Jason Mejia 12/16/2021
CSC 332 Section M – Task 6

For my semaphore approach, I decided that I would have five locks to control the four processes. One lock for each process, and another main lock for the three smokers. The locks have also been placed for the child process as the semaphores have been created. For my pthread approach, I used one mutex that was locked and unlocked in each of my processes. I only had to write one agent process and one smoker process. Then three pthreads are created for the smokers as the implemented smoker process.

Both approaches printed what the agent put onto the table, and then printed the appropriate smoker that would pick the items up from the table and smoke. The pure ease of implementation with pthreads makes it easier to solve a problem like cigarette-smokers. When it came to the semaphore implementation it required more lock mechanisms. Specifically, I used five locks. One lock was generally used throughout the code to maintain processes from leaking into each other. One lock was used for the agent. The remaining three locks were used for each smoker. My pthreads only required one mutex for both processes so as previously mentioned it was easier to implement.

Besides that, I appreciate the semaphore approach to the problem more than the pthreads approach. With semaphores there is more control in the code and is a better representation, in my opinion, of what we learned as far as concurrency in the recitation.