CSC 33200 Lab Jonathan So

**Task 6 Report**

In this lab, we have to write a program that synchronizes 4 processes: agent, smoker with matches, smoker with paper and smoker with tobacco, using two ways, semaphores and pthreads. Both ways provide synchronization through locking. In the semaphore method, we use 5 semaphores, one for the lock, one for the agent and 1 for each of the smokers. In the agent process, represented as a child process, using the semaphore operation, P, we lock the critical section. Then in the agent CS, according to which two random ingredients is given out, the smoker with the last remaining ingredient gets signaled using the semaphore operation V. The agent gets puts to sleep after releasing the lock to the CS. The smoker processes are represented in the parent process as nested child processes. Locks and unlocks (P and V) of the smokers are placed accordingly. In each of the smoker’s process, after their lock to the CS has been released, the agent gets signaled (V) to disperse the next two random ingredients. This process repeats until the agent runs of out of ingredients, which in this case, after 10 times.

For the pthread method, it is very similar. However, instead of representing the 4 processes as nested child processes of the parent process, each of the processes are represented as thread functions, which are used by their corresponding threads. Similar to semaphores, we create a mutex for each process and for the lock (5 mutexes). In the main function, we create and join the threads. This method produces the same results as the semaphore method. Both methods are really similar, as it involves locking, which help provide synchronization. The only major difference is how the processes are represented, with processes being represented as child processes (semaphore) and processes being represented as thread functions (pthreads).