Daniel Gaston

CSC 33200

Lab 6 - Pthreads Vs Semaphores

So for lab 6, we have to control synchronization between one agent process and three smoker processes. Using semaphores, I created a semaphore for each process and a semaphore for the lock that is to be used to lock the processes. In the agent process, I request for the lock and choose the materials to be placed on the table. Then I return the smoker that has the third material so he can create the cigarette and smoke. I say that creating this process is a bit longer as there were more variables to utilize the operations P() and V() and be careful into running into errors.

When using pthreads, I had to create a function for the smoker processes and the agent process. The I create the threads in the main function and join them together. First the agent process gets lock by the mutex then place the materials then unlock the mutex for the appropriate to come through. It was easier to setup than the semaphore due to having less resources.

It was easier doing in finite time than semaphores due to only changing the while(1) to a for loop. For pthreads, if you aren’t careful ,there would be synchronizations problems, such as the program ending earlier or the programs halting and running infinitely. To solve the issue had to account for each smoker process and add a counter so the agent can check if the process did its job N times before exiting.