

Ejercicio 1: Servidor de Chat Multihilo

Implementa un servidor de chat multihilo que pueda manejar múltiples clientes simultáneamente. Cada cliente debe ser capaz de enviar mensajes al servidor, que luego se distribuirán a todos los otros clientes conectados. El servidor debe mantener una lista de clientes conectados y reenviar los mensajes recibidos a todos los clientes.

Ejercicio 2: Transferencia de Archivos Segura

Desarrolla una aplicación cliente-servidor que permita la transferencia segura de archivos entre un cliente y un servidor a través de sockets. El servidor debe autenticar al cliente y solo permitir la transferencia de archivos si el cliente está autenticado correctamente. Además, implementa un cifrado de extremo a extremo para garantizar la confidencialidad de los archivos transferidos.

cifrado de extremo a extremo.

Ejercicio 3: Sistema de Votación Electrónica

Crea un sistema de votación electrónica utilizando sockets multihilo. El servidor debe manejar la autenticación de votantes y el registro de votos de forma segura y verificable. Los clientes deben poder conectarse al servidor, autenticarse como votantes válidos y emitir su voto de manera segura. El servidor debe ser capaz de contar los votos y generar un informe final una vez que se cierre la votación.

Ejercicio 4: Juego Multijugador en Tiempo Real

Desarrolla un juego multijugador en tiempo real utilizando sockets multihilo. El servidor debe gestionar la lógica del juego y mantener la sincronización entre todos los clientes conectados. Los clientes deben ser capaces de enviar acciones al servidor, que luego se distribuirán a todos los demás jugadores. El servidor debe manejar la lógica del juego, la detección de colisiones y la actualización del estado del juego para todos los clientes.

Ejercicio 5: Sistema de Mensajería Segura

Implementa un sistema de mensajería segura utilizando sockets multihilo. El servidor debe autenticar a los usuarios y permitirles enviar mensajes cifrados de extremo a extremo entre ellos. Los clientes deben poder conectarse al servidor, autenticarse y enviar mensajes de forma segura. Además, el servidor debe ser capaz de almacenar los mensajes encriptados de manera segura y entregarlos a los destinatarios adecuados.

Ejercicio 6: Sistema de Transmisión de Video en Tiempo Real

Desarrolla un sistema de transmisión de video en tiempo real utilizando sockets multihilo. El servidor debe capturar el video de una cámara o archivo y transmitirlo a todos los clientes conectados de manera eficiente. Los clientes deben poder recibir y reproducir el video en tiempo real, manteniendo la sincronización entre todos los espectadores.

Ejercicio 7: Sistema de Gestión de Archivos Distribuido

Crea un sistema de gestión de archivos distribuido utilizando sockets multihilo. El servidor debe actuar como un nodo central que almacena y gestiona los archivos compartidos por los clientes. Los clientes deben poder conectarse al servidor, cargar archivos, descargar archivos de otros clientes y realizar operaciones de administración de archivos de forma segura y eficiente.

Ejercicio 8: Plataforma de Juegos en Línea

Desarrolla una plataforma de juegos en línea utilizando sockets multihilo. El servidor debe manejar la autenticación de usuarios, la gestión de salas de juego y la comunicación en tiempo real entre los jugadores. Los clientes deben ser capaces de unirse a partidas, interactuar con otros jugadores y jugar juegos multijugador en tiempo real de manera fluida y segura.

Ejercicio 9: Sistema de Gestión de Flotas de Vehículos

Implementa un sistema de gestión de flotas de vehículos utilizando sockets multihilo. El servidor debe rastrear la ubicación en tiempo real de múltiples vehículos y proporcionar información actualizada sobre su posición, velocidad y estado. Los clientes deben poder conectarse al servidor, visualizar la ubicación de los vehículos en un mapa y recibir notificaciones en tiempo real sobre eventos importantes.

Ejercicio 10: Plataforma de Subastas en Línea

Desarrolla una plataforma de subastas en línea utilizando sockets multihilo. El servidor debe gestionar la creación y ejecución de subastas, permitir a los usuarios pujar en tiempo real y proporcionar información actualizada sobre el estado de las subastas. Los clientes deben ser capaces de participar en subastas, realizar pujas y recibir notificaciones sobre el progreso de las subastas en tiempo real.

COMPLICADOS

Ejercicio 1: Sistema Distribuido de Procesamiento Paralelo

Desarrolla un sistema distribuido de procesamiento paralelo utilizando sockets multihilo. El sistema debe permitir la distribución de tareas entre varios nodos de procesamiento. Cada nodo actuará como un servidor que acepta tareas de procesamiento desde un cliente central. El cliente enviará tareas al servidor, que luego asignará a los diferentes nodos de procesamiento disponibles. Cada nodo ejecutará su tarea de manera independiente y enviará los resultados al cliente central. El cliente central deberá coordinar la ejecución de las tareas y recopilar los resultados de todos los nodos para generar una salida final.

Ejercicio 2: Sistema de Monitoreo y Control de Redes de Sensores

Implementa un sistema de monitoreo y control de redes de sensores utilizando sockets multihilo. El sistema constará de un servidor central y múltiples nodos de sensores distribuidos. Cada nodo de sensor actuará como un servidor que enviará datos de sensores al servidor central. El servidor central recopilará datos de todos los nodos de sensores y proporcionará una interfaz para monitorear y controlar los sensores de manera remota. Los clientes podrán conectarse al servidor central para visualizar datos en tiempo real, enviar comandos de control a los nodos de sensores y configurar la operación del sistema de monitoreo.