Top 50 Linux command

📂 1. Navigation & Workspace Setup

- `pwd`: Displays the full path of the folder you are currently in.
  - *Example:* `pwd`
- `ls`: Lists all files and folders in your current location.
  - *Example:* `ls -alh` (Lists all files, including hidden ones, in human-readable sizes).
- `cd`: Changes your current directory to a new path.
  - *Example:* `cd ~/projects/ReactApp`
- `mkdir`: Creates a new folder.
  - *Example:* `mkdir -p src/components` (Creates the parent folder and sub-folder at once).
- `touch`: Creates a new empty file.
  - *Example:* `touch server.js`
- `alias`: Creates a custom shortcut for a long command.
  - *Example:* `alias gs='git status'`
- `clear`: Wipes the terminal screen clean for a fresh start.
  - *Example:* `clear`

🛠️ 2. File Manipulation & Management

- `cp`: Copies files or directories from one place to another.
  - *Example:* `cp main.cpp main_backup.cpp`
- `mv`: Moves or renames a file or folder.
  - *Example:* `mv old_name.js new_name.js`
- `rm`: Deletes files or folders (use with caution).
  - *Example:* `rm -rf node_modules/` (Forcefully deletes a folder and everything inside).
- `ln`: Creates a "symbolic link" or shortcut to another file.
  - *Example:* `ln -s /var/www/html site_link`
- `tar`: Compresses or extracts archived files.
  - *Example:* `tar -xvf source_code.tar.gz`
- `zip`: Compresses files into a .zip format.
  - *Example:* `zip -r backup.zip ./src`
- `unzip`: Extracts files from a .zip archive.
  - *Example:* `unzip project.zip`
- `chmod`: Changes file permissions (Read, Write, Execute).
  - *Example:* `chmod 755 run.sh` (Makes a script executable).
- `chown`: Changes the owner or group of a file.
  - *Example:* `sudo chown vishal:devs app.log`

🔍 3. Text Processing & Searching (The SDE Power Tools)

- `cat`: Prints the entire content of a file to the screen.
  - *Example:* `cat .env`
- `less`: Allows you to scroll through large files without loading the whole thing.
  - *Example:* `less huge_log_file.log`
- `grep`: Searches for specific text inside files.
  - *Example:* `grep -r "API_KEY" .` (Searches for "API_KEY" in all files in the current folder).
- `find`: Locates files based on name, size, or time.
  - *Example:* `find src/ -name "*.cpp"`
- `head`: Shows the very beginning of a file.
  - *Example:* `head -n 5 index.js` (Shows the first 5 lines).
- `tail`: Shows the end of a file (essential for debugging logs).
  - *Example:* `tail -f access.log` (Updates the screen in real-time as the log grows).
- `diff`: Compares two files line by line and shows what is different.
  - *Example:* `diff version1.cpp version2.cpp`
- `sed`: Automatically finds and replaces text in a file.
  - *Example:* `sed -i 's/localhost/127.0.0.1/g' config.js`
- `awk`: Used for extracting specific columns or data from text files.
  - *Example:* `awk '{print $1, $3}' data.txt`
- `sort`: Alphabetizes or numerically orders the lines in a file.
  - *Example:* `sort names.txt`
- `wc`: Counts the number of lines, words, or characters.
  - *Example:* `wc -l main.cpp` (Tells you how many lines of code are in the file).
- `echo`: Prints text to the terminal or writes it to a file.
  - *Example:* `echo "Build Successful" >> build.log`
- `tee`: Saves output to a file while also showing it on the screen.
  - *Example:* `ls | tee file_list.txt`

⚡ 4. System Monitoring & Performance

- `top`: A live task manager showing CPU and Memory usage.
  - *Example:* `top`
- `htop`: A more colorful, user-friendly version of `top`.
  - *Example:* `htop`
- `ps`: Lists the programs (processes) currently running on your system.
  - *Example:* `ps aux | grep node` (Finds all running Node.js processes).
- `kill`: Stops a running program using its ID (PID).
  - *Example:* `kill -9 1234` (Forcefully stops process 1234).
- `df`: Shows how much disk space is left on your drives.
  - *Example:* `df -h`

- `du`: Shows how much space a specific folder is taking up.
  - *Example:* `du -sh ./node_modules`
- `free`: Displays how much RAM memory is available.
  - *Example:* `free -h`
- `uptime`: Shows how long the system has been running since the last reboot.
  - *Example:* `uptime`
- `history`: Lists all the commands you have typed in the past.
  - *Example:* `history 20` (Shows your last 20 commands).

🌐 5. Networking & Remote Access

- `ssh`: Securely logs you into a remote server.
  - *Example:* `ssh vishal@remote-server-ip`
- `scp`: Securely copies files from your PC to a remote server.
  - *Example:* `scp build.zip vishal@server:/var/www/`
- `ping`: Checks if a website or server is online.
  - *Example:* `ping google.com`
- `curl`: Downloads files or tests API endpoints from the terminal.
  - *Example:* `curl -I https://google.com` (Gets the header info of the site).
- `wget`: Downloads a file directly from a web link.
  - *Example:* `wget https://example.com/installer.sh`
- `ip a`: Shows your local and public IP addresses.
  - *Example:* `ip a`

⚙️ 6. Administration & Utilities

- `sudo`: Runs a command with administrative ("Root") permissions.
  - *Example:* `sudo apt update`
- `apt / yum`: Package managers used to install new software.
  - *Example:* `sudo apt install g++`
- `whoami`: Tells you which user account you are currently using.
  - *Example:* `whoami`
- `uname`: Displays information about the Linux system you are on.
  - *Example:* `uname -a`
- `passwd`: Changes the password for a user.
  - *Example:* `passwd vishal`
- `date`: Shows the current system time and date.
  - *Example:* `date`
- `man`: Opens the "manual" (instructions) for any other command.
  - *Example:* `man grep`