Linux Command Cheat Sheet

The everyday working set. Anything that looks like a name, path or number is yours to replace. When in doubt, `man cmd` or `cmd --help`.

NAVIGATING AND MANAGING FILES

`pwd` Where am I
`cd /path` Change directory
`cd -` Back to the previous directory
`ls -lah` Long listing, hidden files, human sizes
`ls -ltr` Sort by time, newest last
`tree -L 2` Tree view, two levels deep
`mkdir -p a/b/c` Create nested directories
`touch file` Create empty file, or bump its timestamp
`cp -r src dst` Copy a directory
`cp -a src dst` Copy, keeping permissions and times
`mv old new` Move or rename
`rm -i file` Delete, asking first
`rm -rf dir` Delete a tree. No prompts, no undo
`ln -s target link` Make a symbolic link
`readlink -f file` Resolve to the real absolute path
`basename /a/b/c.txt` Prints c.txt
`dirname /a/b/c.txt` Prints /a/b
`stat file` Size, inode, permissions, timestamps
`file report.bin` Guess the file type
`du -sh dir` Total size of one directory

READING AND COMPARING FILES

`cat file` Dump the whole thing
`less file` Page through it. q quits, / searches
`head -n 20 file` First 20 lines
`tail -n 50 file` Last 50 lines
`tail -f app.log` Follow as it grows
`tail -F app.log` Follow across log rotation
`nl file` Number the lines
`wc -l file` Count lines
`diff -u a b` Unified diff
`diff -r dir1 dir2` Compare two trees
`sha256sum file` Checksum a download
`xxd file | head` Hex dump the first bytes
`split -b 10M big part_` Cut into 10 MB chunks
`column -t file` Line up ragged columns

PERMISSIONS AND OWNERSHIP

Digits add up: 4 read, 2 write, 1 execute. Order is owner, group, other.

`chmod 644 file` Owner writes, everyone reads
`chmod 755 script.sh` Everyone may read and run
`chmod +x script.sh` Make it executable
`chmod -R g+w dir` Group write, all the way down
`chown user:group file` Change owner and group
`chown -R user dir` Change owner recursively
`chgrp staff file` Change the group only
`umask 022` Defaults for newly created files
`ls -ld dir` A directory's own permissions
`setfacl -m u:bob:rw file` Grant one extra user access
`getfacl file` Show the access control list
`sudo -u www-data cmd` Run as another user
`sudo -i` Interactive root shell
`visudo` Edit sudoers with a syntax check

FINDING THINGS

`find . -name "*.log"` Match by name below here
`find . -iname "*.LOG"` Same, ignoring case
`find . -type f -size +100M` Files over 100 MB
`find . -mtime -7` Changed in the last week
`find . -type d -empty -delete` Prune empty directories
`find . -name "*.tmp" -delete` Delete every match
`find . -name "*.c" -exec grep -l TODO {} +` Run a command on the matches
`locate nginx.conf` Instant lookup from an index
`updatedb` Rebuild that index
`which python3` Path of the command that runs
`type -a ls` Alias, builtin or file
`whereis gcc` Binary, source and man pages

TEXT PROCESSING

`grep -i error app.log` Search, ignoring case
`grep -r TODO src/` Search a whole tree
`grep -n pattern file` Show line numbers
`grep -v debug app.log` Show lines that do not match
`grep -c pattern file` Count matching lines
`grep -A3 -B3 error log` Three lines of context each side
`grep -E "cat|dog" file` Either word
`grep -oE "[0-9]+" file` Print just the match
`sed 's/old/new/g' file` Substitute everywhere
`sed -i.bak 's/a/b/g' file` Edit in place, keep a backup
`sed -n '10,20p' file` Print lines 10 to 20
`sed '/^#/d' file` Drop comment lines
`awk '{print $3}' file` Print the third field
`awk -F: '{print $1}' /etc/passwd` Split on a custom delimiter
`awk '$3 > 100' data` Rows where field three is over 100
`awk '{s+=$2} END {print s}' f` Sum a column
`cut -d, -f2,5 data.csv` Pull two fields out of a CSV
`sort -u file` Sort and drop duplicates
`sort -k2 -n file` Numeric sort on field two
`sort file | uniq -c` Tally repeated lines
`tr 'a-z' 'A-Z'` Translate characters
`tr -d '\r' < win.txt` Strip Windows line endings
`jq '.items[].name' d.json` Query JSON
`xargs -n1 command` Turn input lines into arguments

ARCHIVES AND COMPRESSION

`tar -czf out.tgz dir/` Create a gzipped archive
`tar -xzf out.tgz` Extract it here
`tar -xzf f.tgz -C /dest` Extract somewhere else
`tar -tzf f.tgz` List contents without unpacking
`tar -cJf out.txz dir/` xz archive: smaller, slower
`gzip file` Compress one file in place
`gunzip file.gz` Expand it again
`zip -r out.zip dir/` Build a zip
`unzip out.zip -d dest/` Unpack into a directory
`unzip -l out.zip` Peek inside a zip
`zcat file.gz` Read a gzip without expanding
`zgrep pattern log.gz` Grep compressed logs directly

REDIRECTION, PIPES AND CHAINING

`cmd > file` Write output to a file, replacing
`cmd >> file` Append instead
`cmd 2> err.log` Capture errors only
`cmd &> all.log` Capture output and errors
`cmd 2>&1 | less` Pipe both streams to a pager
`cmd < input.txt` Feed a file in as input
`cmd | tee log` Watch it and save it
`cmd1 && cmd2` Second runs only if the first worked
`cmd1 || cmd2` Second runs only if the first failed
`cmd1 ; cmd2` One after the other regardless
`cmd &` Run in the background
`$(cmd)` Substitute a command's output
`cmd > /dev/null 2>&1` Throw all output away
`cmd <<'EOF' ... EOF` Feed a literal block in

WILDCARDS AND QUOTING

`*` Any run of characters
`?` Exactly one character
`[abc]` Any one of those characters
`{a,b}.txt` Expands to a.txt b.txt
`~` Your home directory
`'single quotes'` Nothing inside is expanded
`"double quotes"` $variables still expand
`\ at line end` Continue the command on the next line

EDITING IN PLACE: NANO AND VIM

`nano file` Ctrl+O saves, Ctrl+X exits
`vim file` Modal. Esc always returns to normal mode
`i` Start typing. Esc when done
`:w :q :wq :q!` Write, quit, both, quit discarding
`dd / yy / p` Cut a line, copy a line, paste
`/text then n` Search, then jump to the next hit
`gg / G` Top of file, bottom of file
`:%s/old/new/g` Replace throughout the file
`u / Ctrl+R` Undo, redo

Linux Command Cheat Sheet

Running systems: processes, hardware, disks, users, network, SSH, packages, services, shell.

PROCESSES AND JOBS

ps aux Every process on the box
ps -ef | grep nginx Find a process by name
pgrep -a sshd Matching PIDs and command lines
top Live view of the load
htop The same, friendlier
kill 1234 Ask it to stop
kill -9 1234 Make it stop. Last resort
pkill -f "python app" Kill by full command match
nice -n 10 cmd Start at lower priority
jobs This shell's background jobs
Ctrl+Z Suspend the foreground job
bg %1 / fg %1 Resume it, back or front
nohup cmd & Survives you logging out
timeout 30 cmd Give up after 30 seconds
watch -n 2 'cmd' Rerun every two seconds
lsof -p 1234 Files that process has open
lsof -i :8080 What is holding a port
strace -p 1234 Trace its system calls

SYSTEM AND HARDWARE

uname -a Kernel version and architecture
cat /etc/os-release Distribution and release
hostnamectl Hostname, OS and kernel at a glance
uptime Time since boot, load averages
free -h Memory and swap
vmstat 1 Live memory, CPU and IO counters
lscpu Cores, model, cache
lsblk Block devices and mount points
lspci / lsusb PCI and USB hardware
dmesg -T | tail Kernel messages, real timestamps
sensors Temperatures and fan speeds
timedatectl Time zone and clock sync
date "+%F %T" Time in a chosen format

DISKS AND FILESYSTEMS

df -h Free space per filesystem
df -i Inodes, when space looks fine but writes fail
du -sh * | sort -h Size of everything here, sorted
du -ah dir | sort -rh | head Biggest files in a tree
ncdu / Browse disk usage interactively
mount /dev/sdb1 /mnt Mount a device
umount /mnt Unmount it
mount | column -t What is mounted now
blkid UUIDs and filesystem types
mkfs.ext4 /dev/sdb1 Format a partition. Erases it
fsck -f /dev/sdb1 Check an unmounted filesystem
rsync -avh src/ dst/ Mirror a directory locally
dd if=x.iso of=/dev/sdX bs=4M status=progress Write an image to a device

USERS AND GROUPS

whoami / id Who you are, UID and groups
who / w Who is on, and what they are doing
useradd -m -s /bin/bash bob New user with home and shell
passwd bob Set a password
usermod -aG sudo bob Add to a group
userdel -r bob Remove user and home directory
groups bob Their group memberships
groupadd devs Create a group
su - bob Switch user, full login shell
last Recent login history

NETWORKING

ip a Interfaces and addresses
ip r Routing table
ping -c 4 host Four packets, then stop
traceroute host The path packets take
mtr host Traceroute plus live loss stats
ss -tulpn Listening ports and their processes
dig example.com +short DNS lookup, answer only
dig -x 1.1.1.1 Reverse lookup
curl -I https://site Response headers only
curl -fsSL url -o file Download quietly, fail loudly
wget -c url Download, resume if it drops
nc -zv host 443 Is that TCP port open
ufw allow 22/tcp Open a port, Debian firewall
ufw status verbose Review the rules
firewall-cmd --list-all The same, Red Hat side
tcpdump -i any port 80 Capture traffic on a port

SSH AND REMOTE COPIES

ssh user@host Shell on another machine
ssh -p 2222 user@host Non-standard port
ssh-keygen -t ed25519 Generate a key pair
ssh-copy-id user@host Install your public key there
ssh-add ~/.ssh/id_ed25519 Load a key into the agent
ssh -L 8080:localhost:80 host Forward a remote port to yours
ssh -J jump user@host Hop through a bastion
scp file user@host:/path Copy a file up
scp -r dir user@host:/path Copy a directory up
rsync -avzP src/ host:dst/ Resumable sync with progress
rsync -avz --delete src/ host:dst/ Mirror exactly, prune extras
~/.ssh/config Save Host, User, Port and key per server

INSTALLING SOFTWARE

apt update && apt upgrade Debian, Ubuntu: refresh then update
apt install pkg Install
apt remove pkg / apt purge pkg Remove, keeping or dropping config
apt search term / apt show pkg Look for it, inspect it
dpkg -i file.deb Install a downloaded .deb
dnf install pkg / dnf remove pkg Fedora, RHEL
dnf search term / dnf upgrade Look for it, update everything
rpm -qa | grep pkg Query installed RPMs
pacman -Syu Arch: refresh and upgrade
pacman -S pkg / pacman -Rns pkg Install or fully remove
snap install pkg Sandboxed, any distro
flatpak install pkg Desktop apps, any distro

SERVICES, LOGS AND SCHEDULING

systemctl status nginx Running or not, plus recent log
systemctl restart nginx Stop and start
systemctl reload nginx Reread config, keep connections
systemctl enable --now nginx Start now and at every boot
systemctl disable nginx Stop it starting at boot
systemctl daemon-reload Pick up edited unit files
systemctl list-units --type=service Everything systemd manages
journalctl -u nginx -f Follow one service's log
journalctl -xe Recent failures with context
journalctl --since "1 hour ago" Logs from a time window
journalctl -p err -b Errors since this boot
journalctl --vacuum-time=7d Trim the journal to a week
crontab -e / crontab -l Edit or list scheduled jobs
systemctl list-timers systemd's answer to cron

SHELL SHORTCUTS AND HISTORY

Ctrl+C Interrupt what is running
Ctrl+D End of input, or exit
Ctrl+L Clear the screen
Ctrl+R Search history backwards
Ctrl+A / Ctrl+E Start or end of the line
Ctrl+U / Ctrl+K Cut to start or to end
Ctrl+W Delete the word behind you
Alt+. Insert the last argument you typed
Tab, Tab Tab Complete, or list the options
history Numbered list of what you ran
!! / sudo !! Repeat it, or repeat it as root
!$ Last argument of the previous command
alias ll='ls -lah' Short name for a long command
export VAR=value Set an environment variable
echo $PATH Where the shell looks for programs
source ~/.bashrc Reload your shell config
man cmd / cmd --help The manual, and the quick version