

Comandos gestion archivos y directorios

Linux

Carpetas en ubuntu / mint

/: Esta es la raíz.

/usr: Aquí se encuentra la gran mayoría de los archivos existentes en un sistema Linux, como documentación, ejecutables, etc.

/bin: Aquí están los comandos que pueden usar todos los usuarios (incluido el root).

/sbin: Aquí están los comandos que sólo puede usar el root.

/dev: Aquí están todos los dispositivos de nuestra máquina.

/home: Lugar donde se almacenan las cuentas de usuarios. Algo parecido a como es “mis documentos” en Windows.

/lib: Aquí están las librerías que se necesitan para el sistema.

/var: Contiene información variable, como por ejemplo los logs del sistema (*/var/log*), correo local, etc.

/tmp: Directorio temporal.

/etc: Aquí se encuentran todas las configuraciones. Por ejemplo si queremos modificar la configuración de Samba tan sólo hay que editar el archivo de texto */etc/samba/smb.conf*

/root: Cuenta del administrador.

/boot: Aquí está todo lo relacionado con el arranque del sistema.

/media: Punto de montaje para sistemas de archivos montados localmente.

/mnt: Es el predecesor de */media*, se lo conserva sólo por razones históricas

/proc: Sistema de archivos virtual de información de procesos y del kernel.

Ángel González M.

ls → (list: listar)

Nos muestra el contenido de la carpeta que le indiquemos después. La sinapsis del comando sería:

>ls [opciones] [ruta]

Opciones

-a → Muestra todos los ficheros incluyendo algunos que ordinariamente están ocultos para el usuario (aquellos que comienzan por un punto). Recordemos que el fichero punto . indica el directorio actual y el doble punto .. el directorio padre, que contiene, al actual.

-l → Esta es la opción de lista larga: muestra toda la información de cada fichero incluyendo: protecciones, tamaño y fecha de creación o del último cambio introducido,...

-c → Muestra ordenando por día y hora de creación.

-t → Muestra ordenando por día y hora de modificación.

-r → Muestra el directorio y lo ordena en orden inverso.

-R → Lista también subdirectorios.

ls subdir → Muestra el contenido del subdirectorio subdir.

-l filename → Muestra toda la información sobre el fichero filename.

--color → Muestra el contenido del directorio coloreado.

Ángel González M.

Ejemplo de uso ls

Si queremos que nos muestre lo que contiene el directorio o carpeta "/etc":

>ls /etc

Si no ponemos nada interpretará que lo que queremos ver es el contenido de la carpeta donde estamos actualmente:

>ls

Además acepta ciertos argumentos que pueden ser interesantes:

Para mostrar todos los archivos y carpetas, incluyendo los ocultos:

>ls -a

Para mostrar los archivos y carpetas junto con los permisos que tiene, lo que ocupa, su dueño, ...:

>ls -l

Además se pueden solapar los argumentos:

Si quisiéramos mostrar los archivos de la misma forma que antes, pero que muestre también los ocultos:

>ls -la

ver los ficheros de un directorio.

>ls -F

Mostrar los detalles de archivos y directorios y el tamaño en Kb, Mb, Gb o Tb.

>ls -lh

mostrar los ficheros y carpetas que contienen números.

>ls *[0-9]*

Mostrar los archivos ordenados por fecha los más recientes al final.

>ls -ltr

Mostrar los archivos por tamaño, los más grandes al final (en Kb, Mb, Gb o Tb)

>ls -lhSr

Mostrar detalles de archivos y directorios con la fecha completa

>ls -l --full-time

Ángel González M.

Como ver los archivos ocultos en el explorador de archivos

En la consola:

>ls -a

En el explorador gráfico nautilus o nemo

> sudo nemo

> sudo nautilus

> sudo thunar

Otra opción (en el interfaz gráfico)

pulsar la combinación **ctrl + h**

¿Cómo puedo crear archivos en Unix?

Con el comando "**touch**" seguido del nombre de archivo que quiero crear.

Ejemplo:

Crear un fichero simple

```
>touch miFichero.txt
```

Modificar el tiempo real (tiempo de creación) de un fichero o directorio.

```
>touch -t 0712250000 miFichero.txt
```

¿En qué directorios se guardan generalmente los programas?

Generalmente en

- /bin
- /sbin
- /usr/sbin

, aunque depende también de la distribución.

File

Este comando realiza una serie de comprobaciones en un fichero para tratar de clasificarlo, mostrando sus características.

La sinapsis del comando sería:

>file [OPCIÓN...] [ARCHIVO...]

Tras su ejecución este comando muestra el tipo del fichero e información al respecto del mismo. Este comando se puede aplicar también a directorios.

cd → (change directory: cambiar directorio)

Lo utilizamos para cambiar de directorio o carpeta en la terminal.

Podemos usarlo con rutas absolutas o relativas.

En las absolutas le indicamos toda la ruta desde la raíz (/). Por ejemplo, estemos donde estemos, si escribimos en consola ...

```
>cd /etc/apt
```

... nos llevará a esa carpeta directamente. Del mismo modo si escribimos ...

```
>cd /
```

... nos mandará a la raíz del sistema de ficheros.

Cd (continuación)

Las rutas relativas son relativas a algo, y ese algo es la carpeta donde estemos actualmente. Imaginad que estamos en /home y queremos ir a la carpeta "Imágenes" dentro de vuestra carpeta personal. Con escribir ...

>cd *Imágenes*

... nos situará allí. Como véis hemos obviado el "/home/carpeta_personal" inicial ya que si no lo introducimos toma como referencia el directorio donde estamos, que es ese.

¿Y qué sucede si escribimos tan sólo ...

>cd

Sí, sólo "cd". Esto lo que hace es que te lleva a tu carpeta personal directamente, estemos donde estemos. Es algo realmente muy práctico, muy simple y que no todos conocen.

Ejemplos de uso del comando cd

entrar en el directorio '/home'.

```
>cd /home
```

retroceder un nivel.

```
>cd ..
```

retroceder 2 niveles.

```
>cd ../../
```

ir al directorio raíz /.

```
>cd /
```

Ir al directorio home del usuario actual '/home/\$USERNAME/'

```
>cd
```

ir al directorio user1 del home del usuario.

```
>cd ~user1
```

al directorio anterior.

```
>cd – ir (regresar)
```

mkdir → (make directory: hacer directorio)

Crea una carpeta o directorio con el nombre que le indiquemos.

Nuevamente podemos usar rutas absolutas y relativas. Podemos indicarle toda la ruta que le precede al directorio que queremos crear:

```
>mkdir /home/carpeta_personal/nueva_carpeta
```

O si estamos ya en la carpeta que lo va a contener basta con poner tan sólo el nombre de la nueva carpeta. Por ej. si ya estamos en /home/carpeta_personal:

```
>mkdir nueva_carpeta
```

Ejemplo uso mkdir

>mkdir dir1: crear una carpeta o directorio con nombre 'dir1'.

>mkdir dir1 dir2: crear dos carpetas o directorios simultáneamente (Crear dos directorios a la vez).

>mkdir -p /tmp/dir1/dir2: crear la ruta de directorios, cualquiera que no exista se crea.

>mkdir -m 777 dir1 : crear un directorio y asignar los permisos para ese directorio

rm → (remove: borrar)

Borra el archivo o la carpeta que le indiquemos.

Como antes se puede indicar la ruta completa o el nombre del archivo. Esto a partir de ahora lo vamos a obviar, creo que ya ha quedado claro con los dos comandos anteriores.

Para borrar un archivo:

>rm nombre_archivo

Para borrar un directorio o carpeta vacía:

>rm nombre_carpeta

Para borrar un directorio o carpeta que contiene archivos y/o otras carpetas que pueden, a su vez, contener más carpetas y archivos:

>rm -r nombre_carpeta

rm (continuación)

Otras opciones:

"-f", no te pide una confirmación para eliminar.

"-v", va mostrando lo que va borrando.

¿Cómo borrar directorios con todos sus archivos dentro?

`rm -r`

Borrar respondiendo Si

Si solicita confirmación, responder yes a todos los archivos y borrarlos sin demora.

```
>yes | rm archivo*
```

rmmdir (borrar directorios)

También existe el comando "rmmdir" para borrar carpetas o directorios:

>rmmdir nombre_directorio

Pero solo borrará directorios vacíos. Para borrar un directorio no vacío, junto con todo lo que tenga debajo, emplear "rm -r".

cp → (copy: copiar)

Copia el archivo o directorio indicado donde le digamos.

Aquí podemos también jugar con las rutas, tanto para el fichero origen, como en el del destino. También podéis cambiar el nombre que le queréis poner a la copia.

La sinapsis del comando sería:

>cp [/ruta/de/original...] [/ruta/de/copia...]

Por ejemplo, en nuestra carpeta personal vamos a crear una copia de seguridad "sources.list.backup", de nuestros repositorios "/etc/apt/sources.list". Lo voy a explicar según donde estemos colocados en la terminal, para comprender lo primordial que es saber en todo momento el directorio donde estamos colocados en la terminal:

cp (continuación)

- Si estamos colocados en nuestra carpeta personal, debemos de poner la ruta absoluta del original y la ruta relativa de la copia:

>cp /etc/apt/sources.list sources.list.backup

- Si nos colocamos en el directorio que contiene el archivo original (cd /etc/apt), debemos de poner la ruta relativa del original y la ruta absoluta de la copia:

>cp sources.list /home/tu_usuario/sources.list.backup

Nota: no olvides cambiar "tu_usuario" por el nombre de tu usuario o en su defecto sustituye "/home/tu_usuario" por el símbolo "~" (pulsar la combinación de teclas Alt Gr+Ñ). Sería así:

>cp sources.list ~/sources.list.backup

- Si estuviéramos en cualquier otro directorio o simplemente para no tener problemas, escribimos las dos rutas absolutas:

>cp /etc/apt/sources.list /home/tu_usuario/sources.list.backup

Nota: no olvides cambiar "tu_usuario" por el nombre de tu usuario o en su defecto sustituye "/home/tu_usuario" por el símbolo "~" (pulsar la combinación de teclas Alt Gr+Ñ). Sería así:

>cp /etc/apt/sources.list ~/sources.list.backup

mv → (move: mover)

Es igual que el anterior, sólo que en lugar de hacer una copia, mueve directamente el archivo con el nombre que le indiquemos, pudiendo ser otro distinto al original:

La sinapsis del comando sería idéntica a copiar:

>mv [/ruta/de/original...] [/ruta/de/destino...]

mv (continuación)

Ejemplo para mover un "archivo.flv" del directorio "/tmp" (temporales) a nuestra carpeta personal y de paso cambiarle el nombre a "mi_archivo.flv". Lo pongo con las dos rutas absolutas para no repetir todo lo anterior.

```
>mv /etc/archivo.flv /home/tu_carpeta/mi_archivo.flv
```

O en su defecto sustituye "/home/tu_usuario" por el símbolo "~" (Alt Gr+Ñ)

Otro uso muy práctico que se le puede dar es para renombrar un archivo. Basta con indicar el nuevo nombre en el segundo argumento con la misma ruta del primero. En este ejemplo suponemos que ya estamos en la carpeta que lo contiene:

```
>mv archivo.flv mi_archivo.flv
```

`pwd` → (print working directory)

Visualiza o imprime la ruta del directorio en el que nos encontramos en este momento. Este comando es uno de los pocos que no tiene

opciones y se utiliza escribiendo simplemente:

```
>pwd
```

Which

El comando which nos sirve para averiguar dónde se encuentra instalado un determinado programa.

>which find

/usr/bin/find

Ésto nos mostrará la primera aparición del programa buscado. Si queremos que nos muestre todas las ocurrencias que encuentre, utilizaremos el parámetro -a.

>which -a find

/usr/bin/find

/usr/bin/X11/find

whereis

Este comando se utiliza para localizar el archivo binario, el código fuente y la página de manual de un determinado comando. Su sintaxis es como sigue:

whereis [opciones] archivo...

La lista de opciones mas utilizadas es:

- -b: Buscar solamente el archivo binario.
- -m: Buscar solamente la página manual.
- -s: Buscar solamente el código fuente.

Ejemplos:

Pedimos donde está situado el manual del comando ls

>whereis -m ls

Ahora pedimos donde se encuentra el binario (ejecutable) del comando ls

>whereis -b ls

En este ejemplo se han pedido todos los archivos que tengan que ver con el comando passwd

>whereis passwd

Ángel González M.

find → (find: encontrar)

Busca archivos o carpetas en la ruta que le indiques:

La sinapsis del comando sería:

>find [/directorio/donde/buscar...] [-expresión] [búsqueda]

Donde "expresión" es el tipo de búsqueda y siempre se le antepone el signo "-"

find (continuación)

La expresión "-name" sería para realizar una búsqueda por nombre. Por ejemplo, para buscar en todo el sistema de archivos o raíz "/" las carpetas y archivos que se llamen "pepino". Sería:

>find / -name palabra

Si tuviéramos la seguridad de que se encuentra en /var por ejemplo, se lo indicaríamos:

>find /var -name palabra

Si no estamos muy seguros del nombre podemos indicárselo con comodines. Supongamos que el nombre de lo que buscamos contiene "pepi", en la misma carpeta de antes:

>find /var -name *palab*

Otra expresión sería "-size" para realizar la búsqueda por tamaño. Por ejemplo podemos decirle que encuentre los archivos/carpetas de más de 1500 KB:

>find / -size +1500

Se pueden combinar varios atributos para afinar la búsqueda. Por ejemplo, buscar los archivos/carpetas que contienen el nombre "pepi" y tienen menos de 1000 KB:

>find / -name *palab* -size -1000

La opción "2>/dev/null" es muy interesante para que no muestre los errores de "Permiso denegado". Por ejemplo para buscar en la raíz "/" el archivo "gdmflexiserver":

>find / -name gdmflexiserver 2>/dev/null

Ángel González M.

Locate (introducción)

En Linux en muchas ocasiones usamos comandos de búsquedas, suele ser habitual el uso del comando find pero suele ser algo lento, hay que hacer bastantes filtrados, y sobrecarga el sistema bastante. En su lugar tenemos el comando Locate, que hace búsquedas instantáneas, tiene algunas peculiaridades pero si lo conocéis lo adoráis y si no, creo que es uno de esos comandos imprescindibles en Linux.

En primer lugar hay que tener en cuenta que para realizar las búsquedas usa una base de datos, y sí como lo estáis leyendo, esa base de datos tiene toda la información y este comando la consulta a ella directamente, se encuentra en esta ruta: ***/var/lib/mlocate/mlocate.db***

Ahora bien la base de datos tiene información relativamente reciente, ya que se actualiza una vez al día, no obstante se puede forzar la actualización mediante un updatedb que escanea el sistema entero y actualiza el fichero de la base de datos, mlocate.db se puede modificar la configuración de las variables que usa en el fichero: ***>cat /etc/updatedb.conf***

```
PRUNE_BIND_MOUNTS="yes"
# PRUNENAMES=".git .bzr .hg .svn"
PRUNEPATHS="/tmp /var/spool /media /home/.ecryptfs"
PRUNEFS="NFS nfs nfs4 rpc_pipefs afs binfmt_misc proc smbfs autofs iso9660 ncpfs coda devpts ftpts devfs mfs shfs sysfs cifs lustre_lite tmpfs usbfs udf fuse.glusterfs fuse.sshfs
curlftpts ecryptfs fusesmb devtmpfs"
```

Este es un resumen de lo que tiene:

- PRUNEFS – Lista los filesystem que no serán escaneados. Por defecto no obstante no se salva ninguno. Lee el fichero /etc/mtab
- PRUNENAMES – Lista de nombres de directorios que no serán escaneados, no hace falta el path entero.
- PRUNEPATHS – Lista de rutas que usan los directorios que no serán escaneados.
- PRUNE_BIND_MOUNTS – puede tomar valores 0 ó 1, si es 1 los montajes “bind” (ya lo explicaremos más adelante) no son escaneados.

Ángel González M.

Locate (uso)

Búsqueda de un fichero simple

```
>locate httpd.conf
```

```
/etc/httpd/conf/httpd.conf
```

Desplegar número de ficheros que se llaman igual

```
> locate -c httpd.conf
```

```
1
```

Ignorar mayúsculas con Locate

```
>locate -i test.txt
```

more

more [archivo]

El comando `more` accede a la página seleccionada de un archivo de texto página tras página. Pulsando la tecla <Intro> puedes pasar las páginas hasta llegar al final del documento. Para salir puedes pulsar `ctrl+c`

Normalmente se usa con una tubería:

>cat archivo.txt | more

>ls | more

head

head -# [archivo]

head accede a las x **primeras** líneas del archivo de texto elegido. Sustituya # por el número de líneas que desea mostrar en head.

>head -2 fichero.txt

se suele usar en forma de tubería

>cat fichero.txt | head -2

tail

tail -# [archivo]

tail es parecido a head, pero accede a las x **últimas** líneas del archivo de texto elegido. Sustituya # por el número de líneas que desea mostrar en head.

>tail -2 fichero.txt

se suele usar en forma de tubería

>cat fichero.txt | tail -2

grep → (localizar)

El comando grep localiza una palabra, clave o frase en un conjunto de directorios, indicando en cuáles de ellos la ha encontrado. Este comando rastrea fichero por fichero, por turno, imprimiendo aquellas líneas que contienen el conjunto de caracteres buscado. Si el conjunto de caracteres a buscar está compuesto por dos o más palabras separadas por un espacio, se colocará el conjunto de caracteres entre apóstrofes ('). S

La sinapsis del comando sería:

```
>grep [OPCIÓN] 'conjuntocaracteres' [ARCHIVOS...]
```

siendo 'conjuntocaracteres' la secuencia de caracteres a buscar, y file1, file2, y file3 los ficheros donde se debe buscar. Veamos Ejemplo para buscar TRIANGULARIZACION MATRIZ entre las líneas de los ficheros matrix.f y scaling.f.:

```
>grep 'TRIANGULARIZACION MATRIZ' matrix.f scaling.f
```

grep (continuación)

Las opciones principales del comando son:

- c → lo único que se hace es escribir el número de las líneas que satisfacen la condición.
- i → no se distinguen mayúsculas y minúsculas.
- l → se escriben los nombres de los ficheros que contienen líneas buscadas.
- n → cada línea es precedida por su número en el fichero.
- s → no se vuelcan los mensajes que indican que un fichero no se puede abrir.
- v → se muestran sólo las líneas que no satisfacen el criterio de selección.

A continuación se muestra una serie de ejemplos.

- >grep “d” text** → líneas que comienzan por d.
- >grep “[^d]” text** → líneas que no comienzan por d.
- >grep -v “C” file1 > file2** → quita las líneas de file1 que comienzan por C y lo copia en file2.

cat → (Visualización sin formato de un fichero)

Este comando permite visualizar el contenido de uno o más ficheros de forma no formateada. También permite copiar uno o más ficheros

como apéndice de otro ya existente. Algunas formas de utilizar este comando son las siguientes:

Sacar por pantalla el contenido del fichero filename:

>cat filename

Sacar por pantalla, secuencialmente y según el orden especificado, el contenido de los ficheros indicados (file1 y file2):

>cat file1 file2

Aceptar lo que se introduce por el teclado y lo almacena en file1 (se crea file1):

>cat >file1

¿Cómo borro un archivo sin posibilidad de recuperación?

con shred

Ej

```
shred -n 10 -uvz fichero.txt
```

iconv: Listado de cifrados conocidos

listas de cifrados conocidos.

```
>iconv -l
```

crea una nueva forma del fichero de entrada asumiendo que está codificado en fromEncoding y convirtiéndolo a ToEncoding.

```
>iconv -f fromEncoding -t toEncoding inputFile > outputFile:
```

Establecer caracter y conversión de ficheros

dos2unix filedos.txt fileunix.txt: convertir un formato de fichero texto desde MSDOS a UNIX.

unix2dos fileunix.txt filedos.txt: convertir un formato de fichero de texto desde UNIX a MSDOS.

recode ..HTML < page.txt > page.html: convertir un fichero de texto en html.

recode -l | more: mostrar todas las conversiones de formato disponibles.

Editores de archivos

Hay varias opciones cuando se trata editar, ya que hay muchos editores todos ellos disponibles a través del terminal. Usaremos **gedit**.

Para iniciar un archivo de texto en gedit simplemente introduzca:

gedit *[archivo]*

No se olvide de elegir el directorio correcto y, por supuesto, sustituir *[archivo]* con el título correcto.

Otros editores posibles son nano y vi, y pueden abrirse con la misma estructura de mando que gedit:

nano *[archivo]*

pico *[archivo]*

vi *[archivo]*

vim *[archivo]*