

UNIDAD 2: SISTEMAS OPERATIVOS. ELEMENTOS, ESTRUCTURA Y FUNCIONES GENERALES

**Sistemas Operativos Monopuesto
Curso 2012/2013**

U.T.2: SISTEMAS OPERATIVOS. ELEMENTOS, ESTRUCTURA Y FUNCIONES GENERALES

- 1. Introducción
- 2. Evolución Histórica
- 3. Tipos de Sistemas Operativos
- 4. Estructura de un Sistema Operativo
- 5. Funciones de un Sistema Operativo
 - 5.1. Gestión de Procesos
 - 5.2. Gestión de Memoria
 - 5.3. Gestión de E/S
 - 5.4. Gestión de Archivos
- 6. Ejercicios

1. INTRODUCCIÓN

○ Definición de Sistema Operativo

“Conjunto de programas y funciones, relacionados entre sí, que gestionan y coordinan el funcionamiento del hardware y del software del sistema, permitiendo hacer un uso del ordenador de una forma eficiente y apropiada”

1. INTRODUCCIÓN

- Objetivos del Sistema Operativo:
 - Incrementar la productividad (fundamental)
 - Seguridad de la información, los usuarios y el propio sistema
 - Abstracción: se enmascaran los recursos

1. INTRODUCCIÓN

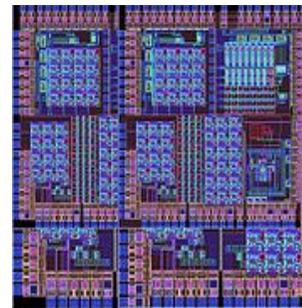
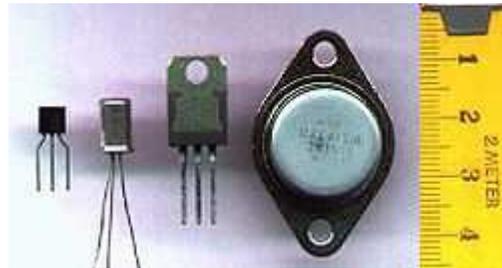
- Funciones del Sistema Operativo:
 - Comunicación con el usuario
 - Control y gestión de elementos físicos
 - Utilización por varios usuarios
 - Ejecución de varios programas
 - Compartición de Información
 - Compartición de Periféricos
 - Comunicación entre sistemas

2. EVOLUCIÓN HISTÓRICA

- Sistemas operativos monolíticos
- Evolución del sistema operativo paralela a la evolución hardware
- Charles Babbage:
 - Primera computadora
 - Secuencia **Entrada-Proceso-Salida**
- George Boole
 - Teoría de la lógica matemática
 - Álgebra de Boole

2. EVOLUCIÓN HISTÓRICA

- Primera generación (1943 a 1959)
- Segunda generación (1960 a 1965)
- Tercera generación (1966 a 1971)
- Cuarta generación (1971 a 1981)
- Quinta generación (1981 hasta la actualidad)



3. TIPOS DE SISTEMA OPERATIVO

- Distintas clasificaciones:
 - Según el tiempo de respuesta
 - Según el número de usuarios
 - Según el número de procesos
 - Según el número de procesadores
 - Trabajo en Red

3. TIPOS DE SISTEMA OPERATIVO

- Tiempo de Respuesta
 - Tiempo que tarda el usuario del sistema en obtener los resultados después de iniciar la ejecución de un programa

3. TIPOS DE SISTEMA OPERATIVO

- Tiempo de Respuesta
 - **Procesamiento por lotes**
 - Disponibilidad de la CPU total para cada proceso, se ejecutan uno tras otro
 - **Tiempo real**
 - La respuesta es inmediata (o casi inmediata) después de iniciar el proceso
 - **Tiempo compartido**
 - Cada proceso utilizará ciclos de la CPU hasta que finalice

3. TIPOS DE SISTEMA OPERATIVO

- Número de usuarios

- **Monousuario**

- Sólo un usuario trabaja con un ordenador
- Todos los dispositivos de hardware están a disposición de ese usuario y no pueden ser utilizados por otros hasta que éste no finalice su uso

3. TIPOS DE SISTEMA OPERATIVO

- Número de usuarios
 - **Multiusuario**
 - Varios usuarios pueden utilizar los recursos del sistema simultáneamente
 - Pueden compartir los dispositivos externos de almacenamiento, los periféricos de salida , el acceso a una misma base de datos instalada en el ordenador principal

3. TIPOS DE SISTEMA OPERATIVO

- Número de procesos: es independiente el número de procesadores
- **Monoprogramación o monotarea**
 - Sólo puede ejecutar un programa o proceso por vez
 - Los recursos del sistema estarán dedicados al programa hasta que finalice su ejecución

3. TIPOS DE SISTEMA OPERATIVO

- Número de procesos
 - **Multiprogramación o multitarea**
 - Varios programas o procesos de forma concurrente o simultánea.
 - La CPU comparte el tiempo de uso del procesador entre los diferentes programas que se desean ejecutar
 - Todos los procesos tardarán individualmente más tiempo en ejecutarse, pero, comparándolo con la monotarea, el tiempo medio de espera será mucho menor

3. TIPOS DE SISTEMA OPERATIVO

- Número de procesadores
 - **Monoproceso**
 - El ordenador sólo tiene un procesador
 - Todos los trabajos a realizar pasarán por él

3. TIPOS DE SISTEMA OPERATIVO

- Número de procesadores
 - **Multiproceso**
 - El ordenador cuenta con dos o más procesadores
 - Determinados SO pueden aprovechar las ventajas de este tipo de hardware
 - Todos los sistemas operativos multiusuario actuales son multitarea
 - A excepción de Windows 9X y ME, en desuso
 - Ejemplos: Windows NT 4.0, Windows 2000, Windows XP, Windows Vista, Linux, Unix, etc

3. TIPOS DE SISTEMA OPERATIVO

- Número de procesadores
 - La forma de utilizar los diferentes procesadores por parte del SO puede ser de dos tipos:
 - **Multiproceso simétrico**
 - El sistema operativo utilizará los procesadores por igual
 - Alternará el uso de los procesadores con los que cuenta el sistema de forma simultánea

3. TIPOS DE SISTEMA OPERATIVO

- Número de procesadores
- **Multiproceso asimétrico**
 - El SO reparte las tareas que está realizando a los procesadores
 - Determinados procesos los ejecutará siempre un procesador
 - El otro procesador sólo se utilizará para realizar procesos o programas de usuario
 - Es posible que un procesador esté siempre trabajando y el otro, en ocasiones, sin actividad

3. TIPOS DE SISTEMA OPERATIVO

- Número de procesadores
- **Multiproceso simétrico y asimétrico**
- Mediante estas dos técnicas se consigue es que el tiempo de respuesta de la ejecución de varios programas en un mismo ordenador sea similar para todos

3. TIPOS DE SISTEMA OPERATIVO

- Trabajo en Red
- Centralizados
- En Red
- Distribuidos

4. ESTRUCTURA DE UN S.O.

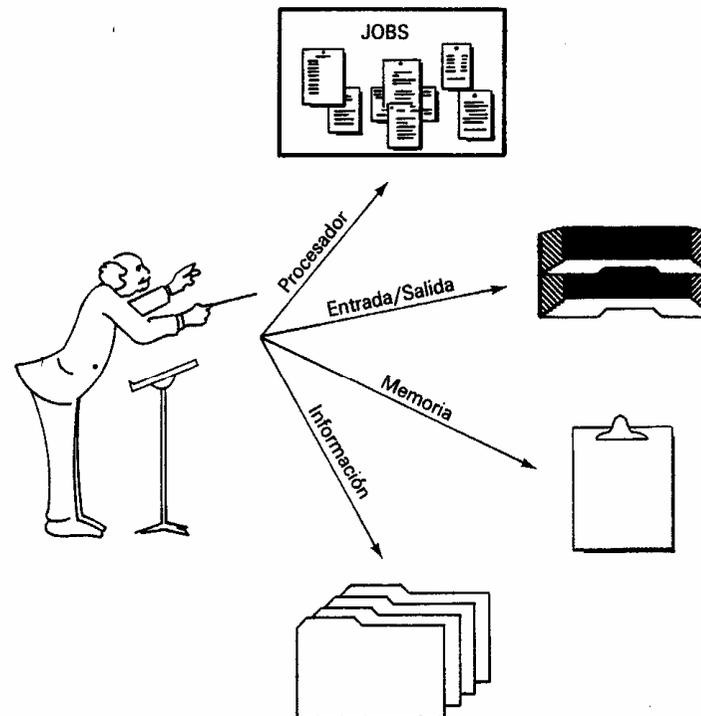
- Sistemas operativos monolíticos
- Sistemas operativos modernos
 - Se organizan en capas o niveles
 - Cada capa o nivel se comunica con su nivel superior e inferior
 - Cada capa se coordina para que el S.O. cumpla con sus funciones

4. ESTRUCTURA DE UN S.O.

Nivel	Descripción
4	<p style="text-align: center;">Usuario</p> <p>Se encarga de controlar los procesos a nivel usuario, es decir, muestra al usuario el proceso que está ejecutando o que quiere ejecutar</p>
3	<p style="text-align: center;">Supervisor</p> <p>Se encarga de realizar la comunicación de cada proceso entre el sistema y el usuario. Controla y coordina la gestión de entrada/salida de los diferentes procesos hacia los periféricos y los dispositivos de almacenamiento externo</p>
2	<p style="text-align: center;">Ejecutivo</p> <p>Sobre este nivel se realiza la administración y la gestión de la memoria. Se encarga de almacenar los procesos en páginas (segmentación, paginación, buffering, etc.), tanto en memoria principal como en disco</p>
1	<p style="text-align: center;">Núcleo</p> <p>Es el que se encarga de controlar la UCP. Gestiona qué procesos llegan al ordenador para ser ejecutados. Son sistemas operativos multiusuario. Este nivel se encarga de realizar tareas básicas del sistema, comunicación con hardware, planificación de procesos, etc</p>

5. FUNCIÓN DEL S.O.

- ¿Qué es un Recurso?
- Tipos fundamentales de recursos:
 - Procesador
 - Memoria principal
 - Periféricos E/S
 - Información



5. FUNCIONES DE UN S.O.

- Distinta gestión según componentes:
 - Realizada por el **planificador de procesos**:
 - Proceso o tarea: programa en ejecución
 - Para ejecutar un proceso ha de residir en memoria y tener asignados los recursos que necesite
 - Cada proceso está protegido del resto de procesos
 - Procesos propios del s.o.: modo kernel o privilegiado
 - Procesos propios de usuario: modo usuario (con restricciones de acceso a los recursos hardware)

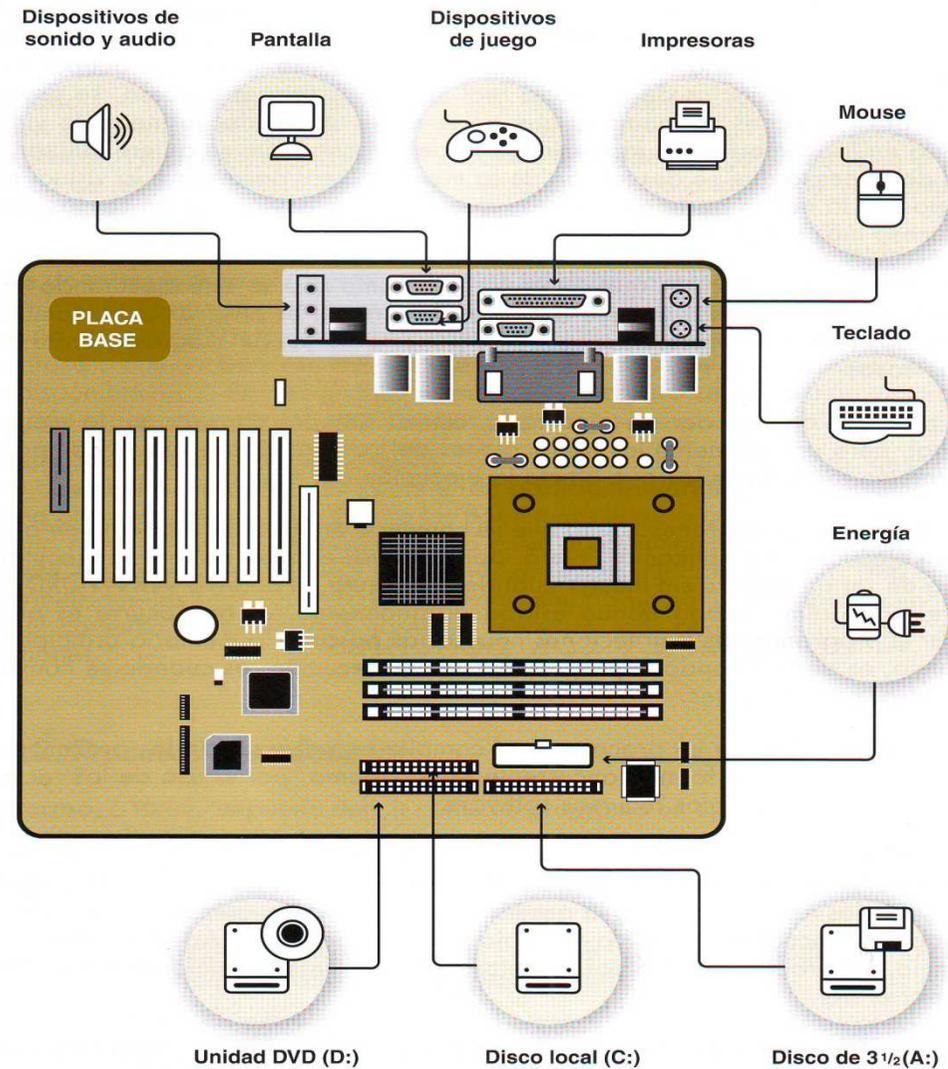
5. FUNCIONES DE UN S.O.

- Distinta gestión según componentes:
 - Memoria: gestionado por el **administrador de memoria**
 - Lleva un registro de las partes de la memoria que se están utilizando y las que no
 - Reserva espacio de memoria para nuevos procesos y libera espacio de los procesos finalizados
 - Protege las zonas de memoria ocupadas de otros procesos
 - Gestiona el intercambio de datos entre memoria y disco
 - Importante en sistemas operativos multiproceso

5. FUNCIONES DE UN SISTEMAS OPERATIVOS

- Distinta gestión según componentes:
 - Periféricos:
 - Alimentación eléctrica regida y coordinada por la UCP
 - Buses de comunicación
 - Permiten a la UCP comunicarse con los elementos de dicho ordenador
 - Comunicación a través del gestor de entrada/salida (**chipset**) ubicado en la placa base del ordenador
 - Interfaces como medio de comunicación entre hardware y software:
 - Interfaz tipo texto
 - Interfaz tipo gráfico

5. FUNCIONES DE UN SISTEMAS OPERATIVOS



5.1. GESTIÓN DE PROCESOS

- Necesita que el S.O. les suministre una serie de recursos
 - Sincronización y asignación de procesos
 - UCP
- Un programa se convierte en un proceso:
 - Se está ejecutando
 - Se ubican en memoria las instrucciones que lo componen y sus datos asociados
 - Se le asocia una estructura de datos: BCP

5.1. GESTIÓN DE PROCESOS

○ BCP:

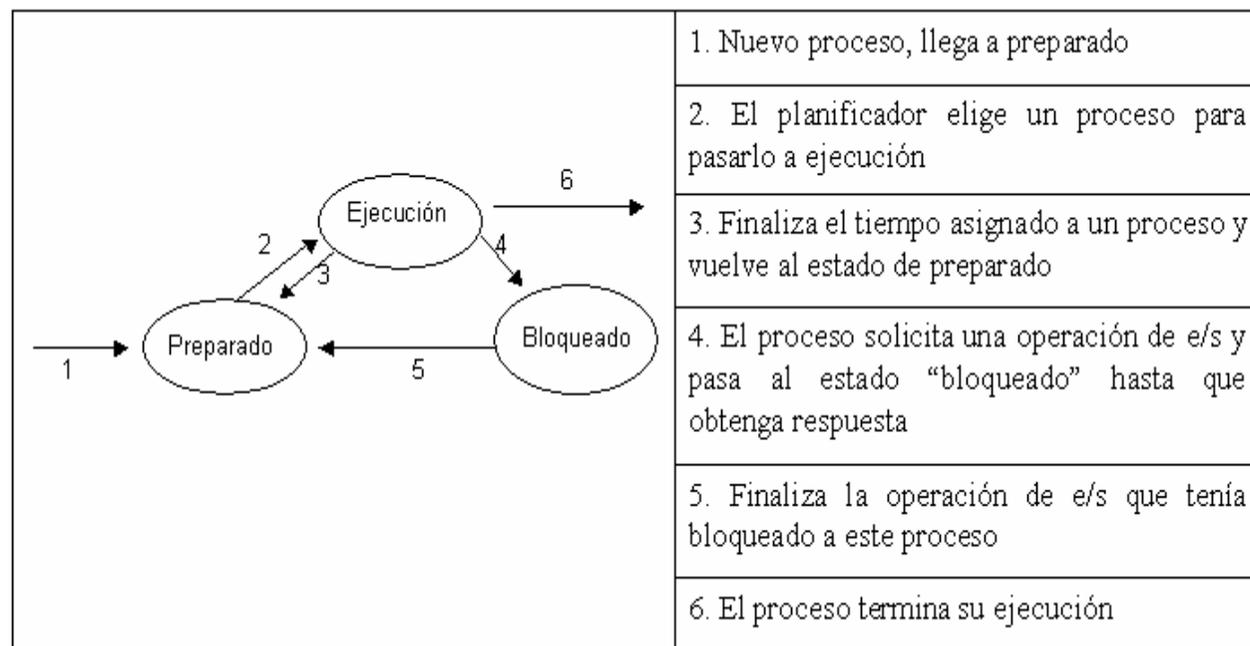
- Estructura de datos única para cada proceso
 - Código de identificador de proceso (PID)
 - Estado actual del proceso
 - Prioridad del proceso
 - Ubicación en memoria de las instrucciones y datos
 - Recursos utilizados

5.1. GESTIÓN DE PROCESOS

- Proceso clásico:
 - Sólo posee un hilo
- Proceso multihilo
 - Más de un hilo
 - Comparten parte de la memoria → Comunicación entre hebras mucho más rápida y eficiente

5.1. GESTIÓN DE PROCESOS

- Existen diferentes estados en los que pueden estar los procesos:
 - En ejecución
 - Preparado, en espera o activo
 - Bloqueado



5.1. GESTIÓN DE PROCESOS

- Cambio de contexto
 - Un proceso pasa de un estado a otro
- Necesidad de planificación para realizar el cambio de contexto:
 - Planificador
 - Algoritmos de planificación:
 - Expropiativo
 - No expropiativo

5.1. GESTIÓN DE PROCESOS

- Durante su ejecución, un proceso compite con el resto de los procesos del sistema por el uso de los recursos
- El reparto de estos entre los procesos y su ejecución concurrente se denomina multiprogramación
- Toda la información de un proceso que el S.O. necesita para controlar se mantiene en el bloque de control de proceso o BCP
- En sistemas operativos multiproceso, el S.O. mantiene listas de bloques de control de procesos para cada uno de los estados del sistema
- Se denomina planificador a aquella parte del S.O. que se encarga de asignar los recursos del sistema de manera que se consigan los objetivos de comportamiento especificados

5.1. GESTIÓN DE PROCESOS

- Procesos se deben ejecutar en el orden adecuado y sin interferencias entre ellos
- Control del uso de recursos
 - No se pueden utilizar por más de un proceso en el mismo instante de tiempo
- Los procesos cargados en un sistema cooperan
 - Espacios comunes de memoria, recursos, hardware, etc.
 - Deben ejecutarse en el orden adecuado, el espacio de memoria adecuado y sin alterar los otros procesos
- El S.O. para o lanza un proceso utilizando técnicas de bloqueo específicas, como los semáforos

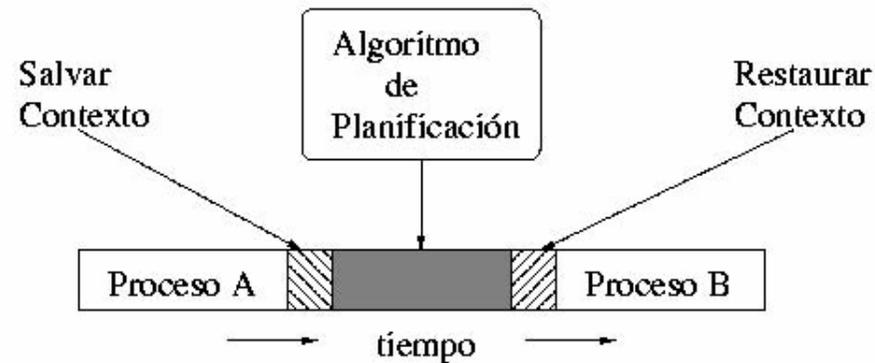
5.1. GESTIÓN DE PROCESOS. ALGORITMOS DE PLANIFICACIÓN

- Cómo y cuando debe ejecutarse un proceso
- Deben ser imparciales, equitativos, eficientes y tener buen tiempo de respuesta y rendimiento
- Entre los diferentes criterios que existen para comparar los algoritmos tenemos:
 - **Eficacia**
 - **Productividad**
 - **Tiempo de Retorno**
 - **Tiempo de Espera**
 - **Tiempo de Respuesta**
- El mejor algoritmo será el que maximiza el uso del procesador y la productividad y minimiza los tres tiempos

5.1. GESTIÓN DE PROCESOS. ALGORITMOS DE PLANIFICACIÓN

○ Algoritmos de Planificación:

- Algoritmo FIFO
- Algoritmo SJF
- Algoritmo SRT
- Planificación por prioridades
- Round Robin
- Colas Multinivel



5.1.1. PLANIFICACIÓN FIFO

- El más sencillo
- El primero que pide la CPU es el primero al que se le asigna
- Inconveniente: tiempo medio de espera largo
- Tiempo medio de espera dependiente del orden de llegada

5.1.1. PLANIFICACIÓN FIFO

○ Ejemplo:

- Cinco procesos A, B, C, D y E
- Los procesos llegan en los tiempos 0, 1, 4, 5 y 8 respectivamente
- Cada uno de ellos necesita respectivamente un tiempo de ejecución de 3, 5, 2, 6 y 4 unidades de tiempo
- Obtener tiempo de respuesta t_f , tiempo de retorno T , tiempo de espera E e índice de servicio I

5.1.1. PLANIFICACIÓN FIFO

○ Ejemplo:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A		█	█	█																	
B			█	█	█	█	█	█													
C						█	█	█	█	█	█										
D							█	█	█	█	█	█	█	█	█	█					
E										█	█	█	█	█	█	█	█	█	█	█	█

5.1.1. PLANIFICACIÓN FIFO

○ Ejemplo:

Proceso	Instante llegada	Duración	t_f	T	E	I
A	0	3	3	3	0	1.00
B	1	5	8	7	2	0.71
C	4	2	10	6	4	0.33
D	5	6	16	11	5	0.54
E	8	4	20	12	8	0.33

5.1.2. PLANIFICACIÓN SJF(SHORTEST JOB FIRST)

- Los procesos se eligen en función del tiempo estimado en que terminarán (su duración total)
- Si dos procesos tienen el mismo hay que desempatar, usando FCFS
- Es óptimo en minimizar tiempo de espera
- Problema: averiguar la duración del uso de CPU de cada proceso

5.1.2. PLANIFICACIÓN SJF(SHORTEST JOB FIRST)

○ Ejemplo:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A		█	█	█																	
B			█	█	█	█	█	█													
C						█	█	█	█	█											
D							█	█	█	█	█	█	█	█		█	█	█	█	█	█
E										█	█	█	█								

5.1.2. PLANIFICACIÓN SJF(SHORTEST JOB FIRST)

○ Ejemplo:

Proceso	Instante llegada	Duración	t_f	T	E	I
A	0	3	3	3	0	1.00
B	1	5	8	7	2	0.71
C	4	2	10	6	4	0.33
D	5	6	20	15	9	0.40
E	8	4	14	6	2	0.67

5.1.3. PLANIFICACIÓN SRT (SHORTEST REMAINING TIME)

- Asigna la CPU al proceso al que menos tiempo le resta para concluir
- Si dos procesos tienen el mismo, FCFS
- Si durante la ejecución de un proceso llega otro al que le quede menos tiempo para acabar, se le cede la CPU (expropiación)
- Es óptimo en minimizar tiempo de espera
- Problema: averiguar la duración del siguiente uso de CPU de cada proceso

5.1.3. PLANIFICACIÓN SRT (SHORTEST REMAINING TIME)

○ Ejemplo:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A		█	█	█																	
B			█	█	█	█	█	█	█	█											
C						█	█														
D							█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
E											█	█	█	█							

5.1.3. PLANIFICACIÓN SRT (SHORTEST REMAINING TIME)

○ Ejemplo:

Proceso	Instante llegada	Duración	t_f	T	E	I
A	0	3	3	3	0	1.00
B	1	5	10	9	4	0.55
C	4	2	6	2	0	1.00
D	5	6	20	15	9	0.40
E	8	4	14	6	2	0.67

5.1.4. PLANIFICACIÓN POR PRIORIDADES

- A cada proceso se le asigna una prioridad
- La CPU se asigna al proceso más prioritario
- Con o sin expropiación
- SJF es un caso particular
- La prioridad es un número en cierto rango
- Favorece a los procesos más prioritarios
- Convenio: prioridad más alta = número más pequeño
 - Ejemplo: 1 es la mayor prioridad

5.1.4. PLANIFICACIÓN POR PRIORIDADES

- Criterios para definir prioridades:
 - Internamente (por el S.O.): tiempo de CPU (SJF), uso de memoria, relación entre período de CPU y de E/S, etc.
 - Externamente: importancia del proceso, importancia del usuario en la organización, dinero pagado por el usuario

5.1.4. PLANIFICACIÓN POR PRIORIDADES

- Las prioridades asignadas pueden ser:
 - Estáticas
 - Dinámicas
- El algoritmo de planificación por prioridades puede ser:
 - Expropiativo
 - No expropiativo
- Problema de Inanición:
 - Un proceso con baja prioridad en un sistema muy cargado puede no ejecutarse nunca
 - Una solución: incrementar la prioridad de un proceso según el tiempo de espera

5.1.4. PLANIFICACIÓN POR PRIORIDADES NO EXPROPIATIVO

○ Ejemplo:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A(2)																					
B(1)																					
C(2)																					
D(0)																					
E(1)																					

5.1.4. PLANIFICACIÓN POR PRIORIDADES NO EXPROPIATIVO

○ Ejemplo:

Proceso	Instante llegada	Duración	t_f	T	E	I
A	0	3	3	3	0	1
B	1	5	8	7	2	0.71
C	4	2	20	16	14	0.13
D	5	6	14	9	3	0.66
E	8	4	18	10	6	0.40

5.1.4. PLANIFICACIÓN POR PRIORIDADES EXPROPIATIVO

○ Ejemplo:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A(2)		█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█		
B(1)		█	█	█	█	█	█	█	█	█	█	█	█								
C(2)																				█	█
D(0)																					
E(1)																					

5.1.4. PLANIFICACIÓN POR PRIORIDADES EXPROPIATIVO

○ Ejemplo:

Proceso	Instante llegada	Duración	t_f	T	E	I
A	0	3	18	18	15	0.17
B	1	5	12	11	6	0.45
C	4	2	20	16	14	0.13
D	5	6	11	6	0	1.00
E	8	4	16	8	4	0.50

5.1.5. PLANIFICACIÓN ROUND ROBIN

- A cada proceso se le asigna el uso de la CPU una cantidad fija de tiempo llamada **Quantum**
 - Si q es grande, entonces como FCFS
 - Si q es pequeño, como si cada proceso tuviese un procesador n veces más lento
- Cola circular: el orden de los procesos según su llegada a la cola de preparados, no al sistema
- Utiliza expropiación para rotar el proceso que se está ejecutando.
- Utiliza un temporizador

5.1.5. PLANIFICACIÓN ROUND ROBIN

- Ejemplo (quantum = 1):

Nota: El proceso que sale se sitúa al final de la cola.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A		Red	Grey	Red	Grey	Red															
B			Green	Grey	Green	Grey	Grey	Green	Grey	Grey	Green	Grey	Grey	Green							
C					Grey	Brown	Grey	Grey	Brown												
D						Grey	Grey	Purple	Grey	Grey	Grey	Purple	Grey	Grey	Purple	Grey	Purple	Grey	Purple	Purple	Purple
E										Grey	Grey	Pink	Grey	Grey	Pink	Grey	Pink	Grey	Pink		

5.1.5. PLANIFICACIÓN ROUND ROBIN

- Ejemplo (quantum = 1):

Proceso	Instante llegada	Duración	t_f	T	E	I
A	0	3	5	5	2	0.60
B	1	5	13	12	7	0.42
C	4	2	9	5	3	0.40
D	5	6	20	15	9	0.40
E	8	4	18	10	6	0.40

5.1.5. PLANIFICACIÓN ROUND ROBIN

- Ejemplo (quantum = 3):

Nota: El proceso que sale se sitúa al final de la cola.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A																					
B																					
C																					
D																					
E																					

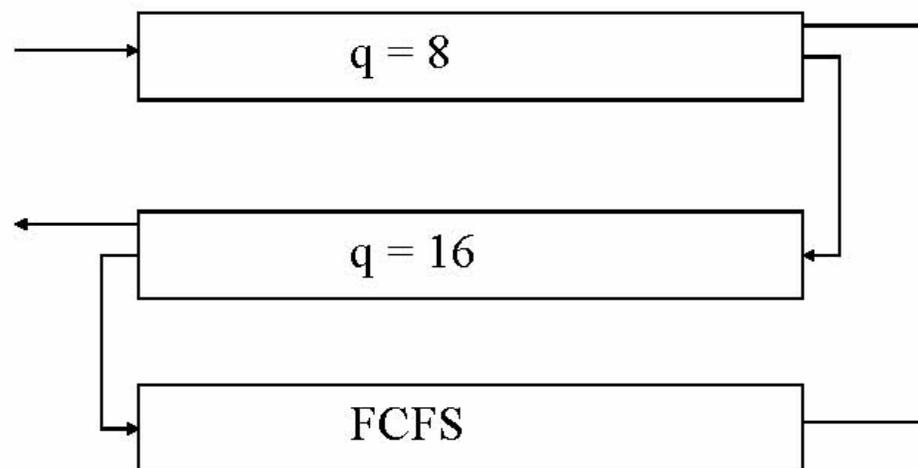
5.1.5. PLANIFICACIÓN ROUND ROBIN

- Ejemplo (quantum = 3):

Proceso	Instante llegada	Duración	t_f	T	E	I
A	0	3	3	3	0	1.00
B	1	5	13	12	7	0.42
C	4	2	8	4	2	0.50
D	5	6	19	14	8	0.43
E	8	4	20	12	8	0.33

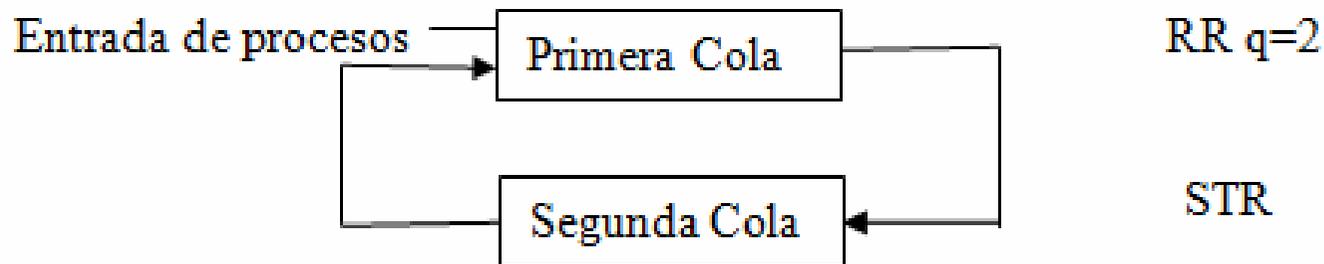
5.1.6. MULTICOLAS

- Cada cola: un algoritmo de planificación
- Algoritmo entre colas:
 - Realimentadas
 - Sin realimentación



5.1.6. MULTICOLAS

○ Ejemplo:



- Obtener tiempo de respuesta t_f , tiempo de retorno T , tiempo de espera E e índice de servicio I , y tiempos medios de T y E
- El paso de un proceso de una cola a otra se produce cuando éste se ha ejecutado durante 1 Quantum de tiempo en la cola correspondiente
- Un nuevo proceso se coloca al final de la cola (delante del proceso que acaba de hacer uso de la CPU)

5.1.6. MULTICOLAS

- Ejemplo:

Proceso	Instante de llegada	Duración
A	0	4
B	1	4
C	2	4
D	7	1
E	8	3
F	12	2

5.1.6. MULTICOLAS

○ Ejemplo:

Proceso	Instante llegada	Duración	t_f	T	E	I
A	0	4	9	9	5	0,44
B	1	4	16	15	11	0,27
C	2	4	18	16	12	0,25
D	7	1	8	1	0	1
E	8	3	12	4	1	0,75
F	12	2	14	2	0	1

5.1.6. MULTICOLAS

○ Ejemplo (Desempate FIFO):

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A		█	█	█	█	█	█	█	█	█									
B			█	█	█	█	█	█	█	█	█	█	█	█	█	█	█		
C				█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
D								█											
E									█	█	█	█							
F													█	█					

Cola 1		B	B C	C	C			D A	A E	E			F						
Cola 2			A	A	A B	A B	A B C	B C	B C	B C	B C	B C E	B C	B C	B C	C	G		

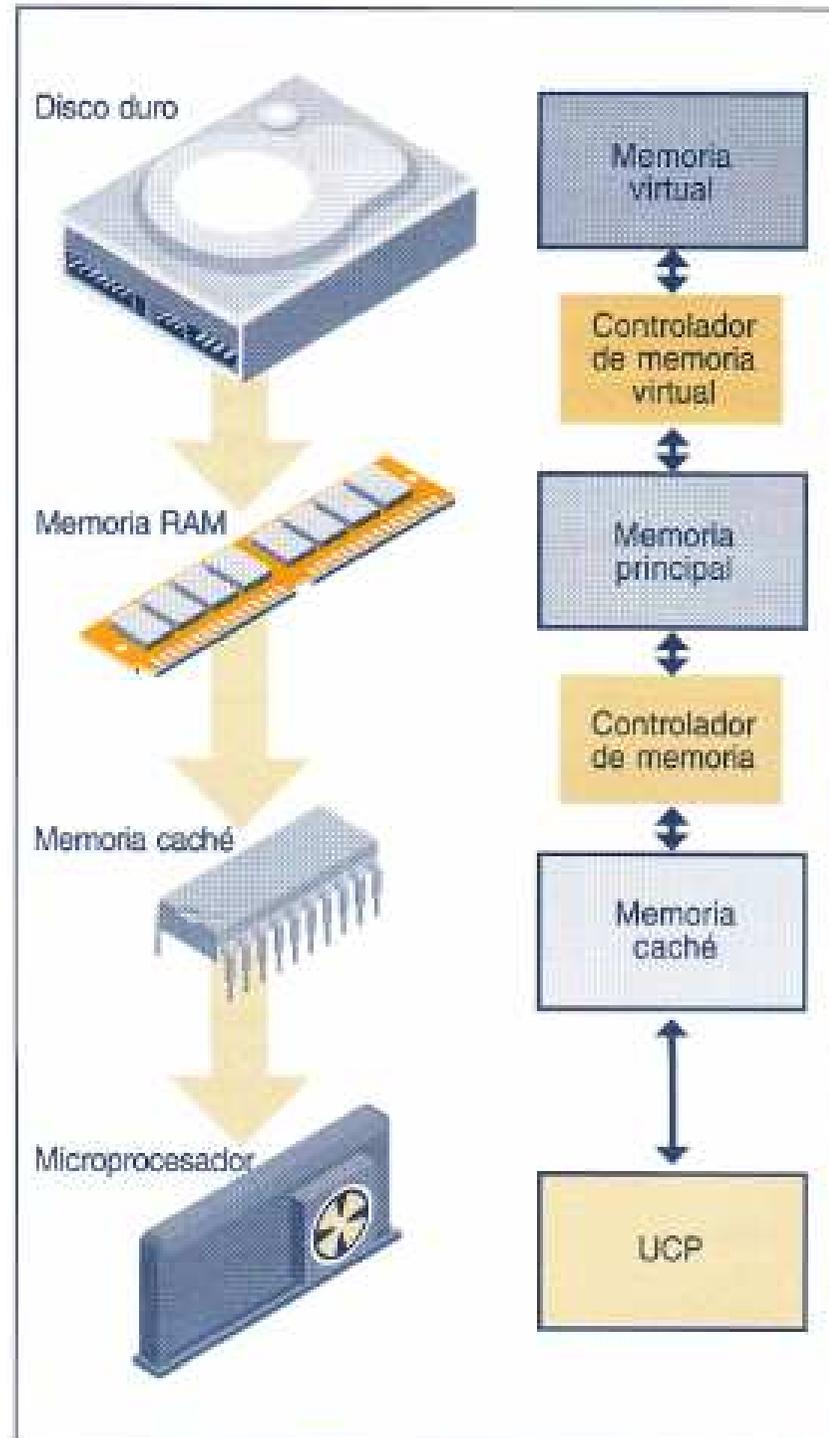
5.2. GESTIÓN DE MEMORIA

- Para poder ejecutar un programa en un ordenador, es necesario que esté cargado en memoria principal
- En los primeros sistemas informáticos:
 - Toda la memoria se dedicaba al único proceso en ejecución (sin contar la parte reservada para el S.O.)
- SSOO actuales:
 - Realizan una gestión de la memoria eficiente
 - No es necesario tener en memoria todas las instrucciones de un proceso

5.2.1. MEMORIA RAM Y MEMORIA VIRTUAL

- La memoria es limitada -> esto produce problemas
- Para solucionarlo:
 - Dividir el programa en partes denominadas capas: las capas se iban ejecutando según fuera necesario
 - En primer lugar, se pasa del disco duro (o soporte de almacenamiento) a memoria la capa 0
 - Cuando sea necesario utilizar otra parte del programa que no este en la memoria principal se accede de nuevo al disco para cargar la siguiente capa

5.2.1. MEMORIA RAM Y MEMORIA VIRTUAL



5.2.1. MEMORIA RAM Y MEMORIA VIRTUAL

○ Memoria virtual:

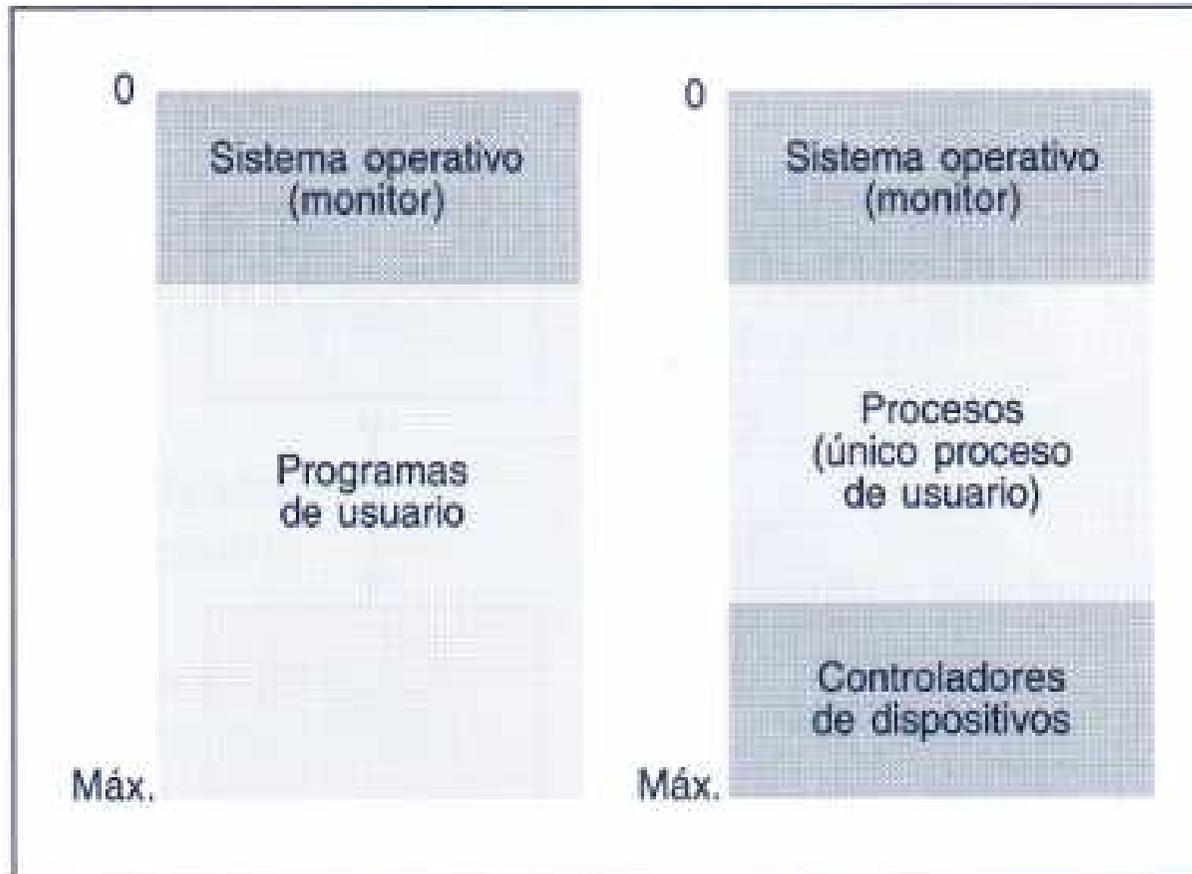
- El programa que se vaya a ubicar en memoria puede ser excesivamente grande para el tamaño de ésta
 - Permanece en memoria la parte del programa que se este ejecutando
 - El resto estará en el disco
- Esta técnica se aplica en la mayoría de los sistemas operativos actuales
- Considera el espacio en disco libre como si se tratase de memoria RAM

5.2.2. INTERCAMBIO

- Gestión de memoria inicial:
 - S.O. monotarea:
 - Memorias de poca capacidad
 - Se reservaba una parte de ella para el sistema operativo
 - El resto era una zona para la ejecución de programas de usuario. Al tratarse de un sistema operativo monotarea, sólo podía haber un proceso en cada momento
 - En el caso del sistema operativo MS-DOS, además se reservaba una tercera zona en la parte alta de la memoria, para ubicar los controladores de dispositivos

5.2.2. INTERCAMBIO

- Gestión de memoria inicial:



5.2.2. INTERCAMBIO

- Función del administrador de memoria
 - Llevar en un registro las partes de memoria que se están utilizando y las que no
 - Se reservará espacio de memoria para los nuevos procesos
 - Liberará el espacio de los procesos que han finalizado
 - Se encarga de gestionar el intercambio de datos entre memoria y disco
 - Siempre y cuando los procesos sean tan grandes que no quepan de una sola vez en memoria

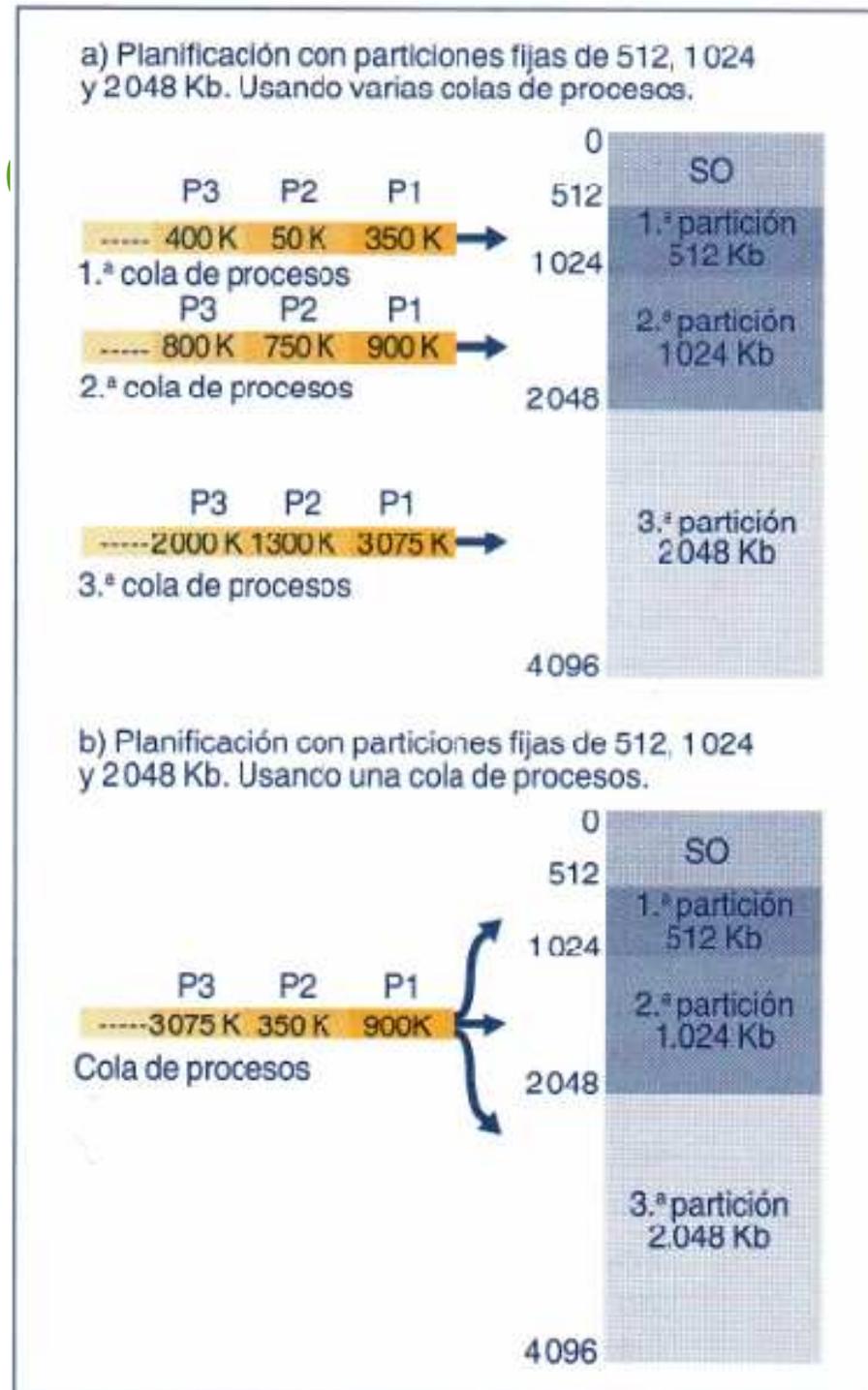
5.2.2. INTERCAMBIO

- La gestión de memoria es complicada en los sistemas multitarea
 - Se necesita disponer de varios procesos residentes simultáneamente en memoria
- Intercambio:
 - Los procesos en espera pueden ser llevados al disco y dejar libre la parte de memoria que ocupan para que otros procesos entren en ejecución
 - Los procesos se pueden cargar siempre en la misma posición de memoria o reubicarse en otra. La reubicación puede ser estática o dinámica

5.2.2. INTERCAMBIO

- Primera opción:
 - Dividir la memoria en particiones fijas
 - Cola de los procesos que solicitan entrar en memoria
 - El planificador tiene en cuenta los requerimientos de memoria de cada uno de los procesos y las particiones de memoria disponibles
- Segunda opción:
 - Cada partición tenga asociada una cola de tareas

5.2.2. INTERCAMBIO



5.2.2. INTERCAMBIO

- Dificultad en el diseño con las particiones fijas
 - Selección de los tamaños de las particiones
 - Desaprovechamiento o fragmentación de la memoria
 - Esta fragmentación puede ser :
 - Interna, que afecta a la parte de la memoria que no se está usando pero que es interna a una partición asignada a un proceso
 - Externa, que ocurre cuando una partición disponible no se emplea porque es muy pequeña para cualquiera de los procesos que esperan
 - Con un conjunto dinámico de procesos ejecutándose, no es posible encontrar las particiones de memoria adecuadas. La opción es disponer de particiones variables

5.2.2. INTERCAMBIO

○ Problema:

- Se necesita un registro con información de las particiones libres y ocupadas
- Se siguen presentando problemas de fragmentación externa

○ Solución:

- Permitir que los procesos puedan utilizar memoria no contigua, lo que se consigue mediante técnicas de **paginación**
- Hay un mecanismo de traducción de las direcciones lógicas a las físicas mediante una tabla de páginas. Hay que tener en cuenta: el tamaño de la tabla y el tiempo de asignación

5.2.2. INTERCAMBIO

- Alternativa:
 - En contraposición al concepto memoria como un array o lista unidimensional, se halla la **segmentación**
 - El espacio de direcciones lógicas es un conjunto de segmentos con diferentes nombres y tamaños, sin ninguna ordenación entre ellos
 - No se produce fragmentación interna, pero sí externa, que ocurre cuando todos los bloques de memoria libres son demasiado pequeños para acomodar a un trozo o bloque de proceso
- Aunque la segmentación y la paginación son esquemas diferentes de gestión de la memoria, se pueden considerar estrategias combinadas

5.2.3. PAGINACIÓN

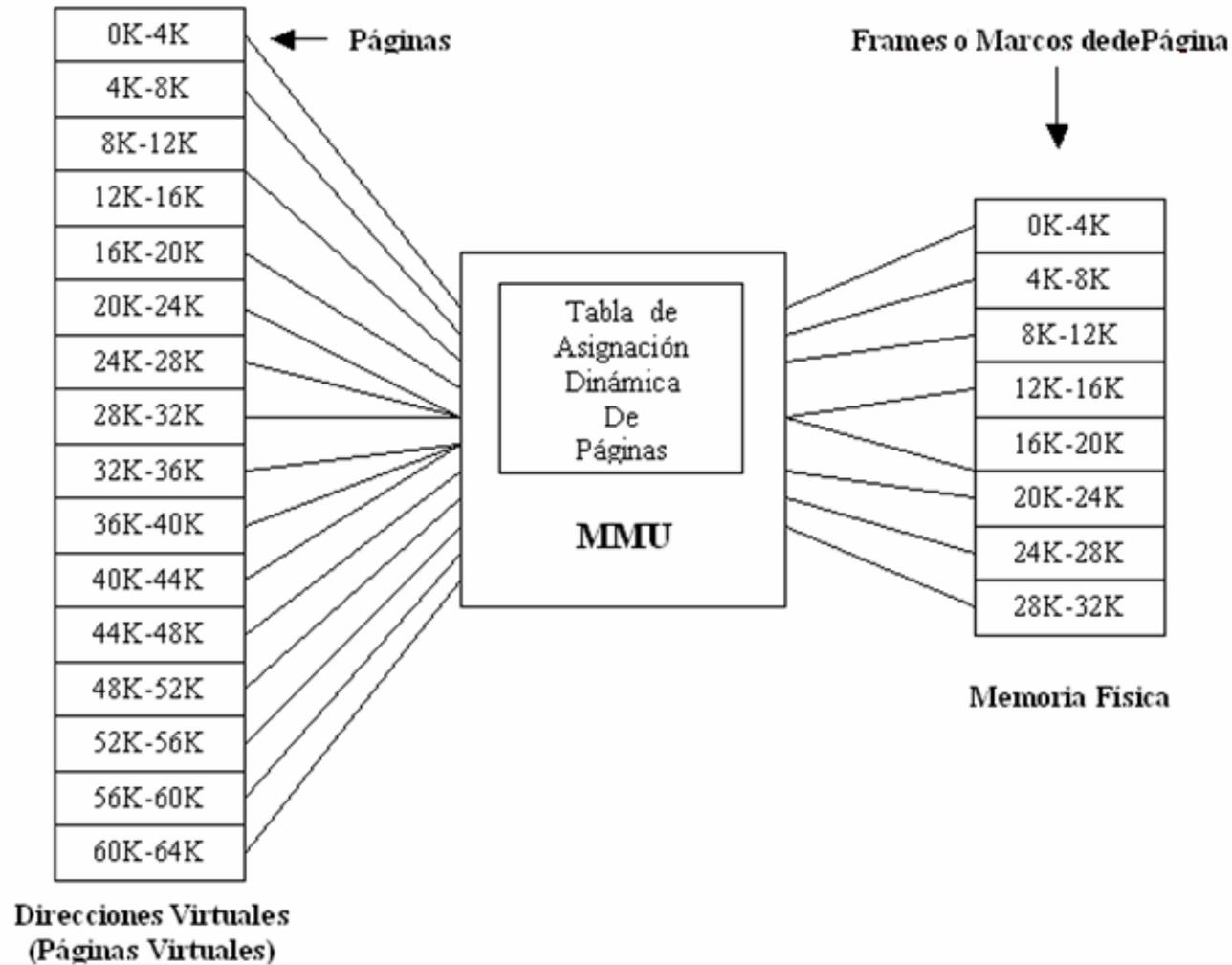
- Técnica que consiste en dividir la memoria interna o RAM en zonas iguales, llamadas *frames* o marcos, y los programas en partes de idéntico tamaño, denominadas páginas
- Para ubicar un programa en memoria, el sistema operativo buscará en la memoria física los marcos que tenga libres
- El tamaño de estos marcos se diseña mediante hardware

5.2.3. PAGINACIÓN

- Si utilizamos un sistema de multiprogramación y sólo hay un trabajo, éste tendrá asignados todos los marcos necesarios para él
- Mediante la tabla de páginas, la UCP asigna las direcciones físicas de los marcos a las páginas en las que se ha dividido el programa
- La asignación de los marcos no tiene que ser necesariamente consecutiva

5.2.3. PAGINACIÓN

En el ejemplo siguiente disponemos de 64K de direcciones virtuales, pero solo de 32K de memoria física. Las páginas son de tamaño de 4K:



5.2.3. PAGINACIÓN

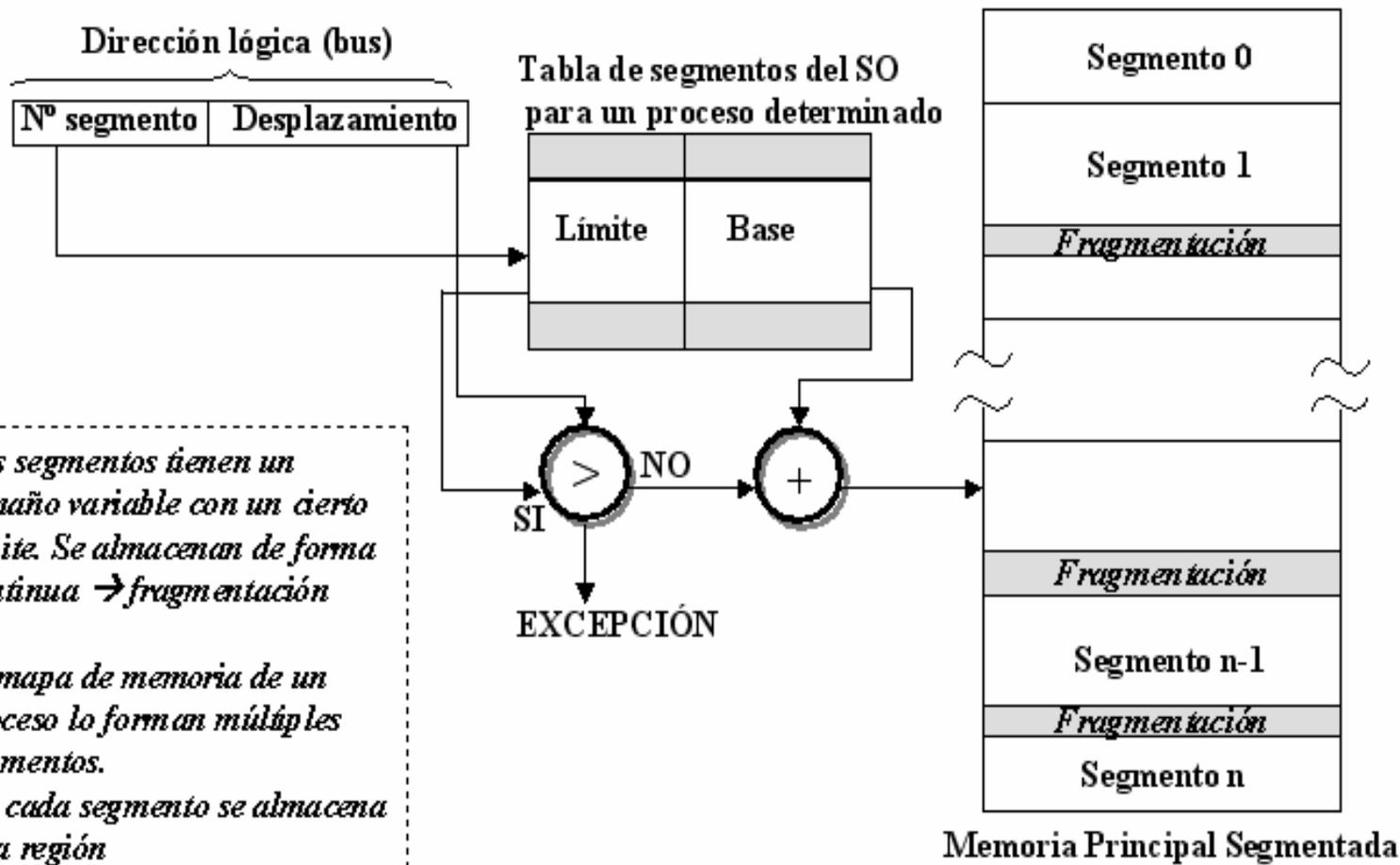
- En resumen:
 - La paginación es una técnica de reasignación o redireccionamiento dinámico, con la consideración de que la tabla de páginas se puede almacenar en registros especiales destinados a tal efecto o en una parte de la propia memoria
- La transformación de las direcciones lógicas en físicas la realiza la unidad de administración de memoria ***Management Memory United (MMU)***

5.2.4. SEGMENTACIÓN

- Técnica similar a la paginación :
 - Permite definir los bloques de memoria, en este caso, de tamaño variable
 - Cada segmento puede variar desde 0 hasta un máximo permitido
 - Estos segmentos pueden tener longitudes distintas
 - La longitud de un segmento puede variar según las necesidades del programa

5.2.4. SEGMENTACIÓN

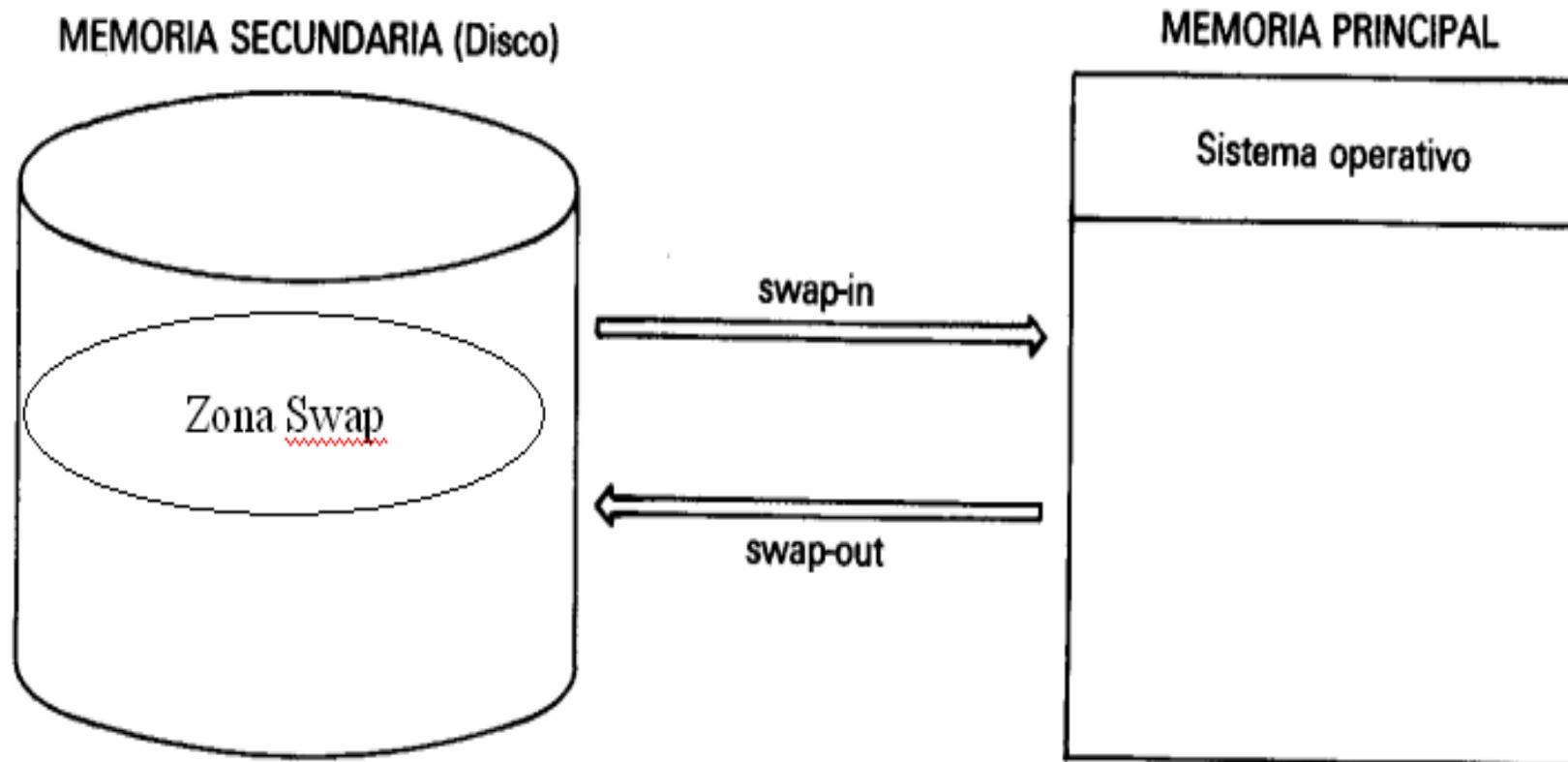
- Esquema de traducción de la segmentación:



5.2.5. SWAPPING

- Técnica similar a la de memoria virtual
- Cuando varios usuarios están ejecutando procesos en un mismo ordenador, éste se ve obligado a cargarlos en RAM
- Según el estado en el que se encuentre el proceso de cada usuario, la memoria se irá liberando de su proceso, y pasará a la zona de **swap** mediante la técnica llamada **swap-out**
- Si el usuario vuelve a solicitar su proceso para seguir ejecutándolo, se produce el denominado **swap-in**, que consiste en pasar el programa de la zona de swap a la memoria interna

5.2.5. SWAPPING



5.2.5. SWAPPING

- Esta zona de swap se suele utilizar en sistemas operativos como UNIX y Linux
- Está formada por un espacio físico del disco en el que tenemos el sistema operativo y las aplicaciones que se van a ejecutar
- Los fabricantes de estos sistemas operativos recomiendan que esta zona sea del 20%, aproximadamente, del espacio en disco o el doble de la capacidad de RAM del ordenador

5.2.6. PROGRAMAS REUBICABLES, REENTRANTES, RESIDENTES Y REUTILIZABLES

- Procesos reubicables
- Procesos reentrantes
- Procesos residentes
- Procesos reutilizables

5.3. GESTIÓN DE E/S

- Una de las funciones principales de un sistema operativo es el control de los periféricos de entrada/salida del ordenador.
- Periférico:
 - Dispositivo electromecánico, electromagnético o electrónico que permiten la comunicación directa con el exterior, permitiendo la entrada y/o salida de datos hacia o desde el mismo

5.3. GESTIÓN DE E/S

- Cada periférico está compuesto por un componente mecánico y por otro u otros componentes electrónicos
- El sistema operativo accede a la información de la memoria principal, la extrae en forma de impulsos eléctricos y la envía a los diferentes dispositivos periféricos

5.3. GESTIÓN DE E/S

- Una vez instalado el periférico el SO se encarga de:
 - Eliminar los errores
 - Determinar qué dispositivo requiere atención
 - Enviar órdenes a los dispositivos
- Eliminar los posibles errores, por ejemplo, durante la impresión de documentos, durante la escritura/lectura desde CDs, pendrives, etc.
- Determinar qué dispositivo requiere atención
 - El dispositivo envía una petición de interrupción (IRQ) para solicitar la atención del SO. Al teclado según se introducen los caracteres, al ratón al hacer clic...

5.3. GESTIÓN DE E/S. INTERRUPCIONES

- Interrupción: señal de llamada generada por un dispositivo cuando necesita ser atendido por la CPU
- Vector de interrupción: IRQ: Interrupt ReQuest
 - Canal de comunicación única para hacer llegar las peticiones de interrupción a la CPU
 - Los PCs suelen contar con al menos 20 IRQs, que en principio suponen 20 posibles vías diferentes de que la CPU reciba interrupciones
 - Tratar en profundidad el tema de las interrupciones resultaría demasiado complejo, por lo que no nos adentraremos en su estudio en este modulo. En cualquier caso puede encontrarse más información de interés en diversas páginas de internet y en la presentación [Interrupcion.pps](#) que se incluye en la unidad

5.3. GESTIÓN DE E/S. INTERRUPCIONES

- En los sistemas por interrupciones una vez iniciado el tratamiento de una interrupción, la rutina de servicio suele ejecutarse hasta el final. Las rutinas de servicio suelen ser cortas y provocando un retraso aceptable para la mayoría de dispositivos
- A pesar de ello, en algunos sistemas es necesario que los dispositivos de E/S se organicen en unos niveles de prioridad
- De esta manera una interrupción con prioridad alta podrá ser aceptada por la CPU, incluso cuando esté atendiendo una de un dispositivo de menor prioridad

5.3. GESTIÓN DE E/S. DMA

- Acceso Directo a Memoria
- Usado por ciertos periféricos que quieren transferir una gran cantidad de información
- Se suprime el acceso mediante IRQ y direcciones de E/S
- La transferencia se realiza a través de unas líneas llamadas DRQ (DMA Request):
 - Acceso a memoria más rápido
 - Se deja libre el procesador para otras tareas
- Utilizado por HD, tarjetas gráficas y de sonido.

5.3. GESTIÓN DE E/S. DMA

- Error en un periférico al usar DMA:
 - S.O. o usuario desactivan DMA
 - Activación de modo PIO:
 - Funcionamiento del periférico más lento
- Modo Ultra-DMA:
 - DMA de alta velocidad

5.3. GESTIÓN DE E/S. DMA

- Técnicas para mejorar el rendimiento del sistema:
 - Caching
 - Buffering
 - Spooling

5.4. GESTIÓN DE ARCHIVOS

- Datos:
 - programas, documentos, imágenes, etc. que se almacenan de forma permanente en memoria secundaria
- Los datos se almacenan en unas estructuras llamadas ficheros o archivos

5.4. GESTIÓN DE ARCHIVOS. FICHEROS

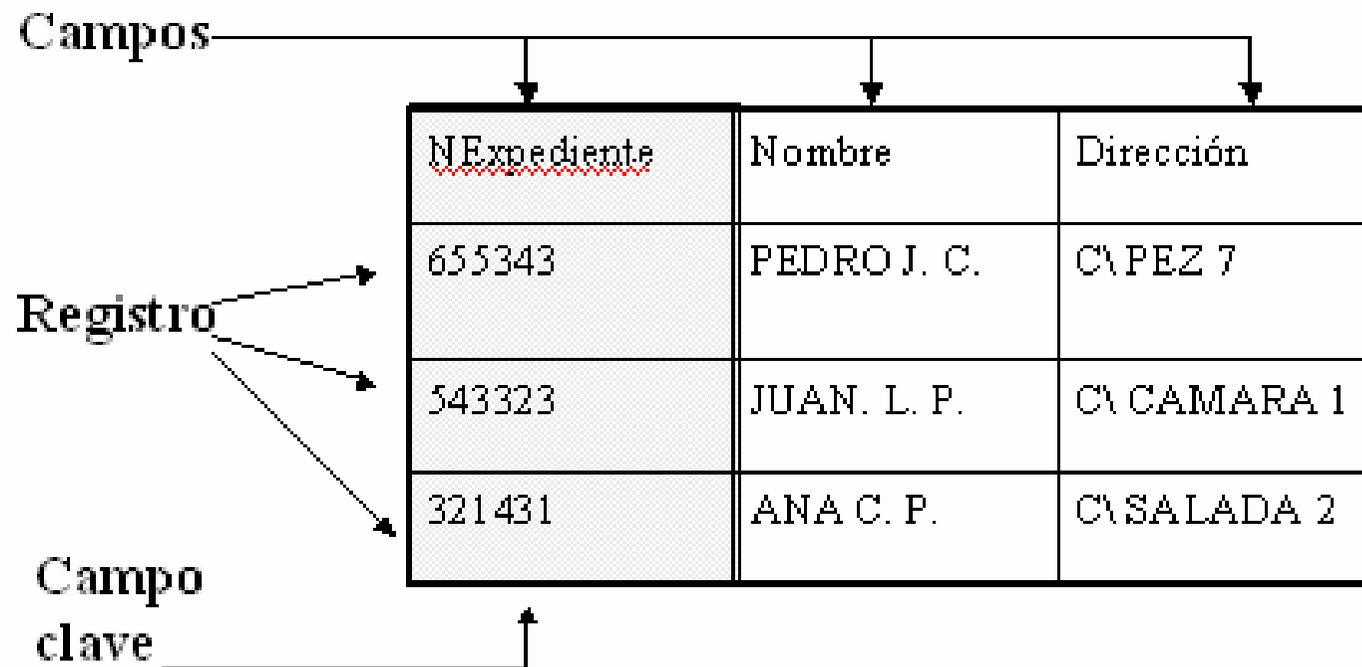
- Fichero: colección organizada de datos que se guarda de forma permanente en dispositivos de almacenamiento secundario (discos duros, CDs,...)
- Los ficheros almacenan los datos en forma de registros
- Registro: agrupación lógica de datos
 - Se compone de campos de información
 - Ejemplo: Fichero alumno

Número de expediente → campo1

Nombre → campo2

Dirección → campo3

5.4. GESTIÓN DE ARCHIVOS. FICHEROS



5.4. GESTIÓN DE ARCHIVOS. FICHEROS

- Cuando hablamos de ficheros hay que tener claros varios conceptos:
 - Registro físico
 - cantidad de información que se transfiere físicamente desde un soporte de almacenamiento externo a la memoria interna de ordenador en una operación de entrada o de salida
 - Clave
 - campo o conjunto de campos que identifican unívocamente un registro dentro del fichero (en el fichero alumnos, el número de expediente)

5.4. GESTIÓN DE ARCHIVOS. FICHEROS

- Cada SO nombra a los archivos a su manera:
 - MS-DOS nombres de 8 caracteres como máximo.
 - Unix distingue entre mayúsculas y minúsculas: PEPE.DOC y pepe.doc son ficheros diferentes.
 - Windows no distingue entre mayúsculas y minúsculas: PEPE.DOC y pepe.doc serían el mismo fichero.
- Además del nombre, la mayoría de los SSOO añade una extensión al nombre, para indicar el tipo de contenido.
 - Ejecutables (.exe, .out, .com...)
 - Fuentes (.h, .c, .cpp, .ada,...)
 - Objetos (.obj, .o)
 - Datos (.txt, .doc, ...)

5.4. GESTIÓN DE ARCHIVOS. FICHEROS

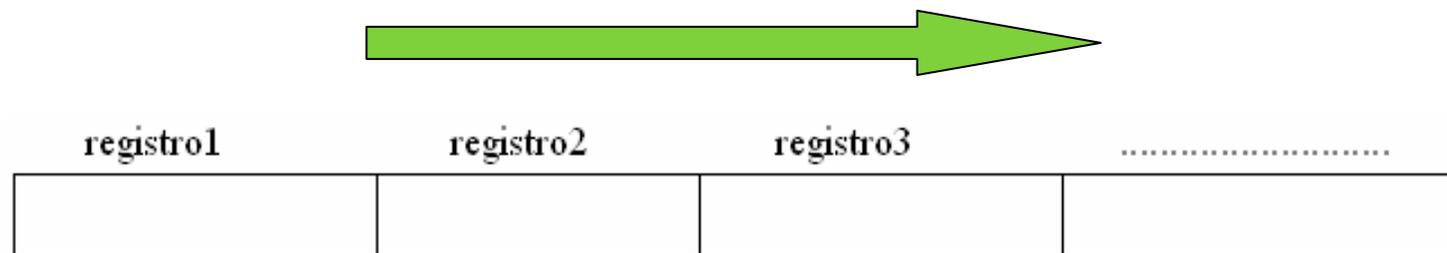
- Desde el punto de vista del SO se distinguen los siguientes tipos de ficheros:
 - Ordinarios
 - Directorios
 - Especiales o de dispositivos

5.4. GESTIÓN DE ARCHIVOS. FICHEROS

- El acceso a un fichero
 - Define el procedimiento a seguir para acceder a los datos que almacenan
 - Está condicionado por el tipo de soporte donde ésta se almacena
- Existen distintos tipos de acceso:
 - Secuencial
 - Directo
 - Indexado

5.4. GESTIÓN DE ARCHIVOS.. ACCESO SECUENCIAL

- Para acceder a un registro, hay que pasar por todos los demás:
- Este tipo de ficheros se almacena en soportes secuenciales (como las cintas de vídeo)



```
-----  
+- a  b  c  d  ...  
|-----  
+> cabeza lectora.
```

Para leer el registro 'd', la cabeza lectora, deberá pasar antes por los que le preceden.

5.4. GESTIÓN DE ARCHIVOS. ACCESO DIRECTO

- Se puede acceder a un registro directamente, sin pasar por los demás.
- Se accede al registro concreto a través de la posición del registro dentro del fichero.
- El acceso es más rápido que en el modo secuencial.
- Este tipo de ficheros se almacena en soportes de acceso directo, como los CDs, DVDs...

Reg	D A T O
1	LUIS
2	CARLOS
3	TERESA
4	JOAQUIN
5	INMA
6	JOSE

Fíjese que los datos están almacenados en el orden en el que han sido introducidos por el usuario.

Accedemos a los datos por medio del valor de la posición del registro.

5.4. GESTIÓN DE ARCHIVOS. ACCESO INDEXADO

- Para acceder a un registro, primero se consulta una tabla de índices
- La tabla de índices contiene por cada registro dentro del fichero:
 - clave que identifica al registro (por ejemplo NumExpediente, ...)
 - dirección relativa
- Para realizar el acceso directo al registro buscado bastará con buscar la clave en la tabla y a partir de ahí se tiene la dirección donde se ubica

5.4. GESTIÓN DE ARCHIVOS. ACCESO INDEXADO

```

+-----+-----+-----+-----+
| Clave | Clave | Clave | Clave |
| Principal | Secundaria | Secundaria | Secundaria |
+-----+-----+-----+-----+
| (Direc) | (D.N.I.) | (Nombre) | (Provincia) | (Edad) |
+-----+-----+-----+-----+
| 1 | 55.366.546 | LUIS | Las Palmas | 16 |
| 2 | 42.386.225 | CARLOS | Salamanca | 17 |
| 3 | 32.387.203 | TERESA | Oviedo | 16 |
| 4 | 46.399.554 | INMA | Palencia | 20 |
| 5 | 60.643.434 | JOAQUIN | Salamanca | 17 |
| 6 | 22.543.986 | JOSE | Las Palmas | 23 |
+-----+-----+-----+-----+

```

Como podemos observar, esto sería un ejemplo de un fichero indexado.

Para cada campo clave, el fichero genera una tabla, donde dicha clave aparece ordenada alfabéticamente y se relaciona con la dirección de los datos. Así las tablas para la clave principal (DNI) y la clave secundaria (Nombre) serían:

```

+-----+-----+
| (D.N.I.) | (Direc) |
+-----+-----+
| 22.543.986 | 6 |
| 32.387.203 | 3 |
| 42.386.225 | 2 |
| 46.399.554 | 4 |
| 55.366.546 | 1 |
| 60.643.434 | 6 |
+-----+-----+
| Tabla de Ac. Clave Princ. |
+-----+-----+

+-----+-----+
| (Nombre) | (Direc) |
+-----+-----+
| CARLOS | 2 |
| INMA | 4 |
| JOAQUIN | 5 |
| JOSE | 6 |
| LUIS | 1 |
| TERESA | 3 |
+-----+-----+
| Tabla de Ac. Clave Secund. |
+-----+-----+

```

5.4. GESTIÓN DE ARCHIVOS. FICHEROS

○ Atributos

- Información adicional, además de la que ya contienen, con la que cada archivo está caracterizado
- Indican el nombre, la hora y fecha de creación, longitud, protección, contraseña de acceso, fecha de actualización, etc.
- Atributos de protección: indican que tipo de operaciones o qué tipo de usuarios pueden usar o no los archivos:
 - Sólo lectura
 - Modificable
 - Oculto
 - Sistema

5.4. GESTIÓN DE ARCHIVOS. SISTEMA DE ARCHIVOS

○ Sistema de archivos (file system)

- Técnica que determina cómo almacenar los datos de manera estructurada
- Tipos: cada sistema de archivos establece una gestión diferente del espacio de almacenamiento:

Sistema de Archivos	Sistema/s Operativo/s
FAT16	MS-DOS y todas las versiones de Windows
FAT32	Windows 98/ME/2000/XP/Vista/7
NTFS	Windows NT/2000/XP/Vista/7
EXT2/EXT3	LINUX
ReiserFS	LINUX
S5	UNIX
HTFS	OS/2 (IBM)

5.4. GESTIÓN DE ARCHIVOS. SISTEMA DE ARCHIVOS

- Los distintos sistemas de archivos no son siempre compatibles entre sí:
 - Unix no tiene problemas con los sistemas FAT de Windows, pero no pasa lo mismo al contrario.
 - Windows NT es compatible con FAT16, pero no con FAT32.

5.4. GESTIÓN DE ARCHIVOS. SISTEMA DE ARCHIVOS: FAT

- FAT (*File Allocation Table*) o Tabla de asignación de archivos
- Permite organizar la información en forma de ficheros
- La FAT funciona como el índice de un libro, es decir, almacena la información sobre:
 - Dónde comienza cada fichero, qué posición del dispositivo de almacenamiento está la primera parte de éste y cuánto ocupa
- La FAT ha ido evolucionando a medida que lo hacían los SSOO que la utilizaban, como MS-DOS y Windows
- Formatos: FAT 16 y FAT 32

5.4. GESTIÓN DE ARCHIVOS. SISTEMA DE ARCHIVOS: FAT16

- Limitación en cuanto al tamaño que es capaz de gestionar
- Los sectores libres se ocupan secuencialmente:
 - el primer archivo ocupará los sectores libres contiguos, que necesite en la zona de datos
 - el siguiente archivo a continuación de los sectores ocupados por el primer archivo y así sucesivamente
- Cuando se borra un fichero, el espacio libre debe poder volver a utilizarse
- La FAT se encarga de indicar al SO qué clústeres ocupa un fichero

5.4. GESTIÓN DE ARCHIVOS. SISTEMA DE ARCHIVOS: FAT16

- Los archivos se nombran con caracteres ASCII:
 - Como máximo 8 caracteres.
 - Seguido de un separador (.) y Una extensión de hasta tres caracteres
 - El nombre no puede contener espacios en blanco.
 - El nombre debe empezar con una letra o número y puede contener cualquier carácter excepto:

. " / \ [] : ; | = ,

- Todos los caracteres se convertirán a mayúsculas
- Nombres reservados: CON, AUX, COM1, COM2, COM3, COM4, LPT1, LPT2, LPT3, PRN, NUL

5.4. GESTIÓN DE ARCHIVOS. SISTEMA DE ARCHIVOS: FAT32

- Respuesta para superar el límite de tamaño de FAT16
- Aún se queda pequeño para aplicaciones de vídeo, que trabajan con archivos muy grandes

5.4. GESTIÓN DE ARCHIVOS. SISTEMA DE ARCHIVOS: NTFS

- NTFS (*New Technology File System*) ó Sistema de archivos de nueva tecnología
- Nace con Windows NT, que al ser un sistema operativo en red necesitaba más prestaciones que las que podía ofrecer FAT
- NTFS no es aconsejable para discos con menos de 400 Mb libres, ya que necesita mucha cantidad de disco duro para sí mismo
- MTF (*Master File Table*): tabla maestra es la estructura central de NTFS
- Lo utilizan los SO Windows NT, 2000, XP, 2008 Server, Vista, 7
- Es más seguro, rápido y da mayor calidad al aprovechar mejor el espacio físico del disco que FAT
- Es compatible con cualquier formato FAT

6. EJERCICIOS

1. ¿Se puede explotar un sistema monoproceso por varios usuarios?
2. ¿De qué se encarga el nivel ejecutivo de un sistema operativo? ¿Y el supervisor?
3. Consulta información en Internet o en revistas de informática y contesta a las siguientes preguntas:
 - a) ¿Cuáles son los cuatro sistemas operativos de mayor difusión y aceptación?
 - b) ¿Cuáles son los precios actuales de esos sistemas operativos?
4. Comentar que sistema operativo es mejor dependiendo de las siguientes situaciones:
 - a) Trabajar en casa con varios ordenadores conectados en red.
 - b) Trabajar en una empresa con datos a compartir por los usuarios de la empresa.
 - c) Trabajar en una empresa solo conectado a Internet.
5. Realiza un esquema completo del tema.

6. EJERCICIOS

6. Completa la siguiente tabla:

Sistema operativo	Número de usuarios	Número de procesos	Número de procesadores	Tiempo de respuesta
DOS				
Windows NT				
Windows 9X, ME				
Windows 2000 Professional				
Windows 2000 Server				
Windows XP				
Windows Vista				
Windows 7				
Windows Server 2003				
Windows Server 2008				
UNIX				
Linux				

6. EJERCICIOS

7. Ejercicios Hoja 1
8. Ejercicios Hoja 2
9. Ejercicios Hoja 3

1.4.1. MODELO TABLA PLANIFICACIÓN

○ Ejemplo:

Proceso	Instante llegada	Duración	t_f	T	E	I
A	0	3				
B	1	5				
C	4	2				
D	5	6				
E	8	4				

5.1.1. PLANIFICACIÓN

○ Ejemplo:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A																					
B																					
C																					
D																					
E																					