

OS_setup

September 1, 2020

1 Setting up your computing environment

Computational exercises you will be performing in this course require operating systems (OS) with Unix or Linux-like commands. This means you need to set up your environment to make sure those commands are available in a commandline environment, for example a terminal window. It is **crucial** that you set this up correctly/properly as the course heavily relies on tools that need to be run in command line environment.

The following sets of instructions will help you to set up your environment correctly to make sure that these sets of commands are available. If you are already using Linux or Unix operating system (OS), then you are all set. If not, you will want to follow the instructions to configure your environment.

1.1 Windows OS

Your Windows OS should already come installed with Windows Powershell (if the OS version is fairly recent), which you will need for the following instructions.

You will need to turn on Windows Subsystem for Linux (WSL) feature on in order to enable and run Unix/Linux commands natively on Windows operating system. You can follow instructions on this page on how to set up and configure your Windows OS to enable Unix commands:

<https://www.howtogeek.com/249966/how-to-install-and-use-the-linux-bash-shell-on-windows-10/>
and here:

<https://docs.microsoft.com/en-us/windows/wsl/install-win10>

Please follow the instructions shown in both pages I have referred to above. WSL would allow you to use **bash** and Unix commands natively in Windows OS. You will also have an opportunity to install a Linux distribution as shown in both pages. I would highly recommend that you choose Ubuntu as the distribution as it is more widely used and has more user support than other distributions.

Briefly, the steps described in those two websites involve the following:

- enabling WSL on Windows Features
- installing a Linux distribution system (Ubuntu, for example) on Windows

Some instructions require that you run the Windows Powershell in administrator mode. To do that, you need to right click on the Windows Powershell icon to display other options and select the option that says “run as Admin” or something similar.

Setting up the command line environment for Windows can be a bit challenging compared to Mac OS but it is not impossible.

1.2 Mac OS

Setting up the command line environment in Mac is a lot easier as Mac OS evolved from BSD, which is an offshoot of earlier Unix operating systems. Consequently, there are more Unix-like tools already built into the operating system. Nevertheless, it is not quite the usual GNU Unix/Linux type of commands you might want to use. In order to set up the environment correctly, follow the instructions below.

This website has pretty good instructions (although a bit old) on how to set up the command line environment without having to install the whole Xcode, which is usually a precursor needed to install command line tools and requires a large space.

<https://mac-how-to.gadgethacks.com/how-to/install-command-line-developer-tools-without-xcode-0168115/>

Basically, you would first:

1. Open your terminal
2. Then type this: `xcode-select --install`
3. Then follow the instructions from the pop-up windows and it should install the necessary tools for the command line environment.

A few additional websites on how to set up the command line environment on Mac OS are shown below:

<https://osxdaily.com/2014/02/12/install-command-line-tools-mac-os-x/>

<https://www.maketecheasier.com/install-command-line-tools-without-xcode/>

Once the steps shown in these pages are followed, your command line environment should be ready for this course.

1.2.1 Install Homebrew

Next, you want to install Homebrew, which allows you to use the `brew` command on your Mac OS terminal. It is a package managing system that allows you to install basic utilities and even some bioinformatics tools on your Mac without needing admin privileges.

- First, go here and follow the instructions: (<https://docs.brew.sh/Installation>)
- Open your terminal
- Then type:

```
cd
```

```
mkdir homebrew
```

```
curl -L https://github.com/Homebrew/brew/tarball/master | tar xz --strip 1 -C homebrew
```

```
brew install coreutils findutils gnu-tar gnu-sed gawk gnutls gnu-indent gnu-getopt grep
```

The first `cd` commands takes you back to your home directory where you want to be installing Homebrew. The rest of the commands shown will install GNU Unix commands, which are different from original Unix-like commands that come installed with Mac OS.

Once Homebrew is installed, it will be very easy to install other tools such as `git`, which you will need to check out the course Github code repository and see the instructions. Remember, `brew` command is your best friend on any Mac OS to install and configure tools you would need for pretty much anything.