OCSALY ACADEMY

Linux Command Line Cheat Sheet

The commands I use every day on servers, labs and desktops. Keep it next to your terminal. **v1.0**

1. Files and navigation

Command	Does
`pwd` / `cd dir` / `cd ..` / `cd ~`	Where am I, move around
`ls -la`	List all files with details
`mkdir -p a/b/c`	Create nested folders
`cp -r src dst` / `mv a b`	Copy, move or rename
`rm file` / `rm -r dir`	Delete (no recycle bin!)
`ln -s target link`	Symbolic link
`cat` / `less` / `head -n 20` / `tail -f`	Read files, follow logs
`nano file` / `vim file`	Edit a file

2. Permissions

Number	Meaning	Use
`644`	`rw-r--r--`	Normal files
`755`	`rwxr-xr-x`	Dirs, scripts
`600` / `700`	`rw-------` / `rwx------`	Private keys, ~/.ssh
`777`	`rwxrwxrwx`	Almost never

```
chmod u+x script.sh        chmod -R u=rwX,go=rX dir
sudo chown user:group file  umask   (022 = files 644, dirs 755)
```

r = 4, w = 2, x = 1. On a directory, x = may enter it.

3. Users and groups

Command	Does
`whoami` / `id`	Who am I, which groups
`sudo command` / `sudo -i`	Run as root / root shell
`sudo adduser alice`	Create a user (Debian/Ubuntu)
`sudo usermod -aG docker alice`	Add to a group (log in again)
`passwd`	Change your password

4. Processes

Command	Does
`ps aux` / `top` / `htop`	What is running
`kill PID` / `kill -9 PID`	Stop politely / force
`command &` / `jobs` / `fg`	Background jobs
`systemctl status nginx`	Service status
`sudo systemctl restart nginx`	Restart a service
`journalctl -u nginx -f`	Follow a service's logs

5. Networking

Command	Does
`ip a` / `ip r`	Addresses / routes
`ping -c 4 host`	Is it reachable?
`ss -tulpn`	Listening ports and programs
`curl -I https://site`	HTTP headers
`dig example.com`	DNS lookup
`ssh user@host` / `scp f user@host:`	Remote shell / copy

6. Finding things

Command	Does
`find . -name "*.log"`	Files by name
`find / -perm -4000 -type f 2>/dev/null`	setuid programs
`grep -rn "text" dir`	Text inside files
`grep -i error app.log | wc -l`	Count matching lines
`which python3`	Where a command lives

OCSALY ACADEMY

Linux Command Line Cheat Sheet

Page 2: packages, disks, archives, redirection, bash scripting and a practice plan. **v1.0**

7. Packages

Debian / Ubuntu	Fedora / RHEL
`sudo apt update`	`sudo dnf check-update`
`sudo apt upgrade`	`sudo dnf upgrade`
`sudo apt install nmap`	`sudo dnf install nmap`
`sudo apt remove nmap`	`sudo dnf remove nmap`
`apt search word`	`dnf search word`

8. Disks and archives

Command	Does
`df -h`	Free space per disk
`du -sh *`	Size of each item here
`lsblk`	Disks and partitions
`tar -czf backup.tar.gz dir`	Create a compressed archive
`tar -xzf backup.tar.gz`	Extract it
`zip -r a.zip dir` / `unzip a.zip`	Zip files

9. Redirection and pipes

Syntax	Does
`cmd > file` / `cmd >> file`	Write / append output
`cmd 2> err.log`	Write errors only
`cmd > all.log 2>&1`	Output and errors together
`cmd1 | cmd2`	Pipe output into the next command
`cmd1 && cmd2` / `cmd1 || cmd2`	Run next if OK / if failed
`sort | uniq -c | sort -nr`	Count and rank lines

10. Bash scripting basics

```
#!/usr/bin/env bash
set -euo pipefail    # stop on error
name="world"         # no spaces
echo "Hello, $name"
if [[ -f /etc/os-release ]]; then
  echo "file exists"
fi
for host in 10.0.0.{1..5}; do
  if ping -c1 -W1 "$host" >/dev/null; then
    echo "$host is up"
  fi
done
echo "exit code: $?"
```

11. Keyboard shortcuts

Keys	Does
`Tab`	Complete names (press twice to list)
`Ctrl+C` / `Ctrl+Z`	Stop / pause the running command
`Ctrl+R`	Search your command history
`Ctrl+A` / `Ctrl+E`	Start / end of line
`!!` / `sudo !!`	Repeat last command (as root)

12. Your 5-day practice plan

1. **Day 1:** `ls -la ~` and explain 5 permission strings.
2. **Day 2:** make a script, `chmod u+x` it, run it, then remove x and read the error.
3. **Day 3:** list setuid programs (section 6) and look up each one.
4. **Day 4:** `ps aux`, `top`, `ss -tulpn`: what runs, what listens?
5. **Day 5:** a 10-line script that backs up a folder into a dated .tar.gz.

Want the command line to do the work for you?
Variables, loops, functions and exit codes, then a network monitor and a network scanner in pure bash.

ocsaly.com/courses/shell-bash