Universidad de Costa Rica
Ciencias de la Computación e informática

Fabián Orozco Chaves - B95690
Redes de Comunicación de Datos - Tarea 2

Ejemplos de sockets de los tutoriales

1) Python (Echo Server) - https://realpython.com/python-sockets/

echo-client.py

```python
# echo-client.py

import socket

HOST = "127.0.0.1"  # The server's hostname or IP address
PORT = 65432  # The port used by the server

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT))
    s.sendall(b"Hello, world")
    data = s.recv(1024)

print(f"Received {data!r}")
```

echo-server.py

```python
# echo-server.py

import socket

HOST = "127.0.0.1"  # Standard loopback interface address (localhost)
PORT = 65432  # Port to listen on (non-privileged ports are > 1023)

# Use the socket object without calling s.close().
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen()
    conn, addr = s.accept()
      #The with statement is used with conn to automatically close the socket at the end of the block.
    with conn:
        print(f"Connected by {addr}")
        while True:
            data = conn.recv(1024)
            if not data:
                break
            conn.sendall(data)
```

TERMINAL

```
[fabian:Tarea2]
  $ python echo-server.py
Connected by ('127.0.0.1', 55556)
```

```
[fabian:Tarea2]
  $ python echo-client.py
Received b'Hello, world'
```

2) C++ - https://www.geeksforgeeks.org/socket-programming-cc/

server.c

```c
// Server side C/C++ program to demonstrate Socket
// programming

#include <netinet/in.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <unistd.h>

#define PORT 8080

int main(int argc, char const* argv[]) {
    int server_fd, new_socket, valread;
    struct sockaddr_in address;
    int opt = 1;
    int addrlen = sizeof(address);
    char buffer[1024] = { 0 };
    char* hello = "Hello from server";

    // Creating socket file descriptor
    if ((server_fd = socket(AF_INET, SOCK_STREAM, 0))
```

client.c

```c
// Client side C/C++ program to demonstrate Socket
// programming

#include <arpa/inet.h>
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <unistd.h>

#define PORT 8080

int main(int argc, char const* argv[]) {
    int sock = 0, valread;
    struct sockaddr_in serv_addr;
    char* hello = "Hello from client";
    char buffer[1024] = { 0 };
    if ((sock = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
        printf("\n Socket creation error \n");
        return -1;
    }

    serv_addr.sin_family = AF_INET;
```

TERMINAL

```
[fabian:src]
  $ gcc server.c -o server

  [fabian:src]
  $ ./server
Hello from client
Hello message sent
```

```
[fabian:src]
  $ gcc client.c -o client

  [fabian:src]
  $ ./client
Hello message sent
Hello from server
```

3) Python & C++ - https://tinyurl.com/socketPyAndC

Receiver.cpp

```cpp
// C++ UDP Receiver

#include "Network.h"
#include <iostream>

#pragma once

int main() {
  int PORT = 8888;

  try {
    WSASession Session;
    UDPSocket Socket;
    char buffer[100];

    Socket.Bind(PORT);
    while (true) {
      sockaddr_in add = Socket.RecvFrom(buffer, sizeof(buffer));
      std::string input(buffer);
```

Transmitter.cpp

```cpp
// C++ UDP Transmitter

#include "Network.h"
#include <iostream>

#pragma once

int main() {
  std::string IP = "127.0.0.1";
  int PORT = 8888;

  try {
    WSASession Session;
    UDPSocket Socket;
    std::string data = "hello world";
    char buffer[100];

    while (true) {
      std::cout << "Enter data to transmit : " << std::endl;
```

EXPLORADOR

BOTH

- Network.h
- PY_Receiver.py
- Receiver.cpp
- receiver.exe
- Transmitter.cpp
- transmitter.exe

TERMINAL PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN

```
[Estudiante:Both]
$ g++ Receiver.cpp -lws2_32 -o receiver
Receiver.cpp:6:9: warning: #pragma once in main file
    6 | #pragma once
      |         ^~~~
[Estudiante:Both]
$ ./receiver.exe
Data received: Hello World
Data received: Fabian Orozco end program
Data received:
[Estudiante:Both]
$
```

```
[Estudiante:Both]
$ g++ Transmitter.cpp -lws2_32 -o transmitter
Transmitter.cpp:6:9: warning: #pragma once in main file
    6 | #pragma once
      |         ^~~~
[Estudiante:Both]
$ ./transmitter.exe
Enter data to transmit :
Hello World
Enter data to transmit :
Fabian Orozco end program
Enter data to transmit :
E[Estudiante:Both]
$
```

Utilizando el `receiver` de Python.

PY_Receiver.py

```python
    else:
        return data

def main():
    print ("[Fabian] Inicia Py_Receiver\n")

    my_sock = initUDP(IP, PORT)
    msg_received = "init_true"

    while (msg_received):

        msg_received = readUDP( my_sock )
        print (msg_received.decode("utf-8"))

    print ("\n[Fabian] Termina Py_Receiver\n")
main()

```

Transmitter.cpp

```cpp
// C++ UDP Transmitter

#include "Network.h"
#include <iostream>

#pragma once

int main() {
  std::string IP = "127.0.0.1";
  int PORT = 8888;

  try {
    WSASession Session;
    UDPSocket Socket;
    std::string data = "hello world";
    char buffer[100];

    while (true) {
      std::cout << "Enter data to transmit : " << std::endl;
```

EXPLORADOR

BOTH

- Network.h
- PY_Receiver.py
- Receiver.cpp
- receiver.exe
- Transmitter.cpp
- transmitter.exe

TERMINAL PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN

```
[Estudiante:Both]
$ python PY_Receiver.py
[Fabian] Inicia Py_Receiver

Hello
World


[Fabian] Termina Py_Receiver
```

```
[Estudiante:Both]
$ ./transmitter.exe
Enter data to transmit :
Hello
Enter data to transmit :
World
Enter data to transmit :
E[Estudiante:Both]
$
```