Assessment 2. Operating Systems programming

Demonstration + online submission; week 15; 15% - group submission

Students will be working in pairs to write script related to the Linux Bash Script features they have learned in their lab sessions, then presenting and submitting it online. The final version of their software will be demonstrated in week 15.

Students will develop a Bash Script program based on a given criteria.

Students will have to write the code up to good programming standards (correct naming, indentation, comments in code), present it to lab tutors in week 15 lab session and submit it online by the end of week 15.

Students are supposed to use parts of shell script programming learned in lab sessions, but they are encouraged to use any new technologies that were not explicitly taught in lab sessions.
Final assessment will be done in lab session of week 15, students will be supposed to bring the written script to the lab session (mail, OneDrive or memory stick), compile it during the presentation, run the code to show it's functional and explain features of the code. Students will be given exact timeslots during the session to be able to accommodate high volume of students. No presence in a given time slot will result in fail mark.

To summarise, project will be marked based on the code functionality, criteria fulfilment as well as code understanding in according to rubric given below.

The Task:

Based on the lab activities as well as external readings, each group needs to write a Bash script that will implement menus using '**dialog**' or any other utility. Create function for each action, e.g., to show Calendar on screen create function show_calender( ). Menu-items and action according to select menu-item are shown as follows:

Menu-Item	Purpose	Action for Menu-Item
Date/time	To see current date and time	Date and time must be shown using infobox of dialog utility or otherwise
Calendar	To see current calendar	Calendar must be shown using infobox of dialog utility or otherwise and it should have the flexibility of choosing any particular date from the calendar to add any relevant information if needed.
Delete	To delete selected file	First ask user name of directory where all the files are present, if no name of directory given assumes current directory, then show all files only of that directory. Files must be shown on screen using menus of dialog utility or otherwise. Let the user select the file, then ask the confirmation to user whether he/she wants to delete the selected file, if answer is yes then delete the file, report errors if any while deleting file to user.
Exit	To Exit this shell script	Exit/Stops the menu driven program i.e. this script

Your shell script should also show the following five system configuration information:

1) Your operating system type,

2) Computer cpu information,

3) Memory information,

4) Hard disk information, and

5) File system (Mounted).

Try to show as much detailed information as possible for each of the above five cases. These five example should be written in a separate shell script file.

In addition to writing the script, each group needs to show that everybody has contributed equally in the project work. There should be plenty of comments throughout the script which would explain all the relevant steps in the scripts.

Item	1	2.1	2.2	3	Fail
Quality of code 15%	Clean code, using of proper naming standards, code fully commented	Code written up to given standards, high amount of comments	Good structure of code with some mistakes, fair amount of comments	Poor quality of code, minimal number of comments	Poor quality of code, no comments
Technologies used 30%	Use of most required elements (if/else statements, while/do loops etc.), using technologies that were not used in the lab	Use of most required elements (if/else statements, while/do loops etc.), no new technologies used	Use of some of required elements(if/else statements, while/do loops)	Use just one of the required elements(if/else statements, while/do loops)	Use none of the required elements(if/else statements, while/do loops)
Fulfilling the task 30%	Achieved or exceeded required functionality, code is running and compiling without errors	Achieved required functionality, code is running some compilation errors are allowed	Achieved most of the required features, code is running some compilation errors are allowed	Achieved part of the required features, code might not compile	Didn't achieve required functionality, code is not compiling
Understanding of the code 25%	Fully understands the code, able to explain it line by line	Good understanding of code, able to explain most of it	Average understanding of code, able to explain at least half of it	Poor understanding of code, not able to explain more than 25% of code	Lack of understanding of code, not able to explain