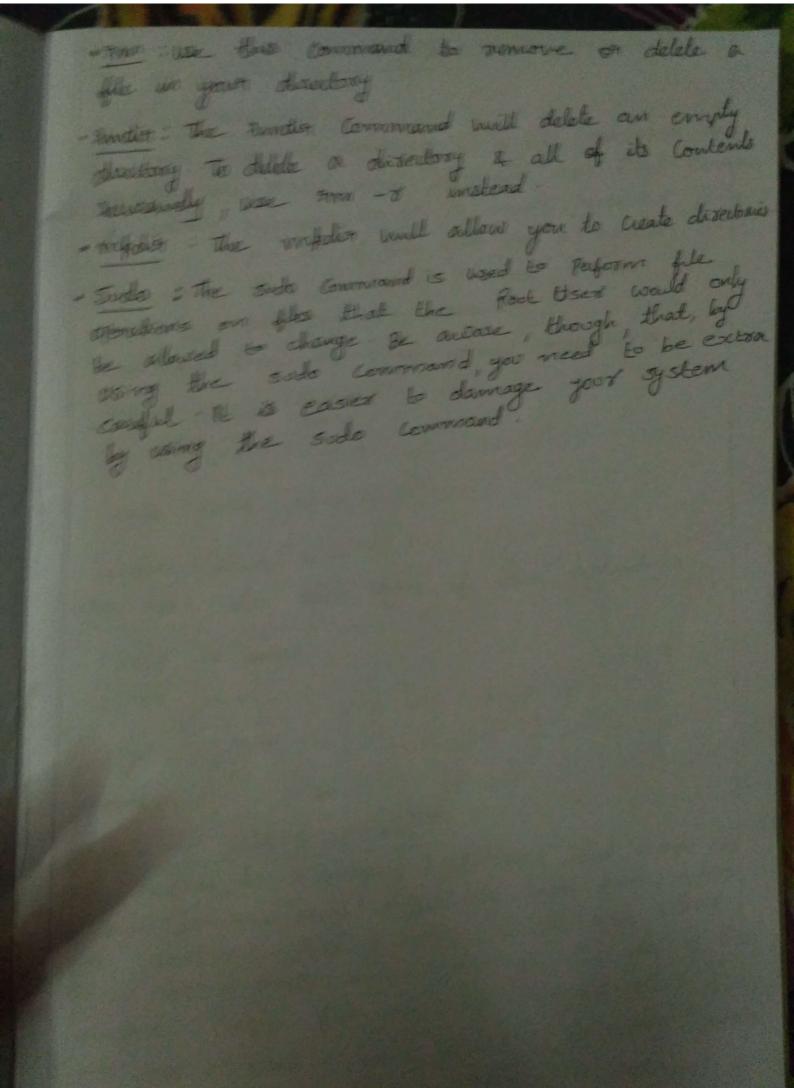# Linux Basic Commands

Aim

To Find the basic commands in Linux.

Basic commands

cd — changes a shell's current working directory.

cp — copies files

find — Searches files matching certain given criteria

less — Displays texts by page

locate — Finds files by name in a file name database

ls — Lists file information or directory contents.

mkdir — Creates new directories.

mv — Moves files to different directories or renames them.

pwd — Displays the name of current working directory

rm — removes files or directories

File & Directory Commands

- The tilde (~) symbol stands for your home directory. If you are user, then the tilde(~) stands for /home/user.

- Pwd : The pwd command will allow you to know in which directory you are located. Pwd stands for Print working directory Pwd in the Desktop directory will show "~/Desktop"

- **ls** : The ls command will show you (list) the files in your current directory

- **cd** : The cd command will allow you to change directories. When you open a terminal you will be in your home directory. To move around the file system you will use cd. Examples:
  - To navigate into the root directory, use "cd/".
  - To navigate to your home directory, use "cd" or "cd~"
  - To navigate up one directory level, use "cd.."
  - To navigate to the previous directory, use "cd-"
  - To navigate through multiple levels of directory at once, specify the full directory path that you want to go to. For example, use "cd/var/www" to go directly to the /www subdirectory of /var/.

- **cp** : The cp command will make a copy of a file for you. Example: "cp file foo" will make an exact copy of "file" and name it "foo", but the file "file" will still be there. If you are copying a directory, you must use "cp-r directory foo".

- **mv** : The mv command will move a file to a different location or will rename a file. Eg: "mv file foo" will rename the "file" to "foo"

- **rm** : Use this command to remove or delete a file in your directory

- **rmdir** : The rmdir command will delete an empty directory. To delete a directory & all of its contents recursively, use rm -r instead

- **mkdir** : The mkdir will allow you to create directories

- **Sudo** : The sudo command is used to perform file operations on files that the Root User would only be allowed to change. Be aware, though, that, by using the sudo command, you need to be extra careful - It is easier to damage your system by using the sudo command.

2. File System

File Permissions

In Linux and Unix, everything is a file. Directories are files, files are of files and devices are files. Devices are usually referred to as a node; however they are still files. All of the files on a system have Permissions that allow or prevent others from viewing, modifying or executing.

There are three types of access restrictions:

Permission	Action	Chmod option
read	(view)	r or 4
write	(edit)	w or 2
execute	(execute)	x or 1

There are also three types of user restrictions:

User	ls output
owner	-rwx . . . .
group	. . . . rwx . .
other	. . . . rwx

Adding & deleting Users

The process for managing local users and groups is straight forward and differs very little from most other GNU/Linux operating system. Ubuntu and other debian based distributions, encourage the use of the "adduser" Package for account management.

• To add a User account, use the following syntax and follow the Prompts to give the account a

Shell Scripting

Aim

To study shell scripting.

Theory

Bash is primarily a scripting language. You can create a bash script by opening your favorite text editor to edit your script and then saving it (typically the .sh file extension is used for your reference).

1) **logged user and his logname**

```
x = $(logname)
echo "currently logged user name & his log : $x"
```

2) **Your current shell**

```
echo $SHELL
```

3) **your home directory**

```
echo $HOME
```

4) **your operating system type**

```
x = $(arch)
echo "your operating system : $x"
```

5) **Your current path setting**

```
echo $PATH
```

6) **your current working directory**

```
echo $Pwd
```

7) **Show currently logged no. of users**

```
echo $users
```