UNIDAD DE TRABAJO 1: ALMACENAMIENTO DE LA INFORMACIÓN

TEMA 1: FICHEROS: TIPOS, CARACTERÍSTICAS

1.1. CONCEPTOS. SUBDIVISIÓN DE UN FICHERO

El nombre de fichero puede tomarse en varios sentidos:

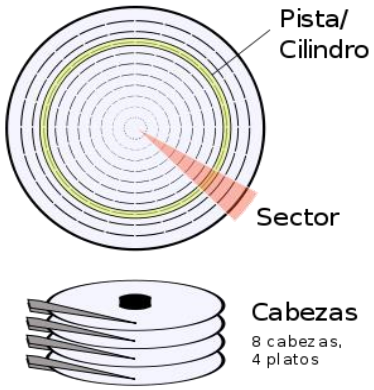
a) En el sentido amplio es un conjunto de información ordenada reservada en cualquier tipo de soporte.
b) En sentido informático es un conjunto de información contenido en un periférico de memoria masiva de datos, formando una sola unidad con un solo nombre.

Este nombre suele constar de dos partes: el nombre propiamente dicho y la extensión. El nombre nos sirve para diferenciar unos archivos de otros y la extensión para atribuirle unas propiedades concretas. Estas propiedades asociadas o "tipo de archivo" vienen dadas por las letras que conforman la extensión.
Normalmente su máximo son tres letras aunque existen algunas excepciones (.jpeg, .html, .java, etc.). Cada uno de estos pequeños grupos de caracteres está asociado a un tipo de archivo.
El final de un fichero viene determinado por el tamaño y, a veces, por uno o más bytes que marcan el final de fichero (señal de Fin de Fichero, EOF).
El tamaño puede ser cualquiera, dentro de unos límites de capacidad, pero la ocupación real puede ser mayor. (Caracteres de control para lecturas).

Para entender el subpunto “subdivisión de un fichero”, tenemos que tener claro como es la estructura de un disco (explicamos la del disco duro, dispositivo de almacenamiento más usado):
Se compone de uno o más platos o discos rígidos, unidos por un mismo eje que gira a gran velocidad dentro de una caja metálica sellada. Sobre cada plato, y en cada una de sus caras, se sitúa un cabezal de lectura/escritura que flota sobre una delgada lámina de aire generada por la rotación de los discos.

Hay varios conceptos para referirse a zonas del disco:

- **Plato**: cada uno de los discos que hay dentro del *disco duro*.
- **Cara**: cada uno de los dos lados de un *plato.*
- **Cabeza**: número de cabezales.
- **Pistas**: una circunferencia dentro de una *cara*; la *pista* 0 está en el borde exterior.
- **Cilindro**: conjunto de varias *pistas*; son todas las circunferencias que están alineadas verticalmente (una de cada *cara*).



- **Sector**: cada una de las divisiones de una pista. El tamaño del sector no es fijo, siendo el estándar actual 512 bytes, aunque próximamente serán 4 Kb. Antiguamente el número de sectores por pista era fijo, lo cual desaprovechaba el espacio significativamente, ya que en las pistas exteriores pueden almacenarse más sectores que en las interiores. Así, apareció la tecnología ZBR (**grabación de bits por zonas**) que aumenta el número de sectores en las pistas exteriores, y utiliza más eficientemente el disco duro.

El primer sistema de direccionamiento que se usó fue el CHS (**cilindro-cabeza-sector**), ya que con estos tres valores se puede situar un dato cualquiera del disco. Más adelante se creó otro sistema más sencillo: LBA (**direccionamiento lógico de bloques**), que consiste en dividir el disco entero en *sectores* y asignar a cada uno un único número. Éste es el que actualmente se usa.

1.1.1 Subdivisión de un fichero

Desde el punto de vista físico:

Se pueden considerar los siguientes elementos:

- Bloque (cluster), que equivale a un registro físico (uno o más sectores).
- Sector
- Byte

Desde el punto de vista lógico

Refiriéndonos a los ficheros de datos (que son los que nos interesan en este módulo) frecuentemente están distribuidos en unidades lógicas llamadas registros. A su vez, estos registros suelen dividirse en campos, que pueden ser de distintos tipos (numéricos, alfanuméricos, lógicos, etc.). Cada registro puede tener asociada una clave (key) de posición o búsqueda (indicativo del registro). Se llama *factor de bloque* al número de registros lógicos por bloque físico.

1.2. TIPOS DE FICHEROS

Los tipos de ficheros los podemos clasificar según diversos criterios:

Según su función:

Archivos Permanentes:

Son aquellos cuyos registros sufren pocas o ninguna variación a lo largo del tiempo, se dividen en:

Constantes o maestros: Están formados por registros que contienen campos fijos y campos de baja frecuencia de variación en el tiempo.

De situación: Son los que en cada momento contienen información actualizada.

Históricos: Contienen información acumulada a lo largo del tiempo de archivos que han sufridos procesos de actualización o bien acumulan datos de variación periódica en el tiempo.

Archivos de Movimiento:

Son aquellos que se utilizan conjuntamente con los maestros (constantes), y contienen algún campo común en sus registros con dichos archivos para el procesamiento de las modificaciones experimentados por los mismos.

Archivo de Maniobra o Transitorio:

Son los archivos creados auxiliares creados durante la ejecución del programa y borrados habitualmente al terminar el mismo.

Según sus elementos:

Archivo de Entrada: una colección de datos localizada en un dispositivo de entrada que tienen como objetivo ser utilizados por aplicaciones para realizar algún tipo de operación, ya sea ser grabados en BD, utilizarlos para actualizaciones, comparaciones, etc.

Archivo de Salida: una colección de información que será visualizada por la computadora a través del monitor, de una impresora, o cualquier dispositivo de salido, suelen ser archivos de información.

Archivo de Programa: un programa codificado en un lenguaje específico y localizado o almacenado en un dispositivo de almacenamiento.

Archivo de Texto: una colección de caracteres almacenados como una unidad en un dispositivo de almacenamiento.

Binarios: es un archivo que lee byte por byte sin asumir ninguna estructura. Contiene información de cualquier tipo codificada en binario para el propósito de almacenamiento y procesamiento. Por ejemplo los archivos informáticos que almacenan texto formateado o fotografías, así como los archivos ejecutables que contienen programas.

Muchos formatos binarios contienen partes que pueden ser interpretados como texto. Un archivo binario que *sólo* contiene información de tipo textual sin información sobre el formato del mismo se dice que es un archivo de texto plano. Habitualmente se contraponen los términos 'archivo binario' y 'archivo de texto' de forma que los primeros no contienen solamente texto.

Entre otras características, a las técnicas de archivo binario no les preocupa los caracteres EOF intercalados (Control+Z = Chr$(26)) que pueda tener un archivo.

Según el método de acceso:

Se refiere al método utilizado para acceder a los registros de un archivo prescindiendo de su organización. Existen distintas formas de acceder a los datos:

Ficheros secuenciales simples: los registros se leen desde el principio hasta el final del archivo, de tal forma que para leer un registro se leen todos los que preceden. (Cuidado, no hay que confundir el concepto de soporte secuencial con el de fichero de acceso secuencial, el primer concepto se refiere a la materialidad

del tipo de soporte físico y el segundo es un concepto lógico que dependerá de la programación).

Pueden grabarse en soporte secuencial o en soporte direccionable. Normalmente se organizan en registros, pero estos registros no están necesariamente organizados en campos y además estor registros pueden ser de longitud variables, con la ventaja de mejor aprovechamiento del espacio disponible.

Suele existir un registro de fin de fichero, para indicar que no hay más registros posteriores.

Para **consultar** un registro determinado se van leyendo todos los registro hasta que se encuentra el que se busca o el fin de fichero.

Para **insertar, añadir, borrar o modificar** un registro no se puede hacer directamente. El procedimiento será: se copia el fichero en otro fichero auxiliar. Se va copiando registro a registro del antiguo al nuevo, hasta encontrar el registro que se busca para modificar, o para insertarlo en caso de que sea una inserción ordenada, si es borrado, se saltaría dicho registro (no se copia). Por fin se sustituye el fichero antiguo por el nuevo.

Ficheros secuenciales encadenados: cada registro tiene un *campo puntero* (campo enlace) que señala la posición del siguiente registro. Son ficheros ordenados. Se emplean en soportes de acceso directo. El final del fichero se establece con el código EOF o por el marcado de un puntero especial. Suelen tener un registro ficticio o falso al comienzo del fichero para mejorar la inserción. Ejemplo:

Dirección Física	Clave de ordenación	Número de orden	Puntero enlace	Observación
0		(0)	2	Ficticio
1	CC	2	3	
2	AA	1	1	
3	FF	3	5	
4	PP	5	6	
5	LL	4	4	
6		(6)		Ficticio

Para **insertar,** por ejemplo GG se inserta en la dirección física 6, FF ya no apuntaría al 5, si no al 6, y GG apuntaría al 5.

Para **buscar** un registro se hace como en los ficheros secuenciales, es más lento. Para modificar se puede hacer de dos formas: suprimiendo e insertando o con una modificación directa.

Ficheros de acceso directo: en estos ficheros la posición de un registro viene dada por la clave y no por el instante en que se escribe. Los registros son del

mismo tamaño. Nos podemos encontrar con distintas posibilidades en cuanto a estos ficheros:

a) El caso más sencillo resulta cuando las claves son numéricas y numéricamente correlativas y el número máximo de claves coincide con el número máximo de registros. Por ejemplo, un hotel de 60 habitaciones, las habitaciones numeradas del 1 al 60 y la clave de la habitación es su número.

b) El caso más general es que la clave no sea numérica o, aunque lo sea, no se corresponde con el espacio reservado. Por ejemplo, en una empresa la clave es el DNI, pero el número máximo de empleados es 500. Si reserváramos espacio para todos los números posibles del DNI resultaría un espacio inmenso, inviable.

Normalmente el **acceso** a los registros de un fichero de acceso directo se hace a través de un índice. O también, de forma más sencilla, se busca simplemente por el número de registro. Pero además hay otra forma, a través de funciones ***hash.***

Las funciones hash: se corresponden a unas fórmulas que sirven para hallar la dirección a partir de su clave. Una dificultad que hay que resolver son los *sinónimos* que son las claves para las cuales la función hash da una misma dirección. Las funciones hash se eligen para que repartan el espacio disponible de la mejor manera y den el menor número de sinónimos. Ejemplo: el resultado de sumar las cifras de un número:

hash(4768)=25

hash(3972)=21

pero generarían sinónimos:

hash(1589)=23

hash(2975)=23

Otras posibilidades de funciones hash: división (por ejemplo tomando el resto de una división por una constante), extracción (por ejemplo tomando las tres primeras cifras de un número), elevación al cuadrado, etc.

Para la elección de una función hash entran variables como el factor de carga, también habría que ver cómo se resuelven las colisiones. No vamos a profundizar más en estas funciones.

¿Cómo resolveríamos el caso de que la clave sea el DNI y el número de empleados 800 con una función hash que haga la clave más pequeña y más rápida?: podríamos hacer DNI mod 1000.

La **inserción, supresión, modificación y consulta** en este tipo de ficheros se debe hacer teniendo en cuenta la función hash, y luego, viendo la correspondencia. En la **inserción** la cosa no es tan sencilla, pues habrá que ver si la función hash nos da una dirección vacía o no, si no es así, se tendrá que ver dónde se coloca este fichero (tratamiento de colisiones).

Ficheros secuenciales indexados: son un **conjunto** de dos o más ficheros: un fichero principal o de acceso directo y uno o más ficheros auxiliares índices. **¿Qué es un índice?:** es un fichero cuyos registros contienen dos campos: una clave para cada registro del fichero principal y la dirección de cada uno de esos

registros. El índice se crea al mismo tiempo que el fichero principal (o posteriormente, claro).

Se accede indirectamente a los registros por el índice, mediante consulta secuenciales al fichero índice que contiene la clave y la dirección de cada registro, y posterior acceso directo al registro.

Número de orden	Dirección	Campo clave	Otros campos	
5	0	Sánchez	Juan	…
2	1	García	Ana	…
0	2	Álvarez	Felipe	…
3	3	Hernán	Pedro	…
6	4	Zapata	Andrés	…
1	5	Benítez	Sara	…
4	6	Pérez	Gil	…

Campo Clave	Dirección
Álvarez	2
Benítez	5
García	1
Hernán	3
Pérez	6
Sánchez	0
Zapata	4

Para ordenar el fichero no tocamos el fichero en ningún momento, sólo tocamos el índice. Esto nos ofrece la ventaja de ocupar en memoria menos espacio durante la ordenación, ya que sólo pasamos el índice a memoria, fichero más pequeño y por tanto la operación será más rápida.

Para el **acceso**: se puede hacer como hemos dicho antes a través del índice de forma secuencial y luego al fichero principal. También por dicotomía (para índices ordenados). La **inserción** de un registro se hace introduciéndolo en el fichero principal al final o en la primera posición libre y luego reordenando el fichero índice. Para la **supresión** se cambia el índice solamente y se marca el registro como libre.

Por su función:

Ejecutables: están creados para funcionar por sí mismos

No ejecutables: almacenan información que tendrá que ser utilizada con ayuda de algún programa.

Por la temática:

De imágenes

De texto

De vídeo

Comprimidos

Etc.

En cuanto a las extensiones de los archivos, hay millones de ellas, lo mejor, si desconocemos la extensión de algún fichero, es recurrir a un buscador o páginas especializadas con extensas bases de datos de archivos.

1.3. CARACTERÍSTICAS DE LOS FICHEROS

Ya hemos visto que la característica principal de un archivo es su **nombre y extensión.**

Los nombres de archivos originalmente tenían un límite de ocho caracteres más tres caracteres de extensión, actualmente permiten muchos más caracteres dependiendo del sistema de archivos.

Datos sobre el archivo: Además para cada fichero, según el sistema de archivos que se utilice, se guarda la fecha de creación, modificación y de último acceso. También poseen propiedades como oculto, de sistema, de solo lectura, etc.

Tamaño: Los archivos tienen también un tamaño que se mide en bytes, kilobytes, megabytes, gigabytes y depende de la cantidad de caracteres que contienen. **Ubicación:** Todo archivo pertenece a un directorio o subdirectorio. La ruta de acceso a un archivo suele comenzar con la unidad lógica que lo contiene y los sucesivos subdirectorios hasta llegar al directorio contenedor.

La estructura de directorios suele ser jerárquica, ramificada o "en árbol. En los sistemas de archivos jerárquicos, usualmente, se declara la ubicación precisa de un archivo con una cadena de texto llamada "ruta" —o *path* en inglés—. La nomenclatura para rutas varía ligeramente de sistema en sistema, pero mantienen por lo general una misma estructura. Una ruta viene dada por una sucesión de nombres de directorios y subdirectorios, ordenados jerárquicamente de izquierda a derecha y separados por algún carácter especial que suele ser una diagonal ('/') o diagonal invertida ('\') y puede terminar en el nombre de un archivo presente en la última rama de directorios especificada.

1.4. EL SISTEMA DE ARCHIVOS

1.4.1. ¿Qué es un sistema de archivos?

Aunque los discos duros pueden ser pequeños contienen millones de bits, y por lo tanto, necesitan organizarse para poder ubicar la información. Éste es el propósito del **sistema de archivos**. Ya hemos visto que un disco duro se conforma de varios discos circulares que giran en torno a un eje. Las pistas (áreas concéntricas escritas a ambos lados del disco) se dividen en piezas llamadas sectores (cada uno de los cuales contiene 512 bytes). El formateado lógico de un disco permite que se cree un sistema de archivos en el disco, lo

cual, a su vez, permitirá que un sistema operativo (Windows 9x, Windows XP, Linux, Windows 7, Unix...) use el espacio disponible en disco para almacenar y utilizar archivos. El sistema de archivos se basa en la administración de clústeres, la unidad de disco más pequeña que el sistema operativo puede administrar.

Un clúster consiste en uno o más sectores. Por esta razón, cuanto más grande sea el tamaño del clúster, menores utilidades tendrá que administrar el sistema operativo. Por el otro lado, ya que un sistema operativo sólo sabe administrar unidades enteras de asignación (es decir que un archivo ocupa un número entero de clústers), cuantos más sectores haya por clúster, más espacio desperdiciado habrá. Por esta razón, la elección de un sistema de archivos es importante.

1.4.2. Sistemas de Archivos y Sistemas Operativos

En realidad, la elección de un sistema de archivos depende en primer lugar del sistema operativo que esté usando. Nos saltamos los sistemas operativos en desuso como DOS o Windows 95, que usaban FAT16 ya que los tamaños de los discos en esa época no excedían los 2 GB.

Para particiones con una capacidad menor a 500 Mb se recomienda FAT16, de lo contrario, es preferible usar FAT32.

Para sistemas operativos Windows versión servidor (hablamos de Windows NT, Windows Server 2000, Windows Server 2003) se usa NTFS que brinda una mayor seguridad y un mejor rendimiento que el sistema FAT.

Linux: ext2, ext3, ext4, Linux Swap

MacOS: HFS (Sistema de Archivos Jerárquico),

ZFS es un sistema de archivos desarrollado por Sun Microysystemes para su sistema operativo Solaris.

OS/2: HPFS (Sistema de Archivos de Alto Rendimiento)

1.5. ASIGNACIÓN DEL ESPACIO DE ALMACENAMIENTO

El subsistema de archivos se debe encargar de localizar espacio libre en los medios de almacenamiento para guardar archivos y para después borrarlos, renombrarlos o agrandarlos. Para ello se vale de localidades especiales que contienen la lista de archivos creados y por cada archivo una serie de direcciones que contienen los datos de los mismos. Esas localidades especiales se llaman directorios. Para asignarle espacio a los archivos existen tres criterios generales:

- **Asignación contigua**: Cada directorio contiene la los nombres de archivos y la dirección del bloque inicial de cada archivo, así como el tamaño total de los mismos. Por ejemplo, si un archivo comienza en el sector 17 y mide 10 bloques,

cuando el archivo sea accedido, el brazo se moverá inicialmente al bloque 17 y de ahí hasta el 27. Si el archivo es borrado y luego creado otro más pequeño, quedarán huecos inútiles entre archivos útiles, lo cual se llama *fragmentación externa*.

- **Asignación encadenada**: Con este criterio los directorios contienen los nombres de archivos y por cada uno de ellos la dirección del bloque inicial que compone al archivo. Cuando un archivo es leído, el brazo va a esa dirección inicial y encuentra los datos iniciales junto con la dirección del siguiente bloque y así sucesivamente. Con este criterio no es necesario que los bloques estén contiguos y no existe la fragmentación externa, pero en cada "eslabón" de la cadena se desperdicia espacio con las direcciones mismas. En otras palabras, lo que se crea en el disco es una lista ligada.
- **Asignación con índices (indexada):** En este esquema se guarda en el directorio un bloque de índices para cada archivo, con apuntadores hacia todos sus bloques constituyentes, de manera que el acceso directo se agiliza notablemente, a cambio de sacrificar varios bloques para almacenar dichos apuntadores. Cuando se quiere leer un archivo o cualquiera de sus partes, se hacen dos accesos: uno al bloque de índices y otro a la dirección deseada. Este es un esquema excelente para archivos grandes pero no para pequeños, porque la relación entre bloques destinados para índices respecto a los asignados para datos es incosteable.

1.6. Operaciones soportadas por el subsistema de archivos

Independientemente de los algoritmos de asignación de espacio, de los métodos de acceso y de la forma de resolver las peticiones de lectura y escritura, el subsistema de archivos debe proveer un conjunto de llamadas al sistema para operar con los datos y de proveer mecanismos de protección y seguridad. Las operaciones básicas que la mayoría de los sistemas de archivos soportan son:

- **Crear (create)** : Permite crear un archivo sin datos, con el propósito de indicar que ese nombre ya está usado y se deben crear las estructuras básicas para soportarlo.
- **Borrar (delete)**: Eliminar el archivo y liberar los bloques para su uso posterior.
- **Abrir (open)**: Antes de usar un archivo se debe abrir para que el sistema conozca sus atributos, tales como el dueño, la fecha de modificación, etc.
- **Cerrar (close)**: Después de realizar todas las operaciones deseadas, el archivo debe cerrarse para asegurar su integridad y para liberar recursos de su control en la memoria.
- **Leer o Escribir (read, write)**: Añadir información al archivo o leer el carácter o una cadena de caracteres a partir de la posición actual.
- **Concatenar (append)**: Es una forma restringida de la llamada ‘write’, en la cual sólo se permite añadir información al final del archivo.
- **Localizar (seek)**: Para los archivos de acceso directo se permite posicionar el apuntador de lectura o escritura en un registro aleatorio, a veces a partir del inicio o final del archivo.
- **Leer atributos**: Permite obtener una estructura con todos los atributos del archivo especificado, tales como permisos de escritura, de borrado, ejecución, etc.

- **Poner atributos**: Permite cambiar los atributos de un archivo, por ejemplo en UNIX, donde todos los dispositivos se manejan como si fueran archivos, es posible cambiar el comportamiento de una terminal con una de estas llamadas.
- **Renombrar (rename)**: Permite cambiarle el nombre e incluso a veces la posición en la organización de directorios del archivo especificado.

Los subsistemas de archivos también proveen un conjunto de llamadas para operar sobre directorios, las más comunes son crear, borrar, abrir, cerrar, renombrar y leer. Sus funcionalidades son obvias, pero existen también otras dos operaciones no tan comunes que son la de 'crear acceso directo' y la de 'destruir acceso directo'. La operación de crear acceso directo sirve para que desde diferentes puntos de la organización de directorios se pueda acceder a un mismo directorio sin necesidad de copiarlo o duplicarlo. La llamada a borrar acceso directo lo que hace es eliminar esas referencias, siendo su efecto la de eliminar esos *enlaces* y no el directorio real. El directorio real no es eliminado hasta que se realice la operación de borrar directorio.