# Lab - Terminal & Git

We will use lab time today to work on terminal and git.

# Let’s make sure your forked repository is connected with our repository so you can get the latest homework/lab files. In the git world, we call this “configuring a remote for a fork”.

1. Open terminal
2. You can see the current remote repository by typing the command “**git remote -v**”. You should see your fork repo as origins.
3. Then specify a new remote upstream repository by typing “**git remote add upstream https://github.com/dandanbang/IOLab.git”**
4. Verify the upstream repository by typing “**git remote -v**”. You should see two additional remotes now.
5. Now, let’s try to sync the fork, you should do this before you start every homework.
6. Fetch all the branches of that remote into remote-tracking branches, such as upstream/master: Do it by typing the command “**git fetch upstream’**
7. Make sure that you're on your master branch by typing “**git checkout master”**
8. Merge the changes from upstream/master into your local master branch. This brings your fork's master branch into sync with the upstream repository, without losing your local changes. Do this by typing “**git rebase upstream/master**
9. Now, let’s do “**git push origin master**” to make sure your origin master has is up to date as well. (so far only your local/master is up to date)

# Now you’re ready to start the lab.

# Part 1 - Find out the Commands

We use the shell to interface with our computer through the operating system and tell it to do something. For this class, we are using the Bash shell, there are other types of shells available such as, ZSH or Fish, however, for what we are going to be using in this class, Bash will work for us nicely.

Using your preferred text editor, create a lab.txt file in your **IOLab code repository**, under the lab1 directory. In your lab1.txt file, type which bash command you would use to perform the following tasks:

Example: Change file permissions: chmod 777 filename.txt

1.     Change Directory

2.     Create a Directory

3.     Rename a Directory

4.     List the contents of a directory

5.     Move a file to a new location

6.     Copy a file to a new location

7.     Search for a string in a file

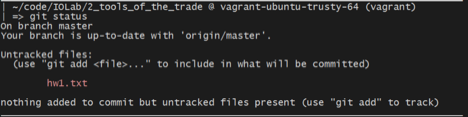
8.     Access the shell of a remote machine

9.     Remove a directory folder

You should be able to find the answers relatively easily by searching for them on Google. You should get comfortable searching Google and Stack Overflow or the relevant documentation to find answers to questions like above.

# Submitting Your Homework

You are going to submit this lab via Git. Using the shell, navigate to your IOLab directory. If you are in the correct directory, you should be able to run “git status” and see something like below (the red will be lab1.txt instead of hw1.txt):



Before you add the homework and submit it, first create a new branch by running:

**git checkout –b lab-1-branch**

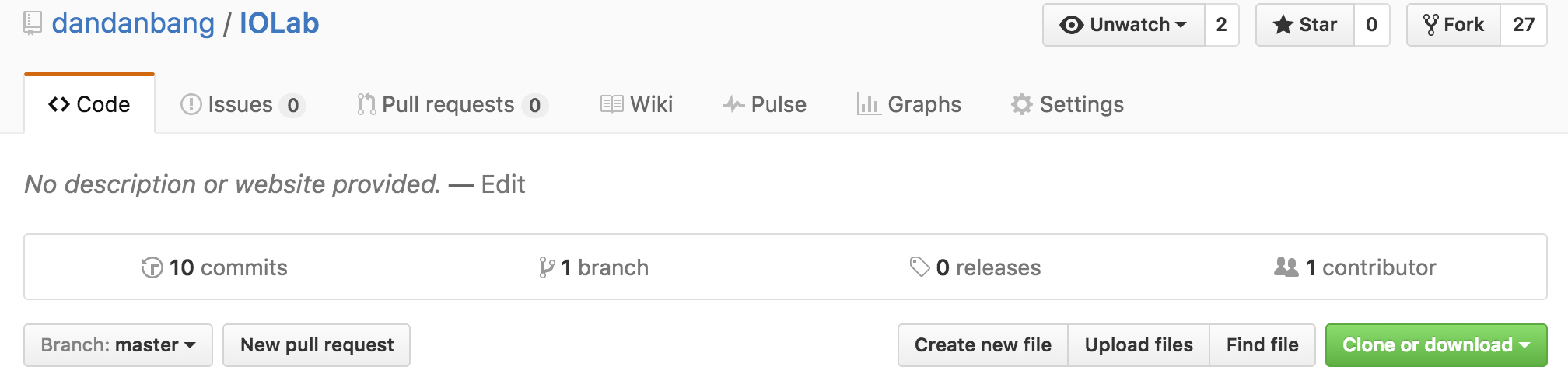
After you create a branch, stage the lab1.txt file, commit, and push the changes to your repository. You can do that by running the following commands:

**git add .**

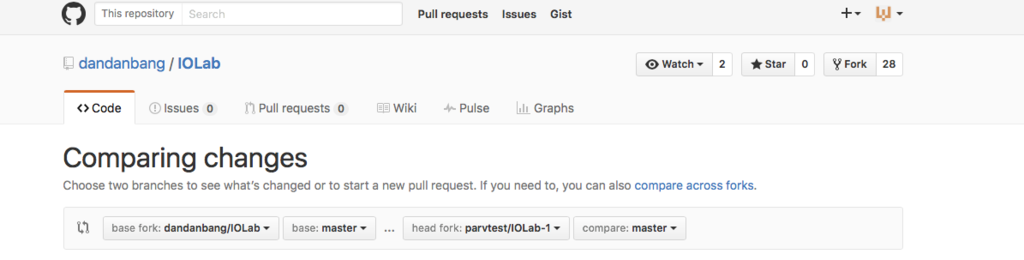
**git commit –m “submitting lab1”**

**git push origin lab1-branch**

As a final step, navigate to your repository on Github’s website. You should see something similar to this page, click on “new pull request”.



In the following page, you will see something similar to below:



Ensure that you select the base fork as “dandanbang/IOLab” and select the head fork as your own instead of parvsondhi/IOLab, and the branch to branch-1-branch. Finally, you can click **“Create Pull Request”** to submit your homework.

\*\* The reason for turning in your code in this manner is because it allows to give you pointed feedback on your code inline.  This is a common practice in the industry and is formalized through a process called a code review. If you are interested, you can read more about it here: [http://blog.codeship.com/github-code-review/.](http://blog.codeship.com/github-code-review/)