

## Sistema de archivos

Javier Fernández Rivera - [www.aurea.es](http://www.aurea.es)

**Nota:** Archivo y fichero es lo mismo. Se usaran en las líneas de esta manual de forma aleatoria.

**Archivos:** Los archivos o ficheros son unas unidades lógicas de almacenamiento que define el propio sistema operativo. No son mas que una serie de bits cuyo significado esta definido por su creador. Pongamos el ejemplo de un fichero grafico, de una escala de grises. Este archivo seria un conjunto de bits definidos por el creador, y que internamente seria una matriz numérica, cuyos elementos representan los niveles de grises de cada uno de los píxeles de la imagen que contiene el fichero.

Los ficheros o archivos se identifican por su nombre y su extensión. Según que sistema operativo se utilice, podemos introducir un mayor o menor numero de caracteres a este nombre. Por ejemplo, MSDOS, solo permite 8 caracteres para el nombre, de hay que si en Windows ficheros con un nombre de mas de 8 caracteres, MSDOS, los reemplaza por "~1" ((Alt. +126) 1). En Windows, en realidad como sistema operativo solo hace un apaño para esto, identifica solo los primeros caracteres, el resto de nombre de un fichero o archivo lo añade pero no lo identifica con él, es un apaño XD. En otros sistemas operativos como LINUX si permite la introducción de un nombre para ficheros con mas de 8 caracteres y si los identifica con cuantos sean.

Los ficheros almacenan dentro de ellos sus propias características, como son: la fecha de su creación, la fecha de su ultima modificación, sus atributos (solo lectura, etc.), su tamaño, etc.

**Tipos y estructuras de los ficheros/archivos:** La extensión que era la segunda parte que identificaba a un fichero, es la parte que define el tipo de fichero de que se trata. Así pues si tenemos un fichero con extensión: jpg, bmp, gif, estaremos hablando de un fichero de imagen. Si es con extensión doc, será de documentos, si es: exe (ejecutable), com (de comandos), bat (procesamiento por lotes (programación lineal)"baths") ficheros ejecutables.

Un sistema operativo puede o no tener conocimiento de los distintos tipos de extensión, lo que permite un mayor servicio y rapidez. UNIX solo reconoce e interpreta los ficheros de periféricos y los directorios. El resto de extensiones de los ficheros se los pasa a la hora de ejecución al programa con el que se han creado o son compatibles.

En Windows pasa esto con ciertas extensiones. Cuando el usuario ejecuta un fichero, el sistema operativo mira haber si la extensión es reconocida por el mismo para pasar directamente al procesamiento del fichero. Si no la reconoce, coge y hace una llamada a un programa (que si reconozca esa extensión, que pueda crear ese tipo de ficheros, o que sea compatible con ellos) y junto a esa llamada al programa le pasa un parámetro (ruta \ nombre del fichero). Entonces el programa capta la orden y ejecuta el procesamiento del parámetro, el cual alberga la localización del fichero.

**Operaciones con los ficheros:** Las principales operaciones que un sistema operativo puede llevar a cabo sobre un fichero son:

- **De primer orden:** create (crear), delete (eliminar).
- **De segundo orden:** copy (copiar), open (abrir), close (cerrar).
- **De tercer orden:** rename (renombrar), write (escribir), read (leer).
- **De cuarto orden:** leer y modificar los atributos.

**Directorio de archivos:** Los directorios son tablas de ficheros donde cada entrada guarda información referida y relativa a ellos.

Así, pues un directorio no es mas que una tabla, con columnas y filas fijas variables. Esto es porque en un directorio podremos meter tantos ficheros como queramos y siempre podemos ir metiendo mas o quitando, de hay que sea obligatoriamente variable y redefinida cada vez que hacemos cambios dentro de un directorio. En cada casilla de esa tabla, podría decirse que se almacena un fichero. Llegando mas lejos, podríamos decir que un directorio viene a ser como otro tipo de archivo, el cual en vez de contener una imagen o un texto, contienen tablas de ficheros dentro del. Al igual que los ficheros un directorio también es una unidad lógica de almacenamiento, (no es tangible).

El sistema de directorios mas extendido es el del árbol. En un sistema operativo multiusuario, cada usuario dispone de su propia estructura jerárquica de directorios.

Otro de los problemas es donde ubicar el almacenamiento de directorios con sus ficheros dentro?

Si pensamos en la rapidez lo más factible sería ubicarlos en la memoria principal (RAM). Pero esto ocuparía muchísimo espacio, así que se pensó en el disco duro, y de ahí su existencia. Lo bueno de los ficheros y directorio es que estén en el ordenador, almacenados dentro del. Y que no haya que andar cargando con ellos en fuentes de almacenamiento externas como disquetes y CDs.

Bien, la estructura en árbol, es el tipo de estructura jerárquica y organizada más empleada para el sistema de directorios. Pero con ella también se plantea un problema. Y es: cuál es la raíz de ese árbol?, cuál es el directorio raíz?. Todo árbol nace de sus raíces y va creciendo y ramificando. Como sabemos cuál es el principio, la raíz, el directorio donde comenzó.

Bueno, normalmente el directorio raíz es aquel donde se arranca el sistema. Donde se encuentra almacenado el sistema operativo y donde también se almacenan los ficheros de inicio del sistema.

Cuando pulsamos el botón de arranque del ordenador, lo que hacemos es mandar una determinada cantidad de corriente, que activa los componentes del hardware, este luego se comunica con el software que en primera instancia es el sistema operativo. Y es este el encargado de inicializar todo el software que requiere el ordenador para iniciarse.

**Operaciones con directorios:** Varían mucho dependiendo del sistema operativo.

Operaciones	MSDOS
Make dir	MD
Remove dir	RD
Read dir	DIR
Rename dir	
Enlazar (link)	En LINUX con la orden ( LINK )
Desenlazar (unlink)	En LINUX con la orden ( UNLINK )
Copiar dir	Deltree

**Relación del sistema de archivos:** El sistema operativo aparte de realizar estas operaciones, también debe ser capaz de gestionar el espacio que hay en disco duro de manera que se acceda de forma eficiente y rápida a los distintos directorios y ficheros.

Esto supone elegir una política de asignación de espacio de espacio libre. Al igual que en memoria principal. Estos pueden ser consecutivos (particiones, fijas o variables), o no consecutivos, en las que se divide el fichero en bloques o paginas.

Para asignar el espacio a disco los 3 métodos más usados son: asignación contigua, listas enlazadas e índices.

**Gestión del espacio libre:** Al igual que en memoria principal para llevar un registro de que bloques del disco están llenos o vacíos sería bueno usar mapas de bits o listas enlazadas de bloques libre.

Otro problema es la elección del tamaño del bloque, si este es muy grande se puede desperdiciar mucho espacio. Con lo que se está desperdiciando memoria. Y si es muy pequeño, como cada lectura de un bloque lleva implícito un retraso para la búsqueda de ese bloque, haríamos que el disco duro se llenase de muchísimos bloques de pequeño tamaño y su lectura y búsqueda se retardaría muchísimo más.

### Métodos de asignación

**Métodos de asignación contigua:** Haber en este método el sistema operativo debe coger un grupo de bloques contiguos y vacíos en el disco duro para llenarlos con un fichero. Entonces el sistema operativo, solo requiere conocer el tamaño del fichero y la dirección del primer bloque libre. Sabiendo eso ya puede asignar el fichero a una serie de bloques libre contiguos en disco duro.

Pero este método, plantea dos grandes problemas:

- La fragmentación externa: Que se puede solucionar con procesos complejos de compactación.
- La obligatoria reubicación: Si un fichero crece es necesario reubicarlo en un sitio con más bloques contiguos. Esto se podría solucionar si el sistema operativo conociera de antemano cuál será el tamaño máximo de un fichero.

**Método de asignación mediante listas enlazadas:** Habíamos visto que un directorio tenía almacenada una tabla con los ficheros. Haber, supongamos que un fichero cuando se almacena en esa tabla o disco duro, se divide en varios bloques. Ahora bien, en este método los directorios poseen unas tablas con la información de la dirección en disco duro del primer bloque de cada fichero. Luego cada bloque tiene grabado en su cabecera la dirección al bloque siguiente y así sucesivamente hasta constituir el fichero completo. Como vemos en este método no se da la fragmentación externa y los ficheros pueden crecer sin problemas de reubicación, solamente modificando los punteros de direcciones a los bloques posteriores. No surge por tanto la necesidad del reubicamiento como anteriormente, debido a que en este caso los bloques se disponen de forma no contigua.

Pero si tiene un problema este tipo de asignación. Y es que si desde un programa queremos acceder a una utilidad que es la que internamente hace la llamada a un determinado bloque de un fichero, el sistema operativo debe primero buscar el bloque inicial de ese fichero y luego ir recorriendo secuencialmente bloque a bloque hasta que llegue al bloque que se quiere.

**Método de asignación mediante la indexación:** Una forma de subsanar el problema del acceso a un bloque determinado. Es la indexación. La indexación es una técnica que desde un "lugar" madre, origen o partida, podemos referirnos a "sublugares". Haber, lo que se hace en este método es crear por cada fichero un bloques index (índices), que contiene el puntero o la dirección en memoria de todos los bloques de ese fichero. Con lo que sí un programa necesita hacer una llamada a un bloque determinado de un fichero lo que hace es acudir al index y este ya la redirecciona directamente al bloque que desea acceder. Pero este método también plantea su problema y es la necesidad de crear ese bloque index por cada fichero. Con lo que estamos ocupando cada vez mucho más espacio, estamos "desperdiciando" memoria con "archivos / bloques de sistema"

**Método de asignación mediante indexación de varios niveles:** Este método es el que soluciona el problema del método anterior. Es el que usa UNIX. En él disponemos de un nodo index (un nodo índice) que guarda el puntero de comienzo de cualquier fichero del directorio, luego se disponen bloques de índice de primer nivel, luego de segundo nivel, luego de tercer nivel, y por ultimo los bloques de datos.

**Aspecto de diseño:** Por un lado hay que diseñar el interface para el usuario. Vendría a ser como el elemento comunicacional entre el usuario y el sistema de archivos o directorios. Y por otro lado la definición del código interno para la gestión del sistema de ficheros y directorios.

Windows usa una interface grafica, lo que es conocido como una GUI (interface grafica de usuario), en Windows los archivos vienen representados con unos iconos identificativos a cada fichero, y los directorios se representan como carpetas, etc.

El método de diseñar la interface grafica para el sistema de ficheros y directorios, explica la rápida difusión y comercialización de este sistema operativo. Haciéndolo más cómodo, rápido, manejable, intuitivo, etc. LINUX pose su modalidad de interface grafica para su sistema de archivos y directorios, con la modalidad de las XWINS.

**Comparición de ficheros:** Compartiendo un fichero lo que obtenemos es que veamos un mismo fichero en varios directorios donde hemos definido que se comparta. En UNIX se realiza con la orden LINK. En realidad no es que cree dos ficheros iguales uno para cada directorio, sino que crea una imagen lógica y virtual del fichero, pero tal imagen sola contiene almacenado en ella una dirección al fichero padre, o al fichero del cual se ha linkado o compartido. Así pues cuando ejecutamos esta imagen o fichero, lo que hace el sistema operativo es ver el contenido de ese fichero y ve que contiene un enlace al fichero padre del cual se ha linkado y por tanto devuelve el resultado de este fichero. Vendría a ser como un espejo, en realidad solo hay una persona mirándose al espejo, aunque aparentemente se vean dos personas. En Windows los accesos directos causan un efecto similar. Lo que contienen únicamente es la ruta a la dirección en memoria del fichero del cual se han creado. Y cuando se ejecuta ese acceso directo estamos en realidad haciendo una llamada de ejecución al fichero raíz.

**Caches de disco:** El acceso a disco duro es del orden de 100.000 veces mas lento que a la memoria principal, por este motivo se usan las denominadas caches de disco. Las caches son un tipo de memoria alternativa y de mas rápido acceso. El sistema consiste en llevar bloques del disco duro que son normalmente utilizados, de uso repetitivo, a la cache. Hacer un volcado de bloques del disco duro a la memoria cache de disco. De esta forma, cuando el usuario desee hacer sus operaciones comunes, estará recurriendo al acceso a la cache y no al disco duro, aunque parezca lo contrario. De hay que se optimice mas la eficiencia y velocidad. Directamente el sistema operativo llama a la cache sin recurrir en ningún momento al disco duro, en caso pues de que las operaciones que realiza el usuario estén grabadas en la cache, de no ser así, recurre al disco duro y luego lo vuelca o lo lleva a la cache, para las posteriores veces. El problema de esto es cuando se escribe en un bloque de la cache y hay que actualizarlo en el disco. De no ser así, podrían darse situaciones de corrupción de ficheros.

La solución ha esto es, volcar la cache sobre el disco duro, obligando al sistema operativo a actualizar los bloques del disco que estén cambiados en la cache. Esta operación se puede realizar periódicamente. El propio sistema operativo se encarga de hacerlo de forma automática o preprogramada por el usuario.

**El uso de la cache presenta estos problemas:**

- Aumenta el riesgo de inconsistencias o corrupciones.
- La gestión es más compleja.

**Las ventajas que presenta el uso de la cache son:**

- Mejora los tiempos de acceso y respuesta.
- Al utilizar escritura retardada las aplicaciones no tienen que esperar a la terminación de dicha escritura.
- Se puede conseguir que los ficheros temporales no se escriban en disco, por ejemplo: los que se generan al compilar en cualquier lenguaje de programación, que duran mientras se compilan.

**Seguridad y protección:** Los problemas de seguridad de la información son debidos a diferentes causas.

1. Incendios, apagones de luz, etc.
2. Averías en el ordenador, como el mal funcionamiento del procesador, errores en el disco duro, etc. Problemas del hardware.
3. Errores en los programas o en el SO (sistema operativo), etc. Problemas en el software.
4. Errores humanos como ejecuciones incorrectas, borran archivos indebidos, etc.

Los mecanismos de protección pretenden luchar contra estos ataques.

**Integridad del sistema de archivos y directorios:** Si el sistema operativo, falla en mitad de una operación de modificación y escritura de un bloque puede ser crítico y pueden ocurrir cosas como:

- Que el bloque aparezca en las listas de bloques libre y usados a la vez.
- Que el bloque este repetido.
- Que un bloque este asignado a mas de un fichero.

Existen utilidades que detectan y corrigen estos errores siempre que no sean muy graves. Cuando el deterioro del sistema es irreparable se necesitan de copias de seguridad del sistema de archivos.

Para realizar estas copias se usan unidades de cintas, o un disco duro de iguales dimensiones o mayores, para hacer un volcado del contenido del primero al segundo disco duro.

Otra técnica consiste en realizar un backup de todo el sistema. Por ejemplo, la noche del lunes de cada semana.

**Otra forma de atentar contra la integridad del sistema de archivos es:** Que un usuario legítimo abandone su puesto o terminal sin realizar un log out dejando el terminal abierto para otro usuario indebido o intruso que pueda continuar con la sesión abierta que dejó el usuario legítimo.

**Programas caballos de trola:** Estos ocultan su funcionalidad. Por ejemplo: un programa falso de login, con idéntica identificación que el original. EL usuario teclea su password y posteriormente el intruso es capaz de acceder a todo su sistema de archivos.

Un caballo de trola o un troyano, consta de dos partes principales, por un lado tiene la consola, este es el dispositivo desde el cual se controlan a los usuarios infectados. Y por otro lado tiene el fichero que cumple la misión de infectar. Este fichero al ejecutarse deja como una línea virtual abierta, deja la sesión en login. Y el diseño del troyano puede desde su consola actuar sobre todo, absolutamente todo el sistema de ficheros del usuario que ha infectado.

A continuación explicare un ejemplo, muy conocido y propio de lamers con ganas de sentirse hackers.

El lamer en cuestión es el que dispone de un troyano, se encuentra en el IRC. Y encuentra a una persona que es un poco novatillo. Entonces le dice al usuario novatillo que le va a pasar un programa de guerra o de defensa para nose que royo, wueno la típica chorada. Lo que en verdad le va a pasar es el fichero para infectarse, el que abre la sesión una vez clikeado, el que hace un auto-login. Wueno, el lamercillo se lo pasa por dcc al novato, y este lo coje con ilusión, lo abre y ve que aquello, no hace nada (en algunos troyanos el fichero desaparece). Bueno, pues apartir de que el novato ejecuto ese fichero, es como si dejara una sesión abierta, una línea de comunicación directa entre, la consola del troyano, que la tiene el lamer y todo el sistema de ficheros del novato. A partir de entonces el lamer puede actuar a sus anchas dentro del ordenador del novato, modificar, borrar ficheros, etc. Actúa totalmente sobre el software del novato. Pudiendo pues joderle completamente el ordenador.

**Algunos de estos troyanos son:**

NETBUS (NT) = puerto: 12345

BACKORIFICE (BO) = puerto: 31337

SUBSEVEN (S7) = puerto: ----

PARADISE (P) = puerto: ----

### **Gusanos y virus informáticos:**

- Los primeros son programas en si que usan desproporcionadamente los recursos del sistema bloqueando.
- Los segundos son trozos de código que se copian en otros programas o que son capaces de borrar ficheros, etc.

Normalmente el virus se suele distribuir a traves de medios que lleguen a muchos usuarios, cuantos más mejor. Por ello Internet es un buen medio para la transmisión de virus. También lo sería un juego, etc.

### **Identificación de usuarios:**

1. La seguridad de acceso de un sistema se basa en la combinación de tres tipos de identificación:
2. Por contraseña
3. Artefactos
4. Física

**Contraseñas:** Suelen estar almacenadas en ficheros. El problema es que muchos usuarios usan el nombre de ciudades, parientes, fechas de nacimiento, etc.; fáciles de obtener. Hay software especial, que tras un numero de intentos te denegan el acceso, bloquean la cuenta. Y otros que tras un periodo de tiempo determinado piden y obligan a cambiar la password.

**Artefactos:** Suelen ser bandas magnéticas, tarjetas electrónicas, donde la password esta almacenada internamente en la tarjeta lo que hace difícil hacerse con ella.

**Identificación física:** consiste en usar características físicas del usuario:

Fisiológicas: huellas dactilares, características faciales, geometría de la mano, etc.

De comportamiento: análisis de firma, o patrones de voz.

### **Mecanismos de protección**

Entendemos por objeto tanto a las unidades del ordenador (discos impresoras, etc) como a las informaciones que se almacenan (archivos de datos, programas, etc.). A cada objeto podemos acceder a traves de operaciones (leer, escribir, ejecutar). Un dominio es un conjunto de objetos y las operaciones permitidas para cada objeto. La capacidad para ejecutar una operación sobre un objeto es un derecho de acceso. Así pues un dominio es un conjunto de derechos de acceso cada uno de los cuales esta formado por un par de la forma: <nombre del objeto, conjunto de sus derechos>

En UNIX a traves del identificador de usuarios (uid) y del grupo (gid) se define el dominio. Es decir 1 lista de todos los objetos a los que puede tener acceso cada usser y el tipo de permiso de acceso