

**INSTITUTO POLITECNICO NACIONAL**

*ESCUELA SUPERIOR DE COMPUTO*

INGENIERIA EN SISTEMAS COMPUTACIONALES

ACADEMIA DE SISTEMAS DISTRIBUIDOS

**APLICACIONES Y COMUNICACIONES EN RED**

PRACTICA 2

SOCKETS ORIENTADOS A CONEXIONES NO BLOQUEANTES

INTEGRANTES:

* Lopez Torres Etnan Jezreel – 2015630260

GRUPO:6CV3

PROFESOR:Ojeda Santillán Rodrigo

Objetivo:

El objetivo de esta práctica es comprender e implementar sockets orientados a conexiones no bloqueantes en un entorno cliente-servidor. Se busca explorar la capacidad de los sockets no bloqueantes para permitir múltiples conexiones simultáneas sin detener la ejecución del programa, mejorando la eficiencia en aplicaciones de alta concurrencia como servidores web, juegos en línea y sistemas de mensajería.

A través de esta práctica, se configurará un servidor que pueda aceptar conexiones de múltiples clientes sin bloquear el flujo de ejecución, mientras que los clientes podrán enviar y recibir datos sin necesidad de esperar a que se completen otras operaciones. Además, se emplearán funciones como select() para gestionar múltiples conexiones de manera eficiente, proporcionando un enfoque práctico para el desarrollo de aplicaciones en red de alto rendimiento.

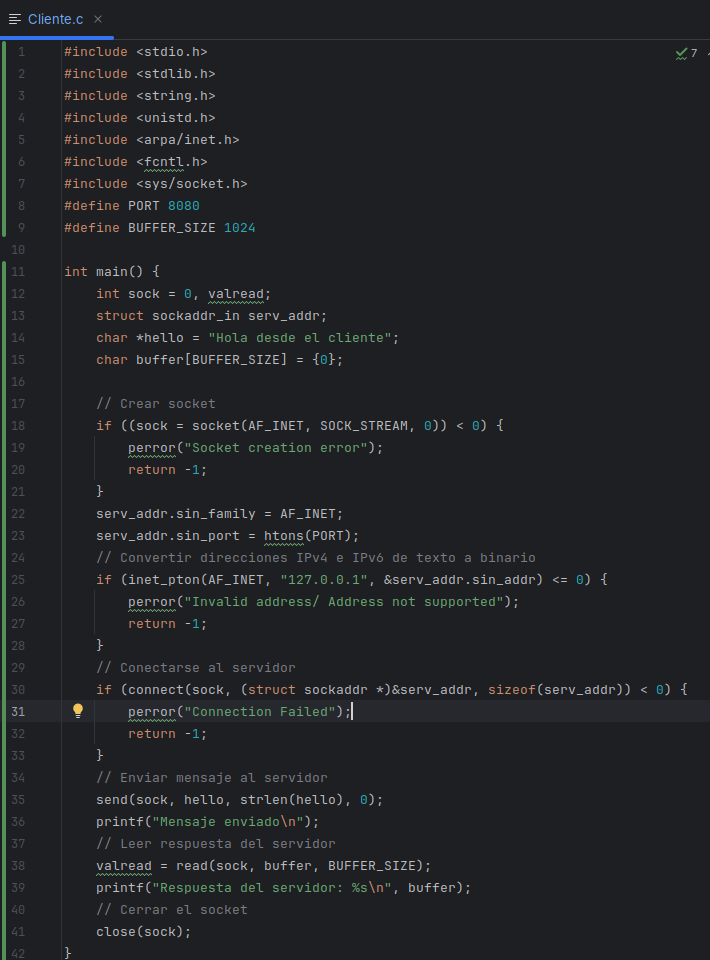
Introducción:

En el desarrollo de aplicaciones en red, la capacidad de manejar múltiples conexiones de manera eficiente es fundamental. Los sockets no bloqueantes permiten que un programa continúe ejecutándose sin necesidad de esperar a que una operación de red, como la recepción de datos o la aceptación de una conexión, se complete.

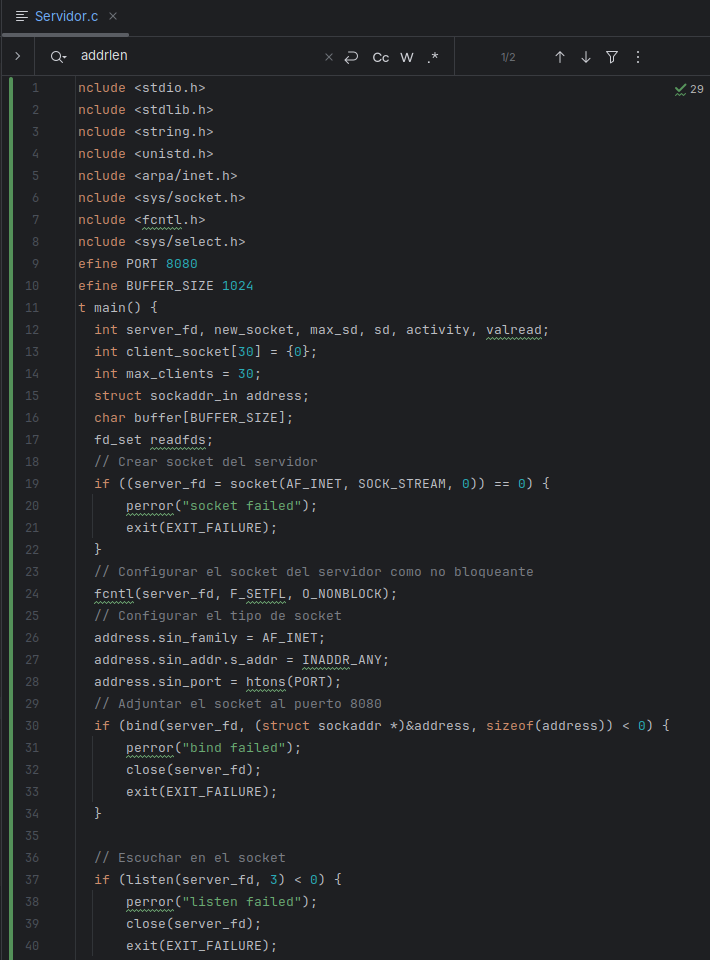
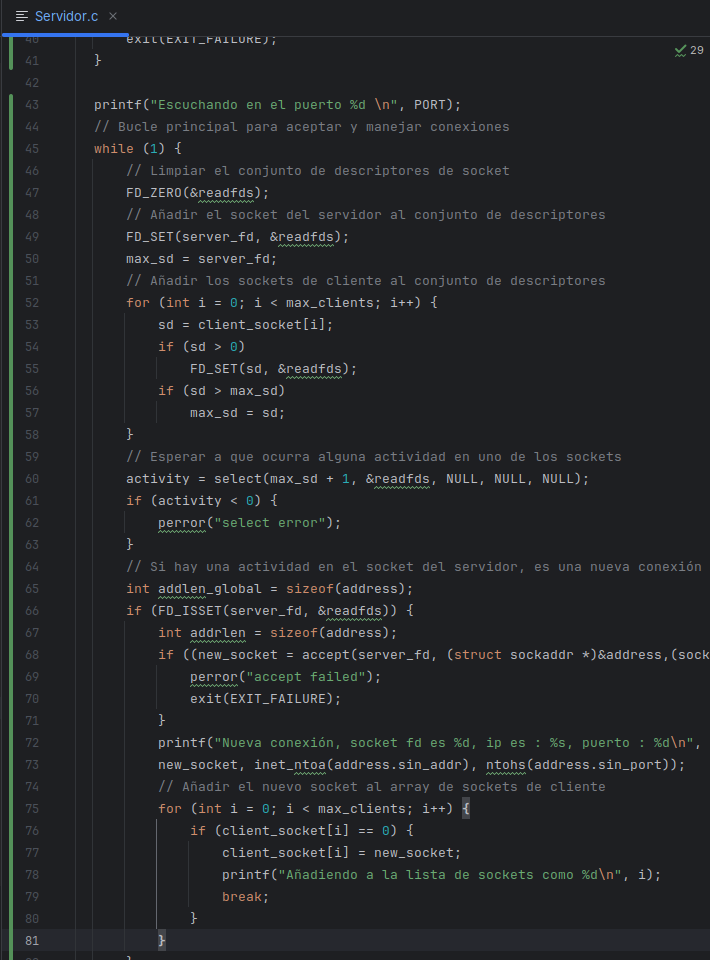
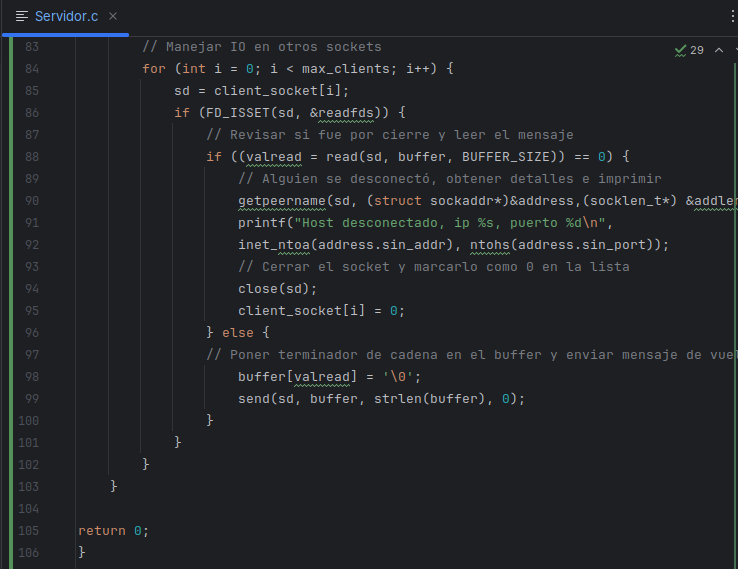
En una implementación tradicional con sockets bloqueantes, el servidor solo puede atender una conexión a la vez, lo que puede generar retrasos y bloqueos innecesarios en el sistema. En cambio, con los sockets no bloqueantes, el servidor puede aceptar y procesar múltiples conexiones simultáneamente sin detener su ejecución. Esto se logra utilizando funciones como select(), que permite monitorear múltiples sockets y reaccionar cuando hay actividad en alguno de ellos.

Esta práctica implementa un servidor en C que usa sockets no bloqueantes para gestionar múltiples clientes al mismo tiempo. Por otro lado, se desarrolla un cliente que se conecta al servidor, envía un mensaje y recibe una respuesta sin interrumpir su ejecución. Al finalizar la práctica, se comprenderá la importancia de los sockets no bloqueantes en el diseño de aplicaciones escalables y de alto rendimiento.

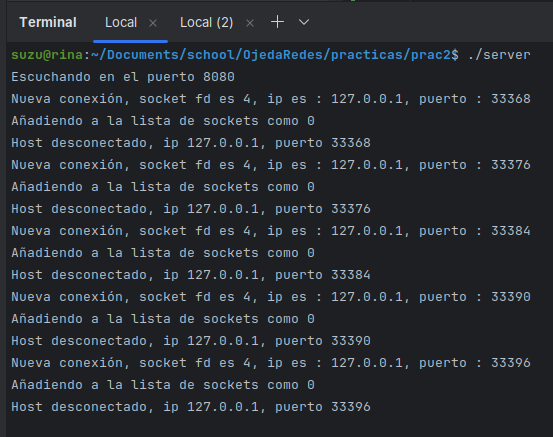
Desarrollo:



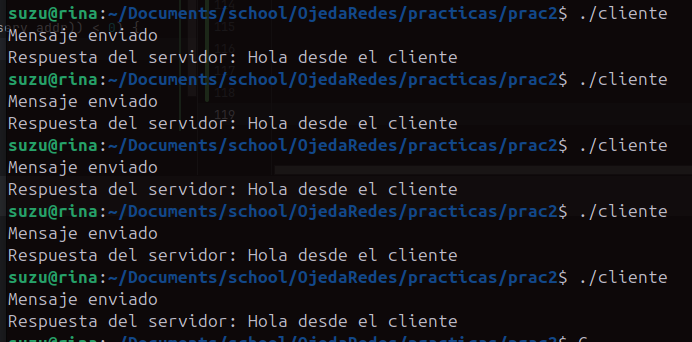
La imagen muestra el código fuente en C de un cliente que se conecta a un servidor mediante sockets orientados a conexión. Este código permite que un cliente se conecte a un servidor en el puerto 8080, envíe un mensaje y reciba una respuesta.

La imagen muestra el código fuente de un cliente en C que establece una conexión con un servidor utilizando sockets. Este código implementa un cliente simple que se conecta a un servidor en 127.0.0.1:8080, envía un mensaje y recibe una respuesta.

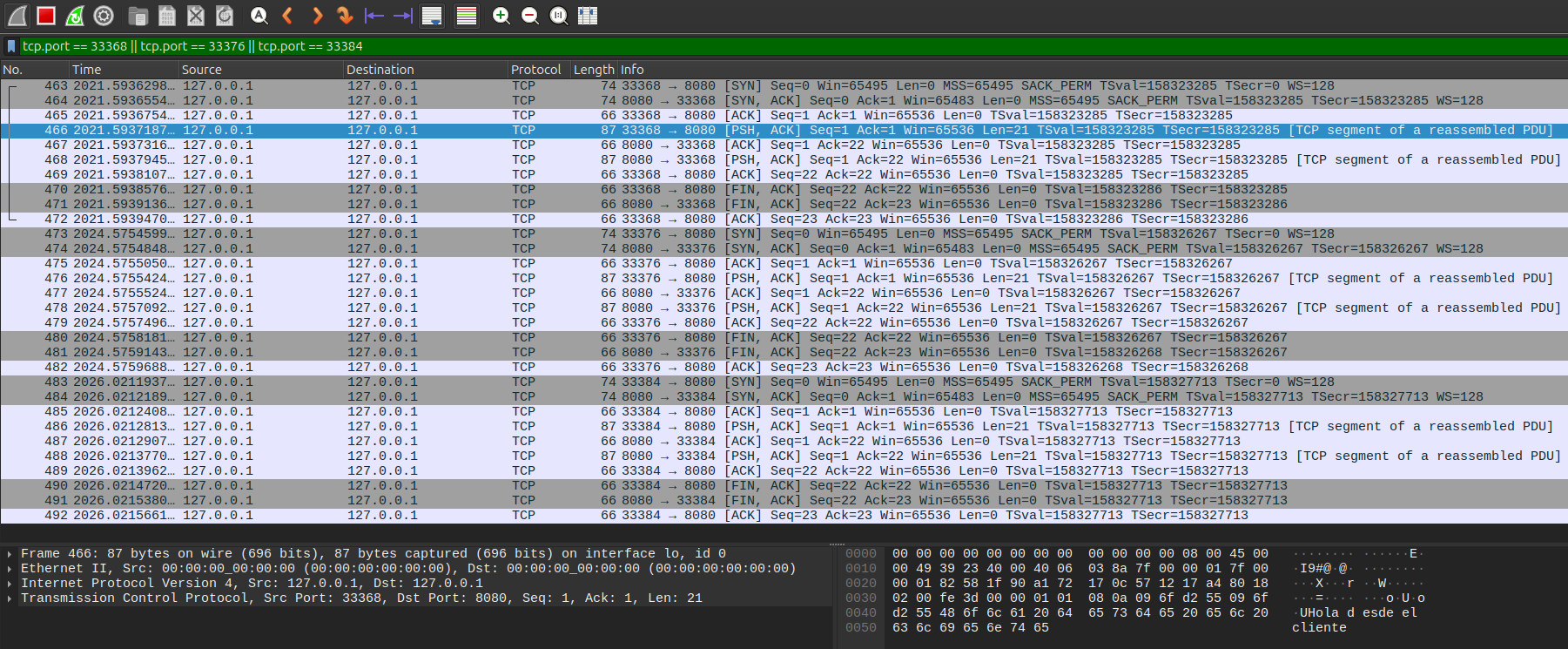


El servidor está funcionando correctamente, aceptando conexiones de clientes, añadiéndolos a su lista y detectando cuando un cliente se desconecta. Esto coincide con el código que maneja la comunicación de sockets no bloqueantes y múltiples clientes.



La captura confirma que:

* El cliente y el servidor están funcionando correctamente.
* La comunicación entre ambos es exitosa.
* Cada cliente se conecta, envía un mensaje y recibe una respuesta antes de cerrar la conexión.

La captura confirma que:

* El cliente y el servidor están enviando y recibiendo mensajes correctamente.
* Se puede ver el contenido del mensaje en los datos TCP, lo que demuestra que la comunicación es exitosa.
* Todo el tráfico es local, lo que sugiere que el programa está en pruebas en la misma máquina.

Conclusión:

Durante la práctica, se demostró cómo un servidor no bloqueante puede manejar múltiples clientes de manera simultánea sin afectar el rendimiento general. Se utilizó la función select() para monitorear múltiples sockets, lo que permitió responder de manera eficiente a las solicitudes sin necesidad de hilos adicionales o procesos separados.

¿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 servidores escalables: Crear servidores para juegos en línea, plataformas de streaming, chats en tiempo real o servicios web que requieren manejar múltiples conexiones simultáneas sin retrasos.
* Ciberseguridad y auditoría de redes: Implementar herramientas que monitoreen el tráfico de red en tiempo real sin bloquear la ejecución de otras tareas.
* Desarrollo de software en la nube: Trabajar en empresas que requieren servidores de alto rendimiento, como AWS, Google Cloud o Azure, donde la gestión de múltiples conexiones simultáneas es esencial.