

Part 2

Linux Command Line Basics :

First : What is the Command Line ?

The **Command Line (Terminal)** is just a place where :

- You type commands
- Linux executes them
- You control the system directly

Think of it like talking to Linux in it's own Language.

Section 1 : "Where am I and what is around me ?"

Before doing anything, Linux users always ask two questions :

1. **Where am I ?**
2. **Where files/folders are here ?**

1. `pwd` — Where am I ?

```
pwd
```

Output Example :

```
/home/hiten/Documents
```

Meaning :

- You are inside the `Documents` folder
- Which is inside `hiten`
- Which is inside `home`

Memory trick :

`pwd` = Present Working Directory

2. `ls` — Where is here ?

```
ls
```

shows :

- Files
 - Folders
- in the current Directory

Common useful options :

```
ls -l
```

```
ls -a
```

Command	Meaning
<code>ls -l</code>	Detailed view (permissions, size, date)
<code>ls -a</code>	Shows hidden files

hidden files starts with `.` (Example : `.bashrc`)

3. `cd` — move between folders

```
cd downloads
```

Moves you into `Downloads`

Special cases (very important) :

```
cd ..
```

→ Move one level up.

```
cd ~
```

→ Go to home Directory.

```
cd /
```

→ Go to root of the system.

Think of folders like room in a building — `cd` lets you walk between them.

Section 2 : Understanding Linux Files Structure

Linux starts from :

```
/
```

Key Folders :

Folder	Purpose
<code>/home</code>	User files
<code>/etc</code>	Configuration
<code>/bin</code>	Essential commands
<code>/var</code>	Logs, variable data

You don't need all details now — just remember :

Your work happens inside `/home/yourname`

Section 3 : Creating Files and Folders

1. `mkdir` — Create Folder

```
mkdir projects
```

Creates a folder names `projects`

Create multiple folders :

```
mkdir frontend backend
```

2. `touch` — Create file

```
touch notes.txt
```

Creates an empty file.

`touch` is also used to update file timestamps.

Section : 4 Copy, Move, Rename, Delete

1. Copy Files — `cp`

```
cp file1.txt file2.txt
```

Creates a copy.

Copy Folder :

```
cp -r folder1 folder2
```

`-r` = recursive (required for folders).

2. Move or Rename — `mv`

```
mv old.txt new.txt
```

Rename file

```
mv file.txt Documents/
```

Move file.

Same command does two jobs.

3. Delete — `rm` (use carefully)

```
rm file.txt
```

Delete folder :

```
rm -r myfolder
```

No recycle bin. Once deleted, it's gone.

Professor rule :

if unsure, list first(ls) then delete.

Section 5 : Viewing File Content

1. `cat` — Show entire file

```
cat notes.txt
```

Good for small files.

2. `less` — Scroll through file

```
less bigfile.txt
```

Controls :

- `Enter` → next line
- `Space` → next page
- `q` → quite

This is how professionals read logs.

3. `head` and `tail`

```
head file.txt  
tail file.txt
```

Default : first / last 10 lines

Live log monitoring :

```
tail -f logfile.log
```

Keep updating automatically.

Section 6 : Why Commands sometimes Fails ?

Common reasons :

1. Wrong Directory
2. File doesn't exists
3. Permission denied

```
pwd  
ls  
ls -l
```

Always observe before acting.