
Project Documentation: The Sleeping Teaching Assistant

Overview:

The Sleeping Teaching Assistant project is designed to simulate the coordination of activities between a Teaching Assistant (TA) and students in a university computer science department. The TA's primary role is to assist undergraduate students with programming assignments during regular office hours. The project utilizes Java threads, mutex locks, and semaphores to manage the flow of students seeking help and the TA's availability, ensuring a synchronized and efficient process.

Functionality:

1. TA's Office Setup:

- The TA's office accommodates only one desk with a chair and computer.
- There are three chairs in the hallway outside the office for students to wait if the TA is busy.

2. Student Interaction:

- When no students need help during office hours, the TA takes a nap at the desk.
- If a student arrives and finds the TA sleeping, they must awaken the TA to ask for help.
- If the TA is currently helping another student, the new student sits on one of the hallway chairs and waits.
- If no chairs are available, the student will return later.

3. Program Input:

- The program has a GUI. The program takes input parameters such as the number of TAs, the number of chairs available for waiting students.

Implementation:

1. TA Thread:

- The TA thread manages the TA's activities, including sleeping and helping students.

2. Student Threads:

- Each student is represented by a thread that interacts with the TA based on their availability.

3. Semaphore Controls:

- Semaphores are used to limit the number of students that can be in the TA's office simultaneously and manage access to hallway chairs.

4. Mutex Lock:

- A mutex lock ensures exclusive access to the TA's desk, preventing multiple students from interacting with the TA at the same time.
-