Linux Basic Commands:

NOTE: To **view Hidden Files and Folders** in the window:
⇒ **Ctrl + H**

A. Terminal Command:

NOTE: Inside Terminal, for **auto completion:**
⇒ press **Tab**, while typing

1. **To open the terminal:**
   ⇒ **Ctrl + Alt + T**

2. **To close the terminal:**
   ⇒ **Ctrl + D**

3. **To view all historical commands on terminal:**
   ⇒ **history**

4. **To clear terminal all historical commands:**
   ⇒ **history -c**

B. Directory/Folder Commands:

1. **List/View all files and directories/folders in the current directory:**
   ⇒ **ls**

   ⇒ **ls -l** (list **information** of directories)
   ⇒ **ls -lh** (list **information** of directories in better **readable format**)

   ⇒ **ls -a** (list both regular & **hidden** files and folder)
   ⇒ **ls -al** (list/show all **files permissions**)

   NOTE: To only list directories/folders:
   ⇒ **dir**

2. **Print current working directory:**
   ⇒ **pwd**

3. **To change the current directory:**
   ⇒ **cd**

   eg: To go one step back of the directory ⇒ **cd ..**

   eg: To go to linux root directory ⇒ **cd /**

   eg: To go inside User directory ⇒ **cd ~**

4. **Creates a new directory/folder:**
   ⇒ **mkdir directory_name**
   eg ⇒ mkdir hello

5. **To remove a directory/folder:**
   ⇒ **rm -r directoryname**
   eg: rm -r test_folder

   NOTE: To forcefully & recursively remove a folder/directory:
   ⇒ **rm -rf directoryname**
   eg: rm -rf test_folder

6. **For copying a file or folder to the directory/folder:**
   ⇒ **cp filename.extension directory_path**
   eg: cp hello.txt dhiraj/

7. **For moving a file or folder to the directory:**
   ⇒ **mv filename.extension directory_path**
   eg: mv hello.txt dhiraj/

8. **For creating a hidden folder:**
   ⇒ **mkdir .folder_name**
   eg: mkdir .hello

9.

C. File Commands:

1. **To create a file:**
   ⇒ **touch filename.extension**
   eg: touch test_file.txt

   NOTE: if u want to create a file & at same time write some content to it:
   ⇒ **cat > filename.extension**
   eg: cat > test2_file.txt

   Now, write the content to the file and press **Ctrl + D** to **save** and **exit**.

2. **To create a hidden file:**
   ⇒ **touch .hiddenfilename.extension**
   eg: touch .hidden_hello.txt

3. **To view type of any file:**
   ⇒ **file filename.extension**
   eg: file hello.txt

4. **To print no. of lines, words & byte of/in a file:**
   ⇒ **wc filename.extension**
   eg: wc hello.txt

   NOTE: For only, **lines counts → wc -l hello.txt**
   NOTE: For only, **words counts → wc -w hello.txt**

5. **To rename a file:**

   Linux doesn't really have a command for renaming files, we also make use of the **mv** command to rename files and folders.

   ⇒ **mv filename.extension newfilename.extension**
   eg: mv hello.txt hello_from_dhiraj.txt

   NOTE: while renaming using mv command, if there already exists newfilename in that directory, then pre existing file having same name will be overridden. So, u better be cautious.

6. **To copy a file:**
   ⇒ **cp filename copyfilename**
   eg: cp hello hello1

7. **To remove/delete a file:**
   ⇒ **rm filename**

   NOTE: To **forcefully** remove a file:
   ⇒ **rm -f filename**
   eg: rm -f hello1

8. **To output/read the content of a file in terminal:**
   ⇒ **more filename.extension**
   eg: more hello.txt

D. Installation Commands:

1. apt (Advanced Package Tool):

- Used in **Debian and Debian-based** systems like **Ubuntu**.

To install a package:
⇒ **sudo apt-get install package_name**

To update all installed package lists:
⇒ **sudo apt-get update**
To upgrade all installed packages:
⇒ **sudo apt-get upgrade**

To list out all installed packages:
⇒ **apt list --installed**

To check if the package is installed or not:
⇒ **apt list --installed | grep package_name**
eg: apt list --installed | grep google-chrome-stable

To uninstall/remove package:
⇒ **sudo apt remove package_name**

To completely remove package with it's config files:
⇒ **sudo apt purge package_name**

2. snap:

- A universal package manager **for many linux distributions** including **Ubuntu, Linux Mint, Debian and Fedora.**

To install a package:
⇒ **sudo snap install package_name**

To update a packages:

⇒ **sudo snap refresh package_name**

To list out all installed packages:

⇒ **snap list**

To check if a particular package is installed or not:

⇒ **snap list | grep package_name**

If the package is installed it will show single line information in terminal, else it won't show any thing in terminal.

To uninstall/remove package:

⇒ **sudo snap remove package_name**

3. **'.deb' file:**
   - using **dpkg** package manger for .deb file.
   - The low-level package manager **behind APT**.
   - It is primary package manager for **Debian and Debian-based systems, like Ubuntu.**
   -

To install a .deb package:

⇒ **sudo dpkg -i package_file.deb**

To list out all installed packages:

⇒ **sudo dpkg -l**

If all installed packages is not shown clearly in the terminal, then redirect the output to a file:

⇒ **sudo dpkg -l > output.txt**

To check if a particular package is installed or not:

⇒ **dpkg -l package_name**

eg: dpkg -l google-chrome-stable

If the package is installed, it will display information about the package.

To uninstall/remove .deb package:

⇒ **sudo dpkg -r package_name**

To completely remove package with it's configurations:

⇒ **sudo dpkg --purge package_name**

4. **‘.rpm’ file:**

- Using **RPM (Universal method)**:

To install a ‘.rpm’ package:

⇒ **sudo rpm -i package_name.rpm**

OR

⇒ **sudo rpm -ivh package_name.rpm**

Where,

-i : install a package

-v : verbose for a nicer display

-h : print hash marks as the package archive is unpacked.

To check if a package is installed or not:

⇒ **rpm -qa | grep package_name**

5.

E. Sudo Commands:

sudo: It stands for "Super User DO" and allows authorized users to execute commands with administrative (root) privileges. It requires the user's password to verify their identity and grant temporary elevated privileges.

1. **‘sudo su’:**

NOTE: su: It stands for **"Switch User"** and is used to change to another user account.
By default, if no username is provided**,** it **switches to** the **root user.**

2. **‘sudo -i’:**

NOTE: This command starts a new shell with the root user's environment, similar to "sudo su". It gives you a root shell with root's home directory and sets the environment variables accordingly.

F. Search (using ‘grep’) and Locate Commands:

‘grep’ is a powerful command-line utility in Linux and Unix-like operating systems used to search for specific patterns in text files or input streams. The name "grep" stands for "Global Regular Expression Print," which hints at its primary function: searching for regular expressions and printing the matching lines.

Here’s the basic **syntax** of grep command:

⇒ **grep [OPTIONS] PATTERN [FILE...]**

- `OPTIONS`: Optional flags that modify the behavior of the `grep` command.
- `PATTERN`: The regular expression or plain text string you want to search for.
- `FILE`: The name of the file(s) in which you want to search. If not provided, `grep` reads from standard input (e.g., output from a pipe).

Here are some common options you can use with `grep`:

- `-i` or `--ignore-case`: Ignore case distinctions when searching.
- `-r` or `--recursive`: Recursively search in directories and subdirectories.
- `-n` or `--line-number`: Print line numbers along with the matching lines.
- `-w` or `--word-regexp`: Match only whole words, not part of words.
- `-v` or `--invert-match`: Invert the search, printing lines that do not match the pattern.

1. **Search for the word "example" in a file named `file.txt`:**

   ⇒ grep "example" file.txt

2. **Search for the word "apple" ignoring case in all text files in the current directory:**

   ⇒ grep -i "apple" *

3. **Search for the word "success" in a file named `data.txt`, displaying line numbers:**

   ⇒ grep -n "success" data.txt

4. **Counts the number of lines having letter ‘b’ in the file.txt:**

   ⇒ grep -c b file.txt

‘locate’ command is a powerful utility in Linux-based systems used to quickly find files and directories based on their names.

It allows you to perform searches across the entire file system(whole disk) or specific directories, making it a useful tool for finding files even if you are unsure of their exact locations.

Befor using locate command, u have to install ‘mlocate’:

If you are using Ubuntu or Debian-based systems, you can install `mlocate` using the following command:

⇒ sudo apt-get update

⇒ sudo apt-get install mlocate

1. Building the mlocate system database:

Before you can use `locate`, you need to ensure that mlocate database is up-to-date. The mlocate database contains a list of filenames and their corresponding paths. To build or update the database, run the following command with superuser privileges (usually, you need to be the root user or use `sudo`):

⇒ sudo updatedb

This command will update the database, and it's typically scheduled to run regularly (e.g., daily or weekly) via a cron job.

2. Basic usage of 'locate' command:

Once the database is up-to-date, you can use the `locate` command to find files. The basic syntax of `locate` is:

> **Syntax: locate [options] pattern**

`'pattern'`: The search pattern you want to match against filenames. It can be a partial or complete file or directory name.

Some Examples:

1. **To find all files and directories that contain the word "example":**

   ⇒ locate example

2. **To find files with specific extensions (e.g., .txt, .jpg):**

   ⇒ locate *.txt

   ⇒ locate *.pdf

   ⇒ locate *.jpg

   ⇒ locate *.conf

3. **To find files and directories that match a specific pattern:**

   ⇒ locate "my_document"

   ⇒ locate "documents/*"

4. **Case sensitivity locate:**

   use the `-i` option:

   ⇒ locate -i "Example"

5. **Limiting search scope (only inside specific directories):**

   By default, locate searches the entire filesystem. However, you can narrow down the search to a specific directory by using the path as part of the pattern:

   ⇒ locate /path/to/directory/pattern

6. **To limit the output of locate command:**

   Using --limit

   locate -i --limit 100 *.txt

G.

Top 50 Linux Commands You Must Know as a Regular User

1. **ls** - The most frequently used command in Linux to list directories
2. **pwd** - Print working directory command in Linux
3. **cd** - Linux command to navigate through directories
4. **mkdir** - Command used to create directories in Linux
5. **mv** - Move or rename files in Linux
6. **cp** - Similar usage as mv but for copying files in Linux
7. **rm** - Delete files or directories
8. **touch** - Create blank/empty files
9. **ln** - Create symbolic links (shortcuts) to other files
10. **cat** - Display file contents on the terminal
11. **clear** - Clear the terminal display
12. **echo** - Print any text that follows the command
13. **less** - Linux command to display paged outputs in the terminal
14. **man** - Access manual pages for all Linux commands
15. **uname** - Linux command to get basic information about the OS
16. **whoami** - Get the active username
17. **tar** - Command to extract and compress files in Linux
18. **grep** - Search for a string within an output
19. **head** - Return the specified number of lines from the top
20. **tail** - Return the specified number of lines from the bottom
21. **diff** - Find the difference between two files
22. **cmp** - Allows you to check if two files are identical

23. **comm** - Combines the functionality of diff and cmp
24. **sort** - Linux command to sort the content of a file while outputting
25. **export** - Export environment variables in Linux
26. **zip** - Zip files in Linux
27. **unzip** - Unzip files in Linux
28. **ssh** - Secure Shell command in Linux
29. **service** - Linux command to start and stop services
30. **ps** - Display active processes
31. **kill and killall** - Kill active processes by process ID or name
32. **df** - Display disk filesystem information
33. **mount** - Mount file systems in Linux
34. **chmod** - Command to change file permissions
35. **chown** - Command for granting ownership of files or folders
36. **ifconfig** - Display network interfaces and IP addresses
37. **traceroute** - Trace all the network hops to reach the destination
38. **wget** - Direct download files from the internet
39. **ufw** - Firewall command
40. **iptables** - Base firewall for all other firewall utilities to interface with
41. **apt, pacman, yum, rpm** - Package managers depending on the distro
42. **sudo** - Command to escalate privileges in Linux
43. **cal** - View a command-line calendar
44. **alias -** Create custom shortcuts for your regularly used commands
45. **dd** - Majorly used for creating bootable USB sticks
46. **whereis** - Locate the binary, source, and manual pages for a command
47. **whatis** - Find what a command is used for
48. **top** - View active processes live with their system usage
49. **useradd and usermod** - Add new user or change existing users data
50. **passwd** - Create or update passwords for existing users