

Shell de Unix | ProgPLN

Víctor Peinado v.peinado@filol.ucm.es

9 de octubre de 2014

Comandos de la shell para gestión de directorios y ficheros

Cuando abrimos una shell (o línea de comandos) en un sistema tipo Unix, nos encontramos con una interfaz como la siguiente:

```
user@mypc:~$
```

En este caso, estamos trabajando en un ordenador que se llama mypc y nuestro nombre de usuario es user.

- El ordenador suele indicar, mediante un cursor que parpadea, que queda a la espera de instrucciones.
- Esta interfaz es muy potente y nos permite (si conocemos los comandos adecuados) ejecutar y manipular cualquier proceso del sistema operativo.
- Nosotros nos conformaremos con menos ya que solo nos interesan los comandos o instrucciones para gestionar directorios, ficheros y procesar texto.

¿Dónde estoy?

pwd (print working directory)

Para que el sistema no indique en qué directorio estamos actualmente ejecutamos el comando pwd.

```
user@mypc:~$ pwd  
/home/user
```

En los sistemas de tipo Unix, las unidades de disco no utilizan letras (en Windows, las distintas unidades se llaman C:, D:, A:, etc). La estructura jerárquica de directorios cuelga de una única raíz, indicada con la barra /.

El directorio de trabajo de cada usuario (el equivalente a Mis Documentos en Windows) cuelga siempre de /home.

Las versiones modernas de la shell (como las que encontramos en Linux o MacOS) muestran habitualmente el directorio en el que estamos trabajando. Sin embargo, a menudo resulta útil pedirle al sistema que nos indique el directorio de trabajo actual.

Rutas absolutas y rutas relativas

Las rutas que apuntan a ficheros (*paths*) pueden ser de dos tipos: absolutas o relativas.

Las rutas absolutas parten desde el directorio raíz / e indican la jerarquía completa de directorios hasta el lugar de destino: /home/user/Ling/sintaxis/ejercicio1.doc.

Las rutas relativas parten desde el directorio de trabajo actual, sea el que sea en cada momento. Si estamos en nuestro directorio de usuario: Ling/sintaxis/ejercicio1.doc.

En la rutas relativas, utilizamos un par de símbolos especiales para representar posiciones.

- Un punto . indica el directorio actual.
- Dos puntos seguidos .. indican el nivel inmediatamente superior en la jearquía de directorios.

En ambos casos, separamos los directorios por barras /.

`ls [ARGUMENTOS] [RUTA] (list)`

Para lista los nombres de los directorios y carpetas del directorio actual de trabajo, ejecutamos el comando `ls`.

Si lo ejecutamos sin especificar ruta, se asume que el directorio a listar es el directorio actual de trabajo:

```
user@mypc:~$ ls
contratos facturas
```

Si lo ejecutamos especificando el nombre de un directorio, nos listará su contenido:

```
user@mypc:~$ ls contratos
gas internet luz
```

Si ejecutamos `ls` sin especificar ningún argumento, la información que nos muestra es escasa. Es más habitual solicitar un listado de directorios en versión larga (usando el argumento `ls -l`), que proporciona información sobre tipos de ficheros, permisos, usuarios y grupos, tamaño y fecha de la última modificación:

Si listamos el directorio `contratos`, comprobamos que contiene tres directorios `gas`, `internet`, `luz` y un fichero de texto llamado `notas.txt`.

```
user@mypc:~$ ls -l contratos
drwxrwxr-x 2 user group 4096 2014-05-02 17:29 gas
drwxrwxr-x 2 user group 4096 2014-05-02 17:29 internet
drwxrwxr-x 2 user group 4096 2014-05-02 17:29 luz
-rw-rw-r-- 1 user group 130 2014-05-02 17:32 notas.txt
```

Al ejecutar `ls -l` se nos muestran información completa sobre los ficheros.

En la mayoría de las shells modernas, podemos teclear directamente el *alias* `ll` en lugar del comando y el argumento `ls -l`. Ambas instrucciones nos permiten ejecutar la versión larga de un listado.

- La 1ª columna indica los distintos permisos de lectura (r), escritura (w) y ejecución (x).
- Si el fichero es un directorio, la primera letra es siempre una d. Si es un fichero normal, es -.
- La 3ª y 4ª columna indican el propietario y el grupo de usuarios que tiene acceso a estos ficheros.
- La 5ª columna indica el tamaño en bytes. Los directorios, independientemente de su contenido, siempre *marcan* el mismo tamaño.
- Las columnas 6ª y 7ª indican la fecha y la hora de la última modificación.
- La 8ª columna indica el nombre del fichero.

¿Cómo me muevo?

`cd [RUTA/]DIRECTORIO` (change dir)

Para cambiar de directorio utilizamos el comando `cd`.

```
user@mypc:~$ pwd
/home/user
user@mypc:~$ cd contratos
user@mypc:~$ pwd
/home/user/contratos
```

Si no especificamos ni ruta ni directorio y ejecutamos simplemente `cd` se asume que el cambio de directorio se refiere al directorio del usuario, y volvemos a `/home/user`, estemos donde estemos.

```
user@mypc:~$ pwd
/home/user/contratos/gas/2010
user@mypc:~$ cd
user@mypc:~$ pwd
/home/user
```

Podemos movernos a cualquier punto de la jerarquía de directorios utilizando tanto rutas absolutas o relativas.

```
user@mypc:~$ cd contratos/gas
user@mypc:~$ pwd
/home/user/contratos/gas
user@mypc:~$ cd /usr/share/doc/java-common
user@mypc:~$ pwd
/usr/share/doc/java-common
```

Para subir un nivel (o los que sean necesarios), utilizamos el símbolo especial que hemos visto antes, los dos puntos seguidos `..`.

```

user@mypc:~$ pwd
/home/user/contratos/gas
user@mypc:~$ cd ../../
user@mypc:~$ pwd
/home/user

```

Creando directorios y ficheros

mkdir [RUTA/]DIR (make dir)

Para crear un directorio utilizamos la instrucción `mkdir`.

```

user@mypc:~$ ls
contratos
user@mypc:~$ mkdir papeleo
user@mypc:~$ ls
contratos papeleo

```

Al igual que con otros comandos, con `mkdir` también podemos utilizar rutas absolutas o relativas.

Podemos utilizar una ruta relativa o absoluta para crear un directorio donde queramos, sin necesidad de estar situados en el directorio que lo contiene.

```

user@mypc:~$ mkdir contratos/seguroDelCoche
user@mypc:~$ ls contratos
gas internet luz notas.txt seguroDelCoche

```

touch [RUTA/]FICHERO

Para crear ficheros vacíos utilizamos el comando `touch`.

```

user@mypc:~$ touch contratos/contratosPendientes.txt
user@mypc:~$ ls contratos
contratosPendientes.txt gas internet luz notas.txt seguroDelCoche

```

Obviamente, crear ficheros vacíos no tiene mucho sentido y la mayoría de las veces crearemos ficheros para guardar algún tipo de contenido. Más adelante veremos cómo.

echo "CADENA" [DESTINO]

El comando `echo` nos permite imprimir mensajes o cadenas de texto. Las cadenas tienen que ir entrecomilladas. Si no especificamos un destino, la cadena se imprime por pantalla.

```

user@mypc:~$ echo "OLA K ASE?"
OLA K ASE?

```

Si indicamos como destino la ruta hacia un fichero de texto a través del símbolo >> la cadena se concatenará al contenido de dicho fichero.

```
user@mypc:~$ echo "OLA K ASE?" >> saludo.txt
```

Copiando, moviendo, renombrando, borrando

cp [RUTA/]FICHERO DESTINO (copy)

Para copiar un fichero en otro directorio utilizamos el comando cp.

Por ejemplo, para copiar el fichero `saludo.txt` en el nivel superior, en este ejemplo `/home/user`, ejecutamos:

```
user@mypc:~$ pwd
/home/user/contratos
user@mypc:~$ cp saludo.txt ..
```

Para especificar las rutas del fichero y el destino de la copia podemos utilizar tanto rutas relativas como absolutas, incluido el símbolo especial `..` para indicar el directorio superior.

mv [RUTA/]FICHERO DESTINO (move)

Para mover un fichero a otro directorio utilizamos el comando mv.

Por ejemplo, para mover el fichero `saludo.txt` desde el directorio de trabajo actual al directorio `/tmp/`, ejecutamos:

```
user@mypc:~$ mv saludo.txt /tmp/
```

rm [RUTA/]FICHERO (remove)

Para eliminar un fichero, ejecutamos la instrucción rm. Por ejemplo, para borrar definitivamente el fichero `saludo.txt` que acabamos de mover a `/tmp/`, ejecutamos:

```
user@mypc:~$ rm /tmp/saludo.txt
```

rmdir [RUTA/]DIRECTORIO (remove dir)

Para eliminar directorios, necesitamos ejecutar una instrucción específica: `rmdir`. Antes, debemos asegurarnos de que el directorio en cuestión está vacío y no contiene ficheros. Por ejemplo, para eliminar definitivamente un directorio previamente vaciado, ejecutamos:

```
user@mypc:~$ mkdir directorioBasura
user@mypc:~$ ls
```

```

directorioBasura otroDirectorio
user@mypc:~$ rmdir directorioBasura
user@mypc:~$ ls
otroDirectorio

```

Visualizando el contenido de ficheros de texto

cat [RUTA/]FICHERO

Para imprimir el contenido de un fichero de texto utilizamos el comando cat.

```

user@mypc:~$ cat saludo.txt
OLA K ASE?
user@mypc:~$ echo "Pues mira, muy bien." >> saludo.txt
user@mypc:~$ cat saludo.txt
OLA K ASE?
Pues mira, muy bien.

```

more [RUTA/]FICHERO

Para poder visualizar el contenido de un fichero de texto de manera paginada podemos utilizar more.

A medida que pulsamos el espacio, avanzamos una página. Para salir del paginador, llegamos al final del documento o pulsamos q.

less [RUTA/]FICHERO

Una alternativa más potente para visualizar el contenido de un fichero de texto de manera paginada es less.

Además de avanzar página a página podemos movernos hacia atrás en el texto utilizando las teclas de los cursores. Para salir del paginador pulsamos q.

¿Cómo busco ayuda?

No es sencillo recordar las opciones disponibles para cada comando (solo algunos *übergeeks* lo consiguen), así que es habitual echar mano de los comandos de la shell.

Tenemos dos tipos de ayuda que podemos consultar desde la propia línea de comandos.

- Todos los comandos tienen una opción `--help` (a veces también `-h`) que podemos ejecutar para acceder a la ayuda en formato abreviado.

Para acceder a la ayuda del comando `cat`, ejecuta:

```
user@mypc:~$ cat --help
```

- El comando `man` (de *manual*) nos da acceso al manual completo de cada comando.

Para abrir las páginas del manual del comando `sort` ejecuta:

```
user@mypc:~$ man sort
```

¡OJO! El comando `man` abre la ayuda en el paginador `less`. Recuerda que para salir de `less` hay que pulsar `q`.