JOMACS BASH SCRIPTING ASSIGNMENT

Submission Deadline: 18:00GMT Tuesday, August 22, 2023

Submission Procedure:

1. You will submit the assignment using the **scp** command from your working environment (local or on AWS).
2. On your local or AWS server, in your home directory, create a directory and name it in this structure **full_name** (use underscores to separate your names).
3. Inside the created directory, create **.sh** script files for each question by naming them in this format **question_1.sh**
4. Complete and test the scripts for all the questions.
5. Come to your home directory and now use the **scp** command to forward your assignment directory to my server for evaluation.
6. Syntax of **scp** command to use

```
scp -i aws-linux-test.pem -r assignment_directory_name
ubuntu@3.142.151.124:/home/ubuntu/assignments
```

Explanation of commands

- **scp** - secured copy (a linux command, **man scp** , **curl cheat.sh/scp** or **google** will give more information).
- **-i aws-linux-test.pem** - the **-i** flag is identifying the **aws-linux-test.pem** as an identity file (keypair) in order for you to gain easy access to my server.
- **-r** - the **-r** flag is also being used to inform bash that the next argument, that is, **assignment_directory_name** is a folder/directory so it should copy it recursively to the destination server. The **assignment_directory_name** is to be replaced with the directory name which houses your bash scripts
- **ubuntu@3.142.151.124** - this is indicating the destination username and hostname/servername/ip address.
- **:/home/ubuntu/assignments** - this is an important file path (absolute) that will guide scp where to place the item being copied in the destination server. Failure to bring it, command will throw an error.

  Take note, commands and syntax are case-sensitive. Pay particular attention when typing out commands or scripts.

NB: If you have a connection timed out whilst using the scp, let us know in the Slack channels.

<u>Use the below guidelines to create scripts for questions 1 - 3</u>

Kofi Ezedike is a newcomer to Canada, and he opened his first checking account at RBC. He also applied for a credit card at the same bank where he was approved for a credit limit of **$2000.** The credit card is ending in **XXXXXXX4412** with CVC of **113** which expires on **05/26**

Question 1

Kofi needs his last four card number and CVC to be able to activate his card and set up a pin. Write a bash shell script to direct Kofi on how to activate his card at an RBC cash machine. For courtesy, don't forget to welcome Kofi to RBC before giving out the directions. Also, he must be made to know that he needs to insert his card to start. Make sure the pin is entered twice and both entries cross-checked with each other verify they are the same. The entering of the password should be invisible.

Question 2

Assuming Kofi's pin is 1957, write a script that directs Kofi on how to use his card for the first CNP transaction at Udemy.com. For a transaction to be successful, he needs to enter his postcode T4E5J1 and CVC correctly after entering his 16-digit card number (come up with the first part of the 16-digit number but ending numbers must be 4412). If not, the transaction should be declined, and Kofi's card blocked with an onscreen message advising Kofi of the same.

Question 3

Given Kofi's approved credit limit, write a script that automatically approves any transaction less than or equal to his credit limit. For all transactions above the credit limit, they should be declined. Let Kofi know whenever a transaction is declined or passed.

Question 4

Write a Bash script that organizes a given directory.

We have a directory containing; music, videos, images, logs and perhaps other files. Your task is to organize files into categories and get rid of some of them.

Specifically, after your script has made changes, the directory should appear as follows:

- A new directory, "music", should contain all files with ".mp3" and ".flac" extensions.

- A new directory, “images”, should contain all files with “.jpg” and “.png” extensions.
- A new directory, “videos”, should contain all files with “.avi” and “.mov” extensions.
- All files with the “.log” extension should be removed.
- Other files not already mentioned in this list must not be changed.

Note that the script takes no arguments and will be executed inside the directory that has to be organized.

To create the files for the work, include instructions in your script to create the files below;
b2.sh bashass.sh bash.sh black.mp3 bubu.avi dame.mov david.flac dev.log dino.log edochie.jpg efe.log games.avi hunger.mov mapple.jnp naija.jpg nana.mov peace.py pete.png python.py shata.flac sheriff.flac toyin.pdf van.avi wale.mp3 wike.avi wiz.mp3 yul.jpg Yvonne.txt zanku.mp3 ghana.avi

You can use wildcards like (*) for simpler and efficient scripting

Question 5

a) Write shell script that displays “This will exit with a 0 exit status”. Be sure the script does indeed exit with a 0 exit status.

b) Write a shell script that accepts a file or directory name as an argument. Have the script report if it is a regular file, directory, or other type of file. If it is a regular file, exit with a 0 exit status. If it is a directory, exit with a 1 exit status. If it is some other type of file, exit with a 2 exit status.

c) Write a script that executes the command “cat /etc/shadow”. If the command returns a 0 exit status, report “Command succeeded” and exit with a 0 exit status. If the command returns a non-zero exit status, report “Command failed” and exit with a 1 exit status.

Question 6

a) Write a shell script that loops through the /etc/passwd file one line at a time. Prepend each line with a line number followed by a colon and then a space.

Example output:

```
1: root:x:0:0:root:/root:/bin/bash
2: daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
3: bin:x:2:2:bin:/bin:/usr/sbin/nologin
4: sys:x:3:3:sys:/dev:/usr/sbin/nologin
```

b) Write a shell script that asks the user for the number of lines they would like to display from the /etc/passwd file and display those lines.

Example output:

```
How many lines of /etc/passwd would you like to see? 3
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
```

c) Write a shell script that allows a user to select an action from the menu. The actions are to show the disk usage, show the uptime on the system, and show the users that are logged into the system. Tell the user to enter qto quit. Display "Goodbye!" just before the script exits. If the user enters anything other than 1, 2, 3, or q, tell them that it is an "Invalid option." You can show the disk usage by using the dfcommand. To show the uptime, use the uptime command. To show the users logged into the system, use the whocommand. Print a blank line after the output of each command.

Example output:

```
1. Show disk usage.
2. Show system uptime.
3. Show the users logged into the system.
What would you like to do? (Enter q to quit.) 2
14:59:08 up 3 days, 7:36, 7 users, load average: 0.13, 0.22, 0.33
1. Show disk usage.
2. Show system uptime.
3. Show the users logged into the system.
What would you like to do? (Enter q to quit.) 4
Invalid option.
1. Show disk usage.
2. Show system uptime.
3. Show the users logged into the system.
What would you like to do? (Enter q to quit.) q
Goodbye!
```

Question 7

a) Write a shell script to output a given directory's size
b) Create a bash script that has an array of five best Linux distributions. Let the script cycle through and print all the items in the array and the array size.
c) Create a bash script that creates a new ubuntu user. The script should accept name, group name and password of the new user and use it to create an account for the user.
d) Create a script that accepts an input from 1 - 7 and should print the day of the week depending on the number entered. 1 - Sunday and so forth. Use a case statement. If the user enters any other input aside from 1 - 7, print an error message.
e) Create a script that accepts a number and determines if the number is odd or even. The response should be printed to the user.

As an advice, do make use of the **sleep** command to slow the script's execution as and when it is appropriate.
Also, utilise the **set** command to debug your script when you have issues.

Wish you all the best!
Any issues, challenges or clarification, do post in the main channels on Slack for resolution.

Thank you.
Aminu Mohammed Twumasi