LABORATORIO 5

SERVIDOR FTP MULTIUSUARIO.

Propósito

Aprender a programar 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, 2, 3 y 4, usted está en condiciones de comenzar con este nuevo laboratorio. El mismo consiste en un servidor FTP multiusuario y varios clientes 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.

Procedimientos y Detalles

1. Con el código desarrollado en el cuarto laboratorio, deberá implementar un `servidor` multiusuarios. El cliente `FTP` ya lo tiene desarrollado, por lo cual puede reusarlo. Nuevamente, el servidor escuchará en un puerto determinado y los clientes intentarán conectarse con él, en dicho puerto.

2. El servidor `FTP` deberá escuchar indefinidamente hasta un total de `n` comunicaciones y debe de ser capaz de atenderlas al mismo tiempo.

4. Una vez establecida la comunicación con un cliente, tanto el cliente como el servidor pondrán enviar mensajes en cualquier momento

5. Una detalle a tener en cuenta. En esta comunicación, los clientes manejan dos tipos de entradas de datos principalmente: tanto de teclado, como del usuario. Esto requiere el uso de la herramienta `select` provista por python. Deberá investigar la misma y su funcionamiento.

6. A diferencia del laboratorio previo, donde el servidor no habia de responder ante posibles ingresos de consola sino sólo a pedidos del cliente, la herramienta *select* no era necesaria. En este caso, para poder antender varias clientes al mismo tiempo, `select` es necesaria en el servidor.

7. 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.

8. Modifique el cliente y el servidor para que ambos requieran al usuario un nombre (un nombre arbitrario). Incluir el nombre junto con un signo mayor (>) al comienzo de cada línea de texto antes de que sea enviada a través de internet.