Cátedra de Sistemas Operativos

Unidad 2 – Administración de Archivos
Parte 2

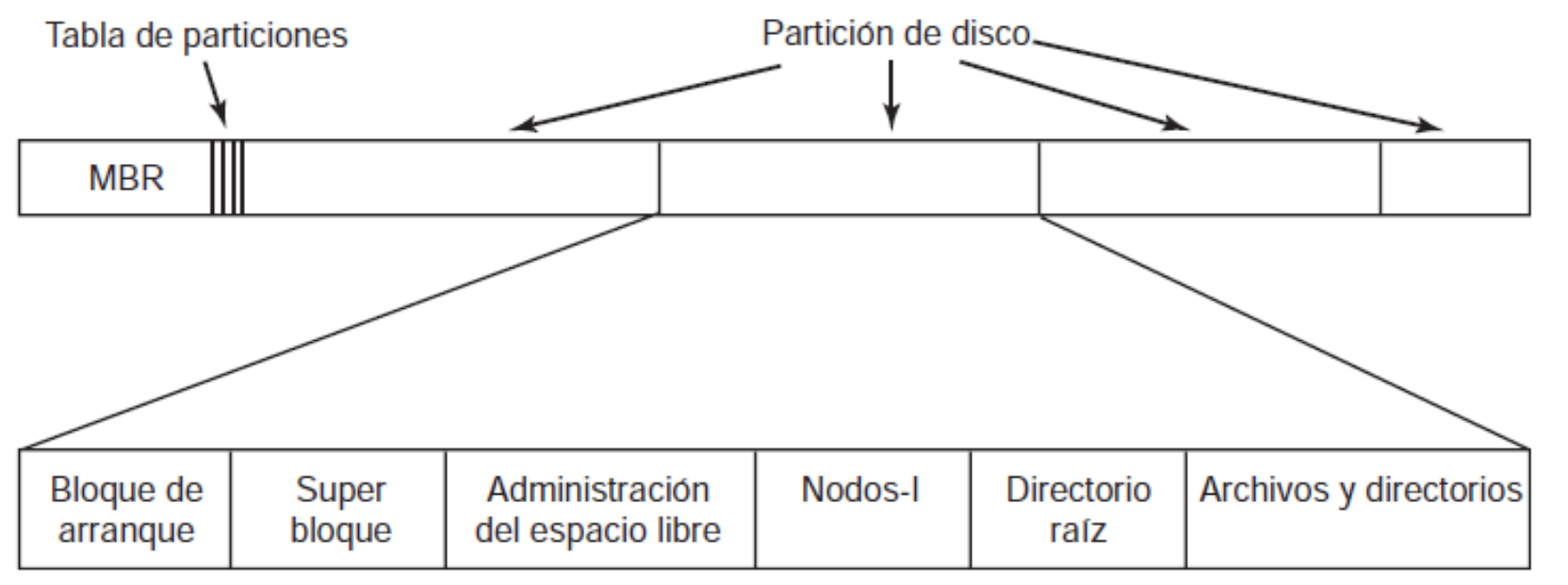
2024

Temario - Implantación del Sistema de Archivos

- Organización del sistema de archivos
- Implantación de archivos:
  - Asignación contigua
  - Asignación de lista enlazada (ligada)
  - Asignación de lista enlazada y un índice
  - Nodo-i (nodo índice)

Organización del Sistema de Archivos

- Los sistemas de archivos se almacenan en discos
- Los discos se pueden dividir en una o más particiones con sistemas de archivos independientes en cada partición



Particiones de disco

- Una partición de un disco es una unidad lógica de almacenamiento que permite dividir un disco rígido en varias partes
- Cada partición puede tener un sistema de archivos diferente
- Tipos de particiones:
  - Partición primaria → son divisiones crudas o primarias del disco (4 como máximo) o 3 primarias y una extendida
  - Partición extendida (secundaria) → se utiliza para contener múltiples unidades lógicas. Sólo puede haber UNA partición extendida por disco físico
  - Partición lógica → ocupa una parte (o toda) de la partición extendida. Puede contener un sistema de archivos (FAT, NTFS, ext4, etc.)

Partición de disco en Linux

- En GNU/Linux puede haber:
  - hasta cuatro particiones primarias, o
  - hasta tres primarias y una extendida.
- Una partición ***extendida*** es aquella cuyo contenido es a su vez particionado en varias particiones lógicas.
- Como mínimo debemos definir en una partición extendida, una partición lógica.

Partición de disco en Linux (cont.)

- Las cuatro particiones primarias/extendida en Linux, reciben los números del 1 al 4, estén presentes o no.
- Las particiones lógicas empiezan con el número 5 en adelante.
- En Linux, los discos como todos los dispositivos se representan y referencian por archivos, estos se encuentran en el directorio /dev.
- Los nombres de discos rígidos serán:
  - /dev/hd[a-h] para discos IDE
  - /dev/sd[a-p] para discos SCSI
  - /dev/ed[a-d] para discos ESDI
  - /dev/xd[ab] para discos XT

Partición de disco en Linux (cont.)

- El nombre del dispositivo se refiere al disco entero.
- La partición es un nombre de dispositivo seguido por un número de partición. Por ejemplo, /dev/hda1 es la primera partición del primer disco duro IDE en el sistema.
- Los discos IDE pueden tener hasta 63 particiones.
- Los SCSI, hasta 15.

Partición de disco en Linux (cont.)

- A la hora de instalar un Linux se necesita al menos dos particiones de disco:
  - una donde irá el sistema operativo y programas.
  - otra llamada swap o partición de intercambio (aunque también puede usar archivos swap) las particiones son más eficientes.
- Por razones de administración, copias de seguridad o pruebas, se puede utilizar más particiones de las mínimas recomendadas anteriormente.

Proceso de particionamiento de disco en Linux

- Si se necesita particionar el único disco disponible, se recomienda utilizar **fdisk**.
- Es una utilidad de la línea de comandos de Linux, que está basada en texto y es guiado por menú para administrar particiones de disco duro a través de la manipulación de las tablas de partición.
- La información sobre cómo se particiona un disco, se almacena en su primer sector.
- Este primer sector es el llamado ***registro de arranque maestro*** (MBR) del disco.
- Técnicamente, los primeros bytes del MBR contienen un código que se ejecuta en modo real y que explora la tabla de particiones, buscando la ***partición activa***, que debe contener un sistema de archivos y los archivos necesarios para iniciar la ejecución y carga del sistema operativo.

Proceso de particionamiento de disco en Linux

- Sólo una partición puede estar apuntada como ***activa***, a pesar de que puede haber más de una partición conteniendo un sistema operativo.
- Utilizando **fdisk** se puede crear un máximo de 4 particiones primarias o hasta 3 primarias y un número mayor de particiones lógicas en 1(una) partición extendida.
- En resumen, un disco rígido posee una MBR (donde hay reservado espacio para cuatro estructuras de datos), la tabla de partición y las particiones propiamente dichas.
- En el proceso de particionado se escriben los sectores que conformarán la tabla de partición (la cual contiene información acerca de la partición: tamaño en sectores, posición con respecto a la partición primaria, tipos de partición existentes, sistemas operativos instalados, etc.).

Proceso de particionamiento de disco en Linux Comando fdisk

fdisk permite crear, modificar o eliminar particiones en el disco rígido manipulando la tabla de particiones.

Sintaxis:

fdisk [opciones] [dispositivo]

dispositivo *nombre del dispositivo o disco a particionar*.

- Si no pasamos un argumento, éste seleccionará la primera unidad de disco que encuentre.
- fdisk proporciona la posibilidad de ejecutar el comando sin opciones, indicando solo el dispositivo, y la utilidad le devuelve por la salida estándar el siguiente menú de ayuda, de manera que el usuario elija la opción.
- Sin opciones ni argumentos, fdisk muestra todos los discos/particiones definidas, indicando cantidad de cilindros, sectores, nodos-i, etc.

Proceso de particionamiento de disco en Linux Comando fdisk (cont.)

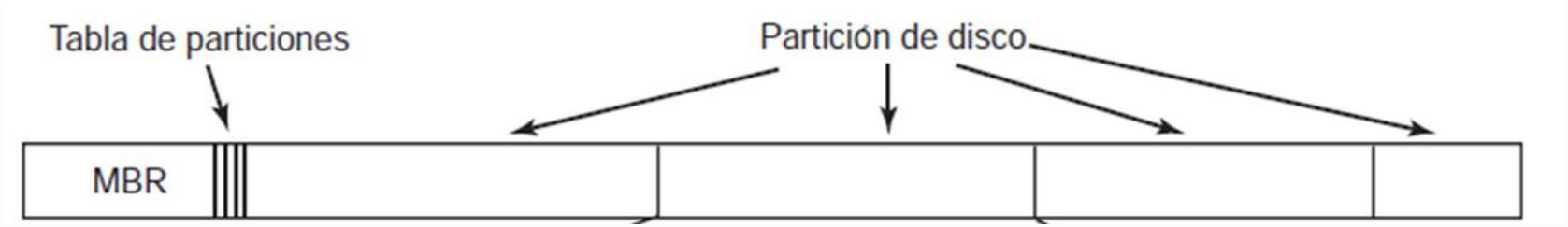
- Al iniciar el proceso de particionado podemos visualizar algunas de las siguientes opciones:
- Comando (m para ayuda): m

d	eliminar una partición.
l	lista los tipos de partición conocidos.
m	imprime el menú.
n	añadir una nueva partición.
p	imprimir la tabla de particiones.
q	salir sin guardar los cambios.
t	cambiar el id del sistema de una partición.
v	verificar la tabla de particiones.
w	escribe la tabla en el disco y salir.

Proceso de particionamiento de disco en Linux Comando fdisk - Ejercicio

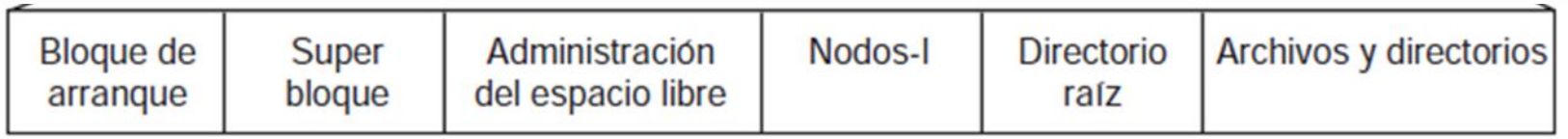
- Identificar el o los archivos que representan a los dispositivos de disco de su sistema.
- Visualizar la tabla de particiones de su disco.
- Crear una partición primaria de 2 GB aproximadamente.
- Guardar los cambios.
- Verificar que la partición acabada de crear se pueda listar en el directorio /dev.

Organización del Sistema de Archivos (cont.)



- MBR (master boot record):
  - Sector 0 del disco
  - Utilizado para arrancar la computadora
  - Contiene la tabla de particiones
- Tabla de particiones → dirección inicial y final de cada partición
- Cada partición inicia con un “bloque de arranque”
- Una partición de la tabla se marca como “activa”
- Cuando arranca la PC, el BIOS lee el MBR y lo ejecuta

Organización del Sistema de Archivos (cont.)

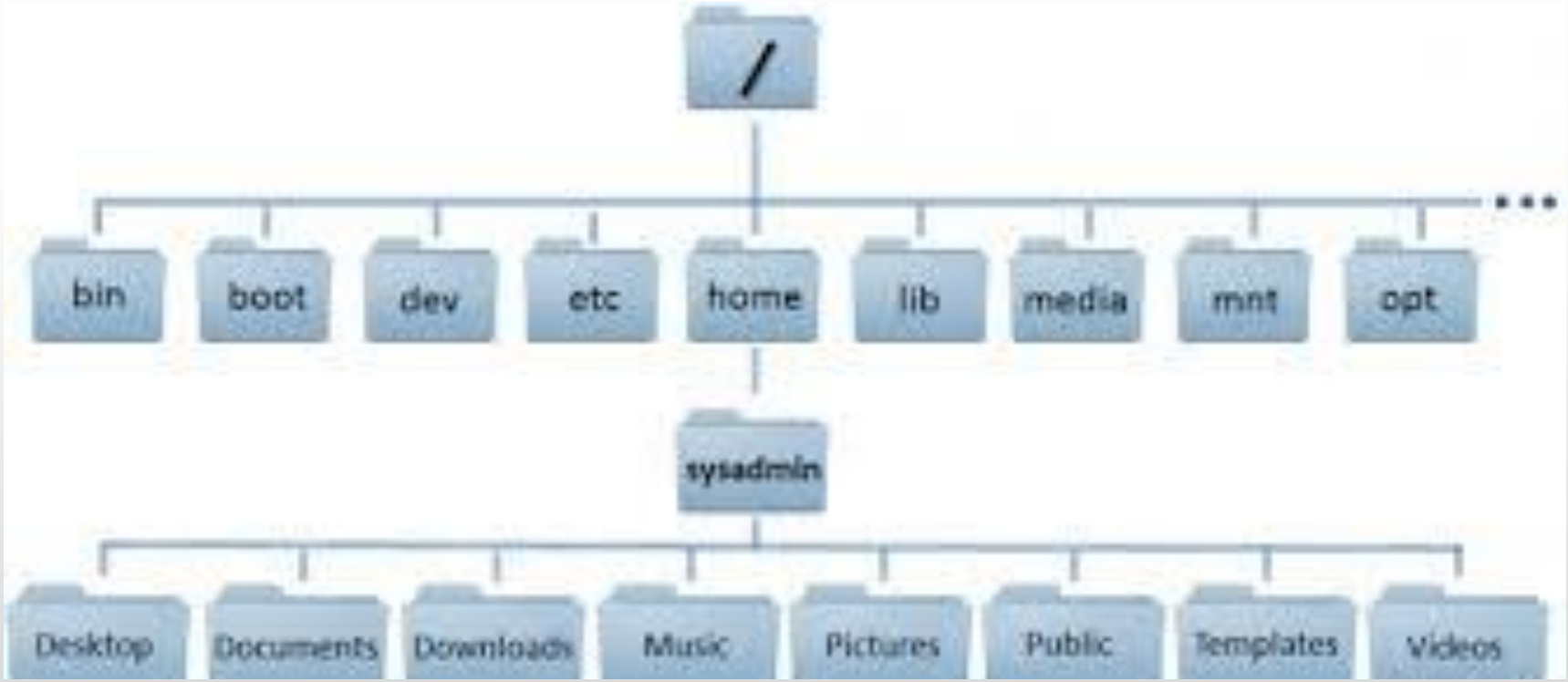


Partición del disco (Linux)

- Bloque de arranque → carga en RAM el S.O. de esa partición
- Superbloque:
  - Contiene un número mágico → identifica el tipo de sistema de archivos
  - Cantidad de bloques que tiene el sistema de archivos
- Administración del espacio libre → contiene un mapa de bits
- Nodos-i → uno por cada archivo y/o directorio
- Directorio raíz
- Archivos y directorios

Sistema de Archivos en Linux

- Estructura de árbol invertido



Detalle de cada directorio

/ es el directorio raíz

/bin comandos del sistema *(son los comandos que vamos a usar como usuario común*

/boot archivos estáticos de inicio (boot)

/cdrom punto de montaje histórico de cd-roms

/dev todos los dispositivos del Sistema

/etc archivos de configuración

/home carpetas de inicio de cada usuario

/lib librerias compartidas

Detalle de cada directorio (cont.)

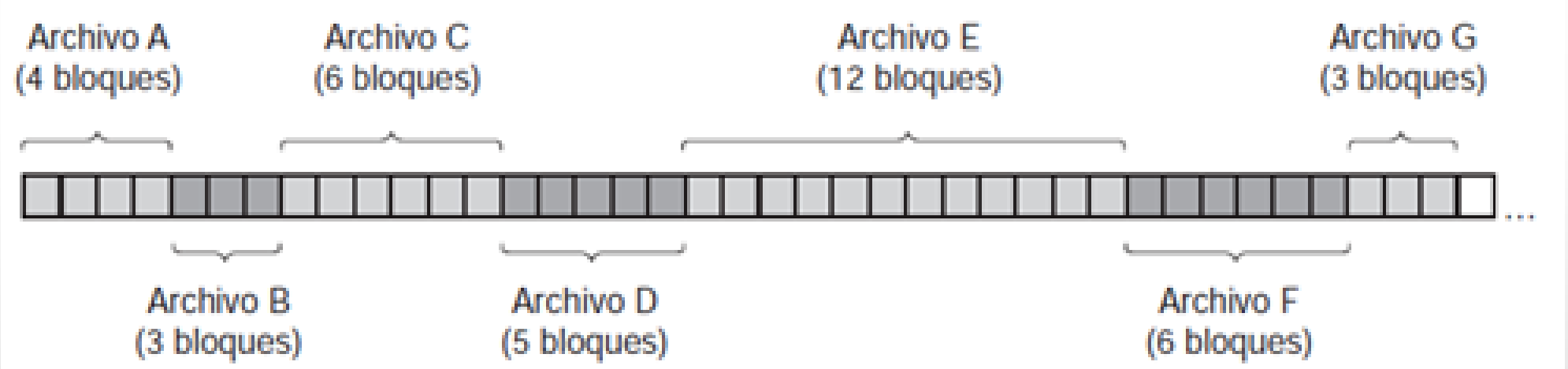
/media	unidades removibles
/mnt	puntos de montajes temporales
/opt	paquetes/instalaciones opcionales
/proc	archivos de los procesos
/root	directorio de inicio del usuario root
/run	archivos de estado de las aplicaciones
/sbin	comandos del Sistema (solo puede utilizarlo el usuario root)
/srv	datos de los servicios
/tmp	archivos temporales
/usr	binarios del usuario y datos de solo lectura
/var	archivos con datos variables

Implementación o implantación de archivos

1. Asignación contigua
2. Asignación de lista enlazada (ligada)
3. Asignación de lista enlazada y un índice
4. Nodo-i

1. Asignación contigua

- Cada archivo se almacena como una serie contigua de bloques de disco



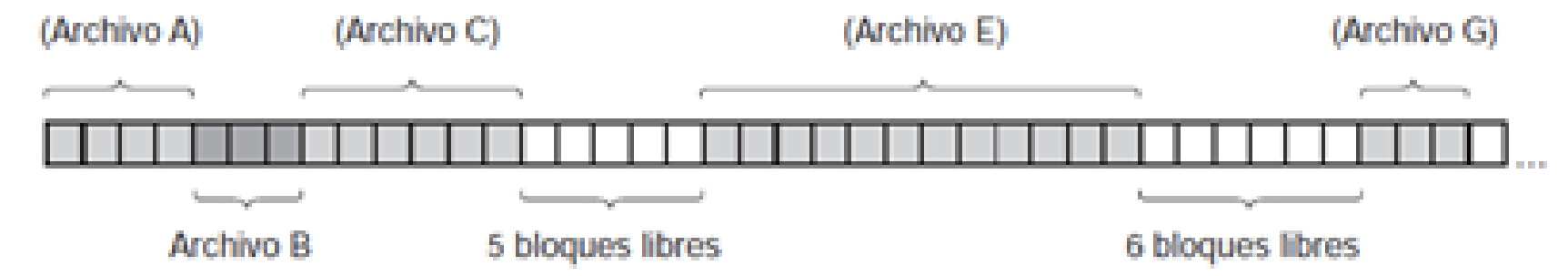
¿Cuánta memoria asignar cuando se crea el archivo?

¿Qué sucede si crece el archivo “C”?

1. Asignación contigua

Ventajas:

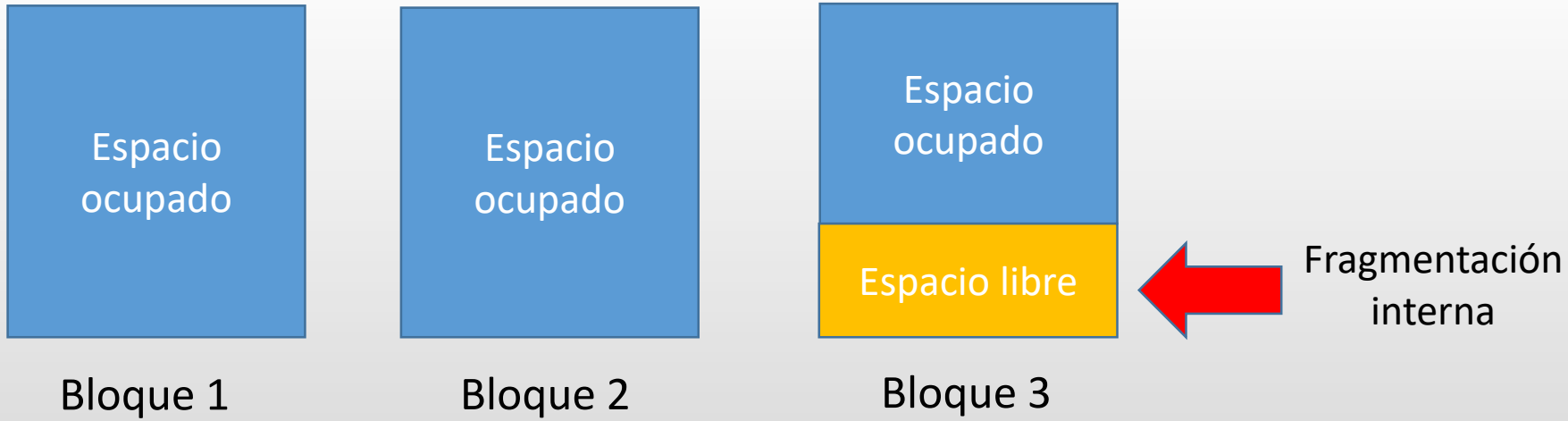
- Es simple de implementar
- Se requiere saber el número del primer bloque de cada archivo y su tamaño
- Rápida lectura del archivo



¿Cuáles serán las desventajas?

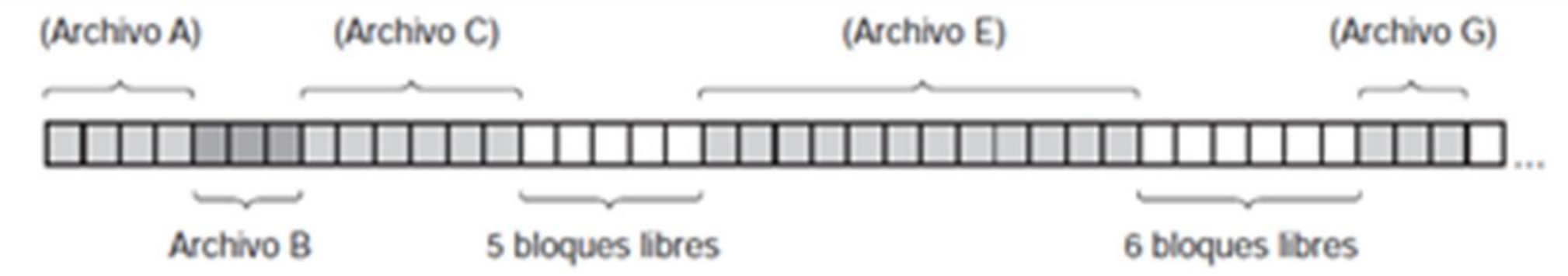
Fragmentación interna y externa

- Fragmentación interna → espacio de memoria NO utilizado y que no puede aprovechar ningún proceso o archivo
- Ejemplo: Archivo que ocupa 3 bloques de disco



Fragmentación interna y externa

- Fragmentación externa → espacio de memoria libre que sí puede ser aprovechado por un proceso o archivo

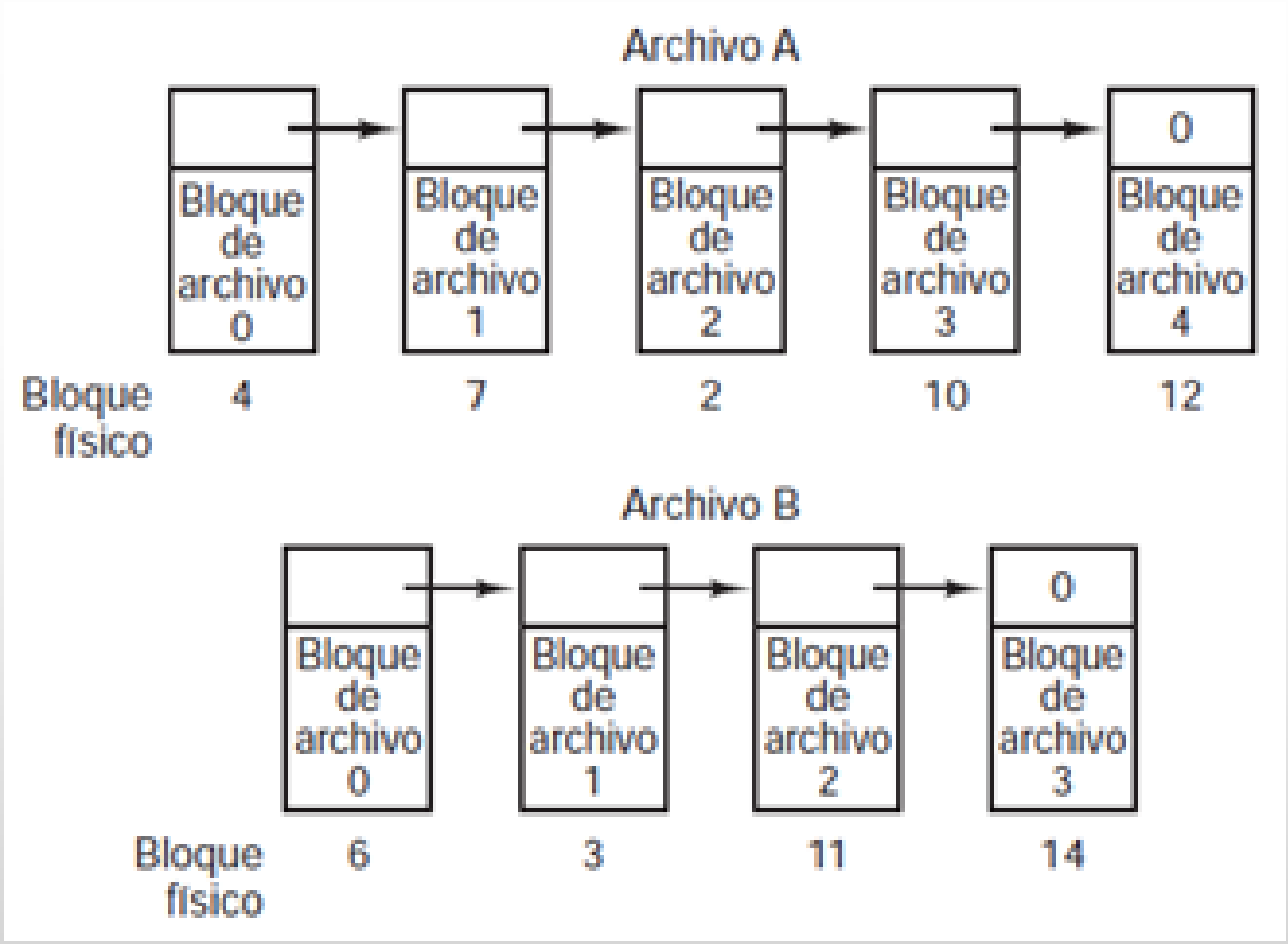


¿Se puede almacenar un archivo de 8 bloques de disco?

- Necesidad de desfragmentación o compactación

2. Asignación de lista enlazada (ligada)

- La primer palabra de cada bloque es un puntero al siguiente bloque del archivo
- El resto del bloque se utiliza para almacenar datos



2. Asignación de lista enlazada (ligada)

Consideraciones a tener en cuenta:

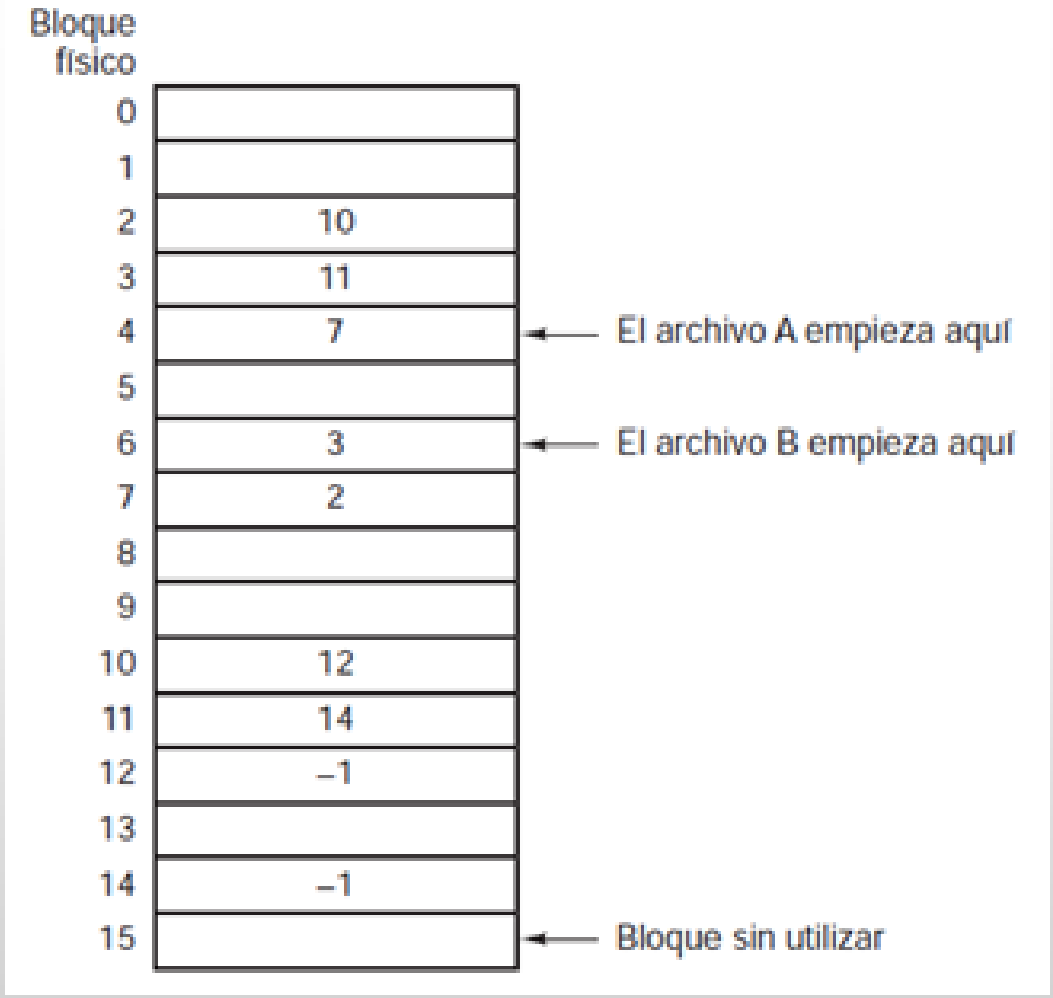
- Se elimina la fragmentación externa
- En el directorio sólo se almacena la dirección del primer bloque de disco
- El archivo puede crecer fácilmente
- Lenta lectura del archivo si el acceso es aleatorio, ya que se debe pasar por todos los bloques físicos anteriores

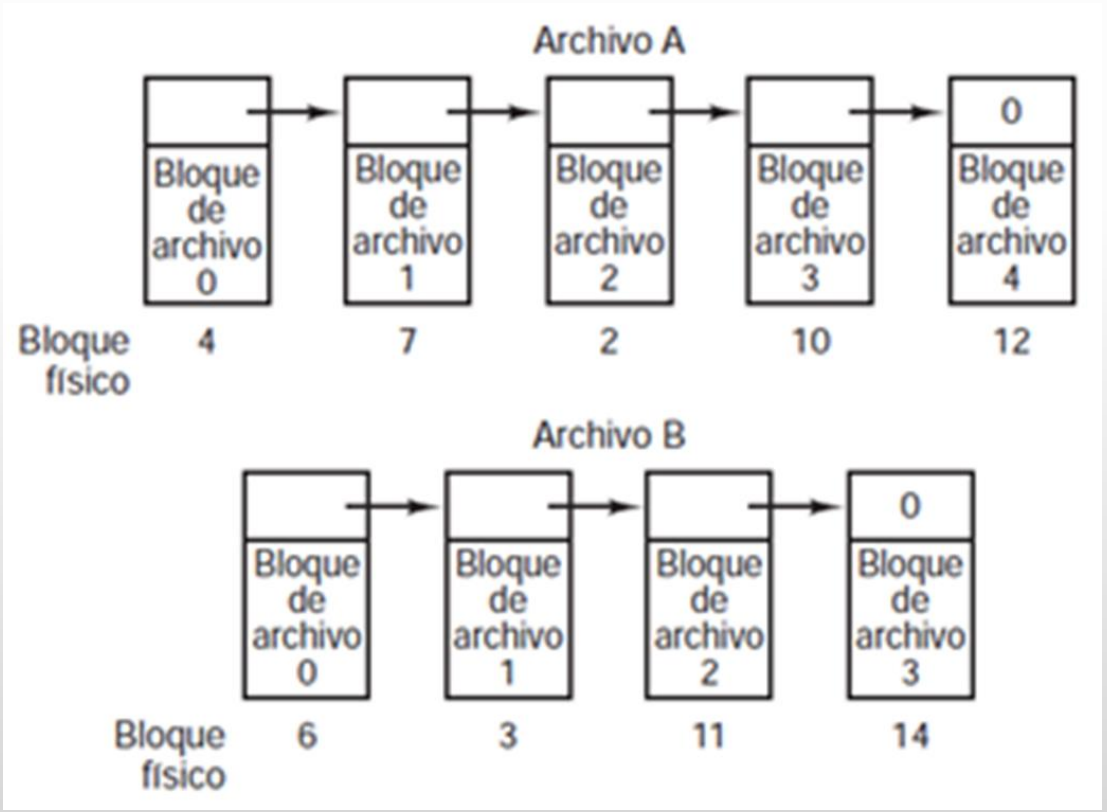
3. Asignación de lista enlazada y un índice (FAT)

- La primer palabra de cada bloque (puntero) se almacena en una tabla en la memoria
- El bloque completo se utiliza para almacenar datos
- Agiliza el acceso aleatorio, ya que la búsqueda de los bloques se hace en la tabla almacenada en RAM
- En el directorio sólo se almacena la dirección del primer bloque de disco
- FAT (file allocation table)

3. Asignación de lista enlazada y un índice

Tabla FAT





3. Asignación de lista enlazada y un índice

Desventajas:

- Toda la tabla FAT debe estar en memoria
- ¿Qué sucede si se destruye la FAT?
- No es escalable para discos grandes

- FAT-16 → las direcciones de disco tienen 16 bits
- FAT-32 → las direcciones de disco tienen 32 bits

¿Cuántas entradas tiene la tabla FAT?

Tamaño de direcciones de disco

- Con 2 bits → 4 combinaciones diferentes (2^2)

  0 0
  0 1
  1 0
  1 1

- Con 3 bits → 8 combinaciones diferentes (2^3)

  0 0 0 1 0 0
  0 0 1 1 0 1
  0 1 0 1 1 0
  0 1 1 1 1 1

Tablas FAT

- Con 16 bits → 2^{16} combinaciones diferentes

FAT-16 → direcciones de disco de 16 bits

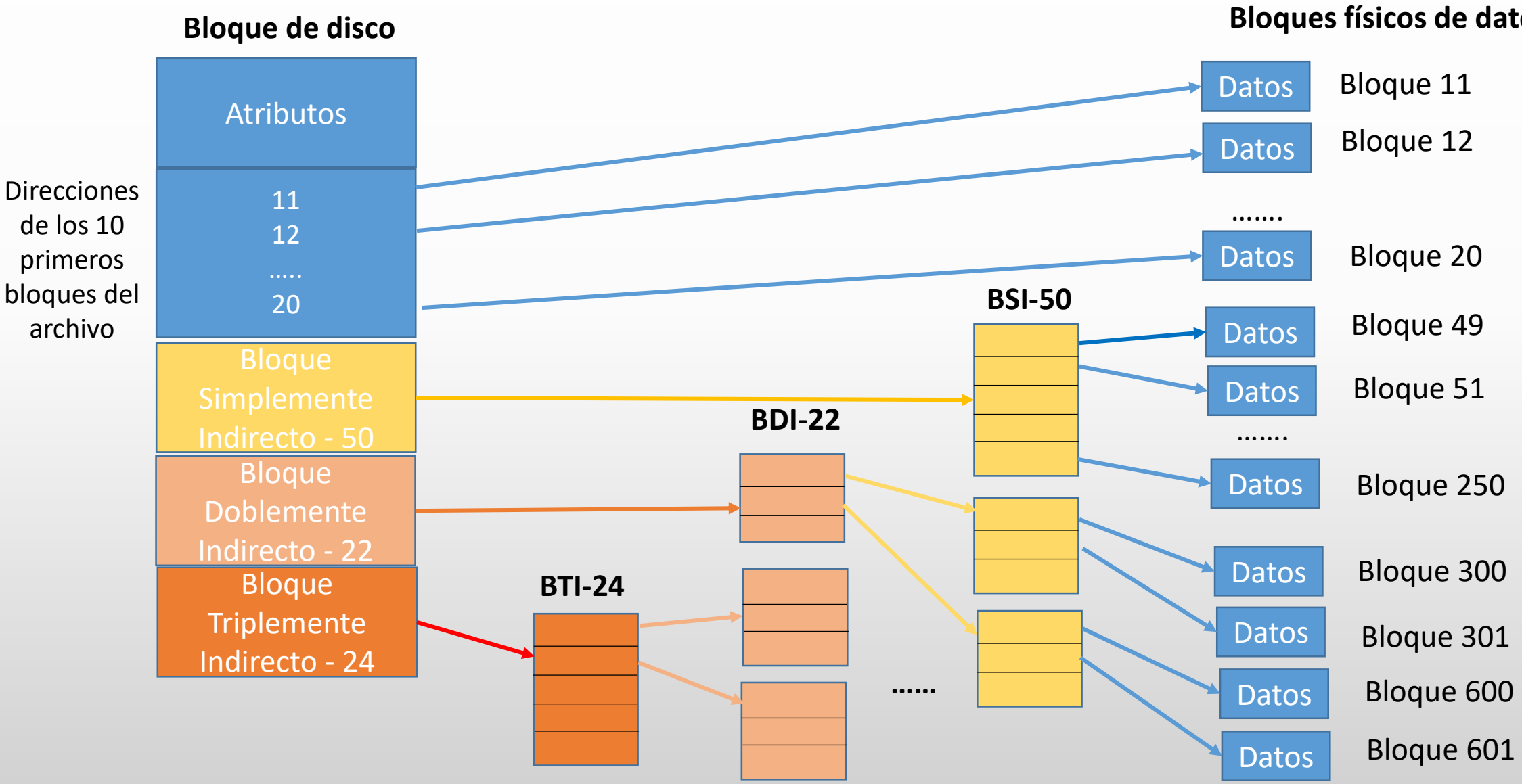
- Con 32 bits → 2^{32} combinaciones diferentes

FAT-32 → direcciones de disco de 32 bits

4. Nodo-i o i-nodo (nodo índice)

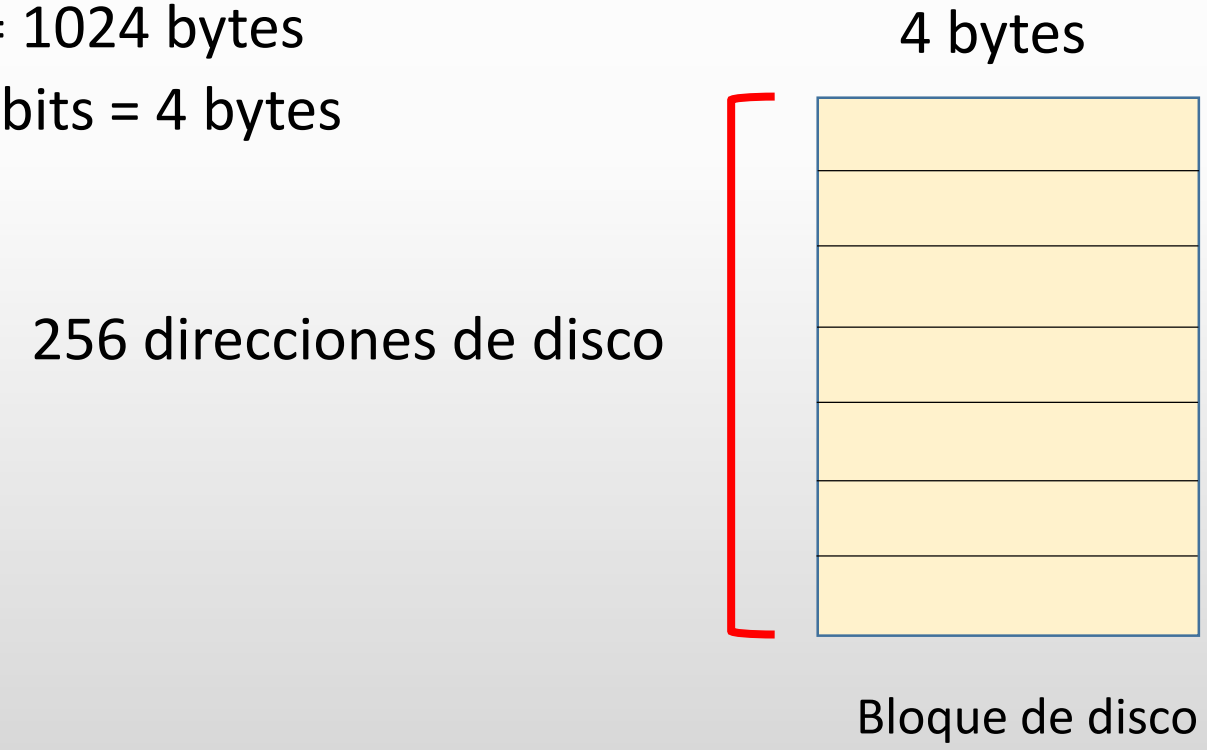
- El i-nodo es un bloque de disco que contiene los atributos de un archivo y las direcciones de los bloques de disco de los datos
- Todo archivo o directorio en Linux tiene asociado un número de i-nodo
- El i-nodo es un número entero
- El i-nodo es una estructura de datos almacenada en un bloque de disco
- La información de qué bloques pertenecen a cada archivo está descentralizada en cada i-nodo, esto lo hace muy robusto
- Sólo es necesario llevar a RAM la información del i-nodo, cuando se abre un archivo y/o directorio

4. Nodo-i o i-nodo (nodo índice)



4. Nodo-i o i-nodo (nodo índice)

- Si tenemos bloques de disco de **1 KByte** y manejamos direcciones de disco de **32 bits**
  - 1 bloque → 1 KB = 1024 bytes
  - Direcciones de 32 bits = 4 bytes



4. Nodo-i o i-nodo (nodo índice)

- Si tenemos bloques de disco de **2 KByte** y manejamos direcciones de disco de **32 bits**
  - 1 bloque → 2 KB = 2048 bytes
  - Direcciones de 32 bits = 4 bytes

512 direcciones de disco

La cantidad de direcciones que se puede almacenar en un bloque de disco, depende del tamaño del bloque y de la cantidad de bytes de la dirección

¿Dudas o Inquietudes?