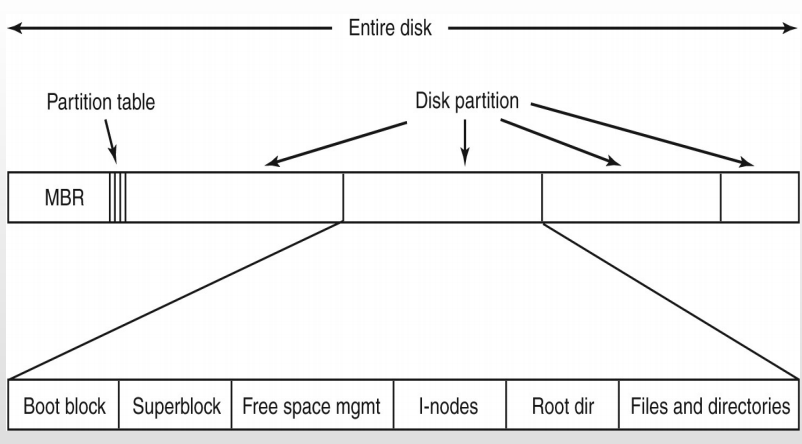
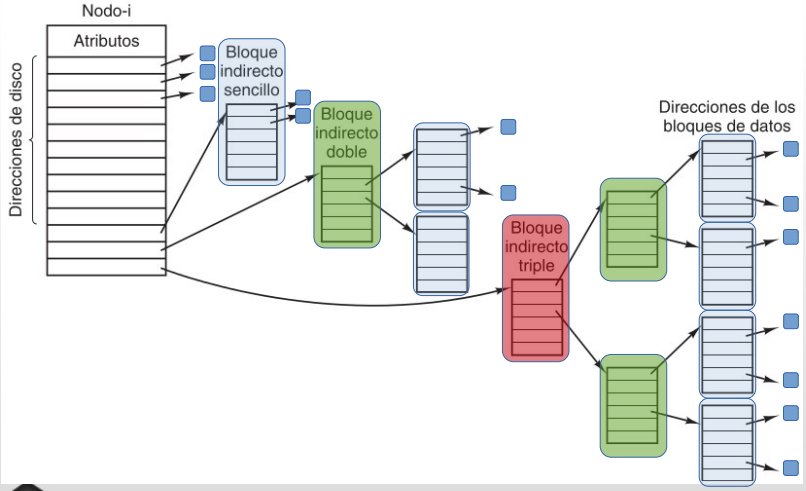
**Unix – Manejo de archivos**

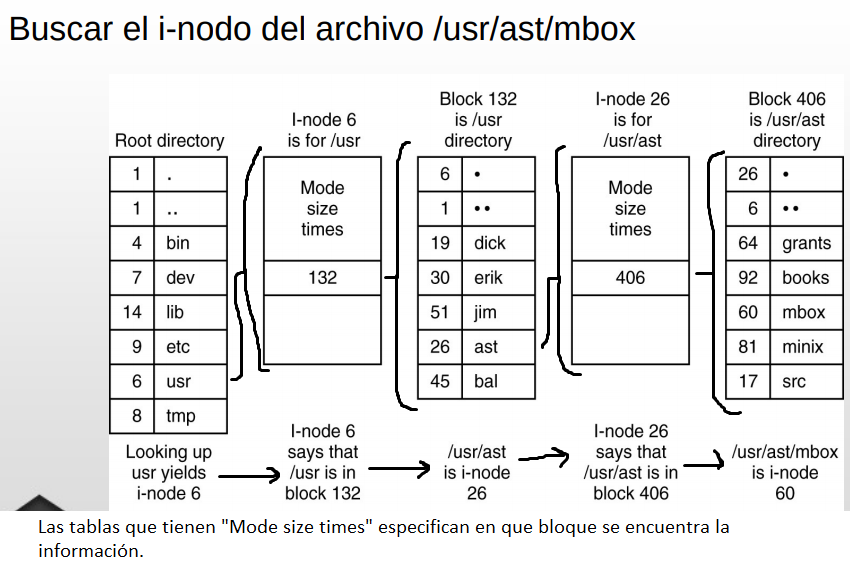
* Tipos de archivos
  + Archivo común.
  + Directorio.
  + Archivos especiales (dispositivos).
  + Named pipes (comunicación entre procesos).
  + Links (comparten el i-nodo, solo dentro del filesystem).
  + Links simbólicos (como si fueran accesos directos, tiene i-nodo propio, para filesystems diferentes).

**Unix – Estructura del volumen**

* Cada disco físico puede ser dividido en uno o más volúmenes (particiones). Cada volumen contiene un filesystem. Cada filesystem contiene.
  + Boot block: código para bootear el S.O
  + Superblock: atributos sobre el filesystem que incluyen que bloques/clusters están libres.
  + Tabla de Inodos: tabla que contiene todos los inodos.
    - Inodo: estructura de control que contiene la información clave de un archivo.
  + Bloques de datos: de los archivos.
* 

**Unix - INODO (Como se ve y un ejemplo de búsqueda)**

****

****

**Windows – File Systems soportados**

* CD-ROM filesystem (CDFS) 🡪 CD
* Universal Disk Format (UDF) 🡪 DVD, Blu-Ray
* File Allocation Table
  + FAT12
  + FAT16 🡪 nombres cortos de archivos
  + FAT32 🡪 nombres largos de archivos pero no soportados en MS-DOS
* New Technology File System (NTFS)

**Windows – FAT**

* FAT es un sistema de archivos utilizado originalmente por DOS y Windows 9x
* Por qué Windows aun soporta FAT?
  + Proporciona compatibilidad con otro SO en sistemas multiboot.
  + Permite upgrades desde versiones anteriores.
  + Para formato de dispositivos como diskettes.
* Las distintas versiones de FAT son diferenciadas por un número que indica la cantidad de bits que son usados para identificar diferentes bloques o clusters.
  + FAT12
  + FAT16
  + FAT32
* Es utilizado un mapa de bloques del FILESYSTEM, llamado FAT.
* La FAT tiene cantidad de entradas = cantidad de bloques.
* La FAT, su duplicado, y el directorio raíz se almacenan en los primeros sectores de la partición.
  + 
* En FAT se utiliza la asignación encadenada.
* El puntero al próximo bloque está en la FAT y no en los bloques.
* Los bloques libres y dañados tienen códigos especiales.

**Windows – FAT12**

* FAT12 utilizan 12 bits para la identificación del sector, se limita a 2^12 (4096) sectores.
  + Windows utiliza tamaños de sector que van desde los 512 bytes hasta los 8kb, esto limita el tamaño total del volumen a 32 MB 🡪 4096 sectores \* 8 KB
  + FAT12 se usa como filesystem para diskettes de 3,5 y 12 pulgadas que son capaces de almacenar hasta 1,44 MB de datos (probablemente lo que pesen 3 o 4 resumenes mios de la materia).

**Windows – FAT16**

* FAT16 usa 16 bits para identificar cada sector. Puede haber 2^16(65.536) sectores en un volumen.
  + El tamaño del sector varía entre 512 bytes hasta los 64 KB lo que limita el tamaño máximo de la partición a 4 GB (64 KB \* 65.536)
  + El tamaño de sector dependía del tamaño de la partición al formatearla.

**Windows – FAT32**

* FAT32 fue el filesystem más reciente de la línea esa (posteriormente salió exFAT que algunos lo conocen como FAT64).
* FAT32 usa 32 bits para la identificación de sectores, PERO reserva los 4 bits superiores. Efectivamente se usan 28 bits para identificar.
  + El tamaño de sector en FAT32 puede ser de hasta 32KB con lo cual tiene una capacidad teórica de direccionar particiones de hasta 8 TB.
  + El modo de identificación y acceso a los sectores lo hace de forma más eficiente que FAT16. Con tamaño de sector de 512 bytes, puede direccionar volúmenes de hasta 128 gb.

**Windows – NTFS**

* NTFS es el filesystem nativo de Windows desde Windows NT.
* Usa 64 bits para referenciar sectores.
  + Teóricamente permite tener volúmenes de hasta 16 Exabytes (16 billones de GB).
* Porqué usar NTFS en lugar de FAT?. A pesar de que FAT es simple y más rápido para ciertas operaciones, NTFS soporta:
  + Tamaños de archivo y de disco mayores.
  + Mejora performance en discos grandes.
  + Nombres de archivos de hasta 255 caracteres.
  + Atributos de seguridad.
  + Es transaccional.