The Edward S. Rogers Sr. Department
of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

COMPUTER SYSTEM PROGRAMMING
ECE454 – FALL 2023
ASSIGNMENT-4

Dues are defined at the end of this guideline.

Objectives

- Understanding/Implementing Parallel Processing through Multi-Threading
- Understanding/Implementing Locking Process

Note- This assignment has two parts (1 & 2) and the following is Part-1. The guideline for the Part-2 will be posted right after the related lecture being provided. Any clarification and revisions to this assignment will be posted on Quercus,

4.1 Implementing Parallel Processing through Multi-Threading (A4 – Part-1)

In this part you will implement a concurrent program for accessing 3 shared resources using Semaphore mechanism. This Semaphore should control, for example 3 shared banking tellers, to be accessed by several banking customers (10 of them) as defined in class. Customers are in a queue and can get service by being removed from the queue. Each service time is random but should be less than a fix amount (for example 30 seconds).
Your program should use multi-threading technique and be written using an object-oriented language, preferably Java. [30 Marks]

The system requirements are as follows:

- Design a proper data structure for the shared 3banking tellers and the semaphore queue
- Define a proper prototype (signature) for each semaphore operation (wait and signal)
- Design a proper threading technique (extending Thread class or implementing proper interface)
- Define a proper interface for input/output in your program (you can use a command line user interface or a graphical one preferably)
- Measure the number of threads in your program with their life time (from the time being created until the time being detached) and save the results in a table
- Measure the execution time of your program (the whole program) by changing the number of threads; Run your program 5 times and find the average among the execution times and save the results in a table

The final report / presentation should contain the following items:

- Small test cases and their sample outputs, representing how your program works.
- You need to make a queue of requests trying to have access to the shared resources (as test inputs). It is recommended to make two scenarios or two queues so that in one of them all requests can get access to the shared resources properly but in the other one, a request in the queue has to wait exceeding the limit time due to the service being provided to another request and it was generated randomly, and so make n error message for the system).
- Make a proper function to display the queue situation as well as the shared resources situation at the same time, for example after each service.
- Full source codes with proper comments at proper points, representing each class and function definition
- Discussion about which concurrent program features are created, used and deleted in the flow of your program
- A readme file of how to run and test your program completely

Deliverables & Submission for Assignment-4 (Part-1) [Total 30 Marks = weight 5%]
-The teams should submit their final report including items above through the submission folder on Quercus and then present their work in the labs (on the same day) to be evaluated by the TAs through Q/A by Thursday Nov16, at 8:59am [20 Marks]
The TAs will also evaluate your partial work (defining the structure of classes, and functions to create /detach the threads), in the Labs on Thursday Nov2, at 8:59am [Partial 10 Marks]
-Make sure to add the names and student numbers of your Team members, and the Lab section your team attends at the top part of your final report. Only ONE final submission is required from each team.
-The teams should save all their answers in a final report (plain text format) under this naming rule:
"PRAxx_Tyy_Az.txt" where "xx" is the Lab section (01 for 12-3pm, and 02 for 9am-12pm), "yy is the Team number and "z" is the Assignment number, e.g. "PRA01_T06_A3.txt" is the name for submission from Team06 for A3 working in the Lab on Thursday 12-3pm.

Good Luck.
Useful Resources:
C, Java References
https://www.geeksforgeeks.org/multithreading-in-cpp
https://stackoverflow.com/questions/58443465/stdscoped-lock-or-stdunique-lock-or-stdlock-guard