CMPSC 111
Introduction to Computer Science I
Spring 2018
Janyl Jumadinova

Practical 1
19 January 2018
Due by midnight of the day of your practical
“Checkmark” grade

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. You will also continue to practice using Slack to support communication with the teaching assistants and the course instructor.

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 101, 103, or 109. Your files are stored on a central server; you don’t have to use the same machine every time you log on in a laboratory.

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, built into operating systems like Linux, instead of using the graphical user interface. 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 the “pumping” motion of 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 a file is with the `touch` command. This will create an empty file using the name specified: `touch file1` or multiple files as: `touch file1 file2`. Remember to execute a command, you should press “Enter” 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.

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. Send your screenshot as a direct message to me via Slack.

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

5. Now create a directory called `cs111s2018` in your home directory, by typing `mkdir cs111s2018` command in your terminal. This is where all of the work you do in this class should reside.

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

7. You can now close the terminal window by typing the `exit` command.

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. You may see the course instructor or one of the teaching assistants if you are not able to complete a certain step or if you are not sure how to proceed.

1. If you do not already have a GitHub account, then please go to the GitHub web site and create one, making sure that you use your "allegheny.edu" email address so that you can join GitHub as a student at an accredited educational institution. 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. 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. Now, in the #practicals channel of our Slack team, please type on one line your full name, "allegheny.edu" email address, and your new GitHub username.

2. 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. But, to start, this task requires you to type commands in a terminal. 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 involves typing the key combination "<Ctrl>−<Alt>−t".

3. 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 save them in the default directory. That is, you should press "Enter" after 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. What files does "ssh−keygen" produce? Where does this program store these files by default? Do you have any questions about completing this step?

4. Once you have created your ssh keys, you should raise your hand to invite either a teaching assistant or the course instructor to help you with the next steps. First, you must log into GitHub and look in the right corner for an account avatar with a down arrow. Click on this link 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 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.

5. 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 there is something that you do not understand, then please ask the course instructor or the teaching assistant about it.

6. 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. Remember, you can use Slack to talk with the instructor by typing "@jjumadinova" in a channel.

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, offer your help. Don't "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 a course instructor.