Andy Zhou

Lab #6 Report

In this lab, students need to code the Cigarette Smoker's Problem using semaphores and threads. There will be two files using each respectively. To code the Cigarette Smoker's Problem, there is an agent and three smokers. Each smokers have one of three ingredients for a cigarette while the agent has all three but will only provide two at a time. After the agent leaves two ingredients, the smoker with the remaining ingredient will it. This process will loop for 10 times. Therefore, there is a reliance on communication between processes/threads. It should be noted that there is a lock semaphore/mutex which is used to ensure only one operation is happening at a time. It should also be noted that there is a random number of seconds (between 1 and 3) between each transaction of the agent and the smoker. This is to make it more realistic.

In the semaphore variation of the Cigarette Smoker's Problem, semaphores are created for each ingredient as well as for one for the agent and one for the lock. There are also 4 total processes. 3 of which are for each smoker and the parent process is the agent. The program starts off with the agent generating a random number from 1 to 3 that determines which two ingredients are given. This increments the ingredient semaphore which signals which smoker process can take the ingredients. The process then prints a message saying the smoker took the two ingredients. This then decrements the ingredient semaphore then increments the agent semaphore, allowing the agent process to give two of the ingredients again. Therefore, by incrementing and decrementing the semaphores appropriately, it will act as a signal to let the appropriate process know it can take the ingredients.

In the thread variation of the Cigarette Smoker's Problem, the logic is similar. However, instead of using semaphores, mutex is used and instead of using forks, we use threads. There are a total of 4 threads, 3 for the smokers with different ingredients and 1 for the agent. To allow communication between the threads, we utilize mutex locks for each ingredient. This allows the program to know which thread is allowed to run. The rest is like the semaphore variation of the problem.

Ultimately, the semaphores act as a signal to each process while the mutex acts as a locking mechanism to allow certain program threads to run.