

- Ubuntu Linux
- Red Hat Enterprise Linux
- Linux Mint
- Debian
- Fedora

About 90% of the internet is powered by Linux servers. This is because Linux is fast, secure, and free!

Linux Shell or “Terminal”

Basically, a shell is a program that receives commands from the user and gives it to the OS to process, and it shows the output. Linux's shell is its main part. Its distros come in GUI (graphical user interface), but basically, Linux has a CLI (command line interface). In this tutorial, we are going to cover the basic commands that we use in the shell of Linux.

Linux Commands

Basic Commands

1. pwd -To know which directory you are in; you can use the “**pwd**” command (**print working directory**).

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**”.

3. cd — Use the “**cd**” command to go to a directory. you can type in “**cd Downloads**” for example if you want to go to the Download directory.

- If you want to go to a directory that has a double name such as **Raspberry pi**, you can use “**cd Raspberry\ Pi**” in this case.

-To go back from a folder to the folder before that, you can type “**cd..**” .

- If you just type “**cd**” and press enter, it takes you to the home directory.

```
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.

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.

```
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.

8. cp — Use the **cp** command to copy files through the command line.

```
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.

11. echo — The "echo" command helps us move some data, usually text into a file. For example, if you want to create a new text file or add to an already made text file, you just need to type in, "echo hello, my name is alok >> new.txt".

12. cat — Use the **cat** command to display the contents of a file. It is usually used to easily view programs.

```
nayso@Alok-Aspire:~/Desktop$ echo hello, my name is alok >> new.txt
nayso@Alok-Aspire:~/Desktop$ cat new.txt
hello, my name is alok
nayso@Alok-Aspire:~/Desktop$ echo this is another line >> new.txt
nayso@Alok-Aspire:~/Desktop$ cat new.txt
hello, my name is alok
this is another line
```

13. nano, vi, jed — **nano** and **vi** are already installed text editors in the Linux command line. The **nano** command is a good text editor that denotes keywords with color and can recognize most languages. And **vi** is simpler than **nano**. You can create a new file or modify a file using this editor. For example, if you need to make a new file named "**check.txt**", you can create it by using the command "**nano check.txt**". You can save your files after editing by using the sequence Ctrl+X, then Y (or N for no). In my experience, using **nano** for HTML editing doesn't seem as good, because of its color, so I recommend **jed** text editor. We will come to installing packages soon.

14. sudo — A widely used command in the Linux command line, **sudo** stands for "SuperUser Do".

15. tar — Use **tar** to work with tarballs (or files compressed in a tarball archive) in the Linux command line. It has a long list of uses. It can be used to compress and uncompress different types of tar archives like **.tar**, **.tar.gz**, **.tar.bz2**, etc. It works on the basis of the arguments given to it. For example, "**tar -cvf**" for creating a **.tar** archive, **-xvf** to untar a tar archive, **-tvf** to list the contents of the archive, etc. Since it is a wide topic

the below example command will create a **tar** archive file **tecmint-14-09-12.tar** for a directory **/home/tecmint** in the current working directory. See the example command in action.

```
# tar -cvf tecmint-14-09-12.tar /home/tecmint/
```

16. apt-get — Use **apt** to work with packages in the Linux command line. Use **apt-get** to install packages. This requires root privileges, so use the **sudo** command with it. For example, if you want to install the text editor **jed** (as I mentioned earlier), we can type in the command “**sudo apt-get install jed**”. Similarly, any packages can be installed like this. It is good to update your repository each time you try to install a new package. You can do that by typing “**sudo apt-get update**”. You can upgrade the system by typing “**sudo apt-get upgrade**”.

17. curl - curl is a command-line utility for **transferring data** from or to a server designed to work without user interaction. With curl, you can download or upload data using one of the supported protocols including HTTP, HTTPS, SCP , SFTP , and FTP .

- **Install Curl on Ubuntu and Debian**

```
sudo apt update
```

```
sudo apt install curl
```

- The syntax for the curl command is as follows:

```
curl [options] [URL...]
```

Tips and Tricks for Using Linux Command Line

- You can use the **clear** command to clear the terminal if it gets filled up with too many commands.
- **TAB** can be used to fill up in terminal. For example, you just need to type “**cd Doc**” and then **TAB** and the terminal fills the rest up and makes it “**cd Documents**”.
- **Ctrl+C** can be used to stop any command in terminal safely. If it doesn't stop with that, then **Ctrl+Z** can be used to force stop it.
- You can exit from the terminal by using the **exit** command.
- You can power off or reboot the computer by using the command **sudo halt** and **sudo reboot**.

Directories

/bin - directory contains the essential user (programs) that must be present when the system is mounted in single-user mode. Applications such as Firefox are stored in /usr/bin, while important system programs and utilities such as the bash shell are located in /bin.

The /boot directory contains the files needed to boot the system

/dev — Device Files

Linux exposes devices as files, and the `/dev` directory contains a number of special files that represent devices. These are not actual files as we know them, but they appear as files — for example, `/dev/sda` represents the first SATA drive in the system.

/etc — Configuration Files

The `/etc` directory contains configuration files, which can generally be edited by hand in a text editor. Note that the `/etc/` directory contains system-wide configuration files — user-specific configuration files are located in each user's home directory.

/home — Home Folders

The `/home` directory contains a home folder for each user. For example, if your username is bob, you have a home folder located at `/home/bob`. This home folder contains the user's data files and user-specific configuration files.

/lib — Essential Shared Libraries

The `/lib` directory contains libraries needed by the essential binaries in the `/bin` and `/sbin` folder. Libraries needed by the binaries in the `/usr/bin` folder are located in `/usr/lib`.

/opt — Optional Packages

The `/opt` directory contains subdirectories for optional software packages. It's commonly used by proprietary software that doesn't obey the standard file system hierarchy — for example, a proprietary program might dump its files in `/opt/application` when you install it.

/sbin — System Administration Binaries

The `/sbin` directory is similar to the `/bin` directory. It contains essential binaries that are generally intended to be run by the root user for system administration.

/usr — User Binaries & Read-Only Data

The /usr directory contains applications and files used by users, as opposed to applications and files used by the system.

/var — Variable Data Files

The /var directory is the writable counterpart to the /usr directory, which must be read-only in normal operation. Log files and everything else that would normally be written to /usr during normal operation are written to the /var directory.

References

- 1- <https://www.howtogeek.com/117435/htg-explains-the-linux-directory-structure-explained/>
- 2- <https://maker.pro/linux/tutorial/basic-linux-commands-for-beginners>
- 3- <https://linuxize.com/post/curl-command-examples/>