

CS: 431 Programming LanguageLab

Assignment – 1

Report

Devansh Gupta: 170101022

❖ Q1

- The role of concurrency and synchronization in the Socket Matching System:
 - Concurrency: There are many arms(threads) available and they are executing concurrently to pick a sock from the heap and pass it to the socket matcher and also the socket matcher and shelf manager are working simultaneously with the arms(threads).
 - Synchronization: The heap of sockets can be accessed simultaneously by multiple arms(threads) but no two arms(threads) should pick same. So, picking up of a socket has to be synchronized. Matching of the sockets present in the matcher buffer and addition of a new socket in the matcher machine's buffer is done synchronously and also arrangement of the pair of sockets from the shelf manager buffer to the correct shelf and addition of a new sock to the shelf manager's buffer is done synchronously.
- Handling
 - Concurrency is achieved by multithreading.
 - Synchronization for accessing the socks by the multiple arms is done by using **semaphores**. A semaphore controls access to a shared resource through the use of a counter. Here socks are the shared resources and we have used individual semaphores to lock each of the sock and set the semaphore counter value equal to 1 i.e. each sock can be picked up by only one robotic arm. We used block synchronization method to put a sock in the matching machine's buffer and similarly to put a sock in the shelf manager's buffer

❖ Q2

- Importance of Concurrency: TA1,TA2 and CC can simultaneously update the marks of the students. In absence of concurrency, others have to wait for if they want to update the marks of some other student if someone else is editing file. So, concurrency is required for allowing them to concurrently update the marks of different students simultaneously.

- Shared Resources: Files containing Stud_Info are shared resource;
- Absence of Synchronization: If synchronization is not taken care of, some updation of marks may not happen because of independent threads for TA's and CC

Ex:-

Initial Marks: 20

TA1 increasing marks by 10 and simultaneously TA2 increasing marks by 20. If both Queries are executed simultaneously then both TA1 will see the initial marks of to be 20 and increase it by 10. Now TA2 will also see the initial marks of to be 20 as TA1 haven't yet written back the updated marks. And TA2 increments marks by 20. Both TA1 and TA2 write back their updated marks for the student. Finally marks in the records will be 40 (TA2 writes back the updated marks after TA1) , but actually it should be 50. So the Query-1 execution haven't affected the marks.

- Handling:
 - Concurrency is achieved by multithreading.
 - We used **block synchronization** on accessing the student's record, so that at a time only one teacher either of TA1, TA2 or CC can access the student's record.