

## Tabla de contenidos

### 1. Introducción

- 1.1. ¿Como se compone un comando?
- 1.2. ¿Como encuentro el comando que necesito?, (apropos)
- 1.3. ¿donde encuentro las opciones?, uso del manual (man).

### 2. Comandos básicos

#### 2.1. Administración de archivos

- 2.1.1. Mover archivos
- 2.1.2. Copiar archivos
- 2.1.3. Eliminar archivos
- 2.1.4. Renombrar archivos
- 2.1.5. Administrar permisos

-----

### 1. Introducción

#### 1.1. ¿Como se compone un comando?

Un comando no se compone por si mismo solamente, son escasos los comandos que se forman por una sola palabra, sino que a este lo acompañan opciones y argumentos.

\$comando [opciones] <argumentos>

por ejemplo:

\$ ls -l /home

comando: ls (listar)

opción: -l (descripción larga)

argumento: /home (directorio que quiero listar)

#### 1.2. ¿Como encuentro el comando que necesito?, (apropos)

Es útil acordarse de ciertos comandos que nos hacen la vida mas sencilla para la administración de un sistema GNU/Linux, uno de esos es apropos, este comando junto con el propósito que buscamos (move, remove, clear, etc) no va a detallar los comandos que tengas alguna funcionalidad parecida al propósito pedido, por ejemplo para renombrar (rename).

```
>miusuario@localhost:~$ apropos rename
>btrename (1) [btrename.bittorrent] - change the suggested filename inside a bittorrent file
>btrename.bittorrent (1) - change the suggested filename inside a bittorrent file
>dpkg-name (1) - rename Debian packages to full package names
>mmove (1) - move or rename an MSDOS file or subdirectory TQ
>mren (1) - rename an existing MSDOS file TQ
>mv (1) - move (rename) files
>rename (1) - renames multiple files
>rename (2) - change the name or location of a file
>XStoreName (3x) - set or read a window's WM_NAME property
>XStoreNamedColor (3x) - set colors
```

el que puedo usar para renombrar un archivo es mv.

#### 1.3. ¿donde encuentro las opciones?, uso del manual (man).

Para encontrar las opciones lo mas útil es usar el comando man, su uso es el siguiente:

```
>miusuario@localhost:~$ man mv
>MV(1)
```

MV(1)

```

>
>NOMBRE
> mv - mueve (renombra) ficheros
>
>SINOPSIS
> mv [opción...] origen destino
> mv [opción...] origen... destino
>
> Opciones de POSIX: [-fi]
>
> Opciones de GNU (en la forma más corta): [-bfuiuv] [-S sufijo] [-V {numbered,existing,simple}] [--help] [--version] [--]
>
>DESCRIPCIÓN
> mv mueve o renombra ficheros o directorios.

```

(el manual continua, por un tema de espacio mostramos hasta aqui)  
de este manual se sale con la tecla [q].

otra posibilidad es usar el comando con la opción --help  
ejemplo:

```

>miusuario@localhost:~$ mv --help
>Modo de empleo: mv [OPCIÓN]... ORIGEN DESTINO
> o bien: mv [OPCIÓN]... ORIGEN... DIRECTORIO
> o bien: mv [OPCIÓN]... --target-directory=DIRECTORIO ORIGEN...
>Renombra ORIGEN a DESTINO, o mueve ORIGEN(es) a DIRECTORIO.
>
>Los argumentos obligatorios para las opciones largas son también obligatorios
>para las opciones cortas.
> --backup[=CONTROL] crea una copia de seguridad antes de borrar
> -b como --backup pero no acepta ningún argumento
> -f, --force no pregunta nunca antes de sobrescribir
> (equivalente a --reply=yes)
> -i, --interactive pide confirmación antes de sobrescribir
> (equivalente a --reply=query)
> --reply={yes,no,query} especifica cómo tratar la pregunta acerca de
> un fichero de destino que ya exista
> --strip-trailing-slashes elimina todas las barras finales de cada
> argumento ORIGEN
> -S, --suffix=SUFIJO reemplaza el sufijo de respaldo habitual
> --target-directory=DIRECTORIO mueve todos los argumentos ORIGEN al
> directorio DIRECTORIO
> -u, --update mueve solamente cuando el fichero ORIGEN
> es más moderno que el fichero de destino,
> o cuando falta el fichero de destino
> -v, --verbose da detalles de lo que va haciendo
> --help muestra esta ayuda y finaliza
> --version informa de la versión y finaliza
>
>El sufijo de respaldo es '~', a menos que se establezca con --suffix o con
>SIMPLE_BACKUP_SUFFIX. El método de control de versión se puede seleccionar
>con la opción --backup o a través de la variable de entorno VERSION_CONTROL.
>Estos son los valores:
>
> none, off nunca realiza copias de seguridad (incluso si se da la
> opción --backup)
> numbered, t crea copias de seguridad numeradas
> existing, nil numeradas si existen copias de seguridad numeradas,
> simples en caso contrario
> simple, never siempre crea copias de seguridad simples
>
>Comunicar bichos a <bug-coreutils@gnu.org>.

```

## 2. Comandos básicos

Ya con las herramientas antes presentadas estamos en condiciones de manejarnos por nuestros medios.

Ahora seguimos con algunos comandos básicos, pero también no queremos dejar de recordar que se complica o es peligroso en algún aspecto tratar de memorizar tantos comandos con sus respectivas opciones, por eso aconsejamos recordar los comandos mas útiles como man o apropos, estos nos serán de guía para encontrar todos los demás.

### 2.1. Administración de archivos

#### 2.1.1. Mover archivos

Como ya vimos en los puntos anteriores se usa para este fin el comando mv

```
USO: $mv [opciones] <ORIGEN> <DESTINO>
ejemplo $ mv /home/miusuario/archivo.txt /home/miusuario/Desktop
```

con este comando mueve archivo.txt hacia el escritorio,

#### 2.1.2. Copiar archivos

Para copiar se utiliza el comando cp

```
USO: $cp [opciones] <ORIGEN> <DESTINO>
ejemplo $ cp -r /home/miusuario/carpeta /home/miusuario/Desktop
```

la opción -r (recursive) mueve todos los contenidos de la carpeta recursivamente.

#### 2.1.3. Eliminar archivos

Se utiliza el comando rm

```
USO: $rm [opciones] <ARCHIVO/DIRECTORIO>
ejemplo $ rm /home/miusuario/archivo.txt
```

#### 2.1.4. Renombrar archivos

Para renombrar se utiliza tambien el comando mv

```
USO: $mv [opciones] <ORIGEN> <DESTINO>
ejemplo $ mv /home/miusuario/archivo.txt /home/miusuario/archivo_nuevo.txt
```

#### 2.1.5. Administrar permisos

Los comandos para administrar los atributos de los archivos son chmod(cambia los permisos), chown(cambiar el dueño), chgrp(cambia el grupo).

El que requiere mas explicación es el cambio de permisos, si vemos un listado:

```
>miusuario@localhost:~/Desktop$ ls -l
>total 228
>-rw-r--r-- 1 miusuario miusuario 7027 2005-06-14 01:28 archivo1
>-rw-r--r-- 1 miusuario miusuario 196208 2005-06-14 01:31 archivos2.pdf
>-rw-r--r-- 1 miusuario miusuario 1034 2005-06-12 22:07 archivo3.txt
>drwx----- 2 miusuario miusuario 20480 2005-06-14 10:33 Trash
```

la primer columna contiene los permisos, hay dos formas de usar este comando, la forma literal y la forma octal, explicaremos solo la forma octal por ser la mas fácil de recordar:

```
Arwxr-xr-x
```

La primer columna (A) nos informa de que tipo de dato nos estamos refiriendo, los valores que mas encontramos son:

l: enlaces simbólico

d: directorio

-: cuando no contiene ninguna letra es un archivo

las siguientes son tres grupos (owner,group,other) de tres columnas cada uno, y que contiene rwx (Read,Write,eXecute), asignándole valores numéricos a estas letras r=4, w=2, x=1 y sumando obtenemos una combinación única por cada grupo, veamos algunos ejemplos:

```
>-rw-r--r-- 1 miusuario miusuario 7027 2005-06-14 01:28 archivo1
```

En este caso la primer columna nos dice que tratamos con un archivo, luego de esta vemos los tres grupos:

```
>owner(dueño) [rw-] r(4)+w(2)=(6) le falta el permiso de ejecucion
>group(grupo) [r--] r(4)=(4) le faltan los permisos de lectura y ejecucion
>other(otros) [r--] r(4)=(4) le faltan los permisos de lectura y ejecucion
```

para darle todos los permisos al usuario y al grupo, y quitarle todos a los otros uso chmod:

```
USO: $ chmod [opciones] <PERMISOS> <ARCHIVO>
ejemplo $ chmod /home/miusuario/archivo.txt /home/miusuario/archivo_nuevo.txt
ejemplo $ chmod -r 755 /home/miusuario
```

(le doy posibilidad de que vean mis documentos pero que no puedan escribir. La opción -r(recursive) cambia los permisos de todos los archivos y carpetas contenidos en mi directorio recursivamente)

seguimos con el ejemplo anterior:

```
$ chmod 770 archivo
```

NOTA: tengo que pasarle la ruta completa (path) al archivo, en este caso se toma que estoy "parado" en el directorio que contiene al archivo.

Las combinaciones de números mas conocidas son:

```
777 rwxrwxrwx Todos los permisos para todos (cuidado con esto)
666 rw-rw-rw-
755 rwxr-xr-x escribe solo el dueño, los demás leen y ejecutan
555 r-xr-xr-x todos leen y ejecutan
444 r--r--r-- todos solo pueden leer
```

NOTA: el dueño del archivo por mas que no tenga permiso de escritura o lectura puede modificar los atributos, si no tuviera posibilidad hay que modificarlo siendo root.

NOTA: en los directorios para poder entrar, se necesitan permisos de lectura y ejecución.

```
USO: $ chown [opciones] <USUARIO> <ARCHIVO>
ejemplo $ chown usuario /home/miusuario/archivo
```

```
USO: $ chgrp [opciones] <GRUPO> <ARCHIVO>
ejemplo $ chgrp grupo /home/miusuario/archivo
```

Estos dos comandos no necesitan mayor explicación de que el usuario y el grupo deben estar contenidos en el archivos /etc/passwd, /etc/shadow, /etc/group según nuestra configuración. Su uso es similar a los anteriores.