SISTEMA DE FICHEROS

Capítulo 6

Contenidos

- **Introducción**
- **Interfaz con el Sistema de Archivos**
  - Archivo
  - Modos de trabajo
  - Método de acceso
  - Directorios
- **Implementación del Sistema de Archivos**
  - Organización del sistema de archivos
  - Métodos de asignación
  - Gestión del espacio libre en disco
  - Implementación de directorios
  - Ejemplos de estructura de discos

Conceptos básicos

- El computador para ejecutar programas necesita tenerlos en memoria principal (almacenamiento primario).
- Como la capacidad de memoria principal es reducida y además es volátil es necesario un Sistema de Almacenamiento Secundario (SAS) para contener los programas y datos que se vayan a utilizar.

Conceptos básicos

- El SO es el software encargado de la gestión del SAS. Dispone de un módulo que realiza un segundo nivel de gestión de dispositivos periféricos:
  - Abstrae las propiedades físicas de los dispositivos de almacenamiento
  - Proporciona una interfaz a los usuarios de acceso a la información
  - Administra el espacio libre
  - Realiza la asignación del almacenamiento
- El sistema de archivos está formado por:
  - Colección de archivos. Cada uno de los cuales contiene datos relacionados.
  - Estructura de directorios. Organiza todos los archivos del sistema y proporciona información sobre ellos.
  - Particiones. Permite separar grandes colecciones de directorios.

Interfaz con el Sistema de Ficheros

fichero

Concepto

- Podemos definir un archivo o fichero como:
- Una abstracción para la manipulación de memoria secundaria ofrecida por el sistema operativo.
- Una colección de información relacionada que reside en el almacenamiento secundario a la cual se le asigna un nombre.
- Una secuencia de bits, bytes, líneas o registros con un significado definido por el creador y el usuario.

Interfaz con el Sistema de Ficheros

fichero

Estructura

- En un archivo se distingue su estructura lógica y su estructura física
- Estructura lógica:
  - Secuencia de bytes. Por ejemplo, un archivo de texto es una secuencia de caracteres organizados en líneas.
  - Secuencia de registros. Un registro puede ser una línea de texto, información de tamaño fijo o información de tamaño variable.
  - Estructura compleja. Por ejemplo, los archivos objeto, los archivos ejecutables, los archivos de formato gráfico.

Interfaz con el Sistema de Ficheros

fichero

Estructura

- Estructura física:
  - Secuencia de bloques. Los archivos se pueden considerar como una secuencia de bloques (sectores).
  - Se debe realizar una conversión de bloques lógicos a bloques físicos.
  - Se produce fragmentación interna.

Interfaz con el Sistema de Ficheros

fichero

Atributos

- Nombre del archivo: En algunos sistemas operativos pueden existir más de uno, distinguen entre mayúsculas y minúsculas.
- Tipo de archivo: Información necesaria en sistemas que soportan más de un tipo.
- Ubicación: Información sobre el dispositivo y bloques de memoria secundaria que le dan soporte físico al archivo.
- Tamaño: El número de bytes de la información que contiene el archivo.
- Propietario: Identificador del creador del archivo.
- Protección: Información sobre que usuarios pueden acceder al archivo y con que permisos (lectura, escritura, ejecución, etc.).
- Fechas de creación, último acceso, última modificación
- Información: Es el contenido del archivo.

Interfaz con el Sistema de Ficheros

fichero

Operaciones

- Crear archivo: El archivo se crea sin datos. Hay que encontrar espacio en el disco y anotar su existencia. Ejemplo UNIX: fd=creat (filename, mode)
- Leer archivo: Ejemplo UNIX: read (fd, buffer, nbytes)
- Escribir en archivo: Ejemplo UNIX: write (fd, buffer, nbytes)
- Borrar archivo: Ejemplo UNIX: unlink (filename)
- Rebobinado de archivo: Ejemplo UNIX: lseek (fd, offset, where)
- Truncar archivo: Ejemplo UNIX: fd=creat (filename, mode)
- Obtener atributos: Ejemplo UNIX: fd=fstat (filename, starbuf)

Interfaz con el Sistema de Ficheros

fichero

Operaciones

- Utilizando y combinando las llamadas al sistema básicas, se pueden realizar otras operaciones más complejas:
- *Copiar un archivo (cp)*
- *Concatenar archivos (cat)*
- *Renombrar archivos (mv)*

Interfaz con el Sistema de Ficheros
tipos de fichero

- Un SO se puede diseñar para que reconozca y manipule diferentes tipos de archivos:
  - Si reconociera muchos tipos el SO sería más complejo ya que tendría que tener los conversores de tales tipos. Esto le da facilidades al usuario o programador.
  - Si reconociera pocos tipos el SO sería más sencillo y el usuario o programador tendrían que tener los conversores para poder acceder a esos archivos.

Interfaz con el Sistema de Ficheros
tipos de fichero

- Una técnica común para implementar los tipos de archivos es incluir el tipo como parte del nombre. Por ejemplo, MSDOS divide el nombre del archivo en nombre y extensión, siendo la extensión la que indica el tipo de archivo (“com”,”exe”,”bat”).
- Otra técnica es indicar en un campo el tipo de archivo. Por ejemplo, UNIX emplea un número mágico para indicar el tipo de archivo.

Interfaz con el Sistema de Ficheros
tipos de fichero

- Para cada SO se puede establecer una clasificación de los archivos desde el punto de vista funcional. Por ejemplo, para UNIX se distinguen los siguientes archivos:
  - Archivos ordinarios o regulares. Son archivos que contienen información introducida por un usuario, programa de aplicación o programa de utilidad del sistema.
  - Archivos directorios. Contienen una lista de nombres de archivo y punteros a nodos-i asociados. Son archivos ordinarios especiales ya que tienen unos privilegios especiales de protección de forma que solo el sistema de archivos puede escribir en ellos.

Interfaz con el Sistema de Ficheros
tipos de fichero

- Archivos especiales. Usados para acceder a los dispositivos. Se pueden distinguir dos tipos:
  - Caracteres. Permiten modelar dispositivos orientados a caracteres (impresoras, redes, terminales, etc.).
  - Bloques. Permiten modelar dispositivos orientados a bloques (discos, cd-rom, etc.)
- Pseudoarchivos. Son archivos que modelan los mecanismos de UNIX para la comunicación entre procesos (tubos, tubos con nombre, sockets).

Interfaz con el Sistema de Ficheros
modos de trabajo

Sesiones

- Para utilizar un archivo hay que definir una sesión con las llamadas al sistema open (abrir) y close (cerrar).
- **Abrir: fd = open(filename, mode) (UNIX)**
- Buscar el archivo en la estructura de directorios y llevar sus atributos a una entrada de una tabla de archivos abiertos en memoria. También registra algunos atributos adicionales como:
  - Puntero de posición actual.
  - Contador de aperturas de archivos abiertos.
  - Ubicación del archivo en disco. La información necesaria para localizar el archivo en el disco se mantiene en la memoria principal.
- El contenido del archivo es llevado parcialmente a buffers en memoria.

Interfaz con el Sistema de Ficheros
modos de trabajo

Sesiones

- **Cerrar: close(fd) (UNIX)**
- Liberar la entrada correspondiente en la tabla de archivos abiertos. Refrescar todos los buffers de memoria sobre el disco.
- Para acceder a su contenido hay que utilizarlas llamadas read y write.

Interfaz con el Sistema de Ficheros
modos de trabajo

Mapeado en memoria

- Para trabajar en el archivo hay que llevarlo a un espacio de direcciones de memoria virtual.
- El archivo es mapeado en el espacio de direcciones de uno o más procesos.
- Para leer y escribir en disco no hace falta operaciones como read o write, sino operaciones “normales” de lectura y escritura en memoria (=, :=).
- El sistema de memoria virtual se encarga de refrescar la imagen del disco con la imagen de memoria.
- Se utiliza con paginación o segmentación de memoria.
- Problemas
  - Un proceso puede intentar leer del disco el archivo cuando éste ha sido modificado por otro en memoria.
  - Un archivo puede ser mayor que el espacio de direcciones virtuales
  - No se conoce la longitud exacta del archivo

Interfaz con el Sistema de Ficheros
métodos de acceso

- Técnica o modalidad empleada para acceder al contenido (información, datos) de un archivo. Un SO puede trabajar con un tipo o con varios.
  - Acceso secuencial
  - Acceso directo
  - Acceso indexado

Interfaz con el Sistema de Ficheros
métodos de acceso

Acceso secuencial

Se procesa la información de un archivo en el mismo orden en que esta figura en él, esto es, un registro tras otro. Los registros tienen un tamaño fijo.

- Existe un puntero de posición que avanza automáticamente al hacer una operación de lectura/escritura.
- Las operaciones de lectura/escritura no tienen como parámetro el número de registro y están ligadas siempre a la posición actual del puntero de posición.

  Leer siguiente: read(fd, buffer, nbytes)

  Escribir siguiente: write(fd, buffer, nbytes)
- Puede existir una operación de rebobinado que sitúe el puntero de posición al inicio del archivo.
- Es la forma de acceder a la información empleada por programas como editores, compiladores, etc.
- Este tipo de acceso funciona tanto en dispositivos de acceso secuencial como en dispositivos de acceso directo..

Interfaz con el Sistema de Ficheros
métodos de acceso

Acceso directo

- El archivo se contempla como una serie numerada de bloques o registros.
- No existen restricciones en cuanto al orden de lectura/escritura en los registros.
- Es necesario especificar, como un parámetro, el número de registro n sobre el que se requiere realizar la operación.

  read nreg: read(fd, buffer, nbytes, nreg)

  write nreg: write(fd, buffer, nbytes, nreg)
- Los archivos de acceso directo son muy útiles para realizar accesos, de forma selectiva, en grandes cantidades de información (bases de datos).
- Se emplea con dispositivos de acceso directo.

Interfaz con el Sistema de Ficheros
métodos de acceso

Acceso indexado

- Se accede a sus registros través de una clave.
- El conjunto de claves y los punteros hacia los registros que contienen dicha clave es lo que se conoce como índice del archivo y suele estar cargado en memoria.
- Problemas
  - Con archivos de gran tamaño el archivo índice puede ser demasiado grande para conservarse en memoria principal.
  - Búsqueda de la clave en el archivo índice excesivamente larga.

Interfaz con el Sistema de Ficheros
directorios

Concepto

- Debido a que los sistemas de archivo almacenan gran cantidad de información es necesario organizarlos. Esta organización se realiza en dos partes: particiones y directorios.
- Los discos físicos se estructuran en zonas de tamaño y ubicación fijos denominadas particiones/minidiscos/volúmenes.
- Las particiones se estructuran en directorios. Los directorios se pueden implementar como:
  - Una estructura de datos al principio de cada partición.
  - Una estructura de datos dentro de un archivo.
- Los directorios pueden verse como una tabla de entradas que asocia nombres simbólicos a archivos y almacenan información relacionada con los archivos (ubicación, atributos, etc.).

Interfaz con el Sistema de Ficheros
directorios

Concepto

La organización de un directorio debe proporcionar:

- Esquema de nombrado de archivos: Un modo de denominar a los archivos que sea conveniente para los usuarios. Algunas características deseables son:
  - El nombre no debe hacer referencia a la unidad física (A:, B). Los nombres no deben depender de la unidad física.
  - El nombre no debe ser demasiado largo.
  - Dos usuarios pueden tener el mismo nombre para archivos diferentes. Los usuarios no deben colisionar en el nombre de un archivo.
  - Un archivo puede tener varios nombres.
- Esquema de agrupamiento de archivos: Un modo de agrupación lógica de archivos por propiedades como son:
  - Pertenecer a la misma aplicación.
  - Ser del mismo tipo de archivo (p.e. formato gráfico).
- Eficiencia: Poder localizar un archivo con rapidez.

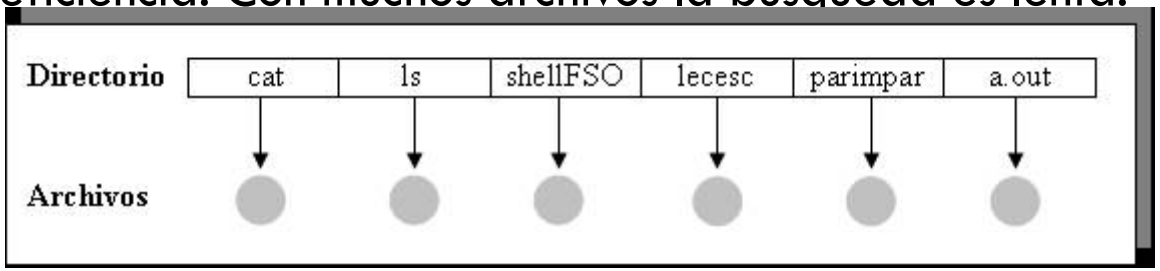
Interfaz con el Sistema de Ficheros
directorios

Estructura

- La estructura de un directorio indica cómo están organizados los archivos en el sistema de almacenamiento de archivos

Directorio de un solo nivel

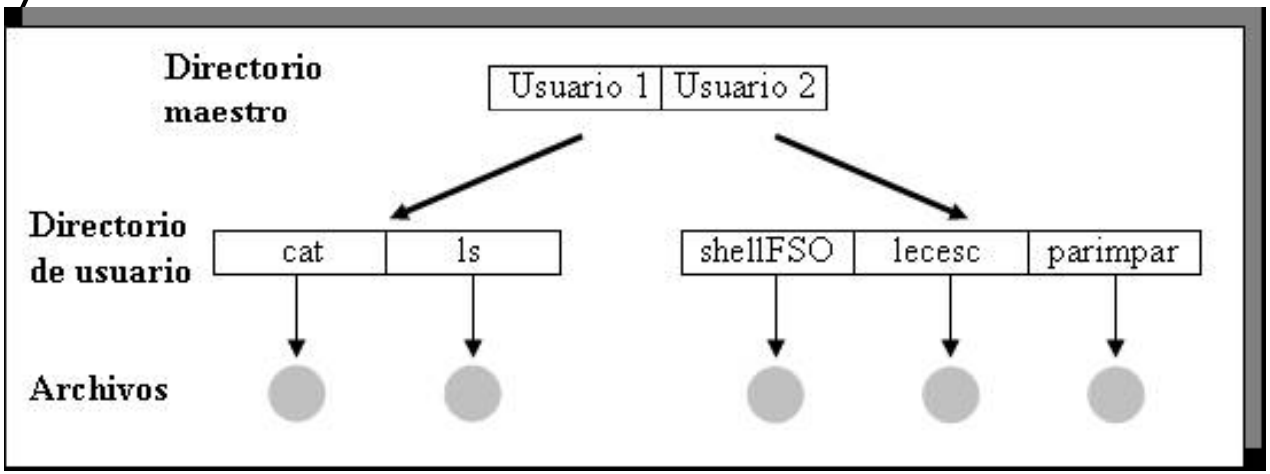
- Todos los archivos están contenidos en un único directorio. Todos los usuarios comparten el mismo directorio.
- Ventaja: Fácil de manejar y mantener.
- Desventajas:
  - Nombres únicos. Dos usuarios no pueden darle el mismo nombre a un archivo.
  - Baja eficiencia. Con muchos archivos la búsqueda es lenta.



Interfaz con el Sistema de Ficheros
directorios

Directorio de dos niveles

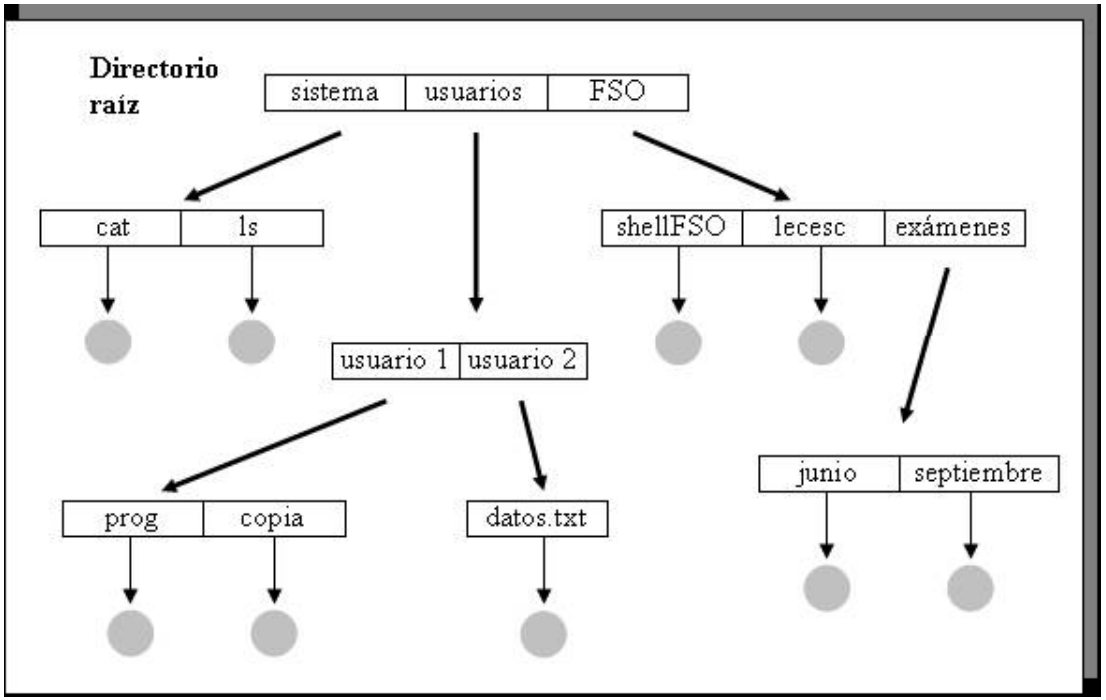
- Un directorio para cada usuario. Normalmente un usuario sólo ve su directorio. Ej: CP/M.
- Rutas de búsqueda por defecto para manejar los archivos del sistema.
- Desventaja: No se pueden compartir archivos.
- Ventajas:
  - Mejora la eficiencia: sólo busca en el directorio de usuario.
  - No hay colisión en los nombres de archivos.



Interfaz con el Sistema de Ficheros
directorios

Directorio con estructura de árbol

- Un directorio contiene un conjunto de archivos y/o directorios. Un directorio es un tipo especial de archivo. Ej: MS-DOS
- Hay un directorio raíz para cada partición y subdirectorios para cada usuario, aplicación, sistema operativo, etc.



Interfaz con el Sistema de Ficheros
directorios

Directorio con estructura de árbol

Nombre de ruta

- Nombres de rutas absolutos. Nombran un archivo partiendo del directorio raíz.
- Nombres de rutas relativos. Nombran al archivo a partir del directorio actual.

- Ventajas
  - Facilidad para agrupar archivos. Usuarios puede definir sus directorios y subdirectorios.
  - Usuarios diferentes pueden tener archivos con el mismos nombre. Ruta única para cada archivo.
  - Facilidad para acceder a los archivos de otros usuarios. Se puede acceder a los archivos:
    - Especificando el nombre de ruta del archivo.
    - Utilizando la operación cambiar de directorio.
    - Definiendo una ruta de búsqueda con varias opciones.

Interfaz con el Sistema de Ficheros
directorios

Directorio con estructura de árbol

- Desventaja: Nombres de archivos pueden ser largos.
- Posibles soluciones:
  - Utilizar rutas de búsqueda por defecto.
  - Utilizar nombres de rutas relativos.
  - Mantener un archivo que contenga los nombres y la localización de todos los programas a los que se ha accedido recientemente que han aparecido.
  - Eliminar la partición del nombre de ruta. En UNIX el nombre de ruta no especifica la partición.

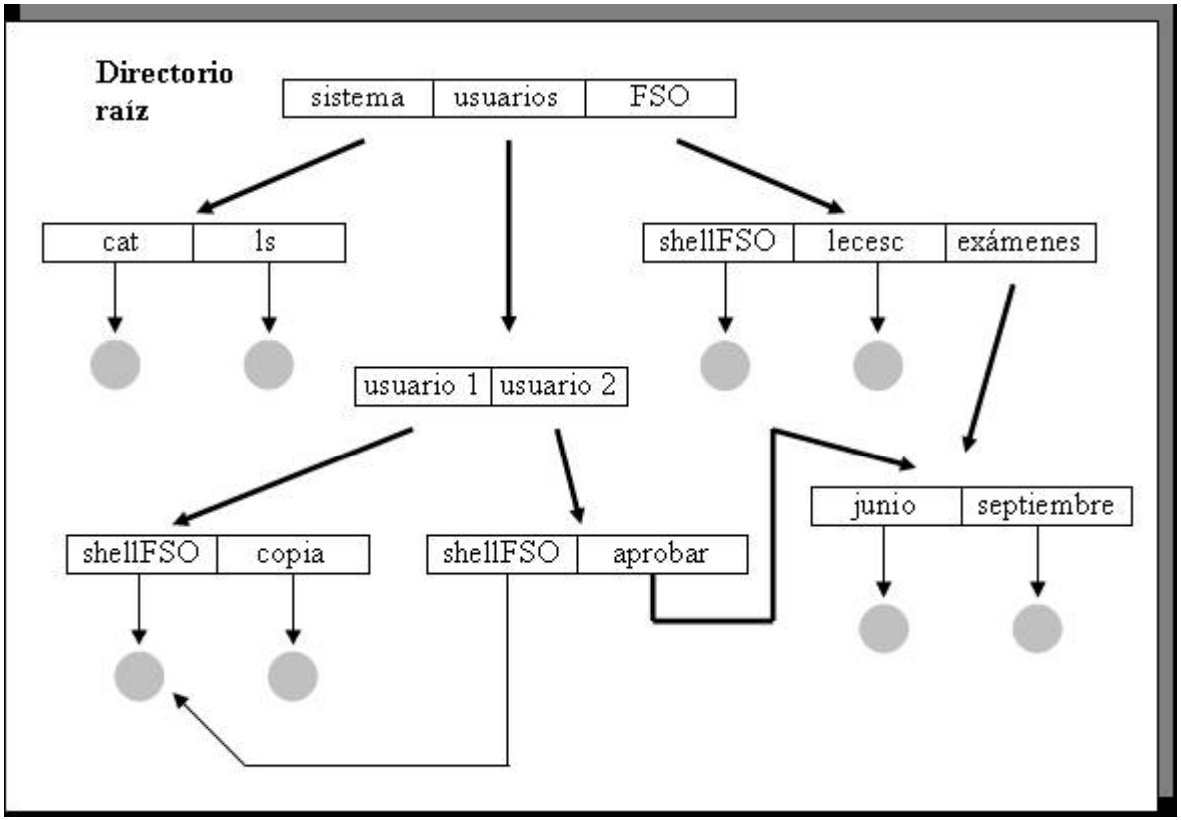
Interfaz con el Sistema de Ficheros
directorios

Directorio de grafo acíclico

- Permite archivos y directorios con varios nombres. Ej: UNIX.
- Aplicación: Compartir archivos entre usuarios.
- Posibles implementaciones
  - Duplicar todos los archivos de los directorios que los comparten.
  - Enlaces: Puntero o referencia a un archivo o directorio.
    - Enlaces físicos
      - El archivo sólo se elimina del disco cuando se borran todos los enlaces (todas las entradas de directorio que lo referencian).
      - Sólo se permite (salvo al administrador) enlazar archivos con archivos (no directorios).
    - Enlaces simbólicos
      - El archivo se elimina cuando se borra el enlace físico. Si permanece el enlace simbólico provoca errores al tratar de accederlo.
      - Se puede hacer con archivos y directorios, existe la posibilidad de ciclos.

Interfaz con el Sistema de Ficheros
directorios

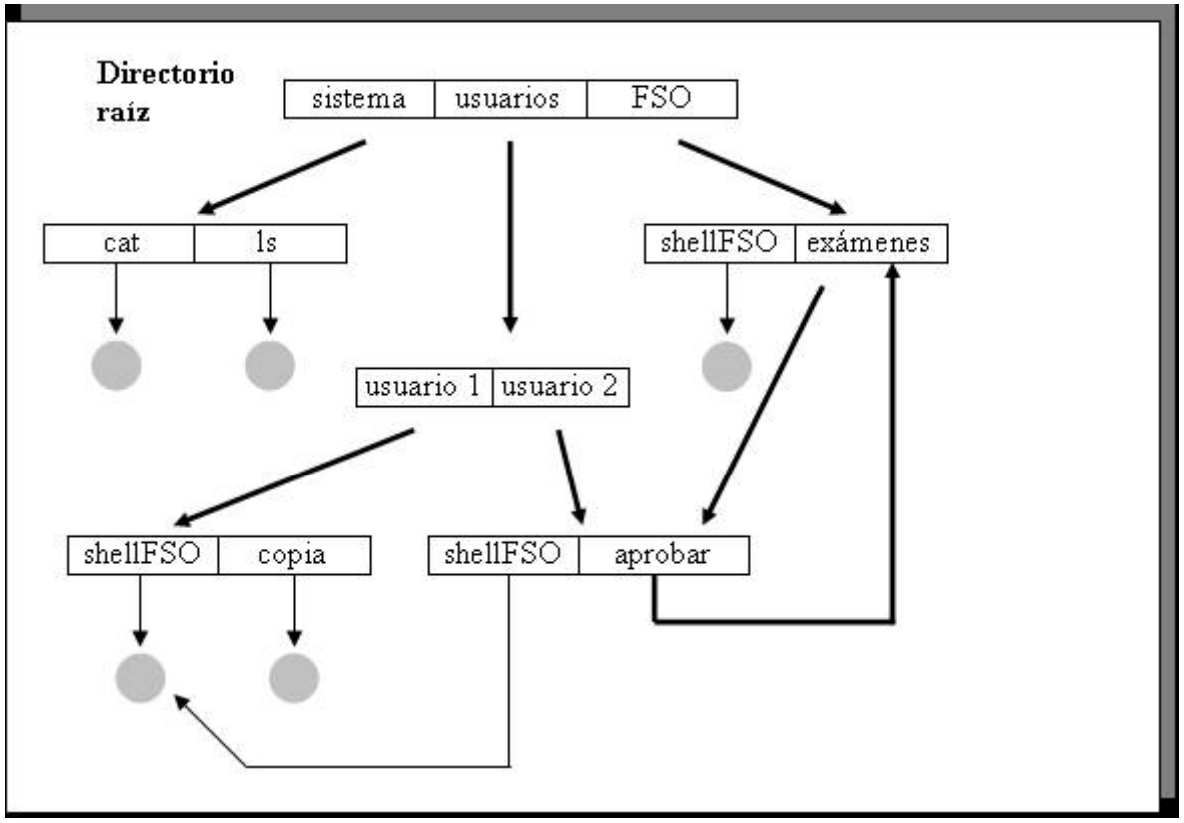
Directorio de grafo acíclico



Interfaz con el Sistema de Ficheros
directorios

Directorio de grafo general

- Es un grafo con posibilidad de ciclos.
- Aparecen cuando los directorios pueden tener varios enlaces.
- Problemas
  - Evitar bucles infinitos en algoritmos de búsqueda de archivos.
  - Borrado de archivos.



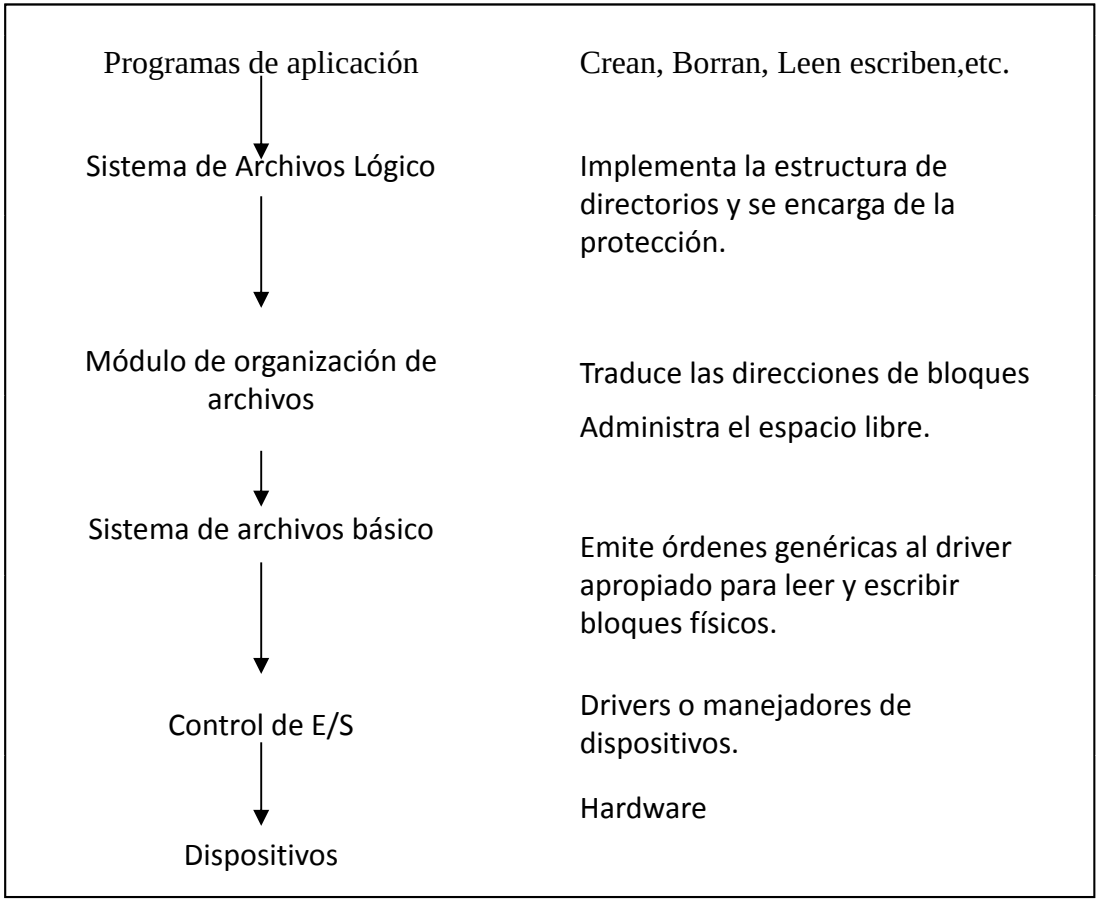
Interfaz con el Sistema de Ficheros
directorios

Operaciones

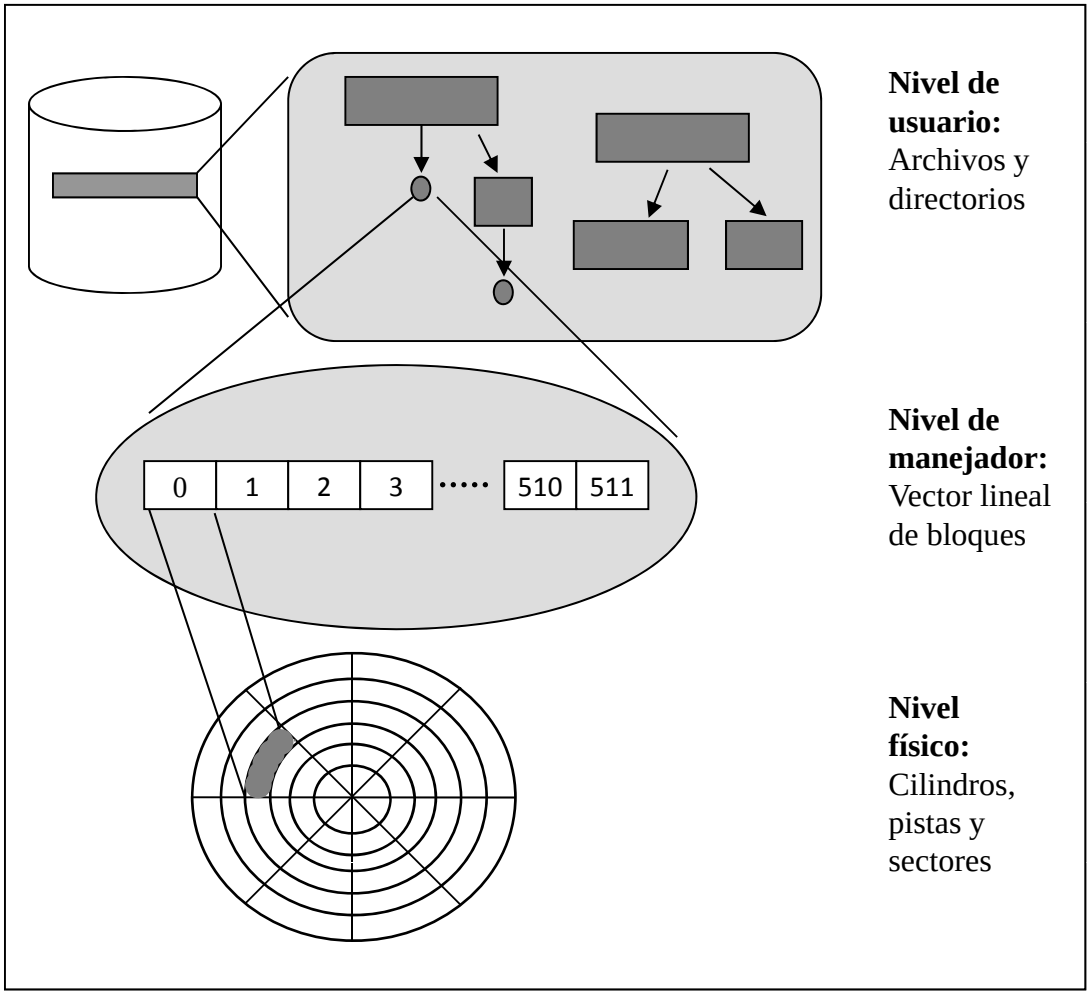
- Un directorio no es más que un tipo abstracto de datos que representa a un conjunto de archivos sobre el que se ha de poder realizar diferentes operaciones:
  - Insertar entradas/Crear un archivo. Al crear un archivo, debe añadirse una entrada al directorio.
  - Borrar entradas/Borrar un archivo. Al borrar un archivo, debe eliminarse una entrada del directorio.
  - Buscar una entrada/Buscar un archivo. Cuando un usuario o aplicación referencia a un archivo, debe buscarse en el directorio la entrada correspondiente al archivo.
  - Listar todas las entradas de un directorio. Puede solicitarse todo el directorio o una parte.

Implementación del sistema de ficheros
organización

- El sistema de archivos se compone de varios niveles en el que cada nivel aprovecha las funciones de los niveles inferiores para crear nuevas funciones que se emplearán en los niveles superiores.



Implementación del sistema de ficheros
organización

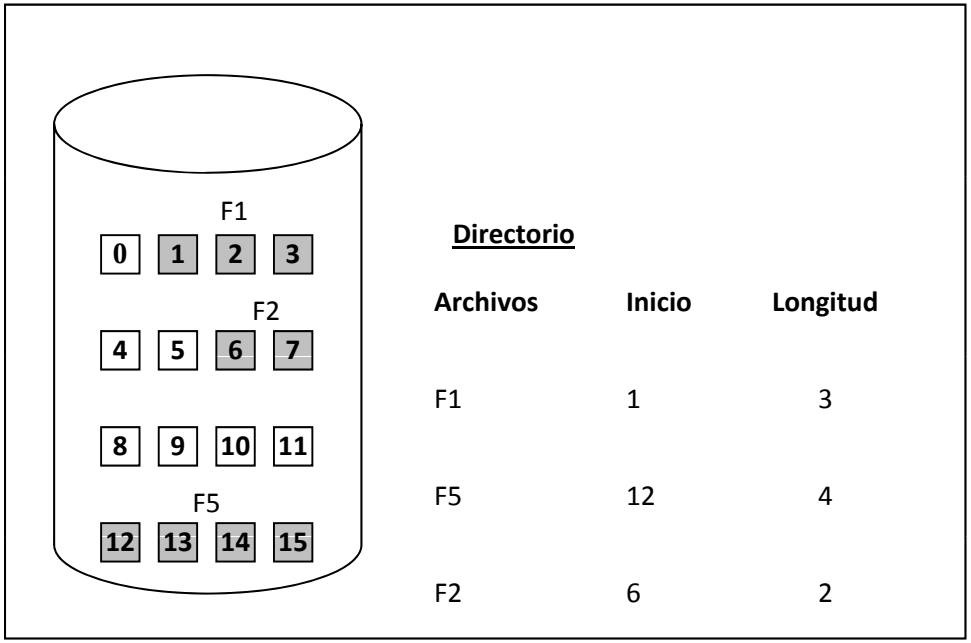


Implementación del sistema de ficheros
métodos de asignación

Desde el punto de vista de la implementación un archivo es una colección de bloques. Existen varias técnicas que permiten implementar esa colección de bloques de modo que el espacio se aproveche de forma eficaz y se pueda acceder rápidamente a ellos.

Asignación contigua

- Cada archivo ocupa un conjunto de bloques consecutivos en el disco.
- Un archivo queda definido por el primer bloque y su tamaño en bloques



Implementación del sistema de ficheros
métodos de asignación

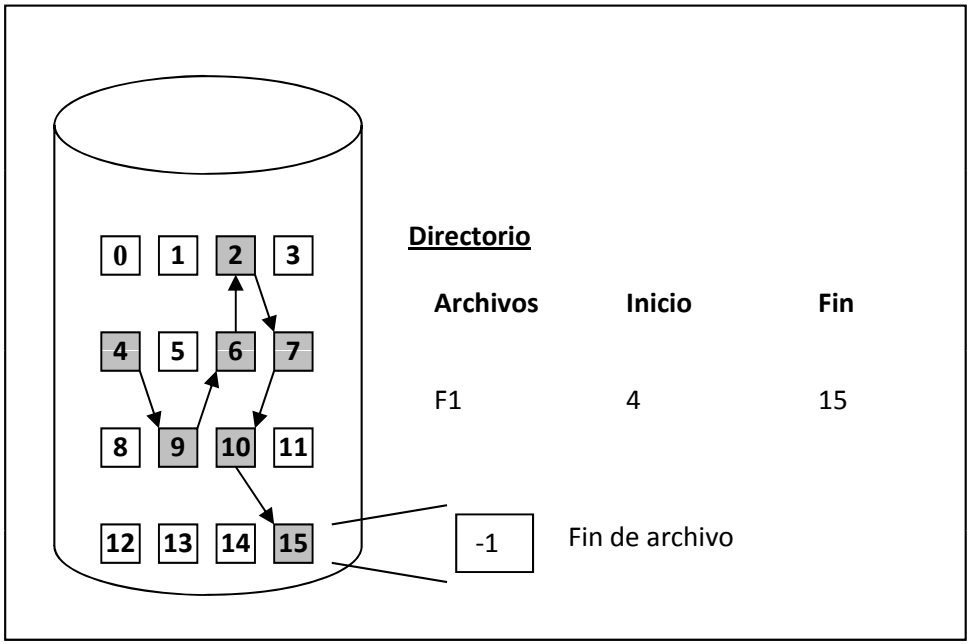
Asignación contigua

- Ventaja: Rapidez de acceso ya que para acceder a bloques consecutivos no hace falta mover el cabezal del disco.
- Métodos de gestión: Algoritmos: primer, mejor y siguiente hueco.
- Desventajas:
  - El crecimiento de los archivos está limitado.
  - Posible solución: Reservar un número de bloques superior al requerido inicialmente (Producirá fragmentación interna y un escaso aprovechamiento del disco)
  - Fragmentación externa
  - Posible solución: Compactación.

Implementación del sistema de ficheros
métodos de asignación

Asignación enlazada

- Cada archivo es una lista enlazada de bloques.
- La entrada de directorio da el número del primer bloque.
- Cada bloque tiene un puntero al siguiente y una zona de datos.



Implementación del sistema de ficheros
métodos de asignación

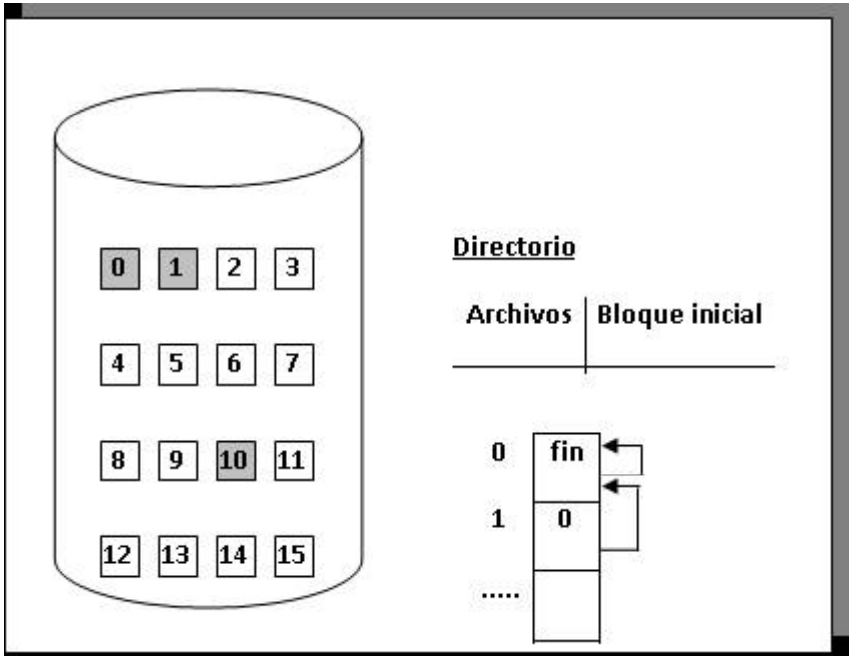
Asignación enlazada

- Ventajas
  - No limita el crecimiento de los archivos.
  - Es fácil el aumento y reducción del tamaño de los archivos
  - No existe fragmentación externa por lo que no existe necesidad de compactación.
- Desventajas
  - Puede ser utilizado de manera efectiva para acceso secuencial pero el acceso aleatorio es costoso de implementar.
  - Elevado número de desplazamientos del cabezal del disco.
  - Baja fiabilidad: Debido a punteros mal calculados.

Implementación del sistema de ficheros
métodos de asignación

FAT de MS-DOS (File Allocation Table)

- Es una variante de la asignación enlazada en la que los punteros no se encuentran en el bloque, sino en una estructura de datos separada (FAT) localizada en los primeros bloques del disco.



Implementación del sistema de ficheros
métodos de asignación

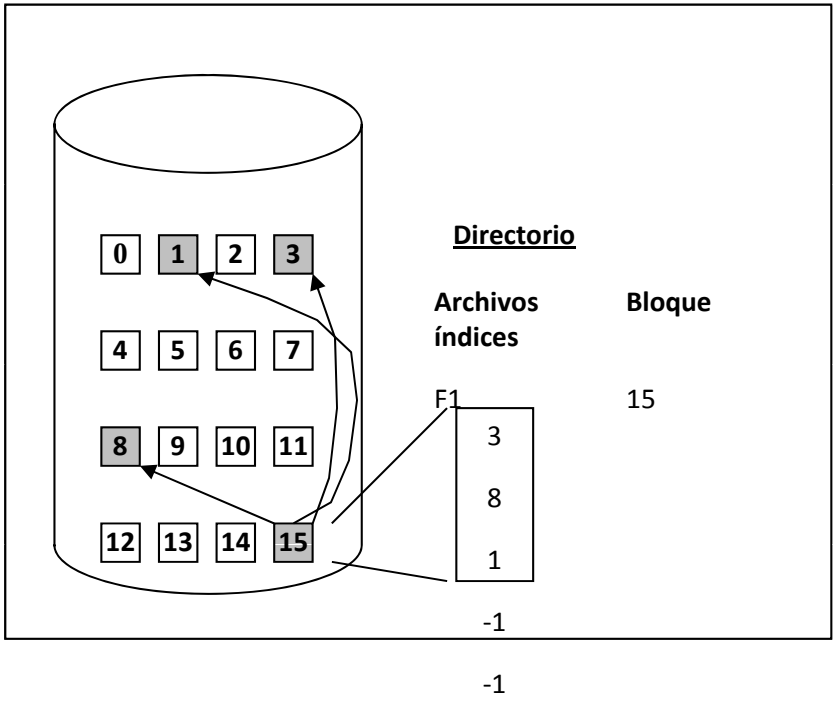
FAT de MS-DOS (File Allocation Table)

- Ventajas
  - Los bloques solo contienen datos
  - Si la FAT se coloca en una cache en memoria mejora significativamente el acceso aleatorio o directo. No obstante, para discos grandes no cabe toda la FAT en la cache
- Desventajas
  - Ocupa memoria principal o memoria caché.
  - Diseño poco elegante: en una misma estructura de datos se mezcla información de diferentes archivos.
  - Baja fiabilidad: la pérdida de un bloque de la FAT puede ocasionar la perdida del disco entero

Implementación del sistema de ficheros
métodos de asignación

Asignación indexada

- Cada archivo tiene un bloque de índice donde hay ubicado un vector con todos los punteros a los bloques del archivo.
- La i-ésima entrada del vector contiene el puntero al i-ésimo bloque.



Implementación del sistema de ficheros
métodos de asignación

Asignación indexada

- Ventajas
  - El acceso aleatorio se implementa eficientemente.
  - Diseño elegante: cada archivo tiene su estructura de datos separada.
  - No hay fragmentación externa.
- Desventajas
  - Con pocos bloques, el bloque de índice supone un desperdicio importante de espacio.
  - El tamaño máximo del archivo está limitada por el número de punteros que cabe en un bloque.
  - Fragmentación interna en los bloques índice.

Implementación del sistema de ficheros
métodos de asignación

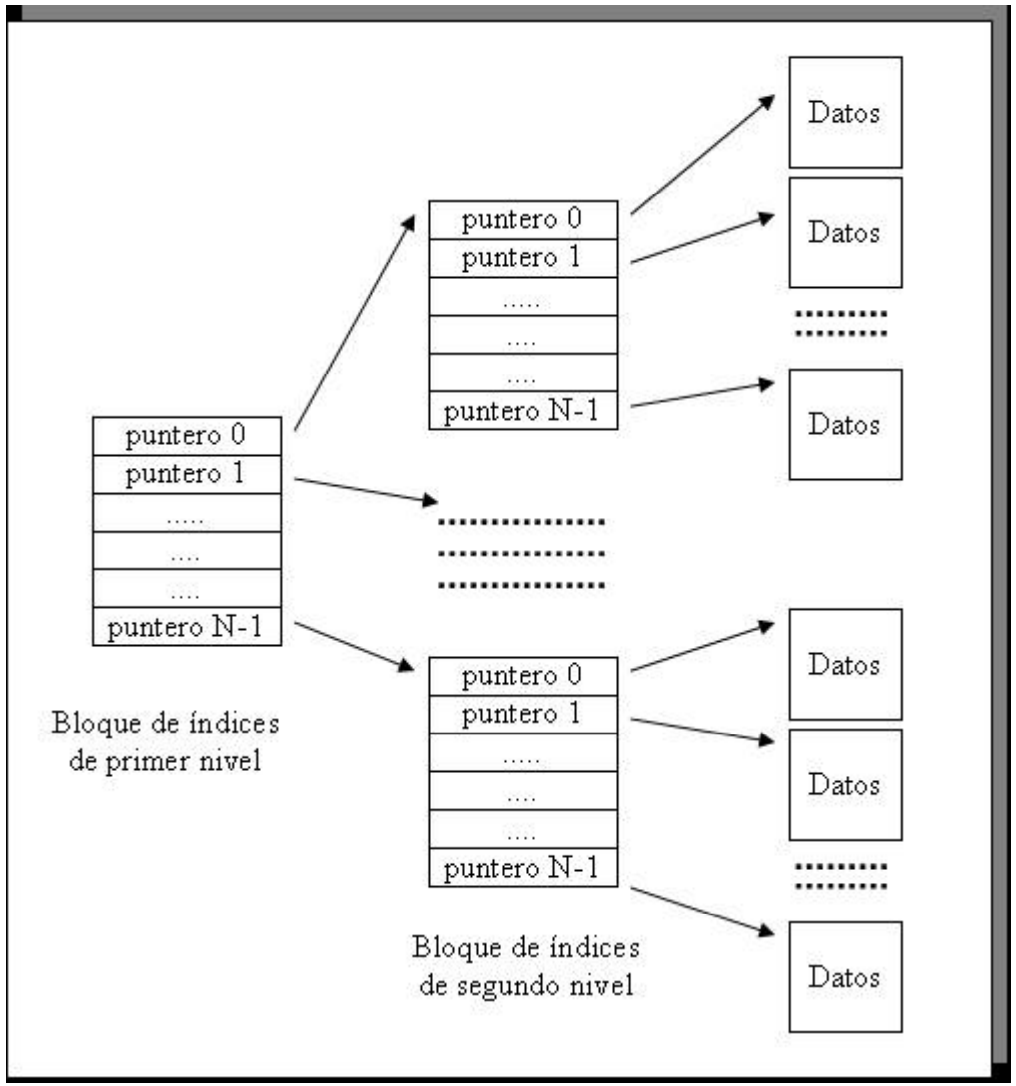
Variantes de la asignación indexada

- Para poder implementar archivos grandes se necesita más de un bloque de índices.
- Los posibles esquemas son:
  - Bloques de índice enlazados. La última posición del bloque índice puede ser un NULL o bien un puntero a otro bloque de índices si el archivo fuera grande.

Implementación del sistema de ficheros
métodos de asignación

Variantes de la asignación indexada

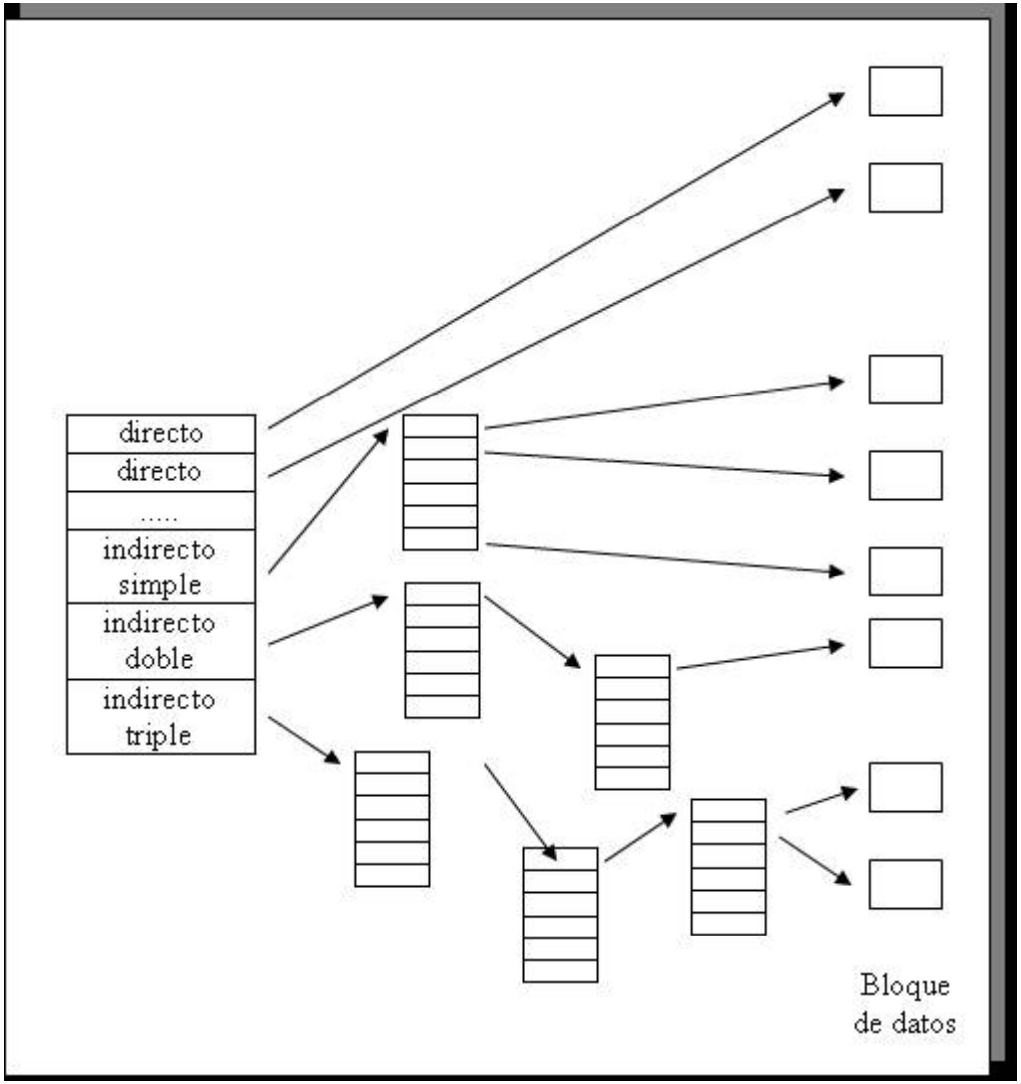
- Indexación multinivel. Se emplea un bloque índice de primer nivel que apunte a un conjunto de bloques índice de segundo nivel, que a su vez apuntan a bloques de datos.



Implementación del sistema de ficheros
métodos de asignación

Variantes de la asignación indexada

- Esquema combinado. El bloque índice contiene punteros a bloques de datos, punteros a índices de primer nivel, punteros a índices de segundo nivel y punteros a índices de tercer nivel (Ej: UNIX).

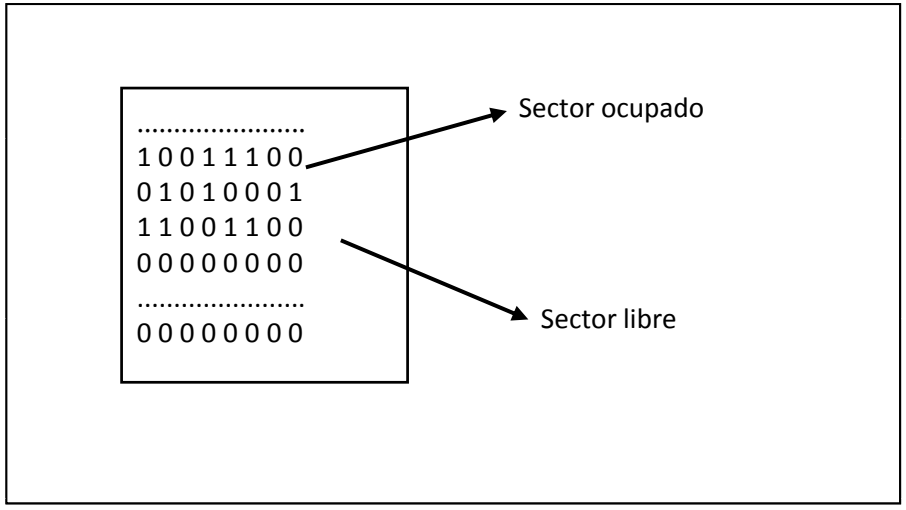


Implementación del sistema de ficheros
gestión de espacio libre

- Para llevar a cabo cualquiera de las técnicas de asignación anteriores, es necesario saber qué bloques del disco están disponibles. Para ello hay que mantener una estructura de datos con todos los bloques no asignados a archivos.

Mapa de bits

- Es un vector de bits en el que cada bit indica el estado de un bloque en disco:
- bit[i]=0 bloque[i] libre y bit[i]=1 bloque[i] ocupado
- Facilidad de asignación contigua.
- Tamaño reducido (facilidad de tenerlo completo en memoria).

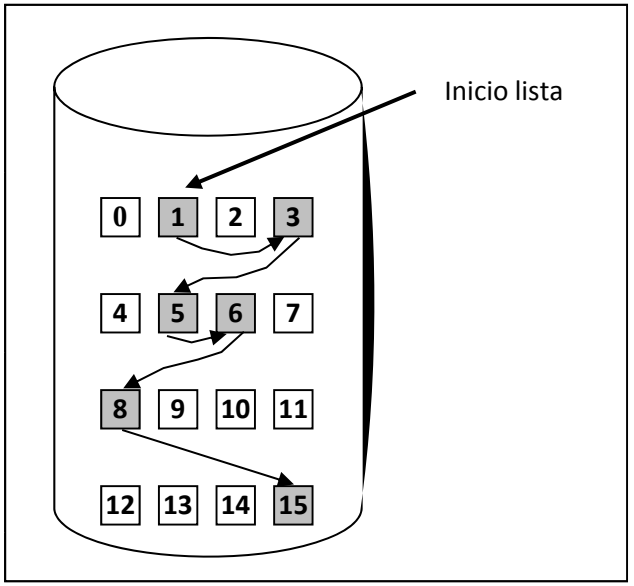


Implementación del sistema de ficheros
gestión de espacio libre

Lista enlazada

Lista de bloques que contienen punteros a los bloques libres.

- Dificultad en la asignación contigua.
- Tamaño variable (máximo cuando el disco está vacío). Se mantiene en memoria únicamente el primer bloque de la lista enlazada.



Implementación del sistema de ficheros
implementación de directorios

- Un directorio es una estructura de datos que consiste en una colección de entradas de directorio.
- Cada entrada de directorio es un registro en el que consta:
  - El nombre del archivo.
  - Los atributos del archivo (o una referencia).
  - Una referencia a los bloques de datos del archivo

Implementación del sistema de ficheros
implementación de directorios

- Existen dos posibles estructuras:
- ***Lista lineal*** de nombres de archivo con punteros a bloques de datos. Para crear un archivo nuevo, primero hay que examinar el directorio para comprobar que ningún archivo existente tenga el mismo nombre. Luego se añade una nueva entrada al final de la lista. Para eliminar un archivo, lo buscamos en el directorio y liberamos el espacio que tiene asignado.
  - Ventaja: Fácil de programar
  - Desventaja: Lenta para buscar una entrada.
- ***Tabla de dispersión*** (hash): Lista lineal + tabla de códigos.
  - La función hash le asigna un código a cada nombre de archivo.
  - La tabla de códigos es un vector, indexado para códigos, que proporciona un puntero a la entrada de directorio de cada código.
  - Mejora el tiempo de búsqueda.

Implementación del sistema de ficheros
implementación de directorios

Directorios en MSDOS

- Directorios con estructura de árbol.
- Cada entrada de directorio ocupa 32 bytes

Nombre	Ext.	Attr.	Reservado	Tiempo	Fecha	Bloque 1	Tam.
8	3	1	10	2	2	2	4

- La entrada de directorio contiene todos los atributos del archivo: nombre, extensión, atributos (Attr: del sistema, oculto, etc.), tiempo de última modificación (Tiempo), fecha de creación (Fecha).
- La entrada de directorio sólo proporciona el puntero al primer bloque de datos. La FAT proporciona el resto de punteros.
- Proporciona el tamaño del archivo (Tamaño).
- Los directorios son archivos con un número arbitrario de entradas excepto el directorio raíz que está en la cabecera del disco y tiene un tamaño máximo prefijado.

Implementación del sistema de ficheros
implementación de directorios

Directorios en UNIX

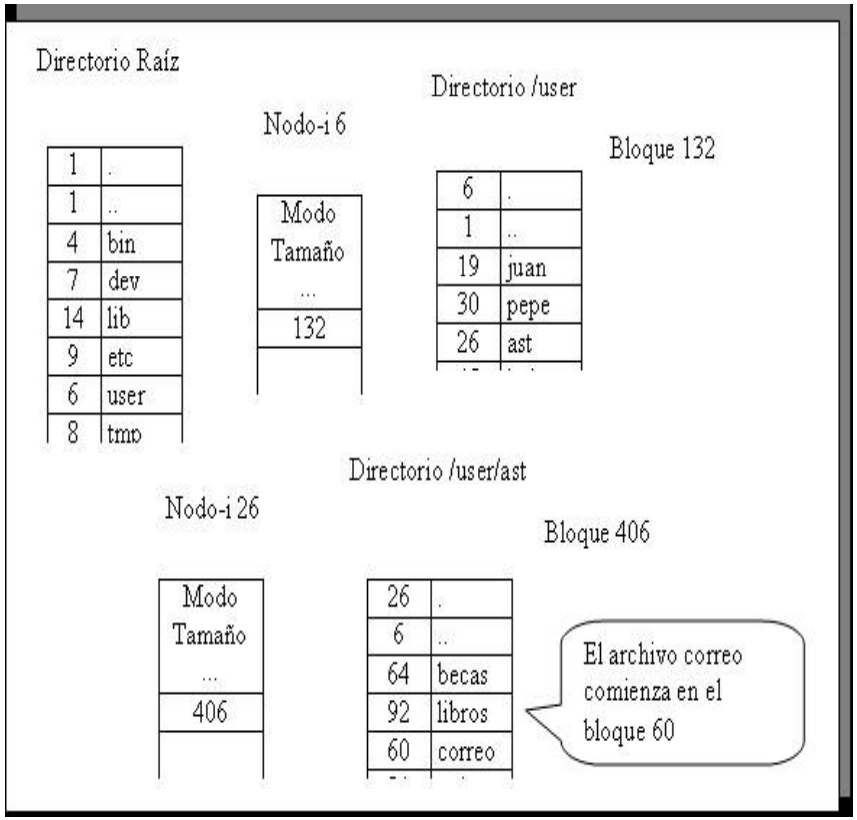
- Directorios con estructura de grafo acíclico.
- La entrada es extremadamente sencilla: no contiene todos los atributos del archivo ni ningún puntero a bloques de datos: sólo una referencia al nodo-i (Num. nodo-i).
- El nombre del archivo incluye la extensión (total 14 bytes por defecto, pero con posibilidad de especificar nombres más largos en la configuración del sistema).
- Los directorios son archivos.

Núm. nodo-i	Nombre de archivo
2	14

- Existen dos entradas que todos los directorios tienen:
  - "." indica el número del nodo-i del directorio actual.
  - ".." indica el número del nodo-i del directorio padre.

Implementación del sistema de ficheros
implementación de directorios

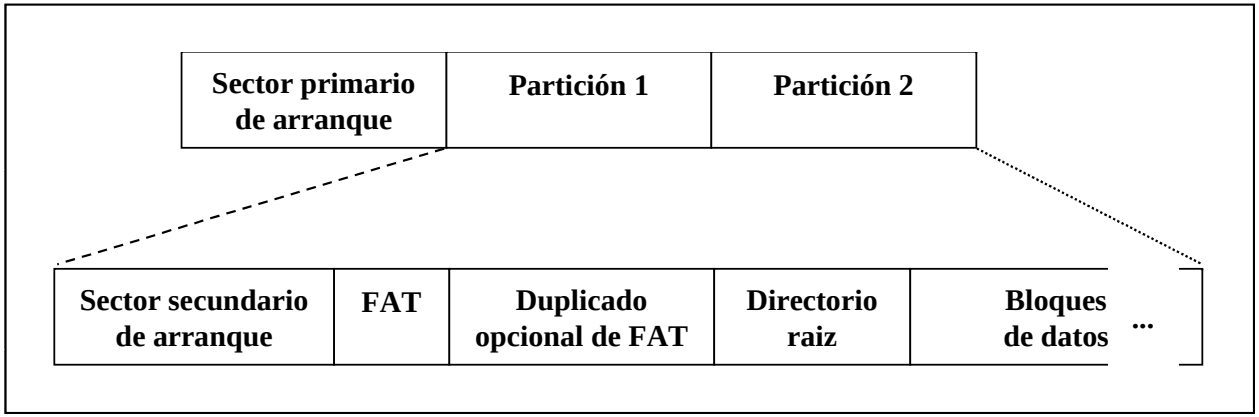
Directorios en UNIX



Implementación del sistema de ficheros
ejemplos estructura discos

Estructura del disco en MS-DOS

- Sector de arranque primario. Contiene información crítica sobre la tabla de particiones y el código para el arranque del sistema.
- Tabla de particiones: Indica el principio, el tamaño y el tipo de cada partición. Se manipula con comandos como fdisk



Implementación del sistema de ficheros
ejemplos estructura discos

Estructura del disco en MS-DOS

- El sector de arranque de MS-DOS contiene información sobre el sistema de archivos como:
  - Número de bytes por sector.
  - Número de sectores por bloque.
  - Tamaño del directorio raíz.
  - Número de tablas FAT.
  - Tamaño del dispositivo.
  - Código para iniciar el SO:
    - Búsqueda del directorio raíz.
    - Carga de los archivos “io.sys” y “msdos.sys”.
    - Transfiere control al “io.sys”.

Implementación del sistema de ficheros
ejemplos estructura discos

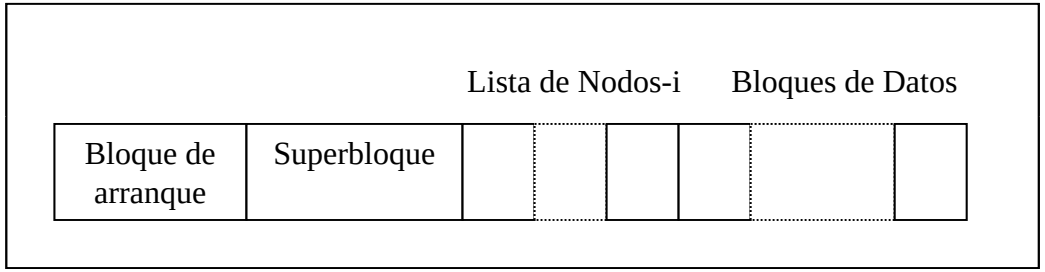
Estructura del disco en MS-DOS

- Organización de una partición MS-DOS
  - La información previa a los bloques de datos (sector de arranque, FAT y directorio raíz) se organiza en sectores.
  - Los bloques de datos donde se almacenan los archivos (incluidas las entradas de directorios que no pertenecen al Raíz), se organizan mediante una lista enlazada (FAT).
  - Los bloques libres están marcados con un código especial. Cuando se asigna espacio a un archivo, MS-DOS busca en la FAT las entradas de bloque libre necesarias.

Implementación del sistema de ficheros
ejemplos estructura discos

Estructura del disco en UNIX

- Bloque de arranque. Ocupa el primer sector del sistema de archivos y puede contener código de arranque.
- Superbloque: Contiene información sobre el sistema de ficheros
  - Tamaño del sistema de archivos
  - Lista de bloques libres
  - Tamaño de la lista de nodos-i, etc
- Lista de inodos (nodos-i). Hay una entrada por cada archivo. El administrador especifica el tamaño de la lista de nodos índice.
- Bloques de datos. Ocupa el resto del sistema de archivos. Un bloque de datos pertenece a un único fichero



Implementación del sistema de ficheros

ejemplos estructura discos

Estructura del disco en UNIX

Nodo índice

- Estructura de control que contiene la información de un archivo.
- Pueden asociarse varios nombres de archivo a un mismo nodo-i.
- Cada archivo es controlado por un único nodo-i.