University of Westminster

School of Computer Science and Engineering

7SENG012W Software Development Environments
Assignment Specification (2023/24)

Module leader	George Charalambous
Unit	Coursework:
Weighting:	50%
Qualifying mark	NA
Description	To analyse, evaluate and use the appropriate software tools and develop Unix/Linux shell program
Learning Outcomes Covered in this Assignment:	The coursework rationale is: • (LO1) Critically evaluate, select and use the most appropriate type of software development tools for a specific task. • (LO3) Evaluate, use and construct a range of command line tools appropriate for a standard operating system.
Handed Out:	**13th of November 2023**
Due Date	**7th of December 2023, 1 pm**
Expected deliverables	1 A single report as pdf 2 Your script as a text file
Method of Submission:	Via Blackboard link
Type of Feedback and Due Date:	Written feedback and marks 15 working days (3 weeks) after the submission deadline. **All marks will remain provisional until formally agreed by an Assessment Board.**

Assessment regulations

Refer to section 4 of the "How you study" guide for undergraduate students for a clarification of how you are assessed, penalties and late submissions, what constitutes plagiarism etc.

Penalty for Late Submission

If you submit your coursework late but within 24 hours or one working day of the specified deadline, 10 marks will be deducted from the final mark, as a penalty for late submission, except for work which obtains a mark in the range 40 – 49%, in which case the mark will be capped at the pass mark (50%). If you submit your coursework more than 24 hours or more than one working day after the specified deadline you will be given a mark of zero for the work in question unless a claim of Mitigating Circumstances has been submitted and accepted as valid. It is recognised that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website: **http://www.westminster.ac.uk/study/current-students/resources/academic-regulations**

Coursework Description

- Using Bash create a script that will emulate the behaviour of the **selfish round robin scheduling algorithm**.

- your script should use three required positional parameters:
  - 1st positional parameter: Data file name
  - 2nd positional parameter: The increment integer value of new queue
  - 3rd positional parameter: The increment integer value of accepted queue

- your script should use a fourth optional positional parameter, the quanta number

- The script should prompt the user the option to
  - output to standard output only
  - output to a named text file (if the file exists then it should be overwritten)
  - output to both standard output and a named text file

- The format of the named data file provided by the user should contain the following data for each process.
  1. The process label
  2. The NUT value
  3. The arrival time

Have the script read the data and then emulate and output the behaviour of the algorithm showing for each time interval of 1 quanta, which process is running (R), waiting (W), completed (F) or not arrived (-), with respect to each time step

A typical output should look like:

Time	P1	P2	P3
0	-	R	W
1	-	W	R
2	W	R	W
3	R	F	W
4	F	F	R
5	F	F	F

- Your script should terminate when all the processes complete.

You need to submit your work in 2 items onto the Blackboard submission link:

Item1 a pdf report:

- Include a list of prioritised requirements for your script.
- Include the design of your script.
- Include your commented code in text format (no images)
- Include testing of your script
- Include a critical evaluation of your script.
- Provide a video link demonstrating your application at the end of your report.
- Reference any sources

Item 2 your uncommented script as a .txt file to be tested on the University's Linux server

Coursework Marking scheme

The Coursework will be marked based on the following marking criteria:

Criteria	Mark per component	Comments
Requirements	**10**	Best to tabulate
Design of System	**10**	detailed specification of system
Commenting of code	**10**	Include required comments, do not include unnecessary comments
implementation	**40**	See Table 2 below
Testing	**20**	Evidence of Testing
Evaluation/comparison of built system	**10**	Critical Evaluation
Total	**100**	

Table 1 :Top level breakdown of marks

Detailed Selfish Round Robin code marks allocation

Category	Maximum MARK
1.Read and store data from file	
Testing for correct number of parameters	1
Test that filename is a regular file	1
read data and store in appropriate data structure	2
2.Implement the algorithm for a quanta value of 1	
loop over Time	1
add/set processes that AT == T to the end of the existing list initially from index 0 & set status to W	2
test queues if not empty ad job to appropriate queue	2
Set top process in accepted queue to R decrement NUT	2
Print out process status in order of header	1
test if all process completed -- exit loop	1
test top process if NUT == 0 then set status to F	1
increment of priority values of for each queue	2
move process from new to accepted queue	4
loop and move or set index so that top process is moved to back of queue; STOP when top process NUT > 0	2
3.Output correct sequence to standard output and/or to a file	
correct header on output based on order provided by user	1
display correct symbolism; - : not on system; W: waiting; R: running; F: finished	4
display time	1
display process states correctly under header	2
outputs also written to stdout and named file	2
4. Option for different quanta values	
read 3rd parameter to define quanta level & validate	2
set process to sit at the top of accepted queue to set quanta level	2
adjust the increment of priority based on quanta value for both queues	2
move process if completes before quanta level expires	2
Total	40

Table 2: Breakdown of marks for Selfish round Robin script

Coursework Rubric

Criteria	**Limited** **0– 20%**	**Adequate** **21 - 49%**	**Good** **50 – 60%**	**High Quality** **61 – 80%**	**Very High Quality** **81 – 100%**
Requirements Weighted 10%	Incomplete Requirements provided	Key Requirements provided but no detail	Good level of Requirements covered with some detail	Detailed, Requirements coved	Detailed, prioritized Requirements coved
Design of System Weighted 20%	Very top level of design	Overall simplistic design provided	Detailed design provided with some limitations	Detailed design given with key variables defined with requirements.	Detailed design given with key variables defined with requirements with pre and post conditions. Also included viable alternative to design components.
Implementation Weighted 40%	Little working code with no comments	Incomplete coding, with little functionality Some comments	Functional coding, with some features missing. And some comments	Good clear coding, high level of functionality Good level of comments showing understanding	Very high level of functionality, code robust. Good formal commenting given throughout
Testing Weighted 20%	Some testing carried out, but restricted	Testing tables given but did not cover the whole system	Good testing, tabulated, showing results with screen shots	Detailed testing, tabulating expected and observed results. With screen shots	Detailed testing, tabulating expected and observed results. With screen shots. Including critical scenarios, and points of failure.
Evaluation of built System Weighted 10%	Code did not fulfil the specification with no analysis or comparison	Code mostly fulfilled the specification, with limited analysis and comparisons	Code fulfilled the specification, based on a limited specification. Some critical Analysis with tabulated comparisons given	Code fulfilled the specification, based a good specification. Some critical Analysis given and with tabulated comparisons given.	Code fulfilled all aspects of a detailed and complex specification. With detailed critical analysis provided with detailed qualitative and quantitative metrics used.

Table 3: rubric