

# *Introducción a los Sistemas Operativos*

Administración de  
Archivos - II



- Versión: Noviembre 2017
- Palabras Claves: Archivo, Directorio, File System, Asignación, Espacio Libre

Algunas diapositivas han sido extraídas de las ofrecidas para docentes desde el libro de Stallings (Sistemas Operativos) y el de Silberschatz (Operating Systems Concepts). También se incluyen diapositivas cedidas por Microsoft S.A.



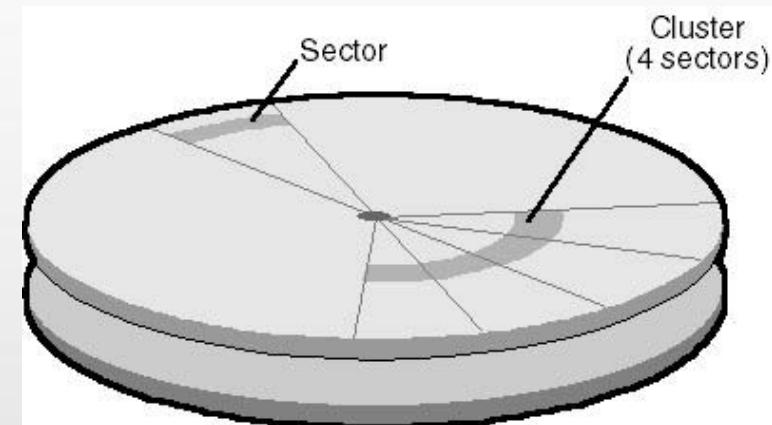
# *Metas del Sistema de Archivos*

- Brindar espacio en disco a los archivos de usuario y del sistema.
- Mantener un registro del espacio libre. Cantidad y ubicación del mismo dentro del disco.



# Conceptos

- Sector
  - ✓ Unidad de almacenamiento utilizada en los Discos Rígidos
- Bloque/Cluster
  - ✓ Conjuntos de sectores consecutivos
- File System
  - ✓ Define la forma en que los datos son almacenados
- FAT: File Allocation Table
  - ✓ Contiene información sobre en qué lugar están alojados los distintos archivos



# *Pre-asignación*

- Se necesita saber cuánto espacio va a ocupar el archivo en el momento de su creación
- Se tiende a definir espacios mucho más grandes que lo necesario
- Posibilidad de utilizar sectores contiguos para almacenar los datos de un archivo
- ¿Qué pasa cuando el archivo supera el espacio asignado?
- Esta técnica suele usar la forma de asignación continua (podría usar otras también)

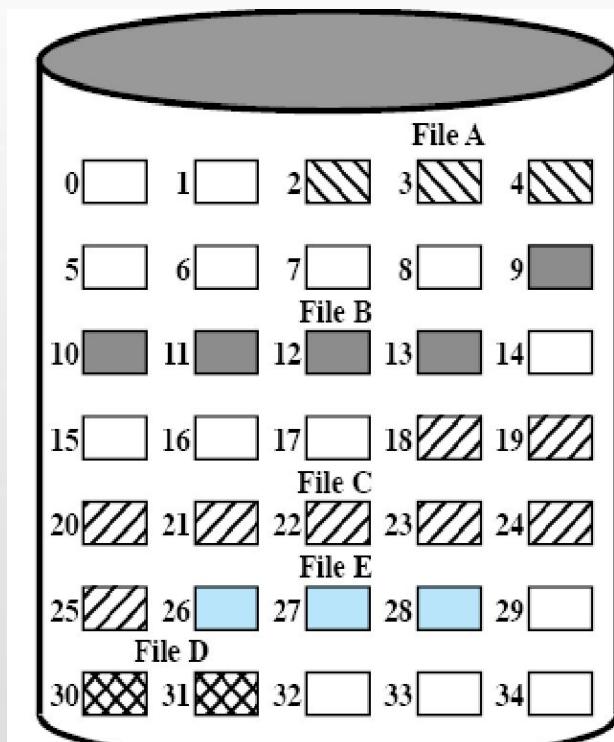


# Asignación Dinámica

- El espacio se solicita a medida que se necesita
- Los bloques de datos pueden quedar de manera no contigua



# Formas de Asignación - Continua



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

Que sucedería si necesitamos agregar un nuevo archivo de 6 bloques?



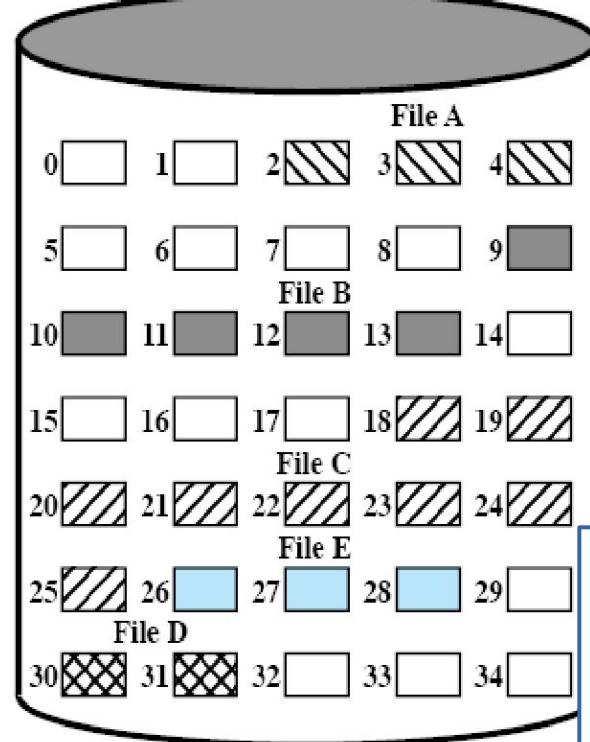
# Formas de Asignación - Continua

- Conjunto continuo de bloques son utilizados
- Se requiere una pre-asignación
  - ✓ Se debe conocer el tamaño del archivo durante su creación
- File Allocation Table (FAT) es simple
  - ✓ Sólo una entrada que incluye Bloque de inicio y longitud
- El archivo puede ser leído con una única operación
- Puede existir fragmentación externa
  - ✓ Compactación

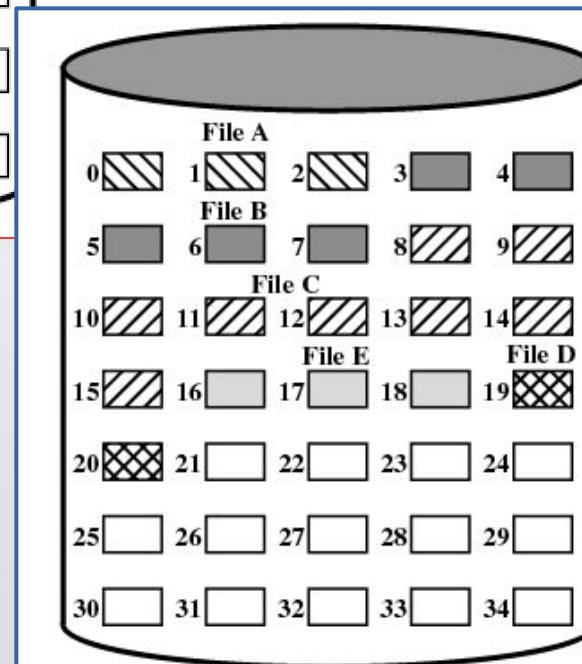
File Allocation Table		
File Name	Start Block	Length
File A	2	3
File B	9	5
File C	18	8
File D	30	2
File E	26	3



# Compactación



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



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



Figure 12.8 Contiguous File Allocation (After Compaction)

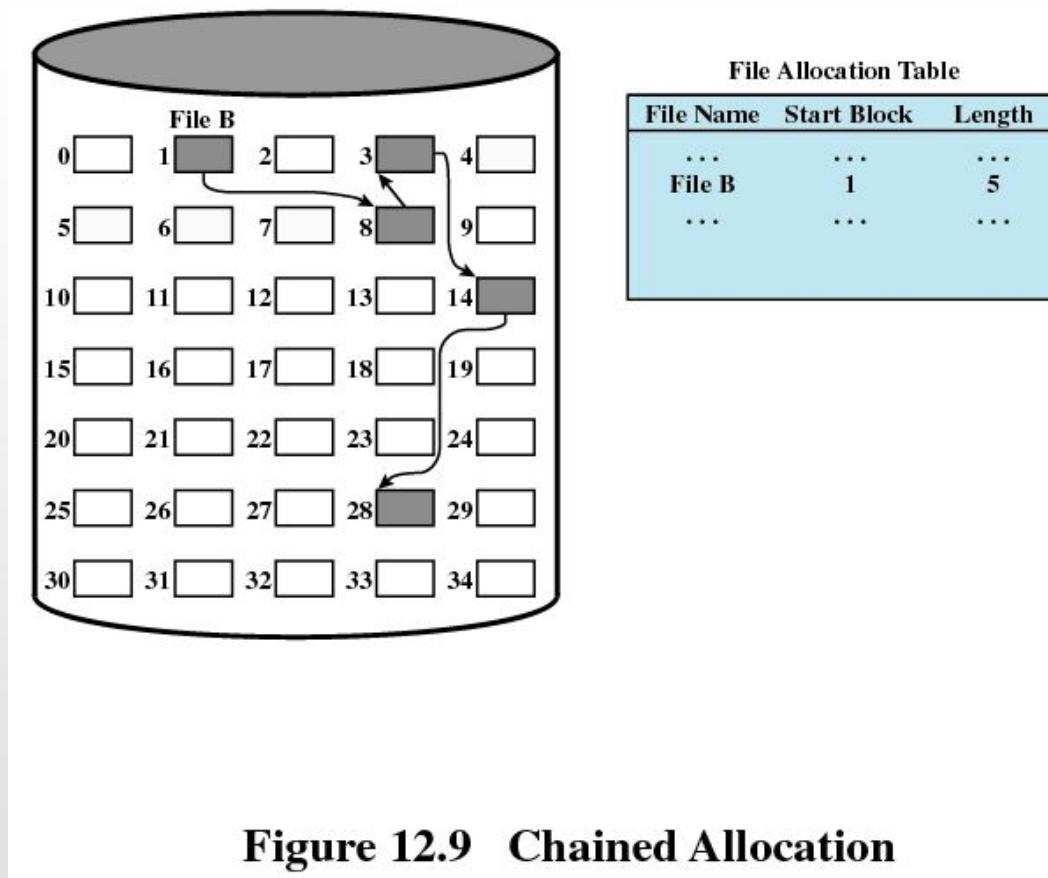
# *Formas de Asignación - Continua*

## □ Problemas de la técnica

- ✓ Encontrar bloques libres continuos en el disco
- ✓ Incremento del tamaño de un archivo



# Formas de Asignación - Encadenada



**Figure 12.9 Chained Allocation**



# *Formas de Asignación - Encadenada*

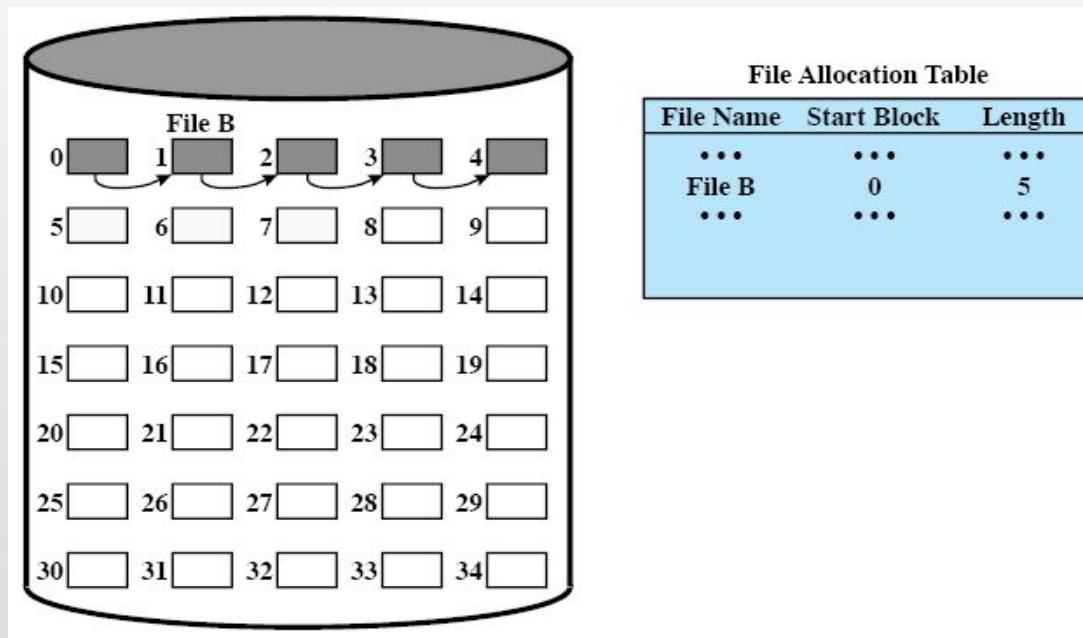
- Asignación en base a bloques individuales
- Cada bloque tiene un puntero al próximo bloque del archivo
- File allocation table
  - ✓ Única entrada por archivo: Bloque de inicio y tamaño del archivo
- No hay fragmentación externa
- Útil para acceso secuencial (no random)
- Los archivos pueden crecer bajo demanda
- No se requieren bloques contiguos

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

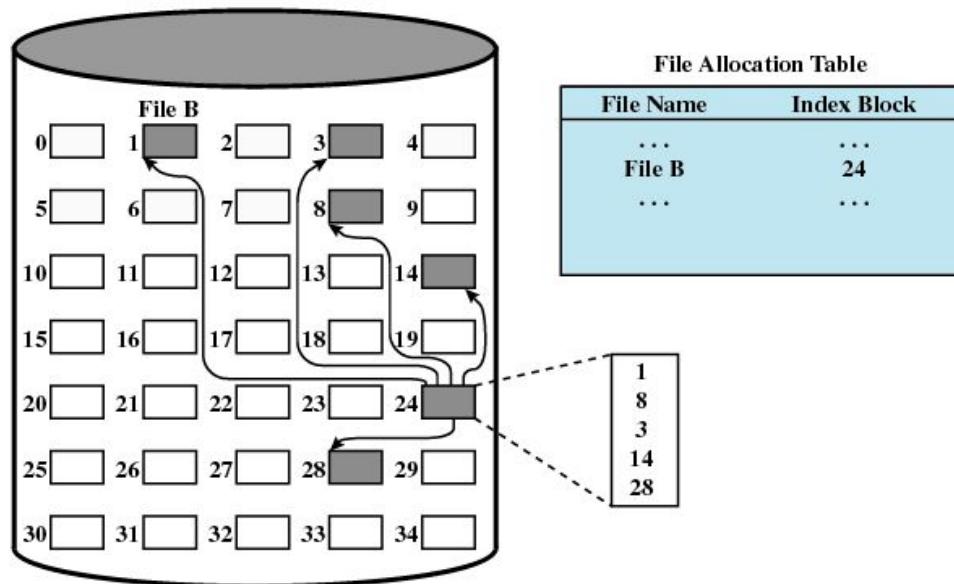


# Formas de Asignación - Encadenada

- Se pueden consolidar los bloques de un mismo archivo para garantizar cercanía de los bloques de un mismo archivo.



# Formas de Asignación - Indexada



- La FAT contiene un puntero al bloque índice
- El bloque índice no contiene datos propios del archivo, sino que contiene un índice a los bloques que lo componen

Figure 12.11 Indexed Allocation with Block Portions



# *Formas de Asignación - Indexada*

- Asignación en base a bloques individuales
- No se produce Fragmentación Externa
- El acceso “random” a un archivo es eficiente
- File Allocation Table
  - ✓ Única entrada con la dirección del bloque de índices (index node / i-node)

File Allocation Table	
File Name	Index Block
...	...
File B	24
...	...



# Formas de Asignación - Indexada

- Variante: asignación por secciones
- A cada entrada del bloque índice se agrega el campo longitud
- El índice apunta al primer bloque de un conjunto almacenado de manera contigua

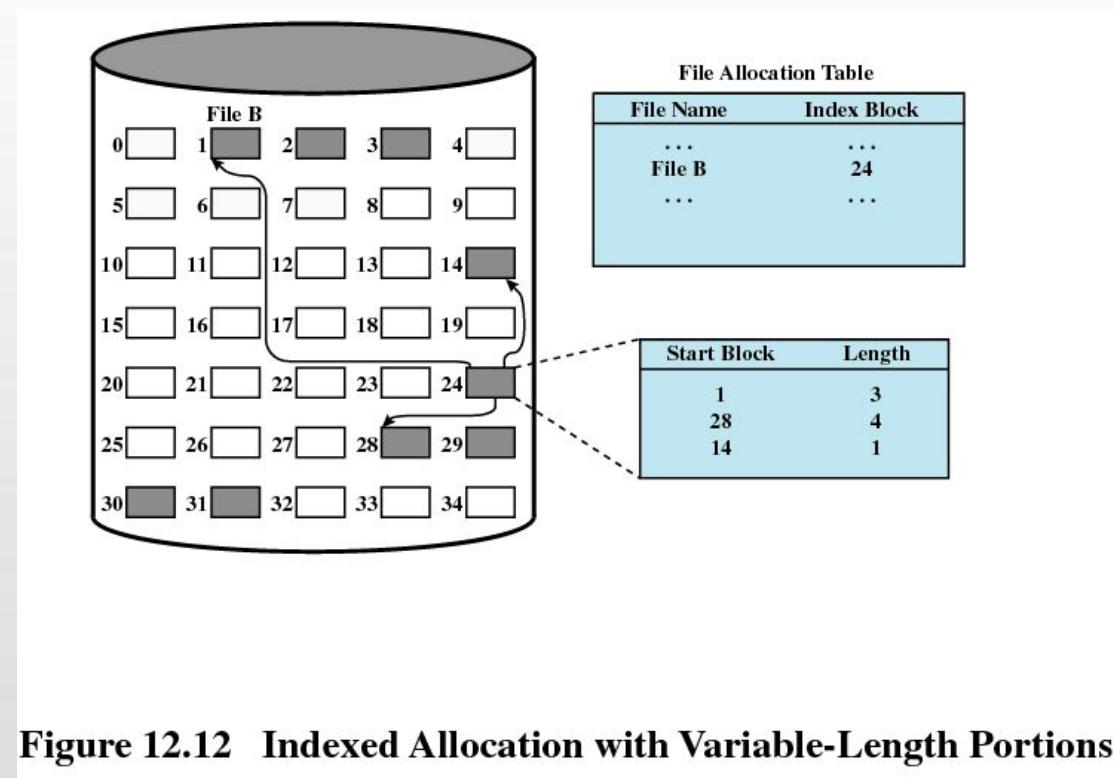


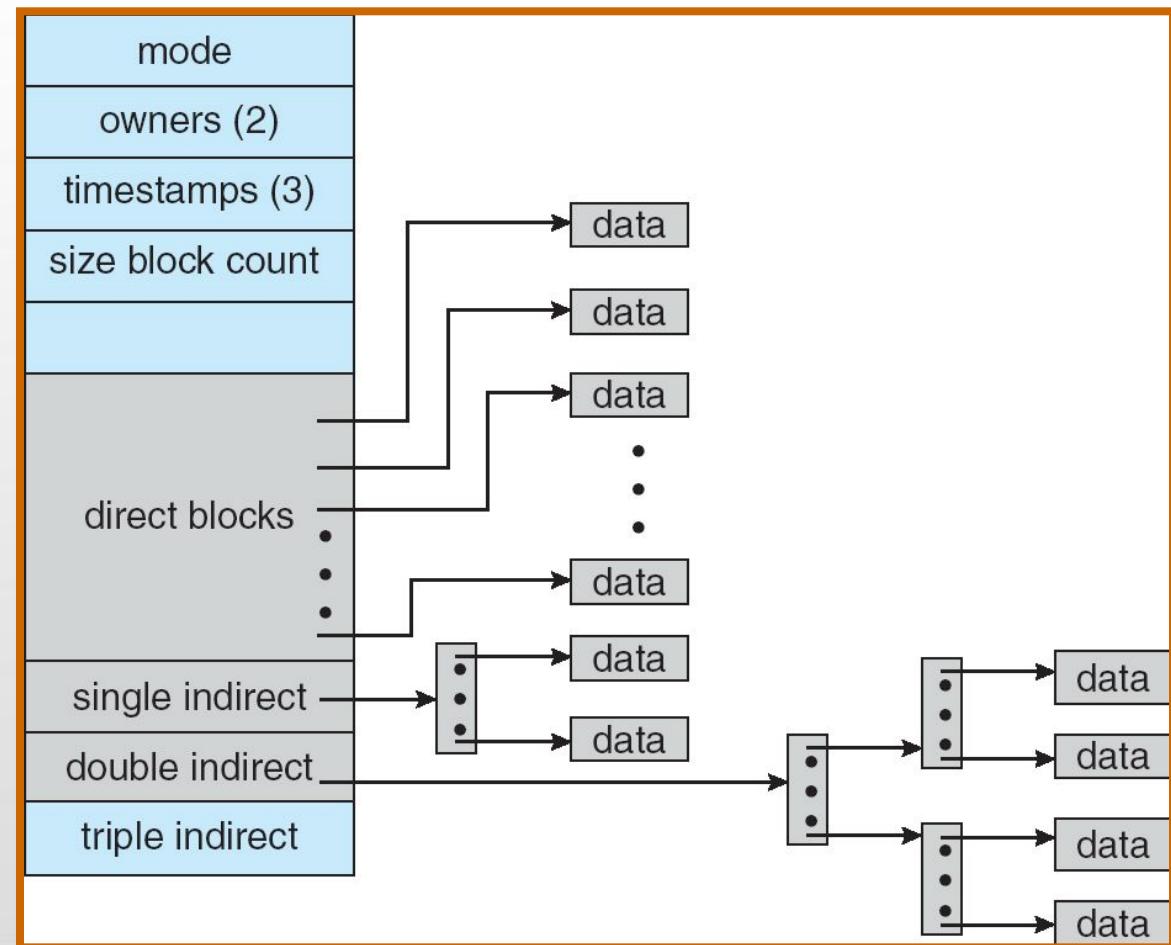
Figure 12.12 Indexed Allocation with Variable-Length Portions



# Formas de Asignación - Indexada

## □ Variante: niveles de indirección

- Existen bloques directos de datos
- Otros bloques son considerados como bloque índices (apuntan a varios bloques de datos)
- Puede haber varios niveles de indirección



# Asignación Indexada - Ejemplo

Cada I-NODO contiene 9 direcciones a los bloques de datos, organizadas de la siguiente manera:

- ◆ 7 de direccionamiento directo.
- ◆ 1 de direccionamiento indirecto simple
- ◆ 1 de direccionamiento indirecto doble

Si cada bloque es de 1KB y cada dirección usada para referenciar un bloque es de 32 bits:

- ✓ ¿Cuántas referencias (direcciones) a bloque pueden contener un bloque de disco?

$$1 \text{ KB} / 32 \text{ bits} = 256 \text{ direcciones}$$

- ✓ ¿Cuál sería el tamaño máximo de un archivo?

$$(7 + 256 + 256^2) * 1 \text{ KB} = 65799 \text{ KB} = 64,25 \text{ MB}$$



# *Gestión de Espacio Libre*

- Control sobre cuáles de los bloques de disco están disponibles.
- *Alternativas*
  - Tablas de bits
  - Bloques libres encadenados
  - Indexación



# *Espacio Libre - Tabla de bits*

- Tabla (vector) con 1 bit por cada bloque de disco
- Cada entrada:
  - ✓ 0 = bloque libre              1 = bloque en uso
- Ventaja
  - ✓ Fácil encontrar un bloque o grupo de bloques libres.
- Desventaja
  - ✓ Tamaño del vector en memoria  
tamaño disco bytes / tamaño bloque en sistema archivo  
Eje: Disco 16 Gb con bloques de 512 bytes □ 32 Mb.



# Espacio Libre - Tabla de bits (cont.)

## □ Ejemplo

00111

00001

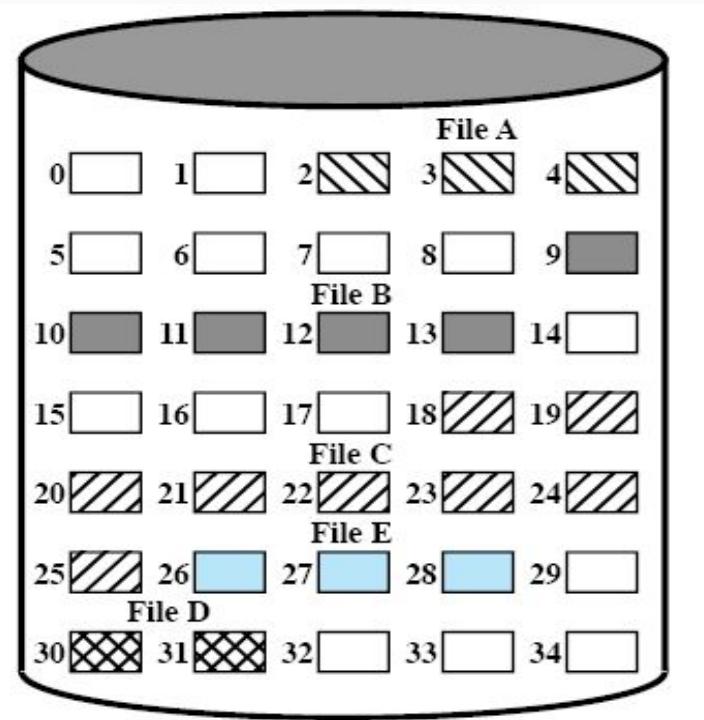
11110

00011

11111

11110

11000

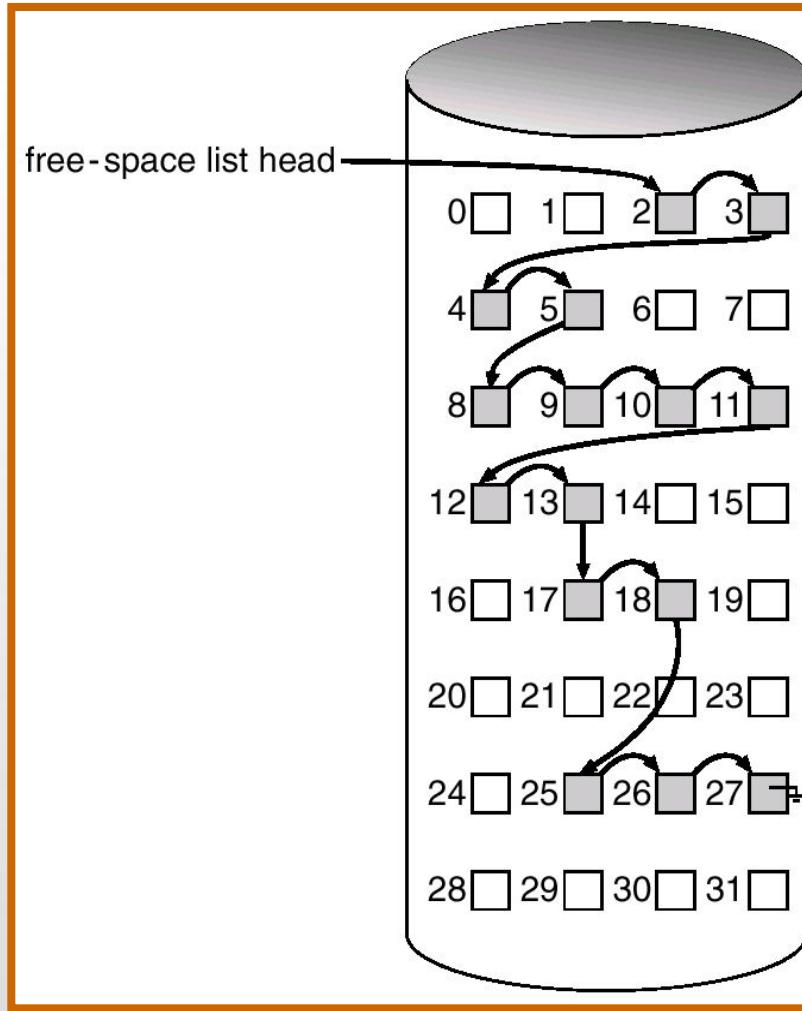


# *Espacio Libre - Bloques Encadenados*

- Se tiene un puntero al primer bloque libre.
- Cada bloque libre tiene un puntero al siguiente bloque libre
- Ineficiente para la búsqueda de bloques libres
  - Hay que realizar varias operaciones de E/S para obtener un grupo libre.
- Problemas con la pérdida de un enlace
- Difícil encontrar bloques libres consecutivos

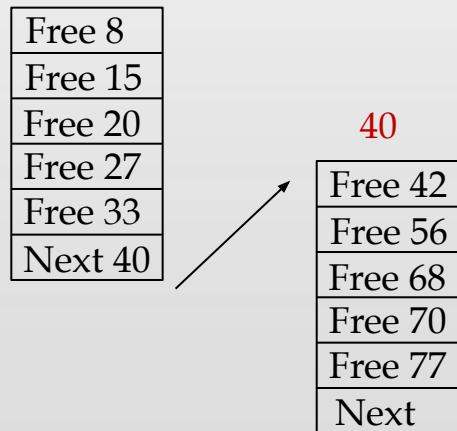


# Espacio Libre - Bloques Encadenados



# Espacio Libre - Indexación (o agrupamiento)

- Variante de “bloques libres encadenados”
- El primer bloque libre contiene las direcciones de N bloques libres.
- Las N-1 primeras direcciones son bloques libres.
- La N-ésima dirección referencia otro bloque con N direcciones de bloques libres.



# Espacio Libre - Recuento

- Variante de Indexación
- Esta estrategia considera las situaciones de que varios bloques contiguos pueden ser solicitados o liberados a la vez (en especial con asignación contigua).
- En lugar de tener N direcciones libres (índice) se tiene:
  - ✓ La dirección del primer bloque libre
  - ✓ Los N bloques libres contiguos que le siguen.  
(#bloque, N siguientes bloques libres)

Free 8/3	40
Free 15/1	
Free 20/5	
Free 27/2	
Free 33/5	
Next 40	

Free 42/3	
Free 56/4	
Free 68/1	
Free 70/5	
Free 77/9	
Next ...	

- Busca achicar estructuras
- Es compleja de mantener

