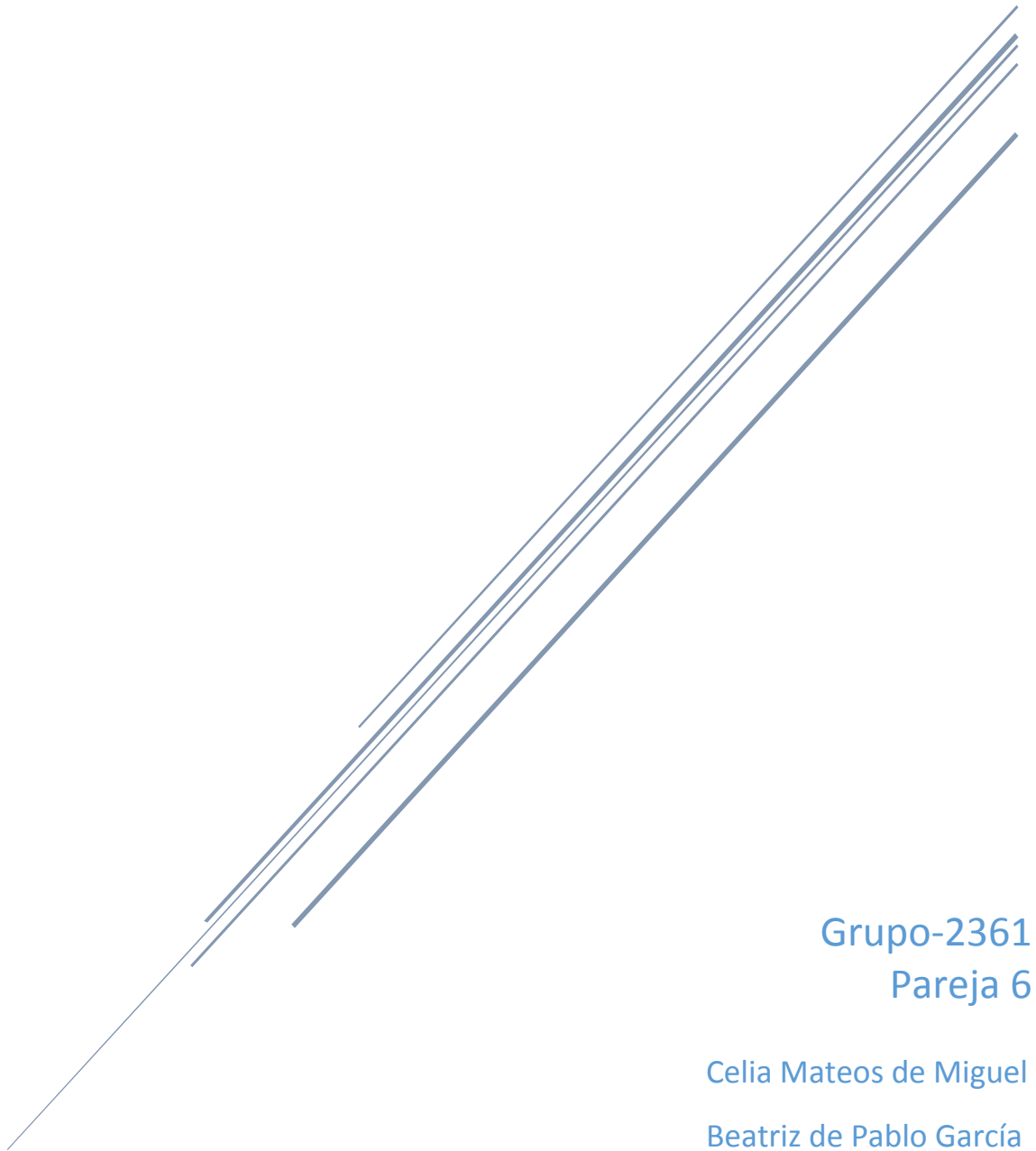


PRÁCTICA 2

El cliente IRC



Grupo-2361
Pareja 6

Celia Mateos de Miguel

Beatriz de Pablo García

Alfonso Sebares Mecha

Contenido

1. Introducción	2
2. Diseño.....	2
3. Funcionalidad IRC.....	4
4. Conclusiones Técnicas.....	4
5. Conclusiones Personales/Dificultades	4

1. Introducción

El objetivo de esta práctica ha sido la implementación del cliente IRC a partir del RFCs correspondientes. Se nos proporciona una interfaz gráfica sobre la que trabajar, añadiendo las funciones apropiadas (callbacks) para la interacción con un servidor IRC.

2. Diseño

A nivel de diseño, el cliente no presentaba demasiados problemas a nivel lógico: dado que nos conectamos únicamente con 1 sólo servidor a la vez, la gestión de hilos se simplifica notablemente. En nuestra implementación, hemos utilizado 3 hilos en total para gestionar los distintos aspectos del cliente:

- Un hilo que lee constantemente datos del buffer del socket mientras se mantenga la conexión.
- Un hilo para gestionar la transmisión de ficheros, estableciendo una conexión con el host que nos va a transmitir el fichero.
- Un hilo principal de interfaz.

Como se verá en la documentación del .c principal, haremos uso de globales que nos ha parecido que simplifican el diseño de estructuras de datos (Nick del usuario operando el cliente, descriptor del socket abierto con el servidor, último canal activo, etc). Aunque podíamos haber contenido estas variables en una estructura, no hemos considerado que añada más claridad, rendimiento o riesgo al programa.

Se incluye una librería implementada en el fichero G-2361-06-P2-logger.c con su fichero de pruebas que generará una carpeta con un archivo .log por ejecución. Todas las transmisiones por el socket con el servidor son registradas cronológicamente en texto plano, simplificando notablemente el debug de problemas asociados a la conexión y transmisión de mensajes.

También se incluye una versión mejorada de la librería utilizada para transmisión de datos por socket con conexión, lo que supone una mejora de legibilidad en el código con respecto a la práctica anterior.

Por último, se ha incluido nuestro diseño de cliente y servidor echo utilizados para el aprendizaje y prueba de conexión con sockets. Se pueden compilar tanto el cliente como el servidor con la directiva 'make extra'.

Respecto a la estructura de la carpeta, contiene los siguientes directorios y archivos:

- Carpeta **src**:
 - **G-2361-06-P2-xchat2.c** Fichero con la implementación de las funciones principales del cliente, directamente relacionadas con los callbacks.
 - **G-2361-06-P2-log_test.c** Fichero con una pequeña prueba de la librería de loggeo de eventos y errores mencionada.
 - **G-2361-06-P2-echo_client.c** Cliente echo
 - **G-2361-06-P2-echo_server.c** Servidor echo
 - **G-2361-06-P2-conexion_tcp_test.c**. Fichero sobre el que hemos realizado pruebas no persistentes de la librería de transmisión de datos por sockets.
- Carpeta **srclib**:
 - **G-2361-06-P2-conexión_tcp.c**. Implementación de la librería mencionada.
 - **G-2361-06-P2-aux_functions.c** Librería de funciones auxiliares de apoyo.
 - **G-2361-06-P2-user_commands.c** Librería de funciones de usuario.
 - **G-2361-06-P2-logger.c** Implementación de la librería de loggeo de eventos.
- Carpeta **includes**:
 - Aquí se encuentran cada uno de los .h que se corresponden a los distintos .c de la carpeta src y srclib.
- Carpeta **man**:
 - En esta carpeta se encuentra toda la documentación de cada uno de los ficheros generada con doxygen, así como esta memoria.

El resto de carpetas son para la generación de objetos durante la compilación y enlazado.

En el directorio raíz se encuentra el Makefile. Con hacer make se compila el servidor y se genera el ejecutable.

3. Funcionalidad IRC

En cuanto a la funcionalidad, se han implementado todos los comandos de usuario definidos en el fichero **includes/G-2361-P2-user_commands.h**

Se han intentado definir todas las posibles respuestas del servidor que garantizan una implementación funcional de un cliente IRC. En caso de que devuelva una respuesta o error no contemplado, aun así se mostrará por la ventana **System** a modo informativo.

Cabe destacar que no hemos implementado la transmisión de audio y la transmisión de ficheros no ha sido probada exhaustivamente.

4. Conclusiones Técnicas

El aspecto más interesante de esta práctica ha sido seguir una implementación bastante libre a partir de unas especificaciones de un protocolo con su RFC y observar qué aspectos están más abiertos a decisiones de implementación que otros.

5. Conclusiones Personales/Dificultades

Destacar algunas complicaciones y lo aprendido de ello:

- La programación eficiente en hilos no es para nada trivial. Habría sido interesante probar una implementación más eficiente basada en select y multiplexación.
- De volver a hacer un proyecto similar, nos habríamos planteado una estructuración mucho más modular desde el principio.
- El control de versiones en un proyecto de este tamaño y con 3 personas realizando ediciones, es fundamental.
- Hemos encontrado complicaciones a la hora de debuggear errores. Aunque inicialmente probamos a trabajar con un entorno, por la naturaleza de la interfaz gráfica y los hilos rápidamente acabamos utilizar prints, cosa que no es óptima en el desarrollo.