

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.

RiddleQuest: The Treasure Hunt Interface

You are a digital adventurer, one of the brave navigators seeking to uncover the secrets of the Code Chest. Armed with your terminal, you connect to the Guardian of Riddles server to face its challenges. With each riddle solved, you get closer to discovering the location of the treasure. It can be assumed that to reach the trophy and receive its coordinates, three challenges must be solved. Once they are solved, you will receive the coordinates, and the server-client connection should be terminated.

Your objective is to implement a client in C that can connect to the challenge server, interact with it to solve riddles, and follow the hints to eventually discover the treasure.

To do so, you must first establish a connection to the Guardian Server and begin your adventure. To do so you'll input the IP and port as arguments to your program. You'll have to implement a navigation menu to guide you in your adventures.

When the program starts, you must print the message: "Welcome to RiddleQuest. Prepare to unlock the secrets and discover the treasure!"

Once this message is shown, you should ask for the user's name and send it to the server. Anytime you send something to the server, it should end with `\n`. So in this case you should send: "`<username>\n`". Responses from the server will also always be strings delimited with `\n` at the end. Print the replies from the server on screen.

Once you do this, you can go directly to the navigation menu. This menu must have:

- **Request Current Challenge:** Asks the server to send the current riddle, to start solving it. To do so, send "`1\n`" to the server. The server will reply with the current challenge with "`<response>\n`".
- **Send Response to Challenge:** Allows the user to solve the challenge provided in option 1, and have the server verify the response. To do so, send "`2\n`" to the server, followed by "`<response>\n`".
- **Request Hint:** To request a hint if you are unsure how to solve the challenge. To do so, send "`3\n`" to the server. The server will reply with a clue. You will find additional material to solve the clues below the examples.
- **View Current Mission Status:** To see how many challenges remain to be solved, and this option will also provide the coordinates of the chest, thereby ending the connection. To do so, send "`4\n`" to the server.
- **Terminate Connection and Exit:** To end the connection with the server. To do so, send "`5\n`" to the server. There will be no replies from the server in this option.

Unless the option to exit has been selected, the menu must be shown after each option again.

Once all challenges are completed, the server will reveal the treasure's coordinates, marking the end of the digital adventure with a victory.

```

||      Navigation Menu      ||
=====
1. Receive Current Challenge
2. Send Response to Challenge
3. Request Hint
4. View Current Mission Status
5. Terminate Connection and Exit
Select an option: 2
Enter your response to the challenge: 100
Incorrect answer, try again.

```

Example 1 - Challenge

Server:

```
  \____; ;____\
  | /          \
  |  ( )oo( )  |
  | \(%()*^^( )^ \
  |  |-%-----|
  |  \ | %  ))  |
  |  \ | %-----|
```

Welcome to the Guardian of Enigmas Server. Prepare to embark on a journey of puzzles and mysteries!

Welcome Cris!

Cris - request challenge...

Cris - sending answer...

Checking answer...

□

Client:

```
=====
||      Navigation Menu      ||
=====
```

1. Receive Current Challenge
2. Send Response to Challenge
3. Request Hint
4. View Current Mission Status
5. Terminate Connection and Exit

Select an option: 1

Challenge: Result of 123 + 234?

```
=====
||      Navigation Menu      ||
=====
```

1. Receive Current Challenge
2. Send Response to Challenge
3. Request Hint
4. View Current Mission Status
5. Terminate Connection and Exit

Select an option: 2

Enter your response to the challenge:

Correct answer! Proceeding to the next challenge.

Example 2 – View current mission Status and Request Hint

Server:

```

  ____  _
 /  _ \| | | |
| |_) | | | |
|  _ \| | | |
| |_) | | | |
|  _ \| | | |
|_| \_|_|_|_|

```

```

Welcome to the Guardian of Enigmas Server. Prepare to embark on a journey of puzzles and mysteries!
Welcome Cris!
Cris - request challenge...
Cris - sending answer...
Checking answer...
Cris - request to view current mission status...
Cris - request challenge...
Cris - request hint...
Hint sent!

```

Client:

```

=====
||      Navigation Menu      ||
=====
1. Receive Current Challenge
2. Send Response to Challenge
3. Request Hint
4. View Current Mission Status
5. Terminate Connection and Exit
Select an option: 4
Current mission status:
You are currently on challenge 2 out of 5.

=====
||      Navigation Menu      ||
=====
1. Receive Current Challenge
2. Send Response to Challenge
3. Request Hint
4. View Current Mission Status
5. Terminate Connection and Exit
Select an option: 1
Challenge: Decode this text: "Khoor Zruog"

=====
||      Navigation Menu      ||
=====
1. Receive Current Challenge
2. Send Response to Challenge
3. Request Hint
4. View Current Mission Status
5. Terminate Connection and Exit
Select an option: 3
Your hint is coming...
Use Caesar cipher with a shift of -1.

```

Example 3 – Finish Challenges

```
_____  
 \  _____  \  
  | /           \  
  | |           |  
  | | oo() .  
  | | (%)*^( ) \  
  | | -%-----|  
  | | %  ) )  
  | | %  
  | |
```

Welcome to the Guardian of Enigmas Server. Prepare to embark on a journey of puzzles and mysteries!

Welcome Cris!

Cris - request challenge...

Cris - sending answer...

Checking answer...

Cris - request to view current mission status...

Cris - request challenge...

Cris - request hint...

Hint sent!

Cris - request challenge...

Cris - sending answer...

Checking answer...

Cris - request to view current mission status...

Cris - request challenge...

Cris - request hint...

Hint sent!

Cris - sending answer...

Checking answer...

Cris - request challenge...

Cris - sending answer...

Checking answer...

Cris - sending answer...

Checking answer...

Cris - sending answer...

Checking answer...

```
=====
||      Navigation Menu      ||
=====
```

1. Receive Current Challenge
2. Send Response to Challenge
3. Request Hint
4. View Current Mission Status
5. Terminate Connection and Exit

Select an option: 2

Enter your response to the challenge:

Congratulations! You've completed all challenges. Press 4 to get the treasure coordinates.

```
=====
||      Navigation Menu      ||
=====
```

1. Receive Current Challenge
2. Send Response to Challenge
3. Request Hint
4. View Current Mission Status
5. Terminate Connection and Exit

Select an option: 4

Current mission status:

Congratulations! You've found the treasure at coordinates: X:100, Y:200. Disconnecting.

Terminate connections:

Server:

```

  _____
 / \      ; ;  \
| / \      / \
 \  ()oo()  .
  \|%( )*^^( )^ \
   | -%-----|
   \|  %  ))
   \| %
   \| %
   \| %

Welcome to the Guardian of Enigmas Server. Prepare to embark on a journey of puzzles
and mysteries!
Welcome Cris!
Cris - request challenge...
Cris - sending answer...
Checking answer...
Cris - decide to terminate the connection
^CExiting server...
```

Client:

```

=====
||      Navigation Menu      ||
=====
1. Receive Current Challenge
2. Send Response to Challenge
3. Request Hint
4. View Current Mission Status
5. Terminate Connection and Exit
Select an option: 5
Terminating connection...
```

Clues

Cesar Code

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Atbash Code

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A

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.