

Assignment 1 Report

Q1: Sock Matching Robot:

Introduction:

The task is to simulate the sock matching process. The simulation has basically three components i.e. multiple Robotic Arms, Matching Machine, and Shelf manager. To be able to work in coordination, there are also two buffers, one between robotic arms and matching machine (BufferSockMatching) and the other between matching machine and shelf manager (BufferMatchingShelf).

The role of concurrency and synchronization in the above system:

- Concurrency: As given in the problem there can be multiple robots so concurrency is followed when multiple robotic arms pick the socks at the same time. One other way in which concurrency is followed is that the matching machine, shelf manager, and robotic arms are working simultaneously i.e., the matching machine does not wait for robotic arms to finish their work and the same with the shelf manager.
- Synchronization: As all the robots are picking socks from the same heap so synchronization is needed so that no two robots pick the same sock. The buffer between robotic arms and the matching machine is a shared resource so its access has to be synchronized i.e. matching machine will accept the socks when it's ready. Same is the case with buffer present between the matching machine and shelf manager.

How did you handle it?

-
- Concurrency: It is handled by using **multithreading**. Each robotic arm is given a different thread as is the case with matching machine and shelf manager.
 - Synchronization: Synchronization for the heap is handled using **ReentrantLock**. Each sock has a separate lock (SockLock). A robotic arm will pick a sock only if it can acquire its lock. Other shared resources are the two buffers since both the buffers have four elements (corresponding to the four colors) so we have 4 locks for each buffer (BufferSockMatchingLocks[4] and BufferMatchingShelfLocks[4]). Block Synchronization was used along with the locks for the two buffers.

Q2: Data Modification in Distributed System:

Introduction:

We have to simulate the evaluation process conducted by three evaluators (CC, TA1, TA2) with CC having more priority than the two TAs. Since we have to simulate concurrency and synchronization so we take the updates in batch format and then execute them concurrently.

Why Concurrency is important here?

As mentioned in the problem we have to allow TA1, TA2, and CC to update the student records simultaneously so concurrency is important. If concurrency is not present then while one evaluator is updating student A's marks other evaluators will not be allowed to update student B's marks thus decreasing the overall efficiency and latency of the system.

What are the shared resources?

The file "Stud_Info.txt" is a shared resource and also the database in which this data is stored in a shared resource. The database is a hashmap containing information about each student which is used by all the evaluators.

What may happen if synchronization is not taken care of?

If synchronization is not taken care of then some of the updates may not happen properly. For e.g. We have a student with roll no 174101035 whose current marks are 80. Suppose we have two queries at the same time:

1. TA1 wants to update this student's marks by +5.
2. TA2 wants to update this student's marks by -5.

If synchronization is not taken care of then both the TAs will see 80 as current marks and update them accordingly so the student might get 85 or 75 as final marks depending on which TA was last to update.

If synchronization is taken care of then whichever TA first gets access to this student's data will make the update then the other TA will update the marks thus final marks will be 80.

How you handled concurrency and synchronization?

- Concurrency: It is achieved by using **multithreading**, each evaluator is given a different thread i.e. total of 3 threads so they will be able to perform updates simultaneously. Concurrency happens in file-level modifications i.e. when different evaluators want to update different student's marks.
- Synchronization: It is handled using a **synchronized block** for each student's record in the database. Synchronized is useful in record-level modification i.e. when more than 1 evaluators want to update the same student's marks.