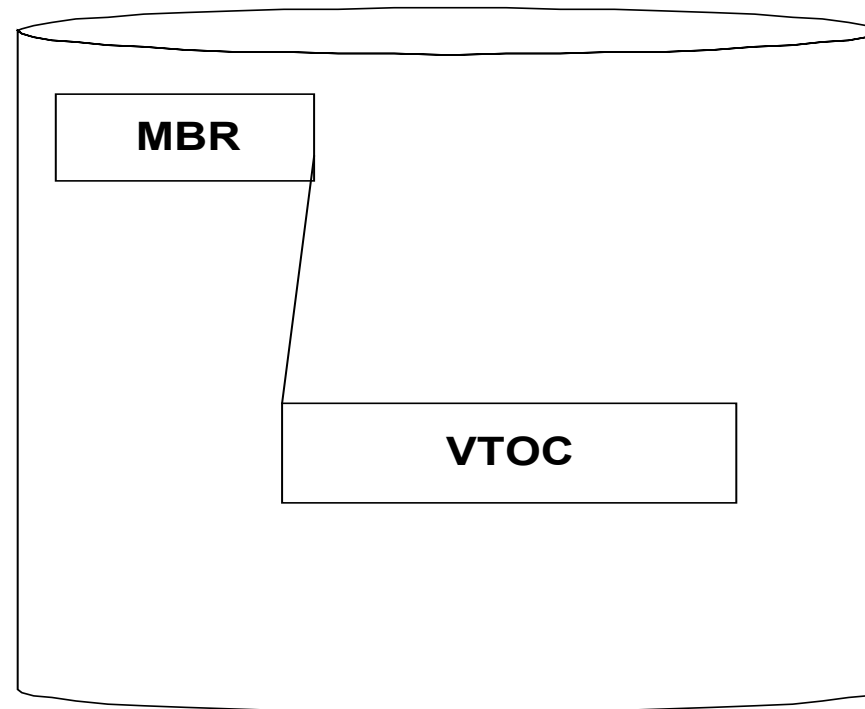


# Administración de la Información (FS)

- Objetivos
- Administración de archivos y espacios en dispositivos compartidos
- Administración de los accesos a los archivos
- Directorios

# Directorio

- Directorio (de 1er. Nivel)

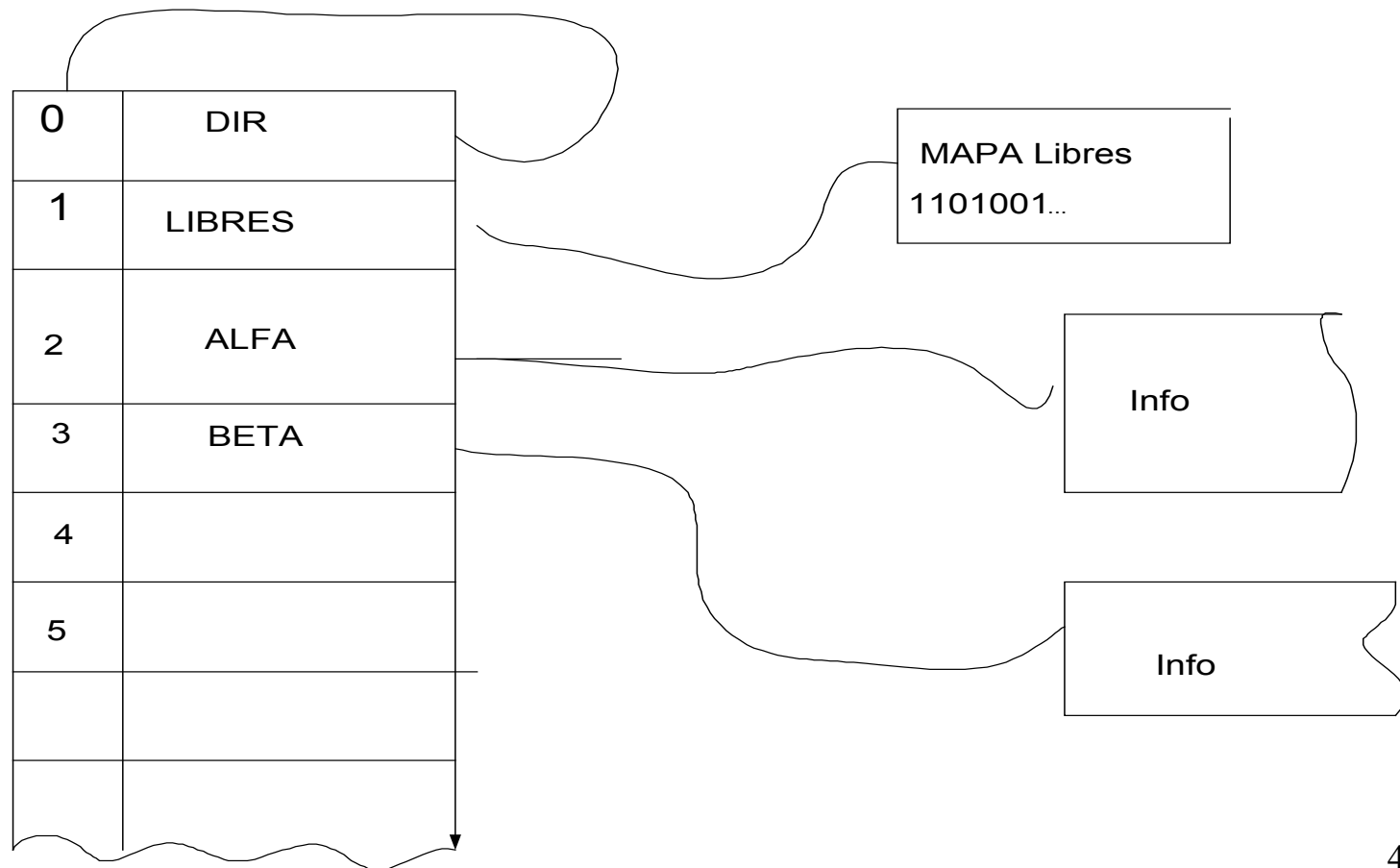


- VTOC: Volume table of contents

# Cada entrada del Directorio

- En la VTOC o Directorio se encuentra la información que caracteriza a cada archivo
  - el nombre,
  - ubicación,
  - longitud (cantidad de registros del archivo o unidad de almacenamiento),
  - longitud del registro lógico,
  - longitud del registro físico,
  - formato de registros (fijo, variable, etc.),
  - organización (secuencial, indexada, al azar, etc.),
  - fecha de creación,
  - fecha de expiración,
  - último cambio,
  - derechos de acceso, (permisos - concurrencia)
  - extensiones.

# Ejemplo de Directorio

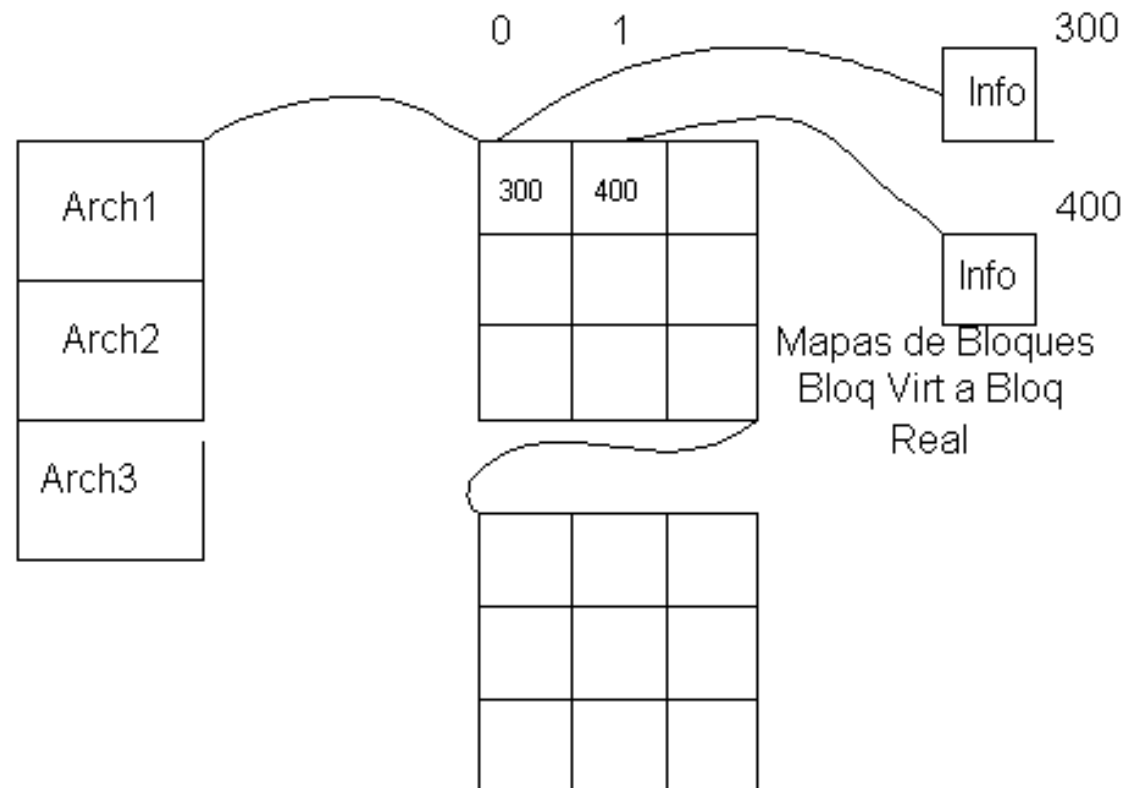


# Ocupación Espacio

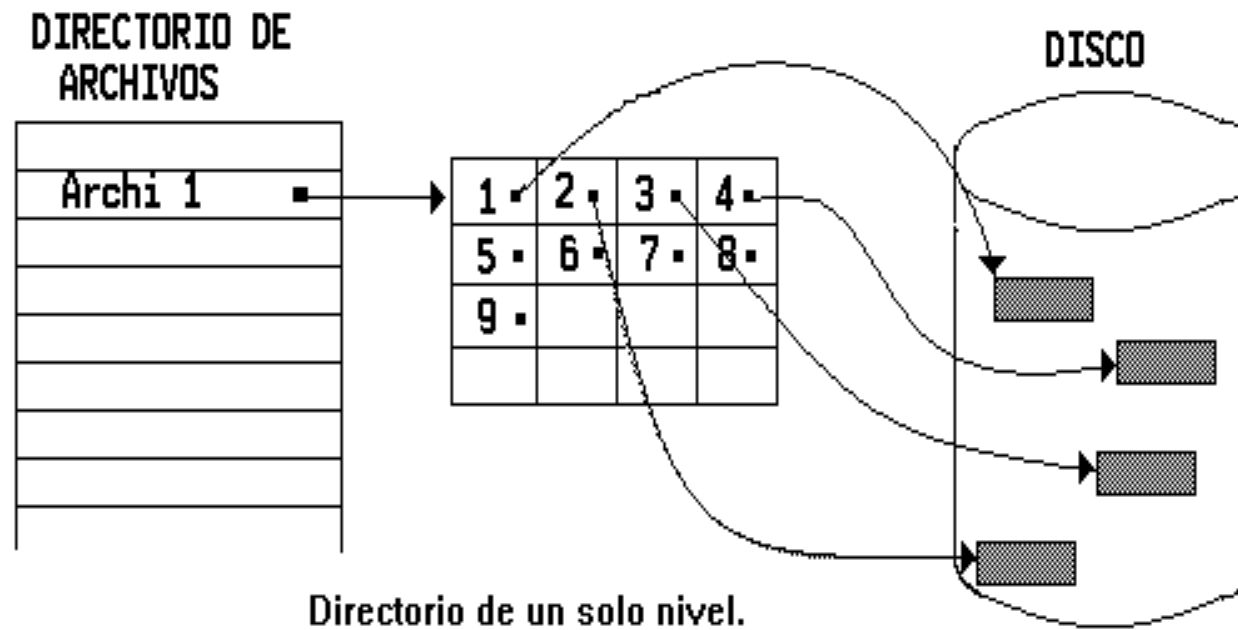
- Sin extensión ➔ Contigua
- Con extensiones (limitada) ➔ ocupan más de una entrada
- Políticas
  - 1er. Lugar libre
  - Mejor lugar
    - ➔ **Compactación (volvimos ...)**

# Ocupación Espacio

- **Dinámica (por mapeo de bloques) (no más compactación)**



# Estructura de Directorios



# Catálogo de Usuarios por Volumen

## **Volumen**

HD1

## **Usuarios**

JOSE

PEDRO

JUAN

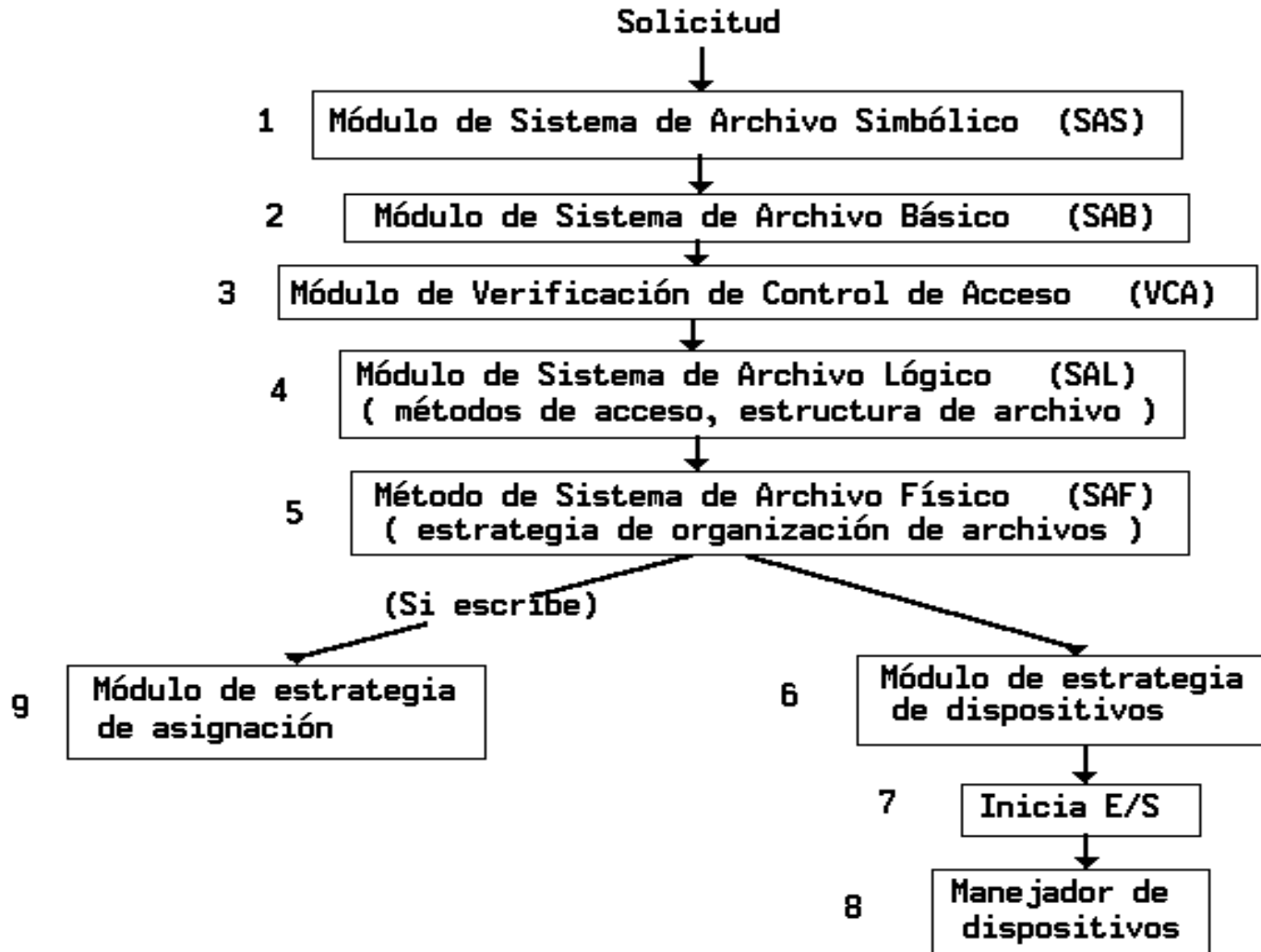
HD2

ANA

MARIA

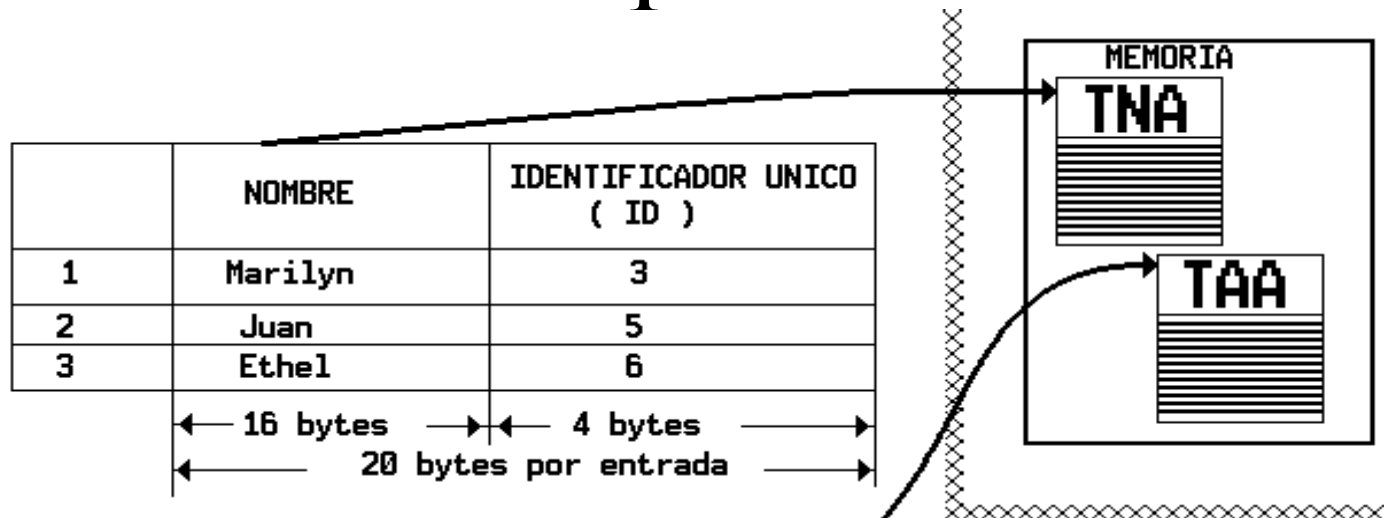


# Modelo General de Acceso



Modelo jerárquico de un sistema de archivos.

# Evitar Múltiples Accesos



a) Directorio de archivo simbólico

ID	LONGITUD DE REGISTRO LOGICO	CANTIDAD DE REGISTROS LOGICOS	DIRECCION AL PRIMER BLOQUE FISICO	DIRECTORIO Y CONTROL DE ACCESO
1	----	----	0	Dir. Básico
2	20	3	1	Dir. Simbólico
3	80	10	2	Todos Leen
4	1000	3	3	Libre
5	500	7	6	Todos Leen
6	100	30	12	MARTA Lee JUAN Lee/Graba
7	1000	2	10	Libre
8	1000	1	15	Libre

b) Directorio de archivo básico

# Modelo General de Acceso

- Búsqueda en Catálogo (SAS y SAP o TNA TAA)
- Control de Accesos (VCA) (Permisos)
- Cálculo dirección Lógica (SAL)
- $DL = (N \text{ Reg} - 1) \times \text{Long. Lógica} (*)$
- Cálculo Dirección Física (\*\*)
- $DF = [DL / \text{Long Bloq}] \text{ (Nro. de Bloque)}$
- $\text{Resto} = [DL / \text{Long Bloq}] \text{ Byte dentro del Bloque}$

# Cálculos

- Supongamos Reg Lóg de 100 Bytes
- Reg Físicos de 300 Bytes
- Buscamos Reg Lóg 3
- $DL = (3 - 1) \times 100 = 200 (*)$
- $DF = [ 200 / 300 ] = 0 \quad (**)$
- $Resto = [ 200 / 300 ] = 200$

# Método General de Acceso

- (\*) Es lo que llamamos Método de Acceso
- (\*\*) Verificamos si la Información ya está en memoria y si el buffer que contiene la información está en memoria (si no page-fault)
- Asignación de Espacio (MEA)
- Lanzamiento de E/S (MEP)

# TAA

Identificación	Long Reg. lógico	Long. Reg. físico	Formato	Organización
Permisos	Concurrencia	Lista de Procesos que lo están usando		

- Tabla de Archivos Activos (TAA).

# Directorio de 2 niveles

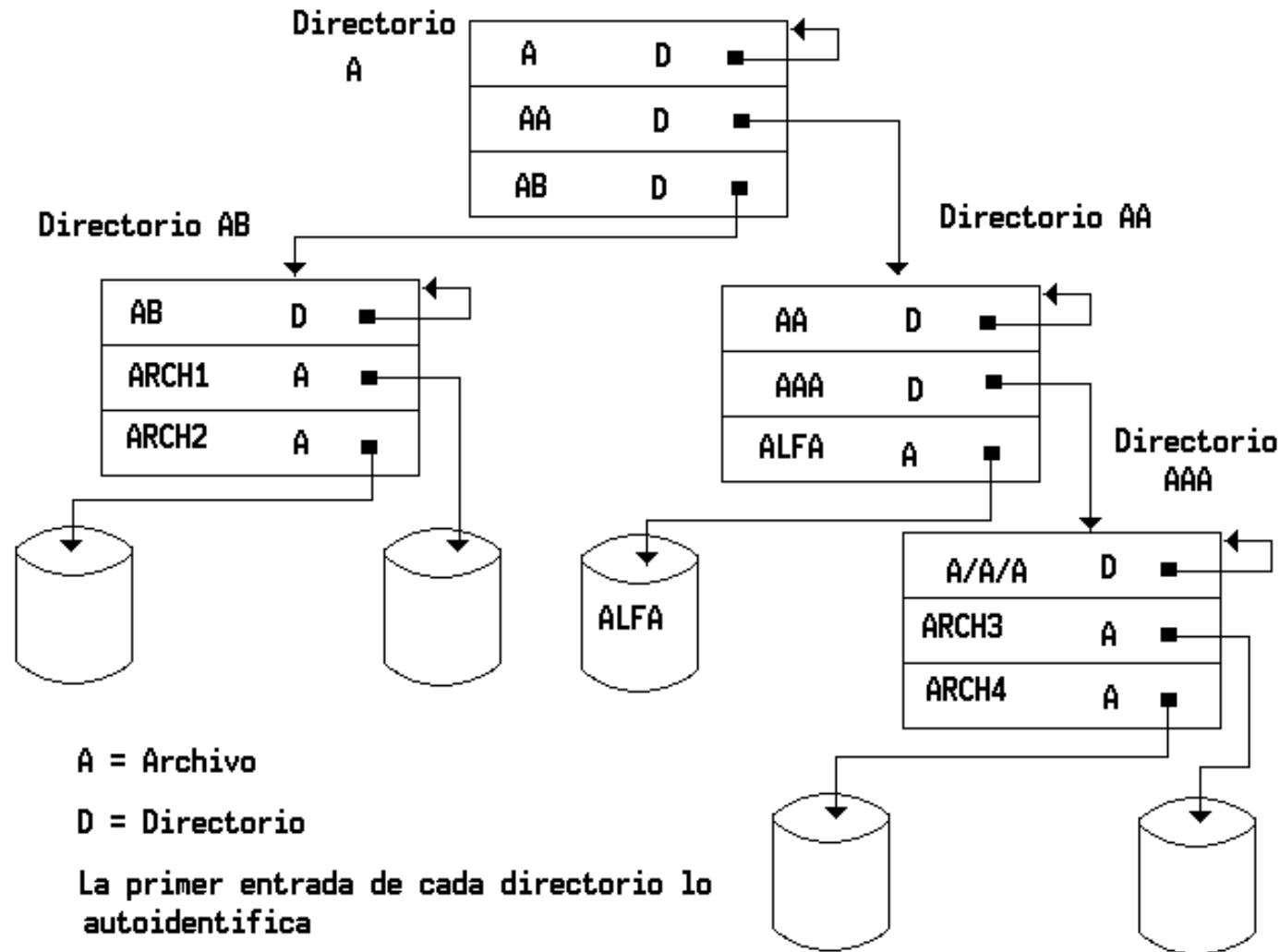
Catálogo Usuarios (Simbólico o Maestro)

Nombre	Dirección	Longitud
Simbólico	0	1
Jose	1	3
Maria		

Catálogo de Archivos (Básico o Dir Usuarios)

Nombre	Dirección	Resto de la Info
Dato1	7	
Dato2	807	

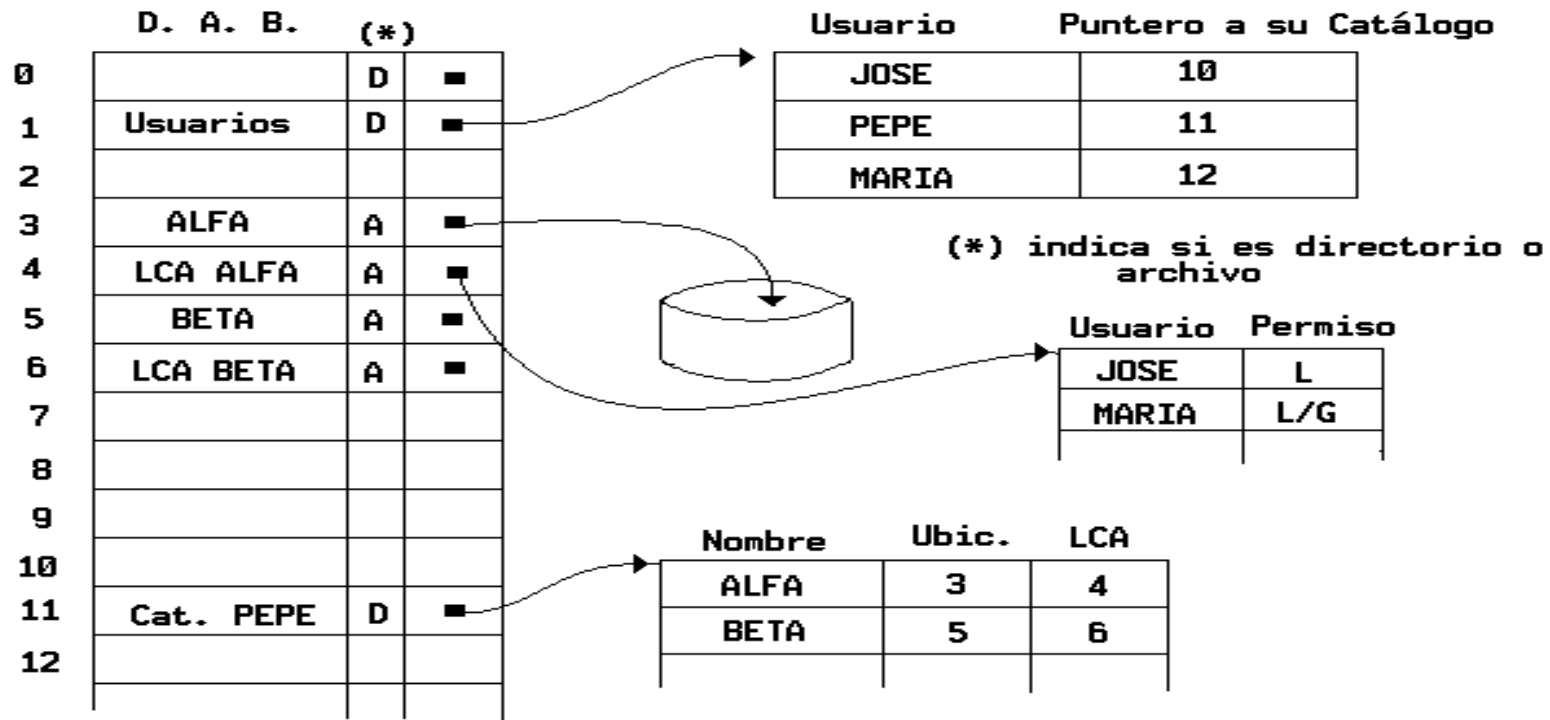
# Directorios de 2 Niveles



Directorios de varios niveles.

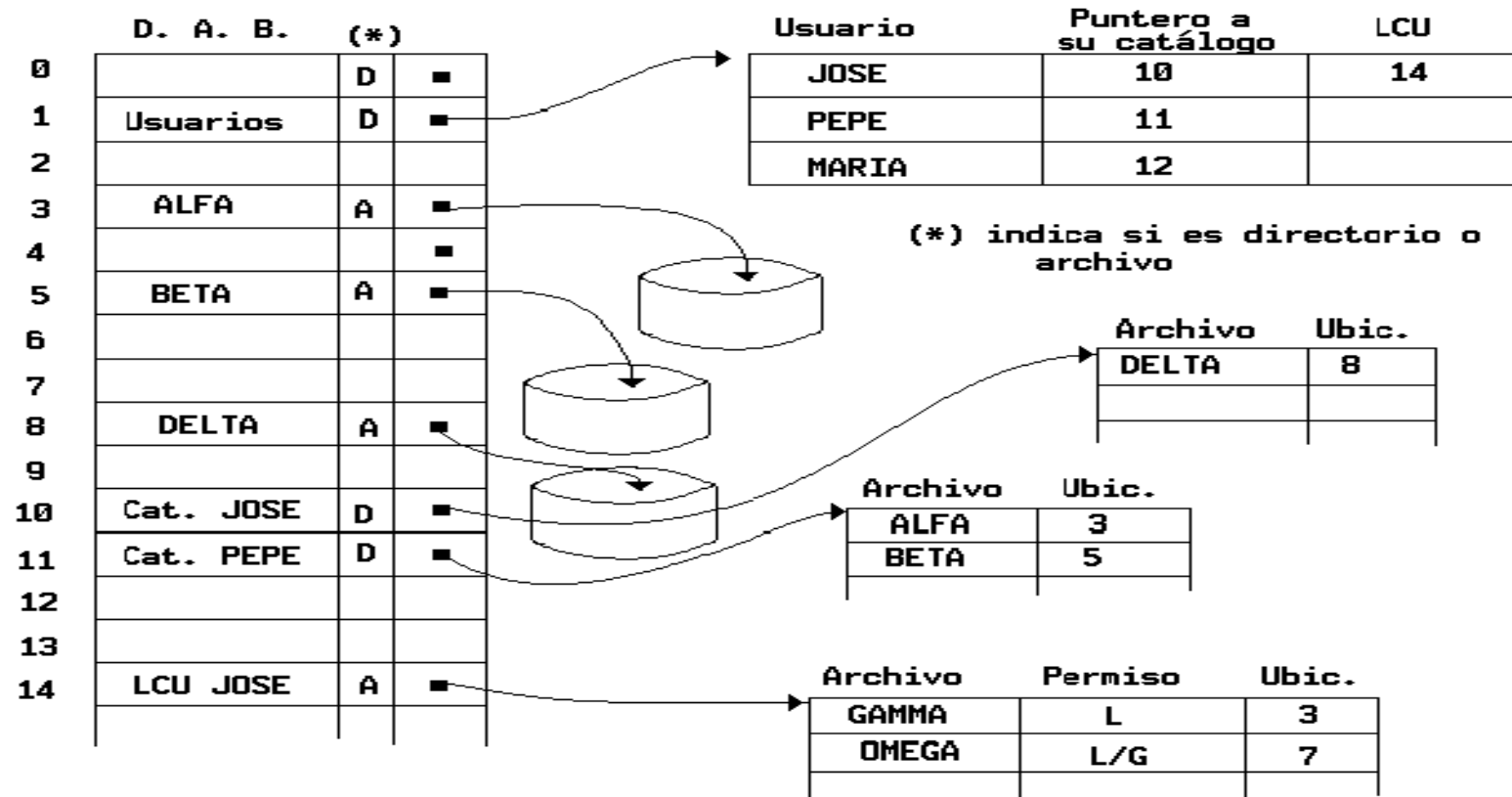


# Lista de Control de Accesos (LCA)



- Sistema de Lista de Control de Acceso (LCA).

# Lista de Control de Usuarios (LCU)



- Sistema de Lista de Control de Usuario (LCU).

# Simbólico

- CALL SAS (READ, "JUAN", 4, 1200)
- Donde estamos pidiendo que se lea el registro lógico número 4 del archivo "JUAN", para colocar su contenido en la dirección 1200 de memoria principal.
- Debe devolver un 5 (ID JUAN)

# Básico

- CALL SAB (READ, 5, 4, 1200)
- Donde todos los parámetros son iguales a la llamada del módulo SAS, a excepción del segundo parámetro que constituye el identificador que le pasó el SAS.

# VCA (Verificación de control de Acceso)

- Supongamos que la entrada del archivo Juan (que era la 2da entrada en la TNA y que estaba almacenada como 5ta entrada del DAB) fue guardada en la entrada número 67 de la TAA (tabla de archivos activos)
- La invocación al módulo de Verificación de Control de Acceso será entonces de la siguiente forma:
- `CALL VCA (READ, 67, 4,1200)`
- El 67 corresponde a la entrada de la TAA del archivo "JUAN" y los demás son exactamente los mismos parámetros de la llamada al módulo anterior.

# SAL (Sistema Archivos Lógicos)

- CALL SAL (READ, 67, 4,1200)
- El Sistema de Archivo Lógico convierte el pedido de un registro lógico en el pedido de una secuencia de bytes lógicos, la cual se entrega al Sistema de Archivo Físico (SAF).

# SAF – MEA - MEP

- SAF (determina dirección física o sea número de bloque)
- MEA (determina si la dirección obtenida está dentro de los límites del archivo)
- MEP (si la información no está en le buffer de memoria prepara la información para lanzar la operación de E/S física, (Nro de RF, Dir del Buffer de memoria))

# Resumen

- 1) **Sistema de archivos simbólicos (SAS)**: transforma el nombre del archivo en el identificador único del Directorio de archivos. Utiliza la Tabla de nombres activos (TNA) y el directorio de archivos simbólico (DAS).
- 2) **Sistema de archivos básico (SAB)**: copia la entrada de la VTOC en memoria. Utiliza el directorio de archivos básicos (DAB) y la Tabla de archivos activos (TAA).
- 3) **Verificación de control de acceso (VCA)**: verifica los permisos de acceso al archivo.



# Resumen

- **4) Sistema de archivo lógico (SAL):** transforma el pedido lógico en un hilo de bytes lógicos.
- **5) Sistema de archivo físico (SAF):** calcula la dirección física.
- **6) Módulo de estrategia de asignación (MEA):** consigue espacio disponible en el periférico (casos de grabación).
- **7) Módulo de estrategia de periférico:** transforma la dirección física según las características exactas del periférico requerido

# Ejemplo con JCL

- \* Trab1 Pedro
- \* Archivo 8,volumen1,JOSE/ALFA,G
- \* Ejectuar Prog1
  
- Prog1
- ...
- Read 8
- ...

# Ejemplo

- BCP
- Trab1 ...SS... Dispositivo Archivo
- volumen1 8  
**JOSE/ALFA**  
**apunt. TNA**  
**buffer (nro RF)**

# Ejemplo

- 1. Se busca en dispositivos volumen1
- Si está se asocia a Trabl
- 2.Se busa en TNA/TAA **JOSE/ALFA**
- Si está se controla permiso
- Sino se busca JOSE y luego ALFA y se cargan TNA/TAA
- Se asocia 8 con **JOSE/ALFA**

# Ejemplo Ejecución

- Open (puede hacer todo lo anterior y asocia buffer a BCP)
- Controla Permisos
- READ
- CDL
- CDF (si está pasa puntero)
- Sino Lanzamiento de E/S Física

# Ejemplo en Cluster

- `#PBS -N TEST_MPI`
- `#PBS -l nodes=40:ppn=2`
- `#PBS -S /bin/bash`
- `#PBS -q verylong`
- `#PBS -m ae`
- `cd /home/robevi/test_intelmpich`
- `echo `hostname``
- `echo `date``
- `echo `pwd``
- `echo `cat $PBS_NODEFILE``
- `cat $PBS_NODEFILE > $PBS_O_WORKDIR/machines`
- `/opt/mpich/intel/bin/mpirun -nolocal -np 80 -machinefile $PBS_NODEFILE xhpl`

# Algunso FS

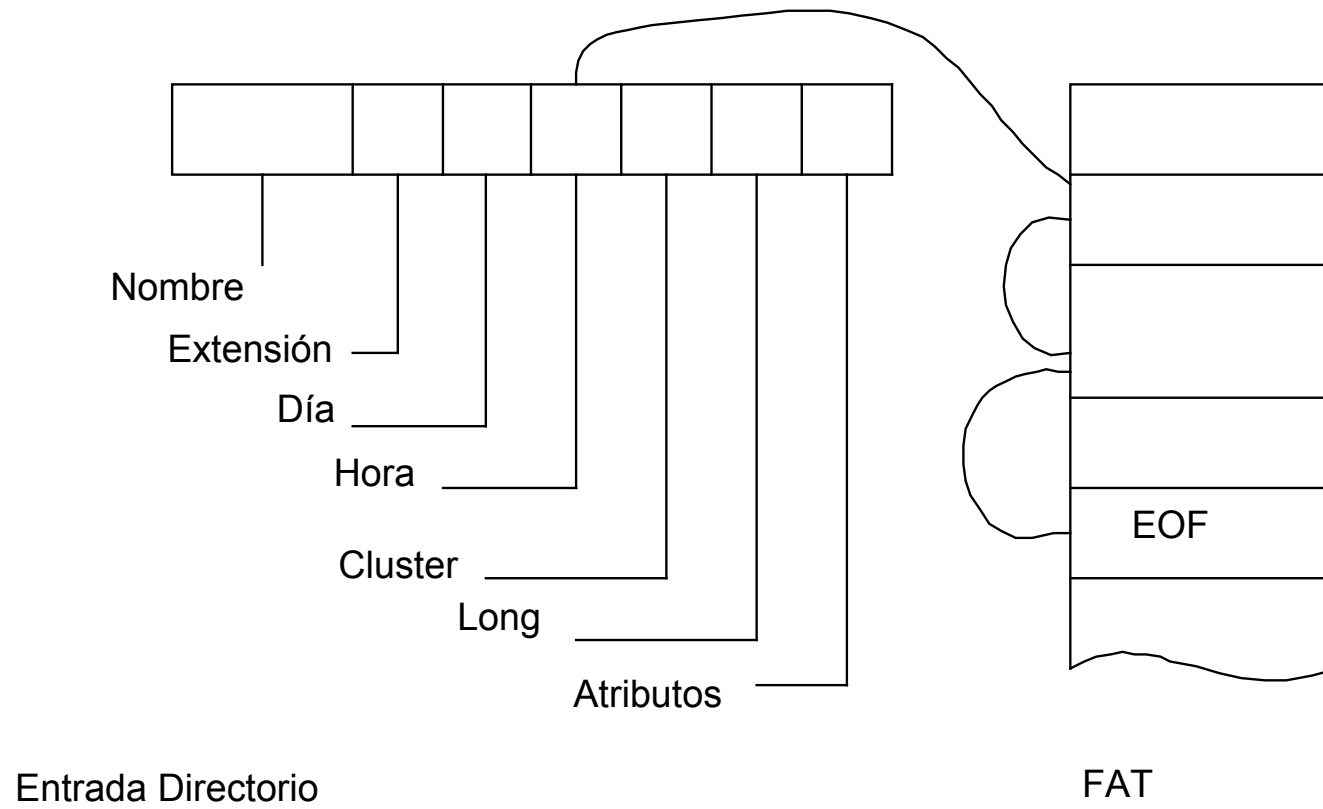
- FAT (DOS)
- UNIX
- LINUX
- NTFS
- HPFS

# FAT

- DOS formatea
  - Area Reservada
  - 1era. Copia FAT
  - Copias adicionales FAT (opcional)
  - Directorio \
  - Area de Datos

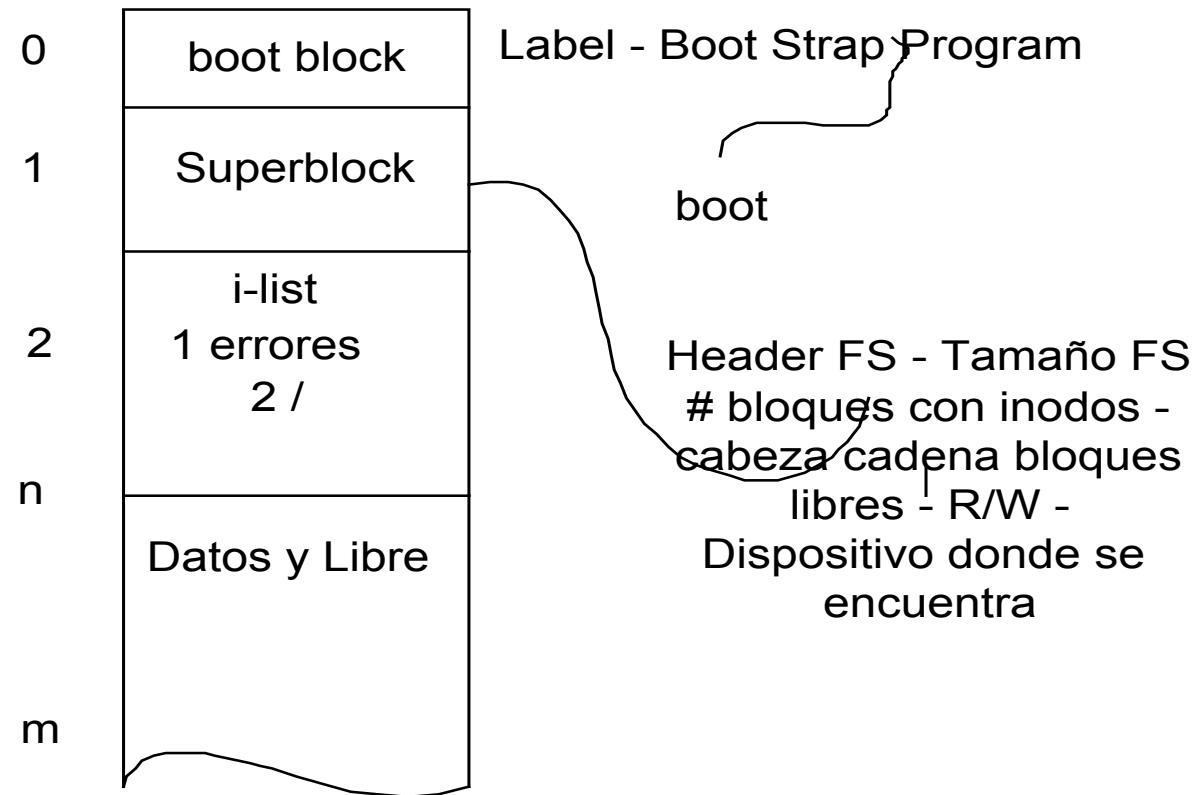


# FAT



# UNIX

- FS y swap



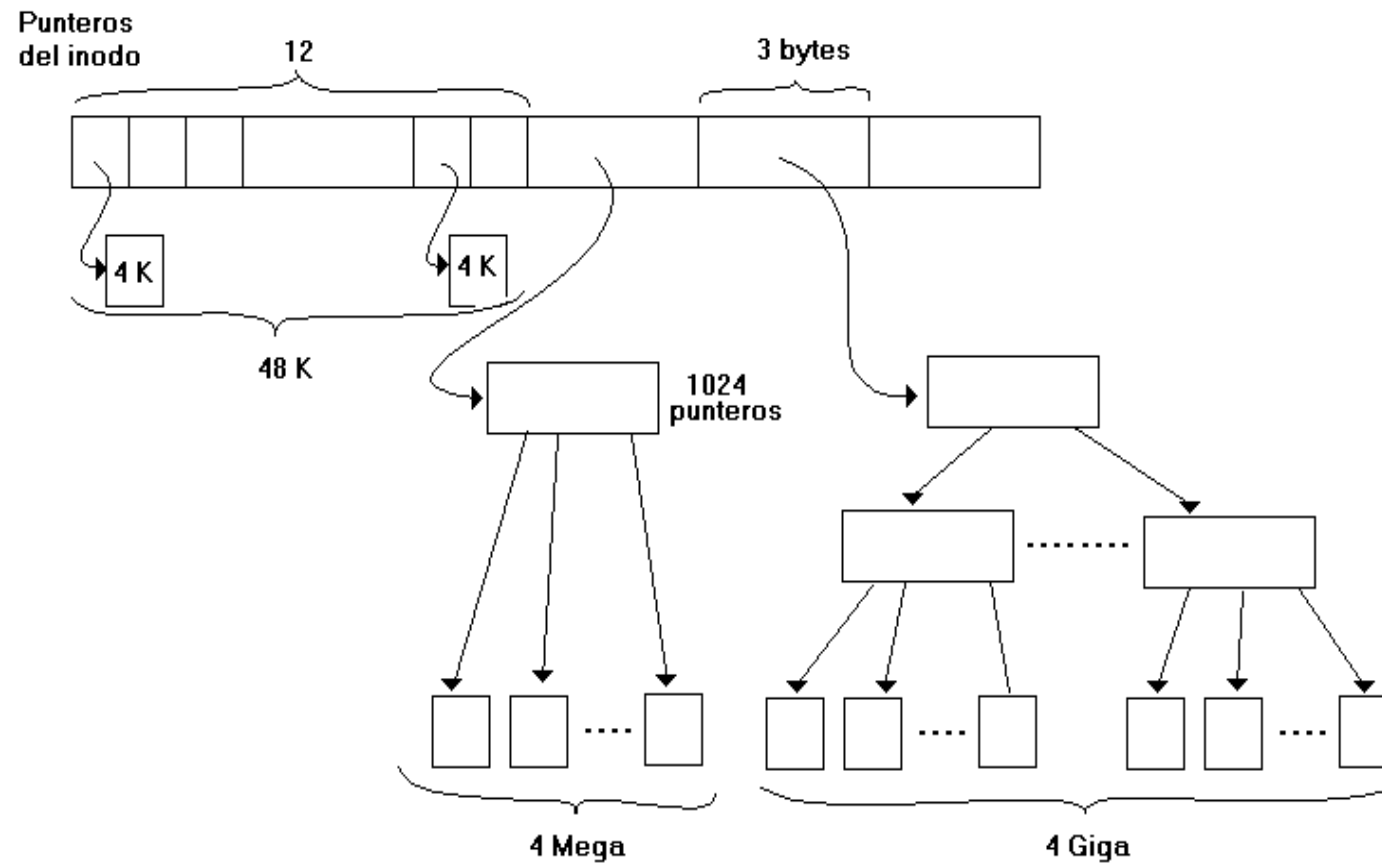
# Unix

- FS tiene Archivos (sin formato) Directorios (con formato)
- Inodo
  - Id usuario – id grupo – permisos – tiempos –
  - # hard. Links –
  - Tipo archivo (archivo – directorio – link simbólico – disp c ó b – sockets ...)
  - 15 apuntadores a bloques de disco

# Unix (15 apunadores)

- 12 bloques directos (4K → 48K ref. directamente)
- 13 indirecto 1024 bloques (4bytes → 4 MB)
- 14 doble indirección 4GB  $2^{32}$
- 15 triple indirección 4 TB

# Unix (15 apunadores)



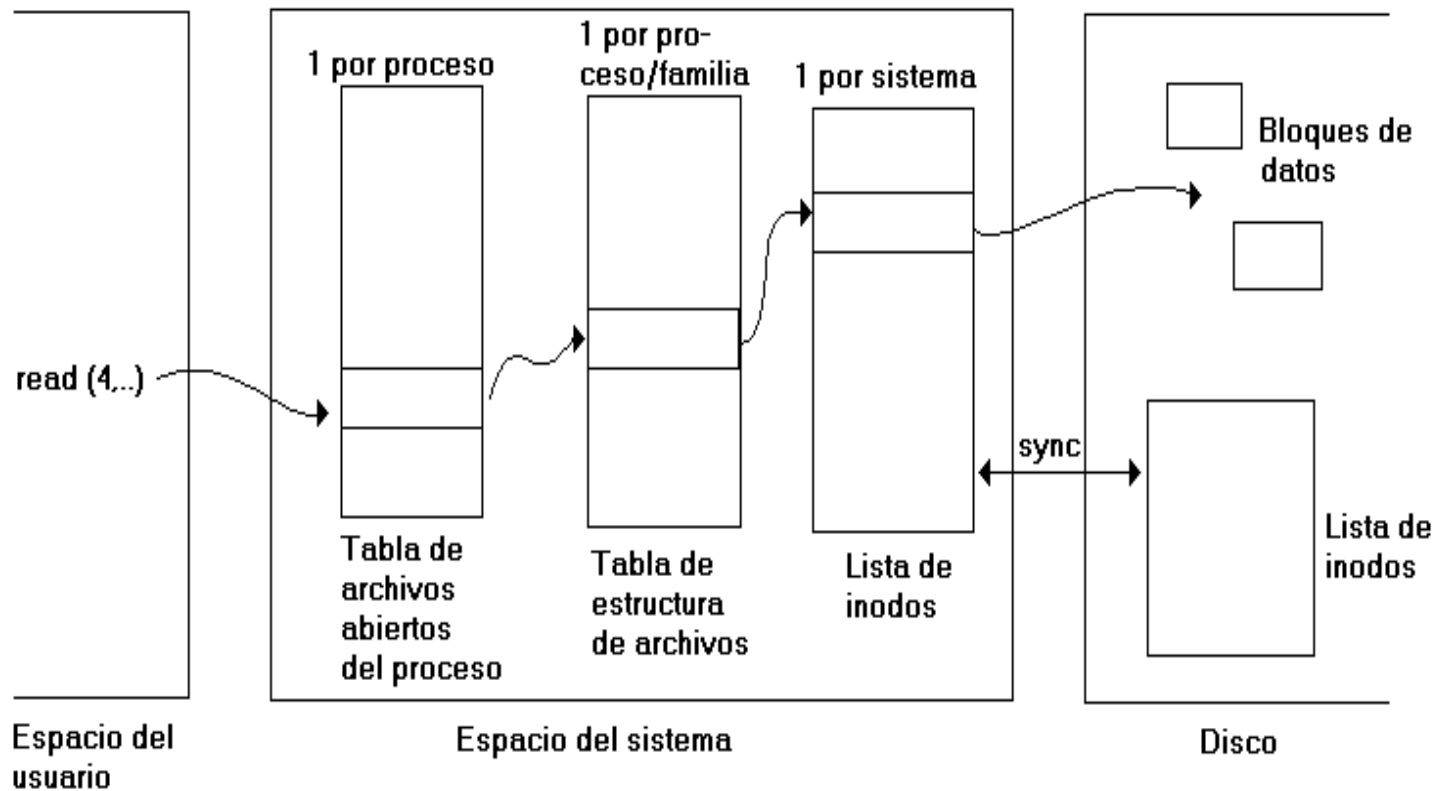
# UNIX

- Directorio
  - Sus contenidos están guardados en bloques de datos y están representados por inodos
  - Estructura (n bytes nombre arch/direct y 2 para inumber
  - Los 1eros. dos nombres son . y ..
  - Se busca dentro de él en forma secuencial

# UNIX

- Unix busca por FS/inodo
- Cuando un archivo se abre se carga su inodo en memoria
- Sync cada (30 seg.) y se guarda info de memoria a disco (info y superbblock)
- Ext3 → journaling

# UNIX – File descriptor





# Linux

- Soporta distintos FS
- (ext, ext2, ext3, minix, msdos, ufs, etc.)
- Tiene VFS

# Linux VFS

- VFS superblock
  - Device: tipo(/dev/hda1) indentificador 0X301
  - Inode pointer: apunta al 1er. Inodo montado
  - Blocksize: tamaño bloque en bytes (1024)
  - Superblock-operations: punteros a rutinas que manejan el superblock
  - FS type: puntero a estructuras de datos del FS específico
  - FS specific: puntero a la info del FS específico

# Linux VFS

- VFS inode
  - Device = VFS
  - Inode number = inodo
  - Modo = describe este inode-VFS
  - Userid = propietario
  - Times
  - Blocksize = tamaño bloque
  - Inode operations: puntero a rutinas específicas de este FS
  - Count = # de usos de este inodo (=0 puede eliminarse)
  - Lock = bloquea inodos en operaciones concurrentes
  - Dirty = inodo modificado
  - FS specific = VFS

# NTFS

- Unidad de almacenamiento físico  $2^n$  (sector 512k)
- Agrupamiento: Cluster de sectores contiguos en la misma pista ( $2^n$ )
- Volumen: Partición Lógica (uno o más discos. Tamaño máximo  $2^{64}$  bytes)
- Los archivos pueden usar agrupamientos no contiguos. Soporta hasta  $2^{32}$  agrupamientos aprox.  $2^{48}$  bytes

# NTFS (directorio)

- Partición sector de arranque
  - Información del volumen
  - Estructura FS
  - Código de arranque

# NTFS (directorio)

- MFT (Master File Table)
  - Información sobre directorios y archivos
  - Espacio libre
  - Está organizado por filas como una base de datos relacional

# NTFS (directorio)

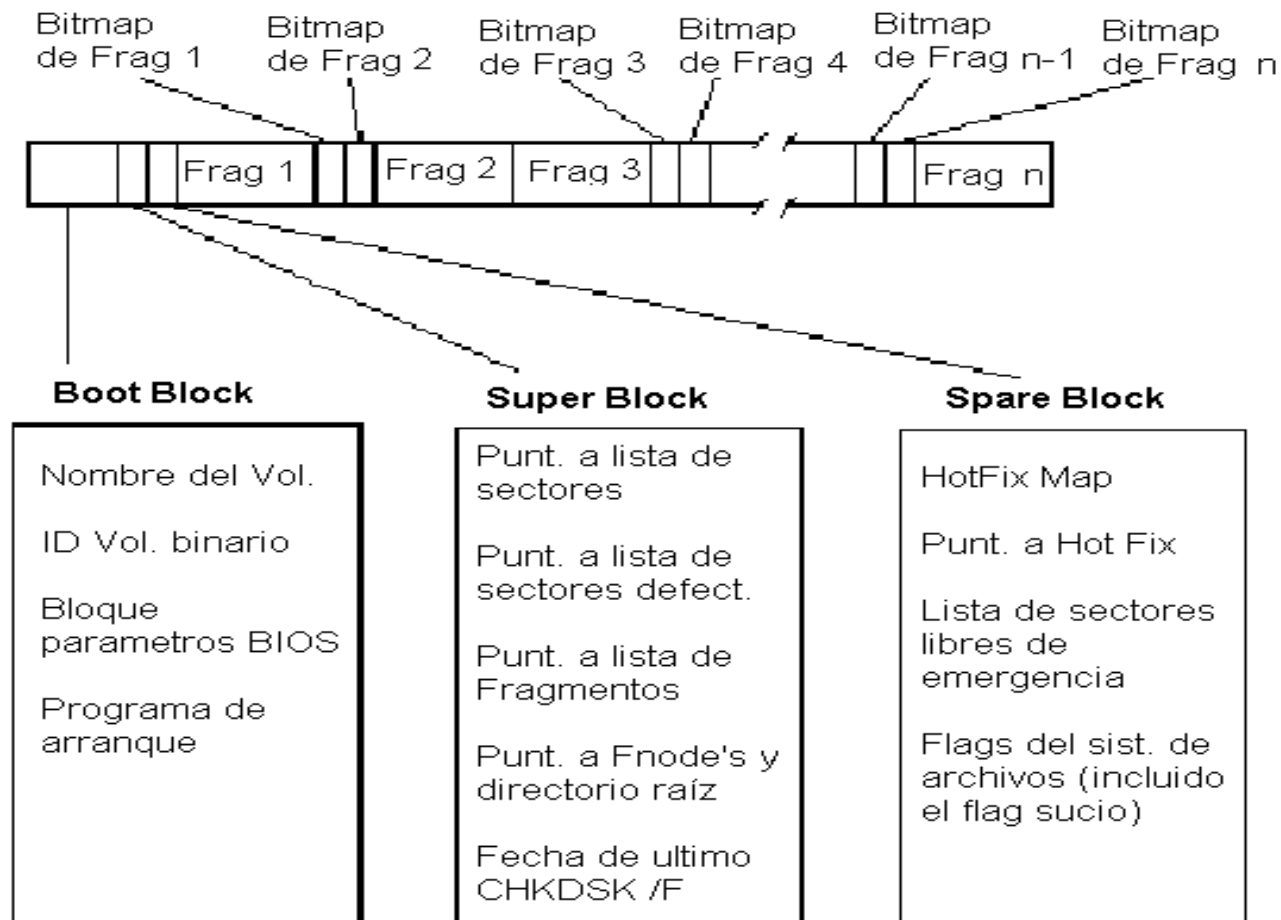
- Archivos del Sistema
  - MFT2 espejo de algunas filas de MFT
  - Registros (transacciones para recuperar NTFS)
  - Mapa de bits de agrupamientos (libres – ocupados)
- Area de datos

# NTFS (directorio)

- MFT está compuesto por filas de longitud variable
  - Cada fila describe un directorio o archivo
  - (incluye a la MFT que se trata como un archivo
  - Un archivo pequeño queda en la MFT
  - Si el archivo es grande desborda sobre otro agrupamiento (ubicados por punteros)
  - El MFT también puede desbordar por ser un archivo



# HPFS (High Performance File System)



# HPFS (High Performance File System)

