CS 149

Operating Systems

Spring Semester 2015

Department of Computer Science
San José State University
Instructor: Ron Mak

Assignment #3

Assigned: Tuesday, February 17
Due: Friday, February 27 at 11:59 pm
100 points, team assignment

Multithreaded class enrollment

This project will give you experience writing a multithreaded program using C and the Pthreads library. Write a C program that simulates multiple students simultaneously attempting to enroll into sections of CS 149. Simulate each student with a thread.

There are three types of students: graduating seniors (GS), “regular” seniors (RS), and everyone else (EE). Students rush to enroll into three sections of CS 149. GS students have priority over RS students, who in turn have priority over EE students.

Have 75 students “arrive” for enrollment at random times spread over two minutes of real time. Make arrival times whole seconds. Give each student a unique ID and randomly make him or her a GS, RS, or EE with equal probability. Also with equal probability, randomly specify that the student can enroll only in Section 1, only in Section 2, only in Section 3, or in any section.

There are three queues. When a student arrives, enter him or her into the tail of the GS queue, the RS queue, or the EE queue, depending on his or her type. Students leave at the heads of the queues to be enrolled into a CS 149 section.

The three queues work simultaneously, but each queue can process and enroll only one student at a time. Because GS students have highest priority, the GS queue takes randomly 1 or 2 seconds with equal probability to process the student at its head and then enroll the student to a section that he or she can enroll in. The RS queue takes randomly 2, 3, or 4 seconds with equal probability to process and enroll the student at its head, and the EE queue takes randomly 3, 4, 5, or 6 seconds with equal probability to process and enroll the student at its head. Each queue computes a new processing time for each student.

Each CS 149 section has a maximum capacity of 20 students. Drop any student who after being processed cannot enroll into a section, and process the next student in the queue.

The simulation finishes when all 75 students have been processed or all three sections are full.

Your program output

Print a line for each event as it occurs. An event is:

- A student arrives and enters a queue.
- A queue starts to process a student at its head
- A queue finishes processing a student and then either enrolls the student into a section or drops the student.

Begin each event print line with a time stamp, such as 0:05 and 1:12, showing the minutes and seconds. The simulation starts at time 0:00.

Each event print line should indicate what the event is. Identify a student by id and type, such as #12.GS.

At the end of the simulation, print:

- Who is enrolled in each section of CS 149.
- What was the turnaround time for each student (from arrival time to being enrolled or dropped).
- The average turnaround time of each queue.
- Who could not enroll and was dropped.
- Any other statistics you find interesting.

Tips

Besides the student threads, what other threads should your program have? Identify the critical regions. What process synchronization is necessary?

Extra credit (up to 10 points)

Students are impatient! A student gives up and leaves after waiting 10 seconds in a queue before being processed. At the end of the simulation, print which students gave up and left. (You may need to adjust this impatience time so that only a small number of students give up and leave.)

What to turn in

Create a zip file containing:

- Your C **source files**.
- A text file containing **output** from a simulation run.
- **A 1- or 2-page report** that describes your software design. What threads did you create? What data was shared and what were the critical regions? What thread synchronization did you use? **Note clearly in your report if you did the extra credit.**

Name the zip file after your team name and email it to ron.mak@sjsu.edu.

Your email subject line should be: **`CS 149-`*section* `Assignment #3` *team name***
CC all your team members so I can "reply all" with your score.