Basic Linux Quick Reference

Terminal, Bash, Programme, Pakete und systemd

Fast jeder Befehl erklärt sich selbst mit `-help` oder `man`. Mit `Tab` vervollständigst du Namen; mit `Ctrl-r` durchsuchst du die History.

1. Terminal und Dateien

Orientierung

`pwd`	Aktuelles Verzeichnis
`whoami`	Aktueller Benutzer
`hostname`	Rechnername anzeigen
`clear`	Terminal leeren
`man cp`	Handbuch zu cp
`cp -help`	Kurzhilfe anzeigen

Navigieren und Auflisten

`ls -lah`	Auch versteckte Dateien, lesbare Größen
`cd /etc`	In /etc wechseln
`cd ..`	Eine Ebene nach oben
`cd -`	Zum vorigen Verzeichnis
`cd ~`	Ins Home-Verzeichnis
`find . -maxdepth 2`	In zwei Ebenen auflisten

Dateien und Verzeichnisse

`mkdir -p log/old`	Verzeichnisse anlegen
`touch test.txt`	Leere Datei anlegen
`cp -a config backup/`	Kopieren, Attribute erhalten
`mv old.txt new.txt`	Verschieben / umbenennen
`rm -rI build/`	Rekursiv, mit Sicherheitsabfrage
`ln -s target link`	Symbolischen Link anlegen

Anzeigen, Bearbeiten, Archive

`less /var/log/syslog`	Datei seitenweise lesen
`head -n 20 file`	Erste 20 Zeilen
`tail -f app.log`	Datei live verfolgen
`nano file / vim file`	Textdatei bearbeiten
`tar -czf logs.tgz log/`	Komprimiertes Archiv erzeugen
`tar -xzf logs.tgz`	Archiv entpacken

Rechte und Eigentümer

`ls -l file`	Rechte und Eigentümer zeigen
`chmod u+x script.sh`	Für Eigentümer ausführbar
`chmod 640 config`	Besitzer rw, Gruppe r
`sudo chown root:root file`	Eigentümer und Gruppe ändern

Targets und Timer

`systemctl get-default`	Standard-Boot-Target
`systemctl list-units -type=target`	Aktive Targets auflisten
`sudo systemctl set-default multi-user.target`	Textmodus als Standard
`sudo systemctl isolate rescue.target`	Sofort in Rettungsmodus wechseln
`systemctl list-timers -all`	Alle Timer und nächste Läufe

2. Suchen, Prozesse und Bash

Pipes und Umleitungen

`cmd > out.txt`	Ausgabe überschreiben
`cmd >> out.txt`	Ausgabe anhängen
`cmd 2> error.log`	Fehlerausgabe umleiten
`cmd1 | cmd2`	Ausgabe an nächsten Befehl
`make && ./app`	Zweiten Befehl nur bei Erfolg
`test -f x || touch x`	Zweiten Befehl nur bei Fehler

Suchen und Text verarbeiten

`grep -Rni TODO src/`	Rekursiv nach TODO suchen
`find . -name '*.log'`	Dateien nach Namen suchen
`grep ERROR app.log`	Passende Zeilen anzeigen
`sort names | uniq -c`	Sortieren und Treffer zählen
`wc -l *.c`	Zeilen zählen
`sed -n '10,20p' file`	Zeilen 10 bis 20 ausgeben

Prozesse und System

`ps aux`	Alle Prozesse anzeigen
`top / htop`	Prozesse live beobachten
`pgrep -af iperf3`	Prozess nach Namen finden
`kill PID`	Prozess normal beenden
`kill -9 PID`	Nur als letzte Maßnahme
`df -h / du -sh *`	Dateisystem / Verzeichnisse
`free -h / uname -a`	Speicher / Kernel anzeigen

Netzwerk und entfernte Systeme

`ip -br address`	Interfaces und IP-Adressen
`ping -c 4 1.1.1.1`	Erreichbarkeit testen
`curl -I https://example.com`	HTTP-Header abrufen
`ssh pi@192.168.1.20`	Remote-Shell öffnen
`scp file pi@host:/tmp/`	Datei per SSH kopieren

Pakete mit APT

`sudo apt update`	Paketlisten aktualisieren
`sudo apt upgrade`	Installierte Pakete aktualisieren
`sudo apt install tree`	Paket installieren
`sudo apt remove tree`	Paket entfernen
`apt search serial`	Pakete suchen
`apt show iperf3`	Paketdetails anzeigen

Andere Distributionen verwenden z. B. `dnf` oder `pacman`.

Bash, History und Tastentricks

`name=Philipp`	Shell-Variable setzen
`echo "$name"`	Variable sicher ausgeben
`today=$(date +%F)`	Befehlsausgabe einsetzen
`export EDITOR=vim`	Variable an Programme vererben
`alias ll='ls -lah'`	Abkürzung definieren
`history | grep ssh`	History nach ssh durchsuchen
`Ctrl-r / !! / !$`	Suchen / letzter Befehl / letztes Argument
`Ctrl-a / Ctrl-e`	Zum Zeilenanfang / -ende

3. Dienste, Logs und Diagnose

Dienste steuern

`systemctl status ssh.service`	Status und letzte Meldungen
`sudo systemctl start ssh.service`	Jetzt starten
`sudo systemctl stop ssh.service`	Jetzt stoppen
`sudo systemctl restart NetworkManager.service`	Dienst neu starten
`sudo systemctl enable ssh.service`	Beim Booten aktivieren
`sudo systemctl disable ssh.service`	Autostart deaktivieren
`sudo systemctl enable -now my-app.service`	Aktivieren und sofort starten

`start` = ändert nur den aktuellen Laufzeitstatus. `enable` = konfiguriert den Start bei künftigen Boots.

Fehler und Logs

`systemctl -failed`	Fehlgeschlagene Units
`journalctl -u ssh.service`	Logs eines Dienstes
`journalctl -u my-app.service -b`	Dienst seit aktuellem Boot
`journalctl -b`	Alle Logs seit dem Boot
`journalctl -f`	Neue Meldungen live verfolgen
`journalctl -p warning..alert -b`	Nur wichtige Meldungen

Units und Service-Dateien

`systemctl list-units -type=service`	Geladene Dienste auflisten
`systemctl list-unit-files`	Installierte Unit-Dateien
`systemctl cat my-app.service`	Effektive Datei anzeigen
`sudo systemctl edit -full my-app.service`	Service-Datei bearbeiten
`sudo systemctl daemon-reload`	Unit-Dateien neu einlesen
`sudo systemctl restart my-app.service`	Änderung anwenden

Typische Fehleranalyse

`systemctl status my-app.service`	Fehler, Exit-Code, Pfad prüfen
`journalctl -u my-app.service -n 50`	Letzte 50 Meldungen
`systemctl cat my-app.service`	User, ExecStart und Abhängigkeiten
`systemd-analyze verify my-app.service`	Unit-Syntax prüfen
`command -v my-app`	Programm im Suchpfad finden
`ss -lntup`	Lauschende Ports und Prozesse