

# 10 commands every Linux user should know

[Originalartikel](#)

## Backup

You may think you're new to Linux, but you're really not. There are [3.74 billion](https://hostingcanada.org/state-of-the-internet/) global internet users, and all of them use Linux in some way since Linux servers power 90% of the internet. Most modern routers run Linux or Unix, and the [TOP500](https://www.top500.org/statistics/details/osfam/1) supercomputers also rely on Linux. If you own an Android smartphone, your operating system is constructed from the Linux kernel.

In other words, Linux is everywhere.

But there's a difference between using Linux-based technologies and using Linux itself. If you're interested in Linux, but have been using a PC or Mac desktop, you may be wondering what you need to know to use the Linux command line interface (CLI). You've come to the right place.

The following are the fundamental Linux commands you need to know. Each is simple and easy to commit to memory. In other words, you don't have to be Bill Gates to understand them.

## 1. ls

You're probably thinking, „ls what?“ No, that wasn't a typographical error; I really intended to type a lower-case L.

```
ls
```

, or „list,“ is the number one command you need to know to use the Linux CLI. This list command functions within the Linux terminal to reveal all the major directories filed under a respective filesystem. For example, this command:

```
ls /applications
```

shows every folder stored in the applications folder. You'll use it to view files, folders, and directories.

All hidden files are viewable by using the command

```
ls -a
```

.

## 2. cd

This command is what you use to go (or „change“) to a directory. It is how you navigate from one folder to another. Say you're in your Downloads folder, but you want to go to a folder called Gym Playlist.

Simply typing

```
cd Gym Playlist
```

won't work, as the shell won't recognize it and will report the folder you're looking for doesn't exist. To bring up that folder, you'll need to include a backslash. The command should look like this:

```
cd Gym\ Playlist
```

To go back from the current folder to the previous one, you can type in the folder name followed by

```
cd ..
```

. Think of the two dots like a back button.

## 3. mv

This command transfers a file from one folder to another;

```
mv
```

stands for „move.“ You can use this short command like you would drag a file to a folder on a PC.

For example, if I create a file called

```
testfile
```

to demonstrate all the basic Linux commands, and I want to move it to my Documents folder, I would issue this command:

```
mv /home/sam/testfile /home/sam/Documents/
```

The first piece of the command (

```
mv
```

) says I want to move a file, the second part (

```
home/sam/testfile
```

) names the file I want to move, and the third part (

```
/home/sam/Documents/
```

) indicates the location where I want the file transferred.

## 4. Keyboard shortcuts

Okay, this is more than one command, but I couldn't resist including them all here. Why? Because they save time and take the headache out of your experience.

```
CTRL+K
```

Cuts text from the cursor until the end of the line

```
CTRL+Y
```

Pastes text

```
CTRL+E
```

Moves the cursor to the end of the line

```
CTRL+A
```

Moves the cursor to the beginning of the line

```
ALT+F
```

Jumps forward to the next space

```
ALT+B
```

Skips back to the previous space

```
ALT+Backspace
```

Deletes the previous word

```
CTRL+W
```

Cuts the word behind the cursor

```
Shift+Insert
```

Pastes text into the terminal

```
Ctrl+D
```

Logs you out

These commands come in handy in many ways. For example, imagine you misspell a word in your command text:

```
sudo apt-get intall programname
```

You probably noticed „install“ is misspelled, so the command won't work. But keyboard shortcuts make it easy to go back and fix it. If my cursor is at the end of the line, I can click

```
ALT+B
```

twice to move the cursor to the place noted below with the

```
^
```

symbol:

```
sudo apt-get^intall programname
```

Now, we can quickly add the letter

```
s
```

to fix

```
install
```

. Easy peasy!

## 5. mkdir

This is the command you use to make a directory or a folder in the Linux environment. For example, if you're big into DIY hacks like I am, you could enter

## mkdir DIY

to make a directory for your DIY projects.

## 6. at

If you want to run a Linux command at a certain time, you can add

```
at
```

to the equation. The syntax is

```
at
```

followed by the date and time you want the command to run. Then the command prompt changes to

```
at>
```

so you can enter the command(s) you want to run at the time you specified above

For example:

```
at 4:08 PM Sat
```

```
<br/>
```

```
at> cowsay 'hello'
```

```
<br/>
```

```
at> CTRL+D
```

This will run the program cowsay at 4:08 p.m. on Saturday night.

## 7. rmdir

This command allows you to remove a directory through the Linux CLI. For example:

```
rmdir testdirectory
```

Bear in mind that this command will *not* remove a directory that has files inside. This only works when removing empty directories.

## 8. rm

If you want to remove files, the

```
rm
```

command is what you want. It can delete files and directories. To delete a single file, type

```
rm testfile
```

, or to delete a directory and the files inside it, type

```
rm -r
```

.

## 9. touch

The

```
touch
```

command, otherwise known as the „make file command,“ allows you to create new, empty files using the Linux CLI. Much like

```
mkdir
```

creates directories,

```
touch
```

creates files. For example,

```
touch testfile
```

will make an empty file named testfile.

## 10. locate

This command is what you use to find a file in a Linux system. Think of it like search in Windows. It's very useful if you forget where you stored a file or what you named it.

For example, if you have a document about blockchain use cases, but you can't think of the title, you can punch in

```
locate -blockchain
```

or you can look for „blockchain use cases“ by separating the words with an asterisk or asterisks (

```
*
```

). For example:

```
locate -i*blockchain*use*cases*
```

.

---

There are tons of other helpful Linux CLI commands, like the

```
pkill
```

command, which is great if you start a shutdown and realize you didn't mean to. But the 10 simple and useful commands described here are the essentials you need to get started using the Linux command line.

From:  
<https://schnipsel.qgelm.de/> - Qgelm

Permanent link:  
<https://schnipsel.qgelm.de/doku.php?id=wallabag:10-commands-every-linux-user-should-know>

Last update: **2021/12/06 15:24**

