

Sistemas de archivos

El Sistema de Archivos es el componente del sistema operativo encargado de administrar y facilitar el uso de las memorias secundarias (o terciarias). Independientemente de las complejidades de los dispositivos de almacenamiento o del sistema, da al usuario una visión de alto nivel, y puede ver la información estructurada y organizada jerárquicamente bien sea de forma textual (CMD en DOS) o gráficamente utilizando un gestor de archivos (Explorador de Windows).

Teniendo en cuenta la naturaleza de los recursos de almacenamiento gestionados por los Sistemas de Archivos, los objetivos principales que deben proporcionar son:

- Implementación de los conceptos de Archivo y Directorio
- Uniformidad de nombres y rutas
- Gestión del espacio libre en el dispositivo
- Gestión del espacio usado por archivos en el dispositivo
- Eficiencia en la gestión del espacio
- Consistencia de Archivos y Directorios
- Seguridad de acceso para Archivos y Directorios
- Rendimiento y eficiencia en uso de los dispositivos
- Sencillez y ausencia de errores
- Interfaz amigable

ARCHIVO

Un archivo es un elemento de información abstracto. Los archivos son elementos identificables de forma única en el Sistema de Archivos. La manera de identificar ese elemento depende del contexto:

- Para un usuario un archivo o directorio se identifica mediante su path o ruta, es decir, un identificador textual único que contiene la secuencia de directorios en el sistema de archivos a los que es necesario acceder para llegar al elemento en cuestión, más el nombre del propio elemento.
- Para el propio sistema operativo todo archivo es una secuencia de bytes que identifica mediante un identificador numérico, cuando éste está siendo utilizado por algún proceso. Además todo archivo o directorio tiene asociados un conjunto de metadatos, o atributos, que son utilizados por el sistema de archivos para la gestión del elemento.

Los archivos tienen un nombre. Dependiendo del Sistema operativo puede tener un límite máximo de caracteres y pueden o no diferenciar las mayúsculas o minúsculas.

La extensión de un archivo es un identificador, a la derecha del último carácter "." (punto) en el nombre del archivo. Suelen ser de 3 caracteres, pero las hay con más o menos caracteres. Por ejemplo: casa.pdf index.html database.db

Las extensiones de los archivos suelen indicarnos el tipo de archivo que son y el sistema operativo suele tener algunos programas que son encargados de abrir por defecto ciertos tipos de tipos de archivos (aunque se puede cambiar el programa por defecto o abrirlo con otro).

Los atributos o metadatos que suelen tener los archivos dependerán en gran medida del tipo de Sistema de archivo empleado, pero algunos ejemplos serían:

- Fechas de creación o modificación
- Usuario propietario del archivo y usuarios y grupos con permisos de acceso
- Permisos de acceso (lectura, escritura, lectura/escritura, ejecución)
- Atributo de oculto
- Cifrado de datos

DIRECTORIO

Un directorio es un contenedor virtual en el que se almacenan una agrupación de archivos y otros subdirectorios, atendiendo a su contenido, a su propósito o a cualquier criterio que decida el usuario.

Con la llegada de los gestores de archivos y entornos gráficos en los sistemas operativos “modernos”, los directorios también se denominaron “carpetas” y se suelen representar gráficamente con esa figura.

Normalmente los Sistemas de Archivos están organizados en forma de árbol invertido, por lo que suele haber un directorio raíz y a partir se ramifican los siguientes subdirectorios y archivos. Desde el directorio raíz es desde el que se parte cuando se busca un archivo mediante una ruta de acceso absoluta. Cuando se usa una ruta de acceso relativa el archivo se busca partiendo del directorio en el que se esté trabajando (directorio activo). Las rutas de acceso se separan en **Windows con el carácter ** y en **Linux con /**

Los atributos de un directorio son similares o parecidos a los atributos de los archivos.

CONCEPTOS BÁSICOS DE ALMACENAMIENTO

Un **sector** de un disco duro es la sección real en que el disco almacena los datos. No se puede realizar lecturas o escrituras en un disco menores al tamaño de sector que tenga puesto. En un sector se puede almacenar una cantidad fija de bytes en potencias de 2. Normalmente el tamaño del sector de los discos duros es de 512bytes o 4KB.

Un **bloque o clúster** (también se le suele llamar “unidad de asignación”) es la unidad lógica y mínima de almacenamiento de un sistema de archivos. Es un tamaño normalmente configurable en el sistema de archivos y suele estar formado por un conjunto de 2^n sectores contiguos (siendo n normalmente valores del 0 al 6). No se pueden realizar lecturas o escrituras que ocupen menos del tamaño que tenga el bloque, es decir, aunque el tamaño real del archivo sea menor que el tamaño del bloque, lo ocuparía al completo. Normalmente el tamaño del bloque suele ser mayor que el del sector, pero por ejemplo, el sistema de archivos FAT podría tener un clúster de 8 bytes mientras que el disco duro el sector sería de 512bytes.

Es importante saber el tipo de sistema de archivos que vamos a usar y el tamaño de bloque que configuraremos, ya que eso nos limitará el tamaño máximo de la partición y tamaño máximo de ficheros con los que podemos trabajar. Si el bloque es muy grande y todos los archivos que tenemos son muy pequeños desperdiciaremos el espacio del disco. Por el contrario, si el bloque es muy

pequeño y los archivos son más grandes, repercutirá en el rendimiento porque tendrá que localizar todos los bloques que componen el archivo.

GESTIÓN DEL ESPACIO LIBRE

Los discos duros o memorias secundarias tienen un espacio limitado. Se hace necesario rehusar el espacio que van dejando los archivos que vamos borrando. Para mantener un control del espacio en disco, el sistema operativo tiene que mantenerlo y tener guardado en una lista o de alguna forma los bloques que están libres. Por ejemplo, cuando se crea un archivo el sistema busca en esa lista la cantidad de bloques o espacio libre que necesitará. Coloca la información en ese espacio y borra de la lista de espacio libre los bloques ocupados. Cuando se borra ese archivo, se ponen los bloques que ocupaba en la lista de espacio libre.

Hay varias formas de implementar esa lista de espacio libre:

Mapa de bits

Se representa cada bloque con un bit. Si el bit es un 1, estará libre, y si es un 0 es que está ocupado.

Ejemplo: Si un disco tiene libres los bloques 2,3,6,8 y 10 y el resto ocupados, el mapa de bits sería:

0011001010100000...

El mapa de bits requiere espacio. Ejemplo:

Con un disco de 1,3GB y un tamaño de bloque de 512bytes se necesita un mapa de bits de más o menos 332KB para llevar el seguimiento de los bloques libres.

*Bloques en el disco = $(1,3*1024*1024*1024)/512 = 2726297$ bloques y como cada bloque ocupa un bit en el mapa de bits, entonces se necesitarán igual número de bits, que pasados a bytes son $340787 \text{ bytes} / 1024 = 332.8\text{KB}$*

Ventajas:

- Relativamente simple y eficiente para encontrar espacio libre en el disco.
- Fácil de conseguir archivos contiguos
- Para discos pequeños se puede mantener la lista en la memoria principal.

Desventajas:

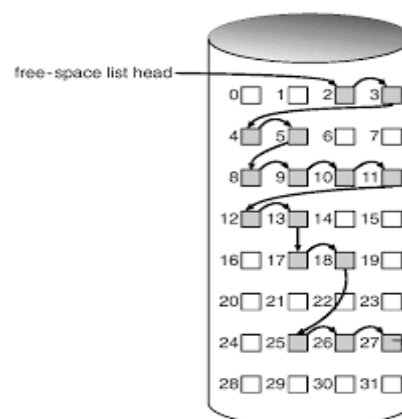
- Se desperdicia espacio por lo que no se usa en discos de gran capacidad

Lista enlazada (Linked List)

En la lista enlazada tenemos enlazados todos los bloques del disco libres entre ellos. Se pone un puntero al primer bloque libre y este bloque contiene otro puntero al siguiente bloque libre y así continuamente.

Ventajas:

- No se desperdicia espacio de disco
- Cuando se tenga que guardar un archivo en un bloque libre, el SO simplemente tendrá que localizar y ocupar el primer bloque libre de la lista y mover la cabecera del puntero al siguiente bloque libre.



Desventajas:

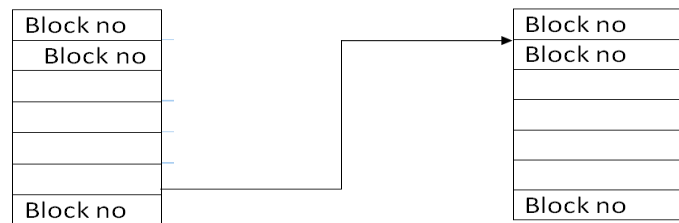
- No se puede conseguir espacio contiguo de un cierto tamaño fácilmente.
- No se accede rápidamente a cualquier bloque. Buscar en la lista es bastante costoso porque cada bloque tendrá que ser leído en el disco (requiere tiempo E/S), y la lectura de disco es muy lenta en comparación con la memoria principal.

Agrupamiento (Grouping)

Guardamos las direcciones de los n bloques libres en el primer bloque libre. De esta forma los primeros $n-1$ bloques están libres. El último bloque (n) contiene las direcciones de los siguiente bloques libres y así continuamente.

Ventajas:

- Podemos encontrar rápidamente las direcciones de un gran número de bloques libres.



Conteo (Counting)

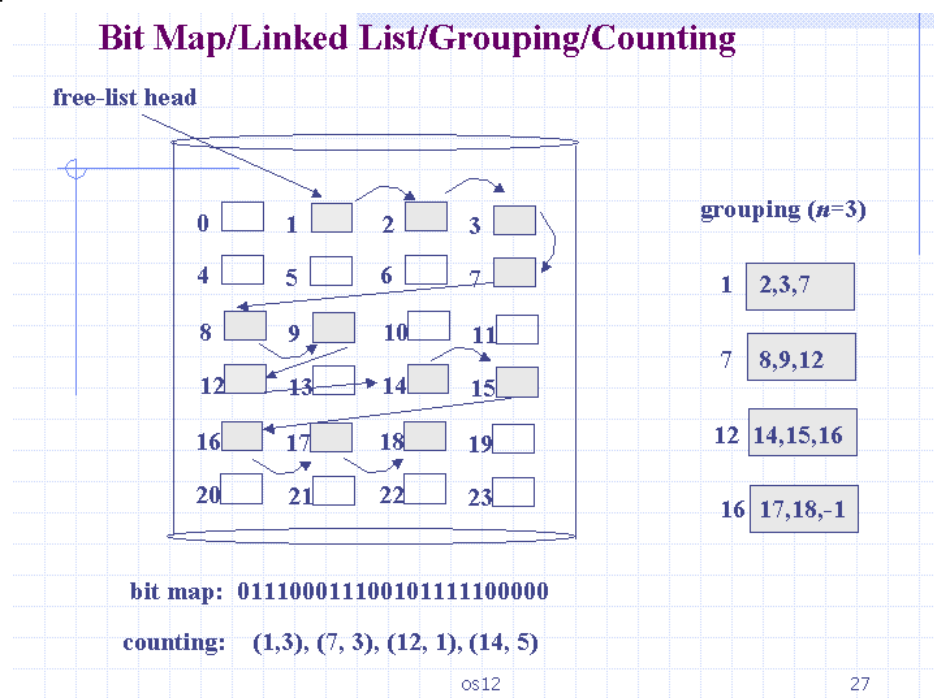
En el conteo tenemos en cuenta que normalmente muchos bloques suelen ser usados o liberados de forma contigua y simultáneamente. Por lo tanto, en este caso en vez de tener una lista que contenga todos los bloques, la lista tendrá guardar “el par” de la dirección el primer bloque libre y el número “ n ” de bloques contiguos libres siguientes. Por lo tanto, cada entrada de la lista consiste en una dirección de un bloque y un contador.

Ventajas:

- La lista será más corta (simple y cuando el número de contador sea grande).

Desventajas:

- Cada entrada en la lista de espacio libre requiere más espacio que una simple dirección de bloque.

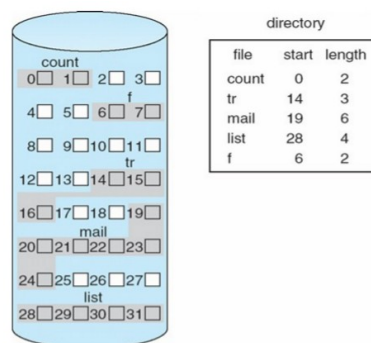


GESTIÓN DE LA ASIGNACIÓN DE ESPACIO DE ALMACENAMIENTO

De igual modo que el sistema operativo necesita gestionar y controlar el espacio que queda libre en el sistema, también gestiona la asignación de ese espacio y controla el almacenamiento de archivos.

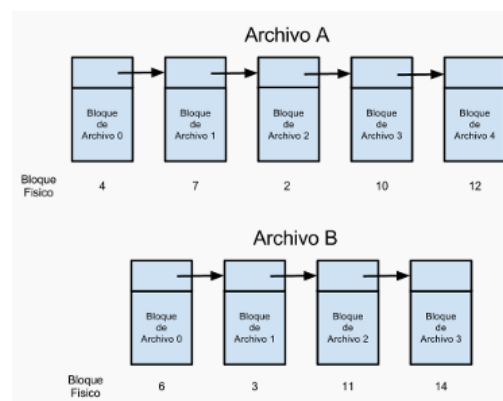
Para manejar los bloques asociados a cada archivo se pueden utilizar varias técnicas:

- **Asignación contigua o adyacente:** se almacenan los archivos mediante bloques adyacentes en el disco. Ventaja: fácil de implementar. Inconveniente: es necesario **conocer a priori el número de bloques que ocupará el fichero** y genera **fragmentación**, lo que produce, pérdida de espacio.



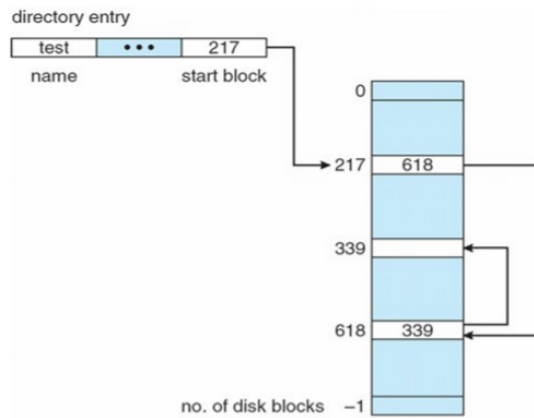
Asignación contigua

- **Asignación en forma de lista enlazada:** el directorio contiene la **dirección del primer bloque** y cada **bloque la dirección del siguiente bloque** o null en caso de que sea el último bloque del fichero.



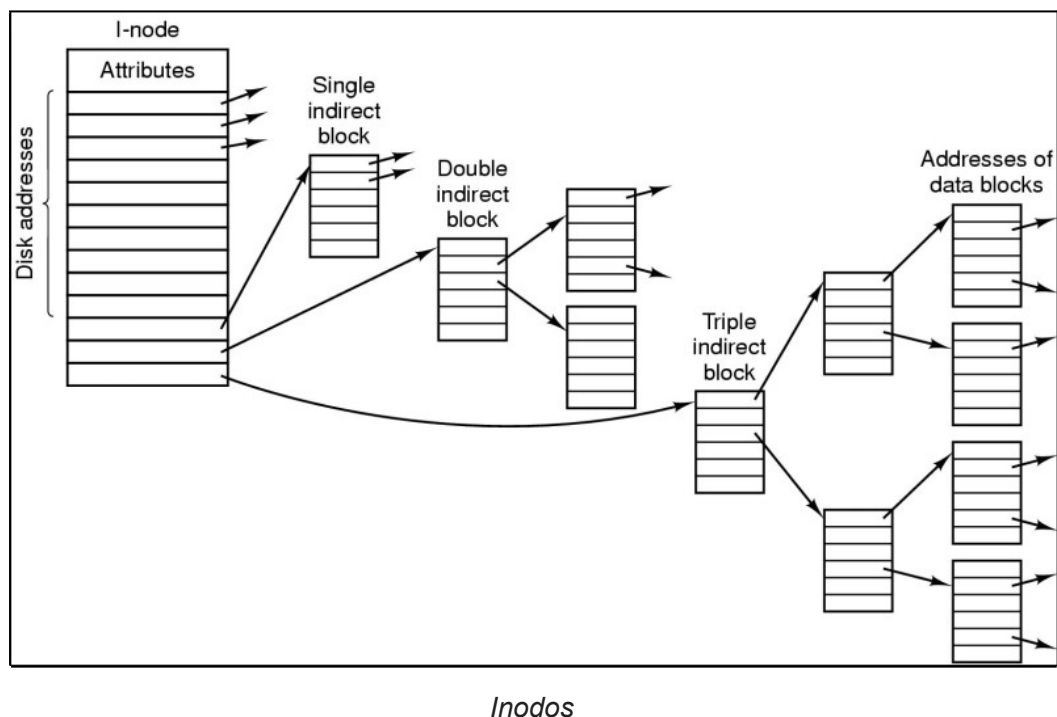
Asignación lista enlazada

- **Asignación mediante una lista enlazada y un índice:** se crea una tabla con un **registro por cada uno de los bloques del disco**, en cada registro se indica si dicho bloque está libre o cuál es la dirección del siguiente bloque. Así, en el directorio se asocia con el nombre del archivo el número de bloque en el que comienza dicho archivo. **Utilizada en FAT 16 y FAT 32.**



Asignación lista enlazada con índice

- **Basado en inodos:** se asocia a cada archivo una pequeña tabla, llamada **inodo**, que contiene los atributos y direcciones en disco de los bloques del archivo. Utilizado en **Linux**. Un archivo ordinario tiene algunas de las siguientes propiedades:
 - **Identificador de dispositivo** que alberga al sistema de archivos.
 - **Numero de inodo** que identifica al archivo dentro del sistema de archivos
 - **Longitud** del archivo en bytes.
 - Identificador de **usuario** del propietario del archivo
 - Identificador de **grupo** de usuarios con derechos diferenciados
 - Modo de **acceso**: capacidad de leer, escribir, y ejecutar el archivo.
 - Marcas de **tiempo**: fechas de última modificación, acceso y modificación.
 - Número de **enlaces**: número de entradas de directorio asociadas al inodo. Se utiliza para eliminar el archivo cuando se han borrado todos los enlaces.
 - Estructura de punteros, para direccionar hacia los bloques de datos (contenido) del archivo. Son 12 punteros que apuntan directamente a bloques de datos del archivo:
 - Puntero de **indirección simple**: apunta a un bloque de punteros, los cuales apuntan a bloques de datos del archivo.
 - Puntero de **indirección doble**: apunta a un bloque de punteros, los cuales apuntan a otros bloques de punteros, estos últimos apuntan a bloques de datos del archivo.
 - Puntero de **indirección triple**: apunta a un bloque de punteros que apuntan a otros bloques de punteros que apuntan a otros bloques de punteros que luego apuntan a bloques de datos del archivo.



Inodos

Los i-nodos se almacenan en la partición del dispositivo sobre la que se crea el sistema de archivos, en una zona reservada para ello. Durante el proceso de formateo del volumen físico se reserva una zona para albergar estas estructuras definitorias de los archivos y directorios.

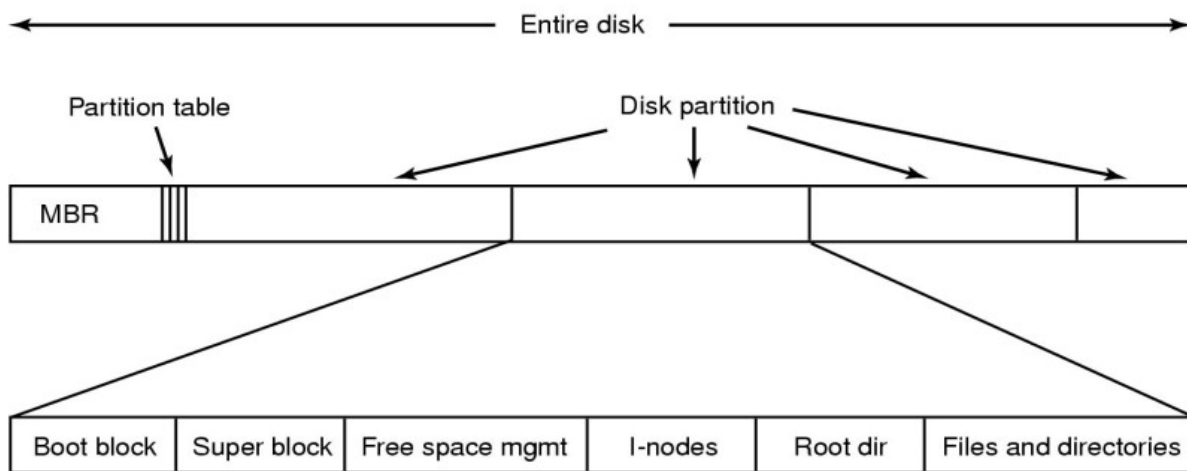
Cada i-nodo tiene asociado un identificador numérico único, de este modo cada archivo o directorio es gestionado por un único i-nodo. Este es el motivo por el cual las entradas de directorio en sistemas de archivos que utilizan i-nodos solo contienen el nombre del archivo y el número de su i-nodo asociado. Una vez localizado el i-nodo toda la información del archivo o directorio puede ser localizada a partir de éste.

Estructura del sistema de archivos en una partición de disco

Un Sistema de archivos no es más que una **estructura de almacenamiento normalizado de archivos y directorios en un volumen de almacenamiento de datos asociado a un dispositivo de bloques** (disco duro, pendrive, unidad óptica, dispositivo SSD, etc.).

Para poder desplegar un sistema de archivos necesitamos una partición dentro de un dispositivo de bloques. Dentro de la partición, durante el proceso de formateado del sistema de archivos, se crearán todas las estructuras de datos necesarias para dar soporte al almacenamiento de archivos, como por ejemplo:

- **Superbloque:** Zona en la que se indica el tipo y datos del propio sistema de archivos
- **Zona de gestión del espacio libre:** en el que se almacena por ejemplo el mapa de bits
- **Bloque de arranque,** utilizado para arrancar el sistema operativo, en caso de que sea un sistema de archivos asociado a un sistema operativo
- **Zona para almacenamiento de i-nodos**
- **Zona de almacenamiento de archivos**



Clases de sistemas de archivos

Transaccionales

Una **transacción** es un conjunto de operaciones en las que **se ejecutan todas ellas o no se ejecuta ninguna**. Las órdenes de ejecución se envían todas, una a una, pero el efecto se realiza al final, mediante una única instrucción commit (realizar la transacción) o rollback (deshacer la transacción).

Ejemplo: cuando se retira dinero del banco se debe grabar un registro en la tabla de movimientos y actualizar el saldo de la cuenta. No se puede hacer a medias, o se graban ambos, o ninguno.

Ejemplo: un programa instalador que tiene que copiar ficheros, borrar otros, renombrar otros, etc...

Este proceso se puede complicar si se produce una caída de la maquina o un simple corte de luz.

Los sistemas de ficheros transaccionales (journaling) aportan **seguridad a la integridad de los datos** y metadatos contenidos en el disco.

El **Journaling** es la propiedad de **recordar las operaciones parciales que conforman una actualización mediante un registro de diario**.

Cuando haya que modificar algún archivo, se escriben sus **datos en un nuevo bloque en lugar de sobrescribirlos y cuando todo esté correcto, se borrará el bloque antiguo con los datos iniciales**. Esta técnica se denomina **copia por escritura** y ofrece varias ventajas:

- El estado en disco siempre es válido
- Las copias de seguridad son coherentes y fiables
- Se pueden deshacer las modificaciones de los datos hasta un momento determinado

Ejemplos de sistemas de ficheros transaccionales: ReiserFS, Reiser4, Ext3, NTFS (Solo metadatos).

Distribuidos

Un sistema de archivos distribuidos (DFS) permite que los **directorios localizados en cualquier lugar de una red sean visualizados como un árbol de directorios único** sin necesidad de que los usuarios sepan en qué servidor reside cada archivo.

Cifrados

El sistema de achivos de cifrado (**EFS, Encrypting File System**) permite a los usuarios almacenar sus datos en el disco de forma cifrada.

El **cifrado** es el proceso de **conversión de los datos a un formato que no puede ser leído por otro usuario**. El **descifrado** es el proceso de **reconversión de los datos** de un formato cifrado a su formato original.

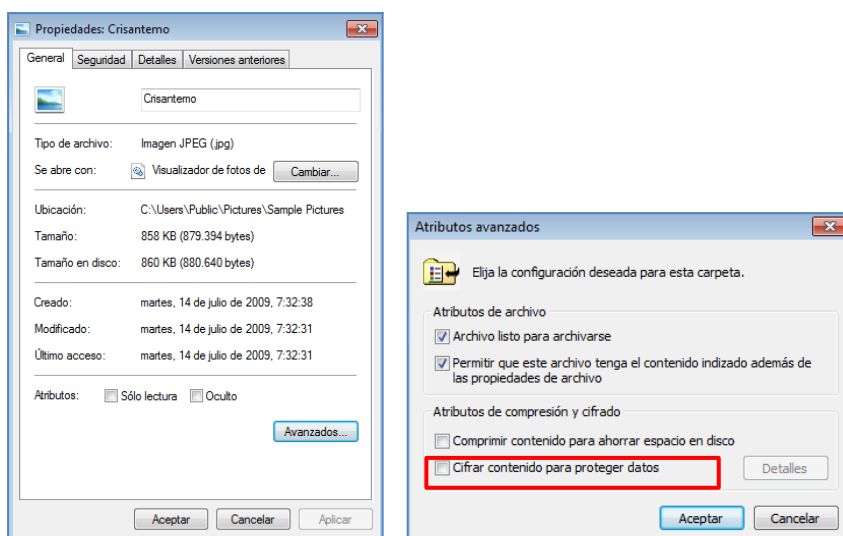
Un usuario necesita autenticarse para poder acceder a los ficheros encriptados empleando el sistema operativo. Puede encriptar los archivos fácilmente y necesitará una clave para poder descifrarlos.

Características del sistema de archivos encriptado:

- Sólo funciona en el sistema de archivos **NTFS**.
- La codificación **no altera los permisos** de los archivos y carpetas.
- Varios usuarios pueden compartir los mismos archivos cifrados.
- Si se **mueven** archivos cifrados **a otro sistema de archivos**, el cifrado desaparece.
- El cifrado **aparece como un atributo de archivo**.
- El **sistema de archivos y las carpetas no pueden ser codificados**.

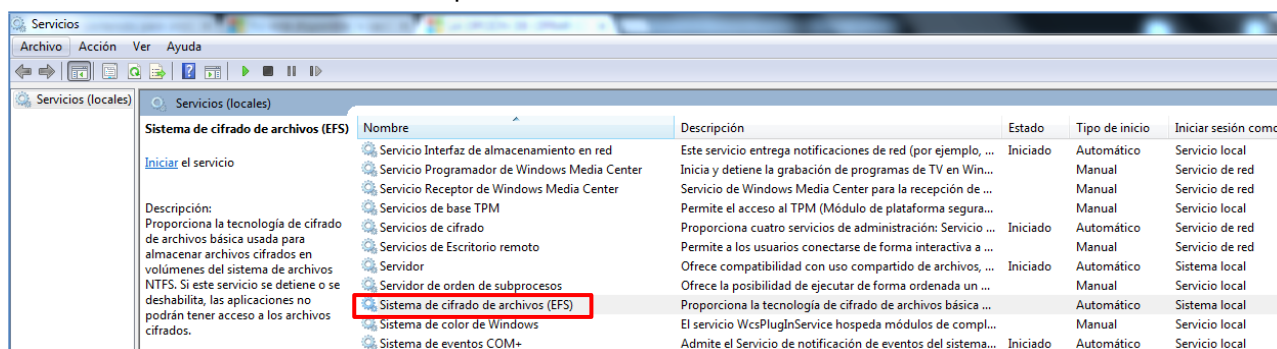
Los datos en un volumen NTFS pueden almacenarse en la unidad tal cual, pero también existe la alternativa de **comprimir o cifrar** esa información. Dichas opciones son **excluyentes entre sí**, es decir, se puede comprimir o cifrar, pero no ambas cosas al mismo tiempo.

En el **menú contextual** de cualquier archivo se pulsa sobre **"propiedades"**. En la pestaña **"general"** se pulsa sobre el botón **"avanzadas"**.



Atributos avanzados de un archivo

Si en la ventana que aparece **no está habilitada** la opción de **"cifrar contenido para proteger datos"** puede ser por varios motivos; que el **servicio EFS no esté activado**, que la partición en la que está almacenado el archivo no es ntfs o que la **versión de Windows no es la adecuada**.



Servicios

El mecanismo de **encriptación** de archivos de NTFS se basa en el uso de **una clave pública de usuario** y da lugar al sistema de archivos EFS (Encrypted File System).

Comprimidos

El sistema de archivos comprimido de sólo lectura para Linux **SquashFS** comprime archivos, inodos y directorios, y soporta tamaños de bloque de hasta 1024 KB. Es software libre (licenciado como GPL). Se utiliza en las versiones en Live CD de Debian, Finnix, Gentoo Linux, Ubuntu y Mandriva, y en sistemas embebidos.

Virtuales

Un **sistema de archivos virtual** (VFS) es una capa de abstracción encima de un sistema de archivos concreto. El propósito de un VFS es permitir que las aplicaciones cliente tengan **acceso a diversos tipos de sistemas de archivos** concretos de una manera uniforme. Puede ser utilizada para tender un **punto entre los diferentes sistemas de archivos de Windows, Mac y Unix** de modo que las aplicaciones pudieran tener acceso a archivos sin tener que saber a qué tipo de sistema de archivos están teniendo acceso.