Math 4610 Fundamentals of Computational Mathematics - Topic 8.

Using git to Work Locally on Your Computer

You can chose to work on projects on Github by logging onto the Github web site. However, if your internet connection is not as good as you might like, you can use "git" to synchronize the work on your project. The git environment is a Version Control System (VCS) which allows different groups to work on the same project without stepping on each other's toes. In this topic we will learn a bit about git and start using git to synchronize your work on Github. Your work will be downloaded from Github for grading.

Using git to Work Locally on Your Computer

The first thing to do is to make sure that git is available within your terminal or terminal emulator. To do this use the following command:

```
koebbe% which git
```

If git is available, you will see output like that in the following figure. In this case, the executable for git is in the folder /usr/bin. If git is not available, you can install the software on most computers. If you need some help getting the sofware installed, talk to your instructor.

Figure 1: Checking for git using the which command.

Command in the git VCS

This section of notes will not cover all possible ways to use and modify git. Instead, we will look at some basic commands that git uses to share the data in a repository. The general form for a command using git is the following:

```
% git command [options]
```

To start, we can type in the command

```
koebbe% git --help
```

This produces a couple of screens of output. Another way to display the same output, one screen at a time, is to pipe the output from the command above into another Unix command. The result is

```
koebbe% git --help | more
```

The concept of a pipe in Unix is to take the output from one command and use this as input to another command. So, the output from

```
koebbe% git --help
```

is piped into the more command. The more command will display output one screen at a time. You should try this some time.
What you should notice is all of the options for running commands within git. The end result is that you can read the first screen and then hit a space bar to read the next screen. The rest of this section of notes will go over a bare minimum of git so that you can work on your own computer.

Figure 2: The first part of the output gives about half of the possible options we can use with git.

Figure 3: Checking for git using the which command.

Initializing a Folder Using the git init

To start, there are two ways to initialize a git folder. The first is to take a new folder and initialize the folder using

```
koebbe% mkdir tempdir
koebbe% cd tempdir
koebbe% git init
```

As the output states, the result is an empty Git repository. The other method will work well for making a copy of the Github repository named math4610. You can clone a repository from another site. To do this, change into the folder where the repository folder will end up.

Figure 4: Creation of a local repository using the git init command.

Initializing a Folder Using the git clone

We can use the clone command in git.
To do this, type

```
koebbe% cd foldername
koebbe% git clone https://www.github.com/username/repositoryname
```

So, if Fred has chosen fred as a username on Github, the pair of commands would be

```
koebbe% cd repository_location
koebbe% git clone https://www.github.com/fred/math4610
```

This will put a copy of everything in math4610 in a directory on your computer. Note that in the figure below a different repository was used. The math4610 repository is too large and takes a bit of time to clone.

Figure 5: Creation of a local repository using the git clone command.

Using the git status Command

One of the most used commands in git is the status command. This will tell you about any changes that have been made. For example,

```
koebbe% git status
```

You will use this command over and over if you are being efficient. An example of the output is shown in the next figure when working in repository.

Figure 6: Using the git status command to List Changes.

The git commit Command

The git push Command

The git pull Command

Figure 7: Screenshot taken using **Snip & Sketch**. This is an app on my Windows 10 box

Figure 8: Screenshot taken using **Snip & Sketch**. This is an app on my Windows 10 box