*/\**

*Csc332 - Operating System Lab*

*DUE: November 30, 2018*

*Hasibul Islam*

*Task 6 – Cigarette Smokers Problem*

*Dad-Son Problem*

*\*/*

**Brief Report:** Cigarette Smokers Problem

We were asked to write a program which include an agent process and three smokers processes that needs three ingredients: tobacco, paper and matches. The agent will have all these ingredients and we need to find a way to synchronize these four processes in order to share the resources.

The solution I came up with required only one process to be in the critical section at any given time to synchronize these processes. The semaphores for the smokers were initialized to 0. This won’t allow them to run until the agent supplies the materials. The agent will also be set to 0 initially which will allow them to sleep after supplying ingredients. The lock mutex will be initialized to 1 to allow the agent to run the first loop, and will give permission to the compatible smoker after supplying the items. The agent won’t run until the smoker uses all the materials. The smokers will start of sleeping and will only be woken by the agent when the items are put on the table. Any smokers without necessary ingredients will be sleeping since they cannot smoke without any missing ingredients. The agent will sleep after distributing the items and will be woken up by a smoker after they used up ingredients. Since the all semaphores are initialized to 0 besides lock, it will guarantee that they must wait for agent to be done. Then agent will have to wait for smokers to done smoking to be run again. I also decided to add a sleep in each critical section to show that no other process has entered critical section while there is one already in process.