Programming Exercise 4 – Group Lab Exercise with a Single Lab Room

This week you continue to tackle a synchronization problem directly related to Assignment 1. Again, no suggested solution programs will be given.

Group Lab Exercise with a Single Lab Room: In Assignment 1 there are *K* lab rooms. Each lab room has one tutor who assists students to conduct the exercise in the lab. A lab room can only hold one group of students at a time. Please read Assignment description very carefully.

To simplify the problem, in this programming exercise you are asked to solve Group Lab exercise problem with just a single lab room.

In Programming Exercise 3 you have written a Pthread program to solve a synchronization problem of group *id* assignment to students (which is first part of Assignment 1). You may revise that program by adding lab room assignment to student groups. Now you need to write three thread routines *teacher_routine()*, *tutor_routine()* and *student_routine()* in addition to *main()*.

In the *main()* function, your program needs to ask the user to supply the following parameters:

- *N*: the total number of students in the class;
- *M*: the number of groups;
- *T*: the time limit for each group of students to do the lab exercise.

To assist you to tackle the problem, general logical structures for teacher, tutor and student routines are provided below. You need to properly add necessary global variables, mutexes and conditional variables for correct thread synchronization and communication.

Teacher routine:

```
//group id assignment to student - you have done in Exercise 3

gid = 0;
while (gid < m){
   //wait for lab room to become available
   printf("Teacher: I'm waiting for lab room to become available\n");

   //signal tutor to take group gid for exercise

   //signal students in group gid to enter and start exercise
   printf("Teacher: The lab is now available. Students in group %d can enter
the room and start your lab exercise.\n", gid);

   gid++;
}

//signal tutor to exit
printf("Teacher: There are no students waiting. Tutor, you can go home
now\n");

printf("Teacher: I can now go home.\n");
```

Tutor routine:

```
While(1){
   //wait for teacher to assign a group of students
   printf("Tutor: The lab room is vacated and ready for one group\n");
```

```
   //get group id

   //if signalled by teacher to exit
   printf("Tutor: Thanks Teacher. Bye!\n");
   exit

   //wait for all students in group gid to enter room
   printf("Tutor: All students in group %d have entered the room. You can
start your exercise now.\n", gid);

   //students in group gid conduct the lab exercise

   //signal students the end of exercise
   printf("Tutor: Students in group %d have completed the lab exercise in %d
units of time. You may leave this room now.\n", gid, ex_time);

   //wait for lab to become empty

   //signal teacher the room is vacated
}
```

Student routine:

```
//group id assignment from teacher – you have done in Exercise 3

//wait for teacher to call to enter lab and conduct exercise
printf("Student %d in group %d: My group is called. I will enter the lab
room now.\n", *my_sid, my_gid);

//signal tutor after all students in group are in lab

//wait for tutor to call the end of lab exercise
printf("Student %d in group %d: Thanks Tutor. Bye!\n", *my_sid, my_gid);

//signal tutor the room is vacated
```