

COMANDOS BÁSICOS DE LINUX

Comandos para gestionar ficheros y
directorios

7.- Comandos para hacer búsquedas ficheros y patrones

- **Comando find**

Se utiliza este comando para buscar archivos dentro de una jerarquía de directorios. La búsqueda, como veremos más adelante, se puede realizar mediante varios criterios.

Sintaxis :

`find [ruta...] [expresión]`

La **expresión** se conforma de opciones, pruebas y acciones.

Algunas opciones son:

Opción	Resultado
<code>-name <expresión></code>	Permite especificar patrones para los nombres de los ficheros a buscar
<code>-iname <expresión></code>	Igual que anterior, pero ignora mayúsculas/minúsculas
<code>-type <tipo></code>	Permite indicar el tipo de fichero a buscar
<code>-size +/- <n></code>	Permiten indicar tamaño máximo y/o mínimo de los ficheros a buscar
<code>-perm[+ -] <modo></code>	Permisos de los ficheros a buscar en octal
<code>-exec <comando> \;</code>	Permite definir un comando a ejecutar para cada resultado de la búsqueda. La cadena {} se sustituye por el valor encontrado

7.- Comandos para hacer búsquedas ficheros y patrones

Ejemplos:

- **find ./PDF/ ./xarxa/ -iname "*.pdf"**

Todos los pdf que hay en los directorios PDF y xarxa.

- **find . -name "*.txt"**

Nos devolverá todos los archivos con extensión **txt** que haya en el directorio y recursivamente en todos sus subdirectorios.

- **find . -name "*.bak" -exec rm {} \;**

Borrara todos los ficheros acabados en .bak, esto lo hace a través de la opción -exec acompañada de la instrucción a ejecutar, los corchetes {} se refieren a los ficheros que la instrucción find ha encontrado y la barra invertida \; indica el final de la instrucción

- **find . !-iname "*.pdf" !-name "*.jpg"**

Que nos devuelve todos los archivos menos los acabados en .pdf o .jpg.

- **find . !-iname "*a*" -iname "*.jpg"**

Nos devolvería todos los jpg que no tengan en su nombre la letra "a".

- **find . -empty -type f** Nos buscara ficheros vacios.
- **find . -type d** buscar directorios
- **find . -type l** buscar enlaces simbólicos.

7.- Comandos para hacer búsquedas ficheros y patrones

- **find . -group admin** buscar ficheros del grupo **admin**
- **find . -user pepe** buscar ficheros del usuario **pepe**
- **find . -nogroup** ficheros que no pertenezcan a ningún grupo o ningún usuario.
- **find . -nouser**
- **find . -perm 777** ficheros que tengan todos los permisos concedidos para todos
- **find . -iname "*.html" -exec cat {} \;**

Sobre los resultados de **find** se pueden ejecutar ciertas acciones, esto se hace indicándolo con **-exec orden**. la cadena **{}** se reemplazará por el nombre del fichero que se esta procesando.

- **find . -iname "*.html" -exec cat {} \;**

7.- Comandos para hacer búsquedas ficheros y patrones

Todas las expresiones que aquí se han mostrado y muchas más que hay disponibles (consultar man find), se pueden combinar entre ellas. Se han de cumplir las dos expresiones (and implícito)

- **expr1 expr2**

Solo se ha de cumplir una de ellas

- **expr1 -or expr2**

Si no se cumple la expr1

- **! expr1**

si no se cumple la expr1 y se cumple expr2

- **! expr1 expr2**

Copia todas las imágenes jpg o png del directorio Imágenes al directorio dibuix.

- **find ./Imágenes/* -iname "*.jpg" -or -iname "*.png" -exec cp {} tmp/dibuix \;**

7.- Comandos para hacer búsquedas ficheros y patrones

- **Comando locate**

Comando que provee una fácil y rápida manera de buscar archivos en el sistema completo basado en patrones de nombres.

Sintaxis: `locate [opciones] archivos/directorios`

El comando es mucho más rápido (aunque menos flexible) que `find`, pues utiliza los datos en una base de datos temporal que almacena los nombres de todos los archivos en el sistema, por lo que también se pueden experimentar fallas al buscar un archivo creado recientemente. Para renovar la base de datos se utiliza el comando `updatedb`. la base de datos (índice) que usa `locate` es:

/var/lib/mlocate/mlocate.db

Opciones:

- c muestra por pantalla el número de resultados de la consulta
- r **expr_regular** búsqueda a partir de expresiones regulares.
- i permite indicar que los resultados hagan caso omiso a mayúsculas/minúsculas.

8.- Comandos para filtrar ficheros

- Comando **sort**

Ordena las líneas de un fichero mostrándolas por la salida estándar. De no especificarse un fichero toma la entrada estándar.

Sintaxis:

```
sort [opciones] [fichero]
```

Algunas opciones:

-r: ordena al revés.

-f: trata las mayúsculas y minúsculas por igual.

-n : ordenación numérica, ideal para columnas con cifras.

-r : ordenación en orden decreciente.

-t c : delimitador de campo c

-k n : campo o columna de ordenación

Ejemplo:

```
sort -f /etc/passwd
```

```
sort -r /etc/passwd
```

```
sort -r -n -k 3 -t: /etc/group
```

8.- Comandos para filtrar ficheros

- Comando **uniq**

Este comando elimina las líneas repetidas de un fichero ordenado, imprimiéndolo por la salida estándar o en otro fichero.

Sintaxis:

```
uniq [opciones] [fichero] [salida]
```

Opciones:

- c : utiliza como prefijo en cada línea su numero de ocurrencias.
- d : solo imprime las líneas duplicadas.

Ejemplo: `uniq -d lista.txt`

8.- Comandos para filtrar ficheros

- Comando **tail** y **head**

Los comandos tail y head muestran respectivamente el final y el comienzo (10 líneas por defecto) de uno o varios ficheros. De no especificarse al menos un fichero toman la entrada estándar.

Sintaxis:

```
tail    [opciones]  [ficheros]
head    [opciones]  [ficheros]
```

Algunas opciones:

-q : no coloca los encabezamientos con el nombre de los ficheros cuando se indican varios ficheros.

-<num> : imprime las n ultimas (tail) o primeras (head) líneas en lugar de las diez establecidas por defecto.

Ejemplos:

```
tail -f /var/log/messages
```

```
tail -20 /var/log/messages
```

```
head -50 /var/spool/mail/pepe
```

```
head -2 -q /etc/*.conf
```

8.- Comandos para filtrar ficheros

- Comando **wc**

Este comando imprime el numero de líneas, palabras y bytes de uno o varios ficheros. Si son varios ficheros hace un resumen de los totales. De no especificarse un fichero toma la entrada estándar.

Sintaxis:

```
wc [opciones] [ficheros]
```

Algunas opciones:

- l : solo cuenta líneas
- m : solo cuenta caracteres
- c: solo cuenta bytes.
- w : solo cuenta palabras.

Ejemplos:

```
wc -l /etc/passwd  
wc -w /etc/passwd
```

8.- Comandos para filtrar ficheros

- Comando **cut**

Este comando tiene la característica de poder cortar caracteres y campos, con la posibilidad de usar delimitadores y otras opciones, para finalmente, extraer las partes seleccionadas de cada fichero en la salida estándar.

Sintaxis:

```
cut  opciones  [ficheros]
```

Opciones:

-b , --bytes=LISTA : muestra solo estos bytes

-c , --characters=LISTA : muestra solo estos caracteres.

-d, --delimiter=DELIM : usa DELIM en vez de caracteres de tabulación para delimitar los campos.

-f, --fields=LISTA : selecciona solo estos campos; también muestra cualquier línea que no tenga un carácter delimitador, a menos que se especifique la opción **-s**.

Ejemplos:

```
echo "Esto es una prueba, 1 2 3 , probando" > fichero
```

```
cut -d"," -f3 fichero
```

```
cut -c2-4 fichero
```

```
cut -d"a" -f2 fichero
```

8.- Comandos para filtrar ficheros

- Comando **tr**

Comando utilizado para traducir caracteres. Permite cambiar unos caracteres por otros, eliminar caracteres, etc.

Sintaxis:

```
tr    [opciones]  cadena1  cadena2
```

Si no se utiliza ninguna opción, tr se limita a sustituir todos los caracteres de cadena1 por los caracteres de cadena2 que estén en el mismo lugar.

Este comando esta pensado para trabajar con tuberías (pipes) y no permite su uso sobre ficheros directamente.

Ejemplo:

```
alumno@alumno-VirtualBox:~/scripts$ echo canasta de 3 puntos | tr [3] [6]
canasta de 6 puntos
alumno@alumno-VirtualBox:~/scripts$ echo canasta de 3 puntos | tr [123] [678]
canasta de 8 puntos
alumno@alumno-VirtualBox:~/scripts$ echo canasta de 3 puntos | tr [a3] [j6]
cijnjstj de 6 puntos
```

8.- Comandos para filtrar ficheros

Opciones:

- d** : borra los caracteres definidos en la cadena1
- s** : elimina la secuencia continua de caracteres repetidos definidos en la cadena1.
- c** : Usa el complemento de la cadena1. Esto significa que define la cadena1 como todos los caracteres que no se encuentran en la definición dada por el usuario. Este parámetro es útil para indicar caracteres que no queremos que se vean afectados.

Ejemplos:

```
$ echo 'esto es un ejemplo!!!....Algo práctico es?????... | tr -s '!.?'
```

```
$ echo 'esssstte eeessss oooottttro eeeeejjjemplo praaactiiccco!!!! | tr -c -s " "
```

```
$ echo 'hola mundo ' | tr -d " "
```

```
$ echo 'hola mundo ' | tr -s " " ";"
```

9.- Comandos para compactar y agrupar ficheros

- Comandos **gzip** y **gunzip**

Ambos comandos permiten compactar y descompactar (comprimir y descomprimir) respectivamente uno o varios ficheros.

Sintaxis:

```
gzip          [opciones]      <fichero/directorio>
gunzip        [opciones]      <fichero/directorio>
```

Algunas opciones son:

-r : dado un directorio comprime todos los ficheros presentes en él recursivamente.

-1 a **-9** : especifica el grado de compresión [-1 menor y mas rápido, -9 mayor y mas lenta].

-S <sufijo> : permite especificar un sufijo o extensión para el fichero resultado (por defecto es gz)

9.- Comandos para compactar y agrupar ficheros

Ejemplos:

\$ gzip -9 fichero1 -> comprime fichero1 como fichero1.gz

\$ gunzip fichero1.gz -> descomprime fichero1.gz

\$ gzip -S .zip -r ./scripts / -> comprime todos los ficheros del directorio ./scripts con extensión .zip

\$ gunzip -S .zip -r ./scripts/ -> descomprime todos los ficheros del directorio con extensión .zip

\$ gzip fichero.gz fichero1 fichero2 fichero3 /user/work/scripts -> comprimirá fichero1 fichero2 fichero3 y el directorio /user/work/scripts y lo guardará en el fichero fichero.gz

9.- Comandos para compactar y agrupar ficheros

- Comando **tar**.

Es una herramienta que permite agrupar varios ficheros aislados o el contenido de un directorio en otro fichero o dispositivo especial. El comando tar no comprime o compacta absolutamente nada, se limita a agrupar varios ficheros en uno solo, sin comprimirlos.

Sintaxis:

```
tar [opciones] [archivo .tar] [Archivo/directorio a archivar]
```

Opciones:

- c** Crea el archivo
- x** Extrae el archivo
- f** Crea el archivo con el nombre de archivo dado
- t** Muestra o lista archivos en el archivo
- u** Archiva y añade a un archivo existente
- v** Muestra la información por pantalla. Verbose
- A** Concatena los archivos
- p** Mantiene los permisos originales de los archivos.
- z** Comprime el archivo tar con gzip

9.- Comandos para compactar y agrupar ficheros

Ejemplos:

\$ tar cvf archivo.tar /etc -> empaqueta todos los archivos del directorio /etc en el archivo.tar , mostrando por pantalla todos y cada uno de ellos.

\$ tar xvf archivo.tar -> extraerá archivo.tar en el directorio actual.

\$ tar czvf archivo.tar.gz /etc -> empaqueta y comprime los archivos del directorio /etc en un archivo llamado archivo.tar.gz

\$ tar xzvf archivo.tar.gz -> descomprime y extrae el contenido de archivo.tar.gz en el directorio actual.

\$ tar cxvf archivo.tar.gz file1 file2 file3 -> empaqueta y comprime los tres ficheros en un archivo llamado archivo.tar.gz

10.- Comandos para desconectarse del sistema.

- Comando **exit**

Este comando permite terminar el shell actual . Si se tiene un único shell es equivalente a desconectarse del sistema, pero si se esta en un subshell solo se terminara este, retornando al shell anterior.

- Comando **logout**.

Este comando permite desconectarse del sistema a partir de un login shell (primer shell que se ejecuta al establecer la conexión.)

La secuencia de caracteres Ctrl+D permite terminar el shell actual. Si es un login shell equivaldrá a hacer logout y si no, a hacer exit.

10.- Comandos para desconectarse del sistema.

- Comando **shutdown**.

Este comando provoca el cese de toda la actividad del sistema, cerrando todos los procesos y apagando el equipo. Para ejecutar esta orden se ejecuta como administrador del sistema.

Sintaxis:

```
shutdown [opciones] tiempo [mensaje de aviso]
```

argumento:

Tiempo; Tiene dos posibles formatos; puede ser una hora absoluta en formato **hh:mm**, donde **hh** representa la hora (uno o dos dígitos) y **mm** los minutos (con dos dígitos), también puede estar en formato **+m**, donde **m** es el numero de minutos que se debe esperar, la palabra **now** es un alias de **+0**.

Opción;

- t seg** : Hace que **init** espere **seg** segundos entre el envío del aviso y las señales de apagado, antes de cambiar a otro nivel de ejecución.
- k** : No apaga realmente, solo envía los mensajes de aviso a todos los usuarios.
- r** : Reinicia tras el apagado, correspondiente al nivel de ejecución 6.
- h** : Parada tras el apagado, correspondiente al nivel de ejecución 0.
- n** : No llama a **init** para iniciar el apagado, se ha de hacer manualmente.
- f** : Omite la comprobación del sistema de archivos (**fsck**) al reiniciar.
- c** : cancela un apagado en marcha, sin argumentos temporales.

10.- Comandos para desconectarse del sistema.

Ejemplos:

- Apaga el sistema transcurridos 10 minutos.

sudo shutdown -h +10

- Apaga el sistema de inmediato y después lo reinicia.

sudo shutdown -r now

- Apaga el sistema en 5 minutos y se envía un mensaje de alerta a los usuarios.

sudo shutdown -h +5 "El sistema se apagará en 5 minutos"

- Cancelar la orden shutdown

sudo shutdown -c