COEN 383 ADVANCED OPERATING SYSTEMS

Multi-threaded Ticket Sellers

BY:

GROUP 6

Dhaval Pujara - W1459995
Aditi Parikh - W1550758
Aayushi Mehta - W1549753
Laxmi Samal - W1505865
Shivangi Vyas - W1536926

Design

This program is written in C using pthread library to create threads and mutex. Also, we have used condition functions to wait and broadcast.

Data Structure:

We have created six structures as a part of data structures to store different information such as :

- **Customer**: To store the customer information such as Customer ID and Arrival Time.
- **Customer Queue**: To maintain a queue for customer served by different type of seller.
- **Row**: To store information about the next available seat and mutex.
- **Pthread_args**: To combine the information about the seller and customer served by it.
- **Seat_state**: To maintain the status of each seat.
- **Seat_manager**: Allocates the seats to all seller according to the seller type.

We initial the theater structure to Null and seat_manager initializes the starting seat numbers to each seller based on the seller type. Creating a customer queue that stores the customer id and arrival time which is generated randomly, and we sort it based on the arrival time and initial all the mutex value to Null. Every seller is given row ids and customers are assigned to each customer queue and then processed based upon the matching row id queue and customer queue. The seller will server one clock time and processor each customer. The seller of type M will maintain 3 counter which contains the number of customers served by each seller M1, M2 and M3. Also, the seller of the type each seller L1, L2, L3, L4, L5, and L6. A customer from the customer queue depending on the value of Seller type will be is assigned to the seller L1 to L6 based on the availability and will be processed. Once the seller has processed the ticket the seat status will change L will maintain 6 counter which contains the number of customers served by to 1(sold). All the customer will be served in a similar manner until the seat is filled out or the clock cycle hits. For any of the remaining customers that couldn’t be served will be turned down.

Customer queues for all sellers, counters for each seller queues, the total number of customers, pthread arguments and seat specific structure variables are shared among all the functions.

Critical Region : We have applied lock inside three functions get_H_seat_to_sell(), get_M_seat_to_sell() and get_L_seat_to_sell() where the available seat is fetched and assigned to the customer so that no single seat can be sold out to more than one customer.

Synchronization : When a customer is processed by the seller, the seller locks the seat and will only be released the lock after the seat has been allocated to the customer successfully. This whole process will make sure that synchronization has been maintained throughout the execution.