**: Limit auf <count> Probezyklen, dann beenden (für Report).
- **-r**: Report-Modus (nicht curses, druckt Tabelle und endet).
- **-i <sec>**: Sekunden zwischen Probe-Paketen (default 1.0).
- **-p**: Pausiert Ausführung beim Start (Press enter to start).
- **-t**: Text Mode (like curses but in dumb terminal).
- **-u**: UDP mode (Standard mtr uses ICMP Echo, as -l in traceroute terms). -T for TCP mode also available.

Anzeige:

MTR listet Hops fortlaufend wie traceroute, plus Spalten: Loss%, Snt (packets sent), Last (ms last latency), Avg, Best, Wrst (worst), StDev. Es aktualisiert kontinuierlich.

Beispiel-Zeile:

```
3. 203.0.113.5    0.0% 10 15.4 16.1 14.8 20.3 1.3
```

ping (ICMP Echo)

Beschreibung: Schickt ICMP Echo-Request-Pakete an eine Adresse und misst die Antwortzeiten (Echo-Reply), um Erreichbarkeit und Round-Trip-Time festzustellen .

Syntax:

```
ping
```

(Fortsetzung)

```
### ping (ICMP Echo Requests)
```

Beschreibung: `ping` prüft die Erreichbarkeit eines Hosts im Netzwerk mit **ICMP-Echo-Paketen** und misst die Round-Trip-Zeit (RTT) der Antwort . Es sendet fortlaufend Anfragen (standardmäßig, bis abgebrochen) und berechnet Statistik über Verlust und Latenz.

Syntax:

```
``` bash
```

```
ping [Optionen] <Ziel>
```

Hier kann <Ziel> ein Hostname oder eine IP-Adresse sein.

### **Wichtige Optionen:**

- **-c <Anzahl>** – Anzahl der ICMP-Pakete festlegen. Z.B. ping -c 4 8.8.8.8 sendet genau 4 Pakete .
- **-i <Sekunden>** – Interval zwischen den Pings (Standard 1 Sekunde). Man kann z.B. -i 0.2 für fünf Pings pro Sekunde (root-Recht nötig bei <0.2).
- **-W <Timeout>** – Wartezeit in Sekunden auf eine Antwort, bevor als **timeout** gezählt (Default meist 1 Sekunde, bei -c am Ende kann man höher setzen).
- **-q** – "Quiet": keine per-Ping Ausgabe, nur eine Zusammenfassung am Ende.
- **-p <Pattern>** – Sende individuelles Byte-Muster im Paket (Hexadezimal angegeben). Gut für bestimmte Diagnosen (z.B. bestimmte Bits toggeln sehen).
- **-s <Packetsize>** – Größe des zu sendenden Pakets in Bytes (ohne ICMP-Header). Standard 56 Bytes (plus 8 Byte ICMP-Header = 64 Bytes in Ausgabe) .
- **-t <TTL>** – Setzt die Time-To-Live (IPv4) oder Hop Limit (IPv6) für die gesendeten Pakete. (Nützlich, um gezielt nur bis zu einem gewissen Hop zu gelangen, ähnlich traceroute Step).
- **-I <Interface>** – Bindet an ein bestimmtes Quell-Interface (oder IP) für Multi-homed hosts. Alternativ **-S <Source\_Address>**.
- **-4, -6** – Erzwingt IPv4 oder IPv6.
- **-f** – "Flood ping": schickt so schnell wie möglich Pings (sehr viele pro Sekunde) und druckt einen Punkt pro empfangenes Echo. (Nur root, zur Stresstest oder schnelle Packetloss Erkennung).
- **-L** (bei einigen ping-Versionen) – Für Echo an Broadcast/Multicast-Adresse (ermöglicht Broadcast-ping, was standardmäßig blockiert ist).
- **-a** – Bei Empfang einen Terminal-Bell ausgeben (akustisches Signal).
- **-D** – Zeitstempel vor jeder Zeile (UNIX time).
- **-U** – Zeitstempel in Us seit Boot vor jeder Zeile (für Messungen).

### **Ausgabe-Interpretation:**

Jede empfangene Antwort erscheint als Zeile:

```
64 bytes from 8.8.8.8: icmp_seq=1 ttl=117 time=8.34 ms
```

Bedeutet: Echo-Reply von 8.8.8.8 erhalten, Größe 64 Bytes, Sequenznummer 1, TTL im Reply 117, RTT 8.34 ms. Wenn Antworten ausbleiben, erscheinen Zeilen wie "Request timeout".

Am Ende kommt eine **Statistik**:

4 packets transmitted, 4 received, 0% packet loss, time 3005ms

rtt min/avg/max/mdev = 8.334/8.512/8.790/0.177 ms

Zeigt Verlust und RTT-Werte.

**Beispiele:**

- ping 1.1.1.1 – Kontinuierlicher Test zu 1.1.1.1 (Cloudflare DNS). Abbruch mit Strg+C.
- ping -c 5 www.example.com – 5 Pings an example.com, dann stop mit Ausgabe der Statistik.
- ping -l eth1 192.168.2.1 – Ping von Interface eth1 aus (wenn Multi-Interface).
- ping -s 1000 -c 3 host – Sende 3 Pings mit Paketgröße ~1000 Bytes (ICMP Payload).
- ping -6 -c 4 ipv6.google.com – IPv6 Ping zu Google.

**traceroute (siehe oben)**

(bereits behandelt in vorgehendem Teil, siehe „traceroute“ in Netzwerkdagnostik.)

**tracepath (siehe oben)**

(siehe „tracepath“ bei Netzwerkdagnostik.)

**resolvectl (Systemd-resolved Query)**

**Beschreibung:** CLI-Tool zum Abfragen des systemd-resolved-Dienstes (Namensauflösung). resolvectl kann DNS-Lookups durchführen, DNS-Settings anzeigen, LL/MNR Browsing etc. Ersetzt Tools wie systemd-resolve (veraltet).

**Verwendung:**

- **resolvectl status** – Zeigt die aktuellen DNS-Einstellungen pro Link (Interface): DNS-Server, Suche-Domains, LLMNR/MulticastDNS Status, DNSSEC, usw.
- **resolvectl query <Hostname>** – Löst einen Hostnamen auf (A/AAAA-Record). Ausgabe enthält v6 und v4 Adressen, CNAME falls, DNSSEC info, Cache info, antwortender Server etc.  
Beispiel: resolvectl query example.com.

- **resolvectl query -t MX <Domain>** – DNS-Abfrage bestimmten Typs (hier MX-Records). Unterstützt -t ANY, A, AAAA, MX, SRV, SOA etc.
- **resolvectl query <IP>** – Reverse Lookup (IP zu Name via PTR).
- **resolvectl flush-caches** – Löscht den DNS-Cache von systemd-resolved.
- **resolvectl dns <Interface> [Server...]** – Setzt DNS-Server für ein bestimmtes Interface (über DBus). resolvectl domain <iface> ... analog. (Normal wird das via NetworkManager/Netplan/etc gemacht).
- **resolvectl revert <Interface>** – Löscht manuelle Overrides und kehrt zu vom Netzwerkdienst gelieferten DNS zurück.

**Beispiel:**

```
$ resolvectl query -t MX example.com
example.com IN MX 0 .
-- Information acquired via protocol DNS in 15.1ms.
```

-- Data is authenticated: no

Bedeutet example.com hat keinen MX (Mail).

resolvectl status Ausschnitt:

Global

Protocols: LLMNR=resolve mDNS=no DNSOverTLS=no DNSSEC=no/unsupported

resolv.conf mode: stub

Link 2 (eth0)

Current Scopes: DNS

Protocols: +DefaultRoute +LLMNR -mDNS DNSOverTLS=opportunistic DNSSEC=...

Current DNS Server: 192.168.1.1

DNS Servers: 192.168.1.1 1.1.1.1

DNS Domain: ~.

Zeigt z.B. that eth0 hat default DNS 192.168.1.1 plus Cloudflare, und ~. (default domain).

### **dig (DNS Lookup Utility)**

**Beschreibung:** dig (domain information groper) führt DNS-Abfragen aus. Es ist Teil von BIND-Tools (bind-utils). Im Gegensatz zu host oder nslookup bietet dig ausführlich

formatierbare Antworten. Standard macht eine A-Record-Abfrage und gibt Antwort mit Abschnitt **ANSWER SECTION** aus.

### Syntax:

dig [@DNS-Server] <Name> [<Typ>] [<Klasse>] [Optionen]

<Klasse> normalerweise IN (Internet), selten was anderes.

### Wichtige Optionen:

- **@server** – Spezifischen DNS-Server benutzen statt System-resolver. Z.B. dig @8.8.8.8 example.com.
- **<Typ>** – Abfragetyp: A, AAAA, MX, TXT, SRV, NS, SOA, CNAME, PTR, ANY etc. (Default A).
- **+short** – Ausgabe nur die Antworten knapp (einfacher, scripterfreundlich) . Z.B. dig +short example.com A -> "93.184.216.34".
- **+noall +answer** – Unterdrückt alle außer Answer-Section (ähnlich short aber mit TTL und Info).
- **+trace** – Iterativer Trace durch DNS-Hierarchie (Root bis Ziel, ähnlich dig an Root und dann NS referrals folgen). Zeigt jeden Schritt. Hilfreich zum Debug.
- **+tcp** – Erzwinge TCP statt UDP (z.B. bei großen DNS-Records oder EDNS issues).
- **+nssearch** – Findet alle NS des Ziel und fragt diese nach SOA (um Delegation correctness zu checken).
- **-x <IP>** – Reverse lookup (PTR) für IP. (Bequem: dig -x 8.8.8.8).
- **+multi** – Schöne Ausgabe in Spalte falls mehrere TXT oder so.
- **+nodnssec oder +dnssec** – Anfordern oder nicht (um RRSIG etc. zu sehen).
- **+ttlshort** – Output TTL in short mode. (e.g. dig +short +ttl google.com -> "300 172.217.168.14").
- **+stats / +nostats** – Schaltet Statistik-Banner am Ende an/aus.
- **+cm** – Colorize output (if compiled with color support).

### Beispiele:

- dig example.com ANY – Fragt alle Records (die server je nach config liefern).
- dig google.com MX +noall +answer – Zeigt nur MX Antworten.
- dig @1.1.1.1 cloudflare.com A – Abfrage an 1.1.1.1 nach cloudflare's A.
- dig -x 8.8.4.4 +short – Reverse Lookup, kurze Ausgabe ("dns.google.").

- dig +trace example.org – Startet vom Root Nameserver und verfolgt Delegation bis example.org resolved ist.

### **nslookup (Name Server Lookup - veraltet)**

**Beschreibung:** Ein einfaches interaktives DNS-Abfrageprogramm. War Teil von BIND, gilt als veraltet im Vergleich zu dig. Viele Admins kennen es dennoch.

#### **Verwendung:**

nslookup [<Host> [<DNS-Server>]]

Ohne args geht es in interaktiven Modus. Dessen Kommandos: server <DNS>, set q=MX, set type=AAAA, set debug, exit.

#### **Beispiele:**

```
$ nslookup example.com 8.8.8.8
```

Server: 8.8.8.8

Address: 8.8.8.8#53

Non-authoritative answer:

Name: example.com

Address: 93.184.216.34

Reverse: nslookup 93.184.216.34 -> zeigt PTR.

**Warnung:** Neuere doc empfehlen dig oder host statt nslookup.

### **whois (Domain Whois Query)**

**Beschreibung:** Fragt Whois-Datenbanken nach Registrierungsinformationen einer Domain oder 5】. Das Tool verbindet sich zum passenden Whois-Server (verwendet /etc/whois.conf, TLD heuristics) via TCP 43 und gibt den Text aus.

#### **Syntax:**

whois [Optionen] <DomainOrIP>

**Ausgabe:** Je nach TLD und Registry differierend. Enthält Domain-Inhaber (bei Generic oft DSGVO anonymisiert), Registrar, Erstellungs- und Ablaufdatum, Nameserver, Contact-IDs. Bei IP WHOIS (z.B. whois 8.8.8.8) gibt es Informationen über den zugeteilten IP-Block (ISP, RIR wie ARIN/RIPE...).

#### **Optionen:**

- **-H** – Unterdrückt das Banner "Whois Server" in RIPE responses.

- **-p <port>** – Andere Port.
- **--verbose** – Mehr Info über Server selection.
- **--help** – Display help.

### **Beispiele:**

- whois example.com -> zeigt Registrar (ICANN info), Status, Name der Organisation, etc.
- whois 2620:0:2d0:200::7 -> IPv6 whois (gibt i.d.R. ARIN oder so mit OrgName).
- whois AS15169 -> WHOIS an RADb für Autonomous System 15169 (Google).

*(Beachte: Manche TLDs haben separate Tools, z.B. jwhois config. Das Standard whois ist meist in Debian whois package (Marco d'Itri) which covers a lot. GDPR hat Personendaten stark eingeschränkt in Output.)*

**arp (Address Resolution Protocol table) – Hinweis: arp wird mittlerweile durch ip neigh ersetzt.**

**Beschreibung:** Zeigt den ARP-Cache (IPv4) an oder manipuliert Einträge. (Linux modern: use ip neigh aber arp Kommando aus net-tools ist noch gebräuchlich).

### **Befehle:**

- arp -a – Listet ARP-Einträge (Hostname, IP, MAC, Interface).
- arp -n – Wie -a, aber keine DNS für IP.
- arp -d <IP> – Löscht ARP-Eintrag für IP.
- arp -s <IP> <MAC> – Fügt statischen ARP-Eintrag hinzu (MAC fix für IP) – gleich ip neigh add ... nud permanent.

*Beispiel: arp -a -> router.lan (192.168.1.1) at aa:bb:cc:dd:ee:ff [ether] on eth0.*

## **Sicherheit und Firewall**

### **iptables/iptables-nft (Packet Filter)**

**Beschreibung:** Linux Kernel-Paketfilter (Firewall) via Netfilter-Schnittstelle. iptables verwaltet Regeln in verschiedenen Tabellen: **filter, nat, mangle, raw, security**, für IPv4. (Für IPv6, ip6tables). Mit Kernel 4.x/5.x viele Distros nutzen "iptables-nft" as a frontend to nftables, aber Syntax bleibt für user gleich.

### **Syntax Grundkommandos:**

`iptables [-t <table>] <operation> <chain> <matches...> <target>`

Operationen: **-A** (append), **-I** (insert), **-D** (delete), **-C** (check exist), **-L** (list), **-F** (flush chain), **-N** (new chain), **-X** (delete user chain).

Standard Tabellen:

- **filter** (Default): Chains **INPUT, OUTPUT, FORWARD**.
- **nat**: Chains **PREROUTING, OUTPUT, POSTROUTING** (masquerading, port forwarding etc.).
- **mangle**: For packet alteration (TOS, mark), Chains in multiple points (PREROUTING, OUTPUT, etc.).
- **raw**: For conntrack exemption (NOTRACK), Chains PREROUTING, OUTPUT.
- **security**: SELinux related (rare).

**Matches / Options:**

- **-p <proto>**: Protocol (tcp, udp, icmp, icmpv6, all).
- **-s <IP/CIDR>, -d <IP/CIDR>**: Source or destination address (or network).
- **-i <iface>, -o <iface>**: Inbound or outbound network interface (for PREROUTING/INPUT/forward vs OUTPUT/POSTROUTING).
- **--sport, --dport** (with -p tcp/udp): Source/dest port or port range (e.g. --dport 80 or --dport 1000:2000).
- **-m state --state <states>**: State match (connection tracking states: INVALID, ESTABLISHED, NEW, RELATED). Common: allow ESTABLISHED,RELATED on INPUT for responses.
- **-m conntrack --ctstate**: (newer usage, same as state module now).
- **-m connlimit, -m limit** etc: Many modules. limit --limit 5/minute for rate-limit logging for example. connlimit --connlimit-above N limit parallel connections.
- **-m multiport**: match multiple ports in one rule if not contiguous, e.g. -m multiport --dports 80,443,8080.
- **-m tcp --tcp-flags SYN,RST,ACK SYN**: match specific TCP flags combination (here SYN only). Often used for --syn (alias) which matches syn packets (to identify new connections).
- **--comment "text"**: attach comment to rule (for documentation).

**Targets (Actions):**

- **ACCEPT**: Packet erlaubt.
- **DROP**: Verwerfen ohne Antwort.
- **REJECT**: Verwerfen mit Antwort (ICMP port unreachable o.Ä.).

- **LOG**: Loggt Paket (via kernel log), dann Verarbeitung geht weiter (so typ. Kombi LOG dann DROP). Options: --log-prefix "msg", --log-level warning.
- **MASQUERADE** (in nat POSTROUTING): SNAT für dynamische IP (Router Use-case) – ersetzt Source-IP durch Interface-IP.
- **SNAT** (in nat POSTROUTING): Statisches Source-NAT (requires --to-source IP[:port-range]).
- **DNAT** (in nat PREROUTING/OUTPUT): Destination-NAT (Port-Forwarding; --to-destination IP:port).
- **REDIRECT** (nat PREROUTING/OUTPUT): Umleiten an lokalen Host (for transparent proxy etc.).
- **MARK** (mangle): Markiert Paket mit fwmark (für routing decisions via ip rule, etc.).
- **RETURN**: In user-defined chain springt zurück zum Aufrufer-Chain (stop further rules in current chain). In builtin chain, RETURN = apply chain policy.
- **QUEUE**: Übergibt Paket an Userspace (via libnetfilter\_queue).
- **DROP** vs. policy: If chain policy is DROP, falling off end = drop.

### **Beispiele:**

- **List Regeln:** iptables -L -n -v (filter table). iptables -t nat -L -n -v NAT rules. -n no DNS, -v verbose (packet counters).
- **Allow incoming SSH:**

② iptables -A INPUT -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT  
 iptables -A OUTPUT -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT  
 (typical stateful allow SSH in).

② **Drop all forward by default:** iptables -P FORWARD DROP.

② **Masquerading (NAT):**

② iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE  
 (for home router with ppp0 outbound).

② **Port Forward:**

Forward ext eth0 port 80 to internal 192.168.1.100 port 8080:

② iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 \  
 -j DNAT --to-destination 192.168.1.100:8080

```
iptables -A FORWARD -p tcp -d 192.168.1.100 --dport 8080 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
```

#### Logging Drop:

- iptables -A INPUT -p tcp --dport 23 -j LOG --log-prefix "Telnet attempt: " --log-level info
- iptables -A INPUT -p tcp --dport 23 -j DROP
- (log then drop telnet).

### firewall-cmd (firewalld frontend)

**Beschreibung:** Steuerprogramm für **firewalld**, eine dynamische Firewall (standard in RHEL/CentOS 7+, Fedora, etc.). firewall-cmd ermöglicht das Konfigurieren von Zonen, Regeln (services, ports, masq) zur Laufzeit und in Config. Firewalld nutzt intern iptables/nftables.

#### Syntax:

```
firewall-cmd [--zone=<zone>] [--permanent] <aktion> <Parameter>
```

**Grundkonzepte:** Zonen (trusted, public, home, internal, work, external, dmz, block, drop etc.), Schnittstellen sind Zonen zugewiesen. Dienste (vordefinierte Regelsets) können pro Zone erlaubt werden oder einzelne Ports/Protokolle. Man kann Einstellungen permanent (persistiert in XML config) oder nur runtime vornehmen.

#### Wichtige Befehle:

- **--state:** Zeigt an, ob firewalld läuft (running/not running).
- **--get-active-zones:** Zeigt welche Zonen aktiv sind und welche Interfaces zugeordnet (z.B. "public: wlp2s0").
- **--get-zones:** Listet alle bekannten Zonen. **--list-all** [--zone=Z] listet alle Einstellungen einer Zone (services, ports, masquerade etc0】 .
- **--zone=<zone> --add-service=<service>:** Erlaubt den vordefinierten service in angegebener Zone (runtime). Bsp: firewall-cmd --zone=public --add-service=http – ermöglicht HTTP (Port 80) in Zone *public*.
- **--zone=<zone> --add-port=<port>/<proto>:** Erlaubt individuellen Port. Bsp: --add-port=5000/tcp.
- **--remove-service=..., --remove-port=...** analog zum Entfernen.
- **--add-masquerade:** Aktiviert NAT/Masquerading in der Zone (z.B. für Internet-sharing zone).
- **--query-service=<svc>:** Fragt ob Service in zone erlaubt (exit code 0/1).

- **--permanent:** Gibt man diese Option an, wird die Änderung in der persistenten Konfiguration vorgenommen (aber nicht an laufende sofort angewendet, es sei denn man ruft extra an). In firewalld, runtime und permanent config sind getrennt.
- **--reload:** Lädt die permanente Konfiguration neu (in runtime) – im Grunde wendet Config an.
- **--runtime-to-permanent:** Speichert aktuellen Laufzeit-Zustand als neue permanente Konfiguration (was man on-the-fly geändert hat, persistieren).
- **--zone=<zone> --change-interface=<iface>:** Weist ein Interface einer Zone zu (persistente Zuordnung).
- **--new-zone=<name>, --delete-zone=...:** Eigene Zonen erstellen/löschen.
- **--list-services, --get-services:** Listet bekannte Service-Namen (die in /etc/firewalld/services/\*.xml definiert sind).
- **--service=<svc> --get-ports:** Zeigt welche Ports hinter einem Service stecken.
- **--add-rich-rule '<rule>':** Fügt sog. Rich Rule hinzu (eine erweiterte Regel-Syntax in firewalld, z.B. mit Logging, Zeitplänen). Z.B.:
  - `firewall-cmd --add-rich-rule='rule family="ipv4" source address="192.0.2.0/24" port port="22" protocol="tcp" accept'`
  - (Erlaubt SSH nur von 192.0.2.0/24).

### **Beispiele:**

- Öffnen von HTTPS permanent:
  - `firewall-cmd --add-service=https --permanent`
  - `firewall-cmd --reload`
  -
- Port 12345/udp temporär öffnen in Zone public:
  - `firewall-cmd --zone=public --add-port=12345/udp` (nach Reboot weg).
- Interface eth0 zur Zone external zuweisen:
  - `firewall-cmd --zone=external --change-interface=eth0 --permanent && firewall-cmd --reload`.
- Status: `firewall-cmd --list-all` (wenn default zone, z.B. public, dann deren config).

### **ufw (Uncomplicated Firewall)**

**Beschreibung:** Vereinfachtes Firewall-Frontend (Ubuntu/Debian) für iptables. ufw erlaubt einfache allow/deny-Regeln definieren, typischerweise nach Services oder Ports, optional in/out interface.

### Syntax:

ufw [Optionen] <Aktion>

### Grundbefehle:

- **ufw enable / disable** – Aktiviert oder deaktiviert die Firewall (bei disable: setzt iptables default ACCEPT und löscht Regeln).
- **ufw status [verbose]** – Zeigt aktuelle Regel-Liste. Mit verbose sieht man auch Interfaces, logging status etc.
- **ufw default allow|deny [incoming|outgoing|routed]** – Setzt Standardpolicy. Standard-Einstellung: incoming deny, outgoing allow.
- **ufw allow <Port/Service> [proto tcp/udp] [from <IP> [to <IP>]] [port <P2>]** – Erlaubt eingehenden Traffic. Beispiele:
  - ufw allow 22 (erlaubt TCP/22 auf allen Interfaces).
  - ufw allow proto udp from 10.0.0.0/8 to any port 53 (erlaubt UDP DNS von privatem Netz).
  - ufw allow out 80/tcp (erlaubt ausgehende HTTP).
  - ufw allow in on eth0 to any port 443 (erlaubt eingehendes HTTPS auf eth0).
- **ufw deny <Spec>** – Blockiert (mit REJECT) entsprechend.
- **ufw reject <Spec>** – Synonym zu deny (deny und reject meist ähnlich in ufw; auf Debian might differ).
- **ufw limit <Port/Service>** – Wie allow, aber mit Rate-Limit (gegen Bruteforce). Z.B. ufw limit ssh erlaubt SSH, aber blockiert IP temporär bei >6 Verbindungen/30s.
- **ufw delete <RuleSpec>** – Entfernt Regel (man kann Regel genau wie bei allow angeben, mit 'delete' davor, oder Nummer aus ufw status numbered).
- **ufw logging on|off|LEVEL** – Logging ein/aus und Level (off, low, medium, high, full).
- **ufw reset** – Setzt ufw auf Auslieferungszustand (alle Regeln weg, policies reset, und disabled).
- **ufw route allow/deny ...** – Regeln für Forwarded Traffic (Routen). Bsp: ufw route allow in on eth1 out on eth0 to any port 80 for forwarded.

- **ufw insert <num> <rule>** – Fügt Regel an bestimmter Position ein.
- **ufw show raw** – Zeigt resultierende iptables-Regeln.

### **Beispiele:**

- Standard: ufw default deny incoming; ufw default allow outgoing; ufw enable.
- Webserver: ufw allow 'Nginx Full' (in /etc/ufw/applications.d vordefinierte Profile, 'Apache Full' etc., Nginx Full = 80+443).
- Only LAN access to a service: ufw allow from 192.168.0.0/16 to any port 3306 (MySQL offen nur intern).
- Block eine IP: ufw deny from 203.0.113.5.
- Show rules: ufw status verbose -> e.g.
- Status: active
- Logging: on (low)
- Default: deny (incoming), allow (outgoing), deny (routed)
- New profiles: skip
- 
- To Action From
- -- ----- -----
- 22/tcp LIMIT Anywhere # SSH
- 80,443/tcp ALLOW Anywhere # WWW
- 22/tcp (v6) LIMIT Anywhere (v6)
- 80,443/tcp (v6) ALLOW Anywhere (v6)
- 

*(UFW ist beliebt wegen einfacher Syntax, aber fehlender Komplexität bei komplexen Setups. Firewalld und raw iptables/nft sind mächtiger in Feinheiten.)*

### **nft (nftables Firewall)**

**Beschreibung:** nftables ist der modernere Packet-Filter (ersetzt iptables). Der Befehl nft wird zum Konfigurieren verwendet. Sehr flexibel (Datenstrukturen, one-rule multiple matches).

### **Syntax:**

nft [options] <command>

Man kann Kommandos interactive eingeben oder als Skript. Der Kommandostil ähnelt C:  
e.g.

```
nft add table inet filter
```

```
nft add chain inet filter input { type filter hook input priority 0 \; }
```

```
nft add rule inet filter input tcp dport 22 accept
```

(Librally, but given complexity and that the original question references iptables and ufw, likely beyond scope).

## Abschluss

Dieses Kompendium deckt die wesentlichen Befehle für die Linux+ (XK0-005) Prüfung ab, gegliedert nach Themengebieten. Die Befehle umfassen Benutzer- und Rechteverwaltung, Prozess- und System-Tools, Netzwerk-Konfiguration und -Diagnose, Software-Installation, Container, Sicherheit (Firewall, SELinux/AppArmor siehe oben Rich-Rules, etc.), Dienste und Planung.

Durch die strukturierte Darstellung mit Syntax, Beschreibung, Optionen und Beispielen dient es als umfassendes Nachschlagewerk. **Wichtig:** In einer Prüfungssituation sollte man sich auf die korrekte Anwendung der Optionen und das Verständnis der Konzepte dahinter konzentrieren (z.B. was macht chmod 775 oder umask 002 bewirkt).

Mit dieser Referenz sollte die Vorbereitung auf typische Linux-Aufgaben – vom Anlegen eines Benutzers, dem Ändern von Dateirechten, über das Einrichten eines Cronjobs bis zum Konfigurieren eines Apache-Dienstes oder Setzen einer Firewallregel – bestens unterstützt werden.

## Kompendium der Linux+ (XK0-005) Befehle

Dieses Nachschlagewerk umfasst alle wichtigen Befehle aus dem offiziellen **Command Reference Guide** für die CompTIA Linux+ Prüfung (XK0-005). Die Befehle sind thematisch gegliedert (Benutzerverwaltung, Netzwerk, Prozesse, Dateisystem, Paketmanagement, Container, Sicherheit etc.), mit Beschreibung, vollständiger Syntax, allen verfügbaren Optionen und ggf. wichtigen Unterbefehlen. Es dient als umfassendes Cheat-Sheet zur Prüfungsvorbereitung.

## Grundlegende Linux-Befehle

### ls (List Directory Contents)

**Beschreibung:** Listet den Inhalt eines Verzeichnisses auf. Ohne Parameter zeigt ls die nicht versteckten Einträge im aktuellen Verzeichnis in alphabetischer Reihenfolge [wiki.ubuntuusers.de/ionos.de](http://wiki.ubuntuusers.de/ionos.de).

#### Syntax:

```
ls [OPTION]... [VERZEICHNIS]
```

*Wenn kein Verzeichnis angegeben wird, wird das aktuelle Verzeichnis aufgelistet*[wiki.ubuntuusers.de](http://wiki.ubuntuusers.de).

### Optionen:

Option	Beschreibung
<b>-a, --all</b>	Zeigt alle Einträge an, inklusive versteckter Dateien (auch . und ..) <a href="http://wiki.ubuntuusers.deionos.de">wiki.ubuntuusers.deionos.de</a> .
<b>-A, --almost-all</b>	Wie -a, aber ohne . und .. – listet versteckte Dateien außer den Verzeichnissen ./.. <a href="http://wiki.ubuntuusers.deionos.de">wiki.ubuntuusers.deionos.de</a> .
<b>-l</b>	Langformat: Detaillierte Ausgabe (Rechte, Besitzer, Größe, Datum usw.) je Eintrag <a href="http://wiki.ubuntuusers.deionos.de">wiki.ubuntuusers.deionos.de</a> .
<b>-d, --directory</b>	Listet Verzeichniseintrag selbst statt dessen Inhalt (nützlich, um Infos über Ordner zu sehen) <a href="http://wiki.ubuntuusers.deionos.de">wiki.ubuntuusers.deionos.de</a> .
<b>-h, --human-readable</b>	Im Langformat (-l) Größen in menschenlesbare Einheiten (KB, MB ...) anzeigen <a href="http://wiki.ubuntuusers.deionos.de">wiki.ubuntuusers.deionos.de</a> .
<b>-i</b>	Zeigt die Inode-Nummer jedes Eintrags an <a href="http://wiki.ubuntuusers.deionos.de">wiki.ubuntuusers.deionos.de</a> .
<b>-C</b>	Ausgabe in Spalten nebeneinander (Standardformat) <a href="http://wiki.ubuntuusers.deionos.de">wiki.ubuntuusers.deionos.de</a> .
<b>-m</b>	Listet Einträge kommasepariert in einer Zeile <a href="http://wiki.ubuntuusers.deionos.de">wiki.ubuntuusers.deionos.de</a> .
<b>-R, --recursive</b>	Rekursiv: Zeigt auch Inhalte von Unterverzeichnissen (durchläuft Verzeichnisbaum) <a href="http://wiki.ubuntuusers.deionos.de">wiki.ubuntuusers.deionos.de</a> .
<b>-t</b>	Sortiert nach Zeitstempel (Änderungszeit, neueste zuerst) anstatt alphabetisch.
<b>-r</b>	Kehrt die Sortierreihenfolge um (Reverse order).
<b>--hyperlink</b>	Macht Dateinamen (in unterstützten Terminals) als klickbare Links ausgebbar <a href="http://wiki.ubuntuusers.de">wiki.ubuntuusers.de</a> .
<b>--help</b>	Zeigt die Hilfe zum Befehl an.
<b>--version</b>	Zeigt Versionsinformationen von ls.

*(Weitere Optionen sind in den Manpages dokumentiert)*[wiki.ubuntuusers.de](http://wiki.ubuntuusers.de).

### cat (Concatenate and Print Files)

**Beschreibung:** Gibt den Inhalt von Dateien auf der Standardausgabe aus oder verknüpft (konkateniert) mehrere Dateien. Oft verwendet, um Textdateien anzuzeigen oder Inhalte zu pipen[ionos.at](#). Der Name steht für „concatenate“ (verketten).

**Syntax:**

cat [OPTION]... [DATEI]...

*Ohne Optionen schreibt cat den Inhalt der angegebenen Datei(en) in der angegebenen Reihenfolge auf die Standardausgabe. Kann mehrere Dateien hintereinander ausgeben oder mit Umleitungen/ Pipes genutzt werden.*

**Optionen:**

**Option Beschreibung**

- n** Nummeriert *alle* ausgegebenen Zeilen fortlaufend[ibm.comionos.at](#).
- b** Nummeriert nur **nicht-leere** Ausgabelinien (Leerzeilen werden nicht nummeriert)[ibm.comionos.at](#).
- s** Squeeze: Unterdrückt aufeinanderfolgende Leerzeilen (mehrere leere Zeilen werden als eine ausgegeben)[ionos.at](#).
- v** “Visible” – Macht nicht druckbare Steuerzeichen sichtbar (außer Tab und Zeilenende)[ionos.at](#).
- e** Entspricht -v mit Anzeige des Zeilenende-Zeichens \$ am Zeilenende (kombiniert -v und Kennzeichnung von Zeilenumbrüchen)[ionos.at](#).
- t** Entspricht -v mit Darstellung von Tabulatoren als ^I (Tab-Markierung)[ionos.at](#).
- E** Zeigt \$ am Ende jeder Zeile (Äquivalent zu -vE, oft identisch zu -e je nach Implementierung).
- T** Zeigt Tabulatoren als ^I an (ähnlich -vT, meist wie -t).
- A** Kombiniert mehrere der obigen: entspricht oft -vET (alle nicht-druckbaren Zeichen, Tabs und Zeilenenden sichtbar machen).
- u** (Ignoriert in GNU cat – historisch für ungepufferte Ausgabe).
- help** Zeigt Hilfe-Informationen an.
- version** Zeigt Versionsinfo an.

**cd (Change Directory)**

**Beschreibung:** Wechselt in ein anderes Verzeichnis. Mit cd kann das aktuelle Arbeitsverzeichnis geändert werden („change directory“)[ionos.at](#). Ohne Argument wechselt cd standardmäßig ins Heimatverzeichnis des Benutzers.

**Syntax:**

cd [OPTION] [VERZEICHNIS]

*Man kann relative Pfade (bezogen auf das aktuelle Verzeichnis) oder absolute Pfade (ausgehend vom Wurzelverzeichnis /) angeben. Erfolgt der Wechsel in ein Zielverzeichnis, so wird dieses das neue aktuelle Arbeitsverzeichnis; bei nicht vorhandenen Zielen oder fehlenden Rechten bleibt das Verzeichnis unverändert[ionos.at](#).*

**Optionen:**

**Option Beschreibung**

- L      Folge symbolischen Links **logisch**. Dies ist die Voreinstellung. (Beim Navigieren durch einen Symlink bleibt man in der logischen Pfadstruktur)[ionos.at](#).
- P      Folge symbolischen Links **physisch**. Wechselt in das tatsächliche Ziel im Dateisystem und löst symbolische Links auf[ionos.at](#).

**Unterbefehle / spezielle Aufrufe:**

- cd ~ wechselt ins Heimatverzeichnis des aktuellen Benutzers[ionos.at](#).
- cd - wechselt in das vorherige Arbeitsverzeichnis zurück (Toggle zwischen zwei Verzeichnissen)[ionos.at](#).
- cd /Pfad/zu/Verz wechselt absolut in ein angegebenes Verzeichnis vom Wurzelpfad aus (hier z.B. nach /Pfad/zu/Verz).
- cd .. wechselt eine Ebene höher (ins Parent-Verzeichnis), cd ../../ entspricht zwei Ebenen usw.
- cd . bleibt im aktuellen Verzeichnis (. bezeichnet das aktuelle Verzeichnis, meistens kein Effekt).

**pwd (Print Working Directory)**

**Beschreibung:** Gibt den vollständigen Pfad des aktuellen Arbeitsverzeichnisses aus[ionos.at](#). Mit pwd kann man sich anzeigen lassen, *in welchem Verzeichnis man sich gerade befindet*.

**Syntax:**

pwd [OPTION]...

**Optionen:**

## **Option**   **Beschreibung**

- Ausgabe des Pfads mit Berücksichtigung symbolischer Links (logischer Pfad, Standard). Wenn z.B. ein symbolischer Link als aktuelles Verzeichnis dient, wird der Linkpfad angezeigt[ionos.de](#).
- P** Ausgabe des *physischen* Pfads: Symbolische Links werden aufgelöst, der tatsächliche Verzeichnispfad wird angezeigt[ionos.de](#).
- help** Zeigt Hilfe-Informationen zu pwd an.
- version** Zeigt Versionsinformationen an.

*Hinweis:* pwd ist häufig als Shell-Builtin implementiert. Das externe Programm /bin/pwd unterstützt ggf. andere oder weniger Optionen (bei GNU Coreutils nur --help und --version zusätzlich)[ionos.de](#). In der Regel genügt der Shell-eigene pwd Befehl.

## **whoami**

**Beschreibung:** Zeigt den Benutzernamen des aktuell angemeldeten Users anfile-4fdk7rvx39azjretinarqb. Praktisch, um schnell zu prüfen, unter welchem Account ein Prozess läuft.

### **Syntax:**

whoami

*Keine spezifischen Optionen.* Der Befehl gibt äquivalent echo \$USER aus (den Namen des aktuellen Unix-Accounts).

## **touch**

**Beschreibung:** Erzeugt eine neue leere Datei oder ändert Zeitstempel (Zugriffs- und Änderungszeit) bestehender Dateien. Wenn die Datei nicht existiert, wird sie (sofern nicht -c verwendet wird) als leere Datei angelegt[man7.org](#). Oft genutzt, um Dateien schnell anzulegen oder deren *timestamp* zu aktualisieren.

### **Syntax:**

touch [OPTION]... DATEI...

### **Optionen:**

<b>Option</b>	<b>Beschreibung</b>
<b>-a</b>	Ändert nur die Zugriffszeit (atime) der Datei <a href="#">man7.org</a> .
<b>-m</b>	Ändert nur die Änderungszeit (mtime) der Datei <a href="#">man7.org</a> .

Option	Beschreibung
<b>-c, --no-create</b>	Legt keine neue Datei an. Wenn die Datei nicht existiert, wird kein neues leeres File erstellt (ansonsten Standardverhalten ohne -c) <a href="#">man7.org</a> .
<b>-d, --date=STRING</b>	Verwendet den angegebenen Datums-String anstelle der aktuellen Zeit. Der String wird geparsst (z.B. "2025-05-01 12:00:00" oder relative Angaben wie "next Friday" sind möglich) <a href="#">man7.org</a> .
<b>-t</b> [[CC]YY]MMDDhhmm[.ss]	Verwendet einen konkreten Zeitstempel im angegebenen Format (JJMMTThhmm[.ss]) anstatt der aktuellen Zeit <a href="#">man7.org</a> . Beispiel: -t 202312312359.59 setzt 31.12.2023 23:59:59.
<b>-r, --reference=DATEI</b>	Übernimmt die Zeitstempel (Zugriffs- und Änderungszeit) der angegebenen Referenzdatei. Die Zielfile erhält also dieselben Zeiten wie DATEI <a href="#">man7.org</a> .
<b>-h, --no-dereference</b>	Ändert bei symbolischen Links <i>den Link selbst</i> (deren Metadaten), nicht die referenzierte Datei. Ohne diese Option würde touch auf den Ziel-Dateipfad wirken (nicht auf den Symlink). ( <i>Diese Option ist nur wirksam, wenn das System das Verändern von Symlink-Zeitstempeln unterstützt.</i> ) <a href="#">man7.org</a>
<b>--time=WORD</b>	Bestimmt, welcher Zeitstempel geändert wird. Mögliche Werte: atime, access, use (entspricht -a), oder mtime, modify (entspricht -m) <a href="#">man7.org</a> .
<b>--help, --version</b>	Anzeigen der Hilfe bzw. Versionsinformation zu touch.

## man (Manual Page Viewer)

**Beschreibung:** Zeigt die Handbuchseite (Manual Page) für einen angegebenen Befehl oder ein Thema an. man ist das Offline-Hilfe- und Dokumentationssystem unter Unix/Linux.

### Syntax:

man [SECTION] <Thema/Command>

*Beispiel:* man ls zeigt die Manpage zu **ls**. Manpages sind meist nach Kategorien/Abschnitten (1=Benutzerkommandos, 5=Dateiformate, 8=Administrator-Kommandos etc.) gegliedert – optional kann man mit man 5 passwd z.B. direkt die Manpage zum Dateiformat **passwd** (Abschnitt 5) öffnen.

## Optionen:

Option	Beschreibung
<b>-k</b> <i>Suchwort</i>	Durchsucht die Manpage-Kurzzusammenfassungen nach <i>Suchwort</i> (entspricht dem Befehl apropos). Beispiel: man -k network listet alle Manual-Topics auf, deren Beschreibung "network" enthält.
<b>-f Befehl</b>	Findet den Namen und Abschnitt einer Manpage zu <i>Befehl</i> (entspricht whatis). Beispiel: man -f passwd zeigt kurz "passwd (1) - change user password".
<b>-a</b>	Zeigt nacheinander alle Manpages an, die zum angegebenen Namen passen (wenn z.B. ein Name in mehreren Abschnitten existiert).
<b>-P Pager</b>	Legt ein anderes Pager-Programm fest (Standard ist less). Beispiel: man -P cat ls würde die ls-Manpage durch cat leiten.
<b>--help, --version</b>	Zeigt Hilfe bzw. Version von man.

## whatis

**Beschreibung:** Zeigt eine kurze Ein-Satz-Beschreibung zu einem Befehl an (frägt die Datenbank der whatis-Einträge ab). Es ist eine Kurzform von man -f und liefert die Synopsis aus der Manpage.

### Syntax:

whatis <Befehl/Topic>

## Optionen:

Option	Beschreibung
<i>(keine üblichen Optionen)</i>	whatis wird meist ohne Optionen verwendet. Es greift auf die gleichen Datenbanken wie apropos/man -k zu. Sollte die Datenbank nicht aktuell sein, kann sie mit mandb erneuert werden.

## Benutzer- und Gruppenverwaltung

### passwd (Passwort ändern/setzen)

**Beschreibung:** Ändert das Passwort eines Benutzerkontos oder verwaltet Konto-Sperren. Als normaler Benutzer kann man mit passwd sein eigenes Passwort ändern; als root kann man mit passwd <user> das Passwort eines anderen Accounts setzen oder bestimmte Eigenschaften (Sperren, Ablaufdatum) ändern.

### Syntax:

`passwd [OPTION] [BENUTZER]`

*Standardaufruf ohne Benutzer wirkt auf den aktuellen Benutzer. Bei Angabe eines Benutzernamens werden dessen Passwortinformationen geändert (Root-Rechte erforderlich).*

**Optionen:**

**Option Beschreibung**

- Sperrt das Benutzerkonto, indem das Passwort in /etc/shadow mit einem **-l, --lock** Präfix (meist ! oder \*) unbrauchbar gemacht wird[wiki.ubuntuusers.de](http://wiki.ubuntuusers.de). Der Benutzer kann sich danach nicht mehr anmelden, bis das Konto wieder entsperrt wird.
- Entsperrt ein zuvor gesperrtes Konto (entfernt das Sperr-Präfix im **-u, --unlock** verschlüsselten Passwort)[wiki.ubuntuusers.de](http://wiki.ubuntuusers.de).
- Löscht das Passwort des Benutzers (leer setzen). Achtung: Damit wird eine **-d, --delete** passwortlose Anmeldung möglich, sofern nicht andersweitig gesichert[wiki.ubuntuusers.de](http://wiki.ubuntuusers.de).
- Markiert das Passwort als abgelaufen. Beim nächsten Login muss der **-e, --expire** Benutzer ein neues Passwort setzen.
- Setzt die maximale Gültigkeitsdauer eines Passworts auf *TAGE*. Nach Ablauf **-x TAGE** muss es geändert werden. (Änderung des PASS\_MAX\_DAYS in /etc/shadow.)
- Setzt die minimale Anzahl an Tagen zwischen zwei Passwortänderungen **-n TAGE** (PASS\_MIN\_DAYS). Der Benutzer kann sein Passwort erst nach *TAGE* Tagen erneut ändern.
- Warnzeit in Tagen vor Passwortablauf (PASS\_WARN\_AGE). Der Benutzer erhält *TAGE* Tage vor Ablauf eine Warnung beim Login.
- Zeigt den Passwortstatus an (Gesperrt? nicht gesetzt? Datum der letzten Änderung usw.). `passwd -S <user>` gibt eine Statuszeile aus.

*Interaktives Verhalten:* Wird passwd ohne Optionen aufgerufen, fordert es zur Eingabe des aktuellen Passworts (zur Authentifizierung, falls nicht root) und zweimal zur Eingabe des neuen Passworts auf (zwecks Bestätigung). Es verweigert einfache Passwörter gemäß der System-Policy (Pluggable Authentication Modules/pam Richtlinien). Root-Benutzer muss das alte Passwort nicht kennen.

**chage (Change Age – Passwortalter und Gültigkeit)**

**Beschreibung:** Verändert Einstellungen zum Passwort-Aging – also Ablaufdatum, Mindestalter, Warnzeit etc. Mit chage kann ein Administrator festlegen, wann ein Benutzer sein Passwort ändern muss bzw. wann der Account deaktiviert wirdfile-4fdk7rvx39azretinarqb.

**Syntax:**

chage [OPTION]... BENUTZER

**Optionen:**

**Option Beschreibung**

**-l** Listet die aktuellen Passwort- und Alterseinstellungen für den Benutzer in einem lesbaren Format auf (Letzte Änderung, Min/Max Alter, Warnzeit, Inaktivzeit, Ablaufdatum) – Übersicht.

**-m TAGE** Setzt die minimale Anzahl von Tagen *TAGE*, die ein Passwort gültig sein muss, bevor es geändert werden darf (PASS\_MIN\_DAYS).

**-M TAGE** Setzt die maximale Gültigkeitsdauer in Tagen (PASS\_MAX\_DAYS). Nach *TAGE* Tagen läuft das Passwort ab und muss geändert werden.

**-w TAGE** Setzt die Anzahl der Tage *TAGE* Vorwarnzeit vor Passwortablauf (PASS\_WARN\_AGE).

**-I TAGE** Legt fest, wie viele Tage nach Passwortablauf der Account **inaktiv** wird (PASS\_INACTIVE\_DAYS). Ein Wert von 0 deaktiviert den Account sofort nach Ablauf des Passworts, -1 schaltet diese Funktion aus[wiki.ubuntuusers.de](http://wiki.ubuntuusers.de/wiki.ubuntuusers.de).

**-E DATUM** Setzt ein Ablaufdatum für das Benutzerkonto (im Format YYYY-MM-DD). Nach diesem Datum wird der Account deaktiviert[wiki.ubuntuusers.de](http://wiki.ubuntuusers.de). Das Datum -1 (oder "" leer) entfernt ein gesetztes Ablaufdatum (kein Ablauf).

**-d YYYY-MM-DD** Setzt das Datum der letzten Passwortänderung auf das angegebene Datum (wirkt wie Backdating oder um zu triggern, dass ein Passwort abläuft). Kann auf 0 gesetzt werden, um das Passwort als sofort ablaufend zu markieren.

**-h, --help** Zeigt Hilfe an.

**Hinweis:** Viele dieser Einstellungen lassen sich auch mit passwd (Optionen -x, -n, -w, -i, -e) vornehmen. chage bietet eine dedizierte Schnittstelle dafür. Ohne Optionen (als root) startet chage <user> interaktiv einen Dialog, um alle Werte einzustellen.

**w**

**Beschreibung:** Zeigt an, welche Benutzer gerade am System angemeldet sind und was sie gerade tun. w fasst Informationen über aktuell eingeloggte User und deren laufende Prozesse zusammen (inkl. Login-TTY, Leerlaufzeit, genutzte CPU-Zeit, aktueller Befehl)file-4fdk7rvx39azjretinarqb.

**Syntax:**

w [OPTION]

**Optionen:**

**Option    Beschreibung**

- h        Blendet die Kopfzeile in der Ausgabe aus (unterdrückt die Überschriftenzeile).
- s        Kurze Ausgabe. Unterdrückt die Spalten für Login-Zeit, Leerlauf etc. (nicht auf allen Systemen vorhanden).
- f        Zeigt Informationen zu von utmp als *geflagt* markierten Einträgen. Auf einigen Systemen steuert dies die Anzeige von Remote-Host oder Exit-Informationen. (Bei neueren procps-Versionen hat -f keine Funktion, da Host immer gezeigt wird.)
- V, --version    Zeigt Versionsinfo von w.

*Die Ausgabe von w enthält in der ersten Zeile System-Uptime und Load Average. Danach folgen pro eingeloggtem Benutzer: Benutzername, Terminal, Remote-Host, Login-Zeit, Leerlaufdauer, CPU-Verbrauch und der aktuell laufende Prozess (TTY-Prozess).*

**who**

**Beschreibung:** Zeigt Informationen über die aktuell angemeldeten Benutzer. Im Gegensatz zu w liefert who primär Login-Zeitpunkte und Terminalsessions, jedoch keine Prozess- oder Idle-Infos.

**Syntax:**

who [OPTION]... [Datei]

*Standard liest who die System-Login-Datenbank /var/run/utmp und gibt für jeden Eintrag Benutzer, Terminal, Datum/Zeit und ggf. Host aus.*

**Optionen:**

**Option    Beschreibung**

- H        Zeigt eine Kopfzeile (Überschriften) über den Spalten an.

<b>Option</b>	<b>Beschreibung</b>
<b>-q</b>	"Quick who": Zeigt nur Benutzernamen und die Anzahl der angemeldeten Nutzer. (Gleichbedeutend mit users-Befehl.)
<b>-b</b>	Zeigt die Zeit des letzten Systemstarts (Boot time) an.
<b>-r</b>	Zeigt den aktuellen Runlevel des Systems.
<b>-a</b>	Zeigt <i>alle</i> Einträge aus utmp, inklusive Einträgen für laufende Prozesse am System (wie getty oder systemd-Spawns). Kombination von mehrere Flags: entspricht oft -b -d --login -p -r -t -T -u je nach Implementierung.
<b>-m</b>	Zeigt nur die Einträge für das aktuelle Terminal an (beschränkt auf den Benutzer, der den Befehl aufruft).
<b>-u</b>	Zeigt zusätzlich die Leerlaufzeit (Idle) und den PID des Login-Prozesses für jeden Benutzer.
<b>-T</b>	Zeigt den Terminal-Status ( $\pm$ Flag) an: + bedeutet terminal ist bei mesg schreibbar, - bedeutet nicht schreibbar, ? unbekannt. (In vielen who-Implementierungen bereits Standardverhalten ohne Option.)
<b>--help, --version</b>	Anzeigen von Hilfe bzw. Versionsinfo.

### **useradd (Benutzer anlegen)**

**Beschreibung:** Legt ein neues Benutzerkonto an. useradd ist ein Low-Level-Befehl, der die Systemkontodateien (wie /etc/passwd, /etc/shadow, /etc/group) direkt verändert, um einen neuen User einzurichten [pq.hosting](#). In Debian/Ubuntu-Umgebungen wird oft das Wrapper-Skript adduser verwendet, das auf useradd aufbaut und interaktiv ein paar Dinge abfragt; useradd selbst ist nicht interaktiv und benötigt alle Optionen zum Festlegen der Eigenschaften des neuen Accounts.

#### **Syntax:**

useradd [OPTION]... BENUTZERNAME

**Optionen:** (häufigste Optionen beim Anlegen eines Benutzers)

<b>Option</b>	<b>Beschreibung</b>
<b>-c</b>	Kommentarfeld (GECOS) für den Benutzer, z.B. voller Name oder Hinweis. Kommentar Wird in /etc/passwd eingetragen <a href="#">wiki.ubuntuusers.de</a> .

<b>Option</b>	<b>Beschreibung</b>
<b>-d HOME</b>	Pfad des Home-Verzeichnisses. Standard: /home/<Benutzername>. Kann geändert werden; wenn -m verwendet wird, wird dieses Verzeichnis erstellt und mit Standard-Dateien aus /etc/skel befüllt <a href="#">wiki.ubuntuusers.de</a> .
<b>-m</b>	Legt das Home-Verzeichnis an und kopiert Standard-Dateien hinein. Sollte gesetzt sein, außer man möchte <i>kein</i> Home-Verz. (Ohne -m wird das Home nicht automatisch erstellt) <a href="#">wiki.ubuntuusers.de</a> .
<b>-s SHELL</b>	Login-Shell des Benutzers. Standard ist normalerweise /bin/bash (oder was in /etc/default/useradd als SHELL definiert ist). Kann z.B. auf /sbin/nologin gesetzt werden für System-Accounts <a href="#">wiki.ubuntuusers.de</a> .
<b>-u UID</b>	Gewünschte User-ID (numerisch). Darf nicht bereits vergeben sein (außer mit -o). Wenn nicht angegeben, wird die nächste freie UID > 999 gewählt <a href="#">wiki.ubuntuusers.de</a> .
<b>-o</b>	Erlaubt die Verwendung einer bereits vergebenen UID (UID-Doppelungen). Zusammen mit -u benutzt, um zwei Benutzer mit identischer UID anzulegen. (Normalerweise nicht empfohlen, da UID eindeutig sein sollen.)
<b>-g GRP</b>	Primäre Gruppe (Name oder GID), der der Benutzer angehört <a href="#">wiki.ubuntuusers.de</a> . Standard: eine neue Gruppe mit gleichem Namen wie der Benutzer (bei entsprechender Systemkonfiguration) oder eine in /etc/default/useradd definierte GROUP. Die angegebene Gruppe muss existieren.
<b>-G Liste</b>	Liste zusätzlicher Gruppen (sekundäre Gruppen), denen der Benutzer angehören soll <a href="#">wiki.ubuntuusers.de</a> . Gruppennamen durch Komma getrennt angeben (ohne Leerzeichen).
<b>-p Passwort</b>	Das verschlüsselte Passwort für den Account (wie es z.B. von openssl passwd oder crypt(3) erzeugt würde) <a href="#">wiki.ubuntuusers.de</a> . <b>Wichtig:</b> Aus Sicherheitsgründen sollte man hier <i>nicht</i> ein Klartextpasswort übergeben. Bleibt -p leer, wird der Account ohne gesetztes Passwort angelegt (Anmeldung evtl. gesperrt bis passwd gesetzt wird).
<b>-e</b>	(bei useradd nicht üblich – wird bei usermod benutzt für Ablauf setzen. Bei einigen Systemen könnte useradd -e <Datum> ein Ablaufdatum setzen.)

<b>Option</b>	<b>Beschreibung</b>
<b>-f TAGE</b>	Anzahl an Tagen nach Passwortablauf, bis der Account deaktiviert wird (Analog zu chage -l). -f 0 würde den Account sofort mit Passwortablauf sperren. Standard -1 (nie sperren).
<b>-N</b>	(Bei einigen Systemen: <i>No user group</i> – verhindert das automatische Anlegen einer gleichnamigen Gruppe.)
<b>-r</b>	Erstellt einen System-Account (UID unter einer gewissen Schwelle, z.B. <1000, kein Expire, kein Mail-Spool). Oft genutzt für Systembenutzer ohne Login.
<b>-M</b>	Kein Home-Verzeichnis anlegen (selbst wenn Default CREATE_HOME aktiviert ist).
<b>-k SKELDIR</b>	Gibt ein alternatives Skeleton-Verzeichnis an (anstatt /etc/skel), von wo aus bei -m die initialen Dateien ins Home kopiert werden.
<b>-D</b>	Zeigt die aktuellen Standardwerte für useradd an (oder mit anderen -D Optionen können diese verändert werden).

*Nach dem Anlegen mit useradd sollte meist ein Passwort gesetzt werden (passwd <user>), außer es handelt sich um einen Account ohne Login.*

### **usermod (Benutzerkonto ändern)**

**Beschreibung:** Modifiziert ein bestehendes Benutzerkonto – z.B. Name, UID, Primärgruppe, zusätzliche Gruppen, Login-Shell, Expire-Datum usw.[wiki.ubuntuusers.de](http://wiki.ubuntuusers.de/wiki.ubuntuusers.de). Im Grunde die gleichen Optionen wie useradd, aber angewendet auf bestehende Benutzer. Häufig genutzt, um Benutzer umzubenennen, in andere Gruppen aufzunehmen oder Accounts zu sperren.

#### **Syntax:**

usermod [OPTION]... LOGIN

#### **Optionen:**

<b>Option</b>	<b>Beschreibung</b>
<b>-c Kommentar</b>	Neuen Kommentar (GECOS-Feld) setzen <a href="http://wiki.ubuntuusers.de">wiki.ubuntuusers.de</a> .
<b>-d Neues_Home</b>	Ändert das Home-Verzeichnis des Benutzers auf <i>Neues_Home</i> . <b>Hinweis:</b> Ohne weiteres passiert das alte Home nicht automatisch mit. Mit zusätzlicher Option <b>-m</b> werden die Inhalte des alten Home-Verz. ins neue Verzeichnis verschoben und das neue Verzeichnis ggf. erstellt <a href="http://wiki.ubuntuusers.de">wiki.ubuntuusers.de</a> .

Option	Beschreibung
<b>-m</b>	(Mit -d verwenden) Verschiebt beim Ändern des Homepfads (-d) die Dateien vom alten ins neue Home und erstellt es falls nötig <a href="#">wiki.ubuntuusers.de</a> .
<b>-l Neuer_Login</b>	Ändert den Login-Namen des Benutzers (Umbenennung des Accounts) <a href="#">wiki.ubuntuusers.de</a> . Der Benutzername in allen Systemkontodateien wird angepasst, <i>nicht</i> jedoch automatisch der Name des Home-Verzeichnisses oder Mail-Spool (diese müssten manuell umbenannt werden, falls gewünscht).
<b>-u Neue_UID</b>	Ändert die numerische Benutzer-ID auf <i>Neue_UID</i> . Falls die UID bereits vergeben ist, muss -o benutzt werden. Dateien im Home-Verzeichnis mit alter UID werden auf neue UID umgestellt; Dateien außerhalb des Homes muss der Admin selbst anpassen <a href="#">wiki.ubuntuusers.dewiki.ubuntuusers.de</a> .
<b>-o</b>	Erlaubt die Verwendung einer bereits vergebenen UID (nur zusammen mit -u sinnvoll).
<b>-g Neue_PrimGruppe</b>	Ändert die primäre Gruppenzugehörigkeit (Name oder GID) <a href="#">manpages.ubuntu.com</a> . Die neue Gruppe muss existieren. Alle Dateien im Home, die der alten primären Gruppe gehörten, werden auf die neue Gruppe übertragen <a href="#">manpages.ubuntu.com</a> . (Dateien außerhalb des Homes müssen ggf. manuell angepasst werden.)
<b>-G Gruppe(n)</b>	Setzt die Liste der sekundären Gruppen neu für den Benutzer <a href="#">manpages.ubuntu.com</a> . <i>Wichtig:</i> Ohne -a entfernt diese Option den Benutzer aus allen Gruppen, die nicht in der Liste stehen <a href="#">wiki.ubuntuusers.demanpages.ubuntu.com</a> . Um einen Benutzer zusätzlichen Gruppen hinzuzufügen, sollte man -a -G gemeinsam verwenden (siehe -a).
<b>-a</b>	<i>Append</i> – fügt den Benutzer zusätzlichen Gruppen hinzu, anstatt die Gruppenzugehörigkeit zu überschreiben <a href="#">wiki.ubuntuusers.dewiki.ubuntuusers.de</a> . <b>Nur in Kombination mit -G verwenden</b> , um den Benutzer der/die angegebenen Gruppen hinzuzufügen, ohne ihn aus anderen zu entfernen. (Siehe Warnung unten.)
<b>-s Shell</b>	Ändert die Login-Shell des Benutzers <a href="#">wiki.ubuntuusers.de</a> . Z.B. auf /sbin/nologin setzen, um interaktive Logins zu unterbinden.

<b>Option</b>	<b>Beschreibung</b>
<b>-e YYYY-MM-DD</b>	Setzt das Ablaufdatum des Accounts (das Datum, an dem das Konto deaktiviert wird) <a href="#">wiki.ubuntuusers.de</a> . Nach diesem Datum kann sich der User nicht mehr anmelden. usermod -e "" user entfernt ein Ablaufdatum (setzt es auf <i>nie</i> ).
<b>-f Tage</b>	Setzt die Anzahl an Tagen nach Passwortablauf, bis der Account deaktiviert wird <a href="#">wiki.ubuntuusers.de</a> . -f 0 deaktiviert sofort nach Ablauf; -f -1 entfernt eine evtl. gesetzte Deaktivierung (nie sperren nach Ablauf).
<b>-L</b>	Sperrt das Benutzerkonto (lock). Fügt ein ! dem Passwort-Hash voran (im Shadow-Eintrag) <a href="#">wiki.ubuntuusers.de</a> , was die Anmeldung verhindert. Kann nicht mit -p oder -U kombiniert werden.
<b>-U</b>	Entsperrt ein zuvor gesperrtes Passwort (passwd -l). Entfernt das ! im Shadow-Passwortfeld, sodass der ursprüngliche Hash wieder wirksam wird <a href="#">wiki.ubuntuusers.de</a> . Nicht zusammen mit -L oder -p verwenden.
<b>-p Hash</b>	Setzt das <i>verschlüsselte</i> Passwort (Hash) des Users auf den angegebenen Wert <a href="#">wiki.ubuntuusers.de</a> . Sollte mit Vorsicht genutzt werden – für Passwortänderung ist passwd meist besser geeignet.
<b>--help</b>	Hilfe anzeigen.

#### **Wichtige Hinweise:**

- *Warnung:* Die Kombination von -G ohne -a überschreibt die Gruppenliste vollständig. Das heißt, der Benutzer wird aus allen Gruppen entfernt, die nicht explizit bei -G genannt sind[wiki.ubuntuusers.de](#). Um einen User nur hinzuzufügen, niemals ohne -a verwenden!
- Man kann keinen angemeldeten Benutzer umbenennen oder dessen UID ändern, solange er eingeloggt ist oder laufende Prozesse hat. Zudem keine Änderungen an dem eigenen Account durchführen, während man noch mit diesem angemeldet ist (kann zu Inkonsistenzen führen).
- Beim Ändern des Benutzernamens oder UIDs sollten evtl. Crontab-Dateien und geplante at-Jobs manuell umbenannt/angepasst werden, und in NIS/LDAP-Umgebungen müssen Änderungen dort vorgenommen werden[wiki.ubuntuusers.de](#).

#### **userdel (Benutzer löschen)**

**Beschreibung:** Löscht ein Benutzerkonto aus dem System (entfernt die Einträge in /etc/passwd, /etc/shadow und ggf. /etc/group falls der Benutzer eine eigene Gruppe hatte). Optional können auch das Home-Verzeichnis und die Mail-Spool des Benutzers gelöscht werden.

**Syntax:**

userdel [OPTION] BENUTZER

**Optionen:**

**Option Beschreibung**

**-r** Entfernt *rekursiv* das Heimatverzeichnis des Benutzers **und** seine Mail-Spool (üblicherweise /var/spool/mail/<user>), sowie alle Dateien darin, beim Löschen des Accounts. Ohne -r bleiben Home und Mails unangetastetfile-4fdk7rvx39azjretinarqb.

**-f** Force: Erzwingt das Löschen des Accounts, selbst wenn der Benutzer noch angemeldet ist oder laufende Prozesse hat. (Warnung: kann zu inkonsistenten Systemzuständen führen, Prozess bleibt ohne Benutzer etc. Diese Option sollte vermieden werden.)

Ohne Option wird der Account aus den Systemdateien gelöscht, aber das (*keine*) Home-Verzeichnis, die Dateien und Mails bleiben bestehen. Admin muss sie ggf. manuell entfernen oder anderweitig zuordnen.

*Hinweis:* userdel bricht ab, wenn der Benutzer gerade angemeldet ist (außer mit -f). Es ist ratsam, vor dem Löschen sicherzustellen, dass keine Prozesse mehr unter der betreffenden UID laufen (z.B. mit pkill -u username beenden). Dateien außerhalb des Home, die dem Benutzer gehören, verbleiben auf dem Dateisystem mit der nun unbenutzten UID – diese müssten manuell gesucht und entfernt oder neu zugeordnet werden (z.B. mit find / -uid <oldUID>).

**id**

**Beschreibung:** Zeigt die Benutzer- und Gruppen-Identitäten eines Benutzerkontos an. Standardmäßig gibt id die UID, primäre GID und zugehörigen Gruppennamen des aktuellen Benutzers zurückfile-4fdk7rvx39azjretinarqb. Mit Angabe eines Benutzernamens kann man die IDs für einen anderen Account abrufen.

**Syntax:**

id [OPTION]... [BENUTZER]

**Optionen:**

<b>Option</b>	<b>Beschreibung</b>
(ohne)	Ohne Options werden UID, Benutzername, GID, Gruppenname und alle zusätzlichen Gruppen des Benutzers aufgelistet (z.B. uid=1000(max) gid=1000(max) Gruppen=1000(max),27(sudo),...).
<b>-u</b>	Gibt nur die effektive User-ID (UID, numerisch) aus. Beispiel: id -u alice könnte 1001 ausgeben.
<b>-g</b>	Gibt nur die effektive Gruppen-ID (GID, numerisch) aus.
<b>-G</b>	Gibt alle Gruppen-IDs (numerisch) aus, in denen der Benutzer Mitglied ist (Primär- und Sekundärgruppen).
<b>-n</b>	Gibt Namen anstelle von numerischen IDs. Kann in Kombination mit -u, -g, -G verwendet werden. Z.B. id -Gn alice listet alle Gruppennamen von alice.
<b>-r</b>	Zeigt die reale UID/GID an (falls z.B. mit sudo aufgerufen, reale vs. effektive ID unterscheiden).
<b>-Z</b>	Zeigt SELinux-Sicherheitskontext, falls SELinux aktiv ist (z.B. unconfined_u:unconfined_r:unconfined_t:s0 für einen unconfined Benutzer).
<b>--help, --version</b>	Hilfe bzw. Version anzeigen.

### **groupadd (Gruppe anlegen)**

**Beschreibung:** Legt eine neue Gruppe im System an. Fügt einen Eintrag in /etc/group (und ggf. /etc/gshadow) hinzu.

#### **Syntax:**

groupadd [Optionen] GRUPPENNAME

#### **Optionen:**

<b>Option</b>	<b>Beschreibung</b>
<b>-g GID</b>	Verwendet die angegebene numerische GID für die neue Gruppe. Sollte noch nicht vergeben sein (oder mit -o überschrieben werden).
<b>-o</b>	Erlaubt die Verwendung einer bereits existierenden GID (zusammen mit -g zu nutzen, um GID-Doppel zu erzwingen).
<b>-f</b>	"Force": falls die Gruppe schon existiert, keinen Fehler melden; falls die angegebene GID bereits vergeben ist, wählt eine andere (ignoriert -o).

<b>Option</b>	<b>Beschreibung</b>
<b>-r</b>	Erstellt eine Systemgruppe (GID wird aus dem System-Bereich gewählt, z.B. <1000).
<b>-K Schlüssel= Wert</b>	Überschreibt einen Eintrag aus /etc/login.defs temporär (z.B. -K GID_MIN=500 um untere GID-Bereiche anzupassen).
<b>-p Passwort</b>	Setzt das (verschlüsselte) Gruppenpasswort direkt. (Gruppenpasswörter werden selten genutzt – ermöglicht mit newgrp temporär in eine Gruppe zu wechseln.)

### **groupmod (Gruppe ändern)**

**Beschreibung:** Ändert Eigenschaften einer bestehenden Gruppe – hauptsächlich Name oder GID.

**Syntax:**

groupmod [Optionen] GRUPPE

**Optionen:**

<b>Option</b>	<b>Beschreibung</b>
<b>-n NeuerName</b>	Ändert den Gruppennamen in <i>NeuerName</i> . Der alte Name in allen relevanten Systemdateien wird ersetzt.
<b>-g NeueGID</b>	Ändert die GID der Gruppe. Muss eindeutig sein (oder mit -o doppelt erlaubt). Dateien mit alter GID außerhalb, die nicht automatisch erfasst werden, müssen eventuell manuell angepasst werden.
<b>-o</b>	Erlaubt das Setzen einer bereits vergebenen GID (nur mit -g relevant).
<b>-p Passwort</b>	Ändert das (verschlüsselte) Gruppenpasswort. (Entspricht Editieren von /etc/gshadow).

### **groupdel (Gruppe löschen)**

**Beschreibung:** Entfernt eine Gruppe aus der Systemgruppe-Datei. Die Gruppe wird aus /etc/group (und /etc/gshadow) getilgt.

**Syntax:**

groupdel GRUPPENNAME

**Hinweis:** Dieser Befehl hat keine besonderen Optionen – er löscht die Gruppe, sofern sie nicht als Primärgruppe eines Benutzers in /etc/passwd noch verwendet wird. Man kann keine Gruppe löschen, die noch als Hauptgruppe eines existierenden Benutzerkontos eingetragen ist; in dem Fall muss man ggf. erst den Benutzer anpassen

oder löschen. Dateien auf dem System, die dieser Gruppe gehören, behalten die nun ungültige GID bei, bis man sie manuell ändert.

## **su (Switch User)**

**Beschreibung:** Wechselt den Benutzerkontext im laufenden Terminal. Mit su („substitute user“) kann man einen anderen Benutzeraccount annehmen – standardmäßig root, wenn kein Benutzer angegeben wird. Dies erfordert die Eingabe des Zielbenutzer-Passworts (außer beim Wechsel zu root, dort abhängig von den Systemeinstellungen – oft ist root-Login per Passwort deaktiviert und man nutzt sudo).

### **Syntax:**

`su [OPTION]... [BENUTZER [-]]`

*Wird BENUTZER weggelassen, versucht su auf den Superuser (root) zu wechseln.*

### **Optionen:**

#### **Option      Beschreibung**

**-l** Startet eine *Login-Shell* der Zielusers, d.h. die Umgebung wird vollständig umgeschaltet als würde sich der Benutzer frisch anmelden. Das Home-Verzeichnis, Pfad (\$PATH) etc. werden übernommen. Beispiel: `su - alice` wechselt zu Benutzer *alice* mit deren Umgebung (äquivalent zu vollem Logout/Login als *alice*).

**-c Befehl** Führt einen angegebenen *Befehl* als Zielbenutzer aus und kehrt dann zurück. Beispielsweise `su -c "systemctl restart apache2"` - root führt als root einen Neustart des Apache durch und beendet sich sofort wieder.

**-m, -p** Behalten die aktuelle Umgebung (*nicht* auf Ziel-User umschalten). Der aktuelle \$HOME, \$PATH, usw. bleiben erhalten, nur die Rechte wechseln. Standardverhalten ohne -l. (In vielen su Implementierungen ist -m identisch mit -p: *preserve environment*).

**-s Shell** Startet die angegebene *Shell* statt der Standard-Login-Shell des Zielbenutzers. Der Pfad zur Shell muss im /etc/shells stehen, sofern nicht root.

**-g Gruppe** Wechselt in Kombination mit Benutzer auch die primäre Gruppe auf *Gruppe*. (Nur verfügbar in neueren util-linux su).

**-<Ziffer>** (BSD su) Angabe einer Session-Klasse – unter Linux nicht genutzt.

**--help, --version** Anzeigen von Hilfe bzw. Version.

### *Hinweise:*

- su ohne - (also z.B. su root) wechselt zwar die Benutzer-ID, aber behält das aktuelle Verzeichnis und weite Teile der Environment von Ursprungskonto bei (\$LOGNAME, \$HOME bleiben z.B. vom ursprünglichen Nutzer). Das kann zu irritierenden Effekten führen (z.B. Schreibrechte im falschen Home). Daher nutzt man meist su - für einen *sauberen* Wechsel.
- Viele Distributionen konfigurieren aus Sicherheitsgründen su so, dass ein direkter Wechsel zu *root* per Passwort entweder deaktiviert ist oder kein *root*-Passwort gesetzt ist. Stattdessen wird sudo bevorzugt, um Root-Rechte zu erlangen.

## **sudo (Superuser Do)**

**Beschreibung:** Führt ein Kommando mit *erhöhten Rechten* aus, typischerweise als *root*, ohne die Benutzeridentität dauerhaft zu wechseln. Benutzer können mit sudo Befehle mit Administratorrechten ausführen, sofern sie in /etc/sudoers entsprechend autorisiert sind. Im Gegensatz zu su wird hier das eigene Passwort abgefragt (nicht das des Zielusers), und nur spezifische Befehle/Optionen sind erlaubt gemäß Konfiguration.

### **Syntax:**

sudo [OPTION]... [COMMAND]

*Wird ein Befehl angegeben, führt sudo diesen als root (Standard) aus. Ohne Befehl zeigt sudo ggf. die erlaubten Befehle an oder wechselt in eine sudo-Shell (bei -s).*

### **Optionen (Auswahl):**

<b>Option</b>	<b>Beschreibung</b>
<b>-l</b>	Listet die erlaubten (und verbotenen) Befehle für den aufrufenden Benutzer auf (nach sudoers-Regeln). Hilfreich, um zu sehen, was man mit sudo ausführen darf.
<b>-u Benutzer</b>	Führt das angegebene Kommando als <i>Benutzer</i> aus (anstatt als root). Beispiel: sudo -u alice whoami zeigt "alice", wenn man als anderer User ausführt. Standard ist root.
<b>-s</b>	Startet eine Shell als Zielbenutzer (Standard root). Ähnlich wie su, aber mittels sudo-Rechte. Es wird die im \$SHELL oder in passwd eingetragene Shell geöffnet.
<b>-i</b>	Simuliert eine vollständige Login-Shell wie su - (initialisiert Umgebung des Zielusers). Entspricht sudo -s plus Login-Umgebung.
<b>-b</b>	Führt den Befehl im Hintergrund aus ( <i>background</i> ). Sudo verlangt dennoch ggf. ein Passwort, aber detach danach.

<b>Option</b>	<b>Beschreibung</b>
<b>-E</b>	Beibehaltung der aktuellen Environment-Variablen (normalerweise bereinigt sudo einige Variablen aus Sicherheitsgründen). Erlaubt das Mitnehmen der eigenen Umgebung. (Ob das erlaubt ist, kann in sudoers eingeschränkt sein.)
<b>-p</b> <i>Prompt</i>	Definiert einen eigenen Passwort-Prompt. z.B. sudo -p "Passwort für %u:" -v (Platzhalter %u = Username).
<b>-v</b>	"Validate": Aktualisiert/überprüft das eigene sudo-Authentifizierungsticket. D.h. fordert ggf. Passwort an, speichert das sudo Token (5min Standard) aber führt sonst nichts aus. (Verlängert die Zeit, in der man kein Passwort erneut eingeben muss.)
<b>-k</b>	Invalidiert das eigene sudo-Ticket sofort (nachfolgende sudo-Aufrufe verlangen wieder Passwort) – ( <i>kill credential</i> ).
<b>-H</b>	Setzt \$HOME in der Environment auf das Home des Zielusers (root). Normalerweise belässt sudo \$HOME vom Aufrufer; mit -H verhält es sich wie su -.
<b>--help, -h</b>	Hilfe anzeigen (kurze Übersicht der Optionen).
<b>-K</b>	Löscht das Zeitstempel-Ticket komplett (ähnlich -k aber gründlicher, forciert Passwort prompt beim nächsten Mal).
<b>--version</b>	Versionsinfo ausgeben (auch Liste der unterstützten Sicherheitsmechanismen etc.).

*Hinweise:* Bei erfolgreicher Authentifizierung legt sudo einen Zeitstempel (Ticket) an, sodass für nachfolgende Aufrufe (Standard ~5 Minuten) kein erneutes Passwort nötig ist. Dies kann per sudoers Einstellung (timestamp\_timeout) angepasst oder mit sudo -k/-K beeinflusst werden.

### **pkexec (PolicyKit Execute)**

**Beschreibung:** Führt ein Programm als ein anderer Benutzer aus (ähnlich wie sudo, aber Teil von PolKit/PolicyKit). Standardmäßig wird, sofern nicht anders angegeben, das Programm mit root-Rechten gestartet[freedesktop.org](http://freedesktop.org). pkexec erfordert, dass der aufrufende Benutzer berechtigt ist, die Aktion auszuführen (PolKit Policies); falls ja, wird eine Authentifizierungsdialog (grafisch oder textbasiert) angezeigt, um ggf. das Passwort einzugeben. Nach erfolgreicher Auth kann das gewünschte Kommando mit erhöhten Rechten ausgeführt werden.

#### **Syntax:**

pkexec [--user <BENUTZER>] <KOMMANDO> [ARG...]

## Optionen:

Option	Beschreibung
--user <Benutzer>	Führt das angegebene Programm als <i>Benutzer</i> aus. Standard ist root (Administratorkonto) <a href="http://freedesktop.org">freedesktop.org</a> .
--disable-internal-agent	Unterdrückt den internen Authentifizierungs-Agent von pkexec. Normalerweise, falls kein grafischer PolKit-Auth-Agent verfügbar ist, startet pkexec einen eigenen textuellen Auth-Dialog im Terminal. Mit dieser Option wird in so einem Fall statt dessen das Programm <b>nicht</b> ausgeführt (Fehler, wenn kein Agent aktiv) <a href="http://freedesktop.org">freedesktop.org</a> .
--help / --version	Zeigt Hilfe bzw. Version von pkexec.

*Hinweise:* pkexec erbt ein Minimal-Umfeld (Environment), um das Einschleusen von Code via Umgebungsvariablen zu verhindern[freedesktop.org](http://freedesktop.org). Display- und Xauth-Variablen werden aus Sicherheitsgründen *nicht* an das Zielprogramm weitergegeben, außer es wurde in den PolKit-Regeln explizit erlaubt (GUI-Programme via pkexec sind daher tricky ohne entsprechende PolKit-Policy). In modernen Distributionen ist pkexec oft als Ersatz für grafische gksu-Aufrufe gedacht, um z.B. aus der Desktop-Umgebung heraus GUI-Administrationsprogramme mit Root-Rechten zu starten (wobei PolKit dann i.d.R. das Benutzer-Passwort abfragt).

## Berechtigungen und Zugriffskontrolle

### umask (Set Default Permissions Mask)

**Beschreibung:** Legt eine Berechtigungsschutzmaske fest, die bei neu erstellten Dateien und Verzeichnissen die Standardzugriffsrechte beeinflusst. umask subtrahiert Bits von den vollen Zugriffsrechten. So bestimmt sie, welche Rechte *NICHT* automatisch gesetzt werden dürfen. Dies wirkt pro Prozess/Session. Als Shell-Builtin zeigt umask ohne Parameter die aktuelle Maske an.

#### Syntax:

umask [Option] [Maske]

#### Optionen / Verwendung:

- Ohne Argument zeigt umask die aktuelle Mask in Oktalnotation an (z.B. 0022). Mit Option **-S** in symbolischer Form (z.B. u=rwx,g=rx,o=rx entspricht Maske 022).
- Mit Angabe einer neuen Maske (z.B. umask 027) wird diese gesetzt. Die Maske wird meist oktal angegeben: jedes Bit, das in der Maske 1 ist, nimmt der neu erstellten Datei die entsprechende Berechtigung weg. Beispiel: Maske 022

bedeutet, neuen Dateien werden keine Gruppe- und Andere-Schreibrechte gegeben (d.h. max Resultatrechte für Datei typ. 644, für Verzeichnis 755).

- **Symbolische Maske:** Man kann auch wie bei chmod symbolisch angeben, z.B. umask u=rwx,g=rx,o=rx für dasselbe Ergebnis (Maske 022).

*Hinweis:* Die Umask wirkt auf Systemaufruf-Ebene: Standard-Dateirechte sind 666 (rw-rw-rw) und für Verzeichnisse 777 (rwxrwxrwx); die Maske wird davon abgezogen. Z.B. Umask 002 ergibt Dateien mit 664 (rw-rw-r-) und Verzeichnisse mit 775. Umask 077 würde sehr restriktiv alles nur für den Benutzer selbst erlauben (Datei 600, Dir 700).

### **chmod (Change Mode - Dateirechte ändern)**

**Beschreibung:** Ändert die Zugriffsberechtigungen einer Datei oder eines Verzeichnisses. Man kann mit chmod sowohl numerisch (Oktalwerte) als auch symbolisch (u/g/o +-/=r,w,x) die Rechte setzen.

#### **Syntax:**

chmod [Optionen] <Modus> <Datei/Verzeichnis>...

#### **Optionen:**

Option	Beschreibung
<b>-R, --recursive</b>	Rekursive Rechteänderung: traversiert Unterverzeichnisse und wendet chmod auf alle darin befindlichen Dateien/Ordner an. Wichtig bei Verzeichnisbäumen.
<b>-v, --verbose</b>	Gibt für jede Datei eine Meldung aus, deren Rechte erfolgreich geändert wurden (und ggf. was geändert wurde).
<b>-c, --changes</b>	Wie verbose, aber nur Ausgaben für tatsächlich geänderte Rechte (keine Ausgabe, wenn Rechte schon so eingestellt waren).
<b>-f, --silent</b>	Unterdrückt Fehlermeldungen (force).
<b>--reference=&lt;Vorlage&gt;</b>	Setzt die gleichen Berechtigungen wie die Referenz-Datei <Vorlage> hat, auf die angegebenen Dateien. Statt Modus muss man --reference nutzen.

#### **Modus-Angaben:**

- **Numerisch (Oktal):** Eine dreistellige (oder vierstellige mit Sonderbit) Oktalzahl. Z.B. chmod 640 datei.txt setzt Rechte auf rw- r-- --- (6=rw,4=r,0=kein Recht). Jede Ziffer entspricht Owner, Group, Others. Beispiel: chmod 755 script.sh => rwx für Besitzer (7), rx für Gruppe (5), rx für andere (5). Eine führende vierte Ziffer kann

spezielle Bits setzen: 4 für Setuid, 2 für Setgid, 1 für Sticky-Bit (z.B. 1755 für Sticky).

- **Symbolisch:** Format: [ugoa]\*[+-=][rwxXst]\* evtl. durch Kommata getrennt für mehrere Operationen in einem Aufruf. Beispiele:
  - chmod u+x,g-w,o= file – dem Besitzer Ausführungsrecht geben, der Gruppe Schreibrecht entziehen, andere bekommen exakt keine Rechte.
  - chmod a+r file – allen (a für all) Leserechte hinzufügen (ohne andere Bits zu ändern).
  - chmod g=u Datei – setzt die Gruppenrechte gleich den Benutzerrechten der Datei (nützliche Kurzform).
  - X: Spezialflag in symbolischer Notation – steht für "execute if directory or if any execute bit is set on any of owner/group/other". Wird meist bei -R verwendet, um rekursiv nur Verzeichnissen das x zu geben.
  - s für setuid/setgid (in u/g Kontext), t für Sticky-Bit (im o-Kontext).
  - Beispiel: chmod u+s script.sh – setzt das setuid-Bit auf die Datei (falls ausführbar).

## chown (Change Owner)

**Beschreibung:** Ändert den Dateibesitzer und/oder die Gruppe einer Datei oder eines Verzeichnisses

file-4fdk7rvx39azjretinarqbfile-4fdk7rvx39azjretinarqb. chown kann sowohl den User als auch die Gruppe in einem Befehl setzen. Nur root (oder Capability CHOWN) darf Eigentümer von Dateien ändern. Normale Benutzer können in der Regel mittels chgrp die Gruppenzugehörigkeit ändern, wenn sie Mitglied der Zielgruppe sind (und Besitzer der Datei bleiben).

### Syntax:

chown [Optionen] <neuerEigentümer>[:<neueGruppe>] DATEI...

### Optionen:

Option	Beschreibung
-R	Rekursiv – wendet die Besitzänderung auf alle Dateien und Unterverzeichnisse an (ganzer Baum).
-h	Ändert den Besitzer von Symbolic Links selbst, statt des Ziels. (Auf den meisten Linux ist das Standard mit Kernel 2.6+; teils wird -h ignoriert, da chown auf Symlinks ohnehin Link selbst ändert, außer mit -L Option).

Option	Beschreibung
<b>-f</b>	Keine Fehlermeldungen ausgeben (silent/force).
<b>-v</b>	Verbose – listet jede Datei mit geänderten Owner/Group auf.
<b>--</b>	Ändert nur, wenn aktuelle Owner/Group genau wie angegeben ist. (Damit kann man selektiv bestimmte Kombis ersetzen, dient als Sicherheitsprüfung).
<b>from=&lt;old_owner:old_group&gt;</b>	<i>Follow symlinks:</i> Symbolische Links auflösen (statt Link selbst zu ändern wird das Ziel geändert). Standardverhalten für die meisten chown-Versionen ist bereits, dem Link <i>nicht</i> zu folgen (außer man setzt -L).
<b>--dereference / -L</b>	Standardverhalten für die meisten chown-Versionen ist bereits, dem Link <i>nicht</i> zu folgen (außer man setzt -L).
<b>-P</b>	<i>Don't follow symlinks</i> (explizit Link behandeln, Standard).
<b>-H</b>	Folgen von Symlinks im Kommandozeilen-Argument (Top-Level), aber nicht rekursiv tiefer. (Selten benötigt).

### Verwendung:

- Besitzer ändern: chown alice datei.txt – setzt *alice* als neuen Besitzer der Datei (Gruppe bleibt unverändert).
- Gruppe ändern mit chown: chown :developers projektX – behält Besitzer, setzt Gruppe auf *developers*. Notation: Vor dem Doppelpunkt nichts heißt Besitzer unverändert, aber *Gruppe* ändern.
- Besitzer und Gruppe zugleich: chown alice:staff datei.txt – Besitzer = *alice*, Gruppe = *staff*.
- Nur Gruppe ändern könnte auch mit eigenem Befehl chgrp staff datei.txt gemacht werden.

*Besonderheit:* Man kann numerische IDs angeben. chown 1001:100 datei setzt Besitzer UID=1001, GID=100. Falls man einen Benutzer- oder Gruppennamen mit Doppelpunkt hat, der numerisch beginnt, kann man -- verwenden um Verwechslung zu vermeiden oder explizit username: schreiben.

### chgrp (Change Group)

**Beschreibung:** Ändert die Gruppenzugehörigkeit einer Datei oder eines Verzeichnissesfile-4fdk7rvx39azjretinarqbfile-4fdk7rvx39azjretinarqb. Äquivalent zu chown :<gruppe>. Wird benutzt, um den Gruppeneigentümer anzupassen.

### Syntax:

chgrp [Optionen] <neueGruppe> DATEI...

#### Optionen:

Option	Beschreibung
<b>-R</b>	Rekursive Änderung der Gruppe in Unterverzeichnissen (alle Dateien/Dirs innerhalb werden angepasst).
<b>-f</b>	Keine Fehlermeldungen bei Problemen ausgeben (silent).
<b>-v</b>	Ausführliche Ausgabe pro Datei.
<b>--reference=Vorlage</b>	Übernimmt die Gruppenzugehörigkeit der <i>Vorlage</i> -Datei auf alle angegebenen Dateien. (Ignoriert das Gruppen-Argument und nutzt stattdessen die Gruppe der Referenz.)

*Hinweis:* Normale Benutzer dürfen chgrp nur verwenden, um eine Datei auf eine Gruppe zu setzen, der sie selbst angehören. Root kann beliebige Zuordnungen vornehmen.

#### lsattr (List Attributes)

**Beschreibung:** Listet die *erweiterten Dateiattribute* (Filesystem-Attribute) von ext-Dateisystemen auffile-4fdk7rvx39azjretinarqb. Diese Attribute wie *immutable* oder *append-only* steuern besondere Verhalten auf niedriger Ebene (nur auf bestimmten FS wie ext2/3/4, XFS - mit anderen Tools - etc. vorhanden). lsattr zeigt pro Datei eine Zeichenfolge von Attribut-Flags.

#### Syntax:

lsattr [Optionen] [Datei/Verzeichnis]...

#### Optionen:

##### Option Beschreibung

<b>-a</b>	Zeigt auch versteckte Dateien (Dateien, die mit Punkt beginnen) an. Standardmäßig ignoriert lsattr diese analog zu ls ohne -a.
<b>-d</b>	Listet das Verzeichnis selbst auf, nicht dessen Inhalt. (Bei Angabe eines Verzeichnisses wird normalerweise dessen Inhalt rekursiv aufgelistet; mit -d sieht man nur die Attribute des Verzeichniseintrags selbst)file-4fdk7rvx39azjretinarqb.
<b>-R</b>	Rekursive Ausgabe der Attribute in allen Unterverzeichnissen.
<b>-v</b>	Zeigt die Dateiversionsnummer (Version im ext-Format) mit an, falls vorhanden. (Das Attribut 'version' wird kaum benutzt, -v gibt eine zusätzliche Spalte mit Versionsnummer).

## Option Beschreibung

**-V** Ausführliche Informationen über die Version des Programms und detaillierte Fehler ausgeben (verbose).

### Beispiel Ausgabe:

```
$ lsattr wichtige_datei
----i-----e-- wichtige_datei
```

Hier bedeuten die Buchstaben: **i** = immutable (unveränderlich), **e** = extents (standard auf ext4). Bindestriche zeigen nicht gesetzte Flags. Wichtige Flags: **i** (immutable: Datei kann nicht verändert/gelöscht umbenannt werden bis Attribut entfernt), **a** (append-only: nur anhängen erlaubt), **d** (no dump: Datei wird von Dump-Backup ignoriert), **A** (no atime updates), **S** (synchronous updates) etc.

## chattr (Change Attributes)

**Beschreibung:** Ändert die erweiterten Dateisystem-Attribute (wie immutable, append-only etc.) unter Linux auf ext2/ext3/ext4 Dateisystemenfile-4fdk7rvx39azjretinarqb. Nur Root darf die meisten dieser Attribute setzen.

### Syntax:

```
chattr [Optionen] [+--Attribute] Datei...
```

### Optionen:

Option	Beschreibung
<b>-R</b>	Rekursive Attributänderung in Unterverzeichnissenfile-4fdk7rvx39azjretinarqb.
<b>-V</b>	Ausführlich – zeigt an, welche Änderungen für welche Datei vorgenommen wurden.
<b>-f</b>	Unterdrückt Fehlermeldungen (force).
<b>-v</b> <b>&lt;Nummer&gt;</b>	Setzt die Versionsnummer der Datei (falls vom Dateisystem unterstützt). Selten benutzt – ext-Dateisysteme haben optional Dateiversionsnummern.

**Attribute setzen (Symbolik):** Man gibt nach möglichen Optionen eine Zeichenfolge an, der ein +, - oder = vorangestellt ist:

- + bedeutet diese Attribute *hinzufügen* (zusätzlich zu bestehenden).
- - bedeutet diese Attribute *entfernen*.
- = bedeutet *exakt setzen* (alle aktuellen Attribute durch diese ersetzen).

Die wichtigsten Attribut-Buchstaben auf ext4 sind:

- **i – immutable**: Datei kann nicht modifiziert, umbenannt oder gelöscht werden file-5y1edouysrow4zj6phjcqm. Prozesse können sie nicht öffnen im Schreibmodus. Schützt kritische Dateien (z.B. /etc/passwd) vor Änderungen, sogar von root (root muss erst chattr -i machen).
- **a – append only**: Datei kann nur angehängt, aber nicht überschrieben oder gelöscht werden. Typisch für Logfiles, damit niemand alte Einträge löscht.
- **e – extent format**: zeigt an, dass die Datei extents benutzt (automatisch auf ext4, nicht manuell gesetzt).
- **d – no dump**: Datei soll von Backup-Programmen, die dem Dump-Standard folgen, ignoriert werden.
- **A – no atime updates**: beim Lesen der Datei wird der atime-Zeitstempel nicht aktualisiert (Performance-Optimierung).
- **S – synchronous updates**: Änderungen an der Datei werden synchron auf den Datenträger geschrieben (wie bei chattr +S – ähnlich wie Ordner mit Dirsync mount option).
- **t – no tail-merging**: (relevant bei ext-Dateisystem, deaktiviert Tail-Merging bei Reverse-Mapping; selten benutzt).
- **T – top of directory hierarchy**: Verzeichnis wird als Top-Level für Ordnungszwecke in ext behandelt (verhindert bestimmte Ordnungsoptimierungen; selten).
- **u – undeletable**: Dateiinhalt soll beim Löschen inhaltlich gesichert werden, sodass man ihn wiederherstellen könnte. (Wird von ext4 i.d.R. nicht implementiert).

### Beispiel:

- chattr +i datei.txt – setzt *immutable*.
- chattr -i datei.txt – entfernt dieses Flag wieder (dann ist Bearbeiten/Löschen wieder möglich).
- chattr =a Datei.log – setzt **nur** append-only und entfernt alle anderen Attribute (falls welche gesetzt waren). Danach kann in Datei.log nur noch angehängt werden.

### getfacl (Get File ACL)

**Beschreibung:** Liest die Access Control Lists (erweiterten Zugriffsrechte) von Dateien oder Verzeichnissen aus und zeigt sie an file-4fdk7rvx39azjretinarqb. Auf Linux-Systemen mit POSIX-ACL-Unterstützung (z.B. ext4 mit acl-Option eingehängt) kann man pro Datei feinere Berechtigungen für einzelne Benutzer oder Gruppen setzen. getfacl zeigt diese

ACL-Einträge zusätzlich zu den Standardrechten.

### Syntax:

getfacl [Optionen] Datei...

### Optionen:

Option	Beschreibung
<b>-a</b>	Zeigt auch die <i>verschachtelten ACLs</i> (falls vorhanden – nicht üblich bei POSIX ACL).
<b>-e</b>	Zeigt die Ausgabe im <i>effective Modus</i> – Standard. (Jeder Eintrag inkl. "effective permissions", relevant wenn Mask gesetzt ist).
<b>-s</b>	Stripschutz: Zeigt nur die tatsächlichen ACL-Einträge, ohne die Grundeinträge (user/group/other).
<b>-d</b>	Zeigt die <i>Default-ACL</i> eines Verzeichnisses (die Vorlage, die an neu darin erzeugte Dateien vererbt wird). In der Ausgabe mit default: gekennzeichnet.
<b>--omit-header</b>	Unterdrückt den Kommentar-Kopf in der Ausgabe, der Dateinamen und Hinweise beinhaltet.
<b>-R</b>	Rekursiv über Verzeichnisse, getfacl für jede Datei auflisten.

### Ausgabe-Beispiel:

```
$ getfacl projekt/
file: projekt/
owner: alice
group: developers
user::rwx
user:bob:rwx- # bob hat rw Rechte zusätzlich
group::r-x
mask::rw- # effektive Maske für ACL = rw
other::---
default:user::rwx # Default ACL für neu angelegte Objekte
...
```

Hier sieht man die Standard-Owner/Group-Einträge und eine explizite ACL, die User bob Schreibrechte gibt, sowie eine *mask*, die die maximal wirksamen Rechte auf rw beschränkt.

### **setfacl (Set File ACL)**

**Beschreibung:** Setzt oder ändert Access Control Lists (ACLs) von Dateien und Verzeichnissen file-4fdk7rvx39azjretinarqb. Damit kann man zusätzliche Berechtigungen für bestimmte Nutzer oder Gruppen vergeben (oder entziehen), die über die klassischen Besitzer-Gruppe-Andere Rechte hinausgehen.

#### **Syntax:**

```
setfacl [Optionen] { -m|-x <ACL-Eintrag> ... | -b | -k } Datei...
```

#### **Optionen:**

##### **Option Beschreibung**

- m *ACL*** *Modify*: Fügt einen ACL-Eintrag hinzu oder aktualisiert ihn. Beispiel: setfacl -m u:bob:rw file gibt Benutzer *bob* rw-Rechte auf *file*. Mehrere -m Einträge oder Einträge mit Komma getrennt sind möglich.
- x *ACL*** *Remove*: Entfernt einen spezifischen ACL-Eintrag. Z.B. setfacl -x g:marketing file entfernt den ACL-Eintrag für Gruppe "marketing".  
**-b** *Remove all* – Löscht alle ACL-Einträge (außer den Basis-Einträgen für Owner/Group/Other). Damit werden alle erweiterten Berechtigungen zurückgesetzt. (Datei hat dann nur noch die klassischen Unix-Rechte.)
- k** *Remove Default* – Entfernt die Default-ACL eines Verzeichnisses (vererbte ACL-Vorlage). Hat auf Dateien keine Wirkung.
- R** Rekursiv – wendet die ACL-Änderung auf alle Dateien im Verzeichnis (und Unterverzeichnisse) an. Nützlich um z.B. ganze Bäume eine neue ACL zu geben.
- d** Setzt Einträge als *Default-ACL* für Verzeichnisse. Benutzt in Kombination mit -m oder -x. Beispiel: setfacl -m d:g:team:rwX projekt setzt eine Default-ACL, so dass neue Dateien unter *projekt*/ der Gruppe *team* rw Rechte erben, und Verzeichnisse rwx.
- mask** Berechnet und setzt die ACL-Maske basierend auf den aktuellen Gruppen-Einträgen neu. (Die Maske begrenzt die maximalen Rechte der ACL-Einträge außer Owner/other). Standardmäßig wird die Maske auto-anangepasst.

## Option Beschreibung

- Don't recalc mask:* Verhindert, dass setfacl die ACL-Maske automatisch anpasst. Normalerweise soll man diese Option nicht verwenden, außer man setzt die Maske manuell mit -m m::....

**Beispiel:** setfacl -m u:bob:rw,g:marketing:r-- report.txt – gibt Benutzer *bob* Lese- und Schreibrecht und der Gruppe *marketing* nur Leserecht auf *report.txt*.

*Hinweis:* Nicht alle Dateisysteme unterstützen POSIX-ACL. Bei ext4, XFS etc. muss i.d.R. das Volume mit Option acl eingehängt sein (moderne Systeme haben das meist per Default). Mit getfacl kann man prüfen, ob vorhandene ACLs existieren.

## Datei- und Verzeichnisverwaltung

### stat (Dateistatus anzeigen)

**Beschreibung:** Zeigt detaillierte Meta-Informationen zu Dateien oder Dateisystemen an. stat gibt z.B. Größe, Blocks, Zeitstempel (Zugriff, Änderung, Statusänderung), Berechtigungen in Zahlenform, Device-ID, Inode-Nummer usw. in strukturierter Form aus.

#### Syntax:

stat [Optionen] <Datei>...

#### Optionen:

Option	Beschreibung
<b>-f, --file-system</b>	Zeigt anstelle der Datei-Details die Informationen zum Dateisystem, auf dem die Datei liegt. Z.B. Statistiken wie Blockgröße, Gesamtblöcke, freie Blöcke, FS-Typ etc.
<b>-c FORMAT, --format=FORMAT</b>	Gibt die Ausgabe gemäß <i>FORMAT</i> formatiert aus. Man kann Platzhalter verwenden, um bestimmte Werte auszugeben. Z.B. stat -c "%A %h %U %G %s %n" datei gibt in einer Zeile: modestr, Linkanzahl, Owner, Group, Size, Name. (Siehe man stat für alle Platzhalter wie %A = Rechte in rwx, %a = oktal, %U = Ownername, %s = Größe etc.)
<b>--printf=FORMAT</b>	Ähnlich wie --format, jedoch ohne automatisch Zeilenumbruch am Ende. Kann benutzt werden, um mehrere stat-Aufrufe zu skripten.
<b>-t, --terse</b>	"Knapp": Gibt die Ausgabe in einer kompakten einzeiligen Form, Felder durch Leerzeichen getrennt, aus (für einfache maschinelle Weiterverarbeitung).

<b>Option</b>	<b>Beschreibung</b>
<b>--help, --version</b>	Anzeigen von Hilfe bzw. Version.

*Beispielausgabe:*

```
$ stat /etc/hosts
 File: /etc/hosts
 Size: 258 Blocks: 8 IO Block: 4096 regular file
Device: 802h/2050d Inode: 13422506 Links: 1
Access: (0644/-rw-r--r--) Uid: (0/ root) Gid: (0/ root)
Access: 2025-04-01 10:22:13.000000000 +0200
Modify: 2025-04-01 10:22:13.000000000 +0200
Change: 2025-04-01 10:22:15.000000000 +0200
 Birth: -
```

Dies zeigt z.B., dass /etc/hosts 258 Bytes groß ist, auf einem bestimmten Device, inode etc., Rechte 0644, Eigentümer root, Zeitstempel für Zugriffs-/Änderungs-/Änderungszeit (Birth ist Erstellung, nicht unterstützt).

### **file (Dateityp bestimmen)**

**Beschreibung:** Untersucht eine Datei und versucht den Typ bzw. Format der Datei zu bestimmen. file nutzt einen umfangreichen Magie-Pattern-Katalog (/usr/share/file/magic), um anhand der Bytes einer Datei (Magic Numbers, Struktur etc.) menschlich lesbare Dateitypinformationen auszugeben. Praktisch, um z.B. Binärformate, Skripttypen oder Kodierungen zu erkennen.

#### **Syntax:**

file [Optionen] <Datei>...

#### **Optionen:**

##### **Option Beschreibung**

- Zeigt den MIME-Typ der Datei anstatt einer Beschreibung. Z.B. text/plain;
- i** charset=utf-8 statt "UTF-8 Unicode text". Hilfreich für Scripting oder Web-Kontext.
- Brief* – gibt nur die Beschreibung aus, ohne den Dateinamen voranzustellen.
- b** (By default schreibt file: "<Dateiname>: <Beschreibung>"). Mit -b erhält man nur "<Beschreibung>".

## Option Beschreibung

- Folgt symbolischen Links und untersucht das Ziel statt den Link selbst.
- L   Standardmäßig wird der Link selbst untersucht und meist als "symbolic link to <target>" ausgegeben.
- (alias --no-dereference) Bewirkt das Gegenteil: behandelt Symlinks wie
- h   normale Dateien (gleichbedeutend mit Standardverhalten, -h ist nützlich falls -L als default kompiliert wäre).
- Untersucht auch Gerätedateien (block/char special) und Auslagerungsdateien.
- s   Standardmäßig werden einige spezielle Dateitypen nicht inhaltlich analysiert  
(um z.B. nicht eine Device-Datei zu lesen). -s kann riskant sein (kann z.B. /dev/zero "lesen" -> blockiert).
- z   Durchsucht komprimierte Dateien (gzip-komprimierte) – versucht innerhalb eines .gz die Dateisignatur zu prüfen.
- E   Gibt (Exit)Status 1 zurück falls *irgendeine* der Dateien kein bekannter Typ bestimmt werden konnte. (Nützlich in Skripten).
- help,   Hilfe bzw. Versionsnummer und unterstützte Magic-Datei-Version ausgeben.  
-v

Beispiele:

- file /bin/ls -> **ELF 64-bit LSB executable**, x86-64, dynamically linked ...
- file archive.tar.gz -> **gzip compressed data**, was probably a tar archive (reads inside and erkennt es als tar in gz).
- file picture.jpg -> **JPEG image data**, JFIF standard 1.01, ...

## ln (Link erstellen)

**Beschreibung:** Erstellt Verknüpfungen zu Dateien – entweder **Hard Links** oder **Symbolische Links**. Ein Hardlink ist ein weiterer Directory-Eintrag für dieselbe Inode (nur innerhalb desselben Dateisystems möglich). Ein Symbolischer Link (Symlink) ist eine kleine Spezialdatei, die auf einen Pfad verweist.

**Syntax:**

ln [Optionen] <Ziel> [Linkname]

Standard ohne Optionen erstellt *Hardlink* mit Namen *Linkname*, der auf Datei *Ziel* zeigt. Wenn *Linkname* nicht angegeben, wird im aktuellen Verzeichnis ein Link gleichen Namens wie *Ziel* erstellt. Für Symlinks Option -s nutzen.

**Optionen:**

<b>Option</b>	<b>Beschreibung</b>
<b>-s, --symbolic</b>	Erstellt einen <i>symbolischen Link</i> statt eines Hardlinks. Der Link verweist auf den angegebenen <Ziel>-Pfad. Wenn <Ziel> kein absoluter Pfad ist, wird er relativ vom Link aus interpretiert. Beispiel: ln -s /etc/hosts hostslink erzeugt Symlink <i>hostslink</i> -> /etc/hosts.
<b>-f, --force</b>	Existierende Zielfile (Linkname) ggf. entfernen, falls schon vorhanden. Ohne -f würde ln bei existierendem Linknamen einen Fehler ausgeben. (Bei -s mit -f: überschreibt existierenden Link.)
<b>-n, --no-dereference</b>	Beim Überschreiben eines existierenden symbolischen Links (mit -f) behandle den Link selbst statt dessen Ziel. (Verhindert, dass versehentlich die Datei, auf die ein bestehender Link zeigt, gelöscht wird).
<b>-v, --verbose</b>	Gibt eine Meldung pro verarbeitetem Link aus.
<b>-T, --no-target-directory</b>	Behandelt das letzte Argument immer als <i>genauen Linknamen</i> , selbst wenn es ein existierendes Verzeichnis ist. (Verhindert, dass ln im Zielverzeichnis einen Link mit Standardname erstellt.)
<b>-d, -F, --directory</b>	Erlaubt es root, auch <i>Verzeichnis-Hardlinks</i> zu erstellen. (Normalerweise verboten – Option nur auf speziellen Systemen mit -f kombinierbar, in der Praxis kaum verwendet).
<b>-b, --backup</b>	Erzeugt ein Backup der Zielfile falls -f zum Überschreiben genutzt wird. (Nützlich, um das ersetzte File nicht zu verlieren.) Siehe auch -- backup=t etc.
<b>-S &lt;Suffix&gt;</b>	Gibt ein Suffix für Backup-Dateien an (Standard ~). Bsp: -b -S .bak.

*Beispiele:*

- Hardlink: ln /path/to/original.txt copy.txt – copy.txt ist nun ein weiterer Verweis auf *original.txt* (gleiche Inode). Änderungen an einer erscheinen in der anderen, da es dieselbe Datei ist. Löscht man eine, bleibt die andere erhalten (Datei wird erst gelöscht, wenn letzter Link entfernt).
- Symlink: ln -s /etc/apache2 siteconfig – erstellt *siteconfig* -> /etc/apache2. Aufruf von ls -l siteconfig zeigt dann siteconfig -> /etc/apache2.

### **cd (Change Directory) – siehe oben bei Grundlegende Befehle**

*(Im Kontext Datei/Verzeichnis-Verwaltung sei auf den bereits beschriebenen Befehl cd verwiesen, um Verzeichnisse zu wechseln.)*

## **tree (Verzeichnisbaum anzeigen)**

**Beschreibung:** Zeigt die Verzeichnisstruktur grafisch in Form eines Baumes an. tree listet rekursiv alle Unterverzeichnisse und Dateien in hierarchischer eingerückter Darstellung. Es erleichtert das Visualisieren der Verzeichnis-Hierarchie.

### **Syntax:**

`tree [Optionen] [Verzeichnis]`

### **Optionen:**

<b>Option</b>	<b>Beschreibung</b>
<b>-a</b>	Zeigt <i>alle</i> Dateien an, inkl. versteckter (auch solche, die mit . beginnen). Standardmäßig ignoriert tree versteckte Dateien, außer . und .. immer.
<b>-d</b>	Listet <i>nur Verzeichnisse</i> auf, keine Dateien. Nützlich, um reinen Verzeichnisbaum zu sehen.
<b>-L Level</b>	Beschränkt die Tiefe der Ausgabe auf <i>Level</i> Ebenen. Z.B. tree -L 2 zeigt nur zwei Ebenen tief (akt. Verzeichnis + Unterverzeichnisse, aber keine tieferen).
<b>-f</b>	Zeigt den vollen Pfad vor jedem Eintrag (statt nur Namen relativ).
<b>-I</b>	Ignoriert Einträge, die auf das Muster passen (Groß/Kleinschreibung
<i>PATTERN</i>	berücksichtigt, mehrere Patterns mit `
<b>-P</b>	Zeigt nur Einträge an, die dem Pattern entsprechen (Whitelist). Umgekehrt zu
<i>PATTERN</i>	-I.
<b>-F</b>	Markiert Verzeichnisse mit abschließendem /, ausführbare Dateien mit *, Symlinks mit @, Pipes mit `
<b>-p</b>	Zeigt UNIX-Dateirechte (Permission bits) vor jedem Eintrag.
<b>-u / -g</b>	Zeigt Benutzer bzw. Gruppenbesitzer an. (Wie ls -l Infos).
<b>-h</b>	<i>Human-readable size</i> : Zeigt Dateigrößen in menschenlesbarem Format (K, M, G) anstatt in Bytes (wirkt mit -s).
<b>-s</b>	Zeigt Dateigrößen in Bytes neben jedem Dateinamen an.
<b>--du</b>	Summiert Verzeichnisgrößen wie du (statt wie normal lediglich die Einzeldateigröße zu zeigen).
<b>-t</b>	Sortiert die Ausgabe nach Zeit (Neueste zuerst) statt alphabetisch.
<b>-o Datei</b>	Gibt die Ausgabe in die angegebene Datei (überschreibt) anstatt auf stdout.

<b>Option</b>	<b>Beschreibung</b>
<b>-n</b>	Keine farbliche Hervorhebung (Standard ist bunt, -n erzwingt Monochrom).
<b>--help</b>	Hilfe anzeigen (Übersicht der Optionen).

*Beispiel:*

```
$ tree -L 2 /etc
```

/etc

```

└── apache2
 ├── apache2.conf
 ├── envvars
 └── mods-available

 └── cron.daily
 ├── apt-compat
 └── man-db

 └── default
 ├── cron
 └── rcS
```

6 directories, 5 files

Hier sieht man /etc mit zwei Ebenen Tiefgang.

### **mkdir (Make Directory)**

**Beschreibung:** Erstellt ein neues Verzeichnis (oder mehrere) im Dateisystem. Kann rekursiv auch mehrere Ebenen auf einmal anlegen.

**Syntax:**

```
mkdir [Optionen] <Verzeichnisname>...
```

**Optionen:**

<b>Option</b>	<b>Beschreibung</b>
<b>-p, --parents</b>	Erzeugt auch alle notwendigen übergeordneten Verzeichnisse, falls sie nicht existieren. Somit werden keine Fehler ausgegeben, wenn

Option	Beschreibung
	Verzeichnisse in der Pfadangabe fehlen. Beispiel: mkdir -p /tmp/foo/bar legt sowohl /tmp/foo (falls nicht da) als auch /tmp/foo/bar an.
<b>-m, --mode=MODE</b>	Setzt die Zugriffsrechte des neu erstellten Verzeichnisses auf <i>MODE</i> (numerisch oder symbolisch) anstatt der Standardrechte (die Standardrechte sind i.d.R. 0777 abzüglich Umask). Z.B. mkdir -m 750 projDir erstellt projDir mit rwx für Besitzer, rx für Gruppe, nichts für andere.
<b>-v, --verbose</b>	Ausgabe einer Meldung für jedes angelegte Verzeichnis (zeigt Pfad an).
<b>--help, --version</b>	Anzeige von Hilfe bzw. Version.

### cp (Copy File)

**Beschreibung:** Kopiert Dateien oder Verzeichnisse. Bei Angabe mehrerer Quelldateien muss das letzte Argument ein Verzeichnis sein (Zielverzeichnis). cp kann auch Verzeichnisbäume rekursiv kopieren.

#### Syntax:

cp [Optionen] <Quelle>... <Ziel>

*Wenn <Ziel> ein Verzeichnis ist, werden alle <Quelle>-Dateien dorthin kopiert (mit gleichem Namen). Ist nur eine Quelle und <Ziel> kein Verzeichnis, wird die Datei dorthin kopiert/umbenannt.*

#### Optionen:

Option	Beschreibung
<b>-r, -R, --recursive</b>	Kopiert Verzeichnisse rekursiv (Inhalt und Unterverzeichnisse). Ohne diese Option kopiert cp standardmäßig keine Verzeichnisse. (GNU cp hat auch -r/-R – beide meist gleichwertig. POSIX fordert -R.)
<b>-a, --archive</b>	Archivmodus: Kopiert rekursiv und bewahrt dabei möglichst alle Attribute. Entspricht -dR --preserve=all (also -R, und -p für timestamps, owner, group, perms, ACLs, xattrs, Links etc.). Dies ist für Vollbackups ideal <a href="#">ionos.at</a> .
<b>-p, --preserve</b>	Bewahrt Eigentümer, Gruppe, Berechtigungen und Zeitstempel der Datei beim Kopieren <a href="#">ionos.at</a> . (Ohne -p übernimmt cp Standard: setzt aktuellen Zeitstempel "jetzt" und erstellt als aktueller

Option	Beschreibung
	Benutzer). Kann Argumente haben wie --preserve=mode,ownership,timestamps,links,xattr,all.
<b>-f, --force</b>	Überschreibt vorhandene Ziel-Dateien ohne Nachfrage, und versucht auch nicht-schreibbare Ziele zu entfernen, um Kopie dennoch zu platzieren. (Im Konflikt mit -n gilt letzter Schalter.)
<b>-i, --interactive</b>	Fragt nach, bevor eine existierende Zielfile überschrieben wird (Confirm Y/N). Gibt dem Benutzer die Möglichkeit, Überschreiben abzulehnen.
<b>-n, --no-clobber</b>	Überschreibt <i>keine</i> existierenden Dateien. D.h. falls Ziel existiert, wird die Kopie für diese Datei ausgelassen. (Verhindert unabsichtliches Überschreiben)
<b>-u, --update</b>	Kopiert nur, wenn <i>Quelle</i> neuer als <i>Ziel</i> ist, oder Ziel noch nicht existiert. Praktisch für inkrementelle Updates. Existiert Ziel mit gleich/neueren Datum, wird es nicht überschrieben.
<b>-d, --no-dereference</b>	Kopiert <i>Symbolic Links</i> als Links (anstatt deren referenzierten Inhalt). Standard: cp folgt symbolischen Links und kopiert die Datei dahinter. Mit -d wird der Link selbst neu erstellt am Ziel (Hardlinks bleiben hardlinks sofern möglich; -d ist in GNU cp gleichbedeutend mit --preserve=links).
<b>-L, --dereference</b>	<i>Folgt allen Symlinks:</i> kopiert stets den referenzierten Inhalt (Standardverhalten bei regulärem cp für Dateien). Bei Angabe von Verzeichnis mit -R: -L würde bewirken, dass cp <i>alle</i> gefundene Symlinks auflöst und deren Ziel kopiert (kann zu Loops führen, daher mit Vorsicht).
<b>-P, --no-dereference --preserve=links</b>	Kopiert Links als Links (das ist Standard mit -a enthalten). (Im GNU cp -P bedeutet No-dereference; beachten: -d und -P ähnlich).
<b>--parents</b>	Erhält die Quellpfadstruktur im Ziel. Bsp: cp --parents dir1/dir2/file.txt /backup legt /backup/dir1/dir2/file.txt an <a href="#">man7.org</a> . Nur sinnvoll, wenn Ziel ein Verzeichnis ist.
<b>-l, --link</b>	Erstellt <i>Hardlinks</i> statt Kopien der Datei, sofern im selben Dateisystem. (Keine Datenkopie, nur neuer Name, ähnlich ln).

<b>Option</b>	<b>Beschreibung</b>
<b>-s, --symbolic-link</b>	Erstellt <i>Symlinks</i> statt die Daten zu kopieren. (Zieldatei wird symbolischer Verweis auf Quelldatei).
<b>-H, -L, -P (bei -R)</b>	Steuerung des Symlink-Verhaltens <b>während</b> rekursivem Kopieren: -P (Default) ignoriert Linkinhalte (kopiert Symlinks als Symlinks), -L folgt Symlinks zu Dateien, -H folgt nur Symlinks angegeben auf Befehlzeile. (-a impliziert -P).
<b>--</b> <b>backup[=Control]</b>	Erzeugt Backup von Zieldateien, die überschrieben würden. (Etwa file.txt~ anlegen). Kann =simple, numbered etc. steuern. Siehe auch VERSION_CONTROL Env.
<b>--help, --version</b>	Hilfe/Version anzeigen.

## **mv (Move/Rename)**

**Beschreibung:** Verschiebt Dateien oder Verzeichnisse an einen neuen Pfad, bzw. benennt sie um. mv kann auch zum einfachen Umbenennen einer Datei verwendet werden (Quell- und Zielpfad auf selbem FS = schneller Rename, kein Kopieren). Falls Quelle und Ziel auf unterschiedlichen Dateisystemen liegen, arbeitet mv durch Kopieren und anschließendem Löschen der Quelle.

### **Syntax:**

mv [Optionen] <Quelle>... <Ziel>

### **Optionen:**

<b>Option</b>	<b>Beschreibung</b>
<b>-i, --interactive</b>	Fragt nach Bestätigung, falls eine Zieldatei durch den Move überschrieben würde (falls im Zielpfad eine Datei gleichen Namens existiert).
<b>-f, --force</b>	Erzwingt das Überschreiben ohne Nachfrage. Ziel-Dateien werden ohne Warnung ersetzt. (Standardverhalten ohne -i).
<b>-n, --no-clobber</b>	Überschreibt keine bestehenden Dateien. Bei Namenskollision bricht mv für diese Datei ab (gibt keinen Fehler, exit code trotzdem 0). In GNU mv hat -n Vorrang vor -i falls beide angegeben.
<b>-u, --update</b>	Bewegt/überschreibt nur, wenn die Quelle neuer ist als die vorhandene Zieldatei <i>oder</i> die Zieldatei nicht existiert. (So

<b>Option</b>	<b>Beschreibung</b>
	werden neuere Versionen moved, ältere Quellen, die schon am Ziel sind, bleiben unberührt.)
<b>-v, --verbose</b>	Gibt aus, welche Datei wohin verschoben/umbenannt wurde (für jede Operation eine Zeile).
<b>--</b>	
<b>backup[=CONTROL]</b>	Wie bei cp: legt Backup der Ziel-Datei an, falls existiert, bevor sie überschrieben wird. CONTROL kann none, numbered, existing, simple sein (siehe Doku oder ENV VERSION_CONTROL).
<b>--strip-trailing-slashes</b>	Entfernt evtl. nachgestellte Schrägstriche bei den Pfadangaben der Quellen (um Verwirrung mit Verzeichnissen zu vermeiden). Standard: neuere mv machen das ohnehin.
<b>--help, --version</b>	Hilfe/Version.

*Hinweis:* mv kann mehrere Dateien in ein Zielverzeichnis verschieben oder eine einzelne Datei/Ordner umbenennen. Bei Verzeichnissen rekursiv, aber mv muss dafür nicht extra Option haben (im Gegensatz zu cp), da der Move innerhalb desselben FS nur Metadaten ändert. Bei verschiedenen FS kopiert mv im Hintergrund (rename() Syscall schlägt dann fehl, worauf cp+unlink passiert).

### rmdir (Remove Directory)

**Beschreibung:** Löscht leere Verzeichnisse. Im Gegensatz zu rm entfernt rmdir nur Verzeichnisse und auch nur, wenn sie keine Dateien/Unterverzeichnisse mehr enthalten. Es kann optional ganze verschachtelte Verzeichnisbäume entfernen, sofern diese leer sind (Eltern werden dann rekursiv entfernt).

#### Syntax:

rmdir [Optionen] <Verzeichnis>...

#### Optionen:

<b>Option</b>	<b>Beschreibung</b>
<b>-p, --parents</b>	Entfernt das angegebene Verzeichnis und ggf. dessen übergeordnete Verzeichnisse, falls diese nach jedem Entfernen leer werden. Beispiel: rmdir -p projekt/build/output versucht erst output zu löschen. Ist danach build leer, wird auch build gelöscht. Ist danach projekt leer, wird auch projekt gelöscht.
<b>-v, --verbose</b>	Ausgabe einer Meldung für jedes gelöschte Verzeichnis.

<b>Option</b>	<b>Beschreibung</b>
<b>--ignore-fail-on-non-empty</b>	Unterdrückt die normale Fehlermeldung, falls ein Verzeichnis nicht leer ist. (Das Verzeichnis wird nicht gelöscht, aber es kommt kein Fehlerstatus; mit Vorsicht verwenden).
<b>-r, -R, --recursive</b>	Löscht Verzeichnisse <i>rekursiv</i> mitsamt ihrem gesamten Inhalt. Damit können komplette Verzeichnisbäume entfernt werden. Ohne -r verweigert rm das Löschen von Verzeichnissen mit Fehler "is a directory".
<b>-d</b>	Löscht ein <i>leeres</i> Verzeichnis. (Ohne -r löscht rm normalerweise gar keine Verzeichnisse; -d erlaubt es aber für leere, ähnlich rmdir. Nützlich, aber -r deckt diesen Fall auch ab.)

**Nützlich:** Um verschachtelte leere Ordner zu entfernen. Aber um Verzeichnisse mit Inhalt zu löschen, nutzt man rm -r.

### **rm (Remove Files/Directories)**

**Beschreibung:** Löscht Dateien oder Verzeichnisse. Standardmäßig entfernt rm nur Dateien; mit -r kann es auch Verzeichnisbäume löschen (inkl. aller enthaltenen Dateien). **Vorsicht:** Gelöschte Dateien werden nicht ohne Weiteres wiederherstellbar gespeichert (kein Papierkorb, außer man nutzt safe-rm Tools).

#### **Syntax:**

rm [Optionen] <Datei/Pfad>...

#### **Optionen:**

<b>Option</b>	<b>Beschreibung</b>
<b>-r, -R, --recursive</b>	Löscht Verzeichnisse <i>rekursiv</i> mitsamt ihrem gesamten Inhalt. Damit können komplette Verzeichnisbäume entfernt werden. Ohne -r verweigert rm das Löschen von Verzeichnissen mit Fehler "is a directory".
<b>-d</b>	Löscht ein <i>leeres</i> Verzeichnis. (Ohne -r löscht rm normalerweise gar keine Verzeichnisse; -d erlaubt es aber für leere, ähnlich rmdir. Nützlich, aber -r deckt diesen Fall auch ab.)
<b>-f, --force</b>	<i>Force/ignore nonexistent:</i> Unterdrückt Rückfragen und Fehlermeldungen. Existiert eine angegebene Datei nicht, wird es ignoriert. Schreibschutz wird ignoriert (d.h. es fragt nicht nach Bestätigung bei schreibgeschützten Dateien). Kombiniert mit -r sehr gefährlich, da es kompromisslos alles löscht. Wird oft in Skripten genutzt.
<b>-i</b>	Interaktiv nachfragen vor <i>jedem</i> Löschvorgang. (Für viele Dateien unpraktisch).
<b>-I</b>	"ein wenig interaktiv": Fragt nur einmal pauschal nach, <b>wenn</b> mehr als 3 Dateien gelöscht werden sollen <i>oder</i> ein rekursiver Lösch auf einem Verzeichnis durchgeführt wird. (GNU rm Feature als Sicherheitsnetz)file-5y1edouysrow4zj6phjcqm.

Option	Beschreibung
-- <b>preserve-</b> <b>root</b>	Schützt vor rm -rf /. Diese Option ist in GNU rm standardmäßig aktiv – es verweigert das komplette Löschen des Root-FS. Man kann mit --no-preserve-root überschreiben (nicht empfohlen).
<b>-v, --verbose</b>	Zeigt für jede Datei eine Meldung an, dass sie gelöscht wurde.
<b>--help, --version</b>	Hilfe bzw. Version.

*Anwendung:*

- **Einzelne Datei löschen:** rm datei.txt.
- **Mehrere Dateien löschen:** rm \*.o (löscht alle .o Dateien im aktuellen Verzeichnis).
- **Verzeichnis und Inhalt löschen:** rm -r docs/ – löscht Ordner docs mitsamt allen Unterdateien.
- **Ohne Rückfrage alles löschen:** rm -rf tempdir. (Vorsicht!).

### head (Dateianfang anzeigen)

**Beschreibung:** Gibt die ersten Zeilen einer Datei aus (standardmäßig 10 Zeilen). head ist nützlich, um schnell die ersten Zeilen (z.B. Header einer CSV, Beginn eines Logs etc.) anzuschauen, ohne die ganze Datei zu öffnen. Es kann auch binäre Dateien auf Anfang bytes prüfen.

#### Syntax:

head [Optionen] [Datei]...

Ohne Datei liest head von der Standardeingabe (z.B. in Pipes). Bei mehreren Dateien wird jeweils ein Header mit dem Dateinamen ausgegeben.

#### Optionen:

Option	Beschreibung
<b>-n K, --lines=K</b>	Gibt die ersten <i>K</i> Zeilen aus. <i>K</i> kann eine Zahl sein oder -K für alle bis auf die letzten <i>K</i> Zeilen (negativer Wert). Standard ist 10. Beispiel: head -n 20 file zeigt 20 Zeilen, head -n -5 file zeigt alle außer der letzten 5 Zeilen.
<b>-c Bytes, --bytes=B</b>	Gibt die ersten <i>B</i> Bytes aus (statt Zeilen). Kann auch mit K, M, etc. Suffix (Potenzen von 1024) angegeben werden. Z.B. head -c 100K bigfile. Negative Werte analog: -c -10 alles außer letzte 10 Bytes.

<b>Option</b>	<b>Beschreibung</b>
<b>-q, --quiet/-silent</b>	Unterdrückt die Überschriften, die vor dem Inhalt stehen, wenn mehrere Dateien angegeben wurden. Normal zeigt head sowas wie "=> filename <==" vor dem Block jeder Datei. Mit -q wird alles aneinandergehängt ohne solche Banner.
<b>-v, --verbose</b>	Zeigt immer die Dateiüberschriften an, auch wenn nur eine Datei angegeben ist. (Normalerweise bei nur einer Datei keine Überschrift).
<b>--help, --version</b>	Hilfe bzw. Version ausgeben.

*Beispiel:* head -n 3 /etc/passwd zeigt die ersten 3 Zeilen der passwd-Datei (typisch Root, daemon, bin Accounts).

### **tail (Dateiende anzeigen)**

**Beschreibung:** Gibt die letzten Zeilen einer Datei aus (standardmäßig 10 Zeilen). Sehr nützlich, um z.B. das Ende von Logfiles zu sehen. Kann im *Follow-Modus* (-f) genutzt werden, um fortlaufend neuen Inhalt anzuzeigen (Echtzeit-Log-Überwachung).

#### **Syntax:**

tail [Optionen] [Datei]...

#### **Optionen:**

<b>Option</b>	<b>Beschreibung</b>
<b>-n K, --lines=K</b>	Zeigt die letzten <i>K</i> Zeilen an. Negative Werte bedeuten ab einer bestimmten Zeile von Anfang an ( <i>ab</i> Zeile <i>K</i> ). Standard ist 10. Z.B. tail -n 50 /var/log/syslog zeigt die letzten 50 Zeilen.
<b>-c B, --bytes=B</b>	Zeigt die letzten <i>B</i> Bytes an. Kann auch negative ( <i>ab</i> Byte <i>X</i> ) sein.
<b>-f, --follow</b>	Bleibt nach dem Anzeigen der letzten Zeilen geöffnet und zeigt fortlaufend neue Zeilen an, falls die Datei wächst (wie tail -f logfile zum Mitlesen).
<b>-F</b>	Ähnlich wie -f, aber zusätzlich "Wiederöffnen", falls Datei rotiert/neu erstellt wurde. (Follow Name statt Descriptor; praktisch für Logrotation – tail -F /var/log/app.log folgt auch nach Neustart des Logs dem neuen File).
<b>-s Sek, --sleep-interval=Sek</b>	Im Follow-Modus: Warte <i>Sek</i> Sekunden zwischen den Überprüfungen auf neue Daten. Standard 1.0 Sekunde. Kann z.B. auf 0.1 verringert werden, um schneller zu pollen, oder erhöht, um Last zu reduzieren.

Option	Beschreibung
<b>--max- unchanged- stats=N</b>	(Bei -F relevant) Wie oft in Folge unveränderte Inode geprüft wird, bevor tail davon ausgeht, dass ggf. Logrotate stattgefunden hat.
<b>-q, --quiet</b>	Unterdrückt Überschriften bei mehreren Dateien (analog head).
<b>-v, --verbose</b>	Erzwingt Überschriften, selbst bei einer Datei.
<b>--help, --version</b>	Hilfe bzw. Version.

*Anwendung:*

- **Letzte Zeilen einer Datei:** tail -n 20 messages.log (Zeigt die letzten 20 Zeilen).
- **Live-Verfolgung:** tail -f /var/log/syslog (Druckt neu geschriebene Logzeilen fortlaufend). Abbruch mit Ctrl+C.
- **Combine mit grep:** z.B. tail -f server.log \| grep "ERROR" filtert live nur Zeilen mit "ERROR".

## Dateien finden und Text filtern

### grep (Global Regular Expression Print)

**Beschreibung:** Durchsucht die Eingabe (Dateien oder Standard Input) nach Zeilen, die einem bestimmten Muster entsprechen, und gibt diese Zeilen ausfile-4fdk7rvx39azjretinarqb. grep arbeitet mit regulären Ausdrücken (Standard *Basic Regex*, erweiterte mit -E). Es ist das klassische Suchwerkzeug in Textstreams. Name stammt von der ed-Editor Anweisung g/re/p (global regex print).

#### Syntax:

grep [Optionen] 'Muster' [Datei...]

#### Optionen:

Option	Beschreibung
<b>-i, --ignore-case</b>	Ignoriert Groß-/Kleinschreibung beim Mustervergleichfile-4fdk7rvx39azjretinarqb. "Test" findet auch "test" oder "TEST".
<b>-v, --invert- match</b>	Gibt alle Zeilen <b>aus, die nicht</b> auf das Muster passen (Inverse Treffer)file-4fdk7rvx39azjretinarqb. Nutzt man oft zur Ausschlussfilterung.
<b>-r, -R, -- recursive</b>	Durchsucht rekursiv alle Dateien in angegebenen Verzeichnissen (und Unterverz.)file-4fdk7rvx39azjretinarqb. Ohne diese Option

Option	Beschreibung
	behandelt grep Verzeichnisargumente normalerweise als Fehler oder ignoriert sie. (GNU grep: -r folgt Symlinks nicht, -R folgt Symlinks.)
<b>-n, --line-number</b>	Zeigt vor jeder gefundenen Zeile die Zeilennummer in der Datei an <a href="#">ibm.com</a> . Format: Datei:Nr:Zeilentext.
<b>-A NUM</b>	(After) Zeigt <i>NUM</i> Zeilen <b>nach</b> jeder Trefferzeile zusätzlich an (Context) <a href="#">ibm.com</a> .
<b>-B NUM</b>	(Before) Zeigt <i>NUM</i> Zeilen <b>vor</b> jeder Trefferzeile an.
<b>-C NUM</b>	(Context) Zeigt <i>NUM</i> Zeilen <i>vor und nach</i> jedem Treffer (gleichzeitig -A NUM -B NUM). Standard -C 2 beispielsweise gibt 2 Kontextzeilen um jede Fundstelle.
<b>-l, --files-with-matches</b>	Listet nur die Dateinamen auf, die mindestens einen Treffer enthalten. Pro Datei ein Name, Inhalt wird nicht ausgegeben. (Nützlich um zu sehen wo etwas vorkommt)
<b>-L, --files-without-match</b>	Listet nur Dateinamen, die <i>keinen</i> Treffer enthalten (Gegenteil von -l).
<b>-c, --count</b>	Zählt die Treffer pro Datei und gibt nur diese Anzahl aus (zusammen mit Dateinamen, außer mit -h). D.h. "file.txt:3". In Kombination mit -v zählt es Nicht-Treffer-Zeilen.
<b>-o, --only-matching</b>	Gibt nur den Teil der Zeile aus, der dem Suchmuster entspricht, nicht die ganze Zeile. Bei mehreren Treffern pro Zeile werden diese getrennt ausgegeben.
<b>-q, --quiet/-silent</b>	"Quiet": gibt keine Ausgabe; liefert lediglich Exit-Status 0 falls wenigstens ein Match gefunden wurde, 1 falls kein Match. Praktisch in Skripten (nur Rückgabewert interessiert).
<b>-s, --no-messages</b>	Unterdrückt Fehlermeldungen über nicht lesbare Dateien (z.B. wenn Zugriffsrechte fehlen oder Datei nicht existiert).
<b>-E, --extended-regexp</b>	Interpretiert das Muster als <i>Extended Regular Expression (ERE)</i> <a href="#">ionos.at</a> . Das erlaubt z.B. Nutzung von +, ?, `
<b>-F, --fixed-strings</b>	Interpretiert das Muster als <i>fixen String</i> (kein Regex, jedes Zeichen literal). Entspricht fgrep (das eine schnellere Suche ohne regex

Option	Beschreibung
	ermöglicht). Alle Muster (wenn -e mehrfach) werden als Literalstrings gesucht.
<b>-P, --perl-regexp</b>	Interpretiert das Muster als Perl-Compatible Regex (PCRE), falls grep mit PCRE-Unterstützung kompiliert wurde. Damit stehen mächtigere Regex-Features zur Verfügung.
<b>-h, --no-filename</b>	Unterdrückt die Ausgabe des Dateinamens vor jeder Trefferzeile (wenn mehrere Dateien durchsucht werden). Standardmäßig präfixt grep Datei: bei mehreren Input-Dateien. Mit -h erscheinen nur die Zeilen. (Bei Einzeldatei ist das sowieso Standard.)
<b>-H, --with-filename</b>	Zeigt Dateinamen auch bei einziger Datei an. (Umgekehrt zu -h). Nützlich in Pipelines wo Source unbekannt.
<b>--color[=WHEN]</b>	Hebt das gefundene Pattern farblich hervor in der Ausgabe. WHEN kann auto (Standard, nur wenn Terminal), always oder never sein. Setzt Umgebungsvariable GREP_COLOR/GREP_COLORS ggf. benötigt.
<b>-e Pattern</b>	Gibt das Suchmuster explizit an. Nötig z.B. wenn das Pattern mit - beginnt (damit es nicht als Option interpretiert wird). Kann mehrfach angegeben werden, um mehrere Muster (OR-Verknüpfung) zu suchen.
<b>-f Datei</b>	Liest Suchmuster aus angegebener Datei (ein Pattern pro Zeile). Nützlich bei sehr vielen oder komplexen Patterns.
<b>--exclude=GLOB</b>	Schließt Dateien vom Durchsuchen aus, deren Name auf GLOB (Wildcard-Muster) passt. Z.B. --exclude=*.min.js würde Minimierte JS ignorieren. (Nur wirksam bei rekursiver Suche oder wenn Verzeichnis als Input). Es gibt auch --include analog.
<b>--exclude-dir=GLOB</b>	Wie --exclude, aber nur auf Verzeichnisnamen angewandt, um ganze Verzeichnisse von der Rekursion auszunehmen (z.B. --exclude-dir=.git).
<b>--help, --version</b>	Hilfe / Version.

**Exit-Status:** 0 falls mindestens ein Match gefunden, 1 falls kein Match, 2 falls ein Fehler auftrat (z.B. Datei nicht lesbar).

### Beispiele:

- grep -i "error" \*.log – sucht case-insensitive "error" in allen .log im aktuellen Ordner, gibt Trefferzeilen mit Dateipräfix aus.
- grep -R "TODO" src/ – findet rekursiv alle Vorkommen von "TODO" in Dateien unterhalb von src/ (typisch in Codebasis).
- grep -nE "^Error|Warning:" report.txt – sucht Zeilenbeginn "Error:" oder "Warning:" und gibt mit Zeilennummer aus [pq.hosting](#).
- ps ax | grep "[c]ron" – findet Prozesse mit "cron" im Namen. (Trick: [c] im Pattern matcht 'c', verhindert aber, dass der grep selbst im ps-Listing gefunden wird, da dessen Kommandozeile "grep cron" kein 'cron' substring hat sondern 'cro n' durch char class unterbrochen).

## **find (Dateien suchen)**

**Beschreibung:** Durchsucht Verzeichnisse nach Dateien/Verzeichnissen, die bestimmten Kriterien entsprechen (Name, Typ, Größe, Datum, Rechte, Eigentümer etc.)  
**file-4fdk7rvx39azjretinarqb.** find ist sehr mächtig: man kann logische Bedingungen kombinieren und Aktionen auf gefundene Dateien ausführen (löschen, verschieben, Exec von Kommandos etc.). Es arbeitet rekursiv ab dem angegebenen Pfad.

### **Syntax:**

find [Start-Verzeichnis] [Optionen/Ausdrücke] [Aktion(en)]

### **Wichtige Suchkriterien (Ausdrücke):**

- **-name Muster** – Dateiname entspricht *Wildcards* (Globbing wie bei Shell, z.B. "\*.txt"). Groß/Kleinschreibung beachten.
- **-iname Muster** – wie -name, aber case-*insensitive*.
- **-type c** – Dateityp ist c. Häufig: **f** für regular file, **d** für directory, **l** symlink, **b** blockdevice, **c** chardevice, **p** named pipe, **s** socket. Z.B. -type d nur Verzeichnisse  
**file-4fdk7rvx39azjretinarqb.**
- **-user name/uid** – Datei gehört dem Benutzer *name* (oder UID numerisch). Analog **-group name/gid**.
- **-uid +n/-n/n** – UID der Datei ist ( $>/</=$ ) n. -uid +500 UID größer 500. (Entspricht -user nur numerisch und mit Vergleichen). Genauso -gid.
- **-size N\*[bckMG]\*** – Dateigröße-Kriterium. *N* kann mit Vorzeichen angegeben werden (+ größer als, - kleiner als, nichts = genau). Suffix: standardmäßig in 512-Byte Blöcken, oder **c** = Bytes, **k** = KiB, **M** = MiB, **G** = GiB. Beispiel: -size +100M Dateien größer 100 MiB [wiki.foxtom.de](#), -size -10k kleiner 10 KiB.
- **-mtime Tage** – Letzte Änderung (Modify timestamp) liegt *Tage* zurück. *Tage* als ganze Zahl: +n = älter als n Tage, -n = jünger als n Tage, n = genau n Tage (im

Bereich ±0.5d). Ähnlich **-atime** (Access), **-ctime** (Statusänderung). Z.B. -mtime -7 in letzten 7 Tagen modifiziert [wiki.foxtom.de](http://wiki.foxtom.de).

- **-newer Datei** – Datei wurde später geändert als *Datei* (Vergleich mtime). **-newermt Datum** (GNU find) erlaubt Vergleich mit einem Zeit-String, z.B. -newermt "2025-05-01".
- **-perm mode** – Dateirechte entsprechen *mode*. Mode kann exakt (z.B. 644) oder mit / oder - angegeben werden für Masken. Beispiel: -perm 644 findet genau 644. -perm -644 findet Dateien, die mindestens diese Bits haben (also rw- r-- r-- oder mehr Rechte). -perm /222 findet Dateien, bei denen irgendein Schreibbit gesetzt ist (Owner oder Group oder Other).
- **-maxdepth N** – Nur bis Tiefe N ins Verzeichnis hinab suchen (0= nur Startdir selbst). **-mindepth N** analog minimal tiefe. Z.B. find . -maxdepth 1 -type f nur Dateien im aktuellen Verzeichnis, nicht tiefer.
- **-mount** oder **-xdev** – Verlasse nicht das Dateisystem. D.h. keine Verzeichnisse auf anderen gemounteten FS durchsuchen (z.B. ignoriert find / -xdev dann andere Partitionen).
- **-prune** – Überspringt (schließt vom Durchsuchen aus) ein Verzeichnis. Bsp: find . -path "./node\_modules" -prune -o -name "\*.js" -print durchsucht alles außer den node\_modules Ordner.

#### Aktionen (wenn Suchkriterien wahr sind):

- **-print** – Gibt gefundene Pfade aus (Standard, wird angenommen, wenn keine andere Aktion).
- **-ls** – Führt ls -dils auf die gefundene Datei aus (anzeige von detaillierten Infos, inkl. Inode, Größe, Rechte etc.).
- **-delete** – Löscht gefundene Dateien *unmittelbar*. Achtung: find -delete entfernt auch Verzeichnisse, wenn leer geworden. Sollte meist mit Kriterien kombiniert werden (z.B. -type f -delete nur Dateien). **Keine** Aktion mehr nach -delete wird ausgeführt (beendet Durchlauf in diesem Pfad).
- **-exec Befehl {} ;** – Führt einen externen Befehl für jede gefundene Datei aus. {} wird durch den Pfad ersetzt. Die Kommandozeile wird so oft ausgeführt wie Treffer. Beispiel: find . -name "\*.bak" -exec rm {} \; löscht alle .bak-Dateien (funktional wie -delete, aber universeller) file-5y1edouysrow4zj6phjcqm. Das \; markiert Ende des -exec Befehls.
- **-exec ... {} +** – Variante, die versucht, mehrere gefundene Dateien in *einen* Aufruf zu packen (fügt die Pfade an Stelle von {} möglichst gesammelt, wie xargs). Effizienter, falls das Programm mehrere auf einmal verarbeiten kann. Z.B. -exec chmod 644 {} + ändert die Rechte aller gefundenen Files, ggf. in batched Calls.

- **-execdir** – Wie -exec, aber führt das Kommando im jeweiligen Verzeichnis der Datei aus (Working Directory wechselt). Erhöht Sicherheit bei {} mit {} containing spaces.

### **Beispiele:**

- find /var/log -type f -name "\*.log" -size +1M -mtime +30 – findet Log-Dateien über 1MB, älter als 30 Tage unter /var/log.
- find . -maxdepth 1 -type d ! -name ":" -exec rm -r {} \; – löscht alle Unterverzeichnisse im aktuellen Ordner (maxdepth 1, alles was Type d ist und Name nicht ":"), dann rm -r).
- find /home/user -xtype l -delete – löscht gebrochene Symlinks (xtype l findet Links, deren Ziel nicht existiert).
- find . -perm /u+x -print – findet Dateien, bei denen das User-execute Bit gesetzt ist (also alle ausführbaren Dateien).
- find . -name ".\*.tmp" -execdir gzip {} \; – packt alle \*.tmp im jeweiligen Verzeichnis in eine .tmp.gz (ausgeführt im Ort der Datei).

## **locate (Datenbank-Dateisuche)**

**Beschreibung:** Durchsucht eine vorgefertigte Index-Datenbank nach Dateinamen (Pfaden), die einem angegebenen Suchstring entsprechenfile-4fdk7rvx39azjretinarqb. locate ist viel schneller als find, da es nicht live das Dateisystem scannt, sondern auf einer regelmäßig aktualisierten Datenbank (typisch via updatedb in Cron) basiert. Nachteil: findet keine erst sehr kürzlich hinzugefügten Dateien (bis zum nächsten updatedb-Lauf) und standardmäßig nur solche, auf die der User zugreifen darf.

### **Syntax:**

locate [Optionen] <Suchmuster>

*Das Suchmuster ist im Grunde ein Substring oder einfache Glob. Standardmäßig sucht locate alle Datenbankeinträge, die <Suchmuster> als Teilstring enthalten.*

### **Optionen:**

<b>Option</b>	<b>Beschreibung</b>
<b>-i, --ignore-case</b>	Sucht case-insensitive (Groß/Kleinschreibung ignorieren)file-4fdk7rvx39azjretinarqb.
<b>-r, --regex</b>	Interpretiert das angegebene Muster als regulären Ausdruck (POSIX ERE) statt als einfachen Textfile-4fdk7rvx39azjretinarqb.

Option	Beschreibung
<b>-w, --wholename</b>	(Standardverhalten) Sucht im gesamten Pfadnamen. (Bei einigen Implementierungen war locate nur substring-suche, wholename explizit wie default, hier nur Info.)
<b>-b, --basename</b>	Sucht nur im Dateinamen selbst, nicht im gesamten Pfad. (Also ignoriert Verzeichnisnamen im Match.)
<b>-c, --count</b>	Zählt nur die Anzahl der Fundstellen und gibt diese Zahl aus, anstatt alle Ergebnisse zu listen.
<b>-l N, --limit=N</b>	Gibt maximal N Ergebnisse aus (dann Abbruch).
<b>--existing</b>	Prüft gefundene Pfade und filtert jene heraus, die aktuell nicht (mehr) existieren. (Die DB könnte veraltete Einträge haben; mit --existing werden nur noch valide Treffer angezeigt, obwohl es Zeit kostet, jeden Pfad zu verifizieren.)
<b>-0, --null</b>	Trennt die Ausgabe mit Nullbytes anstatt neuer Zeile. Nützlich, um in xargs -0 weiterzuverarbeiten.
<b>-e, --nofollow</b>	Ignoriert Einträge, die Symlinks auf nicht-existente Dateien sind (könnte Implementation-spezifisch).
<b>-S</b>	Zeigt Statistik der Datenbank (Anzahl Einträge etc.) statt zu suchen.
<b>-d DB, --database=DB</b>	Nutzt eine alternative Datenbank-Datei anstelle der Standard-Datenbanken (/var/lib/mlocate.db z.B.). Man kann mehrere via -d db1:db2:... angeben.
<b>--help, --version</b>	Hilfe bzw. Version.

*Beispiel:* locate -i '\.pdf' listet alle Pfade, die ".pdf" (unabhängig von Großschreibung) enthalten – das würde i.d.R. alle PDF-Dateien auf dem System listen, die zum Zeitpunkt der letzten Datenbankaktualisierung existierten. Oft benutzt: locate <Programmname> um herauszufinden, wo ein Programm oder dessen Konfigdateien liegen.

### **updatedb (Update Database for locate)**

**Beschreibung:** Baut die Datei-Datenbank für locate neu auf. Es durchsucht das Dateisystem (oder definierte Pfade) und speichert alle Dateinamen/Pfade in einer komprimierten Datenbank (typischerweise /var/lib/mlocate/mlocate.db). updatedb wird meist regelmäßig (täglich) via Cron ausgeführt, man kann es aber auch manuell starten, z.B. nach größeren Änderungen im FS, damit locate sofort aktuell ist.

**Syntax:**

updatedb [Optionen]

**Optionen:** (Häufig werden diese über /etc/updatedb.conf voreingestellt)

Option	Beschreibung
--prune-bind-mounts {yes no}	Ob Bind-Mountpoints ausgelassen werden (Default ja – doppelte Scans vermeiden).
--prune- paths='PFADLISTE'	Doppelpunkt-separierte Liste von Verzeichnissen, die <i>nicht</i> indexiert werden sollenfile-4fdk7rvx39azjretinarqb. Standardmäßig z.B. /tmp, /proc, /sys, /run, /dev, /var/cache etc., um unnötiges zu sparen. Dies entspricht PRUNEPATHS in /etc/updatedb.conf.
--prune- filesystems='TypList'	Liste von Dateisystemtypen, die nicht durchsucht werden (z.B. nfs,squashfs etc.). Standard meist Netzwerk-FS, Pseudo-FS (proc, sysfs) etc.file-4fdk7rvx39azjretinarqb. Entspricht PRUNEFS.
-l N	--localpaths=LIST – falls man nur lokale Pfade indexieren will (veraltet, use prune).
-U 'PFADLISTE'	Nur diese durch Doppelpunkt getrennten Pfade als Wurzel für Index verwenden (Standard "/" – gesamtes FS). Kann man nutzen um z.B. nur /home zu indexieren.
-o DBFILE	Output: speichere DB in angegebener Datei statt Standard. (So kann man separate DBs erzeugen.)
-f 'FS-Liste'	Nur diese Dateisystemtypen berücksichtigen (Inverse von PRUNEFS).
-v	Ausführliche Ausgabe während des Indizierens.

In der Regel wird updatedb ohne Optionen (ggf. als root über sudo updatedb) ausgeführt und nutzt die System-Voreinstellungen. Danach steht die Datenbank locate zur Verfügung.

### which (Welches Kommando - Pfadsuche)

**Beschreibung:** Zeigt den Pfad zu einem ausführbaren Programm, das in der Umgebungsvariablen \$PATH gefunden wirdfile-4fdk7rvx39azjretinarqb. Damit kann man herausfinden, welches Programm aufgerufen würde, wenn man einen bestimmten Befehl tippt (nützlich bei mehreren Versionen in verschiedenen Verzeichnissen).

**Syntax:**

which [Optionen] Befehlsname...

#### Optionen:

Option	Beschreibung
<b>-a</b>	Zeigt alle Vorkommen des angegebenen Befehls im Suchpfad an (nicht nur das erste gefundene) file-4fdk7rvx39azjretinarqb. Z.B. which -a python listet evtl. /usr/bin/python und /usr/local/bin/python falls beide existieren.
<b>--skip-dot</b> , <b>--skip-tilde</b>	Ignoriert . bzw. ~ am Anfang des Pfades (also aktuelles Verzeichnis oder Home in PATH) in der Suche. (Sicherheitsaspekt – Standard bei which).
<b>--show-dot</b> , <b>--show-tilde</b>	Zeigt explizit, wenn gefundene Einträge in . oder ~ liegen, durch Präfix "./" oder "~/".
<b>--tty-only</b>	Zeigt nur etwas an, wenn stdout ein Terminal ist. Unterdrückt z.B. Ausgabe, wenn in Pipe (verhindert sich einzumischen).
<b>--help</b> , <b>--version</b>	Hilfe bzw. Versionsinfo (gilt bei GNU which; einige shells haben eingebautes which mit anderem Verhalten).

*Hinweis:* which ist oft eine Shell-Funktion/Alias und verhält sich u.U. leicht unterschiedlich. Ähnliche Tools: type (shell-builtin, zeigt auch Aliases, Keywords), command -v.

**Beispiel:** which gcc -> /usr/bin/gcc (Pfad zum C-Compiler). which -a ls könnte z.B. zeigen /usr/bin/ls und /bin/ls falls beides symlinks aufeinander oder verschieden, etc.

### Textverarbeitung und Streams (Shell, Scripting)

#### awk (Muster/Aktions-Scanner)

**Beschreibung:** AWK ist ein mächtiges Textverarbeitungswerkzeug und Skriptsprache, mit dem man Textdateien zeilenweise nach Mustern durchsuchen und pro Zeile Aktionen ausführen kann file-4fdk7rvx39azjretinarqb. Typischerweise wird awk benutzt, um Spalten auszugeben oder zu filtern, da es Eingabezeilen automatisch in Felder zerlegt. Der Name stammt von den Anfangsbuchstaben seiner Schöpfer (Aho, Weinberger, Kernighan).

#### Syntax:

awk [Optionen] 'Muster { Aktion }' Datei...

Jede Zeile der Datei wird gelesen; wenn das *Muster* passt, wird die *Aktion* ausgeführt (z.B. etwas ausgeben). AWK hat eigene Variablen (\$1 = erstes Feld, \$0 = ganze Zeile) und eingebaute Rechen- und String-Features.

#### Optionen:

<b>Option</b>	<b>Beschreibung</b>
<b>-F separator</b>	Legt das Eingabefeld-Trennzeichen fest (Field Separator). Standard sind whitespace (Leerzeichen/Tabs)file-4fdk7rvx39azjretinarqb. Z.B. -F ":" für /etc/passwd mit Doppelpunkten.
<b>-v Var=Wert</b>	Setzt eine AWK-Variable <i>Var</i> auf <i>Wert</i> vor dem Verarbeiten. Z.B. -v ORS="\n\n" um eine Leerzeile nach jeder Ausgabezeile einzufügen (ORS = Output Record Separator).
<b>-f Scriptdatei</b>	Lies das AWK-Programm aus einer Datei anstatt aus der Befehlszeile. Damit kann man größere AWK-Skripte in eigenem File verwalten.
<b>-m, --sandbox</b>	(gawk) AWK in sicheren Modus schalten (kein System(), keine Schreibrechte auf FS etc).
<b>--posix</b>	Schaltet in POSIX-kompatiblen Modus (ggf. einige GNU-Erweiterungen aus).
<b>-W version</b>	Zeigt AWK-Version an (für gawk).
<b>--help, --version</b>	Hilfe / Version.

### **AWK Sprachkonzepte (Kurz):**

- Ein AWK-Programm besteht aus *Muster { Aktion }* Blöcken. Muster ist ein Regex oder Ausdruck, Aktion ist Code in AWK.
- Felder: Eingabezeile wird in Felder zerlegt, zugreifbar als \$1, \$2, ..., \$NF (NF=Anzahl Felder). \$0 ist die ganze Zeile.
- Beispiele:
  - awk '{ print \$1 }' file – gibt die erste Spalte jeder Zeile aus (mit Standard-Trenn whitespace).
  - awk '/error/ {print NR ":" "\$0}' log.txt – sucht Zeilen, die "error" enthalten (Regex als Muster), und druckt Zeilennummer und kompletten Inhalt.
  - awk -F ',' '\$3 > 100 { sum += \$3 } END { print "Summe:", sum }' data.csv – mit Komma als Separator summiere Wert der 3. Spalte aller Zeilen >100 in Spalte3; am Ende (END-Block) gib Summe aus.
- AWK hat viele eingebaute Funktionen (Mathe, String), assoziative Arrays, Kontrollstrukturen (if, for, while).

### **sed (Stream Editor)**

**Beschreibung:** sed ist ein nicht-interaktiver Stream-Editor zum Bearbeiten von Texteingaben nach Mustern, insbesondere für Suchen&Ersetzen oder das Extrahieren von Zeilenfile-4fdk7rvx39azjretinarqb. sed liest zeilenweise Eingabe (Dateien oder STDIN), wendet die angegebenen *Editier-Befehle* darauf an und schreibt das Ergebnis auf STDOUT (oder optional in-place zurück in Dateien). Am verbreitetsten ist die *Substitution* mit dem s/alt/neu/ Kommando.

#### Syntax:

sed [Optionen] 'Script' Datei...

Das *Script* besteht aus einem oder mehreren Editierbefehlen, die auf die Eingabe angewendet werden. Beliebte sed-Befehle: **s** (substitute), **p** (print), **d** (delete), **q** (quit), **{}** für Gruppen, Adressierungsmodi (Zeilen-Nr oder Regex).

#### Optionen:

Option	Beschreibung
<b>-n, --quiet/--silent</b>	Suppress normal output. Normal druckt sed jede Eingabezeile (ggf. modifiziert) automatisch. Mit -n wird nichts automatisch gedruckt, nur explizite p (print) Befehle geben ausfile-4fdk7rvx39azjretinarqb. Typisch in Kombination mit p in Script, um nur bestimmte Zeilen auszugeben.
<b>-e Script</b>	Gibt das zu verwendende sed-Skript (Editierbefehle) direkt an. Kann mehrfach genutzt werden für mehrere Befehle in Reihenfolge. Wenn nur ein Skript-Argument ohne -e gegeben ist, kann -e weggelassen werden.
<b>-f Datei</b>	Lies das sed-Skript aus einer Datei. Erlaubt komplexere Edits aus einem File anstatt Kommandozeile.
<b>-i[SUF], --in-place[=SUF]</b>	Bearbeitet Dateien <i>in place</i> , d.h. schreibt Änderungen direkt zurück in die Eingabedateien. Optional kann ein <i>Suffix</i> angegeben werden, um vor Änderungen ein Backup der Originaldatei anzulegen (z.B. -i.bak). Ohne Suffix wird die Datei direkt überschrieben <a href="http://ionos.de">ionos.de</a> .
<b>-r, -E</b>	Schaltet die Verwendung erweiterter regulärer Ausdrücke ein (so dass in Patterns +, ?, {} etc. ohne Backslash verwendet werden können). Bei GNU sed ist -r (old syntax) und -E (wie bei BSD sed) identisch.
<b>-s</b>	Treat files separately. Normalerweise bei mehreren Dateien behandelt sed den Input als einen langen Stream. Mit -s wird nach jeder Datei der interne Zustand (z.B. Zeilennummer, \$-Regex-Flag) zurückgesetzt.
<b>-z, --null-data</b>	Input wird als Null-terminiert (Zeilen getrennt durch ASCII NUL statt newline) angenommen. Ermöglicht Verarbeitung von Binär oder

Option	Beschreibung
	multiline-chunks. Entspricht dann, dass . nicht auch \n matcht und \$ nur am NUL erkennt. (GNU sed)
<b>--follow-symlinks</b>	Wenn in-place auf Symlinks angewendet, ändert sed standardmäßig den Link selbst (also das Zielfile, aber eben durch den Link?). Mit dieser Option folgt sed dem Symlink und editiert die Ziel-Datei. (GNU sed)
<b>--sandbox</b>	Verhindert, dass sed w oder s///w zum Schreiben verwendet oder ! Shell-Aufrufe macht. (GNU sed 4.4+)
<b>--help, --version</b>	Hilfe / Version.

### Häufige sed-Anwendungen:

- **Substitution (Ersetzen):**  
Syntax: s/REGEX/REPLACEMENT/[Flags]. Ersetzt das erste Vorkommen von REGEX in jeder Zeile durch REPLACEMENT. Flags: g = global (alle Vorkommen in Zeile ersetzen), i = case-insensitive, p = print (selbst wenn -n), w file = gefundene Zeilen nach file schreiben.  
Beispiel: sed -i 's/oldword/newword/g' file.txt – ersetzt *oldword* durch *newword* überall in file.txt, direkt in der Datei file-5y1edouysrow4zj6phjcqm.  
Besonderheit: Im Ersatz kann man & verwenden für den gesamten Match oder \1 ... für Capturing Groups aus dem Regex.
- **Zeilen löschen:**  
z.B. sed '/^#/ d' config – löscht alle Zeilen, die mit # beginnen (d = delete). Oder sed '5,10d' file löscht Zeilen 5 bis 10.
- **Nur bestimmte Zeilen ausgeben:**  
Mit -n und p. Z.B. sed -n '1,5p' file druckt nur Zeilen 1-5 (andere werden durch -n nicht ausgegeben). Oder sed -n '/ERROR/p' log druckt nur Zeilen mit "ERROR".
- **Mehrere Edits:**  
z.B. sed -e 's/foo/bar/' -e '/BAZ/d' – erst foo->bar ersetzen, dann Zeilen mit "BAZ" löschen. (Die Reihenfolge ist wichtig).
- **In-Place Backup:**  
sed -i.orig 's/^#/.\*conf' – kommentiert alle Zeilen in \*.conf aus, speichert vorher Backup mit .orig-Suffix.

Sed kann viel mehr (Insert/Append mit i\ und a\, Transform mit y///, Shell-Befehle mit !, Holding Space etc.), was in Tiefe für Prüfung wohl nicht nötig ist. Wichtig ist Suchen&Ersetzen und Basisfilter.

## **tee (Output verdoppeln)**

**Beschreibung:** Liest die Standard-Eingabe und schreibt sie *gleichzeitig* auf die Standard-Ausgabe **und** in eine oder mehrere Dateienfile-4fdk7rvx39azjretinarqb. Es "verzweigt" den Datenstrom (daher der Name T-Stück). So kann man die Ausgabe eines Kommandos in eine Datei umleiten, aber trotzdem im Terminal sehen.

### **Syntax:**

... | tee [Optionen] <Datei>...

### **Optionen:**

Option	Beschreibung
<b>-a, --append</b>	Hängt die Ausgabe an bestehende Dateien an, anstatt sie zu überschreibenfile-4fdk7rvx39azjretinarqb. Standard ohne -a überschreibt Ziel-Dateien neu.
<b>-i, --ignore-interrupts</b>	Ignoriert Interrupt-Signale (SIGINT). D.h. Ctrl+C wirkt nicht auf tee selbst (nur auf vorheriges Kommando?), damit tee garantiert die Daten schreibt.
<b>--output-error MODE</b>	Verhaltensweise falls ein Schreibfehler auftritt: warn (Fehler melden, aber weiter versuchen andere outputs zu schreiben), exit (sofort beenden, evtl. ungeschriebene outputs gehen verloren). Standard ist warn.

### **Anwendung:**

- make 2>&1 | tee build.log – Zeigt alle Ausgaben von make im Terminal und speichert sie gleichzeitig in *build.log*.
- ls -l | tee -a listing.txt – Ausgabe von ls -l normal anzeigen und ans Ende von listing.txt anfügen.
- echo "strict mode on" | sudo tee /proc/sys/kernel/yama/ptrace\_scope – Durch sudo tee kann man Inhalte in Dateien schreiben, für die man Rechte braucht, während Eingabe evtl. aus einem un-privilegierten Echo kommt.

Tee ist also nützlich bei Logging und bei Verwendung mit sudo zum in Dateien schreiben, da sudo echo >> file wegen Redirection sonst nicht klappt.

## **xargs (Execute Arguments)**

**Beschreibung:** Liest Eingaben (Standardinput oder Datei) und führt daraus Kommandoaufrufe zusammengebaut aus. Typisch, um die Ausgabe eines Programms als Argumentliste für ein anderes Programm zu verwendenfile-4fdk7rvx39azjretinarqbfile-4fdk7rvx39azjretinarqb. xargs nimmt zeilen- oder spaltengetrennte Wörter und packt möglichst viele davon in einen Aufruf eines

angegebenen Kommandos (Standard: echo). Damit überwindet man die Shell-Längenbeschränkung und kann sequentiell Befehle auf Listen anwenden.

### Syntax:

xargs [Optionen] [Befehl [Initial-Argumente]]

Wenn kein Befehl angegeben ist, benutzt xargs per Default echo (einfaches Ausgeben).

### Optionen:

Option	Beschreibung
<b>-n MaxArgs</b>	Maximal <i>MaxArgs</i> Argumente pro aufgerufenem Kommando verwenden. Z.B. -n 1 bedeutet, xargs nimmt jeweils 1 Argument und führt das Kommando damit aus (also für jede Eingabezeile einmal). Ohne Angabe bündelt xargs so viele Argumente wie möglich pro Aufruf, bis die Systemlimite (Getconf ARG_MAX) fast erreicht ist.
<b>-L MaxLines</b>	Nimmt maximal <i>MaxLines</i> Input-Zeilen pro Kommando-Aufruf. Z.B. -L 1 bewirkt, dass jede einzelne Zeile als eine xargs-Ausführung behandelt wird (selbst wenn Zeile mehrere Wörter enthält).
<b>-d Delim</b>	Verwendet <i>Delim</i> als Trennzeichen anstelle von Whitespace oder newline. Kann auch mehrere Zeichen (String) sein (GNU). Beispiel: -d ':' um : als Separator zu nehmen.
<b>-0, --null</b>	Erwartet Input-Felder durch Nullbyte getrennt (wie von find -print0 oder grep -Z erzeugt). Dies ermöglicht sichere Übergabe von Dateinamen mit Leer-/Sonderzeichen <a href="http://stackoverflow.com">stackoverflow.com</a> .
<b>-E EOFSTR</b>	Definiert eine Zeichenfolge, die als Ende der Eingabe interpretiert wird. Wenn xargs diese Zeichenfolge liest, bricht es die weitere Verarbeitung ab (auch wenn nicht EOF).
<b>-e</b>	(veraltet, äquivalent zu -E ohne Parameter) – historically to specify EOF marker, oft nicht mehr benötigt.
<b>-I ReplaceStr</b>	Nutzt <i>ReplaceStr</i> als Platzhalter in dem auszuführenden Kommando, der durch den gesamten Input (pro Aufruf) ersetzt wird. file- Z.B. xargs -I {} mv {} /backup nimmt jeden Input (eine Datei) und substituiert anstelle von {} im mv-Aufruf. Bei -I wird implizit -L 1 angenommen (d.h. pro Zeile ein Aufruf).
<b>-J ReplaceStr</b>	(BSD xargs) Ähnlich -I, aber ersetzt nur einmal in der Kommandozeile (kann mit -n kombinieren). In GNU xargs gibt es -J nicht.

Option	Beschreibung
<b>-t, --verbose</b>	Gibt das Kommando, das ausgeführt wird, vorher auf stderr aus (also "echo" what is being done). Zum Debuggen hilfreich.
<b>-p, --interactive</b>	Fragt vor <i>jedem</i> Kommando nach Bestätigung (y/n), ähnlich wie -t zeigt es Befehl. Bei 'y' ausführen, 'n' überspringen.
<b>-r, --no-run-if-empty</b>	Führt das Kommando <b>nicht</b> aus, wenn keine Argumente aus Input entstehen. Standard bei GNU xargs ist, <i>kein</i> Leerlaufaufruf zu machen (sprich diese Option ist implizit an). In POSIX könnte Standard anderes sein.
<b>-0, --null</b>	(siehe oben) Null-terminierte Eingabe verwenden – wichtig in Kombination mit find -print0 um sicher Dateilisten zu pipen ohne Probleme mit Sonderzeichen <a href="http://stackoverflow.com">stackoverflow.com</a> .
<b>--max-chars=K, -s K</b>	Maximal K Zeichen pro Kommandozeilen-Aufruf verwenden (inklusive Kommando und env). Standard an System ARG_MAX orientiert (oft ~131072). Manchmal zu reduzieren bei seltsamen Umgebungen.
<b>--max-procs=N, -P N</b>	Bis zu N Prozesse <i>parallel</i> ausführen statt nacheinander. Z.B. -P 4 führt bis zu 4 Instanzen des Kommandos gleichzeitig aus, um Parallelisierung zu erreichen. N=0 bedeutet unbegrenzt viele parallel (Vorsicht!). Hilfreich z.B. in Multi-Core-Systemen.
<b>--process-slot-var=NAME</b>	Setzt eine Umgebungsvariable \$NAME in den aufgerufenen Prozessen, die eindeutige ID des xargs-Slots enthält (0..N-1) – bei -P parallel. Hilft um Ausgaben zuzuordnen.
<b>-a Datei</b>	Liest Input aus <i>Datei</i> anstatt STDIN. So kann xargs auch ohne Pipe aus File speisen.
<b>--help, --version</b>	Hilfe/Version.

### Beispiele:

- find /tmp -type f -name '\*.txt' -print0 | xargs -0 rm -f – Löscht alle .txt-Dateien unter /tmp (sicher gegenüber Sonderzeichen, dank -print0 und -0).
- grep -l "ERROR" \*.log | xargs -r zip errors.zip – Sucht in allen .log nach "ERROR", die -l Option listet nur Dateinamen mit Treffer. Diese Liste wird an xargs übergeben, das dann zip errors.zip <files> ausführt. Falls kein Treffer, -r sorgt dafür, dass zip nicht ohne Argument aufgerufen wird.

- `cat list.txt | xargs -I {} echo File: {} >> files.txt` – Für jede Zeile aus list.txt, führe `echo File: <line>` aus und append an files.txt. (So fügt man "File: " vor jede Zeile).
- `seq 1 1000 | xargs -n 100 > numbers.txt` – Nimmt die Zahlen 1 bis 1000 (jede auf neuer Zeile) und packt jeweils 100 Zahlen in eine Zeile, schreibt nach numbers.txt.
- `printf "%s\n" * | xargs -0 -n 1 -P 4 sha256sum` – Berechnet parallel (4 at a time) die SHA256 Summen aller Dateien im aktuellen Verzeichnis. `printf "%s\n" *` generiert Null-separierte Liste aller Dateien, xargs nimmt je 1 (-n1) und startet bis zu 4 sha256sum gleichzeitig.

## Software- und Paketmanagement

### **tar (Tape ARchiver)**

**Beschreibung:** Archiviert mehrere Dateien zu einer einzigen Datei (tarball). Tar kann Dateien **bündeln** (zu .tar Archiven) und in Zusammenarbeit mit Kompressionsprogrammen (gzip, bzip2, xz etc.) zugleich komprimierenfile-4fdk7rvx39azjretinarqb. Oft verwendet, um Verzeichnisse oder viele Dateien als ein Archiv zu speichern oder zu verteilen (.tar.gz, .tgz, .tar.bz2 etc.).

#### **Syntax:**

`tar [Optionen] -f <Archiv> [Dateien/Verzeichnisse...]`

Tar hat **Modi** (Mutual exclusive Optionen): Erstellen, Extrahieren, Listen etc. Üblich ist die kombinierte Kurzoption wie -czf oder -xvf. Wichtig: -f (file) gibt an, welches Archivfile zu benutzen ist, und kommt als *ein* Argument, meist zuletzt bei Short-Options-Kombination.

#### **Haupt-Modi:**

- **-c, --create** : Erstelle ein neues Archiv aus den **(Fortsetzung)**

### **tar (Tape Archiver)**

**Beschreibung:** Erstellt oder entpackt Archive aus mehreren Dateien. tar bündelt Dateien in **einem** Archiv (Containerdatei mit Endung .tarfile-4fdk7rvx39azjretinarqb】. Oft wird tar mit Kompressionsprogrammen kombiniert (z.B. gzip oder bzip2), um komprimierte Archive wie .tar.gz zu erstellefile-4fdk7rvx39azjretinarqb】.

#### **Syntax:**

`tar <Modus-Option> [Optionen] -f <Archivdatei> [Dateien/Verzeichnisse...]`

Tar benötigt **eine** der Hauptoptionen: -c (create), -x (extract) oder -t (list). Dazu wird **-f** angegeben, um den Namen der Archivdatei zu bestimmen. Häufig werden mehrere Kurzoptionen kombiniert (z.B. -czvf für "create, gzip, verbose, file").

## Wichtige Optionen:

- **-c – create:** Neues Archiv erstellefile-4fdk7rvx39azjretinarqb】 .
- **-x – extract:** Dateien aus Archiv entpacken.
- **-t – list:** Inhalt des Archivs auflisten (nicht extrahieren).
- **-f Datei** – Gibt den Archivnamen an (Pfad/Name der Tar-Datei). Diese Option **muss** direkt gefolgt von dem Dateinamen kommen (z.B. -f archive.tar).
- **-v – verbose:** Ausführliche Ausgabe (listet während der Verarbeitung die Dateinamen auf).
- **-z – Filter durch gzip:** Bei -c wird das Archiv mit gzip komprimiert (\*.tar.gz); bei -x wird erwartet, dass das Archiv gz-komprimiert isfile-4fdk7rvx39azjretinarqb】 . (Äquivalent: --gzip oder mit externem gzip pipen).
- **-j – Filter durch bzip2:** Erstellt/entpackt ein bzip2-komprimiertes Archiv (\*.tar.bz2). (Alternativ --bzip2.)
- **-J – Filter durch xz:** Für .tar.xz-Archive (--xz).
- **-A – concatenate:** Hängt ein oder mehrere Archive an ein bestehendes Archiv an.
- **-d – diff/compare:** Vergleicht Archiveinträge mit Dateien im Dateisystem (prüft, ob Unterschiede bestehen).
- **-C DIR – Change Directory:** Wechselt ins Verzeichnis *DIR*, bevor Dateien hinzugefügt oder extrahiert werden [freedesktop.org](#)】 . Z.B. tar -xvf archive.tar -C /tmp entpackt nach /tmp.
- **--strip-components=N** – Beim Extrahieren lässt tar N führende Pfadkomponenten weg. Z.B. entpackt /usr/local/share/doc/file als file bei --strip-components=3. Praktisch, um Archiv ohne Top-Level-Ordner auszupacken.
- **--exclude=PATTERN** – Schließt Dateien, die auf PATTERN passen, vom Archivieren aus. Z.B. --exclude="\*.o".
- **--wildcards, --anchored** – steuert, wie Muster bei -t oder -x --wildcards -f archive ".\*.txt" interpretiert werden (glob-Muster an/aus etc.).
- **--numeric-owner** – Beim Entpacken Besitzer/Gruppen als IDs übernehmen (statt Namen auf dem Zielsystem zuzuordnen).

## Beispiele:

- Archiv erstellen: tar -cvf archive.tar Verzeichnis/ erzeugt ein unkomprimiertes Archiv archive.tar mit allen Dateien aus Verzeichnis/.

- Mit Kompression: tar -czvf etc.tar.gz /etc packt ganz /etc als gzip-komprimiertes Archiv.
- Entpacken: tar -xvzf backup.tar.gz -C /restore entpackt *backup.tar.gz* nach */restore*.
- Auflisten: tar -tvf archive.tar zeigt die Dateiliste mit Größen, Timestamp, Rechten etc.

## **gzip (GNU Zip Compression)**

**Beschreibung:** Komprimiert oder dekomprimiert Dateien im **gzip-Format** (.gz). Es ersetzt beim Komprimieren standardmäßig die Originaldatei durch die komprimierte Version (und fügt .gz anfile-4fdk7rvx39azjretinarqb】). Gzip ist schneller, aber erzeugt etwas größere Dateien als bzip2/xz. Häufig mit tar (siehe oben) eingesetzt.

### **Syntax:**

gzip [Optionen] [Datei...]

gunzip [Optionen] [Datei...]

(gunzip ist äquivalent zu gzip -d.)

### **Optionen:**

- **-c, --stdout:** Schreibt das komprimierte Ergebnis auf die Standardausgabe, anstatt die Datei zu ersetzen. Original bleibt erhalten (kann in Pipeline genutzt werden[ionos.de](#)】).
- **-d, --decompress:** Dekomprimiert die .gz-Datei (auch via gunzip aufrufbar[ionos.de](#)】).
- **-k, --keep:** Behält Originaldatei beim Komprimieren/Entpacken (standardmäßig wird Original gelöscht).
- **-r, --recursive:** Komprimiert alle Dateien in angegebenen Verzeichnissen rekursiv.
- **-1 bis -9:** Einstellen der Kompressionsstufe (1 = schnellste/geringste Kompression, 9 = langsamste/beste). Standard ist 6.
- **-t, --test:** Prüft eine gz-Datei auf Integrität, ohne sie zu entpacken (exit-Status zeigt OK/NOK).
- **-l, --list:** Zeigt Informationen über eine komprimierte Datei: Originalgröße, komprimierte Größe, Kompressionsrate, Zeitstempel.
- **--stdout, -c (nochmal):** Wird oft mit -d kombiniert, um nach stdout zu entpacken (z.B. gzip -dc file.gz).

- **-N, --name:** Bewahrt ursprünglichen Dateinamen und Zeitstempel (in .gz-Header) oder stellt diese beim Entpacken wieder her (ist Standardverhalten bei GNU gzip).
- **--help, --version:** Hilfe bzw. Versionsinfo.

#### **Beispiele:**

- gzip bigfile komprimiert *bigfile* zu *bigfile.gz* (und löscht *bigfile*).
- gzip -9 \*.txt erstellt maximal komprimierte \*.txt.gz Dateien.
- gunzip data.csv.gz entpackt *data.csv.gz* zurück zu *data.csv*.
- gzip -c file.txt > file.txt.gz komprimiert nach stdout und schreibt in separate Datei (Original bleibt).
- In Pipe: tar -cvf - /folder | gzip > folder.tar.gz – tar erzeugt Archiv auf stdout, gzip komprimiert es zu *folder.tar.gz*.

### **xz (LZMA Compressor)**

**Beschreibung:** Komprimiert/Dekomprimiert Dateien mit dem **xz/LZMA2-Algorithmus**. XZ erreicht hohe Kompressionsraten, aber benötigt mehr CPU und RAM. Dateiendung: *.xz*. Wird z.B. für Kernel-Quellen-Archive genutzt. Syntax und Optionen ähneln gzip.

#### **Syntax:**

`xz [Optionen] [Datei...]`

`unxz [Optionen] [Datei...]`

(`xz -d` entspricht `unxz`.)

#### **Wichtige Optionen:**

- **-z (Default):** Komprimieren (Standardmodus, kann entfallen).
- **-d, --decompress:** Entpacken.
- **-k, --keep:** Originaldatei behalten (nicht löschen).
- **-f, --force:** Vorhandene .xz-Dateien überschreiben, falls vorhanden; oder erzwingt Komprimieren auch bei scheinbar nicht sinnvoll (z.B. sehr kleine Datei).
- **-9 (auch -0 .. -9):** Kompressionsgrad (0 = none, 6 = Standard, 9 = max). XZ erlaubt auch -e für extreme Mode (noch etwas bessere Kompression, deutlich langsamere Kodierung).
- **-T N, --threads=N:** Nutzung von N Threads für Kompression (für große Dateien auf Multi-Core). Standard 1. (Für Dekomprimierung wird threads automatisch benutzt).

- **-c, --stdout:** Ergebnis auf stdout schreiben (bzw. von stdin lesen und auf stdout entpacken).
- **-l, --list:** Listet das .xz-Archiv (Inhalt, ratio, etc. – bei Single-File Komprimierung liefert original und comp size).
- **-q, -v:** quiet bzw. verbose Mode.
- **-C, --check=TYPE:** Prüfsummenverfahren wählen (Standard CRC64). Kann z.B. auf CRC32, SHA256 gestellt werden.

### **Beispiele:**

- xz hugefile erzeugt *hugefile.xz* (Original wird gelöscht).
- xz -dk archive.tar.xz entpackt *archive.tar.xz* nach *archive.tar* und behält auch die .xz.
- tar -c /usr/bin | xz -9e > bin.tar.xz erstellt maximal-komprimiertes tar-Archiv.
- unxz data.csv.xz oder xz -d data.csv.xz entpackt Datei.
- xzcat file.txt.xz = xz -dc file.txt.xz entpackt nach stdout (ähnlich zcat).

### **bzip2 (Burrows–Wheeler Compressor)**

**Beschreibung:** Komprimiert mit dem älteren bzip2-Algorithmus (.bz2). Langsamer als gzip, aber bessere Kompression; heute oft von xz abgelöst. Bedienung ähnlich gzip.

**Syntax:** bzip2 [Optionen] [Datei...] – bunzip2 zum Entpacken.

**Optionen (Auswahl):** -k (keep file), -d (decompress), -1 .. -9 (Stufe, Standard 9 bei bzip2), -c (stdout), -t (Test), -v (verbose stats).

### **zip (Archiv mit Kompression, ZIP-Format)**

**Beschreibung:** Erstellt ZIP-Archive (mit Kompression) oder fügt Dateien hinzu. Anders als tar erwartet zip direkt die zu archiviernden Dateien und komprimiert sie einzeln ins Archiv. ZIP-Archive enthalten Verzeichnisstruktur + komprimierte Dateien (beliebtes Format unter Windows). unzip entpackt sie.

#### **Syntax:**

zip [Optionen] <Archiv.zip> <Dateien...>

#### **Wichtige Optionen:**

- **-r** – Rekursiv Verzeichnisse einschließen. Z.B. zip -r archiv.zip Ordner/.
- **-9** (bzw. -1 .. -9) – Kompressionsgrad (9 = beste, Standard meist 6).
- **-q** – quiet, **-v** – verbose (gibt Kompressionsraten aus).

- **-m – move:** Dateien nach dem Hinzufügen ins Archiv löschen (verschieben).
- **-j – junk paths:** Pfadnamen ignorieren, legt alle Dateien „flach“ im Archiv ab (ohne Verzeichnisstruktur).
- **-x <Pattern>** – bestimmte Dateien ausschließen (z.B. -x "\*.o").
- **-F** – Versucht ein beschädigtes Archiv zu reparieren (freshen).
- **-u – update:** nur neuere Dateien (bzgl. Zeitstempel) ins Archiv aktualisieren.
- **-g – grow:** hängt Dateien an ein bestehendes ZIP an (statt neu zu erstellen).
- **-e** – erstellt ein verschlüsseltes Archiv (fordert Passwort-Eingabe, traditionelles ZipCrypto; -P <pw> unsicher ohne Prompt).

### **unzip Wichtige Optionen:**

unzip <file.zip> entpackt im aktuellen Verzeichnis. -d <DIR> entpackt ins DIR. -l listet Inhalte. -t testet. -x schließt beim Entpacken bestimmte aus. -n überschreibt nichts, -o überschreibt ohne Nachfrage.

## **Paketmanagement-Befehle (Distributionen)**

### **rpm (RPM Package Manager)**

**Beschreibung:** Das Low-Level-Werkzeug zum Installieren, Entfernen und Abfragen von RPM-Paketen (Red Hat, CentOS, Fedora, SUSE etc.) . rpm arbeitet auf .rpm-Dateien und der lokalen Paketdatenbank. Häufig nutzt man für höhere Ebene yum oder dnf, aber rpm bleibt wichtig für direkte Paketoperationen.

### **Syntax (Basis):**

rpm [Optionen] <Paketdatei> / rpm [Optionen] <Paketname>

### **Operationen (Hauptoptionen, nicht kombinierbar):**

- **-i – install** ein neues RPM-Paket . Beispiel: rpm -i nginx.rpm. Erfordert Root-Rechte.
- **-U – upgrade** (installiert neu oder aktualisiert älteres Paket). Wird häufig anstelle -i benutzt, da es sowohl installiert als auch updatet.
- **-F – freshen:** wie -U, aber nur wenn schon ältere Version installiert ist (sonst kein Install).
- **-e – erase** (deinstalliert) ein installiertes Paket nach Paketname . Beispiel: rpm -e nginx.
- **-q – query:** Abfrage-Modus (ohne Änderungen) . Kann mit Zusätzen Info liefern. Beispiel: rpm -q nginx prüft, ob Paket installiert ist und welche Version.

### **Wichtige Optionen/Zusätze:**

## Install/Upgrade Optionen:

- **-h** – zeigt Hash-Fortschriffs Balken beim Installieren ("#" -Zeichen).
- **-v** – verbose, mehr Ausgabe. Oft kombiniert -Uvh oder -ivh.
- **--force** – erzwingt Installation, selbst wenn z.B. ein Paket bereits installiert ist oder Dateien konfigurieren. (Vorsicht!).
- **--nodeps** – ignoriert Abhängigkeits-Prüfung (installiert trotzdem).
- **--replacefiles** – überschreibt Dateien, die zu anderen Paketen gehören könnten (Konflikte ignorieren).
- **--noscripts** – führt keine Pre/Post-Install-Skripte im RPM aus (für spezielle Fälle).

## Query Optionen (mit -q):

- **-a** – query all (listet alle installierten Pakete).
- **-i** – (mit -q) zeigt detaillierte Paketinformationen (Name, Version, Summary, Größe, Install-Datum etc. [linux-praxis.de](http://linux-praxis.de)) . Z.B. rpm -qi bash.
- **-l** – listet alle vom Paket installierten Dateien. ` rpm -ql <paket> file-5y1edouysrow4zj6phjcqm` .
- **-p** – query ein *Paketfile* (nicht installiert). Z.B. rpm -qp -l paket.rpm zeigt, welche Dateien im RPM drin sind.
- **-f Dateipfad** – findet heraus, zu welchem installierten Paket eine bestimmte Datei gehört. Z.B. rpm -qf /usr/bin/ls -> coreutils-8.xx.
- **-c** – listet nur Konfigurationsdateien des Pakets (mit -ql kombinierbar: -qlc).
- **--changelog** – zeigt das Changelog des Pakets (sofern im RPM hinterlegt).
- **--provides, --requires** – zeigen welche Provides/Requires (Abhängigkeiten) ein Paket liefert bzw. benötigt. Z.B. rpm -q --requires httpd.
- **-s** – zeigt den Status der Dateien (ob sie verändert wurden seit Installation, (vom user modifiziert = **M**, missing etc.)). rpm -qs <paket>.

## Erase (Uninstall) Optionen:

- **--nodeps** beim -e: ignoriert Abhängigkeitsprobleme (entfernt auch wenn andere Pakete darauf angewiesen sind - kann zu kaputten Programmen führen).

## Beispiele:

- rpm -Uvh httpd-2.4.rpm – installiert Apache httpd, zeigt Hashtags und Namen.
- rpm -qa | grep kernel – listet alle installierten Pakete, filtert "kernel".

- rpm -qpi package.rpm – zeigt Infos über nicht installiertes RPM.
- rpm -e httpd – entfernt Apache-Paket.

## yum (Yellowdog Updater Modified)

**Beschreibung:** Ein High-Level-Paketmanager für RPM-basierte Distr. (v.a. CentOS/RHEL 7 und älter). yum löst Abhängigkeiten automatisch auf und kann aus Repositories Pakete installieren, entfernen, aktualisieren . (In neueren RHEL8+ abgelöst durch dnf, aber Syntax ähnlich.)

### Syntax:

yum [Optionen] <Sub-Befehl> [Paketnamen/...]

### Haupt-Subkommandos:

- **install Paket** – Installiert ein Paket (aus den konfigurierten Repos oder einer RPM-Datei). Z.B. yum install httpd. Resolved dependencies automatisch und fragt um Bestätigung.
- **update [Paket]** – Aktualisiert alle installierten Pakete (wenn kein Name angegeben) oder ein bestimmtes. (yum update führt System-Update durch) .
- **remove Paket, erase** – Entfernt angegebene Pakete mit Abhängigkeiten, die dadurch verwiesen.
- **search Stichwort** – Durchsucht die Paketnamen, Beschreibungen etc. nach dem Stichwort (um herauszufinden, welches Paket etwas bietet).
- **info Paket** – Zeigt Infos zu Paket(en) (Name, Version, Summary, Größe, Repo, Abh.) ähnlich rpm -qi aber aus Repos wenn nicht installiert.
- **list [installed|available] <Pattern>** – Listet Pakete (installierte oder verfügbare). Ohne Parameter alle Pakete, optional mit Pattern filtern. Z.B. yum list available 'php\*'.
- **provides Datei/Capability** – Findet heraus, welches Paket eine bestimmte Datei oder "Provide" bereitstellt (ähnlich rpm -qf/--whatprovides). Z.B. yum provides /usr/bin/zip.
- **clean <was>** – Bereinigt lokale Caches: z.B. yum clean all (löscht Metadaten- und Paketcache), oder yum clean metadata etc.
- **check-update** – Zeigt verfügbare Updates, ohne sie zu installieren (Exitcode 100 wenn Updates vorhanden, 0 wenn keine).
- **repolist** – Listet eingerichtete Repositories und ob aktiviert.
- **groupinstall / groupremove** – Installiert/entfernt ganze Paketgruppen (z.B. yum groupinstall "Development Tools").

### **Optionen (Auswahl):**

- **-y** – Automatisch "yes" beantworten (nicht nach Bestätigung fragen). Wichtig für Skript】.
- **-q** – Weniger Ausgabe (quiet).
- **-C** – Nur aus dem Cache arbeiten, keine neuen Metadaten laden (offline).
- **--enablerepo=X, --disablerepo=Y** – Repositories gezielt ein-/ausschalten für diesen Befehl (Overrides config).
- **--nogpgcheck** – Deaktiviert GPG-Signaturprüfung der Pakete (unsicher, nur falls Repo-Key nicht installiert etc.).
- **--downloadonly** – Lädt die Pakete herunter, installiert aber nicht.
- **--security** – (bei update/info/list) berücksichtigt nur sicherheitsrelevante Updates (wenn Repo das unterstützt).

### **Beispiele:**

- yum install nginx php php-mysql – installiert mehrere Pakete in einem Rutsch.
- yum update – bringt System auf neuesten Stand.
- yum search "pdf viewer" – sucht nach Paketen mit "pdf viewer" im Namen/Beschreibung.
- yum remove httpd – deinstalliert Apache (und abhängige Module, wenn keine anderen es benötigen).
- yum provides "\*bin/rar" – findet Paket, das rar-Binary liefert.

(*Hinweis: Auf neueren Systemen ersetzt dnf den Befehl yum. dnf Syntax und Subcommands sind weitgehend gleich, aber z.B. dnf upgrade statt yum update.)*

### **apt (Advanced Package Tool – Debian/Ubuntu)**

**Beschreibung:** Höher-level Frontend für Debian-Paketverwaltung (dpkg). apt kombiniert Funktionen von apt-get und apt-cache in einem Kommando. Es verwaltet .deb-Pakete: Installation, Updates aus Repositories, Paket-Suche usw】. (Ubuntu 16.04+ und Debian Stretch+ empfehlen apt für interaktive Nutzung, während apt-get skriptfreundlich bleibt.)

### **Syntax:**

apt [Optionen] <Sub-Befehl> [Pakete...]

### **Wichtige Sub-Befehle:**

- **update** – Aktualisiert die Paketquellen-Datenbank (lädt neue Paketlisten von den konfigurierten Repos). Immer vor Install/Upgrade ausführen.
- **upgrade** – Installiert verfügbare Updates für alle **schon installierten** Pakete (führt keine Entfernung durch). Hält Pakete zurück, wenn Abhängigkeitswechsel nötig wären. (*vgl. apt full-upgrade*).
- **full-upgrade** – Führt vollständige Distribution-Upgrades durch, d.h. aktualisiert alle Pakete und darf auch Pakete entfernen oder neue Abhängigkeiten hinzufügen, um ein konsistentes System herzustellen (entspricht apt-get dist-upgrade).
- **install <pkg>...** – Installiert angegebene Pakete (aus Repos oder .deb Dateien via Pfad). Löst Abhängigkeiten auf und fragt bei Bedarf. Akzeptiert Paket=Version oder Paket:arch.
- **remove <pkg>...** – Entfernt Pakete, lässt jedoch veränderte Konfigurationsdateien bestehen (als *.config* in /etc).
- **purge <pkg>...** – Entfernt Pakete **inklusive** Konfigurationsdateien (bereinigt vollständig).
- **search <regex>** – Durchsucht Paketnamen und Beschreibungen nach einem Begriff (regex möglich). *Beispiel:* apt search editor pdf.
- **show <pkg>** – Zeigt detaillierte Informationen zu einem Paket (Version, Abhängigkeiten, Beschreibung, Maintainer etc.), egal ob installiert oder nur verfügbar】.
- **list** – Listet Pakete. Mit --installed, --upgradeable oder --all-versions. (Ähnlich dpkg -l aber filterbar). Z.B. apt list --installed | grep nginx.
- **autoremove** – Entfernt automatisch Pakete, die als Abhängigkeiten installiert wurden und nun nicht mehr benötigt werden (Waisen).
- **depends <pkg>, rdepends <pkg>** – Zeigt Abhängigkeiten bzw. Reverse-Abhängigkeiten eines Pakets (welche Pakete benötigen es).
- **edit-sources** – Öffnet die sources.list im Editor (um Repos zu bearbeiten).

#### Optionen:

- **-y, --yes** – Automatisch alle Fragen mit Yes beantworten (nicht interaktiv bestätigen). Praktisch für Skripte.
- **-q** – Weniger Ausgabe (bis zu -qq für gar keine).
- **--no-install-recommends** – Installiert nicht automatisch als "empfohlen" markierte Pakete. Standard apt installiert Recommends mit.

- **--allow-remove-essential, --allow-change-held-packages** – Erlaubt riskante Änderungen (Standard verhindert z.B. Entfernen essentieller Pakete).
- **-s, --simulate** – Simulation: Zeigt an, was würde passieren, aber führt keine Änderung aus (auch Exitcode 0 immer). Gut zum Test.
- **-V, --verbose-versions** – Zeigt bei Auflistungen detailliertere Versionsinfos.
- **--help** – Hilfe-Seite, **--version** – Version von apt.

### **Beispiele:**

- apt update && apt upgrade – Aktualisiert Paketlisten und installiert alle verfügbaren Updates (ohne Dist-Upgrade).
- apt install build-essential – Installiert eine Paketgruppe (hier C/C++ Compiler Toolchain).
- apt remove --purge openssh-server – Entfernt SSH-Server komplett mit Config.
- apt search docker – Sucht alle Pakete mit "docker" im Namen/Beschreibung.
- apt full-upgrade – Aktualisiert System auch über Distribution-Upgrades hinweg (entspricht auf Ubuntu einem Release-Upgrade, sofern Repos angepasst).

*(Hinweis: apt-get und apt-cache sind Vorgänger: z.B. apt-get install, apt-get remove, apt-get dist-upgrade, apt-cache search, apt-cache show. apt vereinfacht den Alltag, aber Skripte nutzen weiterhin oft apt-get -y etc. dpkg siehe unten ist noch eine Stufe tiefer für einzelne .deb ohne Abhängigkeitsauflösung.)*

### **dpkg (Debian Package Manager - Low-Level)**

**Beschreibung:** dpkg ist das unterliegende Werkzeug auf Debian/Ubuntu, das .deb-Pakete entpackt, installiert, konfiguriert und entfernt. Es löst keine Abhängigkeiten – dafür sind apt/apt-get zuständig. Man verwendet dpkg meist für manuelle Offline-Installation von .deb-Dateien.

### **Syntax:**

```
dpkg -i <Paket.deb> # Installieren
dpkg -r <Paketname> # Entfernen
dpkg -P <Paketname> # Entfernen + Purge
dpkg -L <Paketname> # Liste Dateien des installierten Pakets
dpkg -s <Paketname> # Statusinfo (ähnlich apt show)
```

### **Wichtige Optionen/Parameter:**

- **-i, --install Datei.deb**: Installiert das angegebene Deb-Paket. Gibt Fehler aus, falls Abhängigkeiten fehlen. (Diese müssen vorher mit apt installiert werden).
- **-r, --remove Paketname**: Entfernt ein installiertes Paket (Paketname ohne ".deb", wie in dpkg -l gelistet). Konfigdateien unter /etc bleiben bestehen.
- **-P, --purge Paketname**: Entfernt Paket inkl. seiner Konfigurationsdateien.
- **-S, --search Dateiname**: Sucht, welches installierte Paket eine bestimmte Datei besitzt (ähnlich rpm -qf). Z.B. dpkg -S /usr/bin/foo.
- **-L, --listfiles Paketname**: Listet alle Dateien, die das installierte Paket auf dem System installiert hat [file-5y1edouysrow4zj6phjcqm].
- **-l, --list Muster**: Listet Pakete, optional gefiltert nach Muster. dpkg -l ohne Filter zeigt eine Tabelle aller Pakete und deren Status (ii = installiert, rc = removed config-files left, etc.).
- **-s, --status Paketname**: Zeigt den Status und Informationen eines installierten Pakets (Version, Beschreibung, Maintainer, Abhängigkeiten...).
- **--configure Paketname**: Führt die Konfigurationsschritte (dpkg-configure) für ein installiertes aber noch nicht konfiguriertes Paket aus. (Wird z.B. nach entpacken aufgerufen).
- **-B, --auto-deconfigure**: Beim Entfernen eines Pakets, das Abhängigkeiten bereitstellt, markiert abhängige Pakete als "deconfigured".
- **\*\*--force-\*\*X**: Viele Force-Flags, um dpkg anzuweisen, bestimmte Fehler zu ignorieren (z.B. --force-dependencies ignoriert fehlende Abhängigkeiten, --force-confold/--force-confnew bei Config-Datei-Konflikten etc.). Vorsicht: Kann System inkonsistent machen.
- **--print-architecture**: Zeigt die Architektur, für die dpkg Pakete installiert (z.B. amd64).
- **--add-architecture <arch>**: Fügt Multiarch-Unterstützung für andere Arch hinzu (z.B. i386 auf einem amd64 System).

### Beispiele:

- dpkg -i google-chrome.deb – Installiert Chrome .deb-Paket (scheitert, wenn Abhängigkeiten fehlen – in dem Fall via apt --fix-broken install auflösbar).
- dpkg -r postfix – Entfernt postfix, belässt aber ggf. geänderte /etc/postfix Dateien.
- dpkg -P postfix – Entfernt postfix vollständig inklusive Config.
- dpkg -L bash – Listet alle vom *bash*-Paket installierten Dateien.

- `dpkg -S /bin/ls` – Welches Paket stellt `/bin/ls` bereit? (Antwort: coreutils).

## wget (Web GET Utility)

**Beschreibung:** Ein Kommandozeilen-Downloader für Dateien über HTTP, HTTPS und FTP. wget kann in Skripten genutzt werden, um Dateien aus dem Web herunterzuladen, unterstützt Fortsetzen abgebrochener Downloads, rekursives Herunterladen von Websites etc. Es funktioniert nicht-interaktiv (kein Browser, nur Download).

### Syntax:

`wget [Optionen] <URL>`

### Wichtige Optionen:

- **-O Datei** – Speichert die heruntergeladene URL in die angegebene *Datei* (statt den Namen aus URL zu verwenden). Z.B. `wget -O ubuntu.iso http://.../ubuntu.iso`.
- **-c, --continue** – Setzt einen abgebrochenen Download fort (sofern Server Range Requests unterstützt). Prüft, wieviel schon vorhanden ist und lädt den Rest.
- **-q** – Quiet-Modus (keine Ausgabe). **-nv** (not verbose) eine dazwischenliegende Stufe. **-v** für ausführlicher (Standard).
- **-r, --recursive** – Rekursiver Download einer Website. Erfordert oft zusammen mit **-np** (no parent), **-L** (stay on same domain), **-k** (Links anpassen für Offline). Z.B. `wget -r -np -k http://example.com/docs/`.
- **-np, --no-parent** – Geht beim rekursiven Download nicht über das Startverzeichnis hinaus.
- **-l <Tiefe>** – Legt die Rekursionstiefe fest (Default 5, **-l0** = unendlich).
- **-p, --page-requisites** – Lädt alles mit, was nötig ist, um HTML-Seite offline darzustellen (Bilder, CSS...).
- **-N, --timestamping** – Lädt eine Datei nur herunter, wenn sie neuer ist als die lokale (verwendet Last-Modified). Gut zum Aktualisieren.
- **--limit-rate=<Rate>** – Begrenzt Downloadrate, z.B. `--limit-rate=200k` (200 KB/s). Hilfreich, um Bandbreite zu drosseln.
- **-U, --user-agent=<Agent>** – Setzt einen bestimmten User-Agent-String. Standard ist "Wget/Version". Kann geändert werden, um z.B. nicht blockiert zu werden.
- **--no-check-certificate** – Ignoriert SSL-Zertifikatfehler (z.B. selbstsigniert).
- **--user=<user> --password=<pw>** – Falls erforderlich, für HTTP-Auth oder FTP-Login. (Achtung: Sichtbar in Prozessliste).

- **-i <Datei>** – Liest eine Liste von URLs aus <Datei> (jede Zeile eine URL) und lädt alle herunter.
- **-b, --background** – Startet im Hintergrund (nur auf Unix, leitet Ausgabe nach wget-log).
- **-P <Verz>** – Speichert Dateien in angegebenem Verzeichnis (statt aktuelles).

### Beispiele:

- wget http://example.com/file.zip – Lädt file.zip in aktuelles Verzeichnis herunter.
- wget -c http://largefile – setzt Download fort, falls vorhanden.
- wget -r -np -N http://example.com/photos/ – Lädt Verzeichnis "photos" vollständig herunter, aktualisiert nur neue Dateien beim Wiederaufruf.
- wget -O - https://api.example.com/data | jq . – Lädt Daten von API und pipet direkt in jq zur Verarbeitung (hier -O - bedeutet Ausgabe auf stdout statt Datei).

## curl (Client URL)

**Beschreibung:** curl ist ein umfangreiches Kommandozeilen-Tool für Datenübertragungen mit URL-basierten Protokollen (HTTP, HTTPS, FTP, etc.). Im Gegensatz zu wget liegt Fokus auf Flexibilität bei HTTP-Requests (Header setzen, verschiedene HTTP-Methoden, Formulare absenden, APIs ansprechen etc.). Es eignet sich für Web-APIs (GET/POST) und kann ebenfalls Dateien herunterladen oder hochladen.

### Syntax:

curl [Optionen] <URL>

### Wichtige Optionen:

- **-o Datei** – Speichert die Ausgabe in Datei. (Kleinbuchstabe o, analog wget -O).
- **-O** – Speichert unter dem **Originalnamen** aus der URL (Großes O, analog wget ohne -O). Achtung: curl -O http://server/dir/file.txt legt file.txt an.
- **-L – Location folgen:** folgt HTTP-Redirects (Status 3xx) automatisch [freedesktop.org](https://freedesktop.org). (wget macht das default, curl nicht ohne -L).
- **-C –** Setzt abgebrochenen Download fort (wenn Server unterstützt; '-' heißt automatischen Offset ermitteln).
- **-# –** Fortschrittsleiste im CLI (oder --progress-bar). Standard curl zeigt laufende Stats Zeile, -# macht wie wget progressive Bar.
- **-s –** Silent (kein Progress oder Fehlerausgaben). **-S** kombiniert mit -s zeigt Fehler dennoch an.

- **-v / -vv** – Verbose Mode (zeigt detaillierte Request/Response, Header etc. -vv noch mehr, inkl. Low-level). Gut zum Debuggen.
- **-I** – Führt eine HEAD-Anfrage aus (nur Header vom Server holen, keine Body-Daten). Nützlich um Meta-Informationen (Content-Length, Last-Modified) abzufragen.
- **-X METHOD** – Explizite HTTP-Methode festlegen (z.B. PUT, DELETE). Standard bei Datenvorhandensein oder -d ist POST, sonst GET.
- **-G** – Erzwingt, dass bei Verwendung von -d (Daten) diese als Query-Parameter an URL angehängt werden (statt POST Body).
- **-d Daten** – Sendet die angegebenen *Daten* im HTTP-Request-Body (Standard via POST). Z.B. curl -d "user=test&pass=123" http://.../login. Implizit -X POST wenn nicht -G.
  - **--data-urlencode Daten**: URL-encodet die Daten vor Versand (z.B. wenn Daten Sonderzeichen enthalten).
  - **--data-raw**: ähnlich -d, aber ohne spezielle Form-Codierung (sendet genau so).
- **-H "Header: Wert"** – Fügt einen HTTP-Header hinzu oder überschreibt ihn. Z.B. -H "Authorization: Bearer <token>". Mehrfach nutzbar für mehrere Header.
- **-u Benutzer:Pass** – Nutzt HTTP Basic Auth mit gegebenen Credentials (wird in Header "Authorization: Basic ...Base64" codiert[freedesktop.org](#)). Vorsicht in Shell-History. Kann auch nur -u user (dann fragt curl interaktiv Passwort ab). Funktioniert auch für FTP.
- **-F Feld@Datei** – HTTP Datei-Upload via *multipart/form-data*. Z.B. curl -F "file=@bild.png" -F "desc=Urlaubsfoto" http://.../upload.
- **-K Konfigdatei** – Liest curl-Parameter aus Datei (kann nützlich sein für komplexe Setups).
- **-b CookieDatei** – Sendet Cookies aus angegebener Datei (. Netscape cookie format oder einfach "name=value" Paare). Kann auch -b "name=value" um Cookies zu setzen.
- **-c CookieAusgabe** – Speichert Cookies, die vom Server kommen, in angegebener Datei (um sie später wieder zu verwenden).
- **--limit-rate <Speed>** – Begrenzt Übertragungsrate (z.B. --limit-rate 100k).
- **--retry N** – Anzahl der automatischen Wiederholungsversuche bei Fehler (z.B. Timeout) oder --retry-all-errors.

- **--cacert Datei / --insecure:** Spezielles SSL: anderes CA-Zertifikat nutzen bzw. bei --insecure SSL-Zertifikat überprüfen *ignorieren* (wie wget --no-check-certificate).

### **Beispiele:**

- curl -L -o latest.zip https://github.com/user/proj/releases/latest – Lade neueste Release, folge Redirects, speichere als latest.zip.
- curl -u admin:secret -X DELETE http://server/api/item/123 – Sende DELETE-Request mit Basic Auth.
- curl -H "Accept: application/json" 'https://api.example.com/data?id=5' – Hole JSON-Daten von API mit angepasstem Accept-Header.
- curl -d '{"user":"john"}' -H "Content-Type: application/json" -X POST https://api.example.com/users – Sende JSON-Body per POST (hier ein neuer User).
- curl -F 'file=@report.pdf' -F 'desc=Monthly Report' https://upload.example.com – HTTP File-Upload (Multipart).
- curl -I https://example.com – Nur Header einer HTTP-Response anzeigen (z.B. um Last-Modified zu prüfen).

## **Speicher- und Dateisystem-Verwaltung**

### **lsblk (List Block Devices)**

**Beschreibung:** Zeigt eine Übersicht aller erkannten Blockgeräte (Festplatten, SSDs, Partitionen, LVM-Volumes etc.) in Form einer Baumstrukturfile-4fdk7rvx39azjretinarqb】 . lsblk listet wichtige Attribute wie Gerätename, Größe, Typ (disk/part), Mountpoint usw. Es hilft beim Identifizieren von Device-Namen und Partitionierungsstruktur.

### **Syntax:**

lsblk [Optionen] [Gerät...]

### **Optionen:**

- **-a, --all:** Zeigt auch leere bzw. unverwendete Geräte an (z.B. Geräte ohne Speicher, Loop ohne Backing). Standard lässt einige unwichtige aus.
- **-d, --nodeps:** Zeigt nur "Top-Level" Geräte (Disks) ohne Partitionen/Slaves.
- **-e <LIST>, --exclude <LIST>:** Schließt bestimmte Gerätetypen durch Major-Nummern oder Gerätekategorien aus (z.B. -e 7,11 um Loop und DVD zu ignorieren).
- **-f, --fs:** Zeigt zusätzlich Dateisystem-Informationen (Spalten FSTYPE, LABEL, UUID】 . Ähnlich blkid, aber in lsblk integriert.

- **-p, --paths:** Zeigt vollständige /dev/Pfadnamen an statt nur Gerätenamen (z.B. "/dev/sda" statt "sda").
- **-l, --list:** Ausgabe im Listenformat (eine Eintrag pro Zeile, ohne Baum-Struktur).
- **-J, --json:** Ausgabe als JSON-Struktur (maschinenlesbar).
- **-O, --output-all:** Zeigt alle verfügbaren Spalten an. Standardanzeige ist begrenzt. Mit -O sieht man z.B. auch MAJ:MIN, RM (removable), RO (read-only), MODEL, SERIAL etc.
- **-n, --noheadings:** Unterdrückt Überschrift.
- **-b, --bytes:** Größe in Bytes anzeigen (statt lesbar in KiB/MiB etc.).
- **-r, --raw:** Rohformat ohne Ausrichtung/Padding (für Skripting).

### **Beispielausgabe:**

```
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda 8:0 0 100G 0 disk
 ↳ sda1 8:1 0 512M 0 part /boot
 ↳ sda2 8:2 0 99.5G 0 part
 ↳ vg-root 253:0 0 90G 0 lvm /
 ↳ vg-swap 253:1 0 8G 0 lvm [SWAP]
sr0 11:0 1 1024M 0 rom (cdrom)
```

Hier sieht man Disk *sda* mit 100G und zwei Partitionen *sda1*, *sda2*. *sda2* ist ein PV (Physical Volume) mit Volume Group *vg* -> daraus LVs *vg-root* und *vg-swap*, gemountet entsprechend.

### **lsscsi (List SCSI Devices)**

**Beschreibung:** Listet erkannte SCSI-Geräte (inkl. SATA, SAS, USB Storage – diese werden SCSI-Subsys emuliert) auf. **lsscsi** zeigt je Gerät Host/Channel/Target/LUN, dann Gerätetyp, und eventuell Model und Seriennummer je nach Infile-  
4fdk7rvx39azjretinarqb】 . Hilfreich, um z.B. alle angeschlossenen Festplatten und optischen Laufwerke anzuzeigen.

### **Syntax:**

**lsscsi [Optionen]**

### **Optionen:**

- **-v, --verbose:** Ausführlich – zeigt zusätzliche Infos (z.B. SCSI-Transport oder Zustände).

- **-t, --transport:** Zeigt SCSI-Transportinformationen (z.B. SATA, USB UAS, iSCSI etc.).
- **-L, --lunhex:** Zeigt die LUN in hexadezimaler Notation statt dezimal.
- **-H, --hosts:** Zeigt nur SCSI-Host-Adapter (Controller) an.
- **-c, --classic:** Benutzt "klassisches" Ausgabeformat (alter Stil, ohne Abkürzungen).
- **-g, --generic:** Fügt die entsprechenden sg-Device-Nodes hinzu (SCSI Generic /dev/sgX).
- **-s, --size:** Zeigt Größe (Kapazität) der Block-Geräte an, falls anwendbar (nur für Disk-type).
- **-p, --protection:** Zeigt Protection-Information (DIF/DIX).
- **-i, --scsi\_id:** Zeigt die SCSI-IDs im klassischen Sinn (T10 IDs).
- **-B, --usb:** Zeigt Bus-IDs (für USB-Geräte) mit an.

### **Beispiel:**

[0:0:0:0] disk ATA Samsung SSD 860 1B6Q /dev/sda 500GB

[2:0:0:0] cd/dvd TSSTcorp CDDVDW SH-224DB SB00 /dev/sr0

[3:0:0:0] disk WD MyBook 25DA 4004 /dev/sdb 4TB

Format: [H:C:T:L] type vendor model rev /dev/sgX -> /dev/sdX. Hier sieht man SCSI-Gerät auf Host 0 (SATA), Host 2 (DVD), Host 3 (USB-HDD).

### **fdisk (Partition Table Editor)**

**Beschreibung:** Interaktives Konsolen-Tool zum Anzeigen und Bearbeiten von Partitionstabellen auf Blockgeräten (MBR, GPT). Meistens wird fdisk verwendet, um schnell die Partitionierung eines Laufwerks anzusehen (fdisk -l) oder bei MBR-Partitionierung Partitionen einzurichten. Für GPT kann man gdisk oder parted benutzen (fdisk unterstützt aber inzwischen auch GPT).

### **Syntax:**

fdisk [Optionen] <Gerät>

Ohne Optionen startet es ein interaktives Menü für das angegebene Gerät (z.B. /dev/sda).

### **Optionen:**

- **-l, --list:** Listet die Partitionstabellen aller verfügbaren Laufwerke auf [file-4fdk7rvx39azjretinarqb]. Oder wenn ein Gerät angegeben, nur diese. Z.B. fdisk -l /dev/sda zeigt Partitionen auf sda.
- **-s Partition:** Gibt die Größe der angegebenen Partition in Sektoren aus (veraltet, blockdev --getsz moderner).
- **-u (veraltet):** Wechselt die Ausgabeeinheit (Sektoren vs Zylinder) in alter fdisk. Heutige fdisk (util-linux) nutzt standard 512-Byte Sektoren in -l.
- **-B, -C, -H, -S:** Kann man Geometrie (Zylinder, Heads, Sectors) angeben (MBR Legacy, heute irrelevant).
- **-t Typ:** Zeigt Partitionen in bestimmtem Format (Dos, GPT, ...).

Interaktiver Modus: Kommandos wie m (Hilfe), p (Print Partitionstabelle), n (neue Partition), d (löschen), t (Typ ändern, z.B. 83 Linux, 82 Swap, in GPT GUIDs), w (write und exit), q (quit ohne save).

### **Beispiel (List):**

Disk /dev/sda: 238.5 GiB, ...

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1	*	2048	534527	532480	260M	7	HPFS/NTFS/exFAT
/dev/sda2		534528	409600000	409065473	195.1G	83	Linux
/dev/sda3		409600001	419430400	9830400	4.7G	82	Linux swap / Solaris

Zeigt z.B. 3 Partitionen (sda1 NTFS, sda2 Linux, sda3 Swap).

### **parted (Partitionierungstool)**

**Beschreibung:** Moderner Partitionierer, unterstützt GPT und MBR sowie Größe > 2TB. Kann sowohl interaktiv (im eigenen Prompt) als auch direkt per Kommandozeile genutzt werden. Im Gegensatz zu fdisk beherrscht parted auch **Resizing** von Partitionen und Filesystems (begrenzte FS-Typ Unterstützung) und kann in Skripten genutzt werden.

### **Syntax (non-interactive):**

parted [-s] <Gerät> <Befehl> [Parameter...]

Mit -s (script) ohne interaktive Ausgabe.

### **Häufige parted Befehle:** (Nicht alle Optionen, nur exemplarisch)

- **print** – Zeigt Partitionstabelle des Geräts. Z.B. parted /dev/sdb print.
- **mklabel <Typ>** – Erstellt neue Partitionstabelle (Typ: gpt, msdos, etc.). Achtung: löscht alle bestehenden Partitionen.

- **unit <Einheit>** – Setzt Maßeinheit (MB, GB, %, etc.) für folgende Befehle.
- **mkpart Name Typ Anfang Ende** – Erstellt Partition. Z.B.: mkpart primary ext4 1MiB 10000MiB erstellt eine ~10GB Partition vom 1MB Offset. **Typ** hier "primary"/"logical" (bei msdos) oder nur Partitions-TypName (bei GPT optional ein FS-Typ Label ext4 etc.).
- **rm <Nr>** – Löscht Partition mit Nummer.
- **resizepart <Nr> <Ende>** – Passt Größe der Partition Nr an (End-Sektor). Muss danach FS separat resized werden (z.B. mit resize2fs).
- **name <Nr> <Name>** – Setzt den Partitionsnamen (GPT Label).
- **toggle <Nr> boot** – Boot-Flag toggeln (bei MBR Partitionen).
- **align-check Typ Nr** – Überprüft Ausrichtung von Partition Nr (optisch/ Zylinder).

Parted kann auch einfache FS-Aufgaben: mkfs integration, aber meist nutzt man separate Tools (mkfs.\*).

### **Beispiele:**

- parted /dev/sdc --script mklabel gpt mkpart primary ext4 0% 50% mkpart primary linux-swap 50% 100% – Initialisiert /dev/sdc als GPT, teilt in zwei: erste Partition bis Mitte ext4, zweite Swap ab Mitte bis Ende.
- Im interaktiven parted (ohne -s) bekommt man (parted) Prompt zum Eingeben der obigen Befehle nacheinander.

## **mkfs (Make FileSystem)**

**Beschreibung:** Erzeugt ein neues Dateisystem auf einer Partition oder Festplatte. mkfs ist ein Frontend, das je nach -t <FSTYP> das entsprechende Tool (mkfs.ext4, mkfs.xfs, etc.) aufruft. Häufig ruft man direkt das spezifische Formatprogramm auf.

### **Syntax:**

mkfs -t <FSTYP> [Optionen] <Geräte/Partition>

Beispiele: mkfs -t ext4 /dev/sdb1, mkfs -t vfat /dev/sdb1.

### **Wichtige Optionen (allgemein):**

- **-t <typ>** – Dateisystemtyp (ext4, ext3, ext2, xfs, vfat, ntfs, etc.).
- **-V** – zeigt ausführliche Versions-/Debug-Information (welches Tool aufgerufen wird).
- **-q** – quiet (weniger Ausgabe).

Jedes Dateisystem hat eigene spezifische Optionen, wenn man direkt mkfs.ext4 etc. nutzt:

**ext4 (mkfs.ext4):** -L <Label>, -U <UUID> (benutzerdefiniert), -m <Proz> (Reserve-Blöcke Prozent, default 5%), -O <Feature> (ein/aus bestimmter FS-Features), -E stride= (RAID optimierungen), -b blocksize, etc.

**vfat (mkfs.vfat):** -n <Label>, -F <32|16> (FAT type), -I (Force if not partition), etc.

**xfs (mkfs.xfs):** -L <Label>, -m crc=1 (z.B. CRC enable, default ON neu), etc.

**swap (mkswap):** -L <Label>, (swap selbst ist via mkswap erstellt).

*Wichtig:* Partition sollte entsprechend markiert/Dimension formatiert sein (z.B. parted type=linux-swap für swap etc.).

### Beispiele:

- mkfs.ext4 /dev/sda3 – Formatiert sda3 mit ext4 Standardparametern.
- mkfs.vfat -F 32 -n "MYUSB" /dev/sdb1 – Formatiert Partition sdb1 als FAT32 mit Label "MYUSB".
- mkswap /dev/sda5 – Initialisiert Swap auf sda5 (später via swapon).
- mkfs.xfs -f -L data /dev/sdc1 – Erzwingt Format auf sdc1 mit XFS, Label "data" (XFS verlangt -f wenn Partition schon ein FS enthält).

## mount (Dateisystem einhängen)

**Beschreibung:** Bindet ein Dateisystem (Partition, Datenträger oder virtuelles FS) in den Verzeichnisbaum ein. mount ohne Argumente listet aktuelle Einhängungen. mount <Gerät> <Verzeichnis> hängt ein Gerät an den angegebenen Ordner ein (der Ordner muss existieren und leer sein, typischerweise unter /mnt oder /media oder definierter Mountpoint in /etc/fstabfile-4fdk7rvx39azjretinarqb】 .

### Syntax:

mount [Optionen] <Quelle> <Zielverzeichnis>

### Wichtige Optionen:

- **-t <fstype>** – gibt den Dateisystemtyp an (ext4, vfat, nfs, cifs, etc.). Kann oft weggelassen werden, da mount es automatisch erkennt (über blkid).
- **-o <Optionen>** – übergibt eine durch Komma getrennte Liste von Mount- Optionenfile-4fdk7rvx39azjretinarqb】 :
  - **defaults** (rw,suid,dev,exec,auto,nouser,async),
  - **ro/rw** (read-only / read-write),
  - **noexec** (keine Ausführung von Programmen erlauben), **nosuid** (SUID/SGID-Bits ignorieren), **nodev** (keine Gerätedateien interpretieren),

- **sync** (synchrone IO), **async** (Standard, asynchron gepuffert),
  - **user/nouser** (Allow normal user to mount / nur root),
  - **uid=, gid=** (für FAT/NTFS, setze Besitzer der gemounteten Dateien),
  - **umask=, dmask=, fmask=** (Rechte-Masken für FAT/NTFS),
  - **iocharset=, codepage=** (für vfat),
  - **loop** (Loop-Device mount einer Datei als FS, evtl. -o loop oder auto),
  - **remount** (ein bereits gemountetes FS mit neuen Optionen versehen, z.B. mount -o remount,rw /mnt/point),
  - FS-spezifisch: z.B. **acl**, **user\_xattr** (Ext4 Features), **compression**, **encrypt** (btrfs), etc.
- **-L <Label>, -U <UUID>** – anstatt Device-Node, per Label oder UUID identifizieren. mount -U 1234-ABCD /mnt/usb.
- **--bind** – führt einen Bind-Mount aus: mountet ein vorhandenes Verzeichnis an eine zweite Stelle im Verzeichnisbaum. Syntax: mount --bind /pfad/quelle /pfad/ziel. (Nützlich, um z.B. Verzeichnisse in Chroots sichtbar zu machen). Option **--rbind** rekursiv (inkl. Unterpunkte).
- **-a** – Mountet alles was in /etc/fstab als "auto" markiert ist und nicht noauto. (Ausführung typ. beim Boot).
- **-n** – führt Mount ohne Eintrag in /etc/mtab aus (mtab ist heute oft Symlink zu /proc/self/mounts). Selten benötigt.
- **-v** – verbose (gibt Meldungen aus).

### **Beispiele:**

- mount /dev/sdb1 /media/usb -o uid=1000,gid=1000,umask=022 – hängt einen FAT32-Stick so ein, dass Benutzer mit UID/GID 1000 Besitzer ist (typisch eigener User), und Rechte 755 auf Dateien/Ordner (umask 022).
- mount -t nfs 192.168.1.10:/export/share /mnt/share – hängt eine NFS-Freigabe ein (NFS FS muss installiert sein).
- mount -o loop diskimage.iso /mnt/iso – mountet ein ISO-Image als Loop-Device unter /mnt/iso.
- mount | grep sda – sieht, ob /dev/sdaxX gemountet ist (ohne Parameter listet mount alles).

### **umount (Unmount FileSystem)**

**Beschreibung:** Hängt ein gemountetes Dateisystem wieder aus. Schreibvorgänge werden abgeschlossen und der Puffer geleert. Syntax kann entweder der Mountpoint oder das Device angegeben werden】 .

### Syntax:

umount [Optionen] <Mountpunkt | Gerät>

### Wichtige Optionen:

- **-f, --force:** Erzwingt das Aushängen eines FS. Sollte nur im Notfall (z.B. NFS-Server down) benutzt werden, da es evtl. Datenverlust verursachen kann.
- **-l, --lazy:** "Lazy unmount". Markiert das FS als ausgeworfen, aber falls noch offene Handles bestehen, wird es erst tatsächlich ausgehängt, wenn diese geschlossen sind. (Hängt unmittelbar virtuell aus, lässt Rest im Hintergrund erledigen). Nützlich bei hängenden Prozessen.
- **-R, --recursive:** Hängt auch alle Unter-Mounts mit ab, die auf dem angegebenen Pfad liegen (rekursiv).
- **-v:** verbose.

### Beispiele:

- umount /media/usb – hängt das unter /media/usb eingehängte Gerät aus.
- umount /dev/sdb1 – selbes, angesteuert über Gerät.
- umount -f -a -t nfs – erzwingt das Aushängen aller NFS-Mounts (z.B. bei Netzproblemen).

(Tipp: Wenn umount meldet "device is busy", kann man mit lsof oder fuser herausfinden, welcher Prozess noch auf das FS zugreift.)

### df (Disk Free)

**Beschreibung:** Zeigt Übersicht des belegten und freien Speicherplatzes auf eingehängten Dateisystemen】 . df listet pro gemountetem FS die Gesamtgröße, verwendeten und freien Bytes sowie den Einhängepunkt.

### Syntax:

df [Optionen] [Dateipfad...]

Ohne Argumente zeigt alle gemounteten FS. Mit Pfad ermittelt df, auf welchem FS dieser Pfad liegt, und zeigt nur dessen Werte.

### Optionen:

- **-h, --human-readable:** Formatiert Größen in lesbarer Form (KiB, MiB, GiBfile-4fdk7rvx39azjretinarqb】 . Sehr gebräuchlich (Standard rohes Blockzahlen unhandlich).
- **-a, --all:** Schließt pseudo- oder 0-Size-Dateisysteme ein (wie proc, sysfs, etc.), die sonst ausgelassen werden.
- **-T, --print-type:** Zeigt Spalte mit Dateisystemtyp (ext4, tmpfs, nfs, etc.).
- **-i, --inodes:** Zeigt Inode-Auslastung statt Blockgrößen (frei/belegt Inodes pro FS). Nützlich, wenn "no space" durch Inode-Erschöpfung.
- **-t <typ>, -x <typ>:** Nur bestimmte FS-Typen anzeigen bzw. ausschließen. Z.B. df -x tmpfs -x devtmpfs um echte Disks anzuzeigen.
- **-B <Größe>:** Ausgabe in Blöcken von angegebener Größe. Z.B. -B MB (MByte) oder -B 1M.
- **--output[=FELDER]:** Benutzerdefinierte Spaltenausgabe. Z.B. df --output=source,fstype,size,used,avail,pcent,target.

### **Beispielausgabe (df -h):**

```
Filesystem Size Used Avail Use% Mounted on
/dev/sda2 50G 30G 18G 63% /
udev 3.9G 0 3.9G 0% /dev
tmpfs 798M 1.2M 797M 1% /run
/dev/sda1 300M 50M 250M 17% /boot
```

### **du (Disk Usage)**

**Beschreibung:** Summiert und zeigt die Größe von Verzeichnissen und Dateien an. du kann rekursiv durch Verzeichnisse gehen und den Platzverbrauch berechnenfile-4fdk7rvx39azjretinarqb】 . Standardmäßig zeigt es für jedes Unterverzeichnis den Wert. Oft nutzt man du -sh <dir> um eine schnelle Gesamtgröße zu bekommen.

### **Syntax:**

du [Optionen] [Pfad...]

Ohne Pfad = aktuelles Verzeichnis.

### **Optionen:**

- **-h, --human-readable:** Größen in menschenlesbare Einheiten (K,M,G) ausgebene-4fdk7rvx39azjretinarqb】 .

- **-s, --summarize:** Nur Summen pro angegebenem Argument ausgeben, *nicht* rekursiv für Unterordnerfile-4fdk7rvx39azjretinarqb】 . (Gibt also Gesamtgröße des Pfades).
- **-A, --apparent-size:** Rechnet mit logischen Dateigrößen statt belegtem Blockplatz. (Z.B. sehr sparsame Sparse Files würden sonst weniger anzeigen; --apparent-size zeigt nominelle Größe).
- **-d N, --max-depth=N:** Rekursionstiefe begrenzen. Z.B. du -h --max-depth=1 /var gibt Größe von /var und dessen Unterordner erste Ebene. (--max-depth=0 wäre wie -s).
- **-c, --total:** Am Ende eine Gesamtzeile "total" ausgeben (Summe aller ausgegebenen Einträge). Bei -s von mehreren Pfaden praktisch.
- **-x, --one-file-system:** Geht nicht in andere eingehängte Dateisysteme über. (Beschränkt auf Partition/FS der Startpfade).
- **-L, --dereference:** Folgt symbolischen Links (sonst ignoriert du Symlinks, rechnet sie als 0 Bytes). Vorsicht: Kann zu doppeltem Zählen führen, wenn Links in Kreis oder auf Oberverzeichnis zeigen.
- **--exclude=<PATTERN>:** Schließt Pfade aus, die zum Pattern passen. Z.B. du -sh -exclude="\*.git" oder mehrere --exclude.

### **Beispiele:**

- du -sh /home/\* – Zeigt Summe der Größen aller Benuterverzeichnisse unter /home (jedes in human-readable).
- du -a /etc | grep -E "mysql.\*\..cnf\$" – Listet die Größe jeder einzelnen Datei unter /etc, filtert jene, deren Name auf mysql\*.cnf endet. (Option -a listet auch Dateien und nicht nur Verzeichnisse).
- du -d2 /var – Größe von /var, dessen Subdirs und Sub-Subdirs (Tiefe 2).

### **e2label (ext2/3/4 Volume Label)**

**Beschreibung:** Zeigt oder ändert das Label (Namen) eines ext2/ext3/ext4 Dateisystems. Das *Label* ist eine im Superblock gespeicherte Zeichenkette, die man z.B. in fstab zum Mounten verwenden kann (LABEL=<name>).

### **Syntax:**

e2label <Gerät> [NeuesLabel]

Ohne zweites Argument zeigt es das aktuelle Label des ext-Volumes an. Mit Label schreibt es dieses in den FS.

### **Beispiel:**

- `e2label /dev/sda3` -> Ausgabe z.B. UbuntuRoot.
- `e2label /dev/sda3 Archive` -> setzt das Label von sda3 auf "Archive".

(Für andere FS gibt es analoge Tools: XFS nutzt `xfs_admin -L`, FAT/NTFS `mlabel` oder `dosfslabel/ntfslabel` etc.)

### **resize2fs (ext2/3/4 Dateisystem verkleinern/vergrößern)**

**Beschreibung:** Passt die Größe eines ext2/3/4-Dateisystems an. Wird verwendet nach dem Ändern der Partitionsgröße (mit `fdisk/parted`). `resize2fs` kann im online-Modus vergrößern (wenn gemountet, ext3/4), fürs verkleinern muss das Volume ausgehängt und ein Filesystem-Check gemacht werden.

#### **Syntax:**

`resize2fs [Optionen] <Gerät> [NeueGröße]`

NeueGröße kann in K, M, G oder als Anzahl 4K-Blöcke angegeben werden. Lässt man sie weg, wird auf Maximum (Part.-Ende) vergrößert.

#### **Optionen:**

- **-p** – progress anzeigen (Fortschritt).
- **-f** – zwingen, auch wenn FS als nicht sauber markiert (normal will `fsck` vorher laufen bei Verkleinern).
- **-F** – flush caches vorher (sorgt für up-to-date Infos; Standard normal).
- **-M – minimal**: verkleinert das FS auf die kleinstmögliche Größe (alle Daten gerade passend). Äquivalent zu `resize2fs <dev> -M` (dauert Berechnung).

#### **Beispiele:**

- Nach Vergrößern der Partition: `resize2fs /dev/sda2` – erkennt neue Partitionsgröße und wächst das ext4 FS auf vollen Platz.
- Verkleinern: `umount /dev/sda2; e2fsck -f /dev/sda2; resize2fs /dev/sda2 20G` – erst aushängen, `fsck` erzwingen, dann auf 20 GiB verkleinern. Anschließend Partition selbst verkleinern mit `parted`.

### **tune2fs (Tune ext2/3/4 Filesystem Parameters)**

**Beschreibung:** Ändert diverse Parameter eines bestehenden ext-Dateisystems: Reserveblöcke, Check-Intervalle, Label, Journal einschalten, usw.

#### **Syntax:**

`tune2fs [Optionen] <Gerät>`

#### **Häufige Optionen:**

- **-l** – Listet alle Superblock-Parameter (Filesystem-Info) auf [freedesktop.org](http://freedesktop.org) . Zeigt z.B. Inode-Größe, Features, Letzter Mount, Nächster geplanter Check, etc. (Read-only Operation).
- **-c <MaxMounts>** – Setzt die maximale Anzahl Mount-Vorgänge bis zur erzwungenen fsck-Prüfung. Z.B. -c 0 = deaktiviert mount-count-basierten Check.
- **-i <Interval>** – Setzt das Zeit-Intervall für Zwangs-FSCK. z.B. -i 30d = alle 30 Tage. -i 0 = kein zeitbasierter Check.
- **-r <AnzRes>** – Setzt Anzahl der reservierten Blöcke (für root). Standard 5% bei ext4 auf <1TB. Kann man z.B. auf 1% reduzieren auf großen Volumes, z.B. tune2fs -r 0 /dev/sdb1 um Reserven komplett abzuschalten (nicht empfohlen auf /). Alternativ -m <Prozent> (z.B. -m 1).
- **-L <Label>** – Ändert Label (wie e2label).
- **-U <UUID>** – Ändert die UUID (zufällig oder angegeben). (Vorsicht: fstab etc. müssen angepasst).
- **-J** – Journal-Optionen: z.B. -O has\_journal um Journal hinzuzufügen (macht aus ext2 -> ext3), oder -O ^has\_journal Journal entfernen (ext3->ext2), oder -j Kurzoption (Standardwerte Journal hinzufügen).
- **-O <FeatListe>** – Aktiviert (feature) oder deaktiviert (^feature) Filesystem-Features. Z.B. -O casefold (für ext4 ab 5.2: Case-Insensitive-Folder Support) oder ^huge\_file (Feature abschalten – Achtung, Daten verlieren, nur wenn FS leer). Nur bestimmte Features togglebar, meist erfordert offline und vollständiges Backup.
- **\*\*-E <extendopt>\*\*** – Erweiterte Einstellungen: z.B. stripe=<StripeSize>(RAID stripe in Blöcken),lazy\_itable\_init=1` (schnell format, Inode-Table lazy Clearing).
- **-M <Verzeichnis>** – Markiert FS als gemountet an <Verzeichnis> (setzt den Mountpoint in Superblock – meist rein informativ).

## dumpe2fs (Dump ext2/3/4 Superblock)

**Beschreibung:** Gibt ausführliche Informationen über ein ext2/3/4-Dateisystem aus – Inhalt des Superblocks und der Block Group Deskriptoren . Zeigt ähnlich tune2fs -l Informationen, plus Aufteilung der Blöcke in Gruppen, freie Blöcke/Inodes pro Gruppe etc. Hilfreich für Diagnose (z.B. Backup-Superblock Positionen).

### Syntax:

dumpe2fs [Optionen] <Gerät>

### Optionen:

- **-h** – Zeigt nur den Inhalt des Superblocks und der Gruppen-Deskriptoren (ohne die Liste aller freien Blöcke/Inodes). Das ist die gängigste Verwendung, da ohne -

h die Ausgabe sehr lang sein kann (listet freie Blöcke/Inodes aller Groups[wiki.ubuntuusers.de](https://wiki.ubuntuusers.de)) .

- **-x** – Zeigt auch die detaillierten Gruppen-Informationen (wie freie Blöcke/Inodes-Bitmap). Standard ohne -h tut das sowieso.
- **-b** – Listet die Nummern der freien Blöcke. (Sehr viel Output auf großem FS.)
- **-i** – Listet die freien Inode-Nummern.

#### **Beispiel (gekürzt dumpe2fs -h /dev/sda2):**

Filesystem volume name: <none>

Last mounted on: /

Filesystem UUID: 4e8fea21-...-b6c7

Filesystem magic number: 0xEF53

Filesystem revision #: 1 (dynamic)

Filesystem OS type: Linux

Block size: 4096

Block count: 25600000

Reserved block count: 1280000

Free blocks: 15034567

First Block: 0

...

Journal inode: 8

Journal backup: inode blocks

...

Zeigt Label, UUID, Blocksize, Anzahl Blöcke, Reserve, Anzahl freie etc., und welche Features aktiviert sind.

#### **cryptsetup (LUKS Disk Encryption Setup)**

**Beschreibung:** Tool zur Einrichtung von **verschlüsselten Datenträgern** mit LUKS (Linux Unified Key Setup) sowie zum Öffnen/Schließen solcher Verschlüsselungs-Volume . cryptsetup kann ein Gerät mit LUKS initialisieren, Passphrase setzen, und im Betrieb ein verschlüsseltes Volume als Device-Mapper-Gerät bereitstellen, das dann gemountet werden kann.

#### **Typische Aktionen (Subcommands):**

- **luksFormat** – Initialisiert ein Blockgerät mit LUKS-Verschlüsselungsheader und fordert eine Passphrase zum Einrichten an. Beispiel: cryptsetup luksFormat /dev/sdb1. Achtung: Löscht alle vorhandenen Daten auf dem Gerät.
- **open** oder **luksOpen** – Öffnet ein verschlüsseltes Volume (fragt Passphrase). Syntax: cryptsetup luksOpen /dev/sdb1 geheim\_vol. Dies erstellt ein Device /dev/mapper/geheim\_vol, das nun entschlüsselt agiert und z.B. mit mkfs formatiert oder gemountet werden kann】 .
- **close** – Schließt ein vorher geöffnetes LUKS-Device: cryptsetup close geheim\_vol (dann ist /dev/mapper/geheim\_vol weg).
- **luksAddKey** – Fügt einen weiteren Schlüssel (Passphrase) hinzu. LUKS unterstützt mehrere Key-Slots. cryptsetup luksAddKey /dev/sdb1 /pfad/zur/schlüsseldatei oder interaktiv eine neue Passphrase eingeben (nachdem man eine gültige alte eingibt).
- **luksRemoveKey** – Entfernt einen Schlüssel (kann durch Angabe der zu entfernenden Passphrase oder Slot-Nummer erfolgen).
- **luksChangeKey** – Ändert eine bestehende Passphrase (in einem Slot) durch eine neue (kombi aus Remove+Add effizienter).
- **luksHeaderBackup** / **luksHeaderRestore** – Backup bzw. Wiederherstellen des LUKS-Headers (empfohlen, falls Header korrupt -> Daten verloren).
- **status** – Zeigt Informationen über ein offenes Device-Mapper Volume (Name, verschlüsseltes Dev, Cipher, Keylen, etc.): cryptsetup status geheim\_vol.

#### **Wichtige Optionen:**

- **-y** (im Format-Befehl): fragt neues Passwort zweimal zur Verifizierung ab.
- **-d <Datei>**: verwendet eine Datei als Schlüsseldaten (anstatt interaktiv Passwort). -d - kann genutzt werden, um vom stdin zu lesen.
- **--key-size <bits>**: Schlüsselgröße (in Bits) für die Verschlüsselung (z.B. 256).
- **--cipher <name>**: Zu nutzender Verschlüsselungsalgorithmus und Mode. Standard z.B. aes-xts-plain64 bei LUKS2. Kann z.B. aes-cbc-essiv:sha256 oder anderes angegeben werden.
- **--hash <hash>**: Hash-Algorithmus für Passphrase-Derivation (PBKDF). Standard modern ist z.B. Argon2id oder früher sha256.
- **--iter-time <ms>**: Wie lange (Millisek) auf PBKDF verwendet werden soll (Bestimmt Komplexität).
- **--type luks1/luks2**: LUKS Version festlegen (aktuell Standard LUKS2, kann optional LUKS1 für Kompatibilität gewählt werden).

- **-q:** Quiet (unterdrückt Ausgabe, z.B. Warnung bei Format).

### **Beispiel Workflow:**

1. **Formatieren:** cryptsetup -y luksFormat /dev/sda5 (legt LUKS auf Partition sda5 an, fragt 2x Passwort).
2. **Öffnen:** cryptsetup luksOpen /dev/sda5 geheim (Passwort eintippen, erstellt /dev/mapper/geheim).
3. **Format FS:** mkfs.ext4 /dev/mapper/geheim.
4. **Mount:** mount /dev/mapper/geheim /mnt/secure. Jetzt Daten drauf speichern.
5. **Aushängen+Schließen:** umount /mnt/secure && cryptsetup close geheim.

(LUKS ermöglicht bis zu 8 Passphrases; man kann z.B. mit luksAddKey einen zweiten hinzugeben, sodass entweder Passphrase A oder B das Volume öffnet. Das Gerät /dev/sda5 kann in /etc/crypttab und /etc/fstab eingetragen werden für Boot-Time auto-open via keyfile oder prompt.)

### **shred (Securely wipe file/device)**

**Beschreibung:** Überschreibt eine Datei oder ein ganzes Gerät mehrfach mit zufälligen Daten, um ein Wiederherstellen gelöschter Inhalte zu erschweren. shred ist nützlich zum sicheren Löschen auf magnetischen HDDs. Bei SSDs mit Wear-Leveling ist Effekt begrenzt – dort lieber ATA Secure Erase oder physical destruction.

#### **Syntax:**

shred [Optionen] <Datei1> [Datei2...]

#### **Optionen:**

- **-n N** – Anzahl der *zusätzlichen* Überschreib-Pässe (Standard 3). D.h. 3 bedeutet insgesamt 3 mal random + final pass (siehe -z).
- **-z, --zero** – Überschreibt am Ende noch einmal mit Null-Bytes, um das Muster zu verwischen (anstatt dass zuletzt zufälliges Rauschen bleibt). Standard: aktiv (bei neueren shred ist -z default an).
- **-f, --force** – Ändert Schreibschutz falls nötig (chmod u+w), um schreiben zu können.
- **-u, --remove[=how]** – Nach dem Überschreiben die Datei löschen (unlink). Optional how kann unlink (Standard) oder wipe oder wipesync sein: *wipe* überschreibt erst mit Null dann löscht, *wipesync* macht zusätzlich Sync. Standard -u bereits tut wipesync.
- **-v, --verbose** – Auskunft über Fortschritt.

- **-x, -X, --exact** – Überschreibt Datei genau Byte-für-Byte anstelle Dateigröße auf nächste Blockgröße aufzurunden (Standard shred nutzt ganze Blocks, kann über Dateiende hinaus schreiben falls Dateigröße nicht mod Blocksize). *Exact* vermeidet das, aber hinterlässt evtl. un-geschredderte Slack am Ende.
- **-s N, --size=N** – Behandelt die Datei als N Byte lang (statt aktuelle Länge). Um auch ungenutzten Platz (z.B. Slack Space/EOF beyond) im Inode zu überschreiben, kann man etwas größer angeben.

### **Beispiele:**

- `shred -u file.txt` – Überschreibt *file.txt* viermal (3 rand + 1 zero) und löscht sie anschließend.
- `shred /dev/sdb` – Überschreibt komplett `/dev/sdb` mit 3 Runden Zufallsdaten (zerstört alle Daten/Partitionen darauf).
- `shred -n 0 -z -u secret.dat` – Einmal mit Nullen überschreiben und entfernen (hier wollte man evtl. nur einfaches Überschreiben und löschen).

**Warnung:** Shred wirkt nicht zuverlässig auf: **logischen Volumes** in Files (z.B. Datei in FS kann redundante Kopien (Journal, Snapshots) haben), **RAID, SSD,**

**Netzwerkdateisysteme, Dateisysteme mit Copy-on-Write (btrfs, ZFS)** – dort können Daten an mehreren Orten stehen. In solchen Fällen besser Vollverschlüsselung verwenden oder das gesamte physische Medium überschreiben.

### **iostat (IO Statistics)**

**Beschreibung:** Zeigt Statistik zur Systemauslastung von CPU und Input/Output für Blockgeräte (Datenträger). iostat gehört zum Paket sysstat. Es hilft zu sehen, wie stark Platten ausgelastet sind (%util) und Durchsatz (MB/s), IOPS etc., sowie generelle CPU-Idle/Wait Zeite .

### **Syntax:**

`iostat [Optionen] [Intervall [Durchläufe]]`

Ohne Argument gibt einen Durchschnitt seit Boot aus. Mit Intervall (Sekunden) aktualisiert regelmäßig.

### **Wichtige Optionen:**

- **-c** – Zeigt nur CPU-Statistiken.
- **-d** – Zeigt nur Disk (Geräte) Statistiken.
- **-k, -m** – Ausgabe in KiB/s oder MiB/s (anstatt Blöcke/s). (Aktuelle sysstat erlaubt -block-size).

- **-p [Gerät]** – Detaillierte Partitions-Statistiken einbeziehen (normal zeigt iostat nur Whole-Device). Z.B. -p sda zeigt sda und sda1,sda2,...
- **-x** – Ausgabe in erweiterter Form (Pro Gerät: zusätzlich Auslastungsprozentsatz, Average Request Size, Await Times etc.) . Sehr nützlich, da Standard ohne -x recht knapp.
- **-t** – Zeigt Timestamp vor jeder Ausgabe.
- **-y** – Unterdrückt erste Ausgabe (since boot), beginnt erst mit Werten ab erstem Intervall.

#### **Interpretation (bei iostat -x):**

Spalten pro Device: rrqm/s wrqm/s (Merge von I/O, veraltet bei modernen Kernel meist 0), r/s w/s (reads/writes per sec), rkB/s wkB/s (Throughput), avgrq-sz (Durchschnittl. Requestgröße in Sektoren oder KB), avgqu-sz (durchschn. Queue length), await (Durchschnittliche Wartezeit pro IO ms), svctm (durchschn. Servicezeit pro IO ms – oft nicht mehr zuverlässig), %util (Prozentuale Zeit, die das Gerät aktiv war innerhalb der Messperiode – 100% bedeutet permanent beschäftigt).

#### **Beispiel:**

```
Device r/s w/s rkB/s wkB/s avgrq-sz avgqu-sz await svctm %util
sda 12.5 30.2 500.0 250.0 40.0 0.5 10.0 2.00 85.0
```

Hier sda ~85% ausgelastet, wartet ~10ms per IO, etwa 42.7 IO/s (r+s) mit ~750kB/s.

#### **ioping (I/O Ping)**

**Beschreibung:** Ermittelt Latenz (und ggf. Durchsatz) von Storage in Echtzeit ähnlich dem Prinzip von ping im Netzwerk. Es führt schnelle sequenzielle oder wahlfreie Lesezugriffe durch und misst deren Zeit. Praktisch um Latenz auf Disks oder bestimmten Verzeichnissen (Dateisystem) festzustellen.

#### **Syntax:**

```
ioping [Optionen] <Ziel>
```

Ziel kann ein Pfad (Datei oder Verzeichnis) sein für Filesystem IO, oder z.B. ein Blockgerät (mit -D für Direkten Device Zugriff) oder sogar ein bestimmter I/O-Engine (IO depth).

#### **Wichtige Optionen:**

- **-c <count>** – Anzahl der Ping-Requests (Standard unendlich, bis abgebrochen).
- **-i <interval>** – Wartezeit zwischen Pings (Sekunden, default 1s). Kann in ms angeben z.B. -i 0.1.

- **-s <bytes>** – Größe der zu lesenden Daten pro Ping (Standard 4KiB).
- **-R** – Ausgabe im Rohformat (Maschinenlesbar).
- **-D** – Direct-Device Mode: Arbeitet auf Blockgeräte direkt (bypasses FS-Cache).
- **-A** – Ausgabe als Aggregatstatistik nur (kein laufendes Ping, nur Summary).
- **-S** – Sync/Sequenziell: testet sequenzielles Lesen statt wahlfrei. (Standard ioping macht random position reads).
- **-W** – führt Schreib-Tests durch statt Lesen (Achtung kann Daten überschreiben zerstören, am besten auf extra Gerät/Datei).
- **-P <N>** – Parallel Requests (Thread count).
- **-q** – keine laufende Ausgabe, nur Summary am Ende.

### **Beispiele:**

- ioping . – Sendet kontinuierlich 4K-Leseanfragen ins aktuelle Verzeichnis (random in FS), zeigt Latenzen, bis Ctrl+C.

4096 bytes from . (ext4 /dev/sda2): request=1 time=0.47 ms

4096 bytes from . (ext4 /dev/sda2): request=2 time=0.44 ms

... (zeigt typisch ~0.5ms auf SSD, ~10-15ms auf HDD) ...

Nach Abbruch gibt es eine **Summary** mit min/avg/max/mdev (Jitter).

- ioping -c 10 /mnt/backup – Misst 10 Zugriffe im FS /mnt/backup (z.B. externer USB), dann stop.
- ioping -R /dev/sdb – Führt 1 Sekunde lang Messungen auf /dev/sdb durch und gibt am Ende aggregated IOPS und Bandbreite.
- ioping -A -S -c 50 /file.iso – Liest sequenziell 50 Anfragen durch große Datei und gibt am Ende Durchschnittszeit.

(Tool muss ggf. installiert werden, gehört nicht standard zu allen Distr.)

### **Geräte, Prozesse, Speicher und Kernel**

#### **hwinfo (Hardware Info)**

**Beschreibung:** Liefert detaillierte Informationen zur Hardware des Systems. hwinfo probiert, alle verfügbaren Hardware-Komponenten aufzulisten: CPU, RAM, Festplatten, USB, PCI-Geräte, Netzwerk, Monitor, etc., inklusive ihrer Eigenschaften und Treiber. Oft auf openSUSE installiert, aber auch auf anderen Distros verfügbar.

#### **Syntax:**

`hwinfo [Optionen] [Hardwareklasse]`

#### **Optionen / Klassen:**

- Ohne Angabe gibt `hwinfo` *alles* aus (sehr lang). Besser, man filtert nach Klasse: z.B. `hwinfo --cpu`, `hwinfo --disk`, `--gfxcard`, `--netcard`, `--printer`, `--usb`, `--bios`, `--memory` etc.
- **--short:** Kurzfassung ausgeben (ein Eintrag pro Gerät in einer Zeile).
- **--brief:** Noch knapper (ähnlich short).
- **--summary:** Summaries (z.B. nur Zahl der Disks, CPUs etc.).
- **--verbose <level>:** Detaillierungsgrad (0-9). Default 1 (schon sehr ausführlich).
- **--debug N:** Setzt Debug-Level (für Entwickler, welche Module `hwinfo` benutzt).
- **--hwitem <ID>:** Sucht nach Hardware mit bestimmter ID (z.B. PCI-ID).
- **--save-config <datei> / --load-config <datei>:** Speichert erkannte Hardware in Datei zum späteren Vergleich.

*Beispiel:* `hwinfo --cpu` könnte ausgeben:

01: None 00.0: 10103 CPU

[Created at cpu.462]

Unique ID: rdCR.jX8Un+o2ZT1

Hardware Class: cpu

Arch: X86-64

Vendor: "GenuineIntel"

Model: 6.158.10 "Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz"

... (viele weitere Details Frequenzen, Cache etc.) ...

`hwinfo --short` summarisiert z.B. Hauptgeräte und Partitionen.

#### **dmidecode (DMI Table Decode)**

**Beschreibung:** Gibt Informationen aus dem BIOS/UEFI-DMI (SMBIOS) heraus, wie Hersteller, Produktnamen, Seriennummern von System/Mainboard, BIOS-Version, verbauter Speicher, Anschlüsse etc. . Hilfreich für Hardware-Inventarisierung (Seriennummer, RAM-Module, OEM strings).

#### **Syntax:**

`dmidecode [Optionen]`

## Optionen:

- **-t <string|Nummer>, --type**: Nur bestimmte DMI-Typen ausgeben. Typ kann Name oder Zahl sein. Z.B.:
  - bios (DMI type 0),
  - system (1),
  - baseboard (2),
  - processor (4),
  - memory (Memory Device, 17),
  - cache (7),
  - connector (8), slot (9), etc.  
Man kann -t memory um nur RAM-Infos zu sehen (Größe, Typ, Hersteller, Seriennummer der DIMMs) .
- **-s <keyword>, --string**: Einzelne SMBIOS-Einträge gezielt ausgeben (z.B. bios-vendor, bios-version, system-manufacturer, system-serial-number, baseboard-asset-tag, processor-frequency etc.).
- **-q**: Quiet (unterdrückt weniger wichtige DMI-Eintrags-Labels wo "Not Provided").
- **-u**: Raw dump (hex) der DMI-Daten.
- **--dump-bin file, --from-dump file**: Dump SMBIOS in Datei bzw. lese aus Dump (z.B. offline Analyse).

Beispiel: dmidecode -t system könnte zeigen:

System Information

Manufacturer: Dell Inc.

Product Name: Latitude 5490

Version: None

Serial Number: ABCD123

UUID: 4c4c4544-0034-5810-8055-b6c04f513332

Wake-up Type: Power Switch

SKU Number: 07E6

Family: Latitude

dmidecode -t memory zeigt pro RAM-Riegel z.B. Größe, Form Factor (SODIMM), Type (DDR4), Speed, Manufacturer, Serial, Part Number.

## **lspci (List PCI Devices)**

**Beschreibung:** Zeigt alle PCI/PCIe-Geräte im System (angeschlossene Hardware am PCI-Bus: z.B. Netzwerkkarten, Sound, USB-Controller, Grafikkarten, SATA-Controller etc.). Es listet Geräte-ID und -Namen anhand einer Datenbank (/usr/share/misc/pci.ids).

### **Syntax:**

`lspci [Optionen]`

### **Optionen:**

- **-v, -vv, -vvv:** Verbose-Ausgabe in ein oder mehreren Stufen (mehr Register-Informationen). -vvv kann *sehr* ausführlich sein (inkl. Capabilities, Konfigspace).
- **-k:** Zeigt Kernel-Treiber und Module, die für jedes Gerät verwendet werden (welcher Treiber zugeordnet ist). (Z.B. Kernel driver in use: e1000e für Intel NIC).
- **-t:** Zeigt die hierarchische Baumstruktur der PCI-Bus-Topologie (wer hängt an welchem Bus/Bridge).
- **-nn:** Zeigt neben dem Gerätenamen auch Vendor:Device ID in eckigen Klammern. -n nur numeric IDs ohne Namen. -nn= beide. Z.B. Ethernet Controller [8086:1533] etc.
- **-d [<vendor>]:[<device>]:** Filtert nach bestimmten PCI IDs. `lspci -d 8086`: zeigt nur Intel-Geräte, `-d :1533` nur Geräte mit Device 1533.
- **-s [<domain>]::<bus>:<slot>[.<func>]:** Filtert nach Bus-Adresse (wie in `lspci -t` oder `/sys/bus/pci`). Z.B. `-s 00:1f.2` zeigt nur dieses Gerät.
- **-m:** Maschinenlesbar (gibt in einer CSV-artigen Formatzeile pro Gerät: Domain, Bus Slot, Klass, Vendor, Device, SVendor, SDevice).
- **-xxx:** Dump des gesamten PCI-Konfigurationsspeichers (64 Bytes beyond standard) – Vorsicht sensibel. -xx nur ersten 256 Bytes (std config).
- **-H1:** Nutzen von `/proc/bus/pci` statt direct (historisch, meist nicht nötig).
- **-Q:** Zeigt das Querier-Modell (Kernel vs direct vs `/proc`).

### **Beispiel Ausgabe (gekürzt):**

00:00.0 Host bridge: Intel Corporation Device 5914

00:02.0 VGA compatible controller: Intel Corporation UHD Graphics 620

00:14.0 USB controller: Intel Corporation Sunrise Point-LP USB 3.0 xHCI Controller (rev 21)

00:16.0 Communication controller: Intel Corporation Management Engine Interface (rev 21)

...

03:00.0 Network controller: Intel Corporation Dual Band Wireless-AC 8265 (rev 78)

Mit -nn würde man z.B. sehen 00:02.0 VGA controller [0300]: Intel Corporation UHD Graphics 620 [8086:5917]. Mit -k ergänzt: Kernel driver in use: i915.

## **lsusb (List USB Devices)**

**Beschreibung:** Listet USB-Geräte auf, die am System angeschlossen sind (inkl. interne Hubs etc.). Zeigt Vendor und Produktnamen falls bekannt (Datenbank /usr/share/misc/usb.ids) und ID.

### **Syntax:**

lsusb [Optionen]

### **Optionen:**

- **-v:** Verbose, zeigt detaillierte Descriptor-Infos pro Gerät (ähnlich lsusb -vv noch detaillierter). Das kann sehr lang sein.
- **-t:** Baum-Darstellung der USB-Geräte-Hierarchie (welches Device hängt an welchem Port/Hub) – ähnlich lspci -t.
- **-s <bus>:<dev>:** Zeigt nur spezifisches Gerät (per Bus und Device Nummer). Diese Nummern sieht man in lsusb Standardausgabe vorn.
- **-d vid:pid:** Filtert nach VendorID:ProductID (Hex).
- **-D <device-file>:** Liest Infos aus einer Geräte-Datei (z.B. /dev/bus/usb/001/007) – erfordert Root.
- **-w:** Ausgabe als mit usbdump kompatibler Text (Wire format).
- **-v -v (or -vv):** liefert noch detailliertere Dump (inkl. Hexdump der HID Deskriptoren etc.).

### **Beispiel Ausgabe:**

Bus 002 Device 003: ID 0bda:0316 Realtek Semiconductor Corp. Card Reader

Bus 002 Device 002: ID 8087:0aaa Intel Corp.

Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub

Bus 001 Device 004: ID 0c45:671f Microdia Integrated Webcam

Bus 001 Device 003: ID 8087:07dc Intel Corp. Bluetooth wireless adapter

Bus 001 Device 002: ID 056e:4008 Elecom Co., Ltd

Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

Hier z.B. Card Reader, Webcam, Bluetooth, eine Maus (056e:4008 Elecom), plus Root Hubs.

Mit -t etwa:

/: Bus 02.Port 1: Dev 1, Class=root\_hub, ... (bus 2, 3.0)

|\_\_ Port 1: Dev 2, Intel USB 3.0 Hub

  |\_\_ Port 2: Dev 3, Realtek Card Reader

...

Zeigt Topologie (Hub etc.).

## **lscpu (List CPU Info)**

**Beschreibung:** Zeigt Informationen über die CPU(s) des Systems – Anzahl Kerne, Threads, Sockets, Modellname, Frequenz, Virtualisierungsfunktionen uswfile-4fdk7rvx39azjretinarqb】 .

### **Syntax:**

`lscpu [Optionen]`

### **Optionen:**

- **-e, --extended [=list]:** Ausgabe als Tabelle mit Spalten (erweiterte Auflistung jeder logischen CPU mit Eigenschaften). Optional kann man angeben, welche Felder (z.B. CPU, Node, Socket, Core, Online, ...).
- **-p, --parse [=list]:** Ähnlich --extended aber kommagetrennt (parseable). Standardfelder: CPU,Core,Socket,Node.
- **-x, --hex:** Zeigt hex-Werte wo sinnvoll (z.B. CPU bitmasken).
- **-c, --offline / --online:** Zeigt nur offline oder nur online CPUs (bei Hotplug-Systemen).
- **-y, --physical:** Zeigt nur physische IDs (keine virtuellen IDs).
- **-J, --json:** JSON-Ausgabe (maschinenlesbar).
- **-B, --byte:** Einheit Bytes statt kB für Cache-Größen.
- **-C, --cache:** Zeigt Details zu CPU-Caches (Anzahl, Größen).

*Beispiel (gekürzt lscpu Standard):*

Architecture: x86\_64

CPU op-mode(s): 32-bit, 64-bit  
Sockets: 1  
Cores per socket: 4  
Threads per core: 2  
Model name: Intel(R) Core(TM) i7-8565U @ 1.80GHz  
CPU MHz: 2112.004  
Virtualization: VT-x  
L1d cache: 32K  
L2 cache: 256K  
L3 cache: 8192K  
Flags: fpu vme ... vmx ...

Zeigt also 1 Sockel, 4 Kerne, 8 Threads (4\*2), CPU-Modell und Takt etc.

### **lsmem (List Memory Blocks)**

**Beschreibung:** Zeigt Speicherblock-Informationen des Systems (insbesondere auf Systemen mit Hotpluggable Memory oder NUMA). Es listet Memory-Block-Geräte (wie in /sys/devices/system/memory) auf. Auf x86-PCs weniger relevant; auf IBM System z oder Power etc. nützlich.

#### **Syntax:**

`lsmem [Optionen]`

#### **Optionen:**

- **-n, --online / -o, --offline:** Zeigt nur online bzw. offline Speicherblöcke.
- **-b, --bytes:** Größen in Bytes statt lesbar.
- **-M, --mega:** Größen in MiB. (Standard lsmem gibt in bytes + kMG suffix)
- **-p, --pairs:** Ausgabe im Key:Value Format (für maschinelles Parsing).
- **-J, --json:** JSON-Format.
- **--output <list>:** Spalten auswählen (z.B. --output SIZE,NODE,STATE).
- **--help, --version:** analog.

*Beispiel lsmem (auf PC ohne Hotplug memory):*

RANGE	SIZE STATE REMOVABLE BLOCK
-------	----------------------------

```
0x0000000000000000-0x000000007fffffff 2G online yes 0-15
```

```
0x0000000100000000-0x000000027fffffff 6G online yes 32-79
```

Zeigt 2 memory ranges (entsprechend wohl 2 DIMMs). "Removable yes" heißt vom Kernel als Hot-remove möglich (oft 'no' auf Standard-PC). "BLOCK 0-15" referenzieren Memory block device IDs.

Auf NUMA-Systemen würde Spalte NODE relevant sein, um Memory pro NUMA-Node zu sehen.

### **ps (Process Status)**

**Beschreibung:** Zeigt eine Momentaufnahme der aktuellen Prozesse des Systems. ps kann verschiedene Ausgabeformate haben (BSD-Style, UNIX-style). Häufig nutzt man ps aux oder `ps -effile-4fdk7rvx39azjretinarqb】 .

#### **Syntax:**

ps [Optionen]

(Komplexität: ps hat UNIX Optionen (mit - und bündelbar), BSD Optionen (ohne - und können zusammen geschrieben werden) und GNU long Optionen. Z.B. ps aux ist BSD Syntax, ps -ef ist UNIX syntax – beide gängig, liefern ähnliches.)

#### **Gängige Aufrufe:**

- **ps aux** – Listet alle Prozesse mit Details (BSD format). Spalten: USER, PID, %CPU, %MEM, VSZ, RSS, TT (TTY), STAT, START, TIME, COMMAND.
- **ps -ef** – Listet alle Prozesse (UNIX format). Spalten: UID, PID, PPID, C (CPU usage), STIME (start time), TTY, TIME (CPU time), CMD.
- **ps -eF** – wie -ef, aber "extra full", zeigt threads als separate Linien und vollständige Kommandozeilen etc.
- **ps -ejH** – zeigt Hierarchie (Prozessbaum) mit Prozess-IDs.
- **ps -eo pid,cmd,%mem,%cpu --sort=-%mem** – Zeigt benutzerdefinierte Spalten (hier pid, Befehl, Speicher- und CPU% sortiert nach Speicherverbrauch absteigend).

#### **Wichtige Optionen:**

UNIX-style (präfix mit -, können kombiniert werden):

- **-e** – Alle Prozesse anzeigen (entspricht -A).
- **-f – full** Format (mehr Spalten inkl. PPID, STIME, in Kombination z.B. -ef).
- **-F – Extra full** (noch mehr Infos: SZ, RSS, PSR CPU, etc.).

- **-h** – Ohne Header.
- **-H** – Hierarchie-Tree anzeigen (Indentation für Parent-Child).
- **-j** – Jobs format (PGID, SID etc. Spalten).
- **-l** – Long format (mehr spalten inkl. Flags, NI nice, LWP etc.).
- **-u <user>** – Prozesse eines bestimmten Benutzers.
- **-p <pidlist>** – Nur bestimmte PIDs.
- **-G <group>, -U <user>** – nach Gruppe/Benutzer filtern.

BSD-style (ohne -, einzeln hintereinander):

- **a** – Zeigt Prozesse von *allen Terminals* (nicht nur eigene).
- **x** – Zeigt auch prozesse *ohne Terminal* (Daemons). **ax** zusammen = alle Prozesse.
- **u** – Benutzerformat (inkl. Owner, CPU%, MEM%). => **aux** verbreitet.
- **j** – Jobs format (session, pgrp).
- **w** – breites Output (alle columns of command, use -ww to not truncate).

GNU long options (mit --):

- **--pid, --ppid, --user** etc. Alternative Filter.
- **--sort=<key>**: Sortiert Ausgabe nach Spalte(n). Z.B. --sort=-%cpu (absteigend CPU).
- **--no-headers**: Ohne Header (statt -h).
- **--forest**: ASCII-Art Bäumchen (ähnlich -H, aber macht "|\_" und Einrückungen). Kombiniert mit e.g. -ef.

### **STAT Spalte (Prozessstatus):**

Codes: R (running), S (sleep), D (uninterruptible sleep, i/o), Z (zombie), T (stopped), t (gestoppt im Hintergrund), X (dead). Flags dahinter: < (high prio), N (low prio, nice>0), L (pages locked in mem), s (session leader), + (foreground process group in TTY).

## **sar (System Activity Reporter)**

**Beschreibung:** Teil von sysstat (wie iostat, mpstat, etc.), sammelt und berichtet historische System-Auslastungsdaten: CPU, Speicher, Paging, I/O, Netzwerk etc. sar kann *laufende* Stats anzeigen (wenn mit Interval aufgerufen), oder *vergangene Tageswerte* aus Logs (/var/log/sysstat/...) ausgeben.

### **Syntax:**

`sar [Optionen] [Interval [Count]]`

Ohne Parameter, sar liest per default die Tagesstatistiken (cumulated) vom laufenden Tag (falls sar cron job aktiv war). Mit Interval sar 5 5 z.B. 5x alle 5 Sekunden CPU etc.

### Hauptoptionen (für Bereiche):

- **-u:** CPU-Auslastung (Default, wenn keine angegeben) – zeigt %user, %nice, %system, %iowait, %steal, %idle.
- **-r:** Memory (free, used, buffers, cache, commit, etc.).
- **-b:** I/O (transfer rates, i/o per sec, etc.).
- **-B:** Paging (pgpgin/s, pgpgout/s, fault, majflt...).
- **-W:** Swapping (swap in/out).
- **-q:** Queue length (runq, i/o queue) and load average.
- **-n DEV:** Network: DEV mode – zeigt Netzwerk-Interface Stats (rxpck/s, txpck/s, rxKB/s, txKB/s, etc.). -n ALL für alle Net-Statistiken.
- **-d:** Block device I/O (similar to iostat per device).
- **-p [DEV]:** CPU per processor (if SMP). e.g. sar -P ALL 1 3 – CPU usage per core.
- **-f file:** Lese Stats aus Log-Datei (z.B. /var/log/sysstat/sarXX).
- **-s HH:MM:SS, -e HH:MM:SS:** Start/End time for reading logs (z.B. 9 bis 18 Uhr Stats).

### Beispiel (aktuelle CPU alle 1s):

```
$ sar 1 3
```

Linux 5.4.0 ... (8 CPUs)

```
02:00:00 PM CPU %user %nice %system %iowait %steal %idle
02:00:01 PM all 5.13 0.00 1.28 0.00 0.00 93.59
02:00:02 PM all 7.14 0.00 1.02 0.00 0.00 91.84
02:00:03 PM all 3.06 0.00 0.77 0.00 0.00 96.16
Average: all 5.11 0.00 1.02 0.00 0.00 93.87
```

### nohup (No Hangup)

**Beschreibung:** Lässt einen Prozess auch nach Logout weiterlaufen. nohup fängt das HUP-Signal (Hangup) ab, das normalerweise gesendet wird, wenn ein Terminal geschlossen wird. Mit nohup startet man ein Kommando so, dass es dieses Signal ignoriert und Standardausgabe ggf. in *nohup.out* umleite】 .

## Syntax:

```
nohup <Befehl> [Arg...] [&]
```

## Funktionsweise:

- nohup sorgt dafür, dass der ausgeführte Prozess das SIGHUP ignoriert. Außerdem werden *stdout* und *stderr* umgeleitet in *nohup.out*, falls diese derzeit ans Terminal gebunden sind. (Wenn man & anhängt – also im Hintergrund – ist es üblich, die Ausgabe nicht am Terminal zu lassen).
- Üblicherweise kombiniert mit & (Hintergrund). Man loggt dann aus, und das Programm läuft weiter.

## Beispiele:

- nohup long\_running.sh & – Führt *long\_running.sh* im Hintergrund aus, Ausgaben gehen in *nohup.out*. Der Befehl überlebt Logout. Die Shell gibt z.B. zurück: appending output to nohup.out.
- nohup python server.py > server.log 2>&1 & – Startet Python-Server im Hintergrund, leitet *stdout* und *stderr* in *server.log* (damit *nohup.out* nicht genutzt).

*Hinweis:* Wenn man bereits in einer Screen- oder tmux-Session ist, braucht man kein nohup; auch systemd-run oder Cron-Jobs sind Alternativen. Aber nohup ist ein schnelles Mittel.

## **mkswap/swapon/swapoff (Swap einrichten und verwalten)**

- **mkswap** – Initialisiert eine Partition/Datei als Swapbereich. Syntax: mkswap [Optionen] <Gerät/datei>. Wichtige Option: -L <Label> (Swap-Label setzen), -U <UUID>. Beispiel: mkswap /dev/sda3.
- **swapon** – Aktiviert einen Swapbereich für Nutzung durch das System. Nach mkswap muss Partition via swapon <Device> eingebunden werden (oder in /etc/fstab eingetragen mit swap Typ). Option -a aktiviert alle in fstab markierten Swaps. --show zeigt aktuellen Swap an (Äquivalent cat /proc/swaps).
- **swapoff** – Deaktiviert Swap. Syntax: swapoff <Device> oder swapoff -a für alle.

## Beispiele:

```
mkswap -L "SWAP" /dev/sda3
```

```
Setting up swapspace version 1, size = 8 GiB (some hex UUID)
```

```
swapon /dev/sda3
```

```
swapon --show
```

```
NAME TYPE SIZE USED PRIO
```

```
/dev/sda3 partition 8G 0B -2
```

swapoff -a würde alle Swapbereiche aus dem Betrieb nehmen (z.B. vor Partition-Vergrößerung oder Kernel-Dump etc.).

### **free (Speicherübersicht)**

**Beschreibung:** Zeigt eine Übersicht über den aktuellen Speicherverbrauch (RAM und Swap) im System. Es summiert total, used, free, and cached/buffered memory.

#### **Syntax:**

```
free [Optionen]
```

#### **Optionen:**

- **-h:** Human Readable (z.B. MiB/GiB).
- **-m, -g, -k:** Ausgabe in MiB, GiB, KiB.
- **-b:** in Bytes.
- **-t:** Fügt eine "Total" Zeile (RAM+Swap) hinzu.
- **-s N:** Aktualisiert alle N Sekunden (kontinuierliche Ausgabe wie watch).
- **-c N:** Wenn mit -s genutzt, stop nach N Aktualisierungen.
- **--si:** Benutzt 1000er statt 1024er Basis (KB = 1000 Bytes etc.).

#### **Beispiel free -h:**

	total	used	free	shared	buff/cache	available
Mem:	15Gi	5.2Gi	2.3Gi	1.1Gi	7.4Gi	8.8Gi
Swap:	2.0Gi	0.0Gi	2.0Gi			

available ist eine Kernelschätzung, wieviel noch für neue Prozesse verfügbar, unter Berücksichtigung von caches, etc.

### **vmstat (Virtual Memory and system stats)**

**Beschreibung:** Zeigt verschiedene Systemmetriken: Speicher, Auslagerung, CPU, Interrupts, Context Switches etc. im zeitlichen Verlauf. vmstat ohne Parameter gibt Durchschnitt seit Boot; mit Intervall zeigt periodische Werte (Differenzen pro Intervall).

#### **Syntax:**

```
vmstat [Optionen] [Interval [Count]]
```

#### **Wichtige Spalten (Standardausgabe):**

- **procs:** *r* (Anzahl laufender Prozesse im Runqueue), *b* (blockierte Prozesse).
- **memory:** *swpd* (Swap genutzt), *free* (freier RAM), *buff* (Puffer), *cache* (Page Cache).
- **swap:** *si* (Swap-In KB/s), *so* (Swap-Out KB/s).
- **io:** *bi* (Block In, Blöcke gelesen von Disk/s), *bo* (Block Out, geschrieben/s).
- **system:** *in* (Interrupts/s), *cs* (Context Switches/s).
- **cpu:** *us* (% CPU in Userzeit), *sy* (% Systemzeit), *id* (% Idle), *wa* (% IO-Wait), *st* (% Steal bei virt.).

### Optionen:

- **-S <unit>**: Werte in k (1000) oder K (1024) etc.
- **-a**: Zeigt zusätzlich Aktivität der *aktiven/inaktiven* Memory Listen (sin, sout).
- **-d**: Zeigt Disk-Statistiken (like iostat) statt VM stats.
- **-p <dev>**: Zeigt Partition-Statistiken (reads/writes tot und sek) für Device.
- **-s**: Summiert memory stats (eine Liste aller relevanten Zähler, analog /proc/meminfo).
- **-m, -t**: Variation der Ausgabe (meist uninteressant, -m memory in MB etc).

### Beispiel (alle 2s):

```
$ vmstat 2 3

procs -----memory----- swap-- -----io---- -system-- -----cpu-----
r b free buff cache si so bi bo in cs us sy id wa st
1 0 230000 120000 760000 0 0 5 10 150 300 5 191 3 0
0 0 229000 120000 760500 0 0 0 20 140 280 4 194 1 0
1 0 228500 120000 761000 0 0 0 30 160 320 6 192 1 0
```

Erste Zeile nach Header ist seit Boot (ohne parameter, oft uninteressant); weitere sind Echtzeit pro Intervall.

### modinfo (Module Information)

**Beschreibung:** Zeigt Informationen über ein Linux-Kernelmodul (Loadable Kernel Module) – Version, Beschreibung, Lizenz, Abhängigkeiten, Parameter etc】 .

### Syntax:

modinfo [Optionen] <Modulname oder Modul.ko>

## Optionen:

- **-a, --author:** Nur Autor ausgeben.
- **-d, --description:** Nur Beschreibung.
- **-n, --filename:** Zeigt Pfad der Moduldatei im Filesystem.
- **-p, --parameters:** Listet die Parameter und ihre Typen/Beschreibungen.
- **-F <field>:** Gibt nur das spezifizierte Feld aus (alternativ obige Kurzoptionen). Mögliche Felder: author, description, license, parm, alias, etc.
- **-k <kernelversion>:** Fragt Infos für Kernel-Version (sucht in /lib/modules/<version>) statt laufendem Kernel.

**Beispiel:** modinfo e1000e (Intel NIC Treiber) gibt etwa:

```
filename: /lib/modules/5.4.0/kernel/drivers/net/ethernet/intel/e1000e/e1000e.ko
version: 3.2.6-k
license: GPL
description: Intel(R) PRO/1000 Network Driver
author: Intel Corporation, <e1000-devel@lists.sourceforge.net>
...
srcversion: 5C6D1A8....
alias: pci:v00008086d000010D3sv*sd*bc*sc*i*
...
parm: IntMode:Interrupt Mode (0=Auto, 1=MSI, 2=MSI-X) (int)
parm: InterruptThrottleRate:... (array of int)
```

Man sieht Dateipfad, Version, Parameter etc.

## insmod/rmmod/modprobe (Kernelmodule laden/entladen)

- **insmod <pfad/modul.ko> [args]:** Lädt ein Modul in den Kernel. Nutzt genau Pfad; löst keine Abhängigkeiten auf.
- **rmmod <modulname>:** Entfernt ein geladenes Modul (wenn kein anderer in Benutzung). Option -f zum Erzwingen (nur falls Kernel CONFIG\_MODULE\_FORCE\_UNLOAD).
- **modprobe [Options] <modulname> [args]:** Empfehlenswerter Befehl zum Laden eines Moduls und seiner Abhängigkeiten. Es durchsucht

/lib/modules/<kernel>/modules.dep nach Dependencies. Auch zum Entfernen via modprobe -r <modul> (lädt auch Abhängigkeiten aus, falls unbenutzt).

- Optionen: --dry-run oder -n testet nur, -v verbose, -q quiet, alias handling etc. modprobe Konfig steht in /etc/modprobe.d.
- Bsp: modprobe e1000e (lädt Intel NIC modul + ggf. benötigte mii). modprobe -r e1000e entfernt es.

## depmod (Dependency modules)

**Beschreibung:** Generiert die Modulabhängigkeitsdatei *modules.dep* basierend auf Modulen in /lib/modules/<version>. Wird meist beim Kernel-Install aufgerufen. Kann man manuell ausführen nach Hinzufügen eigener Module. Syntax: depmod -a (für alle).

### Optionen:

- -a, --all (default): alle Kernel-Versionen verarbeiten.
- -A, --quick: nur verändern, wenn *modules.dep* älter als Module-Dateien (incremental update).
- -n: Ausgabe nach stdout statt in *modules.dep* schreiben (Test).
- -v: verbose.

## sysctl (Query/Set Kernel Parameters)

**Beschreibung:** Anzeige oder Änderung von Kernel-Laufzeitparametern (in /proc/sys). sysctl ermöglicht bequemen Zugriff auf z.B. Netzwerktruning, VM, FS Settings zur Laufzeit.

### Syntax:

sysctl [Optionen] [<Name>=<Wert> ...] or sysctl -p

### Optionen:

- Ohne =: liest Parameter. Z.B. sysctl net.ipv4.ip\_forward -> net.ipv4.ip\_forward = 0.
- Mit name=value: setzt Parameter. Z.B. sysctl net.ipv4.ip\_forward=1 aktiviert IP-Forwarding (sofort).
- -w: explizit Schreibmodus (kann man weglassen, war historisch).
- -p [Datei]: Lädt Einstellungen aus Datei (Default /etc/sysctl.conf) und wendet an. Man ruft sysctl -p typ. beim Boot (init-Skripte) oder manuell nach Bearbeiten der config.
- -a: Alle Parameter anzeigen (riesige Liste). sysctl -a | grep ipc etc.
- -A, --pattern <regex>: Filtert mit Regex (statt manuell grep).

- **-e:** Ignoriert unbekannte Keys (useful mit -p, sonst bricht ab).
- **-N:** Nur Namen, keine Werte anzeigen (list keys).
- **-q:** Quiet (unterdrückt Bestätigungsmeldungen).

Parameter-Keys entsprechen /proc/sys Pfaden, z.B. /proc/sys/net/ipv4/ip\_forward -> key net.ipv4.ip\_forward.

### **Beispiele:**

- `sysctl vm.swappiness` -> zeigt swappiness (z.B. `vm.swappiness = 60`).
- `sysctl -w vm.swappiness=10` -> setzt auf 10 (weniger Swapneigung).
- `sysctl -p /etc/sysctl.d/99-sysctl.conf` -> lädt Parameter aus der angegebenen Datei.

## **dmesg (Dump kernel message buffer)**

**Beschreibung:** Zeigt die Nachrichten des Kernel-Ringspeichers (Kernellog) an. Hier stehen Boot-Meldungen und Hardware-Events (z.B. USB eingesteckt, Treiber Ausgabe). dmesg ohne Optionen druckt alle aktuellen Kernelmeldungen.

### **Syntax:**

`dmesg [Optionen]`

### **Optionen:**

- **-T:** Zeigt menschenlesbare Zeitstempel (Jahr-Monat-Tag Stunde:Minute:Sekunde) statt der raw Sekunden seit Boo. (Achtung: Kann ungenau sein, da Kernel keine Zeitzone kennt; -T rechnet anhand aktueller TZ).
- **-l <level>, --level:** Filtert Nachrichten bestimmter Log-Levels. Level: emerg, alert, crit, err, warn, notice, info, debug. Z.B. `dmesg -l err, warn` nur Fehler und Warnungen.
- **-k, --kernel:** Zeigt nur Kernel-Messages (Standard). **-u, --userspace:** Nur Benutzerraum (Logeinträge vom `printk_user`). Kaum genutzt, da meiste aus Kernel.
- **-H, --human:** **Pager-Modus** (lässt scrollen, farbig Highlighting). Interaktiv (Space/PgUp etc.).
- **-w, --follow:** "Tail -f" Modus für dmesg – neue Kernelmeldungen laufend anzeigen.
- **-s N:** Buffer-Größe (Bytes) die gelesen wird (Default könnte begrenzt sein). `dmesg -s 1000000`.

### **Beispiel (gekürzt):**

```
[0.000000] Linux version 5.4.0-80-generic ...
[0.345678] PCIe Bus 0000:00: root hub ...
[1.234567] usb 1-2: new high-speed USB device number 3 ...
[1.345678] ata1: SATA link up ...
[10.567890] Bluetooth: RFCOMM ver 1.11
```

dmesg -T zeigt z.B. [Mon Oct 11 14:23:45 2021] ....

Typisch ruft man dmesg | grep -i error um nach Kernel-Fehlern zu suchen, oder dmesg -w in einem Terminal um z.B. neue USB/Hardware Events sofort zu sehen.

## Dienstverwaltung und Systemstart

### systemctl (Control systemd)

**Beschreibung:** Haupt-Werkzeug zur Steuerung des *systemd*-Initsystems (heutzutage auf vielen Linux-Distributionen). systemctl verwaltet **Services** (Start/Stop/Enable), **Daemon-Konfiguration, System-Zustände** (Reboot/Shutdown), **Unit**-Status und Logs, etc】 .

#### Syntax:

systemctl [Optionen] <Sub-Befehl> [Unit/...]

Units können Services (.service), Mounts (.mount), Timer (.timer), Sockets (.socket), etc. sein. Oft kann man den Suffix weglassen (z.B. "ssh" wird als "ssh.service" erkannt).

#### Wichtige Sub-Befehle (Service Management):

- **start <service>, stop <service>** – Startet oder stoppt einen Dienst \*sofortfile-4fdk7rvx39azjretinarqb】 . Bsp: systemctl start apache2.
- **restart <service>** – Stop + Start (auch wenn nicht lief, wird gestartet).
- **reload <service>** – Lädt Konfiguration neu (sendet i.d.R. SIGHUP). Muss vom Service unterstützt werden (in Unit als ReloadSignal oder ExecReload definiert). Bsp: systemctl reload nginx.
- **status <service>** – Zeigt Status des Dienstes: Läuft er? Letzte Startmeldung, ggf. Auszug aus Journal-Log】 .
- **enable <service>** – Aktiviert den Dienst beim Boot (erstellt Symlinks in passenden WantedBy-Verzeichnissen).
- **disable <service>** – Deaktiviert Autostart.
- **is-enabled, is-active** – Prüfen ob Unit aktiviert ist / gerade aktiv ist (exit 0/1).

- **mask <service>** – "Maskiert" den Service, d.h. verhindert Start vollkommen (setzt Link auf /dev/null). *Unmask* zum Aufheben.
- **daemon-reload** – Falls man eine Unit-Datei geändert oder neu hinzugefügt hat, diese neu einlesen (Konfig neu laden, ohne Dienste neu zu starten).
- **kill <service>** – Sendet ein Signal an alle Prozesse der Service-Unit (z.B. systemctl kill -s SIGUSR1 myapp.service).

### **System Commands:**

- **list-units** – Listet alle geladenen Einheiten (nur aktive standardmäßig). --all um Inaktive auch zu sehen, --type=service filtern auf Services, etc.
- **list-unit-files** – Listet Unit-Dateien und ob sie enabled/disabled/etc. sind (Installationszustand).
- **list-dependencies <unit>** – Zeigt Abhängigkeiten (Requires/Wants) Baum für Unit. --reverse zeigt welche Units diese Unit benötigen.
- **show <unit>** – Gibt alle Properties von Unit aus (oder subset via -p).
- **cat <unit>** – Zeigt den Inhalt der Unit-Datei (und Drop-Ins).
- **edit <unit>** – Öffnet Editor für Drop-In Snippet, um diese Unit zu überschreiben/erweitern.
- **preset <service>** – Setzt enable/disable gemäß Distribution-Vorgabe (in /usr/lib/systemd/system-preset/).

### **Targets (Runlevels):**

- **isolate <target>** – Wechselt in ein target (beendet andere). Z.B. systemctl isolate rescue.target (Einzeluser-Modus ähnlich runlevel 1).
- **set-default <target>** – Legt Default Boot-Target fest (z.B. graphical.target vs multi-user.target).
- **get-default** – zeigt aktuelles Defaulttarget.

### **Power Management:**

- **reboot, poweroff, halt** – führen Neustart bzw. Herunterfahren aus (rufen intern entsprechende target isolation).
- **suspend, hibernate, hybrid-sleep** – starten Energiesparmodus.

### **Journal/Logging Integration:**

- **status** (wie oben) zeigt bereits Log-Auszug.

- **journalctl** (separater Befehl, nicht systemctl subcommand) – zum Betrachten der systemd Logs. Z.B. journalctl -u <service> zeigt Logs zu einem Service, -f zum folgen, --since "2021-10-01" ab bestimmtem Datum etc.

### **Beispiele:**

- systemctl enable --now nginx – Aktiviert **und** startet nginx sofort (Option --now führt gleich start nach enable aus).
- systemctl disable apache2.service – Entfernt Apache2 aus Autostart.
- systemctl restart networking – Dienst neu starten (bei Netz aufpassen).
- systemctl list-units --failed – Zeigt nur fehlgeschlagene Units.
- systemctl mask firewalld; systemctl stop firewalld – Verhindert Start des Firewalld dauerhaft und stoppt ihn sofort (z.B. falls man anderen Firewall nutzt).

### **service (SysVinit Compatibility)**

**Beschreibung:** Der Befehl service war bei SysVinit/Upstart gebräuchlich, um Start/Stop-Skripte in /etc/init.d aufzurufen. Auf systemd-Systemen ruft service intern systemctl auf. Man kann aber service <name> start/stop/status noch benutzen.  
4fdk7rvx39azjretinarqb】 . Moderne Systeme – besser systemctl direkt.

### **Syntax:**

service <ServiceName> [start|stop|restart|status|...]

**Beispiel:** service ssh status ruft systemctl status ssh.service auf.

(Auf SysVinit ohne systemd würde es das init.d Script ausführen.)

### **chkconfig (Runlevel Konfiguration - SysV kompatibel)**

**Beschreibung:** Ein Werkzeug zum Verwalten der Start-/Stopp-Links von SysV-Init-Skripten über verschiedene Runlevels (0-6file-4fdk7rvx39azjretinarqb】 . Auf systemd-Systemen gegenstandslos (meist vorhanden für Kompatibilität, was auf systemctl enable/disable mappt).

### **Wichtigste Befehle:**

- chkconfig --list [name] – Listet alle Dienste und ob sie in runlevel 0-6 on/off sind  
4fdk7rvx39azjretinarqb】 . Mit Name nur diesen.
- chkconfig name on (bzw. off) – Schaltet Service in allen entsprechenden runlevels an oder aus (Standard runlevels 2-5 on für on).
- chkconfig name --level 35 on – Feinsteuerung: aktiviert Dienst in Runlevel 3 und 5.
- chkconfig name reset – Setzt Standard (wie im Init-Skript Header vorgesehen).

\*(Auf systemd delegiert chkconfig intern an systemctl enable/disable. Für Nostalgie, wissen was es tat.)

## Zeitplanung (Cron und At)

### crontab (Cron Table Management)

**Beschreibung:** Planen wiederkehrender Aufgaben. crontab bearbeitet die Aufgabenliste des aktuellen Benutzers (oder eines anderen mit -u) im Cron-Daemon. Jede Zeile in crontab -e hat Format "*Min Stunde Tag Monat Wochentag Befehl*". Der Cron-Dienst prüft pro Minute die Tabellen und führt fällige Kommandos auf.

#### Befehle:

- crontab -e – Ediert die Crontab des aktuellen Benutzers (öffnet \$EDITOR). Beim Speichern wird die neue Tabelle dem Cron-Daemon übergeben.
- crontab -l – Listet die aktuelle Crontab auf.
- crontab -r – Entfernt (löscht) die aktuelle Crontab komplett. (Vorsicht, kein Undo).
- crontab -u username -e/-l/-r – Als root: Bearbeitet/liest/löscht die Crontab eines anderen Benutzers.

#### Cron Format Erinnerung:

# Minute(0-59) Hour(0-23) Day(1-31) Month(1-12) Weekday(0-7) Command

0 2 \* \* \* /usr/bin/backup.sh # täglich 02:00

\*/5 9-17 \* \* 1-5 echo "Working..." # Werktags 9-17 Uhr alle 5 min

\*=jeder Wert, Komma getrennte Listen, Intervalle mit -, Step mit /.

Spezial @-Strings: @reboot, @hourly, @daily, @yearly etc.

**Anacron:** Cron führt Jobs aus, wann Maschine an ist. Für nicht ständig laufende Rechner nutzt man anacron (Tagesjobs nachholen).

### at (One-time scheduled task)

**Beschreibung:** Legt einmalige Ausführungen von Befehlen zu einer bestimmten Zeit in der Zukunft fest. at nimmt eine Zeitangabe entgegen und liest dann von STDIN die Befehle, die ausgeführt werden sollen (oder mit -f aus einer Datei). Der atd Daemon muss laufen.

#### Verwendung:

```
echo "/path/to/script.sh" | at 02:30 # führt Script heute Nacht 2:30 aus
```

Man ruft at <Zeit> auf, landet in einer Stdin-Eingabe (end mit Ctrl+D) oder speist via echo/<<EOF.

**Zeitformat:** Z.B. at 14:00 (heute 14 Uhr, falls schon vorbei dann morgen), at 11pm, at midnight, at now + 5 days, at 08:00 12.12.2025 (8:00 am 12 Dec 2025). Kann englische Wörter: tomorrow, noon, teatime(16:00) etc.

### Befehle:

- atq – Zeigt anstehende at-Jobs (Queue) des aktuellen Userfile-4fdk7rvx39azjretinarqb】 . (Listet Job-IDs und geplante Zeit). Root sieht alle oder atq -q <Queue> für bestimmte Batch-Queues.
- atrm <JobID> – Entfernt einen geplanten Job aus der Queue. (JobID sieht man in atq).
- batch – Alias zu at -q b now (führt Jobs aus, wenn Load Average niedrig ist). "Batch" schiebt den Auftrag in Batch-Queue, läuft sobald System nicht ausgelastet (LA<0.8 default).

### Beispiel:

```
$ at now + 1 minute
```

```
warning: commands will be executed using /bin/sh
```

```
at> echo "Hello after 1 min" >> /tmp/test.txt
```

```
at> <EOT>
```

```
job 5 at Fri Oct 15 16:41:00 2021
```

Dann innerhalb einer Minute atq zeigt e.g. 5 Fri Oct 15 16:41:00 2021 a user. Nach Ausführung landet Text in Datei.

(Zugriff: /etc/at.allow /etc/at.deny definieren wer at nutzen darf – analog cron.allow/deny.)

### Drucken

#### **lpr (Line Print Request)**

**Beschreibung:** Sendet eine Datei (oder STDIN) an den Drucker (Print-Job einreihen). lpr ist Teil des BSD-Linux-Drucksystems (CUPS stellt Kompatibilität bereit). Alternative: lp (System V style).

### Syntax:

```
lpr [Optionen] [Dateien...]
```

### Wichtige Optionen:

- **-P <Druckernname>**: Wählt einen bestimmten Drucker aus (sonst Standarddruckerfile-4fdk7rvx39azjretinarqb】 . Druckernamen findet man via lpstat -a oder CUPS Webinterface.

- **-# <Num>**: Anzahl Kopien drucken (z.B. -#3 dreifach).
- **-h**: Unterdrückt Banner-Seite (header page) falls aktiviert.
- **-r**: Löscht die Datei(en) nach dem Versenden an den Spool (rarely used).
- **-m**: Sendet E-Mail Benachrichtigung nach Druck (sofern eingerichtet).
- **-o <Option>=<Wert>**: Drucker-spezifische Optionen übergeben (z.B. -o landscape für Querformat, oder -o media=A4 oder -o sides=two-sided-long-edge für Duplex). Diese Optionen sind je nach Drucker Treiber definierbar (CUPS options).
- **-C <Klasse>, -J <Jobname>, -T <Titel>**: Legen Klassenname, Jobname, Titel für Banner etc. fest.
- **--help** (GNU lpr) usw.

### **Beispiele:**

- lpr report.pdf – Schickt *report.pdf* an Standarddrucker.
- lpr -P HP\_LaserJet -#2 invoice.txt – Druckt *invoice.txt* 2-fach auf Drucker "HP\_LaserJet".
- echo "Test Page" | lpr – Druckt STDIN (den Text "Test Page").
- lpr -o sides=two-sided-long-edge thesis.pdf – Duplexdruck beidseitig.

(Verwandt: *lpq* zeigt Druckerqueue, *lprm* löscht Jobs aus der Queue.)

### **Lokale Systemkonfiguration**

#### **date (Anzeige/Setzen Datum/Uhrzeit)**

**Beschreibung:** Zeigt das aktuelle Datum und die Uhrzeit an, oder stellt sie ein. Auch Formatierung in beliebigem Muster ist möglich.

#### **Syntax:**

date [Optionen] [+Format]

date -s "STRING" (setzen)

#### **Optionen / Verwendung:**

- **Ausgabe formatieren:** date "+FORMAT" – Mit Format-Pattern ähnlich strftime:  
%Y (Jahr), %m Monat(01), %b Monat (Kurzname), %d Tag, %H Stunde (24h), %I (12h), %M Minute, %S Sek, %A Wochentag, %a (kurz), %Z Zeitzone, %z offset, %T = %H:%M:%S, %F = %Y-%m-%d, etc. Z.B. date "+%d.%m.%Y %H:%M:%S" -> "05.10.2025 14:30:00".

- **Systemzeit stellen:** date -s "2025-10-05 14:30:00" – Setzt Datum. Braucht Root. (Alternative: date --set="...").
- **Hardware-Uhr stellen:** In Linux erfolgt date auf Systemzeit (in Kernel, RTC separiert). Um auch BIOS/RTC Uhr zu setzen, hwclock --systohc (setze Hardware Clock nach System Clock). Umgekehrt hwclock --hctosys.
- **Universale Zeit:** date -u zeigt in UTC. date -u -s "2025-10-05 12:30:00" setzt in UTC.
- **RFC-3339 Format:** date --rfc-3339=ns etc. Standard ISO.
- **Unix Timestamp:** date +%s gibt Sekunden seit 1.1.1970 (UTC).

#### **Beispiele:**

- date -> "Tue Oct 5 14:30:00 CEST 2025".
- date +"%Y-%m-%d\_%H-%M-%S" -> "2025-10-05\_14-30-00".
- sudo date -s "@\$(( \$(date +%s) + 3600 ))" -> Systemzeit um eine Stunde vorstellen (Beispiel mit timestamp math).

#### **timedatectl (Zeiteinstellung via systemd)**

**Beschreibung:** Befehl von systemd zum Anzeigen und Ändern der Systemzeit, Hardware-Uhr, Zeitzone und NTP-Synchronisatiofile-4fdk7rvx39azjretinarqb】 .

#### **Verwendung:**

- **timedatectl** (ohne args): Zeigt aktuellen Status: Lokale Zeit, UTC, RTC, Zeitzone, ob NTP aktiv ist etcfile-4fdk7rvx39azjretinarqb】 .
- **timedatectl set-time "YYYY-MM-DD HH:MM:SS"** – Setzt Systemzeit (wenn kein NTP aktiv).
- **timedatectl set-timezone <Zone>** – Ändert Zeitzone. Z.B. timedatectl set-timezone Europe/Berlin. (Verfügbare Zonen unter /usr/share/zoneinfo/, oder timedatectl list-timezones).
- **timedatectl set-ntp true/false** – Aktiviert bzw. deaktiviert die automatische Zeitsynchronisation (via systemd-timesyncd oder anderes NTP). Bei true startet es den Dienst falls vorhanden.
- **timedatectl set-local-rtc 1/0** – Legt fest, ob die Hardware-Uhr (RTC) in **lokaler Zeit** läuft (1) oder in UTC (0, Standard). Windows dual-boot oft RTC=lokal (dann 1), sonst belässt auf 0=UTC um Probleme zu vermeiden.

#### **Beispiele:**

\$ timedatectl

Local time: Tue 2025-10-05 14:30:00 CEST

Universal time: Tue 2025-10-05 12:30:00 UTC

RTC time: Tue 2025-10-05 12:30:00

Time zone: Europe/Berlin (CEST, +0200)

System clock synchronized: yes

NTP service: active

RTC in local TZ: no

timedatectl set-time "2025-12-24 18:00:00" stellt Heiligabend 18:00. timedatectl set-timezone UTC wechselt Zeitzone auf UTC.

## **localectl (Locale and Keyboard Layout)**

**Beschreibung:** Verwaltet System-Locale (Sprache, Gebietsschema) und Tastaturlayout-Einstellungen in systemd-basierten Systemefile-4fdk7rvx39azjretinarqb】. Es kann die Variablen in /etc/locale.conf sowie Konsolen-Tastaturlayout in /etc/vconsole.conf setzen. (Für X/Wayland Tastatur greift localectl in /etc/X11/xorg.conf.d/ oder mittels setxkbmap separat).

### **Verwendung:**

- **localectl** (ohne args): Zeigt aktuelle Locale(s) und Keymap. Z.B.:  
System Locale: LANG=de\_DE.UTF-8  
VC Keymap: de  
X11 Layout: de  
X11 Model: pc105 etc.
- **localectl list-locales** – Listet alle verfügbaren Locale, die generiert sind (die in /usr/lib/locale oder /etc/locale.gen markiert).
- **localectl set-locale LANG=<locale> [VAR=locale...]** – Setzt System-Locale. Z.B. localectl set-locale LANG=en\_US.UTF-8. Optional mehrere, z.B. LC\_TIME=en\_GB.UTF-8. Wirksam nach Neustart oder neu login shells.
- **localectl list-keymaps** – Listet verfügbare Console-Keymaps (z.B. us, de, ...).
- **localectl set-keymap <map> [<toggle>]** – Setzt Tastaturlayout für Konsole (vconsole) und ggf. X11, falls X11 Layout nicht getrennt definiert. Z.B. localectl set-keymap de.
- **localectl set-x11-keymap <layout> [model] [variant] [option]** – Spezifisch X11 (Graphical) Layout/Modell setzen. Z.B. localectl set-x11-keymap us pc104 "" terminate:ctrl\_alt\_bksp.

*Beispiel:*

localectl set-locale LANG=fr\_FR.UTF-8 (Systemssprache auf Französisch).

localectl set-keymap us (Konsole auf US-Layout).

localectl set-x11-keymap de pc105 nodeadkeys (Grafische Tastatur auf Deutsch mit Variante "nodeadkeys").

(Die Konfigurationen landen in /etc/locale.conf, /etc/vconsole.conf, /etc/X11/xorg.conf.d/00-keyboard.conf entsprechend.)

## Netzwerk-Konfiguration

### ip (Netlink Interface Config)

**Beschreibung:** Moderner Befehl zur Verwaltung von Netzwerkinterfaces, IP-Adressen, Routing, Tunnels etc. (Nachfolger von ifconfig/route). ip hat Unterkommandos für verschiedene Objekte: **link** (Netzwerkinterface), **addr** (IP-Adressen), **route**, **neigh** (ARP/NDP Nachbarn), **rule** (Policy Routing), **maddr** (Multicast), **addrlabel**, **netns** (Net Namespaces), **link set** (Interface-Settings).

#### Syntax (Grundstruktur):

ip [Options] <Object> <Subcommand> [Arguments...]

z.B. ip link show, ip addr add, ip route list, etc.

#### Häufige Befehle/Beispiele:

- **Interface anzeigen:** ip link show [dev <iface>] – Listet alle Netzwerkinterfaces (oder ein spezifisches). Ausgabe enthält Index, Name, MAC, MTU, Flags (UP,LOWER\_UP=Link OK, MULTICAST), etc.
- **Interface aktivieren/deaktivieren:** ip link set <iface> up / down – entspricht `ifconfig iface up/downfile-4fdk7rvx39azjretinarqb】 .
- **Adresse zuweisen:** ip addr add <IP>/<Prefix> [broadcast +] dev <iface> – z.B. ip addr add 192.168.1.100/24 dev eth0. broadcast + lässt broadcast aus Prefix berechnen (optional).
- **Adresse löschen:** ip addr del <IP>/<Prefix> dev <iface>.
- **Adressen anzeigen:** ip addr show [dev <iface>] – Listet IPv4/IPv6 Adressen auf Interfacefile-4fdk7rvx39azjretinarqb】 . Zeigt auch Scope (global/link), flags (tentative, dynamic).
- **Standardroute setzen:** ip route add default via <Gateway-IP> [dev <iface>] – z.B. ip r add default via 192.168.1.1 dev eth0.
- **Statische Route setzen:** ip route add <Netz>/<Präfix> via <nächsterHop> dev <iface>.

- **Routen auflisten:** ip route show (oder ip r).
- **ARP Cache:** ip neigh show – zeigt ARP/ND-Cache (Neighbors).
- **ARP Eintrag fix setzen:** ip neigh add <IP> lladdr <MAC> dev <iface> nud permanent.
- **Secondary IPs / Aliase:** Einfach mehrfach ip addr add auf selbes dev ausführen (oder use Label).
- **MTU ändern:** ip link set dev <iface> mtu 1400.
- **MAC ändern:** ip link set dev <iface> address <NewMAC> (Down interface first).
- **Promisc Mode:** ip link set dev <iface> promisc on.
- **Routing Policy (advanced):** ip rule und ip route add table <n> for source-based routing etc.
- **Network Namespace mgmt:** ip netns (add, del, exec, etc – advanced).

**Options global:** -s (stats), -d (details), -br (brief output), -c (color output). Z.B. ip -br addr zeigt Interfaces in einzeiligem Kurzformat.

### Beispiele:

- ip -br link ->

```
lo UNKNOWN 127.0.0.1/8 ::1/128
eth0 UP 192.168.1.10/24 fe80::.../64
```

- ip addr add 10.0.0.1/16 dev eth1 (fügt IP).
- ip route show (z.B. default via 192.168.1.1 dev eth0 proto dhcp metric 100 etc.).
- ip neigh show (z.B. 192.168.1.1 dev eth0 lladdr aa:bb:cc:dd:ee:ff REACHABLE).

(ifconfig (net-tools) alternative: ip ist ausführlicher und bevorzugt.)

### ifconfig (Interface Config – alt)

**Beschreibung:** Älteres Werkzeug aus net-tools (weitgehend ersetzt durch ip). ifconfig kann Interfaces anzeigen, IP setzen, MTU ändern, promiscuous Mode etc., aber unterstützt z.B. keine IPv6 out-of-the-box je nach Version.

### Syntax:

ifconfig [Interface] [Optionen]

### Beispiele:

- ifconfig – zeigt alle aktiven (UP) Interfaces mit IPv4 etc.

- ifconfig -a – zeigt alle, inkl. down Interfaces.
- ifconfig eth0 192.168.0.5 netmask 255.255.255.0 broadcast 192.168.0.255 – setzt IP, Maske, Bcast.
- ifconfig eth0 up / down – Interface an/aus.
- ifconfig eth0 mtu 1400 – MTU ändern.
- ifconfig eth0 promisc – Promiscuous Mode an. -promisc aus.
- ifconfig eth0 hw ether 00:11:22:33:44:55 – MAC-Adresse setzen.

*(ifconfig ist nicht mehr standard installiert in neueren Distros; ip commands stattdessen nutzen.)*

### **iwconfig (Wireless Config – alt)**

**Beschreibung:** Ähnlich ifconfig, aber für Wireless LAN Schnittstellen (Legacy tool aus wireless-tools). Zeigt/ändert ESSID, Mode, Frequenz, Schlüssel etc. Heutzutage ersetzt durch iw (nl80211), oder NetworkManager.

#### **Beispiele (falls vorhanden):**

- iwconfig wlan0 – zeigt ESSID, Mode (Managed/Ad-Hoc/Master), Channel/Freq, Bitrate, Tx-Power, Signal Level, Encryption key on/off, etc.
- iwconfig wlan0 essid "WLAN-Name" – mit offener SSID verbinden (sofern AP ohne Sicherheit).
- iwconfig wlan0 key s:password – setzt WEP Key (veraltet).
- iwconfig wlan0 mode Monitor – Interface in Monitor Mode (für Sniffing).

*(Iwconfig kann kein WPA2; da nutzt man wpa\_supplicant oder NM, etc.)*

### **nmcli (NetworkManager CLI)**

**Beschreibung:** Kommandozeilen-Interface zum NetworkManager-Dienst, um Netzwerkkonfiguration (insb. Desktop-Umfeld) zu betrachten und steuerfile-4fdk7rvx39azjretinarqb】 . Kann Verbindungen (profiles) verwalten, aktivieren, WLAN scannen, etc.

#### **Verwendung (Struktur):**

nmcli [general|networking|radio|connection|device] < Aktion > [Parameter]

#### **Beispiele:**

- **Status anzeigen:** nmcli general status – Gesamter NM-Status. nmcli connection show – listet gespeicherte Verbindungen (Profiles). nmcli device status – zeigt Geräte und ob connected und welches Profil.

- **WLAN scannen:** nmcli device wifi list – listet verfügbare WLANs.
- **Verbinden WLAN:** nmcli device wifi connect "SSID" password "passwort" – erstellt ggf. Connection-Profil und verbindet.
- **Verbindung aktivieren/deaktivieren:** nmcli connection up <Name> / down <Name>. Oder nmcli device disconnect <iface> (z.B. nmcli device disconnect wlan0).
- **Neue Verbindung anlegen:** nmcli connection add type ethernet ifname eth0 con-name "Wired Manual" ip4 192.168.5.5/24 gw4 192.168.5.1.
- **IPv4 ändern laufend:** nmcli connection modify "Wired connection 1" ipv4.addresses 192.168.1.50/24 ipv4.gateway 192.168.1.1 ipv4.method manual und dann up.
- **Ethernet an/aus:** nmcli networking off (setzt NM generell offline) oder nmcli radio wifi off (schaltet nur WLAN-Funk aus).

#### **Beobachten:**

- nmcli connection show --active – nur aktive Conns.
- nmcli -p device status – pretty (ncurses-like) output.

#### **ethtool (NIC Driver Tool)**

**Beschreibung:** Zeigt und ändert Einstellungen von Ethernet-Netzwerkkarten (Treiber-Level): z.B. Geschwindigkeit, Duplex, Auto-Negotiation, Wake-on-LAN, Offload-Funktionen, Statistikzählefile-4fdk7rvx39azjretinarqb】.

#### **Verwendung:**

- ethtool <iface> – Zeigt Standard-Eigenschaften: unterstützt Geschwindigkeiten (Supported link modes), aktuelle Speed/Duplex, Auto-negotiation on/off, Ports (TP, etc.), driver name, firmware-version, bus-info, etc.
- ethtool -s <iface> speed <Mb/s> duplex <half/full> autoneg <on/off> – Setzt feste Geschwindigkeit/Duplex. (z.B. ethtool -s eth0 speed 100 duplex full autoneg off). Vorsicht: muss vom Link Partner unterstützt werden, sonst Link down.
- ethtool -k <iface> – Zeigt Kernel-Offload-Funktionen (tx-checksumming, scatter-gather, TSO etc.) und ob an/aus. -K ... (capital K) schaltet diese: z.B. ethtool -K eth0 gro off (GRO ausschalten).
- ethtool -i <iface> – Infos zum Treiber (Driver name, version, firmware, bus).
- ethtool -p <iface> [N] – "Physically identify": Blinkt die LED am Port für N Sekunden (Standard 15), falls Karte das unterstützt. Hilft um Kabelport zu finden.

- ethtool -S <iface> – Ausgabe von Statistikzählern (packets, errors, etc. vom Treiber).
- ethtool -g <iface> – Ring Puffer Größen (RX/TX rings). -G zum Setzen.
- ethtool -a <iface> – Flow-Control (Pause Frame) Status (Autoneg etc.), -A zum Setzen on/off rx/tx.
- ethtool -r <iface> – Veranlasst die Karte, Link neu auszuhandeln (Retrain).
- ethtool --identify <iface> – alias für -p (Blinken).

### **Beispiele:**

- ethtool eth0 -> zeigt z.B. Speed: 1000Mb/s, Duplex: Full, Auto-negotiation: on, etc.
- ethtool -s eth0 speed 100 duplex half autoneg off -> zwingt 100Mbit Half-Duplex (nur Test oder Legacy).
- ethtool -p eth1 10 -> blinkt LED an eth1 für 10 Sekunden.
- ethtool -K eth0 tso off -> schaltet TCP Segment Offloading aus (im Troubleshooting falls Problem mit Offload).

### **hostnamectl (Control hostname in systemd)**

**Beschreibung:** Zeigt oder ändert den System-Hostname und verwandte Settings (wie Chassis Typ, Icon). hostnamectl ist Teil von systemd.

### **Verwendung:**

- **hostnamectl** (ohne args): Zeigt Systeminfo: Static hostname, Transient (via DHCP), Pretty (human name), und Kernel: Operating System, Kernel, Architecture, etc.
- **hostnamectl set-hostname <name> [--pretty|--static|--transient]** – Setzt den Hostnamen. Ohne Flag wird *static* und *transient* gleichermaßen gesetzt. *static* = in /etc/hostname gespeichert (persistenter Name), *transient* = Laufzeit (kann z.B. DHCP ändern). *pretty* = nur für Anzeigezwecke (z.B. mit Großbuchstaben/Leerzeichen erlaubt).  
Beispiel: hostnamectl set-hostname server42 – setzt static+transient auf "server42".  
hostnamectl set-hostname "My Home PC" --pretty – setzt Pretty-Name.
- **hostnamectl set-icon-name <icon>** – Setzt Icon (z.B. computer-desktop, rein kosmetisch in einigen GUIs).

- **hostnamectl set-chassis <type>** – Setzt Gehäuse-Typ: desktop, laptop, server, tablet, embedded, etc. (Auch nur informative, kann z.B. in GNOME Systeminfo auftauchen).

#### **Zusatz:**

- hostnamectl status (alias ohne args) – Info.
- Traditional hostname Befehl (ohne ctl) zeigt oder setzt nur transient (bis Reboot).

### **Netzwerk-Fehlerdiagnose und Name-Auflösung**

#### **netcat (nc) (TCP/UDP Swiss Army Knife)**

**Beschreibung:** netcat (oder nc) ist ein universelles Netzwerk-Tool, um Verbindungen zu testen, Daten zu senden/empfangen, Ports zu lauschen etcfile-4fdk7rvx39azjretinarqb】. Kann sowohl Client als auch rudimentärer Server sein.

#### **Verwendung:**

- **Verbindung als Client:** nc [options] <Host> <Port> – Baut TCP-Verbindung zu Host:Port auf, leitet stdin -> Socket, und Socket -> stdout. So kann man z.B. echo "HEAD / HTTP/1.0" | nc example.com 80 (HTTP-Request senden). Oder nc time.nist.gov 13 (Daytime Protocol). Wenn man interaktiv ausführt, kann man manuell tippen (Telnet-Ersatz).
- **Port Listening (Server):** nc -l -p <Port> [Host] – Horcht auf Port (optional an gegebene IP gebunden). Bsp: nc -l -p 1234 lauscht auf TCP/1234 auf alle Interfaces; was reinkommt wird auf stdout ausgegeben, was man tippt geht an Client.
- **UDP mode:** nc -u – Statt TCP benutzt UDP. Bsp: nc -u 192.168.1.5 53 könnte man UDP Query schicken (z.B. an DNS server).
- **Port Scan:** nc -z -v host startport-endport – "Zero-I/O mode" nur scannen, nicht verbinden. Z.B. nc -z -v 192.168.1.10 1-1024 zeigt offene Ports. (Netcat als Portscanner ist möglich, aber Nmap dafür besser).
- **Timeout:** -w <secs> – Wartezeit bis Timeout für Verbindungen/Eingaben.
- **Keep-open (inetd style):** In GNU nc -k (listen bleibt offen nach Verbindungsende, auf neue).
- **Exec:** -e <prog> – *Unsichere Option* (nicht immer kompiliert, oft disabled aus Sicherheitsgründen): Leitet die I/O eines Programms an den Socket. So kann man z.B. remote shell spawn: nc -l -p 5000 -e /bin/bash. (Equivalent zum berüchtigten rm -rf example). Modern: use socat or ssh for shells.

#### **Beispiele:**

- **Chat/Echo Test:** Terminal A: nc -l -p 5000, Terminal B: nc localhost 5000 -> alles was A tippt sieht B und umgekehrt.
- **Datei senden:** Auf Empfänger: nc -l -p 4500 > out.txt, auf Sender: nc destIP 4500 < file.txt. (Netcat baut TCP auf und streamt Datei).
- **HTTP Request:** printf "GET / HTTP/1.1\r\nHost: example.com\r\n\r\n" | nc example.com 80 – zeigt HTML der Homepage (HTTP/1.1 erfordert Host-Header).
- **UDP Ping** (ohne Antwort): echo -n "Hello" | nc -u 192.168.1.5 9000.
- **Port scan:** nc -z -v 192.168.1.20 20-30 -> tries FTP-data, FTP etc, output open/closed.

*(Netcat Implementierungen: Traditional vs OpenBSD netcat vs GNU netcat ncat (part of nmap) – Optionen leicht variierend. Z.B. OpenBSD netcat hat -l und -p zusammen als one option or so, etc.)*

### **iftop (Interface TOP, Bandwidth monitor)**

**Beschreibung:** Ein curses-basiertes Tool zum Anzeigen der aktuellen Bandbreitennutzung pro Verbindung (ähnlich top, aber für Netzwerk Trafficfile-4fdk7rvx39azjretinarqb】). Es zeigt welche Peers mit dem System kommunizieren und wieviel Bytes/s in/out. Hilfreich zum realtime Debugging wer Bandbreite frisst.

#### **Verwendung:**

iftop -i <Interface>

Ohne -i wählt iftop das erste externe Interface.

#### **Optionen:**

- **-n** – Keine DNS-Auflösung (zeigt IPs statt aufzulösen, wichtig um Ausgabe schnell zu haltenfile-4fdk7rvx39azjretinarqb】).
- **-N** – Keine Port-Auflösung (zeigt Portnummern statt Services Namen).
- **-P** – Zeigt Ports zusammen mit Host (separat Spalte). Kombiniert man oft: iftop -nP.
- **-B** – Bytes statt Bits in Anzeige (Standard iftop zeigt in bits/s).
- **-F <FilterNet>** – Nur Traffic, der in bestimmten Netzbereich fällt (Filter) betrachten, z.B. -F 192.168.0.0/16 filtert auf diese Range.
- **-p** – Promiscuous Mode (auch Traffic anzeigen, der nicht an lokalem Host endet? Normal iftop Filter host involvement).
- **-t** – Text-Only Modus (keine curses-Interface, nur in stdout Stats; nicht sehr üblich).

Im Interface kann man mit Tasten: h Hilfe, n togg DNS, s sort by source, d sort by dest, t sort by throughput, p toggle port display, b toggle bar graph, B cycles through rate intervals (default 2s,10s,40s windows).

### **traceroute / tracepath (Route Tracking)**

**Beschreibung:** Sendet eine Serie von Paketen mit steigenden TTL (Time-to-Live) Werten, um den Pfad (Router Hopps) zum Ziel auszumessefile-4fdk7rvx39azjretinarqb】.

Traceroute verwendet standard UDP oder ICMP ECHO, tracepath nutzt UDP with high port (no root needed). Jedes Hop sendet "Time Exceeded" zurück, das traceroute protokolliert. Ergebnis: Liste der Router vom Quell- zum Zielnetz.

#### **Syntax:**

```
traceroute [Optionen] <Host> [PacketSize]
```

```
tracepath <Host>
```

#### **Optionen (traceroute):**

- **-I:** ICMP Echo statt UDP verwenden (pakete wie ping).
- **-T:** TCP SYN statt UDP (z.B. um Firewalls zu durchdringen über port 80).
- **-p <Port>:** UDP port Start (Standard 33434).
- **-q <N>:** Anzahl Probes pro TTL (Default 3).
- **-w <Seconds>:** Timeout pro Hop Antwort (default 5s).
- **-n:** Nicht auflösen (IP statt DNSfile-4fdk7rvx39azjretinarqb】).
- **-m <max\_ttl>:** Max Hop Count (Standard 30).
- **-f <first\_ttl>:** Start TTL (default 1). Manchmal hilfreich, mittendrin einzusteigen.
- **-4 / -6:** Force IPv4 / IPv6.

**Optionen (tracepath)** sind wenige: -n no dns, -b show both hostnames and IP, etc.

#### **Beispiel:**

```
$ traceroute example.com
```

```
1 192.168.1.1 (192.168.1.1) 1.123 ms 0.846 ms 0.799 ms
2 100.64.0.1 (100.64.0.1) 10.123 ms 9.876 ms 9.543 ms
3 203.0.113.5 (203.0.113.5) 15.432 ms 14.876 ms 15.210 ms
4 ae12.dar02.example.net (198.51.100.1) 30.456 ms 29.999 ms 30.123 ms
5 93.184.216.34 (93.184.216.34) 35.678 ms 35.900 ms 35.456 ms
```

Zeigt 5 Hops (Heimrouter, CGNAT router, ISP core, example.net router, Ziel).

tracepath example.com ähnlich, oft mit asymmetrischen result such as pmtu info.

### **mtr (My Traceroute)**

**Beschreibung:** Kombiniert traceroute und ping in einem kontinuierlichen Diagnose-Tool. mtr sendet fortlaufend Pakete über alle Hops und aktualisiert in Echtzeit die Latenz und Packetloss-Statistiken pro Hop. Hilfreich um instabile Verbindungen/Packet Loss auf bestimmten Strecken zu identifizieren.

#### **Verwendung:**

mtr [Optionen] <Ziel>

Interaktiv curses UI standard, oder mit -r (report) Modus für einmalige Ausgabe.

#### **Optionen:**

- **-4 / -6:** Force IPv4/IPv6.
- **-n:** Nicht auflösen, nur IPs (schneller).
- **-w:** Wide-output (zeigt beide host+IP columns).
- **-c <count>:** Limit auf <count> Probezyklen, dann beenden (für Report).
- **-r:** Report-Modus (nicht curses, druckt Tabelle und endet).
- **-i <sec>:** Sekunden zwischen Probe-Paketen (default 1.0).
- **-p:** Pausiert Ausführung beim Start (Press enter to start).
- **-t:** Text Mode (like curses but in dumb terminal).
- **-u:** UDP mode (Standard mtr uses ICMP Echo, as -l in traceroute terms). -T for TCP mode also available.

#### **Anzeige:**

MTR listet Hops fortlaufend wie traceroute, plus Spalten: Loss%, Snt (packets sent), Last (ms last latency), Avg, Best, Wrst (worst), StDev. Es aktualisiert kontinuierlich.

Beispiel-Zeile:

3. 203.0.113.5 0.0% 10 15.4 16.1 14.8 20.3 1.3

### **ping (ICMP Echo)**

**Beschreibung:** Schickt ICMP Echo-Request-Pakete an eine Adresse und misst die Antwortzeiten (Echo-Reply), um Erreichbarkeit und Round-Trip-Time festzustellen.

#### **Syntax:**

ping

\*\*(Fortsetzung)\*\*

### ping (ICMP Echo Requests)

\*\*Beschreibung:\*\* `ping` prüft die Erreichbarkeit eines Hosts im Netzwerk mit \*\*ICMP-Echo-Paketen\*\* und misst die Round-Trip-Zeit (RTT) der Antwort. Es sendet fortlaufend Anfragen (standardmäßig, bis abgebrochen) und berechnet Statistik über Verlust und Latenz.

\*\*Syntax:\*\*

``` bash

ping [Optionen] <Ziel>

Hier kann <Ziel> ein Hostname oder eine IP-Adresse sein.

Wichtige Optionen:

- **-c <Anzahl>** – Anzahl der ICMP-Pakete festlegen. Z.B. ping -c 4 8.8.8.8 sendet genau 4 Pakete.
- **-i <Sekunden>** – Interval zwischen den Pings (Standard 1 Sekunde). Man kann z.B. -i 0.2 für fünf Pings pro Sekunde (root-Recht nötig bei <0.2).
- **-W <Timeout>** – Wartezeit in Sekunden auf eine Antwort, bevor als **timeout** gezählt (Default meist 1 Sekunde, bei -c am Ende kann man höher setzen).
- **-q** – "Quiet": keine per-Ping Ausgabe, nur eine Zusammenfassung am Ende.
- **-p <Pattern>** – Sende individuelles Byte-Muster im Paket (Hexadezimal angegeben). Gut für bestimmte Diagnosen (z.B. bestimmte Bits toggeln sehen).
- **-s <Packetsize>** – Größe des zu sendenden Pakets in Bytes (ohne ICMP-Header). Standard 56 Bytes (plus 8 Byte ICMP-Header = 64 Bytes in Ausgabe).
- **-t <TTL>** – Setzt die Time-To-Live (IPv4) oder Hop Limit (IPv6) für die gesendeten Pakete. (Nützlich, um gezielt nur bis zu einem gewissen Hop zu gelangen, ähnlich traceroute Step).
- **-I <Interface>** – Bindet an ein bestimmtes Quell-Interface (oder IP) für Multi-homed hosts. Alternativ **-S <Source_Address>**.
- **-4, -6** – Erzwingt IPv4 oder IPv6.
- **-f** – "Flood ping": schickt so schnell wie möglich Pings (sehr viele pro Sekunde) und druckt einen Punkt pro empfangenes Echo. (Nur root, zur Stresstest oder schnelle Packetloss Erkennung).

- **-L** (bei einigen ping-Versionen) – Für Echo an Broadcast/Multicast-Adresse (ermöglicht Broadcast-ping, was standardmäßig blockiert ist).
- **-a** – Bei Empfang einen Terminal-Bell ausgeben (akustisches Signal).
- **-D** – Zeitstempel vor jeder Zeile (UNIX time).
- **-U** – Zeitstempel in Us seit Boot vor jeder Zeile (für Messungen).

Ausgabe-Interpretation:

Jede empfangene Antwort erscheint als Zeile:

64 bytes from 8.8.8.8: icmp_seq=1 ttl=117 time=8.34 ms

Bedeutet: Echo-Reply von 8.8.8.8 erhalten, Größe 64 Bytes, Sequenznummer 1, TTL im Reply 117, RTT 8.34 ms. Wenn Antworten ausbleiben, erscheinen Zeilen wie "Request timeout".

Am Ende kommt eine **Statistik**:

4 packets transmitted, 4 received, 0% packet loss, time 3005ms

rtt min/avg/max/mdev = 8.334/8.512/8.790/0.177 ms

Zeigt Verlust und RTT-Werte.

Beispiele:

- ping 1.1.1.1 – Kontinuierlicher Test zu 1.1.1.1 (Cloudflare DNS). Abbruch mit Strg+C.
- ping -c 5 www.example.com – 5 Pings an example.com, dann stop mit Ausgabe der Statistik.
- ping -l eth1 192.168.2.1 – Ping von Interface eth1 aus (wenn Multi-Interface).
- ping -s 1000 -c 3 host – Sende 3 Pings mit Paketgröße ~1000 Bytes (ICMP Payload).
- ping -6 -c 4 ipv6.google.com – IPv6 Ping zu Google.

traceroute (siehe oben)

(bereits behandelt in vorangehendem Teil, siehe „traceroute“ in Netzwerkdagnostik.)

tracepath (siehe oben)

(siehe „tracepath“ bei Netzwerkdagnostik.)

resolvectl (Systemd-resolved Query)

Beschreibung: CLI-Tool zum Abfragen des systemd-resolved-Dienstes (Namensauflösung). resolvectl kann DNS-Lookups durchführen, DNS-Settings anzeigen, LL/MNR Browsing etc. Ersetzt Tools wie systemd-resolve (veraltet).

Verwendung:

- **resolvectl status** – Zeigt die aktuellen DNS-Einstellungen pro Link (Interface): DNS-Server, Suche-Domains, LLMNR/MulticastDNS Status, DNSSEC, usw.
- **resolvectl query <Hostname>** – Löst einen Hostnamen auf (A/AAAA-Record). Ausgabe enthält v6 und v4 Adressen, CNAME falls, DNSSEC info, Cache info, antwortender Server etc.
Beispiel: resolvectl query example.com.
- **resolvectl query -t MX <Domain>** – DNS-Abfrage bestimmten Typs (hier MX-Records). Unterstützt -t ANY, A, AAAA, MX, SRV, SOA etc.
- **resolvectl query <IP>** – Reverse Lookup (IP zu Name via PTR).
- **resolvectl flush-caches** – Löscht den DNS-Cache von systemd-resolved.
- **resolvectl dns <Interface> [Server...]** – Setzt DNS-Server für ein bestimmtes Interface (über DBus). resolvectl domain <iface> ... analog. (Normal wird das via NetworkManager/Netplan/etc gemacht).
- **resolvectl revert <Interface>** – Löscht manuelle Overrides und kehrt zu vom Netzwerkdienst gelieferten DNS zurück.

Beispiel:

```
$ resolvectl query -t MX example.com
example.com IN MX 0 .
-- Information acquired via protocol DNS in 15.1ms.
```

-- Data is authenticated: no

Bedeutet example.com hat keinen MX (Mail).

resolvectl status Ausschnitt:

Global

Protocols: LLNR=resolve mDNS=no DNSOverTLS=no DNSSEC=no/unsupported

resolv.conf mode: stub

Link 2 (eth0)

Current Scopes: DNS

Protocols: +DefaultRoute +LLNR -mDNS DNSOverTLS=opportunistic DNSSEC=...

Current DNS Server: 192.168.1.1

DNS Servers: 192.168.1.1 1.1.1.1

DNS Domain: ~.

Zeigt z.B. that eth0 hat default DNS 192.168.1.1 plus Cloudflare, und ~. (default domain).

dig (DNS Lookup Utility)

Beschreibung: dig (domain information groper) führt DNS-Abfragen aus. Es ist Teil von BIND-Tools (bind-utils). Im Gegensatz zu host oder nslookup bietet dig ausführlich formatierbare Antworten. Standard macht eine A-Record-Abfrage und gibt Antwort mit Abschnitt **ANSWER SECTION** aus.

Syntax:

dig [@DNS-Server] <Name> [<Typ>] [<Klasse>] [Optionen]

<Klasse> normalerweise IN (Internet), selten was anderes.

Wichtige Optionen:

- **@server** – Spezifischen DNS-Server benutzen statt System-resolver. Z.B. dig @8.8.8.8 example.com.
- **<Typ>** – Abfragetyp: A, AAAA, MX, TXT, SRV, NS, SOA, CNAME, PTR, ANY etc. (Default A).
- **+short** – Ausgabe nur die Antworten knapp (einfacher, scripterfreundlic0】 . Z.B. dig +short example.com A -> "93.184.216.34".
- **+noall +answer** – Unterdrückt alle außer Answer-Section (ähnlich short aber mit TTL und Info).
- **+trace** – Iterativer Trace durch DNS-Hierarchie (Root bis Ziel, ähnlich dig an Root und dann NS referrals folgen). Zeigt jeden Schritt. Hilfreich zum Debug.
- **+tcp** – Erzwinge TCP statt UDP (z.B. bei großen DNS-Records oder EDNS issues).
- **+nssearch** – Findet alle NS des Ziel und fragt diese nach SOA (um Delegation correctness zu checken).
- **-x <IP>** – Reverse lookup (PTR) für IP. (Bequem: dig -x 8.8.8.8).
- **+multi** – Schöne Ausgabe in Spalte falls mehrere TXT oder so.
- **+nodnssec oder +dnssec** – Anfordern oder nicht (um RRSIG etc. zu sehen).
- **+ttlshort** – Output TTL in short mode. (e.g. dig +short +ttl google.com -> "300 172.217.168.14").

- **+stats / +nostats** – Schaltet Statistik-Banner am Ende an/aus.
- **+cm** – Colorize output (if compiled with color support).

Beispiele:

- dig example.com ANY – Fragt alle Records (die server je nach config liefern).
- dig google.com MX +noall +answer – Zeigt nur MX Antworten.
- dig @1.1.1.1 cloudflare.com A – Abfrage an 1.1.1.1 nach cloudflare's A.
- dig -x 8.8.4.4 +short – Reverse Lookup, kurze Ausgabe ("dns.google").
- dig +trace example.org – Startet vom Root Nameserver und verfolgt Delegation bis example.org resolved ist.

nslookup (Name Server Lookup - veraltet)

Beschreibung: Ein einfaches interaktives DNS-Abfrageprogramm. War Teil von BIND, gilt als veraltet im Vergleich zu dig. Viele Admins kennen es dennoch.

Verwendung:

nslookup [<Host> [<DNS-Server>]]

Ohne args geht es in interaktiven Modus. Dessen Kommandos: server <DNS>, set q=MX, set type=AAAA, set debug, exit.

Beispiele:

```
$ nslookup example.com 8.8.8.8
```

Server: 8.8.8.8

Address: 8.8.8.8#53

Non-authoritative answer:

Name: example.com

Address: 93.184.216.34

Reverse: nslookup 93.184.216.34 -> zeigt PTR.

Warnung: Neuere doc empfehlen dig oder host statt nslookup.

whois (Domain Whois Query)

Beschreibung: Fragt Whois-Datenbanken nach Registrierungsinformationen einer Domain oder 5】. Das Tool verbindet sich zum passenden Whois-Server (verwendet /etc/whois.conf, TLD heuristics) via TCP 43 und gibt den Text aus.

Syntax:

whois [Optionen] <DomainOrIP>

Ausgabe: Je nach TLD und Registry differierend. Enthält Domain-Inhaber (bei Generic oft DSGVO anonymisiert), Registrar, Erstellungs- und Ablaufdatum, Nameserver, Contact-IDs. Bei IP WHOIS (z.B. whois 8.8.8.8) gibt es Informationen über den zugeteilten IP-Block (ISP, RIR wie ARIN/RIPE...).

Optionen:

- **-H** – Unterdrückt das Banner "Whois Server" in RIPE responses.
- **-p <port>** – Andere Port.
- **--verbose** – Mehr Info über Server selection.
- **--help** – Display help.

Beispiele:

- whois example.com -> zeigt Registrar (ICANN info), Status, Name der Organisation, etc.
- whois 2620:0:2d0:200::7 -> IPv6 whois (gibt i.d.R. ARIN oder so mit OrgName).
- whois AS15169 -> WHOIS an RADb für Autonomous System 15169 (Google).

(*Beachte: Manche TLDs haben separate Tools, z.B. jwhois config. Das Standard whois ist meist in Debian whois package (Marco d'Itri) which covers a lot. GDPR hat Personendaten stark eingeschränkt in Output.*)

arp (Address Resolution Protocol table) – Hinweis: arp wird mittlerweile durch ip neigh ersetzt.

Beschreibung: Zeigt den ARP-Cache (IPv4) an oder manipuliert Einträge. (Linux modern: use ip neigh aber arp Kommando aus net-tools ist noch gebräuchlich).

Befehle:

- arp -a – Listet ARP-Einträge (Hostname, IP, MAC, Interface).
- arp -n – Wie -a, aber keine DNS für IP.
- arp -d <IP> – Löscht ARP-Eintrag für IP.
- arp -s <IP> <MAC> – Fügt statischen ARP-Eintrag hinzu (MAC fix für IP) – gleich ip neigh add ... nud permanent.

Beispiel: arp -a -> router.lan (192.168.1.1) at aa:bb:cc:dd:ee:ff [ether] on eth0.

Sicherheit und Firewall

iptables/iptables-nft (Packet Filter)

Beschreibung: Linux Kernel-Paketfilter (Firewall) via Netfilter-Schnittstelle. iptables verwaltet Regeln in verschiedenen Tabellen: **filter**, **nat**, **mangle**, **raw**, **security**, für IPv4. (Für IPv6, ip6tables). Mit Kernel 4.x/5.x viele Distros nutzen "iptables-nft" as a frontend to nftables, aber Syntax bleibt für user gleich.

Syntax Grundkommandos:

iptables [-t <table>] <operation> <chain> <matches...> <target>

Operationen: **-A** (append), **-I** (insert), **-D** (delete), **-C** (check exist), **-L** (list), **-F** (flush chain), **-N** (new chain), **-X** (delete user chain).

Standard Tabellen:

- **filter** (Default): Chains **INPUT**, **OUTPUT**, **FORWARD**.
- **nat**: Chains **PREROUTING**, **OUTPUT**, **POSTROUTING** (masquerading, port forwarding etc.).
- **mangle**: For packet alteration (TOS, mark), Chains in multiple points (PREROUTING, OUTPUT, etc.).
- **raw**: For conntrack exemption (NOTRACK), Chains PREROUTING, OUTPUT.
- **security**: SELinux related (rare).

Matches / Options:

- **-p <proto>**: Protocol (tcp, udp, icmp, icmpv6, all).
- **-s <IP/CIDR>, -d <IP/CIDR>**: Source or destination address (or network).
- **-i <iface>, -o <iface>**: Inbound or outbound network interface (for PREROUTING/INPUT/forward vs OUTPUT/POSTROUTING).
- **--sport, --dport** (with -p tcp/udp): Source/dest port or port range (e.g. --dport 80 or --dport 1000:2000).
- **-m state --state <states>**: State match (connection tracking states: INVALID, ESTABLISHED, NEW, RELATED). Common: allow ESTABLISHED,RELATED on INPUT for responses.
- **-m conntrack --ctstate**: (newer usage, same as state module now).
- **-m connlimit, -m limit** etc: Many modules. limit --limit 5/minute for rate-limit logging for example. connlimit --connlimit-above N limit parallel connections.
- **-m multiport**: match multiple ports in one rule if not contiguous, e.g. -m multiport --dports 80,443,8080.

- **-m tcp --tcp-flags SYN,RST,ACK SYN**: match specific TCP flags combination (here SYN only). Often used for --syn (alias) which matches syn packets (to identify new connections).
- ****-m comment --comment "text"**: attach comment to rule (for documentation).

Targets (Actions):

- **ACCEPT**: Packet erlaubt.
- **DROP**: Verwerfen ohne Antwort.
- **REJECT**: Verwerfen mit Antwort (ICMP port unreachable o.Ä.).
- **LOG**: Loggt Paket (via kernel log), dann Verarbeitung geht weiter (so typ. Kombi LOG dann DROP). Options: --log-prefix "msg", --log-level warning.
- **MASQUERADE** (in nat POSTROUTING): SNAT für dynamische IP (Router Use-case) – ersetzt Source-IP durch Interface-IP.
- **SNAT** (in nat POSTROUTING): Statisches Source-NAT (requires --to-source IP[:port-range]).
- **DNAT** (in nat PREROUTING/OUTPUT): Destination-NAT (Port-Forwarding; --to-destination IP:port).
- **REDIRECT** (nat PREROUTING/OUTPUT): Umleiten an lokalen Host (for transparent proxy etc.).
- **MARK** (mangle): Markiert Paket mit fwmark (für routing decisions via ip rule, etc.).
- **RETURN**: In user-defined chain springt zurück zum Aufrufer-Chain (stop further rules in current chain). In builtin chain, RETURN = apply chain policy.
- **QUEUE**: Übergibt Paket an Userspace (via libnetfilter_queue).
- **DROP** vs. policy: If chain policy is DROP, falling off end = drop.

Beispiele:

- **List Regeln**: iptables -L -n -v (filter table). iptables -t nat -L -n -v NAT rules. -n no DNS, -v verbose (packet counters).
- **Allow incoming SSH**:

```
② iptables -A INPUT -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
(typical stateful allow SSH in).
```

② **Drop all forward by default**: iptables -P FORWARD DROP.

② **Masquerading (NAT):**

② `iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE`

(for home router with ppp0 outbound).

② **Port Forward:**

Forward ext eth0 port 80 to internal 192.168.1.100 port 8080:

```
② iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 \
-j DNAT --to-destination 192.168.1.100:8080
```

```
iptables -A FORWARD -p tcp -d 192.168.1.100 --dport 8080 -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT
```

② **Logging Drop:**

- `iptables -A INPUT -p tcp --dport 23 -j LOG --log-prefix "Telnet attempt: " --log-level info`
- `iptables -A INPUT -p tcp --dport 23 -j DROP`
- (log then drop telnet).

firewall-cmd (firewalld frontend)

Beschreibung: Steuerprogramm für **firewalld**, eine dynamische Firewall (standard in RHEL/CentOS 7+, Fedora, etc.). firewall-cmd ermöglicht das Konfigurieren von Zonen, Regeln (services, ports, masq) zur Laufzeit und in Config. Firewalld nutzt intern iptables/nftables.

Syntax:

```
firewall-cmd [--zone=<zone>] [--permanent] <aktion> <Parameter>
```

Grundkonzepte: Zonen (trusted, public, home, internal, work, external, dmz, block, drop etc.), Schnittstellen sind Zonen zugewiesen. Dienste (vordefinierte Regelsets) können pro Zone erlaubt werden oder einzelne Ports/Protokolle. Man kann Einstellungen permanent (persistiert in XML config) oder nur runtime vornehmen.

Wichtige Befehle:

- **--state:** Zeigt an, ob firewalld läuft (running/not running).
- **--get-active-zones:** Zeigt welche Zonen aktiv sind und welche Interfaces zugeordnet (z.B. "public: wlp2s0").
- **--get-zones:** Listet alle bekannten Zonen. **--list-all** [--zone=Z] listet alle Einstellungen einer Zone (services, ports, masquerade etc0) .

- **--zone=<zone> --add-service=<service>**: Erlaubt den vordefinierten service in angegebener Zone (runtime). Bsp: firewall-cmd --zone=public --add-service=http – ermöglicht HTTP (Port 80) in Zone *public*.
- **--zone=<zone> --add-port=<port>/<proto>**: Erlaubt individuellen Port. Bsp: --add-port=5000/tcp.
- **--remove-service=..., --remove-port=...** analog zum Entfernen.
- **--add-masquerade**: Aktiviert NAT/Masquerading in der Zone (z.B. für Internet-sharing zone).
- **--query-service=<svc>**: Fragt ob Service in zone erlaubt (exit code 0/1).
- **--permanent**: Gibt man diese Option an, wird die Änderung in der persistenten Konfiguration vorgenommen (aber nicht an laufende sofort angewendet, es sei denn man ruft extra an). In firewalld, runtime und permanent config sind getrennt.
- **--reload**: Lädt die permanente Konfiguration neu (in runtime) – im Grunde wendet Config an.
- **--runtime-to-permanent**: Speichert aktuellen Laufzeit-Zustand als neue permanente Konfiguration (was man on-the-fly geändert hat, persistieren).
- **--zone=<zone> --change-interface=<iface>**: Weist ein Interface einer Zone zu (persistente Zuordnung).
- **--new-zone=<name>, --delete-zone=...**: Eigene Zonen erstellen/löschen.
- **--list-services, --get-services**: Listet bekannte Service-Namen (die in /etc/firewalld/services/*.xml definiert sind).
- **--service=<svc> --get-ports**: Zeigt welche Ports hinter einem Service stecken.
- **--add-rich-rule '<rule>'**: Fügt sog. Rich Rule hinzu (eine erweiterte Regel-Syntax in firewalld, z.B. mit Logging, Zeitplänen). Z.B.:
 - firewall-cmd --add-rich-rule='rule family="ipv4" source address="192.0.2.0/24" port port="22" protocol="tcp" accept'
 - (Erlaubt SSH nur von 192.0.2.0/24).

Beispiele:

- Öffnen von HTTPS permanent:
- `firewall-cmd --add-service=https --permanent`
- `firewall-cmd --reload`
-

- Port 12345/udp temporär öffnen in Zone public:
firewall-cmd --zone=public --add-port=12345/udp (nach Reboot weg).
- Interface eth0 zur Zone external zuweisen:
firewall-cmd --zone=external --change-interface=eth0 --permanent && firewall-cmd --reload.
- Status: firewall-cmd --list-all (wenn default zone, z.B. public, dann deren config).

ufw (Uncomplicated Firewall)

Beschreibung: Vereinfachtes Firewall-Frontend (Ubuntu/Debian) für iptables. ufw erlaubt einfache allow/deny-Regeln definieren, typischerweise nach Services oder Ports, optional in/out interface.

Syntax:

ufw [Optionen] <Aktion>

Grundbefehle:

- **ufw enable / disable** – Aktiviert oder deaktiviert die Firewall (bei disable: setzt iptables default ACCEPT und löscht Regeln).
- **ufw status [verbose]** – Zeigt aktuelle Regel-Liste. Mit verbose sieht man auch Interfaces, logging status etc.
- **ufw default allow|deny [incoming|outgoing|routed]** – Setzt Standardpolicy. Standard-Einstellung: incoming deny, outgoing allow.
- **ufw allow <Port/Service> [proto tcp/udp] [from <IP> [to <IP>]] [port <P2>]** – Erlaubt eingehenden Traffic. Beispiele:
 - ufw allow 22 (erlaubt TCP/22 auf allen Interfaces).
 - ufw allow proto udp from 10.0.0.0/8 to any port 53 (erlaubt UDP DNS von privatem Netz).
 - ufw allow out 80/tcp (erlaubt ausgehende HTTP).
 - ufw allow in on eth0 to any port 443 (erlaubt eingehendes HTTPS auf eth0).
- **ufw deny <Spec>** – Blockiert (mit REJECT) entsprechend.
- **ufw reject <Spec>** – Synonym zu deny (deny und reject meist ähnlich in ufw; auf Debian might differ).
- **ufw limit <Port/Service>** – Wie allow, aber mit Rate-Limit (gegen Bruteforce). Z.B. ufw limit ssh erlaubt SSH, aber blockiert IP temporär bei >6 Verbindungen/30s.

- **ufw delete <RuleSpec>** – Entfernt Regel (man kann Regel genau wie bei allow angeben, mit 'delete' davor, oder Nummer aus ufw status numbered).
- **ufw logging on|off|LEVEL** – Logging ein/aus und Level (off, low, medium, high, full).
- **ufw reset** – Setzt ufw auf Auslieferungszustand (alle Regeln weg, policies reset, und disabled).
- **ufw route allow/deny ...** – Regeln für Forwarded Traffic (Routen). Bsp: ufw route allow in on eth1 out on eth0 to any port 80 for forwarded.
- **ufw insert <num> <rule>** – Fügt Regel an bestimmter Position ein.
- **ufw show raw** – Zeigt resultierende iptables-Regeln.

Beispiele:

- Standard: ufw default deny incoming; ufw default allow outgoing; ufw enable.
- Webserver: ufw allow 'Nginx Full' (in /etc/ufw/applications.d vordefinierte Profile, 'Apache Full' etc., Nginx Full = 80+443).
- Only LAN access to a service: ufw allow from 192.168.0.0/16 to any port 3306 (MySQL offen nur intern).
- Block eine IP: ufw deny from 203.0.113.5.
- Show rules: ufw status verbose -> e.g.
- Status: active
- Logging: on (low)
- Default: deny (incoming), allow (outgoing), deny (routed)
- New profiles: skip
-
- To Action From
- -- ----- -----
- 22/tcp LIMIT Anywhere # SSH
- 80,443/tcp ALLOW Anywhere # WWW
- 22/tcp (v6) LIMIT Anywhere (v6)
- 80,443/tcp (v6) ALLOW Anywhere (v6)
-

(UFW ist beliebt wegen einfacher Syntax, aber fehlender Komplexität bei komplexen Setups. Firewalld und raw iptables/nft sind mächtiger in Feinheiten.)

nft (nftables Firewall)

Beschreibung: nftables ist der modernere Packet-Filter (ersetzt iptables). Der Befehl nft wird zum Konfigurieren verwendet. Sehr flexibel (Datenstrukturen, one-rule multiple matches).

Syntax:

```
nft [options] <command>
```

Man kann Kommandos interactive eingeben oder als Skript. Der Kommandostil ähnelt C:
e.g.

```
nft add table inet filter
```

```
nft add chain inet filter input { type filter hook input priority 0 \; }
```

```
nft add rule inet filter input tcp dport 22 accept
```

(Librally, but given complexity and that the original question references iptables and ufw, likely beyond scope).

Abschluss

Dieses Kompendium deckt die wesentlichen Befehle für die Linux+ (XK0-005) Prüfung ab, gegliedert nach Themengebieten. Die Befehle umfassen Benutzer- und Rechteverwaltung, Prozess- und System-Tools, Netzwerk-Konfiguration und -Diagnose, Software-Installation, Container, Sicherheit (Firewall, SELinux/AppArmor siehe oben Rich-Rules, etc.), Dienste und Planung.

Durch die strukturierte Darstellung mit Syntax, Beschreibung, Optionen und Beispielen dient es als umfassendes Nachschlagewerk. **Wichtig:** In einer Prüfungssituation sollte man sich auf die korrekte Anwendung der Optionen und das Verständnis der Konzepte dahinter konzentrieren (z.B. was macht chmod 775 oder umask 002 bewirkt).

Mit dieser Referenz sollte die Vorbereitung auf typische Linux-Aufgaben – vom Anlegen eines Benutzers, dem Ändern von Dateirechten, über das Einrichten eines Cronjobs bis zum Konfigurieren eines Apache-Dienstes oder Setzen einer Firewallregel – bestens unterstützt werden.