

1. Estructura de directorios en GNU/Linux:

La mayoría de los sistemas operativos GNU/Linux siguen el estándar FHS (Filesystem Hierarchy Standard) que define el contenido y las funciones de los directorios en la estructura jerárquica del árbol de directorios. Los directorios más importantes son:

- ✓ **/**: Directorio raíz o root. Todos los directorios y subdirectorios parten de este directorio raíz.
- ✓ **/bin**: contiene los archivos binarios (ejecutables) a nivel de usuario.
- ✓ **/boot**: almacena los archivos ejecutables y de configuración necesarios para el arranque del sistema.
- ✓ **/dev**: se encuentran los componentes del sistema y todos los dispositivos de almacenamiento representados por archivos (memorias FLASH, particiones, DVD, etc.). A través de este directorio podemos acceder a la información propia del medio de almacenamiento.
- ✓ **/etc**: almacena los archivos de configuración globales del sistema y que afectan a todos los usuarios.
- ✓ **/home**: directorio que aloja los directorios de los diferentes usuarios del sistema, a excepción del usuario root (el cual emplea /root)
- ✓ **/lib**: contiene archivos muy importantes para el sistema operativo, como librerías y módulos del kernel.
- ✓ **/media**: directorio que se emplea para montar dispositivos como discos duros o medios removibles como CD o DVD.
- ✓ **/mnt**: también utilizado para albergar puntos de montaje, pero, en este caso, temporales, como, por ejemplo, un sistema de archivo en red o carpetas compartidas en máquinas virtuales.
- ✓ **/proc**: directorio empleado por el sistema para guardar información relativa a los procesos y al kernel del sistema. No contiene archivos físicos, sino que se generan sobre la marcha archivos virtuales.
- ✓ **/sys**: al igual que /proc almacenan archivos virtuales relativos al kernel del sistema, así como información de drivers y dispositivos.
- ✓ **/sbin**: almacena ejecutables que se suelen emplear para tareas administrativas por el superusuario.
- ✓ **/tmp**: las aplicaciones emplean esta carpeta para almacenar archivos temporales
- ✓ **/usr**: almacena archivos de solo lectura de la mayoría de las aplicaciones y utilidades instaladas en el sistema.
- ✓ **/opt**: contiene el resto de aplicaciones no almacenadas en /usr. Normalmente, son aquellas que no son parte de los paquetes instalados con la distribución Linux objeto de uso.
- ✓ **/srv**: directorio encargado de alojar datos, scripts y carpetas para servidores instalados en nuestro sistema (servidores web, ftp, repositorios, etc.).
- ✓ **/var**: directorio considerado como registro del sistema. En él se incluye información del sistema, logs, información de caché, etc.

2. Gestión de archivos por línea de comandos en Linux:

ls [opciones] [ficheros]

Permite listar el contenido de un directorio e información de archivos. Sus opciones más utilizadas son las siguientes:

- ✓ **l**: muestra en formato largo
- ✓ **t**: ordena por fecha de modificación
- ✓ **r**: invierte el orden de salida
- ✓ **R**: lista recursivamente el contenido de cada directorio
- ✓ **i**: muestra el número de i-nodo
- ✓ **a**: muestra los archivos ocultos. En Linux los archivos ocultos son aquellos que empiezan con ".". Si no indicamos esta opción, el comando ls no listará los archivos ocultos.
- ✓ **h**: muestra el tamaño de cada fichero en K, M, G, etc.
- ✓ **size**: muestra el tamaño de cada fichero en bloques

✓ S: lista los archivos ordenados por tamaño

3. Eliminación de ficheros:

rm [-irf] lista_de_Ficheros

Se pueden eliminar archivos mediante la orden rm.
Sus opciones más utilizadas son las siguientes:

- ✓ i: solicita confirmación antes de realizar la acción
- ✓ r o R: eliminación recursiva sobre directorios
- ✓ f: fuerza la eliminación aun estando protegido el archivo contra escritura

4. Copia de archivos:

cp [-irR] lista_archivos_origen destino

- ✓ i: solicita confirmación antes de sobrescribir un archivo con el mismo nombre en el destino
- ✓ r o R: copia de forma recursiva en directorios.

5. Renombrado o movimiento de archivos:

mv [-iu] lista_archivos_origen destino

Sus opciones más utilizadas son:

- ✓ i: se emplea para asegurarse que en el destino no exista un archivo con el mismo nombre, ya que se sobrescribiría.
- ✓ u: solo mueve archivos o directorios con el mismo nombre entre origen y destino si estos son más actuales en el origen.

6. Impresión de archivos:

cat [lista_archivos] [{>|>>|<|<<} archivo]:

Concatena la lista de archivos, mostrándolos por pantalla. Su salida se puede redireccionar a un archivo mediante:

- > sobrescribe si existe el archivo o, en caso contrario, lo crea.
- >> añade al contenido de un fichero existente o, en caso contrario, lo crea.

Por tanto, se emplea la orden cat para crear archivos de escaso contenido de forma rápida.

Para ello, se emplea el comando cat, seguido de ">" o ">>" y de un archivo. De esta manera, el intérprete de comandos espera que introduzcamos texto hasta que pulsemos la combinación de teclas Ctrl+D. En la siguiente imagen, podemos ver cómo hemos creado 2 archivos de texto para, más tarde, fusionarlos en uno solo.

Otros comandos muy utilizados con:

more [lista_archivos]

less [lista_archivos]

Permiten visualizar el contenido de ficheros por páginas. A diferencia de cat, se puede navegar entre páginas. El

comando less es más versátil y potente que more, resultando ideal para ficheros de gran tamaño.

Existen otras órdenes que ayudan a obtener el principio y el final de archivos:

head [-n] lista_ficheros

tail [-n] lista_ficheros

Se emplean para visualizar las n primeras líneas (head) o últimas (tail) de uno o varios ficheros de texto. Por defecto, mostrará 10 líneas.

7. Cuenteo de un fichero:

La orden wc permite mostrar, por defecto, el número de líneas, palabras y número de bytes de un fichero. Su sintaxis es la siguiente:

wc [-lwcL] lista_ficheros

Podemos seleccionar algunos contadores de manera aislada o conjunta:

l: número de líneas

w: número de palabras

c: número de bytes

L: longitud de la línea más larga

8. Ordenación de un fichero:

La orden sort permite mostrar, de forma ordenada, las líneas de un fichero. Por defecto, ordena siguiendo el orden establecido por los caracteres en ASCII, donde las letras minúsculas tienen un orden mayor que las mayúsculas.

sort [-fnru] [-t <delimitador>] [-k <num_campo>] lista_archivos

Las opciones más importantes son:

- f: ignora las mayúsculas y las minúsculas
- r: invierte el orden
- n: ordena numéricamente en lugar de alfabéticamente
- u: elimina entradas repetidas
- t delimitador: indica un delimitador o separador de campos
- k: indica el número de campo por el que se va a realizar la ordenación. Por defecto, toma los espacios, tabuladores y carácter de fin de línea como delimitadores de campo.

9. Procesamiento de textos:

El comando cut se emplea para obtener información a partir de la división de un fichero o cadena de caracteres en columnas. Estas columnas se pueden establecer por caracteres o por campos delimitados por un delimitador de campo.

Su sintaxis es la siguiente:

cut -c <lista_caracteres> | -f <lista_columnas> [<-d delimitador>] fichero_texto

Donde:

- ✓ -c <lista_caracteres>: corta por caracteres especificados por lista_caracteres.
- ✓ -f <lista_columnas> | <-d delimitador>]: corta por campos establecidos por lista_columnas. Por defecto, los delimitadores son: espacio, tabuladores o espacio fin de línea, a menos que se especifique otro mediante la opción -d. No se pueden ejecutar conjuntamente las opciones -c y -f.

Por otro lado, el comando grep localiza un patrón en uno o varios ficheros, mostrando las líneas donde se encuentra. Su sintaxis es la siguiente:

grep [-nvlicw] patrón fichero_texto [fichero_texto]

Las opciones más empleadas son:

- ✓ l: solo muestra los ficheros que contienen el patrón especificado
- ✓ i: elimina la distinción entre mayúsculas y minúsculas
- ✓ c: muestra el número de líneas totales que cumplen con el patrón para cada fichero
- ✓ w: localiza el patrón como palabra y no como parte de una cadena de texto
- ✓ n: imprime el número de línea del patrón localizado
- ✓ v: busca líneas que no contengan el patrón especificado

10. Montaje y desmontaje:

La orden que realiza el montaje de un sistema de archivos es mount. Gracias a ella, se monta cualquier sistema de archivos reconocible por el núcleo de Linux en un punto de montaje del sistema. Todos los sistemas de archivos se montan directamente por nosotros o indirectamente durante el arranque del sistema, a excepción del sistema de archivos raíz "/", que se asocia a un punto de montaje compilado en el propio kernel y que monta la partición especificada durante la instalación. La sintaxis de la orden mount es la siguiente:

mount [-avwrt] [tipo] [dispositivo] [punto_de_montaje]

Las opciones más empleadas son:

- -a : monta los sistemas de archivos presentes en /etc/fstab, salvo que se indique el parámetro no auto, que impediría el montaje por esta opción.
- -v: muestra información del proceso de montaje
- -w: monta el sistema de archivos con permisos de lectura y escritura
- -r: monta el sistema de archivos con permisos de solo lectura
- -t <tipo>: indica el tipo de sistema de archivos para montar. Este puede ser, entre otros: ext, ext2, ext3, ext4, ntfs, nfs, iso9660, msdos, vfat, hfs, hfsplus o smbfs.

Para dejar de usar completamente o extraer un dispositivo con sistemas de archivos, este se debe desmontar. Para el desmontaje se emplea el comando umount, el cual puede completarse si no existen directorios en uso del sistema de archivos objeto a desmontar o procesos lanzados desde él. Para ejecutar el desmontaje se puede indicar la partición o el punto de montaje.

umount <dispositivo> o umount <punto_de_montaje>

11. Formatear:

Ya sabemos que muchas herramientas de particionamiento, como GNOME-disks, GParted y parted, permiten formatear las particiones creadas. Pero existen otras en modo texto que se encargan únicamente de realizar esa función.

El comando mkfs presenta la siguiente sintaxis:

mkfs [-t tipo_sistema_archivos] [opciones] dispositivo

Donde dispositivo es normalmente una partición, y las opciones más comunes son:

- **t tipo_sistema_archivos**: tipo de sistema de archivos que se desea implantar con ext2, ext3, ext4, vfat (msdos) o ntfs, entre otros.
- **opciones**: se pueden especificar muchas otras opciones, como etiquetas, formato rápido, tamaño del clúster, etc. Estas opciones varían según el constructor, al que llama mkfs. Por tanto, para mayor detalle, hemos de familiarizarnos con la ayuda (man) de estos comandos.

12. Desfragmentación:

La fragmentación de un sistema de archivos se refiere a la dispersión de datos relacionados entre sí en un medio de almacenamiento, lo que puede afectar el rendimiento de los discos duros mecánicos. La desfragmentación es el proceso de unir los bloques de datos de un mismo archivo. Los sistemas de archivos NTFS y FAT a menudo necesitan ser desfragmentados para mejorar su rendimiento, mientras que los sistemas de archivos Linux rara vez necesitan ser desfragmentados debido a sus sistemas inteligentes de asignación de bloques a archivos. Sin embargo, en dispositivos con poco espacio en disco y mucha actividad de creación y eliminación de archivos, como el sistema de archivos ext4, puede ser necesario utilizar la herramienta e4defrag para desfragmentar.

Su sintaxis es:

e4defrag [-cv] objetivo

Donde objetivo puede ser un archivo, un directorio o un dispositivo con sistema de archivos ext4 que emplea extents. Las opciones más comunes son:

- ✓ **c**: muestra un recuento de la fragmentación actual con respecto a la fragmentación ideal. Si se combina con la opción **v**, lo muestra para cada archivo. Con esta opción no se hace efectiva la fragmentación.
- ✓ **v**: muestra el recuento para cada archivo antes y después de la desfragmentación.

13. Chequeo:

En cuanto a utilidades propias de Linux, este dispone principalmente de los comandos fsck y e2fsck. Al igual que ocurre con mkfs, fsck es un front-end de otros programas que chequean sistemas de archivos, como e2fsck que permite chequear sistemas de archivos de la familia ext. Por tanto, este último es el apropiado, y el que se ejecuta para chequear sistemas ext2, ext3 o ext4. Podemos ejecutar `ls -l /sbin/*fsck*` para listar los ejecutables relacionados con el chequeo.

La sintaxis es:

fsck [-A] [-V] [-t tipo_sistema_Archivos] [-a] [-r] [sistema_de_archivos]

Donde las opciones más comunes son:

- **A**: Chequea todos los sistemas de archivos establecidos en /etc/fstab. Esta opción se lanza durante el arranque de Linux. Con esta opción no se puede emplear el argumento sistema_de_archivos.
- **V (verbose)**: detalla las acciones realizadas por fsck.
- **t tipo_sistema_archivos**: tipo de sistema de archivos que se desea chequear como ext2, ext3, ext4, fat (msdos) o ntfs, entre otros.
- **a**: repara sin pedir confirmación.
- **r**: pide confirmación antes de reparar los daños.

El comando e2fsck presenta la siguiente sintaxis:

e2fsck [-pcnyvD] sistema_archivos

Donde las opciones más comunes son:

- ✓ p: repara el sistema de archivos automáticamente. No es compatible con las opciones -n o -y.
- ✓ c: hace uso del programa badblocks, que busca bloques defectuosos. En caso de localizar alguno, lo añade a una lista de bloques defectuosos y evita así su uso.
- ✓ y: responde afirmativamente a todas las respuestas que e2fsck pudiera plantear
- ✓ n: responde negativamente a todas las respuestas que e2fsck pudiera plantear
- ✓ v: detalla las acciones realizadas por e2fsck
- ✓ D: optimiza los directorios para un mejor acceso a los datos.

14. RAID:

14.1. Creación de RAID:

mdadm -- create /dev/mdX --level=Y -- raid-devices=Z dispositivos

Donde:

- ✓ create /dev/mdX indica la creación del multidispositivo, siendo X un número
- ✓ level=Y es el nivel RAID para aplicar, pudiendo ser Y:
 - linear para RAID lineal
 - raid0, 0 o stripe para RAID0
 - mirror, raid1 o 1 para RAID1
 - raid4 o 4 para RAID4
 - raid5 o 5 para RAID5
 - raid6 o 6 para RAID6
 - raid10 o 10 para RAID10

✓ raid-devices=Z dispositivos, donde Z indica el número de dispositivos asociados al RAID y cada uno de ellos separado por espacios (/dev/sdX/dev/sdY ...)

14.2. Establecer un disco como defectuoso de un RAID

mdadm /dev/mdX --fail /dev/sdY

14.3. Eliminar un disco de un RAID

mdadm /dev/mdX --remove /dev/sdY

14.4. Añadir un disco a un RAID

mdadm /dev/mdX --add /dev/sdY

14.5. Comprobar el estado de todos los multidispositivos

cat /proc/mdstat

14.6. Obtener información de todos los multidispositivos

mdadm --detail --scan

14.7. Obtener información de un multidispositivo

mdadm --detail /dev/mdX y mdadm --detail /dev/mdX --scan

14.8. Examinar el estado de un dispositivo asociado a un RAID:

mdadm --examine /dev/sdX

14.9. Detener un RAID

mdadm --stop /dev/mdX

14.10. Eliminar el superbloque de un dispositivo sobrescribiendo ceros

mdadm --sop /dev/mdX