CS 431
Programming Languages Lab
Assignment 1

Anubhav Tyagi

170101009

October 8, 2020

Q1

a

Concurrency is the ability of different parts or units of a program, algorithm, or problem to be executed out-of-order or in partial order, without affecting the final outcome. This allows for parallel execution of the concurrent units, which can significantly improve overall speed of the execution in multi-processor and multi-core systems. In more technical terms, concurrency refers to the decomposability property of a program, algorithm, or problem into order-independent or partially-ordered components or units.
In the given problem, multiple instances of Robotic Arms working together to pick up socks from the newly labelled heap of socks is an example of concurrency.A single task of picking up sock from heap of pile is divided among the multiple instances of robotic arms to speed up the process i.e. to increase throughput.

Synchronization refers to one of two distinct but related concepts: synchronization of processes, and synchronization of data. *Process synchronization* refers to the idea that multiple processes are to join up or handshake at a certain point, in order to reach an agreement or commit to a certain sequence of action. *Data synchronization* refers to the idea of keeping multiple copies of a dataset in coherence with one another, or to maintain data integrity. Synchronization is required when multiple processes are dependent on a resource and they need to access it at the same time the operating system needs to ensure that only one processor accesses it at a given point in time.
In the given problem, synchronization is required when multiple Robotic Arms try to pass socks to a single Mactching Machine. In this case, multiple robots try to access a single resource (Matching Machine) at the same time thus giving rise to a need for synchronization

b

Concurrency was handled with the help of **Java Threads Library**. A thread of execution is the smallest sequence of programmed instructions that can be managed independently by a scheduler. Each thread is scheduled on a single core.In case of multicore systems, multiple threads can be run simultaneously.Java provides **Runnable** class for implementing threads.Each Robotic Arm runs as a separate thread, thus providing concurrency in the

sock picking task. Moreover,the socks were divided equally among the Robotic Arms so that each robot did an equal amount of work and overall work was distributed among the robots

Synchronization was handled with the help of **synchronization block** provided by Java.A synchronized block in Java is synchronized on some object. All synchronized blocks synchronized on the same object can only have one thread executing inside them at a time. All other threads attempting to enter the synchronized block are blocked until the thread inside the synchronized block exits the block.
In the given problem, synchronizaton block was used to maintain the count of socks of each color with Matching Machine.Since multiple robotic arms may try to pass socks to the Matching Machine at the same time,this may result in incorrect count as they may try to simultaneously change the socks count variable.