Memoria Práctica 1
Redes de comunicaciones II

Enrique Cabrerizo Fernández Guillermo Ruiz Álvarez

Curso 2013 - 2014
Universidad Autónoma de Madrid

Índice

1. Introducción y descripción

En este documento se detalla el proceso de elaboración de un servidor IRC que sigue las directrices especificadas en el RFC 2812.

El servidor se ejecutará en modo daemon (background) en sistemas Linux. El desarrollo se ha realizado sobre dos máquinas con Ubuntu con el kernel 3.02 y 3.12.[1].

Las pruebas del funcionamiento del servidor y todos los comandos que acepta han sido realizadas sobre `telnet` y xchat. Dichas pruebas se detallarán más adelante en este mismo documento.

Para realizar este proyecto, los autores han realizado una librería referente a conexiones entre procesos e hilos que también será detallada.

A continuación se explicará la organización de directorios y ficheros, que es crucial para el correcto funcionamiento del programa `make` con el Makefile proporcionado.

2. Organización de directorios y ficheros.

Los ficheros del proyecto se han organizado de la siguiente forma:

El directorio raíz, cuyo nombre es `G-1301-03-P1` contiene los siguientes directorios:

- **includes** contiene los ficheros de cabecera generados para la elaboración de librerias y el programa principal.
- **lib** contiene la libreria generada para realizar el programa principal.
- **man** contiene los manuales[2] de las funciones de la librería. 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. Típicamente son programas de prueba (tests) o el programa principal.
- **srclib** contiene los ficheros fuente que generarán los objetos de la librería.

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

[1] No se asegura el funcionamiento correcto del daemon en máquinas con Ubuntu y un kernel superior al 3.10

[2] Estos 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`.

3.1. Directivas

Las directivas proporcionadas son las siguientes:

- **all** genera todos los binarios en el directorio raíz[3]. Estos ficheros son:

  G-1301-03-P1-main

  G-1301-03-P1-semaphores_test

  G-1301-03-P1-connection_test

  G-1301-03-P1-daemonize_test

  G-1301-03-P1-irc_split_test

- **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 objeto y los ejecutables.
- **G-1301-03-P1-libnetworks.a** genera la librería `G-1301-03-P1-libnetworks.a`.

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 incio del fichero Makefile.

[3] incluyendo los ficheros objeto necesarios para ello

4. Funciones de librería

Para desarrollar el proyecto, se han realizado ciertas funciones independientes al mismo. Estas funciones son, en su mayoría, genéricas e independientes del servidor y nos permiten manejar los siguientes escenarios.

- **G-1301-03-P1-daemonize** ejecutar un programa en modo daemon.
- **G-1301-03-P1-connection** manejar conexiones TCP.
- **G-1301-03-P1-parser** parsear cadenas.
- **G-1301-03-P1-semaphores** manejar semáforos y problemas de concurrencia de lectura y escritura.
- **G-1301-03-P1-thread_handling** manejar hilos asociados a conexiones.

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

Se han usado dos ficheros de cabecera externos para el manejo de tablas hash y listas enlazadas:

- uthash
- utlist

4.1. Pruebas

Las pruebas correspondientes a estas librerias son aquellas implementadas en el directorio src cuyo nombre sigue la expresión regular

G-1301-03-*_test.c

- **G-1301-03-P1-semaphores_test** lanza varios hilos que leen concurrentemente un dato compartido e intentan escribir sobre otro. La escritura trata de agregar a una cadena de caracteres un número. El programa se ejecuta correctamente y la escritura se hace en orden y sin condiciones de carrera ni interbloqueos.
- **G-1301-03-P1-connection_test** acepta conexiones en un puerto concreto y devuelve la misma cadena de texto que recibe.
- **G-1301-03-P1-daemonize_test** Ejecuta un proceso en modo background y espera una señal para terminar. Se comprueba que el padre del proceso sea INIT (salvo en nuevas versiones del kernel de ubuntu).

- **G-1301-03-P1-irc_split_test** Comprueba el reconocimiento de patrones de comandos de IRC tal y como vienen especificados en el RFC indicado anteriormente. Comprueba funcionalidades añadidas (no especificadas en el RFC) como el hecho de que los argumentos se separen correctamente ignorando multiples espacios.

5. Estructuras, comandos y pruebas del servidor

Para realizar el servidor se han utilizado estructuras que organizan los datos usando principalmente listas enlazadas y tablas hash.

El servidor se ha definido de la siguiente forma

```
struct {
    channel* channels_hasht;
    user* users_hasht;
    int readers_num, readers, writer;
    int mutex_access, mutex_rvariables;
} server_data;
```

En primer lugar, se tienen dos tablas indexadas para todos los usuarios y canales del servidor y una serie de variables necesarias para el uso de semáforos.

Los usuarios se identifican por el socket de la conexión, nick, nombre de usuario, y el resto de especificaciones que impone el RFC. Además, cada usuario dispone de banderas de modos y estados y una lista de los canales a los que pertenece. Los modos para canales que se han implementado son los siguientes:

- **+oO** Para hacer squit y ser operador en todos los canales.
- **+i** Modo invisible.

Los canales se identifican por un nombre, tema, contraseña, banderas de estados y el resto de especificaciones que impone el RFC. Además, cada canal dispone de una lista de usuarios y de operadores que pertenecen a dicho canal, así como una lista de invitados. Los modos para canales que se han implementado son los siguientes:

- **+k password** para poner contraseña a un canal.
- **+l limit** para poner límite de usuarios a un canal.
- **+i** modo invitación.

- **+o nick** para hacer operador del canal al usuario con nick.
- **+t** para que solo los operadores del canal puedan cambiar el topic.
- **+n** para que solo puedan mandar mensajes al canal los usuarios del canal

Se han realizado varias funciones de tratamiento de usuarios, canales y búsqueda de los mismos en el servidor, todos disponibles en el fichero de cabecera correspondiente.

Cada usuario es atendido por un hilo independiente, que se encargará de realizar las operaciones necesarias para la correcta ejecución de los comandos. El uso de semáforos tiene el objetivo de evitar el problema de accesos múltiples a secciones críticas de escritura o acceso de escritores mientras haya hilos en proceso de lectura. El problema se ha tratado como un problema de lectores-escritores dando prioridad a escritores.

5.1. Comandos

Los comandos implementados son los siguientes:
PING, MODE, NICK, PASS, USER, PRIVMSG, NAMES, JOIN, LIST, TOPIC, WHO, PART, OPER, INVITE, QUIT, SQUIT[4].

5.2. Pruebas

Tanto la conexión al servidor como la correcta ejecución de todos los comandos implementados han sido probados en `telnet` y `xchat` ofreciendo los resultados esperados.

[4]El usuario y contraseña para ser operador IRC y poder ejecutar este comando son `someuser` y `somepass`, respectivamente