John Lawler

CS4348.501

Professor Greg Ozbirn

October 28th, 2023

Project Two Summary

During the implementation of the Project Two simulation, one of the largest problems I struggled with was ensuring mutual exclusion among the threads. The program is heavily reliant on the utilization of multiple threads accessing shared data (as well as modifying it so that the frontdesk could handle new customers) where I actually had a problem with race conditions during development. This was by far the longest, most annoying bug I've had to deal with in the past year as I needed to re-update the value of shared_customer with customer in Functions.cpp on line 65. This was necessary for the correct customer to request a bellhop, and without this line, the program would start hanging up after the second bellhop was considered busy holding bags for a customer that could never get to their room.

I should have already considered this problem, as before I started coding, I made a diagram to help visualize the flow of the program. I was having difficulty thinking about the three unique kinds of threads and how they should interact with each other, so I made a diagram that would help me determine how I would structure the semaphores and the variables I would be storing. I decided to use an array of a struct of integers that would allow me save the number of bags each customer has, the room they are assigned, the bellhop they are assigned, and the front desk they went to. This allowed me to ensure mutual exclusion with a global variable that accounted for the current customer being attended to by an individual thread. Semaphores would allow me to have each thread wait until they were done with the shared_customer value before moving on to their next scheduled task.

This project had me learning a lot about system operations in a way that few other classes had me thinking. CS 3377 Unix had me using pthreads, but not to the extent that I needed to know for this class. There was a rather large learning curve for me with this project as I attempted to complete the first project, but I was unable to actually turn anything in as I did not give myself enough time to complete it. This had me in a position where I needed to put in a lot of time to become familiar with C/C++ again as I have been primarily using JavaScript and Python this semester. This is on top of the fact that I have fairly little experience with things like creating threads, optimization, and concepts such as mutual exclusion. By the end of this project, I feel that I have greatly improved in these skills. I am able to use pthreads and semaphores in a way that uses mutual exclusion and works on information and data in an efficient manner.