

Operating Systems (CSL 3030)

Readme file

Lab Assignment	08
Roll No.	B20CS018
Name	Harshita Gupta

Part-I

Command lines: gcc Late-Night_Pizza.c
./a.exe

Approach:

If a student finds that the pizza is gone, the student goes to sleep until another pizza arrives. Once the pizza is delivered, the delivery person should wake up all the sleeping students. In order to simulate this behavior, we have to use a special type of construct named **Condition Variable**.

What is a Condition Variable?

Condition variables allow one to check for a condition, and make a thread wait until the condition is satisfied. When the condition is satisfied, it will wake up the sleeping thread. There is a special method to broadcast and wake up all the sleeping threads as well.

Part-II

Ques1. Sleeping barber's synchronization problem

Command lines: gcc Sleeping_barber.c
./a.exe

Approach:

1. The solution to this problem includes four semaphores.
2. The first is for the customer which counts the number of customers present in the waiting room (the customer in the barber chair is not included because he is not waiting).
3. The second mutex is used to provide the mutual exclusion which is required for the process to execute.
4. The third, the barber 0 or 1 is used to tell whether the barber is idle or is working,
5. And, the fourth mutex is used to make the customers wait until the Barber is done cutting his/her hair.

Ques2. Banker's Algorithm

Description

This program implements the Banker's Algorithm and simulates a banking machine, where the banker maintains track of the resources and clients (the user) can request more or fewer resources from the banker. The banker only grants a request if it satisfies the safety algorithm. If a request does not leave the system in a safe state, the banker denies it.

Screenshots

```
Number of customers: 5
Currently available resources: 9 8 7 9
Maximum resources from file:
6 4 7 3
4 2 3 2
2 5 3 3
6 3 3 2
5 6 7 5
Enter Command: RQ 3 1 2 3 1
Resource request is satisfied.
Enter Command: RQ 0 1 1 1 1
Resource request is satisfied.
Enter Command: *
Maximum:
6 4 7 3
4 2 3 2
2 5 3 3
6 3 3 2
5 6 7 5

Current Need:
5 3 6 2
4 2 3 2
2 5 3 3
5 1 0 1
5 6 7 5

Current Allocation:
1 1 1 1
0 0 0 0
0 0 0 0
1 2 3 1
0 0 0 0

Currently Available Resources: 7 5 3 7
Current State: SAFE
```

```
Safe Sequence is: < 1 2 3 0 4 >
Executing threads:
--> Client/Thread 1
    Allocated Resources: 0 0 0 0
    Needed: 4 2 3 2
    Available: 7 5 3 7
    Thread has started.
    Thread has finished.
    Thread is releasing resources.
    New Available: 7 5 3 7
--> Client/Thread 2
    Allocated Resources: 0 0 0 0
    Needed: 2 5 3 3
    Available: 7 5 3 7
    Thread has started.
    Thread has finished.
    Thread is releasing resources.
    New Available: 7 5 3 7
--> Client/Thread 3
    Allocated Resources: 1 2 3 1
    Needed: 5 1 0 1
    Available: 7 5 3 7
    Thread has started.
    Thread has finished.
    Thread is releasing resources.
    New Available: 8 7 6 8
--> Client/Thread 0
    Allocated Resources: 1 1 1 1
    Needed: 5 3 6 2
    Available: 8 7 6 8
    Thread has started.
    Thread has finished.
    Thread is releasing resources.
    New Available: 9 8 7 9
--> Client/Thread 4
    Allocated Resources: 0 0 0 0
    Needed: 5 6 7 5
    Available: 9 8 7 9
    Thread has started.
    Thread has finished.
    Thread is releasing resources.
    New Available: 9 8 7 9
Thread execution complete.
```

Installation

1. Start up a terminal and enter the directory you have chosen.
2. Rename the file *makefile.gmk* to *makefile*.
 - `mv makefile.gmk makefile`
3. Run the command `make`.
4. The program is now ready to be invoked.
 - The program is invoked by passing the number of resources of each type via the command line to initialize the available array by these values. For example, if there were four resource types, with ten instances of the first type, five of the second type, seven of the third type, and eight of the fourth type, you would invoke your program as follows: `./Bankers_Algo 10 5 7 8`
5. Once the program has been invoked, it awaits user input commands.
 - Refer to the Features section for more details regarding commands.