

Memoria Práctica 2

Redes de comunicaciones II

Enrique Cabrerizo Fernández Guillermo Ruiz Álvarez

Curso 2013 - 2014
Universidad Autónoma de Madrid

Índice

1. Introducción y descripción	3
2. Organización de directorios y ficheros.	3
3. Makefile	4
3.1. Directivas	4
3.2. Construcción del Makefile	5
4. Funciones de librería	5
5. Estructura, funcionalidad y pruebas del cliente	5
5.1. Conexión y desconexión	6
5.2. Envío de comandos y mensajes	6
5.3. Recepción de comandos y mensajes	6
6. Ampliación multimedia del cliente	6

1. Introducción y descripción

En este documento se detalla el proceso de implementación de un cliente IRC a partir del código proporcionado de una interfaz GTK. Para la elaboración del cliente se han seguido las directrices especificadas en el RFC 2812.

La interfaz gráfica de usuario se compone de varias partes. En primer lugar, el usuario puede introducir sus datos de conexión en los campos de texto y conectarse y desconectarse de un servidor utilizando los botones de conexión y desconexión. En segundo lugar, el usuario puede utilizar el campo de introducción de texto principal para enviar comandos al servidor y mensajes a los usuarios conectados en dicho servidor. Finalmente, el usuario puede utilizar los *checkboxes* para modificar los modos de un canal en el que tenga privilegios.

Toda la información recibida aparecerá en la pantalla principal de la interfaz.

2. Organización de directorios y ficheros.

En lo que respecta a la organización de los ficheros, se han añadido nuevos directorios modificando ligeramente la estructura:

El directorio raíz, cuyo nombre es G-2301-03-P2 contiene los siguientes directorios:

- **bin** contiene los ejecutables creados después de la compilación y el enlazado.
- **doc** contiene la documentación del proyecto en formato PDF.
- **includes** contiene los ficheros de cabecera generados para la elaboración de librerías y el programa principal.
- **lib** contiene las librerías generadas para realizar los programas principales.
- **man** contiene los manuales¹ de las funciones de las librerías. Para acceder a ellos basta ejecutarlos con el programa **man**.
- **obj** contiene los ficheros objeto generados en la compilación.
- **src** contiene los ficheros fuente que generarán un ejecutable. Estos son programas de prueba (tests) o los programas principales.
- **srclib** contiene los ficheros fuente que generarán los objetos de las librerías.

En el directorio raíz se encuentran el fichero Makefile necesario para la compilación y enlazado de los archivos además del fichero de configuración de Doxygen.

¹Estos manuales han sido generados con doxydoc de doxygen

3. Makefile

Para compilar y enlazar se proporciona un fichero Makefile a ejecutar con el programa **make**. En esta práctica se han añadido directivas que se presentan a continuación.

3.1. Directivas

Las directivas proporcionadas son las siguientes:

- **all** genera todos los binarios en el directorio `bin`². Estos ficheros son:
 G-2301-03-P1-main
 G-2301-03-P2-chat
y todos los binarios creados para realizar tests.
- **IRC-SERVER** crea el ejecutable del servidor de la primera práctica en el directorio `bin`.
- **IRC-CLIENT** crea el ejecutable del cliente de la segunda práctica en el directorio `bin`.
- **bin/*** donde `*` es el nombre de un ejecutable. En este caso se compilará todo lo necesario para compilar y enlazar para la obtención del ejecutable.
- **obj/*.o** donde `*` es el nombre de un objeto. En este caso se compilará lo necesario para compilar el objeto.
- **compress, pack, tar, tgz**. Cualquiera de estas directivas ejecutan el comando `clean` y comprimen la práctica a un fichero `tgz` que será colocado en el directorio padre del raíz de la práctica.
- **clean, clear**. Cualquiera de estas directivas eliminan el fichero `tgz`, los objetos y los ejecutables.
- **G-2301-03-P2-*.a** genera la librería `G-2301-03-P2-*.a`. las librerías creadas son
 G-2301-03-P2-libconnection.a
 G-2301-03-P2-libirc.a
 G-2301-03-P2-libaudio.a

²incluyendo los ficheros objeto necesarios para ello en el directorio `obj` y las librerías en el directorio `lib`

3.2. Construcción del Makefile

El archivo Makefile (los comandos utilizados son de GNU Make) se ha realizado siguiendo las siguientes reglas, de forma que puede ser utilizado por cualquier usuario que siga las mismas:

- Seguir la organización de directorios especificada.
- No existen ficheros de cabecera para los ficheros fuente de src.
- Todos los ficheros de srclib tienen un fichero de cabecera asociado.

Para cambiar el nombre de los directorios o añadir ficheros de cabecera extra, basta con modificar las macros al inicio del fichero Makefile.

4. Funciones de librería

En esta práctica se ha ampliado el número de librerías ofreciendo tres librerías diferenciadas según la utilidad de las mismas:

- **G-2301-03-P2-libconnection.a** ofrece todas las funciones necesarias de conexión, incluyendo el manejo de hilos, la daemonización y el uso de semáforos.
- **G-2301-03-P2-libirc.a** ofrece todas las funciones necesarias relacionadas con IRC. Incluyendo funciones de cliente y servidor.
- **G-2301-03-P2-libaudio.a** ofrece todas las funciones necesarias relacionadas con las llamadas y el manejo de audio.

La descripción precisa de todas estas funciones se encuentra en la páginas de manual (**man pages**) realizadas por los autores en el directorio correspondiente.

5. Estructura, funcionalidad y pruebas del cliente

El cliente implementado parte de la implementación de la interfaz gráfica en el fichero **G-2301-P2-chat.c**. Todos los eventos de dicha interfaz producen la ejecución de las funciones implementadas en **G-2301-P2-chat_funcs.c**. Se ha añadido un módulo auxiliar llamado **G-2301-P2-client_utility_functions.c** donde se implementa el hilo principal que recibe datos del servidor y las funciones necesarias para el envío de comandos y tratamiento de los datos recibidos.

El proceso y estructura de la implementación de la conexión, desconexión y el envío y recepción de comandos se describe a continuación.

5.1. Conexión y desconexión

Cuando el usuario desee conectarse a un servidor IRC es **necesario** que rellene la totalidad de los campos de información de usuario (Apodo, Nombre, Nombre real, Servidor³). En el caso de que el campo de puerto se deje vacío o se escriba IRC se utilizará el puerto predeterminado de IRC 6667. En el momento en el que se pulsa el botón de conexión se ejecuta una función implementada en **G-2301-P2-chat_funcs.c** que se encarga de manejar el evento. Dicha función realiza llamada a una función del fichero **G-2301-P2-client_utility_functions.c** que se encarga de realizar las acciones necesarias para establecer una conexión con el servidor.

En el caso de la desconexión, el evento se captura y se manda una petición de desconexión al servidor. Una vez recibida la respuesta, el cliente se desconecta.

5.2. Envío de comandos y mensajes

Para el envío de datos se distingue si la cadena introducida es un comando o un mensaje. Tras ello, una función de **G-2301-P2-client_utility_functions.c** envía el mensaje o identifica el comando introducido. En el caso de que el texto introducido sea un comando se realizará una llamada a una función de **G-2301-P2-client_commands.c** que realizará las operaciones necesarias para el correcto envío del comando. Entre estas operaciones se encuentra la modificación de parámetros de entorno para mantener el estado del programa.

5.3. Recepción de comandos y mensajes

Al igual que en el envío de los datos, para la recepción de los mismos, tras conectar con un servidor, se lanza un hilo que se encarga de recibir toda la información a través del socket de conexión con dicho servidor.

Una vez recibidos los datos, se identifica el comando/mensaje recibido y se realiza una llamada a una función implementada en **G-2301-P2-client_commands.c**, que se encarga de mostrar el mensaje, su significado o en su caso, modificar los parámetros de entorno necesarios.

6. Ampliación multimedia del cliente

Se compone la cabecera RTP a partir de dos funciones, `rtp_header_setup` que fija una serie de parámetros que van a ser fijos durante toda la conexión y `rtp_header_build` que se llama para completar la cabecera con los parámetros que varían para cada paquete (timestamp y número de secuencia). Se han creado una serie de funciones

³Puede utilizarse tanto la IP con la URL del servidor de chat.

que realizan una configuración de la llamada. Primero tenemos `setup_call` que crea un socket e inicializa un timeout para dicho socket. La función `call()` recibe una ip y un socket y comienza a grabar/enviar y recibir/reproducir los paquetes de audio que llegan de dicha ip y socket mediante dos hilos que se lanzan en el momento de establecer la llamada. Tal como manda el protocolo RTP, se comprueban el SSRC y se ignoran paquetes cuyo número de secuencia es inferior al último reproducido o cuyo formato sea diferente al acordado. Para terminar la llamada se llama a `end_call()` que cierra los hilos y libera los recursos dejando las variables listas para un nuevo `setup_call`.