CMPS-405: Operating Systems Laboratory

Lab Project

Date: September 4, 2021

Warning! – تحذير!

يعد الغش مخالفة أكاديمية وفقا للوائح والقوانين المعمول بها في جامعة قطر، وقد تصل عقوبة هذه المخالفة في بعض الحالات إلى الفصل النهائي من الجامعة وعلى الطلاب تجنب القيام أو المشاركة في أي عمل يخالف ميثاق النزاهة الاكاديمية وإجراءات الاختبارات المعمول بها بجامعة قطر.

Cheating is an academic violation according to Qatar University rules and regulations, and in some cases, it may result in final dismissal from the University. Students should not under any circumstances commit or participate in any cheating attempt or any act that violates student code of conduct.

1 Phase 1 – Part 1 – Shell Scripting

Scenario

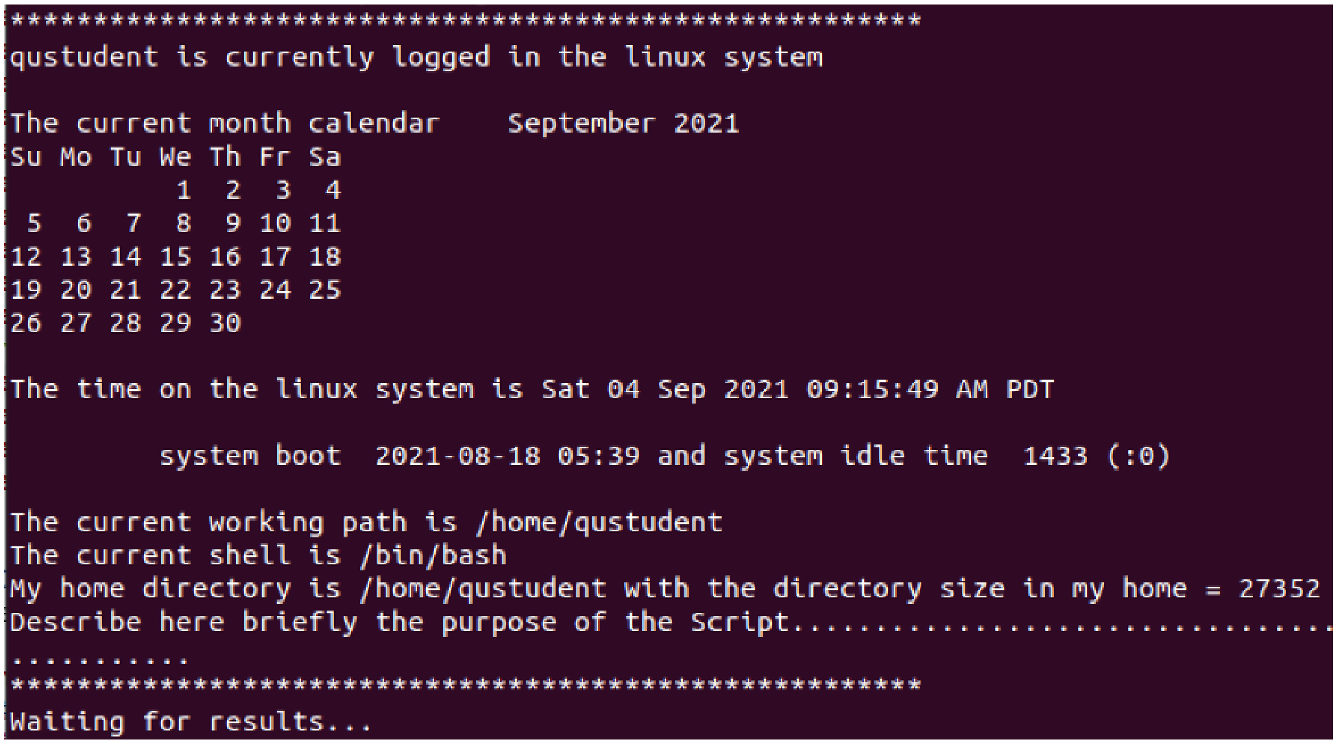
You work as a system admin in a big tech company. The company has several Linux servers to provide different services that power the company. You have been asked to build a set of scripts to log the activity of those servers by keeping track of some important system information over time.

Note that some of the tools below might be different from one Linux distribution to another. But we will assume **Ubuntu 20.04** is being used. This phase is due to **18th September, 2021**

1.1 First Shell Script

Create a shell script called (**Running.sh**). The script must call the following scripts (**FileSystem.sh**), (**Performace.sh**), (**Connectivity.sh**), (**ControlTraffic.sh**) and (**Trap.sh**). The first script **Running.sh** must have two subroutines. The **First Subroutine** will display the following messages on the screen:

- Takes from the user the username and display a welcome username, current day in a calendar format, and the current time.
- Display System boot time and idle time.
- Working path and Shell type
- Display Home directory and the directory size in KB in your Home.
- Message briefly describing the requited tasks.


```
*******************************************************
qustudent is currently logged in the linux system

The current month calendar    September 2021
Su Mo Tu We Th Fr Sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30

The time on the linux system is Sat 04 Sep 2021 09:15:49 AM PDT

         system boot  2021-08-18 05:39 and system idle time  1433 (:0)

The current working path is /home/qustudent
The current shell is /bin/bash
My home directory is /home/qustudent with the directory size in my home = 27352
Describe here briefly the purpose of the Script...............................
...........
*******************************************************
Waiting for results...
```
Figure 1: First shell script sample output

Please note that you must use variables with related messages for the above subroutine.

Second Subroutine must performing the following:

- Running infinitely.
- The script (**Running.sh**) must check the system time. Once the time reaches 11:59 PM, the current script will be calling the two scripts (**Connectivity.sh**) and (**ControlTraffic.sh**)
- The script (**Running.sh**) will also call the following two script (**FileSystem.sh**) and (**Performance.sh**) after 10 minutes from calling the two scripts in step2.
- Any attempt from the user to interrupt (SIGINT, SIGTERM) the execution of (**Running.sh**) will immediately call the script (**Trap.sh**) to catch and ignore signal interrupt to ensure script (**Running.sh**) completion.

1.2 Second Shell Script

Create a second scripts called (**Performance.sh**). The second script Performance.sh will have **three subroutines**.

First Subroutine will perform the following tasks:

- Gets the disk usage of home directory. Get number of used MB (assuming a block is 1K). Sorts the output of the disk usage in reverse order. Save results in a file called Disk_Usage.txt
- Create another file called cpu_inf.txt that contains ONLY information on the CPU specification of the running system including (Architecture, # of CPUs, Model name and vendor ID).
- Write/Save the Kernel output related to memory and hard disk ONLY into a file called memo-

HDMessages_Log.txt.

- Create another file called Message_Count.txt that shows the total word count of memory lines, and the total word counts of hard disks lines from the kernel output/messages [From step 3].
- Add the three files (Disk_Usage.txt, cpu_info.txt, Message_Count.txt) to a compressed tar file called Phase1.tar.gz. Copy the compressed file to a directory. The directory name is the time of the backup. (Example of the Directory name 05553, where 05 hour, 55 minutes, and 03 seconds).

Second Subroutine must check all the files in your home directory for the following:

- Files with read and write permissions for the owner only, make copy of the file, and then redirect that file to a directory.
- The directory name must be the Date and Time of the backup. (Example of the Directory name 20092905553. 20 year, 09 month, 29 day, 05 hour, 55 minutes, and 03 seconds).
- The script should also change the permission of files with read and write permissions to read permission only for the owner of the directory 20092905553 with its contents.

Third Subroutine must perform the following:

- Display the number of files with read permission in the directory 20092905553.
- Display a message indicating the end of the execution.

1.3 Third Shell Script

Create a third scripts called (**FileSystem.sh**). The third script FileSystem.sh must have three subroutines.

First Subroutine will perform the following tasks:

- Insert date and time as follows (**check the example below**) in the first line of the file **OUTFILE.txt**

Example

```
Date/Time of Search: Sep 6 2021 at 18:00:17
```

Second Subroutine will perform the following tasks:

- Searching for files greater than 8MB in your home directory.
- Display the following message on the screen.

Example – 1

```
Searching for Files Larger Than 8Mb starting in /home/HomeDir
Please Standby for the Search Results...
```

Redirect the output to a file called **HOLDFILE.txt**. Test the size of the **HOLDFILE.txt** to find out if any files were found. If the file is empty, display the following info on screen

Example – 1 (Continue)

```
No files were found that are larger than 8MB
Exiting...
```

If the file is not empty, then:

1. Add the content of **HOLDFILE.txt** to **OUTFILE.txt**
2. Count the number of lines found in the **HOLDFILE.txt** and redirect them to **OUTFILE.txt** as follows:

Example – 1 (Continue)

```
Number of files found: 20
```

Third Subroutine will perform the following tasks:

- Display the content of **OUTFILE.txt** on screen.
- Display the following message on screen.

Example

```
These search results are stored in /home/HomeDir/OUTFILE.txt
Search complete...Exiting...
```

Important: Each time the above script runs, the two files **OUTFILE.txt** and **HOLDFILE.txt** are initialized. HINT: Use **/dev/null for initialization**

2 Phase 2 – Part 1 (Cont.)

This phase is due to **2nd October, 2021**

2.1 Fourth Shell Script

Create a fourth shell script called **ControlTraffic.sh** . The fifth script ControlTraffic.sh will have two subroutines"

First Subroutine will perform the following tasks:

- Update and upgrade the system if any updates exist.

Second Subroutine will configure the firewall to perform the following tasks:

- Deny all incoming traffic by default and allow outgoing.

- Allow SSH access.
- Enable and reload the firewall.

2.2 Fifth Shell Script

Create a fifth script called (**Connectivity.sh**). The fifth script Connectivity.sh will have one subroutines. The subroutine will check if the system is online (Connected to the internet). If the system is connected to the internet, then perform the following:

- Ping the gateway with 10 packets each packet is 1KB.
- Query the Domain Name System to qu.edu.qa.
- Save the results in a file called /tmp/**PINGRESULTS.TXT.**

Otherwise perform the following:

- Ping your local machine.
- Display IP configuration and Routing Table of your Machine.
- Store results in a file called /tmp/**NETINFO.TXT**
- Finally reboot the system automatically after 15 seconds.

2.3 Sixth Shell Script

Create a sixth scripts called (**Trap.sh)**. The sixth script Trap.sh must have one subroutine.

The subroutine must check for any attempt from the user to interrupt (SIGINT, SIGTERM [CTRL+C]) the execution of (**Running.sh**) will immediately call the script (**Trap.sh**) to catch and ignore signal interrupt to ensure script (**Running.sh**) completion.

3 Phase 2 – Part 2 – C Programming

Write a C program to benchmark the CPU power of the machine. The program should compute the Fibonacci number of the passed argument.

Write a Makefile to compile and benchmark the C program in the previous part. The Makefile should include two targets. One to compile your C program and another to run it while measuring the time needed for the program to exit.

4 Instructions & Deliverables

1. In MS-Word document submit the following:
    - Cover page includes necessary details of the group members (Student Name, Student ID)
    - In a table format, specify each group member tasks and the contribution percentage into the project.
    - The source code of all shell scripts.

- Copy of the output console view for each shell script.

2. Submit your six scripts with the MS-Word document on the Blackboard in one zipped file no later than **the below due dates**:

   Phase 1 - Due to 18th September, 2021, **Phase 2** - Due to 2nd October, 2021

   Call the Zip file (StuentNAME1_StudentNAME2_StudentNAME3)
3. Add comments to each subroutine of your scripts.
4. Copying and/or plagiarism (-100%).
5. Shell Script does not run (Errors) (-50%).
6. In case of late submission, (-10%) for each day of delay (Max 3 days delay).
7. A group of two to three students can work on the project.
8. Team members are required to meet regularly for discussion and workload distribution.
9. It is the right of the instructor to use any way of testing the student in the discussion and demo session, and according to that in some cases (100%) graded project may be down to (-100%) graded project.
10. Discussion & Demo 50%. + Student Work 50%.

Good Luck‼ ☺
