laSalle
UNIVERSITAT RAMON LLULL

Operative Systems 2023-2024

Session 10: Message Queues 29/11/23

Objective

In this session more complex techniques of operative systems will begin to be used. This session will focus on the creation of message queues to manage messages and data in an environment with more than one thread of execution.

Motivation

Specifically, in this session, the student must train:

- Creation of message queues (*msgget*)
- Closing message queues (*msgctl*)
- Sending messages(*msgsnd*)
- Receiving messages (*msgrcv*)
- Configuration of a message queue

Previous documentation:

It is recommended to read the following references:

SALVADOR, J. (2013). *Programació en UNIX (per a practiques de Sistemes Operatius)*. Páginas 56 - 71.

The pizza

There has been a high request for pizzas in the last days at the university's bar. They have asked us to develop a program to deal with the orders and delivery of the pizzas, we have been the following directions:

- Students will come in groups and each group will be assigned to a table. Table assignments will be done by order of attendance, 1st group in will have table number 1, 2nd group will have table number 2, Xth group will have table number X.
- No one will make an order until everybody is seated and have been told by the employees that they can start making orders.
- Each order can only contain one type of pizza. The name of the type will be written by the students and will have a maximum of 20 characters. Students can make multiple orders, but they must wait until their last order was delivered to make a new one.
- The bar employees will process the order and then deliver-it to the table, the processing time is 3 seconds for each pizza.
- Once students are done, they must tell the bar employees they are going to leave the bar so the table can be cleaned.
- Once all the tables have left the bar employees can go home.

Execution order and communication protocol

The communication between the bar employees and the table will be done via a message queue and the provided struct in Figure 1, when a field is not used it should be filled with padding/dummy data. The communication process will be the following[1]:

```
typedef struct
{
    long mtype;
    char header[HEADER_SIZE];
    char pizzaName[PIZZA_SIZE];
    int pizzaQuantity;
} Msg;
```

Figure 1 - Msg. Queue Struct

1. When all the tables are ready an *ENTER* (\n) should be pressed at the bar, which will send a message to all the tables with the header "ORDER".
2. Tables will wait until they receive a message with "ORDER" as a header. When received, a menu will be shown with two options:
    a. **Order a pizza:** The user should input a pizza name and a pizza quantity and a message with the corresponding data will be sent to the bar, the header of the message must be "PIZZA". At this point we need to wait until pizzas are done.
    b. **Finish:** We need to tell the bar we have finished sending a message with the header "FINISH". At this point the table process must finish correctly.
3. The bar starts receiving petitions from the tables, this can be two options mentioned before:
    a. A message with the header **PIZZA**. At this point the bar will show, which table the order came from, how many pizzas were ordered and its name, how long will it take to make the pizzas. Once the pizzas are ready the bar must send a message to the corresponding table with the header "DELIVERY".

[1] The diagram in Figure 2 might help when reading the following points.

b. A message with the header **FINISH**. The bar will show a message saying which table has finished, when all tables have finished, the bar will quit freeing all the used resources.

4. When a table receives a message with "DELIVERY" as a header it will stop waiting, show a message saying that the pizzas are delicious and get back to point 2.

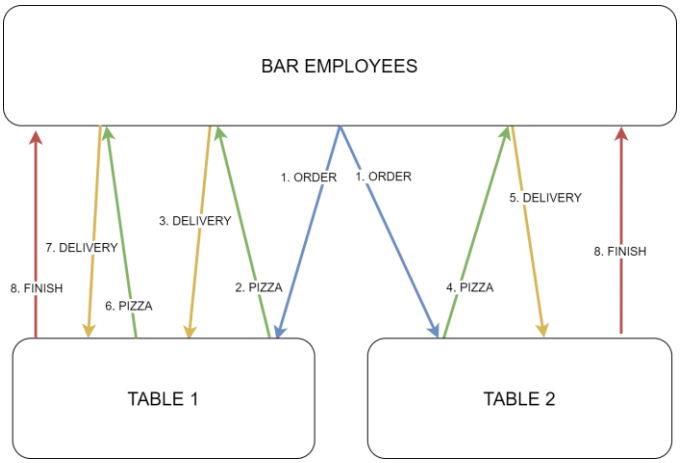

Figure 2 - Comunication protocol diagram

Example of execution

Where n is the number of tables.

Terminal 1	Terminal 2	Terminal 3	···	Terminal n+1
`./bar n`	`./table 1`	`./table 2`	...	`./table n`

```
marc.valsells@montserrat:~>bar
Please introduce the number of tables
marc.valsells@montserrat:~>bar 1 home
Please introduce the number of tables
marc.valsells@montserrat:~>bar 2
Press enter when the tables are ready to make orders

Table 1 ordered 2 Peperoni pizzas
Taking 6 seconds to process them...
Order processed
Table 2 ordered 4 Bacon pizzas
Taking 12 seconds to process them...
Order processed
Table 1 has finished
Table 2 ordered 7 Ham and Chesee pizzas
Taking 21 seconds to process them...
Order processed
Table 2 has finished
All tables have finished, closing the bar...

marc.valsells@montserrat:~>
```

```
marc.valsells@montserrat:~>table
Please introduce the table number
marc.valsells@montserrat:~>table 1 2
Please introduce the table number
marc.valsells@montserrat:~>table 1
Wating to be able to make orders...
You can make orders now

--- Menu ---
1. Order a pizza
2. Exit
Option: 1
Introduce the pizza name (max chars: 20): Peperoni
Introduce the pizza quantity: 2
Waiting for your pizzas...
Your pizzas are ready! Nyaaami

--- Menu ---
1. Order a pizza
2. Exit
Option: 2
Goodbye!
marc.valsells@montserrat:~>
```

```
Wating to be able to make orders...
You can make orders now

--- Menu ---
1. Order a pizza
2. Exit
Option: 1
Introduce the pizza name (max chars: 20): Bacon
Introduce the pizza quantity: 4
Waiting for your pizzas...
Your pizzas are ready! Nyaaami

--- Menu ---
1. Order a pizza
2. Exit
Option: 1
Introduce the pizza name (max chars: 20): Ham and Chesee
Introduce the pizza quantity: 7
Waiting for your pizzas...
Your pizzas are ready! Nyaaami

--- Menu ---
1. Order a pizza
2. Exit
Option: 2
Goodbye!
marc.valsells@montserrat:~>
```

laSalle
UNIVERSITAT RAMON LLULL

Operative Systems 2023-2024

Session 10: Message Queues 29/11/23

Considerations

- You can assume that the bar will always be executed first.
- You need to hand in **two** files: `table.c` and `bar.c`, **one compressed will NOT be accepted**. These files need to have the login and full names of the group members. Otherwise, it'll be considered invalid.
- It is not allowed the use or programming of lists or other data structures.
- **Only one (1) message queue can be created for the solution of this session.**
- In both cases, the files .c will need to have the names and logins of the students.
- It can be assumed that the data introduced will always be correct.