Programming Assignment 3

COMP3033 OS 1001-1003 2st semester 2021-2022

This assignment will involve writing a multithreaded program using POSIX **threads**, **mutex**, **semaphore**， **and a monitor**. Once you are finished, write a report, copy the code of your program to the end of the report, and put your name and student ID number on your report. Also, add some pictures of your program running in the report. Write a paragraph to describe how you solve this problem. Did you encounter any problems? how do you fix it? In addition, upload the C file on iSpace.

Late homework assignments will not be accepted unless you have a valid written excuse (medical, etc.). You must do this assignment alone. No teamwork or "talking with your friends" will be accepted. No copying from the Internet. Cheating means zero.

You can do this assignment on a computer running a Linux system.

Description of the Project:

A university computer science department has a teaching assistant (TA) who helps undergraduate students with their programming assignments during regular office hours. The TA's office is rather small and has room for only one desk with a chair and computer. There are three chairs in the hallway outside the office where students can sit and wait if the TA is currently helping another student. When there are no students who need help during office hours, the TA sits at the desk and takes a nap. If a student arrives during office hours and finds the TA sleeping, the student must awaken the TA to ask for help. If a student arrives and finds the TA currently helping another student, the student sits on one of the chairs in the hallway and waits. If no chairs are available, which means all three chairs are occupied, the student will come back at a later time.

Using POSIX threads, mutex locks, and semaphores, implement a solution that coordinates the activities of the TA and the students. Details for this assignment are provided below.

The Students and the TA

Using Pthreads (Section 4.4.1), begin by creating n(let's say n=5) students where each student will run as a separate thread. The TA will run as a separate thread as well. Student threads will alternate

between programming for a period of time and seeking help from the TA. If the TA is available, they will obtain help. Otherwise, they will either sit in a chair in the hallway or, if no chairs are available, they will resume programming and will seek help at a later time. If a student arrives and notices that the TA is sleeping, the student must notify the TA using a semaphore. When the TA finishes helping a student, the TA must check to see if there are students waiting for help in the hallway. If so, the TA must help each of these students in turn. If no students are present, the TA may return to napping.

Perhaps the best option for simulating students programming—as well as the TA providing help to a student—is to have the appropriate threads sleep for a random period of time.

Coverage of POSIX mutex locks, semaphores and condition variables are provided in Section 7.3. Consult that section for details. You can refer to chapter 6&7 PPT, on page 28(how to use mutex locks), on page 32 (how to use unnamed semaphore). You can refer to sample codes in chapter 6&7 PPT, especially dining philosophers using monitor(posix_dpm1.c).

You are asked to implement it by using a monitor, similar to the dining philosophers solution we provided in PPT.

Command to compile your code:

Suppose your code file name is ta.c, use the following command to build an executable file ta:

```
>gcc -o ta ta.c  -lpthread
```

Or

```
>g++ -o ta ta.c -lpthread
```

Run it:

```
>./ta
```

Please start as early as possible. If you have any problem, please come to our offices to seek help.

Good Luck!

Sample Running Result: (your results don’t need exactly the same)

```
Student 4 hanging out for programming for 4 seconds
Student 3 hanging out for programming for 4 seconds
Student 2 hanging out for programming for 4 seconds
Student 1 hanging out for programming for 4 seconds
Student 0 hanging out for programming for 4 seconds
```

```
       Student 4 takes a seat waiting = 1
       Student 3 takes a seat waiting = 2
       Student 2 takes a seat waiting = 3
TA is serving student(0)
Helping a student for 4 seconds waiting students = 2
Student 4 receiving help
   Student 4 hanging out for programming for 1 seconds
       Student 1 takes a seat waiting = 3
           Student 0 will try later
   Student 0 hanging out for programming for 4 seconds
           Student 4 will try later
   Student 4 hanging out for programming for 2 seconds
           Student 4 will try later
   Student 4 hanging out for programming for 4 seconds
TA is serving student(4)
Helping a student for 1 seconds waiting students = 2
       Student 0 takes a seat waiting = 3
Student 0 receiving help
   Student 0 hanging out for programming for 4 seconds
TA is serving student(0)
Helping a student for 2 seconds waiting students = 2
Student 2 receiving help
   Student 2 hanging out for programming for 3 seconds
       Student 4 takes a seat waiting = 3
TA is serving student(2)
Helping a student for 4 seconds waiting students = 2
Student 1 receiving help
   Student 1 hanging out for programming for 2 seconds
       Student 0 takes a seat waiting = 3
           Student 2 will try later
   Student 2 hanging out for programming for 2 seconds
           Student 1 will try later
   Student 1 hanging out for programming for 5 seconds
           Student 2 will try later
   Student 2 hanging out for programming for 5 seconds
TA is serving student(1)
Helping a student for 3 seconds waiting students = 2
Student 3 receiving help
   Student 3 hanging out for programming for 2 seconds
       Student 3 takes a seat waiting = 3
TA is serving student(3)
Helping a student for 4 seconds waiting students = 2
       Student 1 takes a seat waiting = 3
Student 1 receiving help
```

```
   Student 1 hanging out for programming for 5 seconds
          Student 2 will try later
   Student 2 hanging out for programming for 3 seconds
          Student 2 will try later
   Student 2 hanging out for programming for 4 seconds
TA is serving student(1)
Helping a student for 5 seconds waiting students = 2
Student 0 receiving help
   Student 0 hanging out for programming for 3 seconds
      Student 1 takes a seat waiting = 3
          Student 0 will try later
   Student 0 hanging out for programming for 2 seconds
          Student 2 will try later
   Student 2 hanging out for programming for 4 seconds
TA is serving student(0)
Helping a student for 3 seconds waiting students = 2
Student 3 receiving help
   Student 3 hanging out for programming for 3 seconds
      Student 0 takes a seat waiting = 3
          Student 2 will try later
   Student 2 hanging out for programming for 2 seconds
TA is serving student(3)
Helping a student for 4 seconds waiting students = 2
      Student 3 takes a seat waiting = 3
Student 3 receiving help
   Student 3 hanging out for programming for 2 seconds
          Student 3 will try later
   Student 3 hanging out for programming for 2 seconds
          Student 2 will try later
   Student 2 hanging out for programming for 5 seconds
TA is serving student(3)
Helping a student for 3 seconds waiting students = 2
Student 1 receiving help
   Student 1 hanging out for programming for 2 seconds
      Student 3 takes a seat waiting = 3
          Student 1 will try later
   Student 1 hanging out for programming for 5 seconds
TA is serving student(1)
Helping a student for 4 seconds waiting students = 2
Student 0 receiving help
   Student 0 hanging out for programming for 1 seconds
      Student 2 takes a seat waiting = 3
          Student 0 will try later
   Student 0 hanging out for programming for 2 seconds
```

```
          Student 0 will try later
   Student 0 hanging out for programming for 5 seconds
TA is serving student(0)
Helping a student for 4 seconds waiting students = 2
Student 4 receiving help
   Student 4 hanging out for programming for 4 seconds
      Student 1 takes a seat waiting = 3
          Student 0 will try later
   Student 0 hanging out for programming for 5 seconds
TA is serving student(4)
Helping a student for 5 seconds waiting students = 2
      Student 4 takes a seat waiting = 3
Student 4 receiving help
   Student 4 hanging out for programming for 1 seconds
          Student 4 will try later
   Student 4 hanging out for programming for 2 seconds
          Student 4 will try later
   Student 4 hanging out for programming for 1 seconds
          Student 4 will try later
   Student 4 hanging out for programming for 5 seconds
          Student 0 will try later
   Student 0 hanging out for programming for 3 seconds
TA is serving student(4)
Helping a student for 5 seconds waiting students = 2
Student 2 receiving help
   Student 2 hanging out for programming for 4 seconds
      Student 0 takes a seat waiting = 3
          Student 2 will try later
   Student 2 hanging out for programming for 1 seconds
          Student 4 will try later
   Student 4 hanging out for programming for 3 seconds
TA is serving student(2)
Helping a student for 5 seconds waiting students = 2
Student 1 receiving help
   Student 1 hanging out for programming for 3 seconds
      Student 2 takes a seat waiting = 3
          Student 4 will try later
   Student 4 hanging out for programming for 1 seconds
          Student 1 will try later
   Student 1 hanging out for programming for 4 seconds
          Student 4 will try later
   Student 4 hanging out for programming for 2 seconds
TA is serving student(1)
Helping a student for 4 seconds waiting students = 2
```

```
Student 3 receiving help
   Student 3 hanging out for programming for 2 seconds
      Student 4 takes a seat waiting = 3
         Student 1 will try later
   Student 1 hanging out for programming for 1 seconds
         Student 3 will try later
   Student 3 hanging out for programming for 2 seconds
         Student 1 will try later
   Student 1 hanging out for programming for 3 seconds
TA is serving student(3)
Helping a student for 2 seconds waiting students = 2
Student 0 receiving help
   Student 0 hanging out for programming for 1 seconds
      Student 3 takes a seat waiting = 3
         Student 0 will try later
   Student 0 hanging out for programming for 2 seconds
TA is serving student(0)
Helping a student for 1 seconds waiting students = 2
Student 2 receiving help
   Student 2 hanging out for programming for 5 seconds
      Student 1 takes a seat waiting = 3
TA is serving student(2)
Helping a student for 2 seconds waiting students = 2
Student 4 receiving help
   Student 4 hanging out for programming for 2 seconds
      Student 0 takes a seat waiting = 3
TA is serving student(4)
Helping a student for 1 seconds waiting students = 2
      Student 4 takes a seat waiting = 3
Student 4 receiving help
Student 4 thread is exiting
TA is serving student(4)
Helping a student for 2 seconds waiting students = 2
Student 1 receiving help
Student 1 thread is exiting
      Student 2 takes a seat waiting = 3
TA is serving student(1)
Helping a student for 3 seconds waiting students = 2
Student 0 receiving help
Student 0 thread is exiting
TA is serving student(0)
Helping a student for 2 seconds waiting students = 1
Student 3 receiving help
Student 3 thread is exiting
```

```
TA is serving student(3)
Helping a student for 2 seconds waiting students = 0
Student 2 receiving help
    Student 2 hanging out for programming for 5 seconds
        Student 2 takes a seat waiting = 1
TA is serving student(2)
Helping a student for 3 seconds waiting students = 0
Student 2 receiving help
Student 2 thread is exiting
TA thread is cancelled
```