Alan Padilla Chua

Axp141330

Project 2 Summary

The purpose of this project was to understand the use of semaphores in multithreaded programs to give a perspective in how a processor must share resources with threads but must also be fair and also provide mutual exclusion in critical parts of the program when needed. In this project we had to create 50 threads with 49 of those being a Person thread that must all share the reference to an Elevator. The key detail to keep in mind for this project was that only 7 threads at a time could be inside the elevator.

Using java as my language I wanted to make this project object oriented. Because I only wanted one instance of Elevator to exit and all person threads to have a reference to that one singular instance of elevator. I made elevator an inner static class of Person. Another thing is that I have never worked with multithreaded programming so my main challenge was understanding how the threads would work and how java semaphores are applied to specific threads. Debugging was also interesting because multithreaded debugging requires special tools provided by the ide that I was using.

My results showed that semaphores require coordination between threads and classes so that an infinite loop is avoided or the program is not stalled during a particular thread. The elevator thread is the one that kicks off the programming by signaling 7 person threads to enter the door and enter the elevator. Then each thread individually enters the elevator with the use of semaphores to guarantee mutual exclusion This is to avoid any errors because the threads would be writing to the memory that holds the array list. Mutual exclusion is used again when the threads leave the array list because they are deleting parts of it and it could cause errors. An array of semaphores is used to signal to individual threads that they should exit the elevator. The threads are given and id when the person constructor is called so that each thread is unique from the others.

In general, this project was really fun and a good introduction to semaphores for multithreaded programming. I learned a lot and hope to apply it.