Lab 2: Introduction to Github

UWYO COSC 2030

Introduction:

Welcome back! Github can be a little confusing at first, but if it makes it easier you can just think of it as Dropbox or Google Drive but for code. You can access things from different devices, and most importantly when you leave off on one device, you can pick up exactly where you left off on another device.

Some useful commands:

Of course, as this is a Linux tool, accessing the files isn't quite as simple as double clicking an item in your google drive. But once you get the hang of it, it's honestly pretty similar. Practice makes perfect, right? Here are the commands that you're going to need to interact with Github:

(note: anywhere I leave something like [this], that's meant to be a placeholder. Don't include the [].)

git clone [url] – how you initialize the Github repository on your machine. Only do this once.

git pull – checks the Github repository for changes and updates your local files to reflect them.

git add [file] – adds a specific file to a "commit". We'll get to that in a minute.

git add . – adds all files to the commit.

git commit -m "example" – Creates a commit. Think of this like a more sophisticated drag-and-drop.

git push – pushes the commit to the Github repository. If it goes well, you will see the changes online.

Brief note on authentication tokens:

Github was designed by a bunch of nerds. As such, they have a bunch of nerd security rules. One of these that you're likely to encounter is that you will be unable to access your Github account with your password from the command line. Why? Who knows! Instead, you're going to need to generate a passkey for your account and store it somewhere secure. I will help you through this in lab, but if you can't make it or something goes wrong and you need to make a new one, reach out to me by Discord or email and we will get it taken care of.

Reminder about the README.md

By the end of the lab, you're going to know what a README.md is. If you don't format it correctly, I have to deduct points. What I ask is that for this, as well as all the upcoming labs, you edit the README.md document to show:

> Your name
>
> Help given/received

So for me, this would look like:

> Michael Stoll
>
> Help given/received: I had help from Professor Ward

This might seem obnoxious, but for the sake of avoiding plagiarism allegations, please humor me.

Edited by Michael Stoll for UWYO COSC 2030

Lab 2:

In the same way that the last one was just an introduction to Linux, this lab is really just going to be an intro to Github, though it's going to look more dense because there are more things you need to know here.

First, we'll look at the easiest one, the Github GUI:

Step 1: Click this link to begin the assignment on Github: https://classroom.github.com/a/rSenyZ00

Step 2: Navigate to the homepage, and click through "Add File" -> "Upload Existing File".

Step 3a: Upload anything you would like. A .cpp file, an image, the .pdf of this lab, etc.

Step 3b: Remember to fill out a commit message. I don't know why, but Github really cares.

Step 4: Click "Commit Changes".

And your file(s) should be uploaded!

Now for the command line, where you'll be doing most of your work this semester:

Step 1: SSH into the Hive (look at the cheat sheet or lab 1 if you forgot how).

Step 2: Copy the link from your Github repository.

Step 3: Clone into the repository using your username and passkey (or just passkey).

Step 4: Use "ls" to verify that you successfully cloned into the repository. Then, "cd" into the folder.

Step 5: Make a basic .cpp program, perhaps "Hello World!", in this folder.

Step 6: Using the commands on the last page, add, commit, and push the file back to Github.

Step 7: Check on Github to make sure your file made it.

As before, you should see your file at the top!

Now for one final step, which will be very helpful for you for program 1 testing:

Step 1: Go to the course webpage, click on the "Pi's" link, and choose a Pi. Click the "Claim" button.

Step 2: SSH into your selected Pi by using the listed name as the host name and "padawan" as the username (i.e. "ssh padawan@archer.cs.uwyo.edu" if you use Archer, the password is "wyoming") or by using PuTTy.

Step 3: Clone into your Github repo here, then navigate into the folder.

Step 4: Run your program from here. Notice that it runs exactly the same as it did in the Hive.

And you're all done!

Submission:

There is nothing to submit this week.

Edited by Michael Stoll for UWYO COSC 2030