Lab 1-2

Module 1: Review

Use *one command line* for each step below, and don't create any temporary file

1. Create a link to the cis18c directory on voyager, if you haven't already done so
2. Start the script utility to capture your screen output
3. Run a command to show that you have a link to the cis18c directory
4. Print the line of text: `Found <your full name> among <N> entries of /etc/passwd`
   where `<your full name>` is as it appears in the /etc/passwd file, and `<N>` is the total number of lines in the /etc/passwd file.
5. Show the location of the tcsh shell. The output should just be the path name to the tcsh executable.
6. Show the <u>number</u> of hidden files you have in your home directory
7. Copy the file clean.sed from the /home/distribution/cnguyen/cis18c directory to your local directory. This sed script removes control characters from a script output file.
8. Without using vi, append one sed command to the clean.sed script. This sed command will add a # symbol and a space character to the beginning of every input line
   For example, if the input line is: `[cnguyen@voyager ~]$ ls fileA`
   Then the sed command will turn the line into: `# [cnguyen@voyager ~]$ ls fileA`
9. Exit from the script utility. You should now have a script output file
10. Run the modified clean.sed script with your script output file, and redirect the result into a final output file called `lab`. This `lab` file should show your screen output for steps 2-8, with all control characters removed, and with every line starting with a # and space characters.
11. Use vi to check / remove any typing mistake, backspacing, command error in the `lab` file. Keep this `lab` file to continue with the Module 2 part of the lab next week.

Module 2: Basic Scripting Concepts

1. Start with the lab file that you created last week from Module 1 lab. This file will be part of the shell script for Module 2 lab. So far, you've got all of Module 1 lab output in a big comment block.

2. Use vi to add to lab and create a shell script that does the following steps:
    a. Make lab a bash specific shell script, so that the bash shell will be used to run the script

    b. Add a beginning documentation block with your name and the lab number

    c. Print to screen the number of users who are logged in along with a text explanation

    d. Prompt the user and read in a number, which will be used as the number of minutes in the script

    e. Print to screen the user id and the idle time of users who have been idle for more than the given minutes. The idle time can be found in the finger command output. Here's one sample output line (idle time in bold):
    ```
     cnguyen    Clare Nguyen    pts/17     44   Apr 10  16:55  (153.18.26.5)
    ```
    To do this step, you should be using **awk**. Don't use looping or if else statements of the bash shell when awk can keep track of fields for you

3. When done testing, follow the instruction at the link "How to turn in a lab assignment", which is on the class website, to submit a soft copy by the due date/time

Sample output (user input in blue):

```
$ lab1-2
There are 22 users currently logged in
Enter the number of idle minutes: 3
These users are in idle for more than 3 minutes
cheo 9
cheo 16
cheo 26
cheo 9
czhao 3:20
fallaye 5
lsiebert 38
metcalf 2:10
mjdell 17
nica9850 7
yamini 4
$ lab1-2
There are 22 users currently logged in
Enter the number of idle minutes: 30
These users are in idle for more than 30 minutes
czhao 3:21
lsiebert 39
metcalf 2:11
```