

# Tema 5: Archivos

Objetivo: El alumno comprenderá las organizaciones básicas de los archivos, las operaciones que se pueden realizar sobre ellos y su representación en diferentes medios de almacenamiento secundario.

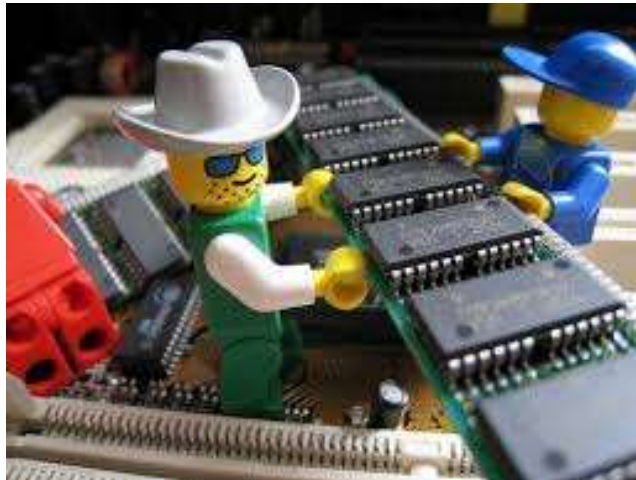
## 5.1 Generalidades

- En muchas situaciones, la cantidad de datos necesarios para realizar algún cómputo es muy grande para almacenarse en memoria principal (RAM).
- En este caso se requiere de un tipo distinto de almacenamiento, los archivos generalmente tienen mucha mayor capacidad, debido al bajo costo por byte de los sistemas de almacenamiento secundario.



## 5.1 Generalidades

- Los archivos son persistentes en la memoria: si se apaga la computadora, se pierden los datos de la memoria principal, mientras que la información almacenada en memoria secundaria se mantiene por tiempo indefinido



## 5.1 Generalidades

- Anteriormente a 1960 el motivo principal para la existencia de una computadora el calculo o procesamiento de información.
- Posteriormente a esa fecha se comenzó a ver a las computadoras como dispositivos para almacenar grandes cantidades de información.

## 5.2 Definición y Operaciones

- Un archivo (file) es una colección de información almacenada como un elemento en alguna unidad de disco de la computadora.
- Un archivo es un flujo lineal de bytes de información con un nombre asociado que se almacena en un dispositivo.
- El sistema operativo es responsable de implementar la abstracción de archivo básico sobre los dispositivos de almacenamiento tales como discos magnéticos, cintas, etc.

## 5.2 Definición y Operaciones

- Se puede almacenar información abriendo un archivo y escribiendo un bloque de bytes.
- De forma similar se puede acceder a la información almacenada en un archivo abriéndolo y leyendo el bloque de bytes almacenados en él.
- Algunas operaciones sobre archivos...
  - crear
  - abrir
  - eliminar

## 5.2 Definición y Operaciones

Los archivos se distinguen de otro tipo de recursos por dos razones:

- Son la principal forma de almacenar información
- Los sistemas operativos utilizan los archivos como primitiva y modelan otras abstracciones de recursos basándose en ella

## 5.2 Definición y Operaciones

- Los archivos se utilizan como contenedores para almacenar la información durante periodos largos de tiempo.
- De manera interna, se utiliza el sistema de archivos para almacenar programas reubicables absolutos y ejecutables.
- Los programadores se apoyan en el sistema de archivos para simplificar el uso de dispositivos de almacenamiento al guardar conjuntos de datos



## 5.3 Organización de archivos

Se refiere a las diferentes maneras en las que puede ser organizada la información de los archivos, así como las diferentes maneras en que ésta puede ser accesada.

Las dos formas de entender la organización de archivos son la organización física y la organización lógica

## 5.3.1 Organización física

- Una unidad de disco consiste en varios platos (*platters*) que rotan a velocidad constante sobre un eje común (*spindle*).
- La superficie de cada plato está cubierta de material magnético.

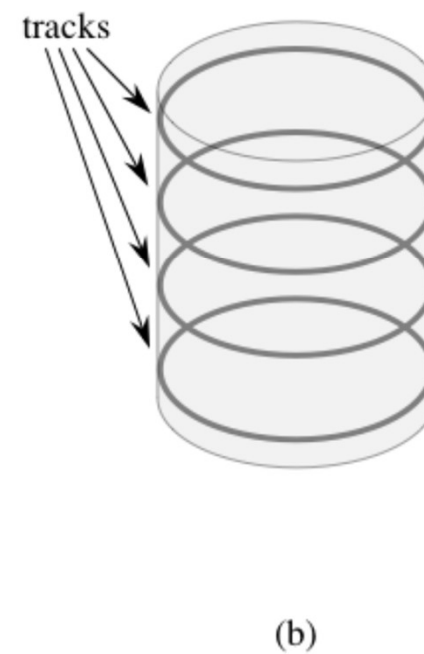
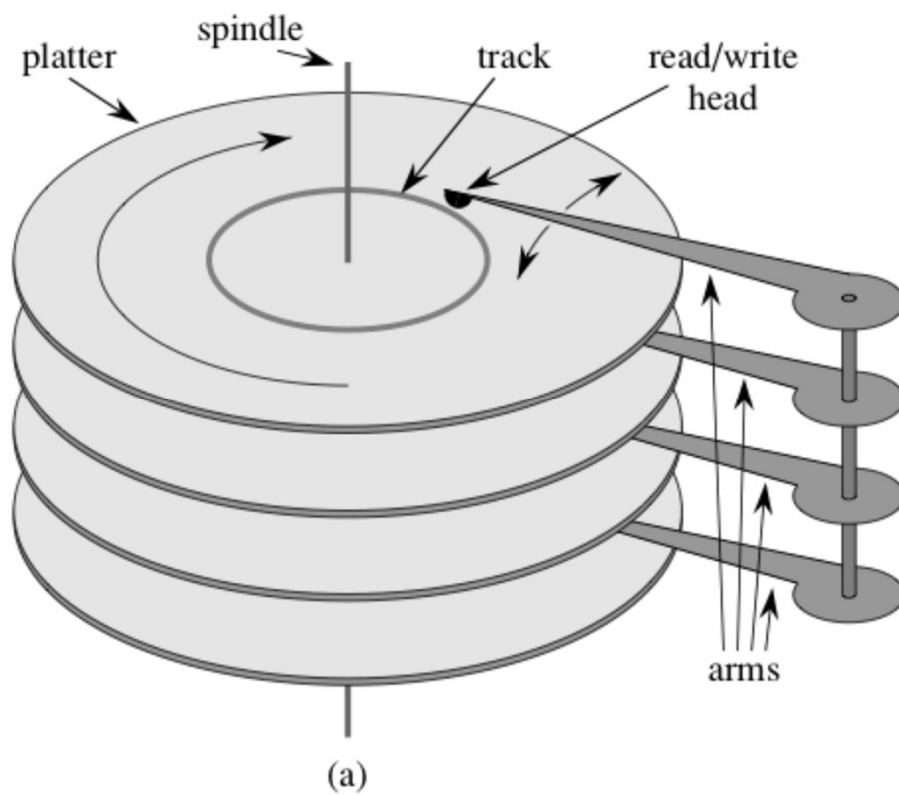
## 5.3.1 Organización física

- Cada plato puede ser leído o escrito por una cabeza (head) que se encuentra al final de un brazo (arm)
- Cada brazo se encuentra físicamente acoplado y en conjunto pueden acercar o alejar las cabezas del eje.
- Cuando una determinada cabeza esta quieta, la superficie que se encuentra debajo se llama pista (track)

## 5.3.1 Organización física

- Las cabezas de lectura/escritura se encuentran alineadas verticalmente en todo momento, por lo tanto el conjunto de pistas bajo cada una se pueden utilizar simultáneamente.
- Este conjunto se conoce como cilindro.

## 5.3.1 Organización física



## 5.3.1 Organización física

- Los dispositivos físicos como los discos son más lentos en su acceso debido a que cuentan con partes móviles.
- En el caso de los discos existen dos componentes de movimiento mecánico: la rotación de los platos y el movimiento de los brazos.

## 5.3.1 Organización física

- Con el fin de disminuir el tiempo de espera por los movimientos mecánicos la lectura no es de un solo elemento, sino de varios al mismo tiempo.
- La información se divide en una cantidad de páginas o bloques de la misma cantidad de bits que aparecen consecutivamente en los cilindros.
- Cada lectura o escritura al disco es sobre una o más páginas

## 5.3.1 Organización física

- Una vez que la cabeza(head, cabezal) se encuentra en la posición correcta y el disco ha rotado al inicio de la pagina deseada, la lectura o escritura de grandes cantidades de datos puede hacerse de forma rápida.
- Es común que tome más tiempo llegar a la pagina de información deseada que el tiempo que le toma a la computadora examinar la información recuperada.



## 5.3.2 Organización Lógica

- Considere las siguientes equivalencias para el correcto manejo de la memoria
  - ✓ 1 byte = 8 bits
  - ✓ 1 kilobyte = 1024 bytes
  - ✓ 1 megabyte = 1'048'576 ( $2^{20}$ ) bytes
  - ✓ 128 megabytes
  - ✓ 256 megabytes
  - ✓ 512 megabytes
  - ✓ 1 gigabyte = 1024 megabytes = ( $2^{30}$ ) bytes

# Ejemplo

- Se tiene una memoria de 1gb.
- Se requiere almacenar los datos de una guía telefónica de 1,200,000 de entradas o registros.
- Cada registro incluye nombre, dirección, teléfono y algunos otros datos de interés.
- Supóngase que cada registro requiere 512 bytes.
- El tamaño completo del archivo sería  $1,200,000 \times 512 = 614,400,000$  bytes (aprox 580 Mb)

# Ejemplo

**¿De qué forma se puede estructurar un archivo para realizar las operaciones sobre el disco de manera rápida?**

- En memoria principal, cualquier byte puede procesarse en velocidades aproximadas a una fracción de microsegundo.
- Una operación similar en un disco requiere mover la cabeza a la pista correcta que incluye movimientos mecánicos.

# Ejemplo

- Cada uno de estos movimientos requiere alrededor de 10 milisegundos. Esto es 10,000 veces más lento que el acceso a memoria principal.
- SOLUCIÓN: ESCRITURA POR BLOQUES

Para minimizar los movimientos mecánicos, siempre se lee o escribe al menos un bloque completo cada vez.

# Ejemplo

El tamaño de los bloques normalmente se define por un número en el que la memoria sea divisible.

Independientemente del tamaño de la memoria, se puede dividir de varias formas siempre considerando potencias de dos.

- Se pueden considerar bloques de tamaños estándar

$2^{10}$  1024 bytes

$2^{11}$  2048 bytes

$2^{12}$  4096 bytes

$2^{13}$  8192 bytes

# Ejemplo

Volviendo a nuestro ejemplo...

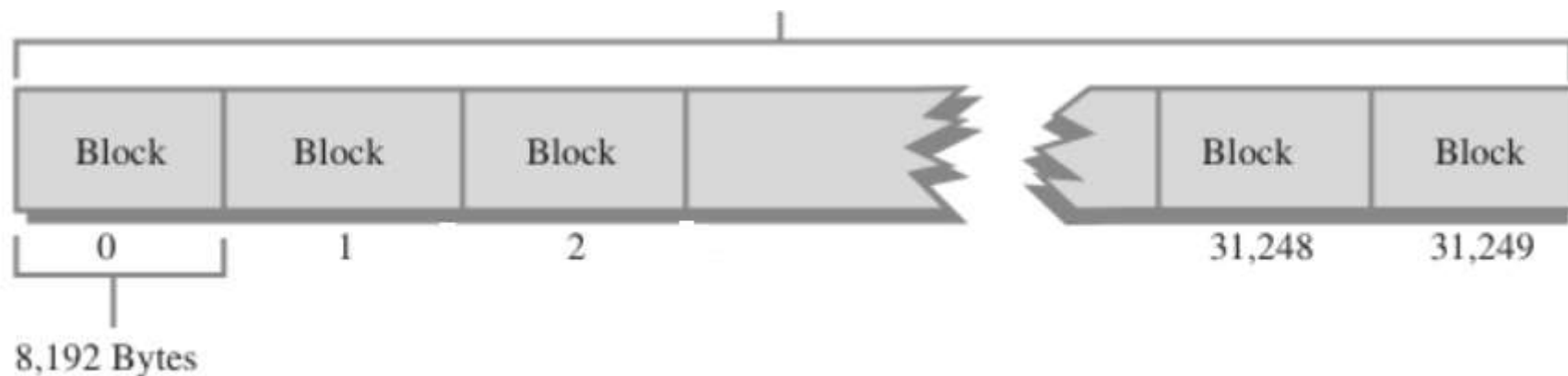
- Si se consideran bloques de 8192 BYTES ( $2^{13}$ )
- Para obtener el número total de bloques. Se divide el tamaño de la memoria entre el tamaño de bloque.

131'072 bloques totales disponibles.

# Ejemplo

- Para obtener la cantidad de registros por bloque se divide el tamaño de bloque entre el tamaño del registro
- $8192 / 512$

En cada bloque se pueden almacenar 16 registros



# Ejemplo

- Para obtener la cantidad de bloques necesarios, se dividen los registros totales requeridos entre los que caben por bloque.
- Se necesitan 75'000 bloques para almacenar el archivo



# Ejercicio

Suponer que se tiene un archivo de 725'000 registros de 256 bytes c/u

- Indicar el tamaño mínimo de memoria que se requiere para almacenar el archivo (en Mb)
- Para los siguientes tres posibles tamaños de bloque:
  - a)  $2^{11}$
  - b)  $2^{12}$
  - c)  $2^{13}$
- El número de bloques disponibles para el tamaño de memoria indicado
- La cantidad de registros que se puede almacenar por bloque.
- El número de bloques necesarios para el archivo

# Solución

- Tamaño de memoria

$$\frac{725'000 \times 256}{1mb} = \frac{185'600'000}{1'048'576} = 177.8 \rightarrow 256 \text{ mb}$$

# Solución

a) Bloques de  $2^{11}$  bytes

$$\# \text{ Bloques disp} = \frac{\text{memoria}}{\text{bloque}} = \frac{2^{28}}{2^{11}} = 2^{17} = 131'072$$

$$\# \text{ Reg x bloque} = \frac{\text{bloque}}{\text{registro}} = \frac{2048}{256} = 8$$

$$\# \text{ Bloques req.} = \frac{\text{reg archivo}}{\text{reg x bloque}} = \frac{725000}{8} = 90625$$

# Solución

b) Bloques de  $2^{12}$  bytes

$$\# \text{ Bloques disp} = \frac{\text{memoria}}{\text{bloque}} = \frac{2^{28}}{2^{12}} = 2^{16} = 65536$$

$$\# \text{ Reg x bloque} = \frac{\text{bloque}}{\text{registro}} = \frac{4096}{256} = 16$$

$$\# \text{ Bloques req.} = \frac{\text{reg archivo}}{\text{reg x bloque}} = \frac{725000}{16} = 45313$$

# Solución

a) Bloques de  $2^{13}$  bytes

$$\# \text{ Bloques disp} = \frac{\text{memoria}}{\text{bloque}} = \frac{2^{28}}{2^{13}} = 2^{15} = 32'728$$

$$\# \text{ Reg x bloque} = \frac{\text{bloque}}{\text{registro}} = \frac{8192}{256} = 32$$

$$\# \text{ Bloques req.} = \frac{\text{reg archivo}}{\text{reg x bloque}} = \frac{725000}{32} = 22657$$

# Implementación de archivos.

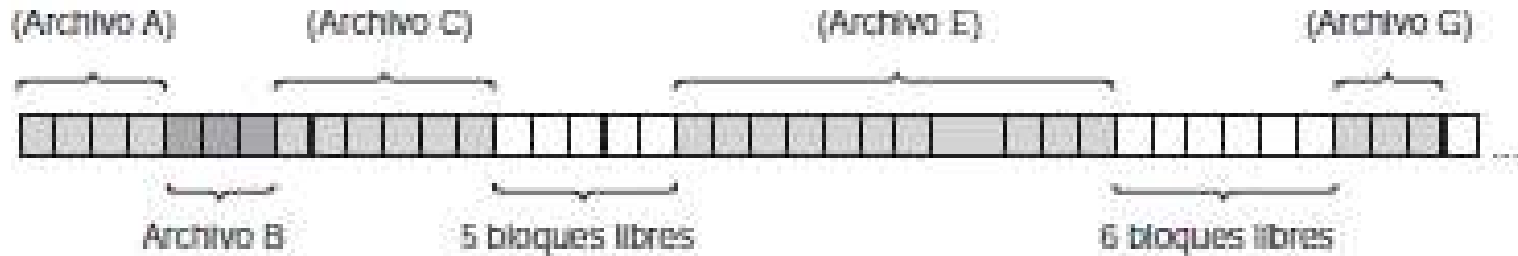
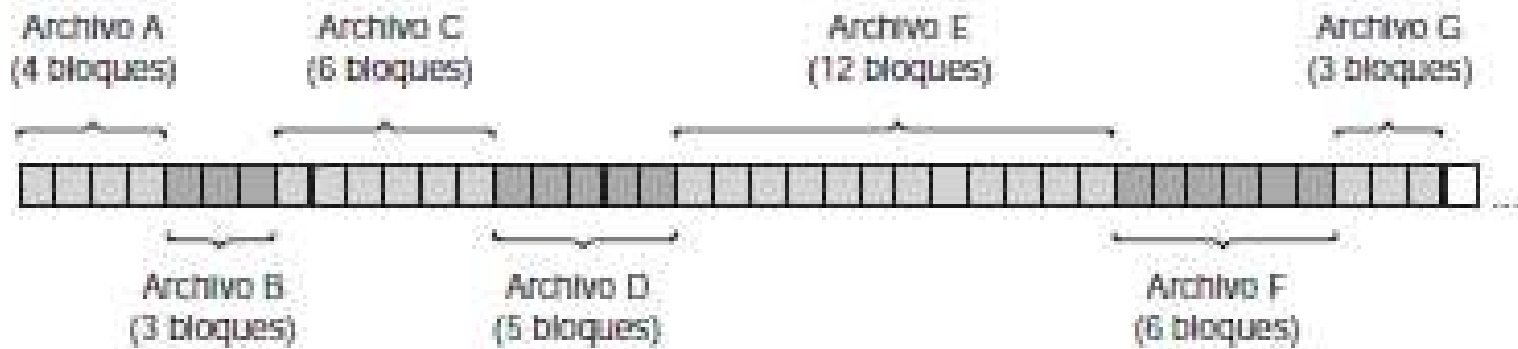
Probablemente la cuestión más importante al implementar el almacenamiento de archivos sea mantener un registro acerca de que bloques de disco se asocian con cual archivo.

- Asignación Contigua.
- Asignación de Lista Ligada

# Asignación Contigua

- Es el esquema más simple, por ejemplo en un disco con bloques de 1mb, a un archivo de 50mb se le asignarán 50 bloques consecutivos. Con bloques de 2mb, se le asignarían 25 bloques consecutivos.
- Ventajas.
  - Simple de implementar
  - Rendimiento de lectura
- Desventaja
  - Fragmentación del disco

# Asignación Contigua

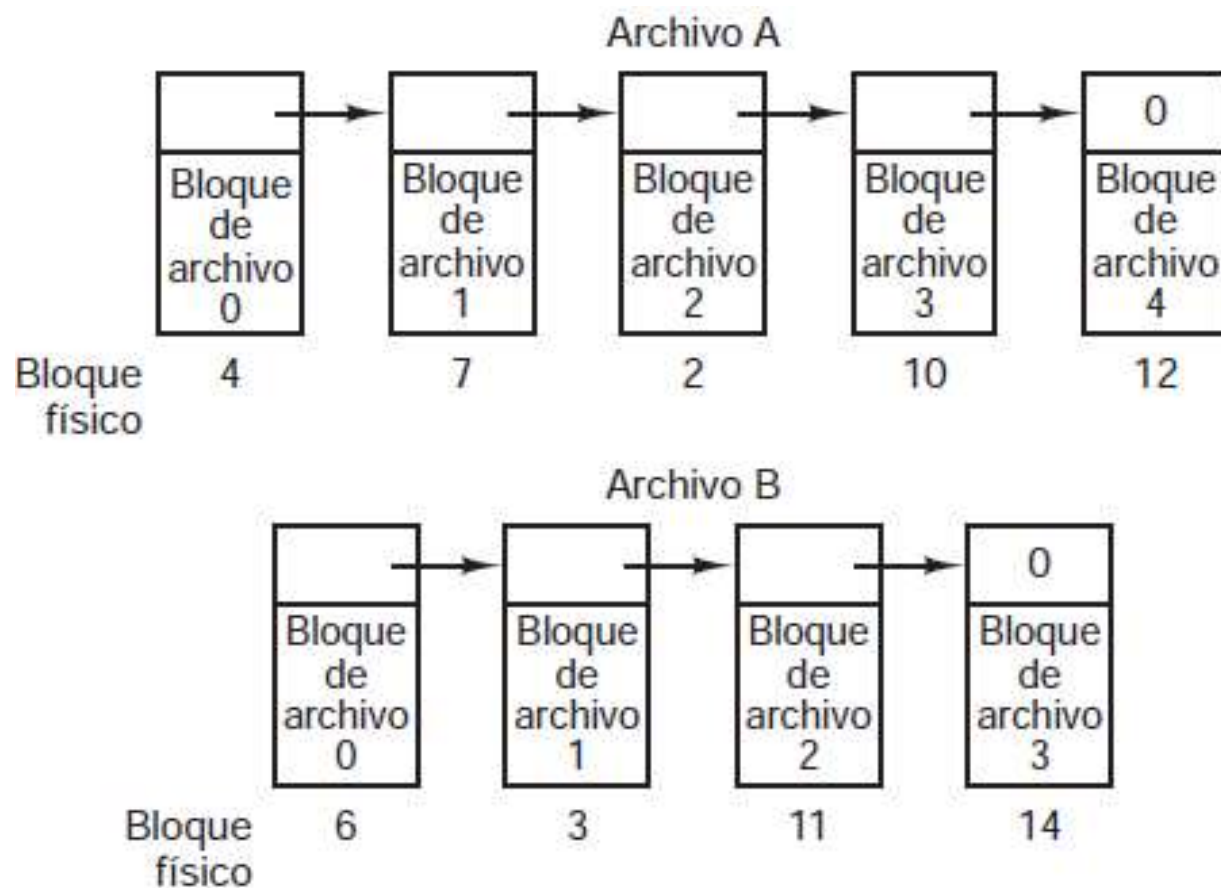




# Asignación de Lista Ligada

- Se trata de mantener cada archivo como una lista ligada de bloques de disco
- Prácticamente no se pierde espacio en disco.
- Para la entrada del directorio, solo le basta con almacenar la dirección de disco del primer bloque.

# Asignación de Lista Ligada



# Tabla de Asignación de archivos

- En esta implementación, se coloca cada dirección de bloque en una tabla en memoria.
- Utilizando esta organización, el bloque completo está disponible para los datos.
- File Allocation Table (FAT)

Bloque físico		
0		
1		
2	10	
3	11	
4	7	← El archivo A empieza aquí
5		
6	3	← El archivo B empieza aquí
7	2	
8		
9		
10	12	
11	14	
12	-1	
13		
14	-1	
15		← Bloque sin utilizar

## 5.4 Acceso a archivos

- Existen varias formas para acceder a la información almacenada en memoria secundaria. Algunas de las más utilizadas son:
  - Secuencial
  - Árboles B
  - Utilizando índices

## 5.4.1 Acceso Secuencial

- El acceso secuencial es una forma natural de organizar la información, se gestiona conforme se almacena.
- A esta información se le puede aplicar algún procedimiento de ordenamiento para mejorar la eficiencia en el acceso a la misma

## 5.4.1.2 Árboles B

- Como se menciona, el acceso al disco es más eficiente cuando se lee o se escribe un bloque completo cada vez
- En un árbol B la entidad que contiene los datos es un nodo, por lo tanto es consistente almacenar un bloque entero en cada nodo del árbol , de esta forma la lectura de un nodo accede a la máxima cantidad de datos posibles en el menor tiempo.

## 5.4.1.2 Árboles B

- La organización de archivos utilizando árboles B. proporciona una mayor eficiencia en cuanto a acceso al disco por la estructura en sí de un árbol.
- El manejo de apuntadores a los otros bloques de información (nodos) debe realizarse de forma adecuada (ordenada) para tener un control exacto de la posición de cada bloque

### 5.4.1.3 Indexación

- Un enfoque diferente para acelerar los accesos a archivos es almacenar los registros de forma secuencial pero utilizar un archivo de índices junto con los datos mismos.
- Un archivo de índices es una lista de parejas (llave, bloque) ordenada de acuerdo a la llave.



### 5.4.1.3 Indexación

- El archivo de índices es mucho más pequeño que el archivo que contiene los registros completos ya que solo se emplea uno de los datos del archivo original.
- Puede estar almacenado en disco, pero se lee en memoria al iniciar el programa, entonces las operaciones sobre el índice pueden llevarse a cabo en memoria y cada determinado tiempo volver a escribir ese índice en el disco.

## 5.5 Sistema de archivos

- El sistema de archivos (filesystem) es la forma en la que se organizan los datos en un dispositivo físico; provee procedimientos para almacenar, extraer y actualizar los datos además de administrar el espacio libre en el mismo dispositivo.
- Sin sistema de archivos, los programas no podrían acceder a los datos por nombre de archivo o directorio y necesitarían tener acceso directo a memoria.

## 5.5.1 Características principales

- **Administración de espacio disponible**

Los sistemas de archivos asignan espacio de forma granular, usualmente múltiples unidades físicas en el dispositivo.

El sistema de archivos es responsable de organizar archivos y directorios, además de mantener un registro del área en la que se encuentra algún archivo específico y las áreas no utilizadas.

## 5.5.1 Características principales

- **Nombre de archivo**

Se utiliza para hacer la referencia a la ubicación en la que se encuentran almacenados los datos.

Cada sistema de archivos tiene diferentes restricciones sobre los nombres de los archivos, principalmente sobre la longitud permitida

## 5.5.1 Características principales

- **Carpetas**

Los sistemas de archivos actuales se organizan por carpetas o directorios, que permiten agrupar archivos y otros directorios, de esta forma se obtiene una estructura jerárquica de tipo árbol

## 5.5.1 Características principales

- **Restricciones sobre el acceso**

Existen diversos mecanismos para limitar el acceso a los datos, generalmente es deseable prevenir que ciertos usuarios puedan leer o modificar el contenido de algunos archivos.

Se pueden modificar los permisos para cada archivo, establecer una contraseña para ciertas acciones, listas de control de acceso, etc.

## 5.5.1 Características principales

- **Aseguramiento de la integridad**

Una de las responsabilidades primordiales de un sistema de archivos es asegurar que la estructura permanezca consistente, sin importar las operaciones realizadas por algún programa.

Esto incluye las acciones que debe tomar cuando un programa que modifica algún archivo termina de forma anormal o que no informa que ha completado la operación sobre el archivo.

## 5.5.1 Características principales

- **Uso del sistema de archivos**

Los lenguajes y programas utilizan alguna API (application programming interface) del sistema de archivos para hacer solicitudes de transferencia, actualización, acceso a directorios, eliminación etc.



## 5.5.2 Tipos de sistemas de archivos

- Sistemas de archivos de disco y cinta
  - Sistemas de archivos de discos magnéticos
  - Discos ópticos
  - Sistemas de archivos flash
  - Sistemas de archivos de cinta
- Sistemas de archivos de red
  - Sistemas de archivos de unidades compartidas

## 5.5.2 Tipos de sistemas de archivos

- Sistemas de archivos especiales
  - Sistemas de archivos de bases de datos
  - Sistemas de archivos de Transacciones

## 5.5.3 Sistemas de archivos y sistemas operativos

- La mayoría de los sistemas operativos soportan uno o más sistemas de archivos.
- Algunas veces las operaciones del sistema operativo y del sistema de archivos están tan entrelazadas que resulta muy difícil separarlos.

## 5.5.3 Sistemas de archivos y sistemas operativos

- UNIX

Los sistemas operativos tipo UNIX crean un sistema de archivos virtual, que produce que todos los archivos en todos los dispositivos parezcan existir en una jerarquía simple, a partir del directorio raíz “/”

- MAC OS X

Utiliza un sistema de archivos llamado HFS Plus, también conocido como MacOS Extended. Contiene metadatos y hace diferencia entre mayúsculas y minúsculas