CSC3320 System Level Programming
Lab Assignment 5 - out-of-Lab

Purpose: Learn how to write basic shell script.

In Chapter 2 and 3, you have learned a list of utilities. However, each time we could only type a single command on command line in terminal. It is inconvenient sometimes when a task has to been accomplished by multiple commands. For example, if the task needs to be repeated, you may have to restart the execution of the list of commands by typing the command one by one. For this reason, the shell script file is used to store the commands interpreted by shell. It is more than a regular file containing only the command. You can even write for loop, if else and switch case statement in the shell script. The shell script file can be executed directly by providing the name of it on command line.

Write a report by answering the questions and upload the report (named as Lab5_P1_FirstNameLastName.pdf or Lab5_P1_FirstNameLastName.doc) to google classroom. This lab assignment is related to the slides

#12 to #14 in chapter 4 Part 1:

Now it is your turn to create your first shell script file by following the steps below.

Step 1: Go to your home directory and create a new file named as simple.sh (vi simple.sh or nano simple.sh), then include following lines in your simple.sh.

Question 1): What did you see in the output of step 3?

Bash: simple.sh : Command not found –

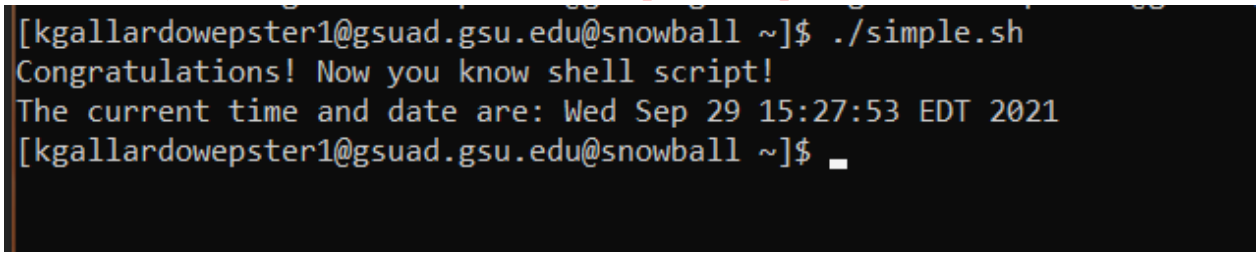
You get that error because the current working directory is not in the PATH.

Question 2): What did you see int the output of step 4?

-Bash .simple.sh Permission denied

The permissions for the file did not include execution permission for the user.

Question 3): Attach a screenshot of the output in step 6.



Question 4): Describe the meaning of -n option in echo command.

The -n option makes the output of the next line appear in the same line where the -n option was used in conjunction with the echo command.

Question 5): Is "Simple Script" a comment? If not, what is the meaning of it or why we use it?

"Simple Script" is a comment. Any line containing # after the first line will be considered a comment.

Question 6): is "#!/bin/bash" a comment? If not, what is the meaning of it or why we use it in first line?

"#!/bin/bash" is not a comment, it is used to determine what shell should be used to run the script.

Question 7): How many directories you can find in the output?

I counted five directories after putting in the command (.:$PATH).

Note: the directories are separated by colon.

Question 8): Can you find errors prompted in step 9? If not, briefly describe why there is no need to put ./ before the line.

There were no errors, the script was able to run. There was no need to put in "./" before the command because with the command "$PATH=.:PATH" you are able to add the current working directory to the PATH.

Question 9): Can you find the current working directory .in the PATH variable?

No, after logging out and logging back in you are not able find the current working directory in the PATH variable.

Question 10) Can you find errors prompted in step 11? If yes, please explain why?

Yes, you get the same error for "command not found" as before adding the current working directory to the PATH variable. This happens because when the current working directory was added to the PATH variable it was not a permanent addition. And once you logged out the Path variables reset to what they were before.

Part 3

```
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ vi checkError.sh
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ schmod a+x checkErrors.sh
-bash: schmod: command not found
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ chmod a+x checkErrors.sh
chmod: cannot access ‘checkErrors.sh’: No such file or directory
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ls
checkError.sh  csc3320  homeworks  Lab3  Lab4  public  simple.sh  test.txt
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ chmod a+x checkError.sh
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ./checkError.sh 002497849
./checkError.sh: line 1: 1/bin/bash: No such file or directory
./checkError.sh: line 3: /bin: Is a directory
./checkError.sh: line 17: warning: here-document at line 9 delimited by end-of-file (wanted `END')
./checkError.sh: line 17: syntax error near unexpected token `newline'
./checkError.sh: line 17: `ech "The regex [^a]*ce can match the string(s):">Result grep '^[^a}*ce$'<<END
>>'
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$
```

The lines 1,3, and 17 have errors in them.