Math 4610 Fundamentals of Computational Mathematics - Topic 4.

This topic covers information on how to work with terminals and using a Linux or Unix environment. In most real computational settings, it is important to be able to work using a command line to create files, modify these files, compile code, and a number of other tasks. Most computers have at least one terminal emulating application right out of the box. If you are using an Apple computer, the operating system that underlies the desktop interface is Linux. You can invoke a terminal and work on the homework and projects assigned in the class. If you are using a PC with Linux installed, you probably already know how to bring up a terminal. If you work on a Windows machine, you can use PowerShell or install a Linux/Unix emulator like Cygwin or MinGW to work your way through the course.

Looking ahead to computational resources that you will use both in the class and beyond most High Performance Computing (HPC) environments require a lot of familiarity with terminals. This topic will cover the various ways to get a terminal up and running. Once we have a few choices, the notes will focus on terminals generated by Cygwin and Xtra-PC. Cygwin is freely available for use on PCs. The instructions on the installation of Cygwin can be found at

    https://cygwin.org

There are simple instructions to get the installation underway. Note that the installation of Cygwin on your computer is both a time intensive job and also requires a bit of disk space. The Xtra-PC platform is stored on a thumb drive that can be plugged into your PC. It costs a little, but works very well.

The benefits of a Cygwin installation are (1) you get a bunch of compilers (gcc, gfortran, and others) for coding up algorithms we will cover in the course, (2) LaTeX can be installed and used directly. There are a ton of packages available for you to explore and use. We will also use git which is installed as a part of Cygwin. With the Xtra-PC version you will likely need to install a few things. This is also very easy to do.

So, let's start into the various ways to start up a terminal and a couple of basic commands to execute in the terminal.

A Brief Introduction to Linux/Unix Emulation with Cygwin:

If you are interested in using Cygwin to pull up a terminal the first thing to do is to install the software locally. Note that the Engineering Computer Lab already has a version of Cygwin installed for student use. In a browser, the first thing you will see is the following.
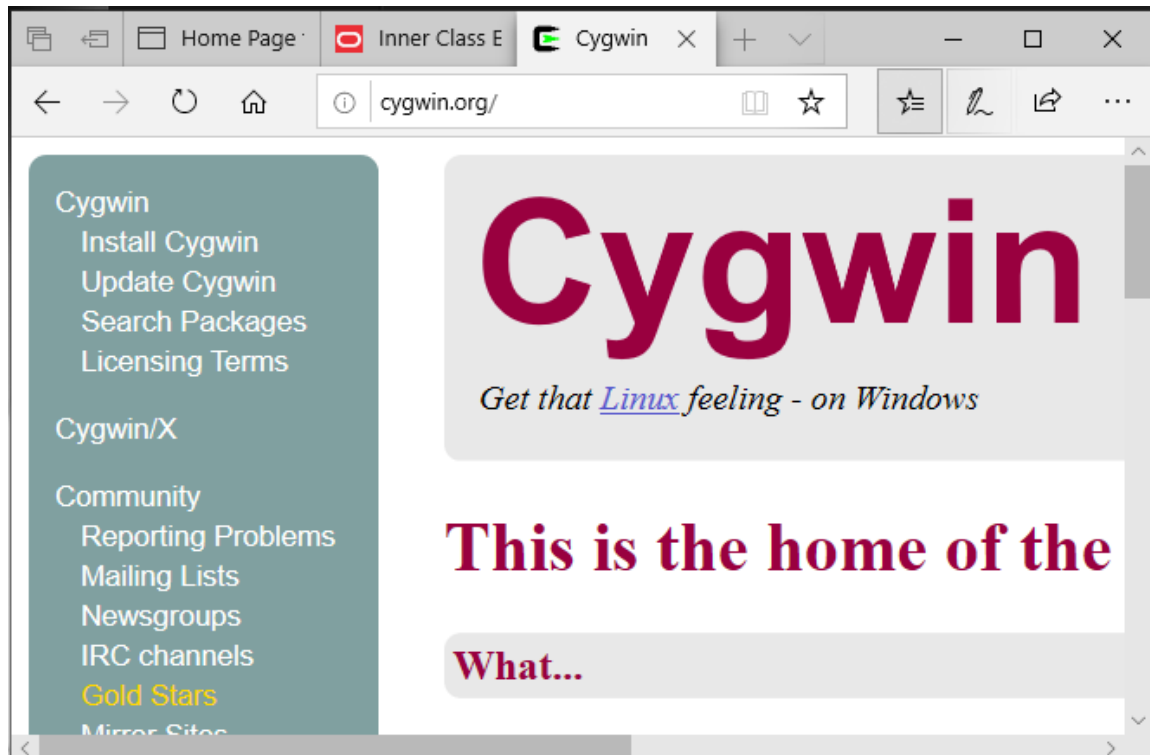
Figure 1: Website for installation of Cygwin. Screenshot taken using **Snip & Sketch**. This is an app on my Windows 10 box

---

You should start with the base installation if you are not used to installing large software packages on your computer. After the installation, there should be an icon on your desktop that can be used to bring up a terminal emulator.
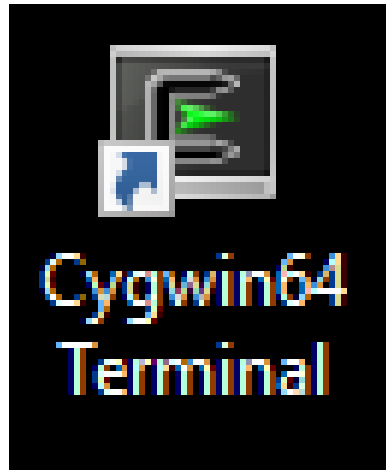

Figure 2: Cygwin icon on your desktop. Screenshot taken using **Snip & Sketch**. This is an app on my Windows 10 box

Double click on the icon and a terminal that looks like the following will appear on your screen. The terminal comes up with a prompt that is based on the computer name. A basic terminal will start up using the bash command terminal.

Figure 3: Example of a terminal that enulates a Linux operating system and command line. Screenshot taken using **Snip & Sketch**. This is an app on my Windows 10 box

The text that automatically appears in the terminal is called a prompt. That is,

```
koebbe%
```

is the prompt and our job will be to enter commands to get our computer to create files, modify files, compile computer programs and the like. The next topic in our list will give you a few standard commands that can be used to do real work for the course.