Tahsin Parvez
CSC33200

Lab 7 Report

For this lab assignment, we use semaphores to help us synchronize the four processes in this problem. I used semaphores in order to represent the smoker match, smoker paper, some tobacco, agent, and run. The semaphores are used to disable a particular resource that can be accessed by all to only be accessed by certain individually opposed to simultaneously. This is why they are useful, as we had to wait for one process to complete to move on to another while utilizing a shared process. The agent signals the other processes, so that each semaphore waits for the next semaphore to finish. I used IPC_PRIVATE and IPC_CREAT to match each semaphore with its process. It can only be accessed by a child process. This process is iterated over and over for 10 different iterations. There are also the run process which will lock and unlock for the agent process in between each separate iteration of the process.

Using semaphores allows all of the four processes to be synchronized and are accessed exclusively, which prevents synchronization issues and race conditions from occurring during this process.

Pthreads solution

For the Pthreads solution, the process is of course still synchronized with the same names. The main process for this solution assigns a random number with a minimum and maximum bound. Depending on which number is chosen, the main program will put any two items on the table. For example, in my code if the random number is 1, the tobacco and the paper will be placed on the table. For 2, tobacco and match, and for 3, match and paper. Pthread functions are used to create and join processes, but also so that they are synchronized, run in the proper order, and do not access the same resources at once by putting processes to sleep.