Lianming Wu CSC332 OS-L Lab

Instructor: Prajwal Khatiwada

Lab 6 Cigarette Problems

Semaphore approach:

- Create 5 semaphores, one for general lock, and one for each child process
- Other than the general lock that is use for critical sections, other locks initialized to 0, which means if we P(smoker1), smoker1 would be put in sleep until someone else called V(smoker1).
- We put all the smokers to sleep initially and let agent call V function to wake up the corresponding smokers based on the randomized result from 0 to 2, each represent a smoker. After that, agent was put into sleep, until the smoker it awakes calls V(agent), and next round begin.
- Pros: By putting the smokers into sleep, lesser system resources are used, no busy waiting required, and the logic are clearer to understand.
- Cons: We need inter-process communication to count the resources as we don't use threads, which make the code harder to read.

pthread solution:

- Create 4 threads each representing a child process
- A global array that contains the number of resources is created for smoker to check if they could smoke the cigarette.
- Pthread mutex are used to lock the critical section.
- Agents increments the resources array accordingly based on the randomized result from 0 to 2. All of the smokers busy checking if it's their turn to smoke, after smoker smoked and used the resources agent would put new resources in, new turn begin. Thus, the project requirement is achieved
- Pros: No inter-process communication required as we are using threads, only one mutex required instead of 5 semaphores.
- Cons: Busy waiting burns CPU resources for nothing.