Escuelas Supeior de Cómputo

Práctica 3
Recepción de datos a través de un socket no orientados a conexión (UDP)

Redes de computadora

Autor:
Héctor David González Tetuán

15/11/2025

Código en lenguaje C del programa para recibir un mensaje a través de un socket UDP.

Captura de pantalla de la terminal donde se muestra que recibió un mensaje a través de un socket UDP.

Código en lenguaje C del programa cliente chat.

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

void writeText( char* msj ){

    printf("\nEscribe un mensaje (Usuario): \n");
    fgets(msj, 512, stdin);
    msj[strcspn(msj, "\n")] = 0;

}

int main() {

    int udp_socket, lbind, tam, lrecv;
    struct sockaddr_in local, remota;
    unsigned char msj[512];
    unsigned char paqRec[512];
    udp_socket = socket(AF_INET, SOCK_DGRAM, 0);

    if(udp_socket == -1){
        perror("Error al abrir el socket");
        exit(EXIT_FAILURE);
    }
    else{

        perror("Exito al abrir el socket");

        local.sin_family = AF_INET;
        local.sin_port = htons(0);
        local.sin_addr.s_addr = INADDR_ANY;

        lbind = bind(udp_socket, (struct sockaddr *)&local, sizeof(local));

        if(lbind == -1){
            perror("Error en bind");
            exit(0);
        }
        else{

            perror("Exito en bind");

            remota.sin_family = AF_INET;
```

```
            remota.sin_port = htons(8080); // Puerto DNS
            remota.sin_addr.s_addr = inet_addr("10.100.82.16");

            while (1){

                writeText(msj);

                tam = sendto(udp_socket, msj, 512, 0, (struct sockaddr *)&
                    remota, sizeof(remota));

                if(tam == -1){
                    perror("Error al enviar");
                    exit(0);
                }
                else
                    printf("\nEnviando mensaje a Servidor: %s\n", msj);

                lrecv = sizeof(remota);
                tam = recvfrom(udp_socket, paqRec, 512, 0, (struct sockaddr
                     *)&remota, &lrecv);

                if(tam == -1){
                    perror("Error al recibir");
                    exit(0);
                }
                else
                    printf("\nEl mensaje recibido es: %s\n", paqRec);

            }

        }

    }

    close(udp_socket);
    return 0;

}
```

Código en lenguaje C del programa servidor chat.

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

void writeText( char* msj ){

    printf("\nEscribe un mensaje (Servidor): \n");
    fgets(msj, 512, stdin);
    msj[strcspn(msj, "\n")] = 0;

}

int main() {

    int udp_socket, lbind, tam, lrecv;
    struct sockaddr_in servidor, cliente;
    unsigned char msj[512];
    unsigned char paqRec[512];
    socklen_t len_cliente;

    udp_socket = socket(AF_INET, SOCK_DGRAM, 0);

    if(udp_socket == -1){
        perror("Error al abrir el socket");
        exit(EXIT_FAILURE);
    }
    else{

        perror("Exito al abrir el socket");

        memset(&servidor, 0, sizeof(servidor));

        servidor.sin_family = AF_INET;
        servidor.sin_port = htons(8080);
        servidor.sin_addr.s_addr = INADDR_ANY;

        lbind = bind(udp_socket, (struct sockaddr *)&servidor, sizeof(
            servidor));

        if(lbind == -1){
            perror("Error en bind");
            exit(EXIT_FAILURE);
        }
```

```
        else{

            perror("Exito en bind");

            while (1){

                lrecv = sizeof(cliente);
                tam = recvfrom(udp_socket, paqRec, 512, 0, (struct sockaddr
                     *)&cliente, &lrecv);

                if(tam == -1){
                    perror("Error al recibir");
                    exit(0);
                }
                else
                    printf("\nEl mensaje recibido es: %s\n", paqRec);

                writeText(msj);

                tam = sendto(udp_socket, msj, 512, 0, (struct sockaddr *)&
                    cliente, sizeof(cliente));

                if(tam == -1){
                    perror("Error al enviar");
                    exit(0);
                }
                else
                    printf("\nEnviando mensaje a Usuario: %s\n", msj);

            }
        }
    }

    close(udp_socket);
    return 0;

}
```

Capturas de pantalla donde se muestra la interacción de envío de mensajes entre cliente y servidor.