**Linux 1-3 SISTEMAS OPERATIVOS MULTIUSUARIO: UNIX (LINUX).**

Índice de Contenidos

[1-3.1 Comandos iniciales: Formato. 2](#_Toc68803971)

# Comandos iniciales: Formato.

* **Comando cp**

cp Se utiliza para copiar directorios y ficheros.

**cp [-r|-R] [-i][-f] {trayectoria-ficheros-origen} {trayectoria-destino}**

Copia los ficheros indicados por trayectoria ficheros origen en el destino señalado que puede ser la trayectoria de un directorio o de ficheros. Si el destino existe lo sobrescribe sin preguntar nada. Si el destino existe y sobrescribimos se cambia el contenido, pero se conserva el propietario y los permisos del destino. Si no existe el destino lo crea y mantiene los permisos del origen. En este caso, la propiedad del fichero destino será del usuario que ha ejecutado la orden cp.

Las distintas opciones que posee este comando son:

**-r|-R** → copia los ficheros indicados y la estructura arborescente que subyace a partir de la trayectoria de esos ficheros, en la trayectoria destino.

**-i** → hace que se pida confirmación antes de sobrescribir un fichero existente.

**-f** → Sobrescribe el destino en caso de existir, aunque no se tenga permiso de escritura sobre él; pero si es necesario tener permiso en el directorio en el que está.

**-p** → Al hacer la copia mantiene fecha, hora, propietario y permisos.

Ejemplos:

**cp [-r|-R] directorio1 directorio2** Copia, con la opcion -R de forma recursiva todo lo que cuelga del directorio1 en el directorio2.

Con la opción -r hace lo mismo, pero si hay ficheros especiales los transforma en normales.

**cp [-ipf] fichero1 fichero2 directoriodestino.** Copia todos los ficheros de la lista en el directorio destino

**cp archi1 nuevoar** .: Se creará una nueva copia del archivo archil y será llamada nuevoar.

**cp arch1 stu1/nuevof**.: El archivo archl del directorio actual es copiado en el subdirectorios stul, con el nombre nuevof.

**cp nuevof doc1 stu3** .: los archivos nuevof y doc1 del directorio actual son copiados con el mismo nombre en el directorio stu3.

**cp archi1 stu2**.: hace una copia de arch1, con el mismo nombre en el subdirectorio sut2.

**cp file1 stu2 stu4 stu5.:** únicamente se copiaría file1 en el directorio sut5. Intenta copiar stu2 y stu4 aa stu5, pero al ser directorios provoca un error.

**cp -R stu1\* stu2** .: Copia los archivos y directorios que cuelgan de stu1 en stu2.

**cp -f stu1/\* stu2** .: Copia los archivos de stu1 en stu2 y sobrescribe si existe algún fichero que se llame como alguno de stu1.

* **Comando mv**

mv Se utiliza para trasladar y/o renombrar ficheros ponemos.

**mv [-i] {trayectoria-ficheros-origen} {trayectoria-destino}**

mediante este comando vamos a poder realizar tres acciones diferentes que son:

1ª sirve para trasladar el fichero o los ficheros indicados al destino que será un directorio.

2º sirve para renombrar un fichero si en destino indicamos el nuevo nombre del fichero.

3º sirve para trasladar y renombrar un fichero si en la trayectoria destino ponemos la trayectoria de un fichero que no existe en otro directorio.

La opción de este comando es:

**-i** → pregunta si algún fichero existe para sobrescribirlo.

* **Comando rm**

rm Borra ficheros.

**rm [-fi] {trayectoria-ficheros}**

Borra los ficheros cuyos nombres indicamos, lo que en realidad realiza es borrar la referencia al fichero indicado en el directorio donde se almacena, lo cual implica que no accede al i-nodo donde se almacena el fichero. ¡Cuidado con la seguridad! Las distintas opciones que tiene este comando son:

**-i** → se pide confirmación antes de borrar los ficheros.

**-r** → borra todo el contenido del directorio especificado (incluidos los subdirectorios y sus ramificaciones), así como el propio directorio.

Sintaxis: ***rm*** *(fichero1) ... (ficheroN)*

*Hay que tener en cuenta que cuando los ficheros son borrados , son irrecuperables.*

Opciones: -i, -r -f

Ejemplo: ***rm*** *-i pedirá confirmación antes de borrar un fichero.*

Ejemplo: ***rm*** *-r \**  Borra todo, incluido los directorios que tengas por debajo. Tener cuidado con esta orden, porque cuando suprima un directorio ya no podrá recuperarlo después ni tampoco los archivos que contenía.

Notas: Ojo. ***rm*** borra de verdad. No hay undelete como en MS-DOS.

Ejemplos:

**rm fil1** Borra fil1 del directorio actual.

**rm fil\*** Borraría todos los archivos que comiencen por fil contenidos en el directorio actual.

**rm \*** Borraría todos los archivos del directorio actual. No borraría el directorio actual.

**rm -i\*** Borraríamos interactivamente todos los archivos del directorio actual. No borra subdirectorios.

**rm -r\*** Borraría todos los archivos y directorios situados bajo el directorio actual, pero no este directorio.

**rm t\*** Borra los ficheros que empiezan por t.

**rm t \*** Borra el fichero t y todos los ficheros del directorio activo

¿pensar que catástrofe podría ocurrir si además usamos la opción -r?

* **Comando tail**

Tail Se utiliza para visualizar las ultimas 10 líneas del fichero especificado o de la entrada estándar

tail [opción...] [archivo...]

Syntaxis🡪 **tail [opciones ] [{ficheros}]**

Muestra la última parte de un archivo, por defecto 10 líneas

Ejemplo: tail /etc/passwd visualiza las 10 últimas líneas del fichero /etc/passwd

1. **-n +numero** Indica que se visualizará desde la línea número hasta el final del fichero.

tail –n +n comienza a mostrar el archivo en el renglón n

Tail –n +3 /etc/passwd visualiza desde la 3 línea hasta el final del fichero.

1. **-n Numero**

Escribe las últimas N líneas. **Numero** Indica cuantas líneas se visualizarán empezando a contar desde el final del archivo.

Tail –n 3 /etc/passwd visualiza las 3 últimas líneas del fichero /etc/passwd

Es igual que poner tail -3 /etc/passwd visualiza las 3 últimas líneas del fichero

1. **-c +N**

Escribe los últimos N bytes.

Tail –c +3 tt comenzando en el byte 3. Comienza a visualizar el fichero desde el número de carácter indicado.

1. **-q:** Evita que se muestren los títulos de los archivos especificados, solo cuando queremos visualizar dos ficheros o más a la vez.

Tail –q /etc/passwd /etc/shells

**EJERCICIOS**

1. Visualice las ultimas 10 líneas de **/etc/passwd**.
2. Visualice las ultimas 5 líneas de **/etc/hosts** y **/etc/shells**.
3. Muestre desde el 76 carácter hasta el final del archivo **/etc/hosts**.

* **Comando head**

head Se utiliza para visualizar las 10 primeras líneas del fichero especificado o de la entrada estándar

**head [-numero] [{ficheros}]**

**-numero** Indica cuantas líneas se visualizarán desde el principio del fichero .

Ejemplo: head /etc/passwd visualiza las 10 primeras líneas del fichero /etc/passwd

head -3 /etc/passwd visualiza las 3 primeras líneas del fichero /etc/passwd

ls –l | head -3 visualiza las 3 primeras líneas de la salida de ls-l

**ARGUMENTOS**

Se pueden especificar uno o más archivos a la vez. De especificarse más de uno, los archivos se desplegarán en el mismo orden en el cual fueron especificados por el usuario, cada uno con su nombre de archivo como título antes de mostrar su contenido.

**OPCIONES**

**-n:** Permite especificar el número de líneas que hay que mostrar. Ejemplo:

Head –n 2 /etc/passwd

ana@ubuntu:~$ head -n 2 /etc/passwd

root:x:0:0:root:/root:/bin/bash

daemon:x:1:1:daemon:/usr/sbin:/bin/sh

ana@ubuntu:~$

ana@ubuntu:~$ head -2 /etc/passwd

root:x:0:0:root:/root:/bin/bash

daemon:x:1:1:daemon:/usr/sbin:/bin/sh

**-q:** Evita que se muestren los títulos de los archivos especificados, solo cuando queremos visualizar dos ficheros o más a la vez.

Head –q /etc/passwd /etc/shells

**-c:** Permite especificar el número de caracteres a desplegar, en vez de líneas.

Head –c 12 /etc/passwd

**EJERCICIOS**

1. Visualice las primeras 10 líneas de **/etc/passwd**.
2. Visualice las primeras 5 líneas de **/etc/hosts** y **/etc/shells**.
3. Muestre las primeras 10 líneas de **/etc/shells** y **/etc/hosts** sin sus respectivos títulos.
4. Muestre los primeros 76 caracteres del archivo **/etc/hosts**.

* **Comando wc**

wc Se utiliza para contar las líneas palabras y caracteres que contengan los archivos referenciados.

**wc [-lwc] [{ficheros}]**

**-l 🡪** Cuenta el número de líneas de un archivo

**-w 🡪** Cuenta el número de palabras de un archivo

**-c🡪** Cuenta el número de caracteres de un archivo

Para **wc**, una palabra es un conjunto de caracteres que hay entre dos blancos( espacios, tabuladores, retornos de carro, etc.).

Al contar, el comando wc entiende que una línea está delimitada por el carácter (\n), una palabra por espacios, tabuladores o (\n). Cuando cuenta caracteres incluye (\n), tabuladores y espacios. El carácter \n equivale a retorno de carro.

Si no se especifica archivo, wc cuenta desde el archivo estándar de entrada (teclado).

Si no se especifica ninguna opción, aparecerán en pantalla e con este orden el número de líneas, palabras, caracteres y nombre del archivo o archivos.

Ejemplos: Wc doc1 Mostrará el número de líneas, palabras y caracteres del archivo doc1.

Wc –w stu2/doc3 🡪 Mostrará el número de palabras del documento doc3, bajo el subdirectorio stu2.

Who | wc –l 🡪 Contaría el número de usuarios que hay conectados.