

COMMUNITY FAVORITES

Save up to 70% on user favorite books, eBooks, and video training. [Learn more.](#)

the trusted technology learning source

[Home](#) > [Blogs](#) > [The 10 Most Important Linux Commands](#)

The 10 Most Important Linux Commands



By [Brad Yale](#)

Sep 22, 2014

Topics: [Information Technology](#), [Open Source](#), [Linux/UNIX/Open Source](#), [Linux/UNIX/Open Source Operating Systems](#), [Operating Systems](#), [Server](#)

I get it.

You're new to the Linux Command Line and truth be told, you might be a little intimidated. Coming from the comfort of a PC or Mac desktop, the Linux Command Line (CLI) looks nothing like what you normally use. The [Linux](#) CLI is dark, it's secretive, it's bare bones minimal and it's anything but friendly to newcomers. And yet, the Linux CLI is highly useful, essential to using your Linux box or virtual machine and can, if done right, provide you with more insight and practical use than the Windows or Mac desktop ever could.

For those reasons, from someone who was also once a Linux newb, I present the top ten Linux CLI commands you need to master for basic Linux CLI comprehension.

This list will not make your a [Linux System Admin](#) however it will get you on your way with your foot in the ocean.

Before my top ten Linux CLI list, I want to preface this countdown by stating the commands presented are being presented to give you a working knowledge of how to get around your Linux terminal from the shell. This list is meant to serve as an introduction to basic yet highly vital [Linux](#) CLI commands.

There are a ton of other commands - more sophisticated commands - which you will one day learn. But for now, like learning to add before you learn to multiply, this list will provide you with your [Linux CLI command](#) foundation.

At the end of the list, some vital information you need to use the Linux CLI.

1. ls

The ls command - the list command - functions in the [Linux terminal](#) to show all of the major directories filed under a given file system. For example, the command:

ls /applications

...will show the user all of the folders stored in the overall applications folder.

The ls command is used for viewing files, folders and directories.

2. cd

Become an InformIT Member

Take advantage of special member promotions, everyday discounts, quick access to saved content, and more! [Join Today.](#)

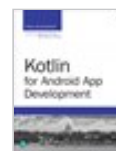
Other Things You Might Like



[Windows 10 Inside Out, 3rd Edition](#)

By [Ed Bott](#), [Craig Stinson](#)

Book \$47.99



[Kotlin for Android App Development](#)

By [Peter Sommerhoff](#)

Book \$35.99



[HTML, CSS, and JavaScript All in One: Covering HTML5, CSS3, and ES6, Sams Teach Yourself, 3rd Edition](#)

By [Julie C. Meloni](#), [Jennifer Kyrnin](#)

Book \$31.99

The `cd` command - change directory - will allow the user to change between file directories. As the name command name suggest, you would use the `cd` command to circulate between two different directories. For example, if you wanted to change from the home directory to the Arora directory, you would input the following command:

`cd/arora/applications`

As you might have noted, the path name listed lists in reverse order. Logically **`cd/arora/applications`** reads change to the arora directory which is stored in the applications directory. All Linux commands follow a logical path.

3. mv

The `mv` command - move - allows a user to move a file to another folder or directory. Just like dragging a file located on a PC desktop to a folder stored within the "Documents" folder, the `mv` command functions in the same manner. An example of the `mv` command is:

`mv/arora/applications/majorapps /arora/applications/minorapps`



The first part of the command **`mv/arora/applications/majorapps`** lists the application to be moved. In this case, arora. The second part of the command **`/arora/applications/minorapps`** lists where arora will be moved to - from majorapps to minorapps.

4. man

The `man` command - the manual command - is used to show the manual of the inputted command. Just like a film on the nature of film, the `man` command is the meta command of the [Linux CLI](#). Inputting the `man` command will show you all information about the command you are using. An example:

`man cd`

The inputting command will show the manual or all relevant information for the change directory command.

5. mkdir

The `mkdir` - make directory - command allows the user to make a new directory. Just like making a new directory within a PC or Mac desktop environment, the `mkdir` command makes new directories in a Linux environment. An example of the `mkdir` command

mkdir testdirectory

The example command made the directory "testdirectory".

6. rmdir

The rmdir - remove directory - command allows the user to remove an existing command using the Linux CLI. An example of the rmdir command:

rmdir testdirectory



The example command removed the directory "testdirectory".

It should be noted: both the mkdir and rmdir commands make and remove directories. They do not make files and they will also not remove a directory which has files in it. The mkdir will make an empty directory and the rmdir command will remove an empty directory.



7. touch

The touch command - a.k.a. the make file command - allows users to make files using the Linux CLI. Just as the mkdir command makes directories, the touch command makes files. Just as you would make a .doc or a .txt using a PC desktop, the touch command makes empty files. An example of the touch command:

touch testfile.txt

The example touch command effectively created the file testfile.txt. As noted by the extension, the file created is a .txt or text file. To equate, a .txt file in Linux is akin to a .txt notebook file within a Windows or Mac OS.



8. rm

The `rm` command - remove - like the `rmdir` command is meant to remove files from your Linux OS. Whereas the `rmdir` command will remove directories and files held within, the `rm` command will delete created files. An example of the `rm` command:

```
rm testfile.txt
```

The aforementioned command removed `testfile.txt`. Interestingly, whereas the `rmdir` command will only delete an empty directory, the `rm` command will remove both files and directories with files in it. This said, the `rm` command carries more weight than the `rmdir` command and should be used with more specificity.

9. locate

The `locate` - a.k.a. `find` - command is meant to find a file within the Linux OS. If you don't know the name of a certain file or you aren't sure where the file is saved and stored, the `locate` command comes in handy. A `locate` command example:

```
locate -i *red*house**city*
```

The stated command will locate a file with a file name containing "Red", "House" and "City". A note on the input: the use of "-i" tells the system to search for a file unspecific of capitalization (Linux functions in lower case). The use of the asterisk "*" signifies searching for a wildcard. A wildcard tells the system to pull any and all files containing the search criteria.

By specifying `-i` with wildcards, the `locate` CLI command will pull back all files containing your search criteria effectively casting the widest search net the system will allow.

10. clear

The `clear` command does exactly what it says. When your Linux CLI gets all mucked up with various readouts and information, the `clear` command clears the screen and wipes the board clean. Using the `clear` command will take the user back to the start prompt of whatever directory you are currently operating in. To use the `clear` command simply type **clear**.

Getting to Know Linux: Vital Information

Having a basic understanding on Linux CLI commands will allow any user to navigate around the Linux shell. This said, there are some basic things about Linux you need to know to more fully operate in the shell. These basics are as follows:

- **All Linux operating systems function in lower case.** The basic idea of Linux is to utilize a simple easy to use operating system. The use of lower case comes out of this. While you can name files, folders and directories using upper case, the system functions in lower case. This means unless you

specify -i (negate case lock), all files, folders and directories named with an upper case will not be shown. Thus, the command

locate thekillersadustlandfairytale.mp3

Will not locate the file TheKillersADustLandFairyTale.mp3

In Linux, upper and lower case matter.

- **Be very careful using the rm command.** The rm command, as noted above, carries more weight than the rmdir command. Using the rm command can wipe out entire directories full of files. Be careful using it. Moreover, if a buddy jokingly tells you to input the command:

rm -rf/

Do not do it. The rm -rf/ command means remove (rm) - recursive (r) force (f) home (/). Spelled out logically, the rm -rf/ command will delete every folder, file and directory within your Linux OS. It is the equivalent of wiping your entire hard drive clean. Use the rm-rf/ command at your own peril.

- **/ means root.** For the vast majority of PC's the home prompt in the Command Prompt (CMD) is some variation on:

The PC user in command prompt will start from the C:\ hard drive. In Linux your CLI starting point - your root directory - is /. / represents your starting point. / is where all other files, folders and directories are stored within.

- **Passwords are kept in the dark.** The first time I logged into a Linux server, I instantly became confused when prompted for my password. This is because when you type in your password into the Linux CLI, the password is kept dark. While typing, you will see nothing. The Linux CLI operates with the assumption that the user typing in the password knows what he/she is typing and thus, for security reasons, has no need to view it while it is input.

Don't freak out when you do not see your password or little asterisks to hide your input. The Linux system is recording every key tap.

- **Master Linux foundations before advancing.** Just like learning new branches of Mathematics, to build higher, your foundation must be strong. Linux operates in the same fashion. If you do not master Linux basics, you will quickly become confused when trying to complete complex tasks and will, like many first time users do, become frustrated and quit. To write a new script, to install a script or to install a basic word editor, you first must know how to move between directories, how to copy (cp command) files and how to make/remove files and directories.

Before you move to more advanced Linux tasks, master the basics.

- **Remember, Linux Web Operations are free.** Maybe the best thing about Linux is that it an open source platform meaning the operating systems and all [Linux web operations](#) you may perform are free. Don't get scammed into buying an operating system. It's free.

- **Pick a more User Friendly Linux OS.** The last bit of advice I can dole out is if you are a first time Linux user, pick an operating system which is considered "beginner friendly". Most users consider [Ubuntu](#), Fedora and Mint to all fit the bill. I'd recommend choosing one and working from there.

Remember, if you like this content and want to chat about it, you can reach me at the following social spaces:

- Twitter: [@bleibowi](#)
- Instagram: [@byalie](#)
- LinkedIn: [Brad Yale](#)
- Google +: [Brad Yale](#)

Share |    