

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	3
2. Organización de directorios y ficheros.	3
3. Makefile	4
3.1. Directivas	4
3.2. Construcción del Makefile	4
4. Funciones de librería	5
4.1. Pruebas	5
5. Estructuras, comandos y pruebas del servidor	6

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 librerías y el programa principal.
- **lib** contiene la librería generada para realizar el programa principal.
- **man** contiene los manuales² 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.

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

²Estos 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 raíz³. 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/*.** 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 `src/lib` 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.

³incluyendo 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 librerías 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.