

## FILE & DIRECTORY BASICS

<code>ls -la</code>	List all files with details & hidden files
<code>cd /path/to/dir</code>	Change directory
<code>pwd</code>	Print current working directory
<code>mkdir -p dir/sub</code>	Create nested directories
<code>cp -r src dest</code>	Copy files/directories recursively
<code>mv old new</code>	Move or rename files
<code>rm -rf dir</code>	Remove directory and contents (careful!)
<code>touch file.txt</code>	Create empty file or update timestamp
<code>ln -s target link</code>	Create symbolic link
<code>tree -L 2</code>	Show directory tree (2 levels deep)

## FILE VIEWING & SEARCHING

<code>cat file.txt</code>	Display entire file content
<code>less file.txt</code>	View file with scrolling (q to quit)
<code>head -n 20 file</code>	Show first 20 lines
<code>tail -f log.txt</code>	Follow file in real-time (live logs)
<code>grep -rn "text" .</code>	Search recursively with line numbers
<code>grep -i "error" log</code>	Case-insensitive search
<code>find / -name "*.log"</code>	Find files by name pattern
<code>find . -mtime -7</code>	Files modified in last 7 days
<code>wc -l file.txt</code>	Count lines in file

## PERMISSIONS & OWNERSHIP

<code>chmod 755 file</code>	<code>rxr-xr-x</code> (owner: all, others: r+x)
<code>chmod 644 file</code>	<code>rw-r--r--</code> (owner: rw, others: read)
<code>chmod +x script.sh</code>	Make file executable
<code>chown user:grp file</code>	Change file owner and group
<code>chown -R user dir/</code>	Change ownership recursively

## SYSTEM INFORMATION

<code>uname -a</code>	Full system information
<code>hostname</code>	Show system hostname
<code>uptime</code>	System uptime and load average
<code>df -h</code>	Disk space usage (human readable)
<code>du -sh dir/</code>	Directory size summary
<code>free -h</code>	Memory usage (human readable)
<code>lscpu</code>	CPU architecture information
<code>lsblk</code>	List block devices (disks)
<code>cat /etc/os-release</code>	Show Linux distribution info

## PROCESS MANAGEMENT

<code>ps aux</code>	List all running processes
<code>top</code>	Real-time process monitor (q to quit)
<code>htop</code>	Interactive process viewer
<code>kill PID</code>	Terminate process by ID
<code>kill -9 PID</code>	Force kill process
<code>killall nginx</code>	Kill all processes by name
<code>bg / fg</code>	Background / foreground a job
<code>nohup cmd &amp;</code>	Run command immune to hangups

## USER MANAGEMENT

<code>whoami</code>	Current username
<code>id</code>	Current user ID, groups
<code>sudo command</code>	Run as superuser
<code>su - user</code>	Switch to another user
<code>useradd -m user</code>	Create user with home directory
<code>passwd user</code>	Set/change user password
<code>usermod -aG grp usr</code>	Add user to group
<code>userdel -r user</code>	Delete user and home directory
<code>groups user</code>	Show user's groups

**Pro Tip:** Use `chmod -R 755` for directories and `chmod 644` for files. Never use `chmod 777` in production! Always check with `ls -la`.

## NETWORKING

<code>ip a</code>	Show all network interfaces & IPs
<code>ip r</code>	Show routing table
<code>ping -c 4 host</code>	Send 4 ICMP packets to host
<code>curl -I url</code>	Fetch HTTP headers only
<code>wget url</code>	Download file from URL
<code>ss -tulnp</code>	Show listening ports with processes
<code>dig domain.com</code>	DNS lookup
<code>traceroute host</code>	Trace packet route to host
<code>scp file usr@h:/p</code>	Secure copy to remote server
<code>ssh user@host</code>	SSH into remote server

## PACKAGE MANAGEMENT

<code>apt update</code>	Update package index (Debian/Ubuntu)
<code>apt install pkg</code>	Install a package
<code>apt remove pkg</code>	Remove a package
<code>dnf install pkg</code>	Install a package (RHEL/Alma)
<code>dnf update</code>	Update all packages
<code>dnf search keyword</code>	Search for packages

## SYSTEMD & SERVICES

<code>systemctl start svc</code>	Start a service
<code>systemctl stop svc</code>	Stop a service
<code>systemctl restart svc</code>	Restart a service
<code>systemctl status svc</code>	Check service status
<code>systemctl enable svc</code>	Enable service at boot
<code>journalctl -u svc -f</code>	Follow service logs real-time
<code>journalctl --since today</code>	Show today's logs

**Pro Tip:** Use `systemctl list-units --failed` to find failed services. Combine with `journalctl -xe` for detailed error logs.

## TEXT PROCESSING

<code>awk '{print \$1}' f</code>	Print first column
<code>sed 's/old/new/g' f</code>	Replace text globally
<code>sort file.txt</code>	Sort lines alphabetically
<code>sort -rn file.txt</code>	Sort numerically, descending
<code>uniq -c</code>	Count unique occurrences
<code>cut -d: -f1 file</code>	Extract field from delimited file
<code>tr 'a-z' 'A-Z'</code>	Translate lowercase to uppercase
<code>diff file1 file2</code>	Compare two files

## COMPRESSION & ARCHIVES

<code>tar czf a.tar.gz d/</code>	Create gzipped archive
<code>tar xzf arch.tar.gz</code>	Extract gzipped archive
<code>tar tf arch.tar.gz</code>	List archive contents
<code>zip -r arch.zip dir/</code>	Create zip archive
<code>unzip arch.zip</code>	Extract zip archive
<code>gzip file</code>	Compress file (replaces original)

## PIPES & REDIRECTION

<code>cmd &gt; file</code>	Redirect output (overwrite)
<code>cmd &gt;&gt; file</code>	Append output to file
<code>cmd 2&gt; error.log</code>	Redirect errors to file
<code>cmd &amp;&gt; all.log</code>	Redirect stdout + stderr
<code>cmd1   cmd2</code>	Pipe output of cmd1 to cmd2
<code>cmd1 &amp;&amp; cmd2</code>	Run cmd2 only if cmd1 succeeds
<code>cmd1    cmd2</code>	Run cmd2 only if cmd1 fails

### Want to master Linux?

Browse our collection of 188+ professional IT eBooks

[dargslan.com/books](https://dargslan.com/books)

Linux | Docker | Kubernetes | Python | AWS | Azure | Security