Gestión de archivos en la terminal de Ubuntu

Visualización de archivos y elementos

ls muestra los directorios y archivos existentes en la ubicación

ls [opciones] [ruta]

ls –c ordenado por fecha de creación

ls –t ordenado por fecha de modificación

ls –l con detalles de cada archivo

ls –lsh con tamaño de cada archivo

ls –ld con propiedades de cada archivo

ls –a incluyendo archivos ocultos

ls –R de manera recursiva a los directorios que existan

ls | wc -l

tree similar al anterior en forma de árbol

file muestra las características de un archivo

file [OPCIÓN...] [ARCHIVO...]

Creación de directorios

mkdir ruta nombredeldirectorio

mkdir –p nombredirectorio1/nombredirectorio2 para crearlos recursivamente

mkdir –p nombredirectorio2/{nombredirectorio2,nombredirectorio3}

por ejemplo:

mkdir –p Música/blues/{80,90,00}

Borrado de directorios

rmdir borra directorios vacíos

rmdir –p borra subdirectorios

por ejemplo:

rmdir –p Música/blues/00

Navegación entre directorios

cd nos lleva al directorio home de usuario (sería lo mismo que cd ~/)

cd rutaabsoluta nos lleva a ese directorio

Por ejemplo:

cd /home/pedro/Música

cd / nos llevaría al directorio raíz

cd rutarelativa

Por ejemplo:

mkdir jazz

cd jazz

cd ../../Descargas

las rutas en Linux se separan por barra inclinada (no la contrabarra o barra invertida en Windows).

En las rutas relativas el uso de los dos puntos determina subir un nivel en el árbol de directorios, por ello, para subir un nivel ponemos:

cd ..

¿Para qué sirve el punto barra ./?

Cuando llamamos a archivos ejecutables que no se encuentran en las rutas de la variable PATH, necesitamos hacerlo incluyendo la ruta absoluta del archivo, aunque estemos en el directorio que lo contiene.

Para no tener que poner toda la ruta podemos poner ./ que hace referencia a la ruta absoluta en la que estamos.

Con cd y dos veces el tabulador podemos ir viendo los directorios que tenemos para navegar al deseado:

Por ejemplo:

cd pedro/ doble tabulador y nos muestra los directorios

Creación de archivos

touch crea un fichero vacío o cambia la fecha de uno existente

Por ejemplo:

touch holamundo.txt

echo sirve para introducir un mensaje.

Podemos usarlo para introducir un texto, por ejemplo:

echo ¡Hola Mundo! > holamundo.txt

Visualización de Archivos

cat muestra el contenido del fichero

cat holamundo.txt

Edición de Archivos

Podemos usar el editor nano

nano nombredelarchivo

Borrado de archivos

rm borra archivos

rm -i pide confirmación

rm –r borra directorios y archivos de manera recursiva

rm –f no solicita confirmación para el borrado

rm –d borra directorios vacíos (similar a rmdir)

rm –v muestra el proceso de borrado

Copia de archivos y directorios

cp copia archivos y directorios

`cp [opciones] ficheroorigen ficherodestino`

`cp [opciones] ficheroorigen directoriodestino`

cd Imágenes

cp jovendelaperla.png ../Descargas/

Otro ejemplo sería copiar en nuestro home usando la virgulilla ~(altgr + ñ) como atajo del home de usuario:

cd Descargas

cp jovendelaperla.png ~/

Si por el contrario queremos copiar varios ficheros, habría que hacer lo siguiente:

touch log1.txt log2.txt log3.txt

mkdir logs

cp log1.txt log2.txt log3.txt /home/pedro/logs/

En caso de que queramos pasar todos los ficheros que se encuentran dentro de un directorio a otro, haremos lo siguiente:

cp /home/pedro/logs/ * /home/pedro/Descargas/

Para copiar un directorio entero (con sus archivos) empleamos

cp –R /home/pedro/logs/ /home/pedro/Descargas/

Mover archivos y directorios

mv similar al anterior para mover archivos o directorios (con la mismas opciones)

Para renombrar usamos también mv, con la sintaxis:

mv archivo archivorenombrado

mv directorio directoriorenombrado

Enlaces simbólicos

Los enlaces simbólicos se pueden definir como los accesos directos en Sistemas Operativos Windows, es decir, su ejecución redirige a otros archivos o directorios.

Para crear un enlace simbólico empleamos

ln –s nombredirectorio nombrenlace

Por ejemplo, en home creamos:

ln –s Música/jazz jz

y listamos con ls –l para comprobar el enlace.

Podemos acceder al enlace simbólico jazz con cd y comprobar que si creamos un archivo, realmente lo estaremos haciendo en el directorio jazz

Nota: Para eliminar un enlace simbólico emplearemos rm no rmdir.

Búsqueda de Ficheros

find

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

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

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 "log.txt”:

find ~/ -name log.txt

Podemos usar asterisco:

find ~/ -name *og*

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

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 "log1.txt":

find / -name log1.txt 2>/dev/null

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 comillas simples (').

La sinapsis del comando sería:

```
grep [OPCIÓN] 'conjuntocaracteres' file1 file2 file3
```

siendo 'conjuntocaracteres' la secuencia de caracteres a buscar, y file1, file2, y file3 los ficheros donde se debe buscar.

Por ejemplo:

touch log1.txt => Escribimos Lorem ipsum…

touch log2.txt => Escribimos En un lugar de la Mancha…

grep ‘Mancha’ log1.txt log2.txt *.* para todos los archivos

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.

Varios

pwd nos devuelve la ruta desde el directorio raíz

clear limpia la Shell

reset resetea la terminal

history muestra el historial de comandos

history -c borra el historial de comandos

hostname nombre de la máquina

who usuarios y terminales virtual

whoami qué usuario soy

date fecha

cal calendario

logout cierra sesión

reboot reinicia el equipo

halt cierra el equipo (exige sudo)

poweroff ídem

shutdown opciones ídem, por ejemplo:

sudo shutdown -r now

sudo shutdown -r +5

sudo shudown -r 22:30

Comandos Internos (no generan PID)

alias permite establecer un alias de un comando que tenga varias opciones para escribirlo más rápido. La sintaxis para generar nuevos alias es:

alias nombrecomando = ‘comando opciones’

El comando va entre comillas simples. Pero la pregunta es ¿Dónde ponemos esto? Pues si queremos que solo sea temporal, simplemente lo escribimos en la consola y durará hasta que la cerremos.

Ahora, si lo queremos de forma permanente, esto lo ponemos dentro del fichero ~/.bashrc el cual está en nuestro /home, y si no está, pues lo creamos (siempre con el punto delante).

Si escribimos alias nos mostrará todos los alias existentes.

echo muestra el valor de lo que pongamos a continuación

exec ejecuta un programa con el mismo PID de la consola, con lo cual al finalizar este programa finaliza también la consola.

env muestra todas las variables de entorno.

Este comando se puede emplear para cambiar el valor de una variable en una aplicación con la sintaxis:

env VARIABLE=nuevovalor binarioprograma

export crea una variable local

export VARIABLE='valor'
echo $VARIABLE

Variable de entorno PATH

Path es una variable de entorno para establecer la ruta de un ejecutable dentro del sistema, de tal manera que los ejecutables que estén en las rutas definidas en esta variable puedan ser ejecutados desde cualquier directorio del sistema.

La forma de almacenar las diferentes rutas en path es separándolas por dos puntos :.

Para ver la variable PATH escribimos:

printenv PATH

Si queremos ver donde se encuentra un ejecutable podemos emplear:

which ejecutable

y nos devolverá su ruta. Si esta se encuentra en PATH lo podremos ejecutar desde cualquier punto.

Vamos a crear un ejecutable y añadimos su ruta a PATH

mkdir misbin

touch holamundo

nano holamundo

Y escribimos:

echo “¡Hola Mundo!”

Guardamos y salimos y le cambiamos los permisos con

chmod 777 holamundo

Y lo ejecutamos de manera local con

./holamundo

Para introducir la ruta de este ejecutable en la variable PATH emplearemos:

export PATH=$PATH:/home/pedro/misbin

Ahora comprobamos como se puede ejecutar el archivo desde cualquier ubicación simplemente con:

holamundo