CS431 Programming Languages Lab

Assignment 1 (Concurrent Programming)

- *Submitted By : Pranshu Srivas (170101048)*

1. **Sock Matching Robot**

   a. The role of concurrency and synchronization in the system**.**
   Concurrency : In our system, there are several robot arms which are working concurrently to pick a sock from the heap and pass it to the matching machine. Along with that matching machine and shelf manager are also working at the same time.
   Synchronization : The heap of socks can be accessed by all the robotic arms simultaneously but no two arms can pick a single sock. Also after picking a sock from, removal of sock from the heap is done synchronously. Sock matching and their pairing is done in a synchronous manner. Arrangement of sock pairs in the shelf is done synchronously.

   b. How do you handle it ?
   Concurrency is achieved by using multithreading. For each robotic arm a thread is created.
   For synchronization in picking a sock from heap, **semaphores** are used. A semaphore controls access to a shared resource using a counter. In our system, socks are the shared resource. We associate a semaphore with each sock in the heap with a counter value of 1 so that only one robot arm can pick it. We use method synchronization for synchronization in matching machine and shelf manager.

2. **Data Modification in Distributed System**

   a. Why is concurrency important here?
   In our system, CC, TA1 and TA2 are independent of each other and they can work simultaneously on the updation of marks. Concurrency will increase the speed of the whole process as if one of them is updating a student’s marks, others don’t need to wait until the completion of his task.

   b. What are the shared resources?
   The file containing the student info are the shared resources here.

c. What may happen if synchronization is not taken care of? Give examples.
If synchronization is not taken care of, then the marks in the file will not get updated in the fashion we want to be.
Suppose a student has marks 50.
TA1 wants to increase it with 10 marks.
TA2 wants to decrease it with 5 marks.
If synchronization is not taken care of then the final marks can be any one of 60 (by TA1) or 45 (by TA2). But here the desired final marks are 55.

d. How did you handle concurrency and synchronization ?
Concurrency : We have multithreading to handle concurrency. Individual threads are created each for CC, TA1 and TA2 which are being run concurrently for the updation of marks.
Synchronization : We have used **block synchronization** on each student for Record level modification. So only one thread (either of CC, TA1 or TA2) can access the same student record and update it.