

Comandos Linux *shell* – ISO

Colegio Retamar

13 de enero de 2022

Índice

1. Comandos para manejo de ficheros	1
1.1. <code>ls</code>	1
1.2. <code>cd</code>	2
1.3. <code>pwd</code>	2
1.4. <code>mkdir</code>	2
1.5. <code>mv</code>	3
1.6. <code>ln</code>	3
1.7. <code>cp</code>	4

1. Comandos para manejo de ficheros

1.1. `ls`

El comando `ls` permite listar el contenido de un directorio, mostrando los ficheros contenidos en su interior, o bien las propiedades de un fichero en particular:

¹ `ls [opciones] [directorio | fichero]`

De entre todas las opciones con las que cuenta el comando, algunas muy útiles son:

- `-l`: para mostrar la salida como una lista descriptiva.

- **-R**: para recorrer de forma recursiva un directorio (y subdirectorios).
- **-a**: para que muestre además los ficheros ocultos.
- **-h**: para mostrar el tamaño de los ficheros legible.
- **-i**: para mostrar el identificador del i-nodo de un fichero.
- **-d**: para mostrar los datos del directorio actual, no del contenido.

1.2. cd

El comando **cd** se usa para cambiar el directorio actual (*change directory*). Su sintaxis es simple:

```
1 cd [directorio]
```

Permite además cambiar al directorio **home** rápidamente cuando no se incluye ninguna opción o se usa la virgulilla (~):

```
1 cd
2 cd ~
```

1.3. pwd

El comando **pwd** muestra el directorio actual en el que estamos (*print working directory*). Es muy útil cuando iniciamos sesión en un servidor remoto o cuando enjaulamos a un usuario.

1.4. mkdir

El comando **mkdir** permite la creación de directorios y subdirectorios en Linux. Su sintaxis es la siguiente:

```
1 mkdir [opciones] directorio
```

Algunas opciones muy potentes son:

- **-m --mode**: crea el directorio estableciendo directamente el modo del mismo (equivalente a: `mkdir -m 0750 test` \equiv `mkdir test && chmod 0750 test/`)
- **-p --parents**: crea todos los directorios y subdirectorios (padres e hijos) aunque no existan. Por ejemplo: `mkdir -p padre/hijo/hijo2` va a crear los directorios **padre**, **hijo** e **hijo2** en caso de que alguno de ellos no exista.

`mkdir` soporta además la creación de múltiples directorios y subdirectorios de forma simultánea y en caso de que no existan. Dicha forma sigue la siguiente sintaxis:

```
1 mkdir -p {directorio1,directorio2,directorio3/directorio4,directorio5/dir6}
```

quedando el árbol de directorios de la siguiente forma:

```
1 .
2 |-- directorio1
3 |-- directorio2
4 |-- directorio3
5 |   '-- directorio4
6 '-- directorio5
7     '-- dir6
```

1.5. mv

El comando `mv` mueve (renombra) un fichero y/o directorio a otro lugar. Como el Linux un fichero está compuesto por su ruta absoluta, moverlo de un sitio a otro se simplifica a cambiarle el nombre (por ejemplo, el nombre de un fichero es `a` y su ruta absoluta (el nombre completo) es `/tmp/a`, moverlo a otra ubicación sería cambiar el nombre completo: `mv a /usr/b` \equiv `mv /tmp/a /usr/b`).

En particular, algunas opciones muy útiles son:

- `--backup`: para crear una copia de seguridad del fichero destino en caso de que exista, y no sobrescribir su contenido.
- `-f --force`: para sobrescribir el fichero sin preguntar.
- `-i --interactive`: para preguntar antes de mover de un sitio a otro.
- `-u --update`: para, en lugar de sobrescribir, actualizar el contenido del destino únicamente de aquellos ficheros que hayan cambiado.

1.6. ln

El comando `ln` sirve para crear enlaces entre dos ficheros/directorios. Los enlaces que se pueden crear son de dos tipos:

1. Enlaces simbólicos, en donde se crea un nuevo fichero que apunta al original, y depende de que este original exista (si se borra, se tiene un enlace roto).
2. Enlaces duros, en donde se crea un “alias” para un fichero, de forma que si se borrara el original el enlace duro seguiría teniendo el contenido del mismo (comparten i-nodo).

1.7. **cp**

El comando **cp** crea una copia de un fichero/directorio en otra ubicación y con el mismo o distinto nombre.