Linux

Shell:

→ Command Line Interface (CLI)

→ Takes command as input

→ Interpret commands from user and tells OS what to do.

/bin - binary file (executable file)

eg? /usr/local/bin/python

↓ ↓ ↓ ↓

user Local binary File name.

~ ls - List commands - Lists all the files/folder in the directory.

~ open . - Shows all the file/folder in the directory.

~ mkdir foldername - Creates folder

~ cd Downloads - Changing directory to downloads folder.

~ Downloads ls → Lists folders/files in downloads.

(Directory, if nothing displayed ie, u r in home directory)

~ Downloads cd .. → Goes back one folder.

~ cd Pictures → Navigates to picture folder

~ where foldername → Gives the pathway to the folder.

~ open usr/local/bin → Opens the pathway.

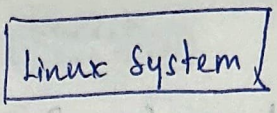
pathway

~ ls -a → Shows all hidden/unhiden files.

Linux Commands vs Window OS

Using commands to open/create/delete files/folders

↳ Uses mouse clicks to open/create/delete files/folders



Hardware

Linux kernel
- Manages Hardware tells how to interact with system
- Between Hardware & UI

User Space.

BASH (Bourne Again Shell) => Shell scripting.

Wtever commands you type in, BASH understands, executes and provides output

Alternatives to BASH; zsh, ksh

File Directory Structure

"/" => Root folder/Directory

/bin, /etc, /home, /opt, /tmp, /usr, /var => Folders/files with Root directory.

Commands:

$ pwd => Provides where u are at currently
↳present working directory

/home/ec2-user => File path.
- /home → Root
- /ec2-user → Home folder
- ↳ ec2-user file.

File path

Absolute path
This is when u give entire path along with cd command
eg:
cd /home/ec2-user/content
file path.

Relative path
This is when u use current directory and switch to another from that
eg: using cd., cd.. commands

$ ls => Lists files in the directory

$ ls -l => long listing => Gives detailed info on files/folder in the directory

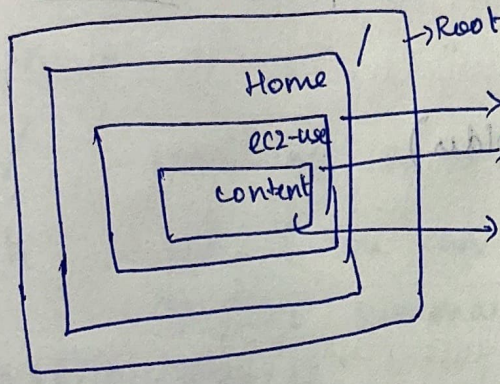
cd => change Directory

relative path commands:

$ cd content/ => Takes u directly to folder

$ cd .. => Takes you to parent directory (one folder above)

$ cd ~ => Takes you to home diretory.



→ Root Directory.
→ Home Directory.
→ Parent Directory (one file above)
→ file.

$ cd - => Takes you to previous file/folder/Directory

Listing commands:

$ ls => Lists files in current directory.

$ ls -l => long listing, detailed information on files.

$ ls -a => Displays all files (hidden as well)
(If a file starts with ".", treated as hidden file)

$ ls -l -a or $ ls -la => Long listing of all files.

$ ls -R => Recursive listing

$ ls -r => reverse order

$ ls -t => sorting files

touch command

- Creates new file

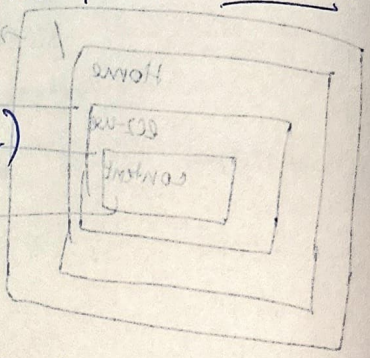
→ $touch newfile ⇒ eg: Let us say this file created at 12/03/2025 14:59:01

You can use this command just to touch the file.

Now if I again execute the command.

→ $touch newfile

→ $ls -l ⇒ Here you can see the datetime modification 12/03/2025 15:01:01



mkdir command:

Creates directory (folder)

→ $mkdir newfolder

file command:

→ $file newfile

output: empty

$file newfile1.

output: ASCII values with very long lines

This command will give description of content in the file

cat command

To view the contents of file.

→ $cat newfile.txt

Displays contents of file.

history commands:

Displays all the commands you have typed at that point

→ $history

clear command:

clears the workspace

→ $clear

less command:

This command will open the file in a editor mode user can navigate through the file, search for a word

→ $less myfile.txt ⇒ Opens a file in a window content.

If you type:

q - Exits the editor window and goes back to terminal

g - moves the content to begining of file.

G - " " " end of file.

/ - searching a word (example: /good) → word to be searched.

h - Help (You can use it in editor window)
Displays summary or instruction manual for "less"

FILE MANIPULATION:

Copy command

cp ⇒ Transferring file from one folder to another

→ $cp myfile.txt newfolder
↓ file to be moved ↳ Where to move.

This will create copy of myfile.txt in newfolder directory

Make Directory/folder:

mkdir ⇒ Make directory or creates folder.

→ $mkdir newfolder1.

→ If you want to move files in bulk:

* ⇒ Wildcard character. (text manipulation)

→ $cp *.txt newfolder1

This will ~~move~~ copy files that ends with ".txt" to newfolder1.

Move command

Use this command to modify file name or to move file from one place to another.

→ $ mv myfile myfile.txt ⇒ Renaming the file.

→ $ mv *.txt cad/ ⇒ Move all files with ".txt" to cad directory

→ $ mv cad/ dog ⇒ Renaming directory.

Remove commands:

→ $ rm -r newfolder/

↳ Recursive removing.

Deletes all the files in newfolder and then deletes newfolder.

→ $ rmdir DirectoryName ⇒ Remove Directory. The directory should be empty.

Copy command with confirmation:

Let's say you have two files with same name in two different folder. When you try to copy the file from one folder to another, it should popup a message for confirmation.

→ $ cp (-i) /home/ec2-user/dog/myfile1.txt . ? ⇒ current directory

↓ filepath.

Copy the file to the current directory.

When you give -i it will ask for confirmation such as overwrite filepath ? You can specify yes or no

If you are not giving -i, it will directly overwrite the file without confirmation

Removing ~~direct~~ non empty directory:

$rmdir ⇒ can remove only empty directory.

To remove all the files within the directory:

$ rm -r content/ ⇒ Recursively deletes the files and directory
↳ Directory name

Creating subfolder

$ mkdir -p newfolder/car
↓ ↳ sub folder
main folder

Find command: To find file/directory.

(syntax)

→ $find /home -name *.docx
↓ ↓
Directory Filename.

Help commands.

→ $man find ⇒ Gives a manual on "find" command.
↓
manual

→ $whatis find ⇒ Gives a short description on command

→ $help echo.

Alias command: Setting shortcut for command.

→ $alias f="ls -la"

I can use 'f' instead of ls -la command

You can set alias for repeatative commands.

→ unalias f ⇒ This will unset the alias for 'f'

Exit command

$ exit ⇒ closes the terminal window

$ logout ⇒ logs out of user.