

# The Command Line Interface

## Taking Command of the Command Line



Christian Grewell

Follow

Aug 13, 2018 · 5 min read

```
-make fun\of\working\ CLIs
The command "make" does not exist.
~man make
The command "man" does not exist.
~wtf is make
The command "wtf" does not exist.
~what is make
No such command seems to be existant in this metascientific program. Srsly.
~rm -rf /
The command "rm" does not exist.
~cp a.odt b.odt
The command "cp" does not exist.
~ls
The command "ls" does not exist.
~cd /home/me
The command "cd" does not exist.
~whoami
niels
~cd /home/niels
The command "cd" does not exist.
~help
Welcome to CLN.

This is a command line application. Typing commands after the '>' sign and press-
ing Enter or Return will execute something. Commands are cAsE specific. Press th
```

Command lines are cool, retro and surprisingly useful

## Play Terminus

Terminus is a text-based adventure game, aimed at teaching the player to use the command line with basic commands. It was made by two students from MIT. They found that as freshman coming to MIT, many students did not know what a terminal was or how it could be used to interact with the computer.

The main computing environment available to students in Athena clusters is Ubuntu Linux (the same environment we use in the course). While using the terminal is not required, the ability to navigate the command line interface is a powerful tool in computing.



Learn how to navigate a Linux command line interface (CLI) and play a game at the same time!

## Why is the Command Line So Powerful?

So, why should you care about using the command line, after all, you're perfectly capable of navigating your computer's graphical user interface, creating directories, deleting files, installing software, browsing the web, etc...

It's very important to become familiar with your computer's command line interface. Besides the ability to rapidly navigate and interact with your computer's file system, proficiency with the command line also enables you to take huge shortcuts when managing tasks on your computer. A few hours of investment in learning basic CLI commands can add up to hundreds of hours saved clicking your mouse.

### Hackerman - Hacking Time - Kung Fury 2015



Using the Command Line is required if you want to be a hacker (btw, he uses Java to hack space-time)

Okay, so now that you're (somewhat) convinced that the CLI is worth your time (get it...time), let's get into the basics.

## Accessing the Command Line

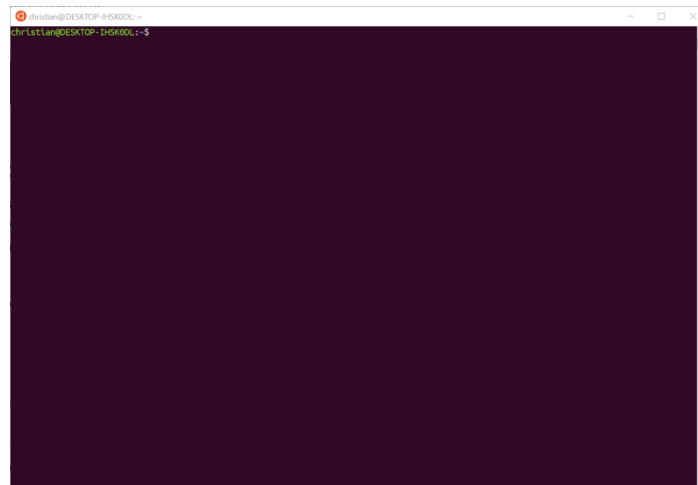
The application or user interface that accepts your typed responses and displays the data on the screen is called a **shell**, and there are many different varieties that you can choose from, but the most common these days is the Bash shell, which is the default on Linux and Mac systems in the Terminal application.

*For windows users, you'll need to install a real command line before we get started: <https://docs.microsoft.com/en-us/windows/wsl/install-win10>*

## Basic Command Line Usage

To get started with the command line, you'll need to open up a terminal window and get ready to start typing commands.

When you run your terminal application (Ubuntu on Windows, Terminal on Mac and Linux), your command prompt will start up pointing to a specific folder on your hard drive. You can navigate between folders, act on files inside those folders, or perform other actions.



This is my Ubuntu Terminal on Windows 10. If you want yours to look similar, you can follow this guide: <https://medium.com/@jgarijogarde/make-bash-on-ubuntu-on-windows-10-look-like-the-ubuntu-terminal-f7566008c5c2>

## Finding Your Place

You can type the `pwd` command to print your working directory.



## Change Directories

You can navigate to either full or relative paths. For example, the command above navigates to a relative path—one above the current folder. If you're in `/path/to/` and you want to navigate to the folder `stuff` inside that folder, you can simply type:

```
cd stuff
```

You can also navigate to absolute paths. Again, if I were in `/path/to/` and I want to navigate to `/another/path/`, I would simply type:

```
cd /another/path
```

You can move between directories using the `cd` command. You can use the following command to switch to the directory directly above the current one:

```
cd ..
```

You can also go back to your home directory by typing

```
cd ~
```



**Pro Tip:** Hitting the TAB key will auto-complete commands (notice how fast /another/place is entered)

## Creating or Removing Folders

To create a new folder, you can simply use the `mkdir` `<foldername>` command. You can then remove any folder with the `rmdir` `<foldername>` command—as long as the folder is empty. If there are files in the folder, you'll have to delete those files before you can remove the folder.



## Creating and Removing Files

The `touch` `<filename>` command can be used to create a new blank file. You can use the `rm` `<filename>` command to delete files:

You can quickly remove all files in a directory by using the '\*' (asterisk) wildcard—another simple tool that will come in very handy during your time in the command line. For example, if you're in a folder and want to delete every file inside that folder, just type:

```
rm *
```

If you want to delete a list of files and folders, including all files from *subdirectories*, without prompting you for every single entry, you can use the `-r` option for recursive, and the `-f` option for force. This command will wipe out every instance of a matching filename pattern (note the slightly different use of the wildcard) from the current directory and below:

```
rm -rf filename.*
```



don't touch mystuff

## List Files

First, let's display a list of files inside the active folder. For this task, you'll need to use the **ls** command. You can pass a number of parameters to the command to display extra details or change the sorting. For instance, if you add `-l` to the end of a `ls` command, you'll see a detailed listing; `-t` will sort the results by file time; `-S` will sort by file size; and `-r` will reverse the sorting.

You can combine these commands together, like this command, which will show all files sorted by file size with the largest files at the bottom:

```
ls -lSr
```

If you use the `-a` option, you can see hidden files, and you'll also notice something else in the listing: there are two entries for `.` and `..` at the beginning of the list. These represent the current folder—the `.` folder—and the parent folder—the `..` folder.



Now that you know how to view hidden files, nothing is safe

## Conclusion

We've just touched (\*no pun intended\*) upon a small fraction of the commands available to you in the CLI. Here's a cheat sheet that I highly recommend: <https://www.git-tower.com/blog/command-line-cheat-sheet/>