

**INSTITUTO POLITECNICO NACIONAL**

*ESCUELA SUPERIOR DE COMPUTO*

INGENIERIA EN SISTEMAS COMPUTACIONALES

ACADEMIA DE SISTEMAS DISTRIBUIDOS

**APLICACIONES Y COMUNICACIONES EN RED**

PRACTICA 3

SOCKETS NO ORIENTADOS A CONEXIONES BLOQUEANTES

INTEGRANTES:

* Lopez Torres Etnan Jezreel – 2015630260

GRUPO:6CV3

PROFESOR:Ojeda Santillán Rodrigo

Objetivo:

El objetivo de esta práctica es comprender y aplicar el uso de sockets UDP en la comunicación entre un cliente y un servidor, explorando cómo funcionan los sockets no orientados a conexión y su comportamiento bloqueante en la transmisión de datos.

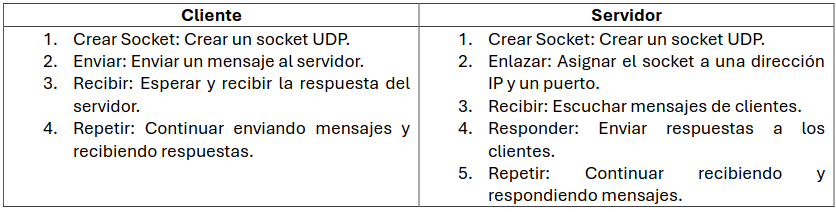
Introducción:

Los sockets UDP (User Datagram Protocol) permiten la transmisión de datos sin establecer una conexión previa entre el cliente y el servidor. A diferencia de TCP, UDP no garantiza la entrega de paquetes ni su orden, lo que lo hace más rápido pero menos confiable.UDP es un protocolo de comunicación de red que permite el envío de datagramas (paquetes de datos) sin necesidad de establecer una conexión previa entre el cliente y el servidor. Este tipo de socket no requiere una conexión continua o un proceso de "handshake" como en el protocolo TCP (Transmission Control Protocol).

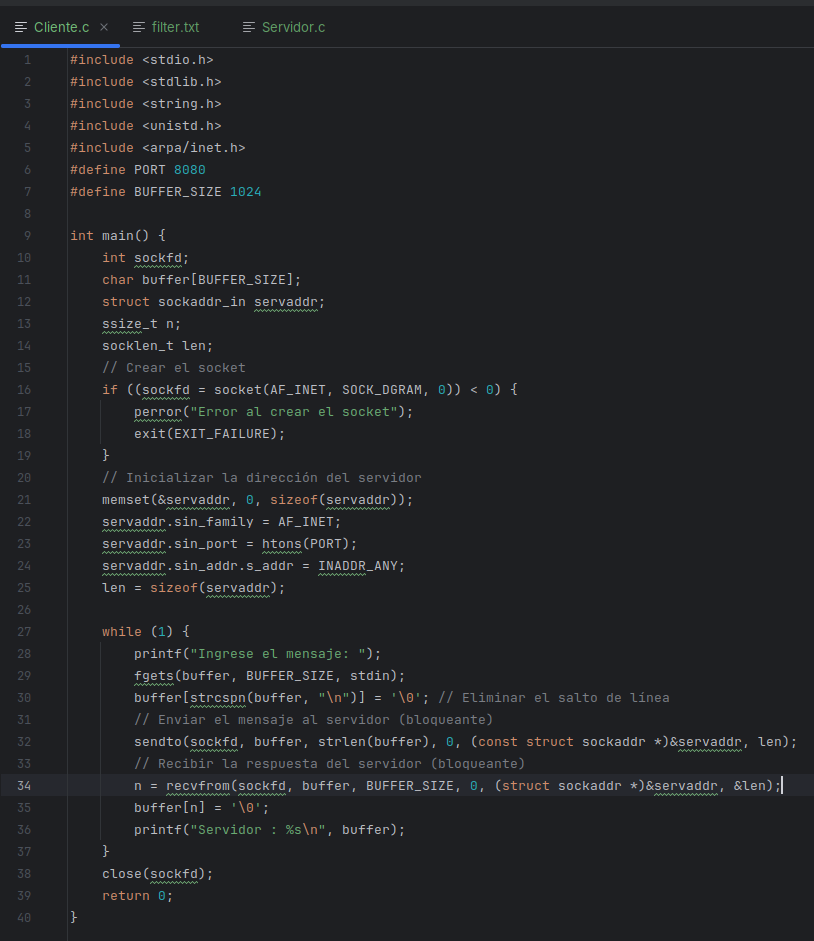
En esta práctica, se implementaron dos programas:

* Cliente: Envía mensajes al servidor y espera una respuesta.
* Servidor: Recibe los mensajes del cliente y responde.

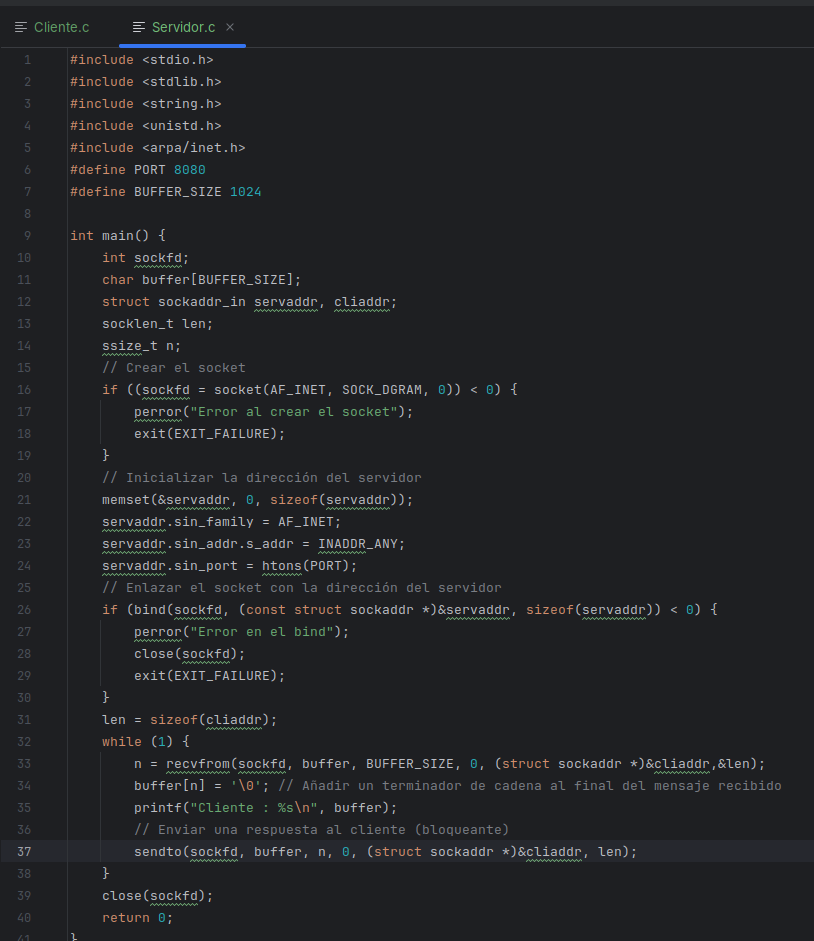
Se utilizaron funciones como socket(), sendto(), recvfrom(), y bind() para gestionar la comunicación entre ambos programas.



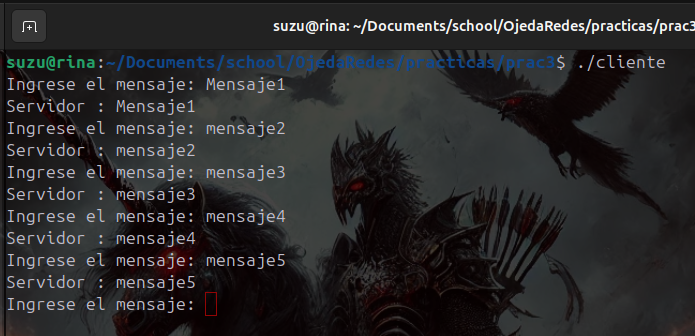
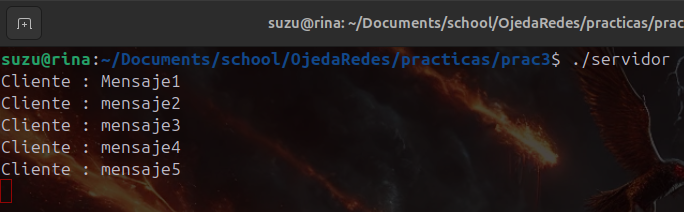
Desarrollo:



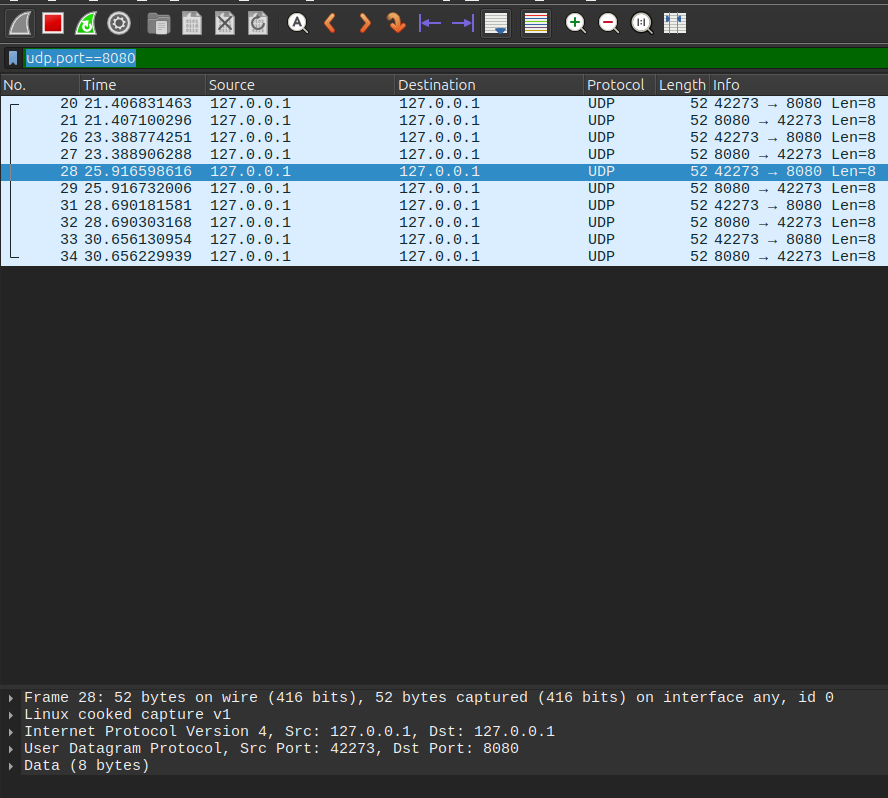
Esta captura muestra la implementación del cliente UDP en C, crea un socket UDP, envía mensajes con sendto(), para finalmente esperar respuesta del servidor con recvfrom() e imprimirbla respuesta recibida.



Esta captura muestra la Implementación del servidor UDP en C, crea de igual forma un socket UDP (socket(AF\_INET, SOCK\_DGRAM, 0)), asigna dirección y puerto (bind)y espera mensajes con recvfrom(), imprime en pantalla y responde con sendto().



La ejecución del programa se ve como se muestra en las capturas.



Esta es la captura de tráfico UDP con Wireshark, indica que el tráfico UDP es detectado correctamente y fluye sin problemas. En este caso se observa comunicación entre el cliente (puerto 42273) y el servidor (puerto 8080).

Conclusión:

A través de esta práctica, se pudo comprobar el funcionamiento de los sockets UDP bloqueantes, donde las operaciones de envío y recepción esperan hasta completarse. También se observó que, al no existir una conexión establecida, cada mensaje se maneja de manera independiente, permitiendo una comunicación rápida, pero sin garantía de entrega.

¿Cómo podría generar un ingreso a través de lo aprendido en esta práctica?

El conocimiento adquirido en esta práctica puede aplicarse en diferentes áreas para generar ingresos, tales como:

* Desarrollo de aplicaciones de red: Crear aplicaciones de comunicación en tiempo real como chat, VoIP o sistemas de monitoreo de red.
* Programación de servidores para videojuegos: Implementar servidores de juegos en línea donde la latencia baja sea más importante que la entrega confiable de paquetes.
* Servicios de transmisión de datos: Desarrollar software para el envío eficiente de datos en entornos de redes congestionadas.