BASh Scripting / Git Usage

Useful help for outlab assignment

---

1. Solution for each assignment is to be typed into the .sh file already kept inside the corresponding folder.

2. Remember, you need to change the permission of the .sh file before executing it:

chmod u+x <scriptfile> or chmod 777 <scriptfile>

3. Do not change the structure and the names.. The automatic checker will give you a "notsubmitted" grade otherwise, if it does not find the names.

4. Remember to follow the exact output formats specified. Extra characters will lead to incorrect evaluation by automatic checker.

5. Inside the script if you use any temporary file, write command to remove it after the operation within that script only.

6. For MacOS USERS: Create a file called .bashrc in the home folder (problem related)

Submission Guidelines for outlab assignment

Strictly follow the instructions given below:

1. Create a file readme.txt in <team_name> directory, which contains contribution of each team member and references, along with git usernames.

2. Note the <team_name> must be exactly as on this sheet. Please follow this for all the labs in the future unless specified.

3. Compress the directory to <team_name>.tar.gz

4. Submit one assignment per team.

5. Put all of the folder structure on Git for us to check, provide all three git usernames in the readme file.

6. Please strictly follow the submission format, else there'll be a 10% penalty.

****

==========
Out-Lab Problem Statement:
==========

Problem 1:

You are required to modify your '.bashrc' file (which is a hidden file in your home folder) and create an alias inside it. The alias should be able to get the space usage of all the files and folders and display it on your screen.

The output of your script should display a list of files and folders in the current directory on the terminal (to be 'precise').

You should take a screenshot of the output, when your script is executed in the "home" folder and paste the 'jpg' or 'png' in the folder 'script1'. You are also required to paste the '.bashrc' file in the folder script1/.

>>script1/ls -lh

Output:
>><screenshot_image_name>.<jpg_or_png>
>>.bashrc

We will check this manually and grade accordingly.

Problem 2:

You are ‘yet-again’ required to change your ‘.bashrc’ file to display a welcome message every time you log on to your linux (e.g. ubuntu) terminal. The welcome message must be an ASCII art which can be created using the web-service:

http://patorjk.com/software/taag/#p=display&f=Graffiti&t=sample

The link above allows you to type any text and create ASCII art of the text using special characters on your keyboard. You can add such text to your ‘.bashrc’ file and try logging in again to the terminal. When your welcome message is displayed, please take a screenshot similar to the last statement and paste in the folder script2/. You must also paste your modified ‘.bashrc’ file in the folder script2/.

>>script1/ls -lh

Output:
>><screenshot_image_name>.<jpg_or_png>
>>.bashrc

Problem 3:

Write a bash script to take a path as an input argument. Your bash script should be able to count the no. of lines in all the files present in the path (“directory”) provided. The output of your script should only be the count of the number of lines present (you should not count blank lines in the above files).

From here onwards your assignments shall be graded automatically using scripts, so please ensure the output is simply a number and nothing else (please ensure you do not count blank lines).

E.g.
Path is test/

>>./script3.sh test/
>>180

Problem 4:

A class of students were given a form to fill out regarding team/group formation for evaluating their grades. The TAs of the class did not take into account that they will have to write a bash script to group the teams and a simple Google form / excel input doesn't really cut it for them. The TAs are now in a soup and need your help desperately. Write a bash script to take as input the 'csv (comma-separated value) file' provided to you in the folder structure. The script should be able to group the students as shown below:

Input file:

team1,student1,rollno_student1,student2,rollno_student2,student3,rollno_student3
team2,student1,rollno_student1,student2,rollno_student2,student3,rollno_student3
team3,student1,rollno_student1,student2,rollno_student2,student3,rollno_student3
team4,student1,rollno_student1,student2,rollno_student2,student3,rollno_student3

Output:
team1,student1,rollno_student1
team1,student2,rollno_student2
team1,student3,rollno_student3
team2,student1,rollno_student1
.
.
.
.

The output file must be sorted (dictionary order) on team name, and conflicts have to be resolved on student name (again dictionary order).

E.g.:

>>./script4.sh <inputfilename>

The output of your script must be stored in the file named 'output' inside your folder script4/

This problem statement shall be graded automatically using scripts, so please ensure your output is contained in the file name ‘output’ and it is sorted based on team names (column 1).

HINT: you should be able to use the commands sort/cut for doing this assignment problem.

P.S.:
This is a work of fiction. TAs, Students, faculty, class, lab and any other incidents are either the products of the author's imagination or used in a fictitious manner. Any resemblance to actual persons in the input data, living or dead, or actual events is purely coincidental.

Problem 5:

You are a nerd-to-the-core and want to know the current courses offered by the department on your terminal. You very well know that using the command ‘wget’ and adding a few flags/options will get you the html data of the webpage on terminal.

The challenge here is to modify your “.bashrc” file and create an alias which would enable you to simply fire a command on your terminal to show you the current courses offered by the department on the screen.

As an output to this program we would like to see a screenshot similar to the problems 1 and 2 above placed in your folder named script5/ along with the .bashrc file.

Hint: Recognising the common pattern for course naming and a simple grep command would do it.

The expected output is shown below. Remember no one likes to see duplicates.

CS 101
CS 152
CS 154
CS 213
CS 218
CS 224
.
.

The above lines if displayed on your screenshot should be enough, but the use of ‘.bashrc’ to create an alias is necessary. Place the .bashrc and screenshot in script5/ .

Problem 6:

You aim to be a famous English rapper in life and as over-ambitious as it may seem, you have volunteered to rap “The Real Slim Shady” by ‘Eminem’ on stage during “Surbahaar”. You are surprised(read: shocked) to suddenly know that your parents/faculty/Dean(SA) will also attend the show. You can make do with your “Dhinchak Pooja” like voice but you definitely cannot sing the actual lyrics of the song in front of the current audience.

One of your friends (read: a TA of this class) was gracious enough to send you a list of cuss words used in the song.

Write a bash script which takes two input arguments as file-names. You bash script should be able to replace the cuss word (case insensitive) with “bleep” in the lyrics file provided. The output of your bash script should be in a file named ‘output’. To be precise, let the first input argument be your ‘lyrics’ file you need to clean, and the second input argument be ‘cuss-word list’ file (one cuss word per line).

P.S.: Later this will help you take a printout of the ‘output’ file and give your “dhinchak” performance.

P.P.S: This problem statement comes with a disclaimer, “People who are easily offended by cuss words/slang usage/offensive rap should not read the lyric file provided”. Simply replace the words, and submit the script and the output file in the folder.

Problem 7:

Thor has decided to write a biography. While writing, he has used ‘Thor is mighty’ in multiple places. Hulk reads the text and wants to do some mischief. He decides to substitute the word ‘mighty’ with ‘dumb’ so that the phrase reads ‘Thor is dumb’.

Unfortunately, Hulk has no idea how this can be done using bash script, since he is new to programming and scripting. As a result, he needs help and you happen to be a passer by.

“Hulk threatens to SMASH you if you don’t help him.”

Help Hulk (rather save yourself) and replace the word ‘mighty’ with ‘dumb’ in the text present in the input file. The output should contain the same text with ‘mighty’ replaced by ‘dumb’.

(basically just replace the word “mighty” with the word “dumb”, and make sure all the occurrences of “mighty”; irrespective of the case; e.g. “Mighty”, “mIghty”, … are substituted by “dumb”).

Write your script in ‘script7.sh’. This file ‘thorBiography’ is present in the directory. The sample output file is provided, however we’ll be checking on other input files also.

(Hint: You can use a command called ‘sed’. For more details, use ‘man’ or search online)

$./script7.sh thorBiography (execution sample)

----------==========----------

MANDATORY, *ji haan* , VERY MANDATORY!

PUT ALL THIS ON GIT and provide your username in readme file for us to check.

----------==========----------

----------==========----------

Extra-credit assignments (for enthu junta/punters!):

----------==========----------

We know this already seems hectic but in case you have some time left, and are enthu to learn more, you can do

1. Bash modification: You know BASh since you are using it now, but do you know “ZSH”, “Oh-my-ZSH” and other bash themes. You could look here, if a fancy bash (with more features and much better looking) interests you:
https://github.com/robbyrussell/oh-my-zsh

2. Googler: Do you know that you can Google stuff over terminal itself (for the ultra-geeks) ? Try this:

https://github.com/jarun/googler

The final directory structure:

<team-name>.tar.gz
-- team-name/
--script<1-7>
--extra/
--readme.txt