

Taller Sockets

Introducción

En este taller aprenderemos a usar la interfaz de sockets de Linux, para crear un programa de descarga. También harán uso de la interfaz E/S.

Descripción

En este taller implementarán dos programas:

1. Un programa **servidor**, que se encargará de enviar el archivo solicitado a quién los requiera
2. Un programa **cliente**, que nos permitirá descargar archivos desde una máquina que esté corriendo el programa servidor.

El programa servidor recibirá tres argumentos: la IP, el puerto donde escuchará las solicitudes y el directorio del cual servirá los archivos al cliente (para descarga). La ruta puede ser relativa o absoluta. Por ejemplo:

`./servidor 127.0.0.1 45343 /home/archivos/`

Este se ligará a esta interfaz y escuchará las solicitudes del cliente. Las solicitudes son una simple de cadena de texto, que pueden ser:

1. El string "**_listar_**". Cuando el servidor recibe este string, el servidor obtendrá el nombre de todos los archivos del directorio que especificamos al iniciar el servidor. Luego mandará cada uno de los nombres al cliente. Cuando ya no tenga mas nombre que mandar, el servidor mandará el string **_fin** (note el subguion). **TIP: para hacer más fácil el manejo de las rutas, cambia el directorio de trabajo del program al directorio que le mandamos como argumento al servidor con `int chdir(const char *path)`.**
2. El string con el nombre del archivo. Si el servidor recibe el nombre de un archivo, el servidor deberá leer los bytes del archivo y enviárselos al cliente. Si el archivo no existe, el servidor debe enviar el string **"no_existe"**.

Para correr el cliente, usamos el siguiente comando:

`./cliente ip puerto`

Por ejemplo:

`./cliente 127.0.0.1 45343`

El cliente deberá mostrar el siguiente menú:

1. Listar archivos a descargar
2. Descargar archivo

Si el usuario ingresa la opción 1 (numero 1), el cliente debe pedir al servidor la lista de archivos disponibles y mostrarlos en pantalla. Al seleccionar la opción 2, se debe pedir el usuario el nombre del archivo que quiere descargar, y se lo envía al servidor. Luego el cliente descarga el archivo y lo guarda con el mismo nombre que ingreso el usuario, en el directorio actual de trabajo. Al final muestre la cantidad de bytes descargados. Por ejemplo:

```
1. Listar archivos a descargar
2. Descargar archivo

Opcion: 1

Archivos disponibles

hola.png
data.zip
lista.txt

1. Listar archivos a descargar
2. Descargar archivo

Opcion: 2

Nombre de archivo a descargar: data.zip

Descargando archivo...

Archivo data.zip descargado (12040 bytes).
```

El programa cliente recibirá los bytes enviados por el servidor, y los guardará en el archivo especificado por el último argumento. **TIP: la función `recv/read` retorna la cantidad de bytes leídos si logró leer algo del socket, -1 en error y 0 si el otro lado cerro la conexión. Si la conexión se ha cerrado, `recv/read` retornará enseguida, en vez de bloquearse (si no hay más bytes para leer).**

Ponga su nombre en el README.