

## Linux Commands

### Basic Commands

**1. pwd** — When you first open the terminal, you are in the home directory of your user. To know which directory you are in, you can use the “**pwd**” command. It gives us the absolute path, which means the path that starts from the root. The root is the base of the Linux file system. It is denoted by a forward slash(/). The user directory is usually something like “/home/username”.

```
nayso@Alok-Aspire:~$ pwd
/home/nayso
```

**2. ls** — Use the “**ls**” command to know what files are in the directory you are in. You can see all the hidden files by using the command “**ls -a**”.

```
nayso@Alok-Aspire:~$ ls
Desktop      itsuserguide.desktop  reset-settings  VCD_Copy
Documents    Music                  School_Resources  Videos
Downloads    Pictures               Students_Works_10
examples.desktop  Public                 Templates
GplatesProject  Qgis Projects         TuxPaint-Pictures
```

**3. cd** — Use the “**cd**” command to go to a directory. For example, if you are in the home folder, and you want to go to the downloads folder, then you can type in “**cd Downloads**”. Remember, this command is case sensitive, and you have to type in the name of the folder exactly as it is. But there is a problem with these commands. Imagine you have a folder named “Raspberry Pi”. In this case, when you type in “**cd Raspberry Pi**”, the shell will take the second argument of the command as a different one, so you will get an error saying that the directory does not exist. Here, you can use a backward slash. That is, you can use “**cd Raspberry\ Pi**” in this case. Spaces are denoted like this: If you just type “**cd**” and press enter, it takes you to the home directory. To go back from a folder to the folder before that, you can type “**cd ..**”. The two dots represent back.

```
nayso@Alok-Aspire:~$ cd Downloads
nayso@Alok-Aspire:~/Downloads$ cd
nayso@Alok-Aspire:~$ cd Raspberry\ Pi
nayso@Alok-Aspire:~/Raspberry Pi$ cd ..
nayso@Alok-Aspire:~$
```

**4. mkdir & rmdir** — Use the **mkdir** command when you need to create a folder or a directory. For example, if you want to make a directory called “DIY”, then you can type “**mkdir DIY**”. Remember, as told before, if you want to create a directory named “DIY Hacking”, then you can type “**mkdir DIY\ Hacking**”. Use **rmdir** to delete a directory. But **rmdir** can only be used to delete an empty directory. To delete a directory containing files, use **rm**.

```
nayso@Alok-Aspire:~/Desktop$ ls
nayso@Alok-Aspire:~/Desktop$ mkdir DIY
nayso@Alok-Aspire:~/Desktop$ ls
DIY
nayso@Alok-Aspire:~/Desktop$ rmdir DIY
nayso@Alok-Aspire:~/Desktop$ ls
nayso@Alok-Aspire:~/Desktop$
```

**5. rm** - Use the **rm** command to delete files and directories. Use "**rm -r**" to delete just the directory. It deletes both the folder and the files it contains when using only the **rm** command.

```
nayso@Alok-Aspire:~/Desktop$ ls
newer.py  New Folder
nayso@Alok-Aspire:~/Desktop$ rm newer.py
nayso@Alok-Aspire:~/Desktop$ ls
New Folder
nayso@Alok-Aspire:~/Desktop$ rm -r New\ Folder
nayso@Alok-Aspire:~/Desktop$ ls
nayso@Alok-Aspire:~/Desktop$
```

**6. touch** — The **touch** command is used to create a file. It can be anything, from an empty txt file to an empty zip file. For example, "**touch new.txt**".

```
nayso@Alok-Aspire:~/Desktop$ ls
nayso@Alok-Aspire:~/Desktop$ touch new.txt
nayso@Alok-Aspire:~/Desktop$ ls
new.txt
```

**7. man & --help** — To know more about a command and how to use it, use the **man** command. It shows the manual pages of the command. For example, "**man cd**" shows the manual pages of the **cd** command. Typing in the command name and the argument helps it show which ways the command can be used (e.g., **cd --help**).

```
TOUCH(1)                                User Commands                                TOUCH(1)

NAME
    touch - change file timestamps

SYNOPSIS
    touch [OPTION]... FILE...

DESCRIPTION
    Update the access and modification times of each FILE to the current
    time.

    A FILE argument that does not exist is created empty, unless -c or -h
    is supplied.

    A FILE argument string of - is handled specially and causes touch to
    change the times of the file associated with standard output.

    Mandatory arguments to long options are mandatory for short options
    too.

    -a      change only the access time

Manual page touch(1) line 1 (press h for help or q to quit)
```

**8. cp** — Use the **cp** command to copy files through the command line. It takes two arguments: The first is the location of the file to be copied, the second is where to copy.

```
nayso@Alok-Aspire:~/Desktop$ ls /home/nayso/Music/  
nayso@Alok-Aspire:~/Desktop$ cp new.txt /home/nayso/Music/  
nayso@Alok-Aspire:~/Desktop$ ls /home/nayso/Music/  
new.txt
```

**9. mv** — Use the **mv** command to move files through the command line. We can also use the **mv** command to rename a file. For example, if we want to rename the file “**text**” to “**new**”, we can use “**mv text new**”. It takes the two arguments, just like the **cp** command.

```
nayso@Alok-Aspire:~/Desktop$ ls  
new.txt  
nayso@Alok-Aspire:~/Desktop$ mv new.txt newer.txt  
nayso@Alok-Aspire:~/Desktop$ ls  
newer.txt
```

**10. locate** — The **locate** command is used to locate a file in a Linux system, just like the search command in Windows. This command is useful when you don't know where a file is saved or the actual name of the file. Using the **-i** argument with the command helps to ignore the case (it doesn't matter if it is uppercase or lowercase). So, if you want a file that has the word “hello”, it gives the list of all the files in your Linux system containing the word “hello” when you type in “**locate -i hello**”. If you remember two words, you can separate them using an asterisk (\*). For example, to locate a file containing the words “hello” and “this”, you can use the command “**locate -i \*hello\*this**”.

```
nayso@Alok-Aspire:~$ locate newer.txt  
/home/nayso/Desktop/newer.txt  
nayso@Alok-Aspire:~$ locate *DIY*Hacking*  
/home/nayso/DIY Hacking
```