Operating Systems
CS 4348

Project #2: Threads

Due Date: Saturday, October 29, 2016

I. Project Organization

This project will study the coordination of multiple threads using semaphores.

You should do the following pieces to complete your project. Each piece is explained below:

• Design	40 points
• Code	25 points
• Output	25 points
• Summary	10 points

Design

The design should consist of two things: (1) a list of every semaphore, its purpose, and its initial value, and (2) pseudocode for each function. The pseudocode should be similar to the pseudocode shown in the textbook for the barbershop problem. Every wait and signal call must be included in the pseudocode.

Code

Your code should be nicely formatted with plenty of comments. The code should be easy to read, properly indented, employ good naming standards, good structure, and should correctly implement the design. Your code should match your pseudocode.

Output

Output will be graded by running your program.

Summary

The summary section should discuss your simulation, any difficulties encountered, what was learned, and results. It should be at least one page in length.

II. Project Description

Language/Platform

This project must target a Unix platform and execute properly on our cs1 or csgrads1 Linux server.
The project must be written in C, C++, or Java.
If using C or C++, you must use POSIX pthreads and semaphores (no mutexes, locks, etc.)
If using Java, you must use Java Threads and Java Semaphores (java.util.concurrent.Semaphore).
You may not use the “synchronized” keyword in Java for mutual exclusion.
You may not use Java data structures that have built-in mutual exclusion.
Other approaches require approval.

Elevator Simulation

In this project threads are used to simulate people using an elevator to reach their floor.

This project is similar to the “barbershop” example in the textbook. The threads to be used are as follows:

Person:

1) 49 people are in line at the elevator at the beginning of the simulation (1 thread per person).
2) Each person begins at floor 1.
3) Each person randomly picks a floor from 2 to 10.
4) A person will wait for an elevator to arrive at floor 1.
5) A person will board the elevator only if there is room.
6) Once at the destination floor, the person exits the elevator.

Elevator:

1) There is 1 elevator (1 thread for the elevator).
2) The elevator can only hold 7 people.
3) The elevator begins on floor 1.
4) The elevator leaves after the 7th person enters.

Main

1) Creates all threads and joins all person threads.
2) When last person reaches their floor, the simulation ends.

Other rules:

1) Each activity of each thread should be printed with identification (e.g., person 1).
2) A thread may not use sleeping as a means of coordinating with other threads.
3) Busy waiting (polling) is not allowed.
4) Mutual exclusion should be kept to a minimum to allow the most concurrency.
5) The semaphore value may not obtained and used as a basis for program logic.
6) All activities of a thread should only be output by that thread.

Sample output:

Your project's output should match the wording of the sample output:

```
Elevator door opens at floor 1
Person 0 enters elevator to go to floor 5
Person 1 enters elevator to go to floor 2
Person 2 enters elevator to go to floor 8
Person 3 enters elevator to go to floor 4
Person 4 enters elevator to go to floor 6
Person 5 enters elevator to go to floor 7
Person 6 enters elevator to go to floor 2
Elevator door closes
Elevator door opens at floor 2
Person 1 leaves elevator
Person 6 leaves elevator
Elevator door closes
Elevator door opens at floor 4
Person 3 leaves elevator
Elevator door closes
Elevator door opens at floor 5
Person 0 leaves elevator
Elevator door closes
Elevator door opens at floor 6
Person 4 leaves elevator
Elevator door closes
Elevator door opens at floor 7
Person 5 leaves elevator
Elevator door closes
Elevator door opens at floor 8
Person 2 leaves elevator
Elevator door closes
Elevator door opens at floor 1
…
Simulation done
```

III. Project Guidelines

Submitting

Submit your project on eLearning. Include in your submission the following files:

1) ‘design.xxx’ where xxx is doc, docx, or pdf.
2) ‘summary.xxx’ where xxx is doc, docx, or pdf.
3) ‘project2.c’, ‘project2.cpp’, or ‘Project2.java’ along with any other source files.
4) ‘readme.txt’ containing:
    a) the complete command line used to compile your program
    b) the complete command line used to run your program
    c) any other details the TA should know

Partial or Missing Submissions

It is your responsibility to upload all of the right files on time. It is recommended that you double-check the files you upload to make sure they are the right ones. Once the deadline passes, changes to the submission are not accepted without a late penalty.

Academic Honesty

This is an individual project. All work must be your own. Comparison software may be used to compare the work of all students. Similar work will be reported to the Office of Judicial Affairs for investigation.

Grading

The written portions will be graded subjectively based on completeness and quality. The code will be graded based on points allocated for each key part of the processing as determined by the instructor. The output will be graded based on expected results.

Resources

The web has many articles on threads and there are books available on threads. The course website also contains example source code.