

Mariana Osorio Rojas - 464679

Fabio Andrés Guzman Figueroa

Universidad Pontificia Bolivariana - Facultad de Ingeniería
Ingeniería de Sistemas e Informática - Sistemas Operativos

PRÁCTICA N8 - SOCKETS

La práctica de Sockets en el contexto de Sistemas Operativos se enfoca en la implementación y demostración del uso de sockets en el lenguaje de programación C, específicamente en un modelo Cliente-Servidor. Los sockets son una forma de comunicación entre procesos que pueden estar en la misma máquina o en máquinas diferentes en una red.

|Descripción del Problema|

En este caso, el problema consiste en establecer una comunicación bidireccional entre un cliente y un servidor utilizando sockets. El servidor debe ser capaz de recibir mensajes del cliente y mostrarlos por pantalla, mientras que el cliente debe poder enviar mensajes al servidor.

|Solución Propuesta|

La solución propuesta implica la implementación de dos programas en C, uno para el servidor y otro para el cliente. El servidor se encarga de crear un socket, vincularlo a una dirección IP y puerto específicos, y escuchar conexiones entrantes. Una vez que se establece una conexión, el servidor lee los mensajes del cliente y los muestra por pantalla.

Por otro lado, el cliente crea un socket, se conecta al servidor a través de una dirección IP y puerto específicos, y envía mensajes al servidor. El cliente puede enviar mensajes de forma continua, los cuales son recibidos por el servidor y mostrados por pantalla.

Esta solución permite una comunicación efectiva entre el cliente y el servidor, demostrando la simplicidad y utilidad de trabajar con sockets en C en el contexto de Sistemas Operativos.

Código Fuente

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

int main(){
    int server_fd, new_socket, val_read;
    struct sockaddr_in address;
    char buffer[1024] = {0};

    server_fd = socket(AF_INET, SOCK_STREAM, 0);

    address.sin_family = AF_INET;
    address.sin_addr.s_addr = INADDR_ANY;
    address.sin_port = htons(5000);

    bind(server_fd, (struct sockaddr *)&address, sizeof(address));
    listen(server_fd, 1);

    new_socket = accept(server_fd, (struct sockaddr *)NULL, NULL);

    while (1){
        val_read = read(new_socket, buffer, 1024);
        printf("%s\n", buffer);
    }

    return 0;
}
```

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

int main(){
    int sock;
    struct sockaddr_in serv_addr;
    char *hello = "Hola, enviando desde el CLIENTE\n";
    char buf[128];

    sock = socket(AF_INET, SOCK_STREAM, 0);

    serv_addr.sin_family = AF_INET;
    serv_addr.sin_port = htons(5000);

    inet_pton(AF_INET, "127.0.0.1", &serv_addr.sin_addr);

    if (connect(sock, (struct sockaddr *)&serv_addr,
sizeof(serv_addr)) < 0){
        printf("Error en la conexión\n");
        return -1;
    }

    //send(sock, hello, strlen(hello), 0);
    //printf("Mensaje hola enviado\n");

    while (1) {
        fgets(buf, 128, stdin);
        send(sock, buf, strlen(buf), 0);
    }
}
```

|Captura de Ejecución|

The image shows a VS Code interface with a terminal window at the top and two source code files below it.

Terminal Window:

- Tab: `cppdbg: client`
- Content:

```
1, 2, 3...
Probando
.
Practica 8
Completada
Exitosamente
xitosamente
```
- Status bar: `Ln 20, Col 55 Spaces: 4 UTF-8 LF {} C Linux`

Editor Windows:

- server.c**

```
1 #include <netinet/in.h>
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <string.h>
5 #include <sys/socket.h>
6 #include <unistd.h>
7
8 int main(){
9     int server_fd, new_socket, val_read;
10    struct sockaddr_in address;
11    char buffer[1024] = {0};
12
13    server_fd = socket(AF_INET, SOCK_STREAM, 0);
14
15    address.sin_family = AF_INET;
16    address.sin_addr.s_addr = INADDR_ANY;
17    address.sin_port = htons(5000);
18
19    bind(server_fd, (struct sockaddr *)&address, sizeof(address));
20    listen(server_fd, 1);
21
```
- client.c**

```
1 #include <netinet/in.h>
2 #include <arpa/inet.h>
3 #include <stdio.h>
4 #include <stdlib.h>
5 #include <string.h>
6 #include <sys/socket.h>
7 #include <unistd.h>
8
9 int main(){
10    int sock;
11    struct sockaddr_in serv_addr;
12    char *hello = "Hola, enviando desde el CLIENTE\n";
13    char buf[128];
14
15    sock = socket(AF_INET, SOCK_STREAM, 0);
16
17    serv_addr.sin_family = AF_INET;
18    serv_addr.sin_port = htons(5000);
19
20    inet_pton(AF_INET, "127.0.0.1", &serv_addr.sin_addr);
21
```