CISC 3595 - Operating Systems Programming Assignment Setup

Introduction

Programming assignments for this course can be completed in C++ with a Linux execution environment. They will be assigned and submitted via Github Classroom.

This document discusses some options for how you can set up your development environment and provides instructions for interacting with Github Classroom.

Github

If you do not already have a Github account that is associated with your fordham.edu email address, sign up at http://github.com. You will need your github credentials for the git setup steps below.

Development Platform

You may do your programming assignments in any environment with which you are comfortable. The target environment on Github Classroom is Redhat Linux, but the programs written for this class will be sufficiently generic that they should run on any Linux variant.

The following are some options that you might consider for your development environment:

Fordham Linux Shell - erdos

You may wish to do your programming on erdos (the shared Linux system available to Fordham students and faculty). If you do not already have an account on erdos, please let me know and I will create one for you.

The CIS Systems page on the Fordham website provides useful assistance for logging in to your erdos account from the campus computer laboratories, via a remote desktop connection or via secure shell (SSH) from another machine on the Internet.

If you will be using the shell mode on erdos, and you have not used git on this system before, enter the following commands from the terminal to properly initialize your git configuration:

```
git config --global user.name "Your Name"
git config --global user.email "your_email@fordham.edu"
git config --global core.editor vim # Or your preference
git config --global core.askpass ""
git config --global credential.helper cache
```

These commands do the following:

- Configure your name and email to identify you when you commit code to a git repository.
- Sets the editor that you would like to use when editing commit messages through git
- Ensure that when git prompts you for a password it uses a simple command prompt
- Caches your github username and password so you don't have to re-enter them multiple times in a session.

For additional information on Git, see the Git Book online.

Windows Subsystem for Linux

If you have a computer running Windows 10, you can program locally using Windows Subystem for Linux (WSL).

To set up your system to program in WSL, follow these steps:

Install WSL

Install WSL by following Microsoft's Quick Start instructions. These instructions refer to two different WSL versions (WSL 1 and WSL 2). Either will be fine for use with our class. These instructions will instruct you to obtain a Linux distribution; I recommend using Ubuntu 20.04 (which is the one I'm using on my own system).

Install git (on both Windows and Linux)

We will be using git for version control as part of our coursework.

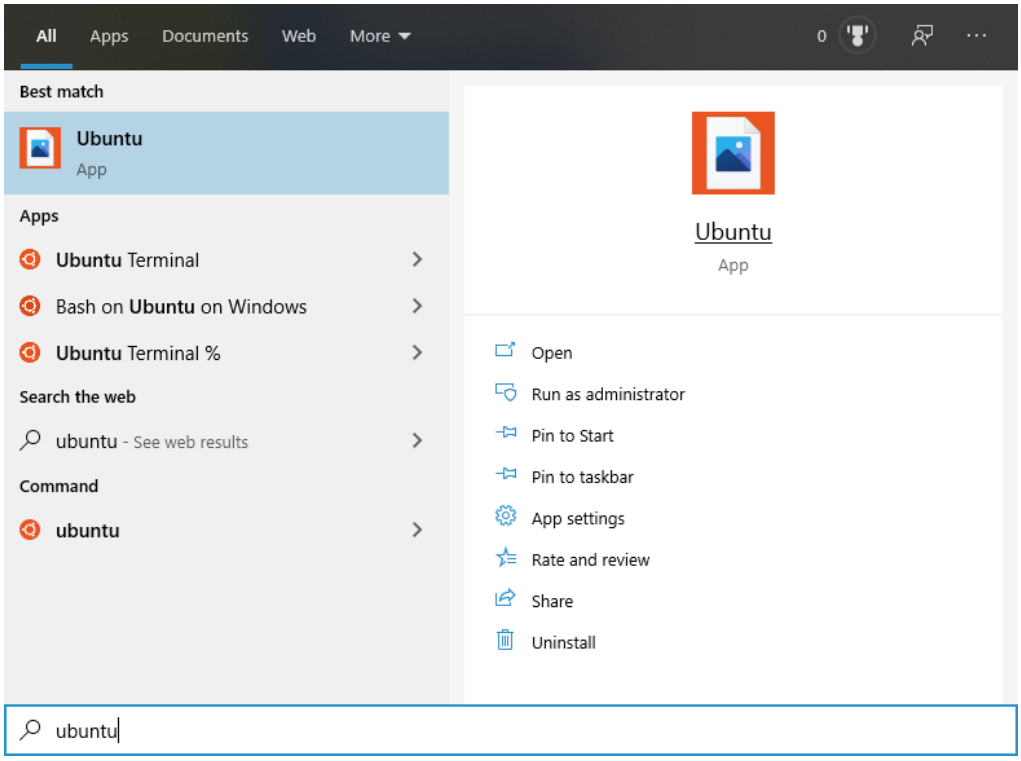
Microsoft's WSL Documentation has a good page on installing git for both Windows and Linux, and for dealing with the interactions between them. Follow the instructions here to set this up.

Install Your Development Language

You will need to install the tools needed to develop in either C++, as follows.

First, open a shell window on your system:

Click the Windows Button (“Start”) and type “Ubuntu” - then click on the Ubuntu App:



This will open a shell window in your Ubuntu subsystem.

Then install one or both of the languages as follows:

Gnu C++ Compiler (g++)

Install the Ubuntu build-essential package, as follows:

```
skamens@Sam:~$ sudo apt install build-essential
```

You will be prompted for the password you created when you set up WSL the first time you use the “sudo” command in your session.

This will install a number of tools, including the g++ compiler and the “make” command, both of which will be needed for our class.

Development Tools/IDE

As noted above, you can use any tools you are comfortable with to do your programming assignments. Some possibilities:

Text Editor and Command Line Tools

If you are comfortable with a Linux text editor such as `emacs` or `vi`, you can use those tools on either erdos or in a WSL environment. If you want to use emacs on Ubuntu, you will need to install the package using `sudo apt-get install emacs`.

Besides the text editor, you will need to use the “make” and/or “g++” commands to build your programs, and possibly “gdb” for debugging.

Visual Studio Code (VSCode)

Visual Studio Code is an integrated development environment (IDE) that runs on multiple platforms. It is available on erdos (if you run in remote desktop mode), but it works particularly well on Windows in conjunction with WSL. To install and configure VSCode for this purpose, follow these instructions for C++ and these instructions for Python.

If you are programming in C++, the assignments can be built using “make”. Makefiles are provided with each assignment template.