

CSEN 383 – Assignment 3

Winter 2024

Group 2

Chaitra Boggaram (W1651213)
Divyanth Chalicham (07700005995)
Shreya Chinthala (07700005606)
Ruchi Manikrao Dhole (W1652116)
Rohit Parthiban (07700006502)

Objective

This project provides hands-on experience in developing a multithreaded program using the pthread library. We've crafted a C program utilizing pthread to create threads and mutex, simulating the concurrent activity of ticket sellers selling concert tickets simultaneously over the course of one hour.

Theory

At the program's initiation, 10 customer queues are established, each accommodating N customers. These queues store customer details, including randomly generated customer numbers, arrival times, and service times. Arrival and service times are quantized in minutes, with all customers arriving at the commencement of each minute. Subsequently, each of the ten sellers is assigned a queue and begins processing customers upon their scheduled arrival. Additionally, each seller is designated a type (L, M and H) dictating their processing speed and seat selection strategy. As a seller initiates service for a customer, it seeks an available seat based on its type. Open seats are marked by an unlocked mutex, ensuring that only one seller can claim a specific seat. Once a seller secures a seat mutex, it records details at that seat location, including response time, turnaround time, and seller type. This process continues until all seats are occupied or an hour elapses. Any remaining unserved customers are turned away. At the simulation's conclusion, sold seats are analyzed to calculate the average response time, turnaround time, and throughput for each seller type.

The aggregate count of customers per seller, the global clock counter, pthread arguments, and seat-specific structure variables are common across all functions. Leveraging the provided code snippet as a foundation, we built our task by simulating clock ticks using the main thread. This approach enables us to manage critical regions and execute the selling process in child threads dedicated to simulating ticket sales. In our project, the main thread generates clock ticks at regular intervals, simulating a one-minute time duration for each tick.

Presumptions

1. Clock Tick: The smallest measurable time unit is one minute, and each child thread is designed to simulate tasks that take exactly one minute. This includes serving customers, waiting for their completion, or finalizing a sale.

2. State of Seller's Thread: Each seller's thread is assumed to be in one of the following states during any given time unit:

- ◆ Waiting: Waiting for a new client
- ◆ Serving: Actively serving a new client from the seller's queue
- ◆ Processing: Engaged in processing and dedicating time to complete the sale
- ◆ Completing: Finalizing the sale for the client

3. New Clock Generation: A new clock is generated when all seller threads have completed their tasks for that time unit to maintain time synchronization.

4. Concert Seat Simulation: Concert seats are represented by a 2-dimensional matrix. To avoid contention in seat assignment, it is assumed that only one thread manipulates the matrix. The overall project workflow is as follows:

- ◆ Initialization: Initial parameters, including mutex locks, seat matrix display, customer queues for each seller, and threads, are initialized.
- ◆ After creating seller threads, all threads enter a waiting state for the next clock tick after completing their initialization.
- ◆ Simulating Clock Tick:
 - When the main thread triggers a clock tick, all seller threads race to acquire a lock on the concert seat matrix based on their current state.
 - Initially, each seller thread checks for new customers based on arrival time and proceeds to serve them one by one.
- ◆ Termination Condition: Each seller thread terminates its processing if either the concert is sold out or the simulation time has expired. In both cases, the customer is required to leave.

Conclusion

Calculated statistics post-program execution

Seller	Number of Customers	Sold Seats	Returned	Throughput	Avg Response Time	Avg Turnaround time
H	10	10	0	0.17	0.700000	34.70
M	30	29	1	0.48	1.400000	30.73
L	60	45	15	0.75	5.300000	25.30