Sistemas operativos.
Gestión de archivos y almacenamiento.

Índice

1. Introducción

1. Introducción.

Los dispositivos de almacenamiento se dividen en particiones, dentro de las particiones en grupos de sectores o cilindros físicamente contiguos.

Una de las particiones primarias puede ser designada como una partición extendida, la cual puede subdividirse en particiones lógicas.

Para un disco duro o disco ssd, la unidad mínima de información serán los sectores.

Los esquemas de particionado más extendidos son el MBR (Master Boot Record) y el GPT (GUID Partition Table)

1. Introducción.

¿Por qué particionar?

Separación: es deseable aislar los datos de las aplicaciones de los archivos del sistema operativo.

Compartición: puede que múltiples sistemas operativos utilicen los mismos sistemas de ficheros.

Seguridad: se desean imponer cuotas o permisos distintos en cada partición.

Tamaño: Alguna información se mantiene constante y otras veces puede ser variable o volátil.

Si una partición se llena no afectará a las demás.

1. Introducción.

Instalar Active Disk Editor

https://www.disk-editor.org/index.html

1. Introducción.

MBR (Master Boot Record)

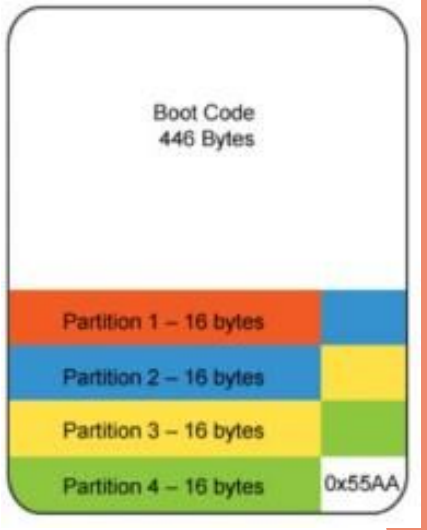
Está ubicado en el primer sector de un dispositivo y contiene tabla de Particiones.

Un disco puede tener hasta cuatro particiones primarias.

La zona de particionado tiene 64 bytes de longitud y se sitúa después de los 446 bytes del registro de arranque y un número mágico indica el final (0x55AA).

Solo una partición se marca como activa.

Cada entrada en la tabla de particiones ocupa 16 bytes y describe las cuatro posibles particiones primarias



1. Introducción.

GPT (GUID Partition Table)

GPT viene a sustituir MBR para mantener las nuevas BIOS (UEFI). Al igual que en el MBR empieza en el sector 0 del dispositivo para mantener compatibilidad con los sistemas con BIOS.

GPT tiene una copia al final del disco

MBR sólo puede tener un máximo de 4 particiones, sin embargo, GPT tiene hasta 128

MBR permite particiones de hasta 2.2TB, GPT hasta 9.4ZB

GPT permite tener 36 caracteres de nombre Unicode para cada partición.

GUIDs son almacenados como valores de 128 bits, y son mostrados en 32 dígitos hexadecimales.

Se mantiene el MBR en el sector 0 pero conteniendo una única partición con identificador de partición OxEE. Se le llama Protective MBR, con el área de código a cero.

1. Introducción.

https://www.cloudcenterandalucia.es/blog/hablemos-de-uefi-gpt-y-esp/#ESP

1. Introducción.

Como ya vimos, la gestión de los archivos es uno de los pilares fundamentales de cualquier sistema operativo.

Los sistemas de archivos proveen la manera de almacenar la información, así como mecanismos que permitan realizar operaciones sobre ella.

Existen multitud de sistemas de archivos que confieren diferentes características al espacio de almacenamiento y repercuten en la seguridad de los datos, su rendimiento o su gestión.

Los sistemas operativos proveen herramientas para la gestión del almacenamiento, para realizar particiones de los medios de almacenamiento, formatear, montar y desmontar los sistemas de archivos, desfragmentar, chequear los sistemas de archivos, buscar información e incluso crear diferentes esquemas RAID.

2.
Sistema de archivos

2. Sistema de archivos.

Los sistemas de archivos emplean el archivo (fichero) como la herramienta fundamental de abstracción lógica de la información.

Un archivo es, por tanto, la unidad lógica mínima de almacenamiento que contiene información. Se evita que el usuario conozca la estructura interna y las propiedades características de los medios de almacenamiento, facilitando la gestión y organización por su parte.

Otro elemento empleado por los sistemas de archivos son los directorios (carpetas). Estos son ficheros que actúan de contenedores lógicos de ficheros o directorios.

El directorio almacena información relativa a la localización física de la información y los atributos propios de cada archivo o directorio que contenga.

2. Sistema de archivos.

Los sistemas de archivos tienen como objetivo:

- Acceder a la información de los ficheros.
- Crear, eliminar y modificar ficheros.
- Acceder a los ficheros mediante diferentes protocolos de comunicación en red u otros
- Facilitar el acceso multiusuario.
- Facilitar el acceso a multitud de medios de almacenamiento.
- Realizar copias de seguridad.
- Utilizar herramientas de recuperación de información.
- Priorizar la eficiencia y la seguridad de acceso a la información.
- Maximizar el rendimiento en las operaciones sobre los archivos.
- Permitir la monitorización y contabilidad sobre ficheros.
- Administrar el espacio de almacenamiento, gestionar la asignación del espacio libre y el espacio ocupado de los archivos.

2. Sistema de archivos.

Para administrar el espacio libre y el espacio ocupado, se han de definir espacios de asignación.

Los sistemas de archivos definen el tamaño del espacio de asignación (también llamado unidad de asignación o clúster) durante la instalación del propio sistema de archivos (formateo). Este espacio determina el tamaño mínimo que ocupará un archivo en el medio de almacenamiento.

A nivel físico (hardware), el dispositivo administra sectores, sin embargo, el sistema de archivos gestiona clústeres.

La planificación de este espacio es muy importante, siendo ideal un equilibrio entre el promedio del tamaño de los archivos que vaya a alojar el sistema de archivos (evitando así que se desperdicie espacio interno al clúster, también conocido como fragmentación interna) y el tamaño del volumen (facilitando la administración del sistema de archivos).

Al formatear una unidad se puede indicar el tamaño de unidad de asignación.

2. Sistema de archivos.

FAT (File Allocation Table)

Es un sistema de archivos creado para el sistema operativo MS-DOS. La administración del espacio del almacenamiento es sencilla, por lo que se convierte en un sistema de archivos muy extendido en la mayoría de los sistemas operativos.

Las principales limitaciones de FAT32 son:

- Imposibilidad de gestionar particiones superiores a 8 TB (32 GB en Microsoft Windows) y archivos de más de 4 GB.
- Bajo rendimiento.
- Inseguro: no permite encriptación, sus atributos y permisos son limitados y no permite journaling.

2. Sistema de archivos.

El Journaling es un registro o diario del sistema de archivos. Los sistemas de archivos que hacen uso de este sistema se denominan transaccionales. Consiste en registrar una serie de acciones previas a la operación sobre el sistema de archivos que se vaya a realizar.

Se evitan muchas situaciones de inconsistencia de ficheros, facilitando la recuperación de datos y chequeos de medios de almacenamiento.

2. Sistema de archivos.

FAT (File Allocation Table)

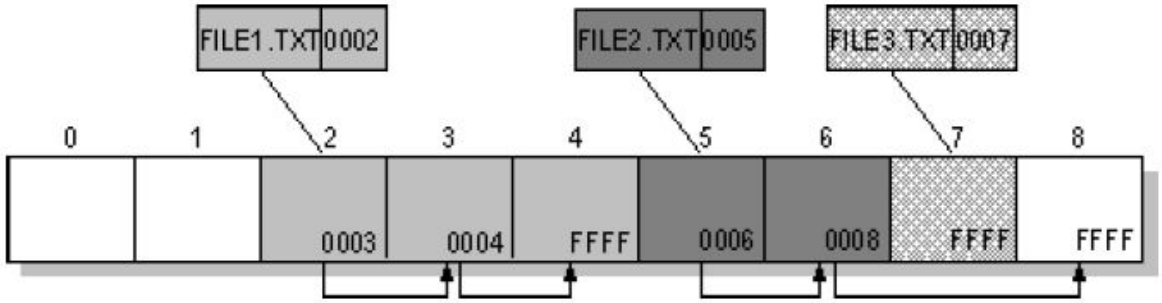
El directorio raíz, es un tipo especial de archivo que almacena las subcarpetas y archivos que conforman el sistema de archivos. Al ser una tabla, dispone una de una entrada para cada fichero o carpeta.

Contiene el nombre, fecha creación, tamaño…

2. Sistema de archivos.

FAT (File Allocation Table)

Las tablas FAT1 y FAT2 son tablas de punteros. Cada posición representa un sector que contiene un archivo y el contenido de la entrada es el valor del siguiente cluster del fichero. El primer cluster aparece indicado en el directorio raid y desde ahí se puede seguir el conjunto de clusters del fichero.



2. Sistema de archivos.

exFAT

Resultado de una evolución del sistema de archivos FAT32, que elimina sus principales limitaciones. Se pueden tratar archivos de hasta 16 EB y mantiene la ligereza frente a sistemas de archivos más avanzados, como NTFS y APFS. Aunque sigue resultando inseguro, es ideal para medios de almacenamiento FLASH portables con gran capacidad y compatibles entre sistemas operativos.

2. Sistema de archivos.

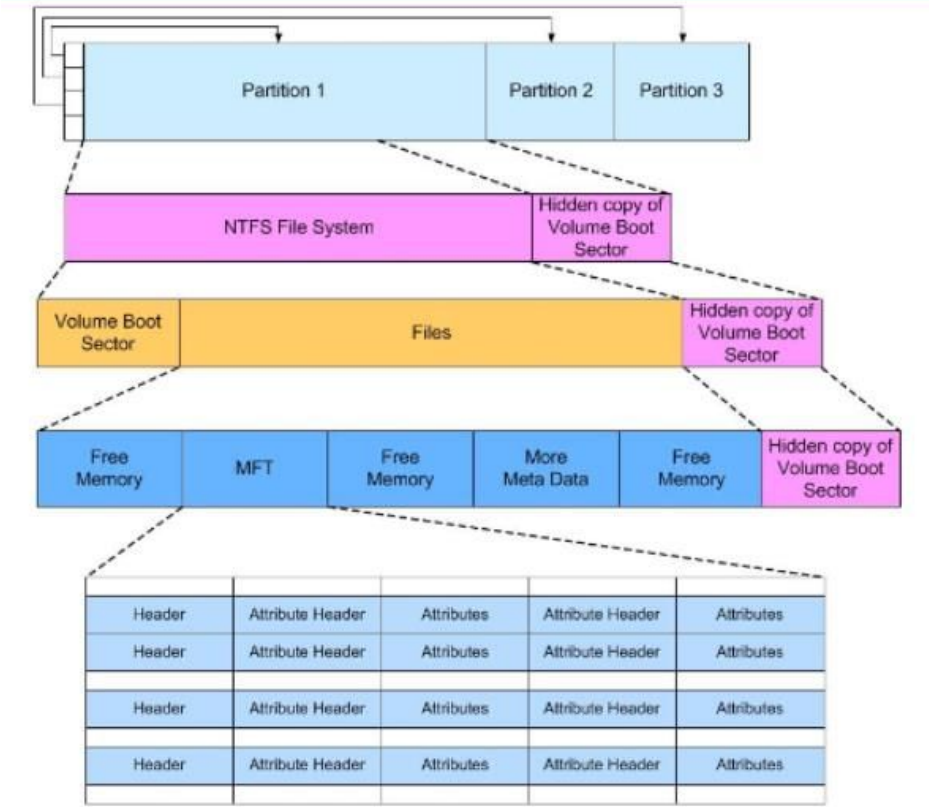
NTFS

Se considera el sistema de archivos estándar de Microsoft Windows. Sus mejoras son considerables con respecto a FAT32, primando la seguridad y la confiabilidad. Sus principales ventajas son:

- Emplea journaling. Favoreciendo una pronta recuperación ante errores inesperados.
- Permite cifrado y compresión.
- Reduce significativamente la fragmentación y aumenta la velocidad de búsqueda de archivos con respecto a FAT32.
- Volume Shadow Copy: mantiene un histórico de los ficheros y directorios.
- Alternate Data Stream: metadatos de ficheros
- Puede llegar a gestionar volúmenes de hasta 16 EB y archivos de hasta 16 TB.
- Emplea Unicode para el nombre de archivos, con hasta 255 caracteres.

2. Sistema de archivos.

NTFS



2. Sistema de archivos.

ReFS (Resilient File System)

El sistema de archivos ReFS (Sistema de archivos resiliente), diseñado para optimizar la disponibilidad de los datos, administrar de manera eficiente la escalabilidad para grandes cantidades de datos y garantizar la integridad de los datos mediante la llamada “resiliencia” a la corrupción de archivos.

ReFS fue diseñado para hacer frente a los nuevos escenarios de crecimiento de datos y como base para futuras innovaciones. ReFS se introdujo con Windows Server 2012, y luego también para Windows 8 y las últimas versiones de Windows 10.

2. Sistema de archivos.

APFS

Sistema de archivos empleado por Apple Inc. para sus medios de almacenamiento, que supone una versión mejorada de su predecesora HFS+. Sus características son similares a NTFS y ext4, por lo que permite administrar archivos y volúmenes de hasta 8 EB. Permite encriptación y está optimizado para almacenamiento Flash.

2. Sistema de archivos.

ext4 (Fourth extended file system)

Sistema de archivos predeterminado para sistemas operativos de tipo Linux en su cuarta versión. Incluye journaling, maneja archivos de hasta 16 TB y volúmenes de hasta 1 EB. Supera a sus antecesores ext2 y ext3, ya que:

- Mejora el rendimiento.
- Reduce la fragmentación.
- Permite trabajar con ficheros de mayor tamaño gracias al uso de extents.

2. Sistema de archivos.

ext4 (Fourth extended file system)

A diferencia de NTFS, ext4 no emplea extensiones como parte del nombre de los archivos. En Microsoft Windows, un nombre de archivo se divide en <nombre>.<extensión>, donde extensión es un conjunto de caracteres (normalmente tres o cuatro) que se asocian con programas para que el sistema operativo reconozca la manera de ejecutar el archivo.

No obstante, muchos nombres de archivos en Linux incorporan sufijos separados por un punto en el nombre del fichero por convención, pero no como requisito establecido por el sistema de archivos.

2. Sistema de archivos.

ext4 (Fourth extended file system)

Una partición con sistema de archivos ext4 se divide en grupos de bloques. Cada grupo de bloques se divide, a su vez, en las siguientes partes:

a) Superbloque: contiene la información más relevante del grupo de bloques.

b) Descriptores de grupos: almacena la información más importante del resto de bloques.

c) Bitmap de bloques de datos: contiene un mapa de bits donde se representa cada clúster, así como su estado (libre u ocupado).

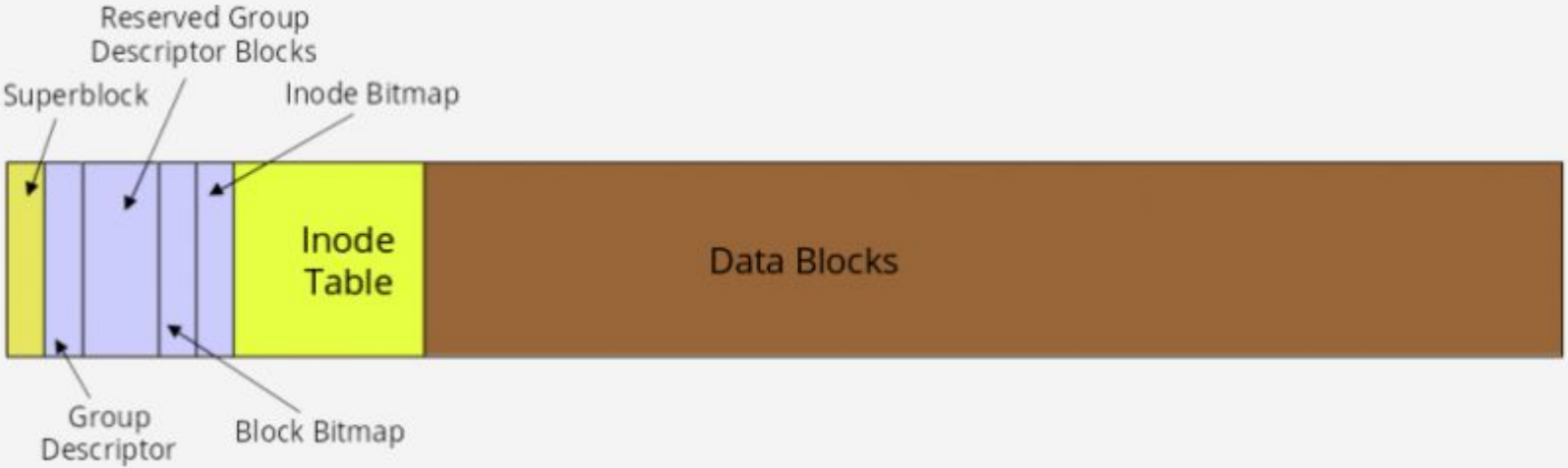
d) Bitmap de i-nodos: mapa de bits representando a cada i-nodo, que además indica su estado (libre u ocupado).

e) Tabla de i-nodos: tabla que contiene una entrada por cada i-nodo. El i-nodo almacena la información propia de cada archivo.

f) Bloques de datos: clústeres con información. Cada bloque de datos está asociado con un archivo.

2. Sistema de archivos.

ext4 (Fourth extended file system) A4 81



2. Sistema de archivos.

ext4 (Fourth extended file system)

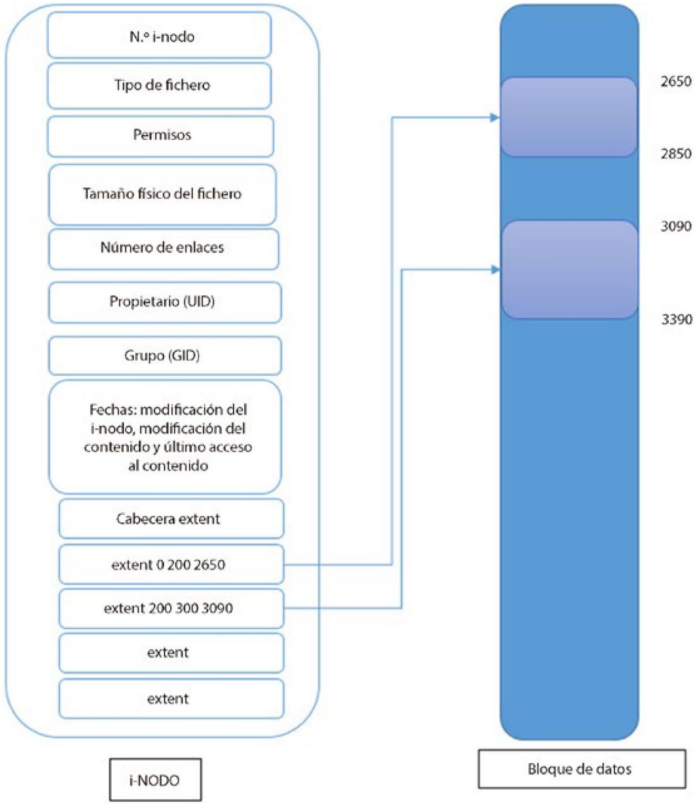
La estructura fundamental es el i-nodo o nodo índice. Este almacena toda la metainformación asociada al archivo que representa: tipo de archivo, propietario, tamaño, fechas, número de bloques de datos, localización de los bloques de datos, etc.

Es importante la compatibilidad entre sistemas de archivos y sistemas operativos. Los sistemas de archivos ext4 y APFS no son reconocidos por Microsoft Windows, sin embargo, se puede emplear NTFS en la mayoría de distribuciones Linux. Por otro lado, exFAT se considera ideal para medios extraíbles de tipo FLASH por su compatibilidad entre la mayoría de sistemas operativos.

2. Sistema de archivos.

ext4 (Fourth extended file system)

En ext4, el i-nodo emplea extents, los cuales intervienen con ficheros grandes. Estos emplean un conjunto de clusters contiguos (en lugar de estar separados) y se gestionan de manera simple, almacenando: el inicio del bloque dentro del archivo, el número de bloques almacenados y el inicio del número de bloque físico en disco.



2. Sistema de archivos.

ext4 (Fourth extended file system)

Un i-nodo está compuesto por:

- Un identificador único de i-nodo o número de i-nodo. Único en el sistema de archivos.
- Tipo de fichero: regular, enlace simbólico, directorio o dispositivo.
- Permisos de lectura, escritura y ejecución para el propietario, grupo y otros usuarios.
- Tamaño del fichero (en bytes).
- Número de enlaces (duros). Hace referencia al número de veces que el i-nodo es referenciado en el árbol de directorios. Identificador del propietario del archivo (UID). Número identificativo único que representa a un usuario.
- Identificador del grupo (GID). Número identificativo único que representa a un grupo de usuarios.
- Fechas de última modificación de la meta-información del i-nodo (ctime), última modificación de su contenido (mtime) y último acceso a su contenido (atime).
- Cabecera extent. Contiene información asociada a los extents que lo siguen: número válido de entradas que siguen a la cabecera, máximo número de entradas que siguen a la cabecera y la profundidad del extent actual (0 apunta a bloques de datos y en otro caso hasta 4 apuntará a nodos extents intermedios que actúan como indirecciones).
- Cuatro nodos hoja extent que contienen el inicio del bloque dentro del archivo, el número de bloques almacenados y el inicio del número de bloque físico en disco.

2. Sistema de archivos.

Btrfs (B-tree FS)

Es un sistema de archivos copy-on-write anunciado por Oracle Corporation para GNU/Linux.

Su objetivo es sustituir al actual sistema de archivos ext4, eliminando el mayor número de sus limitaciones, en especial con el tamaño máximo de los ficheros; además de la adopción de nuevas tecnologías no soportadas por ext3. Se afirma también que se "centrará en la tolerancia a fallos, reparación y fácil administración".

2. Sistema de archivos.

Btrfs (B-tree FS)

Las características finales presentadas son:

- Empaquetado eficiente en espacio de archivos pequeños y directorios indexados
- Asignación dinámica de inodos (no se fija un número máximo de archivos al crear el sistema de archivos)
- Subvolúmenes (raíces del sistema de archivos internas separadas)
- Comprobación de datos y metadatos (alta seguridad de integridad)
- Compresión
- Copy-on-write del registro de todos los datos y metadatos
- Comprobación del sistema de archivos sin desmontar y comprobación muy rápida del sistema de archivos desmontado
- Copias de seguridad incrementales eficaces y mirroring del sistema de archivos
- Modo optimizado para SSD (activado a través de una opción de montaje)
- Desfragmentación sin desmontar

3. Estructura de directorios en Linux y Windows.

Cuando los sistemas operativos GNU/Linux y Microsoft Windows son instalados se crean un conjunto de directorios donde se despliegan los archivos y subdirectorios del propio sistema operativo en una estructura en forma de árbol invertido del sistema de archivos.

Un directorio del que cuelgan un conjunto de subdirectorios, ramas, sobre estos otros subdirectorios hasta llegar al extremo (ramas).

El directorio principal se llama raíz. Windows \ y en Linux /

3. Estructura de directorios en Linux y Windows.

Para hacer referencia a la localización de un directorio dentro de la estructura arbórea, se emplea el término **ruta**, camino o **path**. Así, se localiza fácilmente un archivo o directorio.

- En Linux podemos hacer referencia a la ruta de un directorio como /home/usuario/
- En Microsoft Windows se puede indicar un directorio como C:\Documents and Settings. La letra que simboliza una unidad o volumen. “C:\” indica el directorio raíz de la unidad C.

3. Estructura de directorios en Linux y Windows.

El directorio de trabajo actual es el directorio donde se encuentra situado actualmente, el equivalente a navegar con el explorador de archivos hasta una carpeta.

El directorio de trabajo actual se simboliza por .

En Linux podemos mostrar la ruta de trabajo actual mediante el comando pwd (print working directory).

3. Estructura de directorios en Linux y Windows.

El directorio padre es el que está por encima del directorio actual (en dirección al raíz). Se simboliza por “..”. De la misma manera, un directorio hijo de un directorio es el directorio que se encuentra por debajo del primero en la estructura jerárquica. La ruta de un archivo o directorio se puede indicar de dos maneras:

1. Ruta absoluta: ruta completa indicada desde el directorio raíz.

2. Ruta relativa: ruta indicada desde el directorio de trabajo actual. Se suele hacer uso de “..” y “.” para desplazarse.

El comando cd (change directory) permite cambiar de directorio. Su sintaxis es la siguiente: cd [directorio], si no se indica directorio, cambiará al directorio home del usuario y equivaldría a ejecutar cd ~

3. Estructura de directorios en Linux y Windows.

Realizar en terminal de Windows y Linux:

Nos encontramos en /home/miUsuario como directorio actual y queremos acceder al directorio raíz. ¿Cómo se puede indicar el cambio de directorio mediante ruta absoluta y relativa?

3. Estructura de directorios en Linux y Windows.

Estructura de directorios en GNU/Linux

La mayoría de los sistemas operativos GNU/Linux siguen el estándar FHS (Filesystem Hierarchy Standard). 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.

3. Estructura de directorios en Linux y Windows.

Estructura de directorios en GNU/Linux

/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.

3. Estructura de directorios en Linux y Windows.

Estructura de directorios en GNU/Linux

/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.

3. Estructura de directorios en Linux y Windows.

Estructura de directorios en GNU/Linux

/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 almacenas 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.

3. Estructura de directorios en Linux y Windows.

Actividad. ¿Cuál es la ruta del Escritorio en ambos sistemas Windows y Linux?

Llega a ella por comandos desde el directorio raíz y desde el personal.

3. Estructura de directorios en Linux y Windows.

Estructura de directorios en Windows

El directorio raíz de una partición con Microsoft Windows instalado dispone del siguiente árbol de directorios:

\Archivos de programa (Program Files): en sistemas de 32 bits, se encuentran todos los programas instalados. Sin embargo, en sistemas de 64 bits se almacenarán las aplicaciones de 64 bits.

\Archivos de programa (x86): esta carpeta se encuentra en sistemas de 64 bits y contiene las aplicaciones instaladas de 32 bits.

\PerfLogs: por defecto, está vacía, pero puede contener registros de rendimiento del sistema.

\ProgramData: carpeta oculta que contiene datos de programas genéricos para todos los usuarios del sistema.

3. Estructura de directorios en Linux y Windows.

Estructura de directorios en Windows

\Usuarios (Users): carpeta que contiene subcarpetas por cada usuario del sistema, así como la carpeta \Acceso público (\Public) y \Default (carpeta oculta):

\Acceso público: carpeta compartida por todos los usuarios del sistema donde se definen aspectos comunes a ellos. Por defecto, está compartida en red.

\Default: contiene el perfil base sobre el que se crean nuevos perfiles en el sistema. Así, al crear un nuevo usuario, su carpeta situada en C:\Usuarios, contendrá el perfil y la estructura definida en \Default.

\[nombreUsuario]: contiene un conjunto de carpetas que definen el perfil del usuario (conjunto de valores de configuración del entorno: escritorio, aplicaciones, impresoras, conexiones de red, etc.), así como la carpeta oculta AppData. Esta última carpeta contiene los datos de las aplicaciones asociadas al usuario en sí (a diferencia de \ProgramData). En ella se encuentran tres subcarpetas: Roaming, que aloja perfiles de configuración de que puedan sincronizarse entre equipos; Local y LocalLow, que almacenan el resto de archivos empleados por las aplicaciones.

3. Estructura de directorios en Linux y Windows.

Estructura de directorios en Windows

Windows: contiene el grueso de la instalación del sistema operativo. En esta carpeta destacan las subcarpetas:

\System32: contiene archivos DLL de 32 bits o 64 bits, dependiendo de si la versión de Microsoft Windows es de 32 bits o 64 bits respectivamente.

\SysWOW64: solo en las versiones de 64 bits para almacenar archivos DLL de 32 bits.

\WinSxS: conocido como el almacén de componentes de Microsoft Windows, que contiene archivos utilizados para la instalación, las actualizaciones del sistema, los Service Packs o las características de Microsoft Microsoft Windows.

3. Estructura de directorios en Linux y Windows.

Estructura de directorios en Windows

Las bibliotecas de vínculos dinámicos (DLL) son archivos que contienen código ejecutable y datos. Microsoft Windows los emplea frecuentemente, ya que aumenta la modularidad en las aplicaciones, ahorra recursos del sistema y simplifica la instalación y la ejecución de las aplicaciones.

3. Estructura de directorios en Linux y Windows.

Comandos más importantes Linux:

help	help comando
man	man comando

https://ubunlog.com/poner-las-guias-man-espanol-ubuntu/

Comandos más importantes Windows:

help	help comando o comando /?

3. Estructura de directorios en Linux y Windows.

Actividad. Navega por la estructura de directorios de Ubuntu y lista las carpetas aquí estudiadas. Para mayor detalle, puedes hacer uso del comando man hier en un terminal de Linux, el cual especificará la utilidad de cada carpeta. (comando ls para listar contenido)

Actividad. Navega por la estructura de directorios de Microsoft Windows y lista las carpetas aquí estudiadas, observando su contenido. (comando dir para listar contenido)

4. Gestión de archivos por línea de comandos.

Linux

Los administradores de sistemas suelen hacer uso de la interfaz por línea de comandos, ya que su versatilidad y potencia de uso la convierte en una herramienta ideal.

Los comandos de Linux siguen una sintaxis:

comando [opciones] [argumentos]

Cada comando varía el tipo de opciones y argumentos que pueda utilizar, e incluso puede no utilizarse ninguno en caso de ser opcionales. Las opciones pueden ser cortas (una sola letra) o largas (una palabra), antecedidas por uno o dos guiones. Se pueden emplear ambos tipos de opciones con un mismo comando, aunque las opciones cortas pueden unirse. Tanto los comandos como las opciones y los argumentos que se usen son sensibles a mayúsculas y minúsculas.

4. Gestión de archivos por línea de comandos.

Linux

Uno de los comandos que más se emplea es ls (list).

ls [opciones] [ficheros]

Permite listar el contenido de un directorio e información de archivos.

- l: muestra en formato largo.
- t: ordena por fecha de modificación.
- r: invierte el orden de salida.

4. Gestión de archivos por línea de comandos.

Linux

ls [opciones] [ficheros]

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.

4. Gestión de archivos por línea de comandos.

Linux

ls -l

Cuando se emplea la opción “– l”, aparecen clasificadas en columnas la información propia de cada fichero listado:

1. La primera columna es la lista de control de acceso o máscara de permisos. A su vez, se divide en dos partes:

    a) El primer carácter puede ser: “d” indicando un directorio, “l” un enlace simbólico, “-“ un archivo regular, “c” un dispositivo de tipo carácter, o “b” un dispositivo de tipo bloque.

    b) El resto de caracteres son los permisos asociados al usuario, al grupo y a otros, tomados en grupos de tres.

2. La segunda columna establece el número de enlaces duros asociados al archivo.

3 y 4. La tercera y cuarta columna indican el propietario y el grupo, respectivamente.

4. Gestión de archivos por línea de comandos.

Linux

ls -l

Cuando se emplea la opción “– l”, aparecen clasificadas en columnas la información propia de cada fichero listado:

5. La quinta columna hace mención al tamaño que ocupa en bytes.

6. La sexta columna se refiere a la fecha de la última modificación de su contenido (mtime) o de su creación (si no se ha modificado).

7. La séptima y última columna se refiere al propio nombre del fichero. El nombre del fichero no reside en el i-nodo, sino en el directorio.

4. Gestión de archivos por línea de comandos.

Linux

En Linux los nombres de ficheros tienen entre 1 y 255 caracteres, no pueden tener el carácter “/”, para indicar un nombre con espacios se entrecomilla o se usa \ antes del espacio.

Cualquier archivo o directorio se localiza e identifica dentro del árbol de directorios gracias a su ruta. Por ello, no pueden existir dos archivos con el mismo nombre en un mismo directorio.

4. Gestión de archivos por línea de comandos.

Linux

Actividad. Ejecuta y explica las siguientes instrucciones:

ls /home /usr

ls -l /home

ls –R /home

ls –ltra.

Actividad. Ejecuta y explica las siguientes instrucciones:

cd .

cd ..

cd ./..

4. Gestión de archivos por línea de comandos.

Linux

Todos los elementos físicos(disco, tarjeta de red,...) o lógicos (directorio, enlace,...) se representa como un archivo para estandarizar la gestión.

Tipos:

a) Regulares: ficheros de cualquier tipo (texto, imagen, ejecutable,...)
b) Directorios: almacenan en su bloque de datos i-nodo y nombre de los ficheros que contiene.

   Ejemplo: ls -lai

4. Gestión de archivos por línea de comandos.

Linux

c) Enlaces:

Enlaces duros: Para un mismo i-nodo se le asocia distintos nombre en el mismo o en varios directorios. Solo se puede hacer con ficheros. Se crean con el comando

ln fichero fichero_enlace

Enlace simbólico: se guarda un i-nodo diferente al del fichero que enlaza y un nombre. En el i-nodo se guarda la ruta de acceso al archivo destino. Se puede usar para referenciar otros sistemas de archivos, particiones, equipos en red. Se incluye -s al crear el enlace

ln -s fichero fichero_enlace

4. Gestión de archivos por línea de comandos.

Linux

Ejemplo: crea un fichero con el comando touch

```
touch notas.txt
```

a continuación crea un enlace duro notas.txt.bck y un enlace simbólico notas.txt.s_bck

¿Cómo se sabe cuál es un enlace duro y cuál simbólico?

4. Gestión de archivos por línea de comandos.

Linux

d) Dispositivos: representan elementos físicos. La mayoría están en /dev. Existen 2 tipos:

Dispositivos por caracteres que no tienen sistema de archivos. Se transmiten los datos carácter a carácter. Teclado, impresora…

Dispositivos por bloques que almacenan información como los discos duros.

Tenemos además dispositivos virtuales como /dev/null

4. Gestión de archivos por línea de comandos.

Linux

El comando ls dispone de la opción “- -color”, que se encuentra activada por defecto en Ubuntu y permite discriminar el tipo de archivo según el color:

– Blanco: archivo regular.

– Verde: archivo ejecutable.

– Azul: directorio.

– Cian: enlace simbólico.

– Rojo: enlace roto.

4. Gestión de archivos por línea de comandos.

Linux

El asterisco * y cerrar interrogación ? son caracteres comodín. Estos caracteres permiten generar patrones de texto.

“*” hace referencia a cualquier cadena de caracteres.

“?” solo referencia un único carácter cualquiera.

También se puede especificar una serie de caracteres con:

[caracteres] para referirse a un conjunto de caracteres.

[!caracteres] para referirse a un carácter que no esté en el conjunto.

Los caracteres se pueden identificar como rango, ej. [a-z]

4. Gestión de archivos por línea de comandos.

Linux

El intérprete de comandos, antes de ejecutar la orden donde se incluyan los caracteres comodines, realiza una acción llamada “expansión de comodines”, donde traduce estos comodines a todas las posibles combinaciones de caracteres, según el patrón aportado.

4. Gestión de archivos por línea de comandos.

Linux

Ejemplo:

Creamos los ficheros abbbb.txt abbb.txt abb.txt ab.txt a.txt b.txt c.txt d.txt dentro de un nuevo directorio llamado ejemplo en nuestra carpeta personal.

Después ejecutamos los siguientes comandos:

ls a?.txt

ls a*.txt

ls [ab].txt

ls [!b].txt

ls [!b]*

4. Gestión de archivos por línea de comandos.

Linux

Se pueden eliminar archivos con el comando rm.

rm [opciones] [lista de ficheros]

Opciones más frecuentes:

- 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. Gestión de archivos por línea de comandos.

Linux

Ejercicio:

Elimina los archivos del directorio ejemplo que empiecen por a solicitando confirmación.
Elimina todos los archivos que terminen en .txt

Crea un archivo llamado origen1. Crear un enlace duro sobre origen1 llamado origen_d. Crear un enlace simbólico sobre origen1 llamado origen_s. Lista el contenido del directorio en formato largo mostrando los números de i-nodo. Elimina el archivo origen1. Vuelve a listar el contenido del directorio en formato largo, mostrando los números de i-nodo. ¿Qué ha ocurrido? Eliminar el archivo origen_d. Volver a listar el contenido del directorio en formato largo, mostrando los números de i-nodo. ¿Qué ha ocurrido?

4. Gestión de archivos por línea de comandos.

Linux

Para crear directorios tenemos un comando especial:

mkdir [lista directorios a crear]

Para borrar directorios tenemos:

rmdir [lista directorios vacíos a borrar]

Si queremos borrar un directorio con todo su contenido podemos usar:

rm -r [directorio]

https://en.wikipedia.org/wiki/With_great_power_comes_great_responsibility

4. Gestión de archivos por línea de comandos.

Linux

Para copiar archivos

cp [opciones] lista_archivos_origen directorio_destino

Por ejemplo si queremos hacer una copia de seguridad del directorio actual a uno de backup podemos ejecutar

cp -r . ../backup

4. Gestión de archivos por línea de comandos.

Linux

Para mover de directorio o cambiar de nombre un archivo tenemos mv. Funciona como cp, solo que borra el fichero origen.

mv [opciones] lista_archivos_origen destino

La opción i asegura en destino no exista un archivo con el mismo nombre ya que lo sobrescribirá.

La opción u solo mueve archivos o directorios si son versione modificadas más recientes.

4. Gestión de archivos por línea de comandos.

Linux

Ejemplo:

Creamos una carpeta backup en el directorio personal.

Copiamos todos los archivos del directorio ejemplo.

Dentro de la carpeta backup editamos el fichero a.txt

```
nano a.txt
```

Le añadimos un texto y guardamos.

Hacemo un mv -u ejemplos/a.txt backup

¿Se modifica el fichero? No.

4. Gestión de archivos por línea de comandos.

Linux

Para ver el contenido de los ficheros existen varios comandos.

cat [opciones] lista_ficheros

more [opciones] lista_ficheros

less [opciones] lista_ficheros

head [opciones] lista_ficheros

tail [opciones] lista_ficheros

4. Gestión de archivos por línea de comandos.

Linux

Para ver el contenido de los ficheros existen varios comandos.

cat [opciones] lista_ficheros

Permite ver el contenido completo de un fichero. Lo imprime de inicio a fin.

Comando interesante -n

4. Gestión de archivos por línea de comandos.

Linux

Para ver el contenido de los ficheros existen varios comandos.

more [opciones] lista_ficheros

less [opciones] lista_ficheros

Muestra el fichero por pantallas (para ver sin tener que usar la rueda de ratón)

less hace todas las funciones de more y además retroceder. (Usa los comandos de vi)

4. Gestión de archivos por línea de comandos.

Linux

Funciones de less:

Espacio -> siguiente página

Intro -> siguiente línea

q -> salir

/texto -> busca el texto

n -> busca la siguiente ocurrencia de texto

:n -> siguiente fichero

:p -> fichero previo.

4. Gestión de archivos por línea de comandos.

Linux

Para contar el contenido de un fichero tenemos el comando wc.

wc [opciones] ficheros

l para el número de líneas.

w para el número de palabras.

c para el número de bytes.

L longitud de la línea más larga.

4. Gestión de archivos por línea de comandos.

Linux

Para ordenar las líneas de un fichero tenemos el comando sort. Lo hace según caracteres ASCII

sort [opciones] fichero

f ignorar mayúsculas y minúsculas.

r invertir orden.

n ordenar numéricamente.

u elimina entradas repetidas.

4. Gestión de archivos por línea de comandos.

Linux

Sort permite leer los ficheros como si fuesen tablas donde cada línea es una fila y las columnas se indican mediante caracteres delimitadores. Por defecto son los espacios, tabuladores y carácter de fin de línea.

Con la opción k podemos indicar el número de la columna por la que queremos ordenar y con t podemos definir el delimitador que queramos

Por ejemplo los csv.

4,casa,950123456

2,trabajo,950654321
1,trabajo2,950000001

4. Gestión de archivos por línea de comandos.

Linux

Entradas, salidas y redirecciones.

El sistema operativo asigna automáticamente a cada comando entradas y salidas estándar asociadas a los flujos de entrada y salida, respectivamente.

Por defecto, se asignan tres ficheros: entrada estándar (/dev/stdin), salida estándar (/dev/stdout) y salida de errores estándar (/dev/stderr). Además, estos flujos de entrada o salida se identifican por un número o descriptor de fichero: 0, 1 y 2 para stdin, stdout y stderr, respectivamente.

4. Gestión de archivos por línea de comandos.

Linux

Entradas, salidas y redirecciones.

La entrada estándar nutre al comando de información para su ejecución, la salida estándar transmite el resultado y, si durante la ejecución de un comando sobreviene un error o aviso se enviará a la salida de errores estándar.

Normalmente, la entrada estándar es el teclado y la salida estándar y la salida de errores estándar se asocian a la pantalla.

4. Gestión de archivos por línea de comandos.

Linux

Podemos emplear el operador “ > ” para volcar la salida de una orden sobre un fichero en lugar de a la salida estándar:

orden > fichero

Si el fichero no existe, se crea un nuevo y, si existe, sobrescribe su contenido. También se puede utilizar “ 1> ”.

También se puede emplear el operador “>>” para realizar la misma acción, pero, a diferencia del operador “>”, añade su contenido al fichero sin sobreescribirlo.

4. Gestión de archivos por línea de comandos.

Linux

Ejemplos:

Ejecutar cat /etc/passwd. La entrada estándar es el fichero /etc/passwd y la salida estándar y de error es la pantalla.

Ejecutar cat > notas.txt. La entrada estándar es el teclado y la salida estándar se redirecciona al fichero notas.txt.

Ejecutar ls /usr >> notas.txt. La entrada estándar es el resultado de la ejecución del comando ls /usr, el cual no se muestra por pantalla, ya que se redirecciona al fichero notas. txt, donde se añade al contenido existente en este.

4. Gestión de archivos por línea de comandos.

Linux

Podemos redireccionar un fichero como entrada de una orden, en lugar de la entrada estándar. Para ello, se emplea el operador “<”:

cat < notas.txt

Existe una redirección particular que permite introducir texto hasta que se encuentre una línea únicamente con el delimitador establecido.

cat << EOF

4. Gestión de archivos por línea de comandos.

Linux

La salida de error estándar puede tener o no actividad. Se puede redireccionar empleando el operador 2> .

Del mismo modo, se puede emplear el operador “2>>” para añadir contenido al fichero.

4. Gestión de archivos por línea de comandos.

Linux

Ejemplos:

Ejecutar ls s (dado un archivo s inexistente). Por tanto, el terminal ofrece un mensaje de error por la salida estándar. Si ejecutamos ls s 2> error.txt redireccionamos la salida de error estándar al fichero error.txt y no será mostrado por pantalla:

Ejecutar ls –R /usr > salida.txt 2> salida_err.txt. El comando “ls –R /usr” almacena su salida en el fichero salida.txt y los errores o avisos se almacenarán en salida_err.txt.

Ejecutar ls –R / 1> listado.txt 2> errores.txt. El listado del directorio raíz de manera recursiva se redirige al archivo listado.txt, mientras que todos los errores generados (principalmente, porque no se poseen permisos de acceso a determinados directorios) se envían al fichero errores.txt.

4. Gestión de archivos por línea de comandos.

Linux

Ambas salidas se pueden redireccionar al mismo destino mediante el comando &> o &>>. El primero sustituye y el segundo añade contenido al fichero.

Ejemplo: ls –R / &> listado_y_errores.txt

Se suele emplear el operador 2>&1 para que la salida de error se dirija al mismo lugar que la salida estándar.

Ejemplo: ls –R / > listado_y_errores.txt 2>&1

4. Gestión de archivos por línea de comandos.

Linux

Tuberías o pipes

Con una tubería puedes hacer que la salida estándar o stdout de un comando pueda pasar directamente a la entrada estándar de otro. Es decir, puedes hacer que un programa alimente a otro. En vez de usar parámetros introducidos por el teclado, con una pipe se le entrega la información generada por el comando previo mediante esta canalización representada con el símbolo | (Alt Gr +1)

4. Gestión de archivos por línea de comandos.

Linux

Tuberías o pipes

Al igual que antes, con el operador |& podemos redireccionar la salida estándar y la salida de error estándar de una orden a la siguiente. Podemos emplear el operador “tubería” a la vez que enviamos la información de la salida estándar a la salida estándar y a un archivo mediante la orden tee.

4. Gestión de archivos por línea de comandos.

Linux

Ejemplos:

Ejecutar ls -1 /usr | wc –l. La orden ls -1 /usr lista en una columna una línea por cada fichero o directorio. Su salida estándar se redirecciona a la entrada estándar de wc –l que a su vez envía a su salida estándar (pantalla) el resultado: 9¿?. Hemos obtenido el número de ficheros o directorios de /usr.

Ejecutar ls -1 /usr | tee listado | wc –l. La salida estándar de ls -1 /usr se envía a la orden tee que almacena en el fichero listado el resultado de la orden anterior, a la vez que envía esta a la siguiente orden.

4. Gestión de archivos por línea de comandos.

Linux

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.

4. Gestión de archivos por línea de comandos.

Linux

Procesamiento de textos

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

–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.

4. Gestión de archivos por línea de comandos.

Linux

Ejemplos:

Obtener el primer campo del fichero /etc/passwd: cut –f1 –d: /etc/passwd. El comando cut toma la entrada estándar del fichero /etc/passwd. Los campos se establecen por el delimitador ":" gracias a la opción "–d". La opción –f1 hace referencia al primer campo. Si quisiéramos añadir más campos, bastaría con emplear comas para campos independientes o guiones para intervalos de campos, como: cut –f1,4-7 –d: /etc/passwd.

Obtener los caracteres 1º, 2º, 3º, 4º y 20º del fichero /etc/passwd:

cut –c1-4,20 /etc/passwd

4. Gestión de archivos por línea de comandos.

Linux

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 …]

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.

4. Gestión de archivos por línea de comandos.

Linux

Expresiones regulares:

Símbolo	Significado	Ejemplos
.	Cualquier carácter, excepto el carácter fin de línea	Cas.
*	Cero o más repeticiones del carácter que le precede	C*
[lista]	Coincide con uno de los caracteres presentes en la lista	[aCgh]
	Se puede indicar la negación de la coincidencia de un patrón	[^aCgh]
	Se pueden indicar rangos de caracteres, si se incluyen guiones y estos caracteres se especifican de mayor a menor	[0-9] [^0-9]
^	Comienzo de línea	^C
$	Fin de línea	a$

Los símbolos que tienen un significado especial se pueden emplear si se antecede con el carácter "\" en los patrones. Como, por ejemplo "*" o "\^".

egrep o grep -E permiten más expresiones regulares.

4. Gestión de archivos por línea de comandos.

Linux

Ejercicios:

Localizar aquellas líneas que contengan el patrón “root” en el fichero /etc/passwd:

Crear un fichero de texto para trabajar con él. Este fichero se va a titular Pirata.txt y contendrá el siguiente texto:

Con diez cañones por banda,

viento en popa a toda vela,

no corta el mar, sino vuela,

un velero bergantín;

bajel pirata que llaman,

por su bravura el Temido,

en todo el mar conocido,

del uno al otro confín,

4. Gestión de archivos por línea de comandos.

Linux

Ejercicios:

Listar aquellas líneas que tengan el patrón “el mar”, mostrando además el número de línea donde se encuentren dentro del fichero.

Mostrar aquellas líneas que tienen una “u” seguida de cualquier otro carácter.

Mostrar aquellas líneas con palabras con “u” seguida de cualquier otro carácter.

Mostrar aquellas líneas que comienzan por “en”.

Mostrar aquellas líneas que terminan en “,”.

Mostrar aquellas líneas que comienzan por “C”, seguidas de cero o más repeticiones de cualquier carácter y terminen en “,”.

Mostrar aquellas líneas que contengan cadenas de caracteres que no contienen una letra mayúscula, seguida del carácter “a” y del carácter “,”.

Mostrar aquellas líneas que no contengan la palabra “en”.

Parte 2

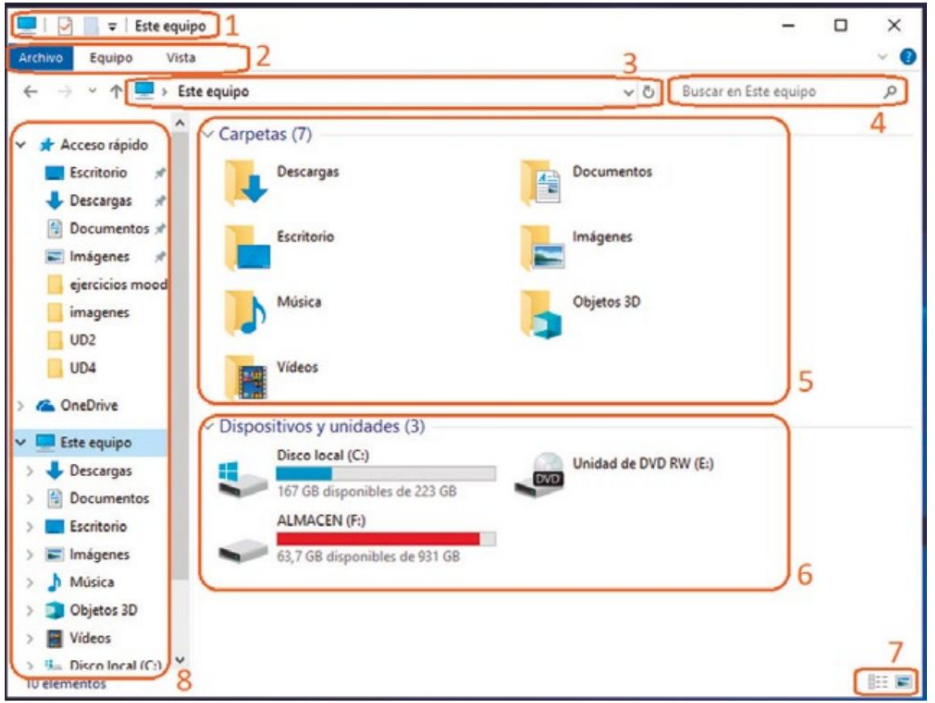
5. Gestión de archivos por interfaz gráfica en Microsoft Windows

El ‘Explorador de archivos’ de Microsoft Windows facilita la gestión de los archivos sin necesidad de conocer la administración interna del sistema.

Este dispone de accesos rápidos a carpetas muy utilizadas: ‘Escritorio’, ‘Imágenes’ o ‘Descargas’, así como a las unidades conectadas.

5. Gestión de archivos por interfaz gráfica en Microsoft Windows

1. Barra herramientas
2. Cinta de opciones
3. Barra direcciones
4. Cuadro de búsqueda
5. Lista de archivos
6. Lista de dispositivos
7. Iconos de vista de archivos
8. Panel de navegación



5. Gestión de archivos por interfaz gráfica en Microsoft Windows

El propio ‘Explorador de archivos’ se puede configurar mediante la ‘Barra de herramientas’ de acceso rápido (pulsando en el botón con forma de triángulo), habilitando o deshabilitando algunas opciones.

El ‘Explorador de archivos’ se adapta a la navegación de la estructura de carpetas. De esa manera, dependiendo si nos encontramos en una unidad, carpeta o archivo, el entorno se va adaptando, así como las operaciones que podemos hacer con cada uno.

La cinta de opciones nos resulta de mucha utilidad, puesto que contiene en modo icono los comandos que podemos realizar sobre cada elemento sobre el que nos encontremos: copiar, pegar, mover a otro lugar, copiar a otro lugar, eliminar, cambiar el nombre, crear una nueva carpeta o un acceso directo, ver sus propiedades, etc. Todo ello en la pestaña ‘Inicio’.

Además, permite de forma muy sencilla compartir de diversas maneras un elemento en la pestaña ‘Compartir’.

5. Gestión de archivos por interfaz gráfica en Microsoft Windows

También podemos modificar la manera de organizar los archivos y directorios en la pestaña ‘Vista’. Por ejemplo, con la vista ‘Detalles’ de archivos, disponemos de más información propia de cada elemento. Incluso podemos añadir o quitar propiedades si nos situamos en la barra de detalles con el botón secundario del ratón.

En la propia pestaña ‘Vista’, el botón ‘Opciones’ permite configurar multitud de características. Entre ellas, destacan las opciones de configuración avanzada de la pestaña ‘Ver’, dónde podemos mostrar archivos, carpetas o unidades ocultas, ocultar archivos protegidos del sistema operativo u ocultar extensiones de los archivos. Algunas de estas opciones se encuentran también en la cinta de opciones.

Además, en la pestaña ‘Herramientas de unidad’ (accesible cuando se selecciona una unidad) se gestionan las unidades de almacenamiento, pudiendo cifrarlas, desfragmentarlas, liberar espacio o formatearlas.

5. Gestión de archivos por interfaz gráfica en Microsoft Windows

Para una mayor agilidad en la gestión de archivos se suelen hacer uso de teclas combinadas junto con el ratón o combinaciones de teclas rápidas.

Combinaciones	Descripción
Ctrl+x	Cortar el elemento seleccionado
Ctrl+c	Copiar el elemento seleccionado
Ctrl+v	Pegar el elemento seleccionado
Ctrl+z	Deshacer una acción
F2	Modificar el nombre del elemento
Ctrl+e	Seleccionar todos los elementos
Ctrl+d	Eliminar el elemento seleccionado y enviarlo a la papelera de reciclaje
Ctrl+click ratón sobre elementos	Seleccionar distintos elementos

6. Gestión de almacenamiento por línea de comandos en Linux

La estructura de directorios de Linux incluye los diferentes discos y particiones que el sistema operativo es capaz de gestionar.

Linux puede tratar con una partición de disco cuando esta contiene un sistema de archivos y se anexa a su árbol de directorios mediante un directorio común, al que se denomina punto de montaje.

Este directorio es uno más entre el conjunto de directorios, al que se le otorga la capacidad de acceder a través de él a un subsistema de archivos.

6. Gestión de almacenamiento por línea de comandos en Linux

Los dispositivos de almacenamiento se administran a través del directorio del sistema /dev (al igual que el resto de dispositivos físicos del equipo), el cual contiene archivos que se agrupan por tipos, atendiendo al comienzo del nombre.

/dev/hd*: interfaz para unidades de disco duro IDE.

/dev/sd*: interfaz para discos SCSI, SATA y unidades con conexión USB (unidades FLASH o discos duros externos).

/dev/tty*: consolas o terminales físicos. Para cambiar entre consolas, se ha de utilizar la combinación de teclas CTRL+ALT+ F1..F6. Si queremos volver a la consola gráfica o entorno gráfico se emplea la combinación de teclas CTRL+ALT+ F7.

/dev/ttyS*: puertos serie.

/dev/sr* y /dev/scd*: interfaz para unidades CD o DVD.

6. Gestión de almacenamiento por línea de comandos en Linux

De tal manera, que la sintaxis para determinar la identificación de cada dispositivo o partición es la siguiente:

/dev/<id_dispositivo><letra_orden><numero_partición>.

Ejemplos:

/dev/hda: primer disco duro IDE.

/dev/sdb: segundo disco duro SCSI, SATA o con conexión USB.

/dev/sdc2: segunda partición del tercer disco duro SCSI, SATA o con conexión USB.

/dev/ttyS0: primer puerto serie.

6. Gestión de almacenamiento por línea de comandos en Linux

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]

6. Gestión de almacenamiento por línea de comandos en Linux

Montaje y desmontaje:

-a: monta los sistemas de archivos presentes en /etc/fstab, salvo que se indique el parámetro noauto, 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. El sistema mantiene actualizada una lista de sistemas de archivos montados

6. Gestión de almacenamiento por línea de comandos en Linux

Montaje y desmontaje:

El sistema mantiene actualizada una lista de sistemas de archivos montados a través del archivo /proc/self/mounts (en versiones anteriores se empleaba /etc/mtab, que actualmente es un enlace simbólico al anterior).

Este se interpreta cuando ejecutamos mount sin argumentos y se actualiza al montar nuevos sistemas de archivos o al desmontar sistemas de archivos existentes. Por tanto, la orden mount sin modificadores y argumentos lista los sistemas de archivos montados actualmente en el sistema.

6. Gestión de almacenamiento por línea de comandos en Linux

Montaje y desmontaje:

Por defecto, Ubuntu monta automáticamente un sistema de archivos. Las unidades Flash USB u otros dispositivos portables son gestionados por el gestor de archivos de Linux. Este gestor automatiza su conexión.

Si realizamos un montaje, hemos de indicar el tipo de sistema de archivos, la partición y el punto de montaje donde se situará el nuevo sistema de archivos (este último ha de existir).

6. Gestión de almacenamiento por línea de comandos en Linux

Montaje y desmontaje:

Para realizar un montaje debemos realizar los siguientes pasos:

1. Detectar el nombre asignado por el sistema a la partición objeto de montaje.
2. Crear el punto de montaje, si este no lo está.
3. Montar el sistema de archivos en el punto de montaje. Por defecto, solo el usuario root tiene permisos para ejecutar este comando.

6. Gestión de almacenamiento por línea de comandos en Linux

Montaje y desmontaje:

Para detectar el nombre asignado por el sistema a la partición objeto de montaje tenemos varias opciones. Si sabemos que es un disco SCSI, SATA o con conexión USB, podemos ejecutar:

ls -1tr /dev/sd*

lsblk: lista la información de los dispositivos por bloques: nombres, tipos, puntos de montaje, tamaños, etc. Si ejecutamos lsblk - -fs, mostrará el sistema de archivos.

lshw –C disk: muestra información detallada de los discos conectados al sistema.

fdisk –l: muestra información de los dispositivos o particiones que figuran en /proc/partitions, el cual registra los dispositivos conectados.

6. Gestión de almacenamiento por línea de comandos en Linux

Montaje y desmontaje:

Para detectar el nombre asignado por el sistema a la partición objeto de montaje tenemos varias opciones. Si sabemos que es un disco SCSI, SATA o con conexión USB, podemos ejecutar:

lsusb: muestra información de los buses USB y los dispositivos conectados a ellos. Esto permite recabar información sobre los tipos de dispositivos USB conectados: lsusb –tv.

dmesg: lista el contenido del buffer de mensajes del núcleo de Linux. Este comando muestra gran cantidad de información. En nuestro caso, podemos averiguar el último dispositivo conectado ejecutando este comando justo después de conectar un dispositivo, mostrando una serie de mensajes relativos a la conexión de un nuevo dispositivo USB y su posterior configuración. En caso de que el sistema tenga mucha actividad, podríamos filtrar, por ejemplo, con dmesg | grep sd.

6. Gestión de almacenamiento por línea de comandos en Linux

Montaje y desmontaje:

Para dejar de usar completamente o extraer un dispositivo con sistema 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>

Por ejemplo:

umount /dev/sdb1

umount /home/david/pendrive

6. Gestión de almacenamiento por línea de comandos en Linux

Montaje y desmontaje:

Ejercicio: Montar una memoria externa en VirtualBox

1. Descarga y añade el disco a VirtualBox

   https://drive.google.com/file/d/1aZqJvL3uxwc00IWikCFL4tJT7TfaciUj/view?usp=sharing

2. Inicia la máquina virtual
3. Detecta el nombre asignado
4. Crea una carpeta en tu directorio personal, llamada disco2 para este disco.
5. Monta el nuevo disco en disco2.
6. Desmonta el nuevo disco.

6. Gestión de almacenamiento por línea de comandos en Linux

Formato del fichero fstab del directorio /etc:

File system: partición.

Mount point: punto de montaje.

Type: tipo de sistema de archivos que contiene la partición. En caso de indicar “auto”, detecta el sistema de archivos automáticamente.

6. Gestión de almacenamiento por línea de comandos en Linux

Formato del fichero fstab del directorio /etc:

Options: opciones de montaje. Son semejantes a las del comando mount. Las opciones más comunes son:

- auto: monta el sistema de archivos durante el arranque. Este es el valor por defecto.
- noauto: el sistema de archivos se montará solo manualmente.
- ro: monta el sistema de archivos en modo solo lectura.
- rw: monta el sistema de archivos en modo lectura-escritura.
- user: permite a cualquier usuario montar el sistema de archivos.
- users: cualquier usuario del grupo users puede montar el sistema de archivos.
- nouser: solo el usuario root puede montar el sistema de archivos.
- defaults: establece una serie de opciones de montaje predeterminadas.
- errors=VALOR, establece una acción en caso de que en el sistema de archivos se produzca un error. Si VALOR es:
  - continue: el sistema sigue funcionando.
  - remount-ro: el sistema se reinicia en modo de solo lectura.
  - panic: se apaga el sistema.

6. Gestión de almacenamiento por línea de comandos en Linux

Formato del fichero fstab del directorio /etc:

- Dump: habilita o deshabilita la copia de seguridad mediante el comando dump. Normalmente no se encuentra instalado este programa, por lo que la opción más común es 0 (deshabilitado). Con valor 1, dump hace una copia de seguridad del sistema de archivos.
- Pass: establece el orden en el que se comprueban los sistemas de archivos. Con un valor 0 no se comprueba el sistema de archivos, 1 se comprueba en primer lugar y 2 se comprueba en segundo lugar. Normalmente, el sistema de archivos raíz ha de comprobarse en primer lugar y el resto en segundo lugar.

6. Gestión de almacenamiento por línea de comandos en Linux

Para identificar la partición no nos podemos fiar por el fichero de bloques que sale en en /etc . Necesitamos un identificador que no cambie si el disco se conecta a otro puerto físico.

1. Mediante una etiqueta: Es el nombre nos sale al montarla. Se puede editar con GParted/Discos cuando no esté montada. Además hay una serie de comandos:
   - NTFS: ntfslabel <partición> <etiqueta>
   - FAT: fatlabel <partición> <etiqueta>
   - ext2/3/4: e2label <partición> <etiqueta>
   - swap: swaplabel –L <etiqueta> <partición>

   Ejemplo: sudo e2label /dev/sda1 sistemaraiz

6. Gestión de almacenamiento por línea de comandos en Linux

Para identificar la partición no nos podemos fiar por el fichero de bloques que sale en en /etc . Necesitamos un identificador que no cambie si el disco se conecta a otro puerto físico.

2. Mediante un UUID. El UUID o identificador único universal se asigna a la partición cuando esta es formateada. Si se vuelve a formatear una partición o se modifican sus características (como su tamaño), cambiará su UUID.

Podemos ver las etiquetas y los UUID de las particiones del sistema mediante lsblk --fs

6. Gestión de almacenamiento por línea de comandos en Linux

La identificación de etiquetas o UUID en el archivo /etc/fstab es sencilla, basta con indicar LABEL=<nombre-etiqueta> o UUID=<código_UUID> en la columna de file system.

Para ello, editamos el archivo /etc/fstab con privilegios de root y añadimos las modificaciones necesarias.

6. Gestión de almacenamiento por línea de comandos en Linux

Crear particiones:

Existen varias herramientas para crear y manipular particiones en Linux, teniendo en cuenta que la mayoría de tareas de administración con particiones necesitan privilegios de root.

Las más conocidas son fdisk, gdisk, parted y GParted. Todas trabajan con esquemas de particionamiento MBR y GPT, a excepción de gdisk, que lo hace solo con GPT.

Las tres primeras se manejan a través de la interfaz de texto mediante un menú interactivo. Tanto fdisk como gdisk (GPT fdisk) presentan un conjunto de opciones muy similares, sin embargo, parted dispone de otras opciones y permite, además, formatear las particiones.

Ubuntu con escritorio GNOME, incorpora la herramienta de partición de discos GNOME-disks (Discos de GNOME) a la que podemos acceder como “Discos”. Permite desmontar, eliminar y establecer opciones adicionales de la partición de forma sencilla.

6. Gestión de almacenamiento por línea de comandos en Linux

Crear de particiones:

Existen varias herramientas para crear y manipular particiones en Linux, teniendo en cuenta que la mayoría de tareas de administración con particiones necesitan privilegios de root.

Las más conocidas son fdisk, gdisk, parted y GParted. Todas trabajan con esquemas de particionamiento MBR y GPT, a excepción de gdisk, que lo hace solo con GPT.

Las tres primeras se manejan a través de la interfaz de texto mediante un menú interactivo. Tanto fdisk como gdisk (GPT fdisk) presentan un conjunto de opciones muy similares, sin embargo, parted dispone de otras opciones y permite, además, formatear las particiones.

Ubuntu con escritorio GNOME, incorpora la herramienta de partición de discos GNOME-disks (Discos de GNOME) a la que podemos acceder como “Discos”. Permite desmontar, eliminar y establecer opciones adicionales de la partición de forma sencilla.

6. Gestión de almacenamiento por línea de comandos en Linux

Crear de particiones:

sudo fdisk /dev/sdX

Ejecutando fdisk con el comando anterior nos salta el menú inicial del comando. Pulsando m podemos ver las opciones:

- d borra una partición
- F lista el espacio libre no particionado
- n añade una nueva partición
- p muestra la tabla de particiones

Todos los cambios que realicemos no se guardarán a menos que pulsemos w.

6. Gestión de almacenamiento por línea de comandos en Linux

Crear de particiones:

sudo gdisk /dev/sdX

Pulsando ? podemos ver las opciones:

b back up GPT data to a file

d delete a partition

i show detailed information on a partition

n add a new partition

o create a new empty GUID partition table (GPT)

Tiene más opciones de fdisk, además incluye opciones de recuperación y avanzadas con GPT.

Todos los cambios que realicemos no se guardarán a menos que pulsemos w.

6. Gestión de almacenamiento por línea de comandos en Linux

Ejemplo: Crear una tabla de particiones GPT en un disco mediante gdisk, estableciendo dos particiones que ocupen espacios similares en cuanto a tamaño.

Cuarto crear la segunda partición con “n”, sin indicar los tamaños para que ocupe el resto del disco.

Quinto con p comprobamos que el esquema es el correcto.

Final pulsa w para escribir cambios.

6. Gestión de almacenamiento por línea de comandos en Linux

Ejemplo: Crear una tabla de particiones GPT en un disco mediante gdisk, estableciendo dos particiones que ocupen espacios similares en cuanto a tamaño.

Lanzamos el programa pasándole como argumento un disco flash USB de 8,1 GB: sudo gdisk /dev/sdb, mostrando la tabla de particiones actual del disco.

Primero crear una tabla de particiones nueva “o”.

Segundo crear una nueva partición “n”, y el programa nos preguntará el número de partición, siendo la primera (opción por defecto). Luego preguntará por el sector de comienzo de la partición, e indicaremos la opción por defecto para que no exista espacio sin particionar. Más tarde, hemos de indicar el último sector o tamaño de la partición que en nuestro caso indicamos 3.5GB (+3.5G). Por último, pregunta el tipo de partición ,que también estableceremos la opción por defecto.

Tercero con p comprobamos que el esquema es el correcto.

6. Gestión de almacenamiento por línea de comandos en Linux

Formateo de particiones:

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

mkfs [-t tipo_sistema_archivos] [opciones] dispositivo

Ejemplo: mkfs –t ntfs /dev/sdb1

Tipo_sistema_archivos: tipo de sistema de archivos que se desea implantar como ext2, ext3, ext4, vfat (msdos) o ntfs, entre otros.

6. Gestión de almacenamiento por línea de comandos en Linux

Formateo de particiones:

mkfs es un front-end de otros programas que implantan sistemas de archivos.

mkfs llama al programa constructor del sistema de archivos, como son mkfs.ext2, mkfs.ext3, mkfs.ext4, mkfs.vfat, mkfs.ntfs, etc.,

Y algunos de estos son enlaces simbólicos a otros, como mke2fs, mkntfs o mkfs.fat. Podemos analizarlo si ejecutamos ls -l /sbin/mk*.

Además podemos incorporar otros que sean reconocidos por GNU/Linux.

El sistema de archivos exFAT no se incorpora dentro del núcleo, por lo que si deseamos hacer uso de él para montar, leer, escribir o formatear un sistema de archivos de este tipo, podemos instalar el paquete asociado:

sudo apt install exfat-utils.

A partir de ese momento ya podríamos usar particiones exFAT

6. Gestión de almacenamiento por línea de comandos en Linux

Ejemplo: Formatear particiones del ejemplo anterior.

Partición 1 vfat

```
sudo sudo mkfs -t vfat -n USB1 /dev/sdb1
```

Partición 2 ntfs

```
sudo sudo mkfs -t ntfs -n USB2 /dev/sdb2
```