Linux Cheat Sheet

> *Based on the Linux Journey curriculum — `linuxjourney.com`*

Grasshopper (Beginner)

Command Line Basics

Command	Description	Example
`pwd`	Print working directory	`pwd`
`ls`	List directory contents	`ls -la`
`cd`	Change directory	`cd /var/log`
`mkdir`	Create directory	`mkdir -p project/src`
`touch`	Create empty file	`touch notes.txt`
`cp`	Copy files/dirs	`cp -r src/ dest/`
`mv`	Move/rename files	`mv old.txt new.txt`
`rm`	Remove files/dirs	`rm -rf unwanted/`
`cat`	Display file contents	`cat /etc/os-release`
`less`	Scrollable file view	`less large.log`
`head` / `tail`	View start/end of file	`tail -f syslog`
`man`	Manual pages	`man ls`
`which`	Find command location	`which python3`
`history`	Command history	`history | grep sudo`

Navigation shortcuts: - `cd ..` — Parent directory - `cd ~` or `cd` — Home directory - `cd -` — Previous directory - `Tab` — Auto-complete

Wildcards: - `*` — Match any characters - `?` — Match single character - `[]` — Match character set: `[aeiou]`

Working with the File System

Command	Description	Example
`find`	Search files	`find / -name "*.conf"`
`locate`	Database file search	`locate sshd_config`
`stat`	File/directory info	`stat document.txt`
`ln -s`	Create symlink	`ln -s /opt/app link`
`du`	Disk usage	`du -sh /var/log`
`df`	Filesystem space	`df -h`
`tree`	Directory tree view	`tree -L 2 project/`

Key directories:

```
/          — Root
/bin       — Essential binaries
/sbin      — System binaries
/etc       — Configuration files
/home      — User home directories
/var       — Variable data (logs, mail)
/tmp       — Temporary files (wiped on reboot)
/usr       — User programs & data
/opt       — Optional/additional software
```

Text Manipulation

Command	Description	Example
`grep`	Search text patterns	`grep -i "error" syslog`
`grep -v`	Invert match	`grep -v "^#" file`
`grep -r`	Recursive search	`grep -r "TODO" ./`
`sort`	Sort lines	`sort -n numbers.txt`
`uniq`	Remove duplicates	`sort file | uniq -c`
`wc`	Word/line/char count	`wc -l file.txt`
`cut`	Extract columns	`cut -d: -f1 /etc/passwd`
`tr`	Translate characters	`tr '[:lower:]' '[:upper:]'`
`sed`	Stream editor	`sed 's/old/new/g' file`
`awk`	Text processing	`awk '{print $1}' file`
`diff`	Compare files	`diff file1 file2`
`paste`	Merge file lines	`paste -d, col1 col2`

Grep flags: `-i` (ignore case), `-n` (line numbers), `-c` (count), `-r` (recursive), `-v` (invert)

User Management

Command	Description	Example
`whoami`	Current user	`whoami`
`id`	User/group IDs	`id username`
`adduser`	Create user	`adduser newuser`
`useradd`	Low-level user creation	`useradd -m -s /bin/bash user`
`userdel`	Delete user	`userdel -r username`
`passwd`	Change password	`sudo passwd username`
`su`	Switch user	`su - root`
`sudo`	Run as superuser	`sudo apt update`
`groups`	Show user groups	`groups username`
`usermod`	Modify user	`usermod -aG sudo user`
`chage`	Password expiry	`chage -l username`

Key files:

```
/etc/passwd    — User accounts
/etc/shadow    — Password hashes (root only)
/etc/group     — Group definitions
/etc/sudoers   — Sudo permissions
```

Permissions

Command	Description	Example
`chmod`	Change permissions	`chmod 755 script.sh`
`chown`	Change owner	`sudo chown user:group file`
`chgrp`	Change group	`chgrp staff file`
`umask`	Default permissions mask	`umask 022`

Permission codes:

```
r = 4  w = 2  x = 1

chmod 755 = rwxr-xr-x  (owner full, group/others read+execute)
chmod 644 = rw-r--r--  (owner read/write, others read)
chmod 600 = rw-------  (owner only)
```

Symbolic notation:

```
chmod u+x file    — Add execute for user
chmod go-w file   — Remove write for group/others
chmod a+r file    — Add read for all
```

Packages

Command	Description	Debian/Ubuntu	RHEL/CentOS	Arch
Install package		`apt install pkg`	`dnf install pkg`	`pacman -S pkg`
Remove package		`apt remove pkg`	`dnf remove pkg`	`pacman -R pkg`
Update index		`apt update`	`dnf check-update`	`pacman -Sy`
Upgrade all		`apt upgrade`	`dnf upgrade`	`pacman -Su`
Search packages		`apt search term`	`dnf search term`	`pacman -Ss term`
List installed		`dpkg -l`	`rpm -qa`	`pacman -Qe`
Get info		`apt show pkg`	`dnf info pkg`	`pacman -Si pkg`

Journeyman (Intermediate)

Processes

Command	Description	Example
`ps`	Process status	`ps aux`
`top`	Interactive process monitor	`top`
`htop`	Enhanced top	`htop`
`kill`	Terminate process	`kill -9 PID`
`killall`	Kill by name	`killall nginx`
`pkill`	Kill by pattern	`pkill -f script.py`
`nice`	Set priority	`nice -n 10 command`
`renice`	Change priority	`renice 5 PID`
`fg` / `bg`	Foreground/background	`fg 1`
`jobs`	List background jobs	`jobs -l`
`&`	Run in background	`command &`
`nohup`	Survive logout	`nohup command &`

Command	Description	Example
`wait`	Wait for jobs	`wait`
`pgrep`	Find process PID	`pgrep python`
`pstree`	Process tree	`pstree`

Signal numbers:

```
1  SIGHUP   — Hangup
2  SIGINT   — Interrupt (Ctrl+C)
9  SIGKILL  — Force kill (cannot be caught)
15 SIGTERM  — Graceful termination (default)
18 SIGCONT  — Continue stopped process
19 SIGSTOP  — Stop process (Ctrl+Z)
```

Process Utilization

Command	Description	Example
`uptime`	System uptime & load	`uptime`
`who`	Logged-in users	`who`
`free`	Memory usage	`free -h`
`vmstat`	Virtual memory stats	`vmstat 1 5`
`iostat`	Disk I/O stats	`iostat -x 1`
`mpstat`	CPU stats per core	`mpstat -P ALL 1`
`lsof`	List open files	`lsof -i :80`
`lscpu`	CPU information	`lscpu`
`uname`	System info	`uname -a`

Load averages: `1min, 5min, 15min — above 1.0 per core means overloaded.`

Advanced Text Fu

Regular expressions (grep -E):

```
.       — Any single character
^       — Start of line
$       — End of line
*       — Zero or more
+       — One or more
?       — Zero or one
|       — OR
[]      — Character class: [a-z0-9]
[^]     — Negated class: [^0-9]
\w      — Word character
\W      — Non-word
\d      — Digit
\s      — Whitespace
{n,m}   — Quantifier: exactly n to m
```

Sed examples:

```
sed 's/old/new/g' file           # Global replace
sed -i 's/old/new/g' file        # In-place edit
sed '3d' file                    # Delete line 3
sed '1,5d' file                  # Delete lines 1-5
sed -n '2p' file                 # Print line 2
sed '/pattern/d' file            # Delete lines matching pattern
```

Awk examples:

```
awk '{print $1}' file            # Print first column
awk -F: '{print $1}' /etc/passwd # Custom delimiter
awk '$3 > 1000 {print $1}' file  # Conditional
awk 'BEGIN{x=0} {x+=$1} END{print x}' file  # Sum column
```

Devices

Command	Description	Example
`lsblk`	List block devices	`lsblk`
`fdisk -l`	Disk partitions	`sudo fdisk -l`
`parted`	Disk partitioner	`sudo parted /dev/sda`
`dmesg`	Kernel messages	`dmesg | tail`
`lspci`	PCI devices	`lspci`
`lsusb`	USB devices	`lsusb`
`lscpu`	CPU info	`lscpu`
`hwinfo`	Hardware info	`hwinfo`

Device naming:

```
/dev/sda1  — SATA disk 1, partition 1
/dev/nvme0n1p1  — NVMe disk 1, partition 1
/dev/loop0 — Loop device (for mountable images)
/dev/cdrom — Optical drive
```

Filesystem

Command	Description	Example
`mount`	Mount filesystem	`sudo mount /dev/sdb1 /mnt`
`umount`	Unmount	`umount /mnt`
`mkfs.ext4`	Create filesystem	`sudo mkfs.ext4 /dev/sdb1`
`blkid`	Block device info	`blkid`
`lsblk`	Block devices	`lsblk -f`
`fuser`	Find process using FS	`fuser -mv /mnt`
`fdisk`	Partition editor	`sudo fdisk /dev/sda`

fstab entry:

```
UUID=xxxx-xxxx  /mnt/data  ext4  defaults  0  2
```

Filesystem hierarchy:

```
/           Root
├── bin/    Essential user commands
├── sbin/   Essential system commands
├── etc/    Configuration
├── var/    Variable data
│   └── log/  System logs
├── home/   User directories
├── root/   Root home
├── usr/    User programs
│   ├── bin/    Non-essential binaries
│   ├── lib/    Libraries
│   └── share/  Shared data
├── tmp/    Temporary files
├── dev/    Device files
├── proc/   Process info (virtual)
└── sys/    Kernel/hardware info (virtual)
```

Boot Process

Stages: 1. **BIOS/UEFI** — Hardware initialization, finds bootloader 2. **GRUB** — Loads kernel and initramfs 3. **Kernel** — Mounts root filesystem, starts init 4. **Init (systemd)** — Starts services, reaches target (multi-user/graphical)

GRUB:

```
sudo update-grub          # Regenerate GRUB config
# /etc/default/grub       — GRUB configuration
# GRUB_TIMEOUT=5          — Boot menu timeout
```

Kernel modules:

```
lsmod                     # List loaded modules
sudo modprobe module      # Load module
sudo rmmod module         # Remove module
# /etc/modprobe.d/        — Module configuration
```

Init (systemd)

Command	Description	Example
`systemctl status`	Service status	`systemctl status nginx`
`systemctl start`	Start service	`sudo systemctl start ssh`
`systemctl stop`	Stop service	`sudo systemctl stop ssh`
`systemctl restart`	Restart service	`sudo systemctl restart ssh`
`systemctl reload`	Reload config	`sudo systemctl reload nginx`
`systemctl enable`	Enable at boot	`sudo systemctl enable ssh`
`systemctl disable`	Disable at boot	`sudo systemctl disable ssh`
`systemctl is-active`	Check running	`systemctl is-active ssh`
`systemctl list-units`	List all services	`systemctl list-units --failed`

System targets (runlevels):

```
0   — poweroff
1   — rescue (single-user)
2   — multi-user (no GUI)
3   — multi-user (no GUI)
4   — unused
5   — graphical (GUI)
6   — reboot

sudo systemctl get-default      # Current target
sudo systemctl set-default multi-user.target  # Change default
```

Logging

Command	Description	Example
`journalctl`	systemd logs	`journalctl -xe`
`journalctl -f`	Follow logs	`journalctl -fu ssh`
`journalctl -u`	Service logs	`journalctl -u nginx`
`journalctl -p`	Priority filter	`journalctl -p err`
`dmesg`	Kernel logs	`dmesg | tail -20`
`last`	Login history	`last`
`lastlog`	Last login per user	`lastlog`

Priority levels: `emerg`, `alert`, `crit`, `err`, `warning`, `notice`, `info`, `debug`

Common log files:

```
/var/log/syslog         — General system log
/var/log/auth.log       — Authentication
/var/log/kern.log       — Kernel
/var/log/dpkg.log       — Package manager
/var/log/apt/           — APT history
```

SSH

Command	Description	Example
`ssh`	Connect to server	`ssh user@server`
`ssh -p`	Custom port	`ssh -p 2222 user@host`
`scp`	Secure copy	`scp file.txt user@host:/tmp/`
`scp -r`	Copy directory	`scp -r dir/ user@host:/path/`
`rsync`	Sync files	`rsync -avz dir/ user@host:/path/`
`ssh-keygen`	Generate keys	`ssh-keygen -t ed25519`
`ssh-copy-id`	Install public key	`ssh-copy-id user@host`

Key files:

```
~/.ssh/id_ed25519       — Private key (NEVER share)
~/.ssh/id_ed25519.pub   — Public key (safe to share)
~/.ssh/known_hosts      — Known servers
```

SSH config (`~/.ssh/config`):

```
Host myserver
    HostName 192.168.1.100
    User deploy
    Port 2222
    IdentityFile ~/.ssh/id_ed25519
```

Bash Scripting

Shebang & basics:

```
#!/bin/bash
# Comment
VARIABLE="value"
echo $VARIABLE
echo ${VARIABLE}
```

Variables:

```
NAME="World"
echo "Hello, $NAME"          # String interpolation
echo $#                      # Number of arguments
echo $1, $2, $3              # Positional arguments
echo $?                      # Last command exit code
echo $$                      # Current PID
echo $!                      # Last background PID
echo ${#string}              # String length
echo ${string:0:5}           # Substring
```

Conditionals:

```
if [ -f "file.txt" ]; then
    echo "File exists"
elif [ -d "dir" ]; then
    echo "Directory exists"
else
    echo "Nothing found"
fi

# String comparison
if [ "$a" = "$b" ]; then
if [ "$a" != "$b" ]; then

# Number comparison
if [ $a -gt $b ]; then      # Greater than
if [ $a -lt $b ]; then      # Less than
if [ $a -eq $b ]; then      # Equal
```

Test operators:

```
[ -f file ]   — File exists
[ -d dir ]    — Directory exists
[ -r file ]   — Readable
[ -w file ]   — Writable
[ -x file ]   — Executable
[ -s file ]   — File not empty
[ -z string ] — String is empty
[ -n string ] — String is not empty
```

Loops:

```
# For loop
for i in 1 2 3; do
    echo $i
done

for file in *.txt; do
    echo "Processing $file"
done

# While loop
counter=1
while [ $counter -le 5 ]; do
    echo $counter
    counter=$((counter + 1))
done

# Until loop (runs until condition is true)
until [ $counter -eq 10 ]; do
    counter=$((counter + 1))
done
```

Functions:

```
greet() {
    echo "Hello, $1!"
}
greet "World"
```

Reading input:

```
read -p "Enter name: " name
echo "Got: $name"
```

Cron Jobs

Crontab format:

```
┌──────────────── minute (0-59)
│ ┌────────────── hour (0-23)
│ │ ┌──────────── day of month (1-31)
│ │ │ ┌────────── month (1-12)
│ │ │ │ ┌──────── day of week (0-7, Sun=0,7)
│ │ │ │ │
* * * * *  command_to_execute
```

Examples:

```
*/5 * * * *    — Every 5 minutes
0 * * * *      — Every hour at :00
0 9 * * *      — Every day at 9 AM
30 8 * * 1-5   — Weekdays at 8:30 AM
0 0 1 * *      — First of every month
0 0 * * 0      — Every Sunday at midnight
@reboot         — Run at boot
@daily          — Every day at midnight
@monthly        — First day of month
@yearly         — January 1st
```

Management:

```
crontab -e      — Edit crontab
crontab -l      — List crontab
crontab -r      — Remove crontab
```

Firewall Basics (UFW)

Command	Description
`sudo ufw status`	Check status
`sudo ufw enable`	Enable firewall
`sudo ufw disable`	Disable firewall
`sudo ufw default deny`	Deny incoming by default
`sudo ufw allow 22/tcp`	Allow SSH
`sudo ufw allow 80,443/tcp`	Allow HTTP/HTTPS
`sudo ufw allow from 10.0.0.0/8`	Allow subnet
`sudo ufw delete allow 80`	Remove rule
`sudo ufw reset`	Reset all rules

Nginx profile: `sudo ufw allow 'Nginx Full'` (allows 80,443)

Networking Nomad

Network Configuration

Command	Description	Example
`ip addr`	IP addresses	`ip addr show`
`ip link`	Network interfaces	`ip link set eth0 up`
`ip route`	Routing table	`ip route show`
`ifconfig`	Legacy interface config	`ifconfig eth0`
`hostname`	Set/view hostname	`hostname myserver`
`hostnamectl`	System hostname	`hostnamectl set-name newname`

DHCP: `sudo dhclient eth0` **Static IP (Netplan, `/etc/netplan/`):**

```
network:
  version: 2
  ethernets:
    eth0:
      addresses: [192.168.1.100/24]
      gates: [192.168.1.1]
      nameservers:
        addresses: [8.8.8.8, 1.1.1.1]
```

Network Troubleshooting

Command	Description	Example
`ping`	Test connectivity	`ping google.com`
`ping -c 4`	Limited pings	`ping -c 4 192.168.1.1`
`traceroute`	Route to host	`traceroute google.com`
`nslookup`	DNS lookup	`nslookup google.com`
`dig`	DNS query	`dig google.com ANY`
`netstat`	Network connections	`netstat -tulnp`
`ss`	Socket stats (modern)	`ss -tulnp`
`tcpdump`	Packet capture	`sudo tcpdump -i eth0`
`curl`	HTTP requests	`curl -I https://example.com`
`wget`	Download files	`wget https://example.com/file`

Command	Description	Example
`arp`	ARP table	`arp -a`
`iwconfig`	Wireless config	`iwconfig`
`nmcli`	NetworkManager CLI	`nmcli dev wifi`

Check listening ports:

```
ss -tulnp                    # Modern
netstat -tulnp               # Legacy
sudo lsof -i :80             # What's using port 80
```

DNS Configuration

Files:

```
/etc/resolv.conf      — DNS resolver config
/etc/hosts            — Static host mappings
/etc/nsswitch.conf    — Name resolution order
```

/etc/hosts example:

```
127.0.0.1   localhost
192.168.1.50  myserver.local
```

DNS lookup tools:

```
dig google.com A              # A record
dig google.com MX             # Mail servers
dig -x 8.8.8.8               # Reverse lookup
host google.com               # Simple lookup
```

Network Sharing

rsync:

```
rsync -avz /source/ user@host:/dest/          # Remote sync
rsync -avz --delete /source/ /dest/           # Mirror (delete extras)
rsync -avz --exclude='*.tmp' /source/ /dest/  # Exclude files
rsync -avn /source/ /dest/                    # Dry run (no changes)
```

NFS (server):

```
# /etc/exports
/var/share  192.168.1.0/24(rw,sync,no_subtree_check)
sudo exportfs -ra
```

NFS (client):

```
sudo mount -t nfs server:/var/share /mnt/nfs
```

Subnetting Reference

CIDR	Netmask	Hosts	Example
/24	255.255.255.0	254	192.168.1.0/24
/23	255.255.254.0	510	192.168.0.0/23

CIDR	Netmask	Hosts	Example
/22	255.255.252.0	1022	192.168.0.0/22
/16	255.255.0.0	65,534	192.168.0.0/16
/8	255.0.0.0	16M	10.0.0.0/8

Private IP ranges:

```
10.0.0.0/8      — 10.0.0.0 – 10.255.255.255
172.16.0.0/12   — 172.16.0.0 – 172.31.255.255
192.168.0.0/16  — 192.168.0.0 – 192.168.255.255
```

Road Warrior (Advanced)

Docker

Command	Description	Example
`docker run`	Run container	`docker run -d nginx`
`docker ps`	List containers	`docker ps -a`
`docker stop`	Stop container	`docker stop name`
`docker rm`	Remove container	`docker rm -f name`
`docker images`	List images	`docker images`
`docker rmi`	Remove image	`docker rmi image:tag`
`docker logs`	View logs	`docker logs container`
`docker exec`	Execute in container	`docker exec -it name bash`
`docker build`	Build image	`docker build -t app .`
`docker-compose up`	Start services	`docker-compose up -d`

Common flags: `-d` (detached), `-p` (port mapping), `-v` (volume), `--name`

---

tmux

Command	Description
`tmux`	New session
`tmux new -s name`	Named session
`tmux ls`	List sessions
`tmux attach -t name`	Attach to session
`Ctrl+b, %`	Split vertical
`Ctrl+b, "`	Split horizontal
`Ctrl+b, arrow`	Navigate panes
`Ctrl+b, z`	Zoom pane
`Ctrl+b, d`	Detach

Git

Command	Description
`git init`	Initialize repo
`git clone url`	Clone repo
`git status`	Working tree status
`git add .`	Stage changes

Command	Description
`git commit -m "msg"`	Commit
`git push`	Push to remote
`git pull`	Pull from remote
`git branch`	List branches
`git checkout -b name`	Create & switch branch
`git merge branch`	Merge branch
`git log --oneline`	Compact log
`git diff`	Show changes
`git reset HEAD file`	Unstage file
`git stash`	Stash changes
`git tag -a v1.0`	Create tag

Disk Management

Command	Description	Example
`lsblk`	Block devices	`lsblk -f`
`fdisk`	Partition tool	`sudo fdisk /dev/sda`
`parted`	Partition tool	`sudo parted /dev/sda`
`mkfs.ext4`	Create ext4	`sudo mkfs.ext4 /dev/sdb1`
`blkid`	Block device IDs	`blkid`
`tune2fs`	Ext filesystem tune	`tune2fs -l /dev/sda1`
`e2fsck`	Filesystem check	`sudo e2fsck /dev/sda1`
`lvm`	LVM management	`sudo lvs` , `sudo vgs`
`df -h`	Disk usage	`df -h`
`du -sh *`	Directory sizes	`du -sh /var/*`

LVM basics:

```
sudo pvcreate /dev/sdb          # Physical volume
sudo vgcreate vg0 /dev/sdb      # Volume group
sudo lvcreate -L 50G -n lv0 vg0 # Logical volume
sudo mkfs.ext4 /dev/vg0/lv0     # Format
```

Swap:

```
sudo fallocate -l 4G /swapfile
sudo chmod 600 /swapfile
sudo mkswap /swapfile
sudo swapon /swapfile
```

---

curl & wget

curl:

```
curl URL                          # Download to stdout
curl -o file.zip URL              # Save to file
curl -O URL                       # Save with original name
curl -I URL                       # Headers only
curl -X POST -d "data" URL        # POST request
curl -H "Authorization: Bearer token" URL  # Auth header
curl -s URL                       # Silent (no progress)
curl -L URL                       # Follow redirects
```

wget:

```
wget URL                          # Download file
wget -r URL                       # Recursive download
wget -c URL                       # Resume interrupted
wget -P dir/ URL                  # Save to directory
```

Nginx (Quick Reference)

Command	Description
`sudo nginx -t`	Test config
`sudo systemctl restart nginx`	Restart
`sudo systemctl reload nginx`	Reload config

Basic config (`/etc/nginx/sites-available/default`):

```
server {
    listen 80;
    server_name example.com;

    location / {
        root /var/www/html;
        index index.html;
    }

    # Reverse proxy
    location /api/ {
        proxy_pass http://localhost:3000/;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}
```

---

fail2ban

Command	Description
`sudo systemctl status fail2ban`	Check status
`sudo fail2ban-client status`	List jails
`sudo fail2ban-client status ssh`	SSH jail status
`sudo fail2ban-client set ssh banip IP`	Ban IP
`sudo fail2ban-client set ssh unbanip IP`	Unban IP

Config: `/etc/fail2ban/jail.local`

Quick Reference

Exit Codes

```
0    — Success
1    — General error
2    — Misuse of shell builtins
126  — Command invoked cannot execute
127  — Command not found
128  — Invalid exit argument
130  — Script terminated by Ctrl+C
137  — Process killed (SIGKILL)
```

Pipes & Redirection

```
command > file          — Overwrite file
command >> file         — Append to file
command < file          — Read from file
command1 | command2     — Pipe output
2> error.log            — Redirect errors
&> file                 — Redirect all (stdout+stderr)
command1 | tee file     — Pipe AND save
```

Job Control

```
Ctrl+C    — Kill foreground process
Ctrl+Z    — Suspend foreground process
Ctrl+D    — End input / logout
fg        — Resume in foreground
bg        — Resume in background
jobs      — List background jobs
```

Useful One-Liners

```
# Find large files
find / -type f -size +100M -exec ls -lh {} \;

# Kill process on port
sudo fuser -k 80/tcp

# Archive and compress
tar -czvf archive.tar.gz /path/

# Extract archive
tar -xzvf archive.tar.gz

# Find and replace in files
find . -name "*.txt" -exec sed -i 's/old/new/g' {} \;

# Count occurrences
grep -c "pattern" file

# Monitor log in real-time
tail -f /var/log/syslog

# Disk usage by directory
du -sh /* | sort -rh | head -20
```

Generated from Linux Journey curriculum — linuxjourney.com / linuxjourney.org