Introduction to Scientific Computing (BSR1015)

Assigned February 14th, Due February 28th

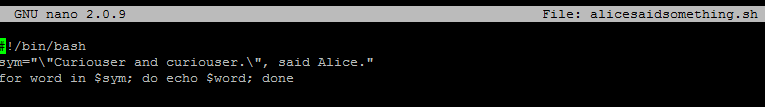
1. Fix the syntax in the following script so that the output is:

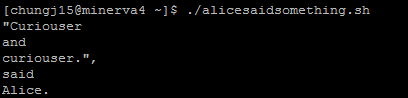
“Curiouser and curiouser.”, said Alice

#!/bin/bash

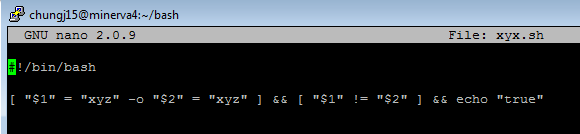
sym=""Curiouser and curiouser.", said Alice."

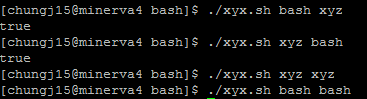
for word in $sym;do echo $word;done





1. You have a script that accepts 2 arguments. What would be the bash conditional test to return true if the first or second argument equals the string “xyz” but not both arguments equal to “xyz”, i.e., an exclusive or (xor) conditional. Use the square bracket [] form of the conditional and only one line.





1. Explain the difference among the use of a trailing &, nohup, and screen.

* This ampersand allows multitasking in one terminal. Adding" &" at the end of the commands makes the commands run in the background in the subshell, while we are waiting for theforeground processto complete. The [shell](https://www.lifewire.com/linux-unix-shell-commands-2180216) does not have to wait for abackground process to complete before it can move on to other processes [1]. The commands run in the background in the subshell are able to immediately return to command lines.
* Nohup, or no hang up is to keep the commands while we exit the shell (close the terminal) [2]. What Nohup does it to provide a separate environment for the child process to shield the child from the SIGHUP, the terminate signal. As the terminals, the input would be redirected from /dev/null, and the output would be redirected to nohup.out, and the standard error would be redirected to stdout and then to the nohup.out. So, when we close the shell, the parent shell exits; the children receive no input from the parent, the background process will still stay alive. Nohup is useful when we run a shell-script or command that take a long time to finish and we want to continue with other work, or when we close the terminal window and open another, the process will still be running [3].
* Screen is the process that allows detachment from the terminal, and we can reattach back later from the same or another computer. We can also restore the screen if we lose connection or disconnect from the **SSH** session. Screen allows multiple shells running in one terminal window, so we can share my sessions with other users by the detach/attach functions.

[1] Lifewire

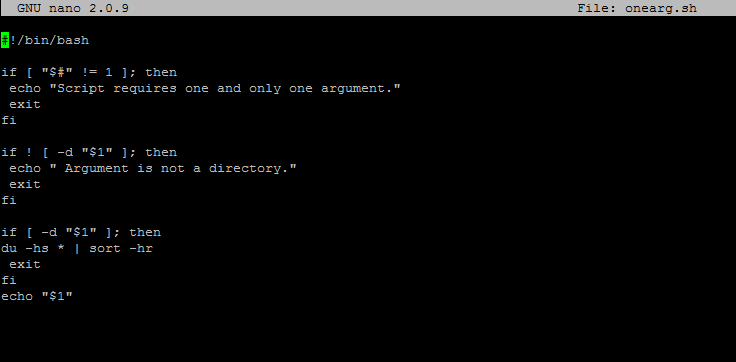
[2] ComputerHope

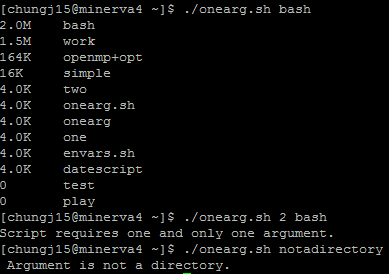
[3] DigitalOcean

1. What would be a command line to return the number of files in /usr/include that are directories?

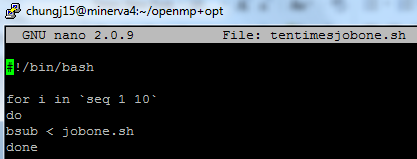


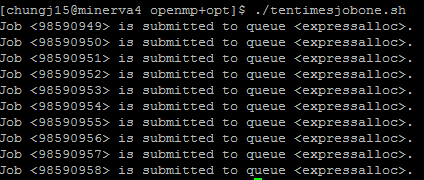
1. Write a script that takes exactly one argument, a directory name. If the number of arguments is more or less than one, print “Script requires one and only one argument”. If the argument is not a directory, print “Argument is not a directory”. For the given directory, print just the names and sizes of the files in that directory.





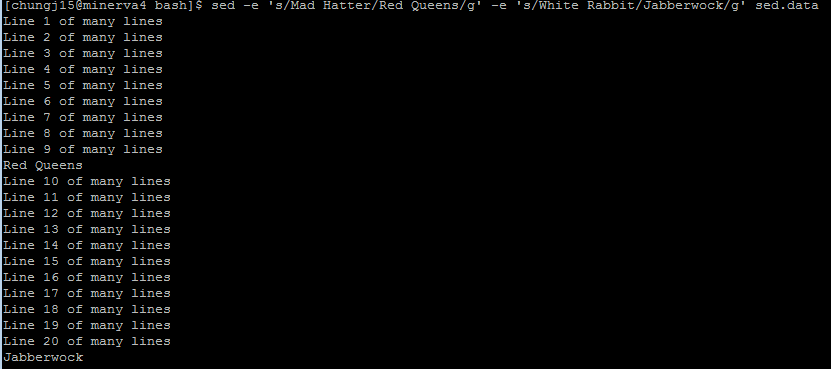
1. Write a command line script that will submit the jobone.sh script you wrote in class 4 10 times.





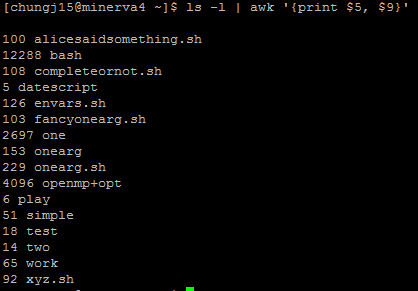
1. Write the sed command that will change the string “Mad Hatter” to “Red Queen” and “White Rabbit” to “Jabberwock” in the file sed.data.

$ sed -e 's/Mad Hatter/Red Queens/g' -e 's/White Rabbit/Jabberwock/g' sed.data



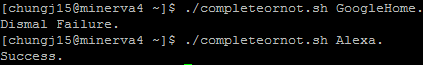
1. Write a command line that will take the output of “ls –l” and output the filename followed by the size.

$ ls –l | awk ‘{print $5, $9}’



1. Write the if test that will check whether a command completed successfully and if so, write “Success!” to the screen and if not, “Dismal Failure” if not.





Source: StackExchange

1. (Research question) Modify the script in question 5 so that it will return an exit code of 0 if the input is only one directory and an exit code of 1 if it is more or less than 1.

