Team Meatball  
Andres Chorro-Dahlgren  
Dora Do  
Jannette Pham-Le  
CS 149 – 02  
28 February 2015

Assignment #3: Report

*What threads did you create? What data was shared and what were the critical regions? What thread synchronization did you use?*

Our design was based on having a thread for each student attempting to enroll in a section, and a thread for each queue that processes the students of a certain priority. We defined a struct that represented a student. It held the student’s ID, section preference, priority type, when the student started processing, when the student finished processing, and their turnaround time.

Three of our arrays functioned as queues for each of the priorities. These were shared by student threads joining the queues and queue threads processing them. We handled this with a semaphore for each queue: students post to the semaphore when they join a queue and the process waits on that semaphore. Another shared piece of data was the final section list which is a two-dimensional array holding three sections of students that have successfully enrolled. Each of the three queue threads are inserting into these lists concurrently, so we protect each one with it’s own mutex. We maintained several other lists to help calculate the statistics. These were shared as well, but did not require locks since they never required any elements to be deleted.

The critical region for the student threads is when they arrived and are being inserted into a queue. The queue threads have two critical regions: popping from a queue and inserting into a section list.