CSc 332 (L) - Operating Systems
Lab – Fall 2020

Task 6: Cigarette Smokers Problem

November 20, 2020

Due: December 04, 2020

Problem Description

Consider a system with 3 smoker processes and 1 agent process. Each smoker continuously rolls a cigarette and then smokes it. The smoker needs three ingredients: tobacco, paper, and matches. One of the smokers has paper, another has tobacco, and the third has matches. The agent has an infinite supply of all three materials and (randomly) places two of the ingredients on the table each time. The smoker who has the remaining ingredient then makes and smokes a cigarette, signaling the agent on completion. The agent then puts out another two of the three ingredients, and the cycle repeats.

TO DO: Write **two** programs to synchronize the agent and smoker processes: one using **semaphores** and another using **pthread libraries.**

Instructions

   • Please see this link for pseudocode: http://www.cs.umd.edu/~hollings/cs412/s96/synch/smokers. html

   • Though the description says the agent process can infinitely supply two of the three ingredients, you can assume that the agent places ingredients only a finite number of times, say for example 10.

   • You need to use the sem.h header file in your semaphore-based solution.

   • You need to compile code with linker **-pthread** when using pthread library.

      Eg: $ gcc -o Task6 task6.c -pthread

Submission Instructions

Submit your solution to the cigarette smokers problem along with a brief report (atmost 1 page) on how your solution synchronizes the four processes. Draw comparison between semaphore and pthread library based solutions highlighting the merits and demerits of both approaches. The C solution files must be internally documented.

Zip your source & explanation text files into a single folder as: task6_*lastname*.zip. Email link to your zip file with the subject line, "Task 6 - CSc 332(L) - *lastname*".

******