

# **CSEN 383: Advanced Operating Systems**

## **Group 3 Project 3 Report**

Members: Lydia Myla, Joshua Kindarara, Praful Mamidi, Samantha Lee, and Guru Sushma Kasa

**Objective:** Write a multithreaded program that simulates ticket sellers simultaneously selling concert tickets during one hour.

**Software Design:** Our software design is a multi-threaded time-simulation. It has a main function that acts as a global clock and ten threads, each representing a ticket seller. The simulation occurs in one-minute intervals, where the main function synchronizes the ten threads using a condition variable. At the beginning of each minute, the function sends a signal to wake up all sellers so that they can sell their tickets for that minute. Each thread waits for the signal and checks its queue for customers whose arrival time matches the current simulation time. It serves the first customer and decreases the remaining service time. It gets a lock on the seats matrix, finds and reserves a seat, updates the matrix, and finally releases the lock.

- **Parameter Adjustments:** A parameter that was changed was N, the number of customers per seller. We ran simulations for N=5, N=10, and N=15. We had other parameters that we didn't need to adjust, since they remained the same for each simulation. These key parameters were total simulation time (fixed at 60 minutes), number of sellers (1 for high sellers, 3 for medium sellers, and 6 for low sellers), and service time (implemented as a random value within a specific range depending on the seller type).
- **Data Shared & Critical Regions:** The seats matrix was shared among the threads. This matrix represented the concert seating chart. All of the threads needed to read it to find empty seats and write to it to reserve them. The current time slice, which represented the simulation clock, was also shared and read by the threads to let them know if a customer arrived. Another shared data was the global statistical counters (response time and turn around time). These counters were updated by the threads. A critical region was when the threads needed to find an available seat to reserve for the customer. This region required threads to read the seats matrix and then write the customer ID to it. Another critical region was the global statistical counters.
- **Process Synchronization:** One process synchronization was the mutex used for the seats matrix. This lock guaranteed mutual exclusion so that only one thread could access the seating chart at any point in time. This ensured that no two sellers could assign the same seat to two different customers. Another process synchronization was the condition variable used to synchronize the simulation's clock. All of the threads waited on the condition variable until the main function signaled to them to continue. This process synchronization made sure that all sellers moved on to the next minute together at the same time.

**Output:** Below we attached individual outputs for each N value.

- N = 5:

```
Final Concert Seat Chart
*****

H101 H102 H103 H104 H105 - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -

M201 M202 M301 M203 M204 M302 M101 M205 M102 M103
M104 M303 M304 M105 M305 - - - - -
- - L605 L105 L405 L305 L104 L203 L304 L604
L404 L505 L303 L202 L403 L603 L504 L302 L402 L201
L401 L503 L103 L102 L502 L602 L301 L601 L501 L101

*****
Multi-threaded Ticket Sellers
Input N = 05
*****

-----
| No of Customers | GotSeat | TurnAway | Throughput |
-----
| H |          5 |      5 |      0 | 0.08 |
| M |         15 |     15 |      0 | 0.25 |
| L |         30 |     28 |      2 | 0.47 |
-----

-----
| Avg Response Time | Avg Turnaround Time |
-----
| H | 0.000000 | 32.60 |
| M | 0.666667 | 29.13 |
| L | 0.533333 | 30.47 |
-----
```

- N = 10:

```
Final Concert Seat Chart
*****

H101 H102 H103 H104 H105 H106 H107 H108 H109 H110
- - - - - - - - - -
M209 - - - - - - - - L307 L206
M309 M208 M109 M110 M310 L508 L607 L108 L408 L306
M307 M207 M308 L205 L507 L606 L407 L107 L305 L506
M101 M102 M201 M301 M202 M103 M104 M203 M105 M302
M204 M303 M106 M304 M107 M205 M305 M306 M108 M206
L204 L406 L605 L106 L505 L203 L105 L405 L304 L604
L104 L404 L202 L504 L303 L603 L503 L103 L602 L302
L403 L102 L502 L301 L402 L201 L101 L601 L501 L401

*****
Multi-threaded Ticket Sellers
Input N = 10
*****

-----
| No of Customers | GotSeat | TurnAway | Throughput |
-----
| H |          10 |     10 |      0 | 0.17 |
| M |          30 |     29 |      1 | 0.48 |
| L |          60 |     44 |     16 | 0.73 |
-----

-----
| Avg Response Time | Avg Turnaround Time |
-----
| H | 0.700000 | 34.40 |
| M | 1.333333 | 30.63 |
| L | 5.316667 | 24.50 |
-----
```

- N = 15:

```
Final Concert Seat Chart
*****

H101  H102  H103  H104  H105  H106  H107  H108  H109  H110
H111  H112  H113  H114  H115  M312  M212  L109  L309  L408
M110  M311  M111  M211  M112  L607  L208  L508  L108  L308
M309  M209  M109  M310  M210  L606  L507  L207  L407  L605
M107  M208  M108  M308  L107  L307  L406  L506  L206  L604
M101  M301  M201  M302  M102  M202  M303  M103  M104  M203
M105  M204  M106  M304  M205  M305  M206  M306  M207  M307
L306  L405  L106  L205  L505  L305  L603  L404  L204  L304
L105  L504  L602  L403  L203  L303  L104  L601  L503  L103
L402  L302  L202  L102  L502  L401  L301  L201  L101  L501

*****
Multi-threaded Ticket Sellers
Input N = 15
*****

-----
|   | No of Customers | GotSeat | TurnAway | Throughput |
-----
| H |           15 |      15 |         0 | 0.25 |
| M |           45 |      36 |         9 | 0.60 |
| L |           90 |      49 |        41 | 0.82 |
-----

-----
|   | Avg Response Time | Avg Turnaround Time |
-----
| H | 0.800000 | 26.80 |
| M | 4.422222 | 26.76 |
| L | 6.788889 | 18.14 |
-----
```