California State Polytechnic University, Pomona
Computer Science Department
CS4310: Operating Systems
Spring 2019
Instructor: Prof. Tony Diaz

Program 2 - Due Mar. 18

(Total: 100 pts)

A dorm on campus wants to show off its commitment to equality by ending the long-standing practice of gender-segregated restrooms. However, as a concession to tradition, it decrees that when a woman is in a restroom, other women can enter, but no men, and vice versa.

The goal of this assignment is to give you experience writing multi-threaded programs. In your system, each person arriving at a restroom will be represented by a thread:

```
OnePerson(int id, int gender, int time) {

Arrive(id,gender);

UseFacilities(id,gender,time);

Depart(id,gender);

}
```

In the code above, *id* is an integer which uniquely identifies each person (ids will be assigned sequentially based on their arrival as 1,2,3,...), *gender* is either 0 or 1 for men and women, *time* is the amount of time the person will stay in the restroom (assume that every person spends 5 seconds in the restroom (I know this is unrealistic but go with it). At most 3 people can be in the restroom at a time.

Implement a "fair" policy that imposes a limit of 2 on the number of people that can enter the restroom while persons of the opposite gender wait to enter. NOTE: if fewer than 3 women (or men) are currently in the restroom and if no men (women) are waiting to enter, more women can be allowed to enter. In other words, your policy should be smart enough to provide fairness if there are people waiting of both genders but should never force a woman (man) to wait to enter if there are no men (women) who want to enter.

Write the procedures **Arrive**, **UseFacilities** and **Depart**, using **mutex locks** and **condition variables** for synchronization. The **Arrive** procedure must not return until it is okay for the person to enter the restroom (it must guarantee that there will either be only women or only men in the restroom and that the **room** limit of 3 is not exceeded). The **UseFacilities** procedure should just delay for time seconds and print out a debug message. **Depart** is called to indicate that the person is ready to exit; **Depart** should take steps to let additional people enter the restroom. In addition, **Depart** should update a shared variable departure index, which keeps track of the order in which the people leave the restroom, i.e., the first person to leave the restroom has departure index 1, the second person has departure index 2, and so on. The **Depart** procedure should also print out the departure index for that person.

NOTES:

- **Your solution must not employ busy waiting.**
- Your solution must use two gender-based queues where people wait until it is okay for them to enter the restroom. In terms of threads, this means that you need two condition variables (or Java objects) where a thread is suspended if it cannot enter the **UseFacilities** procedure.
- For full credit, your solution **must not use** the pthread_cond_broadcast() function (or if you're using Java, the notify all() method), i.e. your solution must only use pthread_cond_signal() or notify() for signaling any threads blocked on a condition variable. NOTE: invoking pthread_cond_signal() or notify() repeatedly in a loop is equivalent to pthread_cond_broadcast() and notify all().
- **You do not have to ensure that people leave the restroom in the same order as they entered it.**
- The debug message printed out in **UseFacilities** should provide a snapshot of the restroom and the gender queues. It is probably also a good idea to print out debug messages in **Arrive** and **Depart**.

In this assignment, you have to run your program for the three arrival schedules given below:

(i) 5 : DELAY(10) : 5 : DELAY(10) : 5 : DELAY(10) : 5

(ii) 10 : DELAY(10) : 10

(iii) 20

Here the numbers indicate the number of people arriving simultaneously at the restroom, while the numbers in parentheses indicate the delay before the next arrival(s). For example, under schedule (i) 5 people arrive simultaneously at the restroom at the start of the experiment, five more people arrive simultaneously 10 seconds after the arrival of the first five people and so on. In each of the three schedules, twenty people arrive at the restroom during the course of the experiment. Note that people arriving simultaneously do not imply that they are all of the same gender. Assume that the probability that a person is a woman is 0.6, you will use a random number generator to determine this.

NOTES:

1. You can use either Java or any multithreaded programming library for doing this assignment.
2. For Pthreads, information on how a thread library call is to be invoked is provided by the corresponding page: https://www.cs.nmsu.edu/~jcook/Tools/pthreads/library.html
3. For Java threads, information on the use of Java's library is provided on the corresponding page: https://www.tutorialspoint.com/java/java_multithreading.htm
4. If you are using Pthreads, use the system call drand48() to determine the gender of the person (according to the probability distribution specified above). The random number stream should be initialized by calling srand48() with the desired seed. If you are using Java for this assignment, use the methods in class java.util.Random to create and use the random number stream that determines the gender.
5. Your code should be well documented and structured, i.e., it should have meaningful comments, meaningful variable names, etc. Your output should make it clear what is happening respect to the entering and exiting of the restroom.

What to turn in:

- A project report analyzing your code and explaining your solution
  (submit it as a .doc or .pdf file to Blackboard)

- A softcopy of all of your source code and the output of runs for the 3 schedules with seed = 9.
  (submit it to Blackboard)