

What does this program do?

This program simulates interactions between a Teaching Assistant and students by using multithreading and synchronization. The TA will assist students (the threads) one at a time. Up to three students can be “in line” waiting for assistance. If there are no empty chairs for students to wait, the program will add them in as they are made available. We achieve this simulation by using semaphores and mutexes to manage resources and synchronize all threads.

TA Thread	Function Description
ta_sleep_semaphore	Signals TA Sleep state
ta_ready_semaphore	Signals TA readiness
chairs_count_mutex	Mutex lock for chair count
chair_semaphores[3]	Array of semaphores
TA_Activity()	Handles TA activity, sleep assist, loop
Initialize resources	Initializes semaphores and mutexes

Student Thread	Function Description
student_id	ID of student thread
chair_semaphores[3]	Array of semaphores
chairs_count_mutex	Mutex lock for chair count
ta_ready_semaphore	Semaphore to check TA readiness
Student_Activity(id)	Simulate student requesting help
request_help()	Signals Ta and waits for assistance

Main Thread	Function Description
Students[]	Array of student threads
TA	The TA thread

num_students	Number of students threads
main(argc, argv)	Initialize system and create threads
pthread_create()	Prepares semaphores and mutexes
pthread_join()	Waits for all threads to finish execution