Operating Systems
CS4348

Project #2: Threads and Semaphores

Due Date: Saturday, April 4, 2015

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 20 points
- Output 20 points
- Summary 20 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 Linux server.
The project must be written in C, C++, or Java.
If using C or C++, you must use POSIX pthreads and semaphores.
If using Java, you must use Java Threads and Java Semaphores (java.util.concurrent.Semaphore).
You should not use the "synchronized" keyword in Java.
You should not use any Java classes that have built-in mutual exclusion.
Any mechanisms for thread coordination other than the semaphore are not allowed.

Post Office Simulation

A Post Office is simulated by using threads to model customer and employee behavior.

This project is similar to the "barbershop" example in the textbook. The following rules apply:

Customer:

1) 50 customers visit the Post Office (1 thread per customer up to 50), all created initially.
2) Only 10 customers can be inside the Post Office at a time.
3) Each customer upon creation is randomly assigned one of the following tasks:
    a) buy stamps
    b) mail a letter
    c) mail a package
4) Times for each task are defined in the task table.

Postal Worker:

1) 3 created initially, one thread each.
2) Serves next customer in line.
3) Service time varies depending on customer task.

Scales:

1) Used by the postal worker when mailing a package.
2) There is only one, which can only be used one at a time.
3) The scales are not a thread. They are just a resource the postal worker threads use.

Other rules:

1) A thread should sleep 1 second in the program for each 60 seconds listed in the table.
2) All mutual exclusion and coordination must be achieved with semaphores.
3) A thread may not use sleeping as a means of coordination.
4) Busy waiting (polling) is not allowed.
5) Mutual exclusion should be kept to a minimum to allow the most concurrency.
6) Each thread should print when it is created and when it is joined.
7) Each thread should only print its own activities. The customer threads prints customer actions and the postal worker threads prints postal worker actions.
8) Your output must include the same information and the same set of steps as the sample output.

Output:

1) Each step of each task of each thread should be printed to the screen with identifying numbers so it is clear which threads are involved.
2) Thread output sample. The wording in your output should exactly match the sample.

Simulating Post Office with 50 customers and 3 postal workers

Customer 0 created
Customer 0 enters post office
…
Customer 9 created
Customer 9 enters post office
Customer 10 created
Customer 11 created
…
Customer 49 created
Postal worker 0 created
Postal worker 0 serving customer 0
Postal worker 2 created
Postal worker 2 serving customer 1
Postal worker 1 created
Postal worker 1 serving customer 2
Customer 0 asks postal worker 0 to buy stamps
Customer 2 asks postal worker 1 to mail a package
Customer 1 asks postal worker 2 to mail a package
Scales in use by postal worker 1
Postal worker 0 finished serving customer 0
Customer 0 finished buying stamps
Postal worker 0 serving customer 3
Customer 3 asks postal worker 0 to mail a letter
Customer 0 leaves post office
Customer 10 enters post office
Joined customer 0
Scales released by postal worker 1
Postal worker 1 finished serving customer 2
Customer 5 asks postal worker 1 to mail a package
Scales in use by postal worker 2
Customer 2 finished mailing a package
Postal worker 1 serving customer 5
Customer 2 leaves post office
Postal worker 0 finished serving customer 3
Customer 3 finished mailing a letter
Postal worker 0 serving customer 4
Customer 4 asks postal worker 0 to mail a letter
Customer 3 leaves post office
Scales released by postal worker 2
Postal worker 2 finished serving customer 1
Scales in use by postal worker 1
Customer 1 finished mailing a package
Customer 1 leaves post office
Joined customer 1
Joined customer 2
Joined customer 3

• • •

Task Table

Task	Time (seconds)
Buy stamps	60
Mail a letter	90
Mail a package (scales are used entire time)	120

III. Project Guidelines

Submitting

Your final project should work correctly on cs1.

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

Academic Honesty

All work must be your own. Comparison software will be used to compare the work of all students. Similar work will be reported to the Office of Judicial Affairs for investigation. They will determine the penalty to be given.

Resources

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