



Copiar en Linux: con CP es muy sencillo

Si quieres trabajar con la línea de comandos en una de las numerosas distribuciones de código abierto de [Linux \(servidores/know-how/linux-una-alternativa-a-windows-de-bajo-coste/\)](#), primero debes conocer varios de los [comandos de Linux \(servidores/configuracion/comandos-de-linux-la-lista-fundamental/\)](#). Estos te permiten navegar por estructuras de carpetas, editar datos o, como en este caso, copiar archivos y directorios. Con el comando CP (abreviatura de “copy”) puedes hacer **copias de archivos individuales o de carpetas enteras**.

Índice

1. ¿Para qué necesitas CP en Linux? (servidores/configuracion/comando-cp-de-linux/#c375146)
2. Sintaxis del comando CP de Linux (servidores/configuracion/comando-cp-de-linux/#c375147)
3. Funciones especiales del comando CP (servidores/configuracion/comando-cp-de-linux/#c375188)
4. Ejemplos para el uso de CP (servidores/configuracion/comando-cp-de-linux/#c375206)
5. Alternativas al comando CP de Linux (servidores/configuracion/comando-cp-de-linux/#c375177)

Dominios web baratos

Dominios tan originales como tus ideas.

Registra tu dominio con IONOS y disfruta de las funciones integrales que tenemos para ofrecerte.

- ✓ Correo incluido
- ✓ Certificado SSL
- ✓ Asistencia 24/7

Continúa

¿Para qué necesitas CP en Linux?

Copiar archivos y carpetas es uno de los elementos básicos de cualquier sistema operativo y la mayoría de los usuarios lo hacen a diario. Las copias se utilizan para compartirlas con otras personas o para almacenarlas como copias de seguridad. Por regla general, esta herramienta es una parte integral (y ya instalada) de todo sistema de Linux, independientemente de la distribución que se utilice. CP pertenece a los **GNU Core Utilities** (abreviado: "coreutils"). Este paquete contiene las herramientas más importantes y está disponible de forma gratuita en todas las distribuciones de Linux.

Sintaxis del comando CP de Linux

La función principal de CP es sencilla: copiar uno o más archivos a una ubicación especificada por el usuario. Para ello, hay dos datos especialmente importantes: el **nombre del archivo y el lugar de destino de la operación**. La sintaxis del comando CP está estructurada de la siguiente manera:

```
1 | cp [Origen] [Destino]
```

Para usar el comando CP, **ve al directorio del archivo que quieras copiar** e introduce en el comando el nombre del archivo y la ruta en la que quieres que se guarde el nuevo archivo.

```
1 | cp texto.txt /home/usuario/carpeta_de_destino/
```

En este ejemplo, creamos un nuevo archivo en la carpeta de destino que tiene el mismo nombre que el archivo de origen. Si quieres **copiar varios archivos al mismo tiempo**, escríbelos uno detrás de otro dentro de la misma línea de comando.

```
1 | cp texto1.txt texto2.txt texto3.txt /home/usuario/carpeta_de_destino/
```

En los ejemplos anteriores, siempre hemos dado a la copia el mismo nombre que el archivo de origen. Puedes **darle el nombre que quieras al archivo nuevo** introduciéndolo en el parámetro de destino:

```
1 | cp texto.txt /home/usuario/carpeta_de_destino/texto_copia.txt
```

Si no indicas la ruta de la carpeta de destino, la copia se crea **con un nuevo nombre en la misma carpeta** que el archivo original.

```
1 | cp texto1.txt texto2.txt
```

Funciones especiales del comando CP

El comando CP de Linux también puede realizar otras tareas. Para ello, al comando se le puede añadir un término que contenga un parámetro:

(/digitalguide/)

1 | cp [Parámetro(s)] [Origen(es)] [Destino]

Puedes introducir los siguientes parámetros (flags) en el campo para que el trabajo de copia sea más complejo o más fácil.

Parámetros	Descripción	Ejemplo
-a	Copia el archivo con la misma configuración de permisos y metadatos que el original.	cp -a texto.txt /home/usuario /carpeta_de_destino/
--archive	Copia el archivo con la misma configuración de permisos y metadatos que el original.	cp --archive texto.txt /home/usuario /carpeta_de_destino/
-b	Crea una copia en la memoria intermedia (también llamada buffer) si el archivo original y el destino tienen el mismo nombre, pero diferente contenido.	cp -b texto.txt /home/usuario /carpeta_de_destino/
--backup	Crea una copia en la memoria intermedia (también llamada buffer) si el archivo original y el destino tienen el mismo nombre, pero diferente contenido.	cp --backup texto.txt /home/usuario /carpeta_de_destino/
--backup=numbered	Crea un nuevo archivo numerado si el archivo original y el de destino tienen el mismo nombre de archivo, pero diferente contenido.	cp --backup==numbered texto.txt /home/usuario /carpeta_de_destino/
--backup=existing	Crea un nuevo archivo numerado si el archivo original y el de destino tienen el mismo nombre de archivo, pero diferente contenido - y ya existen copias de seguridad numeradas.	cp --backup==existing texto.txt /home/usuario /carpeta_de_destino/
-d	Copia los enlaces simbólicos.	cp -d texto.txt /home/usuario /carpeta_de_destino/
-f	Obliga a sobrescribir al copiar.	cp -f texto.txt /home/usuario /carpeta_de_destino/texto.txt
--force	Obliga a sobrescribir al copiar.	cp --force texto.txt /home/usuario /carpeta_de_destino/texto.txt
-i	Pide permiso antes de sobrescribir archivos con el mismo nombre.	cp -i texto.txt /home/usuario /carpeta_de_destino/
--interactive	Pide permiso antes de sobrescribir archivos con el mismo nombre.	cp --interactive texto.txt /home/usuario /carpeta_de_destino/
-l	Crea un enlace duro en lugar de una copia.	cp -l texto.txt /home/usuario /carpeta_de_destino/
--link	Crea un enlace duro en lugar de una copia.	cp --link texto.txt /home/usuario /carpeta_de_destino/

Parámetros	Descripción	Ejemplo
-n	Los archivos existentes en el directorio de destino no se sobrescribirán. (/digitalguide/)	cp -n texto.txt /home/usuario/ /carpeta_de_destino/
--no-clobber	Los archivos existentes nunca se sobrescribirán.	cp --no-clobber texto.txt /home/usuario /carpeta_de_destino/
-p	Los atributos del archivo original se heredan al copiar.	cp -p texto.txt texto1.txt
--preserve	Los atributos del archivo original se heredan al copiar.	cp --preserve texto.txt texto1.txt
--preserve=mode	El modo del archivo original se hereda al copiar.	cp --preserve==mode texto.txt texto1.txt
--preserve=ownership	Las propiedades del archivo original se heredan al copiar.	cp --preserve==ownership texto.txt texto1.txt
--preserve=timestamp	La marca de tiempo del archivo original se hereda al copiar.	cp --preserve==timestamp texto.txt texto1.txt
--preserve=links	Los enlaces del archivo original se heredan al copiar.	cp --preserve==links texto.txt texto1.txt
--preserve=context	El contexto de seguridad del archivo original se hereda al copiar.	cp --preserve==context texto.txt texto1.txt
--preserve=xattr	Los atributos extendidos del archivo original se heredan al copiar.	cp --preserve==xattr texto.txt texto1.txt
--preserve=all	Todos los atributos del archivo original se heredan al copiar.	cp --preserve==all texto.txt texto1.txt
-P	Los enlaces simbólicos se guardan como tales al copiar.	cp -P texto.txt /home/usuario /carpeta_de_destino/
--no-dereference	Los enlaces simbólicos se guardan como tales al copiar.	cp --no-dereference texto.txt /home/usuario/carpeta_de_destino/
-r	Los directorios, incluidos los subdirectorios, se copian de forma recursiva.	cp -r /home/usuario /carpeta_de_origen/ /home/usuario /carpeta_de_destino/
-R	Los directorios, incluidos los subdirectorios, se copian de forma recursiva.	cp -R /home/usuario /carpeta_de_origen/ /home/usuario /carpeta_de_destino/
--recursive	Los directorios, incluidos los subdirectorios, se copian de forma recursiva.	cp --recursive /home/usuario /carpeta_de_origen/ /home/usuario /carpeta_de_destino/
-s	Crea un enlace simbólico para el archivo original.	cp -s texto.txt /home/usuario /carpeta_de_destino/
--symbolic-link	Crea un enlace simbólico para el archivo original.	cp --symbolic-link texto.txt /home/usuario /carpeta_de_destino/



Parámetros	Descripción	Digital Guide	Ejemplo
-S	Sobrescribe un sufijo de backup al copiar con --backup.	(/digitalguide/)	cp --backup=simple -S texto.txt /home/usuario /carpeta_de_destino/
--suffix=own_suffix	Sobrescribe con su propio sufijo el sufijo de backup al copiar con --backup.		cp --backup=simple --suffix=own_suffix texto.txt /home/usuario /carpeta_de_destino/
-u	Copia el archivo solamente si el archivo de destino es más antiguo que el original.		cp -u texto.txt /home/usuario /carpeta_de_destino/texto.txt
--update	Copia el archivo solamente si el archivo de destino es más antiguo que el original.		cp --update texto.txt /home/usuario /carpeta_de_destino/texto.txt
-v	Emite un mensaje al terminar la copia.		cp -v texto.txt texto1.txt
--verbose	Emite un mensaje al terminar la copia.		cp --verbose texto.txt texto1.txt

💡 Consejo

También puedes combinar varios parámetros entre sí. Para ello, simplemente escribe los parámetros separados por un espacio al principio del comando.

Ejemplos para el uso de CP

A partir del gran número de parámetros disponibles para el comando copy se deduce que el comando CP en Linux puede utilizarse para más que simplemente copiar archivos y carpetas. Combinando los distintos parámetros, se pueden abordar tareas mucho más complejas. Los siguientes tres ejemplos reflejan escenarios de aplicación típicos en la vida cotidiana de un usuario de Linux.

Backup numerado

Supongamos que estás trabajando en un archivo y quieres crear un backup del mismo de forma más o menos regular. Para que también puedas hacer un seguimiento de los cambios y posiblemente volver a un estado anterior, se crean archivos individuales. Para ello utilizamos la opción de backup ampliado.

```
1 | cp -f -v --backup=numbered texto.txt texto.txt
```

Creamos este backup con el mismo nombre de archivo y también en la misma carpeta. Para que esto sea posible, utilizamos el **parámetro "forzar"**. Como también hemos activado el parámetro "numerado", se crean nuevos archivos con nombres numerados. Gracias a "verbose", recibirás una notificación después de cada operación de copia.

En la consola aparece de esta manera:

Digital Guide



```
1 ~$ cp -v -f --backup=numbered (digitalguide).txt texto.txt
2 'texto.txt' -> 'texto.txt.~1~'
3 ~$ cp -v -f --backup=numbered texto.txt texto.txt
4 'texto.txt' -> 'texto.txt.~2~'
5 ~$ cp -v -f --backup=numbered texto.txt texto.txt
6 'texto.txt' -> 'texto.txt.~3~'
7 ~$ ls
8 texto.txt  texto.txt.~1~  texto.txt.~2~  texto.txt.~3~
```

💡 Consejo

En Linux, puedes utilizar el comando `ls` para mostrar la lista de los archivos y subcarpetas que hay dentro de la carpeta que quieras.

Clasificación de archivos con wildcards

Tienes una carpeta llena de diferentes archivos: las imágenes, aplicaciones y los documentos están mezclados entre sí. Supongamos que quieres copiar todas las imágenes a una carpeta diferente. En lugar de introducir una a una todas las imágenes en el comando, puedes trabajar con el **asterisco (*)**, que sirve de wildcard. En nuestro ejemplo, sustituimos el nombre completo del archivo por un asterisco y solo introducimos las extensiones de los archivos.

```
1 | cp *.jpg *.png /home/usuario/imágenes
```

Con este comando copiarás todos los archivos con las extensiones JPG y PNG a la carpeta de imágenes. Todos los demás archivos serán ignorados. Los archivos originales se conservan, ya que con el comando CP solo se copian los datos, no se mueven.

Copia de directorios

La **copia recursiva** incluye a una carpeta y todas sus subcarpetas. Por lo tanto, la estructura del directorio se copia entera. Esto no funciona con la copia sencilla del comando CP: si intentas copiar una carpeta sin el parámetro correspondiente, te saldrá un mensaje de error. El parámetro que debes usar es el `'-r'`, también conocido como `'--recursivo'` o `'-R'`.

En nuestro ejemplo, queremos crear regularmente una copia de seguridad de una carpeta, incluyendo su contenido y subcarpetas. Para que no haya que copiar siempre todo el contenido del directorio, puedes utilizar el **parámetro "update"**.

```
1 | cp -r -u /home/usuario/original/ /home/usuario/backups/
```

Digital Guide



Cuando ejecutes el comando por primera vez, se creará el directorio "original" dentro de la carpeta de backups. Todos los archivos y subdirectorios se te copiarán dentro de él. Si se trata de grandes cantidades de datos, esto puede llevar algún tiempo. Si utilizas el parámetro '-u', solo copiarás los elementos modificados posteriormente.

Alternativas al comando CP de Linux

A pesar de los numerosos parámetros que ofrece el comando CP, solo puede utilizarse de forma relativamente limitada. Por ejemplo, si quieres crear una copia y luego borrar el original, tienes que usar el parámetro **MV** (move). Con este comando también se introduce la ruta de origen y la ruta de destino.

El comando **DD** es algo más sofisticado que el CP. Con este comando, también tienes la opción de cambiar el formato de los archivos durante el proceso de copia. Para ello, el proceso se lleva a cabo bit a bit. El comando es útil, por ejemplo, si quieres copiar discos duros o particiones completas.

El comando **PYCP** de Linux también es interesante. Con él, obtienes una visualización gráfica durante el proceso de copia. Sin embargo, para ello necesitas Python 3.5 (o una versión más reciente). En las nuevas versiones de Ubuntu, el lenguaje de programación está preinstalado por defecto.

El [comando SCP de Linux \(servidores/configuracion/scp-de-linux/\)](#) también puede ser útil. Con Secure Copy puedes transferir datos de tu ordenador a otro o viceversa. Esto se hace a través del protocolo [SSH \(servidores/herramientas/protocolo-ssh/\)](#) que, por tanto, es un proceso encriptado.

☰ En resumen

Con Linux-CP puedes copiar archivos y carpetas de forma rápida y relativamente fácil. Mediante los múltiples parámetros disponibles, puedes realizar tareas incluso más complejas. Para llevar a cabo tareas cotidianas, también puedes utilizar la técnica de Shell Scripting. Esta técnica te permite, por ejemplo, crear un script para hacer una copia de seguridad semanal.

Para el uso diario de un terminal de Linux, debes conocer otros comandos. Por ejemplo, para simplificar el flujo de trabajo te recomendamos conocer estos comandos:

- Comando SED de Linux: un editor de texto más sencillo.
- Comando Tail de Linux: muestra las últimas líneas de un archivo.
- [Comando Head de Linux \(servidores/configuracion/linux-head/\)](#): muestra las primeras líneas de un archivo.
- Comando CAT de Linux: muestra el contenido de un archivo.



[Linux \(tags/linux/\)](#)

[\(/digitalguide/\)](#)

Artículos similares



[\(servidores/configuracion/comandos-de-linux-la-lista-fundamental/\)](#)

Los comandos de Linux más importantes (servidores/configuracion/comandos-de-linux-la-lista-fundamental/)

© 08.05.2020 | Configuración (servidores/configuracion/)

En Linux, todas las acciones que se pueden ejecutar en una interfaz gráfica de usuario con ratón y monitor se pueden llevar a cabo también en una consola con comandos, siempre y cuando se conozca la orden adecuada y su sintaxis. Para facilitarte la introducción en el trabajo con el terminal, hemos elaborado un listado con los comandos básicos de Linux que incluye una detallada descripción y...





(servidores/configuracion/eliminar-archivos-en-linux/)

Cómo eliminar archivos en Linux fácilmente (servidores/configuracion/eliminar-archivos-en-linux/)

© 07.01.2021 | Configuración (servidores/configuracion/)

Los archivos de Linux pueden eliminarse tan rápido como se crean. Para ello tienes diferentes posibilidades. Ya sea a través del gestor de archivos o del comando "rm" en la terminal, te explicamos cómo eliminar fácilmente uno o varios archivos de Linux y cómo eliminar directamente todos los archivos de un tipo en concreto con el comando "rm".



(servidores/configuracion/comando-linux-find/)

Linux find: comando para buscar y encontrar archivos en Linux (servidores/configuracion/comando-linux-find/)

© 28.05.2021 | Configuración (servidores/configuracion/)

Cuando se trabaja en un sistema Linux, la línea de comandos se utiliza con frecuencia. Muchas tareas administrativas requieren encontrar archivos y directorios basados en criterios específicos. Para ello, los administradores de Linux están acostumbrados a utilizar el comando `find`. Aquí te mostraremos cómo funciona el comando y cómo usarlo como herramienta práctica.

[Inicio](#) [Año](#) [W.ic](#) [/w](#)



Productos asociados



Servidores Cloud

[Ver tarifas ▶](#)

Hosting para agencias

Ofrece un servicio fiable y de alto rendimiento a tus clientes con un pack hosting de IONOS.

[Ver packs ▶](#)

Artículos Favoritos

<https://www.ionos.es>
IONOS Digital

[Empresa \(https://www.ionos.es/empresa\)](https://www.ionos.es/empresa) [Uso de Cookies \(https://www.ionos.es/cookies\)](https://www.ionos.es/cookies)

[Privacidad \(https://www.ionos.es/terms-gtc/terms-privacy\)](https://www.ionos.es/terms-gtc/terms-privacy)

[Centro de Ayuda \(https://www.ionos.es/ayuda\)](https://www.ionos.es/ayuda)

© 2022 IONOS Cloud S.L.U. (<https://www.ionos.es>)