

Lab 01 Specification – Version Control with Git and Bitbucket
Due (via Bitbucket and hard copy) Wednesday, 6 September 2017
50 points

Lab Goals

- Connect to Bitbucket
- Learn some Git commands
- Understand the behavior of some Java programs
- Write a short paper summarizing your work for this lab

Assignment Details

Practicing software developers normally use a version control system to manage most of the artifacts produced during the phases of the software development life cycle. In this course, we will always use the Git distributed version control system to manage the files associated with our laboratory assignments. In this laboratory assignment, you will learn how to use the Bitbucket service for managing Git repositories and the `git` command-line tool in the Ubuntu Linux operating system. After connecting to the course's Git repository and creating your own repository, you will compile and run two Java programs, write about their output, and commit your final report to a repository.

Linux Command Review

In CMPSC 111, you should have become familiar with a number of basic Linux commands for maintaining files and directories. Table 1 at the top of the next page lists a few of these.

Command	Meaning
<code>ls</code>	list files in current directory
<code>touch <i>filename</i></code>	create a new file " <i>filename</i> " in current directory
<code>mkdir <i>dirname</i></code>	make new directory " <i>dirname</i> "
<code>cd</code>	change directory to home directory
<code>cd <i>dirname</i></code>	change to directory " <i>dirname</i> "
<code>cd ..</code>	change to parent of current directory
<code>rm <i>filename</i></code>	remove " <i>filename</i> "
<code>rmdir <i>dirname</i></code>	remove (empty) directory " <i>dirname</i> "
<code>cp <i>name</i>₁ <i>name</i>₂</code>	copies " <i>name</i> ₁ " to " <i>name</i> ₂ "
<code>cp -r <i>name</i>₁ <i>name</i>₂</code>	copies all files and directories from " <i>name</i> ₁ " to " <i>name</i> ₂ "
<code>mv <i>name</i>₁ <i>name</i>₂</code>	move (rename) " <i>name</i> ₁ " to " <i>name</i> ₂ "
<code>mv <i>name</i> <i>dirname</i></code>	move file or directory " <i>name</i> " into directory " <i>dirname</i> "

Table 1: Some Linux Commands You May Already Know

The `man` command accesses the online “man,” or manual, pages. Thus, the command “`man ls`” brings up information about the `ls` command; “`man mkdir`” gives the manual page for the `mkdir` command. If you feel that you need to review any of these commands, please take some time now to look them up using `man`.

Configuring Git and Bitbucket

During this laboratory assignment and subsequent assignments, we will securely communicate with the `bitbucket.org` servers that will host all of our projects. In this laboratory assignment, we will perform all of the steps to configure the accounts on the departmental servers and the Bitbucket service. Through this assignment, you should refer to the following Web site for additional information:
confluence.atlassian.com/bitbucket/getting-started-with-bitbucket-675385635.html.

As you will be required to turn in a report describing each step that you finish in this assignment, please be sure to keep a record of all of the steps that you complete and the challenges that you face.

1. 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 the Bitbucket servers. Type `man ssh-keygen` to learn more about how to use this program. What files does `ssh-keygen` produce? Where does this program store those files?
2. If you do not already have a Bitbucket account, please go to the Bitbucket website and create one – make sure that you use your `allegheny.edu` email address so that you can create an unlimited number of free Bitbucket repositories. Then, upload your `ssh` key to Bitbucket.
3. Now, you need to test to see if you can authenticate with the Bitbucket servers. Open a terminal window on your workstation and change into the directory where you will store your files for this course. For instance, you might make a `cs112f2017/` directory that will contain the shared Git repository. Once you have done so, please type the following command: `git clone https://YOURUSERNAME@bitbucket.org/amohan_bb/cmpsc112f2017-share.git`. If everything worked correctly, you should be able to download all of the files that you will need to use for this laboratory assignment. Please resolve any problems that you encountered by first reviewing the Bitbucket documentation. If you are still not able to run the `git clone` command, then please see the TA.
4. Using your terminal window, you should browse the files that are in this Git repository. In particular, please look in the `labs/lab1/src/` directory and use Vim to study the two Java programs that you find. Remember, the `cd` command allows you to change into a directory.

At this point in the lab, you should have a directory structure that looks similar to this:

```
<home directory>
|---<maybe some other directories>
|---cs112f2017
|   |---cmpsc112f2017-share
|       |---lectures
|           |---<subfolders containing lecture audio/slides>
|           |
|           |---labs
|               |---lab1
|                   |---<files for lab1>
|                   |
|                   |---syllabus.pdf
|
|---<maybe some other directories and files>
```

Creating a New Repository

Now that you have learned how to clone an existing Git repository, you should make a new repository in the `cs112f2017/` directory that you previously created. First, make a new directory called `cs112f2017-<your user name>`. Then, change into this directory and type the command `git init .`. At this point, you should go into the `cmpsc112f2017-share` repository and use the `cp -r` command to copy the entire `labs/` directory from the `cmpsc112f2017-share` repository to `cs112f2017-<your user name>`. Once the files are in your own Git repository, please use the `git add` and `git commit` commands to add them correctly. If you do not know how to use the `git add` and `git commit` commands in the terminal window, please learn more about them by searching on the Internet and discussing them with the TA.

Next, you should use the Bitbucket website to create a repository that has the same name as the local directory and local repository. You must follow Bitbucket's instructions to push the code and tags in your local repository to the remote one. When you are finished with this step, you should see in your browser that the Bitbucket servers are storing the two Java programs. Once the Git repository contains the correct files, you should share your Bitbucket repository with the course instructor, whose Bitbucket username is `jwenskovitch`.

At this point in the lab, you should have a directory structure that looks similar to this:

```
<home directory>
|---<maybe some other directories>
|---cs112f2017
|   |---cmpsc112f2017-share
|   |   |---lectures
|   |   |   |---<subfolders containing lecture audio/slides>
|   |   |
|   |   |---labs
|   |   |   |---lab1
|   |   |       |---<files for lab1>
|   |   |
|   |   |---syllabus.pdf
|   |
|   |---cs112f2017-<your user name>
|       |---labs
|           |---lab1
|               |---<files for lab1>
|
|---<maybe some other directories and files>
```

From here, you can learn more about Git by consulting websites like <http://try.github.io/> and <http://gitimmersion.com/>. At minimum, you should ensure that you fully understand how to use the following Git commands in the terminal window:

- `git init`
- `git status`
- `git add`
- `git commit`
- `git push`
- `git pull`

Compiling, Running, and Understanding Java Programs

Once you have mastered the use of Git and version control, you should return to the `labs/lab1/src/` directory that contains two Java programs. Now, use the Java compiler to compile the `Hooray.java` program. That is, you should type `javac Hooray.java` in the terminal window. Next, you can run this program by typing `java Hooray` in the terminal window. What output does this program produce? Why does it create this output? How do you stop this program?

After compiling, using, and studying the `Hooray` program, you should complete the same steps for the `Weeee.java` program. Go ahead and compile and run this program. What output does it produce? Why does it create this output? How is the output similar to and different from that which was created by the `Hooray` program? Once you have finished studying and understanding these two programs, add comments to the code to explain what they do and how they work. Finally, please make sure that the commented version of each program is correctly committed to your Git version control repository hosted by Bitbucket.

Submission Details

For this assignment, please submit a written description of the procedures you have followed while completing this lab. Also, please upload this written document to your `cs112f2017-<your user name>` repository. Your submission should include the following:

1. A description of the steps that a user must take to configure Git and Bitbucket.
2. A description of the inputs, outputs, and behaviors of the six Git commands listed in the `Creating a New Repository` section.
3. A commented version of the `Hooray.java` and `Weeee.java` programs.
4. A report that clearly responds to the following four prompts:
 - (a) The steps that you took to compile and run both of these programs.
 - (b) The output that each of these programs produce.
 - (c) An explanation for why these programs create the output that they do.
 - (d) A discussion of the similarities and differences between these programs and their outputs.

Before you turn in this assignment, you also must ensure that the course instructor has read access to your BitBucket repository that is named according to the convention `cs112f2017-<your user name>`.