

SISTEMAS OPERATIVOS

Mg. Leandro Ezequiel Mascarello

<leandro.mascarello@uai.edu.ar>



UAIOnline
ultra >>>

- Conocer los conceptos de fichero y directorio así como sus características.
- Utilizar los servicios de gestión de Ficheros y directorios ofrecidos por el sistema operativo.
- Comprender la estructura de un sistema de ficheros.
- Comprender los mecanismos en los que se apoya un servidor de ficheros y aplicarlos a ejercicios sencillos.

- El sistema de ficheros permite organizar la información dentro de los dispositivos de almacenamiento secundario en un formato inteligible para el sistema operativo.
- Previamente a la instalación del sistema de ficheros es necesario dividir físicamente, o lógicamente, los discos en **particiones** o **volúmenes**.
- Una **partición** es una *porción de un disco a la que se la 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. Para ello se proporcionan mandatos como `format` o `mkfs` al usuario.
 - `#mkswap -c /dev/hda2 20800`
 - `#mkfs -c /dev/hda3 -b 8196 123100`

- Sistema de ficheros: conjunto coherente de metainformación y datos.
- Ejemplos de Sistemas de ficheros:

MS-DOS



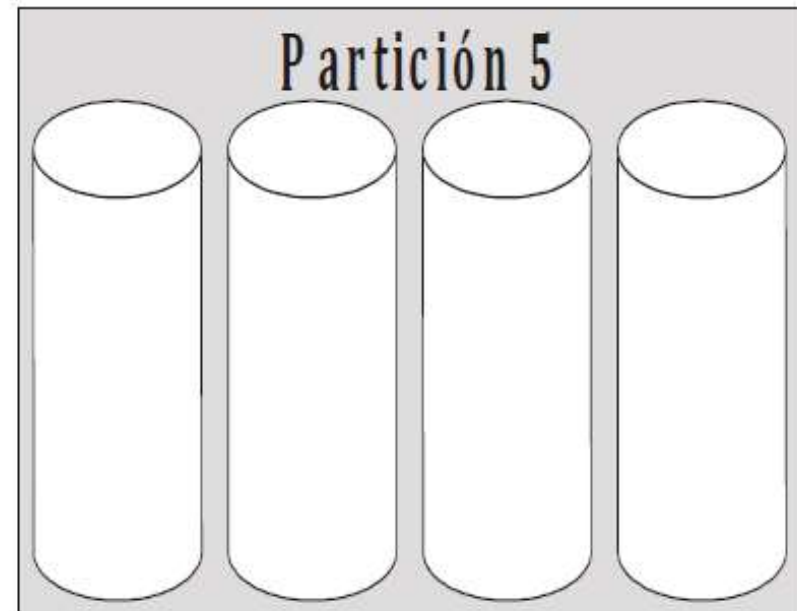
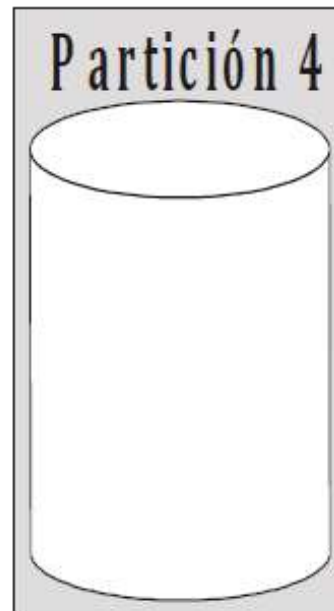
UNIX



- Descripción de sistemas de archivos:
 - El sector de arranque en MS-DOS
 - El superbloque en UNIX
- Relación sistema de archivos-dispositivo:
 - Típico: 1 dispositivo N sistemas de archivos (particiones)
 - Grandes archivos: N dispositivos 1 sistema de archivos
- Típicamente cada dispositivo se divide en una o más particiones (en cada partición sistema de archivos)
- La tabla de particiones indica el principio, el tamaño y el tipo de cada partición.

Tipos de Particiones

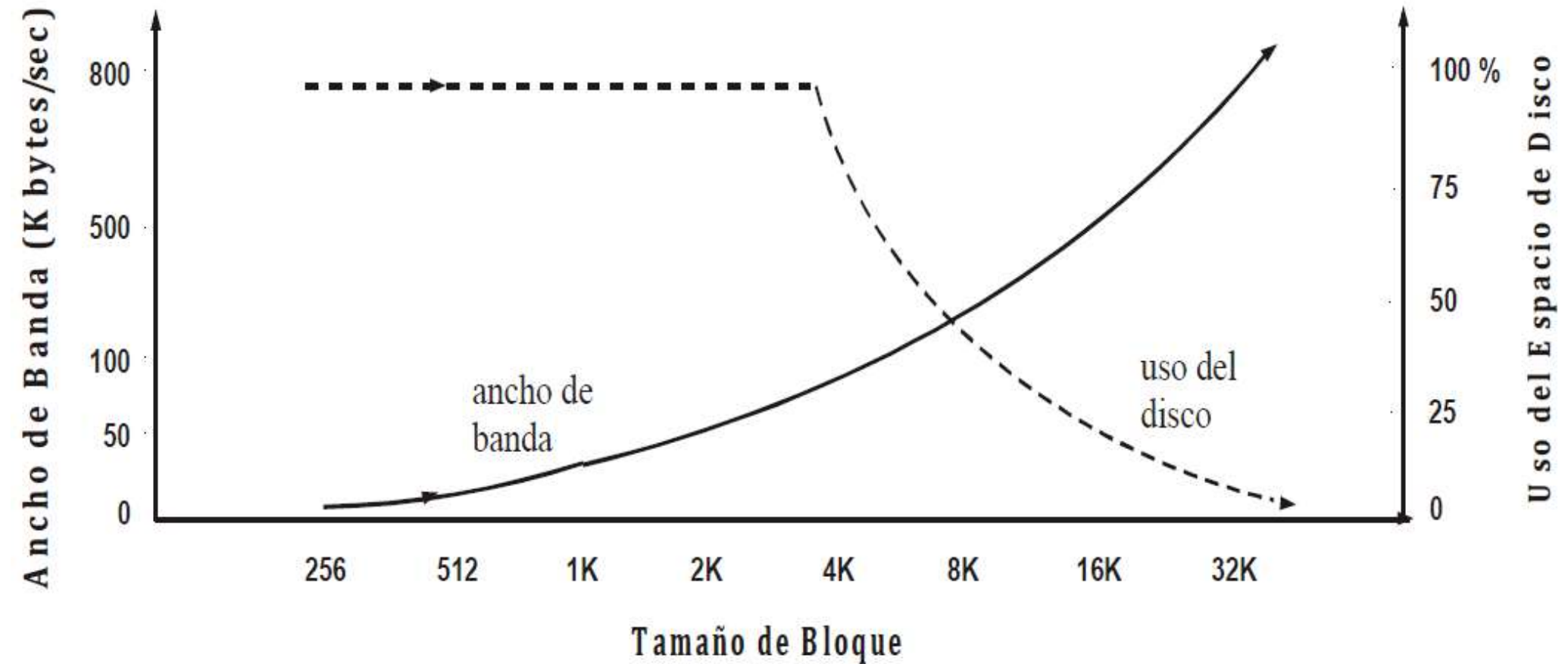
SO - UAI



- **Bloque:** *agrupación lógica de sectores de disco.*
 - Es la unidad de transferencia mínima que usa el sistema de archivos.
 - Optimizar la eficiencia de la entrada/salida de los dispositivos secundarios de almacenamiento.
 - Todos los sistemas operativos proporcionan un tamaño de bloque por defecto.
 - Los usuarios pueden definir el tamaño de bloque a usar dentro de un sistema de archivos mediante el mandato `mkfs`.
- **Agrupación:** *conjunto de bloques que se gestionan como una unidad lógica de gestión del almacenamiento.*
 - El problema que introducen las agrupaciones, y los bloques grandes, es la existencia de fragmentación interna.

Tamaño de bloque, ancho de banda y uso de disco

SO - UAI



- Utilizado fundamentalmente para dispositivos de sólo lectura → asignación contigua de bloques.
- El espacio de almacenamiento se divide en 5 zonas:
 - Área de sistema (16 bloques): Usada por las extensiones Rock-Ridge y Joliet.
 - Descripción de volumen(1 bloque): Información sobre la imagen del CD.
 - Tablas de localización: Lista precompilada de todos los directorios → Aceleración de búsquedas.
 - Directorios: Lista de entradas de los directorios.
 - Ficheros: Bloques de los ficheros.

- Utilizado en DOS y algunas versiones de Windows.
- Se sigue usando para dispositivos de almacenamiento portátil.
- Espacio dividido en:
 - Bloque de carga: Arranque del sistema operativo.
 - FAT: Tabla de asignación de ficheros.
 - Copia de FAT: Copia de seguridad.
 - Directorio Raíz: Directorio principal del volumen
 - Zona de directorios y ficheros.

- Tabla con una posición por bloque de disco.
- Tabla con direcciones de 12 bits.
- Número máximo de bloques: 4096.
- Tamaño de bloque: 512 bytes a 8 KB.
- Tamaño máximo: 32 MB.
- Utilizado en disquetes.

- Tabla con una posición por bloque de disco.
- Direcciones de 16 bits.
- Número máximo de bloques → 65.535.
- Tamaño de bloque: 512 bytes hasta 64KB.
- Utilizado en discos antiguos.

Tamaño de bloque	Tamaño máximo	Tamaño de bloque	Tamaño máximo
512 B	32 MB	8 KB	512 MB
1 KB	64 MB	16 KB	1 GB
2 KB	128 MB	32 KB	2 GB
4 KB	256 MB	64 KB	4 GB

Tamaño máximo de la tabla FAT: 128 KB

- Tabla con una posición por bloque de disco.
- Direcciones de 32 bits (solamente se usan 28).
- Número máximo de bloques → 256 Mbloques.
- Tamaño de bloque: 4 KB hasta 32 KB.
- Windows lo limita a dispositivos de hasta 32 GB.
- Usado en dispositivos de almacenamiento portátil.
- La tabla FAT puede ocupar cantidad considerable de espacio.
 - No se puede tener permanentemente en memoria y hay que consultarla en disco.
- Tamaño máximo de fichero: 4GB

- Estructura:
 - Bloque de carga: Arranque del sistema operativo.
 - Superbloque: Información descriptiva de la estructura del sistema de ficheros.
 - Superbloque virtual: Información genérica.
 - Superbloque específico: Dependiente del sistema operativo.
 - Mapas de bits de bloques: Un bit para cada bloque indicando libre/ocupado.
 - Mapas de bits de i-nodos: Un bit para cada i-nodo de la sección de i-nodos (libre/ocupado).
 - i-nodos: Tantos i-nodos como número de ficheros que puede albergar el sistema.
 - Linux crea un i-nodo por cada dos bloques de datos.
 - Bloques de datos.

SuperBloque de Sistema de Ficheros (LINUX)

SO - UAI

Superbloque Virtual

Dispositivo
Tamaño de bloque del superbloque
Flags, número mágico y tiempos
Operaciones sobre el sistema de archivos
Operaciones sobre el superbloque
Superbloque de EXT2
Superbloque de MINIX
Superbloque de UFS
...
Superbloque de NFS

Superbloque de MINIX

Número de particiones	Información presente en disco y en memoria
Número de agrupaciones	
Número de bloques de mapa de bits de nodos-i	
Número de bloques de mapas de bits de agrupaciones	
Primer bloque de datos	Información presente en disco y en memoria
Relación agrupación/bloque	
Máximo tamaño de archivo	
Número mágico	
Puntero al nodo-i raíz del SF montado	Información presente en memoria
Puntero al nodo-i de montaje	
Nodos-i/bloque	
Numero de dispositivo	
Flag de protección (ro,...)	Información presente en memoria
Versión de SF	
Primer bit libre en el mapa de bits de nodos-i	
Primer bit libre en el mapa de bits de clusters	

- Problemas de sistemas de ficheros tipo UNIX:
 - Metadatos agrupados al principio del disco.
 - Copia única de metadatos → ¿Corrupción del SF?
 - Bloques muy dispersos → Tiempo de búsqueda largo.
- Solución en BSD (FFS) y ext2.
 - Partición dividida en varias áreas: Grupos de cilindros.
 - Superbloque replicado en cada grupo.
 - En cada grupo mapas de bits y nodos-i del grupo.

- Proporciona un acceso eficiente y sencillo a los dispositivos de almacenamiento.
- Funciones: almacenar, buscar y leer datos fácilmente.
- Problemas de diseño:
 - Definir la visión de usuario del sistema de entrada/salida, incluyendo servicios, ficheros, directorios, sistemas de ficheros, etc.
 - Definir los algoritmos y estructuras de datos a utilizar para hacer corresponder la visión del usuario con el sistema físico de almacenamiento secundario.

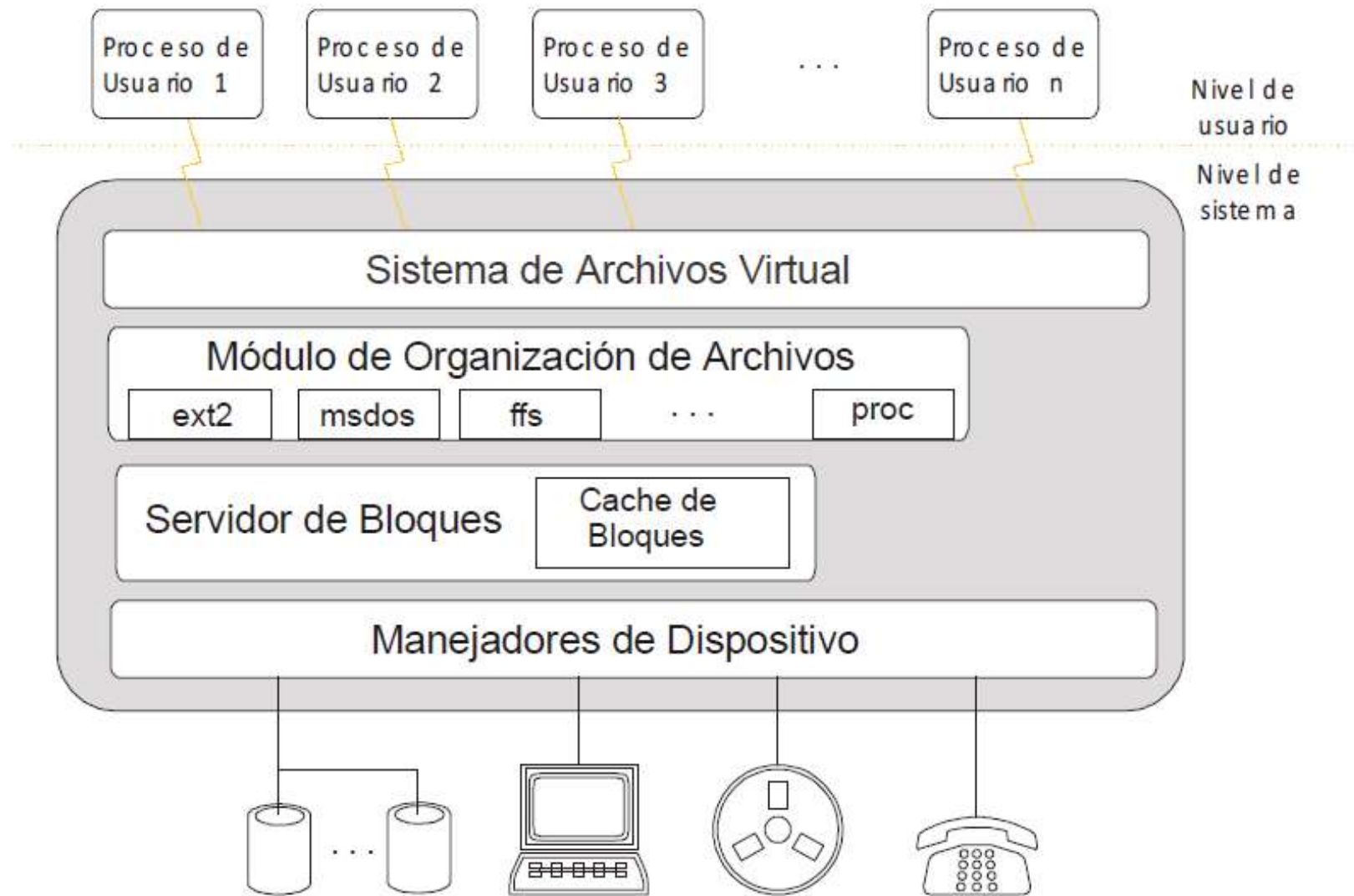
Ya visto



- Servidor de ficheros virtual: Proporciona interfaz de llamadas de E/S.
 - Independiente de sistema de ficheros particular.
- Módulo de organización de ficheros: Transforma peticiones lógicas en físicas.
 - Distinto para cada sistema de ficheros particular.
- Servidor de bloques: Gestiona las peticiones de operaciones de bloques sobre dispositivos.
 - Mantiene una caché de bloques o de páginas.
- Manejador de dispositivo: Transforma peticiones de bloques en peticiones de dispositivo.
 - Políticas de planificación de E/S.

Estructura del Servidor de Ficheros

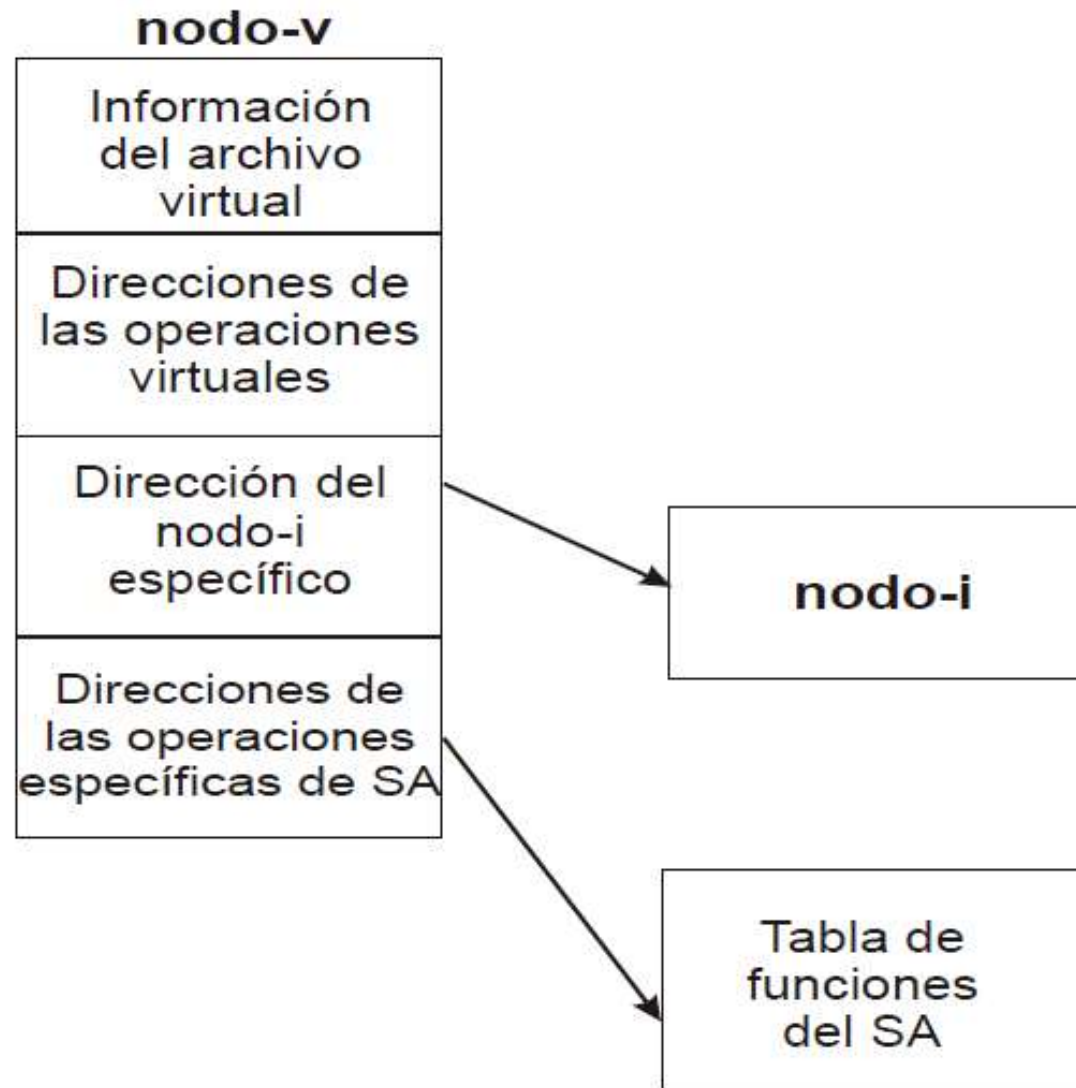
SO - UAI



- Proporciona interfaz de llamadas al sistema para gestión de ficheros.
- Servicios:
 - Gestión de directorios.
 - Gestión de nombres.
 - Servicios de seguridad.
 - Servicios genéricos sobre ficheros y directorios.
- Estructura de datos: v-nodo.
 - Un nodo virtual contiene información genérica sobre el fichero.

Estructura de un nodo virtual en UNIX

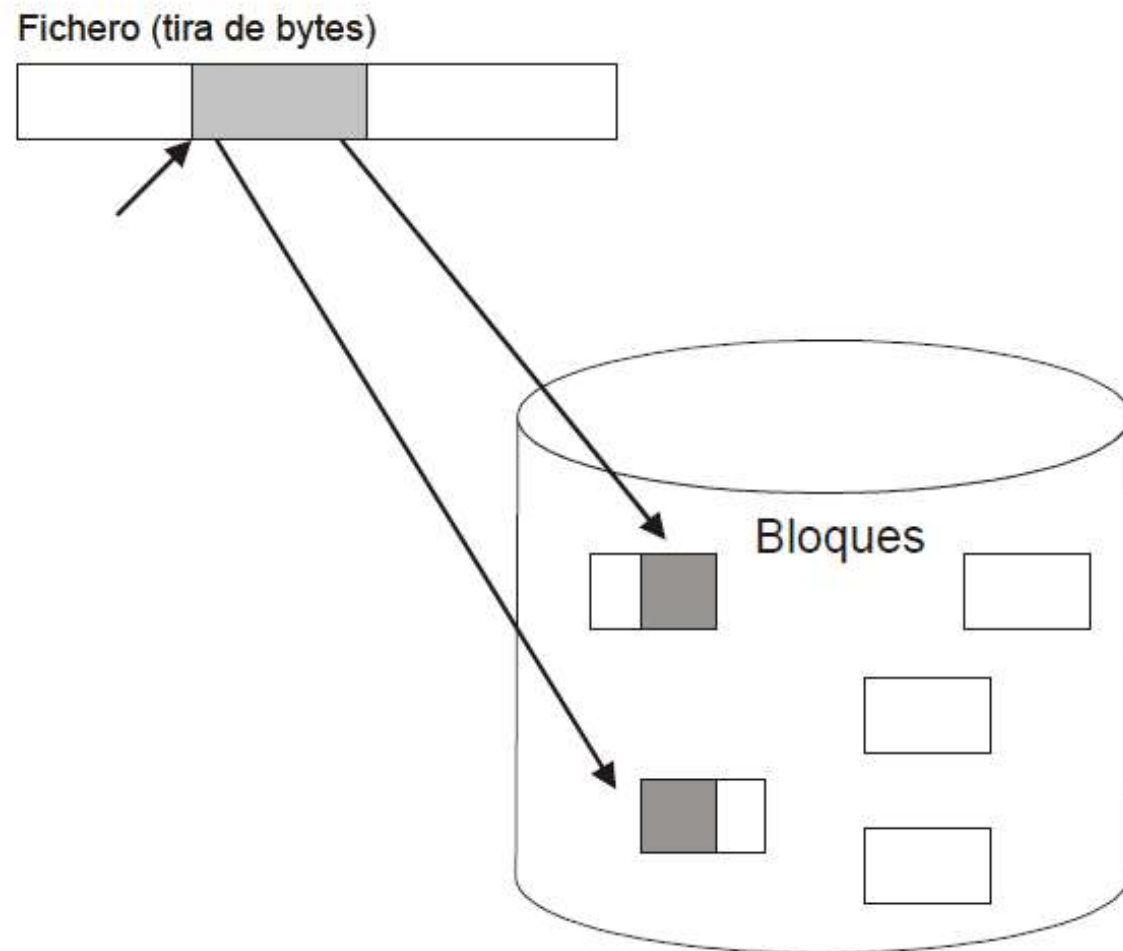
SO - UAI



- Relaciona la imagen lógica del fichero con su imagen física.
 - Algoritmos para traducir direcciones lógicas de bloques a direcciones físicas.
- Gestiona:
 - Espacio de almacenamiento del sistema de ficheros.
 - Asignación de bloques a ficheros.
 - Gestión de descriptores de ficheros (i-nodos).
- Un módulo de organización de ficheros para cada sistema de ficheros soportado (UNIX, AFS, Windows NT, MS-DOS, EFS, MINIX, etc.).
- Dentro de este nivel también se proporcionan servicios para pseudo-archivos, tales como los del sistema de ficheros proc.
- Resuelve las llamadas al sistema que son específicas del sistema de ficheros.
 - Para ello, se usa la información existente en el nodo-i del fichero afectado por las operaciones.

- Envía peticiones al manejador de dispositivo.
 - Leer bloque
 - Escribir bloque.
- Optimizaciones.
 - Caché de bloques.
 - Puede estar integrado con el gestor de páginas de memoria virtual.
- Las operaciones se traducen a llamadas de los manejadores de cada tipo de dispositivo específico y se pasan al nivel inferior del sistema de ficheros.
- Esta capa oculta los distintos tipos de dispositivos, usando nombres lógicos para los mismos. Por ejemplo, `/dev/hda3` será un dispositivo de tipo *hard disk* (`hd`), cuyo nombre principal es `a` y en el cual se trabaja sobre su partición 3.

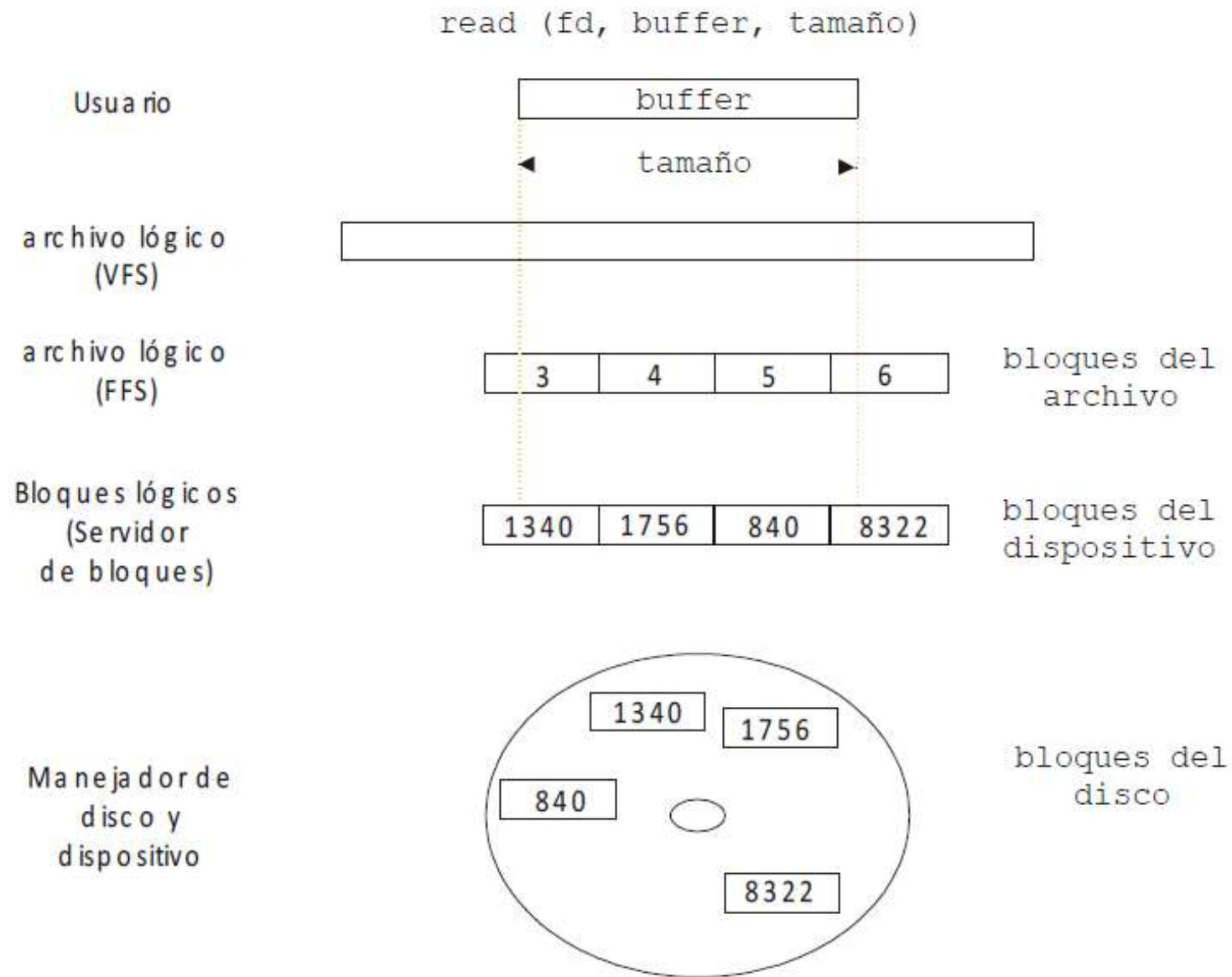
- Ficheros con estructura de bytes



- Un manejador por cada dispositivo.
 - Posible descomposición en manejador genérico más manejador de dispositivo.
- Su función principal es recibir ordenes de E/S de alto nivel, tal como `read_block 234`, y traducirlas al formato que entiende el controlador del dispositivo, que es dependiente de su hardware.
- Habitualmente, cada dispositivo tiene una cola de peticiones pendientes, de forma que un manejador puede atender simultáneamente a varios dispositivos del mismo tipo.
- Funcionalidad clave: Gestión de las colas de entrada/salida.
 - Algoritmo de planificación de entrada/salida.
 - Fusión de peticiones en la cola.

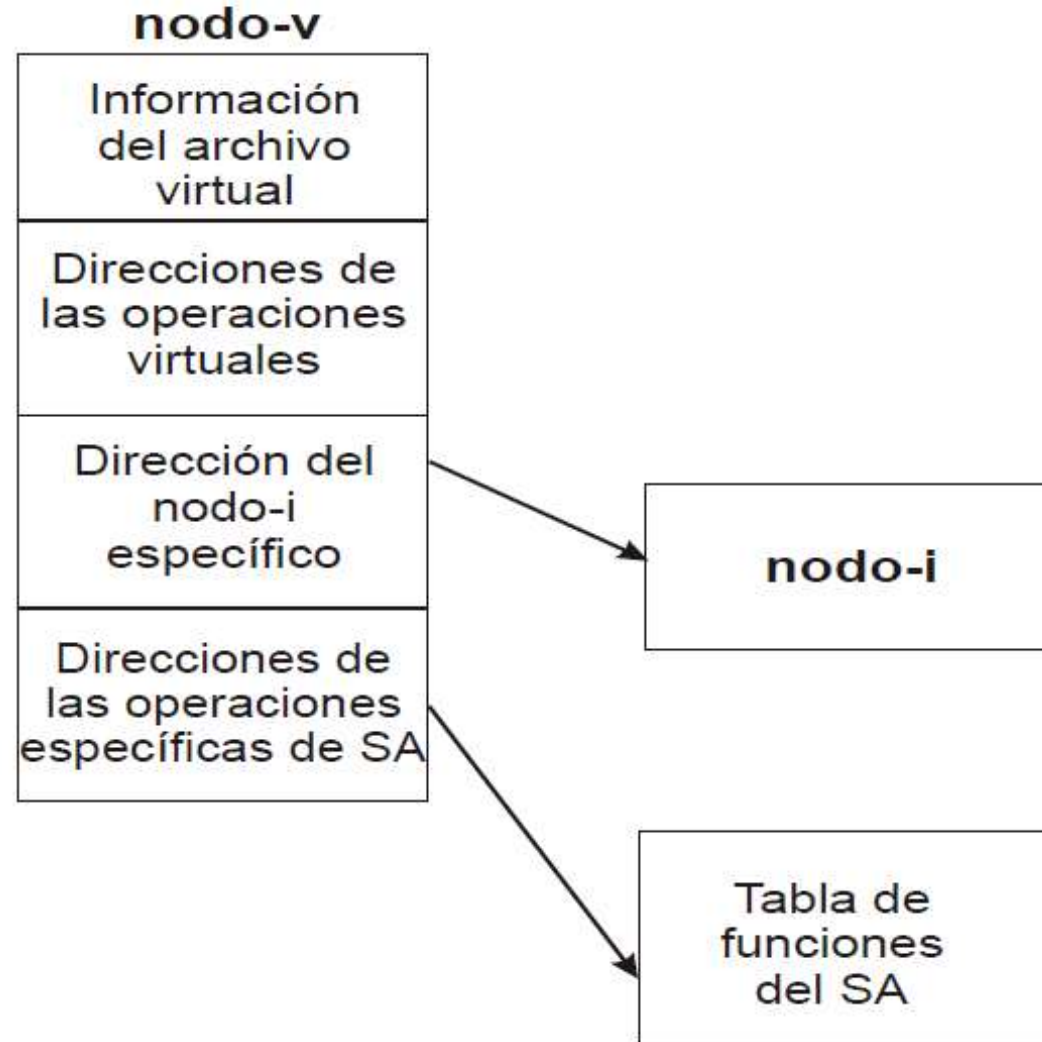
Flujos de Datos en un Sistema de Ficheros

SO - UAI



- Tabla de v-nodos:
 - Tabla única con todos los v-nodos de todos los ficheros abiertos.
- Tabla de ficheros abiertos:
 - Tabla por proceso con una entrada por fichero abierto.
- Tabla de posiciones en ficheros abiertos.
 - Tabla única con posiciones de punteros en archivos abiertos.
- Tabla de i-nodos.
 - Tabla única con los i-nodos todos los ficheros abiertos.

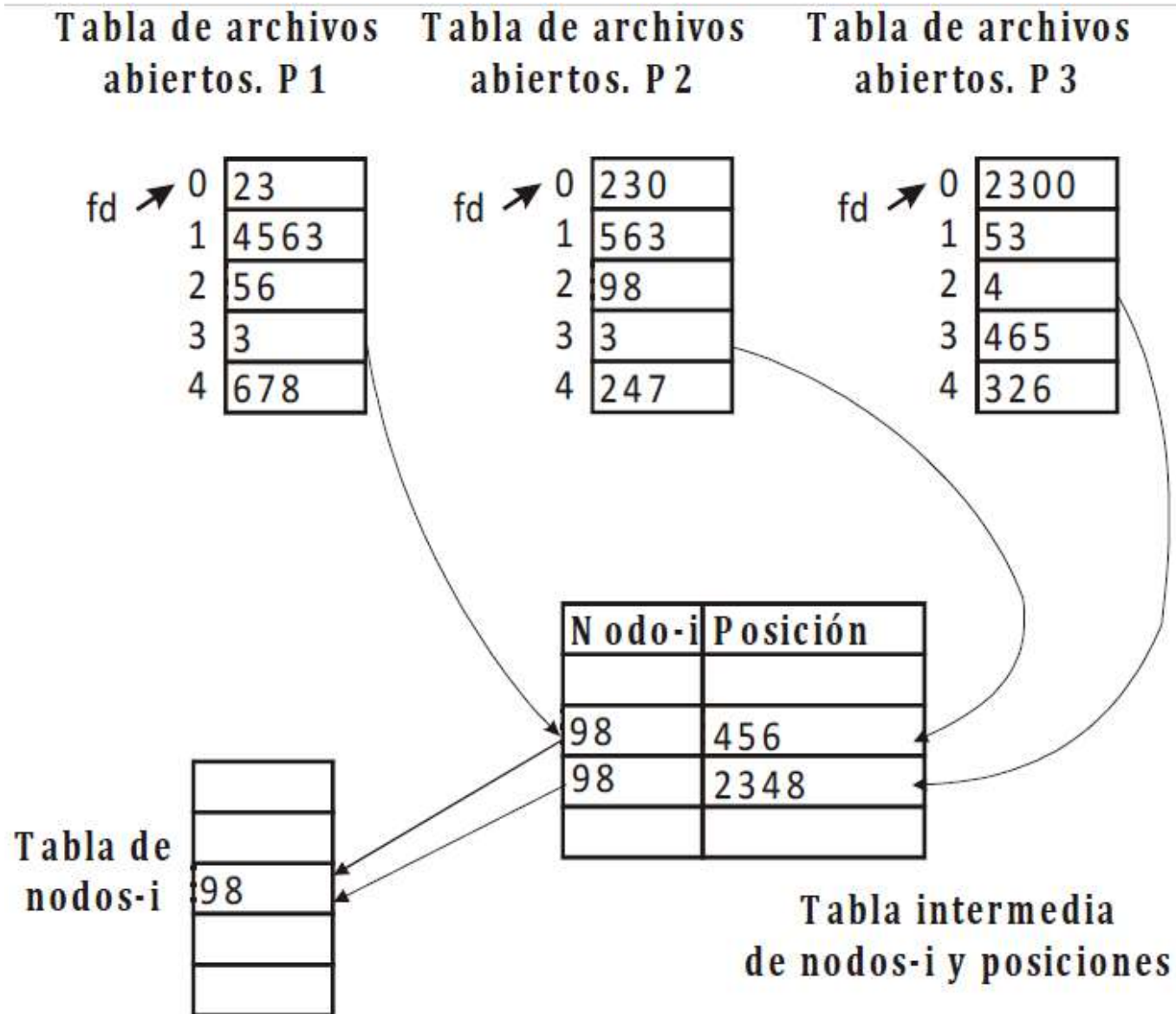
- Mantiene una tabla con todos los v-nodos de ficheros abiertos.
- Supone un límite sobre el número máximo de ficheros simultáneamente abiertos.
- En cada entrada:
 - Información de v-nodo en disco.
 - Información adicional solo en memoria.



- Tabla con una entrada por cada fichero abierto.
- Tamaño de la tabla limita el número máximo de ficheros abiertos por un proceso.
- Incluida en el BCP del proceso.
- Cada entrada mantiene un puntero a una posición de la tabla única de posiciones en ficheros abiertos.
- La tabla se rellena de forma ordenada.
- Descriptores estándar: 0, 1 y 2.
- Operaciones:
 - Open → Busca la primera entrada vacía en la tabla.
 - Close → Marca como vacía una entrada de la tabla.
 - Dup → Copia un valor de una entrada de la tabla a la primera vacía.
 - Fork → Copia todas la entradas de la tabla a la del nuevo proceso.

Tablas que usa el Servidor de Ficheros

SO - UAI



- Tabla única con información de cada fichero abierto.
 - Posición del puntero dentro del fichero abierto.
 - Puntero al v-nodo del fichero.
 - Modo de apertura del fichero.

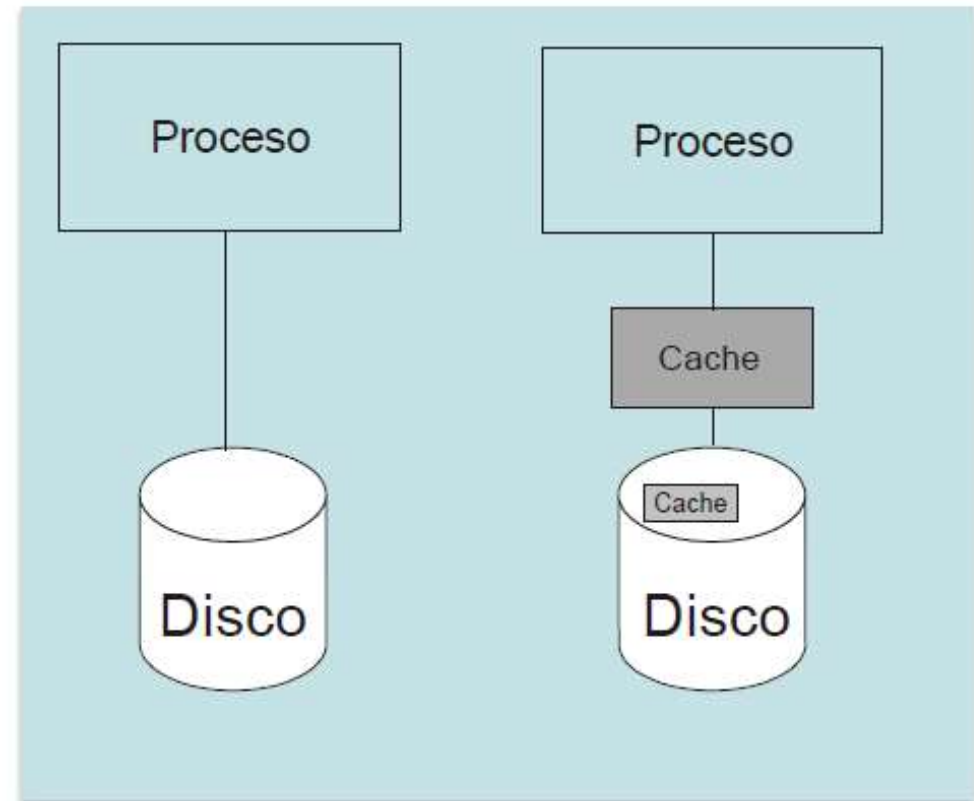
- Mantiene una tabla con todos los i-nodos de ficheros abiertos.
- Supone un límite sobre el número máximo de ficheros simultáneamente abiertos.
- En cada entrada:
 - Información de i-nodo en disco.
 - Información adicional solo en memoria.

- Basados en el uso de **almacenamiento intermedio** de datos de entrada/salida en memoria principal. Estos mecanismos son de dos tipos:
 - **Discos RAM**, cuyos datos están almacenados sólo en memoria.
 - Aceptan todas las operaciones de cualquier otro sistema de ficheros y son gestionados por el usuario.
 - *Pseudodispositivos* para almacenamiento temporal o para operaciones auxiliares del sistema operativo. Su contenido es volátil.
 - **Cache de datos**, instaladas en secciones de memoria principal controladas por el sistema operativo, donde se almacenan datos para optimizar accesos posteriores.
 - Se basan en la existencia de **proximidad espacial y temporal** en las referencias a los datos de entrada/salida.
 - Dos caches importantes dentro del servidor de ficheros: **cache de nombres y cache de bloques**.

- Fundamento:
 - Localidad espacial de referencias.
 - Localidad temporal de referencias.
 - Dos clases de flujos de E/S:
 - Usan una sola vez cada bloque .
 - Usan repetidamente los bloques.
 - Acceso a disco mucho más lento que el acceso a memoria.

- Estructura de datos en memoria con los bloques más frecuentemente utilizados

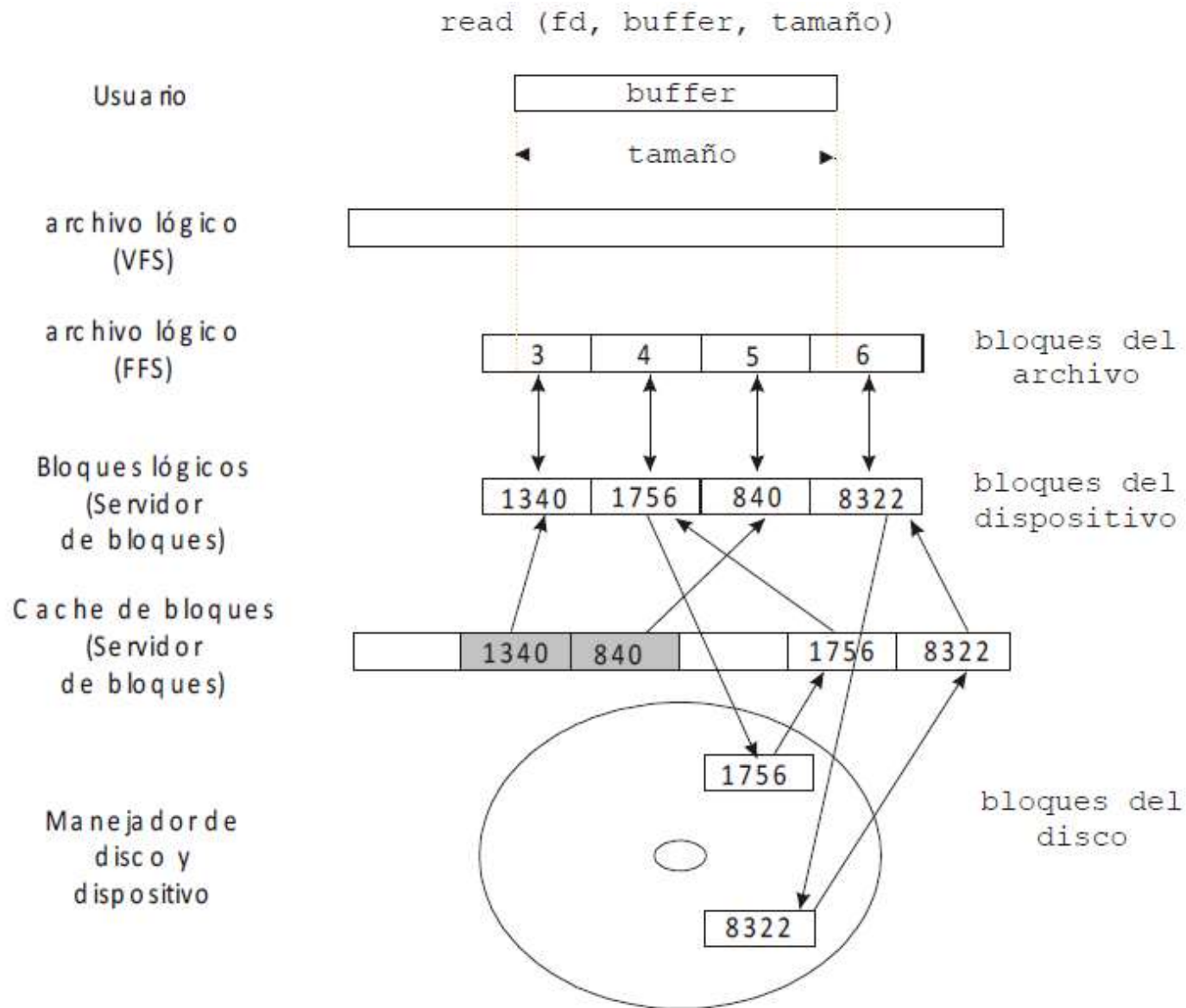
- Lecturas adelantadas
- Limpieza de la cache (sync)



- Principal problema: fiabilidad del sistema de ficheros.

Flujo de datos con caché de bloques

SO - UAI



- Algoritmo:
 - Comprobar si el bloque a leer está en la cache.
 - En caso de que no esté, se lee del dispositivo y se copia a la cache.
 - Si la cache está llena, es necesario hacer hueco para el nuevo bloque reemplazando uno de los existentes: **políticas de reemplazo**.
 - Si el bloque ha sido modificado (sucio): política de escritura.
- Políticas de reemplazo: FIFO (*First in First Out*), segunda oportunidad, MRU (*Most Recently Used*), LRU (*Least Recently Used*), etc.
 - La política de reemplazo más frecuentemente usada es la **LRU**. Esta política reemplaza el bloque que lleva más tiempo sin ser usado, asumiendo que no será referenciado próximamente.
 - Los bloques más usados tienden a estar siempre en la cache y, por tanto, no van al disco. La utilización estricta de esta política puede crear problemas de fiabilidad en el sistema de ficheros si el computador falla.
 - La mayoría de los servidores de ficheros distinguen entre **bloques especiales y bloques de datos**.

- **Escritura inmediata** (*write-through*): se escribe cada vez que se modifica el bloque.
 - No hay problema de fiabilidad, pero se reduce el rendimiento del sistema.
- **Escritura diferida** (*write-back*): sólo se escriben los datos a disco cuando se eligen para su reemplazo por falta de espacio en la cache.
 - Optimiza el rendimiento, pero genera los problemas de fiabilidad anteriormente descritos.
- **Escritura retrasada** (*delayed-write*), que consiste en escribir a disco los bloques de datos modificados en la cache de forma periódica cada cierto tiempo (30 segundos en UNIX).
 - Compromiso entre rendimiento y fiabilidad.
 - Reduce la extensión de los posibles daños por pérdida de datos.
 - Los bloques especiales se escriben inmediatamente al disco.
 - No se puede quitar un disco del sistema sin antes volcar los datos de la cache.
- **Escritura al cierre** (*write-on-close*): cuando se cierra un fichero, se vuelcan al disco los bloques del mismo que tienen datos actualizados.