

# Linux Commands Guide

---

## ASSIGNMENT 1 - PRACTINCING LINUX COMMMANDS

A quick reference to essential Linux commands with short explanations.

---

### 1. `pwd`

Print Working Directory → shows the full path of your current location in the filesystem.  
Useful to confirm where you are before running other commands.

---

### 2. `ls`

Lists files and directories in the current location.  
Options like `ls -l` show details, and `ls -a` shows hidden files.

---

### 3. `cd`

Change Directory → lets you move between folders.  
Example: `cd /home/user/Documents` goes to the Documents folder.

---

### 4. `mkdir`

Make Directory → creates a new folder in the current location.  
Example: `mkdir projects` creates a folder named *projects*.

---

### 5. `touch`

Creates an empty file or updates the timestamp of an existing one.  
Example: `touch notes.txt` makes a blank text file.

```
[mukund@parrot]~  
$ pwd  
/home/mukund  
[mukund@parrot]~  
$ ls  
c_lab  Desktop  Downloads  file2  linux  Pictures  save.txt  Videos  
array1.sh  array2.sh  array3.sh  array4.sh  array5.sh  data1.txt  data2.txt  data3.txt  data5.txt  data6.txt  data.txt  folder1  folder2  prime.sh  sum_of_digits.sh  
[mukund@parrot]~  
$ cd linux  
[mukund@parrot]~/linux  
$ mkdir linux  
[mukund@parrot]~/linux  
$ ls  
armstrong.sh  array3.sh  data1.txt  data5.txt  folder1  prime.sh  
array1.sh  array4.sh  data2.txt  data6.txt  folder2  sum_of_digits.sh  
array2.sh  array5.sh  data3.txt  data.txt  linux  
[mukund@parrot]~/linux  
$ touch a.txt  
[mukund@parrot]~/linux  
$ ls  
armstrong.sh  array3.sh  a.txt  data3.txt  data.txt  linux  
array1.sh  array4.sh  data1.txt  data5.txt  folder1  prime.sh  
array2.sh  array5.sh  data2.txt  data6.txt  folder2  sum_of_digits.sh
```

## 6. `cp`

Copy → duplicates files or directories.

Example: `cp file.txt backup.txt` makes a copy named *backup.txt*.

---

## 7. `mv`

Move → shifts files between directories or renames them.

Example: `mv old.txt new.txt` renames *old.txt* to *new.txt*.

---

## 8. `rm`

Remove → deletes files or directories permanently.

Use with caution; `rm -r folder/` deletes a folder and its contents.

---

## 9. `cat`

Concatenate → displays the contents of a file directly in the terminal.

Example: `cat file.txt` prints the text inside *file.txt*.

---

## 10. `nano`

Opens the Nano text editor for creating or editing files.

Example: `nano file.txt` allows editing *file.txt* inside the terminal.

```
[mukund@parrot]~[~/linux]
└─ $cp data.txt a.txt
[mukund@parrot]~[~/linux]
└─ $mv data1.txt data2.txt
[mukund@parrot]~[~/linux]
└─ $rm a.txt
[mukund@parrot]~[~/linux]
└─ $cat data1.txt
cat: data1.txt: No such file or directory
[✖]-[mukund@parrot]~[~/linux]
└─ $ls
armstrong.sh  array3.sh  data2.txt  data6.txt  folder2  sum_of_digits.sh
array1.sh     array4.sh  data3.txt  data.txt   linux
array2.sh     array5.sh  data5.txt  folder1    prime.sh
[mukund@parrot]~[~/linux]
└─ $cat data2.txt
au65kulyoiuthivu
[mukund@parrot]~[~/linux]
└─ $
```

## ? Extra Questions & Answers

---

Q1: What is the difference between `chmod` and `chown`?

- `chmod` → changes **permissions** (who can read, write, or execute a file).
  - `chown` → changes **ownership** (which user or group owns the file).
- 

Q2: How do you check the current directory and user?

- To check the **current directory**, use:

```
pwd
```

- To check the **current user**, use:

```
whoami
```