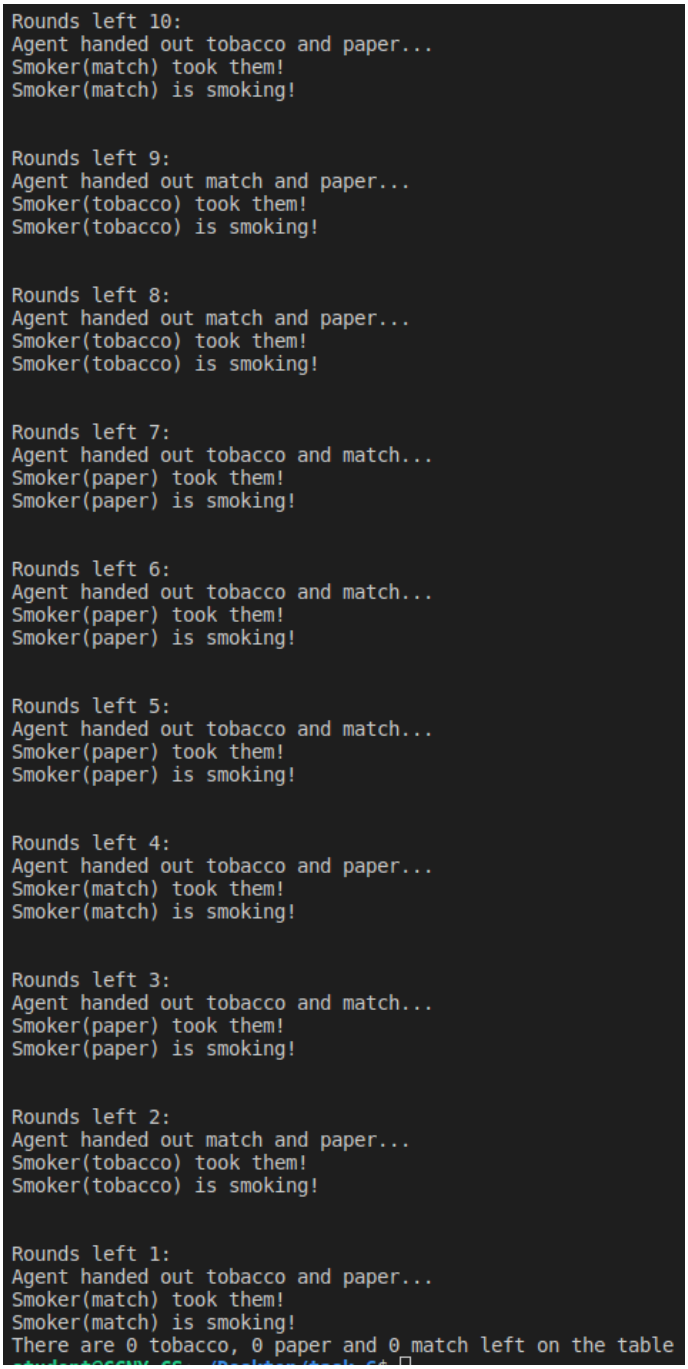
Jia Ming Ma

CSc 332 Lab

May 14, 2020

Task 6: Cigarette Smokers Problem

Output: remember to compile with gcc -o task6 task6.c -pthread (the pthread parameter is important)



Explanation

I will first explain the picture with the outputs. As you can see for all the rounds, the agent hands out 2 random items, and the agent wakes up the smoker that has the 3rd item needed to make a cigarette takes them. After that, the smoker starts smoking, then they will wake the agent up again after they are done. As you can see on the last line, in the end there is 0 of every item left on the table, meaning there is no synchronization problem. Agent puts 2 items; the smokers take 2 items.

I will now explain how mutex synchronization works for this problem. Initially, we create 5 mutexes, lock, agent lock, tobacco smoker lock, paper smoker lock, and match smoker lock. They are all initialized as locked except for lock, which is unlocked. I then have 4 threads for the 3 smokers and agent. The 3 smokers will all be initially be sleeping because they are waiting for their respective lock to be unlocked since they are initially locked. The agent will be looking to lock the lock to access the critical section, which is unlocked. Then it will choose random items to put out and unlock the smoker that has the 3rd item and go to sleep after locking the CS lock (waiting for agent lock to be unlocked). The smoker will wake up because they can now lock their respective lock and the CS lock, and then they will smoke. After smoking, they will unlock the agent lock and the CS lock so that the agent can access it, then the agent can lock its respective lock and choose items again. This repeats until the specified number of rounds.