

MINI PROJECT

Table Allocation Management in a Small Restaurant

Mini Project Presentation:

A small restaurant can accommodate only a limited number of customers at the same time. This limit is determined by the number of tables available in the restaurant. There are only **four tables**, differing by the number of chairs: **two tables with 2 chairs**, and **two tables with 4 chairs**.

Customers enter the restaurant in groups of **2, 3, or 4** people at most. There is only **one host** (receptionist) who manages the allocation of the four tables, aiming to fill them as efficiently as possible. The idea is as follows:

- Customers belonging to the same group must sit together at the same table. So only one group can sit at the table.
- The tables with 2 chairs can only seat a group of 2 customers.
- For the tables with 4 chairs, a group of 4 customers has higher priority than a group of 3 customers, which in turn has higher priority than a group of 2 customers. Therefore, the descending priority order is: **group of 4 > group of 3 > group of 2**.
- If a group of 2 customers arrives and the two types of tables are empty, the priority goes to the tables with 2 chairs.

System operation:

1. As soon as a group of customers arrives at the restaurant, the host directs them to an empty table if one is available, while considering the above priority rules.
2. If all tables are occupied, or if the empty table does not have enough chairs for the group (for example: a group of 3 customers and only a 2-chair table is free), the group must wait.
3. When a table becomes available, the host must assign it to a waiting group, still following the priority rules mentioned above.

Objective of the Mini Project:

The goal of this mini project is to simulate the operation of the table allocation system using **semaphores**.

- The system's process types include: **one host process** who assigns the tables. And multiple processes representing **group-of-2 customers**, **group-of-3 customers**, and **group-of-4 customers** (Each one requests access to the restaurant, then eats, and finally leaves the restaurant).
- The **shared resources** are the four tables in the restaurant.

Work to Be Submitted:

The mini project can be carried out in **groups of up to three students**. However, the evaluation will be done individually.

The final submission must be presented as a **ZIP (WinRAR) archive** containing:

- A **report in PDF format** containing the well-commented algorithmic solution using semaphores, clearly explaining the semaphores used. The cover page must include the names and the groups of the students who carried out the mini-project together.
- A **folder named “source”** containing the well-commented **source file (C program)** of your application, and the executable file of your program.
- A **text file** explaining how to run the program (how to use it), if necessary.

Technical Notes:

- You must implement the mini project in **C language** under **Ubuntu**, using **POSIX** (mandatory).
- The user of the application can enter customer groups in **any order** (for example: P1-G2, P2-G3, P3-G3, P4-G4, P5-G2, P6-G3, P7-G2, ...). Each group is launched as a thread in your program, plus the host Thread, all are executed in parallel.
- The application must **display the table allocation in real time** as it changes. You must display each Thread whenever it is blocked or awakened.
- You may use **random sleep() durations** to simulate the time each group of customers spends eating before leaving the table.

Note: You may use either a **graphical user interface (GUI)** or simply **console messages (printf)** to demonstrate the correct operation of your application, for example:

Enter a series of customer groups:

P1-G2 P2-G3 P3-G4 P4-G4 P5-G3 P6-G3 ...

Table allocation:

Arrival of P1-G2
Assign Table-21 to P1-G2
Arrival of P2-G3
Arrival of P4-G4
Assign Table-41 to P2-G3
Assign Table-42 to P4-G4
Arrival of P5-G3
P1-G2 finished eating
Arrival of P6-G3
P4-G4 finished eating
.....
.....

P1, P2, P3, ...: these are the names of the **Threads** launched in parallel (groups of customers).

G2, G3, G4: represent the **number of customers** in each arriving group.

Table-21 and **Table-22:** the **tables with 2 chairs**.

Table-41 and **Table-42:** the **tables with 4 chairs**.

Submission Date:

Only one student from each group of three must submit their project archive (**WinRAR archive**) on the university platform, in the **MINI-PROJECT** section, **no later than Friday, December 5, 2025 (midnight)**. Mini projects submitted after this deadline will **not be accepted**.

Six evaluation sessions for the application will be scheduled starting **Sunday, December 7, 2025** (during your regular lab session). Each session will concern one sub-group.

Note:

- Make sure to properly comment your source code so that it can be easily understood.
- You will receive a zero if the report or the program is a copy-paste of another work, or if the program was not entirely developed by you.

**** Good Luck ****