

CSC 2510: DevOps

Lab 03 – CLI III

General Instructions

Using your book and previous lecture material, fill out this assignment sheet. **Use red text to signify your answers.** Use information from your textbook and the provided lectures to answer the questions on this lab. You may utilize online resources to answer these questions if you cite them.

Submission Instructions

To submit, **change the name in the header** and save this document as a PDF. Attach your PDF document to the iLearn dropbox.

Lab Directions

Description

The vim program is a full-screen text editor that can almost certainly be found on all UNIX systems. vi does not use a large amount of system resources and yet it is very powerful. We are only going to touch on vi basics so that you will be able to create and edit files in the coming lessons.

There are two modes in vim. The first is input mode. In *input* mode, text is entered in the document. You can insert or append text. The second mode is *command* mode. When vim is in command mode, you can move within the document, merge lines, search, cut, and so on. You can do all of the functions of vim from command mode except enter new text. Text can only be entered when in *input* mode.

Getting Started

1. Install NeoVim
 - a. `sudo apt-get update`
 - i. this goes and checks for updates for your system.
 - b. `sudo apt-get upgrade`
 - i. this will download and install the updates found in step 1.a.
 - c. `sudo apt-get install neovim`
 - i. 'neovim' is NeoVim, a newer slimmer package of 'vim'. These two packages are essentially the same and can be used interchangeably.
2. Type `'nvim showShells.sh'`
3. Tap `'i'` on your keyboard to enter *input* mode.

<Won Lee>

4. Type the following into the file:

```
#!/bin/bash
echo "You are running $(uname -s) version $(uname -r)"
for x in ash bash bsh csh pdksh ksh sh tcsh zsh; do
    test -x /bin/$x && shells="$shells $x"
done
echo "You have at least the following shells installed:$shells"
```

5. Press the escape key to enter command mode.
6. Move your cursor to the word "at" in the second echo statement. In order to do this, you should use the arrow keys, or the following: "l" moves your cursor right, "h" moves your cursor left, "k" moves your cursor up, and "j" moves down. Practice moving around until you get to the first "at" in this script.
7. Is your cursor placed before the word? Place it after the word by hitting the "w" key. Now, hit the "b" key. Pretty neat, isn't it? Play around for a minute and then bring yourself back to the beginning of the word "at."
8. Now, press "dd". We erased the whole line! Type "u". It is fixed!
9. Make sure you are located right at the word "at" and press "dw". This should delete the word. Press the period key on your keyboard to repeat the previous command.
10. Now, let's save the document. Press the ESC key to bring yourself back to command mode. Then type ":wq" and hit enter to save and quit.
11. Enter the following command: `chmod u+x showShells.sh`
12. Now run your new shell script by entering the following command: `./showShells.sh`

Questions

1. (3) What does the shell script print out? **You are running Linux version 5.4.0-1087-gcp**
You have following shells installed: bash sh
2. (3) What does the `u+x` mean when used with the `chmod` command? **give execution permission to the user** How would you write this using Octal values? **100**
3. (3) Why did we need to change permissions on the file? **to execute the file as a user and print the shell script**
4. (3) Why is `sudo` used with `apt-get`? **to override the permission as a super user to use the command apt-get to either update, upgrade, or install**
5. (3) What is the difference between `update` and `upgrade`? **update means to show and get what needs to be updated (up-to-date) and upgrade means to actually install those updates**
6. (5) What `vim` commands were used during the lab exercise? Name them and state what they do. **'w' to go to first char of a string left of current cursor in command mode, 'b' to go to first char of a string right of current cursor in command mode, 'dd' erases the whole line that the cursor is on in command mode, 'i' to enter input mode to type/modify text inside the file, moving cursors: 'l' right, 'h' left, 'k' up, 'j' down; 'u' restores previous action, 'dw' deletes the word that the cursor is on, and ":wq" to save and quit**