Exercício 5: Trabalho Final

Aluna: Naomi Takemoto

RA: 184849

Instituto de Computação

Universidade Estadual de Campinas

Janeiro de 2021

Instruções de execução

Na pasta de códigos raíz, exercise5_code, em src existem os arquivos:

a. run_server.sh
b. run_client1.sh
c. run_client2.sh
d. run_client3.sh

Para reproduzir o experimento descrito aqui basta executá-los em abas direferentes do terminal. Eles irão compilar os arquivos necessários para a pasta bin no diretório raiz de código e irão executar servidor e clientes.

Resumo

Segue abaixo o resumo do funcionamento do jogo:

1. Ao iniciar o cliente, o usuário coloca seu id na pela entrada padrão
2. O cliente se conecta ao servidor (usando protocolo TCP)
3. O cliente manda o seu player id (assumindo que ele é único)
4. O receber o player id o servidor coloca o cliente na lista de jogadores disponíveis para uma partida e manda em resposta a lista de jogadores disponíveis.
5. O cliente recebe a lista de jogadores disponíveis. Se a lista não estiver vazia escolhe um deles (exceto ele mesmo) como oponente e manda para o servidor o seu id e o do oponente.
6. O servidor recebe os ids dos jogadores e os retira da lista de jogadores disponíveis.
7. Os clientes oponentes se enfrentam
8. Se o cliente se desconecta (manda uma mensagem vazia para o servidor).
   a. O servidor retira o id do cliente da lista de jogadores disponíveis.
9. Se o cliente manda o resultado do jogo
   a. O servidor coloca o id do jogador na lista de jogadores disponíveis

Funcionamento do programa

Para a implementação do cliente, optou-se por utilizar uma variável de estado que poderia assumir os seguintes valores:

Tabela 1: estados do cliente

Estado	**Descrição**
PENDING_LOGIN	O jogador tem que enviar ao servidor o seu id, para poder entrar na lista de jogadores disponíveis.
WAITING_PLAYERS_LIST	Depois de sair do estado acima, o jogador está na espera para que o servidor mande uma lista de jogadores disponíveis.
WAITING_NEW_PLAYER	O jogador (X) escolhe um dos jogadores disponíveis e manda o id para o servidor. O servidor recebe o id do oponente e manda o convite para o jogador escolhido (Y)
WAITING_ACCEPT	Se (Y) aceitar, ele manda uma mensagem de Accept para o servidor. O servidor retira X e Y da lista de jogadores disponíveis. Ambos X entra no estado PLAYER_TURN e Y entra no estado WAITING_OTHER_PLAYER_MOVE. Se (Y) recusar, o servidor manda uma mensagem de Deny para X volta ao estado WAITING_PLAYERS_LIST.
PLAYER_TURN	O X deve fazer sua jogada, que consiste em colocar um movimento válido na entrada padrão e enviar ao oponente via uma conexão UDP.
PLAYING	O jogador alterna com o oponente as jogadas até que o jogo acabe.
GAME_OVER	X ou Y detecta o estado do jogo e manda o seu resultado pra o servidor. Neste estado o jogador pode pedir os Scores para o servidor: Se pedir: vai par o estado WAITING_SCORES Caso contrário: volta o para o estado PENDING_LOGIN.
WAITING_SCORES	O cliente requisitou os scores e espera o resultado do servidor. Ao receber ele volta para o estado PENDING_LOGIN.

Nas tabelas abaixo são descritos os formatos de request e de response, que são strings. Cada token é separado por um espaço seguindo a convenção:

<request/response name>: <token1> <token1> …

O | é o operador lógico “ou”.

Tabela 2: Formato das requests

Request	**Descrição**
login: self_player_id port	O cliente manda sou id para o servidor, junto com a porta do servidor UDP (aqui assumimos que o IP é o localhos 127.0.0.1.
invite: self_player_id opponent_player_id port	O cliente manda seu id e o do oponente e porta de seu servidor UDP.
accept: self_player_id opponent_player_id yes	O cliente manda seu id e o do oponente e aceita a partida
deny: self_player_id opponent_player_id no	O cliente manda seu id e o do oponente e nega a partida
uploadResult: self_player_id DRAW | WIN | LOSS	O cliente manda seu id e o resultado de seu jogo.
requestScore: self_player_id	O cliente manda o seu id e pede os Scores.

Tabela 3: formato das responses

Response	
availablePlayers: player_id's	O servidor retorna a lista de players disponíveis
accept: "yes" | "no" address port	O servidor informa que o oponente aceitou a partida, envia o endereço e a porta do oponente.
scores:	O servidor retorna os scores de todos os jogadores
Invitation: player_id address port	O servidor manda para o jogador X o convite feito pelo jogador Y.

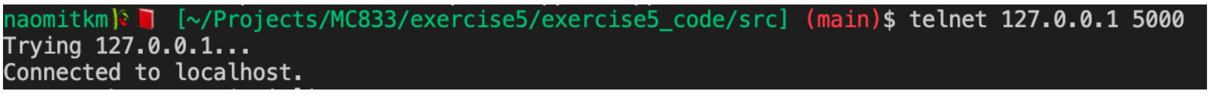
Implementação do Servidor

Para que fosse possível lidar com diversos clientes sem o uso de threads ou processos filhos no servidor, foi utilizada a multiplexação de IO com o comando select, que permite o monitoramento de diversos file descriptors [2].

Para testar se o servidor fato conseguia suportar várias conexões TCP simultâneas, foi utilizado o comando telnet em diversas chamadas concorrentes (diferentes abas do terminal). Mais adiante neste documento há o exemplo com a execução de uma partida.

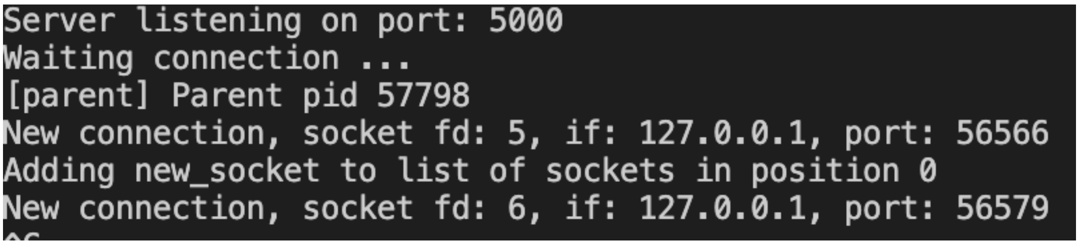
Utilizando o comando telnet, foi possível substituir os clientes para fim de testes de conexão. Neste exemplo em específico, foram criadas duas conexões (executando o comando abaixo em duas abas distintas do termninal)

telnet localhost <#porta>



```
naomitkm)  [~/Projects/MC833/exercise5/exercise5_code/src] (main)$ telnet 127.0.0.1 5000
Trying 127.0.0.1...
Connected to localhost.
```

Nos screenshots abaixo são mostradas a saída do servidor: os socket file descriptors associados a cada conexão telnet, IP e porta sendo utilizados.



```
Server listening on port: 5000
Waiting connection ...
[parent] Parent pid 57798
New connection, socket fd: 5, if: 127.0.0.1, port: 56566
Adding new_socket to list of sockets in position 0
New connection, socket fd: 6, if: 127.0.0.1, port: 56579
```

Implementação do Cliente

Assim como o servidor, o cliente utiliza a multiplexação de IO para possibilitar a comunicação não bloqueante:

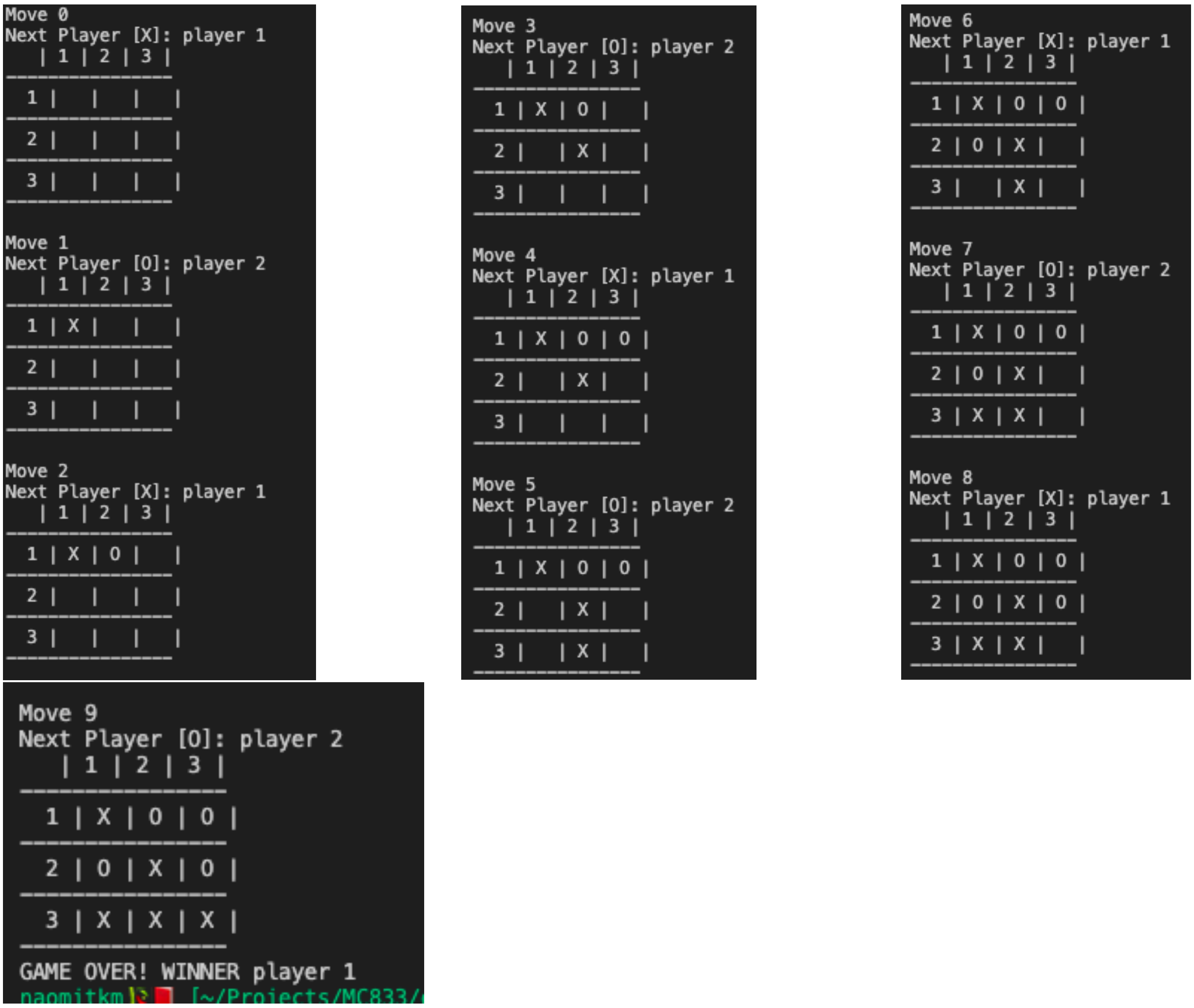
1. Com o servidor
2. Com o oponente

Conforme cliente recebia os resultados do servidor ou do oponente o comando select era utilizado para que o socket em condições de leitura pudesse ser acessado.

Exemplo de execução

Agora mostraremos um exemplo do funcionamento do programa. Algumas limitações:

- Durante os testes não foi possível terminar a execução de uma partida por inteiro, pois aparentemente em um dado momento os dados trafegando via UDP pareciam não chegar de um cliente a outro. Tratamento de perdas de dados não foram implementados.
- Os testes com a classe que cuida da lógica de jogo não detectaram problemas, sendo possível terminar normalmente uma partida (fora do contexto de comunicação UDP).
- O problema pode estar na lógica de estados do cliente, mas não foi possível confirmar esta hipótese até o presente momento.



```
Move 0
Next Player [X]: player 1
    | 1 | 2 | 3 |
 ----------------
  1 |   |   |   |
 ----------------
  2 |   |   |   |
 ----------------
  3 |   |   |   |
 ----------------

Move 1
Next Player [0]: player 2
    | 1 | 2 | 3 |
 ----------------
  1 | X |   |   |
 ----------------
  2 |   |   |   |
 ----------------
  3 |   |   |   |
 ----------------

Move 2
Next Player [X]: player 1
    | 1 | 2 | 3 |
 ----------------
  1 | X | 0 |   |
 ----------------
  2 |   |   |   |
 ----------------
  3 |   |   |   |
 ----------------
```

```
Move 3
Next Player [0]: player 2
    | 1 | 2 | 3 |
 ----------------
  1 | X | 0 |   |
 ----------------
  2 |   | X |   |
 ----------------
  3 |   |   |   |
 ----------------

Move 4
Next Player [X]: player 1
    | 1 | 2 | 3 |
 ----------------
  1 | X | 0 | 0 |
 ----------------
  2 |   | X |   |
 ----------------
  3 |   |   |   |
 ----------------

Move 5
Next Player [0]: player 2
    | 1 | 2 | 3 |
 ----------------
  1 | X | 0 | 0 |
 ----------------
  2 |   | X |   |
 ----------------
  3 |   | X |   |
 ----------------
```

```
Move 6
Next Player [X]: player 1
    | 1 | 2 | 3 |
 ----------------
  1 | X | 0 | 0 |
 ----------------
  2 | 0 | X |   |
 ----------------
  3 |   | X |   |
 ----------------

Move 7
Next Player [0]: player 2
    | 1 | 2 | 3 |
 ----------------
  1 | X | 0 | 0 |
 ----------------
  2 | 0 | X |   |
 ----------------
  3 | X | X |   |
 ----------------

Move 8
Next Player [X]: player 1
    | 1 | 2 | 3 |
 ----------------
  1 | X | 0 | 0 |
 ----------------
  2 | 0 | X | 0 |
 ----------------
  3 | X | X |   |
 ----------------
```

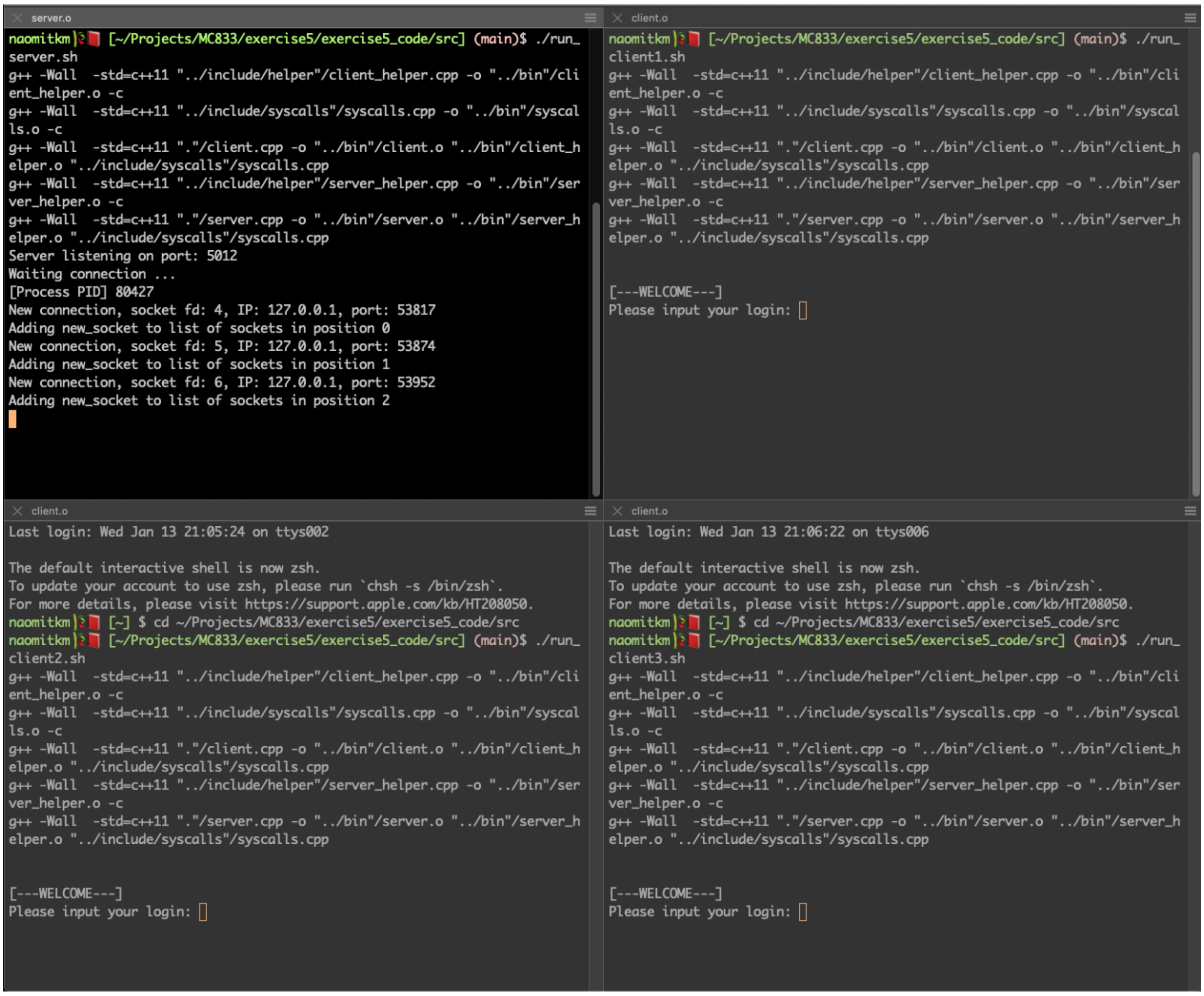
```
Move 9
Next Player [0]: player 2
    | 1 | 2 | 3 |
 ----------------
  1 | X | 0 | 0 |
 ----------------
  2 | 0 | X | 0 |
 ----------------
  3 | X | X | X |
 ----------------
GAME OVER! WINNER player 1
naomitkm  [~/Projects/MC833/
```

Figura: testes com a lógica do tabuleiro.

Procedimentos

2. Iniciar o servidor e os clientes: rodando os scripts, dentro da pasta src
    a. run_server.sh
    b. run_client1.sh
    c. run_client2.sh
    d. run_client3.sh

Da esquerda para direita, de cima para baixo, na figura a seguir, temos as saídas do servidor e dos clientes 1,2 e3; Os clientes devem fazer login, isto é, enviar o seu id. No caso os ids dos clientes 1,2 e3 são ONE, TWO e THREE.



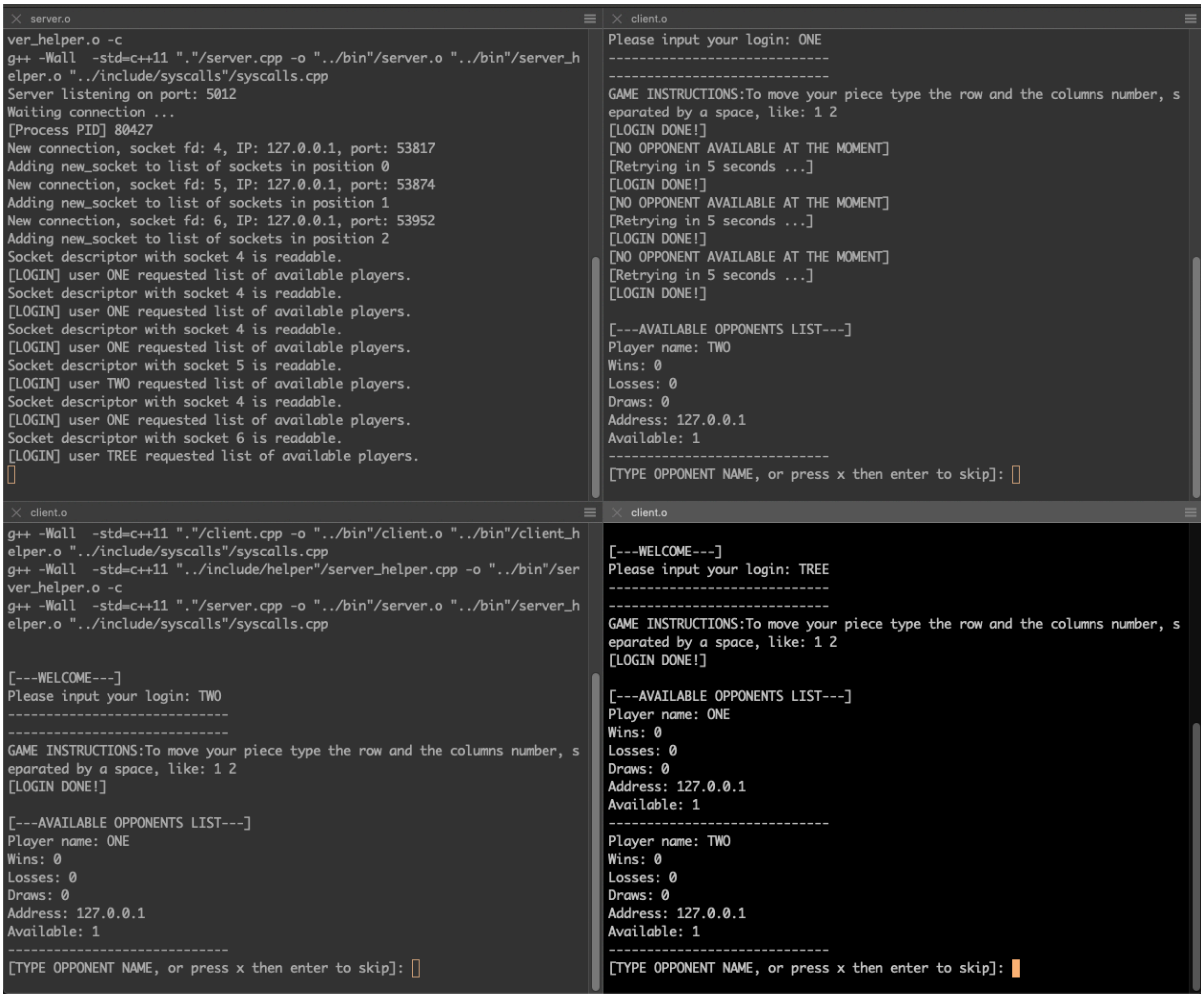
```
naomitkm [~/Projects/MC833/exercise5/exercise5_code/src] (main)$ ./run_
server.sh
g++ -Wall  -std=c++11 "../include/helper"/client_helper.cpp -o "../bin"/cli
ent_helper.o -c
g++ -Wall  -std=c++11 "../include/syscalls"/syscalls.cpp -o "../bin"/syscal
ls.o -c
g++ -Wall  -std=c++11 "."/client.cpp -o "../bin"/client.o "../bin"/client_h
elper.o "../include/syscalls"/syscalls.cpp
g++ -Wall  -std=c++11 "../include/helper"/server_helper.cpp -o "../bin"/ser
ver_helper.o -c
g++ -Wall  -std=c++11 "."/server.cpp -o "../bin"/server.o "../bin"/server_h
elper.o "../include/syscalls"/syscalls.cpp
Server listening on port: 5012
Waiting connection ...
[Process PID] 80427
New connection, socket fd: 4, IP: 127.0.0.1, port: 53817
Adding new_socket to list of sockets in position 0
New connection, socket fd: 5, IP: 127.0.0.1, port: 53874
Adding new_socket to list of sockets in position 1
New connection, socket fd: 6, IP: 127.0.0.1, port: 53952
Adding new_socket to list of sockets in position 2
```

```
naomitkm [~/Projects/MC833/exercise5/exercise5_code/src] (main)$ ./run_
client1.sh
g++ -Wall  -std=c++11 "../include/helper"/client_helper.cpp -o "../bin"/cli
ent_helper.o -c
g++ -Wall  -std=c++11 "../include/syscalls"/syscalls.cpp -o "../bin"/syscal
ls.o -c
g++ -Wall  -std=c++11 "."/client.cpp -o "../bin"/client.o "../bin"/client_h
elper.o "../include/syscalls"/syscalls.cpp
g++ -Wall  -std=c++11 "../include/helper"/server_helper.cpp -o "../bin"/ser
ver_helper.o -c
g++ -Wall  -std=c++11 "."/server.cpp -o "../bin"/server.o "../bin"/server_h
elper.o "../include/syscalls"/syscalls.cpp


[---WELCOME---]
Please input your login:
```

```
Last login: Wed Jan 13 21:05:24 on ttys002

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
naomitkm [~] $ cd ~/Projects/MC833/exercise5/exercise5_code/src
naomitkm [~/Projects/MC833/exercise5/exercise5_code/src] (main)$ ./run_
client2.sh
g++ -Wall  -std=c++11 "../include/helper"/client_helper.cpp -o "../bin"/cli
ent_helper.o -c
g++ -Wall  -std=c++11 "../include/syscalls"/syscalls.cpp -o "../bin"/syscal
ls.o -c
g++ -Wall  -std=c++11 "."/client.cpp -o "../bin"/client.o "../bin"/client_h
elper.o "../include/syscalls"/syscalls.cpp
g++ -Wall  -std=c++11 "../include/helper"/server_helper.cpp -o "../bin"/ser
ver_helper.o -c
g++ -Wall  -std=c++11 "."/server.cpp -o "../bin"/server.o "../bin"/server_h
elper.o "../include/syscalls"/syscalls.cpp


[---WELCOME---]
Please input your login:
```

```
Last login: Wed Jan 13 21:06:22 on ttys006

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
naomitkm [~] $ cd ~/Projects/MC833/exercise5/exercise5_code/src
naomitkm [~/Projects/MC833/exercise5/exercise5_code/src] (main)$ ./run_
client3.sh
g++ -Wall  -std=c++11 "../include/helper"/client_helper.cpp -o "../bin"/cli
ent_helper.o -c
g++ -Wall  -std=c++11 "../include/syscalls"/syscalls.cpp -o "../bin"/syscal
ls.o -c
g++ -Wall  -std=c++11 "."/client.cpp -o "../bin"/client.o "../bin"/client_h
elper.o "../include/syscalls"/syscalls.cpp
g++ -Wall  -std=c++11 "../include/helper"/server_helper.cpp -o "../bin"/ser
ver_helper.o -c
g++ -Wall  -std=c++11 "."/server.cpp -o "../bin"/server.o "../bin"/server_h
elper.o "../include/syscalls"/syscalls.cpp


[---WELCOME---]
Please input your login:
```

3. Cliente 1, 2 e 3 fazem login. O, quando o cliente 1 o faz, não há jogadores disponíveis então o cliente espera 5 segundos para fazer uma nova requisição da lista de jogadores para o servidor. Quando 2 o faz, apenas o jogador 1 está disponível. E quando 3 o faz, 1 e 2 estão disponíveis. Junto aos dados dos jogadores disponíveis são mostrados os scores de cada um (vitórias, derrotas e empates).



server.o

```
ver_helper.o -c
g++ -Wall  -std=c++11 "."/server.cpp -o "../bin"/server.o "../bin"/server_h
elper.o "../include/syscalls"/syscalls.cpp
Server listening on port: 5012
Waiting connection ...
[Process PID] 80427
New connection, socket fd: 4, IP: 127.0.0.1, port: 53817
Adding new_socket to list of sockets in position 0
New connection, socket fd: 5, IP: 127.0.0.1, port: 53874
Adding new_socket to list of sockets in position 1
New connection, socket fd: 6, IP: 127.0.0.1, port: 53952
Adding new_socket to list of sockets in position 2
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 5 is readable.
[LOGIN] user TWO requested list of available players.
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 6 is readable.
[LOGIN] user TREE requested list of available players.
```

client.o

```
Please input your login: ONE
----------------------------
----------------------------
GAME INSTRUCTIONS:To move your piece type the row and the columns number, s
eparated by a space, like: 1 2
[LOGIN DONE!]
[NO OPPONENT AVAILABLE AT THE MOMENT]
[Retrying in 5 seconds ...]
[LOGIN DONE!]
[NO OPPONENT AVAILABLE AT THE MOMENT]
[Retrying in 5 seconds ...]
[LOGIN DONE!]
[NO OPPONENT AVAILABLE AT THE MOMENT]
[Retrying in 5 seconds ...]
[LOGIN DONE!]

[---AVAILABLE OPPONENTS LIST---]
Player name: TWO
Wins: 0
Losses: 0
Draws: 0
Address: 127.0.0.1
Available: 1
----------------------------
[TYPE OPPONENT NAME, or press x then enter to skip]:
```

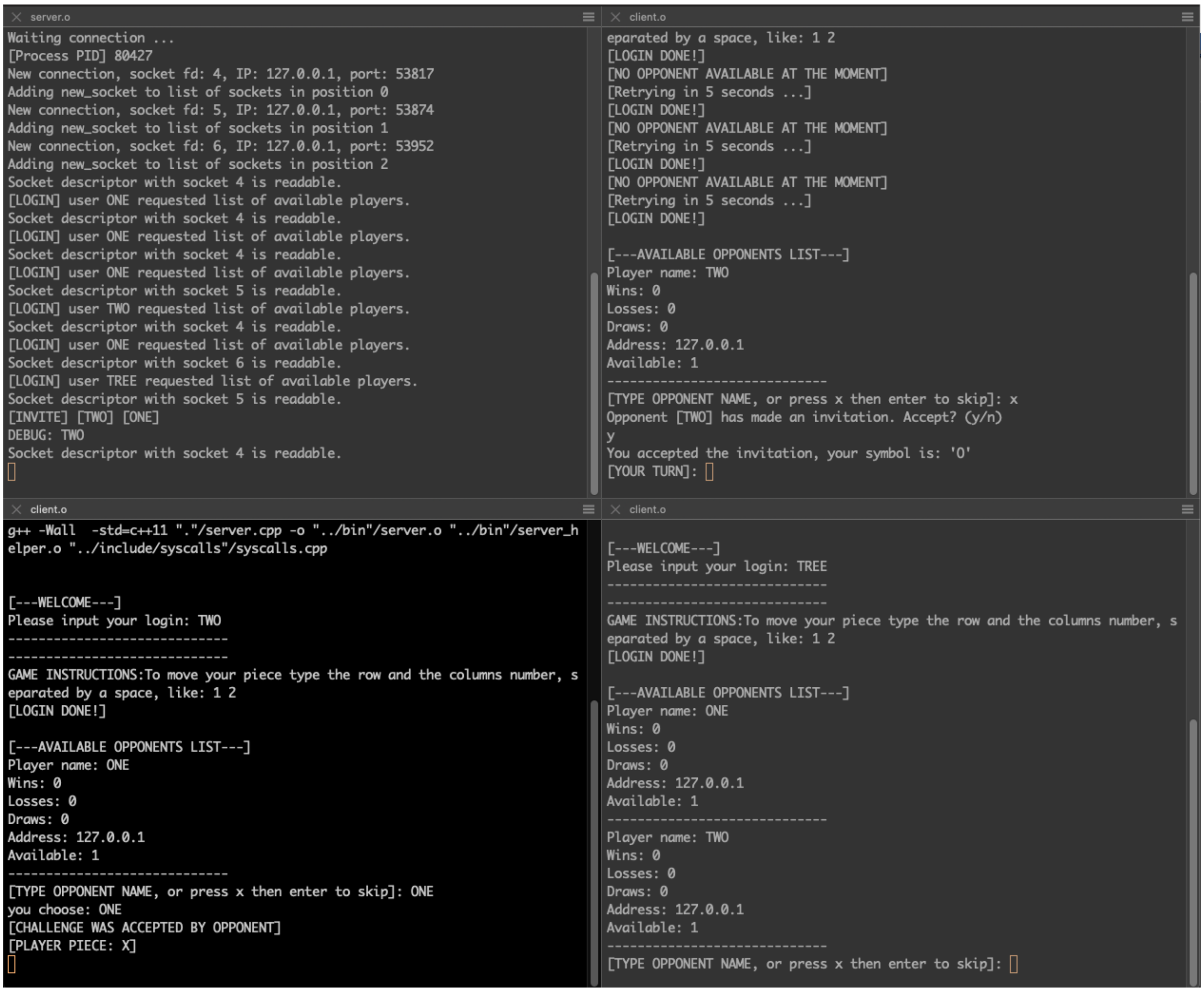
client.o

```
g++ -Wall  -std=c++11 "."/client.cpp -o "../bin"/client.o "../bin"/client_h
elper.o "../include/syscalls"/syscalls.cpp
g++ -Wall  -std=c++11 "../include/helper"/server_helper.cpp -o "../bin"/ser
ver_helper.o -c
g++ -Wall  -std=c++11 "."/server.cpp -o "../bin"/server.o "../bin"/server_h
elper.o "../include/syscalls"/syscalls.cpp


[---WELCOME---]
Please input your login: TWO
----------------------------
----------------------------
GAME INSTRUCTIONS:To move your piece type the row and the columns number, s
eparated by a space, like: 1 2
[LOGIN DONE!]

[---AVAILABLE OPPONENTS LIST---]
Player name: ONE
Wins: 0
Losses: 0
Draws: 0
Address: 127.0.0.1
Available: 1
----------------------------
[TYPE OPPONENT NAME, or press x then enter to skip]:
```

client.o

```

[---WELCOME---]
Please input your login: TREE
----------------------------
----------------------------
GAME INSTRUCTIONS:To move your piece type the row and the columns number, s
eparated by a space, like: 1 2
[LOGIN DONE!]

[---AVAILABLE OPPONENTS LIST---]
Player name: ONE
Wins: 0
Losses: 0
Draws: 0
Address: 127.0.0.1
Available: 1
----------------------------
Player name: TWO
Wins: 0
Losses: 0
Draws: 0
Address: 127.0.0.1
Available: 1
----------------------------
[TYPE OPPONENT NAME, or press x then enter to skip]:
```

4. O cliente 2 (TWO) decide convidar o cliente 1 (ONE). O cliente 1 decide não convidar ninguém e aperta ‘x’ para pular esse passo. O cliente 1 recebe o convite, intermediado pelo servidor, e o jogo se inicia.



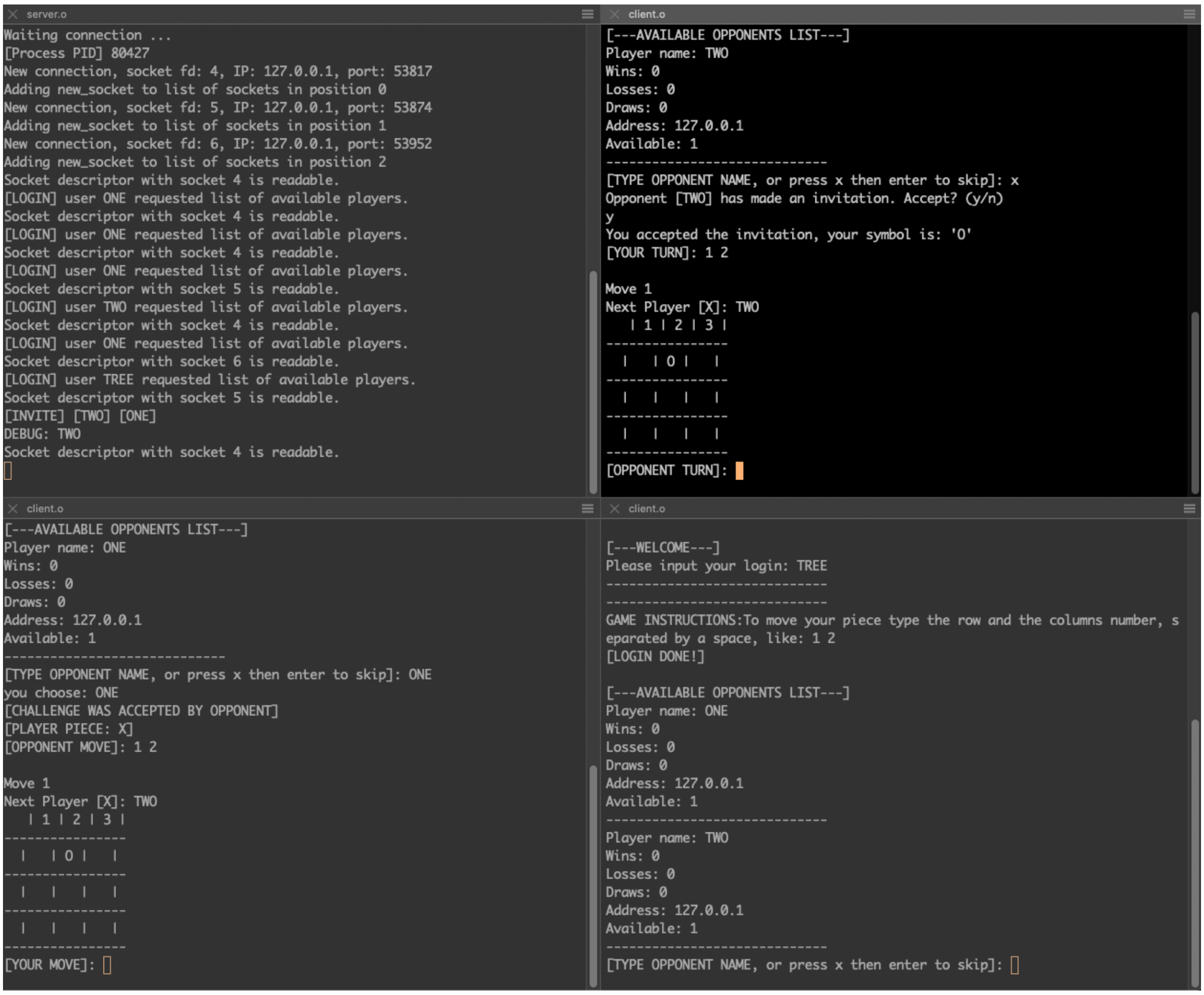
```
Waiting connection ...
[Process PID] 80427
New connection, socket fd: 4, IP: 127.0.0.1, port: 53817
Adding new_socket to list of sockets in position 0
New connection, socket fd: 5, IP: 127.0.0.1, port: 53874
Adding new_socket to list of sockets in position 1
New connection, socket fd: 6, IP: 127.0.0.1, port: 53952
Adding new_socket to list of sockets in position 2
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 5 is readable.
[LOGIN] user TWO requested list of available players.
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 6 is readable.
[LOGIN] user TREE requested list of available players.
Socket descriptor with socket 5 is readable.
[INVITE] [TWO] [ONE]
DEBUG: TWO
Socket descriptor with socket 4 is readable.
```

```
eparated by a space, like: 1 2
[LOGIN DONE!]
[NO OPPONENT AVAILABLE AT THE MOMENT]
[Retrying in 5 seconds ...]
[LOGIN DONE!]
[NO OPPONENT AVAILABLE AT THE MOMENT]
[Retrying in 5 seconds ...]
[LOGIN DONE!]
[NO OPPONENT AVAILABLE AT THE MOMENT]
[Retrying in 5 seconds ...]
[LOGIN DONE!]

[---AVAILABLE OPPONENTS LIST---]
Player name: TWO
Wins: 0
Losses: 0
Draws: 0
Address: 127.0.0.1
Available: 1
----------------------------
[TYPE OPPONENT NAME, or press x then enter to skip]: x
Opponent [TWO] has made an invitation. Accept? (y/n)
y
You accepted the invitation, your symbol is: 'O'
[YOUR TURN]:
```

```
g++ -Wall  -std=c++11 "."/server.cpp -o "../bin"/server.o "../bin"/server_h
elper.o "../include/syscalls"/syscalls.cpp


[---WELCOME---]
Please input your login: TWO
----------------------------
----------------------------
GAME INSTRUCTIONS:To move your piece type the row and the columns number, s
eparated by a space, like: 1 2
[LOGIN DONE!]

[---AVAILABLE OPPONENTS LIST---]
Player name: ONE
Wins: 0
Losses: 0
Draws: 0
Address: 127.0.0.1
Available: 1
----------------------------
[TYPE OPPONENT NAME, or press x then enter to skip]: ONE
you choose: ONE
[CHALLENGE WAS ACCEPTED BY OPPONENT]
[PLAYER PIECE: X]
```

```

[---WELCOME---]
Please input your login: TREE
----------------------------
----------------------------
GAME INSTRUCTIONS:To move your piece type the row and the columns number, s
eparated by a space, like: 1 2
[LOGIN DONE!]

[---AVAILABLE OPPONENTS LIST---]
Player name: ONE
Wins: 0
Losses: 0
Draws: 0
Address: 127.0.0.1
Available: 1
----------------------------
Player name: TWO
Wins: 0
Losses: 0
Draws: 0
Address: 127.0.0.1
Available: 1
----------------------------
[TYPE OPPONENT NAME, or press x then enter to skip]:
```

5. O cliente 1 associado ao símbolo “O” faz a primeira jogada (linha 1, coluna 2).



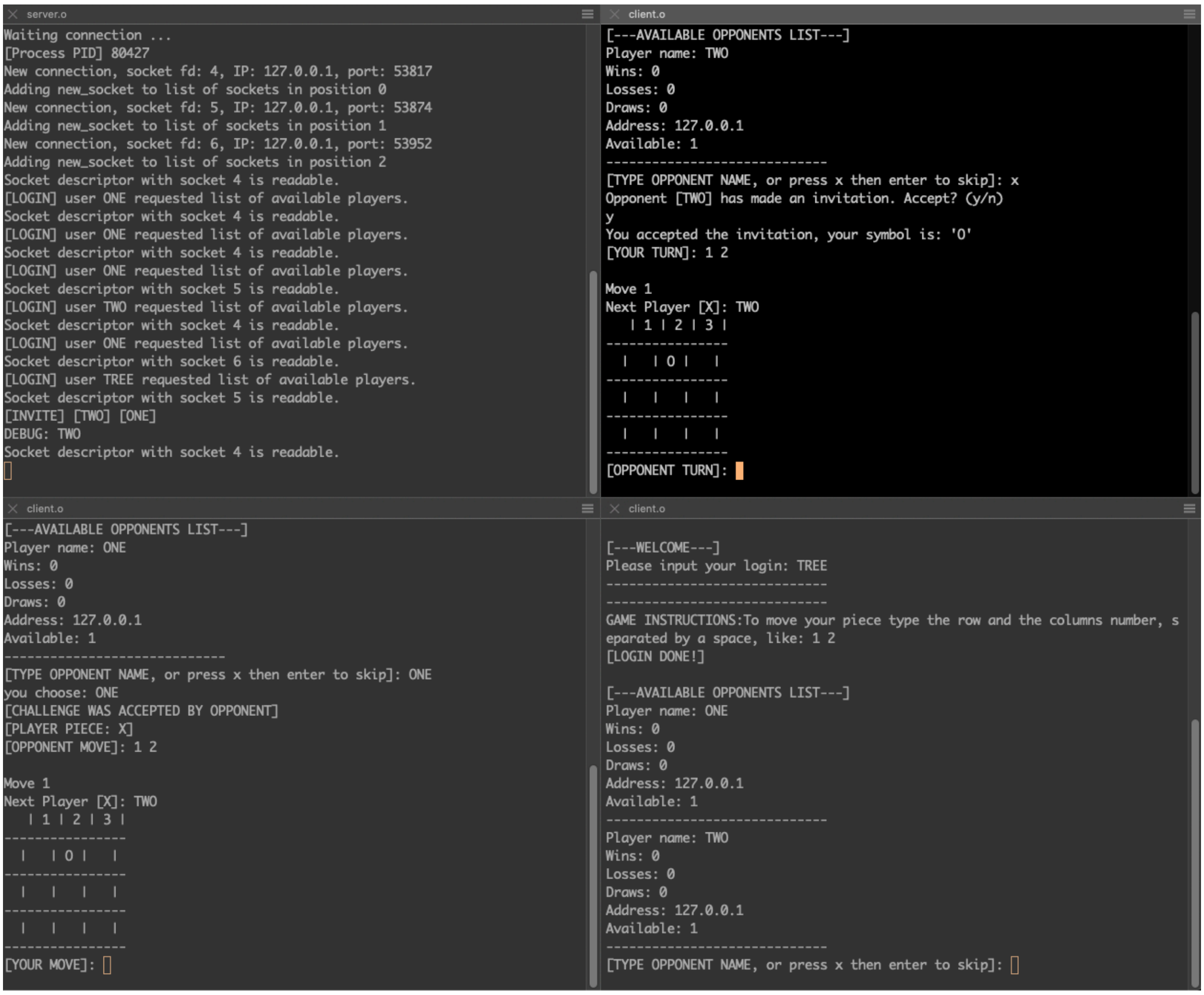
```
Waiting connection ...
[Process PID] 80427
New connection, socket fd: 4, IP: 127.0.0.1, port: 53817
Adding new_socket to list of sockets in position 0
New connection, socket fd: 5, IP: 127.0.0.1, port: 53874
Adding new_socket to list of sockets in position 1
New connection, socket fd: 6, IP: 127.0.0.1, port: 53952
Adding new_socket to list of sockets in position 2
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 5 is readable.
[LOGIN] user TWO requested list of available players.
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 6 is readable.
[LOGIN] user TREE requested list of available players.
Socket descriptor with socket 5 is readable.
[INVITE] [TWO] [ONE]
DEBUG: TWO
Socket descriptor with socket 4 is readable.
```

```
[---AVAILABLE OPPONENTS LIST---]
Player name: TWO
Wins: 0
Losses: 0
Draws: 0
Address: 127.0.0.1
Available: 1
-----------------------------
[TYPE OPPONENT NAME, or press x then enter to skip]: x
Opponent [TWO] has made an invitation. Accept? (y/n)
y
You accepted the invitation, your symbol is: 'O'
[YOUR TURN]: 1 2

Move 1
Next Player [X]: TWO
   | 1 | 2 | 3 |
----------------
  |   | O |   |
----------------
  |   |   |   |
----------------
  |   |   |   |
----------------
[OPPONENT TURN]:
```

```
[---AVAILABLE OPPONENTS LIST---]
Player name: ONE
Wins: 0
Losses: 0
Draws: 0
Address: 127.0.0.1
Available: 1
-----------------------------
[TYPE OPPONENT NAME, or press x then enter to skip]: ONE
you choose: ONE
[CHALLENGE WAS ACCEPTED BY OPPONENT]
[PLAYER PIECE: X]
[OPPONENT MOVE]: 1 2

Move 1
Next Player [X]: TWO
   | 1 | 2 | 3 |
----------------
  |   | O |   |
----------------
  |   |   |   |
----------------
  |   |   |   |
----------------
[YOUR MOVE]:
```

```

[---WELCOME---]
Please input your login: TREE
-----------------------------
-----------------------------
GAME INSTRUCTIONS:To move your piece type the row and the columns number, s
eparated by a space, like: 1 2
[LOGIN DONE!]

[---AVAILABLE OPPONENTS LIST---]
Player name: ONE
Wins: 0
Losses: 0
Draws: 0
Address: 127.0.0.1
Available: 1
-----------------------------
Player name: TWO
Wins: 0
Losses: 0
Draws: 0
Address: 127.0.0.1
Available: 1
-----------------------------
[TYPE OPPONENT NAME, or press x then enter to skip]:
```

6. O cliente 2 (símbolo “X”) faz a jogada linha 1 coluna 1



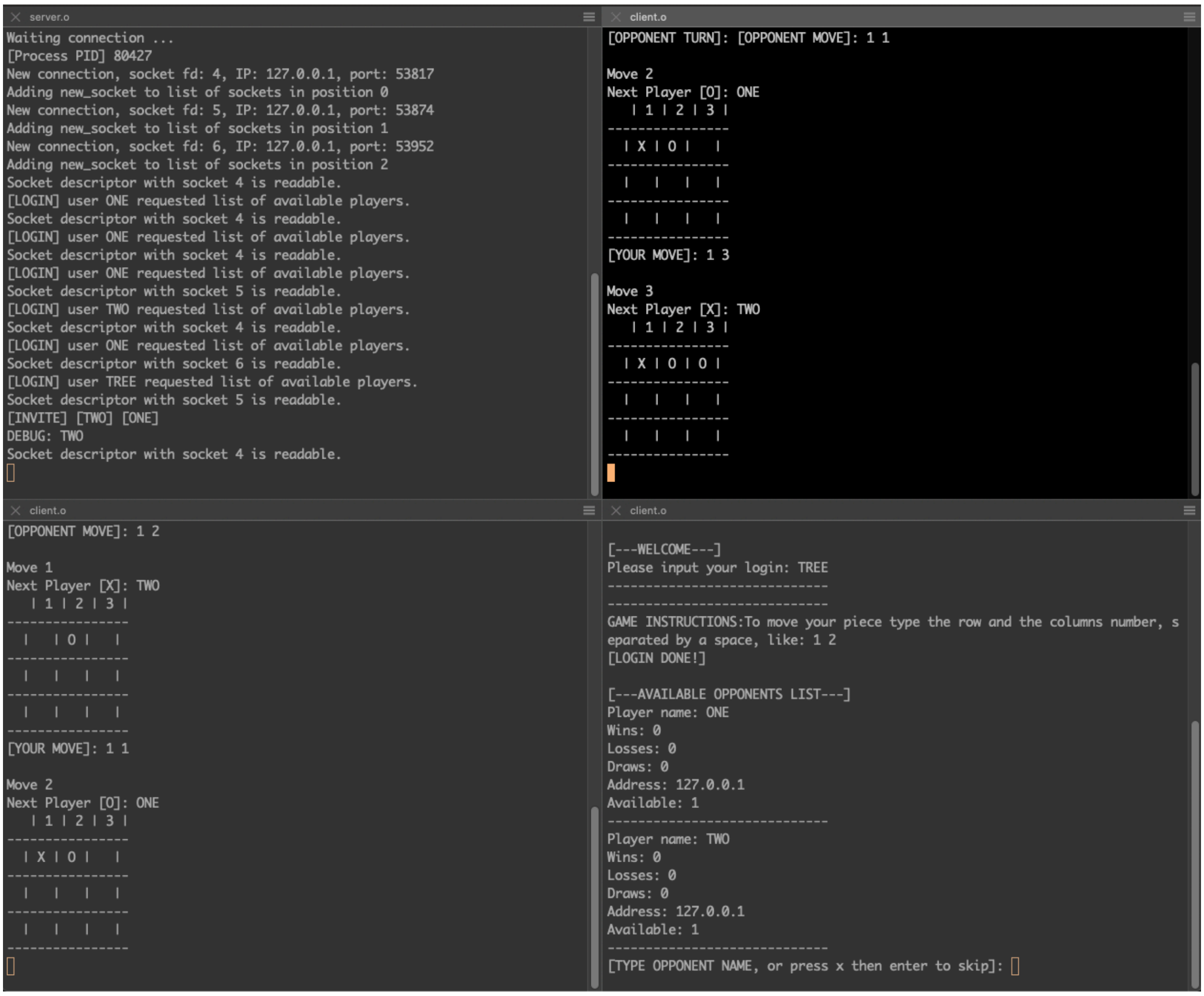
```
Waiting connection ...
[Process PID] 80427
New connection, socket fd: 4, IP: 127.0.0.1, port: 53817
Adding new_socket to list of sockets in position 0
New connection, socket fd: 5, IP: 127.0.0.1, port: 53874
Adding new_socket to list of sockets in position 1
New connection, socket fd: 6, IP: 127.0.0.1, port: 53952
Adding new_socket to list of sockets in position 2
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 5 is readable.
[LOGIN] user TWO requested list of available players.
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 6 is readable.
[LOGIN] user TREE requested list of available players.
Socket descriptor with socket 5 is readable.
[INVITE] [TWO] [ONE]
DEBUG: TWO
Socket descriptor with socket 4 is readable.
```

```
[---AVAILABLE OPPONENTS LIST---]
Player name: TWO
Wins: 0
Losses: 0
Draws: 0
Address: 127.0.0.1
Available: 1
-----------------------------
[TYPE OPPONENT NAME, or press x then enter to skip]: x
Opponent [TWO] has made an invitation. Accept? (y/n)
y
You accepted the invitation, your symbol is: 'O'
[YOUR TURN]: 1 2

Move 1
Next Player [X]: TWO
   | 1 | 2 | 3 |
----------------
  |   | O |   |
----------------
  |   |   |   |
----------------
  |   |   |   |
----------------
[OPPONENT TURN]:
```

```
[---AVAILABLE OPPONENTS LIST---]
Player name: ONE
Wins: 0
Losses: 0
Draws: 0
Address: 127.0.0.1
Available: 1
-----------------------------
[TYPE OPPONENT NAME, or press x then enter to skip]: ONE
you choose: ONE
[CHALLENGE WAS ACCEPTED BY OPPONENT]
[PLAYER PIECE: X]
[OPPONENT MOVE]: 1 2

Move 1
Next Player [X]: TWO
   | 1 | 2 | 3 |
----------------
  |   | O |   |
----------------
  |   |   |   |
----------------
  |   |   |   |
----------------
[YOUR MOVE]:
```

```

[---WELCOME---]
Please input your login: TREE
-----------------------------
-----------------------------
GAME INSTRUCTIONS:To move your piece type the row and the columns number, s
eparated by a space, like: 1 2
[LOGIN DONE!]

[---AVAILABLE OPPONENTS LIST---]
Player name: ONE
Wins: 0
Losses: 0
Draws: 0
Address: 127.0.0.1
Available: 1
-----------------------------
Player name: TWO
Wins: 0
Losses: 0
Draws: 0
Address: 127.0.0.1
Available: 1
-----------------------------
[TYPE OPPONENT NAME, or press x then enter to skip]:
```

7. O cliente 1 (“O”) move linha 1 coluna 3



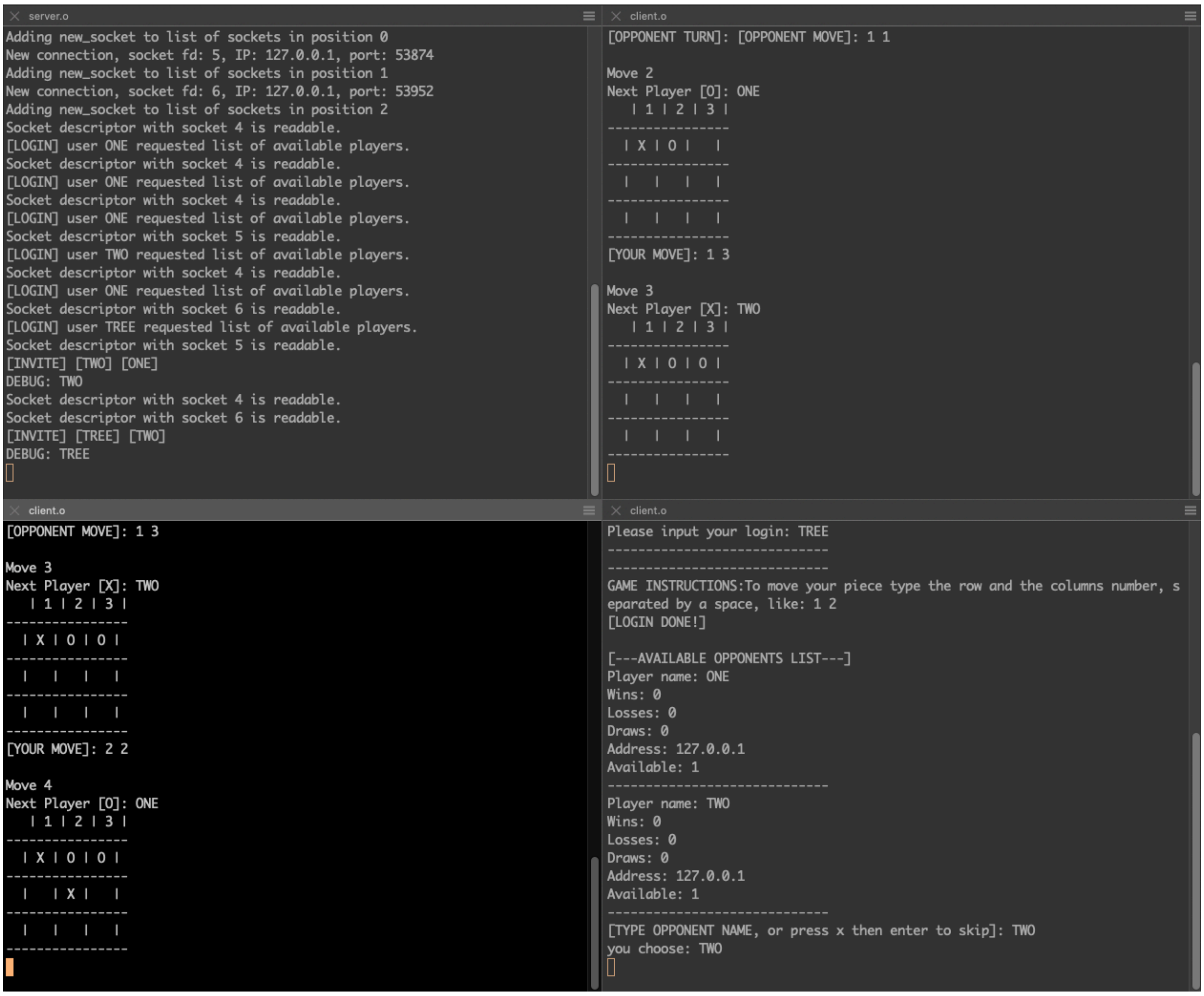
```
Waiting connection ...
[Process PID] 80427
New connection, socket fd: 4, IP: 127.0.0.1, port: 53817
Adding new_socket to list of sockets in position 0
New connection, socket fd: 5, IP: 127.0.0.1, port: 53874
Adding new_socket to list of sockets in position 1
New connection, socket fd: 6, IP: 127.0.0.1, port: 53952
Adding new_socket to list of sockets in position 2
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 5 is readable.
[LOGIN] user TWO requested list of available players.
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 6 is readable.
[LOGIN] user TREE requested list of available players.
Socket descriptor with socket 5 is readable.
[INVITE] [TWO] [ONE]
DEBUG: TWO
Socket descriptor with socket 4 is readable.
```

```
[OPPONENT TURN]: [OPPONENT MOVE]: 1 1

Move 2
Next Player [O]: ONE
   | 1 | 2 | 3 |
----------------
  | X | O |   |
----------------
  |   |   |   |
----------------
  |   |   |   |
----------------
[YOUR MOVE]: 1 3

Move 3
Next Player [X]: TWO
   | 1 | 2 | 3 |
----------------
  | X | O | O |
----------------
  |   |   |   |
----------------
  |   |   |   |
----------------
```

```
[OPPONENT MOVE]: 1 2

Move 1
Next Player [X]: TWO
   | 1 | 2 | 3 |
----------------
  |   | O |   |
----------------
  |   |   |   |
----------------
  |   |   |   |
----------------
[YOUR MOVE]: 1 1

Move 2
Next Player [O]: ONE
   | 1 | 2 | 3 |
----------------
  | X | O |   |
----------------
  |   |   |   |
----------------
  |   |   |   |
----------------
```

```

[---WELCOME---]
Please input your login: TREE
-----------------------------
-----------------------------
GAME INSTRUCTIONS:To move your piece type the row and the columns number, s
eparated by a space, like: 1 2
[LOGIN DONE!]

[---AVAILABLE OPPONENTS LIST---]
Player name: ONE
Wins: 0
Losses: 0
Draws: 0
Address: 127.0.0.1
Available: 1
-----------------------------
Player name: TWO
Wins: 0
Losses: 0
Draws: 0
Address: 127.0.0.1
Available: 1
-----------------------------
[TYPE OPPONENT NAME, or press x then enter to skip]:
```

8. O cliente 2 (“X”) move linha 2 coluna 2, o cliente 3 manda um convite para o cliente 2, por intermédio do servidor.



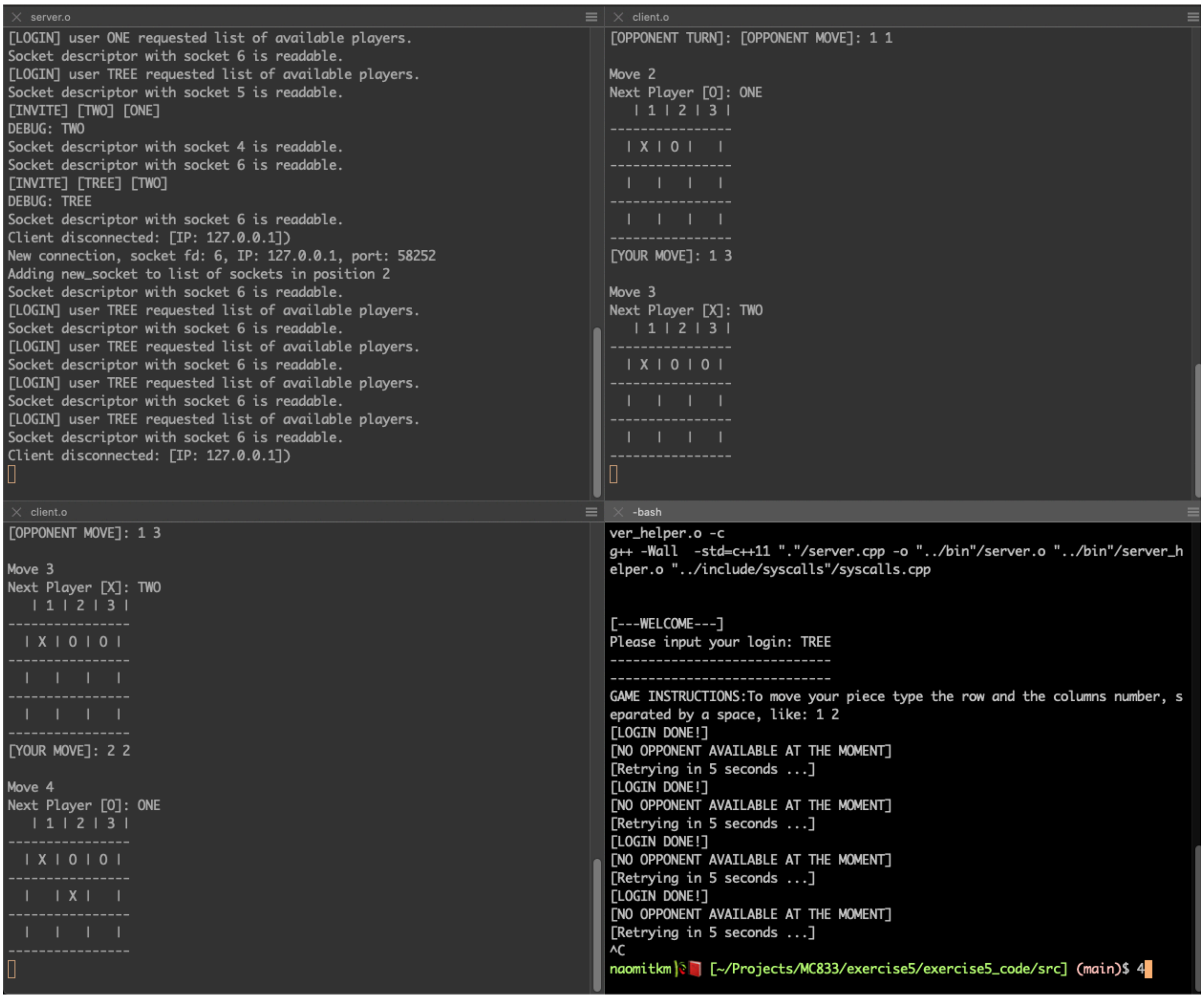
```
Adding new_socket to list of sockets in position 0
New connection, socket fd: 5, IP: 127.0.0.1, port: 53874
Adding new_socket to list of sockets in position 1
New connection, socket fd: 6, IP: 127.0.0.1, port: 53952
Adding new_socket to list of sockets in position 2
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 5 is readable.
[LOGIN] user TWO requested list of available players.
Socket descriptor with socket 4 is readable.
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 6 is readable.
[LOGIN] user TREE requested list of available players.
Socket descriptor with socket 5 is readable.
[INVITE] [TWO] [ONE]
DEBUG: TWO
Socket descriptor with socket 4 is readable.
Socket descriptor with socket 6 is readable.
[INVITE] [TREE] [TWO]
DEBUG: TREE
```

```
[OPPONENT TURN]: [OPPONENT MOVE]: 1 1

Move 2
Next Player [O]: ONE
   | 1 | 2 | 3 |
----------------
  | X | O |   |
----------------
  |   |   |   |
----------------
  |   |   |   |
----------------
[YOUR MOVE]: 1 3

Move 3
Next Player [X]: TWO
   | 1 | 2 | 3 |
----------------
  | X | O | O |
----------------
  |   |   |   |
----------------
  |   |   |   |
----------------
```

```
[OPPONENT MOVE]: 1 3

Move 3
Next Player [X]: TWO
   | 1 | 2 | 3 |
----------------
  | X | O | O |
----------------
  |   |   |   |
----------------
  |   |   |   |
----------------
[YOUR MOVE]: 2 2

Move 4
Next Player [O]: ONE
   | 1 | 2 | 3 |
----------------
  | X | O | O |
----------------
  |   | X |   |
----------------
  |   |   |   |
----------------
```

```
Please input your login: TREE
-----------------------------
-----------------------------
GAME INSTRUCTIONS:To move your piece type the row and the columns number, s
eparated by a space, like: 1 2
[LOGIN DONE!]

[---AVAILABLE OPPONENTS LIST---]
Player name: ONE
Wins: 0
Losses: 0
Draws: 0
Address: 127.0.0.1
Available: 1
-----------------------------
Player name: TWO
Wins: 0
Losses: 0
Draws: 0
Address: 127.0.0.1
Available: 1
-----------------------------
[TYPE OPPONENT NAME, or press x then enter to skip]: TWO
you choose: TWO
```

9. O cliente 3 se desconecta e conecta novamente, nenhum cliente está disponível para o jogo então de 5 em 5 segundos ele pede ao servidor para mandar a lista de jogadores disponíveis.



```
[LOGIN] user ONE requested list of available players.
Socket descriptor with socket 6 is readable.
[LOGIN] user TREE requested list of available players.
Socket descriptor with socket 5 is readable.
[INVITE] [TWO] [ONE]
DEBUG: TWO
Socket descriptor with socket 4 is readable.
Socket descriptor with socket 6 is readable.
[INVITE] [TREE] [TWO]
DEBUG: TREE
Socket descriptor with socket 6 is readable.
Client disconnected: [IP: 127.0.0.1])
New connection, socket fd: 6, IP: 127.0.0.1, port: 58252
Adding new_socket to list of sockets in position 2
Socket descriptor with socket 6 is readable.
[LOGIN] user TREE requested list of available players.
Socket descriptor with socket 6 is readable.
[LOGIN] user TREE requested list of available players.
Socket descriptor with socket 6 is readable.
[LOGIN] user TREE requested list of available players.
Socket descriptor with socket 6 is readable.
[LOGIN] user TREE requested list of available players.
Socket descriptor with socket 6 is readable.
Client disconnected: [IP: 127.0.0.1])
```

```
[OPPONENT TURN]: [OPPONENT MOVE]: 1 1

Move 2
Next Player [O]: ONE
    | 1 | 2 | 3 |
----------------
  | X | O |   |
----------------
  |   |   |   |
----------------
  |   |   |   |
----------------
[YOUR MOVE]: 1 3

Move 3
Next Player [X]: TWO
    | 1 | 2 | 3 |
----------------
  | X | O | O |
----------------
  |   |   |   |
----------------
  |   |   |   |
----------------
```

```
[OPPONENT MOVE]: 1 3

Move 3
Next Player [X]: TWO
    | 1 | 2 | 3 |
----------------
  | X | O | O |
----------------
  |   |   |   |
----------------
  |   |   |   |
----------------
[YOUR MOVE]: 2 2

Move 4
Next Player [O]: ONE
    | 1 | 2 | 3 |
----------------
  | X | O | O |
----------------
  |   | X |   |
----------------
  |   |   |   |
----------------
```

```
ver_helper.o -c
g++ -Wall  -std=c++11 "."/server.cpp -o "../bin"/server.o "../bin"/server_h
elper.o "../include/syscalls"/syscalls.cpp


[---WELCOME---]
Please input your login: TREE
----------------------------
----------------------------
GAME INSTRUCTIONS:To move your piece type the row and the columns number, s
eparated by a space, like: 1 2
[LOGIN DONE!]
[NO OPPONENT AVAILABLE AT THE MOMENT]
[Retrying in 5 seconds ...]
[LOGIN DONE!]
[NO OPPONENT AVAILABLE AT THE MOMENT]
[Retrying in 5 seconds ...]
[LOGIN DONE!]
[NO OPPONENT AVAILABLE AT THE MOMENT]
[Retrying in 5 seconds ...]
[LOGIN DONE!]
[NO OPPONENT AVAILABLE AT THE MOMENT]
[Retrying in 5 seconds ...]
^C
naomitkm| [~/Projects/MC833/exercise5/exercise5_code/src] (main)$ 4
```

Não foi possível dar prosseguimento `a partida além deste ponto por conta do problema mencionado anteriormente.

Referências

[1] Unix Network Programming, Volume 1: The Sockets Networking API (3rd Edition). Disponível em https://www.amazon.com.br/Unix-Network-Programming-Sockets-Networking/dp/0131411551.

[2] Socket Programming in C/C++: Handling multiple clients on server without multi threading. Disponível em https://www.geeksforgeeks.org/socket-programming-in-cc-handling-multiple-clients-on-server-without-multi-threading/.

[3] UPD client server implementation. https://www.geeksforgeeks.org/udp-server-client-implementation-c/