|  |  |  |
| --- | --- | --- |
| A logo with blue text  Description automatically generated | **SCHOOL OF ENGINEERING, TECHNOLOGY & DESIGN**  **ASSIGNMENT GUIDELINES** | |
| **TITLE OF MODULE**:  Advanced Operating Systems (U14553) | | **MODULE COMPONENT:**  **50% of Module** |
| **MODULE TEAM**:  **Donald Shaw-case (module lead)**  Tim Jackson  Vijay Sahota | | **ASSIGNMENT CONTACT**:  Vijay Sahota  [Vijay.Sahota@canterbury.ac.uk](mailto:Vijay.Sahota@canterbury.ac.uk) |
| **ASSIGNMENT DEADLINE**:  20th Mar 14:00 | | **EXPECTED FEEDBACK DATE:**  10th Apr  **Location of Feedback:** VIA TURNITIN ON BLACKBOARD |
| **ASSESSMENT TYPE** | | Report + Artefact |
| **Where to Submit:** BLACKBOARD TURNITIN SUBMISSION TOOL  If you experience any problems with this system, then please contact the Computing Administration Team ([computing@canterbury.ac.uk](mailto:computing@canterbury.ac.uk))  **What to Submit:**  **A 800 (Max) word report of high professional standard + Artefact source code** | | |
| **TITLE OF ASSIGNMENT**: “Bash ’n Sims” | | |

**ASSIGNMENT INSTRUCTIONS**

This is an individual submission and must be your own work.

There is one ***large programming task with its compulsory write up, all parts must be completed.***

**General advice**

* You are required to back up your work regularly onto one drive and on removable storage devices. Always check the date-stamp on your files before submission.
* You **must** submit your work using the software versions we currently have on the University’s network.

**TASK:**

You have been tasked to create a menu system which is for a simple queue simulator.

The menu is a simple one with only 2 options (plus other commands to exit) that will allow you to choose between the simulation of FIFO or LIFO queue.

The menu systems will be a bash script which, upon selecting a valid option will launch another separate script for the respective option.

Once a simulation is completed the user is returned to the initial menu. The user also has the option to exit the systems simply typing/ issuing the command ‘Bye’ at any prompt followed by a confirmation (Y/N) prompt.

This system will operate separately from the host systems (OS) username/ password system and will have separate username/password verification prior to the menu loading. To this end you will need to create an admin script that will have the ability to create usernames, passwords and PIN which will store this information to an existing ‘UPP.db’ file.

Note that there must also be an ability to delete/update existing users too, which will only work when given the correct PIN.

Creating users/ deleting users can only be executed by the admin script, whilst changing the password can be done by both admin & user (after pin verification).

Both the username and password must be 5 characters long (alpha numeric only), with the PIN being a 3 digit number. On creation of a user (which is unique) you are then prompted to enter their new password twice, the script will check if the same password was entered twice (and prompt to try again if wrong, indefinitely) - a similar case stands for PIN entry.

Once all the correct details have been validated, the last step is to write the new user details to the ‘UPP.db’ file.

*Hint: the ‘UPP.db’ file can be a simple text file to which you append data to.*

Since it is expected that this simulator will be used by many users, your systems should also keep a record of all user activity, specifically when they logged onto the system, how long they used the system for and which simulators were used. All this information should be stored on a ‘Uasge.db’ file (*which may also be a text file*). An additional admin script file is also need such that the admin can gain useful information on a per user/ per simulation bases; specifically total time used by a specified user, most popular simulator used (per specified user), most popular simulator overall and a ranking list of the users who have used the system the most.

As this system will simulate FIFO/LIFO simulation data is required. The presences of a simulated data file is needed, this will be stored in ‘simdata\_<username>.job’ file. For this assignment, we will only simulate a queue of 10 bytes of data (B0-B99) which represents a byte task in the queue.

Example:

B00, B99, B89, B33, B55, B01, B29, B18, B10, B11,

When a user successfully logs in, the system will check to see if this file exists (else create one) and prompts the user to if they would like to use these pre-defined sim data (if initially existed) or enter a new set. Once the user has decided, these configurations will be saved (overwritten) and passed on to the selected scripts for simulation.

The implementation of the scripts for FIFO and LIFO will require independent research, with marks awarded for functional correctness.

In addition to the coded solution, you are required to provide a short informal write up (800 words) describing how **key aspects of code meet the requirements** for this task.

Turnitin submissions are now anonymised. So that your source code can be matched to your write up, name the zip with the following convention….

<the date number + month number + year number of your DOB> X <random number between 111-999>.zip

Example, DOB 1st Jan 2025, random number 234

= (1+1+2025) x 234 = 474318.zip

Then on your written submission, state the name of the zip, so that the correct code can be marked along with your write up.

1. You are to upload your written part onto Bb (Turnitin), a bucket/ link will be provided in the same location where you downloaded this assignment.
2. You are to upload the code zip part onto Bb (Turnitin), a bucket/ link will be provided in the same location where you downloaded this assignment.

**Failure to provide a write up will result in ZERO marks.**

**Failure to provide an adequate write up will result in SCALING of marks.**

**ALL CODE TESTED ON LAB TINY CORE IMAGE**

**THE USE OF Awk, Sed AND BUILT IN MENU GENERATOR IS NOT ALLOWED – NO MARKS WILL BE AWARD IF USED FOR THAT SPECFIC SUBTASK(S)**

See mark scheme to find out more exacting details.

Mark scheme:

|  |  |
| --- | --- |
| **Task:** | **Mark** |
| Display Menu using entire screen | 1 |
| Select correctly & launch on a clean screen | 1 |
| Launch separate menu (place holder) files | 1 |
| Ability to exit at any prompt via the “Bye” command (case insensitive) | 2 |
| Prompt users if they really want to exit via a Y/n option (case insensitive) & repeat question until a valid choice is made | 2 |
| After completion of a selected simulator user returns back to menu | 1 |
| Suitable comments in code (comments must not exceed code) | 2 |
| Usage of colour in the menu system | 1 |
| Validate user correctly on entry (login correctly) | 2 |
| Validate user password on entry (login correctly) | 2 |
| Usage of appropriate functions (implemented) | 3 |
| Correct usage of Global/Local variables | 2 |
| Simple animation/ loading bar when menu start/ exits & selection of simulator | 1 |
| Create separate admin script to add/ modify users to a database file | 2 |
| Usernames and passwords are case insensitive & validated | 2 |
| On user creation request password/pin twice & check if both entries are the same | 1 |
| Ability to delete user given correct PIN | 1 |
| Ability to rest user password given correct PIN | 1 |
| PIN validation on creation | 1 |
| Accurate log recorded for user usage | 3 |
| Complete implementation of admin script for usage analysis as outlined | 3 |
| Correct implementation of sim data configuration & usage | 3 |
| Correct implementation of LIFO & FIFO scripts. | 3 |
| Correct and complete implementation (as outlined above) | 4 |
| 800 words write up | 5 |
| **TOTAL** | **50** |

All marks will be provided in line with your Bb –Turnitin submission.

|  |  |
| --- | --- |
| Task 1 (Write up) | |
| **Percentage** | **Description** |
| 75-100 | Complete: Excellent, complete descriptors of code, respective screen shots, code and CLI, all functionality covered. |
| 50-75 | Good: An in-depth descriptor(s) and most covering most of the functionality of the code. |
| 25-50 | Fair: A high level screen shots given (code or cli), with descriptive text and its function. |
| 1-25 | Poor: A basic level screen shots some comments made (not in code). |
| 0 | No Attempt |

**INDICATIVE TIMELINE FOR PLANNING:**

|  |  |
| --- | --- |
| Timeline | |
| **Duration** | **Task/Tips** |
| 1.5-2 days | Task: Get main menu functionality operating. |
| 1-1.5 days | Task: Implement validation code. |
| 2-3 days | Task: Implement queuing codes, and test all system functionality (including new user/ admin scripts) |
|  |  |
| \* | You should be adding to your write up/code as you go along so as to, at any point have material to submit. The suggested timeframes are indicative, you may want to revisits parts and tweak your work – keep this in mind when you plan for this assessment. |

**FORMATTING OF SUBMISSION:**

You are to upload your written report & source code onto Bb (Turnitin), a bucket/ link will be provided in the same location where you downloaded this assignment.

Any **screen shots, tables, figures, charts, illustrations, etc. will not contribute towards the word count**.

Your work must be adequately referenced throughout. Guidelines on using the Harvard Referencing style are available at:

<https://www.canterbury.ac.uk/asset-library/library/harvard.pdf>

The report must be submitted using the dedicated Blackboard grade centre submission bucket on or before the submission deadline.

**LEARNING OUTCOMES ASSESSED (Fully or Partially):**

**Learning Outcomes of this module:**

1. ~~Evaluate critically different operating systems in terms of their technical characteristics, performance, reliability, and related aspects.~~
2. Exploit tools and techniques that allow administrators to manage the operating system efficiently and critically evaluate their efficacy.
3. Demonstrate an advanced understanding of input/output; process/threads: creation, process/thread management and scheduling and process/thread communication; file systems: the differing mechanisms for allocating storage space to files in terms of efficiency, performance, and reliability; memory management and virtual memory, and the consequences of various paging and segmentation strategies.

**GRADUATE / EMPLOYABILITY SKILLS GAINED:**

This assessment is an opportunity to develop an understanding of the advance bash, their respective tool at hand and how to use them for complex tasks/ operations.

All tasks build on skills and knowledge required in industry, namely researching a problem (analysis), developing an idea (conception/ design/ communication) and documentation (though informal).

This assessment has been designed to exercise your abilities to work on your own and progress closer to becoming a proficient self-learner.

**PROGRAMMES OF STUDY:**

BSc (Hons) Computing & All routes.