

LINUX Comandos vía terminal (Tema 13)

Comando **cd** , Este comando se usa para cambiar de directorio. Generalmente cuando el usuario inicia una sesión en GNU/Linux, el directorio donde comienza es su directorio personal. Desde ahí uno puede moverse a los diferentes directorios donde se tenga acceso usando este comando. Su sintaxis es la siguiente:

- **cd directorio** Va al directorio definido, dentro del que se esta.

```
alberto@alberto-desktop:~$ cd asir
alberto@alberto-desktop:~/asir$ █
```

- **cd** Va al directorio raíz del usuario, generalmente /home/usuario.

- **cd ..** Sube un nivel en el directorio.

```
alberto@alberto-desktop:~/profes$ cd ..
alberto@alberto-desktop:~$ █
```

- **cd ../..** Sube 2 niveles en el directorio.

```
alberto@alberto-desktop:~/asir/alumno1$ cd ../..
alberto@alberto-desktop:~$ █
```

- **.** Hace referencia al directorio actual.

- **..** Hace referencia al directorio padre.

Ruta relativa: Si la ruta no comienza con / entonces se considera relativa a la ruta actual.

Ruta absoluta: Si la ruta comienza con / entonces hará referencia a una ruta en el directorio raíz.

Referirse al directorio `bin/` no es lo mismo que referirse al directorio `/bin/`, si uno se encuentra en el directorio `/usr/`, la primera instancia corresponde a `/usr/bin/` mientras que la segunda corresponde a `/bin/`.

Comando **ls** , Este comando se usa para listar archivos y directorios de un determinado directorio, según la opción se lista de una determinada manera la información:

- **ls** Sin argumentos muestra el listado del directorio actual.

```
alberto@alberto-desktop:~$ ls
Descargas  Escritorio  Imágenes  Plantillas  Vídeos
Documentos examples.desktop  Música    Público
```

- **ls -l** Muestra un listado en el formato largo, con información de permisos, número de enlaces asociados al archivo, usuario, grupo, tamaño y fecha de última modificación además del nombre.

```
alberto@alberto-desktop:~$ ls -l
total 40
drwxr-xr-x 3 alberto alberto 4096 2012-02-06 12:53 asir
drwxr-xr-x 3 alberto alberto 4096 2012-01-25 22:41 Descargas
drwxr-xr-x 2 alberto alberto 4096 2012-01-25 22:32 Documentos
drwxr-xr-x 2 alberto alberto 4096 2012-01-25 22:32 Escritorio
-rw-r--r-- 1 alberto alberto 179 2012-01-25 22:20 examples.desktop
drwxr-xr-x 2 alberto alberto 4096 2012-01-25 22:32 Imágenes
drwxr-xr-x 2 alberto alberto 4096 2012-01-25 22:32 Música
drwxr-xr-x 2 alberto alberto 4096 2012-01-25 22:32 Plantillas
drwxr-xr-x 2 alberto alberto 4096 2012-01-25 22:32 Público
drwxr-xr-x 2 alberto alberto 4096 2012-01-25 22:32 Vídeos
```

- **ls -li** Muestra el número del [i-nodo](#) antes del nombre de archivo y junto con l el formato largo (ls -li)

```
alberto@alberto-desktop:~$ ls -li
total 36
395439 drwxr-xr-x 3 alberto alberto 4096 2012-01-25 22:41 Descargas
395759 drwxr-xr-x 2 alberto alberto 4096 2012-01-25 22:32 Documentos
395438 drwxr-xr-x 2 alberto alberto 4096 2012-01-25 22:32 Escritorio
327660 -rw-r--r-- 1 alberto alberto 179 2012-01-25 22:20 examples.desktop
396286 drwxr-xr-x 2 alberto alberto 4096 2012-01-25 22:32 Imágenes
395888 drwxr-xr-x 2 alberto alberto 4096 2012-01-25 22:32 Música
395440 drwxr-xr-x 2 alberto alberto 4096 2012-01-25 22:32 Plantillas
395441 drwxr-xr-x 2 alberto alberto 4096 2012-01-25 22:32 Público
398755 drwxr-xr-x 2 alberto alberto 4096 2012-01-25 22:32 Vídeos
```

- **ls -a** Muestra los archivos ocultos.

```
alberto@alberto-desktop:~$ ls -a
.          .dmrc          .gvfs          .pulse-cookie
..         Documentos    .ICEauthority  .recently-used.xbel
asir       Escritorio     Imágenes       .sudo_as_admin_successful
.bash_history .esd_auth     .local         .update-notifier
.bash_logout examples.desktop Música          Vídeos
.bashrc    .gconf        .nautilus      .xsession-errors
.cache     .gconfd       Plantillas     .xsession-errors.old
.config    .gnome2       .profile
.dbus      .gnome2_private Público
Descargas .gtk-bookmarks .pulse
```

- **ls -s** Muestra delante del nombre del fichero el tamaño en kilobytes del mismo.

```
alberto@alberto-desktop:~/asir$ ls -hs
total 8,0K
  0 ajercicio1.txt    0 apuntes.txt      0 ejercicio3.txt
4,0K alumno1        0 ejercicio2.txt  4,0K fich1.txt
```

- **ls -R** Hace un listado recursivo. Lista primero los archivos del directorio en curso, luego subdirectorios de éste, luego los subdirectorios contenidos en ellos (nietos) y así sucesivamente.

```
alberto@alberto-desktop:~/asir$ ls -R
.:
alumno

./alumno:
alumno1 apuntes.txt
```

- **ls -l /home/a*** Lista directorios que empiecen por 'a' en este caso y su contenido, del directorio home.

- **ls -l *a** Lista directorios que terminan por la letra 'a' en este caso y su contenido.

- **ls -l [pn]*** Lista directorios que empiezan por 'p' y 'n' en este caso y su contenido.

- **ls -l *[ao]** Lista directorios que terminan por 'a' y 'o' en este caso y su contenido.

Se puede también juntar opciones, como por ejemplo:

- **ls -lia** Que muestra el i-nodo, permisos y archivos ocultos.

```
alberto@alberto-desktop:~$ ls -lia
total 152
327657 drwxr-xr-x 23 alberto alberto 4096 2012-02-06 12:53 .
129801 drwxr-xr-x  3 root    root    4096 2012-01-25 22:20 ..
419021 drwxr-xr-x  3 alberto alberto 4096 2012-02-06 12:53 asir
260516 -rw-----  1 alberto alberto 3962 2012-02-02 12:16 .bash_history
327661 -rw-r--r--  1 alberto alberto  220 2012-01-25 22:20 .bash_logout
327658 -rw-r--r--  1 alberto alberto 3103 2012-01-25 22:20 .bashrc
260505 drwx-----  4 alberto alberto 4096 2012-02-06 11:35 .cache
398846 drwxr-xr-x  6 alberto alberto 4096 2012-01-31 14:04 .config
399268 drwx-----  3 alberto alberto 4096 2012-01-25 22:32 .dbus
```

Otras opciones del comando ls:

- **-h** con **-l** imprime el tamaño de los archivos de forma entendible para los humanos (ej. 1K 234M 2G).
- **-d** muestra solamente el nombre del subdirectorio, sin entrar en él ni dar un listado del contenido.
- **-t** muestra ordenado por la fecha de última modificación.
- **-c** muestra ordenado por la fecha de última modificación del estado del archivo.
- **-r** cuando el listado se hace por orden temporal, los archivos más recientes van al principio. Si se indica **-r** se invierte el orden, mostrando los más recientes al final.
- **-L** en los enlaces simbólicos, muestra los datos del archivo referenciado en vez de los del link.
- **-m** muestra los archivos en una línea y separados por comas.
- **--color** muestra cada tipo de archivo de un color distinto: un color para los directorios, otro para los archivos regulares, otro para los enlaces simbólicos, otro para los sockets, otro para las tuberías FIFO, etc. Este parámetro no se acepta en todas las versiones de **ls** y, por supuesto, requiere que la terminal sea capaz de mostrar distintos colores o intensidades.

Comando **dir** , similar a **ls** pero no diferencia directorios de archivos en colores.

```
alberto@alberto-desktop:~$ dir
Descargas  Escritorio      Imágenes  Plantillas  Vídeos
Documentos examples.desktop Música      Público
```

Comando **who** , muestra rápidamente los usuarios que actualmente están autenticados en el sistema

```
alberto@alberto-desktop:~/profes$ who
alberto tty7      2012-01-26 11:41 (:0)
alberto pts/0     2012-01-26 11:43 (:0.0)
```

Comando **whoami** , muestra el nombre del usuario.

```
alberto@alberto-desktop:~/profes$ whoami
alberto
```

Comand **hostname**, muestra el nombre de ordenador.

```
alberto@alberto-desktop:~/profes$ hostname
alberto-desktop
```

Comando **pwd** , indica donde estas (directorio de trabajo actual), indicando la ruta completa.

```
alberto@alberto-desktop:~$ pwd
/home/alberto
```

Comando **id** muestra información del usuario.

```
alberto@alberto-desktop:~$ id
uid=1000(alberto) gid=1000(alberto) grupos=4(adm),20(dialout),24(cdrom),46(plugdev),105(lpadmin),119(admin),122(sambashare),1000(alberto)
```

Comando **echo texto** , escribe el texto escrito en pantalla.

```
alberto@alberto-desktop:~/profes$ echo Hola mundo
Hola mundo
```

Comando **date** , muestra la fecha y hora.

```
alberto@alberto-desktop:~/profes$ date
jue ene 26 12:20:21 CET 2012
```

Comando **date > nombre fichero** , guarda los datos de date en el fichero indicado.

```
alberto@alberto-desktop:~/asir$ date > fich1.txt
```

Vemos que ha creado el archivo y el contenido que le ha metido:

```
alberto@alberto-desktop:~/asir$ ls
ajercicio1.txt alumno1 apuntes.txt ejercicio2.txt ejercicio3.txt fich1.txt
alberto@alberto-desktop:~/asir$ cat fich1.txt
mar ene 31 13:00:02 CET 2012
```

date con opciones para que se visualice con un formato determinado.

```
alberto@alberto-desktop:~/asir$ date +"son las %r del %d de %h de %y"
son las 01:08:48 del 31 de ene de 12
alberto@alberto-desktop:~/asir$ date +"son las %r del %d de %h de %Y"
son las 01:10:10 del 31 de ene de 2012
```

Comando **cal** , Muestra el calendario del mes en curso.

```
alberto@alberto-desktop:~/profes$ cal
Enero 2012
lu ma mi ju vi sa do
                1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
```

Comando **cmp fichero1 fichero2** , Compara dos archivos con las opciones especificadas.

```
alberto@alberto-desktop:~/asir$ cmp fich1.txt fich2.txt
fich1.txt fich2.txt son distintos: byte 20, línea 1
```

Comando **mkdir** , Para creación de Directorios.

```
alberto@alberto-desktop:~$ mkdir asir
alberto@alberto-desktop:~$ ls
asir      Documentos  examples.desktop  Música      Público
Descargas Escritorio  Imágenes          Plantillas  Vídeos
```

Para crear directorios con nombre compuesto y espacios (“Carpeta sin titulo”), se indica con “\” cada espacio.

```
alberto@alberto-desktop:~/asir$ mkdir carpeta\ sin\ titulo/
alberto@alberto-desktop:~/asir$ ls
carpeta sin titulo fich1.txt
```

- **mkdir -p directorio1/directorio2** Crea el directorio especificado y el que lo contiene si no existe.

```
alberto@alberto-desktop:~/asir$ mkdir -p alumno/alumno1
alberto@alberto-desktop:~/asir$ ls -R
.:
alumno

./alumno:
alumno1

./alumno/alumno1:
```

Comando **cp** , Copia archivos donde le indiquemos:

- **cp file1 file2** Crea el archivo file2 que es una copia idéntica del archivo file1.
- **cp file1 /usr/nombre/** Copia el archivo file1 a la carpeta /usr/nombre.
- **cp -r uno/ Anix/.** Copia el directorio **uno** dentro del directorio **Anix** con el mismo nombre.
- **cp -r uno/ Anix/tres** Copia el directorio **uno** dentro del directorio **Anix** cambiándole nombre a tres.

Comando **mv** , comando usado para mover o renombrar archivos o directorios. El archivo original es borrado y el crea un nuevo archivo con el mismo contenido, el nombre puede ser diferente o puede ser el mismo. En caso de que se use en la misma ruta con el nombre del archivo y otro nombre solo cambiara el nombre del archivo.

- **mv file1 file_1** renombra un archivo: el archivo 'file1' pasa a llamarse 'file_1'
- **mv File /file** mueve 'File' a '/file'
- **mv file ./dir/file** mueve 'file' a 'dir/file' Relativo al directorio
- **mv archivo directorio** Mueve un archivo al directorio especificado.

```
alberto@alberto-desktop:~/asir$ mv apuntes.txt alumno/
alberto@alberto-desktop:~/asir$ ls -R
.:
alumno

./alumno:
alumno1 apuntes.txt
```

Comando **rm** , es una orden usada para eliminar archivos y directorios. Esta orden debe utilizarse con cautela, ya que puede ser muy destructiva, debido a que, al momento de ser llamada, por omisión borra los archivos sin pedir confirmación (con la opción **-i** pide confirmación).

- **rm fichero** Borra el archivo especificado.

```
alberto@alberto-desktop:~/asir/alumno$ ls
alumno1 apuntes.txt
alberto@alberto-desktop:~/asir/alumno$ rm apuntes.txt
alberto@alberto-desktop:~/asir/alumno$ ls
alumno1
```

- **rm -r directorio** Borra el directorio especificado y todo su contenido.

```
alberto@alberto-desktop:~/asir$ ls -R
.:
alumno

./alumno:
alumno2

./alumno/alumno2:
alumno3

./alumno/alumno2/alumno3:
alberto@alberto-desktop:~/asir$ rm -r alumno
alberto@alberto-desktop:~/asir$ ls -R
.:
alberto@alberto-desktop:~/asir$
```

- **rm -ri directorio** Borra el directorio especificado y todo su contenido, pero pide confirmación.

```
alberto@alberto-desktop:~/asir$ mkdir pepe
alberto@alberto-desktop:~/asir$ ls
f1 f2 ficheroordenado pepe
alberto@alberto-desktop:~/asir$ rm -ri pepe
rm: ¿borrar el directorio «pepe»? (s/n) s
alberto@alberto-desktop:~/asir$ ls
f1 f2 ficheroordenado
```

Comando **rmdir** , Borra el directorio especificado, pero el directorio tiene que estar vacío.

```
alberto@alberto-desktop:~/asir/alumno$ mkdir alumno1
alberto@alberto-desktop:~/asir/alumno$ ls
alumno1
alberto@alberto-desktop:~/asir/alumno$ rmdir alumno1
alberto@alberto-desktop:~/asir/alumno$ ls
alberto@alberto-desktop:~/asir/alumno$ █
```

Comando **touch** , Crea ficheros dentro de un directorio.

```
alberto@alberto-desktop:~/asir$ touch apuntes.txt
alberto@alberto-desktop:~/asir$ ls
alumno apuntes.txt
```

Editores de ficheros **gedit** , **nano** , **pico** , **vi** , edita el fichero indicado con el programa especificado.

```
alberto@alberto-desktop:~/asir$ gedit apuntes.txt
```

Otra forma de crear un archivo y introducir texto mediante terminal es con **cat** y **>**

cat > fichero Se introducen los datos línea a línea y se validan con **enter**, para salir con "Ctrl C".

Comando **cat** , Muestra el contenido del fichero especificado en el terminal.

```
alberto@alberto-desktop:~/asir$ cat fich1.txt
mar ene 31 13:00:02 CET 2012
```

Comando **less** , muestra el contenido de un fichero, similar a **cat**, hasta donde entre en la pantalla del terminal y lo que no entra se muestra línea a línea usando el cursor.

Comando **more** , muestra el contenido de un fichero y los saltos son en bloque.

Comando **cut** , corta secciones de línea basándose en campo, carácter o byte, las opciones son:

- **-c** indicamos que lo que se desea es cortar exclusivamente columnas que en principio estarán separadas por comas.
- **-f** indicamos que lo que se desea recortar son campos, por lo tanto, el fichero deberá estar formateado con respecto a informaciones que se estructuren en el formato de registro subdividido en campos.

a continuación del **-f** irán los números de columna que para cada fila del fichero se quisieran recortar. La forma de indicar columnas es igual que en la orden **sort**, a diferencia, que en este caso no es necesario precisar el tipo de columna.

-f 1 -5 Cortaría desde la 1 hasta la 5 inclusive

-f 1 5 solo la una y la 5

- **-d** indicamos el carácter separador de campo

- **-d :**

```
alberto@alberto-desktop:~/asir$ echo "Esto es un prueba, 123, probando" | cut -d "," -f1
Esto es un prueba
alberto@alberto-desktop:~/asir$ echo "Esto es un prueba, 123, probando" | cut -d "," -f2
123
alberto@alberto-desktop:~/asir$ echo "Esto es un prueba, 123, probando" | cut -d "," -f3
probando
alberto@alberto-desktop:~/asir$ echo "Esto es un prueba, 123, probando" | cut -c 1-4
Esto
alberto@alberto-desktop:~/asir$ █
```

```
alberto@alberto-desktop:~/asir$ cut -c 1-60 cut.txt
Linea 1 : este texto es despues de dos ptos en la linea 1
Linea 2 . este texto es despues de un pto en la linea 2
Linea 3 , este texto esta despues de una coma en la linea 3
```

```
alberto@alberto-desktop:~/asir$ cut -d ":" -f2 cut.txt
este texto es despues de dos ptos en la linea 1
Linea 2 . este texto es despues de un pto en la linea 2
Linea 3 , este texto esta despues de una coma en la linea 3
cut -d: -f1,2 /etc/passwd
```

Comando **tee** , Graba en un fichero el contenido leído de otro por ejemplo mediante el comando **cat**.

```
alberto@alberto-desktop:~/asir$ cat tee1.txt | tee tee2.txt
Este texto esta escrito en archivo tee1.txt
alberto@alberto-desktop:~/asir$ cat tee2.txt
Este texto esta escrito en archivo tee1.txt
```

Primero con **cat** lee el contenido del fichero **tee1.txt**, añadimos **|** para concatenar otra sentencia y con **tee** copiamos lo leído con **cat** en el fichero **tee1.txt** y lo metemos en el fichero **tee2.txt**, este archivo lo crea si no está y si está, reemplaza su contenido por lo que se copia.

Sintax **>** y **>>** , Similar a usar **tee**.

con **>** se copia el contenido a otro fichero machacando lo que tenga.

```
alberto@alberto-desktop:~/asir$ cat prueba2.txt
Este texto esta en el archivo prueba 2
alberto@alberto-desktop:~/asir$ cat prueba1.txt > prueba2.txt
alberto@alberto-desktop:~/asir$ cat prueba2.txt
Este texto esta en el archivo prueba 1
```

Con >> lo que hace es añadir el contenido sin machacar.

```
alberto@alberto-desktop:~/asir$ cat prueba2.txt
Este texto esta en el archivo prueba 2
alberto@alberto-desktop:~/asir$ cat prueba1.txt >> prueba2.txt
alberto@alberto-desktop:~/asir$ cat prueba2.txt
Este texto esta en el archivo prueba 2
Este texto esta en el archivo prueba 1
```

Comando **head -n x** fichero , Sirve para que liste las filas especificadas de principio a fin (el numero indicado en x define el numero de filas) del contenido de un fichero.

```
alberto@alberto-desktop:~/asir$ head -n 2 fich1.txt
Fila 1
Fila 2
```

Comando **tail -n x** fichero , Sirve para que liste las filas especificadas desde el final al principio (según el numero x) del contenido de un fichero.

```
alberto@alberto-desktop:~/asir$ tail -n 1 fich1.txt
Fila 3
```

tail En el caso de varios ficheros

```
alberto@alberto-desktop:~/asir$ head -n 3 fich1.txt fich2.txt
==> fich1.txt <==
Fila 1 del fichero 1
Fila 2 del fichero 1
Fila 3 del fichero 1

==> fich2.txt <==
Fila 1 del fichero 2
Fila 2 del fichero 2
Fila 3 del fichero 2
```

Comando **sort** y **sort -r** archivo , Ordena el contenido de un fichero ascendente o descendente (con -r).

```
alberto@alberto-desktop:~/asir$ sort fich1.txt
Fila 1 del fichero 1
Fila 2 del fichero 1
Fila 3 del fichero 1
Fila 4 del fichero 1
Fila 5 del fichero 1
Fila 6 del fichero 1
Fila 7 del fichero 1
alberto@alberto-desktop:~/asir$ sort -r fich1.txt
Fila 7 del fichero 1
Fila 6 del fichero 1
Fila 5 del fichero 1
Fila 4 del fichero 1
Fila 3 del fichero 1
Fila 2 del fichero 1
Fila 1 del fichero 1
```

Comando **wc** , Cuenta las lineas, palabras y caracteres, en este orden, contenidas en un fichero.

```
alberto@alberto-desktop:~/asir$ cat fich2
Pepe Romero
Jose Miranda
Alicia Paterna
alberto@alberto-desktop:~/asir$ wc fich2
 3  6 40 fich2
```

Y con las siguientes opciones:

wc -l , Cuenta las líneas de un fichero.

wc -c , Cuenta los caracteres, incluidos espacios y saltos de línea.

wc -w , Cuenta las palabras de un fichero.

```
alberto@alberto-desktop:~/asir$ wc -l fich2
3 fich2
alberto@alberto-desktop:~/asir$ wc -c fich2
40 fich2
alberto@alberto-desktop:~/asir$ wc -w fich2
6 fich2
```

Comando **grep** a [a-z]as fichero1 fichero2 Muestra las filas donde aparece el patrón indicado, con opciones como -c, -v, -i, ^, \$, estos meta-caracteres se usan con grep

^a Empieza por a

a\$ Termina por a

^[^a] No empieza por a

las opciones son:

grep -v 'cadena' fichero devuelve la línea donde no aparece la cadena

grep -i 'cadena' fichero devuelve la línea donde aparece la cadena, no diferencia mayúsculas de minúsculas

grep -c 'cadena fichero' devuelve el número de líneas donde aparece la cadena

```
alberto@alberto-desktop:~/asir$ gedit prueba2
alberto@alberto-desktop:~/asir$ cat prueba2
ahora estamos en implantacion de sistemas operativos
mañana no tenemos clase
alberto@alberto-desktop:~/asir$ grep -v '^a' prueba2
mañana no tenemos clase
alberto@alberto-desktop:~/asir$ grep '^a' prueba2
ahora estamos en implantacion de sistemas operativos
alberto@alberto-desktop:~/asir$ grep -c '^a' prueba2
1
```

```
alberto@alberto-desktop:~/asir$ grep ^Fi fich1.txt
Fila 1 del fichero 1
Fila 2 del fichero 1
Fila 3 del fichero 1
Fila 4 del fichero 1
Fila 5 del fichero 1
Fila 6 del fichero 1
Fila 7 del fichero 1
```

Comando **sed** , Muestra el texto especificado del fichero.

sed '1d' fichero muestra el texto del fichero pero borra la primera línea

```
alberto@alberto-desktop:~/asir$ cat prueba
Linea 1 arenas 250

Linea 2 vientos 350

Linea 3 olas 125
```

```
alberto@alberto-desktop:~/asir$ sed '1d' prueba

Linea 2 vientos 350

Linea 3 olas 125
```

sed '2,4d' fichero muestra el texto del fichero pero borra de la 2ª a la 4ª línea

```
alberto@alberto-desktop:~/asir$ sed '2,4d' prueba
Linea 1 arenas 250
Linea 3 olas 125
alberto@alberto-desktop:~/asir$
```

sed '/^\$/d' fichero Elimina las líneas en blanco que tenga el texto

```
alberto@alberto-desktop:~/asir$ sed '/^$/d' prueba
Linea 1 arenas 250
Linea 2 vientos 350
Linea 3 olas 125
```

sed '3,\$d' fichero muestra el texto del fichero pero borra de la 3 a la última línea.

```
alberto@alberto-desktop:~/asir$ sed '3,$d' prueba
Linea 1 arenas 250
alberto@alberto-desktop:~/asir$ █
```

sed -e '/^#/d' /etc/services muestra el texto del fichero pero borra las líneas que empiezan por #.

sed -n -e '/^[de]p' fichero muestra las palabras que empiecen por de

sed -n -e '/vivimos/,/tecnologia/p' fichero muestra desde línea que empieza por 'vivimos' hasta la que finaliza con 'tecnologia'

sed 's/cadena1/cadena2/' fichero Sustituye la cadena1 por la cadena2 la 1ª vez que la encuentra, en cada línea.

```
alberto@alberto-desktop:~/asir$ cat prueba
Linea Linea 1 arenas 250
Linea Linea 2 vientos 350
Linea Linea 3 olas 125
alberto@alberto-desktop:~/asir$ sed 's/Linea/Fila/' prueba
Fila Linea 1 arenas 250
Fila Linea 2 vientos 350
Fila Linea 3 olas 125
```

sed 's/cadena1/cadena2/g' fichero Sustituye la cadena1 por la cadena2 todas las veces que exista.

```
alberto@alberto-desktop:~/asir$ cat prueba
Linea Linea 1 arenas 250
Linea Linea 2 vientos 350
Linea Linea 3 olas 125
alberto@alberto-desktop:~/asir$ sed 's/Linea/Fila/g' prueba
Fila Fila 1 arenas 250
Fila Fila 2 vientos 350
Fila Fila 3 olas 125
```

sed '4s/cadena1/cadena2/g' fichero Sustituye la cadena1 por la cadena2 en la 4ª línea.

sed 's/\t;/g' ficheroA Sustituye los tabuladores (representado por `\t`) por ;

```
alberto@alberto-desktop:~/asir$ cat ficheroA
Primera linea tiene cosas escritas
Segunda Linea con mas cosas escritas
Tercera linea escrita
Cuarta linea con la palabra USUARIO
Quinta linea con la palabra USUARIO
alberto@alberto-desktop:~/asir$ sed 's/\t;/;' ficheroA
Primera linea;tiene cosas escritas
Segunda Linea;con mas cosas escritas
Tercera linea;escrita
Cuarta linea;con la palabra USUARIO
Quinta linea;con la palabra USUARIO
```

sed 's/a/A/g' fichero Sustituye las a por A

```
alberto@alberto-desktop:~/asir$ cat ficheroA
Primera linea tiene cosas escritas
Segunda Linea con mas cosas escritas
Tercera linea escrita
Cuarta linea con la palabra USUARIO
Quinta linea con la palabra USUARIO
alberto@alberto-desktop:~/asir$ sed 's/a/A/g' ficheroA
PrimerA lineA tiene cosAs escritAs
SegundA LineA con mAs cosAs escritAs
TercerA lineA escrita
CuArtA lineA con lA pAlAbra USUARIO
QuintA lineA con lA pAlAbra USUARIO
```

sed '/^d/s/a/A/g' fichero Sustituye a mayusculas las letras a de las lineas que empiezan por d.

```
alberto@alberto-desktop:~/asir$ cat ficheroA
Primera linea tiene cosas escritas
Segunda Linea con mas cosas escritas
Tercera linea escrita
Cuarta linea con la palabra USUARIO
Quinta linea con la palabra USUARIO
d linea con la palabra USUARIO
alberto@alberto-desktop:~/asir$ sed '/^d/s/a/A/g' ficheroA
Primera linea tiene cosas escritas
Segunda Linea con mas cosas escritas
Tercera linea escrita
Cuarta linea con la palabra USUARIO
Quinta linea con la palabra USUARIO
d lineA con lA pAlAbra USUARIO
```

Comando **find** , Permite buscar en un directorio ficheros que cumplen una serie de condiciones.

find [ruta desde donde se busca] [opciones de busqueda]

find /etc -name '*.conf' busca en /etc todos los ficheros con la extension conf

Comando **whatis** , muestra un resumen de lo que hace el comando que le pasas como parámetro.

whatis cat

Comando **ps** , muestra por pantalla un listado de los procesos que se están ejecutando en el sistema.

- **ps -a** Lista los procesos de todos los usuarios.
- **ps -u** Lista información del proceso como por ejemplo el usuario que lo está corriendo, la utilización de Cpu y memoria, etc.
- **ps -x** Lista procesos de todas las terminales y usuarios
- **ps -l** Muestra información que incluye el UID y el valor “[nice](#)”.

Comando **sudo -s** , Entrar como usuario con privilegios de administrador.

```
alberto@alberto-desktop:~/asir/alumno/alumno1$ sudo -s
[sudo] password for alberto:
root@alberto-desktop:~/asir/alumno/alumno1#
```

comando **sudo passwd root** , Entrar como usuario root (administrador)

Comando **umask** , es una mascara que establece los permisos que tenemos por defecto, el valor del umask por defecto en Ubuntu es de 022, de tal manera que la diferencia de los permisos máximos (777 para directorios y 666 para ficheros) con el valor del umask, da los permisos del fichero/directorio que se creen, así que 777-022 da 755 para directorios y 666-022 da 644 para ficheros.

Comando **chmod** , con este comando se cambian los permisos a ficheros y directorios, el permiso se puede indicar en numérico y mediante identificadores, rwx (lectura, escritura, ejecución), u= usuario, g= grupo, o= otros y a= todos.

chmod permiso fichero/directorio donde permiso se puede poner en numérico o con identificador.

chmod 644 fichero daríamos permiso de rw(4+2+0) al usuario, r al grupo(4+0+0) y r a todos(4+0+0)

chmod u+rw, g+r, o+r fichero seria lo mismo que lo anterior.

Comando **chown** , con este comando se cambia el usuario propietario de un fichero

chown usuario fichero asignamos el nuevo usuario propietario del fichero especificado

chown usuario:grupo fichero asignamos usuario y grupo al fichero en una sola instrucción

Comando **chgrp** , con este comando se cambia el grupo al que pertenece un archivo

chgrp grupo fichero asignamos el nuevo grupo al que pertenece el fichero especificado

Comando **ln** , hace un enlace fuerte/débil del fichero según la opción empleada:

ln 'fichero' 'nombre enlace directo' Crea enlace fuerte

solo referencia archivos y en la misma partición, referencia al inodo, por lo que se pueden crear copias del archivo sin que ocupe mas espacio, ya que solo se copia la referencia al inodo y hasta que no se eliminan todos los enlaces fuertes el archivo no se elimina.

ln -s 'fichero' 'nombre enlace directo' Crea hace un enlace directo (débil) del fichero

puede referenciar a archivos o carpetas en cualquier ruta, referencia la ruta del archivo

```
alberto@alberto-desktop:~/asir$ ln f1 enlacedurof1
alberto@alberto-desktop:~/asir$ ln -s f1 enlacedirectof1
alberto@alberto-desktop:~/asir$ ls -lia
total 28
411433 drwxr-xr-x  3 alberto alberto 4096 2012-02-09 11:43 .
327657 drwxr-xr-x 23 alberto alberto 4096 2012-02-09 11:30 ..
411430 lrwxrwxrwx  1 alberto alberto   2 2012-02-09 11:43 enlacedirectof1 -> f1
419017 -rw-r--r--  5 alberto alberto  15 2012-02-09 10:28 enlacedurof1
419017 -rw-r--r--  5 alberto alberto  15 2012-02-09 10:28 f1
419011 -rw-r--r--  1 alberto alberto  22 2012-02-09 10:28 f2
419035 -rw-r--r--  1 alberto alberto  37 2012-02-09 10:30 ficheroordenado
419071 drwxr-xr-x  2 alberto alberto 4096 2012-02-09 10:54 mio
```

cat /etc/passwd solo un extracto

```
alberto@alberto-desktop:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
```

cat /etc/group solo un extracto del contenido

```
alberto@alberto-desktop:~$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:alberto,prueba
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:alberto,prueba
fax:x:21:prueba
voice:x:22:
cdrom:x:24:alberto,prueba
```

Comando **adduser** "usuario" , Este comando crea un usuario.

Comando **userdel** "usuario" , Este comando borra el usuario

Comando **groupadd** "grupo" , Este comando crea un grupo.

Comando **groupdel** "grupo" , Este comando borra un grupo.

Pipe y Tubería es “|” (Alt Gr+1) y empalma la salida estándar del primer comando con la entrada estándar del segundo comando, ejemplo:

```
alberto@alberto-desktop:~/asir$ ps | wc -l
```

Se aplica el comando ps y después el comando wc

Si quiero añadir un texto determinado al final del fichero, por ejemplo 'buenos días':

echo 'buenos días' > fichA

Si existe el fichA machaca el contenido

Si no existe fichA crea el archivo y le mete el contenido 'buenos días en este caso 'buenos días'

****Permisos por defecto de los archivos de Linux (por defecto a 666 para ficheros y 777 para directorios).**

umask es para dar permisos por defecto los nuevos archivos que creamos (es decir rw- rw- rw- o 666)

r: lectura, w: escritura y x: ejecución

rwX rwX rwX

111 111 111 equivale a 777

421 421 421 Equivale a 3 bit en binario. así si tiene las 3 opciones (rwX) suma 7.

rw- r-- r--

110 100 100 equivale a 644

Meta caracteres

^ empieza por

\$ termina por

^[^] no empieza

Los Repositorios y apt-get

Repositorios: Es una base de datos donde están almacenados los paquetes, que una vez descargados instalan el software en el ordenador.

Ubuntu usa paquetes para almacenar todo lo que un programa necesita para ejecutarse.

Hay distintos tipos de paquetes, los mas conocidos son **.deb** (paquetes preparados para ser instalados en ubuntu) y los **.rpm** (típicos de Red Hat y Mandriva).

Ubuntu divide todo el software en cuatro grupos llamados componentes. Para mostrar diferencias en licencias y la prioridad con la que se atienden los problemas de los que informan los usuarios.

Por defecto se instalan los paquetes de componentes **main** y **restricted**.

- **Main:** Contiene solamente paquetes que cumplen los requisitos de la licencia de Ubuntu y para los que hay soporte disponible.
 - Pensado para que incluya todo lo NECESARIO para la mayoría de sistemas Linux.
 - **Restricted:** Contiene paquetes soportados por los desarrolladores de Ubuntu, pero no están disponibles bajo ningún tipo de licencia libre. Se incluyen los paquetes tales como controladores propios de alguna tarjeta gráfica....
 - **Universe:** Mantenido por la comunidad.
 - **Multiverse:** Software no libre.
- *Activar repositorios Universe y Multiverse:
- x GUI: Sistema → Administración → Orígenes de SW → Añadir
 - x CLI: /etc/apt/sources.list

Instalar aplicaciones de repositorios

- ◆ Método gráfico: Centro de SW de Ubuntu
 - Synaptic (Ubuntu)
 - Adept (Kubuntu)

 - Menú → Aplicac. → Añadir y quitar
 - Sistema → Administrar → Gestor de paquetes Synaptic
- ◆ Modo Texto:
 - aptitude → Hasta ahora si des-instalas, aptitude borra el programa junto con las dependencias.
 - Apt-get → La desinstalacion borra solo el paquete especificado.

Sintaxis:

- sudo apt-get install <<nombre del paquete>>
- sudo aptitude install <<nombre del paquete>>
- sudo aptitude removal Desinstala

- `sudo aptitude update` Actualiza lista de paquetes disponibles
- `sudo aptitude upgrade` Actualiza el sistema con actualizaciones pendientes.

➔ `.deb` → paquetes preparados para ser instalados en ubuntu.

- `sudo dpkg -i <<nombrepaquete.deb>>` instalar el paquete.
- `sudo dpkg -r <<nombrepaquete.deb>>` Desinstala el paquete.

Apunte:

Startup Manager → Aplicación para modificar los parámetros del Grub.

Fdisk - Gestor de Particiones

Fdisk es un gestor de particiones que se ejecuta desde un Terminal con línea de comandos, los pasos a seguir para particionar un Disco duro serían los siguientes:

1. Particionado (fdisk)

`sudo fdisk /dev/sda` Herramienta Particionado de disco duro bajo terminal(sda, sdb) y las opciones son:

- ◆ m Muestra menú
- ◆ a Marca la partición como activa
- ◆ d Borra la partición
- ◆ n Crea nueva partición
- ◆ p Muestra la tabla particiones
- ◆ t Cambia el tipo de partición
- ◆ l Lista los tipos de particiones conocidos
- ◆ w Escribe la tabla de particiones (Hace efectivos los cambios)
- ◆ n Muestra Submenu (nueva partición)
 - p Partición primaria (1-4)
 - e Partición extendida

2. Crear un Sistema de archivos "FileSystem" (mkfs)

- `mkfs -t ext3 /dev/sdb1` Indica que en el disco duro sdb partición 1 sea con formato ext3.
- `mkfs.ext3 /dev/sdb1` Es lo mismo que lo anterior

3. Crear un punto de montaje (mkdir)

- `mkdir /respaldo`

4. Modificar el archivo de configuración fstab (/etc/fstab)

- Se añade la línea: `/dev/hdb1 /respaldo ext3 defaults 2 1`

5. Montar el dispositivo

- `mount -t ext3 /dev/hdb1 /respaldo`

Comandos adicionales:

- ◆ `du` Muestra el espacio de disco usado por los ficheros y directorios de un directorio.
 - `du -a` Muestra valores para ficheros y directorios
 - `du -b` Tamaño en bytes
 - `du -k` Tamaño en Kilobytes
 - `du -s` Muestra la ocupación total
- ◆ `df` Muestra el espacio de disco usado y disponible de los sistemas de ficheros montados.
 - `df -a` Muestra todos los Filesystem.
 - `df -h` Muestra salida mas legible
 - `df -i` Información de inodo
 - `df -l` Solo muestra filesystem locales
 - `df -t` Muestra el tipo de sistema de ficheros

Importante

- Si añadido 1 nuevo disco al sistema ya instalado, debemos crear particiones y sistemas de ficheros, lo que implica los siguientes pasos:
 - **fdisk** Para crear particiones
 - para hacer efectivos los cambios sin tener que reiniciar: **sudo partprobe**
 - **mkfs** Para crear sistema de ficheros
 - **mkdir** Crear directorio para montar el dispositivo
 - **mount** Montar los sistemas de ficheros
- Al iniciar sistema se montan los filesystem listados en /etc/fstab
- Para eliminar particiones con fdisk
 - Desmontar la partición del sistema con **umount** (umount /dev/sdb1)
 - Acceder a la gestión del disco con **fdisk** (fdisk /dev/sdb)
 - Eliminar partición con la opción **-d**

sudo gedit /etc/fstab

Cada línea del fichero tiene la siguiente información:

File System	Mount point	Tipo	Opciones	Dump	Pass
/dev/hda9	/home	ext3	defaults	0	2

cat etc/mstab Contiene una lista de los filesystem que están montados en el sistema.

Copias de Seguridad en Ubuntu

Tipo de copias de seguridad:

- a) **Diaria:** Es un tipo de copia que se realiza diariamente y de forma automática, llevada a cabo por el sistema, previamente ha sido programada. Solo copia los archivos que se han modificado en el día que se realiza la copia.
- b) **Total o completa:** Se realiza una copia de seguridad de todos los archivos y directorios seleccionados en el sistema.
- ventajas:
 - Realiza una copia íntegra de los archivos.
 - Posibilidad de recuperar los archivos con una copia de seguridad.
 - Desventajas:
 - Si las copias de seguridad son de gran tamaño al final acabará usándose demasiado espacio de disco de almacenamiento.
 - Son más lentas de realizarse.
- c) **Diferencial:** La copia de seguridad diferencial, almacena todos los archivos que han sido modificados desde la última copia de seguridad total, (si se ha realizado en medio una copia incremental, la copia diferencial copiará todos los archivos que cambien a partir de esta) almacenando de manera progresiva los archivos que modificamos cada día. Para realizar una restauración de archivos, solo deberemos disponer de la copia total y la última copia diferencial (la cual almacenará los cambios producidos).
- No marca los archivos como copiados. Ejemplo: Si hicimos una copia completa el domingo, el lunes se copiarán aquellos ficheros modificados o creados desde el domingo, y el martes se copiarán los creados o modificados desde el domingo también. Las copias diferenciales se van haciendo más grandes ya que reflejan los cambios desde una marca fija en el tiempo.
- Ventajas:
 - Solo necesitaremos la última copia diferencial creada más la copia total.
 - Cuando la diferencial se realiza, borra las copias echas anteriormente ahorrando espacio.
 - Desventajas:
 - Son más lentas para realizar que las incrementales y ocupan más espacio que las incrementales.
 -
- d) **Incremental:** Hace una copia de seguridad de todos los archivos que han sido modificados desde la última copia de seguridad realizada. Para realizar una restauración de archivos debemos disponer de la copia total y la suma de las copias incrementales. Hace copia de seguridad de los archivos.
- Ejemplo: Solo copia los ficheros creados o modificados desde el último backup ya sea completo o incremental, reduciendo la cantidad de información a copiar, en el caso de tener una copia completa el domingo el lunes se copiarán las novedades respecto al domingo y el martes las novedades respecto a la copia del lunes, con la consiguiente reducción de tamaño de copia
- Ventajas:
 - Son las más rápidas
 - Son las que menos espacio ocupa.
 - Desventajas:
 - Necesidad de realizar copias de seguridad constantemente.
 - Para restaurar el sistema necesitaremos la suma de las incrementales más la total.

Existen dos maneras de hacer copias de seguridad, realizarlas manualmente o utilizando un programador de tareas a través de la utilidad copias de seguridad.

Resumen:

- Normal: Copia los archivos seleccionados y los marca como copiados.
- Copia: Copia todos los archivos seleccionados, pero no los marca como copiados.
- Diferencial: Copia solo los archivos que fueron modificados después de la última copia de seguridad, pero no los marca como incluidos en la copia de seguridad.
- Incremental: Copia solo los archivos que fueron modificados después de la última copia de seguridad.
- Diaria: Hace una copia de seguridad de los archivos que se han creado o modificado hoy.

Carpetas importantes de las que hacer copia de seguridad:

- /home (Carpeta personal de los usuarios)
- /etc (Archivos de configuración del sistema)
- /root (Carpeta personal del usuario root)
- /var/log (Carpeta de logs del sistema)
- /var/www (Web de Intranet)

Porque se debe comprimir la copia de seguridad:

- La copia se realiza más rápidamente.
- El tamaño de la copia es menor, al quedar los datos reducidos la cantidad de datos a ocupar en el soporte de almacenamiento es menor y esto unido a que los datos estén comprimidos en un solo archivo, hace que transmitir los datos al soporte sea menor que si no lo comprime.
- La compresión garantiza la integridad de los datos, el algoritmo de compresión introduce un código de redundancia cíclica (CRC) que se consulta a la hora de descomprimir el archivo, de tal manera que tenemos la seguridad de si están correctos o no.

Nomenclatura de los archivos de copia de seguridad:

Normalmente, el nombre del archivo suele incluir el tipo de copia, las carpetas que contiene y la fecha (en el caso de copias totales) o fechas (en el caso de copias diferenciales e incrementales) de los datos.

Ejemplos de nombre de copia:

- ◆ Nombre de archivo copia de seguridad total
 - CopiaTotal_etc-home_01feb12.tar.bz2
- ◆ Nombre de archivo copia de seguridad diferencial
 - CopiaDiferencial_etc-home_01feb12-08feb12.tar.bz2
- ◆ Nombre de archivo copia de seguridad incremental
 - CopiaIncremental_etc-home_08feb12-12feb12.tar.bz2

Para crear una copia de seguridad se utiliza el comando tar con sus respectivas extensiones:

- j Comprimir utilizando bzip2
- c Crea nuevo archivo
- v Mostrar los archivos añadidos/extraídos
- f Escribir/extraer hacia/desde un archivo
- x Descomprime
- t Mostrar el contenido.

Crear copia de seguridad

```
tar -jcvf Copia.tar.bz2 carpeta1 carpeta2 carpeta3
```

Extrae archivos de la copia de seguridad

```
tar -jxvf Copia.tar.bz2
```

Crea copia de seguridad más nueva desde la fecha que se pone

```
tar -jcvf Copia.tar.bz2 -N 1feb
```

```
tar -jcvf /tmp/copi.tar.bz2 /home /etc
```

Información:

http://www.ite.educacion.es/formacion/materiales/85/cd/REDES_LINUX/backup/Creacion_y_restauracion_de_copias_de_seguridad.html

Automatización

El proceso de creación de copias de seguridad debe ser un proceso automático que no requiera la intervención del usuario para realizarse ya que por un olvido o dejadez del mismo podría ocurrir que el día que necesitemos la copia de seguridad, ésta no se haya hecho.

Para lanzar la realización automática de copias utilizaremos **cron**. Cron es un servicio que nos permite lanzar comandos automáticamente los días y a las horas que deseemos. Cada usuario tiene su propio cron en el que puede configurar sus tareas programadas mediante el comando '**crontab -e**' o con alguna aplicación gráfica como **gnome-schedule**. En nuestro caso, como realizamos copia de seguridad de carpetas que solamente tiene acceso el usuario root, debemos programar la copia mediante el **cron de root**.

Aplicaciones – Herramientas de sistema- Tareas programadas

- a- Gráficamente:gnome-schedule
- apt-get install gnome-schedule

Nuevo repetida

Crear una tarea programada

comando: ./copias/copiatotal.sh

Hora y fecha

Minuto

Hora

Diagnostico

..

En comando seria: Crontab 55 23 1 * * ./copia/ (minuto, hora,dia,mes, dia semana)
video tutorial youtube llamado copia de seguridad con tar y crontab

Otros comandos para copias y backup:

Comando dd

Permite realizar copias exactas de discos duros, particiones o ficheros.

También permite hacer una ISO a partir de un CD/DVD.

El comando **dd** (duplicate disk) es un comando útil para transferir datos desde un dispositivo/archivo hacia un dispositivo/archivo/etc.

- La sintaxis básica del comando es la siguiente:

dd if=origen of=destino

donde **if** significa "*input file*", es decir, lo que quieres copiar y

of significa "*output file*", o sea, el archivo destino (donde se van a copiar los datos); **origen** y **destino** pueden ser dispositivos (lectora de CD, diskettera, etc.), archivos, etc.

Haciendo imágenes **ISO** de un CD:

La forma mas fácil y efectiva de crear nuestras "imagenes" de CD es la siguiente:

- **dd if=/dev/cdrom of=micd.iso**

Clonar discos:

El comando **dd** también sirve para copiar particiones o discos completos unos sobre otros. Básicamente podemos decir que mediante **dd** podemos "*clonar*" particiones o nuestro disco rígido completo. Para hacer esto hace lo siguiente:

- **dd if=/hdx of=/hdyb** (copia una partición en otra)
- **dd if=/hdx of=/hdy** (copia de un disco duro en otro)
- donde: **x**: disco rígido origen, **y**: disco rígido destino, **a**: partición origen, **b**: partición destino.

Tener presente que el tamaño de la imagen resultante va ser exactamente el mismo que el del dispositivo original. Es decir: **dd** te guarda también el espacio no utilizado.

dump

Herramienta que permite hacer copias de seguridad, en entornos Unix y copia sistemas de ficheros completos.

Según las opciones puede hacer copias incrementales, diferenciales, totales, etc.

Restore

Herramienta que permite restaurar las copias echas con dump.

Ejemplo:

Realiza una copia de seguridad de la carpeta /home y guardala en el fichero /root/copia.dump

dump 0 uf /root/copia.dump /home

restore 'opciones' /root/copia.dump