



Facultad de Ciencias Sociales y de la Comunicación

Diplomatura en Gestión y Administración Pública

Asignatura de:

Redes de datos

Temas III:

Componentes de un Sistema Operativo

(Transparencias de clase)

DEPARTAMENTO DE LENGUAJES Y SISTEMAS INFORMÁTICOS

Curso 2009/2010

Profesor: Manuel Fernández Barcell

e-mail: manuel.barcell@uca.es

www.mfbarcell.es

Índice

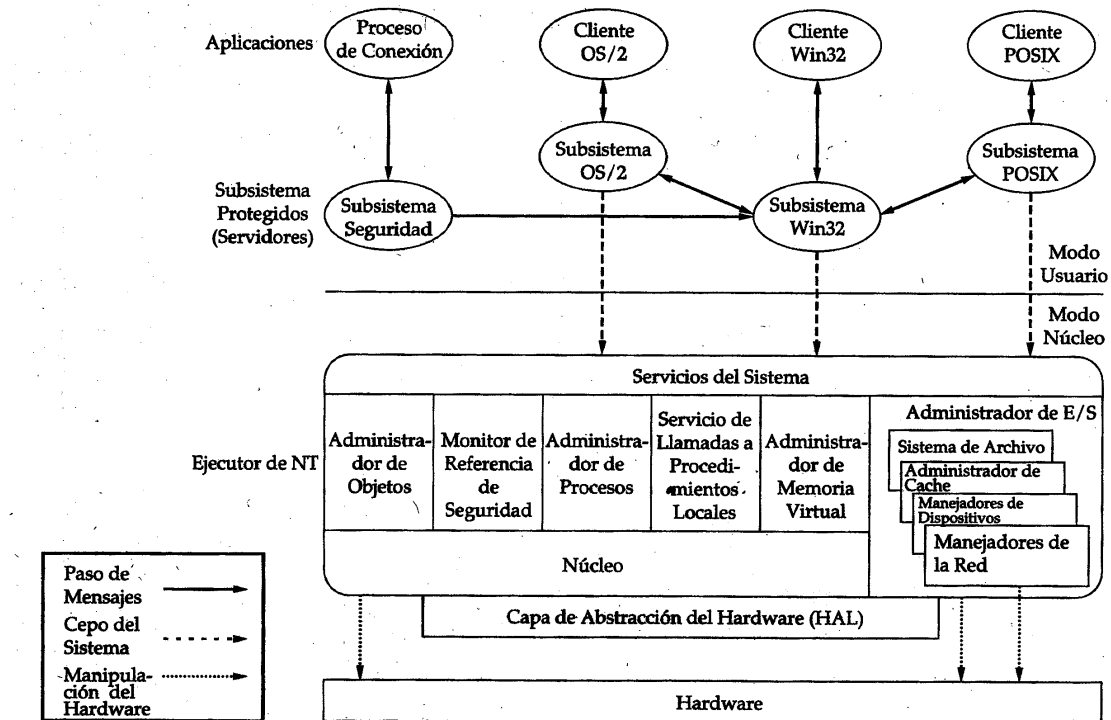
1GESTOR DE PROCESOS.....	4
1.1¿QUÉ ES UN PROCESO?	4
1.2ESTADOS DE UN PROCESO Y TRANSICIONES ENTRE ESTADOS.....	5
1.3MODELO DE 5 ESTADOS.....	5
1.4HEBRAS DE EJECUCIÓN (THREADS).....	6
1.5TIPOS DE PLANIFICACIÓN.....	7
1.5.1Planificación apropiativa (<i>preemptive</i>).....	7
1.5.2Planificación no apropiativa (<i>nonpreemptive</i>).....	7
1.6ALGORITMOS DE PLANIFICACIÓN.....	7
2GESTOR DE MEMORIA.....	9
2.1LA MEMORIA COMO RECURSO.....	9
2.2JERARQUÍA DE LAS MEMORIAS.....	9
2.2.1Tecnología de las memorias:.....	9
2.3MEMORIA CACHÉ.....	10
2.4FUNCIONES DEL ADMINISTRADOR DE MEMORIA.....	11
2.5CLASIFICACIÓN DE LAS POLÍTICAS DE ADMO. DE MEMORIA.....	11
2.6POLÍTICAS DE ASIG. DE MEMORIA COMPLETA Y CONTIGUA.....	11
2.6.1Asignación contigua para un solo proceso (<i>Monoprogramación</i>).....	11
2.6.2Asignación contigua de particiones fijas.....	11
2.6.3Asignación contigua de particiones variables.....	12
2.6.4Intercambio (<i>swapping</i>).....	13
2.7ASIGNACIÓN DE OBJETO COMPLETO NO CONTIGUOS.....	13
2.7.1Paginación.....	14
2.7.2Segmentación.....	14
2.8MEMORIA VIRTUAL.....	15
2.8.1Descripción.....	15
2.8.2Manejo de un fallo de página.....	16
3EL SISTEMA DE GESTIÓN DE FICHEROS.....	19
3.1CONCEPTO.....	19
3.1.1Justificación.....	19
3.1.2Funciones del sistema de gestión de ficheros.....	19
3.2CONCEPTOS PREVIOS.....	20
3.2.1Conceptos sobre ficheros.....	20
3.2.2La estructura de disco.....	20
3.2.3Formateo de un disco.....	21
3.3SISTEMAS DE FICHEROS.....	25
3.3.1Elementos de información de un directorio de ficheros.....	25
3.3.2Aspectos básicos de los ficheros.....	27
3.3.3Métodos de accesos.....	28
3.3.4Ejemplos de sistemas de ficheros:.....	28

3.3.5	Sistema de ficheros del UNIX.....	28
3.3.6	Estructura jerárquica del SGF.....	30
4	SISTEMA DE ENTRADA SALIDA.....	30
4.1	FUNCIONES.....	30
4.2	COMPONENTES HARDWARE DEL SUBSISTEMA DE E/S.....	31
4.2.1	Bus.....	31
4.2.2	Periféricos.....	33
4.2.3	Controladores de dispositivos.....	33
4.3	TÉCNICAS DE COMUNICACIÓN CON EL PROCESADOR.....	34
4.3.1	Operaciones de E/S guiada por interrupciones.....	34
4.3.2	Operaciones de E/S con acceso directo a memoria (DMA).....	35
4.4	PRINCIPIOS DEL SOFTWARE DE E/S.....	36
4.4.1	Objetivos del software de E/S.....	36
4.4.2	Estratos del software de e/s.....	37
	Manejadores o drivers de dispositivos.....	37
	Software de E/S independiente del dispositivo.....	37
4.5	RAID.....	37
4.6	STORAGE AREA NETWORKS (SAN).....	39
4.7	NAS (DEL INGLÉS NETWORK ATTACHED STORAGE)	39
5	CUESTIONES.....	41

En este tema veremos los componentes principales de un sistema operativo. El motivo de ver la estructura de un sistema operativo no es otro que introducir ciertos conceptos "cotidianos" en el uso de cualquier sistema operativo. Hay conceptos que son importantes que se les presten atención. En el tema de procesos, al concepto de proceso y de las hebras (threads), y la diferencia entre los dos. La planificación, su necesidad. Los distintos modos de planificación (apropiada y no apropiada). La jerarquía de memoria, la memoria, la memoria virtual. Los sistemas de ficheros y los distintos niveles de formateo. DMA, drivers y controladoras. RAID.

Introducción

En este tema veremos algunos de los componentes más importantes y frecuentes de un sistema operativo y su estructura de capas. En la figura vemos el esquema de componentes del Windows NT.



1 Gestor de procesos

1.1 ¿Qué es un proceso?

En un sistema multiprogramado, el sistema operativo debe de intercalar la ejecución de un número de procesos para maximizar el uso del procesador con un razonable tiempo de respuesta. Con la multiprogramación, se mantienen en memoria varios trabajos o programas de usuarios. Cada trabajo se encuentra alternativamente utilizando el procesador central y esperando a que realice una E/S. El procesador se mantiene ocupado ejecutando un trabajo mientras los otros esperan. La clave de la multiprogramación es la planificación. La **gestión de los**

procesos es la principal tarea de un Sistema Operativo multiprogramado.

Concepto de proceso

Lo primero que vamos a ver es el concepto de proceso. Hay varias definiciones de **proceso** como:

- La entidad que puede ser asignada y ejecutada en un procesador.
- Un programa en ejecución.
- La unidad "despachable".
- Aquella entidad a la que se le asigna un procesador.
- La entidad que se manifiesta en el sistema operativo por la existencia de un bloque de control de procesos.

Informalmente un proceso secuencial o tarea es: "la actividad resultante de la ejecución de un programa con sus datos, por un procesador secuencial". La definición más frecuentemente aceptada es: Un programa en ejecución.

Comparación entre programa y proceso

Para comprender mejor la diferencia entre programa y proceso comentaremos algunas características de ambos.

- Un proceso es la unidad de trabajo de un sistema.
- Un programa es una entidad pasiva.
- Un proceso es una entidad activa.
- El proceso es un concepto dinámico que se refiere a un programa en ejecución que sufre frecuentes cambios de estado y atributos.

Un proceso esta formado por los siguientes tres elementos:

- Un programa ejecutable.
 - Conjunto de instrucciones ejecutables
- Los datos asociados necesitados por el programa:
 - Variables.
 - Espacio de trabajo.
 - *Buffers* etc...
- El contexto de ejecución del programa.
 - Incluye toda la información que el sistema operativo necesita para manejar el proceso y la que el procesador necesita para ejecutar el proceso.

1.2 Estados de un proceso y transiciones entre estados

Para entender cual es el control que un sistema operativo debe llevar de los procesos, vamos a describir el comportamiento que los procesos pueden tener. Para ello necesitamos considerar el concepto de estado de un proceso. Durante el tiempo de vida de un proceso, la situación en que se encuentra cambia un cierto número de veces. La situación en cada instante de tiempo se denomina **estado**.

Un proceso está caracterizado por: un código ejecutable, unos datos, atributos y un *estado dinámico*.

Un estado de un proceso se caracteriza por un conjunto de información que el sistema operativo tiene de él en un momento dado. El sistema operativo debe tener información referente a cada proceso tal como:

- Su ubicación en memoria principal.
- Tamaño que ocupa.
- Estado actual del proceso.

El sistema operativo contempla la ejecución de un proceso como una progresión a través de un *conjunto de estados*.

1.3 Modelo de 5 estados

Vamos a ver un modelo con los siguientes estados posibles:

- **Estado de ejecución** (*running*): El proceso tiene el dominio de la CPU en ese momento y se está ejecutando.

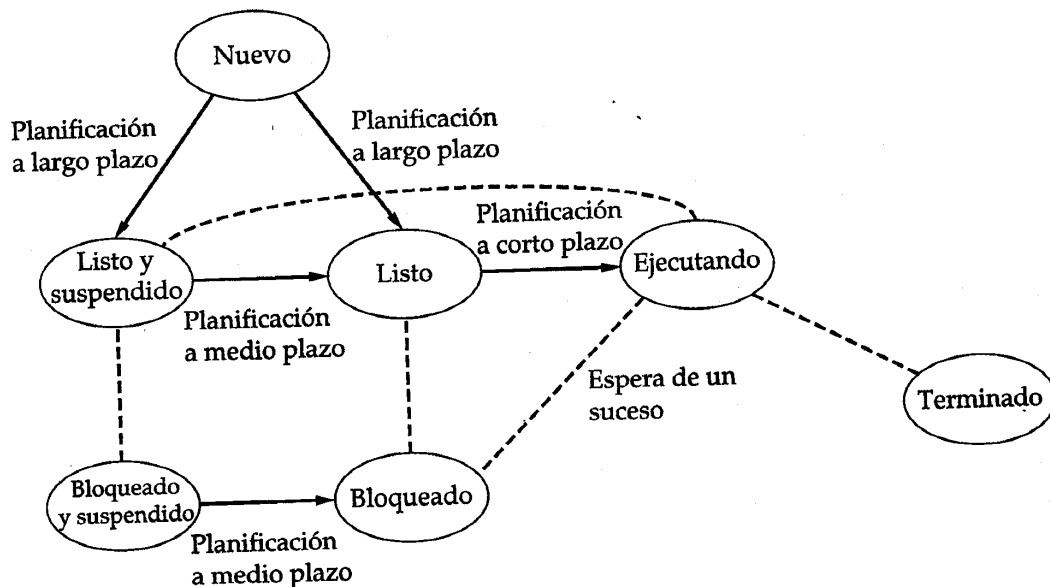
- **Estado de listo** (*ready*): El proceso está preparado y en disposición de usar la CPU, si hubiese una disponible. Está en espera de que le toque el turno.
- **Estado de bloqueado** (*blocked*): El proceso está a la espera de que ocurra algún evento (completar una operación de e/s).
- **Estado de nuevo** (*new*): El proceso acaba de ser creado y no ha sido admitido todavía en el grupo de los procesos ejecutables.
- **Estado de terminado** (*exit*): El proceso ha terminado, o ha sido abortado o detenido.

Los programas cuando aún no han sido remitidos al sistema, podemos considerarlos como en estado de *inactivo*.

Para llevar el control de los procesos, en este modelo se crean dos listas: una lista de **procesos listos** y otra de **procesos bloqueados**.

- La lista de procesos **listos** esta ordenada por orden de prioridad de ejecución. La de procesos **bloqueados**, no.
- Cuando un trabajo llega al sistema entra en la lista de trabajos **listos**, a la espera de CPU.

Un proceso cambia entre estos estados a lo largo de su vida.



1.4 Hebras de ejecución (*threads*)

En los procesos que hemos visto hasta ahora podemos distinguir dos características:

- Una unidad propietaria de recursos. Un proceso tiene su contador de programa, su conjunto de registros y su espacio de direcciones.
- Una unidad de ejecución o despacho (las instrucciones de las rutinas a ejecutar).

Esta segunda entidad es la que se conoce como *thread*. Los sistemas operativos de reciente desarrollo consideran que estas dos características deben tratarse por separado para mejorar los rendimientos. Al trabajar con una entidad más pequeña, es más fácil de manejar. La unidad que asigna el sistema al procesador no es el "proceso completo" (recursos mas unidad de ejecución) sino la hilera. Los recursos son compartidos por las distintas hileras que tenga ese proceso.

La hilera tiene utilidad tanto en sistemas monoprocesadores como en multiprocesadores. En un sistema multiprocesador, el Sistema Operativo podrá repartir las distintas rutinas (unidades de ejecución) de un proceso entre los distintos procesadores.

Múltiples hebras en un solo proceso

En los procesos tradicionales (procesos pesados) existe un sólo espacio de direcciones y un solo hilo de ejecución.

Hay situaciones en que es deseable tener más de un hilo de ejecución, que compartan un único espacio de direcciones, pero que se puedan ejecutar de modo más o menos simultáneo.

- A estos flujos de control se les denomina hebras o hilera de ejecución.

1.5 Tipos de planificación

1.5.1 Planificación apropiativa (*preemptive*)

En la planificación de procesos apropiativa, el proceso que se está ejecutando, puede ser interrumpido en cualquier momento y movido a la cola de listos, antes de su completa ejecución. Es decir, la CPU, una vez asignada a un proceso, puede ser retirada por el sistema operativo. Es adecuada para sistemas de tiempo compartido, con tiempos de respuestas críticos.

Las distintas clases de planificación apropiativa que existen se diferencian en el momento en que pueden retirar el dominio de la CPU al proceso que se esté ejecutando en ese momento. Las posibilidades de momentos se pueden retirar el dominio de la CPU a un proceso son:

- Cada vez que llega un nuevo proceso.
- Cuando se produce una interrupción.
- Cada cierto período de tiempo q (*quantum*).

Este tipo de planificación es más costosa por los tiempos empleados en los cambios de contextos (intercambio de un proceso a otro) que son más frecuentes. Pero impide que un proceso monopolice el procesador por un período de tiempo muy prolongado.

1.5.2 Planificación no apropiativa (*nonpreemptive*)

La planificación **no** apropiativa consiste que una vez asignada la CPU a un proceso, no se le puede retirar hasta que la libera, ya sea porque termina la ejecución o porque cambia a un estado de espera (se autobloquea).

No es adecuada para los sistemas de tiempo real ni de tiempo compartido. Microsoft Windows 95, 98 tiene un esquema de planificación **no apropiativo**.

1.6 Algoritmos de planificación

Es necesario un algoritmo que decida de entre los procesos que son candidatos al dominio de la CPU (cola de procesos listos), a cual se le asigna.

Son posibles encontrar muchos algoritmos dependiendo de los parámetros que se evalúen - memoria, tiempo de CPU, de E/S, carga del sistema, prioridades externas ...-

Vamos a nombrar algunos algoritmos basados en consideraciones de tiempo.

- FIFO (*First-in/First-out*)

Consiste en servir primero al primero que llega, es decir, se les asigna la CPU conforme con el orden de llegada.

- LIFO (*Last-in/First-out*).
- SJF (*Shortest-job-first*) o SJN (*Shortest-job-next*).

Se le asigna la CPU, al proceso que tenga un tiempo estimado de ejecución más corto hasta su terminación.

- Prioridad
- SRT (*Shortest-Remaining-time*)

El proceso de tiempo estimado de ejecución menor para llegar a su terminación, es el siguiente en ser ejecutado, incluyendo las nuevas llegadas.

1Gestor de procesos

- RR (*Round-robin*)

A cada proceso se le asigna un *quántum* de tiempo. Si no termina en este período, el control de la CPU pasa al siguiente proceso de la cola, y el que se estaba ejecutando pasa al último lugar.

- ML (*Multilevel*).

Se crean distintas colas de procesos en estado de preparados. Los trabajos permanecen asignados a una cola en virtud de alguna propiedad del mismo.

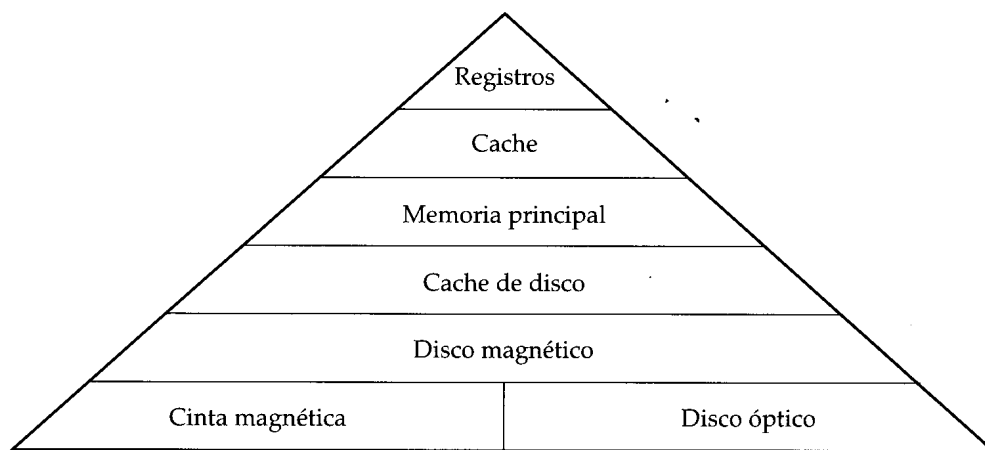
- MLF (*Multilevel Feedback*).

2 Gestor de memoria

2.1 La memoria como recurso

La memoria es un recurso importante que debemos administrar con cuidado aunque en la actualidad sea un recurso abundante. El rendimiento de un Sistema va a depender de la cantidad de memoria que tenga disponible y de como optimiza su uso. En un sistema multiprogramado es vital una gestión eficiente de la memoria para conseguir el mayor grado de multiprogramación que permita la máxima utilización del procesador.

La memoria tiene un carácter central en el funcionamiento de un Sistema informático moderno. La CPU y el Sistema de E/S interactúan con la memoria central.



(b) Jerarquía moderna de memoria

CPU <-----> memoria <-----> S. E/S.

La parte de un Sistema Operativo que se encarga de la gestión del recurso del Sistema "memoria" es el *administrador de la memoria*.

2.2 Jerarquía de las memorias

2.2.1 Tecnología de las memorias:

Las características técnicas de las memorias responden a algunas de las preguntas: ¿Qué cantidad de memoria: capacidad? ¿Qué velocidad: tiempo de acceso? ¿Qué costo?

Hay una regla general con relación a las memorias: Cuanto más pequeño, más rápido, pero más caro. Es decir, a tiempos de acceso menores, mayor costo por bit. Mayor capacidad, mayor tiempo de acceso, menor costo por bit.

Los almacenamientos de un ordenador se organizan en una jerarquía:

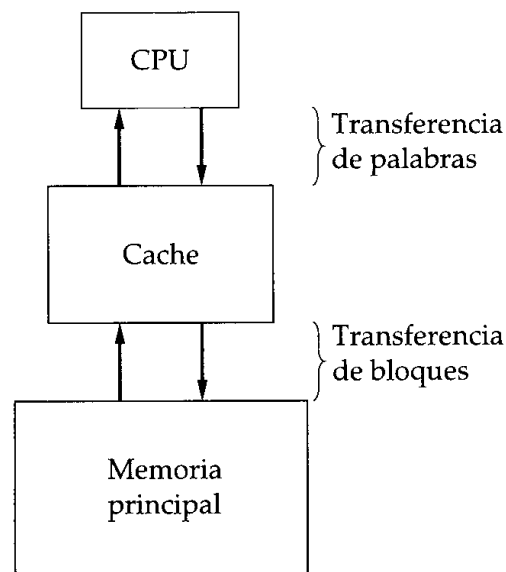
- Los más rápidos, en la parte de arriba.
- Los más lentos, menos costosos y más densos, en la parte de abajo.
 1. Registros.
 2. Caches internas (nivel 1, dentro del procesador).
 3. Caches externas (nivel 2).
 4. Memoria principal
 5. Caché de disco.
 6. Disco magnético.
 7. Cinta- disco óptico.

2.3 Memoria caché

Son memorias rápidas pero de pequeño tamaño. Se construyen con componentes de acceso muy rápido, pero caros.

Las memorias caché contienen una copia de una porción de la memoria principal. El funcionamiento de las memorias caches es el siguiente: cuando el procesador quiere acceder a una palabra de memoria principal, comprueba primero si está situada en la memoria caché. Se consideran dos situaciones:

- Si la palabra buscada se encuentra en la memoria caché: la suministra.
- Si no es así, traslada el bloque que contiene la palabra buscada, de memoria principal a la caché.



Tiempo medio de acceso de la memoria caché se calcula mediante la formula:

$$T_s = H \times T_1 + (1-H) \times (T_1 + T_2) \Rightarrow T_s = T_1 + (1-H) \times T_2$$

T_s = tiempo medio de acceso

T_1 = tiempo de acceso a memoria de 1º nivel (caché).

T_2 = tiempo de acceso a memoria de 2º nivel (memoria principal)

H = porcentaje de aciertos.

Los sistemas que incorporan memorias caches funcionan porque el acceso a memoria principal no se realiza de un modo aleatorio sino basado en dos principios:

- **Localidad temporal** (localidad en el tiempo) si un elemento es referenciado, volverá a ser referenciado pronto.
 - Si hace poco tiempo que llevó un libro de la biblioteca a su mesa del despacho para examinarlo, probablemente pronto necesitará examinarlo de nuevo.
- **Localidad espacial** (localidad en el espacio) Si un elemento es referenciado, los elementos cuyas direcciones están próximas tenderán a ser referido pronto.

Rendimiento

El rendimiento de una memoria cache se calcula por la formula:

$$C_s = (C_1 \times S_1 + C_2 \times S_2) / (S_1 + S_2)$$

C_s = Coste medio por bit de los dos niveles combinados de memoria

C_1 = Coste medio por bit de M1

C_2 = Coste medio por bit de M2

S_1 = Tamaño de M1 (memoria de primer nivel)

S_2 = Tamaño de M2 (memoria de 2º nivel)

2.4 Funciones del administrador de memoria

- Llevar el control de la memoria que está libre y la que está ocupada.
- Determinar a que proceso se le asigna memoria y que cantidad.
- Asignar memoria a los procesos cuando la necesiten y retirársela cuando terminen.
- Determinar donde situar un proceso dentro del almacenamiento principal.
- Administrar el intercambio entre memoria central y el disco cuando la memoria central no baste para contener todos los procesos.

Requerimientos del administrador de memoria:

- Realización de la reubicación de los programas.
- Mecanismos de protección entre procesos.
- Compartición de la misma zona de memoria.
- Organización física (jerarquía) y lógica.

La selección de un esquema de gestión de memoria depende de varios factores, especialmente del *hardware* del Sistema.

2.5 Clasificación de las políticas de admo. de Memoria

1.- Asignación de Memoria completa y contigua

La memoria se asigna de tal manera que todo el espacio direccionable de un objeto lógico esta situado en área contigua de memoria física.

1.- Monoprogramación.

2.- Multiprogramación con particiones múltiples.

2.1.- Asignación contigua de particiones fijas (MFT).

2.2.- Asignación contigua de particiones múltiples (MVT).

2.- Asignación de Memoria completa pero no contigua

La memoria se asigna de tal manera, que las distintas partes de un único objeto lógico pueden estar situados en áreas no contiguas de memoria física, pero el espacio direccionable del objeto lógico se carga completo en memoria.

1.- Paginación.

2.- Segmentación.

3.- Asignación de Memoria parcial y no contigua

1.- Memoria virtual.

2.6 Políticas de asig. de Memoria completa y contigua

2.6.1 Asignación contigua para un solo proceso (Mono programación)

Descripción:

La memoria física (espacio físico de direcciones) se divide en dos secciones básicas:

- Sección para el proceso usuario.
- Sección para el Sistema Operativo.

En la sección del usuario sólo hay un proceso residente, al que se le asigna toda la memoria física disponible. No se utiliza ya.

2.6.2 Asignación contigua de particiones fijas

Descripción:

Se divide el espacio físico de direcciones en un número determinado de particiones de tamaño fijo. El número y tamaño de las particiones se determina en tiempo de generación del Sistema.

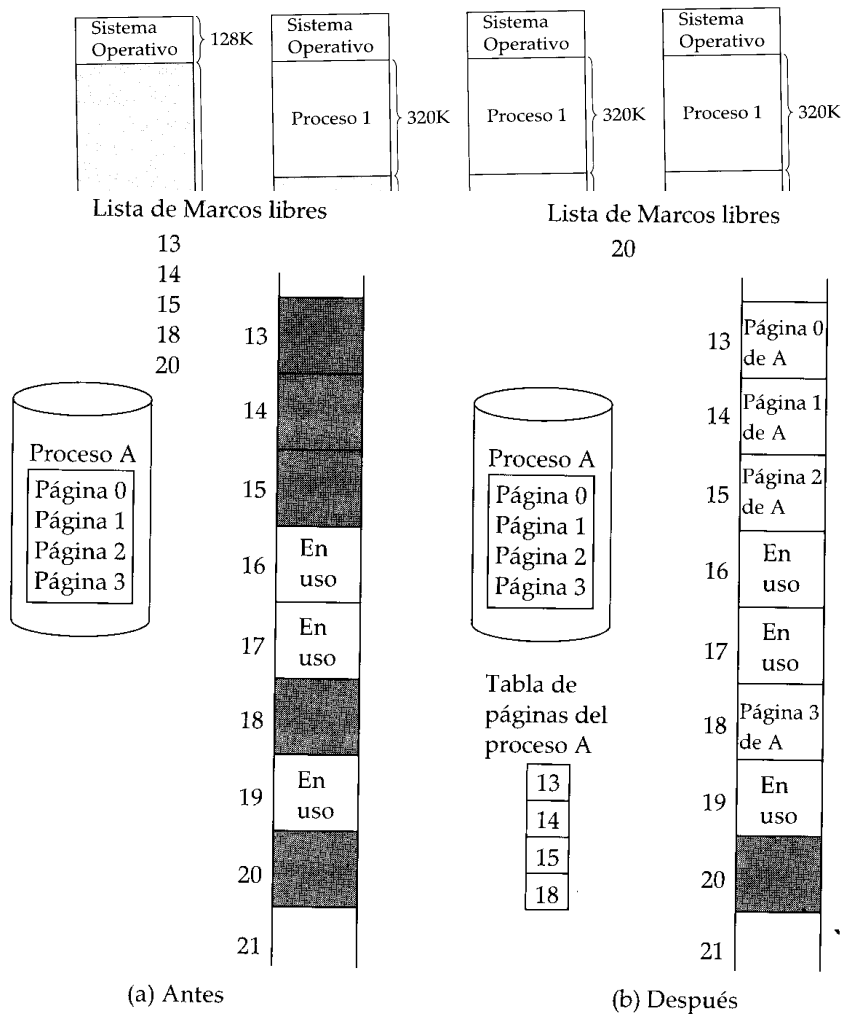
Sistema operativo 128K
64K
192K
256K
384K

2.6.3 Asignación contigua de particiones variables

Las particiones usadas son variables en tamaño y número. Los programas ocupan el espacio de memoria física que realmente necesitan.

Los huecos se producen cuando empiezan a terminar los procesos que inicialmente se cargan en memoria.

Según se gestionen los huecos tenemos la asignación dinámica sin compactación y con compactación.



2.6.4 Intercambio (*swapping*)

Se entiende por *SWAPPING* el intercambio de la memoria imagen de los procesos de memoria principal al almacenamiento secundario (disco) y viceversa.

Se emplea cuando hay sobrecarga en los Sistemas, para disminuir la carga de trabajo o mientras un proceso está a la espera de un suceso para dejar memoria libre. El tiempo necesario para hacer el intercambio es relativamente alto.

Hay Sistemas que reservan un espacio en disco exclusivo para procesos intercambiados (área de *swapping*). Cuando el intercambio se realiza por la llegada de procesos de mayor prioridad se denomina *roll-in roll-out*.

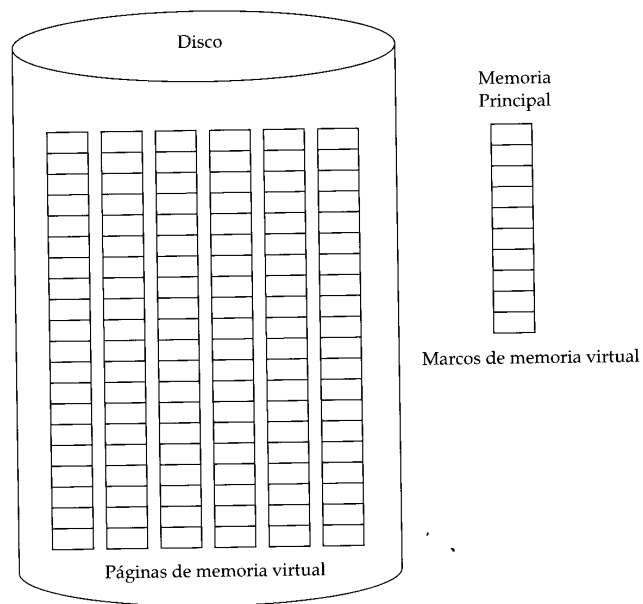
2.7 Asignación de objeto completo no contiguos

El espacio de direcciones lógicas se carga completo en memoria física, pero no todo el espacio tiene que estar en posiciones contiguas de memoria física.

El espacio de direcciones lógicas se dividirá en "fracciones" menores para facilitar su colocación en los huecos existentes en la memoria.

Se necesitarán mecanismos para conocer donde están situadas cada fracción del espacio de direcciones lógicas dentro de la memoria física posibilitando la traducción de las direcciones lógicas en direcciones físicas.

Hay dos aproximaciones a esta idea, fracciones todas del mismo tamaño (páginas) o fracciones de distinto tamaño (segmentos).



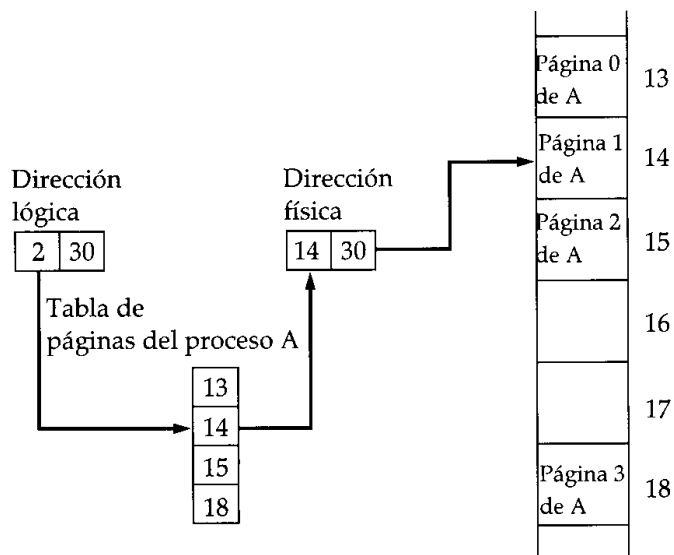
2.7.1 Paginación

Descripción

El espacio de direcciones lógicas del programa se divide en fracciones de igual tamaño que denominamos *páginas*.

Igualmente la memoria física, se divide en un número de fracciones o partes de igual tamaño que vamos a denominar *celdas* o marcos de página (*frames*). Las páginas y las celdas tienen el mismo tamaño.

2.7.2 Segmentación

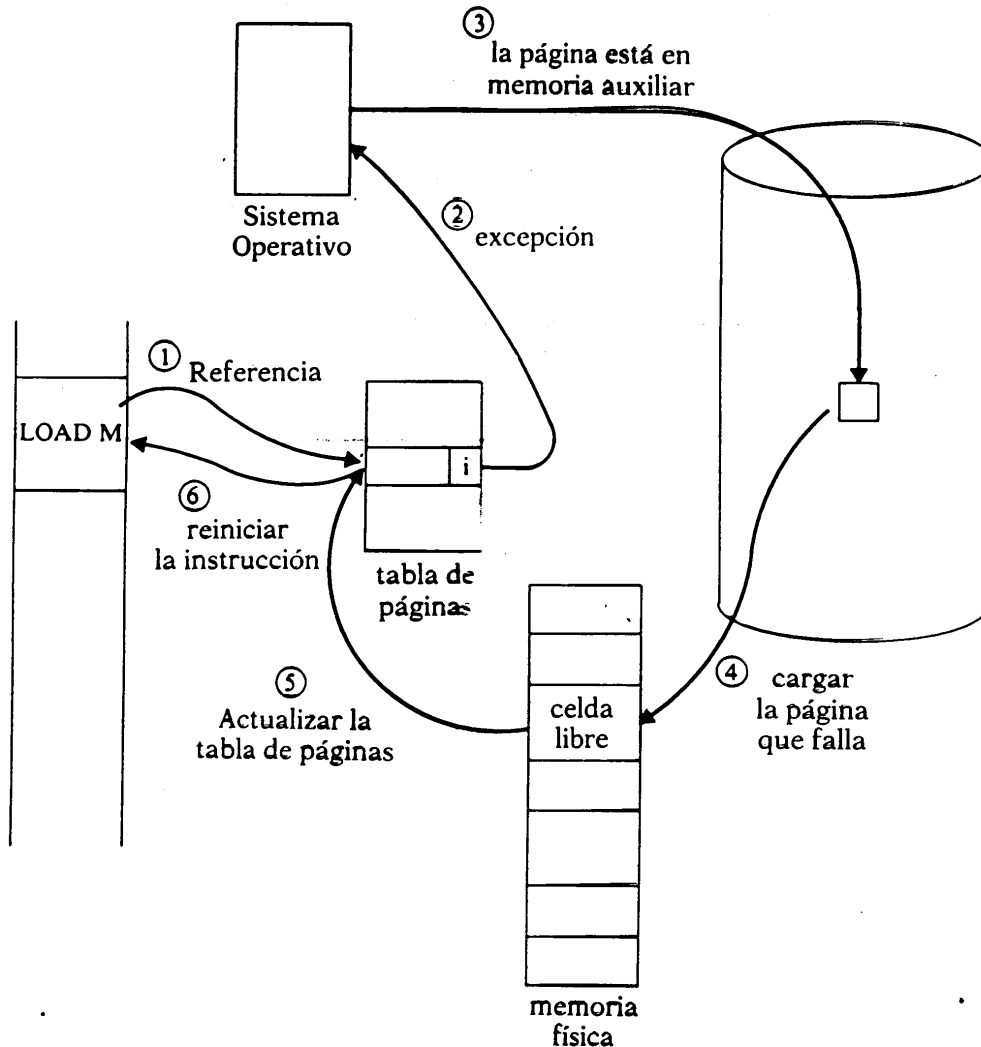


Principios de operación

Consiste en dividir el espacio de direcciones lógicas del programa en unos conjuntos de procedimientos u objetos (segmentos) que tienen una entidad por sí mismo.

Un **segmento** es una entidad definida por el usuario que puede ser tratada como una unidad lógica e independiente.

El espacio de direcciones lógicas es ahora un conjunto de segmentos. Los objetos (segmentos) en que se



Pasos en el tratamiento de un fallo de página

divide el programa son de distinto tamaño. La división *no es transparente al programador*.

La división de un programa en una serie de subrutinas independientes la realiza el programador.

2.8 Memoria virtual

2.8.1 Descripción

Definición: conjunto de técnicas que nos permite ejecutar un proceso en el que su espacio virtual de direcciones no se encuentra en memoria física en su totalidad.

Características:

- Sólo una porción del espacio de direcciones lógicas de un proceso residente está cargada en memoria física.
- La suma de los espacios direccionables de los procesos activos, puede exceder la capacidad de la memoria física.

2Gestor de memoria

- Se lleva a cabo manteniendo en el almacenamiento secundario una imagen del espacio direccionable completo del proceso y trayendo las secciones que son necesarias a memoria principal.
- Se consideran dos niveles de almacenamiento el 11 en memoria física y el 21 en memoria auxiliar.

Justificación

Un examen de los programas reales muestran que no se precisa tener cargado en memoria principal la imagen del programa ejecutable completo:

- Secciones de códigos de error poco usuales.
- Operaciones del programa que se utilizan raramente.
- A las matrices, listas y tablas... se les asigna tamaños máximos que en la mayoría de los casos pocas veces realmente se necesitan. (Ejemplo: compiladores).
- Ejecución de programas con caminos alternativos.

Ventajas:

- El tamaño de un proceso no queda limitado por la cantidad de memoria física.
 - Se pueden ejecutar procesos que requieran una cantidad de memoria mayor que la disponible realmente por el ordenador.
 - Se tiene la "ilusión" de disponer de una memoria mucho mayor de la que realmente se dispone.
- Aumenta el grado de multiprogramación.
 - Al usar cada usuario menos memoria física, más usuarios podrían usar el Sistema.
- Menor tiempo de E/S:
 - Se necesita menos tiempo de E/S para cargar o intercambiar cada proceso a memoria.
- La gestión de la memoria virtual es generalmente transparente para el programador.
- El Sistema Operativo puede acelerar procesos de mayor prioridad asignándole más memoria real y retirársela a otros de menor prioridad.
- Elimina la fragmentación externa y minimiza la interna.
 - Fragmentación externa: espacio de la memoria física desperdiciado o no ocupado por ningún proceso
 - Fragmentación interna: desperdicio o no utilización de parte de la memoria asignada a un proceso.

Funcionamiento

El espacio direccionable del proceso se divide en páginas de igual tamaño que los marcos de memoria física. Las páginas del proceso se reparten entre los marcos de páginas que haya libres. No todas las páginas del proceso se cargan en memoria.

En Windows el fichero imagen de intercambio de memoria virtual es el *pagefile.sys*. Para cambiar el tamaño de este fichero se realiza en el panel de control -> sistema -> rendimiento. Recomendación 2,5 veces el tamaño de la memoria física.

El funcionamiento es similar al descrito en el apartado de paginación. Se requiere un mecanismo de traducción de las direcciones lógicas a las direcciones físicas.

Mecanismo de traducción

Es necesaria una tabla de página que indique donde se encuentran cada página del proceso. Vea la figura.

2.8.2 Manejo de un fallo de página

La secuencia de eventos es la siguiente:

11.- El *hardware* "atrapa" al núcleo del S.O., salvando el contador de programas en la pila.

- En la mayoría de las máquinas, alguna información sobre el estado de ejecución de la instrucción actual se salva en registros especiales de la CPU.

2.- Una rutina en código máquina es arrancada para salvar el contenido de los registros generales y otra información *volátil*, para evitar su pérdida.

3.- El S.O. descubre que ha ocurrido un fallo de página, y trata de descubrir qué página virtual es la que se necesita.

4.- Una vez que la dirección virtual que causó el fallo es conocida, el S.O. comprueba si esa dirección es válida y las protecciones de la página permiten realizar el acceso que se desea.

- Si no es así, el S.O. manda una *señal* al proceso indicándoselo.
- Si la dirección es válida y no hay problemas con las protecciones, el Sistema intenta obtener un marco de página de la lista de marcos de páginas libres.
- Si no hay marcos de páginas libres se ejecuta el algoritmo de reemplazo de página para seleccionar una "víctima".

5.- Si la página seleccionada tiene activado el bit página modificada (*dirty bit*), un cambio de contexto tiene lugar, suspendiendo el proceso de cambio de página, poniendo otro proceso en ejecución hasta que se ha completado la transferencia a disco.

6.- Tan pronto como el marco de página está limpio, el Sistema Operativo averigua la dirección del disco donde se encuentra la página que se necesita, y ordena una operación de disco para traerla.

7.- Cuando la interrupción de disco indica que la página ha llegado, las tablas de página son actualizadas para reflejar su posición, y el marco de página es marcado como de situación normal.

8.- La instrucción causante del fallo de página es retomada en el estado que estaba cuando ocurrió y el contador de instrucciones es restablecido para que apunte a esa instrucción.

9.- El proceso es puesto en la cola de listo, y el Sistema operativo retorna a la rutina en lenguaje máquina que lo llamó.

10.- Esta rutina restaura los registros y toda la información volátil, y retorna al espacio de usuario para continuar la ejecución, como si el fallo de página no hubiese ocurrido.

Consideraciones	Paginación	Segmentación
▪ ¿Necesita el programador ser consciente de que esta técnica está siendo usada?	No	Sí
▪ ¿Cuántos espacios de direccionamiento lineal hay?	1	Muchos
Puede el espacio total de direcciones lógicas superar el tamaño de la memoria física?	Sí	Sí
¿ ▪ Pueden adaptarse fácilmente las tablas cuyo tamaño fluctúa?	No	Sí
▪ ¿Es fácil el compartir procedimientos entre distintos usuarios?	No	Sí
¿ ▪ Pueden ser los procedimientos y los datos fácilmente separados y protegidos?	No	Sí
▪ ¿Porqué se inventan estas técnicas?	Para obtener un espacio de direcciones lógicas virtuales grande sin necesidad de comprar más memoria física.	Para permitir que los procesos puedan ser separados en espacios lógicos independientes y añadirles la protección y compartición

Cuestiones

Comparación entre los distintos Sistemas de Gestión de Memoria			
Esquemas	Problemas solventados	Problemas creados	Cambios en el software
Monoprogramación contigua		* Trabajo limitado al tamaño de la memoria física * CPU frecuentemente inactiva	Ninguno
Particiones fijas	Tiempo de CPU inactiva	* Tamaño del trabajo limitado al de la partición * Fragmentación interna	* Manejo de protección * Proceso de administración
Particiones dinámicas	* Fragmentación interna	* Fragmentación externa	Ninguno
Particiones dinámicas con reubicación	* Fragmentación Interna	* Sobrecarga por compactación * Tamaño del trabajo limitado al tamaño de la memoria física	Algoritmos de compactación
Paginación	* Necesidad de compactación	* Memoria necesaria para las tablas * Trabajo limitado al tamaño de la memoria física. * Fragmentación interna	* Algoritmos de manejo de las tablas de páginas.
Demanda de página	* Trabajo limitado al tamaño de la memoria física * Uso más eficiente de la memoria. * Permite un alto grado de multiprogramación y tiempo compartido	* Gran numero de tablas * Posibilidad de <i>thrashing</i> * Sobrecarga por los fallos de páginas. * Necesidad de hardware de paginación	* Algoritmos de reemplazo de páginas. * Algoritmos de búsqueda para las páginas en almacenamiento secundario
Segmentada	* Fragmentación Interna * Enlaces dinámicos * Compartición de segmentos	* Dificultad del manejo de segmentos de longitud variables en almacenamiento secundario. * Fragmentación externa.	* Esquema de direccionamiento de dos dimensiones. * Enlace dinámico
Demanda segmentada /paginada	* Carga de grandes segmentos en memoria.	* Sobrecarga por el manejo de tablas. * Memoria necesaria para las tablas de segmentos y de páginas.	* Esquema de direccionamiento de tres dimensiones

3 El Sistema de Gestión de ficheros

3.1 Concepto

Sistema de Gestión de Ficheros es la parte del Sistema Operativo encargada:

- Del control de todos los ficheros del sistema.
- De asegurar la conservación de los ficheros.
- De las funciones de accesos a los mismos.

Proporciona a los usuarios y aplicaciones servicios relacionados con el uso de los ficheros. Es el *software* responsable de permitir a los usuarios y aplicaciones:

- Poder crear, modificar, borrar y controlar el acceso a los ficheros por medios de servicios de comandos o llamadas al sistema.
- Oculta las operaciones de bajo nivel.

El sistema de ficheros, gestiona la forma de guardar los datos en los dispositivos de almacenamiento. Vamos a centrarnos en el almacenamiento en disco magnético por ser el más frecuente.

Responde a las preguntas de ¿Dónde se almacena la información, cómo se almacena la información y quién y cuándo se puede acceder a una información?

3.1.1 Justificación

Todos los Sistemas necesitan almacenar información a largo plazo y poder recuperarla posteriormente. Para el almacenamiento físico de la información a largo plazo se emplea los dispositivos de almacenamiento secundario, como discos y cintas.

La información se almacena de modo organizado en unas abstracciones o estructuras de datos denominadas **ficheros**.

3.1.2 Funciones del sistema de gestión de ficheros

SGF se encarga de la gestión del almacenamiento secundario. Entre sus funciones están:

- La gestión de los soportes físicos de la información ocultando al usuario los detalles de la organización de la ubicación física de los ficheros.
 - Saber donde y cómo se encuentran almacenados físicamente los ficheros.
- Optimizar los rendimientos y tiempos de respuestas.
- Traducción de las peticiones de acceso desde el espacio lógico al fichero físico.
 - Organizaciones y tipos de acceso de los ficheros.
 - Transferencia de la información entre el almacenamiento secundario y el principal.

SGF se encarga del control de accesos a los ficheros, es decir:

- Proporcionar medidas de seguridad contra pérdida de información accidental o maliciosa.
- Proporcionar privacidad de la información (control de acceso).
- Integridad y Protección.
- Llevar el control de los ficheros compartidos por varios usuarios.

Proporcionar soporte de e/s para una gran variedad de tipos de dispositivos de almacenamiento.

- E/S independiente del dispositivo
- Interactuar con el gestor de dispositivos.

Según el método y tipo de invocación de los servicios de SGF, los usuarios los podemos clasificar en:

- Usuarios del lenguaje de órdenes (comandos)
- Los programadores de sistemas (llamadas del sistema).

3.2 Conceptos previos

3.2.1 Conceptos sobre ficheros

Campo

- Es el elemento básico de datos. Contiene un valor elemental.
- Se caracteriza por su longitud y tipo de dato.
- Los campos pueden ser de longitud fija o variable.

Registro

- Es una colección de campos relacionados.
- Pueden ser de longitud fija o variable.

Concepto de ficheros

- Un fichero es una estructura de datos, constituida por un conjunto de datos relacionados, y con un nombre que sirve para referido. Tienen un nombre único y pueden ser creados y borrados. Los ficheros se les reconoce por un **nombre**, que sirve para localizar la información asociada a dicho nombre.
 - La información almacenada en los ficheros debe ser persistente.
 - Cada elemento de datos que forman un fichero lo denominamos registro. Un fichero es una colección de registros similares.
 - El fichero es una unidad lógica de almacenamiento, en contraposición al almacenamiento físico.

Base de datos.

- Colección de datos relacionados.
- Son usadas por distintas aplicaciones

Organización de ficheros

Un elemento fundamental en el diseño de un sistema de ficheros es la forma en que los registros están organizados o estructurados.

Dentro de la organización de ficheros distinguimos entre organización física y organización lógica.

La **organización física** hace referencia a cómo se guarda la información que conforma un fichero en el almacenamiento físico. (Contigua, en *cluster* separados...).

La **organización lógica** de los registros que determinan la forma en la cual pueden ser accedidos. Normalmente se le conoce escuetamente por organización de ficheros.

Ejemplos de organizaciones de ficheros:

- Fichero de organización secuencial.
- Fichero de organización secuencial-indexada.
- Fichero de organización directa.

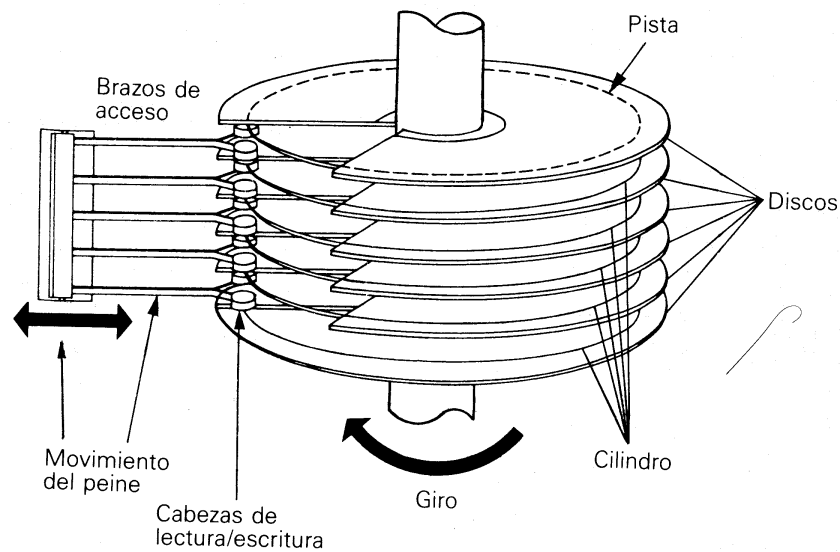
3.2.2 La estructura de disco

La geometría de un disco esta definida por:

- El número de superficies (determinada por el nº de platos y cabezas lectoras/grabadoras)
- El número de pistas por cara (cilindros)
- El número de sectores por pista
- El tamaño del sector

La capacidad del disco se obtiene con una simple operación de cálculo:

- Tamaño del sector x nº de sectores por pista x nº de pistas x nº de superficies

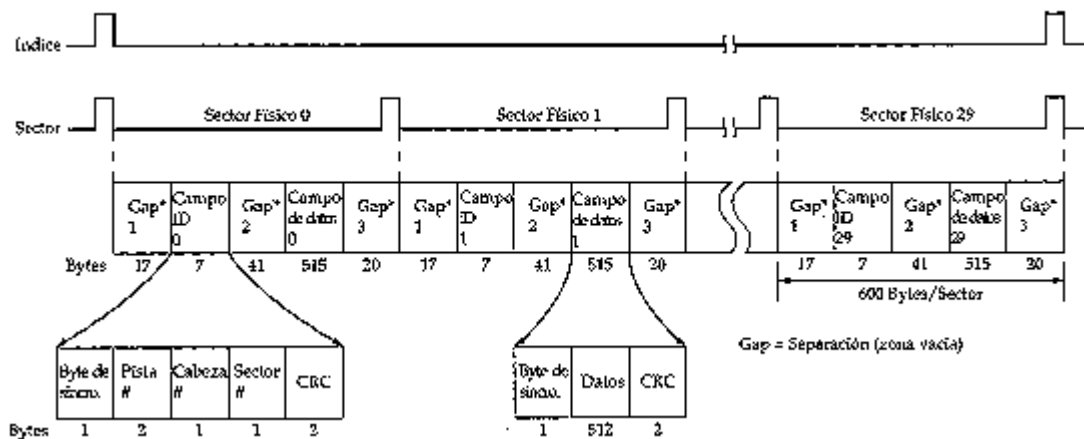


3.2.3 Formateo de un disco

Para que un disco pueda ser usado por un determinado sistema requiere una preparación previa. A esa preparación se le conoce como “formateo”. Existen dos tipos de formateos: el de bajo nivel o formateo físico y el de alto nivel o formateo del sistema operativo

Formateo a bajo nivel (formateo físico):

- Traslada la geometría del disco al soporte magnético del disco
- Es una característica física del disco, por tanto independiente del sistema operativo
- Se tiene que realizar primero
- Suele venir realizado de fábrica



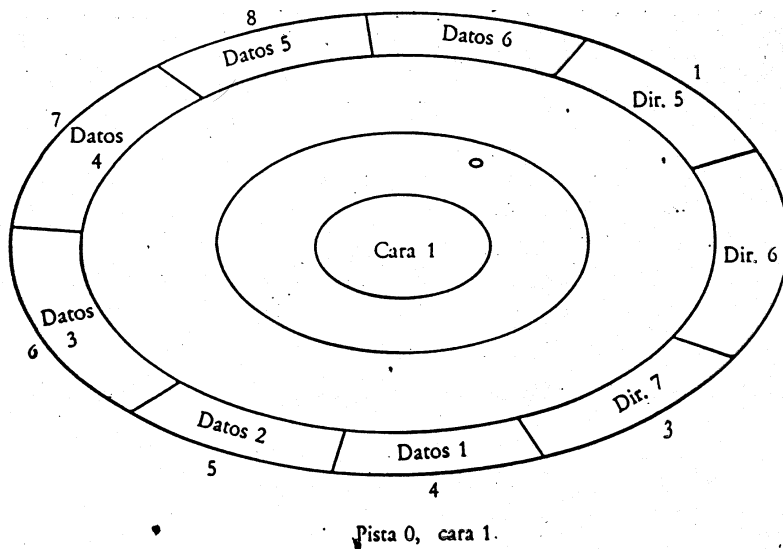
- La estructura física divide el disco según sus elementos físicos

Formateo a alto nivel (formateo Lógico):

- Prepara el disco para contener el sistema o sistemas de fichero que permita el sistema operativo concreto en que estemos trabajando.

Gestor de memoria

- Es propio de cada sistema operativo y del sistema de fichero que queramos utilizar en ese almacenamiento que estamos formateando.
- Un sistema operativo no tiene por qué “entender” el sistema de ficheros de otro sistema operativo.
- Un sistema operativo puede (y suele) trabajar con más de un sistema de fichero (propio o estándar).
- Un sistema operativo puede “entender” el sistema de ficheros de otro.



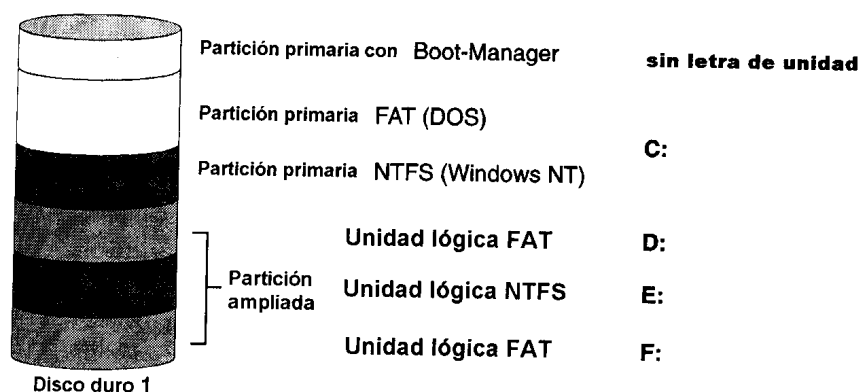
Partición

- Una partición es un conjunto de cilindros que forman una unidad independiente
- En cada partición se puede almacenar un sistema de ficheros distinto.
- En cada partición sólo puede haber un sistema de ficheros
- En un disco duro puede haber hasta cuatro particiones primarias

Tamaño Disquete	360K	720K	1.2MB	1.44MB	2.88MB
Sector Arranque	1	1	1	1	1
FAT (sectores)	4	10	14	18	36
Directorio Raíz (sectores)	7	7	14	14	28
Area Datos Usuario (sectores)	709	1427	2372	2848	5696
Tamaño cluster (sectores)	2	2	1	1	1
Total clusters	354	713	2371	2847	5695
Num. Máximo Entradas Directorio Raíz	112	112	224	224	448

- Los datos sobre las particiones que hay en el disco, se guardan en el *Master Boot Record*.
- La información de cada partición se guarda en el *boot record* de cada partición.

En terminología Microsoft, entiende de particiones primarias y particiones extendidas. Las particiones extendidas se emplean para definir dentro de ellas unidades lógicas de almacenamiento. Es un modo de solventar los problemas de los tamaños máximos de los discos y particiones que pueden utilizar.



Cluster

- Unidad de asignación de espacio de disco.
- Su tamaño depende del tipo de FAT y del tamaño del disco
- Su tamaño es un múltiplo del tamaño del sector. Está formado por un número entero de sectores.
- Si el tamaño del **cluster** es muy grande desperdiciamos espacio del disco sobre todo en los ficheros pequeños

Con FAT de 16 bits podemos direccionar $2^{16} = 65.536$ (64K) unidades de asignación.

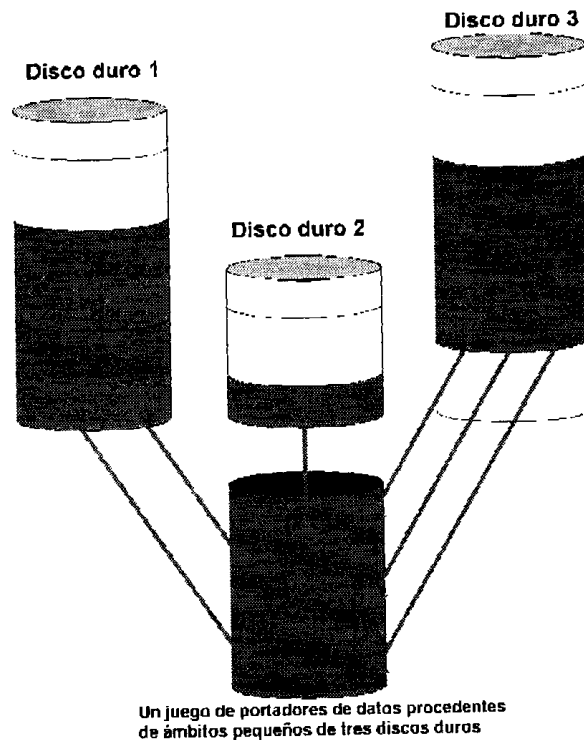
– $65.536 \times \text{Tamaño_del_cluster} = \text{tamaño máximo de la partición}$

Tamaño de la partición	Tamaño del CLUSTER	Nº de sectores
512 MB	8KB	16
Hasta 1 GB	16 KB	32
+ 1GB	32 KB	64

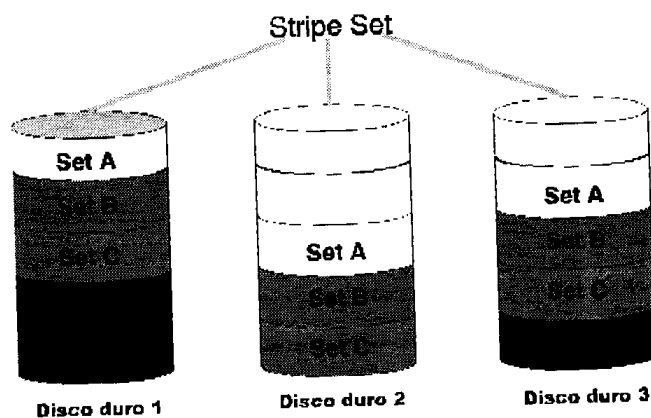
Volumen

Es un Conjunto de unidades de almacenamiento de espacios en disco que unimos bajo una letra de unidad común. El volumen puede ser dividido en particiones, y unidades lógicas.

Permite un uso eficaz de los discos pequeños y los espacios “restantes” disponibles. Pero el fallo de uno de los discos físicos impide el acceso al resto de la información.



Conjunto de bandas



En un conjunto de bandas, los datos no se escriben como una totalidad en un disco duro, sino en pequeñas partes denominadas bandas. Los conjunto de bandas puede extenderse por distintas partes de una unidad física o entre distintas unidades físicas.

Es posible crear conjunto de bandas con o sin paridad. Un conjunto de bandas sin paridad aporta mejoras en el rendimiento exclusivamente. Un conjunto de bandas con paridad, mejora la seguridad del sistema.

Disco Espejo

Con un disco espejo se reproducen los datos de una partición o de una unidad de un disco en un área libre de otro disco duro.

Pasos para preparar un disco duro

1. Formateo a bajo nivel (estructura física); viene del fabricante.
2. Hacer las particiones del disco (Con FDISK o equivalente)

3. Formateo a alto nivel (estructura lógica); Con la orden propia del sistema operativo.

3.3 Sistemas de ficheros

Los sistemas de ficheros organizan los datos en los discos distribuyéndolos en áreas físicamente accesibles del disco. El sistema de ficheros es más o menos "la forma de escribir los datos en el disco duro". En un sistema de ficheros podemos encontrar dos elementos para su administración:

- Los directorios.
- Los Ficheros.

Además el sistema de fichero tendrá que llevar el control del espacio libre y ocupado que hay en cada dispositivo.

3.3.1 Elementos de información de un directorio de ficheros

Un directorio no es más que un índice donde se recoge la información de los ficheros que hay almacenado en el dispositivo. Sirve para localizar a partir del nombre del fichero, dónde se encuentra físicamente almacenada la información. También se les conoce por las siglas FAT (tabla de alojamiento de ficheros).

Un directorio puede verse como una tabla que asocia nombres con direcciones de ubicación.

La información que se guardan en un directorio, varía de un sistema operativo a otro. Un ejemplo del tipo de información podría ser el siguiente:

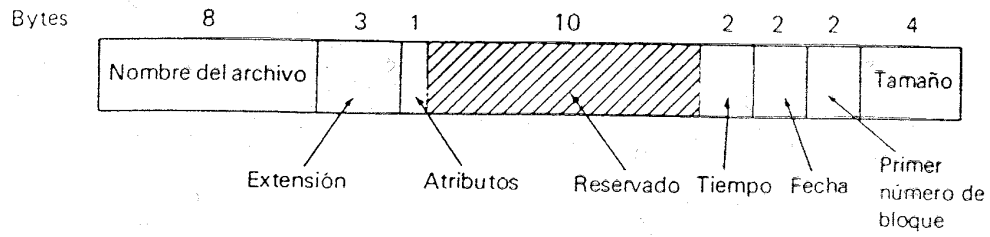
Información básica

- *Nombre del fichero*. Es elegido por el creador del fichero. Debe ser único dentro de un determinado directorio.
- *Tipo de fichero*. Por ejemplo: texto, binario etc.
- *Organización del fichero*. Para sistemas que soporten distintas organizaciones de ficheros.

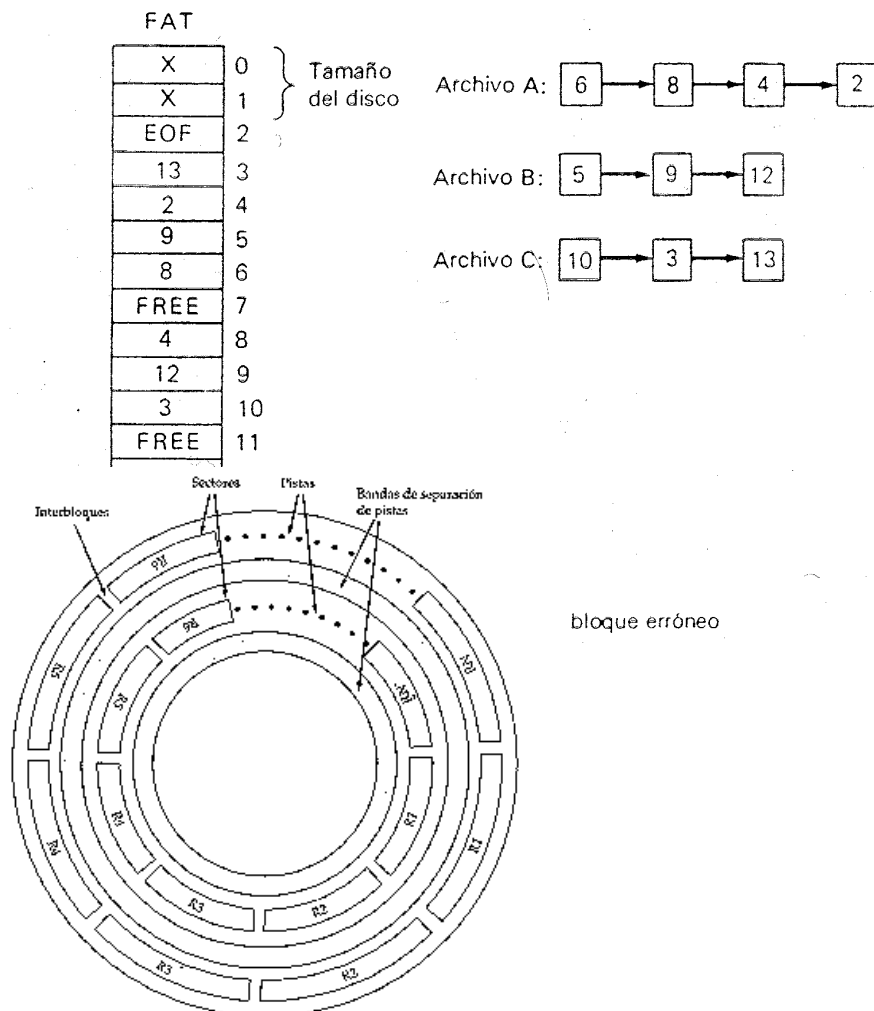
Información de direccionamiento

- *Volumen*. Indica el dispositivo donde el fichero es almacenado.
- *Dirección de comienzo* (localización). Dirección física de comienzo en el almacenamiento secundario (cilindro, pista, bloque del disco).
- *Tamaño usado*. Tamaño actual del fichero en *bytes*, palabras o bloques.
- *Tamaño asignado*. Tamaño máximo del fichero.

Figura ejemplo de la información de directorio de MS-DOS



Entrada del directorio de MS-DOS.



3El Sistema de Gestión de ficheros

- *Propietario*. Usuario que tiene asignado el control del fichero. Puede autorizar o denegar accesos a otros usuarios y cambiar los privilegios.
- *Información de acceso*.
- *Acciones permitidas*. Lectura, escritura, ejecución, transmisión por una red.

Información de uso

- Fecha de creación, identidad del creador, fecha del último acceso de lectura, identidad del último lector, fecha de la última modificación, identidad del usuario de la última modificación, fecha del último *backup*.
- *Posición actual*.
- *Uso actual*. Información sobre la actividad actual del fichero, tal como proceso o procesos que tienen abierto el fichero, si está bloqueado por un proceso, si el fichero ha sido actualizado en memoria principal pero no en disco etc.

El tamaño de un directorio puede variar entre 16 a 1.000 bytes por cada fichero dependiendo de los distintos sistemas operativos.

Los directorios se almacenan en memoria auxiliar debido a que pueden resultar voluminosos, aunque una parte se cargan en memoria principal para facilitar la velocidad de acceso.

3.3.2 Aspectos básicos de los ficheros

Los sistemas operativos además del nombre del fichero guardan otra información adicional sobre los ficheros.

Los atributos son características e informaciones que los sistemas operativos registran de cada fichero.

Atributos de los ficheros

La lista de atributos varía enormemente de un Sistema Operativo a otro.

Ejemplos:

- Información histórica y de medidas: Fecha de creación, del último cambio, o de la última lectura, número de veces que se ha utilizado etc.
- Permisos de accesos: Lectura, escritura, ejecución etc.
- Tamaños: Numero de registros, tamaño del fichero, del registro etc.
- Propietarios etc.
- Organización: secuencial, indexada, directa etc.

Nombre de los ficheros

Los nombres de los ficheros tiene distintas reglas según el sistema que se trate.

- Las longitudes de los nombres están comprendida entre el mínimo de 8 y un máximo de 255.
- El juego de caracteres suele estar formado por los dígitos, las letras y algunos caracteres especiales.
- Muchos sistemas admiten nombres con dos partes separadas por un punto. A la parte que sigue al punto se le llama **extensión**.

MS-DOS comand.com

UNIX prog.c.Z

- Algunas extensiones típicas y sus significados son:

file.bas	programa BASIC
file.pas	programa PASCAL
file.bak	fichero backup
file.dat	fichero de datos
file.txt	fichero texto

file.exe

fichero ejecutable

Algunos sistemas las "extensiones" de los nombres de los ficheros determinan o nos indican que tipo de ficheros son.

3.3.3 Métodos de accesos

El S.G.F debe de permitir realizar los métodos de accesos básicos sobre los ficheros en función de su organización interna.

Entendemos por métodos de accesos, los diversos modos en que podemos acceder a la información de un fichero.

Los métodos de acceso están relacionados con las organizaciones de los ficheros.

- Acceso secuencial.
- Acceso directo.
- Acceso dinámico.

3.3.4 Ejemplos de sistemas de ficheros:

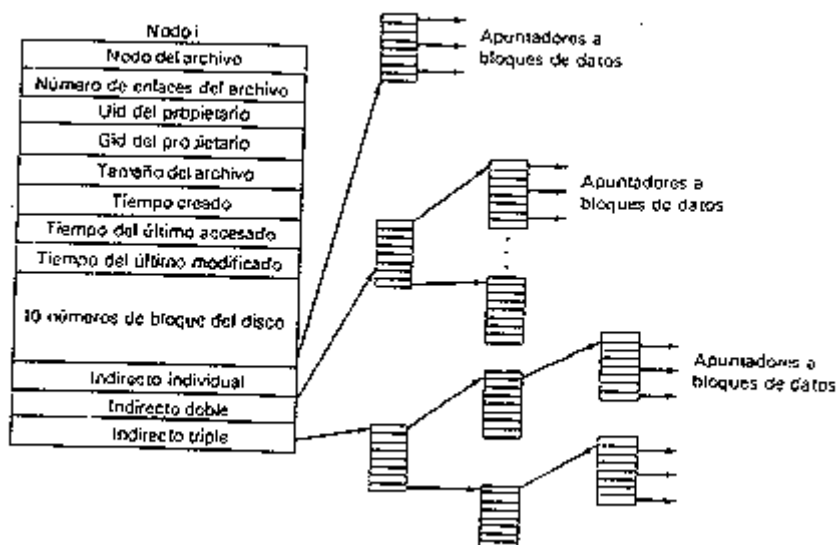
- FAT 16 (MS-DOS, windows 95, 98, NT)
- FAT 32 (Windows 95 SR2)
- NTFS (Windows NT)
- HPFS (*High Performance File System*) OS/2
- 9660 (sistema de ficheros estándar del CD-ROM)
- EXT2
- ext3
- EXT3,
- ReiserFS y
- XFS.

Sistemas de ficheros en los que pueden trabajar distintos sistemas operativos:

WINDOWS 95	WINDOWS NT	UNIX
FAT16	FAT16	EXT2FS
FAT32	NTFS	EXTFS
9660	HPFS (OS/2)	FAT16
	OTROS	SMB
	9660	9660
		NFS
		SWAP
		MINIX
		XIAF
		Y MUCHOS MAS

3.3.5 Sistema de ficheros del UNIX

La estructura de directorios que sigue Linux es parecida a la de cualquier UNIX. No tenemos una "unidad" para cada unidad física de disco o partición como en Windows, sino que todos los discos duros o de red se montan bajo un sistema de directorios en árbol, y algunos de esos directorios enlazan con estas unidades físicas de disco. Las barras en Linux al igual que en cualquier UNIX son inclinadas hacia la derecha. Los directorios principales:



Estructura de directorios en Linux

Directorio	Descripción
/	Es la raíz del sistema de directorios. Aquí se monta la partición principal Linux EXT.
/etc	Contiene los archivos de configuración de la mayoría de los programas.
/home	Contiene los archivos personales de los usuarios.
/bin	Contiene comandos básicos y muchos programas.
/dev	Contiene archivos simbólicos que representan partes del hardware, tales como discos duros, memoria...
/mnt	Contiene subdirectorios donde se montan (se enlaza con) otras particiones de disco duro, CDROMs, etc.
/tmp	Ficheros temporales o de recursos de programas.
/usr	Programas y librerías instalados con la distribución
/usr/local	Programas y librerías instalados por el administrador
/sbin	Comandos administrativos
/lib	Librerías varias y módulos ("trozos") del kernel
/var	Datos varios como archivos de log (registro de actividad) de programas, bases de datos, contenidos del servidor web, copias de seguridad...
/proc	Información temporal sobre los procesos del sistema (explicaremos esto más en profundidad posteriormente).

Nombramiento de dispositivos y particiones

Debemos saber de qué manera nombra Linux a los discos duros que tenemos conectados a nuestra máquina y sobre todo a sus particiones. Todos los discos duros (IDE) comienzan su nombre como `hd`. Un ejemplo de nombre completo de disco duro sería `hda` y de la primera partición de ese disco duro sería `hda1`.

La 'a' significa que ese disco duro está conectado al IDE1 como maestro. Si fuera esclavo tendría la 'b', y si estuviera conectado al IDE2 como maestro, la 'c', y si estuviera al IDE2 como esclavo, la 'd'.

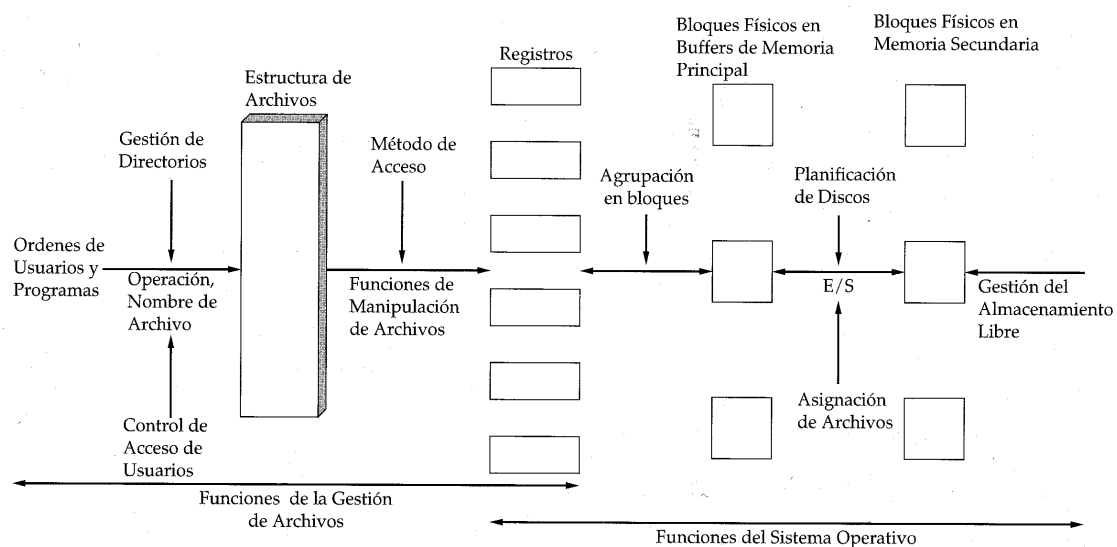
El número 1 indica que es la primera partición (primaria y no lógica) del disco duro en cuestión. La segunda geométricamente hablando (primaria) sería la 2 y así sucesivamente. La primera partición lógica de un disco duro se nombra con el número 5, independientemente de si pertenece a la primaria 1, 2, 3 ó 4. La segunda se nombraría con un 6 y así sucesivamente.

Con saber lo que significa `hda1` o `hdd2` o `hdc5` es suficiente de momento.

3.3.6 Estructura jerárquica del SGF

El sistema de gestión de ficheros, está estructurado en capas. Es dependiente del sistema operativo concreto. Una visión general pudiera ser la que sigue:

1. Programa de aplicación.
2. Sistema de directorios.
3. Métodos y control de accesos.
4. Sistema de fichero lógico (organización lógica).
5. Sistema de fichero físico (organización física).
6. Manejadores de los dispositivos (*devices driver*) Interfaces con los dispositivos y control de la E/S.
7. Controladoras y canales de los dispositivos.
8. Dispositivos de E/S.



4 Sistema de entrada salida

4.1 Funciones

Una de las funciones de los Sistemas Operativos es *controlar todos los dispositivos de entrada/salida*. El **sistema de E/S** es la parte del Sistema Operativo encargada de la gestión de los dispositivos de E/S.

La problemática de la E/S es la gran variedad de dispositivos distintos existentes, cada uno con sus características:

- Eléctricas.
- De velocidad.
- De formatos: códigos -EBCDIC, ASCII, HOLLERITH.

4 Sistema de entrada salida

- Transmisión: serie, paralelo.
- Modo: a carácter o a bloque y de sincronización particulares.

Esto requiere que el sistema operativo se encargue de la diversidad de dispositivos, siempre que se le suministre la información adecuada.

Componentes

Los componentes básicos del subsistema de entrada/salida son:

- Los componentes *hardware* del subsistema de E/S.
- El *software* del subsistema de E/S

4.2 Componentes *hardware* del subsistema de e/s

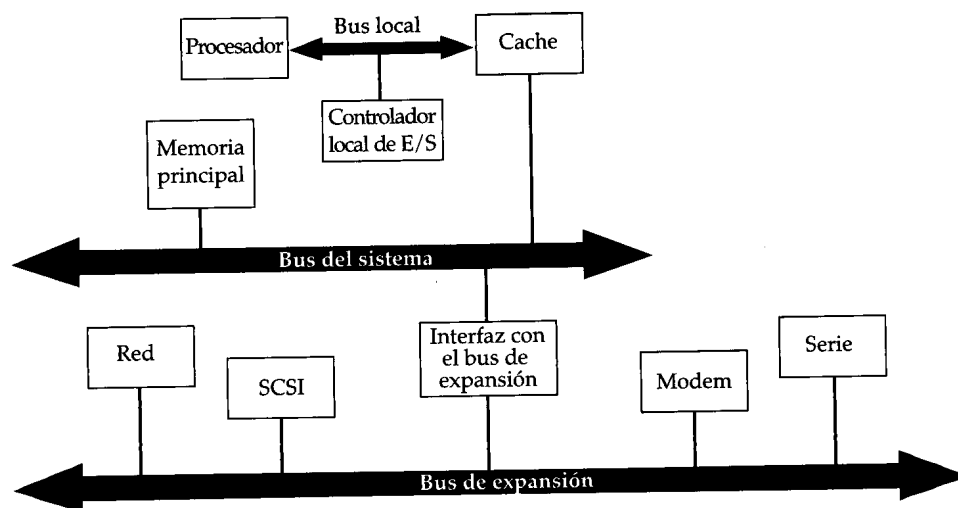
Los elementos básicos del *hardware* que intervienen subsistema de E/S son:

- El procesador
- La memoria principal
- Los periféricos o dispositivos
- Las unidades de control (controladoras)
- DMA
- Canales
- Bus

Del procesador y la memoria principal ya hemos hablado.

4.2.1 Bus

Los tipos de buses y de interconexión lo podemos ver en la siguiente figura:



Tenemos tres categorías de buses:

- *Bus local*: Comunica el procesador con la memoria cache de nivel 2.
- *Bus del sistema*: Interconecta el procesador con la memoria central y los demás sistemas de interconexión.
- *Bus de expansión* o de E/S: Se interconecta con el bus del sistema y los periféricos.

Buses estándares

Bus PCI (*Peripheral Component Interconnet*). Bus de datos de 32 bits y 33Mhz, con anchos de banda de 132 Mb/seg. *Plug and play*. Unos 50 contactos.

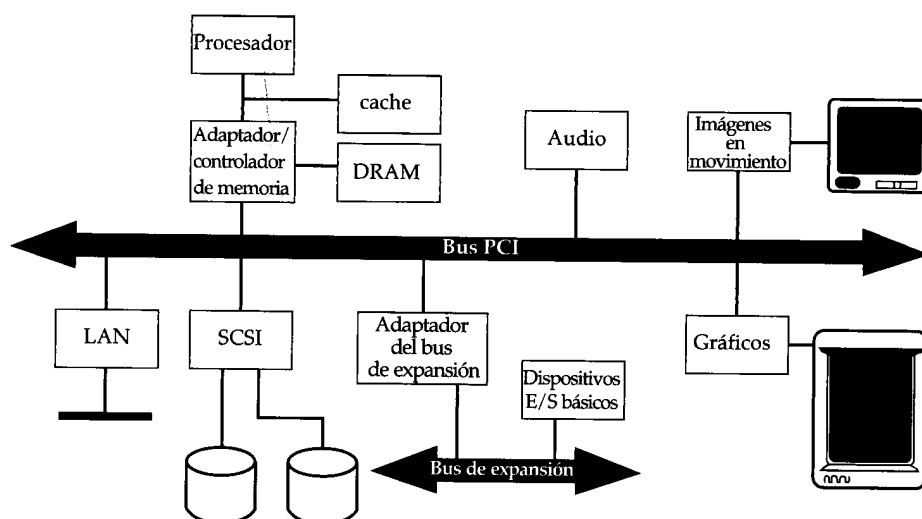
Controladora de disco EIDE (*Enhanced Integrated Drive Electronic*). Velocidad de 11.1 Mb/seg. Hasta 4 unidades por controladora. *Fast ATA-2* hasta 16,6 Mb/seg. *ATA-4* o *UltraDMA* 66Mb/seg.

Bus SCSI (*Small Computer System Interface*)

SCSI-2.	Fast SCSI	8 bits; 10 Mhz; 10 Mb/s; 8 dispositivos
	Wide SCSI	16 bits; 5 Mhz 10Mb/s; 16 dispositivos
	Fast Wide SCSI	16 bits, 10 Mhz; 20 Mb/s; 16 dispositivo
SCSI-3	Ultra SCSI	8 bit 20 Mhz 20Mb/s 8 dispositivos
	Ultra Wide SCSI	16 bit 20 Mhz 40 Mb/s 16 dispositivos
	Ultra 2 SCSI	8 bit 40 Mhz 40Mb/s 8 dispositivos
	Ultra 2 Wide SCSI	16 bit 40 Mhz 80Mb/s 16 dispositivos
	<i>FireWire</i> (IEEE –1394)	1 bit serie 400 Mhz a 1Ghz 400 Mb/s 63 dispositivos
	Ultra 3 SCSI	8 bits 80 Mhz 80 Mb/s 8 dispositivos
SCSI-FCP		266Mb/s a 4Gb/s hasta 10 Km

Bus AGP (*Accelerated graphics Port*) Solo para tarjetas gráficas 3D. 533 Mb/seg. El AGP , (Puerto de Gráficos Acelerados) es una especificación de bus que permite que se despliegan rápidamente gráficos en 3-D en ordenadores personales comunes. El AGP es una interfaz especial diseñada para transmitir imágenes en 3-D (por ejemplo, de páginas Web o CD-ROMs) mucho más veloz y ágilmente de lo que es posible hoy en una ordenador que no sea una costosa estación de trabajo gráfica. La interfaz usa el almacenamiento principal del ordenador (RAM) para refrescar la imagen del monitor y soportar el mapeo de texturas, el *z-buffering* y la mezcla alfa que se requieren para el despliegue de imágenes en 3-D. El uso que hace el AGP de la memoria principal es dinámico, lo cual significa que cuando no se está utilizando para gráficos acelerados, la memoria principal se devuelve para uso del sistema operativo u otras aplicaciones.

Puerto USB. Velocidad de hasta 12 Mb/seg. Puede acceder a un amplio número de dispositivos externos *plug and play* (127). Se pueden añadir o quitar dispositivos mientras el ordenador está encendido, configurándose automáticamente el dispositivo nada más conectarse. El estándar USB 2.0 tiene una alta velocidad de 480 Mbit/s.



FireWire (IEEE 1394) es un estándar multi-plataforma para entrada/salida de datos en serie a gran velocidad. Suele utilizarse para la interconexión de dispositivos digitales como cámaras digitales y video-cámaras a ordenadores. Alcanzan una velocidad de 400 megabits por segundo.

PCI-Express se basa en un sistema de comunicación serie mucho más rápido. PCI-Express está pensado para ser usado sólo como bus local. La velocidad superior del PCI-Express permitirá reemplazar casi todos los demás buses, AGP y PCI incluidos. Altísimo ancho de banda, desde 200MB/seg para la implementación 1X, hasta 4GB/seg para el PCI Express 16X que se empleará con las tarjetas gráficas.

4.2.2 Periféricos

Un periférico es un dispositivo que es capaz de transferir información desde o hacia un soporte externo. Requiere estar conectado de alguna forma con el sistema.

Modos de interconexión

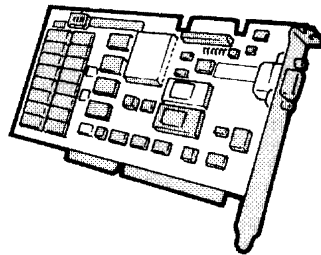
Los periféricos se pueden conectar al sistema de diversas formas:

- Por medio de controladores o interfaces.
- Con DMA.
- Por canales

4.2.3 Controladores de dispositivos

Es posible dentro de los dispositivos de E/S distinguir una parte mecánica y otra electrónica.

- *Los controladores o adaptadores* de dispositivos, que son la placa o **componente electrónico** del peri-

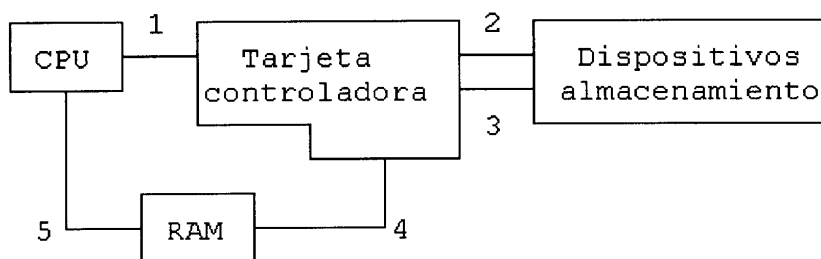


férico que está situado entre el dispositivo y el bus del sistema.

- La *parte mecánica* es el dispositivo en sí mismo.

Su función es hacer de puente entre los periféricos y la CPU adaptando las distintas velocidades, señales eléctricas, códigos etc. existentes entre el periférico y la CPU.

Ejemplo: Convierte una serie de bits en bloques o viceversa.



La controladora se va a entender directamente con el periférico por un extremo y con el bus por otro. El sistema operativo se suele entender con la controladora no con el dispositivo

Ejemplos de controladoras:

- Controladora de disco duro y de disco flexible.
- RS-232-c (Puerta serie), *centronic* (Puerta paralela).
- Controladoras de vídeo.
- Controladoras de redes locales etc.

4.3 Técnicas de Comunicación con el procesador

Hay tres técnicas para la realización de las operaciones de E/S:

- E/S controlada por programa (programada).
- E/S guiada por interrupciones.
- E/S con acceso directo a memoria.

E/S controlada por programa: El procesador emite una orden de E/S de parte de un proceso a un módulo de E/S; el proceso espera entonces a que termine la operación, antes de seguir.

4.3.1 Operaciones de E/S guiada por interrupciones

Las operaciones de E/S guiadas por interrupciones tienen el siguiente modo de operar:

- El procesador inicia una operación de E/S de parte de un proceso y continua con la ejecución de la siguiente instrucción.
- El módulo de E/S interrumpe al procesador cuando está listo para intercambiar datos.

- El procesador no espera la terminación de la operación de E/S.

Con este sistema el procesador sigue siendo el responsable de la transferencia de todos los datos entre el procesador y el dispositivo. La transferencia de información se realiza de memoria al procesador y del procesador al dispositivo.

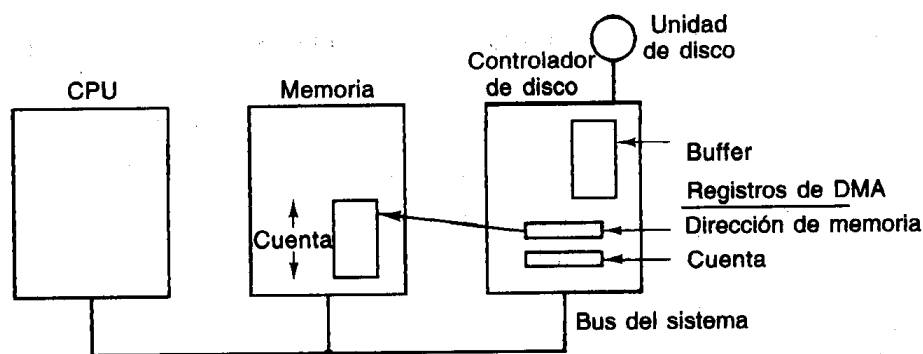
- Las E/S conducidas por interrupciones alivian al procesador **de la espera** durante cada evento de E/S.

En dispositivos con gran ancho de banda, como un disco magnético, la transferencia ocuparía al 100% al procesador.

4.3.2 Operaciones de E/S con acceso directo a memoria (DMA)

Mecanismo para descargar al procesador del trabajo de la transferencia de datos entre la memoria/dispositivo. El DMA permite que el controlador del dispositivo transfiera datos directamente a/o desde la memoria sin involucrar al procesador.

- - El módulo de DMA controla el intercambio de datos entre memoria y el dispositivo.
- - El procesador envía una petición de transferencia de un bloque de datos al DMA y sólo es interrumpido cuando el bloque completo es transferido.
- ∃ El mecanismo de la interrupción es empleado por el dispositivo para comunicarse con el procesador a la terminación de la transferencia de e/s o cuando ocurre un error.



Algunas controladoras permiten la transferencia de forma autónoma bloques de información desde o hacia la memoria sin la intervención de la CPU. Se utiliza en dispositivos de bloques principalmente (discos).

- El DMA se puede utilizar para interconectar un disco, sin que se consuman todos los ciclos del procesador en la operación de E/S.

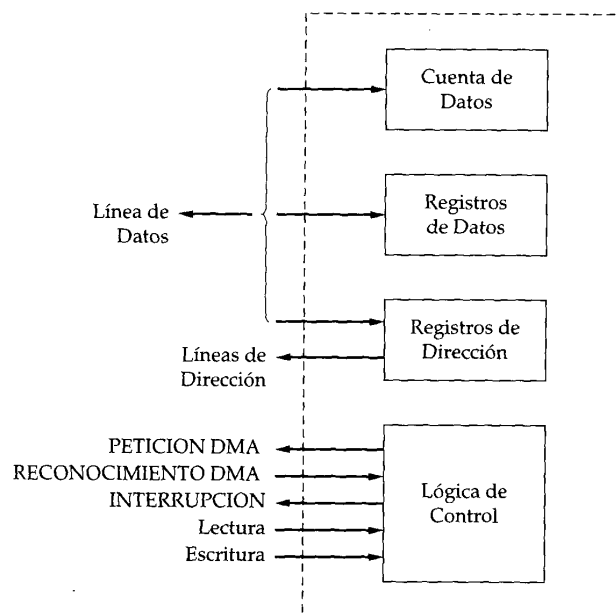
Sin DMA, la CPU es la responsable de la transferencia física de los datos entre el dispositivo y la memoria.

Funcionamiento

- 1.- El procesador inicia la operación de E/S cargando en los registros del DMA, los datos de la operación:
 - Identidad del dispositivo.
 - Operación a realizar.
 - Dirección de memoria de la fuente o destino de los datos.
 - Número de *bytes* a transferir.
- 2.- El procesador queda libre.
- 3.- El DMA va obteniendo el control del bus para realizar la operación.
- 4.- Terminada la operación genera una interrupción.

5.- El procesador interrogando al DMA o la memoria, determina si la operación fue completada con éxito o no.

DMA	Bus Line ?	Typical Default Use	Other Common Uses
0	no	Memory Refresh	None
1	8/16-bit	Sound card (low DMA)	SCSI host adapters, ECP parallel ports, tape accelerator cards, network cards, voice modems
2	8/16-bit	Floppy disk controller	Tape accelerator cards
3	8/16-bit	None	ECP parallel ports, SCSI host adapters, tape accelerator cards, sound card (low DMA), network cards, voice modems, hard disk controller on old PC/XT
4	no	None; cascade for DMAs 0-3	None
5	16-bit only	Sound card (high DMA)	SCSI host adapters, network cards
6	16-bit only	None	Sound cards (high DMA), network cards
7	16-bit only	None	Sound cards (high DMA), network cards



4.4 Principios del software de e/s

Vamos a analizar ahora como es la estructura del *software* de E/S.

4.4.1 Objetivos del software de E/S

Organizar el *software* en estratos de tal forma que los niveles inferiores oculten las peculiaridades del *hardware* a los niveles superiores.

4.4.2 Estratos del *software* de e/s

Es *software* de E/S se puede estructurar en distintos estratos.

- a.- Manejadores de interrupciones.
- b.- Manejadores o *drivers* de dispositivos.
- c.- *Software* del Sistema Operativo independiente del dispositivo.



d.- *Software* a nivel de usuario.

Manejadores o *drivers* de dispositivos

Cada periférico tiene un modo peculiar de trabajar. Cada fabricante conoce el funcionamiento de su impresora o de su disco. El Sistema Operativo, para poder entenderse con el dispositivo necesita un elemento intermedio entre sus órdenes generales de E/S y las específicas de cada dispositivo y cada fabricante. El manejador del dispositivo o *driver* es esa capa intermedia que traduce las órdenes generales del sistema operativo al controlador *hardware* del dispositivo. Es decir

- Es el *software* que está en contacto directo con el *hardware* del dispositivo.
- Maneja a los controladores de dispositivos.
- Acepta órdenes del *software* independiente de dispositivos y las transmite a las controladoras.

Cuando compramos un periférico, suele venir con un dispositivo de almacenamiento (disco o CD-ROM) que incorpora los distintos *Drivers* para los distintos sistemas operativos. Antes de poder funcionar correctamente con nuestro nuevo dispositivo, además de conectarlo físicamente al equipo, debemos de instalar en nuestro sistema operativo su *driver* correspondiente.

Es interesante distinguir entre la controladora de un dispositivo, que es un elemento *hardware* y el *manejador* o *driver* del dispositivo que es un elemento *software*.

Si cambiamos de sistema operativo en nuestro ordenador, tendremos que buscar en Internet, *drivers* actualizados de nuestros dispositivos para el Sistema Operativo que hayamos instalado. Se buscan en los sitios de los fabricantes y en sitios especializados en *drivers*.

Software de E/S independiente del dispositivo

Sus funciones son:

- Ejecutar todas las operaciones de E/S que son comunes a todos los dispositivos
- Proporcionar una interfaz uniforme al *software* del nivel de usuario.

Este elemento preocupa o es útil a los programadores no a los usuarios finales.

4.5 RAID

El RAID (*redundant array of independent [inexpensive] disks*) es una forma de almacenar los mismos datos en distintos lugares (por tanto de modo redundante) en múltiples discos duros. Al colocar los datos en discos múltiples, las operaciones I/O (*input/output*, de entrada y salida) pueden superponerse de un modo equilibrado, mejorando el rendimiento del sistema. Dado que los discos múltiples incrementan el tiempo medio entre errores (*mean time between failure*, MTBF), el almacenamiento redundante de datos incrementa la tolerancia a fallos.

Un RAID, para el sistema operativo, aparenta ser un sólo disco duro lógico. El RAID emplea la técnica conocida como "*striping*" (bandeado o creación de bandas), que incluye la partición del espacio de almacenamiento de cada disco en unidades que van de un sector (512 bytes) hasta varios megabytes. Las bandas de todos los discos están interpaginadas (*interleaved*) y se accede a ellas en orden.

En un sistema de un solo usuario donde se almacenan grandes registros (como imágenes médicas o de otro tipo), las bandas generalmente se establecen para ser muy pequeñas (quizá de 512 bytes) de modo que un solo registro esté ubicado en todos los discos y se pueda acceder a él rápidamente leyendo todos los discos a la vez.

En un sistema multiusuario, un mejor rendimiento demanda que se establezca una banda lo suficientemente ancha para contener el registro de tamaño típico o el de mayor tamaño. Esto permite acciones I/O superpuestas en los distintos discos.

Hay al menos nueve tipos de RAID además de un grupo no redundante (RAID-0):

RAID-0. Esta técnica tiene bandeado pero no tiene redundancia de datos. Ofrece el mejor rendimiento pero no tolerancia a los fallos.

RAID-1. Este tipo también se conoce como creación de discos espejo y consiste de al menos dos discos duros que duplican el almacenamiento de datos. No hay bandeado. El rendimiento de la lectura se mejora pues cualquiera de los dos discos puede leerse al mismo tiempo. El rendimiento de escritura es el mismo que el del almacenamiento en un solo disco. El RAID-1 proporciona el mejor rendimiento y la mejor tolerancia a fallos en un sistema multiusuario.

RAID-2. Este tipo usa bandeado en todos los discos, con algunos de estos dedicados a almacenar información de verificación y corrección de errores (*error checking and correcting*, ECC). No tiene ninguna ventaja sobre el RAID-3.

RAID-3. Este tipo usa bandeado y dedica un disco al almacenamiento de información de paridad. La información de verificación de errores (ECC) incrustada se usa para detectar errores. La recuperación de datos se consigue calculando el O exclusivo (XOR) de la información registrada en los otros discos. Dado que una operación I/O accede a todos los discos al mismo tiempo, el RAID-3 no puede traslapar I/O. Por esta razón, el RAID-3 es mejor para sistemas de un solo usuario con aplicaciones que contengan grandes registros.

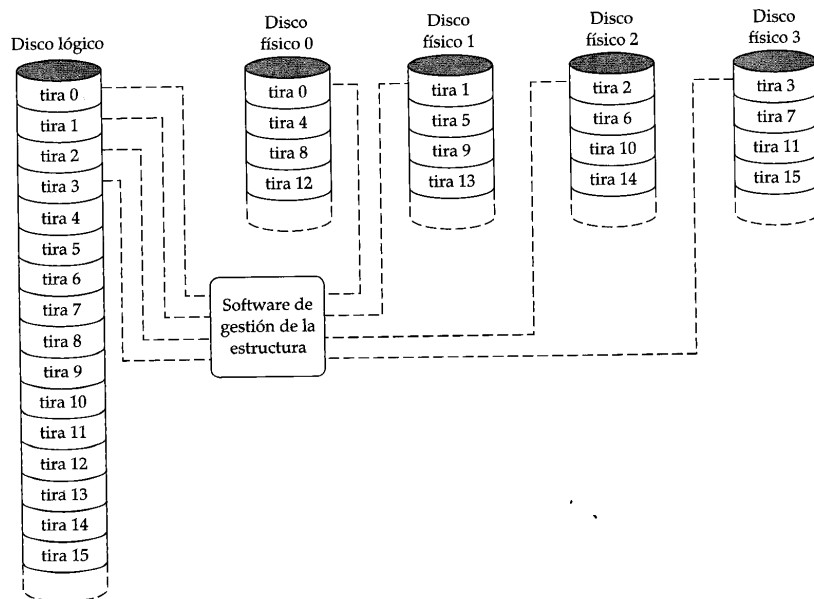
RAID-4. Este tipo usa grandes bandas, lo cual significa que podemos leer registros de cualquier disco individual. Esto nos permite aprovechar la I/O traslapada para las operaciones de lectura. Dado que todas las operaciones de escritura tienen que actualizar el disco de paridad, no es posible la superposición I/O para ellas. El RAID-4 no ofrece ninguna ventaja sobre el RAID-5. **RAID-5.** Este tipo incluye un grupo rotatorio de paridad, con lo que resuelve las limitaciones de escritura en RAID-4. Así, todas las operaciones de lectura y escritura pueden superponerse. El Raid 5 almacena información de paridad pero no datos redundantes (aunque la información de paridad puede usarse para reconstruir datos).

RAID-5 exige al menos tres y usualmente cinco discos en el conjunto. Es mejor para los sistemas multiusuario en los cuales el rendimiento no es crítico, o que realizan pocas operaciones de escritura.

RAID-6. Este tipo es similar al RAID-5, pero incluye un segundo esquema de paridad distribuido por los distintos discos y por tanto ofrece tolerancia extremadamente alta a los fallos y las caídas de disco. Hay pocos ejemplos comerciales en la actualidad.

RAID-7. Este tipo incluye un sistema operativo incrustado de tiempo real como controlador, haciendo las operaciones de caché a través de un bus de alta velocidad y otras características de un ordenador sencillo. Un vendedor ofrece este sistema.

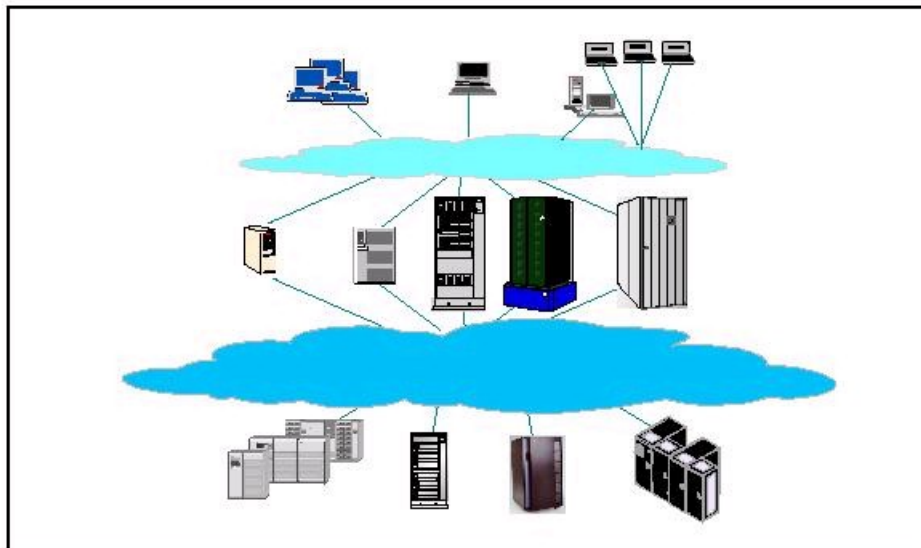
RAID-10. Este tipo ofrece un conjunto de bandas en el que cada banda es un grupo de discos RAID-1. Esto proporciona mejor rendimiento que el RAID-1, pero a un costo mucho mayor.



RAID-53. Este tipo ofrece un conjunto de bandas en el cual cada banda es un conjunto de discos RAID-3. Esto proporciona mejor rendimiento que el RAID-3, pero a un costo mucho mayor.

4.6 Storage Area Networks (SAN)

El nuevo concepto se define como "redes de dispositivos de almacenamiento gestionados de forma centralizada, conectados entre sí y con todos los servidores de la empresa a través de sistemas de alta velocidad y que permiten a las empresas explotar al máximo sus sistemas información".

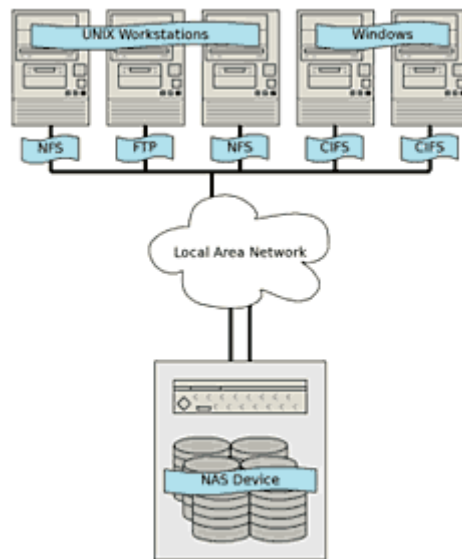


4.7 NAS (del inglés *Network Attached Storage*)

Es una arquitectura que permite el acceso desde los [PC](#) o [servidores](#) a discos externos del mismo modo que si estuvieran conectados directamente, con la ventaja de la compartición de la información. Se accede desde los

Gestor de memoria

equipos a través de protocolos de red (normalmente [TCP/IP](#)).



5 Cuestiones

1. Qué es un proceso
2. Diferencia entre proceso y programa
3. Nombra los tres elementos que forman un proceso
4. Diferencia entre la planificación apropiativa y no apropiativa.
5. Asocia planificación apropiativa y planificación no apropiativa con:
 - El proceso que se está ejecutando, puede ser interrumpido en cualquier momento y movido a la cola de listos antes de su completa ejecución
 - Una vez asignada la CPU a un proceso, no se le puede retirar hasta que la libera
6. Sería adecuado una planificación No apropiativa para un sistema de tiempo real ¿Por qué?
7. Nombra tres algoritmos de planificación.
8. Ordena los siguientes niveles de memoria(caché de disco, cache interna, registro, memoria principal, caches externa)
9. Características de las memorias caches
10. Concepto de localidad temporal
11. Los sistemas operativos ofrecen una serie de “servicios”. ¿Mediante qué mecanismo solicitan los programas los servicios al sistema operativo?
12. Pon por orden de “jerarquía” las siguientes memorias: Almacenamiento secundario, registros, caches externas, memoria principal, caches internas.
13. Las memorias caches son:(Señala la respuesta correcta)
 - Grandes y lentas
 - Pequeñas y lentas
 - Pequeñas y rápidas
14. Clasificación de las políticas de administración de memoria
15. Concepto de intercambio (*swapping*)
16. Concepto de memoria virtual
17. Ventajas de la memoria virtual
18. Concepto de fallo de página
19. Pasos para preparar un disco duro
20. Características principales de la memoria virtual
21. Pon un ejemplo de políticas de gestión de memoria con asignación completa y contigua, asignación completa no contigua y asignación parcial y no contigua
22. ¿Qué entiendes por “Sistema de ficheros”?
23. ¿Cuál es el sistema de ficheros específico del sistema operativo Windows XP?
24. Nombra tres sistemas de ficheros que conozcas y dí los sistemas operativos lo soportan
25. Nombra varios sistemas operativos que conozcas que sí incorporen memoria virtual. ¿Conoces alguno que no incorpore memoria virtual?
26. Parámetros que determinan la geometría de un disco
27. Formateo a alto nivel
28. Concepto de *cluster* (en un sistema de ficheros)
29. Nombra tres sistemas de ficheros
30. A qué corresponde las siglas SCSI ¿Y USB?
31. Diferencia entre controlador y driver o manejador de dispositivo

32. Pon tres ejemplos de controladoras
33. ¿Qué es lo que hace el DMA?
34. ¿Qué significado tienen las siglas RAID?
35. ¿Qué es un disco espejo?
36. Concepto de "Volumen"
37. Concepto de "Banda" de discos
38. ¿Para qué sirven los "directorios" en los sistemas de ficheros?
39. ¿Todos los sistemas operativos utilizan los mismos sistemas de ficheros?
40. ¿Qué diferencia hay entre una FAT16 y una FAT 32?
41. ¿A qué corresponden las siglas NTFS?
42. ¿Cuántas particiones primarias puede tener un disco?
43. Concepto de Particiones extendidas
44. ¿Para que se pueden utilizar las particiones extendidas?
45. ¿Puedo instalar en una misma partición dos sistemas de ficheros?
46. ¿Dónde se almacena la información sobre las particiones de un disco?
47. ¿A qué corresponde las siglas NAS?
48. ¿Y las siglas SAN?