Linux for Devops

Fundamental Linux Commands date, df, touch, mkdir, cp, ls, etc...

Navigation → `ls, cd, pwd`
👉 "Look Carefully Please."

File/Folder Operations → `cp, mv, rm, mkdir, rmdir, touch`
👉 "Copy My Records, Make Real Texts.."

File Viewing → `cat, less`
👉 "Cats Listen."

Storage → `df, du`
👉 "Disk Used."

🔹 Navigation

- **ls** → list files in directory

```
 ls -l   # detailed listing
```

- **cd** → change directory

```
 cd /home/user
```

- **pwd** → print working directory

```
pwd   # /home/user/projects
```

👉 **"Look Carefully Please."**

🔹 File / Folder Operations

- **cp** → **copy file/folder**

```
cp file.txt backup/
```

- mv → move/rename file

(Example: `mv file.txt newname.txt`)

- rm → remove file

(Example: `rm old.txt, rm -r folder/ # remove directory`)

- mkdir → make directory

(Example: `mkdir myfolder`)

- rmdir → remove empty directory

(Example: `rmdir oldfolder`)

- touch → create empty file

  (Example: `touch notes.txt`)

- 👉 Acronym: "Copy My Records, Make Real Texts."

File Viewing → cat, less

- cat → view file content

  (Example: `cat file.txt`)

- less → view file content page by page

  (Example: `less biglog.txt`)

- 👉 Acronym: "Cats Listen."

Storage → df, du

- df → disk free space
- (Example: `df -h # human-readable (GB/MB)`)
- du → disk usage
- (Example: `du -sh *`)
- 👉 Acronym: "Disk Used."

System Info → date, whoami

- date → show system date/time
- (Example: `date # Mon Sep 01 08:40:22 NPT 2025`)
- whoami → show current user
- (Example: `whoami # nyapu)`
- 👉 Acronym: "Don’t Worry."

Command	What it does	Memory Technique & Story
`grep`	**G**lobally search for a **R**egular **E**xpression and **P**rint	**G**et **R**eally **E**xcited **P**attern! You're excitedly searching for a pattern in a file. `grep "error" log.txt`
`find`	Find files	Simple. It **find**s files. Remember the syntax: `find /where/to/look -name "filename"`
`chmod`	**Ch**ange **mod**e (permissions)	**Ch**ange the **Mod**e of a file (Read, Write, eXecute). Think of a file putting on different "modes" of clothing (rwx).
`chown`	**Ch**ange **own**er	**Ch**ange the **Own**er. A file changes its owner like a house changes its **own**er.
`ps`	**P**rocess **S**tatus	**P**rocess **S**napshot. `ps aux` is like taking a detailed **AUX**iliary snapshot of all processes.
`kill`	Kill a process	Simple. It **kill**s a process. `kill -9` is the "**SIGKILL**" signal—imagine a sniper rifle. Unstoppable.
`tar`	**T**ape **Ar**chive	**T**ake **A** **R**oadtrip. You're bundling (**c**reating) files into a car (archive) or **x**tracting them out at the destination. `-z` for zipping through traffic, `-v` for viewing the scenery. `tar -czvf archive.tar.gz /my/folder`
`curl`	**C**lient **URL**	**C**all **URL**. You're using a command-line client to **c**all a **URL** and see what it returns.
`wget`	**W**eb **Get**	**W**eb **Get**. Simple. It **get**s a file from the **w**

Flags modify a command's behavior. Remember these like cheat codes.

Flag	Meaning	Memory Technique
`-r`	**R**ecursive (for directories)	**R** for **R**ecursive. The command goes through every folder, **r**ecursively.
`-f`	**F**orce (bypass warnings)	**F** for **F**orce. Like a superhero forcing an action.
`-h`	**H**uman-readable (shows MB, GB)	**H** for **H**uman. It presents data in a way a **h**uman can easily understand.
`-a`	**A**ll (show hidden files, all procs)	**A** for **A**ll. It shows **a**ll the things, even the hidden ones. `ls -a`, `ps -a`
`-l`	**L**ong listing format (more details)	**L** for **L**ong. It gives you the **l**ong, detailed version. `ls -l`
`-v`	**V**erbose (show more output)	**V** for **V**erbose. It's very **v**erbose and chatty about what it'

🔐 Linux Permissions Overview

Every file/directory in Linux has **3 types of permissions** for **3 types of users**:

1. Types of Users

- **Owner (u)** → person who created the file.
- **Group (g)** → a team of users.
- **Others (o)** → everyone else.

2. Types of Permissions

- **r** → Read (view file / list folder)
- **w** → Write (edit file / create-delete in folder)
- **x** → Execute (run file / enter folder)

📂 Example: Checking Permissions

Command:

```
ls -l
```

output :

```
-rwxr-xr--  1 nyapu devops  1234 Sep  1  notes.sh
```

Breakdown:

- `-` → file type (`-` = file, `d` = directory)
- `rwx` → **owner** (read, write, execute)
- `r-x` → **group** (read, execute)
- `r--` → **others** (read only)

So here:

- Owner can read/write/run.
- Group can read/run.
- Others can only read.

We know that :

- **Owner (u)** → person who created the file.
- **Group (g)** → a team of users.
- **Others (o)** → everyone else

⚙️ Changing Permissions

1.Symbolic Method

```
chmod u+x script.sh   # give execute to owner
chmod g-w file.txt    # remove write from group
chmod o+r report.txt  # add read for others
```

2. Numeric Method

Each permission has a number:

- r = 4
- w = 2
- x = 1

Add them up:

- `rwx = 7`
- `rw- = 6`
- `r-- = 4`

So, `chmod 755`:

- Owner: 4+2+1 = **7** (Read, Write, Execute)

- Group: 4+0+1 = **5** (Read, Execute)
- Others: 4+0+1 = **5** (Read, Execute)

Example : chmod 755 script.sh
Means:

- Owner = 7 (rwx)
- Group = 5 (r-x)
- Others = 5 (r-x)