Goal

The objective of this session is to get a first introduction to the sockets, a mechanism of communication between processes that may be running on the same or different systems.

Motivation

More specifically, with this session, the student has to exercise:
- Creation of sockets (socket)
- Establishment of connections (connect)
- Connection closure (close)
- Sending messages between client and server (read, write)
- Passing parameters to an executable (argc, argv)

Previous documentation

To complete this session, it is recommended that you 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 en adelante.

Tic Tac Toe

At La Salle Bonanova they want to add interactive games in class, and a game that children like is the Tic Tac Toe. Because of this, they asked the University students to help them out. The game has a board game of 3 by 3 where each player needs to place one piece (X, O) each turn. The game ends with a payer placing three of his pieces in a row, including in diagonal and anti-diagonal, in this case the play wins; or when all 9 slots are occupied by a piece, but no player has 3 of them in a row, in this case there is a tie.

As always, they have asked us to make the game at the last minute and with urgency… So, we only have this session to develop it. We are lucky that we only need to implement the client, the server was made by someone else and is provided to us. The communication process between the client and the server will be made with the following frames:

NewPlayer
To start a game, the player sends to the server: **NP#<player name>#**
The server answers acknowledging he has received the player's name: **NP#OK#**

StartGame
Once two players have joined the game, the server tells both players that the game can be started and which piece the player is going to use: **SG#X# / SG#O#**
Note: Here the client must indicate if it's using X or O, and display an empty board.

YourTurn and MyPiece
Every time it is a player's turn, the server will notify the client with the frame: **YT#**
The client must answer to the server indicating row (1-3) and column (A-C) where the piece is going to be placed: **MP#<column>#<row>#**
Note: The client should update the board with the new piece.

NewPice
When the other player places a piece, the server will send to the client which position has been occupied: **NP#<column>#<row>#**
Note: The client should update the board with the new piece.

EndGame
When a player wins, the server will send to both clients the name of the winner, in case of a tie it will send both players' names: **EG#<winner/s name/s>#**

Execution example

We can see an error when server is not running, and then how once it's started the 1st player is connected:

```
marc.valsells@montserrat:~>server 127.0.0.1 9090    marc.valsells@montserrat:~>client 127.0.0.1 9090
Waiting for a client...                              Error connecting to the server!
Client connected                                    marc.valsells@montserrat:~>client 127.0.0.1 9090
Waiting for a client...                              Enter your username: Marc
Player Marc joined                                  Waiting to start the game...
```

Both players have joined, no one has made a move yet:

```
marc.valsells@montserrat:~>server 127.0.0.1 9090    marc.valsells@montserrat:~>client 127.0.0.1 9090
Waiting for a client...                              Error connecting to the server!
Client connected                                    marc.valsells@montserrat:~>client 127.0.0.1 9090
Waiting for a client...                              Enter your username: Marc
Player Marc joined                                  Waiting to start the game...
Client connected                                       · A · B · C ·
Player Bernat joined                                ·  ---- ---- ----·
Two players connected, time to play                 1 |    |    |    |
                                                    ·  ---- ---- ----·
                                                    2 |    |    |    |
                                                    ·  ---- ---- ----·
                                                    3 |    |    |    |
                                                    ·  ---- ---- ----·



                                                    marc.valsells@montserrat:~>client 127.0.0.1 9090
                                                    Enter your username: Bernat
                                                    Waiting to start the game...
                                                       · A · B · C ·
                                                    ·  ---- ---- ----·
                                                    1 |    |    |    |
                                                    ·  ---- ---- ----·
                                                    2 |    |    |    |
                                                    ·  ---- ---- ----·
                                                    3 |    |    |    |
                                                    ·  ---- ---- ----·
                                                    It's your turn, enter the column[A-C]: |
```

After several turns of both players:

```
marc.valsells@montserrat:~>server 127.0.0.1 9090    .  .----.----.----.
Waiting for a client...                             2 |   | X |   |
Client connected                                    .  .----.----.----.
Waiting for a client...                             3 | O |   |   |
Player Marc joined                                  .  .----.----.----.
Client connected
Player Bernat joined                                It's your turn, enter the column[A-C]: A
Two players connected, time to play                 Enter the row[1-3]: 2
Bernat (O): New move at A1                             . A . B . C .
Marc (X): New move at B2                             .  .----.----.----.
Bernat (O): New move at A3                           1 | O |   |   |
Marc (X): New move at A2                             .  .----.----.----.
                                                    2 | X | X |   |
                                                     .  .----.----.----.
                                                    3 | O |   |   |
                                                     .  .----.----.----.

                                                    Waiting for the other player...
                                                    |
```

```
                                                    1 | O |   |   |
                                                     .  .----.----.----.
                                                    2 |   | X |   |
                                                     .  .----.----.----.
                                                    3 | O |   |   |
                                                     .  .----.----.----.

                                                    Waiting for the other player...
                                                      . A . B . C .
                                                     .  .----.----.----.
                                                    1 | O |   |   |
                                                     .  .----.----.----.
                                                    2 | X | X |   |
                                                     .  .----.----.----.
                                                    3 | O |   |   |
                                                     .  .----.----.----.

                                                    It's your turn, enter the column[A-C]:
```

A player has made 3 in a row and wins:

```
marc.valsells@montserrat:~>server 127.0.0.1 9090   1 | O |   |   |
Waiting for a client...                             .  .----.----.----.
Client connected                                    2 |   | O |   |
Waiting for a client...                             .  .----.----.----.
Player Marc joined                                  3 | X | X |   |
Client connected                                    .  .----.----.----.
Player Bernat joined
Two players connected, time to play                 Waiting for the other player...
Bernat (O): New move at A1                             . A . B . C .
Marc (X): New move at B3                             .  .----.----.----.
Bernat (O): New move at B2                           1 | O |   |   |
Marc (X): New move at A3                             .  .----.----.----.
Bernat (O): New move at C3                           2 |   | O |   |
Player Bernat is the winner                          .  .----.----.----.
Exiting server...                                    3 | X | X | O |
                                                     .  .----.----.----.
marc.valsells@montserrat:~>                          The winner/s is/are Bernat!
                                                    marc.valsells@montserrat:~>
```

```
                                                     .  .----.----.----.
                                                    3 | X | X |   |
                                                     .  .----.----.----.
                                                    It's your turn, enter the column[A-C]: C
                                                    Enter the row[1-3]: 3
                                                      . A . B . C .
                                                     .  .----.----.----.
                                                    1 | O |   |   |
                                                     .  .----.----.----.
                                                    2 |   | O |   |
                                                     .  .----.----.----.
                                                    3 | X | X | O |
                                                     .  .----.----.----.
                                                    Waiting for the other player...
                                                    The winner/s is/are Bernat!
                                                    marc.valsells@montserrat:~>
```

All the squares are filled with a piece and none of the players has made 3 in a row:

```
marc.valsells@montserrat:~>server 127.0.0.1 9090
Waiting for a client...
Client connected
Waiting for a client...
Player Marc joined
Client connected
Player Bernat joined
Two players connected, time to play
Bernat (O): New move at A1
Marc (X): New move at B2
Bernat (O): New move at A3
Marc (X): New move at A2
Bernat (O): New move at C2
Marc (X): New move at B1
Bernat (O): New move at B3
Marc (X): New move at C3
Bernat (O): New move at C1
There is a draw
Exiting server...

marc.valsells@montserrat:~>
```

```
1 | O | X |   |
. .----.----.----.
2 | X | X | O |
. .----.----.----.
3 | O | O | X |
. .----.----.----.
Waiting for the other player...
  . A . B . C .
. .----.----.----.
1 | O | X | O |
. .----.----.----.
2 | X | X | O |
. .----.----.----.
3 | O | O | X |
. .----.----.----.
The winner/s is/are Bernat and Marc!
marc.valsells@montserrat:~>
```

```
. .----.----.----.
3 | O | O | X |
. .----.----.----.
It's your turn, enter the column[A-C]: C
Enter the row[1-3]: 1
  . A . B . C .
. .----.----.----.
1 | O | X | O |
. .----.----.----.
2 | X | X | O |
. .----.----.----.
3 | O | O | X |
. .----.----.----.
Waiting for the other player...
The winner/s is/are Bernat and Marc!
marc.valsells@montserrat:~>
```

Help

Since la Salle Bonanova wants the code done urgently, they have provided us the code to print the board so we can focus on the communication protocol, it is up to every developer if they want to use it or not.

```c
#define printF(x) write(1, x, strlen(x))
#define BOARD_COLUMNS 3
#define BOARD_ROWS 3
#define BOARD_HORITZONTAL_LINE "· ------------·\n"
#define BOARD_TOP_LINE "  · A · B · C ·\n"

void printBoard(char board[BOARD_COLUMNS][BOARD_ROWS])
{
    int i;
    char *buffer;
    printF(BOARD_TOP_LINE);
    printF(BOARD_HORITZONTAL_LINE);
    for (i = 0; i < BOARD_ROWS; i++)
    {
        asprintf(&buffer, "%c | %c | %c | %c |\n",'1' + i, board[0][i], board[1][i], board[2][i]);
        printF(buffer);
        free(buffer);
        printF(BOARD_HORITZONTAL_LINE);
    }
}
```

Note: Empty squares of the board are represented by a space (' ') in the board matrix.

Considerations

- The IP and port to which the client must connect and the server must listen are passed as a parameter.
- It can be assumed that the format of the input parameters will always be correct. However, must check if the number of parameters is correct.
- The port assigned to the client and server is the one assigned to the group.
- The client is guaranteed to always enter an input followed by a return (\n).
- The server manages who is the winner, clients must only save and display the board distribution.
- The communication between client and server must be carried out following the guidelines of the frames indicated in the statement.
- The output of the program must be identical to that of the statement (see execution examples).
- The use of "system" or "popen" or analogous functions of the same family is not allowed.
- The use of global variables must be reduced to the minimum possible to make the program work correctly.
- All input and output must be done with file descriptors, the use of printf, scanf, FILE*, getchar, or similar is not allowed.
- It must be compiled using the –Wall and –Wextra flags.
- Any practice that contains warnings will be directly discarded.
- All resources must be released.
- A single "S4.c" file must be delivered that will have the names and logins of the group members commented. Otherwise, the practice will not be corrected.
- The server is guaranteed to never go down as long as a client is connected.
- If an unexpected frame is received the server will print its contents and ignore it.