Assessment Pro Forma

1. Module number	CSN08116
2. Module title	Operating Systems
3. Module leader	Jawad Ahmad
4. Tutor with responsibility for this Assessment	Jawad Ahmad
5. Assessment	Practical Skills Assessment
6. Weighting	40%
7. Size and/or time limits for assessment	1000 to 1200 words for the documentation file
8. Deadline of submission	23:59 08 November 2020
9. Arrangements for submission	Script files and documentation must be submitted via Moodle.
10. Assessment Regulations **All assessments are subject to the University Regulations.**	No exemptions.
11. The requirements for the assessment	See Attached.
12. Special instructions	None.
13. Return of work	Feedback is provided via Moodle
14. Assessment criteria	See Attached.

CSN08116 Operating Systems

Operating Systems
Practical Skills Assessment

The purpose of this coursework is to introduce you to the functions performed by an operating system and to give you experience of the associated command language and the file system facilities.

On completion of this coursework you will be able to:

(a) Use the facilities provided by a Linux-based system and particularly, the file attributes and directory structures.
(b) Use the command language associated with a Linux-based system, including some of the commonly used commands and the commonly encountered structures.
(c) Appreciate the strategy associated with the testing of systems software, and the need to build robust products.
(d) Understand and implement 'best practice' housekeeping procedures to allow for user/system misuse or mistakes.

In every Unix system, you can have a list of tasks scheduled to run at regular time intervals on the system. This is feasible using the "crontab" command. First, you should study the use of "crontab", by executing "man crontab" or "man 5 crontab" (for some Linux systems) in your terminal. For the needs of this coursework the files crontab.pdf crontab5.pdf are being provided.

Your task is to create a script for the shell that you are using (bash) and name it "mycrontab", which will act as a friendly interface for the user who wants to use the "crontab" command, through choices that he can have using a carefully designed menu. **By using this script, the user should be able to check what tasks run periodically and when, add new tasks, edit a task, delete an existing task or delete all tasks in the list.** You should take into consideration that the user is not able to understand the structure of the "crontab" file so it is up to you to help the user understand the information of the "crontab" file, by offering a friendly human readable terminal interface.

For example
$ ./mycrontab

1. Display crontab jobs

2. Insert a job

3. Edit a job

4. Remove a job

5. Remove all jobs

9. Exit

Select > 1

1. <Description of periodicity>1

<Command>1

2. <Description of periodicity>2

<Command>2

.....

8. <Description of periodicity>8

<Command>8

1. Display crontab jobs

2. Insert a job

3. Edit a job

4. Remove a job

5. Remove all jobs

9. Exit

Select > 4

Job to remove > <Command>2

1. Display crontab jobs

2. Insert a job

3. Edit a job

4. Remove a job

5. Remove all jobs

9. Exit

Select > 1

1. <Description of periodicity>1

<Command>1

.....

8. <Description of periodicity>8

<Command>8

1. Display crontab jobs

2. Insert a job

3. Edit a job

4. Remove a job

5. Remove all jobs

9. Exit

Select > 2

<Dialog to get periodicity>

<Dialog to get command>

1. Display crontab jobs

2. Insert a job

3. Edit a job

4. Remove a job

5. Remove all jobs

9. Exit

Select > 1

1. <Description of periodicity>1

<Command>1

.....

8. <Description of periodicity>8

<Command>8

1. Display crontab jobs

2. Insert a job

3. Edit a job

4. Remove a job

5. Remove all jobs

9. Exit

Select > 3

Job to edit > <Command>

1. Display crontab jobs

2. Insert a job

3. Edit a job

4. Remove a job

5. Remove all jobs

9. Exit

Select > 9

$

The communication between the user and the terminal is presented in an abstract way on purpose to give you the flexibility to design your script according to your taste. By studying the format of the "crontab" file, you will notice that it offers a lot of more functionalities. **Your script should not cover all of these functionalities except from the functionalities presented above with the numbers 1-5 and 9 (check what tasks run periodically and when, add new tasks, edit a task, delete an existing task, delete all tasks in the list or exit)**.

Your script should contain comments that explain the functionalities that you have implemented. Along with your script you should submit a documentation file (e.g. a word document). The documentation file should prove the equal distribution of tasks and explain if necessary, code parts that you think are not clarified by your comments. The length of your documentation file should be at least 1000 words up to 1200 words. **Without the documentation file your coursework will not be marked.**

This is a group project. Each group is allowed to have 3 members. You are responsible to pick your team members and register your team at the link provided through Moodle. You are also obliged to demonstrate your work within the dates and times that will be announced. <u>If you do not present your work, you will not be able to claim its ownership, thus failing this coursework.</u>

Submission Deadline: 08 November 2020 at 23:59

Marking scheme:

Documentation	5
Embedded #comments	5
Exception handling of incorrect/missing user input arguments	5
Display crontab jobs	4
Insert a job	4
Edit a job	4
Remove a job	4
Remove all jobs	4
Demo individual performance	5
Total	40