Documentación

Chat IRC

Autores:
Roberto García Teodoro
Mario Valdemaro, García Roqué

Correo:
roberto.garciat@estudiante.uam.es
mariov.garcia@estudiante.uam.es

Estructura del makefile

- Para generar el ejecutable usar el método chat.
- Para borrar los .o y los .a usar el método mrPropert
- Para comprimir usar el método comprimir.

Estructura de inicio de conexión, desconexión y mantenimiento de la conexión.

Para conectarse al servidor se establece la conexión mandando los menajes al servidor de NICK y USER y se abren 2 hilos, estos hilos mantienen la conexión y reciben los mensajes del servidor.

Uno de los hilos que se crea es un hilo que manda constantemente el mensaje PING al servidor, cada 30 segundos, hemos decidido llevar a cabo esta implementación ya que no nos quedo muy claro si hay una forma estándar en los servidores de IRC para mantener la conexión y este método nos asegura que se mantendrá la conexión.

La desconexión cerrara el socket de emisión y todos los hilos que estén activos.

Protocolo de envío de ficheros

Estructura del servicio:

El sistema consta de 2 partes un cliente y un servidor:

- Cliente:El que recibe la información, no inicia la conexión, confirma esta.
- Servidor: Manda la información, inicia la conexión.

Secuencia de ejecución:

- El que hace de servidor manda un fichero con el comando /fsend
- El 'servidor' crea un nuevo hilo.
  - Abre el socket.
  - Abre el bind.
  - Hace aceptar y espera al cliente.
- El 'servidor' manda un mensaje DCC.
- El 'cliente' acepta escribiendo /faccept.
- Abre un nuevo hilo.
  - Abre un socket.
  - Hace connect.
  - Empieza a recibir.
- El 'servidor' empieza a mandar.
- Cuando termina de enviar contenido, envía el mensaje “CLOSE” y cierra el socket.
- Cuando el 'cliente' recibe “CLOSE” cierra el fichero y termina la ejecución del hilo.

Mensajes generados:

- Mensaje de inicio de conexión: Lo manda el servidor, sirve para indicar al cliente que se va a iniciar la conexión para mandar un archivo, sera decisión del cliente indicar si lo acepta o lo rechaza. La estructura del mensaje es la siguiente
  - DCC SEND <usuario> <archivo> <IP> <puerto> <tamaño>
- Mensaje de fin de conexión: Lo manda el cliente al servidor, sirve para indicar al servidor que el cliente no desea el archivo, este mensaje cerrara el hilo del servidor.

Comandos de envío de ficheros:

- Comando “fsend”: Este comando tiene como argumento el nombre de un cliente y el nombre de un fichero. El formato es “/fsend <usuario> <archivo o ruta del archivo>. Este comando creara una mensaje que se enviara en este caso a un cliente. El sistema que mande el mensaje sera el servidor.
- Comando “faccept”: Este comando se realizara en la parte del cliente abrirá un hilo que actuara como cliente, si la conexión esta activa el servidor mandara el archivo al cliente. Es decir este comando confirma la conexión y empieza el intercambio de información. No genera ningún mensaje de respuesta a través del servidor de IRC. El formato del comando es el siguiente. /faccept <archivo> donde archivo es el nombre o la ruta con la que se guardara el nuevo archivo que nos mande el servidor.
- Comando “fcancel”: Cancela/no inicia la conexión con el servidor. El formato del comando es /fcancel <usuario> donde el usuario es el nick del usurario que nos esta mandando el archivo. Este mensaje generara un comando IRC DCC CANCEL que se mandara al otro usuario y que indicara si se cancela la conexión. Una vez recibido este mensaje el servidor cerrara el hilo de emisión.

Estructura del servidor:

El servidor recibe el comando fsend de la entrada de la interfaz, crea el mensaje DCC SEND y lo envía, además gurda en una variable global “args” de tipo argumentos que contiene, en este caso el nombre del archivo y el puerto. Esta información se usara en la función servidorArchivo() que se ejecutara usando un hilo. A parte de esto la conexión con el cliente es la típica conexión cliente-servidor sockect, bind, listen, accept. Una vez hecho este proceso y establecida la conexión con el cliente el servidor mandara el archivo completo al cliente.

Estructura del cliente:

El cliente se ejecuta con un hilo al realizar el usuario el comando /accept la información sobre el archivo y el servidor la ha recibido antes mediante un parser de mensajes del servidor. Este mensaje que recibe el cliente rellenan la estructura argumentos asignando el nombre del archivo, la dirección del servidor y el puerto. Una vez reconocido el accept se abrirá un hilo que ejecute el cliente, este se conectara al servidor e iniciara el intercambio del fichero.

Posibles fallos de implementación/seguridad:

- El servidor conecta con cualquier sistema que se conecte a ese socket, seria recomendable verificar que el cliente mandase su dirección al servidor mediante un mensaje y si la dirección que se mande coincide la dirección que solicita entonces iniciar la transmisión del fichero.
- El servidor debería de tener un mecanismo de timeout cuando pasa un tiempo sin que el cliente no responda, en este caso el servidor debería mandar un mensaje de timeout al servidor.
- Si se elige un nombre de fichero que ya existe no suelen funcionar bien, aveces otras no.
- Formatos que no sean de texto no van bien.
- No esta pensado que se envíen varios ficheros a la vez.