

Universidad Nacional Autónoma de México
Facultad de Ingeniería

Uso de hilos en sistemas de comunicación: Chats Interactivos

Luis Daniel Salazar Islas

Hansel Yael Tepal Briseño

Sistemas operativos



01 de abril de 2025

1. Introducción

En la actualidad, la comunicación tiende a digitalizarse en su mayoría, permitiendo a las personas comunicarse en tiempo real a través de grandes distancias. Uno de los sistemas de comunicación más cotidianos son las aplicaciones de mensajería como whatsapp, por lo que, en esta exposición explicaremos la implementación de un chat, los mecanismos de comunicación requeridos y que papel juegan los hilos.

2. Arquitectura Cliente-Servidor

De inicio, nuestra implementación utiliza la famosa arquitectura Cliente-Servidor, por lo que explicaremos su funcionamiento.

De entrada es un modelo de diseño para sistemas distribuidos donde existen dos tipos de entidades: los servidores y los clientes. Los clientes, análogo a la vida real, se encargan de solicitar servicios y recursos, mientras que los servidores son los proveedores de los clientes.

El flujo de trabajo de este modelo es sencillo, un cliente realiza una petición a un servidor, el servidor recibe la petición, procesa y envía una respuesta de vuelta al cliente. En nuestra implementación, los clientes se representan como los usuarios de nuestra aplicación y el servidor será el encargado de administrar los mensajes entre los clientes.

Administrar la comunicación entre el cliente y servidor es de vital importancia, esta tiene que ser confiable (no se debe perder información), segura (protegido ante agentes externos) y los pedidos del cliente deben llegar en el orden en el que los solicita.

3. Sockets

Una de los mecanismos para resolver este reto son los sockets. Un socket es uno de los extremos en la vía de comunicación de dos sistemas en una red. Existen diversos tipos de sockets, como los que siguen los protocolos TCP y los UDP. Con fines prácticos, en esta implementación utilizaremos sockets TCP (Transmission Control Protocol), ya que son más confiables y permiten la comunicación entre el cliente y el servidor. Los sockets TCP son una forma de comunicación orientada a la conexión, lo que significa que antes de enviar datos, se establece una conexión entre el cliente y el servidor. Esto permite que los datos se envíen de manera confiable y en el orden correcto.

4. Componentes de un socket

- **Dirección IP:** Es la dirección única que identifica a un dispositivo en una red. En el caso de un servidor, es la dirección IP pública o privada que permite a los clientes conectarse a él.
- **Puerto:** Es un número que identifica un servicio específico en un servidor. Los puertos son utilizados por los protocolos de comunicación para dirigir los datos a la aplicación correcta.

- **Tipo de protocolo** Especifica el protocolo de comunicación que se utilizará para la conexión. En este caso, se utiliza el protocolo TCP.
- **Buffer de Entrada y Salida:** Son áreas de memoria utilizadas para almacenar temporalmente los datos que se envían y reciben a través del socket. El buffer de entrada almacena los datos recibidos del cliente, mientras que el buffer de salida almacena los datos que se enviarán al cliente.

5. Flujo de trabajo de un socket (TCP)

1. El cliente realiza una petición de conexión al servidor en un determinado puerto.
2. El servidor rechaza o acepta la conexión, se crea el socket del cliente al servidor y automáticamente el sistema proporciona un socket que conecta al servidor con la ip y puerto del cliente.
3. Una vez establecidas las conexiones, se establece un flujo de datos entre el cliente y el servidor. El cliente envía datos al servidor a través del socket, y el servidor recibe esos datos en su buffer de entrada.
4. El servidor procesa los datos recibidos y puede enviar una respuesta del cliente a través del socket. La respuesta se almacena en el buffer de salida del servidor.
5. Cuando la comunicación ha terminado, el cliente o el servidor pueden cerrar la conexión. Esto libera los recursos asociados al socket y cierra la comunicación.
6. El socket se cierra y los buffers de entrada y salida se liberan.

(Inicia ejemplo de la calculadora.)

6. Participación de los hilos

Los hilos son una forma de ejecutar múltiples tareas simultáneamente dentro de un mismo proceso. En el contexto de la arquitectura cliente-servidor, los hilos son utilizados para manejar múltiples conexiones de clientes al mismo tiempo.

Cuando múltiples clientes se conectan al mismo servidor, se crean hilos para cada conexión. Existe hilos que estarán dedicados únicamente a la escucha de peticiones y respuestas (Empezaría la explicación de los ejemplos del chat).

7. Ventajas y Desventajas

7.1. Ventajas de los sockets en aplicaciones de mensajería

- Permiten la comunicación en tiempo real entre los clientes y el servidor.
- Ofrecen una implementación más sencilla.
- Son confiables y garantizan la entrega de datos en el orden correcto.

- Permiten la comunicación bidireccional, lo que significa que tanto el cliente como el servidor pueden enviar y recibir datos al mismo tiempo.

7.2. Desventajas

- Requieren una conexión constante entre el cliente y el servidor, lo que puede ser un problema en redes inestables.
- Pueden ser más complejos de implementar que otros métodos de comunicación, como las API REST.
- Pueden ser vulnerables a ataques de seguridad si no se implementan correctamente.
- Consumen más recursos del sistema, requieren mantener conexiones abiertas y gestionar múltiples hilos.
- Pueden ser más difíciles de depurar y mantener puesto que la comunicación se realiza a través de sockets y no a través de una interfaz de programación de aplicaciones (API) más sencilla.
- Pueden ser más difíciles de escalar ya que cada conexión requiere un hilo separado y puede haber limitaciones en el número de conexiones simultáneas que un servidor puede manejar.

8. Bibliografía

- IBM i 7.5. (s. f.). <https://www.ibm.com/docs/es/i/7.5?topic=programming-how-sockets-work>
- What Is a Socket? (The Java™ Tutorials Custom Networking.All About Sockets). (Oracle). <https://docs.oracle.com/javase/tutorial/networking/sockets/definition.html>