

Tema4Gestiondearchivos.pdf



cdl_99



Sistemas Operativos



2º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada



MÁSTER EN

Inteligencia Artificial & Data Management

MADRID

Formamos
talento para un futuro
Sostenible

saber más



Esto no son apuntes pero **tiene un 10 asegurado** (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos Holandés con una garantía de hasta 100.000 euros por depositante. Consulta más información en [ing.es](https://www.ing.es)

Tema 4: Gestion de archivos

1. Ficheros

Concepto de fichero

Un fichero es una unidad de almacenamiento lógico no volátil (constante) que agrupa un conjunto de información relacionada entre sí bajo un mismo nombre. Posee un espacio de direcciones lógicas continuas. Para el usuario un fichero es la única forma de sistema de archivos.

Tipos de archivos:

- Archivos ASCII : formado por líneas de texto.
- Archivos binarios: ejecutables, objetos y clases no textuales.
- Archivos especiales (**solo UNIX**): permite modelar dispositivos de E/S como archivo.

Propiedades:

- Existencia a largo plazo → permanecen en memoria secundaria (MS) después de que el usuario se desconecte.
- Compatible entre procesos → los ficheros tienen nombres y permisos de acceso.
- Estructura → dependiendo del sistema de ficheros, un fichero puede tener una estructura interna para aplicaciones particulares. También sirve para expresar relaciones entre los mismos.

Atributos de un fichero (Metadatos)

Nombre	Tipo	Localización
Tamaño	Protección	Tiempo, fecha, info usuario

Operaciones que se pueden hacer sobre los ficheros:

- Gestión: crear, borrar, renombrar, copiar, establecer y obtener atributos.
- Procesamiento: abrir, cerrar, leer y escribir (modificar, insertar, borrar ...)

Estructura de un fichero: es el elemento básico de los datos

- Campo → es el elemento básico de los datos y contiene un único valor, como un apellido, una fecha o un sensor. Se define por su longitud (fija o variable) y su tipo de dato (cadena, decimal ...) Los campos de longitud **variable** pueden incluir subcampos como el valor, el nombre y, a veces, la longitud, o utilizar símbolos de demarcación para separar campos.
- Registro → es una colección de **campos relacionados** que se tratan como una unidad, como un registro de empleado con datos como nombre, número de seguridad social y fecha de contratación. Los registros pueden tener **longitud fija** o **variable**, dependiendo de la longitud de sus campos o del

Consulta condiciones aquí



do your thing

número de campos. Los registros de longitud variable suelen incluir un **campo de longitud** y, a veces, nombres de campo.

- **Fichero** → es una colección de **campos similares** que se maneja como una entidad única. Los usuarios y aplicaciones pueden referenciarlo por **nombre**, crearlo o eliminarlo. Los controles de acceso generalmente se aplican a nivel de fichero, aunque en sistemas avanzados pueden hacerse a nivel de **registro** o incluso de **campo**.
- **Base de datos** → es una colección de datos relacionados.

Hay una serie de operaciones y según como sean afectarán a la forma en la que se organiza el fichero (eg Obtener_todos, Obtener_uno, Insertar_uno, Actualizar_Uno ...)

1.2 Sistemas de gestión de ficheros

Un **sistema de gestión de ficheros** es un software que facilita a usuarios y aplicaciones el acceso y manejo de ficheros de forma consistente, eliminando la necesidad de desarrollar soluciones específicas para cada aplicación. Sus objetivos principales incluyen:

- Satisfacer las necesidades de almacenamiento y manipulación de datos.
- Garantizar la validez de los datos.
- Optimizar el rendimiento del sistema y la experiencia del usuario.
- Soportar diversos dispositivos de almacenamiento.
- Minimizar la pérdida de datos.
- Proveer rutinas estándar de E/S.
- Gestionar el acceso de múltiples usuarios en sistemas multiusuario.

Arquitectura de un sistema de ficheros

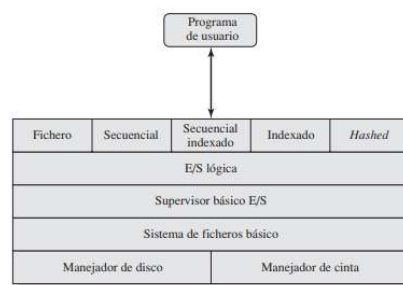


Figura 12.1. Arquitectura software de un sistema de ficheros.

El manejo de ficheros y dispositivos se organiza en varios niveles:

1. **Controladores de dispositivos:** Interactúan directamente con los periféricos o sus controladores, gestionando la E/S de discos y cintas. Son parte del sistema operativo.
2. **Sistema de ficheros básico:** Gestiona bloques de datos intercambiados con almacenamiento secundario (discos o cintas) y realiza buffering. No interpreta datos ni estructuras de ficheros.

3. **Supervisor de E/S básico:** Coordina las operaciones de E/S seleccionando dispositivos, planificando disco y cinta, asignando buffers y memoria secundaria.
4. **E/S lógica:** Permite acceder a registros dentro de los ficheros, proporcionando una interfaz general para trabajar con registros en lugar de bloques.
5. **Método de acceso:** Es la capa más cercana al usuario, ofreciendo una interfaz estándar entre las aplicaciones y los sistemas de ficheros, ajustándose a las estructuras y formas de acceso de los datos.

1.3 Organización y acceso a los ficheros

Para escoger una organización de ficheros, son importantes los siguientes criterios:

- Tiempo de acceso corto.
- Facilidad de actualización.
- Economía de almacenamiento.
- Mantenimiento sencillo.
- Fiabilidad.

Existen cinco organizaciones fundamentales de ficheros, cada una adecuada para diferentes necesidades y aplicaciones:

1. **Pila:** Almacena registros en el orden en que se añaden, ideales para accesos secuenciales simples.
2. **Fichero secuencial:** Los registros están organizados en un orden específico (como alfabético o numérico) para facilitar accesos secuenciales ordenados.
3. **Fichero secuencial indexado:** Combina un índice para accesos directos con la organización secuencial para búsquedas rápidas.
4. **Fichero indexado:** Utiliza un índice que mapea claves a registros, permitiendo accesos rápidos y flexibles sin depender de un orden secuencial.
5. **Fichero de acceso directo o hash:** Emplea funciones hash para localizar registros rápidamente, eficiente para accesos aleatorios frecuentes.

Estas organizaciones pueden combinarse para adaptarse a diversas estructuras en sistemas reales.

Esto no son apuntes pero **tiene un 10 asegurado** (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos Holandes con una garantía de hasta 100.000 euros por depositante. Consulta más información en [ing.es](https://www.ing.es)

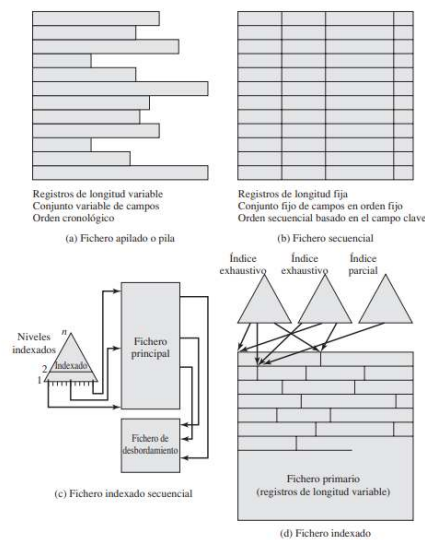


Figura 12.3. Organizaciones comunes de ficheros.

2. Directorios

2.1 Contenido

Un **directorio** es un fichero especial que almacena información sobre otros ficheros, como sus atributos, ubicación y propiedad.

- **Función:** Gestionar y organizar ficheros en un sistema.
- **Gestión:** El sistema operativo controla gran parte de los datos, especialmente los relacionados con el almacenamiento.
- **Acceso:** Aunque es accesible mediante rutinas de gestión de ficheros, la información suele proporcionarse de forma indirecta a usuarios y aplicaciones.

2.2 Estructura

La organización (lógica) de los directorios debe proporcionar:

- **Eficiencia:** localización rápida de un archivo
- **Denominación:** adecuada a los usuarios.
- **Agrupación:** de forma lógica y según sus propiedades.

Consulta condiciones aquí



do your thing



A un unico nivel: unico directorio que contiene todas las entradas de archivos. Dificil de entender e implementar. Tiene problemas de denominacion y agrupacion.

A dos niveles: cada usuario tiene su propio directorio. Varios usuarios diferentes pueden tener archivos con el mismo nombre. No hay posibilidad de agrupación.

En arbol: se presentan todos los directorios y subdirectorios aprtiendo de un directorio raiz, existiendo un camni unico (*path*)
Nodos → directorios, hojas → archivos.

Apararece el concepto de directorio de trabajo para saber cual sera el *path* de ahí a un archivo determinado.

Exite la posibilidad de agrupación. Los nombres de los caminos son absolutos y relativos.

En grafo: en el cual el mismo archivo o subdirectorio puede estar en dos directorio distintos, establenciendo una relacioón unívoca nombre-archivo. se comparte archivo mediante **enlaces**. Son flexibles pero complejos.

2.3 Compartición de ficheros

Derechos de acceso

Los usuarios o grupos reciben permisos específicos que determinan las acciones que pueden realizar, como leer, escribir o ejecutar.

Acceso simultaneo

Cuando se garantiza acceso de adición o actualización de un fichero a más de un usuario, el sistema operativo o sistema de gestión de ficheros debe forzar una disciplina. Una técnica de fuerza bruta consiste en permitir al usuario bloquear el fichero completo cuando se va a actualizar. Un control de grano más fino implica el bloqueo de registros individuales durante la actualización. Se deben tratar aspectos de exclusión mutua e interbloqueos a la hora de diseñar capacidades de acceso compartidas.

2.4 Bloques y registros

Tamaño y métodos de organización de bloques

1. Bloques de longitud fija o variable:

- Los **bloques fijos** simplifican la gestión de E/S, asignación de buffers y organización en almacenamiento, aunque pueden generar fragmentación interna.
- Los **bloques variables** se adaptan mejor a datos de longitud cambiante pero complican la gestión de memoria y E/S.

2. Tamaño del bloque:

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos Holandés con una garantía de hasta 100.000 euros por depositante. Consulta más información en ing.es

Que te den **10 € para gastar**
es una fantasía.
ING lo hace realidad.

Abre la **Cuenta NoCuenta** con el código
WUOLAH10, haz tu primer pago y llévate 10 €.

Quiero el cash

[Consulta condiciones aquí](#)



do your thing

- **Bloques grandes:** Reducen el número de operaciones de E/S, mejorando el procesamiento secuencial y aprovechando la proximidad de referencias.
- **Bloques pequeños:** Reducen transferencias innecesarias en accesos aleatorios pero pueden requerir más operaciones de E/S.

3. Métodos de almacenamiento en bloques:

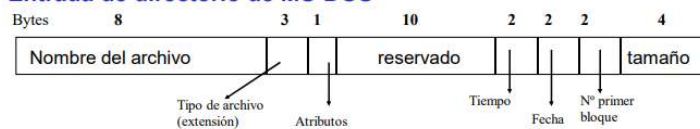
- **Bloques fijos:** Registros de longitud fija; puede haber fragmentación interna.
- **Bloques expandidos de longitud variable:** Empaquetan registros en bloques y permiten expansión entre bloques, enlazándolos con punteros.
- **Bloques no expandidos de longitud variable:** No permiten expansión entre bloques; suelen dejar espacio malgastado si un registro no cabe en el espacio restante.

2.5 Implementación de directorio

Contenido de una entrada de directorio (Casos)

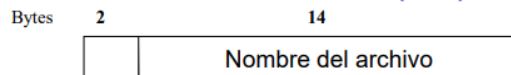
Nombre de Archivo + atributos + dirección de bloques de datos

Entrada de directorio de MS-DOS



Nombre de archivo + puntero de una estructura de datos que contiene toda la información relativa al archivo (UNIX)

Entrada de directorio de UNIX (s5fs)



Posibilidades respecto a la implementación:

1. Lista lineal:

- a. Sencillo de programar.
- b. consume tiempo en las creaciones, búsquedas,..., si no se utiliza una cache software

2. Tabla hash:

- a. Decrementa el tiempo de búsqueda.
- b. Dificultades:
 - i. Tamaño fijo de la tabla hash.
 - ii. Dependencias de la función hash sobre el tamaño de la tabla.
 - iii. Necesita previsión para colisiones.

2.6 Implementación de archivos compartidos (o enlace)

Esto no son apuntes pero **tiene un 10 asegurado** (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos Holandeses con una garantía de hasta 100.000 euros por depositante. Consulta más información en [ing.es](https://www.ing.es)

Enlaces simbólicos: nuevo archivo cuyo contenido es el nombre del archivo enlazado.

- Se crea nueva entrada en el directorio, se indica el tipo de enlace y se almacena el camino del archivo a enlaces.
- Se puede usar en entornos distribuidos, muchos accesos al disco.

Enlaces absolutos(hard): es una entrada en el directorio que apunta directamente al **i-nodo** del archivo en el sistema de archivos. Esto significa que varios nombres de archivo pueden compartir el mismo i-nodo y, por ende, referirse al mismo archivo físico en el disco.

- Se crea una nueva entrada en el directorio y se copia la dirección de la estructura de datos con la info del archivo.
- Problema al borrar enlaces → Solución: contador de enlaces.
- Cuando se crea un enlace duro se incrementa el contador de enlaces.

3. Sistemas de archivos

3.1 Introducción

Protección

Proporcionar un acceso controlado a los archivos, de lo que puede hacerse y quien puede hacerlo.

Listas y grupos de acceso

Principal solución a la protección: hacer el acceso dependiente del **identificativo del usuario**, pero la lista de usuarios de un archivo puede llegar a ser muy extensa.

Esto se puede solucionar con clses de usuario → propietario, grupo o publico.

Se puede hacer uso de una propuesta alternativa que seria las contraseñas, pero tienen un problema que hay que recordarla.

Semantica de consistencia

Especificar cuando las modificaciones son visibles por otros usuarios.

Ejemplos:

- UNIX: la estructura es directamente observable, existe un modo para que los usuarios compartan el puntero actual de posicionamiento en un archivo.
- de sesión: la estructura en un archivo no es directamente observable, cuando un archivo se cierra, sus cambios solo se observan en sesiones posteriores.
- Archivos inmutables: cuando un archivo se declara como compartido, no se puede modificar.

Funciones básicas

- Tener el conocimiento de todo slos archivos del sistema.
- Controlar el comportamiento y forzar la protección.

Consulta condiciones aquí



do your thing

- Gestionar el espacio.
- Traducir las direcciones lógicas del archivo en direcciones físicas del disco.

3.2 Estructura del sistema de archivos

Un sistema de archivos es una entidad lógica autocontenida. Este posee los siguientes problemas de diseño:

1. Como ve el usuario el sistema de archivos (SA):
 - a. definir el archivo y sus atributos.
 - b. definir operaciones permitidas sobre el archivo.
 - c. definir la estructura de directorios.
2. Definir los algoritmos y estructuras de datos que deben crearse para establecer la correspondencia entre el sistema de archivos lógicos y los dispositivos físicos donde se almacenan.

Organización en niveles:

- Por eficiencia,+: el SO mantiene una tabla indexada de archivos abiertos.
- Por bloque de control de archivo: estructura con información de un archivo en uso.

3.3 Métodos de asignación de espacio

Contiguo (no es muy usado)

Cada archivo ocupa un conjunto de bloques contiguos en disco.

Ventajas

- Sencilla: solo hace falta la localización del comienzo (nº de bloque) y la longitud. Se buscan huecos libres.
- Es de fácil acceso

Desventajas

Si no se conoce inicialmente el tamaño y el archivo crece habrá que buscar un nuevo hueco → puede producir fragmentación externa → habrá que hacer comparación del disco a menudo.



Asociación lógica a física

$\text{Direccion_logica} / \text{tam_bloque} \rightarrow C(\text{cociente}), R(\text{resto})$

$\text{bloque a acceder} = C + \text{direccion de comienzo}$

$\text{desplazamiento en el bloque} = R$

No contiguo-encalazado

El archivo se asigna el primer bloque libre, cada archivo es una lista enlazada de bloques de disco. Los bloques pueden estar dispersos en el disco.

Ventajas

- Evita la fragmentación interna.
- Basta almacenar el puntero al primer bloque del archivo.
- El archivo puede crecer dinámicamente cuando hay bloques de disco libres.

Desventajas

- El acceso directo no es efectivo (solo secuencial)
- Hace falta espacio para los punteros de enlace. El cálculo del tamaño de archivo no es real → Solución **agrupaciones de bloque (clusters)**
- Inseguridad por la pérdida de punteros → Solución **lista doblemente enlazada (overhead)**



Asociación lógica a física

$\text{direccion_logica} / \text{tam_bloque} - \text{direccion(1byte)} \rightarrow C(\text{cociente}), R(\text{resto})$

$\text{Bloque a acceder} = C$

$\text{Desplazamiento en el bloque} = R + 1$

Tabla de asignación de Archivos (FAT)

La **Tabla de Asignación de Archivos (FAT)** es una variación del método de asignación enlazada, utilizada en sistemas como **Windows** y **OS/2**. Su funcionamiento se basa en los siguientes principios:

- **Estructura reservada:** Se dedica una sección específica al inicio de la partición del disco para almacenar la FAT.
- **Entradas por bloque:** La tabla contiene una entrada por cada bloque del disco, indexada por su número de bloque.
- **Simplicidad y eficiencia:** Su diseño es simple y proporciona un rendimiento eficiente, especialmente cuando la FAT está almacenada en caché.
- **Optimización del acceso directo:** Para localizar un bloque de datos, solo es necesario leer en la FAT, lo que optimiza el acceso directo a los datos.
- **Limitaciones:** Puede presentar problemas si se pierden punteros. Para mitigar este riesgo, se mantiene una copia redundante de la FAT.

Esta técnica es conocida por su facilidad de implementación, aunque puede tener limitaciones en sistemas más complejos o de mayor escala.

No contiguo-indexado

En este método, todos los punteros a los bloques de datos de un archivo se almacenan en un **bloque índice**. Cada archivo tiene su propio bloque índice, y el directorio guarda la ubicación de este bloque.

Funcionamiento:

Esto no son apuntes pero **tiene un 10 asegurado** (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos Holandes con una garantía de hasta 100.000 euros por depositante. Consulta más información en [ing.es](https://www.ing.es)



- Para acceder al *i*-ésimo bloque, se consulta la *i*-ésima entrada del bloque índice, que contiene su dirección.

Ventajas:

1. **Acceso directo eficiente:** Localiza bloques rápidamente.
2. **Sin fragmentación externa:** Evita problemas de fragmentación en el disco.

Desventajas:

1. **Desperdicio de espacio:** Bloques índices grandes pueden desperdiciar memoria.
2. **Tamaño del bloque índice:**
 - **Soluciones:**
 - **Índices enlazados:** Enlaza bloques adicionales.
 - **Índices multinivel:** Divide la indexación en varios niveles (más accesos a disco, pero almacenables en memoria principal).
 - **Esquema combinado (Unix):** Combina indexación directa, multinivel y enlazada para mayor flexibilidad.

3.4 Gestión de espacio libre

Es necesario conocer el estado de ocupación de los bloques de los dispositivos para poder realizar las operaciones de asignación eficientemente.

Hay varias maneras de hacerlo:

- **Mapas de bist o vectores:** incluyen un bit por recurso existente (1 → libre, 0 → ocupado). Tiene varias ventajas: implementación fácil y sencillo de usar si la lista no está muy llena o fragmentada.
- **Lista de recursos libres:** se mantienen enlazados en una lista todos los recursos disponibles, manteniendo un apuntador que siempre está en memoria que apunta al primer elemento de la lista. Cuando se necesitan recursos se recorre la lista y se desenlazan. Solo es útil en sistemas fragmentados pero es ineficiente si hay muchos recursos libres.

Posible optimización: reunir los bloques en agrupaciones y mantener mapas de bits o listas de recursos para ellas. Cada bloque almacena *n*-1 direcciones libres para evitar fragmentación en ellas se usa un mecanismo de gestión de fragmentos.

Cuenta: cada entrada de la lista tiene un contador de nº de bloques que le sigue.

3.5 Recuperación, eficiencia y rendimiento

Eficiencia y rendimiento

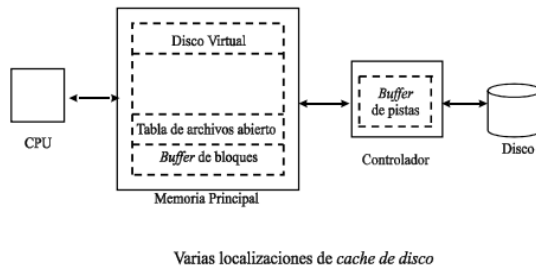
- Los discos suelen ser el principal cuello de botella del rendimiento del sistema
- La eficiencia depende de la asignación de disco y de la implementación de directorios utilizada.

Consulta condiciones aquí



do your thing

- Proporcionan mejor rendimiento



1. **Caché de disco:** secciones de M.P. con bloques usados
2. **Discos virtuales o discos RAM:** almacén temporal. Su contenido es controlado por el usuario

Recuperación

Como los archivos y directorios se mantienen en MP como en disco, el sistema debe asegurar que un fallo no genere pérdida de datos. Se puede tener un comprobador de consistencia o realizar copias de seguridad.

3.6 Algoritmos de bajo nivel de SA

- `iget` = devuelve un i-nodo de la tabla de i-nodos identificado previamente, si no está, lo carga en la tabla
- `iput` = libera un i-nodo de la tabla de i-nodos
- `namei` = convierte un pathname a un número de i-nodo
- `alloc` y `free` = asignan y liberan bloques de disco
- `ialloc` e `ifree` = asignan y liberan i-nodos de disco
- `bmap` = traducción de direcciones lógicas a físicas



Contenido de un superbloque:

- Tamaño SA.
- N° de bloques libres en el SA.
- Lista de bloques libres e índice al siguiente bloque libre.
- Tamaño lista i-nodos.
- N° i-nodos libres
- Lista de i-nodos libres y un índice al siguiente

Cuanto esta en memoria:

- Campos de bloques para las lista de bloque e i-nodos libres.
- Campo que indica si ha sido modificado.

3.7 Subsistemas de archivos UNIX

i-nodo

Es la representación interna de un archivo, cada archivo tiene asociado un *i-nodo*.

El *i-nodo* contiene el identificador del propietario, tipo de archivos, permisos de acceso, tiempos de acceso, contador de enlaces, tabla de contenido para las direcciones de los datos y el tamaño.

Tablas de *i-nodos*

El núcleo lee el *i-nodo* del archivo cuando opera con él.

Contenido:

- Estado del *i-nodo* en memoria y su estado: si está bloqueado, esperando...
- Nº del dispositivo lógico del SA que contiene el archivo.
- Nº del *i-nodo*.
- Punteros a otros *i-nodos* en memoria, se usa la estructura *hash* para enlazar los *i-nodos* cuya clave de acceso es: `nº disco lógico y nº de i-nodo`
- Contador de referencias al *i-nodo* actual.

Tabla de archivos

Mantienen la información del *offset* y los permisos de acceso.

Contenido de cada entrada:

- Puntero al *i-nodo* correspondiente de la tabla de *i-nodos*.
- Puntero lectura/escritura.
- Permisos de acceso.
- Modo de apertura del archivo.
- Contador de entradas de las tablas de descriptores de archivos asociados con la entrada.

Tabla de descriptores

Identifica los archivos abiertos por el proceso, hay una por proceso.

Cada entrada contiene un puntero a la entrada de la tabla de archivos correspondiente.

Las tres primeras entradas por defecto son: *stdin*, *stdout* y *stderr*.

UFS - Unix File System

Es una implementación específica de un sistema de archivos en Unix.

Objetivos:

- Dar soporte a distintos tipos de SAs a la vez.
- Tener diferentes particiones con diferentes SAs.
- Transparencia en el acceso a un SA remoto.
- Se puede añadir nuevos SAs de forma modular.

FFS - Fast File System

Esto no son apuntes pero **tiene un 10 asegurado** (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos Holandés con una garantía de hasta 100.000 euros por depositante. Consulta más información en [ing.es](https://www.ing.es)



Es una mejora del **UFS (Unix File System)**, diseñado para optimizar el rendimiento del sistema de archivos en Unix.

Particion dividida en uno o mas grupos consecutivos de cilindros.

Información del superbloque FFS: numero, tamaño y ocalización d elos grupos de cilindros + tamaño de bloque, nº de i-nodos y nº bloques.

Cada grupo de cilindros tiene: ñista de i-nodos y bloques libres y una copia duplicada del superbloque.

Tamaño minimo del bloque: `4KB`

Politica de asignación: situar los i-nodos de archivos de un mismo directorio en el mismo grupo de cilindros.

3.8 Llamadas al sistema

Sintaxis `descriptor archivo = open(<camino>, <opciones>, [<permisos>])`

Cauces: permiten la transferencia de datos y la sincronización entre procesos. Tienen una capacidad limitada y gestión FIFO.

En la ultima implementacion son bidireccionales y soportan comunicacion full-duplex.

Hay dos tipos:

Sin nombre

- Se crean con un pipe
- El proceso que los crea y sus descendientes pueden compartirlo.
- Son transitorios.
- Tiene asociado un i-nodo en la tabla.
- No tienen asociado un nombre de archivo.

Con nombre

- Se crear con `mknod`.
- Todos los procesos pueden acceder a él.
- Permanecen en el directorio.
- Tienen asociado el nombre de archivo.
- Tienen asociado un i-nodo en la tabla de i-nodos y en disco.

3.9 Montaje y desmontaje de archivos

`mount`: conecta sistemas de archivos existentes

`mount(<camino especial>, <camino directorio>, <opciones>)`

`umount`: lo desmonta-

El nucleo tiene una tabla de montaje con una entrada por cada SA montado.

Contiene:

- Nº de dispositivo que contiene el SA montado.
- Puntero a un buffer que contiene una copia del superbloque.
- Puntero al i-nodo raiz del SA montado.
- Puntero al i-nodo del directorio de punto de montaje.

Consulta condiciones aquí



do your thing

4. Disco

4.1 Estructura del disco

Disco → array de bloques (B0, B1, B2,..., Bn-1).

La información se referencia por una dirección formada por: unidad (nº de dispositivo, superficie o cara, pista, sector).

Existe un esquema de asignación a la dirección de un bloque lógico via dirección física.

El area de asignación mas pequeña es el bloque.

4.2 Planificación de disco

El SO puede mejorar el tiempo medio de servicio del disco.

Una petición se atiende en 3 fases:

1. Posicionamiento de la cabeza en la pista o cilindro.
2. Latencia: espera a que pase el bloque deseado.
3. Trasnferencia de los datos entre MP y disco.

$$T_{total} = T_{posicionamiento} + T_{latencia} + T_{transferencia}$$

La planificación intenta minimizar el tiempo de posicionamiento.

Si el disco esta ocupado, las pelitionen se encolan.

La información necesaria para una petición es la siguiente:

- Si la operación es de entrada o salida.
- Dirección de bloque.
- Dirección de memoria a donde copiar los datos a transferir.
- Cantidad de información a transferir.

Existen diferentes algoritmos de planificación de peticiones:

FCFS: procesa los elementos de la cola en orden secuencial, es esquitativa y tiene buen rendimiento si no hay demasiadas peticiones.

SSTF: se selecciona la petición que requiera menor movimiento de brazo desde su posición actual, tiempo de busqueda minimo.

SCAN: la cabeza recorre el disco de principio a fin y viceversa, sirviendo todas las peticiones para cada pista, favorece los trabajos con pistas cercanas.

C-SCAN: La cabeza del disco recorre las pistas en una sola dirección, sirviendo todas las solicitudes pendientes en su camino. Al llegar al extremo del disco (o la última solicitud), la cabeza regresa directamente al inicio del disco sin atender solicitudes durante el retorno, y continúa sirviendo en la misma dirección. Esto asegura tiempos de respuesta más uniformes para todas las solicitudes.

LOOK: Similar a SCAN, pero la cabeza solo recorre las pistas donde hay solicitudes, sin llegar necesariamente al principio o al final del disco si no

hay solicitudes en esos extremos. Esto optimiza los movimientos y reduce el tiempo total.

C-LOOK: La cabeza recorre únicamente las pistas que contienen solicitudes, moviéndose en una sola dirección hasta la última solicitud pendiente. Al llegar a esta, la cabeza salta directamente al inicio de las solicitudes pendientes en la dirección opuesta, sin recorrer pistas innecesarias.



Selección de un algoritmo de planificación:

Si la cola de peticiones esta practicamente **vacía** → **cualquier** algoritmo sera valido.

Si el servicio de peticiones esta muy influenciado por el metodo de asignación:

- Asignación contigua: peticiones que reducen el tiempo de posicionamiento.
- Asignación no contigua (enlazado o indexado): mayor aprovechamiento de disco pero mayor tiempo de posicionamiento.

4.3 Gestión de disco y del espacio de intercambio.

Gestión de disco

Particion de disco: división lógica en la unidad de almacenamiento en el cual se alojan y organizan los archivos.

Formateo de disco, establece bloque de arranque del sistema. Hay dos tipos:

- Físico: ajusta los sectores a nivel de pista.
- Lógico: establece información que el SO necesita conocer.

Gestión del espacio de intercambio

La memoria virtual requiere el uso de disco como extensión de memoria principal. Su uso depende de los algoritmos de gestión de los algoritmos de gestión del SO.

Hay dos formas de asignar:

- En el propio SA (*windows*)
- En una partición independiente que no utiliza SA ni estructura de directorios.

4.4 Eficiencia y seguridad de disco

Técnica de entremezclamiento (*striping*) de disco

El grupo de discos se trata como una unidad. Cada bloque esta formado por subbloques, cada uno almacenado en un disco diferente.

Los bloques realizan el posicionamiento y transfieren sus bloques en paralelo.

RAID(Redundant Array of Independent Disks)

Esto no son apuntes pero **tiene un 10 asegurado** (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos Holandés con una garantía de hasta 100.000 euros por depositante. Consulta más información en [ing.es](https://www.ing.es)



Tiene mejoras de rendimiento y seguridad. Hay dos tipos:

- Sombra (*shadowing*): mantiene duplicados de cada disco.
- Paridad entremezclada de bloques: los datos se escriben en cada disco del array en un bloque y hay un bloque extra en otro disco.

Los datos dañados de un disco se obtienen a partir de los datos del resto de discos.

Consulta
condiciones aquí



do your thing

WUOLAH