

The background is a deep blue gradient. On the left side, there is a stylized representation of a globe with white puzzle pieces. A black puzzle piece is being placed into a gap in the white pieces. Wavy, light blue lines resembling data or network connections are scattered across the scene. The overall aesthetic is technological and modern.

Tema V

Sistema de Archivos

Problemas al almacenar datos sólo en la memoria

- ◆ El tamaño está limitado por el tamaño de la memoria ¿qué pasa con aplicaciones que necesiten acceder a gran cantidad de información?
- ◆ Cuando un proceso termina, la información de la memoria se pierde ¿cómo hacer para conservar la información por más tiempo?
- ◆ A veces es necesario que múltiples procesos tengan acceso a la información al mismo tiempo ¿cómo hacer para acceder de forma concurrente?

Lo importante es...

- ◆ ¿de qué se constituye un archivo?
- ◆ ¿cómo se nombran y se identifican?
- ◆ ¿cómo se protegen?
- ◆ ¿qué operaciones se pueden ejecutar con archivos?

Desde el punto de vista del sistema operativo

El sistema operativo administra los archivos. La forma como se estructuran, se nombran, se tiene acceso a ellos, se usan, se protegen y se implementan son temas del diseño del sistema operativo.

Esta parte del sistema operativo se denomina:

“Sistema de Archivos”

Objetivos de un Sistema de Archivos

- ◆ Cumplir con las necesidades de almacenamiento y de las operaciones de usuario.
- ◆ Garantizar la validez de los datos de los archivos.
- ◆ Optimizar el rendimiento.
- ◆ Ofrecer soporte de I/O para la variedad de tipos de dispositivos de almacenamiento.
- ◆ Evitar la pérdida de datos.
- ◆ Ofrecer un conjunto estándar de rutinas de interfaz de I/O.

Requisitos mínimos de usuario

- ◆ Capacidad para crear, borrar y modificar los archivos.
- ◆ Acceso controlado a los archivos de otros usuarios.
- ◆ Control sobre los tipos de acceso permitidos que tendrán sus archivos.
- ◆ Capacidad para reestructurar sus archivos según su necesidad.
- ◆ Capacidad para mover datos entre los archivos.
- ◆ Capacidad para hacer respaldos y recuperar datos de los respaldos.
- ◆ Acceso a los archivos mediante nombres simbólicos.

Concepto

- ◆ **Archivo:** Unidad de almacenamiento lógico no volátil que agrupa un conjunto de información relacionada entre sí bajo un mismo nombre.
- ◆ La mayoría de los sistemas operativos reconocen nombres de archivo de dos partes, las cuales se separan con un punto. *Nombre.Extensión.*
- ◆ Algunos atributos de los archivos son: nombre, tipo de archivo, ubicación, tamaño, protección, hora, fecha e identificación del usuario.
- ◆ Tipos de archivos: binarios, texto, imágenes, video, sonidos, etc.

Operaciones con archivos

- ◆ Crear un archivo (*Create*).
- ◆ Eliminar un archivo (*Delete*).
- ◆ Escritura de un archivo (*Write*).
- ◆ Leer un archivo (*Read*).
- ◆ Abrir un archivo (*Open*).
- ◆ Añadir nueva información al final de un archivo ya existente (*Append*).
- ◆ Reubicación dentro de un archivo (*Seek*).
- ◆ Cerrar un archivo (*Close*).
- ◆ Otros :
 - *Get attributes*
 - *Set attributes*
 - *Rename*

Derechos de acceso

- ◆ Ninguno
- ◆ Conocimiento
- ◆ Ejecución
- ◆ Lectura
- ◆ Adición
- ◆ Actualización*
- ◆ Cambio de protección
- ◆ Borrado

* modificar, borrar y añadir datos del archivo

Para:

- ❑ *Usuario específico*
- ❑ *Grupo de usuarios*
- ❑ *Todos*

Métodos de Acceso

Acceso Secuencial:

- ◆ La información es accedida en orden, registro a registro.
- ◆ El registro depende del tipo de archivo (texto plano sería *byte*).
- ◆ Las operaciones de lectura accederán a la información en forma secuencial incrementando el apuntador del archivo *file pointer*.
- ◆ El acceso secuencial es basado en el modelo de cinta (*tape*), en donde los archivos son accedidos de uno a la vez y en forma secuencial.

Desventaja: Problema de inserción de un bloque.

Ventaja: Aprovechamiento del espacio cuando no se necesita hacer modificaciones.

Primitivas: *write(datos)*, *read(datos)*, *rewind*.

Métodos de Acceso

Acceso Directo:

- ◆ La información es accedida en cualquier orden.
- ◆ No existe restricciones sobre el orden de escritura y lectura de un archivo. Es basado en el modelo de disco de un archivo que está estructurado en bloques.

Primitivas:

write(posición, datos)

read(posición, datos)

seek(posición)

Organización de Archivos

*Tema a investigar para evaluar
(no se debe entregar algo)*

- ◆ Pila
- ◆ Archivo secuencial
- ◆ Archivo secuencial indexado
- ◆ Archivo indexado

The background is a deep blue gradient. On the left side, there is a stylized, semi-transparent globe with a wireframe structure. A 3D puzzle piece, which is dark blue and has a complex, interlocking shape, is positioned in the center-left, appearing to be part of the globe's structure. The puzzle piece has a small, lighter blue rectangular area on its top surface. The overall aesthetic is modern and technological.

Estructura de los directorios

Directorio

Estructura que permite organizar y acceder a los archivos.

- ◆ Deben de ser no volátiles.
- ◆ Deben traerse a memoria cuando se necesitan.



Operaciones sobre Directorios

- ◆ Crear un directorio (*Create*).
- ◆ Eliminar el directorio vacío (*Delete*).
- ◆ Leer el directorio (*Opendir*).
- ◆ Cerrar el directorio liberando espacio (*Closedir*).
- ◆ Devuelve la siguiente entrada de un directorio abierto (*Readdir*).
- ◆ Cambiar el nombre del directorio (*Rename*).
- ◆ Permite a un archivo aparecer en más de un directorio (*Link*).
- ◆ Elimina una entrada del directorio (*Unlink*).

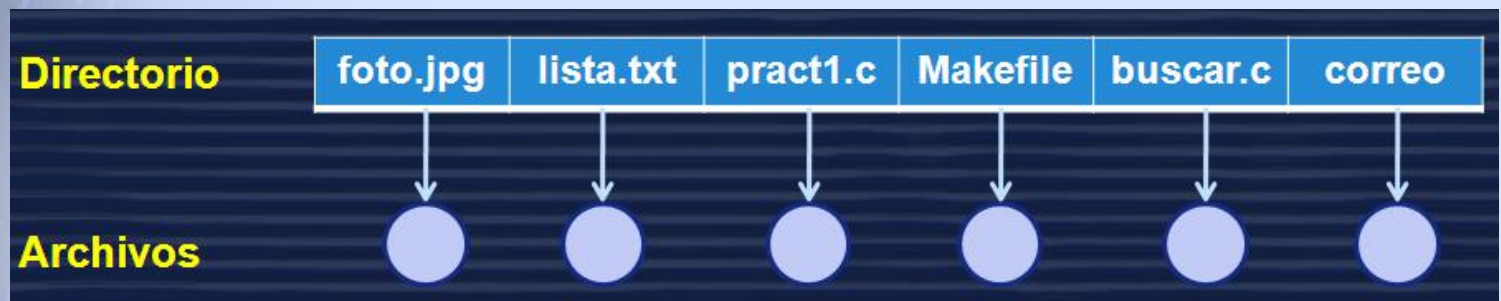
Tipos de organización

Directorio de un solo nivel:

- ◆ Todos los archivos se guardan en un solo directorio.
- ◆ Estructura más simple y más fácil de soportar.

Características:

- ✦ Tiene limitaciones importantes cuando el número de archivos aumenta o si hay más de un usuario.
- ✦ No permite clasificar la información de ninguna manera.
- ✦ Como todos los archivos están en el mismo directorio, deben tener nombres únicos.
- ✦ Problema: En sistemas multiusuario hay problemas para nombrar los archivos, es decir, puede haber confusión de nombres de archivo entre diferentes usuarios.



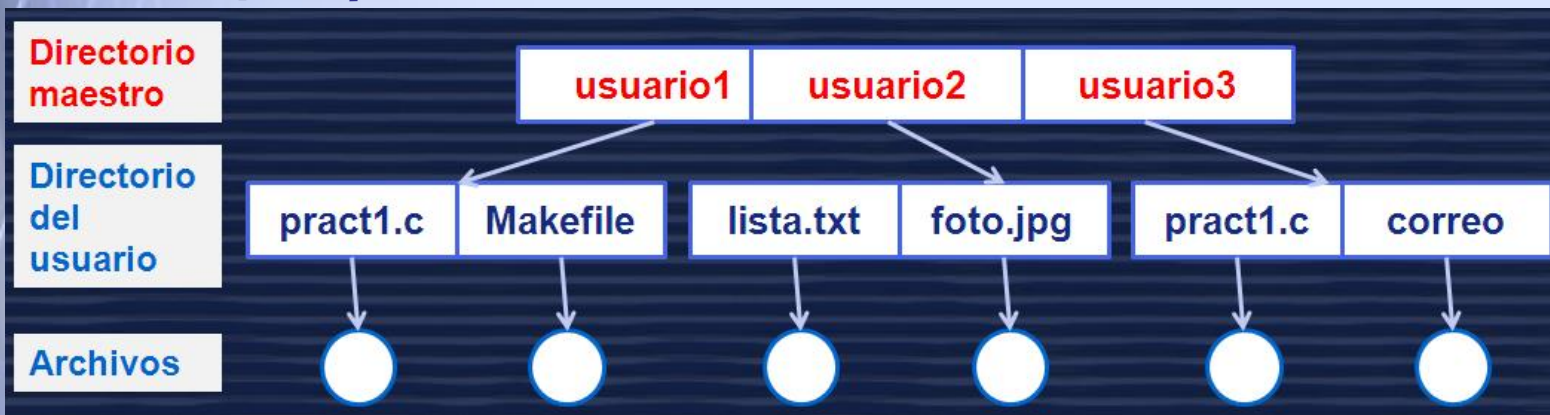
Tipos de organización

Directorio de dos niveles:

- ◆ Cada usuario tiene su propio directorio de archivos de usuario (*UFD, User File Directory*).

Características:

- ✦ Se asigna un directorio a cada usuario o tipo de archivo.
- ✦ Existe un directorio maestro o padre (*MFD, Master File Directory*).
- ✦ Las operaciones de un usuario sobre los archivos están restringidas a su directorio.
- ✦ Existen operaciones de actualización del directorio maestro.
- ✦ Fácil administración en cuanto a la protección.
- ✦ Concepto: *pathname*.



Tipos de organización

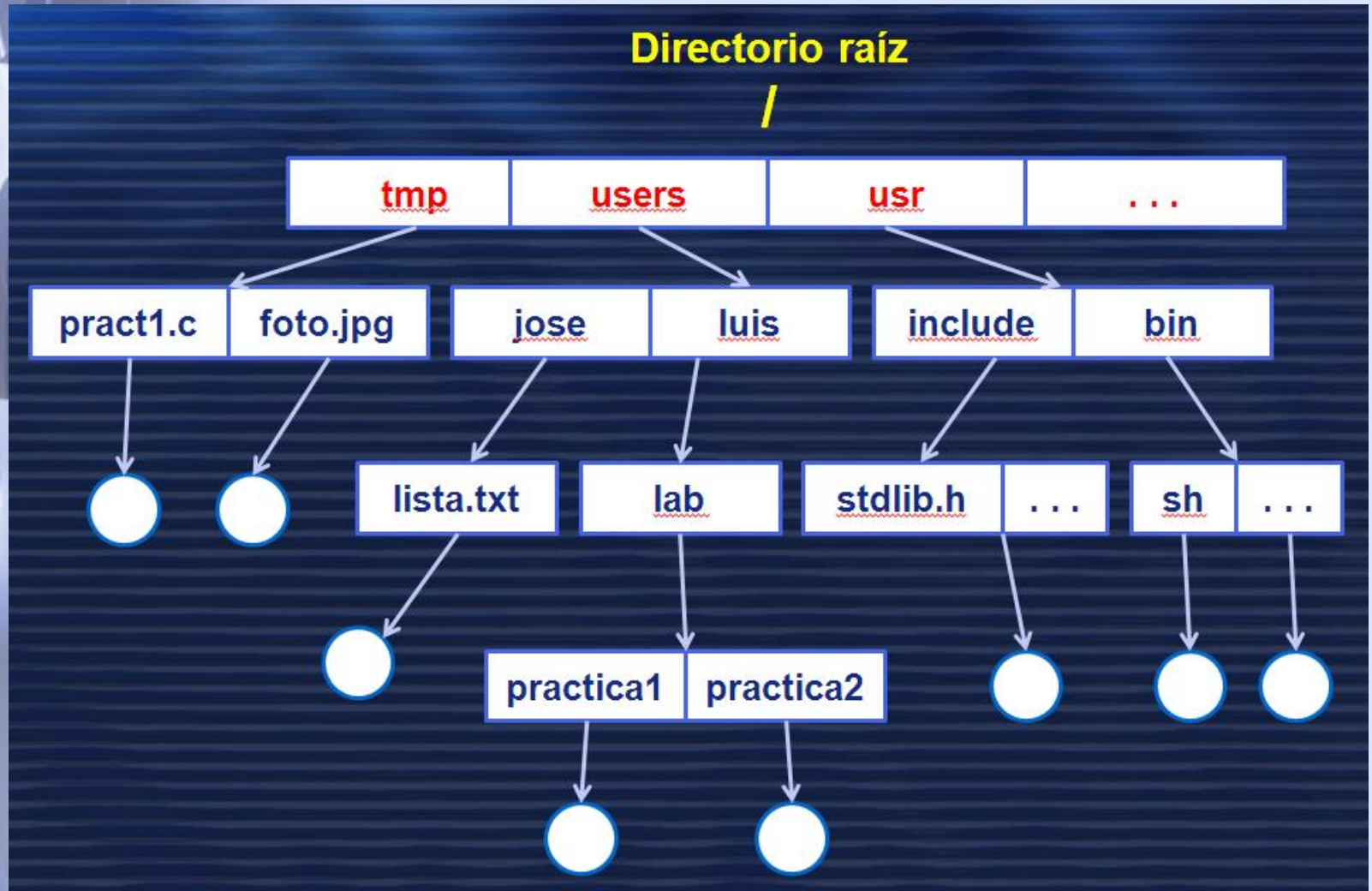
Estructura Jerárquica de Árbol:

- ◆ Generalización del directorio de dos niveles ⇒ estructura de directorios, subdirectorios, archivos, ...

Características:

- ⊕ Directorio raíz del que parten todos los directorios (y archivos).
- ⊕ Simplicidad en la administración agrupando en un mismo directorio todos los archivos relacionados.
- ⊕ Conceptos: directorio de trabajo (*home directory*).
- ⊕ Los nombres de los archivos se pueden especificar mediante rutas relativas y absolutas.
- ⊕ El identificador de archivos distingue entre archivos ordinarios y directorios.

Tipos de organización



The background is a deep blue gradient. On the left side, there is a stylized globe with a grid of latitude and longitude lines. Several 3D puzzle pieces are scattered around the globe, some of which are interlocked. The puzzle pieces are in shades of blue and white. The overall composition suggests a complex system or structure being assembled or analyzed.

Estructura de los Sistemas de Archivos

The background is a deep blue gradient. On the left side, there are several white puzzle pieces, some of which are interlocked. Wavy, white, smoke-like or liquid-like lines flow upwards from the bottom left towards the puzzle pieces. The overall composition suggests a process of assembly or implementation.

Implementación

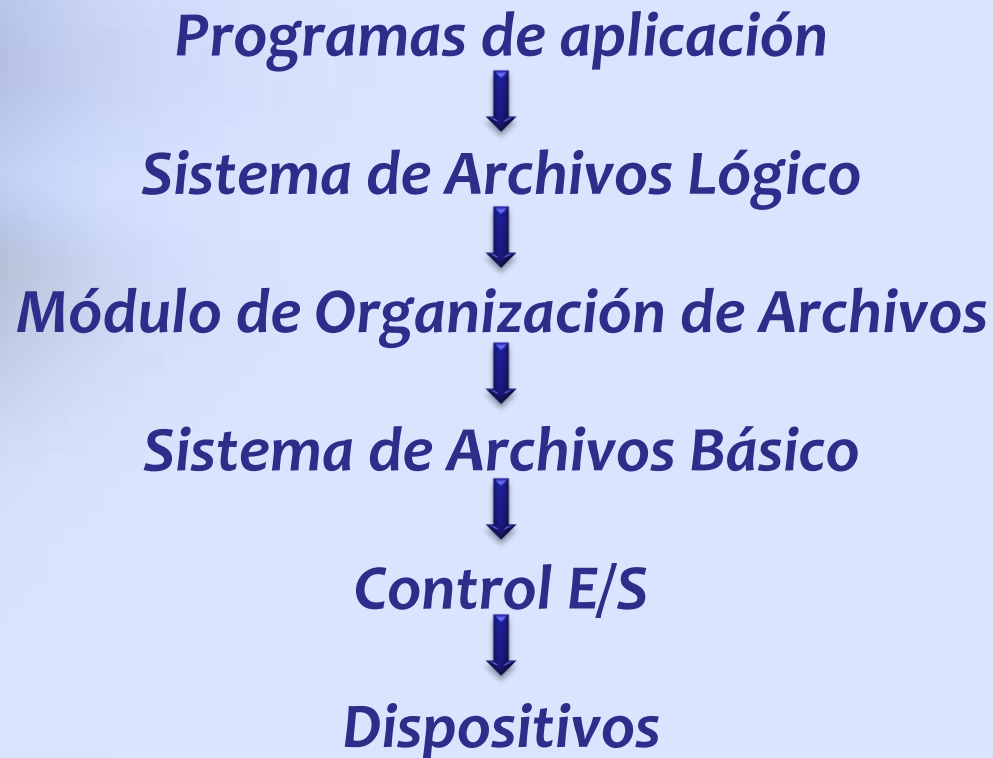
Implementación

El diseño de un *Sistema de Archivos* enfrenta dos problemas:

- ◆ *Como se verá para el usuario.* Implica definir los atributos, las operaciones válidas sobre los archivos y la estructura de directorios para la organización de los archivos.
- ◆ *La creación de algoritmos y estructura de datos* para hacer corresponder el sistema de archivos lógico con los dispositivos físicos de datos.

Implementación

El *Sistema de Archivos* está compuesto de varias capas. Cada una de las cuales utiliza la funcionalidad de la capa inferior.



Implementación

Los dispositivos físicos (discos) contienen la siguiente estructura:

- ◆ **Bloque de control para el boot (*Boot Control Block*):** es necesario para lograr iniciar el sistema operativo.
- ◆ **Bloque de control de partición (*Partition Control Block*):** contiene la información de las particiones que existen en el disco, bloques utilizados y libres, cantidad de archivos, etc.
- ◆ **Estructura de directorios:** para la organización de los archivos.
- ◆ **Bloque de control del archivo (*File Control Block*):** los bloques de control de los archivos que están en el sistema de archivos.

Implementación

El sistema operativo en memoria mantiene las siguientes estructuras:

- ◆ La tabla de partición con los sistemas de archivos cargados.
- ◆ La estructura de directorio de los accedidos últimamente.
- ◆ Tabla de descriptores de archivos abiertos a nivel global del sistema.
- ◆ Tabla de descriptores de archivos abiertos por proceso del sistema.

Implementación

- ◆ Por cada archivo en el sistema se tendrá un **Bloque de Control** (*File Control Block*).
- ◆ El **Bloque de Control** contiene varios atributos de conteo, permisos y donde están los datos del mismo:
 - Permisos del archivo.
 - Fechas (creación, acceso, modificación).
 - Propietario, grupo de propietario, lista de acceso.
 - Tamaño del archivo.
 - Bloques de datos del archivo.

The background is a deep blue gradient. On the left side, there is a stylized globe with several white puzzle pieces attached to its surface. Some puzzle pieces are in the foreground, appearing to be in the process of being placed or removed. The right side of the image is mostly a smooth blue gradient with some subtle light effects.

Métodos de Asignación

Asignación Contigua (Contiguous Allocation)

- ◆ Los datos son dispuestos en forma contigua.
- ◆ Para mantener la información es necesario saber en que bloque comienza y la cantidad de bloques que tiene el archivo
- ◆ Sufre de fragmentación externa.
- ◆ Es necesario reubicar continuamente los archivos si crecen en tamaño.
- ◆ Se utilizan técnicas de asignación de tamaños más grandes para prever el crecimiento futuro de los archivos.

Asignación Contigua

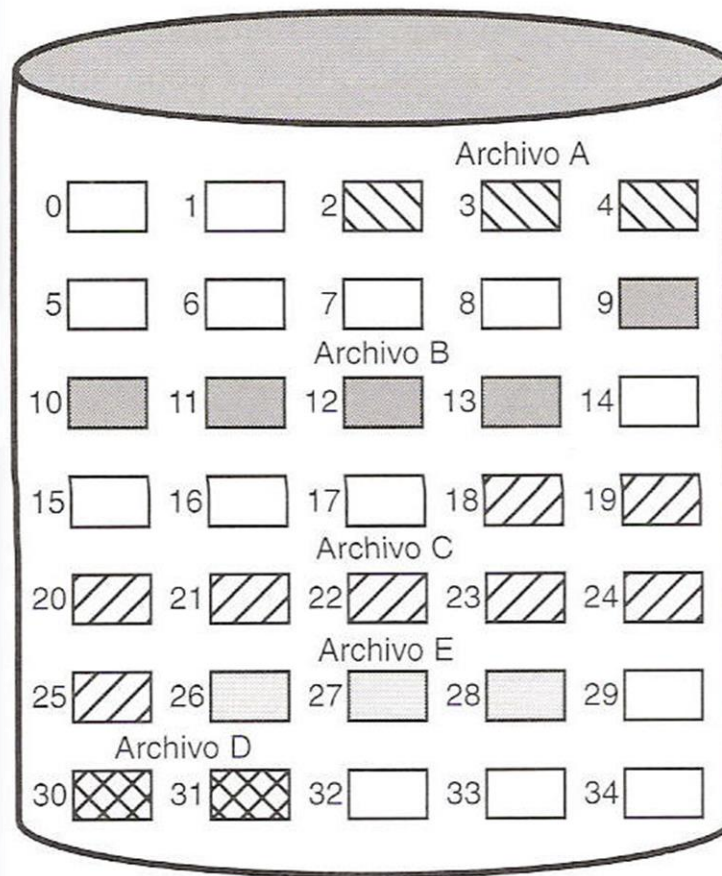


Tabla de asignación de archivos

Nombre de archivo	Bloque de inicio	Longitud
Archivo A	2	3
Archivo B	9	5
Archivo C	18	8
Archivo D	30	2
Archivo E	26	3

Asignación Contigua (Compactación)

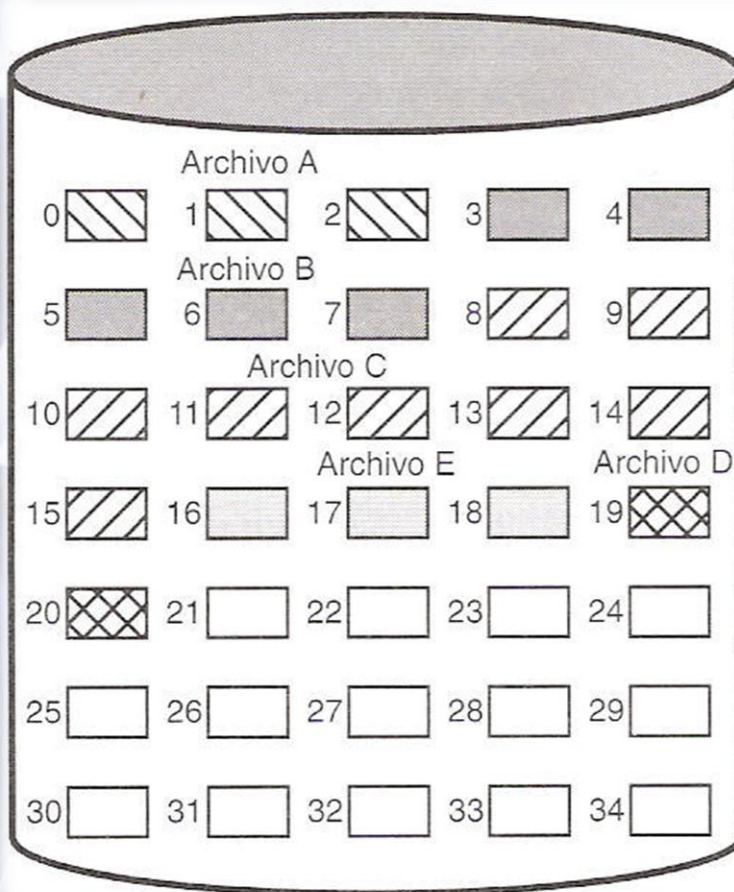


Tabla de asignación de archivos

Nombre de archivo	Bloque de inicio	Longitud
Archivo A	0	3
Archivo B	3	5
Archivo C	8	8
Archivo D	19	2
Archivo E	16	3

Asignación Enlazada (Linked Allocation)

- ◆ Los bloques de datos forman una lista encadenada.
- ◆ Es necesario una referencia al primer y último bloque de datos en el bloque de control de archivo.
- ◆ Soluciona el problema de la fragmentación.
- ◆ El acceso a los bloques es lineal.
- ◆ Los apuntadores ocupan espacio en los bloques.
- ◆ La pérdida de una referencia genera la pérdida de gran parte de información del archivo.

Asignación Enlazada

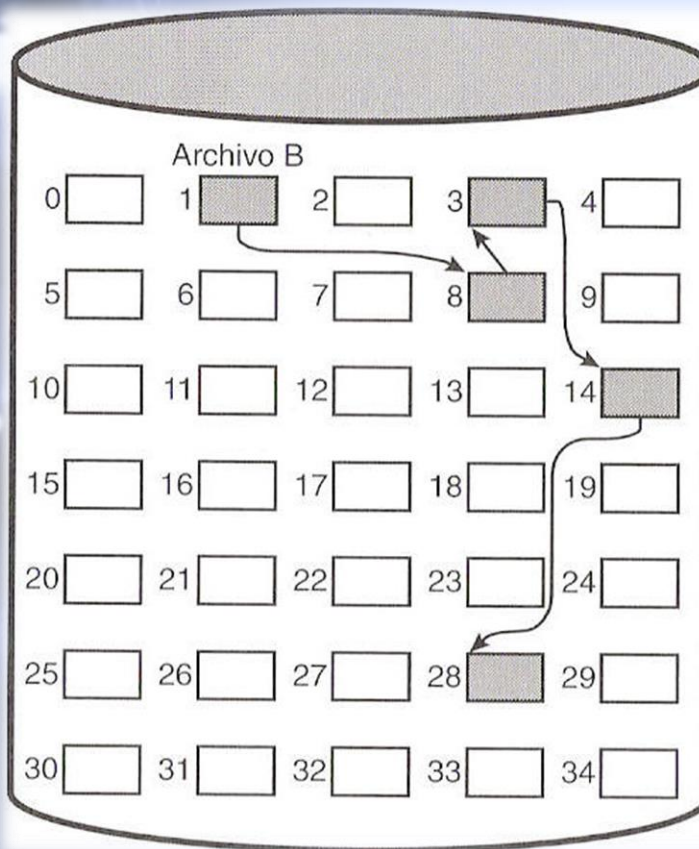


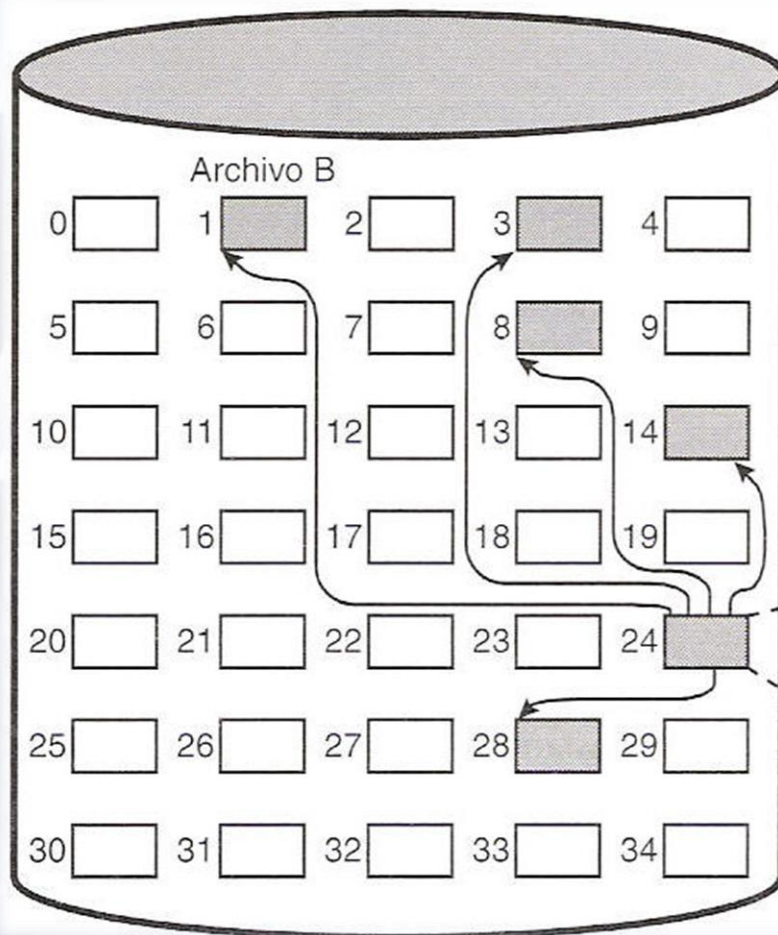
Tabla de asignación de archivos

Nombre de archivo	Bloque de inicio	Longitud
...
Archivo B	1	5
...

Asignación Indexada (Indexed Allocation)

- ◆ Se mantiene una tabla en donde cada entrada referencia a un bloque de datos.
- ◆ Los bloques son accedidos directamente a través del bloque de indexación (*index block*).
- ◆ El bloque de indexación ocupa lugar. Se trata de que sea lo más pequeño posible, pero limita la cantidad de bloques.
- ◆ Una posible alternativa es indexación en varios niveles. Algunos índices hacen referencia a bloques directos y otros a bloques de indexación.

Asignación Indexada



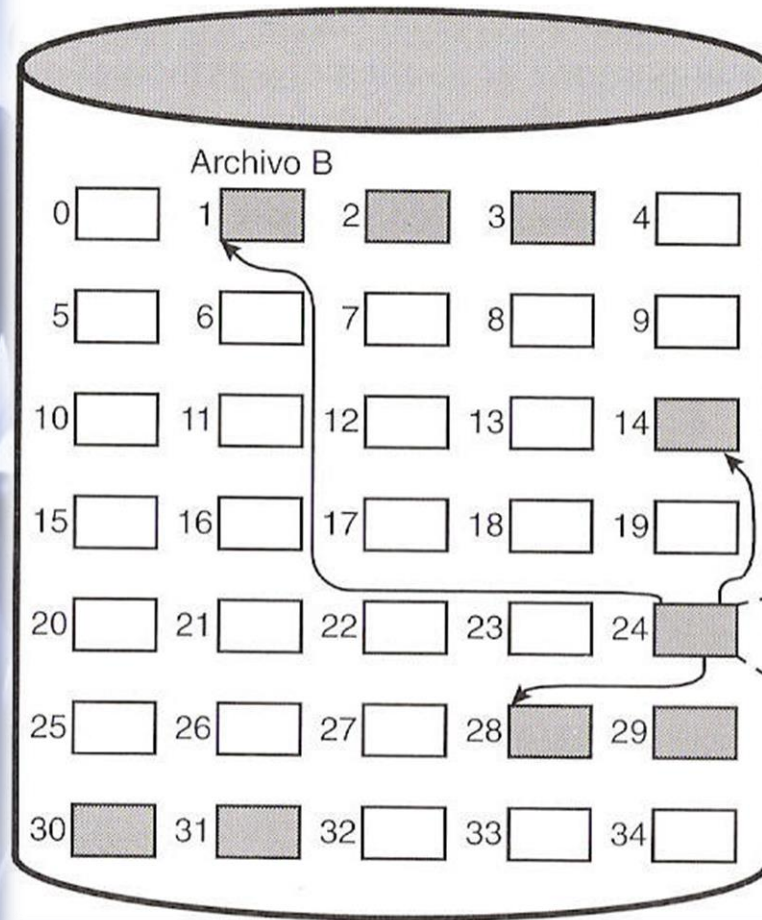
Asignación indexada por bloques

Tabla de asignación de archivos

Nombre de archivo	Bloque índice
...	...
Archivo B	24
...	...

1
8
3
14
28

Asignación Indexada



Asignación indexada por secciones de longitud variable

Tabla de asignación de archivos

Nombre de archivo	Bloque índice
...	...
Archivo B	24
...	...

Bloque de inicio	Longitud
1	3
28	4
14	1

The background is a deep blue gradient. On the left side, there is a stylized representation of a globe with several interlocking puzzle pieces. Some pieces are white and others are dark blue. A bright light source from the upper left creates a strong lens flare effect across the right side of the image. The title text is positioned on the right, within the lens flare area.

Administración del Espacio Libre

Introducción

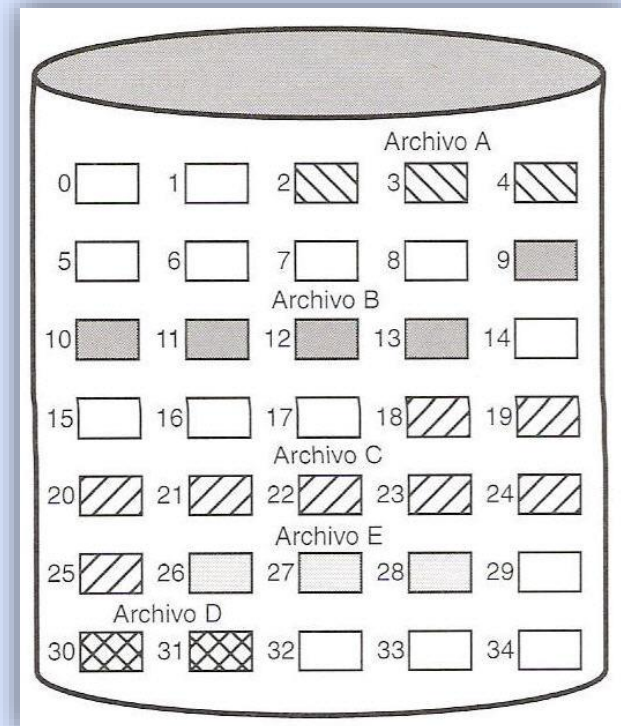
- ◆ En el sistema de archivos es necesario mantener qué bloques están ocupados y cuáles libres.
- ◆ Técnicas de uso común:
 - ✦ Mapa de bits (*Bit Vector, Bit Map*).
 - ✦ Lista de bloques libres (*Linked list*).
 - ✦ Agrupación (*Grouping*).
 - ✦ Conteo (*Counting*).

Mapa de Bits

- ◆ Se dispone de un bit para cada bloque de datos del sistema, que representa si está ocupado o libre.
- ⊕ Si el bloque está libre, el bit es 1.
- ⊕ Si el bloque está asignado, el bit es 0.

Por ejemplo, para la disposición del siguiente disco se necesitaría un vector de longitud 35 que tendría el siguiente valor:

11000111100000111100000000000100111



Mapa de Bits

- ◆ Una forma de calcular la cantidad de memoria (en bytes) requerida por un mapa de bits en bloques se puede calcular de la siguiente manera:

tamaño del disco en bytes

8 x tamaño del bloque en el sistema de archivos

Por ejemplo, para un disco de 16 Gb con bloques de 512 bytes, la tabla ocupará **4 Mb**.

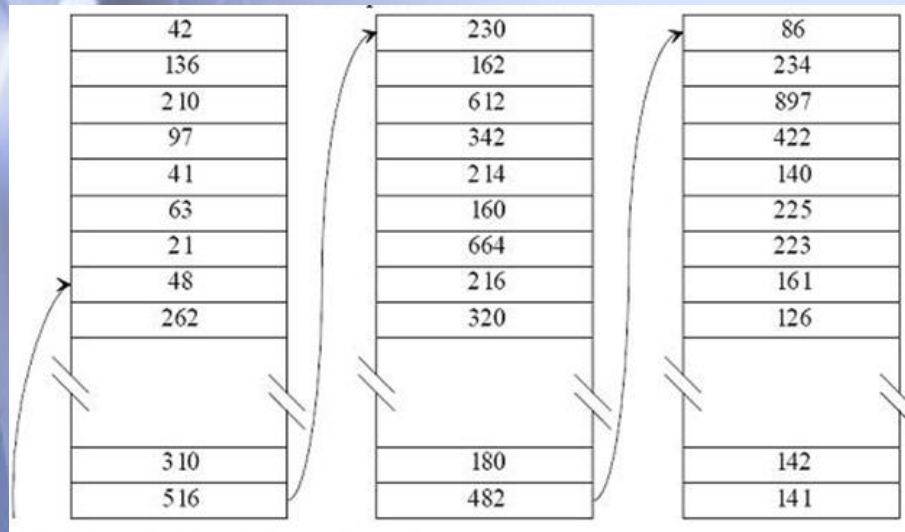
1 Gb = 1073741824 bytes

16 * 1073741824 = 17179869184 bytes

17179869184 / (8 * 512) = 4194304 = **4 Mb**

Lista Enlazada

- ◆ Se mantiene una lista encadenada con los bloques libres a través de los bloques. Es necesario una referencia al primer bloque.
- ◆ El apuntador se coloca en la memoria caché.



Bloques de 1Kb y números de bloque de 32 bits, cada bloque de la Lista Libre contendrá 255 números de bloque libres de 32 bits (se necesita un número de bloque para el apuntador al siguiente bloque).

Un disco de 16 Gb necesita 224 bloques como máximo para contener $224 * 255$ números de bloque.

Administración del espacio libre

- ◆ **Agrupación:** es una variación de la lista encadenada. En cada bloque de la lista se contiene un grupo de bloques libres.
- ◆ **Conteo:** se mantiene una lista en donde cada bloque contiene información de cuántos bloques contiguos, a partir de él, están libres.

Leer y comprender los conceptos de Sistemas de Archivos que tratan las siguientes referencias:

- ❑ <http://social.technet.microsoft.com/wiki/contents/articles/5375.windows-file-systems.aspx>
- ❑ <http://www.ibm.com/developerworks/library/l-linux-filesystem/>
- ❑ <https://developer.apple.com/library/prerelease/mac/documentation/FileManagement/Conceptual/FileSystemProgrammingGuide/FileSystemOverview/FileSystemOverview.html>



Ing. Yesenia Carrera Fournier
sofiunam at gmail dot com