CMPSC 102
Discrete Structures
Fall 2018

Practical 1: Setting Up GitHib Classroom
31 August 2018

Summary

To learn how to navigate the directories within Ubuntu operating system using command line interface. To set up Github for use in the course.

Using Your Computer Science Account

In advance of today's lab you have already received the details about your Alden Hall computer account and learned how to log on. You may use this account on any computer in Alden labs in rooms; 101, 103, or 109. Your files are stored on a central server; and may be reached from any lab machine after logging in to a lab machine.

Hours of lab availability are posted on the bulletin board in each lab and on the following Web site: `http://www.cs.allegheny.edu/`; the on-duty lab monitor is always available in Alden 101.

Navigating using the Command Line Interface

A command-line interface allows the user to interact with the computer by typing in commands. Computing professionals prefer to use the command line interface, called the "Terminal", built into operating systems like Linux, instead of using the graphical user interface for launching programs and etc. In many situations command line interface tends to be very efficient and effective, for example, it allows you to complete some tasks with a simple one line command instead of using having to click on desktop items using the mouse!

1. Read through the supplemental handout on "Tips on Using Linux and the Command Line Interface'.' Locate the terminal window and open it as explained in the reading handout.

2. Now you will practice using the commands discussed in the handout. Using the terminal window type each of the commands found in Table 1 of the supplemental handout. Make sure you understand what each command does. You will have to create new files in order to run some commands such as `cp`, `mv`, etc. The most basic method of creating an empty file is with the `touch` command. This will create an text file using the name specified: `touch file1` or multiple files as: `touch file1 file2`. Remember to execute a command, you should press the "Enter" key after typing a command. Check with your neighbors to see if they are able to open the terminal window, and use commands such as `cd, cp, pwd, ... , ls`, etc. If you can help them, please feel free to do so!

3. Take a screen shot (using "PrtScn" button located at the top right part of the keyboard) of your terminal window, showing the commands you have typed. Verify that your file was saved in the "Pictures" directory of your home directory. Note: when you push this button, your desktop will flash to signify that a screenshot has been made.

4. To avoid confusion and clutter in the future, delete all of the newly created files and directories from the previous practice step.

5. Now create a directory called `cs103F2018` in your home directory, by typing `mkdir cs103F2018` command in your terminal. This is where all of the work you do in this class should reside.

6. From the home directory type the `pwd` command in the terminal.

7. You can now close the terminal window by typing the `exit` command. You can reopen another terminal window and navigate to the the this directory by using the `cd` (change directory) command, followed by the new location in your current directory. Remember that `cd ..` will move back one directory, while `cd <directoryName>` will move you up to the directory called `<directoryName>`.

Configuring Git and GitHub

During the subsequent practical and laboratory assignments, we will securely communicate with the GitHub servers that will host all of the project templates and your submitted deliverables. In this assignment, you will perform all of the steps to configure your account on GitHub, so that you are ready to start your first lab assignment using GitHub Classroom next week. You can also learn more about GitHub Classroom by visiting `https://classroom.github.com/`. As you will be required to use Git, an industry standard tool, in all of the laboratory and remaining practical assignments and during the class sessions, you should keep a record of all of the steps that you complete and the challenges that you face to be used in your code documentation. Please ask for help from the course instructor or a teaching assistant if you have trouble completing certain steps.

Figure 1: GitHub allows students to package and ship their own software to the community.

1. If you do not already have a GitHub account, then please go to the GitHub web site (at `https://github.com/`) and create one, making sure that you use your `allegheny.edu` email address so that you can join GitHub as a student from Allegheny College which has permission to use various parts of the GitHub service. You are also encouraged to sign up for GitHub's "Student Developer Pack" at `https://education.github.com/pack`, qualifying you to receive free software development tools. Complete the form and once you gain access to the GitHub site, you should be able to see a screen where the icon of your account is shown. On my page, the upper right side is shown in Figure 1 on the GitHub page.

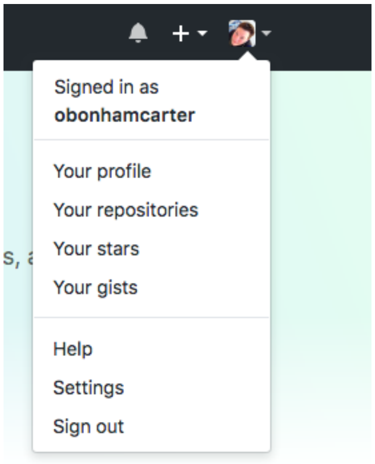

Figure 2: GitHub settings menu.

2. Additionally, please add a description of yourself and an appropriate professional photograph to your GitHub profile. Unless your username is taken, you should also pick your GitHub username to be the same as Allegheny's Google-based email account.

3. If you have never done so before, you must use the `ssh-keygen` program to create secure-shell keys that you can use to support your communication with GitHub. These keys enable you to send your files to GitHub without having to type in a password each time. Note: your ssh keys serve as the password and so you have to be using the machine of these keys.

   Open the terminal as you have done in the previous step. Alternatively, you can search for it by starting to type the word "terminal", and then select that program. Another way to open a terminal window involves typing the key combination `<Ctrl> +<Alt> + t`.

4. Now that you have started the terminal, you will now need to type the `ssh-keygen` command in it. Follow the prompts to create your keys and allow the generator program to save them in the default directory (`.ssh` of your root directory). Press "Enter" when you are prompted to `Enter file in which to save the key ...` and then type your selected passphrase whenever you are prompted to do so. Please note that a "passphrase" is like a password that you will type when you need to prove your identify to GitHub. Note: passphrases generally involve more keys than passwords and so are thought to be more secure. Verify that the generator has stored its key files in the `ssh-keygen` root directory (`.ssh`).

5. Once you have created your ssh keys, log (again) into GitHub and look in the right corner for your avatar information. Click on this link (similar to that featured in Figure 2 and then select the "Settings" option. Now, scroll down until you find the "SSH and GPG keys" label on the left, click to create a "New SSH key", and then upload your ssh key to GitHub.

   You can copy your (PUBLIC!) SSH key to the clipboard by going to the terminal and typing "`cat ~/.ssh/id_rsa.pub`" command and then highlighting this output. When you are completing this step in your terminal window, please make sure that you only highlight the letters and numbers in your key—if you highlight any extra symbols or spaces then this step may not work correctly. Then, paste this into the GitHub text field in your web browser.

6. Again, when you are completing these steps, please make sure that you take careful notes about the inputs, outputs, and behavior of each command. If you have trouble, then please ask the course instructor or the teaching assistant.

7. Since this is your first assignment and you are still learning how to use the appropriate software, don't become frustrated if you make a mistake. Instead, use your mistakes as an opportunity for learning both about the necessary technology and the background and expertise of the other students in the class, the teaching assistants, and the course instructor.

8. In the class repository that you created earlier, clone the *classDocs* repository which is hosted the GitHub website at. The website for the repository is the following:
`https://github.com/Allegheny-Computer-Science-102-F2018/classDocs`.

   Before you can make the clone, you will have to locate the SSH clone address from the green button (labelled, *Clone or download*) at the website. After clicking on the button, you will use the displayed SSH address in the following command to copy the *classDocs* files to your local machine:
   `git clone <ssh address>`.

   This operation will only be used once. To collect all updated documents from here on, you will type the command, `git pull` within the *classDocs* directory on your machine. Please ask questions if you have trouble.

General Guidelines for Practical Sessions

- **Experiment!** Practical sessions are for learning by doing without the pressure of grades or "right/wrong" answers.

- **Submit *Something*.** Your grade for this assignment is a "checkmark" indicating whether you did or did not complete the work.

- **Try to Finish During the Class Session.**

- **Help One Another!** If your neighbor is struggling and you know what to do, please offer your help. Do not feel that you are to "do the work" for them, but advise them on what to type or how to handle things. If you are stuck on a part of this practical session, ask your neighbor, a teaching assistant, or the course instructor.