

Linux Command Reference - 250 Commands

A practical Linux administration and troubleshooting reference covering shell navigation, files, text processing, permissions, processes, systemd, networking, storage, packages, and security.

DOCUMENT CENTER

Compiled September 2026

CAUTION: Commands that remove files, change accounts or permissions, alter services, modify storage, or change firewall policy can interrupt service or cause data loss. Prefer read-only inspection first and maintain backups/recovery access.

Use this guide as a practical reference. Verify commands, targets, permissions, paths, versions, and organizational policy before making changes. Examples with placeholders must be adapted to the environment.

Contents

#	Section	Entries
1	Help, shell navigation, and environment	25
2	Files, directories, links, and text viewing	25
3	Search, compare, transform, and parse text	25
4	Permissions, ownership, identity, and access	25
5	Processes, jobs, CPU, memory, and performance	25
6	Services, boot, systemd, and logs	25
7	Networking, DNS, sockets, and routes	25
8	Storage, filesystems, mounts, and block devices	25
9	Archives, compression, packages, and software inventory	25
10	Security, hashes, certificates, firewall, and auditing	25

Total reference entries: 250

1. Help, shell navigation, and environment

Command	Purpose / Notes
<code>man ls</code>	Open the manual page for <code>ls</code> .
<code>man -k network</code>	Search manual-page descriptions for a keyword.
<code>whatis ssh</code>	Show one-line manual descriptions.
<code>apropos filesystem</code>	Search manual page names and descriptions.
<code>help cd</code>	Show Bash help for a shell builtin.
<code>type ls</code>	Show how the shell resolves a command.
<code>command -v python3</code>	Show the command path the shell would execute.
<code>which ssh</code>	Locate a command in <code>PATH</code> where supported.
<code>pwd</code>	Print the current working directory.
<code>cd /var/log</code>	Change to a directory.
<code>cd -</code>	Return to the previous directory.
<code>pushd /tmp</code>	Push a directory onto the directory stack.
<code>popd</code>	Return to the prior directory in the stack.
<code>dirs -v</code>	Show the directory stack with indexes.
<code>echo "\$PATH"</code>	Display the executable search path.
<code>printenv</code>	List environment variables.
<code>env sort</code>	List environment variables in sorted order.
<code>export EDITOR=vim</code>	Set an environment variable for the current shell.
<code>unset TEMPVAR</code>	Remove an environment variable from the current shell.
<code>history</code>	Show shell command history.
<code>history tail -20</code>	Show the most recent history entries.
<code>alias</code>	List shell aliases.
<code>alias ll='ls -aF'</code>	Create a temporary shell alias.
<code>unalias ll</code>	Remove a shell alias.
<code>clear</code>	Clear the terminal display.

2. Files, directories, links, and text viewing

Command	Purpose / Notes
ls	List directory entries.
ls -lah	Long listing including hidden files with human-readable sizes.
ls -lt	Sort a long listing by modification time.
tree -L 2	Show a directory tree to two levels when tree is installed.
stat file.txt	Show metadata for a file.
file image.bin	Identify a file type by content.
touch notes.txt	Create an empty file or update timestamps.
mkdir project	Create a directory.
mkdir -p a/b/c	Create nested directories as needed.
rmdir emptydir	Remove an empty directory.
cp source.txt backup.txt	Copy a file.
cp -a config/ backup/	Archive-copy a directory while preserving metadata.
mv old.txt new.txt	Move or rename a file.
rm file.txt	Remove a file. Verify the target first.
rm -i file.txt	Prompt before removing a file.
rm -r olddir	Remove a directory tree. Use caution.
ln file.txt hardlink.txt	Create a hard link.
ln -s /opt/app app-link	Create a symbolic link.
readlink app-link	Show the target of a symbolic link.
realpath app-link	Resolve a path to an absolute canonical path.
cat file.txt	Print a file to standard output.
less file.txt	Page through a text file interactively.
head -n 20 file.txt	Show the first 20 lines.
tail -n 20 file.txt	Show the last 20 lines.
tail -f /var/log/syslog	Follow appended log entries in real time.

3. Search, compare, transform, and parse text

Command	Purpose / Notes
grep -n 'error' app.log	Search for matching text and show line numbers.
grep -Rni 'password' /etc/app	Recursively search text, ignoring case.
grep -v '^#' config	Exclude lines matching a pattern.
find /var/log -type f -name '*.log'	Find log files by name.
find /tmp -type f -mtime +7	Find files modified more than seven days ago.
find . -type f -size +100M	Find files larger than 100 MB.
locate ssh_config	Search the locate database for a path.
updatedb	Refresh the locate database; typically requires root.
diff -u old.conf new.conf	Compare two files in unified format.
cmp file1 file2	Compare two files byte by byte.
comm sorted1 sorted2	Compare two sorted text files by lines.
sort users.txt	Sort lines alphabetically.
sort -u users.txt	Sort and remove duplicate lines.
uniq -c sorted.txt	Count adjacent duplicate lines.
cut -d: -f1 /etc/passwd	Extract the first colon-delimited field.
paste file1 file2	Merge lines from files side by side.
tr '[:lower:]' '[:upper:]'	Translate lower-case text to upper-case.
sed -n '1,20p' file.txt	Print a selected line range.
sed 's/foo/bar/g' file.txt	Replace text in the output stream.
awk -F: '{print \$1,\$3}' /etc/passwd	Print selected fields using awk.
wc -l file.txt	Count lines in a file.
wc -w file.txt	Count words in a file.
xargs -n1 echo < items.txt	Build commands from standard input.
tee output.txt	Copy standard input to a file and standard output.
column -t -s, data.csv	Align delimiter-separated text into columns.

4. Permissions, ownership, identity, and access

Command	Purpose / Notes
id	Show the current user and group IDs.
whoami	Show the effective user name.
groups	List group memberships.
who	Show currently logged-in users.
w	Show logged-in users and their activity.
last	Show login history.
lastlog	Show the most recent login for accounts.
getent passwd	Query account entries through NSS.
getent group	Query group entries through NSS.
chmod 640 file.txt	Set owner/group/other permissions numerically.
chmod u+x script.sh	Add execute permission for the owner.
chmod -R g+rX shared/	Recursively grant group read and directory execute access.
chown user:group file.txt	Change file owner and group.
chgrp admins file.txt	Change the file group.
umask	Show the current file-creation mask.
umask 027	Set a restrictive creation mask for the shell.
sudo -l	List commands the current user may run with sudo.
sudo command	Run an authorized command with elevated privileges.
su - user	Start a login shell as another user if permitted.
passwd	Change the current account password.
passwd username	Change another account password as root.
useradd -m analyst	Create a local user with a home directory.
usermod -aG sudo analyst	Append a user to a supplemental group.
userdel -r olduser	Remove a local user and home directory. Use caution.
groupadd security	Create a local group.

5. Processes, jobs, CPU, memory, and performance

Command	Purpose / Notes
<code>ps aux</code>	Show a broad process snapshot.
<code>ps -ef</code>	Show processes in full-format listing.
<code>ps -eo pid,ppid,user,cmd,%cpu,%mem --sort=-%cpu</code>	Sort process output by CPU usage.
<code>pgrep sshd</code>	Find process IDs by name.
<code>pgrep -a python</code>	Find process IDs and show command lines.
<code>pidof sshd</code>	Show PIDs for a program where supported.
<code>top</code>	Interactive process and load monitor.
<code>htop</code>	Interactive process monitor when installed.
<code>uptime</code>	Show uptime and load averages.
<code>free -h</code>	Show memory and swap usage.
<code>vmstat 1 5</code>	Sample CPU, memory, and I/O counters.
<code>iostat -xz 1 3</code>	Show extended device I/O statistics when sysstat is installed.
<code>mpstat -P ALL 1 3</code>	Show per-CPU utilization when sysstat is installed.
<code>nice -n 10 command</code>	Start a process with adjusted scheduling priority.
<code>renice 10 -p <pid></code>	Change the priority of a running process.
<code>jobs</code>	List jobs started from the current shell.
<code>bg %1</code>	Continue a stopped shell job in the background.
<code>fg %1</code>	Bring a shell job to the foreground.
<code>nohup command &</code>	Run a command immune to terminal hangup and in background.
<code>kill <pid></code>	Send SIGTERM to a process.
<code>kill -HUP <pid></code>	Send SIGHUP to a process.
<code>kill -9 <pid></code>	Force termination with SIGKILL. Use only when graceful stop fails.
<code>pkill -f pattern</code>	Signal processes matching a command-line pattern.
<code>timeout 30 command</code>	Stop a command if it exceeds a time limit.
<code>/usr/bin/time -v command</code>	Measure runtime and resource usage.

6. Services, boot, systemd, and logs

Command	Purpose / Notes
<code>systemctl status ssh</code>	Show service status and recent log context.
<code>systemctl start ssh</code>	Start a service.
<code>systemctl stop ssh</code>	Stop a service.
<code>systemctl restart ssh</code>	Restart a service.
<code>systemctl reload ssh</code>	Reload service configuration if supported.
<code>systemctl enable ssh</code>	Enable a service at boot.
<code>systemctl disable ssh</code>	Disable boot-time activation.
<code>systemctl is-active ssh</code>	Check whether a service is active.
<code>systemctl is-enabled ssh</code>	Check whether a service is enabled.
<code>systemctl list-units --type=service</code>	List loaded service units.
<code>systemctl list-unit-files --type=service</code>	List service unit files and enablement state.
<code>systemctl --failed</code>	List failed systemd units.
<code>systemctl daemon-reload</code>	Reload systemd unit definitions after edits.
<code>journalctl -b</code>	Show journal entries for the current boot.
<code>journalctl -b -1</code>	Show journal entries for the previous boot.
<code>journalctl -u ssh --since today</code>	Show today's entries for a service.
<code>journalctl -p warning..alert</code>	Show warning through alert priority events.
<code>journalctl --disk-usage</code>	Show journal disk usage.
<code>dmesg --level=err,warn</code>	Show kernel warnings and errors.
<code>dmesg -T tail</code>	Show recent kernel messages with readable timestamps.
<code>hostnamectl</code>	Show host and operating-system identity.
<code>timedatectl</code>	Show time, timezone, and NTP status.
<code>localectl</code>	Show locale and keyboard settings.
<code>reboot</code>	Restart the system. Save work and verify impact first.
<code>shutdown -h now</code>	Shut down immediately. Use caution on remote systems.

7. Networking, DNS, sockets, and routes

Command	Purpose / Notes
<code>ip addr show</code>	Show IP addresses and interfaces.
<code>ip -br addr</code>	Show a compact address summary.
<code>ip link show</code>	Show link-layer interfaces and state.
<code>ip route show</code>	Show the routing table.
<code>ip neigh show</code>	Show the neighbor/ARP cache.
<code>ip route get 8.8.8.8</code>	Show the route the kernel would use to a destination.
<code>ss -tulpen</code>	Show listening TCP/UDP sockets with process details where permitted.
<code>ss -tan state established</code>	Show established TCP connections.
<code>ping -c 4 1.1.1.1</code>	Test IP reachability with four ICMP echo requests.
<code>ping -c 4 example.com</code>	Test reachability and name resolution.
<code>traceroute example.com</code>	Trace network hops where traceroute is installed.
<code>tracepath example.com</code>	Trace path and path-MTU information where available.
<code>dig example.com</code>	Query DNS using dig.
<code>dig +short example.com</code>	Return concise DNS answers.
<code>dig @1.1.1.1 example.com</code>	Query a specific DNS resolver.
<code>host example.com</code>	Perform a simple DNS lookup.
<code>nslookup example.com</code>	Query DNS using nslookup.
<code>resolvectl status</code>	Show systemd-resolved DNS configuration where used.
<code>nmcli device status</code>	Show NetworkManager device state.
<code>nmcli connection show</code>	List NetworkManager connection profiles.
<code>nmcli device wifi list</code>	List nearby Wi-Fi networks where permitted.
<code>curl -I https://example.com</code>	Fetch HTTP response headers.
<code>curl -v https://example.com</code>	Show verbose HTTP/TLS connection diagnostics.
<code>wget --spider https://example.com</code>	Check resource reachability without downloading content.
<code>ethtool eth0</code>	Show Ethernet interface driver/link details where installed.

8. Storage, filesystems, mounts, and block devices

Command	Purpose / Notes
<code>df -h</code>	Show filesystem space usage.
<code>df -i</code>	Show inode usage.
<code>du -sh /var/log</code>	Summarize disk use for a directory.
<code>du -xhd1 /var sort -h</code>	Summarize first-level disk use within one filesystem.
<code>lsblk</code>	Show block devices and mount points.
<code>lsblk -f</code>	Show block devices with filesystem metadata.
<code>blkid</code>	Show block-device UUIDs and filesystem types.
<code>findmnt</code>	Show mounted filesystems in a tree.
<code>findmnt /</code>	Show the filesystem mounted at root.
<code>mount</code>	List current mounts.
<code>mount /dev/sdb1 /mnt/data</code>	Mount a filesystem. Verify device and mount point.
<code>umount /mnt/data</code>	Unmount a filesystem.
<code>mount -o remount,ro /mnt/data</code>	Remount a filesystem read-only.
<code>fsck -N /dev/sdb1</code>	Show what fsck would run without executing it.
<code>fsck /dev/sdb1</code>	Check/repair an unmounted filesystem. Use the filesystem-specific guidance.
<code>tune2fs -l /dev/sdb1</code>	Show ext filesystem metadata.
<code>xfs_info /mountpoint</code>	Show XFS filesystem geometry where applicable.
<code>swapon --show</code>	Show active swap areas.
<code>free -h</code>	Show memory and swap usage.
<code>fdisk -l</code>	List partition tables. Modifying partitions requires extra caution.
<code>parted -l</code>	List disk partition information.
<code>smartctl -a /dev/sda</code>	Show SMART data when smartmontools is installed.
<code>nvme smart-log /dev/nvme0</code>	Show NVMe health information when nvme-cli is installed.
<code>sync</code>	Flush buffered filesystem writes to storage.
<code>lsdf +L1</code>	Find open files that have been deleted but still consume space.

9. Archives, compression, packages, and software inventory

Command	Purpose / Notes
<code>tar -tf archive.tar</code>	List files in a tar archive.
<code>tar -czf backup.tar.gz folder/</code>	Create a gzip-compressed tar archive.
<code>tar -xzf backup.tar.gz</code>	Extract a gzip-compressed tar archive.
<code>gzip file.txt</code>	Compress a file with gzip.
<code>gunzip file.txt.gz</code>	Decompress a gzip file.
<code>xz file.log</code>	Compress a file with xz.
<code>unxz file.log.xz</code>	Decompress an xz file.
<code>zip -r archive.zip folder/</code>	Create a ZIP archive when zip is installed.
<code>unzip -l archive.zip</code>	List ZIP archive contents.
<code>unzip archive.zip</code>	Extract a ZIP archive.
<code>dpkg -l</code>	List installed Debian packages.
<code>dpkg -S /usr/bin/ssh</code>	Find which Debian package owns a file.
<code>apt list --installed</code>	List packages known to APT as installed.
<code>apt update</code>	Refresh APT package metadata.
<code>apt upgrade</code>	Upgrade installed APT packages after review.
<code>apt install <package></code>	Install a package with APT.
<code>apt remove <package></code>	Remove a package with APT.
<code>rpm -qa</code>	List installed RPM packages.
<code>rpm -qf /usr/bin/ssh</code>	Find which RPM package owns a file.
<code>dnf check-update</code>	Check for available DNF updates.
<code>dnf upgrade</code>	Upgrade packages with DNF after review.
<code>dnf install <package></code>	Install a package with DNF.
<code>dnf remove <package></code>	Remove a package with DNF.
<code>snap list</code>	List installed Snap packages where Snap is used.
<code>flatpak list</code>	List installed Flatpak applications where Flatpak is used.

10. Security, hashes, certificates, firewall, and auditing

Command	Purpose / Notes
sha256sum file.iso	Calculate a SHA-256 hash.
sha512sum file.iso	Calculate a SHA-512 hash.
md5sum file.iso	Calculate an MD5 checksum for compatibility; do not use MD5 as a security proof.
openssl dgst -sha256 file.bin	Calculate a SHA-256 digest with OpenSSL.
openssl x509 -in cert.pem -text -noout	Display an X.509 certificate.
openssl s_client -connect example.com:443 -servername example.com	Inspect a remote TLS connection and certificate chain.
ssh-keygen -lf ~/.ssh/id_ed25519.pub	Display an SSH public-key fingerprint.
ssh -v user@host	Run SSH with verbose connection diagnostics.
sshd -t	Validate OpenSSH server configuration where supported.
ufw status verbose	Show UFW firewall state and rules.
ufw status numbered	Show UFW rules with numbers.
firewall-cmd --state	Check firewalld state.
firewall-cmd --list-all	Show the active firewalld zone configuration.
nft list ruleset	Show the nftables ruleset.
iptables -S	Show iptables rules in command format where iptables is used.
auditctl -s	Show Linux Audit subsystem status where auditd is installed.
auditctl -l	List current Linux Audit rules.
ausearch -m USER_LOGIN -ts today	Search audit records for login events.
aureport --auth	Summarize authentication events from audit logs.
lastb	Show failed login history when btmp is available and permissions allow.
faillock --user <username>	Show failed-authentication records on systems using faillock.
getfacl file.txt	Show POSIX ACLs on a file.
setfacl -m u:analyst:r file.txt	Add a user ACL entry to a file.
capsh --print	Show process capability context where libcap tools are installed.
getcap -r /usr/bin 2>/dev/null	Find file capabilities under a path.

Authoritative references

Use vendor and standards documentation for exact syntax, current behavior, version-specific details, licensing, and policy requirements.

Linux man-pages project

<https://www.man7.org/linux/man-pages/>

GNU Coreutils manual

<https://www.gnu.org/software/coreutils/manual/>

systemd documentation

<https://systemd.io/>

OpenSSH manuals

<https://www.openssh.com/manual.html>