



Assignment 3

[20 marks]

Submission Deadline: SUNDAY, JULY 22, at 11:59PM

The Doctor on Night Duty

A hospital has a doctor who treats patients during her night shift. The doctor has a small examination room with room for only one patient. Attached to her examination room is a waiting room with three seats where patients can sit and wait if the doctor is currently examining another patient. When there are no patients who need help, the doctor takes a nap. If a patient arrives during the night and finds the doctor sleeping, the patient must awaken the doctor to ask for help. If a patient arrives and finds the doctor currently examining another patient, the patient sits on one of the seats in the waiting room and waits. If no seats are available, the patient goes away to have a coffee from the hospital cafeteria and will come back later.

Using multithreading, write a program (using C or C++) that coordinates the activities of the doctor and the patients. Use mutex locks and semaphores to ensure that two patients are not in the examination room at the same time and that the patients waiting in the waiting room get their turn. More details are provided below.

The Doctor and the Patients

Create n patients using PThreads, with each patient running as a separate thread. The doctor runs as a separate thread as well. Patient threads will alternate between having coffee in the cafeteria and being treated by the doctor. If the doctor is available, they will be treated immediately. Otherwise, they will either sit in a chair in the waiting room or, if no chairs are available, will go back to the cafeteria for a random amount of time and will return to see the doctor at a later time. If a patient arrives and notices that the doctor is taking a nap, the patient must notify the doctor using a semaphore. When the doctor finishes treating a patient, she must check to see if there are patients waiting for help in the waiting room. If so, the doctor must help each of these patients in turn. If no patients are present, the doctor may return to napping.

Perhaps the best option for simulating the time that patients spend in the cafeteria as well as the doctor treating a patient, is to have the appropriate threads sleep for a random period of time.

Each patient thread should print its state (drinking coffee, waiting for doctor, getting treated, doctor not available etc.) along with its threadID. Similarly, the doctor should print its state (treating, checking for next patient, sleeping etc.). A sample output is shown on the following page.

Pthreads are covered in section 4.4.1, while mutex locks and semaphores are covered in section 5.9.4 of your textbook.

SUBMISSION DETAILS:

Upload a single zip file containing:

1. your code (consisting of your c or c++ file(s)),
2. an image file (.jpg or .png) showing a screen capture of your program in action, and

3. a readme.txt file containing names of the group members, specific contribution by each team member (if group of 2) and instructions to compile your program including the compilation string.

to the Assignment 2 dropbox on Brightspace. Do not submit the executable.

SUBMISSION DEADLINE:

Sunday, 22-July-2017, 11:59PM

GRADING CRITERIA:

- Code compiles using gcc. Assignment instructions are properly followed. You may get a zero in the assignment if your code does not compile.
- Your application uses mutexes and semaphores properly, and in the correct sequence.
- Output is correctly displayed.
- Code is well written, properly formatted and commented. Adequate error checking is done.

SAMPLE OUTPUT:

Note: The output is meant for guidance pertaining to messages only. Actual sequence of events will vary based on the number of threads, how random numbers are generated, and how scheduler allocates CPU to each thread.

```
Patient 2 drinking coffee for 1 seconds
Patient 3 drinking coffee for 3 seconds
Patient 1 drinking coffee for 1 seconds
Patient 0 drinking coffee for 1 seconds
    Patient 2 waiting. Seats occupied = 1
    Patient 1 waiting. Seats occupied = 2
    Patient 0 waiting. Seats occupied = 3
        Patient 3 will try later
    Patient 3 drinking coffee for 2 seconds
    Patient 4 drinking coffee for 1 seconds
Doctor treating patient for 2 seconds. Seats occupied = 2
Patient 2 getting treatment
    Patient 2 drinking coffee for 2 seconds
        Patient 4 waiting. Seats occupied = 3
Doctor treating patient for 1 seconds. Seats occupied = 2
Patient 1 getting treatment
    Patient 1 drinking coffee for 3 seconds
        Patient 3 waiting. Seats occupied = 3
            Patient 2 will try later
    Patient 1 drinking coffee for 1 seconds
```