CSc332 OS Lab
Weifan Lin
Lab Report
05/19/2016

Cigarette Smokers Problem

In this assignment we were asked to write a program that consists of an agent process and three smokers processes which needs three ingredients: tobacco, paper, and matches. The agent has all three ingredients. In this problem, we would like to find a way to synchronize these four processes in order to share the resources.

I was able to successfully synchronize these processes as only one process was allowed in the critical section at any given time. The semaphores for the smokers were initialized to 0, so they cannot run unless the agent supplies the materials first. Agent was also initialized to 0 so that it will sleep after supplying the items. Finally, the lock mutex was initialized to 1. That way the agent can run the first loop and will grant permission to the compatible smoker after supplying the items. Then agent will sleep, until the smoker uses the materials and wakes up the agent. The smokers all start off sleeping, and get woken up by agent when the items they need are put on the table. The smokers which do not have the necessary ingredients will be sleeping because the smokers cannot smoke without first being given the items by the agent. The agent sleeps after putting the items, and is woken up by a smoker after the items have been used. By initializing all the semaphores to 0 besides lock, it guarantees that they must wait for agent to be done. Then the agent must also wait for smokers to be done to run again. In my code I also added a sleep in each critical section to show that no process has entered the critical section while a process is in there.