

Manual del programa corrector de prácticas R2D2

Práctica 1 - Servidor IRC

Oscar Delgado



Este documento describe el uso y características principales del corrector de la práctica 1, que implementa un servidor IRC, de la asignatura Redes II.

1 Uso básico

El corrector R2D2 es una aplicación nativa para Linux, que debe ejecutarse desde la línea de comandos. Para ello basta abrir una terminal del sistema y ejecutar el comando `r2d2`. Para solicitar ayuda, y obtener un listado de todas las funcionalidades del programa, se debe utilizar el siguiente comando:

```
# r2d2 --help
```

Aunque la mayoría de las funcionalidades deberían ser autoexplicativas, en las siguientes secciones se analizarán como mayor detalle las más significativas.

Otra opción muy útil, que utilizarás a menudo en el desarrollo de la práctica, es `--verbose`, que muestra una traza de ejecución detallada de cada prueba realizada, y de la respuesta devuelta por el servidor. Esta opción puede añadirse a cualquier otra.

1.1 Servidor y puerto

Estas opciones, señalizadas con `--servidor <IP servidor>` y `--puerto <puerto servidor>`, respectivamente, sirven para indicar al corrector una dirección IP y puerto específicos sobre los que lanzar las pruebas. Por defecto, si no se indica lo contrario con estos valores, el corrector se ejecuta contra la dirección IP `127.0.0.1`, o `localhost`, y puerto `6667`.

1.2 Lista de pruebas

Con la opción `--lista-tests` el corrector muestra una lista completa de las pruebas disponibles, junto con su número de identificación, `num`, y una pequeña explicación de cada una:

```
# r2d2 --lista-tests
```

Si se desea mayor información sobre una prueba específica, puede utilizarse la opción `-info-test <num>`. Por ejemplo, para obtener una explicación detallada de la prueba 1, denominada `TestComandoJoin`, basta con ejecutar:

```
# r2d2 --info-test 1
```

1.3 Selección y lanzamiento de pruebas

Una vez informados sobre las distintas pruebas existentes, y su número de identificación, se debe utilizar la opción `--tests <rango>` para seleccionarlas y ejecutarlas. El parámetro `rango` funciona como habitualmente: es posible seleccionar valores discretos, separados por comas, rangos, separando los extremos por guiones, y, finalmente, combinar ambas opciones. Algunos ejemplos de selección son los siguientes:

```
#r2d2 --tests 1,3,5    - Ejecuta las pruebas 1, 3 y 5
#r2d2 --tests 1-5      - Ejecuta todas las pruebas entre 1 y 5
#r2d2 --tests 1,3,5-8  - Ejecuta las pruebas 1, 3, 5, 6, 7 y 8
```

Por último, es posible también utilizar una serie de alias, **basicos**, **adv** y **errores**, que seleccionan los tests denominados **básicos**, **avanzados** y **errores**, respectivamente:

```
#r2d2 --tests basicos
#r2d2 --tests adv
#r2d2 --tests errores
```

La siguiente sección amplía la información sobre cada tipo de prueba.

2 Tipos de pruebas y evaluación

A grandes rasgos, existen tres tipos de pruebas, básicas, avanzadas y de gestión de errores, que, sobre el total de 6 puntos, suponen 3, 2 y 1, respectivamente.

Excepto la prueba del empaquetado, que se explica con detalle en 2.2, no hay ninguna prueba cuya superación sea imprescindible en sí misma. Sin embargo, en la práctica, existe una serie de dependencias entre ellas que hará imposible aprobar la práctica si todas o la mayoría de pruebas básicas no se superan correctamente.

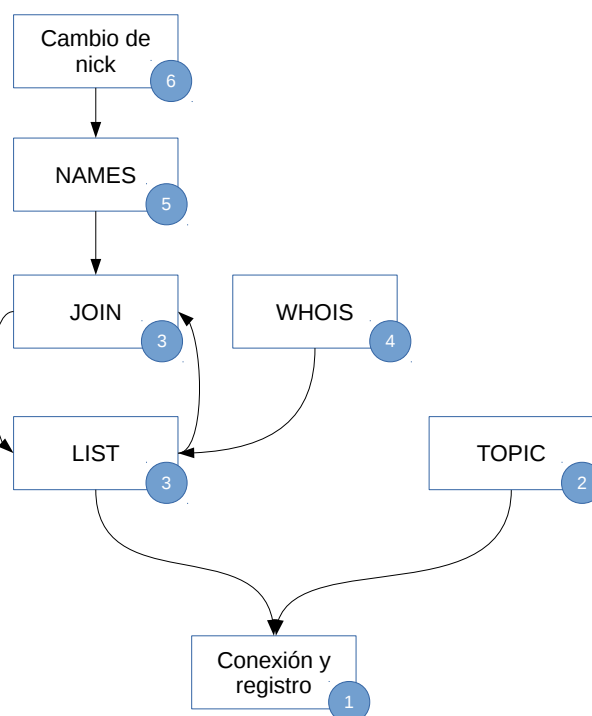
2.1 Orden propuesto de implementación

El orden normal de ejecución de las pruebas es el que se obtiene a través de la opción `--lista-tests`, por lo que es aconsejable implementar los distintos comandos IRC aproximadamente en ese orden. Además, existen una serie de interdependencias entre las pruebas, de forma que algunas no podrán ejecutarse correctamente hasta que no lo hagan las anteriores.

En la figura adyacente puede observarse estas relaciones y el orden recomendado de implementación de los distintos comandos:

- 1.- Funciones de conexión y registro, que incluyen los comandos USER y NICK.
- 2.- Comando TOPIC.

Fig. 1: Interdependencias y orden recomendado (círculos azules) de implementación de los comandos del servidor IRC



- 3.– Comandos LIST y JOIN. Ambos se necesitan mutuamente en las pruebas, por lo que habría que implementarlos simultáneamente.
- 4.– Comando WHOIS.
- 5.– Función de cambio de nick.

Una vez implementados estas funcionalidades básicas, el orden de implementación del resto de funcionalidades del servidor IRC es irrelevante.

2.2 Prueba de empaquetado

La prueba de empaquetado es una prueba básica especial: no puntúa pero **su ejecución correcta es imprescindible para poder entregar y evaluar la práctica**. Es decir, si una práctica no supera esta prueba, aunque su puntuación de ejecución sea superior a 3, se considerará **suspensa**.

Esta prueba es muy sencilla, y únicamente verifica que en el mismo directorio desde donde se ejecuta el corrector, existen dos ficheros llamados **autores.txt** y **G-CCCC-NN-P1.tar.gz**, cada uno de ellos de acuerdo al formato dado en las siguientes secciones.

Formato del fichero 'autores.txt'

Este es un fichero de texto plano que contiene información sobre los autores de la práctica. Su formato es el siguiente:

```
G-CCCC-NN-P1
NIA#Apellido1 Apellido2, Nombre
...
NIA#Apellido1 Apellido2, Nombre
```

Formato del fichero de la práctica

Como ya se comentado, el nombre de este fichero debe ser **G-CCCC-NN-P1.tar.gz**, donde CCCC es el número de la clase de prácticas y NN es el número del grupo **siempre con dos dígitos**.

Este fichero debe crearse comprimiendo en formato `.tar.gz`¹ un directorio, también con nombre G-CCCC-NN-P1, y que debe contener una estructura de directorios y archivos también fija:

- 1.– Un directorio `src`, donde se almacenarán los fuentes.
- 2.– Un directorio `src/lib`, donde se almacenarán los fuentes que se vayan a acompilar para una librería.
- 3.– Un directorio `includes`, donde se guardarán los includes (`.h`)
- 4.– Un directorio `lib`, donde se almacenarán las librerías (`.a`).
- 5.– Un directorio `obj`, donde se almacenarán los objetos (`.o`) que deberá estar vacío en el comprimido.
- 6.– Un directorio `man`, donde se almacenarán los ficheros para el comando `man`.

¹Para más información, se puede estudiar el comando `tar` con el comando `man`). Es importante comprobar que si se descomprime en cualquier directorio este fichero, aparece el directorio base de la práctica. No se admitirá ningún otro tipo de compresión.

7.– Un directorio `doc`, donde se almacenarán los ficheros de documentación.

8.– Dos ficheros en el directorio raíz: el mismo fichero `autores.txt` anterior, y el `Makefile`.

En general, los nombres de los ficheros en el interior de los directorios anteriores son de libre elección del estudiante, excepto en los siguientes casos:

- La memoria de la documentación deberá llamarse `G-CCCC-NN-PX-doc.pdf`. Este nombre implica, por supuesto, que toda la documentación debe entregarse en formato PDF.
- Los ficheros para el comando `man` deben llevar los nombres adecuados para que funcionen correctamente.
- La librería deberá llevar un nombre propio de una librería y es aconsejable que sea una única librería para todas las prácticas.

Por último, debe tenerse en cuenta que **bajo ningún concepto deberán incluirse ejecutables en el archivo comprimido**. Para ello, el *makefile* deberá borrar los ejecutables cuando se solicite la compresión.

3 Entregable

Sólo debe entregarse el archivo comprimido **G-CCCC-NN-P1.tar.gz** a través de Moodle con el formato adecuado, y habiendo confirmado que la práctica pase correctamente la prueba de empaquetado.