Clase 30/09/2020

CLIENTE

Socket Se conecta al servidor

Socket conexion = new Socket(“servidor”,#puerto) -> servidos puede ser localhost o un dominio.

Para poder enviar y recibir datos a travez del socket necesitamos un “stream”, en este caso dos DataInput y DataOutput

DataInputStream entrada = new DataInputStream(connexion.getInputStream());
DataOutputStream salida = new DataOutputStream(conexion.getOutputStream());

Para poder enviar datos por el stream se utilizan los métodos

```
salida.writeInt( );
salida.writeDouble();
salida.write( “string”.getBytes());  -> el metodo wite envia arreglo de bytes por eso se convierte el string
```

ByteBuffer se utiliza para hacer una manera mas eficiente al enviar los conjuntos de datos

ByteBuffer b = ByteBuffer.allocate(N * 8); -> N :Cantidad de números que queremos enviar
b.putDouble( );
byte a[] = b.array(); -> se convierte el objeto en arreglo de byte para poder enviarlo por el método write del stream
salida.write(a);

SERVIDOR

SocketServer Es un socket de servidor

ServerSocket servidor = new ServerSocket(#Puerto);

Aceptamos el socket del cliente con el método accept

Socket conexion = servidor.accept(): -> este metodo es bloquante

Para poder enviar y recibir datos volvemos a usar DataInputStream y DataOutputStream

Para poder leer lo que viene en el stream se usan los métodos
entrada.readInt() -> int n = entrada.readInt();
entrada.double():

Para poder recibir lo que un ByteBuffer envía se necesita un arreglo de byte del tamaño indicado para así poder hacer el casteo con el método wrap y para leer los numeros una vez costeado se usa el método .get

byte[] a = new byte[#N * 8];
read(entrada, a , 0 ,#N*8) -> entrada es el stream, a es donde se va a guardar, 0 donde inicia, y el ultimo parámetro el tamaño
ByteBuffer b = ByteBuffer.wrap(a);
for(i=0;i < #N, i++) b.getDouble();

Actividad

1. Modifique el programa cliente para que envíe 10000 números punto flotante utilizando el método writeDouble (enviar los números 1.0, 2.0, 3.0 ... 10000.0). Mida el tiempo que tarda el programa cliente en enviar los 10000 números, se sugiere utilizar el método System.currentTimeMillis()
    1. R: 5
2. Modifique el programa servidor para que reciba los 10000 números que envía el programa cliente. Mida el tiempo que tarda el programa servidor en recibir los 10000 números.
    1. R: 22
3. Ahora modifique el programa cliente para que envíe los 10000 números utilizando ByteBuffer. Mida el tiempo que tarda el programa cliente en enviar los 10000 números.
    1. R: 3
4. Modifique el programa servidor para que reciba los 10000 números utilizando ByteBuffer. Mida el tiempo que tarda el programa servidor en recibir los 10000 números.
    1. R: 20
5. ¿Qué resulta más eficiente, enviar los números de manera individual mediante writeDouble o enviarlos empacados mediante ByteBuffer?
    ByteBuffer