CS431: Programming Languages Laboratory

Assignment 1: Concurrent Programming

Aneesh Dash	130101006
Anirudh Agnihotry	130101007

Q1. Write a short note on the benefit of having concurrency in modeling the problem.

Ans: We have 10 sensors producing some data and we have to process the data to generate some outputs. Concurrency will help in this problem as the problem can be broken into multiple independent tasks. Thus, the tasks can be assigned to different threads which can execute in parallel resulting in speedup of the process.

- 10 sensors can be run are separate threads to produce data irrespective of other sensors
- Post processing of data can be run as different threads.
- Each task need not wait for other tasks to execute and can continue its work given the required data is available.

Q2. Propose a concurrent solution for the problem.

Ans. We propose the following approach:

1. Create 10 sensor threads which create 8-bit binary strings and add them to their respective queues at certain time intervals.
2. A main thread exists which, at certain time intervals, reads the top values from the queues and converts the strings to integers using separate thread for each queue.
3. If some sensor's queue is empty then data is not processed further in that iteration.
4. Two threads calculate the sum and product of the converted integer values.
5. The main thread then calculates average and compares with threshold.

From java.util.concurrent, locks were used to ensure that no other thread inserts or reads while one thread is accessing the data and executor service was used to execute threads.
The same was implemented using Thread class and synchronized keyword without using the concurrent library.

Implementation of 2nd part (Merge Sort using Fork-Join):
Merge sort was implemented using ForkJoinPool class which creates a pool of threads and uses them for tasks. In merge sort, a new thread was created, using java.util.concurrent.RecursiveAction, for the left split of the array and right part was continued in the same thread. Merging was done in the original thread.

Implementation of 3rd part (Calculator):
javax.swing library was used to create the GUI for the calculator. Main window was created with JFrame and JLabel and JButton was used for building the elements of the calculator.
KeyEvent, KeyListner, ActionEvent and ActionListner from java.awt.event library was used for event handling which listened on the frame and carries out actions based on input from keyboard.
Separate threads were used for highlighting the numbers and operators.