CSE325-Operating Systems

Project Descriptions

**Project Title:** Seeking Tutor Problem

|  |  |  |
| --- | --- | --- |
| **2018-2-60-010** | **Syeda Tamanna Sheme** | **Seeking tutor problem** |

**Project Description:**

The computer science department of EWU runs a programming club to help undergraduate students with their programming assignments. The club has a coordinator and several tutors to assist the students. The waiting area of the center has several chairs. Initially, all the chairs are empty. The coordinator is waiting for the students to arrive. The tutors are either waiting for the coordinator to notify that there are students waiting or they are busy tutoring. The tutoring area is different from the waiting area. A student, while programming for his project, decides to go to a club to get help from a tutor. After arriving at the club, the student sits in an empty chair in the waiting area and waits to be called for tutoring. If no chairs are available, the student will go back to programming and come back to the club later. Once a student arrives, the coordinator queues the student based on the student’s priority, and then the coordinator notifies an idle tutor. A tutor, once woken up, finds the student with the highest priority and begins tutoring. A tutor after helping a student, waits for the next student. A student after receiving help from a tutor goes back to programming. The priority of a student is based on the number of times the student has visited the club. A student visiting the club for the first time gets the highest priority. In general, a student visiting for the ith time has a priority higher than the priority of the student visiting for the kth time for any k > i. If two students have the same priority, then the student who came first has a higher priority.

Using POSIX threads, mutex locks, and semaphores implement a solution that synchronizes the activities of the coordinator, tutors, and the students. The total number of students, the number of tutors, the number of chairs, and the number of times a student seeks a tutor’s help are passed as command line arguments. Once a student thread takes the required number of help from the tutors, it should terminate. Once all the student threads are terminated, the tutor threads, the coordinator thread, and the main program should be terminated. Your program should work for any number of students, tutors, chairs and help sought. Allocate memory for data structures dynamically based on the input parameter(s).