

Report

Q1) An Alternate Course Allocation Portal

- The code takes input as specified in the assignment pdf. After that it launches threads for each of the students, courses and labs into their respective functions: `stu_func`, `course_func`, `lab_func`.
- The `stu_func`:
 - Each thread sleeps for the corresponding time at which the student is scheduled to arrive.
 - Then it enters into a conditional wait which has to be signalled by the course after conducting tutorial for that student. The conditional wait can also be triggered if the course which was the current priority of the student gets removed due to unavailability of TAs.
 - After coming out of conditional wait, if it was due to the tutorial completion the student decides randomly based on the probability = (interest of the course) × (student's calibre) whether or not to choose the course. If he/she decides to choose the course, the student selects it permanently and is removed from simulation. If he/she withdraws from the course, then he moves onto his next preference.
 - If the conditional signal was due to current course being removed, then the student moves on to the next preference. If it was the last preference, then the student exits the simulation. Cases like the next preference also being removed already are appropriately handled.
- The `course_func`:
 - Each thread checks from the list of labs it has been assigned, the eligibility of any TA in a lab. If a TA is eligible, then the TA is assigned to the course. If no TA is eligible from a particular lab, then that lab is removed and a conditional signal is sent to the corresponding lab thread. If no TA is eligible from any of the labs that the course has, then the course is removed and a conditional signal is sent to all the students having that course as their current priority.
 - After getting assigned a TA, a random number is generated with maximum value as the maximum number of slots of the course which is allocated to the course. Then, the course searched for students who have their current preference as that course. **The tutorial is started even with 0 students.**
 - There is a sleep of 5 seconds for conduction of tutorial. After the tutorial is completed, the TA is deallocated and can be considered by other courses. The thread sends a conditional signal to all the students who were allocated in the tutorial.
 - This whole process keeps on repeating with appropriate delays in between.
- The `lab_func`:
 - Each thread in the `lab_func` waits on a conditional variable that signals it about all the TAs exhausting their TA ship.

- After it comes out of conditional wait, the lab thread prints the appropriate statement and returns.

The main program waits on all the student threads to return from the `stu_func` and this finally ends the simulation. At the end of simulation, all the students' final course allotted along with their priorities are also printed as a result of the simulation.