Cuestionario Threads Cliente Servidor

Reyes Granados Enrique

**Usando Threads para esquemas de comunicación**

**Cliente-Servidor**

**Actividades:**

1. Revise el código. Consiste de tres archivos .java: ClientFullDuplex.java, MultiServerThread.java y ServerMultiClient.java.
2. Compile y ejecute en el siguiente orden
   1. ServerMultiClient
   2. ClientFullDuplex
3. En base a la revisión de código que realizó, pruebe el código
4. Después de probar y revisar el código fuente, conteste las siguientes preguntas

**Preguntas**

1. ¿A que IP y puerto se conecta por omisión el Cliente del código revisado?

La IP sería la 127.0.0.1 o localhost y el puesto el 12345

1. ¿Con qué cadena, el Cliente se desconecta del Servidor Central?

Con “FIN”

1. ¿Qué recibe el Cliente como respuesta si manda una cadena al Servidor?

“Echo…” y el mensaje que enviamos.

1. ¿Cómo maneja el Servidor cada conexión con un Cliente?

Cada que recibe una conexión crea un nuevo MultiServerThread y le manda el socket del Cliente para que sea este el que establezca la comunicación con el.

1. ¿Cuántos Clientes se pueden conectar al Servidor?

Como son Threads los que manejan las conexiones se van a poder tantas como tu Hardware los permita, en principio podría pensarse en tantas como se desee, pero los hilos causan peso en la ram de la maquia, sobre todo si se tienen grandes cargas de trabajo, por lo que depende de su equipo.

1. ¿Es posible saber el número de Clientes conectados al Servidor? Explique su respuesta

Como esta actualmente el programa no, pero agregando un par de líneas de código es posible, lo primero es agregar una variable de tipo int y la vamos a inicializar en 0, luego, cada que se conecte un nuevo cliente le sumamos 1 e imprimimos el resultado.

1. La comunicación que se establece entre Clientes y Servidor, ¿de qué tipo es: Half-Duplex, Full-Duplex, otra?

Es una comunicación full-Duplex, puesto que tanto el cliente como el servidor pueden enviar y recibir mensajes.

1. ¿Qué modificaciones haría al código para que se comportara de la siguiente forma: si un Cliente conectado manda la cadena #cuantos#, el Servidor le contestara cuántos Clientes en Total están actualmente conectados?

Lo primero es hacer la variable de arriba un atributo de la clase al que llamaremos numeroClientes y lo inicializamos con 0, despues creamos el método public int saberCliente() que nos va a retornar el número de clientes y finalmente dentro de MultiServerThread agregamos una condicional a que si la línea de entrada es #cuantos# nosotros le regresemos el numero de clientes a través de esta función.

1. ¿Es posible definir y usar algún protocolo de comunicación entre los Clientes y el Servidor? En caso afirmativo, defínalo. En caso negativo, explique.

De hecho, la librería socket de Java hace uso del protocolo TCP para establecer la comunicación, además hace uso del protocolo IPV4 con la IP y con el uso de puertos ya estamos hablando del prtocolo TCP/IP. Además de esto nosotros establecimos las reglas de comunicación entre el cliente y el servidor que es que el cliente se conecte al servidor y este le asigne un Thread para que este sea quien maneje la conexión, además de que el servidor escuche por un puerto especifico, por lo que muy ambiguamente podemos estar hablando que esto sea un protocolo de comunicación de nuestro programa.

1. ¿Es posible establecer un canal seguro entre los Clientes y el Servidor? Qué restricciones tendría?

Si es posible, de hecho, nosotros podemos implementar ciertas características a nuestro programa que implemente la seguridad, podríamos cifrar todos los mensajes a través de una cadena determinada de texto. También podríamos agregar al mensaje un numero que sirviera como suma de verificación o implementar otro tipo de medidas como que solo se admitieran solicitudes en el servidor si provienen de una IP en específico, por mencionar algunas características de seguridad que podríamos implementar.