

## MANEJO DE ARCHIVOS Y DIRECTORIOS A TRAVES DE COMANDOS

Más allá de interfaces gráficas como KFM, Linux dispone de un numeroso conjunto de utilidades para manipular archivos (individualmente o en grupos) tipeando comandos en una terminal. A través de esta modalidad de trabajo es posible alcanzar un alto nivel de flexibilidad y versatilidad para realizar operaciones de cualquier grado de complejidad.

Los comandos utilizados más frecuentemente se describen brevemente a continuación. Para un mayor nivel de detalle, se sugiere consultar el manual en línea de Linux (ver más adelante el comando **man**) y especialmente la documentación del intérprete de comandos utilizado (usualmente, el **bash**).

## COMANDOS LINUX PARA LA GESTION DE ARCHIVOS Y DIRECTORIOS

### cp

---

#### *Propósito*

El comando cp es una abreviatura de copy (copiar); permite copiar archivos y directorios. Para copiar un archivo se usa el siguiente mandato:

#### *Sintaxis*

cp [Opciones] archivo\_fuente directorio\_destino

cp [Opciones] archivo\_fuente archivo\_destino

#### *Opciones*

- a conserva todos los atributos de los archivos.
- b hace un backup antes de proceder a la copia.
- d copia un vínculo pero no el fichero al que se hace referencia.
- i pide confirmación antes de sobrescribir archivos.
- p conserva los sellos de propiedad, permisos y fecha.
- R copia los archivos y subdirectorios.
- s crea enlaces en vez de copiar los ficheros.
- u únicamente procede a la copia si la fecha del archivo origen es posterior a la del destino.
- v muestra mensajes relacionados con el proceso de copia de los archivos.

#### *Descripción*

El comando cp copia un archivo a otro. También puede copiar varios ficheros en un directorio determinado.

*Ej.*

```
cp manual_linux_v1 ../../doc/linux
```

En este ejemplo copia el archivo manual\_linux en un directorio dos niveles más arriba del actual, en el directorio doc/linux

## **mv**

---

### *Propósito*

Modifica el nombre de los archivos y directorios moviéndolos de una ubicación a otra.

### *Sintaxis*

```
mv [Opciones] fuente destino
```

### *Opciones*

-d hace una copia de seguridad de los archivos que se van a mover o renombrar.

-f elimina los archivos sin solicitar confirmación.

-v pregunta antes de sobrescribir los archivos existentes.

### *Descripción*

El comando mv se puede utilizar para modificar el nombre o mover un archivo de un directorio a otro. Trabaja tanto con archivos como con los directorios.

*Ej.*

```
mv manual_linux_v1 manuales/linux
```

```
mv manual_linux_v1 manual_linux_v1_doc
```

```
mv manual_linux_cap1 manual_linux_cap2 manual_linux_cap2 /manual/linux
```

## **rm**

---

### *Propósito*

Elimina uno más archivos (puede eliminar un directorio completo con la opción -r).

### *Sintaxis*

rm [Opciones] archivos

### *Opciones*

-f elimina todos los archivos sin preguntar.

-i pregunta antes de eliminar un archivo.

-r elimina todos los archivos que se encuentran en un subdirectorio y por último borra el propio subdirectorio.

-v muestra el nombre de cada archivo antes de eliminarlo.

### *Descripción*

El comando rm se utiliza para borrar los archivos que se le especifiquen. Para eliminar un fichero ha de tener permiso de escritura en el directorio en el que se encuentra.

Ej.

rm manual\_linux\_v1

rm -r documentos/

## **mkdir**

---

### *Propósito*

crear directorios.

### *Sintaxis*

mkdir [Opciones] nombre\_directorio

### *Opciones*

-m modo, asigna la configuración de permisos especificada al nuevo directorio.

-p crea directorios emparentados (en caso de que no existan).

### *Descripción*

El comando mkdir se utiliza para crear un directorio específico.

Ej.

mkdir manuales

## **rmdir**

---

### *Propósito*

Elimina un directorio (siempre y cuando esté vacío).

### *Sintaxis*

rmdir [Opciones] directorio

### *Opciones*

-p elimina cualquier directorio emparentado que este vacío.

### *Descripción*

El comando rmdir elimina los directorios vacíos. Si tiene algún contenido, tendrá que utilizar el comando rm -r para eliminar el directorio y sus contenidos.

### *Ej.*

rmdir manual

## **ls**

---

### *Propósito*

Listar el contenido de un directorio.

### *Sintaxis*

ls [Opciones] [nombre\_directorio o archivo]

### *Opciones*

-a muestra todos los archivos. Incluyendo a los ocultos.

-b muestra los caracteres no imprimibles de los nombres de los ficheros utilizando un código octal.

-c ordena los archivos de acuerdo con la fecha de creación.

-d muestra una lista en la que aparecen los directorios como si fuesen archivos (en vez de mostrar su contenido).

-f muestra el contenido del directorio sin ordenar.

-i muestra información de i-node.

- l muestra la lista de archivos con formato largo y con información detallada (tamaño, usuario, grupo, permisos etc.).
- p añade un carácter al nombre del archivo para indicar a que tipo pertenece.
- r coloca la lista en orden alfabético inverso.
- s muestra el tamaño (kb) de cada archivo próximo al solicitado.
- t ordena la lista de acuerdo con la fecha de cada fichero.
- R muestra una lista con el contenido del directorio actual y de todos sus subdirectorios.

### *Descripción*

El comando ls muestra el contenido de un directorio determinado. Si se omite el nombre del directorio, mostrará el contenido del directorio en el que se encuentre. Por defecto, ls no muestra el nombre de los archivos cuyo nombre comience con un punto; para verlos tendrá que utilizar la opción -a.

*Ej.*

ls -a

ls -l

ls -la

## **cd**

---

### *Propósito*

Cambiar de directorio.

### *Sintaxis*

cd [directorio]

### *Opciones*

Ninguna

### *Descripción*

Si escribe cd sin ningún nombre de directorio como argumento, se cambiará al directorio home del usuario. En cualquier otro caso se moverá al directorio indicado, si existe.

## **pwd**

---

## Propósito

Mostrar la ruta del directorio de trabajo actual.

## Sintaxis.

pwd

## Opciones

Ninguna

## Descripción

El comando pwd imprime el directorio de trabajo (aquel en el que actualmente se está trabajando).

## chmod

---

### Propósito

Modifica los permisos de uno o más archivos o directorios.

### Sintaxis

chmod [Opciones] [permiso\_descripción] archivo

### Opciones

-c muestra los archivos a los que se les han modificado los permisos.

-f hace que no aparezca en pantalla ningún mensaje de error.

-v muestra los cambios efectuados en los permisos de archivos.

-R cambia los permisos de los archivos de todos los subdirectorios.

### Permisos\_descripción

Quien Acción Permiso

Quien	Acción	Permiso
u: usuario	+: agregar	r: lectura
g: grupo	-: quitar	w: escritura
o: otros	=: asignar	x: ejecutar
a: todos		s: ajustar con el ID

		del usuario.
--	--	--------------

*Ej.*

```
chmod u+rx manual_linux
```

El usuario tendrá los permisos de lectura y ejecución sobre el archivo manual\_linux

### *Descripción*

Para utilizar eficazmente el comando chmod, debe especificarse la configuración de los permisos de acuerdo a la tabla de permisos\_descripción.

Por ejemplo para que todos tengan permiso de lectura en un determinado archivo se tipea, *chmod a+r nombre\_archivo*. También se podría haber tipeado *chmod u=r,g=r,o=r nombre\_archivo*.

Otra forma de modificar los permisos es a través de un número octal de 3 cifras una cifra por cada grupo de permisos, este número surge de realizar la suma de los permisos que se les quiere asignar de acuerdo a los siguientes valores:

Permiso de lectura r = 4

Permiso de escritura w = 2

Permiso de ejecución x = 1

Y si no se le concede cualquier permiso el valor asignado es 0.

El formato para utilizar chmod especificando los permisos por medio de números es el siguiente.

```
chmod permiso_usuario permiso_grupo permiso_otros
```

Ejemplo, supongamos que creamos el archivo permiso.txt y queremos que el usuario tenga todos los permisos, el grupo los permisos de lectura y ejecución y finalmente que el resto de los usuarios tenga sólo el permiso de ejecución.

Para el usuario: lectura r = 4, escritura w = 2, ejecución x = 1 ; sumados = 7

Para el grupo: lectura r = 4, escritura w = 0, ejecución x = 1; sumados = 5

Para el resto de los usuarios: lectura r = 0; escritura w = 0, ejecución x = 1; sumados = 1

Entonces el comando seria: *chmod 751 permisos.txt*

En la lista detallada de los archivos de un directorio (usando el comando ls), los permisos de lectura escritura y ejecución del usuario, grupo y otros se mostrarán a través de la secuencia rwxrwxrwx, cuando algún permiso no está activado aparece un guión en su reemplazo.

## **cat**

---

### *Propósito*

Muestra el contenido de un archivo utilizando la salida estándar (pantalla).

### *Sintaxis*

cat [-benstvA] archivos

### *Opciones*

- b números de líneas que no estén en blanco.
- e muestra el final de una línea (como \$) y todos los caracteres no imprimibles.
- n numera todas las líneas de salida, comenzando por el 1.
- s sustituye varias líneas en blanco por una sola.
- t muestra las tabulaciones como ^I.
- v muestra los caracteres no imprimibles.
- A muestra todos los caracteres (incluidos los no imprimibles).

### *Descripción*

Normalmente, cat se utiliza para mostrar el contenido de un archivo o para *concatenar* varios dentro de un mismo fichero. Por ejemplo,  
cat archivo1, archivo2, archivo3 > todo

combina los tres archivos dentro de uno solo llamado todo.

## **find**

---

### *Propósito*

Muestra una lista con los archivos que coinciden con un criterio específico.

### *Sintaxis*

find [ruta] [opciones]

### *Opciones*

- depth procesa, en primer lugar, el directorio en el que se encuentra y luego sus subdirectorios.
- maxdepth n restringe la búsqueda a n niveles de directorios.



- follow procesa los directorios que se incluyen dentro de los enlaces simbólicos.
- name modelo localiza los nombres de los archivos que coinciden con el modelo propuesto.
- ctime n localiza los nombres de los archivos creados n días atrás.
- user nombre\_usuario localiza los archivos pertenecientes al usuario específico.
- group nombre\_grupo localiza los archivos pertenecientes al grupo específico.
- path ruta localiza a los archivos cuya ruta coincide con el modelo propuesto.
- perm modo localiza los archivos con los permisos especificados.
- size +nK localiza los archivos cuyo tamaño (en kilobytes) es mayor de especificado.
- print imprime el nombre de los archivos que encuentra.
- exec comando [opciones] {} \; ejecuta el comando especificado analizando el nombre del archivo localizado.

### *Descripción*

El comando find es de gran utilidad cuando se quiere localizar todos los archivos que coinciden con algún criterio. Si escribe find sin ningún argumento, la salida mostrará un listado en el que aparecen los archivos de todos los subdirectorios de la carpeta en la que se encuentre.

Para ver todos los archivos cuyo nombre termine con **.gz**, tendrá que escribir:

```
find -name "*.gz "
```

Para buscar a partir del directorio /usr/doc todos los archivos con extensión **.bak** y eliminarlos, utilizar el comando:

```
find /usr/doc -name "*.bak" -exec rm -f {} \;
```

en donde la secuencia {} se substituirá por el nombre completo de cada archivo encontrado.

---

## **grep**

### *Propósito*

Busca en uno o más archivos las líneas que coincidan con una expresión regular (modelo de búsqueda).

### *Sintaxis*

```
grep [opciones] modelo archivos
```

### *Opciones*

- N muestra N líneas que contienen el modelo de búsqueda señalado.
- c muestra el número de líneas que contienen el modelo de búsqueda.
- f archivo lee las opciones del archivo especificado.
- i ignora letras
- l muestra los nombres de los archivos que contienen un modelo.
- q devuelve el número de línea siguiente a aquellas en las que se encuentra el modelo de búsqueda.
- v muestra las líneas que no contienen el modelo de búsqueda.

### *Descripción*

El comando localiza el modelo de búsqueda en los archivos especificados. El modelo es una expresión regular en los archivos especificados que tienen sus propias reglas. Generalmente se utiliza para buscar una secuencia de caracteres en uno o más archivos de texto.

### *Ejemplo*

grep Juan ListadoDeAlumnos.txt

## **OTROS COMANDOS DE LINUX**

**man:** Muestra por pantalla secciones del manual del usuario.

*Formato:* man Nombre del comando.

Ej: man ls.

**mesg:** Habilita o deshabilita la comunicación entre usuarios por medio de *write*.

*Formato:* mesg [n/y].

**lpr:** Imprime el contenido de un archivo.

*Formato:* lpr [Opción] Archivo

Se consideran las principales opciones:

-P cola Indica la cola de impresión a utilizar.

-n<número>: Indica la cantidad de copias a imprimir, por defecto siempre es 1.

-R: Remueve el archivo después de realizada la impresión.

**tree:** Lista todos los directorios a partir del directorio actual o del directorio indicado.

*Formato:* tree [Directorio].

**tty:** Muestra el número de la terminal donde está trabajando el usuario.

*Formato:* tty

**who:** Visualiza los usuarios que están activos en el sistema, sin ningún tipo de argumento éste comando muestra los nombres de usuario, número de terminal y horario de conexión por cada usuario activo del sistema. Utilizando los argumentos who am i el comando muestra con que nombre de usuario está usted conectado.

*Formato:* who [Opción]

**write:** Envía mensajes a otros usuarios hasta que se digite "Control D". La recepción de estos mensajes puede ser deshabilitada por el usuario utilizando el comando MESG.

*Formato:* write Usuario Terminal