UNIVERSITY OF TECHNOLOGY, JAMAICA

SCHOOL OF COMPUTING & INFORMATION TECHNOLOGY

OPERATING SYSTEM (CIT3002) GROUP PROGRAMMING PROJECT

Date Available: August 30, 2019 ***Date Due:*** November 15, 2019

Group Size: 5

A group of SCIT students drinks from a barrel that holds 50 servings of light beer. In order to think the SCIT students need light beer. When a SCIT student wants a drink, they take a flask from the barrel, if it is not empty. The SCIT student can then go away to think. If the barrel is empty, the SCIT Student wakes up the bartender, and then waits until the bartender has refilled the barrel. An unsynchronized SCIT student thread will run the following (pseudo-)code:

```
while True:
get_serving()
drink_and_think()
```

And the bartender will run the following unsynchronized code:

```
while True:
refill_barrel()
```

Any number of SCIT student threads can be running concurrently, but there is only one bartender thread. The problem has the following synchronization constraints:

- SCIT Students cannot get any light beer if the barrel is empty.
- The bartender can refill the barrel only if the barrel is empty. Also, the bartender will not always fill the barrel to maximum capacity.

Providing that the bartender thread can only be awaken three (3) times, code a solution to this synchronization problem that has two processes in the system: SCIT student and bartender. The SCIT student process is multithread and the thread can either be waiting, thinking or drinking. Simulation should also have the capability of pausing. When it is paused, it should display the *Process Control Block (PCB)* including (at minimum) the following elements:

- Process ID
- Thread ID
- Thread State (Refilling, Drinking, Thinking, Waiting, Terminated)
- Process Type (Student or Bartender)
- Light beer required (if student)
- Light beer consumption (if student)
- Wake count (if bartender)
- Turnaround time

In developing solution for simulation, each group has free range in deciding:

- synchronization requirements (must ensure mutual exclusion, progress, and bounded wait)
- the amount light beer each student wants to drink
- the time taken for student to think between drinks

Submission

The project should be done in groups with NO MORE than five (5) students per group. There are two parts to be graded: Mini report (due November 15th) and Demonstration (in week 13).

Mini Report should have

- Description of how simulation works and terminates
- Stating which algorithms were used and why
- Synchronization requirements used
- How solution ensures mutual exclusion, progress, and bounded wait

Late submission: Mark deduction will be applied for days late as follows:

- 10% → 1day,
- 25% → 2days,
- 50% → 3days.

No assignment will be accepted after the third day late, in which case the group will be awarded zero marks for the assignment.

Rubric

	81 - 100%	61% - 80%	21% - 60%	0% - 20%
Program Specifications / Correctness [20mks]	☐ system limit (50 servings) ☐ student drinks before thinking ☐ active thread empty/full ☐ student threads running concurrently ☐ bartender refill ☐ bartender awake limit ☐ pausing capability ☐ PCB minimum			
	No errors and program meet 7-8 specifications.	5-6 specifications are met, Minor details are violated, program functions incorrectly.	3-4 specification met. Significant details are violated, program often exhibits incorrect behaviour.	Program only functions correctly in very limited cases or not at all. 0-2 specifications met.
PCB Data Structure [10mks]	Able to apply required data structure and produce correct results.	Able to apply required data structure and produce partially correct results.	Able to apply required data structure but does not produce correct results.	Unable to identify required data structure or was able to identify required data structure but not applied correctly.
Algorithms [10mks]	Algorithms specified in mini report are all implemented and utilized correctly within program.	One algorithm specified in mini report was not implemented, but all others are utilized correctly.	More than one algorithm specified in mini report was not implemented. Not all implemented algorithms works as it should.	Not seeing any algorithms used that were specified in mini report.
Critical Section [10mks]	Identified the critical section and meets all requirements for its solution.	Identified the critical section but able to meet all requirements for solution.	Only one requirement for the critical solution met.	No solution for the critical section provided
Validation [5mks] *Thread & barrel limits*	Does exceptional checking for errors and out-of-range data.	Check for some errors and out-of-range data.	Does little check for errors and out-of-range data.	Does not check for errors and out-of-range data.
Demonstration [5mks]	Able to explain program design correctly and provide alternative solutions	Able to explain some program design	Able to explain a little program design	Unable to explain program design
Mini Report [10mks]	The report is well written and clearly explains what the program is accomplishing and how.	The report has minor grammatical error but explains what the program is accomplishing and how.	The report explains what the program is accomplishing but not how.	Report is not well written, missing majority if not all information required.

Rate Cooperation of Group Members (check where appropriate)

MEMBERS	Participates actively. Thoroughly completes assigned tasks. [1]	Participates in group. Completes assigned tasks. [0.75]	Sometimes participates in group. Completes some assigned tasks. [0.5]	Participates minimally. Completes assigned tasks late or turns in work incomplete. [0.25]
1.				
2.				
3.				
4.				
5.				

Instructor gives grade for project and grades will be adjusted according to a peer assessment.

Example:

Project Grade = 80/100

Individual Grade = Project grade * avg. peer weight

- 80 *.75 =60
- 80 * 1 = 80