

Instituto Politécnico Nacional

Escuela Superior de Cómputo

Aplicaciones para comunicaciones de red

Práctica No 1

Prieto de la Cruz Felipe

Huerta Larrazolo Jehiel David

18/04/18

Profesor: Bautista Rosales Sandra Ivette

Grupo: 3Cv5

**Objetivo de la práctica:**

El estudiante implementará una aplicación para el envío de **múltiples archivos** a través de la red

haciendo uso de sockets de flujo.

Introducción

Puertos

Cada proceso que se comunica con otro proceso se identifica a sí mismo a la familia de protocolos TCP/IP por uno o más puertos. Un puerto es un número de 16 bits, usado por el protocolo host-a-host para identificar a qué protocolo de más alto nivel o programa de aplicación (proceso) debe entregar los mensajes de entrada.

Como algunos programas de más alto nivel son protocolos por sí mismos, estandarizados en la familia de protocolos TCP/IP, tales como telnet y ftp, usan el mismo número de puerto en todas las realizaciones de TCP/IP. Aquellos números de puerto "asignados" se denominan puertos bien-conocidos y las aplicaciones estándares servicios bien-conocidos.

Los puertos "bien-conocidos" los controla y asigna la Autoridad de Números Asignados de Internet (IANA) y en la mayoría de los sistemas sólo pueden usarlo los procesos del sistema o programas ejecutados con privilegios de usuario. Los puertos "bien-conocidos" asignados ocupan números de puerto en el rango de 0 a 1023. Los puertos con números dentro del rango 1024-65535 no los controla la IANA y la mayor parte de los sistemas únicamente usan programas desarrollados por usuarios.

Sockets

Consideremos la terminología siguiente:

Un socket es un tipo especial de manejador de fichero que utiliza un proceso para pedir servicios de red al sistema operativo.

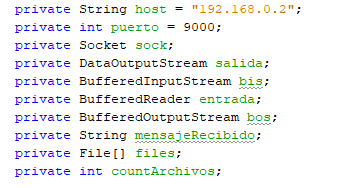
Una dirección de socket es la tripleta: {protocolo, dirección-local, proceso-local}

En la familia TCP/IP, por ejemplo: {tcp, 193.44.234.3, 12345}

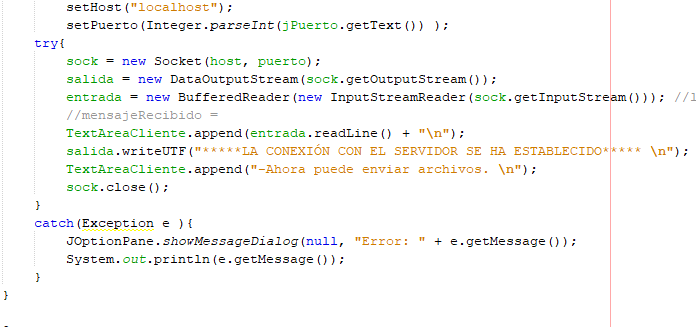
Una conversación es el enlace de comunicación entre dos procesos.

//Desarrollo

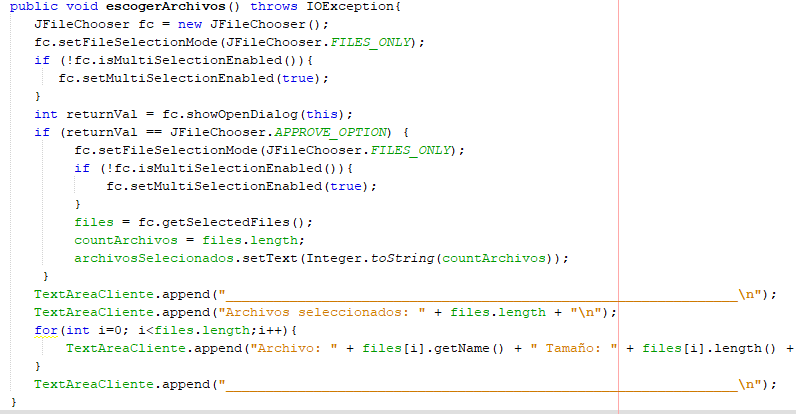
Cómo primer paso tenemos que declarar nuestras variables que vamos a usar, en este caso serán sockets, string donde guardaremos los valores del puerto, y la ip, nuestro buffer, que determinará el tamaño del archivo, estas variables serán similares para nuestras dos interfaces que serán, cliente servidor.



Tendremos nuestro siguiente método donde inicializaremos la conexión con el socket, obteniendo la información del host y puerto, los mensajes de conexión para las interfaces cliente y servidor.



Continuamos con el siguiente método donde incluiremos la librería JFileChooser que nos permitirá seleccionar diferentes archivos.

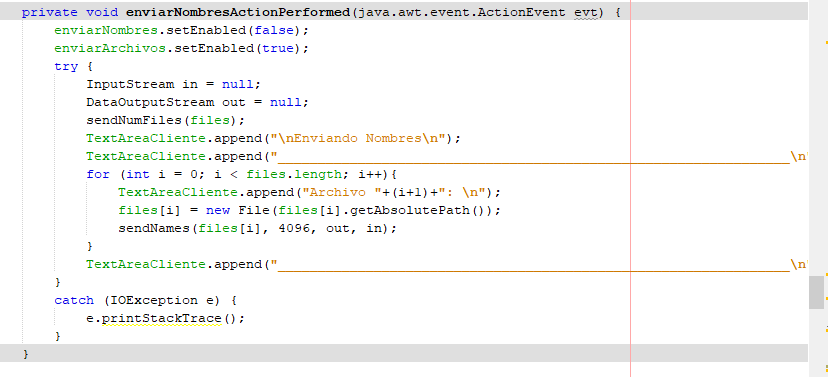


Para el método donde enviamos los nombres de nuestros archivos lo que vamos a hacer es con nuestro arreglo que se llama “Files” es donde guardamos la información de nuestros archivos, obtenemos su información, tanto como su tamaño.

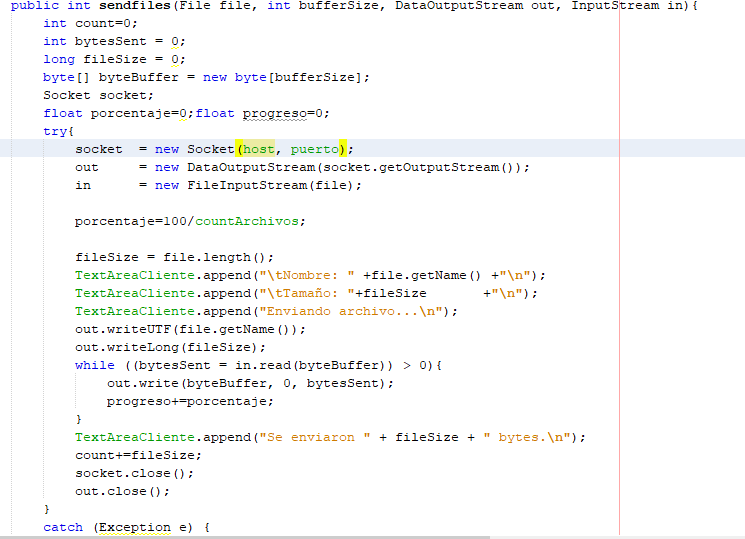
Una vez que tenemos la información de nuestros archivos,

Una vez que tenemos la información lo que haremos será establecer la conexión con nuestro socket, en este método uno de nuestros parámetros será el tamaño de nuestro buffer.

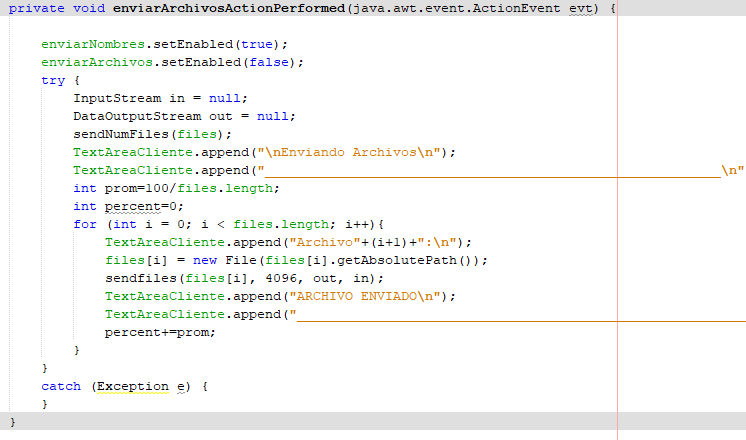
Ya recopila la información y haber establecido la conexión con nuestro socket, procederemos a mandar el nombre de los archivos, aquí se mandará nuestro arreglo, en el cual ya están alojados los archivos, únicamente extrajimos sus datos.



Ahora, para poder enviar nuestros archivos, haremos algo similar como lo que hicimos para poder obtener los nombres, estableceremos el tamaño del buffer, el nombre del archivo y su tamaño, por lo tanto también debemos establecer una conexión con el socket.

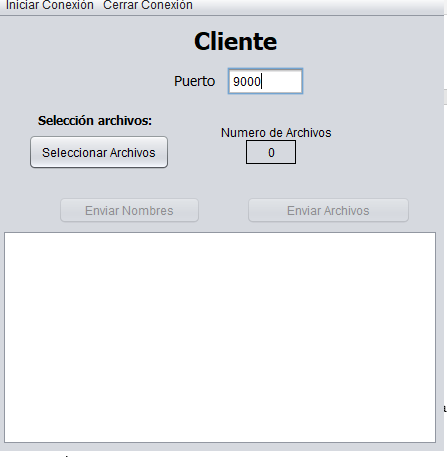


Igualmente, que con nombres el procedimiento para enviar los archivos es similar

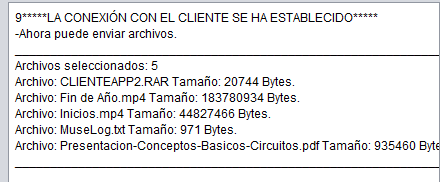


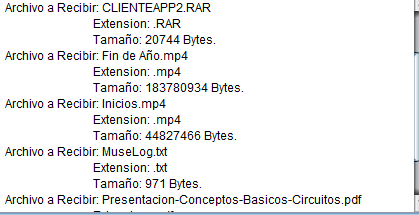
//Cuestio

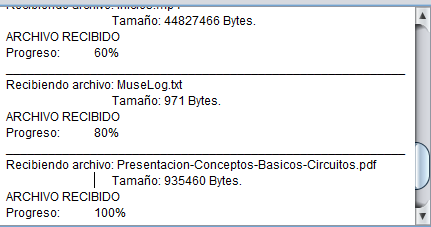
Así es como quedan nuestras interfaces:











1. ¿En qué consiste el algoritmo de Nagle y de qué manera se podría aplicar en la práctica? ¿Cuáles son sus

ventajas y desventajas?

El Algoritmo de Nagle se trata de un procedimiento que supone una mejora y aumento de eficiencia de las redes de comunicación basadas en Transmission Control Protocol (TCP). El algoritmo de Nagle es un método heurístico para evitar enviar paquetes IP particularmente pequeños, también denominados pequegramas (del inglés tinygrams). El algoritmo de Nagle intenta evitar la congestión que estos paquetes pueden ocasionar en la red reteniendo por poco tiempo la transmisión de datos TCP en algunas circunstancias.

Con esté algoritmo podemos mejorar el tiempo en que se envían los archivos haciéndolo que sea un poco más rápido.

Ventajas:  
Utilizando el algoritmo de Nagle, una conexión TCP sólo puede tener un segmento de tamaño pequeño sin que se haya reconocido, solo podemos tener un único segmento.

1. ¿Qué tipo de archivos se enviaron más rápido?

Los archivos con menor peso.

1. ¿Cuál fue el número máximo de archivos que fue posible enviar a la vez?

Para esta prueba intentamos con 13 archivos que tenían un peso menor.

1. ¿Cuál fue el tamaño de archivo más grande que se pudo transferir? ¿por qué?

683 Mb aunque nuestro buffer lo configuramos para que pudiera soportar 1GB.

1. ¿Qué es el orden de red?
2. Si deseáramos enviar archivos de tamaño muy grande, ¿qué cambios sería necesario hacer con respecto a

los tipos de datos usados para medir el tamaño de los archivos, así como para leer bloques de datos del

archivo?

Un cambio sería en el tamaño del buffer, la manera en la que estamos descomponiendo los archivos, ya que lo que hacemos es enviarlos por bytes, trozos.

1. ¿Por qué razón es importante utilizar el orden de red al enviar los datos a través de un socket?

Por la forma en que es la transferencia de los datos.

1. ¿Qué es el orden de red?

La prioridad en que van llegando los datos.

Conclusiones:

David:

En esta práctica reafirmamos nuestros conocimientos sobre sockets, los cuales nos permitieron establecer conexión entre 2 aplicaciones, la definición de socket quedo mucho más clara para mí, ya que fue el medio por el cual pudimos abrir una ventana hacia otro ordenador comunicándonos a través de un puerto, observando como viajaba la información de un ordenador a otro. Esto sin duda es de gran importancia ya que sin el avance en estas tecnologías, la redes sociales o la forma en que nos comunicamos hoy en día no serían lo mismo.

Prieto de la Cruz Felipe:

En esta práctica volví a ver lo eficaz que puede ser Java al igual que sus bibliotecas, para la parte de los sockets, llegue a tener problemas en averiguar cómo se conectaban, ya que tienes que configurar bien los puertos, cuando ejecutas un socket, para que no se trabe tu aplicación es necesario que trabes con hilos, también fue interesante el pensar cómo podrías enviar los archivos y una forma de lograrlo fue en trozos, byte por bye, entre menor peso más rápido se enviará tu archivo.

# References

|  |
| --- |
|  |
|  |
|  |
|  |
| [4] | [En línea]. Available: http://www.serinfor.net/infraestructura-informatica/. |
| [5] | EcuRED. [En línea]. Available: https://www.ecured.cu/Socket. |
| [6] | SOPA. [En línea]. Available: http://sopa.dis.ulpgc.es/ii-dso/leclinux/ipc/sockets/sockets.pdf. |