CPSC 256 - Lecture 2

The goals of this lecture are to:

1. Fork my NetBeansProjects repository (this contains all of my code for this class)
2. Clone your new repository on the Ubuntu laptop in front of you
3. Edit the main.c program of the HelloWorld Project
4. Compile and execute this edited file from the command line
5. Push these changes back up to your newly forked repo

Step 1: Cloning my NetBeansProjects Repo

a) In Firefox, go to https://github.com/ and log into your account
b) In the search bar at the top left of the page, type in "user:brash99"
c) Click on the brash99/NetBeansProjects link
d) Click on "Fork" in the upper right corner

That's it! You now have your OWN COPY of all of my code.

Step 2: Clone your new repository on the Ubuntu laptop in front of you

a) Log into the machine ... username is XXXXXXXX@cnuadmin.cnu.edu ... password is your usual CNU password

b) Open a terminal window (click on the "9 dots" icon in the bottom left, and find the terminal app)

Aside: Some basic Unix commands:

```
whoami – gives your username
ls     – lists all of the files in the
            current directory
ls –altr   – lists all of the files, in long
            format, in reverse time order
pwd    – shows the present working directory
cd     – returns to your 'home' directory

cd ~/Documents  – takes you to the Documents directory
                    ("~" is a shorthand for your
```

```
                                home directory)
```

c) From your home directory, clone your GitHub code repo

```
cd
git clone git@github.com/<githubusername>/
NetBeansProjects.git
```

where <githubusername> is your GitHub username (not necessarily the same as your local username!!!!)

If this step is successful, you should see that there is a new directory, under your home directory, called NetBeansProjects (check by doing 'ls -altr')

Step 3: Edit the main.c file of the HelloWorld project

a) Get to the correct directory

```
cd ~/NetBeansProjects/HelloWorld
ls –altr
```

b) Edit the main.c file

```
nano main.c
```

(We will discuss the changes to be made in class … Save the changes with CTRL-X)

Step 4: Compile and execute the program

```
gcc –c main.c (What did this do?  Check with ls –altr)
gcc –o myFirstProgram main.o (What did this do?
                                Check with ls –altr)
./myFirstProgram (Did it work????)
```

Step 5: Push these changes back up to your newly forked repo

a) Check which files have changed/been added with:

```
git status
```

b) Remove the object file and executable, for cleanliness

```
rm main.o
rm myFirstProgram
git status
```

c) Commit the changes

```
git commit –am “Edits to HelloWorld/main.c”
git status
```

```
You should see that your local repository is now one
commit ahead of your GitHub repo.  We want to now
push this commit up to the GitHub.com servers.  This is,
admittedly, a bit cumbersome, because of
cybersecurity issues.
```

Step 6: Configuring SSH keys to allow passwordless remote pushes in Git

On the local machine:

a) Execute the following commands from a terminal window:

```
cd
ssh–keygen –t rsa
```

(Just hit return a bunch of times until you get the terminal prompt again, i.e. just respond with blank answers to all questions)

```
cd
ssh–keygen –t dsa
```

(Again, just hit return a bunch of times)

```
cd .ssh
cat id_rsa.pub id_dsa.pub > authorized_keys
chmod 600 authorized keys
cd
```

b) Create a config file for SSH GitHub authentication, as follows:

```
cd
```

```
cd .ssh
nano config
```

(In the editor, make this file look like the following:)

```
Host github.com
  Hostname github.com
  User edwardbrash
  IdentityFile /home/localbrash/.ssh/id_rsa.pub
```

where you need to replace ‘edwardbrash’ with your GITHUB username, and ‘local brash’ with your LOCAL username! Save this file and exit.

Now, set up the correct permissions on this file by doing:

```
chmod 600 config
cd
cat .ssh/id_rsa.pub
```

In the Firefox browser, in the tab with your GitHub account open:

i) Click on Settings, in the pulldown menu under your account avatar in the upper right corner

ii) On the left side menu, click on ‘SSH and GPG Keys’

(iii) Click on the green ‘New SSH key’ button

(iv) Choose the Title to be something useful, perhaps CPSC256, and then CUT AND PASTE THE CONTENTS OF id_rsa.pub, PRINTED ABOVE, INTO THE TEXT BOX!!!!!!!!

(v) Click on ‘Add SSH key’

You should now be set up to do password-less authentication to this GitHub account! And, this should work for all repositories, not just this one. :)

Step 7: Pushing Changes

(1) Push the changes

```
cd
```

```
cd NetBeansProjects
git push origin main
```

(2) Go back to your GitHub account in Firefox, and check to see that the newly edited main.c actually got pushed!