Lab 1: Command Line Interface

Long ago, before operating systems with graphical user interfaces (GUIs), everyone used command line interfaces (CLIs). For example, instead of starting a program by clicking the icon for your program in the Launchpad, you instead typed the name of your program. And instead of clicking on menus in your program to adjust settings, open files, etc, you could specify these things as command line arguments.

Hence, the computer that processed data or performed operations was separate from the tool that gave it the instructions to do the processing. There was:

- The terminal: which was used to send commands to the computer and

- The computer: the hardware that processed the commands

Today, there are computers that can both provide commands AND perform the computation, and these computers have graphical user interfaces (known as GUIs) that make it easy to perform tasks.

Although most of us are accustomed to use GUI, terminal (command line) is something every developer should learn and implement into their daily routine. It has become a Swiss Army knife of features behind deceptively simple commands, which allow you to gain greater control of your system, become more productive, and much more. You can also use the terminal to launch and execute open reproducible science tools such as Jupyter Notebook, Python, and git.

1   Why Command Line Interface?

1. Most command line interface tasks only require the keyboard, while GUI systems require both the mouse and the keyboard. Therefore, command line users usually do not have to switch their hands between two places.

2. Furthermore, command line interface usually requires just few lines of code to perform a complicated task. They can be easily automated through scripts, which are essentially recipes of text-based commands. For example, you can write scripts to automate daily, time-consuming tasks, and even quickly commit and push code to a Git repository with just a few simple commands

3. Command line interface also uses fewer resources than a GUI system as a GUI system will load icons, fonts, I/O drivers and other resources.

4. They allow for more detail when running some programs, as you can add modifiers to specify exactly how you want your program to run. .

Because of above reasons, command line users may be able to complete most tasks relatively faster than a GUI user. Command line users can create scripts and save time, while GUI users can do the same with facilities such as creating shortcuts. Although new users might have to learn how to operate the mouse, GUI is easier to pick up than using the Command line. Unlike GUI, Command Line users need a fair amount of familiarity and need to memorize a number of commands in order to get their jobs done smoothly. But, a command line user has much more control of the file and operating system. And for performing some advanced tasks, command line may be the only option (sometimes). GUI systems inherently make it easy to multitask, by providing graphical means of monitoring several things (process) at once (many command line environments offer multitasking, but it is harder to view several things at once).

In this post we'll look at the Bash Shell (Bourne Again SHell), which is a command-line interface (CLI) and is currently the most widely used shell. This is a light introduction into the most popular commands, when you're most likely to use them, and how to extend them with options.

> If you don't know the command line, you're not using your computer to its full potential.

2   Shell and Bash

Shell is the primary program that computers use to receive code (i.e. commands) and return information produced by executing these commands (i.e. output). These commands can be entered and executed via the terminal. This allows you to control your computer by typing in commands with a keyboard, instead of using buttons or drop down menus in a GUI with a mouse/keyboard.

Bash (also known as the "Bourne Again SHell") is an implementation of Shell and allows you to efficiently perform many tasks. For example, you can use Bash to perform operations on multiple files quickly via the command line. You can also write and execute scripts in Bash, just like you can in R or Python, that can be executed across different operating systems.

3   Working on a Terminal

Terminal is the command line interface (CLI) that gives you access to Bash. There are many different terminal programs, and thus, the terminal that you use on your computer will vary according to your operating system. For instance, Mac as well as many Linux computers have a default terminal program installed that provides access to Bash. However, the default terminal on Windows computers does not provide access to Bash.
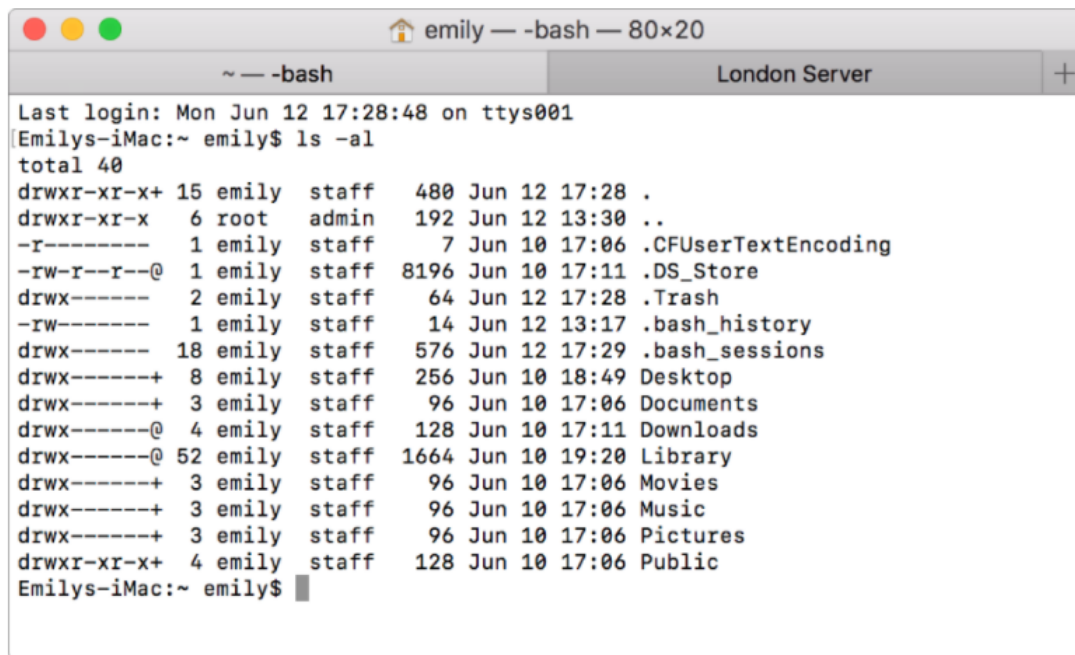
Windows users will need to install a customized terminal. Various ways of doing that have been described below.

Throughout the textbook, the command line environment that you use to access Bash will be referred to as the terminal.

3.1   Open a Terminal Session On Mac (OS X)

You can use the program called Terminal, which uses the Bash implementation of Shell and is installed natively on the Mac OS.

You can open Terminal by finding and launching it from Spotlight (or from /Applications/Utilities).

Figure 1: This is what the Terminal on Mac looks like. Source: Apple.com.

3.2   Open a Terminal Session On Linux

Many Linux computers use the Bash implementation of Shell, which you will learn to test for in the section below.

You can open the program called Terminal (or Terminal Emulator) by finding and launching it from your list of programs.
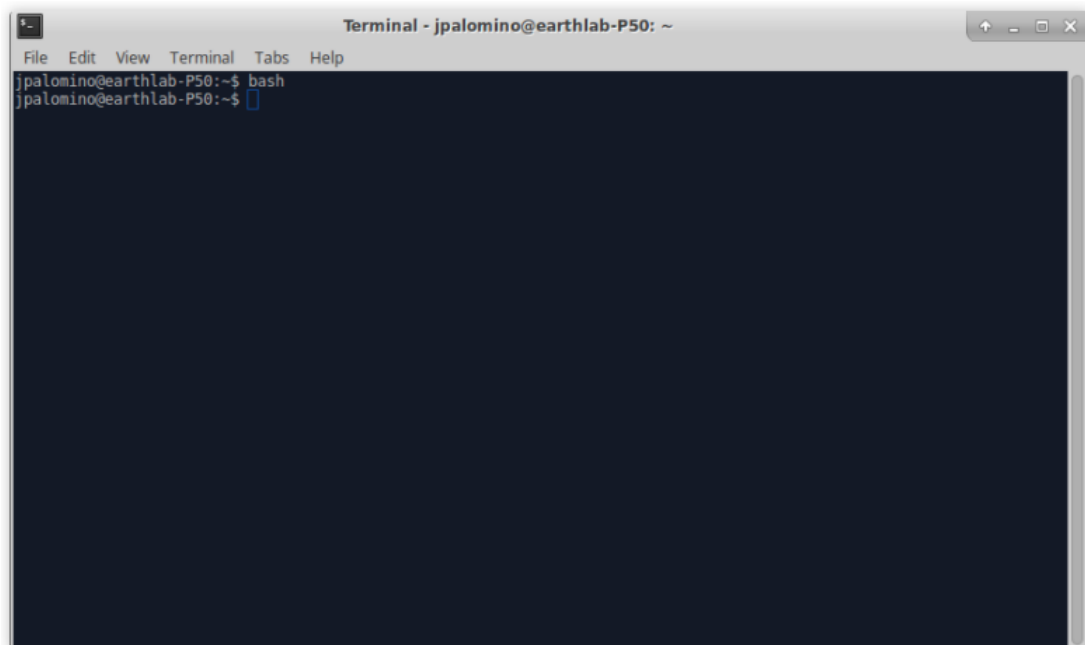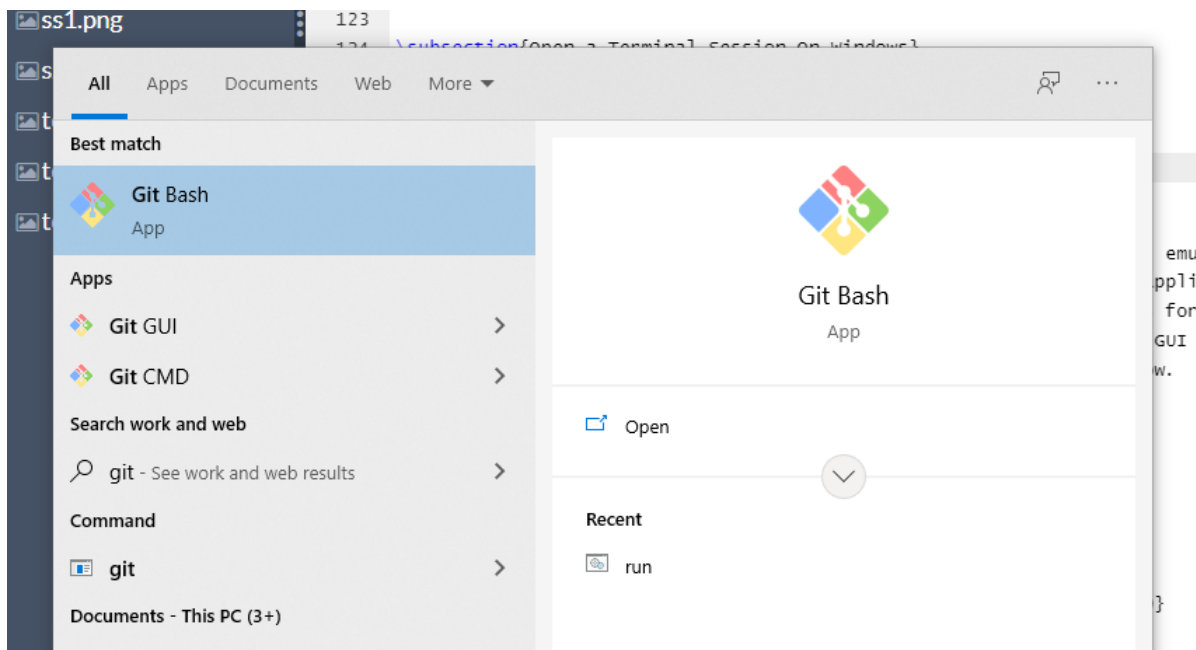
Figure 2: This is what the Terminal on Linux looks like.

3.3 Open a Terminal Session On Windows

You can choose any of the following strategies.

3.3.1 git bash

Install git Windows from `https://git-scm.com/download/win`. Search for git bash and run it.

3.3.2 Install Cygwin

Cygwin is a useful utility that provides a UNIX/Linux-like shell emulated over a Windows environment. It makes it easy to port certain specific applications on Windows and allows you to develop and use Linux-based scripts on Windows for various tasks. Its installation is as simple as any Windows-based program with GUI based setup. Once you install it, you can open Cygwin terminal, as shown below.

3.3.3 "Windows Subsystem For Linux": Only on Windows-10

To install WSL using Setting on Windows 10, use these steps:

1. Open Settings.

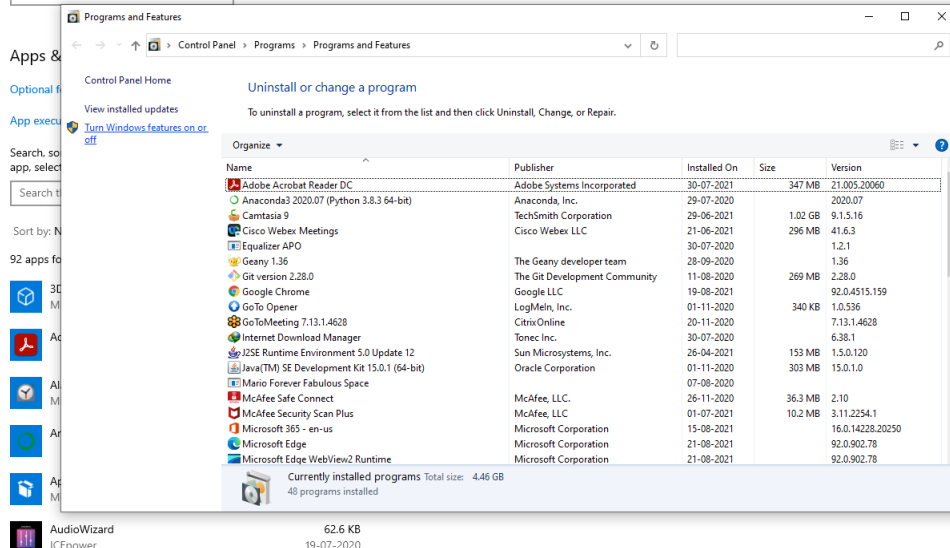2. Click on Apps.

3. Under the "Related settings" section, click the Programs and Features option.

4. Click the Turn Windows features on or off option from the left pane.

5. Check the Windows Subsystem for Linux option.

6. Click the OK button.

7. Click the Restart now button.

8. Install "Ubuntu" from Microsoft Store.

9. Search Ubuntu in the search bar and open it.
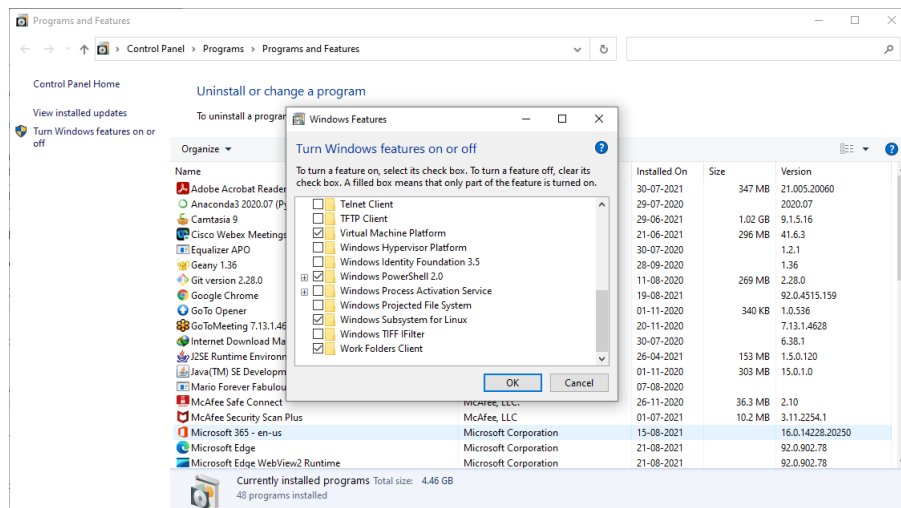
4 Commands to run

Google for the following commands and run them using Bash shell.

1. clear
2. exit
3. history
4. ls
5. pwd
6. mkdir
7. rmdir
8. cp
9. mv
10. rm
11. touch
12. echo
13. cat
14. diff
15. chmod
16. grep
17. locate
18. which
19. cat
20. ps
21. top
22. passwd
23. sudo
24. pico/nano/vim
25. read

To explore a command further, use the "man" command.

5 References

1. https://www.earthdatascience.org/courses/intro-to-earth-data-science/open-reproducible-science/bash/

2. https://pakcivilengrcom.files.wordpress.com/2015/12/gui-and-cli.pdf

3. https://www.cs.princeton.edu/courses/archive/spr15/cos126/precepts/CommandPromptTutorial.pdf

4. https://www.windowscentral.com/install-windows-subsystem-linux-windows-10