Brandon Angeles

CS 4348.002

Dr. Greg Ozbirn

30 April 2022

**Project 2 Summary**

Let me begin with saying that this project was difficult for one main reason, and that was time. When the project was assigned I had three exams I had to study for all split up between two weeks along with assignments and my job so finding time for this project was definitely difficult. My simulation does its best to keep concurrency at a high and with that there is a use of a decent number of “global” variables, meaning all classes can use them, in the main cases to allow for this to work. 50 customers were made with own threads and their own semaphores to help make sure that they can be used individually by the theater workers. Every theater worker was given their own threads and a form of semaphore to make sure the customers could only work with those number of threads. The only mutual exclusion necessary in the whole simulation was the setting and getting of the queues of each set of workers and for when we have to check ticket counts of movies. This was actually the biggest problem I faced because at the beginning of the simulation my only goal was to make the code work. I decided to only use mutexes to make it all work at first meaning only one customer could work with another thread at a time. I talked to Dr. Ozbirn about this problem, and he reminded me of the barbershop problem and how we used queues to fix that issue in that example. I had already looked back at this example for the solution of assigning each customer a semaphore, so it made sense to look back again for the queue usage. The main document or program I looked at during this project was that example. Since I fully understood how the barbershop example worked and how every time we improved it would increase concurrency and fix any other problems it may have had, this made the project quite simple to implement. Other than semaphores and concurrency for this project some other problems I had was the way to read in the file which required “isDigit()” to help separate the movie titles and ticket counts. Lastly I am very glad that Dr. Ozbirn mentioned the “setDaemon()” function as that was a solution to the issue I was having for my code to finish running. The most monotonous part of this project was the testing, as I would have to run the project enough time to wait for an error to occur or if it worked check that every customer was attended to and not mixed up between workers in any way. I truly understand semaphores and mutual exclusion from this project, and I think the result for this project will show that.