

GESTIÓN DEL SISTEMA DE ARCHIVOS

Objetivos

- ❑ Conocer el funcionamiento del sistema de gestión de archivos.
- ❑ Identifica la organización del sistema de archivos
- ❑ Analizar las diferentes formas de implementación del sistema de archivos
- ❑ Definir los métodos que utiliza el SO para el manejo de la seguridad y protección del sistema de archivos

Agenda

- ❑ Almacenamiento físico de archivos
- ❑ Estructura y función del sistema de archivos
- ❑ Propiedades de los archivos
- ❑ Operaciones sobre los archivos
- ❑ Seguridad y protección de los archivos

Almacenamiento físico de archivos

- ❑ Archivo
 - Conjunto de registros, instrucciones de programa o datos agrupados bajo un nombre y almacenados en memoria no volátil.
- ❑ Tecnologías de almacenamiento físico
 - Medios magnéticos
 - Discos
 - Cintas
 - Medios ópticos
 - CD R/W
 - DVD R/W
 - Memorias flash
 - Memorias USB

Almacenamiento físico de archivos

- ❑ Por rendimiento los sectores realmente no se numeran de forma contigua, existe un factor de intercalación.
- ❑ Cuando los datos se organizan físicamente en sectores un archivo es considerado como una serie de cúmulos o bloques de sectores.
 - Un cúmulo (clúster): número fijo de sectores “contiguos” y es la unidad de disco mas pequeña que el sistema operativo puede administrar (*un cúmulo puede ser un sector*).
- ❑ El sistema de archivos se basa en la administración de bloques.
 - Tamaño grande de bloque equivale a mayor desperdicio.



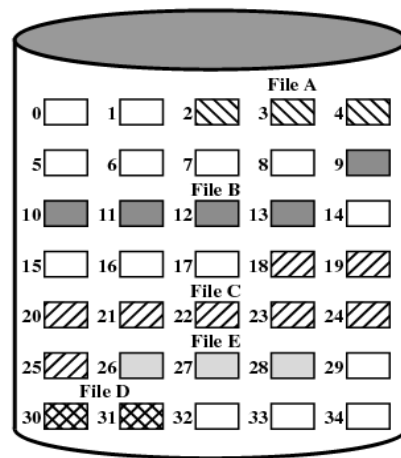
Almacenamiento físico de archivos

- ❑ Asignar espacio en el disco para los archivos requiere que el sistema operativo asigne ciertos bloques para tal fin.
- ❑ En la asignación del espacio se debe procurar que el espacio se aproveche de forma eficaz y se pueda acceder de forma rápida a los archivos
- ❑ Se utilizan principalmente tres métodos de asignación:
 - *Contigua*
 - *Encadenada*
 - *Indexada*

Almacenamiento físico de archivos

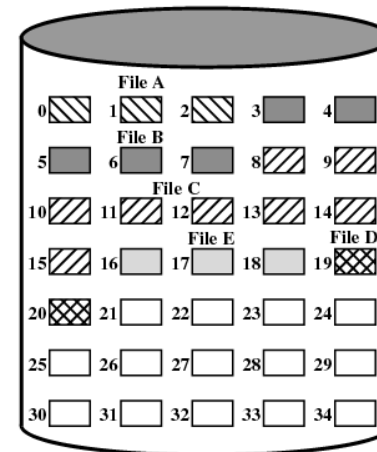
□ **Asignación contigua**

- Cada archivo ocupa un conjunto de bloques contiguos en el disco, dicho conjunto se asigna al crear el archivo.
- La tabla de asignación de archivos (FAT) contiene el número de bloque inicial y la longitud en bloques del archivo.
- Tiene buenas prestaciones para archivos secuenciales.
- Genera fragmentación externa y se dificulta encontrar espacio para un nuevo archivo.
- De vez en cuando se debe compactar para liberar espacio.



File Allocation Table		
File Name	Start Block	Length
File A	2	3
File B	9	5
File C	18	8
File D	30	2
File E	26	3

Asignación contigua
antes de compactar



File Allocation Table		
File Name	Start Block	Length
File A	0	3
File B	3	5
File C	8	8
File D	19	2
File E	16	3

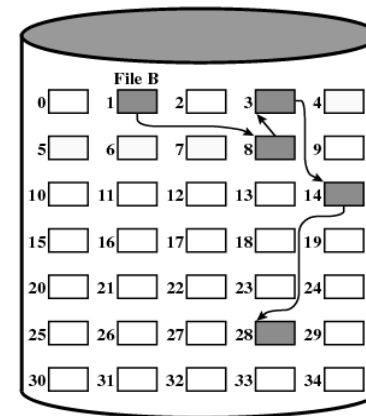
Asignación contigua
después de compactar

Almacenamiento físico de archivos

❑ **Asignación encadenada**

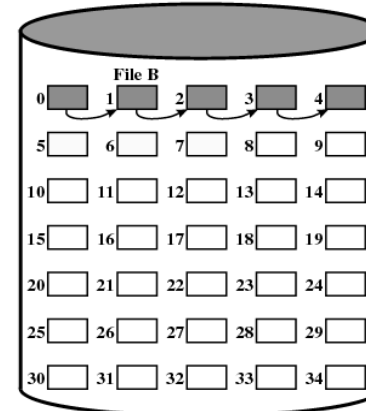
- La asignación se realiza a nivel de bloques.
- Cada bloque contiene un apuntador al siguiente bloque de la cadena.
- La FAT contiene el número de bloque inicial y la longitud en bloques del archivo.
- Buenas prestaciones para archivos secuenciales.
- No existe la fragmentación externa.
- No existe el principio de proximidad lo que puede implicar múltiples accesos a disco.

Asignación encadenada antes de compactar



File Allocation Table		
File Name	Start Block	Length
...
File B	1	5
...

Asignación encadenada después de consolidar



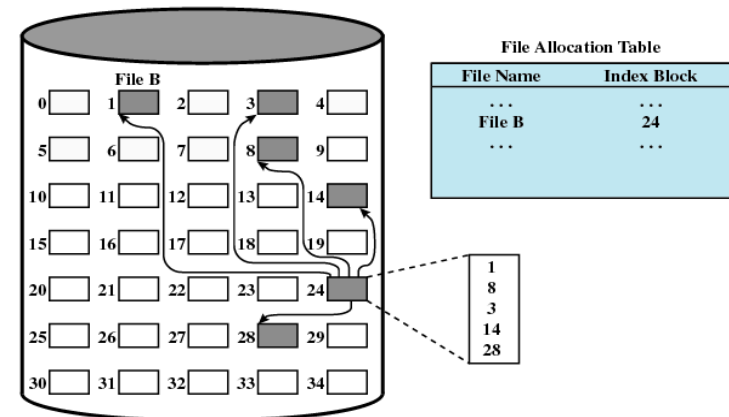
File Allocation Table		
File Name	Start Block	Length
...
File B	0	5
...

Almacenamiento físico de archivos

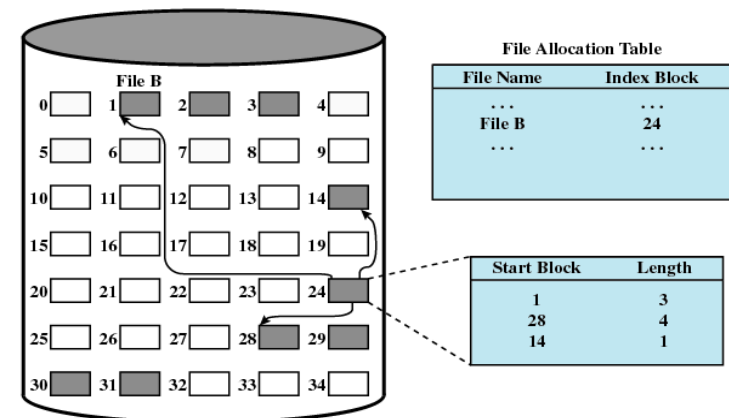
❑ **Asignación indexada**

- Resuelve muchos de los inconvenientes de las asignaciones contigua y encadenada.
- La FAT contiene el número de bloque índice de un nivel por cada archivo.
- El bloque indexado contiene una entrada (registro) por cada porción asignada al archivo.
- La asignación puede realizarse mediante bloques de tamaño fijo o porciones de tamaño variable.
- La consolidación reduce el tamaño del índice.
- Esta es la forma más utilizada en la asignación de archivos.

Asignación indexada con porciones de bloques



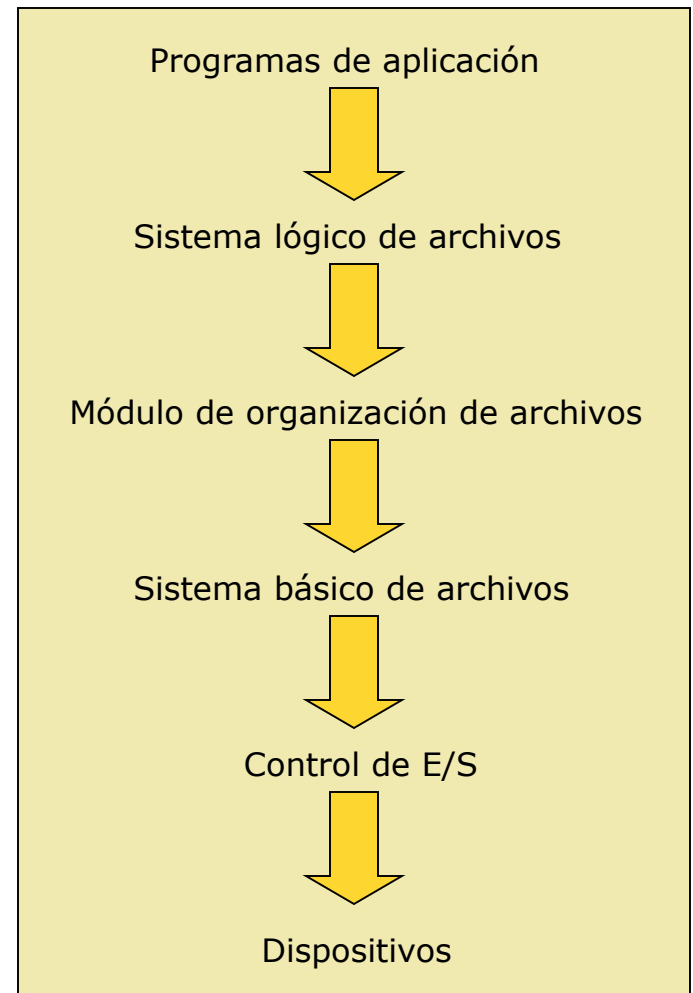
Asignación indexada con porciones de tamaño variable



Estructura y función del sistema de archivos

- ❑ Para ofrecer un acceso eficiente y cómodo al disco, el sistema operativo impone en él un sistema de archivos.
- ❑ Generalmente el sistema de archivos esta compuesto de varios niveles.
 - Cada nivel de diseño aprovecha las funciones de los niveles inferiores para crear nuevas funciones que se usarán en niveles superiores

Sistema de archivos en niveles



Estructura y función del sistema de archivos

❑ **Control de E/S**

- Esta compuesto por los controladores del dispositivo y rutinas de tratamiento de interrupción para transferir información entre la memoria principal y el sistema de disco.

❑ **Sistema básico de archivos**

- Envía comandos genéricos al controlador de dispositivo apropiado para leer y escribir bloques físicos en el disco.

Estructura y función del sistema de archivos

❑ **Módulo de organización de archivos**

- Tiene conocimiento de los archivos y sus bloques físicos y lógicos.
- También incluye el gestor de espacio libre, el cual controla bloques no asignados y asigna bloques cuando se requiere.

❑ **Sistema lógico de archivos**

- Gestiona la información de metadatos, los cuales incluyen toda la estructura del sistema de archivos, excepto los propios datos.
- Gestiona la estructura de directorios, propietarios, permisos, ubicación protección y seguridad.

Estructura y función del sistema de archivos

- ❑ Los objetivos que se sugieren para un sistema de gestión de archivos son:
 - Satisfacer las necesidades de gestión de datos y requisitos del usuario.
 - Optimizar el rendimiento en términos de productividad (sistema) y en términos del tiempo de respuesta (usuario).
 - Proporcionar soporte de E/S a una variedad de tipos de dispositivos de almacenamiento.
 - Identificar y localizar el archivo seleccionado.
 - Utilizar un directorio para describir la localización de todos los archivos más sus atributos.
 - En un sistema compartido debe describir el control de acceso para los usuarios.
 - Bloquear el acceso a los archivos
 - Asignar a los archivos bloques libres
 - Manejar el espacio libre para la disponibilidad de bloques

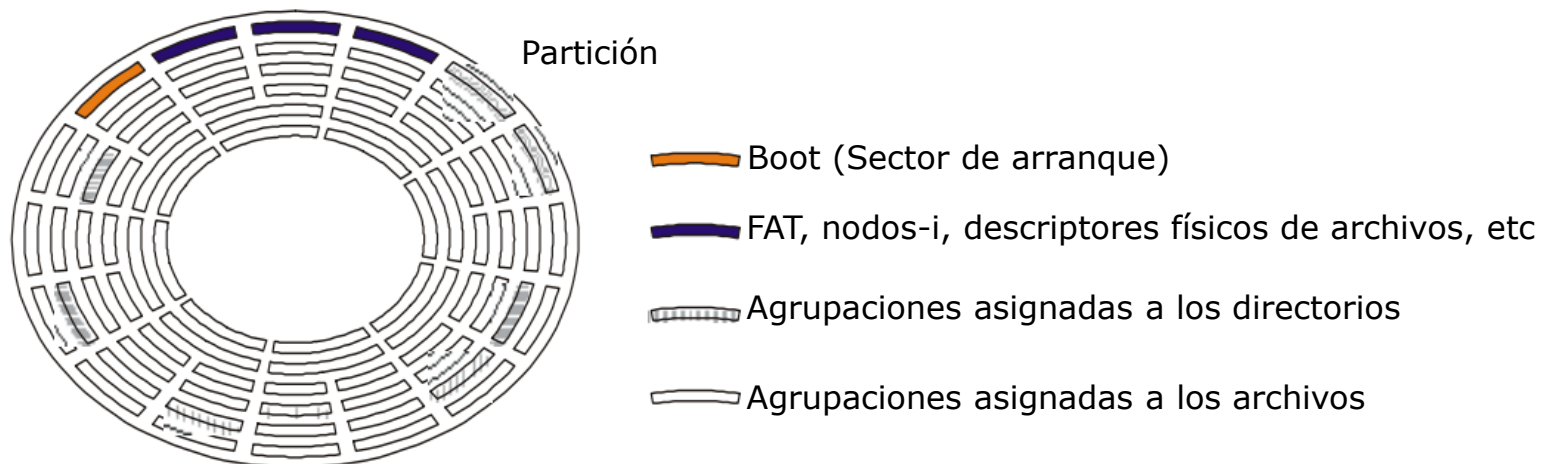
Estructura y función del sistema de archivos

- La estructura y funciones particulares del sistema de archivos dependerá del sistema operativo que se este utilizando.
- Previo a la instalación del sistema de archivos es necesario dividir física o lógicamente los discos en particiones o volúmenes.
 - Una partición es una porción del disco a la que se le dota de una identidad propia y que puede ser manipulada por el sistema operativo como una entidad lógica independiente.
- Una vez creadas las particiones el sistema operativo debe crear las estructuras de los sistemas de archivos dentro de esas particiones.

Sistema operativo	Tipos de sistemas de archivos admitidos
Dos	FAT16
Windows 95	FAT16
Windows 95 OSR2	FAT16, FAT32
Windows 98	FAT16, FAT32
Windows NT4	FAT, NTFS (versión 4)
Windows 2000/XP	FAT, FAT16, FAT32, NTFS (versiones 4 y 5)
Linux	Ext2, Ext3, ReiserFS, Linux Swap (FAT16, FAT32, NTFS)
MacOS	HFS (Sistema de Archivos Jerárquico), MFS (Sistemas de Archivos Macintosh)
OS/2	HPFS (Sistema de Archivos de Alto Rendimiento)
SGI IRIX	XFS
FreeBSD, OpenBSD	UFS (Sistema de Archivos Unix)
Sun Solaris	UFS (Sistema de Archivos Unix)
IBM AIX	JFS (Sistema Diario de Archivos)

Estructura y función del sistema de archivos

- Al crear un sistema de archivos en una partición de un disco, se crea una entidad lógica autocontenida con:
 - Espacio para la información de carga del sistema operativo
 - La descripción de su estructura
 - Descriptores de archivos
 - Información del estado de ocupación de los bloques del sistema de archivos
 - Bloques de datos y directorios



Estructura y función del sistema de archivos

□ Tarea:

- Investigar sobre la estructura y funcionamiento de los siguientes sistemas de archivos:
 - FAT, FAT16, FAT32
 - Ext2, Ext3, Ext4
 - NTFS
- Incluir:
 - Cómo se realiza la asignación de ficheros en Unix
 - Estructuras de ficheros en NTFS

Propiedades de los archivos

- ❑ Desde el punto de vista del usuario, una de las partes más importantes de un sistema operativo es el sistema de archivos.
- ❑ El sistema de archivos permite crear archivos con las siguientes propiedades deseables:
 - **Existencia a largo plazo**
 - Almacenamiento en disco u otra tecnología de almacenamiento secundario permanente.
 - **Compartible entre procesos**
 - Los archivos tienen nombres y permisos de acceso asociados para controlar la compartición.
 - **Estructura**
 - Dependiendo del sistema de archivos, los archivos pueden tener una estructura interna conveniente a las aplicaciones.
 - Los archivos se pueden organizar en estructuras jerárquicas o más complejas.

Propiedades de los archivos

❑ Términos relacionados con los archivos

■ **Campo**

- Es el elemento básico de los datos.
- Contiene un único valor (p.e. Apellido, fecha, temperatura, etc.)
- Tiene un tipo y longitud asociados
- Dependiendo del diseño del archivo su longitud puede ser fija o variable.

■ **Registro**

- Es una colección de campos relacionados que pueden tratarse como una unidad.
- Dependiendo del diseño del archivo pueden ser de longitud fija o variable.
- Puede ser de longitud variable si uno de sus campos lo es o también si el número de sus campos varía.

■ **Archivo**

- Es una colección de campos similares.
- Es tratado como una entidad única por parte de los usuarios o aplicaciones.
- Se referencian a través de un nombre.

Propiedades de los archivos

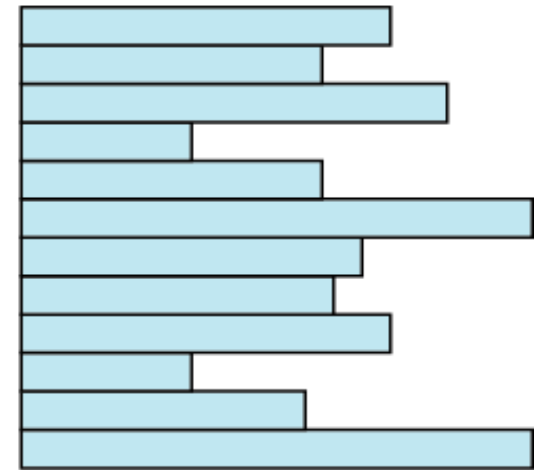
- ❑ Existen diversas formas de organizar los registros de un archivo, las cinco que se consideran fundamentales son:
 - La pila
 - El archivo secuencial
 - El archivo secuencial indexado
 - El archivo indexado
 - El archivo de acceso directo o hash

Propiedades de los archivos

❑ La pila

- La forma menos complicada de organización.
- Los datos se almacenan en el orden en que llegan.
- Los registros pueden tener diferentes campos o similares campos en diferentes órdenes.
- Cada campo debe ser autodescriptivo, incluyendo el nombre del campo y el valor
- La longitud se conoce por defecto, se indica mediante un delimitador o se incluye en un subcampo.
- El acceso a los registros se hace mediante búsqueda exhaustiva.

Archivo tipo pila



Registros de tamaño variable
Orden cronológico

Propiedades de los archivos

El archivo secuencial

- La forma más común de estructurar archivos
- Utiliza un formato fijo para los registros
 - Todos de igual tamaño, compuestos por el mismo número de campos de longitud fija en un orden específico.
- Solo se almacenan los valores de los campos; el nombre y la longitud de cada campo son atributos de la estructura del archivo.
- Normalmente uno de los campos es el campo clave
 - Identifica de forma única al registro
 - La secuencia de los registros es según la clave
- Típicamente la organización lógica encaja con la organización física.

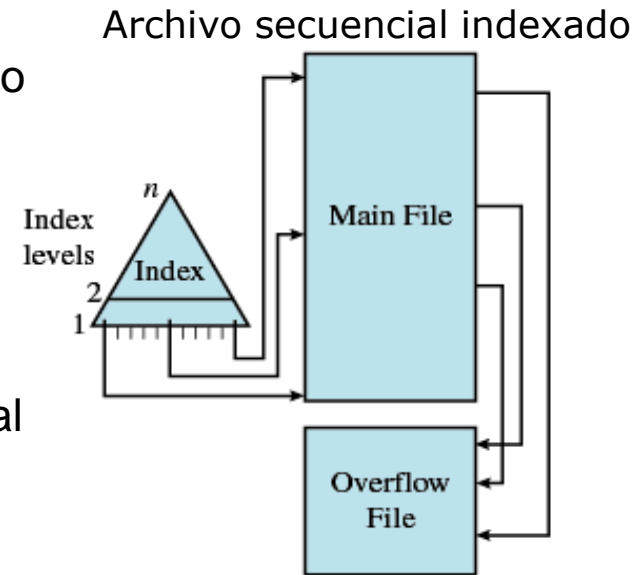
Archivo secuencial

[illegible]

Propiedades de los archivos

□ El archivo secuencial indexado

- Mantiene las características clave del archivo secuencial
 - Registros en secuencia basándose en un campo clave.
- Añade dos características
 - Un índice que da soporte al acceso aleatorio
 - Un archivo de desbordamiento
- En un nivel de indexación
 - El índice esta formado por un archivo secuencial simple.
 - En cada registro dos campos: la clave y el apuntador al archivo principal.
 - Para encontrar un campo específico se busca el índice que contenga la clave anterior más próxima a la del campo deseado, después la búsqueda continúa en el archivo principal.



Propiedades de los archivos

□ Ejemplo de archivo secuencial indexado

Orden de llegada de los datos: 3 2 15 11

1

Índice		F. Secuencial	
1		1	Null
10			Null
20		10	Null
...	...	20	Null
...	Null

Desbordamiento	
	Null
	Null
	Null
	Null
	Null

Situación de partida

2

Índice		F. Secuencial	
1		1	Null
10			Null
20		10	Null
...	...	20	Null
...	Null

Desbordamiento	
	Null
	Null
	Null
	Null
	Null

Dato a insertar: 3 2 15 11

Búsqueda en el índice del valor anterior más próximo: 1

3

Índice		F. Secuencial	
1		1	Null
10			Null
20		10	Null
...	...	20	Null
...	Null

Desbordamiento	
	Null
	Null
	Null
	Null
	Null

Dato a insertar: 3 2 15 11

Búsqueda en el fich. secuencial del dato anterior más próximo: 1

4

Índice		F. Secuencial	
1		1	Null
10		3	Null
20		10	Null
...	...	20	Null
...	Null

Desbordamiento	
	Null
	Null
	Null
	Null
	Null

Dato a insertar: 3 2 15 11

Se inserta el dato a continuación en el fichero secuencial

Propiedades de los archivos

□ Ejemplo de archivo secuencial indexado

Orden de llegada de los datos: 3 2 15 11

5

Índice		F. Secuencial		Desbordamiento	
1		1	Null		Null
10		3	Null		Null
20		10	Null		Null
...	...	20	Null		Null
...	Null		Null

Dato a insertar: 3 2 15 11

Búsqueda en el índice del valor anterior más próximo: 1

6

Índice		F. Secuencial		Desbordamiento	
1		1	Null		Null
10		3	Null		Null
20		10	Null		Null
...	...	20	Null		Null
...	Null		Null

Dato a insertar: 3 2 15 11

Búsqueda en el fich. secuencial del dato anterior más próximo: 1

7

Índice		F. Secuencial		Desbordamiento	
1		1		2	Null
10		3	Null		Null
20		10	Null		Null
...	...	20	Null		Null
...	Null		Null

Dato a insertar: 3 2 15 11

Como no existe hueco, se inserta en el fich. de desbordamiento y se crea puntero

8

Índice		F. Secuencial		Desbordamiento	
1		1		2	Null
10		3	Null		Null
20		10	Null		Null
...	...	20	Null		Null
...	Null		Null

Dato a insertar: 3 2 15 11

Búsqueda en el índice del valor anterior más próximo: 10

Propiedades de los archivos

□ Ejemplo de archivo secuencial indexado

Orden de llegada de los datos: 3 2 15 11

9

Índice	F. Secuencial	Desbordamiento
1	1	2
10	3	Null
20	10	Null
...	20	Null
...	...	Null

Dato a insertar: 3 2 15 11

Búsqueda en el fich. secuencial del dato anterior más próximo:
10

10

Índice	F. Secuencial	Desbordamiento
1	1	2
10	3	15
20	10	Null
...	20	Null
...	...	Null

Dato a insertar: 3 2 15 11

Como no existe hueco, se inserta en el fich. de desbordamiento y se crea puntero

11

Índice	F. Secuencial	Desbordamiento
1	1	2
10	3	15
20	10	Null
...	20	Null
...	...	Null

Dato a insertar: 3 2 15 11

Búsqueda en el índice del valor anterior más próximo: 10

12

Índice	F. Secuencial	Desbordamiento
1	1	2
10	3	15
20	10	Null
...	20	Null
...	...	Null

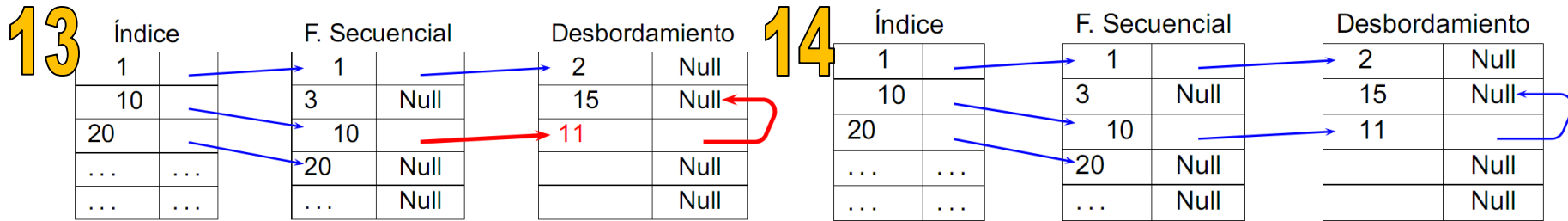
Dato a insertar: 3 2 15 11

Búsqueda en el fich. secuencial del dato anterior más próximo:
10

Propiedades de los archivos

□ Ejemplo de archivo secuencial indexado

Orden de llegada de los datos: 3 2 15 11

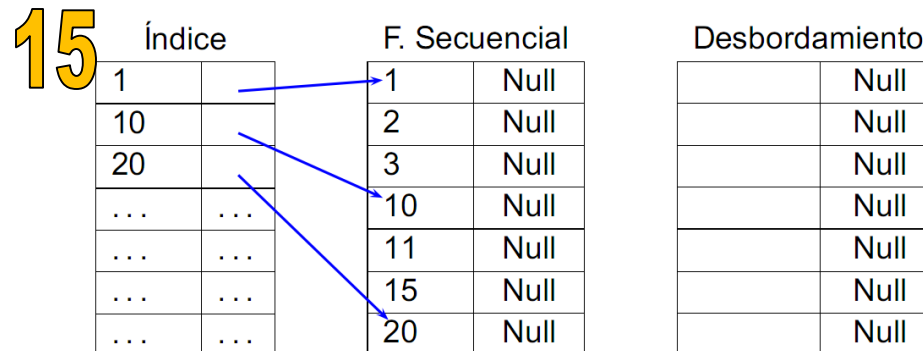


Dato a insertar: 3 2 15 11

Se inserta el nuevo dato en la lista cuya cabecera es el campo puntero asociado al dato 10

Situación final

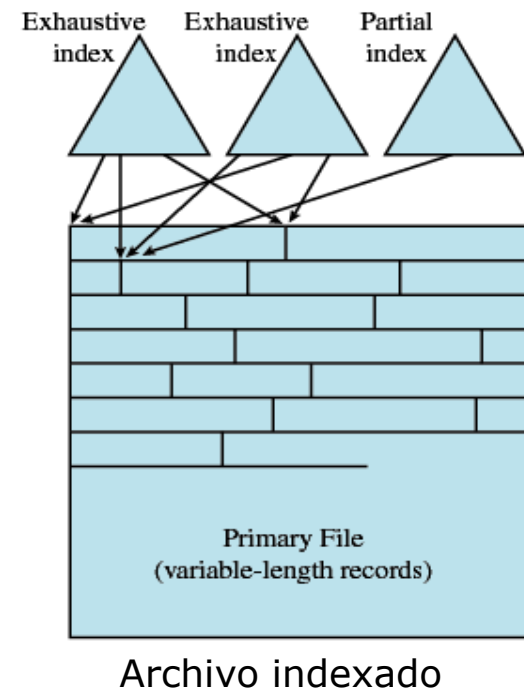
Situación tras una reorganización de la información:



Propiedades de los archivos

❑ El archivo indexado

- Se utilizan dos tipos de índice
 - Un índice exhaustivo que contiene una entrada por cada registro del archivo principal.
 - Un índice parcial que contiene entradas a registros donde el campo de interés existe.
- Se dispone de un índice por cada tipo de campo de búsqueda.
- Los registros pueden ser de longitud variable y se acceden solo a través de sus índices.
- Cuando se añade un registro al archivo principal, todos los archivos índice deben actualizarse.



Propiedades de los archivos

❑ El archivo de acceso directo o hash

- Explota la capacidad de los discos para acceder directamente a cualquier bloque de una dirección conocida.
- Se requiere de una clave para cada registro
- No existe el concepto de ordenación secuencial, se hace uso de una tabla hash sobre un valor clave.
- Hace uso de un archivo de desbordamiento.
- Posible Algoritmo de inserción
 - Asociar al elemento una etiqueta n entre 0 y $m - 1$
 - Donde m es el número de entradas y n el número de elemento.
 - Usar n como índice de la tabla hash
 - Si la entrada n está vacía, almacenar el elemento
 - Si esta ocupada, almacenar el elemento en un área de desbordamiento, creando una lista de elementos con la misma etiqueta.

Propiedades de los archivos

❑ Ejemplo de archivo hash

Orden de llegada de los datos: 50 150 351 250

Función de hash: $Dato \text{ MOD } 100$

1

Tabla hash		
49		Null
50		Null
51		Null
52		Null

Desbordamiento	
	Null
	Null
	Null
	Null

Orden de llegada de los datos: 50 150 351 250

Función de hash: $Dato \text{ MOD } 100$

2

Tabla hash		
49		Null
50	Dato 50	Null
51		Null
52		Null

Desbordamiento	
	Null
	Null
	Null
	Null

$50 \text{ MOD } 100 = 50$

La entrada está vacía. Se introduce el dato en la entrada 50.

Orden de llegada de los datos: 50 150 351 250

Función de hash: $Dato \text{ MOD } 100$

3

Tabla hash		
49		Null
50	Dato 50	Null
51		Null
52		Null

Desbordamiento	
Dato 150	Null
	Null
	Null
	Null

$150 \text{ MOD } 100 = 50$

La entrada está ocupada. Se introduce el dato en el fichero de desbordamiento y se crea una lista de datos con etiqueta 50

Orden de llegada de los datos: 50 150 351 250

Función de hash: $Dato \text{ MOD } 100$

4

Tabla hash		
49		Null
50	Dato 50	Null
51	Dato 351	Null
52		Null

Desbordamiento	
Dato 150	Null
	Null
	Null
	Null

$351 \text{ MOD } 100 = 51$

La entrada está vacía. Se introduce el dato en la entrada 51.

Propiedades de los archivos

❑ Ejemplo de archivo hash

Orden de llegada de los datos: 50 150 351 250

Función de hash: $Dato \text{ MOD } 100$

5

250

Tabla hash

49		Null
50	Dato 50	
51	Dato 351	Null
52		Null

Desbordamiento

Dato 150	
Dato 250	Null
	Null
	Null

$250 \text{ MOD } 100 = 50$

La entrada está ocupada. Se introduce el dato en el fichero de desbordamiento y se añade el dato al final de la lista de datos con etiqueta 50

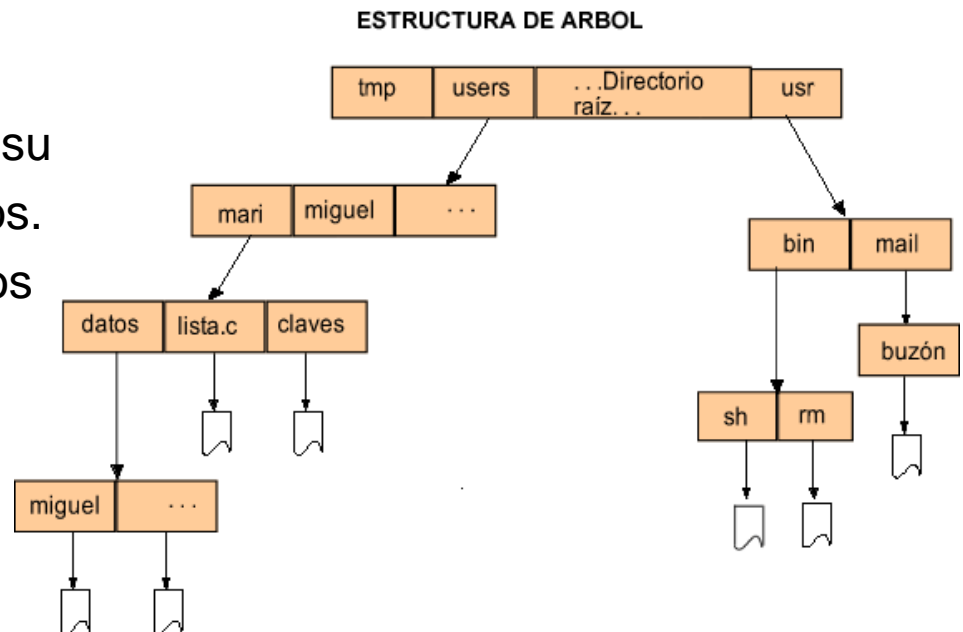
Propiedades de los archivos

- ❑ De forma “**externa**” un archivo puede estar organizado en una estructura jerárquica llamada directorio.
- ❑ Directorio
 - Es un archivo que pertenece al sistema operativo y que es accedido a través de diversas rutinas de gestión de archivos.
 - Permite al usuario abstraerse de la ubicación física de los archivos visualizando solo la ubicación lógica.
 - Las dos estructuras lógicas más utilizadas son:
 - Árbol jerárquico
 - Grafo acíclico

Propiedades de los archivos

❑ Estructura de directorios de tipo árbol jerárquico

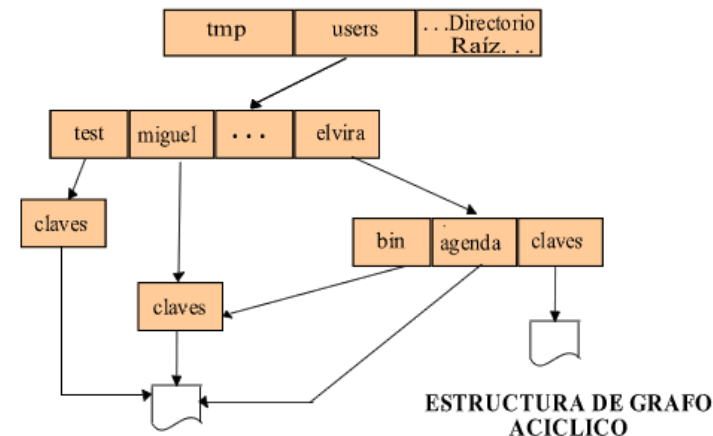
- Se parte de un directorio raíz
- Los nodos del árbol son subdirectorios que a su vez contienen otros subdirectorios o archivos.
- Las hojas del árbol son los archivos.
- Existe un camino (*path*) único para cada archivo.
- Permite a los usuarios definir su propia estructura de directorios.
- Se relaciona con los conceptos de:
 - Ruta absoluta
 - Ruta relativa
 - Directorio actual



Propiedades de los archivos

❑ Estructura de directorios de tipo grafo acíclico

- Permite que un archivo o subdirectorio puedan estar en directorios distintos.
- Para lograr compartir archivos o subdirectorios se utilizan dos tipos de enlaces
 - Físico (*duro*)
 - Apuntador a un archivo o directorio, cuya entrada de directorio tiene el mismo descriptor de archivo que el archivo enlazado.
 - Simbólico (*suave*)
 - Un nuevo archivo cuyo contenido es el nombre del archivo enlazado



Propiedades de los archivos

- ❑ Un archivo también se caracteriza por tener distintos atributos, entre ellos están:
 - **Nombre**
 - Identificador del archivo en formato comprensible para el usuario y es definido por su creador.
 - Algunos SO imponen ciertas restricciones.
 - **Identificador único**
 - Número que utiliza el sistema operativo para manejar el archivo.
 - **Tipo de archivo**
 - Útil en sistemas que soportan múltiples tipos de archivo (ejecutables y datos)
 - **Mapa del archivo**
 - Apuntadores a los dispositivos y a los bloques dentro de estos
 - **Protección**
 - Información de control de acceso (contraseña, dueño, creador, etc.)
 - **Tamaño del archivo**
 - Número de bytes en el archivo, máximo tamaño posible, etc.
 - **Información de control del archivo**
 - Indica si está oculto, si es del sistema, normal, etc.

Operaciones sobre los archivos

❑ Llamadas fundamentales sobre archivos al sistema

- Crear
- Borrar
- Abrir
- Cerrar
- Leer
- Escribir
- Posicionar
- Obtener atributos
- Renombrar

Ejemplo: Llamadas al sistema en Linux

```
fd = creat(nombre,modo)
fd = open(fichero, como)
s = close(fd)
n = read(fd,buffer,nbytes)
n = write(fd,buffer,nbytes)
s = chmod(nombre,modo)
```

❑ Llamadas fundamentales sobre directorios al sistema

- Crear
- Borrar
- Cambiar
- Mostrar
- Enlazar
- Desenlazar

Ejemplo: [llamadas al sistema](#) en Unix

```
s = mkdir(nombre,modo)
s = rmdir(nombre)
s = link(fich1,fich2)
s = unlink(nombre)
s = chdir(nombre_directorio)
```

Seguridad y protección de los archivos?