

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE PILANI,  
K. K. BIRLA GOA CAMPUS  
I SEMESTER 2019-2020  
Operating Systems (CS F372)  
Assignment 5 [Non-evaluative Assignment]**

---

**Question #1:                      The Sleeping Barber Problem**

This program is to test your understanding in pthread & semaphore functions. A barbershop consists of a waiting room with N chairs and the barber room containing the barber chairs. If there are no customers to be served, the barber goes to sleep. If a customer enters the barbershop and all chairs are occupied, then the customer leaves the shop (without waiting for service). If the barber is busy but chairs are available, then the customer sits in one of the free chairs. If the barber is asleep, the customer wakes up the barber. Write a program to coordinate the barber and the customers. Each customer and barber should be a different thread. Use actual semaphores implementation.

**Consider the following issues:**

Is there a way by which you can evaluate through execution of your program whether the three conditions of Critical Section are satisfied?

How would you incorporate priority amongst customers? Let us say that there are two groups: staff and student. A waiting student customer has priority over the waiting staff customer (because the students are supposed to be busy !!!). Therefore, a barber attends to the waiting student customer after finishing the work on present customer. Modify the code to include the priority condition also. Your program should run both with priority and without priority. In priority approach avoid starvation by enforcing the condition of “each staff can allow maximum of 5 students to overtake”.

Another modification is that there is more than 1 barber; say there are four (N) barbers. Now modify the code to include this condition also. (A good code would include the number of barbers as variable implying the barbershop could hire or fire some of the barbers. Hiring and firing can happen with respect to the number of customers per time).

[The basic version of this code is available in one of your reference books – William Stallings, “Operating Systems Internals and Design Principles”, 6th edition, Pearson Edition, 2009]

### **Question #2:                      The Cigarette Smokers Problem**

This program is to test your understanding in pthread & semaphore functions  
PART I:

Consider a system with three smoker processes and one agent process. Suppose a cigarette requires three ingredients, tobacco, paper and match. There are three chain smokers. Each of them has only one ingredient with infinite supply. There is an agent who has infinite supply of all three ingredients. To make a cigarette, the smoker has tobacco (resp., paper and match) must have the other two ingredients paper and match (resp., tobacco and match, and tobacco and paper). The agent and smokers share a table. The agent randomly generates two ingredients and notifies the smoker who needs these two ingredients. Once the ingredients are taken from the table, the agent supplies another two. On the other hand, each smoker waits for the agent's notification. Once it is notified, the smoker picks up the ingredients, makes a cigarette, smokes for a while, and goes back to the table waiting for his next ingredients.

- A. Write a program that simulates this system, with three smokers and the agent being simulated by threads.

## PART II:

Consider there are  $n$  number of smokers and a single agent. In this case, there can be more than one smoker having the same ingredient at the same time. The priority among the smokers is set on a first come first serve basis. You are required to find out, among the  $n$  smokers, which smoker has the ingredient other than the two ingredients placed by the agent (There can be any number of smokers) and then check for the priority among them.

B. Write a program to synchronize the agent and the  $n$  smokers

[<http://www.laynetworks.com/Cigarette-Smokers%20Problem.htm>]

## PART III:

In this part, the smokers have been classified into two categories, staff and student. The staff has more priority than the student because the student respect the staff and the priorities within the staff and the student is again on a first come first serve basis. Here we need to take care of student process being starved. I.e. you need to introduce some aging condition like the student can allow only 3 staffs to overtake. Modify the program to

C. Synchronize the above processes with only one agent.

D. Synchronize the above processes with  $n$  agents.

## **Question #3: Reader / Writer Problem**

This program is to test your understanding in pthread & semaphore functions. Implement a Reader write problem (with 'N' reader and 'M' writers) with following constraints

- Maximum of 5 readers can access the critical section at a time
- If one writer is accessing the critical section no other writers or readers can access the critical section.

- A writer can allow maximum of 2 readers to overtake him while it is waiting for the critical section.
- The problem must take care of precedence (if one writer is executing and 'N' readers and 'M' writer are waiting for critical section, the next process to execute must be the first process came to the critical section).

Implement this with the help of semaphores

---