Steven Tang

CSE 460

1/17/17

Lab 2

**Objective**:

The objective of this lab is to get familiar with a number of UNIX commands. As an addition to learning the commands, we should be able to use it in the command line as well as in a script. We will learn to write a script that will find the user inputted process and terminate all of them if they are running.

* ./ginfo runs the script while .ginfo does not. The ./ directs us to the current directory while just a . is in the current shell. In order for us to run our script, we need to use the ./ syntax.

1. How do you define variable x with value 10 and print it on screen?
2. clear
3. x=10
4. echo $x
5. How do you define variable xn with value “Rani” and print it on screen?
6. clear
7. xn=Rani
8. echo $xn

3. How do you print the sum of two numbers, say, 6 and 3?

1. clear
2. x=6
3. y=3
4. sum=$(($x+$y))
5. echo $sum

4. How do you define two variables x=20, y=5 and then print the quotient of x and y (i.e. x/y)?

1. clear
2. x=20
3. y=5
4. echo $((x/y))

5. Modify the above question to store the result of dividing x by y to a variable called z.

1. clear
2. x=20
3. y=5
4. z=$((x/y))
5. echo $z

* When running ./test.sh and then running echo $XYZ, we get nothing. When we run . ./test.sh and then run the command echo $XYZ, we actually get 2017. This happens because we run ./test.sh, we create a new instance while . ./test.sh does not.

1. When I run ps auxw | awk '{print $1 "\t\t" $2}' it shows all the processes that’s running for the user by printing it to the terminal

The difference when changing system(“ps –ax &”) was that it showed all of the programs that are running in the background.

* Grep –v Shows all the selected lines that doesn’t match the specified pattern. This is called invert-match
* When using top, top consumes the most CPU. It fluctuates between 1.5 and 2%. The one that contains the most memory is mdwork32.

1. process=$(pgrep $1)
2. **if** [ "$process" == "" ]
3. then
4. echo "No such task exists"
5. **else**
6. kill $process
7. Script started on Wed Jan 18 23:49:49 2017
8. [?1034hbash-3.2$ ./robot &
9. [1] 1616
10. bash-3.2$ ./robot &
11. [2] 1617
12. bash-3.2$ ./robot &
13. [3] 1618
14. bash-3.2$ ./robot &
15. [4] 1619
16. bash-3.2$ ./T[KterminateProcess &[Krobot
17. [2]   Terminated: 15          ./robot
18. [4]+  Terminated: 15          ./robot
19. [1]-  Terminated: 15          ./robot
20. [3]+  Terminated: 15          ./robot
21. bash-3.2$ ps
22. PID TTY           TIME CMD
23. 1353 ttys000    0:00.16 -bash
24. 1614 ttys000    0:00.01 script
25. 1615 ttys001    0:00.01 /bin/bash -i
26. bash-3.2$ ei[Kxit
27. exit
29. Script done on Wed Jan 18 23:50:10 2017

**Evaluation**

In this lab, I wrote practice shell scripts and and learned how they work. I clarified the difference between “./ginfo” and “.ginfo”. I answered exercises 1-5 by including my script code. I also denoted the difference between echo $XYZ when using “./testShell.sh” and “. ./testShell.sh”. At the end, I created a short script program that will allow the user to terminate all instances of a program, and show how many was terminated.

**Score: 20/20**