

GNU/Linux para programadores:

Guía completa sobre el manejo de archivos por línea de comandos

La importancia de que un programador tenga la capacidad de moverse libremente por línea de comandos radica en algo tan simple como en recordar que en los entornos de producción (servidores) no existe un entorno gráfico. Es por ello, que aprender a manejar el Sistema Operativo por línea de comandos es indispensable para cualquier programador. En este artículo, nos centraremos en el manejo y manipulación de archivos.

Escrito por: **Eugenia Bahit** (GLAMP Hacker & eXtreme Programmer)



Eugenia es **Arquitecta de Software**, docente e instructora de tecnologías **GLAMP** (GNU/Linux, Apache, MySQL, Python y PHP) y **Agile coach** (UTN) especializada en Scrum y **eXtreme Programming**. Miembro de la **Free Software Foundation**, **The Linux Foundation** y **Debian Hackers**. Creadora de `python-printr`, `Europio Engine` y colaboradora de `Vim`.

Webs:

Cursos de programación: www.cursosdeprogramaciondistancia.com

Web personal: www.eugeniabahit.com

Redes sociales:

Twitter / Identi.ca: [@eugeniabahit](https://twitter.com/eugeniabahit)

Es muy probable que al leer la frase “manipulación de archivos” se piense en comandos como `mv`, `cp` y `rm`. Si bien son comandos que se utilizan para mover, copiar y eliminar archivos y directorios respectivamente, la manipulación de archivos no solo se limita a ellos. Por línea de comandos es necesario acceder a las mismas opciones que accedes por entorno gráfico. Desde descargar archivos de la Web, hasta comprimirlos, descomprimirlos, modificarlos o examinar sus propiedades.

En este artículo intentaremos abarcar la manipulación y manejo de archivos en el sentido más amplio yendo desde lo más básico a lo que puedan ser consideradas funciones un poco más complejas.

Lo más básico: moverse por el sistema de archivos

Ya conocerás al comando `cd` para moverte entre directorios. Sin embargo, me sucede a menudo con muchos alumnos, ver que olvidan los atajos y las buenas prácticas. Pero con `cd` se puede hacer mucho más que simplemente `cd /ruta/`. Incluso, `cd` no lo es todo. Veámoslo en detalle.

Conocer el directorio actual, te permite evitar escribir innecesarias rutas cuando deseas navegar entre directorios:

```
eugenia@cococha-gnucita:~$ pwd
/home/eugenia
```

En el caso anterior, suponiendo que tu usuario fuese `eugenia`, está claro que no necesitarás escribir `cd /home/eugenia/mi-otra-carpeta` para ingresar en esta última.

Auto-completar el nombre de archivos y directorios, reduce el riesgo de errores (y su consiguiente “volver a escribir”) así como también, ayuda a la falta de memoria, pues es muy difícil recordar rutas completas. Tras escribir los primeros caracteres de un nombre se puede completar de forma automática, pulsando la tecla `tab` una vez. Suponiendo que deseas ingresar en una carpeta llamada Documentos (o crees recordar que su nombre comienza por “Do” pero desconoces como sigue):

```
eugenia@cococha-gnucita:~$ cd Do<TAB>
# Auto-completará la palabra como
eugenia@cococha-gnucita:~$ cd Documentos/
```

La instrucción `<TAB>` indica que la tecla TAB (tabulación) debe ser pulsada

Si no se auto-completa, es señal de que existe más de un archivo o directorio en el que su nombre coincide con los caracteres escritos. En ese caso, pulsas la tecla `tab` 2 veces consecutivas, para que en pantalla te muestre las coincidencias.

```
eugenia@cococha-gnucita:~$ cd D<TAB><TAB>
Descargas/ Documentos/
eugenia@cococha-gnucita:~$ cd D
```

RECUERDA: La barra diagonal / delante del nombre de una carpeta, indica que la misma se encuentra en la raíz del sistema de archivos.

Caminar hacia adelante en el sistema de archivos es tan simple como escribir `cd carpeta1/carpeta2/etcétera`, pero para **caminar hacia atrás** puede que te interesen algunos atajos:

Retroceder un directorio

```
eugenia@cococha-gnucita:~/Documentos/documentacion/lisp$ cd ..
eugenia@cococha-gnucita:~/Documentos/documentacion$
```

Retroceder varios directorios

```
eugenia@cococha-gnucita:~/Documentos/documentacion/lisp$ cd ../../..
eugenia@cococha-gnucita:~$
```

Atajo para acceder a tu home independientemente de dónde te encuentres

```
eugenia@cococha-gnucita:~/webprojects/demo.europio.org/public/core/cli$ cd
eugenia@cococha-gnucita:~$ pwd
/home/eugenia
```

Un atajo para no escribir /home/usuario

```
eugenia@cococha-gnucita:~/webprojects/demo.europio.org/public/$ cd ~/Documentos/
eugenia@cococha-gnucita:~/Documentos$
```

Listar documentos y directorios, no se resume solo a `ls`. Algunos trucos:

Listado de documentos con información adicional

```
eugenia@cococha-gnucita:~/test/$ ls -l
total 12
-rw-r--r-- 1 eugenia eugenia 31 Apr 8 15:07 controller.php
drwxr-xr-x 2 eugenia eugenia 4096 Apr 8 15:07 core
-rw-r--r-- 1 eugenia eugenia 0 Apr 8 15:07 settings.php.dist
drwxr-xr-x 6 eugenia eugenia 4096 Apr 8 15:07 static
```

Mostrar también ocultos

```
eugenia@cococha-gnucita:~/test/$ ls -la
total 24
drwxr-xr-x 4 eugenia eugenia 4096 Apr 8 15:07 .
drwxr-xr-x 4 eugenia eugenia 4096 Apr 8 15:07 ..
-rw-r--r-- 1 eugenia eugenia 31 Apr 8 15:07 controller.php
drwxr-xr-x 2 eugenia eugenia 4096 Apr 8 15:07 core
-rw-r--r-- 1 eugenia eugenia 55 Apr 8 15:07 .htaccess
-rw-r--r-- 1 eugenia eugenia 0 Apr 8 15:07 settings.php.dist
drwxr-xr-x 6 eugenia eugenia 4096 Apr 8 15:07 static
```

Tamaño de archivo más legible

```
eugenia@cococha-gnucita:~/test/$ ls -lah
total 24K
drwxr-xr-x 4 eugenia eugenia 4.0K Apr 8 15:07 .
drwxr-xr-x 4 eugenia eugenia 4.0K Apr 8 15:07 ..
-rw-r--r-- 1 eugenia eugenia 31 Apr 8 15:07 controller.php
drwxr-xr-x 2 eugenia eugenia 4.0K Apr 8 15:07 core
-rw-r--r-- 1 eugenia eugenia 55 Apr 8 15:07 .htaccess
-rw-r--r-- 1 eugenia eugenia 0 Apr 8 15:07 settings.php.dist
drwxr-xr-x 6 eugenia eugenia 4.0K Apr 8 15:07 static
```

Listar recursivamente

```
eugenia@cococha-gnucita:~/test/$ ls -lahR
.:
total 24K
```

```

drwxr-xr-x 4 eugenia eugenia 4.0K Apr  8 15:07 .
drwxr-xr-x 4 eugenia eugenia 4.0K Apr  8 15:07 ..
-rw-r--r-- 1 eugenia eugenia  31 Apr  8 15:07 controller.php
drwxr-xr-x 2 eugenia eugenia 4.0K Apr  8 15:07 core
-rw-r--r-- 1 eugenia eugenia  55 Apr  8 15:07 .htaccess
-rw-r--r-- 1 eugenia eugenia   0 Apr  8 15:07 settings.php.dist
drwxr-xr-x 6 eugenia eugenia 4.0K Apr  8 15:07 static

./core:
total 12K
drwxr-xr-x 2 eugenia eugenia 4.0K Apr  8 15:07 .
drwxr-xr-x 4 eugenia eugenia 4.0K Apr  8 15:07 ..
-rw-r--r-- 1 eugenia eugenia  709 Apr  8 15:07 helpers.php

./static:
total 24K
drwxr-xr-x 6 eugenia eugenia 4.0K Apr  8 15:07 .
drwxr-xr-x 4 eugenia eugenia 4.0K Apr  8 15:07 ..
drwxr-xr-x 2 eugenia eugenia 4.0K Apr  8 15:07 css
drwxr-xr-x 2 eugenia eugenia 4.0K Apr  8 15:07 html
drwxr-xr-x 2 eugenia eugenia 4.0K Apr  8 15:07 img
drwxr-xr-x 2 eugenia eugenia 4.0K Apr  8 15:07 js
...
...

```

Si además se desea ordenar el listado, puede sumarse:

```

-X      Ordenar por extensión
-S      Ordenar por tamaño
-T      Ordenar por fecha y hora

```

También puede obtenerse un listado tipo árbol, instalando tree:

```

eugenia@cococha-gnucita:~/test/$ tree
.
├── controller.php
├── core
│   └── helpers.php
├── settings.php.dist
├── static
│   ├── css
│   ├── html
│   │   └── template.html
│   ├── img
│   └── js

```

Algunas opciones muy útiles de tree pueden ser las siguientes:

```

-a      Incluye ocultos
-p      Incluye permisos
-h      Incluye el tamaño de los archivos de forma legible
-u      Incluye el usuario propietario
-g      Incluye el grupo propietario

```

-L NUMERO Para indicar la cantidad máxima de niveles de dependencia a mostrar

```
eugenia@cococha-gnucita:~/test/$ tree -aphug
├── [-rw-r--r-- eugenia eugenia 31] controller.php
├── [drwxr-xr-x eugenia eugenia 4.0K] core
│   ├── [-rw-r--r-- eugenia eugenia 709] helpers.php
│   ├── [-rw-r--r-- eugenia eugenia 55] .htaccess
│   ├── [-rw-r--r-- eugenia eugenia 0] settings.php.dist
│   └── [drwxr-xr-x eugenia eugenia 4.0K] static
│       ├── [drwxr-xr-x eugenia eugenia 4.0K] css
│       ├── [drwxr-xr-x eugenia eugenia 4.0K] html
│       │   └── [-rw-r--r-- eugenia eugenia 1.4K] template.html
│       ├── [drwxr-xr-x eugenia eugenia 4.0K] img
│       └── [drwxr-xr-x eugenia eugenia 4.0K] js
```

Búsqueda y localización de archivos y programas

Para **buscar archivos** te puedes ayudar del comando `find` el cual posee la siguiente sintaxis:

```
find DONDE OPCIONES
```

DONDE, debe ser reemplazado por la ruta de base en la cuál se realizará la búsqueda. Por ejemplo, para buscar archivos en toda mi home (incluyendo los subdirectorios), escribo:

```
find /home/eugenia OPCIONES
```

Entre las opciones de búsqueda, la más frecuente es por nombre de archivo. Se puede buscar por el nombre completo:

```
eugenia@cococha-gnucita:~$ find /home/eugenia -name 'PHP Básico.odt'
/home/eugenia/Cursos/editables/PHP Básico.odt
eugenia@cococha-gnucita:~$
```

O también, realizar búsquedas por nombre parcial, utilizando comodines y expresiones regulares:

Buscará todos los archivos con extensión .odt

```
eugenia@cococha-gnucita:~$ find /home/eugenia -name '*.odt'
```

Buscará todos los archivos que comiencen por la palabra mvc (en minúsculas)

```
eugenia@cococha-gnucita:~$ find /home/eugenia -name 'mvc*'
```

Buscará todos los archivos que comiencen por la palabra mvc pero esta vez

sin distinguir mayúsculas y minúsculas

```
eugenia@cococha-gnucita:~$ find /home/eugenia -iname 'mvc*'
```

Localizar la ubicación del ejecutable de un programa mediante el comando `which`:

```
eugenia@cococha-gnucita:~$ which apache2
/usr/sbin/apache2
eugenia@cococha-gnucita:~$ which php5
/usr/bin/php5
```

Si además se desea localizar el binario más los archivos fuentes y el manual, se utiliza el comando `whereis`:

```
eugenia@cococha-gnucita:~$ whereis apache2
apache2: /usr/sbin/apache2 /etc/apache2 /usr/lib/apache2 /usr/share/apache2
/usr/share/man/man8/apache2.8.gz
eugenia@cococha-gnucita:~$ whereis python
python: /usr/bin/python2.7 /usr/bin/python /etc/python2.7 /etc/python
/usr/lib/python2.7 /usr/bin/X11/python2.7 /usr/bin/X11/python
/usr/local/lib/python2.7 /usr/include/python2.7_d /usr/include/python2.7
/usr/share/python /usr/share/man/man1/python.1.gz
```

Manipulando archivos y directorios

Para **crear directorios** se utiliza el comando `mkdir`:

Crear un único directorio

```
eugenia@cococha-gnucita:~/borrador$ mkdir un_solo_directorio
```

Crear varios directorios anidados

```
eugenia@cococha-gnucita:~/borrador$ mkdir -p varios/directorios/anidados
```

```
eugenia@cococha-gnucita:~/borrador$ tree varios
```

```
varios
├── directorios
│   └── anidados
```

2 directories, 0 files

Crear más de un directorio al mismo tiempo (no anidados)

```
eugenia@cococha-gnucita:~/borrador/varios/directorios/anidados$ mkdir dir1 dir2
```

```
eugenia@cococha-gnucita:~/borrador/varios/directorios/anidados$ tree
```

```
├── dir1
└── dir2
```

2 directories, 0 files

Para **crear archivos vacíos** se puede utilizar el comando `touch` como efecto colateral ya que dicho comando está destinado a cambiar la fecha/hora de un archivo, pero si este no existe, lo crea:

```
eugenia@cococha-gnucita:~/borrador/varios$ touch __init__.py
eugenia@cococha-gnucita:~/borrador/varios$ find ~/borrador/varios -name '__init__.py'
/home/eugenia/borrador/varios/__init__.py
```

Como alternativa, puede utilizarse `echo`, una instrucción *bash* destinada a escribir en pantalla. Dicha instrucción utilizada de forma conjunta con el carácter de direccionamiento de la salida estándar `>` ayuda indirectamente, a crear un archivo:

Crea un archivo vacío

```
eugenia@cococha-gnucita:~/borrador/varios$ echo '' > archivo.php
```

Crea un archivo con las etiquetas `<?php ?>`

```
eugenia@cococha-gnucita:~/borrador/varios$ echo '<?php ?>' > archivo.php
```

Para **eliminar archivos y directorios**:

Eliminar archivos

```
eugenia@cococha-gnucita:~/borrador/varios$ rm __init__.py
```

Eliminar directorios

```
eugenia@cococha-gnucita:~/borrador/varios$ rm -R directorios
```

Tener en cuenta que la opción `-R` siempre indica recursividad de acción en la gran mayoría de los comandos.

Es posible además, **eliminar archivos o directorios de forma segura haciéndolos irrecuperables**. Esto es muy útil cuando por ejemplo hemos guardado información vital (como contraseñas) en un archivo de texto plano y no queremos que el archivo se pueda recuperar mediante un software de recuperación de datos. Este comando sobre-escribirá el archivo tantas veces como le indiquemos y finalmente lo eliminará (es decir, elimina el archivo después de haber sido sobre-escrito N cantidad de veces):

```
eugenia@cococha-gnucita:~/borrador/uno$ shred -n 12 archivo1.copia -u
```

-n CANTIDAD *Indica la cantidad de veces que el archivo será sobre-escrito*
-u *Elimina el archivo tras sobre-escribirlo*

Para mover o **renombrar archivos y directorios** se utiliza la sintaxis:

```
mv NOMBRE_ACTUAL NUEVO_NOMBRE
```

Por ejemplo:

```
eugenia@cocochoa-gnucita:~/borrador$ mv varios nuevo_nombre
eugenia@cocochoa-gnucita:~/borrador$ find ~/borrador/ -name 'nuevo_nombre'
/home/eugenia/borrador/nuevo_nombre
eugenia@cocochoa-gnucita:~/borrador$ find ~/borrador/ -name 'varios'
eugenia@cocochoa-gnucita:~/borrador$
```

Es posible mover los archivos o directorios a una ruta distinta con el mismo nombre:

```
eugenia@cocochoa-gnucita:~/borrador$ mkdir -p uno/dos/tres/cuatro
eugenia@cocochoa-gnucita:~/borrador$ cd uno/dos/tres
eugenia@cocochoa-gnucita:~/borrador/uno/dos/tres/$ mv cuatro ../
eugenia@cocochoa-gnucita:~/borrador/uno/dos/tres/$ cd ..
eugenia@cocochoa-gnucita:~/borrador/uno/dos/$ tree
```

```
├── cuatro
└── tres
```

O moverlos a una ruta distinta con distinto nombre:

```
eugenia@cocochoa-gnucita:~/borrador/uno$ touch archivo1 archivo2
eugenia@cocochoa-gnucita:~/borrador/uno$ mv archivo2 dos/archivo_dos
eugenia@cocochoa-gnucita:~/borrador/uno$ tree dos/
dos/
├── cuatro
├── tres
└── archivo_dos
```

TIP: para repetir un comando que ya hayas ejecutado, puedes utilizar varios métodos: 1) flecha arriba y abajo para ir desde el último comando ejecutado hacia atrás y regresar; o 2) mi preferido: pulsando **ctrl + R** y **escribes los primeros caracteres de un comando ya ejecutado para realizar una búsqueda**. Si tienes que cancelar dicha búsqueda, pulsas **Ctrl + G**.

Para **copiar archivos** se utiliza la sintaxis:

```
cp ARCHIVO_ORIGINAL NOMBRE_COPIA
```

Por ejemplo:

```
eugenia@cocochoa-gnucita:~/borrador/uno$ cp archivo1 archivo1.copia
```

Mientras que para **copiar directorios** es necesario hacerlo de forma recursiva con la opción **-R**:


```
cp -R DIRECTORIO_ORIGINAL NOMBRE_COPIA
```

Por ejemplo:

```
eugenia@cococha-gnucita:~/borrador$ cp -R uno/ uno.copia
eugenia@cococha-gnucita:~/borrador$ cd uno.copia/
eugenia@cococha-gnucita:~/borrador/uno.copia/$
```

Trabajando con archivos externos

Muchas veces te será necesario **descargar un archivo desde alguna dirección de Internet**. Pero ¿cómo hacerlo sin un navegador? Es muy simple. Puedes utilizar el comando `wget`:

```
eugenia@cococha-gnucita:~/borrador$ wget
http://www.euopio.org/downloads/EuropioEngine_3_0_beta_1--app_demo.tar.gz
--2013-04-08 18:12:26--
http://www.euopio.org/downloads/EuropioEngine_3_0_beta_1--app_demo.tar.gz
Resolviendo www.euopio.org (www.euopio.org)... 66.228.52.93
Conectando con www.euopio.org (www.euopio.org)[66.228.52.93]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 109782 (107K) [application/x-gzip]
Grabando a: "EuropioEngine_3_0_beta_1--app_demo.tar.gz"

100%
[=====]
=====>] 109.782      146K/s   en 0,7s

2013-04-08 18:12:27 (146 KB/s) - "EuropioEngine_3_0_beta_1--app_demo.tar.gz"
guardado [109782/109782]

eugenia@cococha-gnucita:~/borrador$
```

También es posible **descargar varios archivos de forma recursiva** agregando el argumento `-r`:

```
wget -r http://www.euopio.org/
```

Copiar archivos mediante SSH suele ser sumamente útil cuando deseas enviar tus archivos de configuración al servidor sin tener que estar sincronizando carpetas. Para copiar archivos desde tu ordenador al servidor debes utilizar:

```
scp /ruta/de/archivo/local usuario@ip-host-server:/ruta/a/directorio/remoto
```

Por ejemplo:

```
eugenia@cococha-gnucita:~$ scp reporte_apache.html user@123.456.78.90:/home/user
reporte_apache.html          100%  34KB  34.0KB/s  00:00
eugenia@cococha-gnucita:~$
```

Y, para “descargar” archivos desde tu servidor hacia tu ordenador local, utilizas la sintaxis inversa:

```
scp user@123.456.78.90:/home/user/archivo.py /home/eugenia/
```

Empaquetado y desempaquetado

En caso que necesites **empaquetar y comprimir** todos los archivos de un directorio, puedes utilizar:

```
tar -czv DIRECTORIO > ARCHIVO.tar.gz
```

Por ejemplo:

```
eugenia@cococha-gnucita:~$ tar -czv borrador/uno/ > directorio-uno.tar.gz
borrador/uno/
borrador/uno/archivo1
borrador/uno/dos/
borrador/uno/dos/tres/
borrador/uno/dos/cuatro/
borrador/uno/dos/archivo_dos
borrador/uno/a
borrador/uno/hola
```

Luego, para **explorar un archivo sin descomprimirlo**, se puede utilizar:

```
eugenia@cococha-gnucita:~$ tar -tf directorio-uno.tar.gz
borrador/uno/
borrador/uno/archivo1
borrador/uno/dos/
borrador/uno/dos/tres/
borrador/uno/dos/cuatro/
borrador/uno/dos/archivo_dos
borrador/uno/a
borrador/uno/hola
```

Finalmente, se podrá **descomprimir un archivo** mediante:

```
eugenia@cococha-gnucita:~$ tar -xzv < directorio-uno.tar.gz
```

Resumen de comandos

Moverse por el sistema de Archivos	
Conocer el directorio actual	pwd
Auto-completar el nombre de archivos y directorios	(pulsar la tecla TAB)
Moverse hacia adelante	cd carpeta1/carpeta2
Moverse hacia atrás	cd ..
Volver a la home	cd
Abreviar /home/usuario en una ruta	~/
Listar archivos y directorios (método más óptimo)	ls -lha
Mostrar árbol de archivos y directorios (requiere instalar el paquete tree)	tree -a
Búsqueda y localización de archivos y programas	
Buscar archivos	find DONDE -iname 'PATRON'
Localizar un programa	which PROGRAMA
Ubicar un programa, su binario, fuentes y manual	whereis PROGRAMA
Manipulación de archivos y directorios	
Crear archivo	touch ARCHIVO
Crear directorios	mkdir -p DIRECTORIO_1/ DIRECTORIO_2
Eliminar archivos	rm ARCHIVO
Eliminar directorio	rm -R DIRECTORIO
Eliminación segura de archivos	shred ARCHIVO -n CANTIDAD -u
Renombrar archivos o directorios	mv ORIGINAL NUEVO_NOMBRE
Copiar archivo	cp ORIGINAL NUEVO_NOMBRE
Copiar directorio	cp -R ORIGINAL NUEVO_NOMBRE
Manejo de archivos externos	
Descargar archivo de Internet	wget URL
Descargar Web de forma recursiva	wget -r URL
Copiar archivos de local a servidor (el comando se ejecuta desde la PC local)	scp /ruta/local user@ip:/ruta/destino
Copiar archivos desde el servidor a la PC local (el comando se ejecuta desde la PC local)	scp user@ip:/ruta/server /ruta/local
Empaquetado y Desempaquetado	
Empaquetar	tar -czv DIRECTORIO > tarball.tar.gz
Explorar un <i>tarball</i>	tar -tf tarball.tar.gz
Desempaquetar en el directorio actual	tar -xzv < tarball.tar.gz
Misceláneas: historial de comandos ejecutados en la terminal	
Moverse en el historial de comandos ejecutados	(flecha arriba / flecha abajo)
Buscar un comando ejecutado, en el historial	(Teclas Ctrl + R)
Cancelar la búsqueda iniciada	(Teclas Ctrl + G)

