COEN 383: Advance Operating Systems

Project 3 **Multi-threaded Ticket Sellers**
Fall 2023 Mon-Wed 7:10 am Group 8

Team members:
Akaash Meghraj Trivedi (W1650955)
Anagha Viswanath (W1651360)
Michael Zarrabi (W1479183)
Nakul Pravin Thombare (W1650130)
Sai Sanjana Reddy Vatte(W1650167)

Objective:

The primary objective of this project is to delve into the practical implementation of multithreading concepts. By designing and coding a program, we aim to create a simulation that mirrors the concurrent operation of multiple ticket sellers, providing a tangible application for the principles of multithreading.

Observations:

Main Thread Responsibilities:

The main thread orchestrates the simulation, initializing data structures, launching seller threads, collecting audience statistics, and deallocating resources. It effectively coordinates the entire simulation.

Global Clock for Synchronization:

Synchronization among seller threads is achieved using a global clock variable. Mutexes ensure exclusive access to the global clock, preventing concurrent modifications and maintaining synchronization.

Quanta-Based Increment of Global Clock:

Incrementing the global clock after all sellers finish their tasks for a specific quanta ensures that the threads progress together. The quanta concept provides a systematic way to advance the simulation.

Separate Thread for Clock Updates:

Clock updates are performed in a separate thread, minimizing synchronization needs. This thread is responsible for maintaining overall time consistency in the simulation.

Local Clocks for Seller Threads:

Each ticket seller thread has a local clock, synchronized with the global clock. Sellers wait for commands from the main thread, and at the end of each quanta, they update and compare their clocks to progress.

Mutexes for Shared Data Access:

Mutexes protect shared data, including seat reservations. The array representing seats is initialized as empty to prevent double-selling, and mutexes control access to this crucial shared resource.

Seat Reservation Logic:

Seat reservations are processed through function calls, and each seller simulates locating available seats as customers request them. The logic ensures a controlled and conflict-free assignment of seats.

Customer Queue Handling:

Customer queues are established at the simulation's outset, but sellers only interact with customers who have already arrived. This approach simplifies the simulation and aligns with real-world ticket-selling scenarios.

Critical Zones with Mutexes:

Critical zones are defined by mutex-protected global variables tracking the most recent seats sold by each seller type. This ensures consistency among sellers of the same kind and prevents conflicts in updates.

Assumptions:

Seller Thread States: Sellers are expected to be in states of Waiting, Serving, Processing, or Completing during any given time interval, reflecting the sequential nature of their tasks.

Clock Tick and Time Interval: The smallest measurable time interval is one minute, with each thread simulating one minute of work. This granularity simplifies timekeeping and task simulation.

Time Synchronization: A new clock is generated to maintain time synchronization among threads, facilitating a cohesive and realistic simulation.

Matrix Representation for Seats: A 2-dimensional matrix represents concert seats, and precautions are taken to ensure only one thread modifies the matrix to avoid seat assignment conflicts. This structure supports the integrity of seat allocation.

Critical Region:

Critical regions, where only one operation can occur at a time, are implemented in three functions: get_H_seat_to_sell(), get_M_seat_to_sell(), and get_L_seat_to_sell(). These functions fetch and assign available seats to customers, ensuring that no seat is sold to more than one customer simultaneously.

The following process synchronization is required:
Synchronization is maintained by locking seats when a customer is being processed. The seat lock is released only after the seat is successfully allocated to the customer. This ensures that synchronization is upheld throughout the program execution.

Output:

Concert seating chart and average statistics:

N = 5

```
Final Seat Allocation
=======================
| H101  H102  H103  H104  H105     -     -     -     -     -
|   | -     -     -     -     -     -     -     -     -     -
|   | -     -     -     -     -     -     -     -     -     -
|   | -     -     -     -     -     -     -     -     -     -
|   | -     -     -     -     -     -     -     -     -     -
| M101  M201  M202  M301  M102  M302  M303  M203  M304  M305
| M204  M205  M103  M104  M105     -     -     -     -     -
|   | -     -     -     -     -     -     -     -  L304  L105
| L603  L504  L303  L503  L104  L502  L404  L103  L302  L403
| L602  L301  L102  L402  L202  L201  L101  L501  L401  L601


Stat for N = 05
===============
| ==========================================
|    | No of Customers | Got Seat | Returned |
| ==========================================
|  H |               5 |        5 |        0 |
|  M |              15 |       15 |        0 |
|  L |              30 |       22 |        8 |
| ==========================================

Average TAT is 33.40
Average RT is 36.60
Throughput of seller H is 0.37
Throughput of seller M is 0.25
Throughput of seller L is 0.08
```

N = 10

```
Final Seat Allocation
=======================
| H101  H102  H103  H104  H105  H106  H107  H108  H109  H110
|   | -     -     -     -     -     -     -     -     -     -
| M208    -     -     -     -     -     -     -     -  L106
| M308  M309  L209  L506  L408  L608  L308  L208  L105  L607
| M206  M110  M207  M307  L407  L505  L307  L406  L207  L104
| M101  M301  M102  M201  M202  M103  M302  M104  M203  M303
| M204  M105  M304  M106  M305  M306  M107  M205  M108  M109
| L606  L306  L504  L206  L605  L405  L205  L305  L503  L103
| L404  L604  L204  L304  L403  L603  L203  L303  L602  L502
| L402  L102  L202  L302  L401  L501  L301  L201  L101  L601


Stat for N = 10
===============
| ========================================
|    | No of Customers | Got Seat | Returned |
| ========================================
|  H |              10 |       10 |        0 |
|  M |              30 |       27 |        3 |
|  L |              60 |       45 |       15 |
| ========================================

Average TAT is 22.30
Average RT is 25.70
Throughput of seller H is 0.75
Throughput of seller M is 0.45
Throughput of seller L is 0.17
```

N = 15

```
Final Seat Allocation
=======================
| H101    H102    H103    H104    H105    H106    H107    H108    H109    H110
| H111    H112    H113    H114    H115    M112    M212    L409    L607    L508
| M110    M312    M211    M111    M313    L108    L208    L308    L408    L507
| M311    M209    M109    M210    L207    L307    L606    L107    L407    L506
| M207    M107    M309    M208    M310    M108    L206    L306    L605    L406
| M301    M302    M303    M201    M304    M101    M202    M102    M305    M203
| M103    M306    M204    M104    M307    M205    M105    M206    M308    M106
| L106    L505    L205    L305    L604    L405    L105    L504    L204    L603
| L304    L404    L104    L203    L503    L602    L103    L303    L403    L601
| L202    L102    L502    L302    L402    L501    L201    L101    L301    L401


Stat for N = 15
===============
| ========================================
|    | No of Customers | Got Seat | Returned |
| ========================================
|  H |              15 |       15 |        0 |
|  M |              45 |       37 |        8 |
|  L |              90 |       48 |       42 |
| ========================================

Average TAT is 34.13
Average RT is 38.73
Throughput of seller H is 0.80
Throughput of seller M is 0.62
Throughput of seller L is 0.25
```

Conclusion:

This assignment provided a comprehensive exploration of multithreading in a practical context. The simulation of ticket-selling dynamics not only reinforced theoretical concepts but also required thoughtful design decisions to create a coherent and realistic multithreaded program.