

# Lab 1

## Connectivity, vi/vim Editor, Unix Environment, & File Transfer

### CSCI 344 Shell Programming

Due Friday January 27, 11:59pm on Canvas  
30 points

## OBJECTIVES

1. Confirm access to a UNIX account on the instructor's Amazon Web Services (AWS) EC2 instance.
2. Students should be able to access their Unix account within the OCNL 244 lab room and remotely (i.e. from home, work, friends place, etc.). Our lab is often closed after-hours so it is important to have remote access.
3. Be familiar in using the vi/vim editor which is commonly used on the Unix environment to edit text data, modify scripts, etc.
4. Introductory to Bash: environment variables, scripts, etc.
5. Students should be able to transfer files between the AWS EC instance and his/her local computer.
6. Ensure the creation of a proper lab write-up format with screen shots and its submission via Blackboard.

### Step 1: Get your own student Unix account on the AWS EC2 instance

Contact instructor via email, during lab, office hours, etc. to get your *USERNAME* and *PASSWORD*

**All students must have access to the AWS EC2 instance.** It will be possible to perform most of the lab assignments from any UNIX environment that a student may already have access to. However, there will be situations where the lab assignment requires access to the AWS EC2 instance.

### Step 2: Log into your account

Confirm that you can access your UNIX account from our lab in OCNL 24 and remotely. There are several ways you can log in remotely.

If you have a Windows-based computer, download the PuTTY executable from [www.putty.org](http://www.putty.org). The PuTTY software is a very commonly used in academia and industry as a tool to create a secure shell (SSH) connection to a Unix system and open a Command Line Interface (CLI). Make sure to use port 22 when establishing a connection. You may want to try MobaXTerm.

On Apple iOS/Linux-based platforms, you should be able to open an ssh terminal session connection using the following command (the \$ character represents the Unix command prompt and xxx.xxx.xxx.xxx is the IP address of the AWS EC2 instance):

```
$ ssh -i mylock.pem username@xxx.xxx.xxx.xxx
```

**On Blackboard under the *Misc* content area, a tutorial on connecting to the AWS EC2 has been posted.**

**Screen shot:** Once you are logged into your UNIX account, at the command prompt, execute the following command and take a screen shot.

```
$ whoami
```

With future Labs, embedding screen shots image files within your lab write-up is critical. Therefore, it is important to determine an efficient method of taking screen shots and embedding them into a document..

### Step 3: Use the vi/vim editor

**If you are already familiar with vi/vim, you can skip this step.**

The vi/vim will be heavily used for all labs throughout this semester. If you are not already familiar with this editor, please take the time to become familiar. On jaguar.ecst.csuchico.edu, vi and vim refer to the same editor. The history behind this is that vi used to be the common editor and now vim is the latest release. A key difference with vim is that it displays code text in different colors for better readability. So vim is the actual editor being used on jaguar.ecst.csuchico.edu but initiating the editor with vi works OK for backward compatibility.

The task is simply to create a new data file, edit it, and display it. Knowing how to do this early during this semester will greatly help. Within the steps below, do not include the double quotes characters with your data entry.

- a. At the Unix command prompt, enter “vim datafile”
- b. Click on the “i” key to toggle into Insert mode.
- c. Type in the number “2345”
- d. Click on the “escape” key to toggle into Command mode.
- e. Type in “:wq” which will write your new text to the file and quit.
- f. Re-open the file again by entering “vim datafile” at the Unix command prompt.
- g. Click on the “i” key to toggle into Insert mode.
- h. Type in the number “1” which will insert the “1” character at the start of the line.
- i. Click on the “escape” key to toggle into Command mode.
- j. Type in “:wq” which will write your new text to the file and quit.
- k. Display the contents of your new file by entering “cat datafile” at the Unix prompt.
- l. Take a **screen shot**.

Review the **viCheatSheet** document posted on Blackboard under the **Misc** content area. This is a two-page document showing the more common UNIX commands.

### Step 4: Unix Environment

- a. Once you log into your account, let’s start by displaying all your environment variables by typing the **env** command as:  
**\$ env**
- b. Display the value of the \$HOME environment variable.  
**\$ echo \$HOME**
- c. From the output above, what is the first character?
- d. Create the following directory structure using the full/absolute path names  
**\$ mkdir \$HOME/dir1**  
**\$ mkdir \$HOME/dir2**  
**\$ mkdir \$HOME/dir1/dir3**
- e. This time navigate the directory structure using the **cd** command and use the relative path names to create directory dir4 under the dir2 directory and dir5 under the dir3 directory.
- f. What is the difference between a full and relative path name?
- g. Use the **touch** command to create an empty file named *myfile* and then use the **ls -l** command to display the file’s metadata.
- h. Take a screen shot(s) that shows completion of the above and make sure to embed your answer to the questions above.
- i. Display the contents of the environment **PATH** variable as:  
**\$ echo \$PATH**

- j. What does it contain?
- k. Use the *whereis* command to locate where the *ls* executable resides.
- l. Within your current shell environment, temporarily change the contents of the **PATH** variable by deleting the directory where *ls* resides. In other words, use the syntax **similar to** the following to change a new list of directories in order to omit the directory where the *ls* command resides:  
`$ export PATH=/usr/local/bin:/blahblahblah:/sbin`
- m. Can you explain how the *export* command works?
- n. Try the *ls* command and notice that its execution will fail.
- o. What will happen if the *ls* executable existed in two different directories referenced in the **PATH** directory? Which one would execute? The next steps will help provide the answer.
- p. Since the contents of the **PATH** environment got fouled up with the earlier export command, let's reset it by logging out from the UNIX server and logging back in. This will cause your initialization profile script to re-execute and hence, to reset the contents of the **PATH** variable.
- q. Create a dummy Bash script named *ls* which simply displays the current date with the following 2 lines and save it in your **HOME** directory.  

```
#!/bin/bash
date
```
- r. Convert the *ls* file created in the previous step into an executable with the following command:  
`$ chmod 700 ls`
- s. Can you explain how the *chmod* command works?
- t. Next, add your **HOME** directory to the first slot in the **PATH** variable and try running the *ls* command. Which *ls* executable ran? Take a screen shot.
- u. Next, move your **HOME** directory from the first slot of the **PATH** variable to the last slot. Which *ls* executable ran?
- v. Take screen shots and embed them (plus your answers) within your lab write-up.

## Step 5: More Unix

Write a Bash script which does the following:

- a. Creates a text file with 1000 lines and name the file *myfile*. The contents of the file should be integers starting at 1 and ending at 1000
- b. Splits the file into 5 separate files using the *head* and *tail* Unix commands
- c. Recombine the file without the “middle” part (range 401-600) of the file. The recombined file, without the “middle” part, should end up with the same filename as the original full *myfile* file.

## Step 6: File transfer

Your next step is to perform the transfer of files between the AWS EC2 instance and your local computer. The transfer should be completed in both directions. There are several popular and good graphical-based software applications available such as WinSCP (for Windows), FileZilla, and gFTP. For Mac users, you can use some of the apps such as MacFusion and Cyberduck.

From a Linux-based platform, other options to transfer files are either *scp* or *sftp* such as:

```
$ sftp username@xxx.xxx.xxx.xxx
```

It is of interest to note that *sftp* is a newer protocol that extends the transfer capability to resume transfers, list files, and create/remove remote directories

**Multiple screen shots:** Take a before and after screen shots of a file successfully transferred to and from jaguar.ecst.csuchico.edu. The screen shots should clearly indicate the command lines used or describe if you performed a drag and drop (windows-based).

## Step 7: Lab submission

Given the steps mentioned above, your deliverables are the following:

1. Start by pasting the screen shots stated above for each step into one continuous document that conveys proof of success.
2. Spend much time on adding your personal comments within your lab submission. Refer to syllabus and example as shown in class. Make sure to include embedded comments within your lab write-up. Anyone should be able to read your lab write-up and understand what it is trying to present. **DO NOT SIMPLY TAKE SCREEN SHOTS.** Include highlights and arrows to describe specific points.
3. Add a conclusion section to your write-up. Elaborate on at any aspect of your lab. Some examples are: what was challenging, how were you able to overcome any hurdle(s), and/or what did you learn.
4. During the Activity lab time, it is mandatory to show your work to the instructor. Otherwise, no credit will be given. Hence, mandatory Activity attendance will be enforced.
5. Turn in **one** PDF/Word/OpenOffice document). No individual image files. And no zipped, no compressed files, etc.

Please refer to the required format example by reviewing the following document posted on Blackboard under the **Misc** content area named ***SampleLabWriteup***. This is a good example of the lab write-up format. Notice the different embedded highlights and comments.

## GRADING

To achieve a maximum score, students will need to clearly prove that they completed the goal. A clear description of the steps taken and screen shots are essential.

- Partial credit may be given if the write-up clearly states what was done and what is not working. If the instructor is required to decipher what was and was not done, much less partial credit will be given.
- Points lost for incompleteness, sloppiness, lateness, or failure to follow instructions.
- Refer to syllabus for late policy.