

CSEN 383

Project 3 – Multithreaded Ticket Sellers

Group 6:

Madhuri Sharma (07700006469)

Vaibhav Somkuwar (7700009607)

Yash Suryavanshi (7700005725)

Nachiket Timmanagoudar (7700012309)

Archana Vellanki (7700009833)

OVERVIEW:

We developed a C program that simulates ticket sellers synchronously selling concert tickets to customers over the course of one hour. Using the pthread library to create threads and mutexes that monitor critical sections, we adhered to the goal of maintaining concurrency, and tracked statistics such as response time, turnaround time and throughput.

KEY ASSUMPTIONS:

- **Clock ticks:** The minimum unit of time is 1 minute. Each child thread simulates tasks in discrete units of time, which include serving a customer, finishing up a sale, or waiting for completion.
- **Synchronizing time:** Each completed transaction begins a new interval of clock time, ensuring that time is synchronized across different sales.
- **States of seller threads:** At each clock tick, the seller thread will be in one of the following states:
 - **Waiting state:** Where it waits for the arrival of a new customer.
 - **Service state:** Where it serves a new customer (ie: the customer at the head of the seller's queue).
 - **Processing state:** Where it engages in the necessary processing to finish the sale.
 - **Completion state:** Where it finishes the sale for the customer.
- **Simulating concert seats:** The seating chart for the concert venue is represented by a 2-D matrix. To avoid any clashes in seat allocation, only one thread accesses and modifies the matrix (via mutex locks). There are 10 rows of seats overall, and each row has 10 seats.
- **Seller types:** There are 3 types of sellers, each with their own customer queue:
 - **High-priced ticket sellers (H):** Serves customers the fastest, completing a sale in exactly 1 or 2 minutes. There is only 1 such seller at the venue.
 - **Medium-priced ticket sellers (M):** Completes a sale in exactly 2, 3, or 4 minutes. There are 3 such sellers at the venue (denoted by M₁, M₂, and M₃).

- **Low-priced ticket sellers (L):** Serves customers the slowest, completing a sale in exactly 4, 5, 6 or 7 minutes. There are 6 such sellers at the venue (denoted by L1, L2, L3, L4, L5 and L6).

IMPLEMENTATION:

- **Initialization:** Here, we set up the seating chart, customer queues for each seller, threads and mutex locks. We also initialize the seat assignment order for each seller type:
 - H assigns seats from front to back, ie: from the first row to the last.
 - L1, L2, L3, L4, L5 and L6 assign seats from the back to the front, ie: from the last row to the first.
 - M1, M2 and M3 assign seats according to a custom order: 5,6,4,7,3,8,2,9,1,10.
 Each customer queue accommodates N customers, where N is provided as a command-line argument.
- **Waiting phase for seller threads:** Once the seller threads are created, they enter the waiting state, anticipating the next clock tick.
- **Active phase for seller threads:** Once the main thread drives a clock tick, all seller threads aim to obtain a lock on the seating chart according to their current state. **Since this seating chart was shared across all seller threads, a critical region for the code involved the threads attempting to access the seating chart to check for an available seat.** This is controlled by using a mutex for tracking open seats, ensuring that no two sellers access the same seat at the same time while serving their respective customers. Each seller thread checks its respective queue for any newly-arriving customers (while keeping in mind the arrival time) and serves them one after another. As a seller thread serves a customer, it searches for any available seats based on its type: an unlocked mutex indicates that there is an open seat. Once the seller thread obtains that mutex, it starts its processing to complete the sale, and records details like turnaround time and response time for each seller type.
- **Termination phase for seller threads:** Each seller thread stops processing if the concert is sold out or the simulation time has ended. In either scenario, customers must exit the queues.
- **Calculating statistics:** For each type of seller, the average turnaround time, average response time and throughput is calculated. We also calculate the number of customers served as well as the number of customers turned away for each seller type.

RESULTS:

The results are stored in files results-5.txt, results-10.txt and result-15.txt. They are summarized as follows:

# of customers per queue	Seller type	# of customers served	# of customers turned away	Avg. turnaround time	Avg. response time	Avg. throughput
5	H	5	0	1.40	0	0.07
	M	15	0	2.87	0	0.21
	L	30	0	6.97	1.47	0.43
10	H	10	0	2.00	0.50	0.14
	M	30	0	4.50	1.47	0.43

	L	53	7	9.55	3.94	0.76
15	H	11	4	1.91	0.18	0.16
	M	38	7	5.76	2.74	0.54
	L	51	39	15.08	9.71	0.73