

## Objective

The objective of this session is to carry out an extension on sockets, a communication mechanism between processes that may be running on the same or different systems.

## Motivation

More specifically, with this session, the student will exercise:

- Creation of sockets (socket)
- Association (bind)
- Receiving connections (listen)
- Accepting connections (accept)
- Closing the connection (close)
- Monitoring connections (select)
- Passing parameters to an executable (argc, argv)

## Previous documentation

To complete this session, it is recommended to read the following references:

SALVADOR J., CANALETA X. (2014). *Programació en UNIX per a pràctiques de Sistemes Operatius*, Publicacions d'Enginyeria i Arquitectura La Salle (Edició PDF). Capítulo 7. Pág. 97.

## LS Broker

The company's equity portfolio is diversified and constantly changing, so we need assistance to maintain effective control. In order to understand our current and potential investments, this session has been organized.

You will need to develop a management system for the stock portfolio, which will receive client requests and inputs from the team's keyboard, so the ability to listen to and process both types of requests is required.

These requests may involve adding new stocks to our portfolio or querying existing investments.

For this session, the client will come implemented, we are only interested in the server implementation. It will receive two arguments:

1. The port on which you want to deploy the server (you will find yours in the list of ports of the practice).
2. The name of the file where the portfolio data will be stored. It will have the following format:

```
<n_products>
<company>|<units>&<price>
<company>|<units>&<price>
<company>|<units>&<price>
...
```

This file should be written between executions.

From the client, the client will be able to launch the following actions:

### Get Company

Information about an object may be requested as follows:

```
Is-<name>|<U/P>\n
```

U/P refers to the fact that if the message contains a U we must return the units of shares we own and P will return the price.

The server will always return the same frame:

```
<name>|<requested parameter>\n
```

And in case of not having the product, the parameter will be -1.

### Add company

To add companies to the portfolio, the client will send the following:

```
Add-<name>|<units>&<price>\n
```

The server will reply with an "OK" once the record has been added or "ERROR" in case of an error, such as if the company you are trying to insert into the database already exists.

To run the server:

```
./S7 <port> <portfolio file>
```

To run the client:

```
./client<ip> <port>
```

## Examples of execution

### Client

```
./client 127.0.0.1 3456
```

```
WELCOME TO LS BROKER
```

```
-----
```

```
1-Get company info
```

```
2-Add company to portfolio
```

```
-----
```

```
2
```

```
Which company would you like to add?
```

```
ready
```

```
How many shares do you want?
```

```
34
```

```
What is the price?
```

```
6574
```

```
The stock has been purchased correctly
```

```
-----
```

```
1-Get company info
```

```
2-Add company to portfolio
```

```
-----
```

```
1
```

```
Which company do you want to consult?
```

```
wuxia
```

```
Do you want to know their price or units? P/U
```

```
U
```

```
The company Microsoft has the value of the parameter entered: 420
```

### Server

```
>./server 3456 portfolio.txt
```

```
Waiting for connections...
```

```
We are waiting for a message or customer
```

```
New customer!
```

```
We are waiting for a message or customer
```

```
New message!
```

```
Thread Launched!
```

```
We are waiting for a message or customer
```

```
A customer has disconnected.
```

```
We are waiting for a message or client.
```

```
New client!
```

```
We are waiting for a message or client
```

```
Is-link|P
```

```
New message!
```

```
Thread Launched!
```

We are waiting for a message or a customer  
The company link has the value of the entered parameter: 32  
We are waiting for a message or customer  
Is-perro|U  
New message!  
Thread Launched!  
We are waiting for a message or a client  
The dog company does not exist in the database.  
Add-Perro|12&32  
New message!  
Thread Launched!  
We are waiting for a message or a customer  
The stock has been successfully purchased  
A

## Considerations

- The server program will receive the keyboard commands in the same format as it would receive the message from a client, i.e. "ls-<name>|<U/P> " and "Add-<name>|<Units>&<Price>".
- The database is case sensitive.
- The monitoring of the clients must be done with select().
- Disconnection of clients from the server must be controlled.
- All shared resources must be protected
- All critical regions must be monitored. Screen included.
- It can be assumed that the server program will be executed before the client program.
- It can be assumed that the format of the input parameters will always be correct.
- It can be assumed that the client will always respect the frames and that the format of the input files will not contain errors.
- The use of "system" or "popen" or analogous functions of the same family is not allowed.
- All input and output must be done with file descriptors, printf, scanf, FILE\*, getchar, or similar are not allowed.
- Compile using the -Wall and -Wextra flags.
- Any practice that contains warnings will be ineligible.
- **All resources must be freed, otherwise the practice will be ineligible.**
- **Connections must be closed as well as possible, including unexpected closures. Hanging connections will not be accepted.**
- The output of the program must be identical to that of the statement (see examples of execution).
- **A file with the name server.c must be submitted with the names and logins of the members of the group. Otherwise, the session will not be corrected.**
- It must be compiled using the -Wall and -Wextra flags.
- Any practice that contains warnings or errors will be ineligible.