

Linux Commands Cheatsheet

A practical reference covering 300+ commands across files, processes, networking, and system administration. Tested on Ubuntu 24.04, Debian 12, and RHEL 9.

1. File & Directory Operations

<code>ls -lah</code>	List files with sizes (human-readable) including hidden
<code>ls -ltr</code>	Sort by modification time, oldest first
<code>cd -</code>	Switch to previous working directory
<code>pwd</code>	Print working directory
<code>mkdir -p a/b/c</code>	Create nested directories at once
<code>cp -av src dst</code>	Copy preserving attributes, verbose
<code>mv old new</code>	Move or rename
<code>rm -rf dir</code>	Remove directory recursively (DESTRUCTIVE)
<code>ln -s tgt link</code>	Create symbolic link
<code>readlink -f path</code>	Resolve absolute path of a symlink
<code>stat file</code>	Show file metadata, permissions, timestamps
<code>touch file</code>	Create empty file or update timestamp
<code>file file</code>	Identify file type by content
<code>du -sh *</code>	Disk usage of each item, human-readable
<code>du -h --max-depth=1</code>	Top-level breakdown
<code>df -h</code>	Disk free per mount, human-readable
<code>tree -L 2</code>	Directory tree, 2 levels deep
<code>find . -name "*.log" -mtime +7 -delete</code>	Delete logs older than 7 days
<code>find . -type f -size +100M</code>	Find files larger than 100MB
<code>locate file.txt</code>	Fast filename search using db
<code>which python3</code>	Show path of executable in PATH
<code>whereis cmd</code>	Show binary, source, and man page locations

2. File Content & Text Processing

<code>cat file</code>	Show entire file
<code>less file</code>	Pager: q to quit, / to search, n for next match
<code>head -n 50 file</code>	First 50 lines
<code>tail -f log</code>	Follow log in real time
<code>tail -F log</code>	Follow even if log rotates
<code>wc -l file</code>	Count lines
<code>sort file uniq -c sort -rn</code>	Count and rank distinct lines
<code>cut -d: -f1 /etc/passwd</code>	Extract 1st field (colon-separated)
<code>awk '{print \$2,\$5}' f</code>	Print 2nd and 5th columns

<code>awk -F, '\$3>100' f</code>	CSV: print rows where col 3 > 100
<code>sed -i 's/old/new/g' f</code>	In-place find/replace
<code>sed -n '10,20p' f</code>	Print lines 10-20
<code>grep -rn "TODO" .</code>	Recursive search with line numbers
<code>grep -E "err warn" f</code>	Extended regex match
<code>grep -v "^#" cfg</code>	Hide comment lines
<code>tr -d "\r" < f > f2</code>	Strip carriage returns
<code>diff -u a b</code>	Unified diff between two files
<code>comm -12 a b</code>	Lines common to both sorted files
<code>xargs -I{} echo {}</code>	Run command per stdin line
<code>paste a b</code>	Side-by-side merge
<code>column -t -s,</code>	Format CSV as aligned table
<code>jq '.users[].name' f.json</code>	Extract names from JSON

3. Permissions & Ownership

<code>chmod 755 file</code>	rwx for owner, rx for group/others
<code>chmod +x script</code>	Make executable
<code>chmod -R g+w dir</code>	Recursively add group write
<code>chown user:group f</code>	Change owner and group
<code>chown -R u: dir</code>	Recursively chown to user
<code>umask 022</code>	Default permission mask
<code>getfacl file</code>	Read POSIX ACL
<code>setfacl -m u:bob:rw f</code>	Grant Bob read+write via ACL
<code>chattr +i file</code>	Make immutable (root only)
<code>lsattr file</code>	List extended attributes
<code>sudo -u alice cmd</code>	Run cmd as Alice
<code>sudo !!</code>	Re-run last command with sudo
<code>passwd</code>	Change own password
<code>su - user</code>	Switch user with login env

4. Process Management

<code>ps auxf</code>	Process tree with all users
<code>ps -eo pid,pcpu,pmem,cmd --sort=-pcpu</code>	Sort by CPU usage
<code>top</code>	Live process monitor
<code>htop</code>	Interactive process viewer (better)
<code>kill -9 PID</code>	Force kill
<code>kill -HUP PID</code>	Hangup signal (reload config)
<code>killall nginx</code>	Kill all by name
<code>pkill -f "node.*server"</code>	Pattern-based kill
<code>pgrep -lf python</code>	List PIDs matching pattern
<code>nohup cmd &</code>	Run detached, survive logout
<code>disown %1</code>	Detach background job
<code>jobs -l</code>	List background jobs
<code>fg %1 / bg %1</code>	Foreground/background a job
<code>renice -n 10 PID</code>	Lower priority
<code>time cmd</code>	Measure command runtime
<code>strace -p PID</code>	Trace syscalls of running process
<code>lsof -i :8080</code>	What's using port 8080?
<code>lsof -p PID</code>	Files opened by process
<code>fuser -v /path</code>	Who is using this file/dir?

5. Networking

<code>ip a / ip addr</code>	Show all interfaces and IPs
<code>ip r / ip route</code>	Show routing table
<code>ip link set eth0 up</code>	Bring interface up
<code>ping -c 4 host</code>	Send 4 ICMP echoes
<code>traceroute host</code>	Show network path to host
<code>mtr host</code>	Live combined ping + traceroute
<code>dig +short example.com</code>	DNS query, terse output
<code>dig @8.8.8.8 host</code>	Query specific DNS server
<code>nslookup host</code>	Classic DNS lookup
<code>host host</code>	Simple DNS lookup
<code>curl -I url</code>	HTTP HEAD request only
<code>curl -sSL url -o f</code>	Download silently, follow redirects
<code>wget -c url</code>	Resume interrupted download
<code>ss -tulpn</code>	Show listening TCP/UDP sockets
<code>ss -tn state established</code>	Active TCP connections
<code>netstat -rn</code>	Routing table (legacy)

<code>nc -lvp 4444</code>	Listen on port 4444 (netcat)
<code>nc -vz host 22</code>	Test if port is open
<code>iptables -L -n -v</code>	Show firewall rules
<code>ufw status</code>	UFW firewall status
<code>tcpdump -ni any port 80</code>	Capture HTTP traffic

6. Package Management

<code>apt update</code>	Refresh package index (Debian/Ubuntu)
<code>apt upgrade -y</code>	Upgrade installed packages
<code>apt install pkg</code>	Install package
<code>apt remove pkg</code>	Remove (keep config)
<code>apt purge pkg</code>	Remove with config
<code>apt autoremove</code>	Remove orphaned dependencies
<code>apt search keyword</code>	Search package database
<code>apt show pkg</code>	Show package details
<code>dpkg -l</code>	List installed packages (Debian)
<code>dpkg -L pkg</code>	List files installed by pkg
<code>dnf install pkg</code>	Install (Fedora/RHEL 9+)
<code>yum update</code>	Update packages (RHEL 7/8)
<code>rpm -qa</code>	List installed packages (RHEL)
<code>snap install app</code>	Install Snap package
<code>flatpak install app</code>	Install Flatpak
<code>pip install pkg</code>	Python package install
<code>npm install -g pkg</code>	Node global install

7. System Information

<code>uname -a</code>	Kernel + arch + hostname
<code>hostnamectl</code>	Hostname and OS info
<code>lsb_release -a</code>	Distribution details
<code>cat /etc/os-release</code>	Same, more portable
<code>uptime</code>	System uptime + load avg
<code>w / who</code>	Who is logged in
<code>lscpu</code>	CPU details
<code>free -h</code>	RAM usage human-readable
<code>vmstat 1 5</code>	Virtual memory stats every 1s, 5 times
<code>iostat -x 1</code>	Per-disk I/O statistics
<code>lsblk</code>	Block devices tree
<code>blkid</code>	UUIDs and filesystem types
<code>lsusb / lspci</code>	USB / PCI devices
<code>dmesg -T tail</code>	Kernel ring buffer with timestamps
<code>journalctl -u sshd</code>	systemd logs for sshd
<code>journalctl -f</code>	Follow system journal
<code>systemctl status nginx</code>	Service status
<code>systemctl restart nginx</code>	Restart service
<code>systemctl enable --now app</code>	Enable + start at once

8. Compression & Archives

<code>tar -czvf out.tgz dir</code>	Create gzip tarball
<code>tar -xzvf in.tgz</code>	Extract gzip tarball
<code>tar -tzvf in.tgz</code>	List contents without extracting
<code>tar -cJvf out.txz dir</code>	Create xz tarball (smaller)
<code>zip -r out.zip dir</code>	Create zip
<code>unzip -l in.zip</code>	List zip contents
<code>gzip / gunzip file</code>	Single-file compression
<code>zstd -19 file</code>	High-ratio modern compression

9. SSH & Remote

<code>ssh user@host</code>	Connect via SSH
<code>ssh -p 2222 user@host</code>	Custom port
<code>ssh -i ~/.ssh/key user@host</code>	Specific key
<code>ssh-keygen -t ed25519</code>	Generate modern keypair
<code>ssh-copy-id user@host</code>	Install your public key on host
<code>scp file user@host:/dst</code>	Copy file to remote
<code>scp -r dir user@host:/dst</code>	Recursive copy
<code>rsync -avz src/ user@host:dst/</code>	Efficient sync, preserve attrs
<code>rsync --delete</code>	Mirror: remove files not in source
<code>ssh -L 8080:localhost:80 host</code>	Local port forward
<code>ssh -D 1080 host</code>	SOCKS proxy via SSH

10. Shell Productivity

<code>!!</code>	Last command
<code>!\$</code>	Last argument of last command
<code>Ctrl-R</code>	Reverse search history
<code>Ctrl-A / Ctrl-E</code>	Cursor to start / end of line
<code>Ctrl-W</code>	Delete word to the left
<code>Ctrl-U</code>	Delete to start of line
<code>history grep apt</code>	Search command history
<code>alias gs="git status"</code>	Define alias
<code>export PATH=\$PATH:/opt/bin</code>	Append to PATH
<code>command1 && command2</code>	Run 2nd only if 1st succeeds
<code>command1 command2</code>	Run 2nd only if 1st fails
<code>command > file 2>&1</code>	Redirect both stdout and stderr
<code>command >> file</code>	Append instead of overwrite
<code>cmd1 tee out cmd2</code>	Pipe and save copy