PRÁCTICO 3

COMPILAR, TESTEAR Y EXTENDER EL SOFTWARE FTP DE EJEMPLO.

Propósito

Aprender a modificar y extender programas de aplicación que se comunican sobre Internet

Generalidades

Desarrolle, compile y testee un programa cliente *FTP* y un programa servidor *FTP*. Luego ejecute ambos programas en computadoras separadas, y verifique que se pueden comunicar.

Luego de haber realizado los laboratorios 1 y 2, usted está en condiciones de comenzar con este nuevo laboratorio. El mismo consiste en un cliente `FTP` y en un servidor `FTP`. Como su nombre lo indica, `FTP`, Se basa fuertemente en el protocolo donde uns ervidor mantiene una base de datos a la espera de ser consultada por clientes.

Para ello, en esta oportunidad, más que un simple chat como lo veníamos trabajando anteriormente, el servidor va a estar dotado de una serie de comandos ante los cuales opera.

Procedimientos y Detalles

1. Con el código desarrollado en el segundo laboratorio, deberá implementar un cliente `FTP` y un servidor `FTP`. Nuevamente, el servidor escuchará en un puerto determinado y el cliente intentará conectarse con él, en dicho puerto

2. El servidor `FTP` deberá escuchar indefinidamente por comunicaciones pero sólo puede atender una a la vez. Una vez establecida dicha comunicación, deberá procesar todo comando que el cliente le envíe y de ser correcto, responder acorde al mismo.

3. Una vez establecida la comunicación entre ambas partes, tanto el cliente como el servidor pondrán enviar mensajes en cualquier momento.

4. Una detalle a tener en cuenta. A diferencia del laboratorio previo, el servidor no ha de responder ante posibles ingresos de consola. Sólo responde a pedidos del cliente. Por lo que el lado servidor no necesitará contar con la herramienta select.

5. Los comandos a implementar son los siguientes:

    a. ***list***: Este comando le solicita al servidor que liste los archivos que tiene presente en su directorio de trabajo

    b. ***get FILENAME***: Este comando pide la transmisión del archivo *FILENAME* presente en el directorio de trabajo del servidor. Si el archivo existe, debe de ser enviado correctamente al cliente. En caso de no existir, deberá devolverse un mensaje de error "*El archivo no existe*".

    c. ***quit***: Este comando finaliza la conexión entre el cliente y el servidor. El servidor deberá quedar listo para recibir nuevas conexiones.

Ejemplo

Desde la consola del cliente se muestra la siguiente interacción:

```
> list

SistemasOperativos.txt
Redes.txt
```

```
BasedeDatos.txt

> get Redes.txt
> ls
Redes.txt
```

Explicación:

En la primer instrucción el cliente le solicita el listado de archivos en su directorio. El servidor los devuelve y el cliente lo muestra en consola.

Para el segundo comando, el cliente le solicita el archivo `Redes.txt`. El archivo es transferido y guardado en el cliente. El comando `ls` muestra el directorio del cliente, y en efecto, se encuentra el archivo pedido.

Obligatorios

6. Agregue un temporizador al programa chat para que la conección TCP sea automáticamente cerrada luego de un minuto de tiempo muerto (es decir, un minuto en el cual ningún caracter es enviado ni recibido).