**LINUX**

1. **ls -ltr** : Display the most recently modified files

-l: long listing format (permissions, owner, size, etc.)

-t: sort by modification time, newest last

-r: reverse the order (so oldest files come first)

If you want to list only files with a specific extension, you can use shell globbing (\*.ext):

**🔸 Example:**

List only .log files sorted by time (oldest first):

ls -ltr \*.log

For .txt files:

ls -ltr \*.txt

For any other extension (e.g., .sh, .conf, .json):

ls -ltr \*.sh

ls -ltr \*.conf

ls -ltr \*.json

1. **mkdir -p d1/d2/d3**  : create a path directory

-p — parents: create intermediate directories as needed without error

Example:

mkdir -p /home/user/logs

mkdir -p /var/www/myapp/logs/errors

1. **chmod 755 filename\_or\_directory**:remove write permissions to all others in the system expect owner

Read :4

write :3

execute:1

1. **chmod -R D1**: change permissions to directory and all sub-files files

chmod — change file/directory permissions

-R — recursive: applies to all files and sub directories inside the given directory

755 — permission setting (explained below)

D1 — the name of the directory you're changing (e.g., D1)

1. **chown -R <owner>:<group> <directory>** : change owner ship of a folder and all its subdirectory and file D1

chown — change ownership

-R — recursive: applies to all files and sub-folders inside the directory

<owner> — username of the new owner

<group> — group name (optional, but common to include)

Example:

chown -R ubuntu:ubuntu /var/www/html

1. **ln -s <target> <link\_name>** :create a softlink to adirectory

ln — create links

-s — symbolic (soft) link

<target> — the original file or directory

<link\_name> — the name/path of the symlink

Eamples:

ln -s /var/log/nginx/access.log nginx-access.log

1. **mv d1 d2** : rename a file d1 to d2
2. cp -pr: copy a directory and its sub directories files into another directory with permissions and timestamps

cp — copy files and directories

-p — preserve file attributes such as:

ownership

permissions

timestamps (modification time)

-r — recursive: copy directories and their contents recursively (all sub-folders and files)

Examples:

cp -pr /source/directory /destination/directory

1. **rm -rf** : remove directory even if the the owner does no have write permission

rm — remove/delete files or directories

-r or -R — recursive: delete directories and their contents, including subdirectory

-f — force: ignore nonexistent files, never prompt for confirmation

**Example:**

rm -rf /path/to/directory

1. **head -3 f1** :display the first three lines of file

head — displays the beginning of a file (by default, first 10 lines)

-3 — option to show only the first 3 lines

f1 — the filename to read

1. **tail -1** : display last line of a file

Example:

tail -1 filename

1. **tail -f F1** : continuosly monitor a running log file f1

tail — shows the end of a file

-f — "follow" mode: keeps the command running and displays new lines as they are added to the file in real-time

F1 — the filename you want to watch

Example:

tail -f /var/log/syslog

1. **find /home/ncodeit -name \*\_xml**: find all the xmal files in /home/ncodeit directory and its subdirectory

/home/ncodeid — directory to start searching from (recursively)

-name '\*\_xml' — search for files whose names end with \_xml (quotes prevent shell expansion)

14**)find /home/ncodeit -sixe +10M** : find all the files larger than 10mb in a directory

find — the command to search files

/home/ncodeit — the directory to start search

-size +10M — files larger than 10 megabytes (M for megabytes, uppercase)

No closing parenthesis needed unless part of a bigger expression

1. **find /home/ncodeit -mtime +7** : find all the files in /home/ncodeit older than 7 days
2. **find /home/ncodeit -mtime -5 :** find all the files newer than 5days
3. **grep kw f1** : find the line in a file with a perticular keyword

grep searches for text patterns inside files

kw is the pattern (a string or keyword) you want to find

f1 is the file where you want to search

1. **grep -i kw f1**:find the keyword in a file F! ignoring the CASE
2. **grep -v kw f1 : f**ind all the lines in the file f1 that do not have the keyword KW
3. **grep -r kw /path/to/dir**:find all the key words in directory and sub directory
4. **tar cvf D1.tar**: create a backup of a directory D!

tar — tool to create or extract archives (tarballs)

c — create a new archive

v — verbose (shows progress)

f — specify the filename of the archive (D1.tar)

1. **tar zcvf D1.tar D1** :

z — compress using gzip

c — create archive

v — verbose output

f — specify filename

D1 — the directory or files you want to archive

1. **gzip F1 / gunzip F1.zip** : zip a file f1 and unsip a file f1.zip
2. **gunzip F1.gz:** to decompress a gzip file
3. **ps -ef** : list all the prossess running in the system

ps — process status command

-e — show all processes running on the system (not just your own)

-f — full-format listing showing detailed info

Example :

ps -ef | grep apache

1. **kill -9 PID** : terminate a process

kill — command to send signals to processes

-9 — signal number 9, which is SIGKILL

This signal immediately stops the process without allowing it to clean up.

PID — the Process ID of the process you want to kill

1. **kill -3 PID** : take a thread dump

Example:

kill -3 1234

Sends SIGQUIT to process 1234.

The process will terminate and create a core dump (if enabled

1. **kill -1** : kill the process inculing the current terminal
2. **top:** find the CPU utilization
3. **top -b -n 1 > top-out.txt :** capture the output of top cammand in file

top — Displays real-time system resource usage (CPU, memory, etc.).

-b — Batch mode (useful for scripting or logging; no interactive display).

-n 1 — Run 1 iteration and then exit.

1. **sar 2 3** : find the system analysis report ( take report times after every 2 seconds)

sar (System Activity Reporter)

1. **free -m** : find the free memory on the sysytem .displays memory usage on your system in megabytes (MB).
2. **ulimit -a**:displays all current user-level resource limits for your shell session
3. **uname -a** : find the type of operating system
4. **id** :identity of current user
5. **cat /etc/passwd** : find all the users in the system
6. **df -k** : find how many file systems are there in the system and how much space is free
7. du -sk\* : find the sizeof directories

38)du -sk\* | sort -n : find the size of directories and sort them in increasing order

du — disk usage command

-s — summarize: only show total size per item (not contents of subdirectory)

-k — show size in kilobytes (KB)

\* — wildcard: applies to all files and directories in the current folder

| — pipe: passes the output of du to sort

sort -n — sorts output numerically

1. **if config -a** :ip address of a sytem

ifconfig = Interface Configuration tool (used to view or configure network interfaces)

-a = Show all interfaces, including those that are down (inactive)

1. **netstat -n | grep 8080** : find is given port is in use or not

netstat — Displays network connections, routing tables, etc.

-n — Shows numeric addresses and ports (no DNS/host name resolution)

| grep 8080 — Filters the output to show only lines that mention port 8080

1. **lsof -l tcp:9090** : find which processor is using this port

lsof = List Open Files (shows files and network sockets opened by processes)

1. **nslookup <ip> or nslookup <host-name>** : find the website name or hostname of a diven ip
2. **traceroute <website-url>** : find the number of routes a request is passing to reach a website
3. **nc -z IP port** : find if a specific port on a specific IP is listening for requests or not

nc = Netcat, a versatile networking tool

-z = Zero I/O mode (just scan for listening daemons, no data sent)

IP = Target IP address or hostname

port = Port number (or port range) to check

1. **ping <ip.of-Remote-system>:** check the connectivity between two systems
2. **wget url** :download a zip file or a webpage at unix prompt
3. **cat /proc/cpu info** : how to find the no of cps in a systems
4. **setup the cammand in .bash\_profile file** :every-time you login, you want to execute date command how to do that
5. ssh :open a session in remote system
6. **xargs**: take the output of previous cammand and send it as inputs to next cammand

example :

echo "file1 file2 file3" | xargs rm

1. **ls -a**:display a hindden file

ls = list directory contents

-a = show all files, including those starting with a dot . (hidden files)

1. **mv F1 .F2** :move a hidden file F1 from file f2
2. Touch F1: create a zero byte file f1
3. **telnet Ip port** : it is used to connect another system just like ssh.but its not available now a days
4. **/etc/hosts** : ip & host name mapping , where is it defined in the system
5. **jps:** find all the java process in the system

**Dynamic / Real-Time Monitoring:**

**top**

Displays real-time running processes and resource usage (CPU, memory, etc.) in a terminal. Useful for monitoring system health interactively.

**htop**

An enhanced version of top with color output, mouse support, and a more user-friendly interface. Shows process trees and makes it easier to manage processes.

**watch <cmd>**

Runs the specified command repeatedly at fixed intervals (default 2 seconds) and shows updated output on the screen, useful for watching changes dynamically.

**vmstat 1**

Reports virtual memory, CPU usage, I/O, and system stats every 1 second, helping track system performance over time.

**iostat 1**

Monitors disk I/O and CPU usage every second, useful to identify bottlenecks in storage performance.

**mpstat 1**

Displays CPU usage statistics per CPU/core every 1 second, helpful for checking load distribution.

**nload**

Shows real-time incoming and outgoing network traffic separately, with graphical representation in the terminal.

**iotop**

Monitors disk I/O usage per process in real-time, ideal to find out which processes are heavy on disk operations.

**dstat**

A versatile tool that combines vmstat, iostat, netstat, and others to monitor system resources all at once.

**tail -f /var/log/syslog**

Displays live appending of the system log, useful for monitoring log messages as they happen.

**journalctl -f**

Similar to tail -f but for systems using systemd, showing real-time logs.

**ip monitor**

Watches and outputs changes to network interfaces in real-time, helpful to track IP address changes or link status.

**2. Troubleshooting:**

**dmesg**

Prints kernel ring buffer messages, typically boot and hardware info, useful for diagnosing hardware or boot problems.

**ping <host>**

Checks network connectivity by sending ICMP echo requests to a host, measuring round-trip time.

**traceroute <host>**

Traces the path packets take to reach a destination, showing each hop along the route.

**netstat -na**

Shows all active network connections and listening ports with numeric IPs and ports.

**ss -tulnp**

Modern alternative to netstat, lists TCP/UDP sockets and associated processes.

**lsof -i**

Lists all open internet sockets (TCP/UDP connections).

**lsof +D /path**

Lists all open files under a specific directory, useful for file or resource locks.

**strace <command>**

Traces system calls and signals of a command, invaluable for debugging program behavior.

**tcpdump**

Captures and analyzes network packets, essential for deep network troubleshooting.

**dig**

DNS lookup utility to query DNS servers for records and troubleshoot DNS issues.

**df -h**

Shows disk space usage on all mounted filesystems in human-readable form (GB, MB).

**\*\*du -sh \*\*\***

Summarizes disk usage of each directory/file in current folder, human-readable.

**free -m**

Displays memory usage (RAM and swap) in megabytes.

**uptime**

Shows system uptime and load averages (1, 5, and 15 minutes).

**ps aux**

Lists all running processes with detailed info.

**kill <pid>**

Sends a signal (default TERM) to stop a process by its PID.

**systemctl status <service>**

Checks status and logs of a systemd service.

**3. User Logins & Activity**

**who**

Shows who is currently logged in.

**w**

Displays logged-in users plus their current activities.

**last**

Shows login history with timestamps.

**lastlog**

Lists last login times for all users.

**id <username>**

Shows user ID (UID), group ID (GID), and group memberships.

**groups <username>**

Lists groups the user belongs to.

**passwd**

Change user password.

**su - <user>**

Switch user with environment loaded (login shell).

**sudo -i**

Get an interactive root shell.

**history**

Shows command history for the current user.

**journalctl \_UID=$(id -u)**

Shows logs for the current user via systemd journal.

1. **Log File Management**

**tail -n 100 /var/log/syslog**

Show last 100 lines of syslog.

**less /var/log/messages**

View large log files with scroll/search support.

**journalctl**

Systemd log viewer, for querying system logs.

**logrotate**

Tool to manage log rotation and compression.

**grep "error" /var/log/syslog**

Search for "error" keyword in syslog.

**find /var/log -name "\*.log"**

Find all .log files in /var/log directory.

**du -sh /var/log**

Check total size of the /var/log directory.

**5. Network & Connectivity**

**ifconfig / ip a**

Show network interfaces and their IP addresses.

**ip r**

Display routing table.

**ping <ip>**

Test network reachability.

**netstat / ss**

Show active connections and listening ports.

**nmap <host>**

Scan remote host ports.

**curl <url>**

Transfer data from or to a server (download/upload).

**wget <url>**

Download files from the web.

**scp user@host:/file .**

Securely copy files from remote host to local machine.

**rsync -avz**

Efficiently sync files/directories between locations.

**hostname -I**

Show all IP addresses assigned to the host.

1. **System & Process Management**

**ps aux:** Show all running processes

**top / htop**: Monitor system resource usage

**kill -9 <PID>**: Forcefully stop a process

**nice / renice**: Adjust process priority

**systemctl status <service>**: Check service status

**journalctl -xe**: View systemd logs for troubleshooting

1. **Package Management**

**apt update**: Update package index (Debian/Ubuntu)

**apt install <package>**: Install a package (Debian/Ubuntu)

**dpkg -l | grep <package>**: List installed packages (Debian/Ubuntu)

**yum install <package>**: Install a package (RHEL/CentOS)

**rpm -qa | grep <package>**: List installed RPM packages

**3. File & Directory Commands**

**ls -al**: List with permissions and details

**cp, mv, rm, mkdir**: File operations

**find / -name <file>**: Locate files

**du -sh \*:** Disk usage summary

**df -h:** Filesystem usage

**4. Permissions & User Management**

**chmod, chown, chgrp**: Change permissions/ownership

**adduser, deluser**: Manage users

**usermod -aG <group> <user>**: Add user to a group

**passwd <user>**: Set/change user password

**5. Networking & Connectivity**

**ip a**: Show interfaces and IPs

**ping <host>**: Test connectivitycurl, wget: Test HTTP endpoints or download files

**netstat / ss**: View ports and connections

**nmap <host>:** Network scanner