

# Entorno de trabajo

Autómatas y Matemáticas Discretas  
Escuela de Ingeniería Informática  
Universidad de Oviedo  
curso 2017-2018

## 1. Breve introducción

En lo sucesivo, las prácticas de la asignatura de *Autómatas y Matemáticas Discretas* se realizarán sobre el host *ritchie*, dicha máquina tiene instalado el sistema operativo *linux* (concretamente una distribución Debian GNU/Linux 8) y su dirección IP es **156.35.94.1**.

Como los ordenadores de los laboratorios tienen instalado Windows, para abrir una sesión en *ritchie* es preciso conectarse a él mediante un cliente *ssh*. El que utilizaremos es *putty*, que está instalado en los equipos de los laboratorios y que también se puede descargar desde la dirección <http://www.putty.org>.

Para autenticarnos, *ritchie* nos pedirá nuestro usuario (por ejemplo UO123456) y nuestra password. Esto nos permite abrir una sesión en modo consola o terminal, algo similar al cmd (command) de Windows, pero infinitamente más potente. La primera vez que nos conectemos, el sistema nos pedirá que cambiemos nuestra password, debemos leer atentamente los mensajes y proceder tal y como nos pide.

Una vez abierto el terminal el sistema indica que está a la espera mostrando una línea similar la siguiente:

```
userlogin@host:~$
```

y en nuestro caso, más concretamente, algo como:

```
UO123456@ritchie:~$
```

donde indica que el usuario **UO123456** está conectado al host **ritchie**, situado en el directorio `~` (que es una abreviatura del directorio *home* del usuario; es decir el directorio `/home/students/UO123456/`); y que se está a la espera de un comando que escribiremos a partir del `$`. De hecho podemos probar a ejecutar el comando `pwd` que nos informa de nuestro directorio actual.

Muchos usuarios no están acostumbrados a trabajar en modo texto, ya que casi todos los sistemas operativos hacen uso de potentes interfaces gráficas. *linux* también las utiliza, pero en muchas ocasiones un terminal en modo texto es mucho más eficiente. Lo cierto es que la combinación de herramientas en modo texto con otras en modo gráfico es la mejor de las opciones.

## 2. El terminal de linux

En esta sección vamos a ver algunos comandos útiles que podemos introducir en nuestro intérprete de comandos (nuestro terminal).

En *linux*, las operaciones de creación, edición, copia, borrado, ejecución, etc. de ficheros y en general todas las órdenes del sistema operativo se realizan a través de nuestro terminal, aunque evidentemente también existen herramientas gráficas para hacerlo.

La estructura típica de un comando es:

```
$ comando opciones arg1 arg2 ...
```

es decir, el nombre del comando seguido de cero o más argumentos separados por espacios. Los primeros argumentos pueden ser *opciones* que modifican ligeramente el comportamiento del comando tecleado. Las opciones son, usualmente, argumentos que se distinguen por comenzar por un guión seguido de una letra.

Algunos comandos imprescindibles para manipular ficheros en *linux* son los siguientes:

| Comando                          | Significado   |
|----------------------------------|---|
| <code>ls</code>                  | Listado del contenido de un directorio                    |
| <code>ls -l</code>               | Listado largo del contenido de un directorio              |
| <code>cat arg1</code>            | Muestra el contenido del fichero <code>arg1</code>        |
| <code>joe arg1</code>            | Editar el fichero de texto <code>arg1</code>              |
| <code>cp arg1 arg2</code>        | Copia <code>arg1</code> en <code>arg2</code>              |
| <code>mv arg1 arg2</code>        | Renombra (o mueve) <code>arg1</code> en <code>arg2</code> |
| <code>rm arg1 arg2 ...</code>    | Borra los argumentos dados                                |
| <code>rm -r arg1 arg2 ...</code> | Borra recursivamente (útese con MÁXIMA precaución)        |
| <code>pwd</code>                 | Muestra la ruta del directorio actual                     |
| <code>cd dir</code>              | Cambiar al directorio <code>dir</code>                    |
| <code>mkdir dir</code>           | Crear el directorio <code>dir</code>                      |
| <code>rmdir dir</code>           | Borrar el directorio <code>dir</code> (si está vacío)     |
| <code>man comando</code>         | Página de manual de <code>comando</code>                  |

Por último, si es preciso repetir o alterar comandos ya tecleados, no hace falta escribirlos de nuevo en su integridad. Pulsando la tecla de *cursor hacia arriba* recorreremos hacia atrás el histórico de los comandos que hemos ejecutado, y podemos alterarlos y repetirlos.

Un aspecto importante a tener en cuenta es que nuestro intérprete de comandos es *case-sensitive*, es decir que es sensible a mayúsculas y minúsculas. Es decir si pretendemos ver el contenido de nuestro directorio mediante el comando *LS* nos dirá que esa orden no se ha encontrado con un mensaje como:

*-bash: LS: no se encontró la orden*

en este caso comprobamos que el intérprete que estamos utilizando es el *bash*, uno de los más utilizados en *linux*.

Los ficheros en *linux* se organizan en un sistema de ficheros en forma de árbol único (no existen unidades ni volúmenes). El directorio raíz del sistema se llama `/` y de éste cuelgan algunos ficheros y directorios, los cuales a su vez pueden contener ficheros y directorios hasta aburrirnos. Un fichero queda determinado por su trayectoria en el árbol, que se describe por los directorios que conducen a él separados por barras, por ejemplo: `/home/students/U0123456/pruebas/fichero.txt`

Si un nombre de trayectoria no empieza por `/` se considera un camino *relativo* al directorio actual. El nombre `..` designa, en especial, al directorio padre. Así `../pruebas2/fichero.txt` será el fichero `fichero.txt` que reside en el directorio `pruebas2` hermano de mi directorio actual. Ahora vamos a probar algunos comandos.

1. Ver el contenido del directorio actual con los comandos `ls` y `ls -l`. Observar las diferencias.
2. Cambiarse al directorio raíz.
3. Observar los ficheros que contiene. Ver también sus tamaños, propietario y fecha.
4. Cambiarse a `/home/students`
5. Observar los directorios que hay.
6. Intentar cambiarse a alguno de esos directorios u observar su contenido
7. Volver al directorio origen con el comando `cd` sin argumentos.
8. Crear un subdirectorio de nombre *pruebas*
9. Cambiarnos al subdirectorio *pruebas*

10. Comprobamos que nos encontramos en el subdirectorio *pruebas*
11. Crear un fichero `HelloWorld.java`. La forma de crear y modificar ficheros de texto es usar un editor (no un procesador de textos). En *linux* existen numerosos editores con diferentes características: *vi*, *emacs*, *pico* y *joe* son algunos de los más populares. Por lo sencillo e inmediato de su manejo, recomendamos el uso de *joe*. En la siguiente dirección puedes encontrar un pequeño manual:

`http://joe-editor.sourceforge.net/manpage.html`

Para crear el fichero utilizando el editor *joe* ejecutaremos:

```
U0123456@ritchie:/home/U0123456/pruebas$ joe HelloWorld.java
```

Ahora le añadimos las siguientes líneas.

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
```

NOTA: En el directorio `/var/asignaturas/AMD/practicas/Entorno` dispones de una versión del fichero `HelloWorld.java` que puedes utilizar y que además está autodocumentada.

Como puede verse en la barra superior nos indica que para acceder a la ayuda del editor debemos teclear un *Ctrl-K H*. Seguidamente se nos muestra una pequeña ayuda en el propio terminal con las que podremos realizar las funciones básicas.

Ahora salvaremos el fichero y volvemos a nuestro intérprete de comandos.

12. Compilar nuestro programa mediante la orden:

```
U0123456@ritchie:/home/U0123456/pruebas$ javac HelloWorld.java
```

13. Ejecutar nuestro programa mediante la orden:

```
U0123456@ritchie:/home/U0123456/pruebas$ java HelloWorld
```

14. Nuestro intérprete es mucho más listo de lo que podemos imaginar. Una característica que nos resultará de gran ayuda es el *autocompletar*. Para ello únicamente tenemos que pulsar el tabulador para que el intérprete nos muestre o complete las posibilidades que tenemos. Como ejemplo teclea *jo* y después pulsa dos veces el tabulador. Nos mostrará las opciones de comando que empiezan por *jo*, si tecleamos ahora *e He* y volvemos a pulsar el tabulador, el intérprete nos completará el comando con un *joe HelloWorld.java* y podemos volver a editar nuestro fichero.

15. Hacer una copia (cp) de `HelloWorld.java` a `borrame1`

16. Hacer otra copia (cp) de `HelloWorld.java` a `borrame2` (acuérdate del autocompletar)

17. Mover (mv) el fichero `borrame1` al directorio superior (..)

18. Borrar (rm, o casi mejor `rm -i`) todos los fichero que empiezan por *bo*, para ello se puede utilizar el comodín (\*), es decir, aquellos cuyo nombre sea *bo\**

19. Crear un directorio (mkdir) `basura`

20. Borrar el directorio (`rmdir`) **basura**
21. Crear en tu directorio por defecto, (directorio `'/home/students/UOXXXXXX'`) un subdirectorio **practicasAMD**
22. Colócate en dicho directorio y copia en él el contenido del directorio de la primera práctica:  
`/var/assignaturas/AMD/practicas/Entorno`

### 3. WinSCP: compartiendo archivos entre Windows y linux

Una de las cosas que también necesitamos ser capaces de hacer es compartir e intercambiar ficheros desde *ritchie* a nuestro ordenador del laboratorio, o a cualquier otro dispositivo. Una de las formas más sencillas para conseguir esto es utilizar el programa **WinScp**. Este programa que ejecutaremos desde Windows nos permitirá acceder a nuestros ficheros de *ritchie* utilizando un protocolo de intercambio de ficheros seguro (encriptado).

WinSCP debe estar ya instalado en el ordenador del laboratorio, pero si no fuera así o quisieras instalar en un ordenador de tu casa puedes descargar el programa desde <http://winscp.net/eng/download.php> en cualquiera de las versiones.

Ahora nos conectaremos a nuestro sistema de fichero de *ritchie*. Para ello ejecutamos WinSCP y veremos que nos pide un host, que en nuestro caso será la IP de *ritchie* (156.35.94.1), un usuario (ejemplo: UO123456) y una password. El resto de valores dejaremos los valores por defecto. A partir de aquí nos conectaremos a *ritchie* y se abrirá una ventana en cuya parte izquierda tendremos nuestro sistema de archivos de Windows y en la derecha el de *linux*. En este punto ya podremos copiar archivos desde un sistema al otro.

### 4. Editando y ejecutando

Ahora ya estamos en disposición de establecer un método de trabajo cómodo para el resto del curso. Nuestro objetivo es poder editar un fichero con un editor gráfico desde Windows (olvidamos el *joe*) y luego poder ejecutarlo en linux. Para probar este método vamos a modificar y ejecutar nuestro fichero **HelloWorld.java**.

#### 4.1. Editando un fichero de linux con WinSCP

Si recuperamos la conexión a nuestros directorios de *ritchie* desde WinSCP podemos ver que en el subdirectorio *pruebas* aún conservamos el fichero **HelloWorld.java**. Si lo seleccionamos, mediante el botón de la derecha nos permitirá editarlo. Esto nos abrirá una nueva ventana con un editor propio del WinScp muy sencillito. Pero podemos configurar el WinScp para utilizar cualquier otro editor que sea más potente, Un buen ejemplo es el **Notepad++**, que permite editar simultáneamente varios ficheros, resaltar aspectos sintácticos como el balanceo de paréntesis, numeración de líneas y otras ventajas.

Para poder utilizar este editor, u otro de nuestro gusto, ir al menú *Options/Preferences*. Seguidamente nos aparecerá una ventana de preferencias y en el apartado de *Editors* añadir (*Add*) un editor externo (*External editor*) especificando el path donde se encuentra dicho editor. Una vez hecho esto podremos elegir el **Notepad++** para editar nuestros ficheros. Y si lo colocamos en la primera posición de la lista de editores (Subir) será el editor que utilice WinScp por defecto, que es lo que nos interesa.

Ahora vamos a modificar el fichero: donde ponía *Hello, World* lo cambiamos por *Hola, Mundo*; y salvamos el fichero mediante el icono correspondiente. Es posible que en este momento se abra una ventana que nos pida la contraseña de *ritchie*, pero ojo, alguna veces dicha ventana queda oculta, tendrás que buscarla. En el momento en el que salvemos nuestro fichero en *ritchie*, ya podemos volver a compilarlo y ejecutarlo, como se explica en la siguiente sección.

#### 4.2. Ejecutando en linux

Por otro lado en nuestra sesión *ssh* del *putty* nos colocamos en el directorio *pruebas* y ejecutamos de nuevo los comandos:

```
UO123456@ritchie:/home/UO123456/pruebas$ javac HelloWorld.java
```

```
U0123456@ritchie:/home/U0123456/pruebas$ java HelloWorld
```

Observamos el resultado. Ahora, de nuevo, volvemos a la ventana del editor del WinSCP y cambiamos 'Hola, Mundo' por 'Hola, Mundo Cruel' y salvamos; volvemos al *putty* y ejecutamos de nuevo veremos que las modificaciones han tenido éxito.

### 4.3. Creando los directorios para las prácticas

Aunque la estructura de directorios la podrías crear desde el propio WinScp, vamos a hacerlo desde la terminal de linux que nos resultará más rápido. Para ello colócate en tu directorio *home* mediante `cd ~` y ejecuta los comandos:

```
U0123456@ritchie:~$ mkdir -p practicasAMD/LexicalAnalysis/Training
U0123456@ritchie:~$ mkdir -p practicasAMD/LexicalAnalysis/ScannerMiniLan
U0123456@ritchie:~$ mkdir -p practicasAMD/SyntacticAnalysis/Training
U0123456@ritchie:~$ mkdir -p practicasAMD/SyntacticAnalysis/ParserMiniLan
```

Ahora comprueba que has creado correctamente todos los directorios.