

**Operating Systems (Comp Sci 3SH3), Fall 2024**  
**Prof. Neerja Mhaskar**

**Assignment 3**

**Due by 11:59pm on October 27th, 2024**

- **No late assignment accepted, unless an MSAF is provided**
- **If an MSAF is provided, then you will get 5 days extension on the assignment.**
- It is advisable to start your assignment early.
- Make sure to submit a version of your assignment ahead of time to avoid last minute uploading issues.
- Note that students/groups copying each other's solution will get a zero.
- The assignment should be submitted on Avenue under Assessments -> Assignments -> Assignment III -> [Group #] folder.
- In your C programs, you should follow good programming style, which includes providing instructive comments and well-indented code. If this is not followed, marks will be deducted.
- **If working in a group of two, a Readme file containing information on your individual contributions should be provided.**
- **Use of generative AI is not allowed.**

**[10 points] Question 1 — The Sleeping Teaching Assistant.**

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, 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.**

Using `pthread`s, begin by creating `n` 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, 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.

Refer to the practice labs on the use of POSIX mutex locks and semaphores.

**Deliverables:**

1. **q3.c** - You are to provide a C file named q3.c that contains the entire solution for question 3. **Please make sure to name your solution with the correct file name for grading purposes.**
2. Please include the command for compiling and running your code at the top of q3.c file as a comment, alternatively you can upload a Makefile.
3. For this question it is important to have a modular and readable code, with comments.
4. **Marks Split:**  
**3 marks for modular code, comments and readable code.**  
**7 marks for achieving the below task. 7 marks for achieving the below task.**