

# Sistemas Operativos

---

TEMA 2: INTRODUCCIÓN A LOS SISTEMAS  
OPERATIVOS

# Definiciones

Un sistema operativo (S.O.) es:

- “**Conjunto de programas que controla la ejecución de los programas de aplicación y actúa como interfaz entre el usuario y el hardware del computador**”
- “**Un programa que tiene encomendadas una serie de funciones diferentes cuyo objetivo es simplificar el manejo y la utilización de la computadora, haciéndolo seguro y eficiente.**

Los programas de usuario:

- Utilizan los **servicios proporcionados por el S.O.** para tener **acceso a los recursos** (ficheros, dispositivos E/S, procesador, memoria, etc.)

# Objetivos

## **Comodidad $\Rightarrow$ Máquina virtual o extendida**

El S.O. hace que sea más fácil de usar el computador (se ocultan los detalles HW)

## **Eficiencia:**

El S.O. utiliza los recursos de la forma más eficiente posible (ficheros, dispositivos E/S, memoria, procesador, etc.)

## **Seguridad:**

Asignación controlada de dispositivos E/S

## **Capacidad de evolución:**

Un S.O. debe poder ampliarse fácilmente:

Actualizaciones del Sistema y controladores

Nuevos servicios

Correcciones, parches de seguridad, etc...

# S.O. - Objetivos

**El sistema operativo puede ser visto como:**

- El interfaz entre el usuario y la máquina –  
[The Operating System as a User/Computer Interface](#)
- El gestor de recursos  
[The Operating System as Resource Manager](#)

**El sistema operativo reside en disco y debe ser cargado a memoria**

[El proceso de arranque de la computadora](#)

# Usuarios y Grupos

- Un usuario es una persona autorizada a usar el sistema. El usuario se autentica mediante usuario y password.
- El S.O. no asocia el usuario con persona física sino con el concepto de Cuenta.
- Cada usuario (cuenta) tiene asociado un identificador (uid) y un perfil de usuario
- Cada usuario y perfil tienen asociados unos permisos.
- Existe un usuario privilegiado (Administrador o super usuario (root)) sin restricción alguna
- Los usuarios se organizan en grupos
- Todo usuario debe pertenecer a un grupo
- Los grupos también tienen privilegios
  - Los privilegios de un usuario son los propios, más los del grupo/grupos al que pertenece el usuario.

# S.O. - Ejemplos

DOS (PC-DOS, DR-DOS, MS-DOS)

Windows (3.1x, 9x, NT, 2000, XP, .

OS/2 (IBM)

Unix / Linux

- HP-UX – AIX etc...

Macintosh (Apple)

OS/400

Solaris (Sun Mic.)

VMS (Digital)

MVS (IBM)

GECOS (BULL)

Symbian

Darwin BSD

Mac Os X

IOS n

Android

BlackBerry OS

...



Microsoft Corporation, 1978

# Estructura de los S.O.

---

# Estructura de un sistema

Estructura de un sistema de información:

- Hardware (CPU, memoria, dispositivos E/S)
- Sistema Operativo
- Utilidades y Librerías
- Aplicaciones de usuario
- Usuarios (personas y otros computadores)

Núcleo (Kernel):

- Parte/s fundamental del sistema operativo que siempre se está ejecutando y lo hace en modo privilegiado.
- Responsable de facilitar el acceso seguro al hardware.
- Encargado de gestionar los recursos.
- Multiplexa el acceso al hardware y recursos.
- Implementa capas de abstracción del hardware.

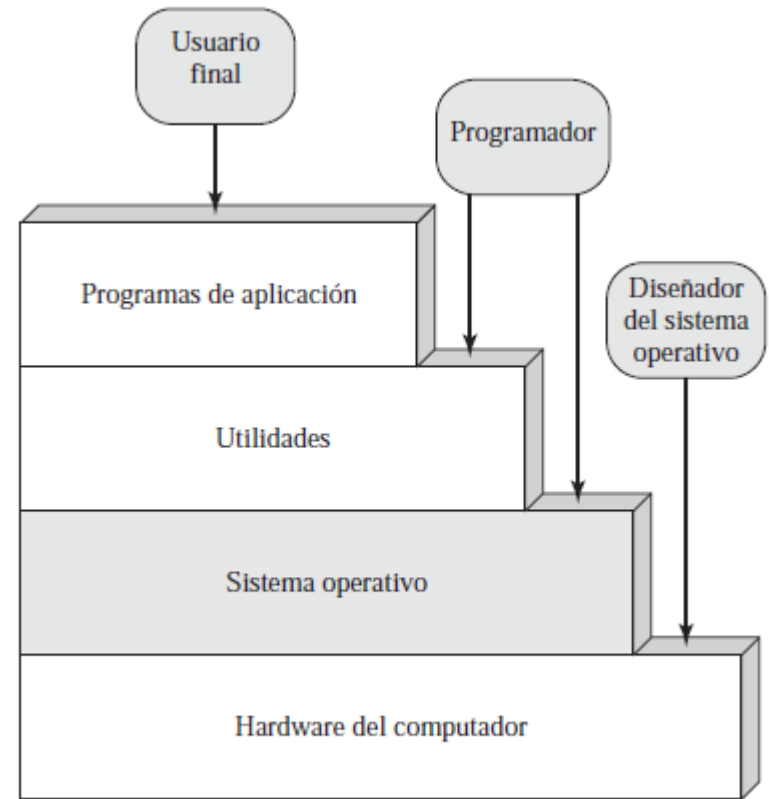


Figura 2.1. Capas y vistas de un sistema de computación.



# Componentes del sistema operativo

El núcleo interactúa directamente con el hardware.

## Gestión del procesador

Tratamiento de interrupciones

Funciones básicas de memoria y DMA

## Gestión de procesos

Creación, planificación y destrucción de procesos

## Gestión de memoria

Mantenimiento del mapa de memoria libre, asignación y liberación

## Gestión de E/S

Acceso a los registros y protocolos de comunicación con periféricos

## Gestión de archivos y directorios

Manejar archivos y directorios. Administración del sistema secundario.

## Seguridad y protección

Identidad de los usuarios, gestión de privilegios, detección de violaciones

## Comunicación y Sincronización de procesos

Ofrece mecanismos para que los procesos puedan comunicarse y sincronizarse entre ellos en la propia máquina y entre máquinas.

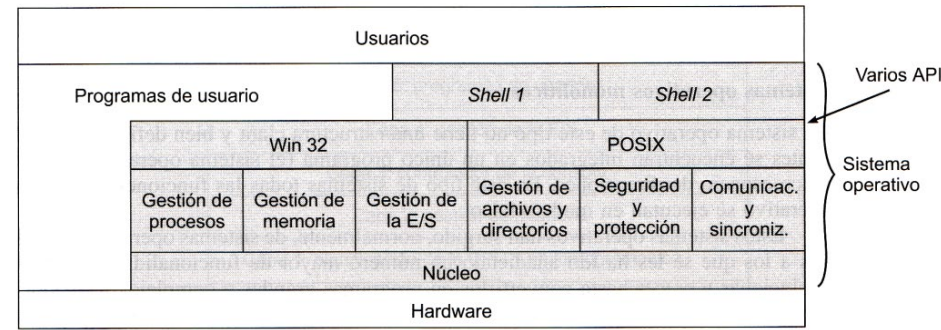
## Interfaz de llamadas al sistema

Son capas de abstracción que incluyen programas que encapsulan las llamadas al sistema junto con utilidades del sistema

Se pueden ofrecer varias interfaces en el mismo sistema, pero solo se puede usar una simultáneamente.

## Interpretes de comandos.

Puede ofrecer varios, de texto y gráficos. No se pueden mezclar mandados de varios simultáneamente.



# Estructura de los S.O.

Cada SO estructura sus componentes en formas diferentes. Esta estructura depende de la evolución del SO y el uso al que está destinado.

Sistemas monolíticos

Sistemas estructurados

- Sistemas por capas
- Sistemas cliente-servidor

Sistemas de máquinas virtuales

# Sistemas Monolíticos

No tiene estructura definida.

Un único programa.

Todas las funciones que oferta el S.O. se ejecutan en modokernel.

Procedimientos y funciones con interfaz bien definidas.

Libertad de llamada entre ellos.

No hay ocultación de información, se ven entre ellos y sus datos.

Dentro de la no estructura existen procedimientos de servicio y de utilidades.

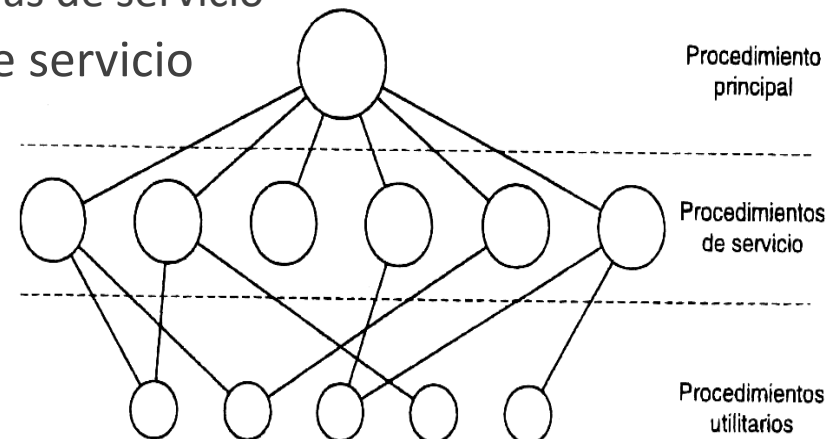
- Programa principal que llama a una rutina de servicio
- Conjunto de rutinas de servicio
- Conjunto de rutinas de utilidades que apoyan a las de servicio

Para cada llamada al sistema hay una rutina de servicio

Problemas con la evolución.

Difícil programación.

Parcheado.



# Sistemas Estructurados - Por capas

Jerarquía de capas.

- Cada capa ofrece servicios y utiliza servicios de otras capas. Solo conocen la interfaz.
- No se conocen detalles de implementación
- Modularidad y ocultación.
- Desarrollo y mantenimiento separado por capas.
- Fácil depuración

Ej. THE  
(Dijkstra 1968)

Ej. MULTICS

Ej. OS/2 Microsoft

Capa 5: Programas de usuario
Capa 4: Gestión de E/S
Capa 3: Controlador de consola
Capa 2: Gestión de Memoria
Capa 1: Planificación de CPU y multiprogramación
Capa 0: Hardware

# Sistemas Estructurados - Cliente/Servidor

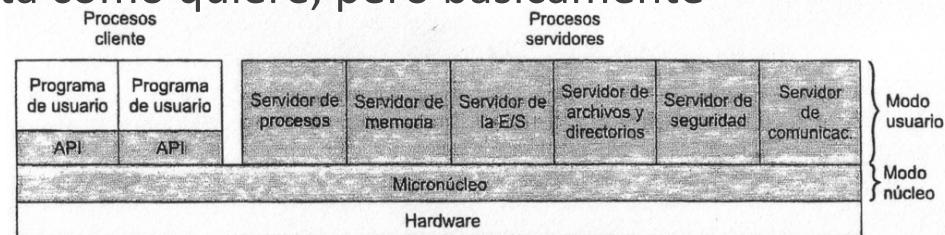
Solo una pequeña parte compone el Microkernel

El resto de los servicios se implementa como servicios en modo usuario.

Son los llamados Servidores.

Microkernel, cada sistema lo implementa como quiere, pero básicamente contiene:

- Gestión de interrupciones
- Gestión de memoria básica
- Gestión básica de procesos
- Gestión de servicios de comunicación entre procesos



Los procesos de usuario solicitan servicio a los servidores, y estos entre ellos si se requiere.

Los servidores utilizan servicios del microkernel.

## Ventajas y Desventajas

- Flexibilidad.  
Funcionalidades manejables.  
Facilita desarrollo y depuración.  
Si falla un servicio no se detiene todo el sistema.  
Adaptabilidad a sistemas distribuidos.
- Mayor sobrecarga y tiempos de servicio.  
Competencia por recursos.

Ej. Minix (Tanenbaum, 1998)  
Mach (Acceta, 1986)  
Amoeba (Mullender, 1990)  
Windows NT (gestión de procesos, E/S, Memoria, etc. al kernel por eficiencia)

# Sistemas de Máquinas Virtuales

El núcleo del sistema es el Monitor de Maquina Virtual. Ej. VM/370

- Se ejecuta sobre el hardware
- Utiliza multiprogramación para acceder al hardware

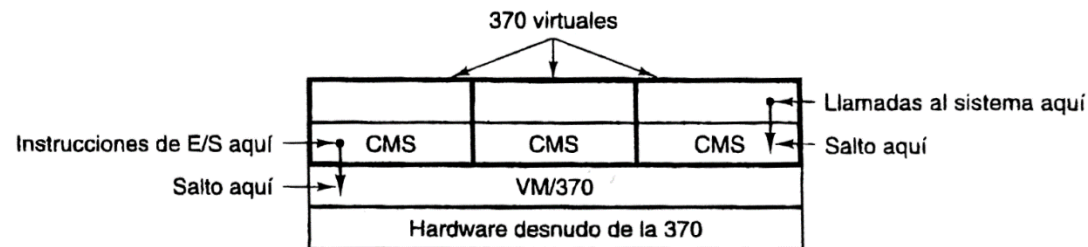
Sobre el Monitor se implementan las Máquinas Virtuales. Replicas lógicas del hardware (registros, cache, etc...)

En cada máquina virtual se puede ejecutar un SO con diferentes características.

VM/370-CMS (emulación total del hardware), MS-DOS sobre Windows NT-XP (emulación de un 8086)

SUN JVM (Java Virtual Machine)

- Un programa se compila y genera código para la JVM
- Esta JVM se puede implementar en cualquier SO para correr en cualquier



# Subsistemas

---

# Principales subsistemas

Gestión de procesos



Gestión de memoria



Gestión de E/S



Gestión de Archivos y Directorios





# Subsistema: Procesos



Es el subsistema principal. Todo gira alrededor del concepto de proceso

El proceso es

- Un programa en ejecución
- Una instancia de un programa funcionando en un computador
- La entidad que puede ser asignada al procesador y ejecutada por el.
- Una unidad de actividad caracterizada por un sencillo tratamiento de ejecución secuencial, un estado actual y asociada a un conjunto de recursos del sistema

No confundir Proceso y Programa

- Un programa es simplemente un conjunto de instrucciones máquina

El subsistema se encarga de generar y gestionar los procesos y atender sus peticiones.

# Subsistema: Procesos



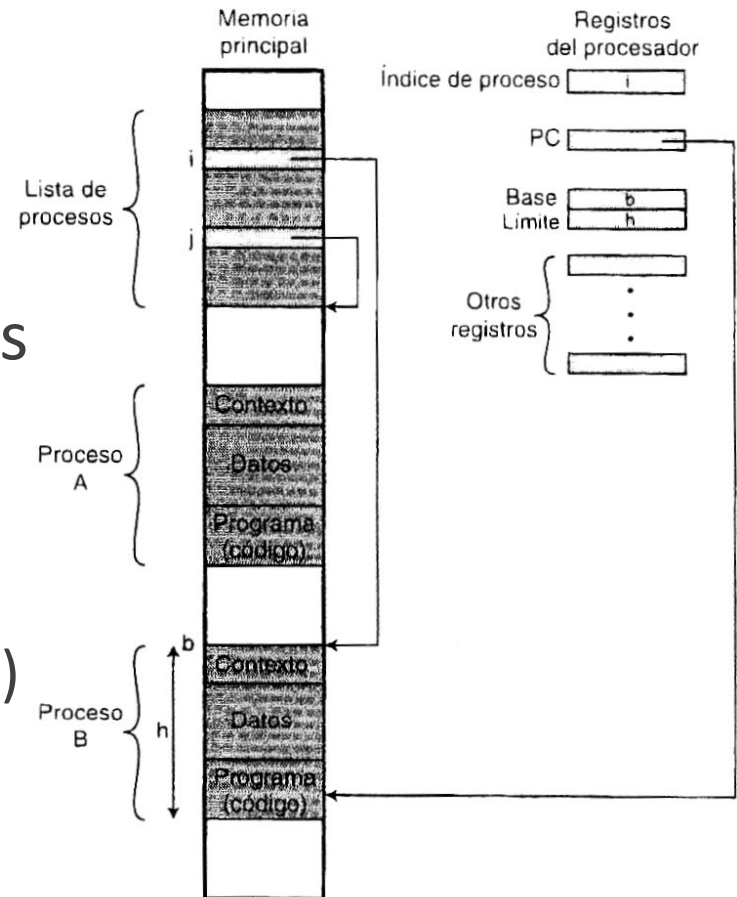
Para ejecutar un programa debe residir en memoria.

- Imagen en memoria – Código - Datos
- Estado del procesador – Parte del PCB

Gestión de las estructuras y servicios que conforman la idea de proceso

- PCB
- Servicios de gestión de procesos
- Resolución de conflictos (interbloqueos)

El proceso se trata como una estructura de datos que está en un **estado** determinado.



# Subsistema: Procesos



## Servicios de procesos

- Creación de procesos
  - Proceso padre crea procesos hijos
  - Creación
    - A partir del proceso padre. UNIX
    - A partir de un archivo ejecutable. Windows
- Ejecución de procesos
  - Ejecución batch (background).
    - El proceso no tiene asociado terminal
    - E/S de y hacia archivos
  - Ejecución interactiva
    - El proceso está asociado a un terminal
- Terminación de procesos
  - Termina la ejecución del programa
  - Se produce una condición de error (p.ej. División por 0, violación de memoria)
  - Otro proceso, el SO o el usuario deciden que debe terminar.
- Cambio de programa del proceso
  - Cambia el código que está ejecutando por el de otro programa almacenado en disco, normalmente a petición propia
  - No crea otro proceso, sustituye el código que el proceso ejecuta.

# Subsistema: Procesos



Un proceso puede dividir su trabajo

- Se crean relaciones entre procesos padre-hijo, hijo-hijo

El subsistema proporciona servicios de comunicación y sincronización

- Locales
- Remotos

## ❑ Comunicación

- ❑ Síncrona: Espera en el servicio
- ❑ Asíncrona: Fuerza al SO a gestionar el almacenamiento temporal

## ❑ Los mecanismos de comunicación tienen entidad propia

- ❑ Se crean
- ❑ Se utilizan
- ❑ Se liberan

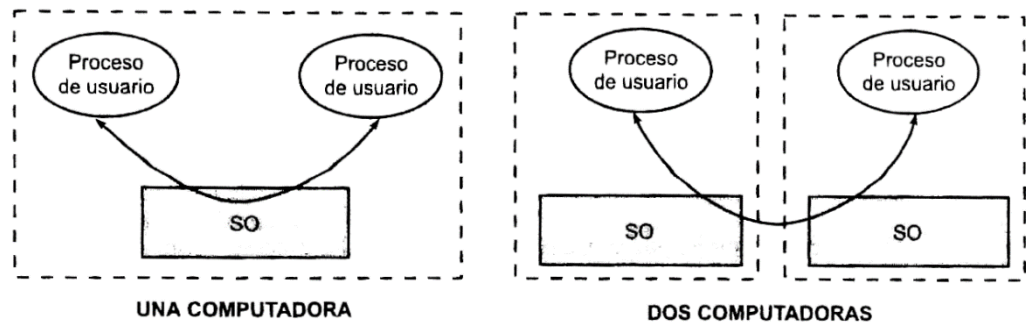


Figura 2.9. Comunicación entre procesos locales y remotos.

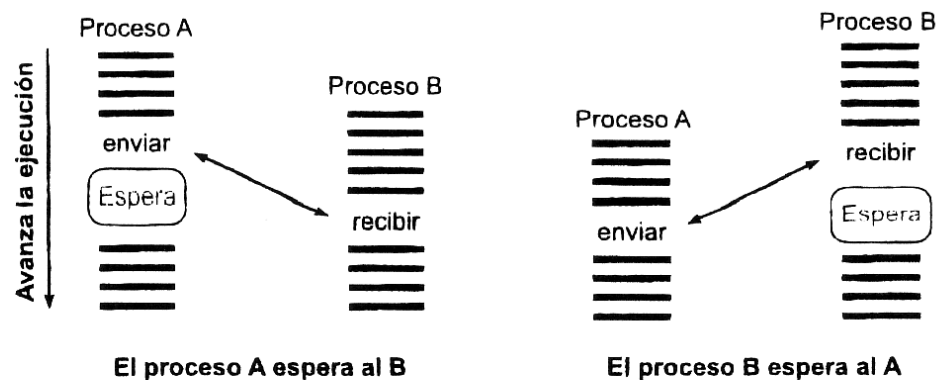


Figura 2.10. Comunicación síncrona entre procesos.

# Planificación de Procesos y Asignación de Recursos



En un entorno de multiprogramación la existencia de las **colas de procesos** permite la planificación y asignación.

## Corto plazo

- La cola a corto plazo mantiene procesos (completos o no) en memoria principal. Listos para ejecución
- El planificador a corto plazo y el distribuidor (Short-Term Scheduler, Dispatcher) se encargarán de llevar el proceso a ejecución
- Round-Robin o Niveles de prioridad

## Largo plazo

- Nuevos trabajos en espera de usar el procesador.
- El Planificador a largo se encarga de balancear la carga del sistema y pasarlos a la cola de corto plazo
- Al entrar en corto se asigna memoria al proceso

## Colas de E/S

- Hay una para cada dispositivo gestionadas por el S.O.
- Los procesos en estas colas están bloqueados o suspendidos

El sistema operativo toma el control cuando se produce una interrupción

- Retorna al planificador a corto

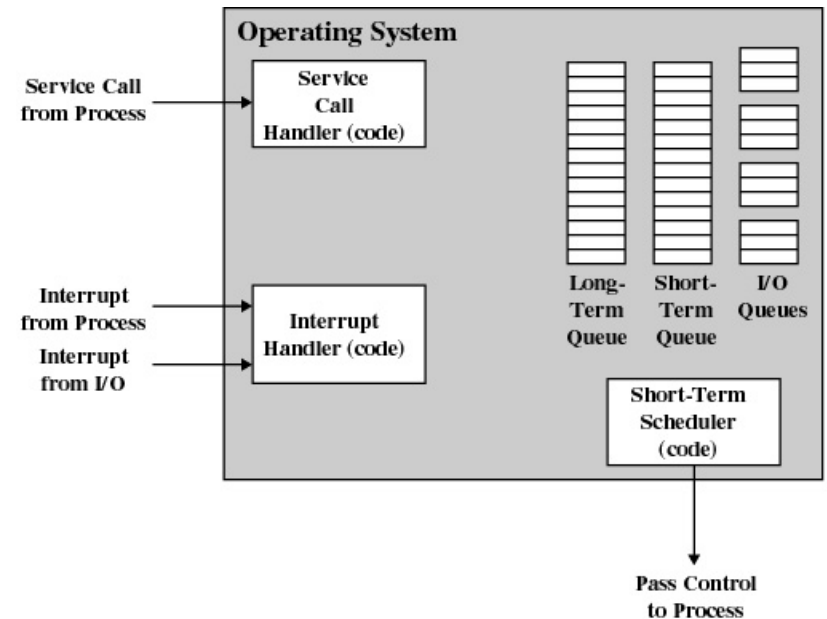
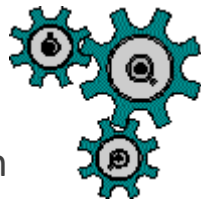


Figure 2.11 Key Elements of an Operating System for Multiprogramming

# Activación del S.O.



¿Cuáles son las acciones que activan la ejecución del S.O.? ¿Cómo es el proceso de activación  
El S.O. es un servidor que está a la espera de que se le encargue trabajo.

## Secuencia normal

- Un proceso A en Ejecución solicita un servicio del S.O.
- Se salva el estado del proceso A
- Se realiza la función pedida
- Se Ejecuta el planificador
- Se Ejecuta el Activador
- Se ejecuta el proceso B

## Solicitud por una de estas vías

- Llamadas al sistema emitidas por los procesos de usuario o sistema (TRAP)
- Interrupciones producidas por los periféricos o hardware
- Condiciones de excepción o error del hardware

Las llamadas al sistema están encapsuladas en funciones de librería (POSIX, WIN32...)

```
int fork() {  
    int r;  
    LOAD R8, FORK_SYSTEM_CALL  
    TRAP  
    LOAD r, R9  
    return (r);  
}
```

Una función de librería que llama al sistema suele componerse de:

- Una parte inicial de preparación de parámetros del servicio conforme lo requiere el S.O.
- Una instrucción TRAP
- Una parte final que recupera los parámetros de contestación del S.O. para devolverlos (en crudo, modificados o codificados) al programa que la llamó.

# Subsistema: Memoria



Asignación de memoria al Gestor de Procesos para crear la imagen del proceso

Garantizar el acceso correcto a las zonas de memoria

- Garantizar que ningún otro proceso viole la memoria de otro
- Permitir acceder a distintas zonas de memoria en distintos modos

Proporcionar memoria a los procesos que lo soliciten y gestionar su liberación, tanto para datos como para código

- Aumentar el tamaño de la asignación del proceso, datos o código
- Siempre que haya memoria disponible devuelve un puntero a la nueva zona asignada
- Disminuir el tamaño de la asignación del proceso y marcar la zona como disponible

Tratamiento de los errores de acceso a memoria

Gestión de la compartición de memoria

- Ofrece servicios para asignar segmentos de memoria compartida a varios proceso cooperantes.

# Subsistema: Memoria



## Gestionar la memoria virtual

- Permitir direccionar más memoria de la que realmente hay disponible de manera transparente para el programador
- Permite a los programas direccionar la memoria desde un punto de vista lógico
- Permite la coexistencia en memoria de varios procesos
- Así no existirá un tiempo muerto entre la ejecución de procesos sucesivos, puesto que el siguiente proceso planificado ya reside en memoria.

## Sistema de paginación

- Permite que los procesos estén formados por varios bloques de tamaño fijo, denominados páginas
- La dirección virtual está formada por un número de página y un desplazamiento dentro de la página
- Cada página puede estar ubicada en un lugar cualquiera de la memoria principal
- La dirección real o dirección física se utiliza en la memoria principal.



# Direccionamiento de la Memoria Virtual

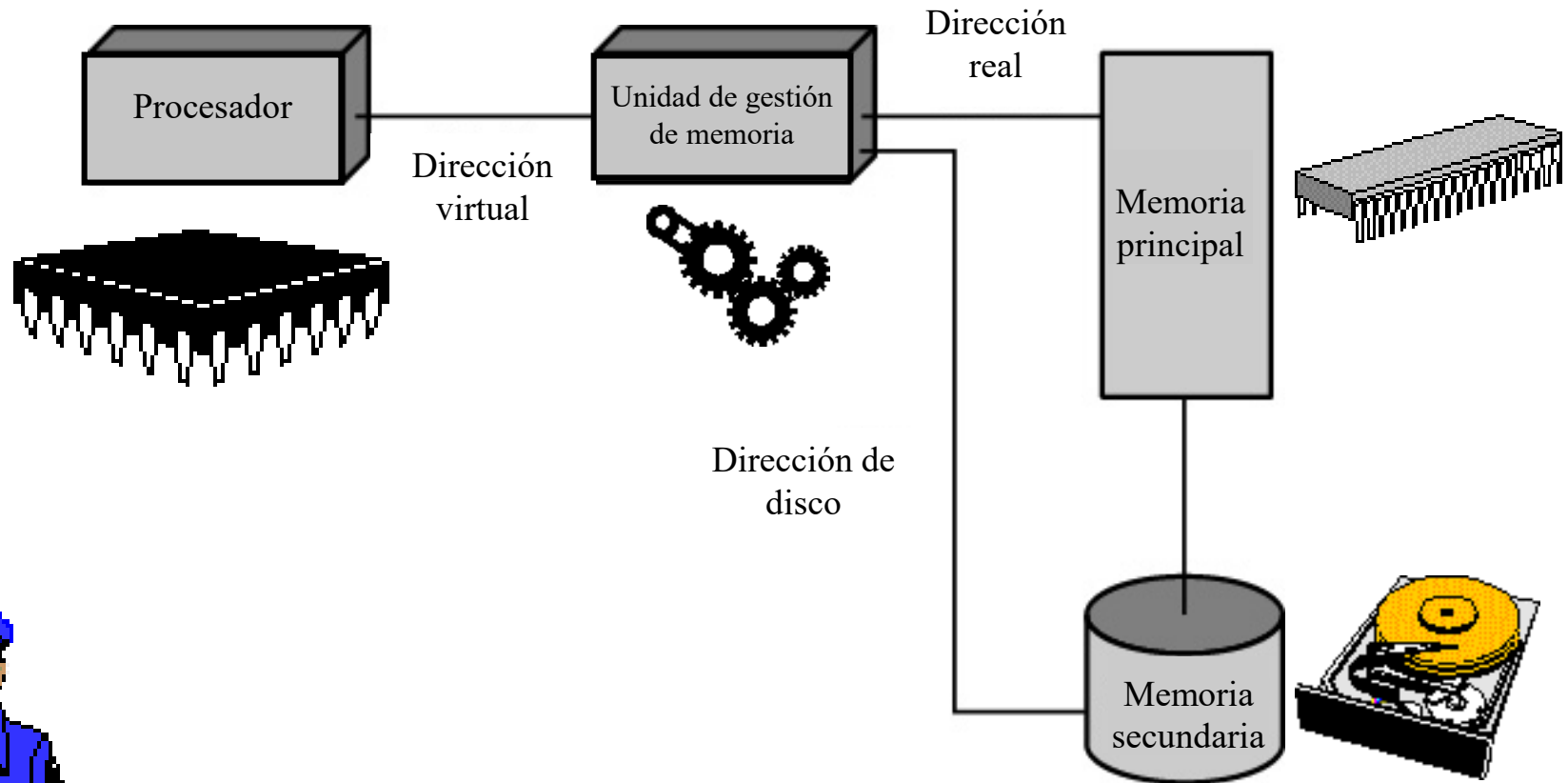
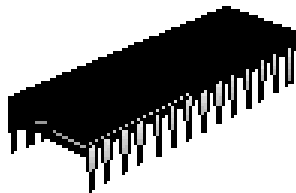
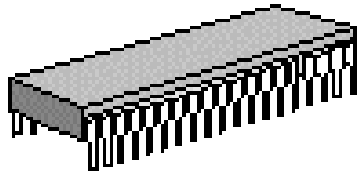


Figure 2.10 Virtual Memory Addressing

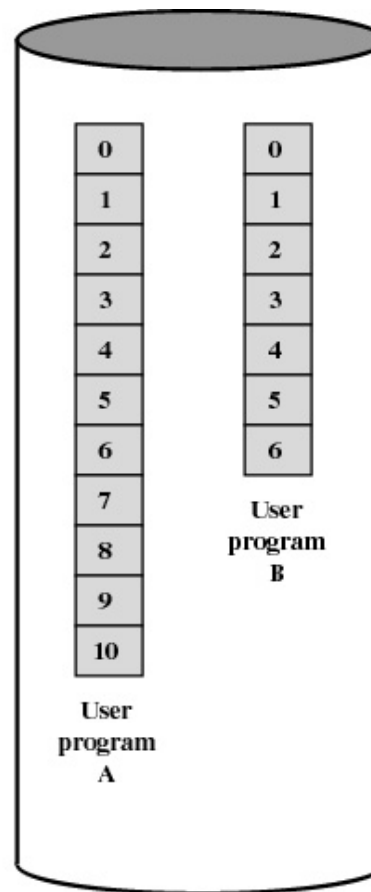




A.1			
	A.0	A.2	
	A.5		
B.0	B.1	B.2	B.3
		A.7	
	A.9		
		A.8	
B.4	B.5	B.6	

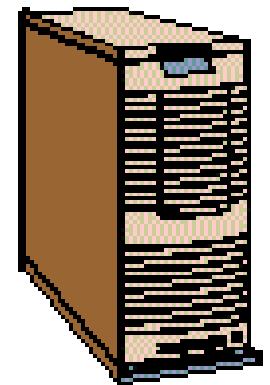
**Main Memory**

Main memory consists of a number of fixed-length frames, equal to the size of a page. For a program to execute, some or all of its pages must be in main memory.



**Disk**

Secondary memory (disk) can hold many fixed-length pages. A user program consists of some number of pages. Pages for all programs plus the operating system are on disk, as are files.



**Figure 2.9 Virtual Memory Concepts**

# Subsistema: E/S

Controlar el funcionamiento de todos los dispositivos de E/S.

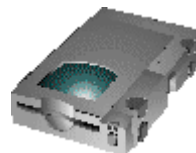
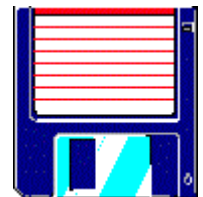
- Relojes, terminales, almacenamiento secundario, terciario, teclado, puertos, etc.

Facilitar su manejo

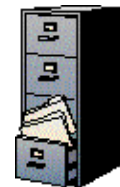
- Ofrecer una interfaz uniforme entre dispositivos
- Gestionar los errores
- Ofrecer mecanismos de protección

Ofrece servicios independientes del dispositivo

- Básicamente orientados a la lectura/escritura
- Orientados a carácter como
  - Impresoras
  - Terminales
  - Teclado ...
- Orientados a bloque
  - Discos



# Subsistema: Archivos y Directorios



## Servidor de archivos

Es la parte del S.O. encargada de:

- Facilitar el manejo de dispositivos periféricos.

El S.O. ofrece una visión lógica de los periféricos como si fueran archivos.

- Proteger y limitar los archivos que cada usuario es capaz de manipular

Los servicios son de dos tipos:

- Orientados al manejo de datos o archivos
- Orientados al manejo de nombres o directorios



□ El servidor de archivos ofrece una Visión lógica y una Visión Física.

- La visión lógica trata objetos (archivos y directorios) identificados por nombre lógico sobre los que se pueden realizar una serie de operaciones
- La visión física trata con los detalles de cómo están almacenados los objetos en los periféricos (discos, etc...)

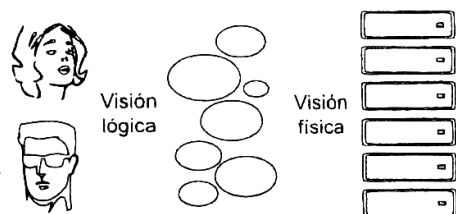


Figura 2.11. Visión lógica y física del sistema de archivos.

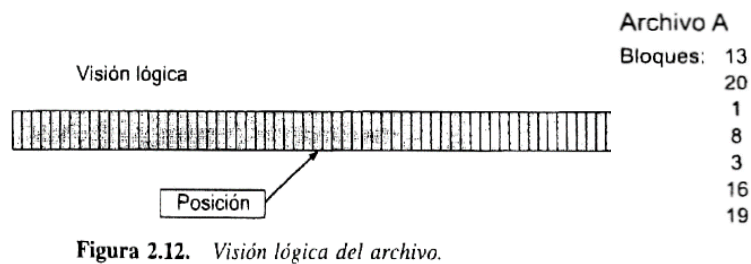


Figura 2.12. Visión lógica del archivo.

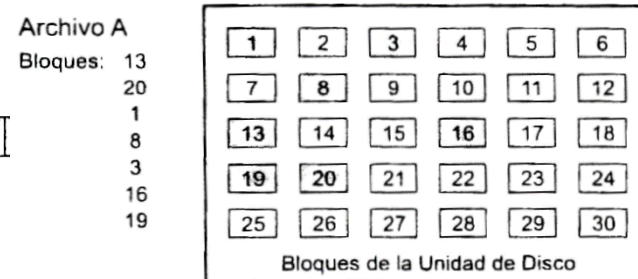


Figura 2.13. Visión física del archivo.

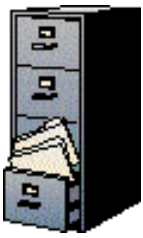
# Subsistema: Archivos y Directorios



## Servicio de Archivos

- Una unidad de almacenamiento lógico no volátil.
- El servidor tiene información asociada al archivo para el uso propio y de los usuarios
  - Tipo de archivo,
  - Propietario,
  - Tamaño (real, asignado),
  - Fechas (creación, acceso, modificación...)
  - Derechos de acceso (sólo lectura, lectura-escritura, ejecución, ... por quién)
  - etc..
- Operaciones sobre la visión lógica del archivo
  - Puntero de posición, posición, movimiento,
  - Crear, Abrir, Leer, Escribir, Añadir, Cerrar, Borrar
- Operaciones sobre la visión física del archivo
  - Gestión de bloques de disco
  - Organización del almacenamiento de los bloques
- Gestionar la información lógica y física
  - La *Descripción Física del Archivo* es una estructura de información que recoge la composición física del archivo
  - Es información es auto contenida en el disco para poder transportarse de un sistema a otro
    - La FAT en MS-DOS,
    - NTFS en Windows NT-XP-2000,Win X,
    - I-nodes en UNIX

# Subsistema: Archivos y Directorios



## Servicio de Directorios

- Es un objeto que relaciona un nombre con un archivo
- Sirve para identificar y ubicar a los archivos
- Un mismo nombre no puede referenciar a varios archivos
- Un archivo puede tener varios nombres

## Visión Lógica

- Esquema jerárquico de directorios y archivos
- Directorio raíz, subdirectorios, archivos
- Nombres: referencia local y referencia absoluta

## Visión Física

- Tablas de i-nodes, Tabla FAT o NTFS
- Relaciones nombres-descriptores o identificadores

Operaciones ofrecidas por el sistema de archivos

Crear, Borrar, Abrir, Leer, Cerrar

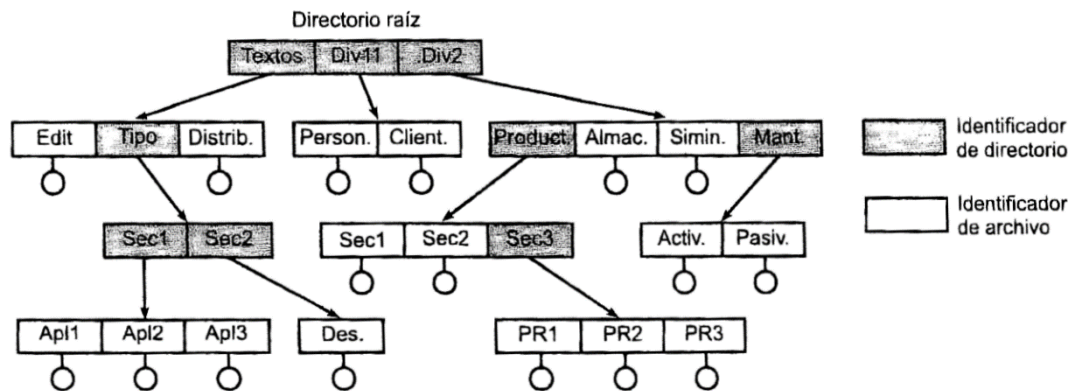


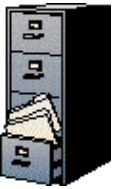
Figura 2.15. Esquema jerárquico de directorios.

### Visión Física:

Los directorios son archivos que contienen parejas NOMBRE-IDENTIFICADOR

NOMBRE: Nombre (lógico) del archivo  
IDENTIFICADOR: Información de dónde reside (o comienza) la descripción física del archivo.

# Subsistema: Archivos y Directorios



## SISTEMA DE ARCHIVOS

Es el conjunto de archivos incluidos en una unidad de almacenamiento (disco)

Compuesta por los datos de los archivos y la información auxiliar (metainformación) para gestionarlos.

La metainformación del sistema de archivos está compuesta por:

- Estructura física de los archivos (i-nodes / FAT / NTFS )
- Directorios (archivos que contienen tablas de nombre-puntero)
- Estructura física del sistema de archivos (superbloque)
- Estructura de información de bloques e i-nodes libres (mapas de bits)

Cada S.O. organiza las particiones de disco de una determinada manera para repartir el espacio entre:

- El programa de carga (boot)
- La meta información
- Los datos
- El fichero de intercambio (swap)

# Interfaz del programador

---



# Interfaz del programador

## POSIX [IEEE96]

Es el estandar para sistemas operativos portables de IEEE basado en UNIX

No define la implementación solo la definición del estandar.  
Los distintos S.O. pueden ofrecer lo mismo programado de distinta forma

Es una familia de estandares en evolución

Se encuentra en todas las versiones actuales de UNIX y LINUX . También disponible para Windows.

Algunos estándares de POSIX son:

- 1003.1 Servicios básicos del S.O.
- 1003.1a Extensiones a los servicios básicos
- 1003.1b Extensiones de tiempo real
- 1003.1c Extensiones de procesos ligeros (threads)
- 1003.1d Extensiones adicionales de tiempo real
- 1003.1e Seguridad
- 1003.2 Shell y utilidades
- 1003.2b Utilidades adicionales

Algunas características

- Hay tipos de datos definidos en el estandar otros se dejan al implementador, por ejemplo en `<sys/types.h>` Estos tipos acaban con `_t` por ejemplo `uid_t` tipo que se emplea para el identificador de usuario.
- Nombres de funciones cortos con todas sus letras, por ejemplo: `fork`, `read`, `close`
- Las funciones normalmente devuelven 0 si ejecutaron con éxito o -1 en caso de error
- Cuando una función devuelve -1 el código de error se guarda en una variable global (`errno`) . Estos códigos están en `<errno.h>`
- La mayoría de los recursos gestionados por el sistema operativo se referencian como descriptores (un valor entero mayor que 0) Por ejemplo el descriptor de fichero al abrir un fichero con `fopen`

# Interfaz del programador

## Win32

Define los servicios ofrecidos por Windows 95/98, Windows NT y Windows 2000.

No es un estandar. Es una definición comercial de Microsoft.

El API es totalmente diferente al de POSIX

- Prácticamente todos los recursos del sistema operativo se tratan como objetos que se referencian mediante manejadores (*handlers*).
- Los nombres de las funciones son largos y descriptivos: `GetFileAttributes`, `CreateNamedPipe`, ...
- Tiene una serie de datos predefinidos
  - `BOOL`, `DWORD`, `TCHAR`, `LPSTR`
  - Evitan el uso de la indirección de C(\*). `LPSTR` está definido como `*TCHAR`
- Los nombres de las variables llevan el prefijo del tipo de datos
  - `lpzFileName` representa un puntero largo a una cadena de caracteres acabada en nulo
- Las funciones en general devuelven `true` si la llamada se ejecutó con éxito y `false` en caso contrario
- Define un completo conjunto de funciones y servicios gráficos.

# Clasificación y Evolución de los Sistemas Operativos

---

# Clasificaciones

La evolución de los sistemas operativos nos ha dejado una gran variedad de tipos de sistemas

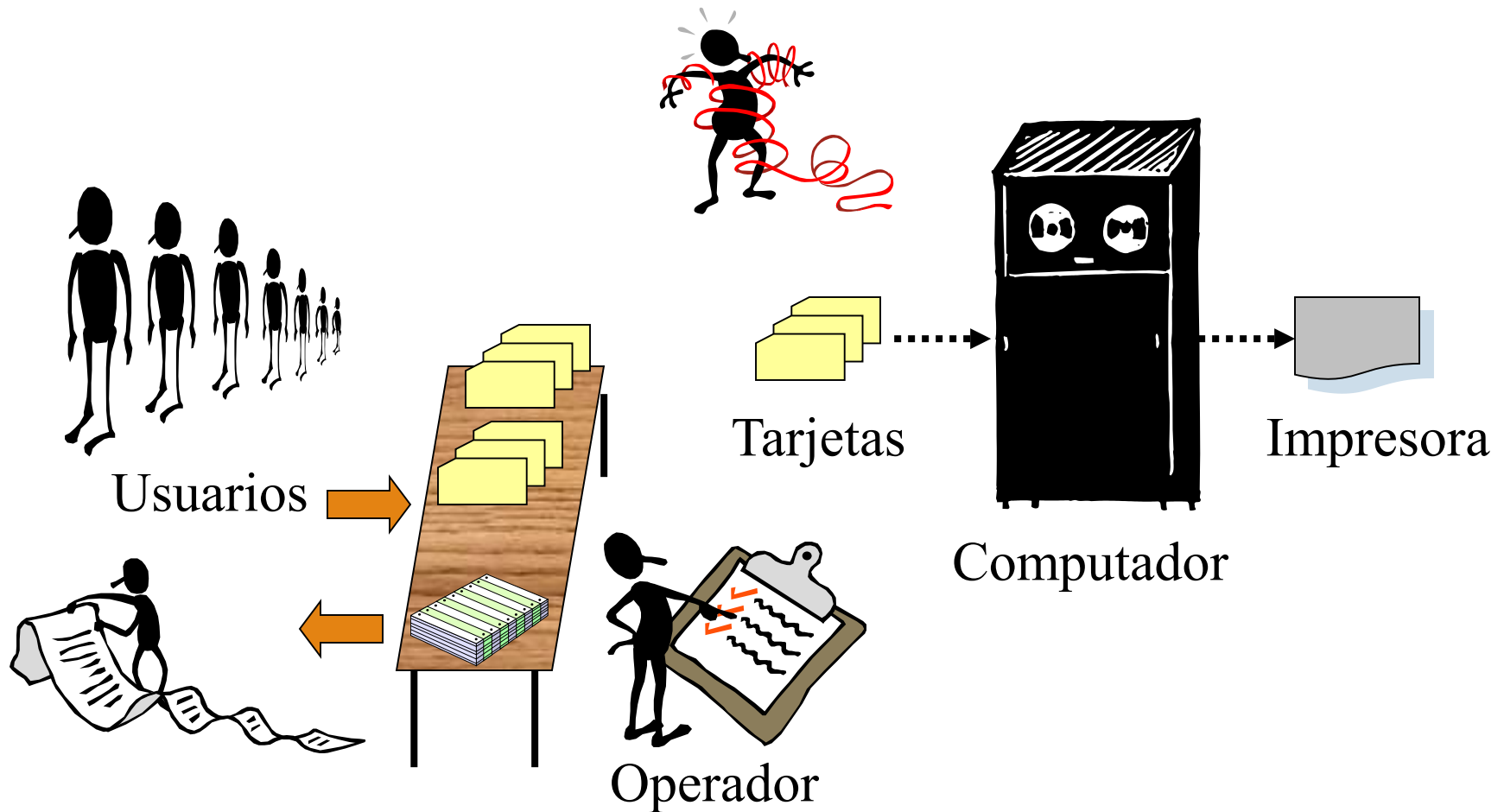
- De Mainframe. Sistemas operativos propietarios
  - Grandes máquinas en capacidad y tamaño
    - Procesamiento por lotes
    - Procesamiento de transacciones
    - Tiempo compartido
- De Servidor. Unix, Linux, Windows Server
  - Servidores – Computadoras personales muy grandes o incluso mainframes dedicados.
    - Múltiples usuarios con diversos perfiles
    - Compartición de recursos, servicios de impresión, servicios de correo, servicios web,
- Multiprocesador y Distribuidos
  - Computadoras paralelas
  - Multicomputadoras o multiprocesadores
  - Son SO para servidor con variaciones en conectividad y comunicaciones
- Computadora Personal
- De Tiempo Real
- Integrados
- De Tarjeta Inteligente

# Evolución – 1. En serie

Procesamiento en serie (1940-50):

- Usuario  $\neq$  Operador
- El usuario escribía los programas en código máquina  $\Rightarrow$  controlaba el HW directamente
- Programas orientados a cálculo matemático
- El operador era el encargado de insertar los trabajos de cada usuario de forma manual
- Cintas papel o tarjetas perforadas
- Problemas: planificación y tiempo preparación

# Evolución – 1. En serie



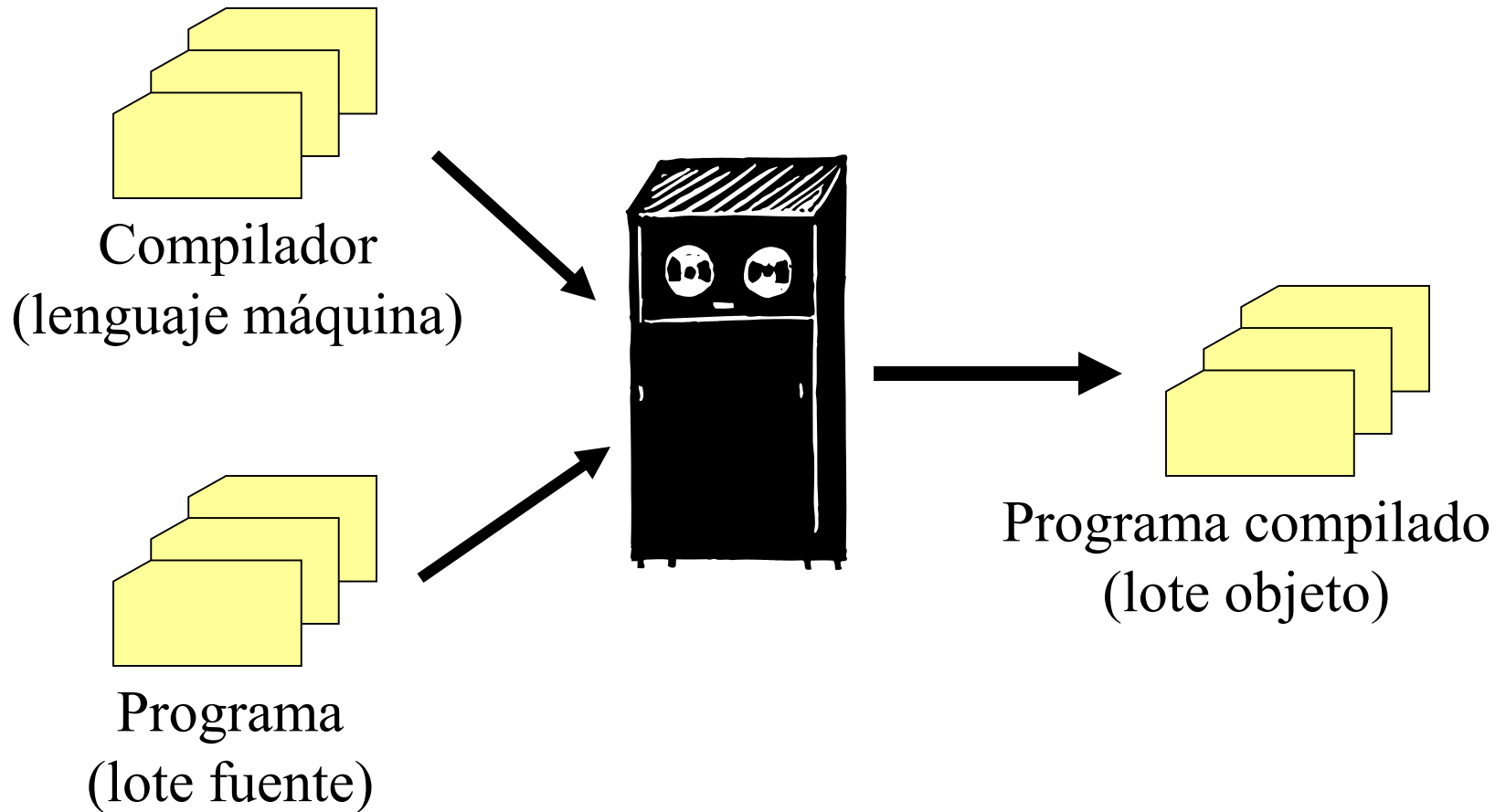
# Evolución – 2. Por lotes (batch)

Procesos por lotes (batch, 50's):

- Usuario  $\neq$  Operador
- Cintas magnéticas o tarjetas perforadas
- Programas escritos en ensamblador o un lenguaje de alto nivel (FORTRAN)
- Se basa en el uso de un “monitor” no residente totalmente, que gestionaba el problema de la planificación
- El tiempo de inicio se reducía para trabajos similares

## Evolución – 2. Por lotes (batch)

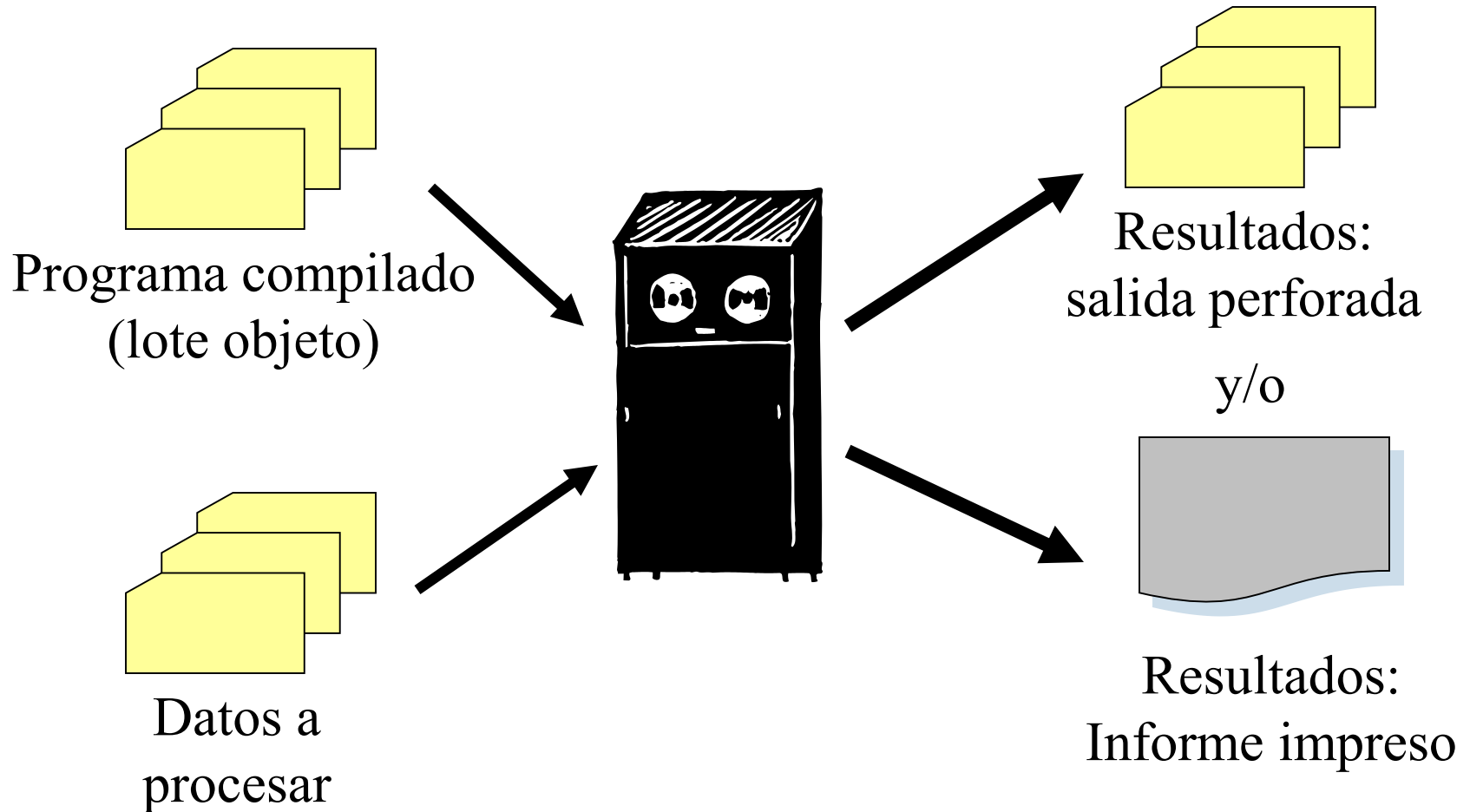
Fase 1: Compilación





## Evolución – 2. Por lotes (batch)

Fase 2: Ejecución



## Evolución – 2. Por lotes (batch)

¿Cómo distingue el monitor ...?

- ¿datos o programas?
- ¿programa escrito en ensamblador o Fortran?

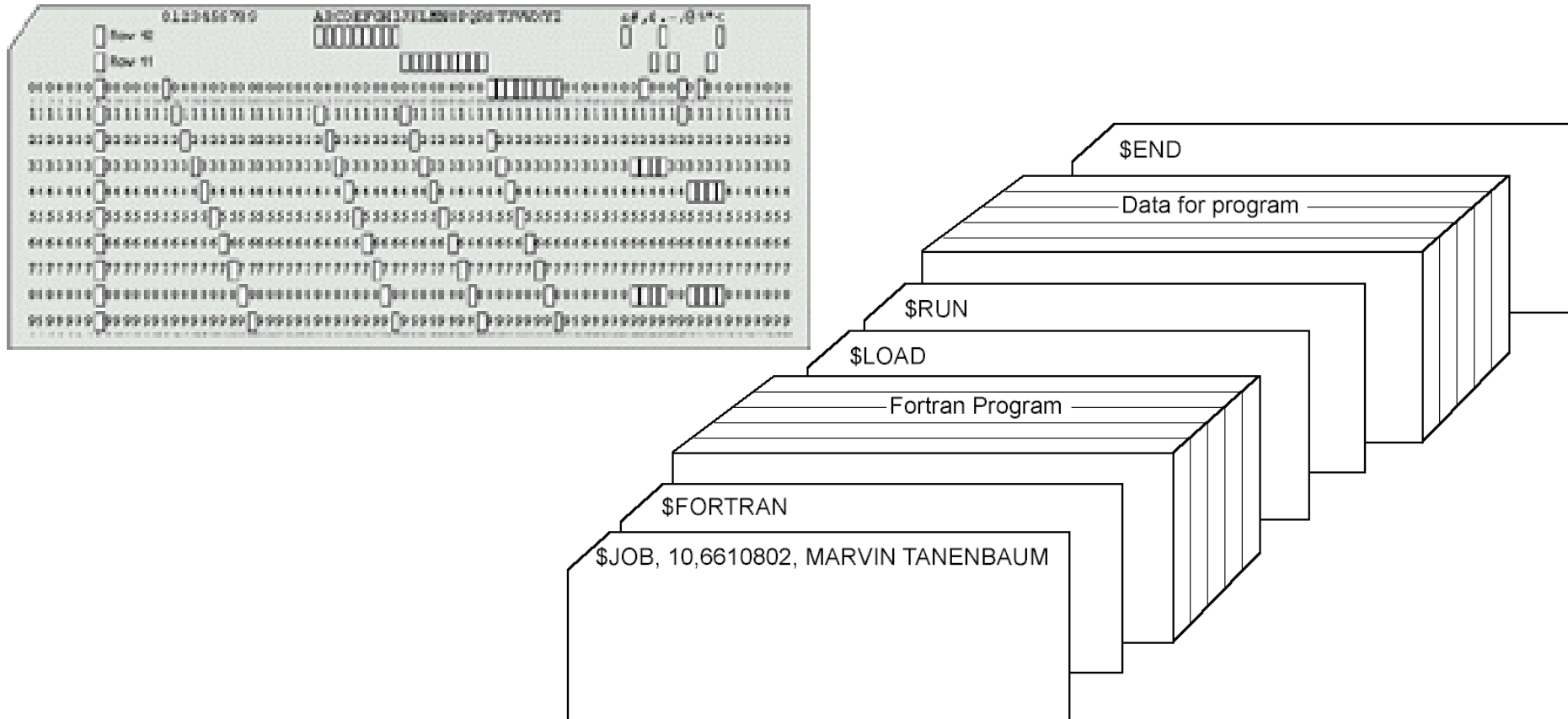


Solución: incluir unas tarjetas de control  
(JCL, Job Control Language):

- \$JOB ⇒ Inicio de un trabajo (programa)
- \$FTN ⇒ Programa en Fortran
- \$LOAD ⇒ Cargar el programa en memoria
- \$RUN ⇒ Ejecutar el trabajo
- \$DATA ⇒ Inicio de datos
- \$END ⇒ Fin del trabajo

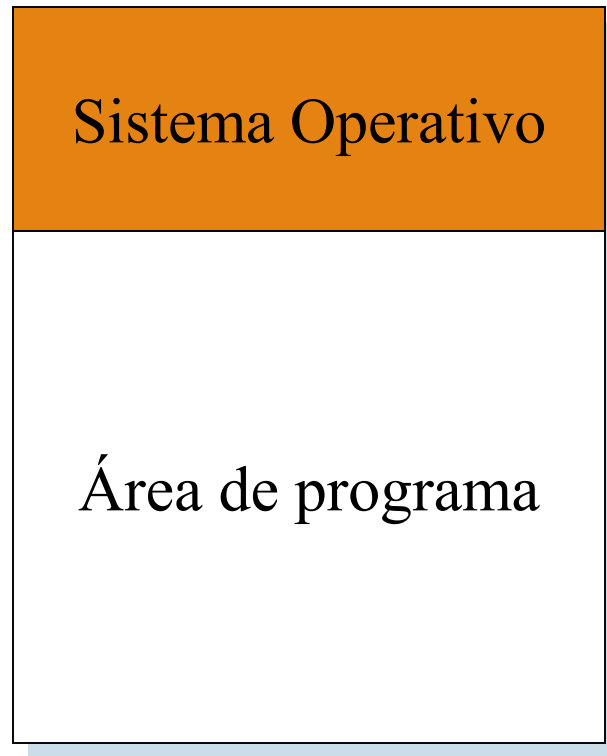
# Evolución – 2. Por lotes (batch)

## Ejemplo de trabajo por lotes



# Evolución – 2. Por lotes (batch)

Mapa de memoria:



# Evolución – 2. Por lotes (batch)

El monitor se encarga de:

- Leer e interpretar las tarjetas perforadas
- Cargar programas de sistema (compiladores) y de usuario en la memoria
- Comunicarse con los dispositivos E/S (drivers)

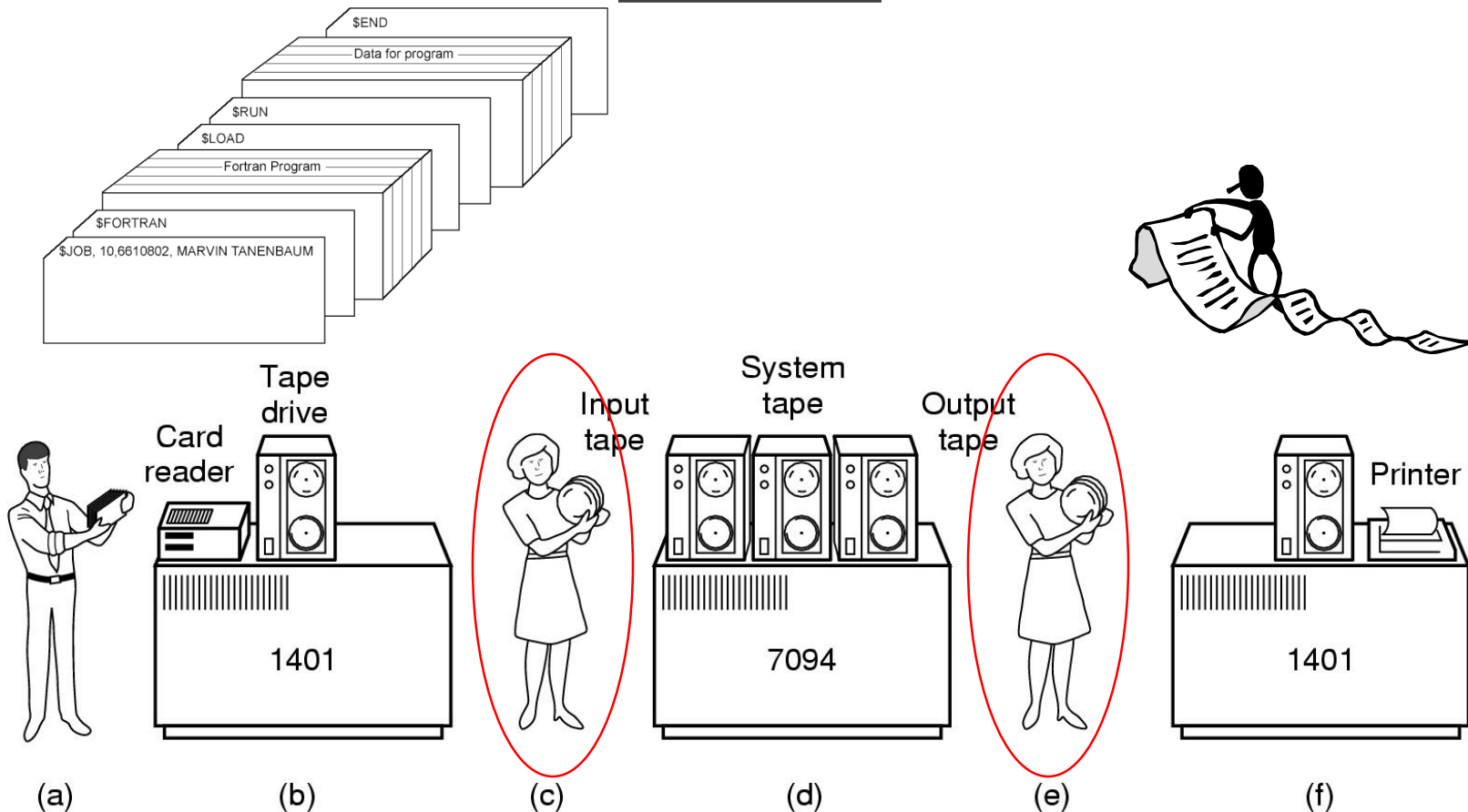
Problemas:

- Depuración difícil: volcados post-mortem
- Bajo rendimiento, ya que no se pueden solapar las operaciones E/S con CPU.

Solución: técnicas “off-line”.

# Evolución – 2. Por lotes (batch)

Técnicas “off-line” ⇒ Sistemas ejecutivos totalmente residentes



# Evolución – 2. Por lotes (batch)

Los sistemas ejecutivos evolucionaron:

- Contabilización del uso del sistema
- Límites temporales de procesamiento
- Protección ...

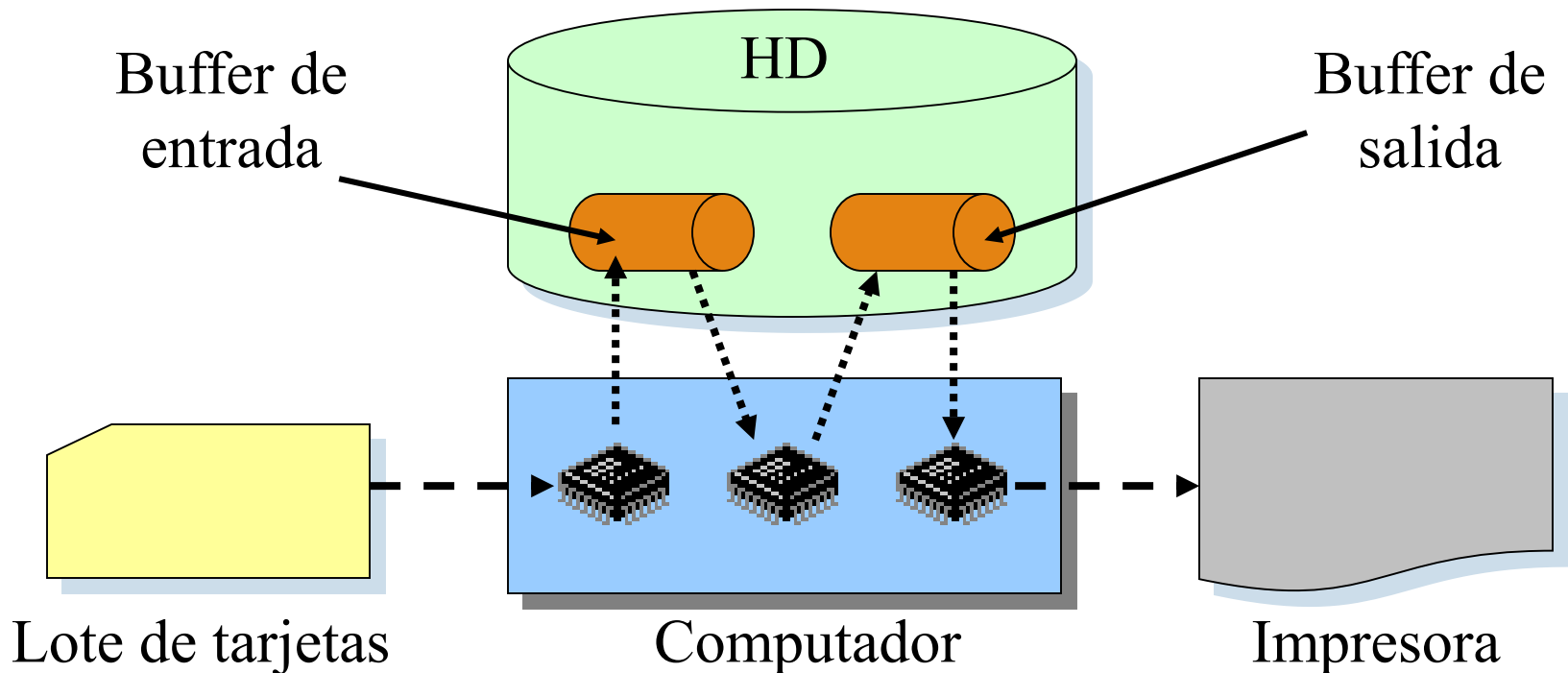
Mantenían dos inconvenientes:

- Secuencialidad estricta (cinta magnética)
- Mucha utilización de CPU (cintas magnéticas)

Solución: procesadores auxiliares (canales DMA)  $\Rightarrow$  SPOOL

## Evolución – 2. Por lotes (batch)

SPOOL (Simultaneous Peripheral Operating On-Line)





# Porcentaje de utilización de la CPU

Leer un registro de un archivo	0,0015 segundos
Ejecutar 100 instrucciones	0,0001 segundos
Grabar un registro en un archivo	0,0015 segundos
<b>Total</b>	<b>0,0031 segundos</b>

Porcentaje de utilización de la CPU =  $0,0001 / 0,0031 = 0,032$

El 3,2% del tiempo se utiliza la CPU el 96,8% esta ociosa.

# Evolución – 3. Multiprogramados

## Concepto de “multiprogramación”

- Idea básica: tener varios procesos en marcha para que el S.O. cambie a otro proceso cuando el que está en ejecución espere una op. E/S.

## Se comparte:

- Espacio de memoria de usuario
- Tiempo de procesador (multiplexación en tiempo)

Mapa de Memoria

Grado o factor de multiprogramación = 3



Sistema Operativo

Proceso 1

Proceso 2

Proceso 3

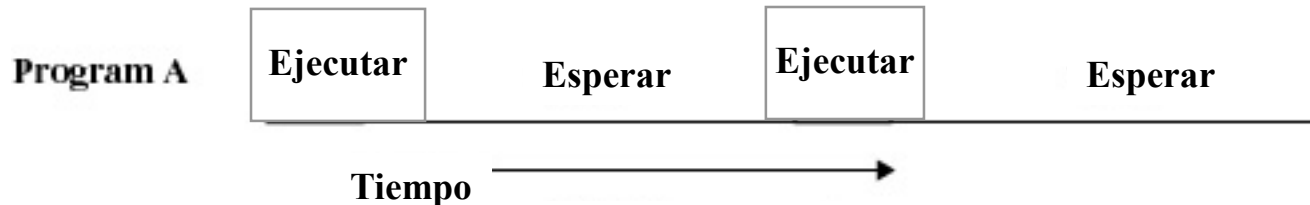
Todos los procesos están en memoria principal, en diferentes estado de ejecución

⇒ **Rápido intercambio de proceso**

- ❑ Grado o factor de multiprogramación  
 $P_{ES}$  = Probabilidad de un proceso en E/S
- ❑ Utilización procesador con 1 solo proceso:  
Utilización de CPU =  $1 - P_{ES}$
- ❑ Utilización procesador con “n” procesos:  
Utilización de CPU =  $1 - (P_{ES})^n$

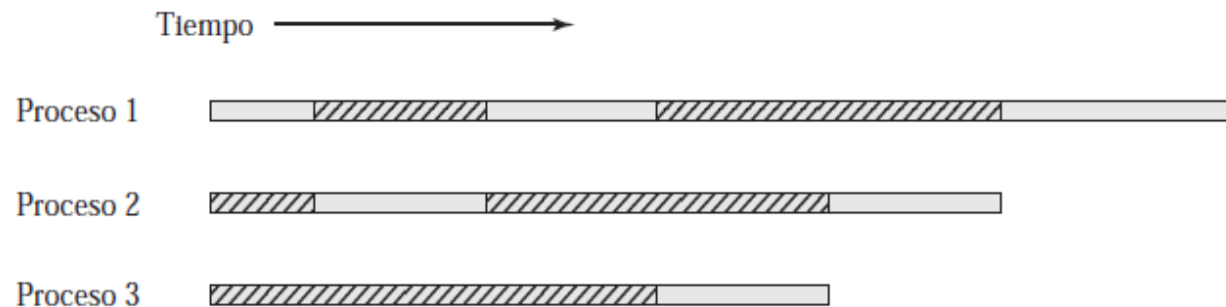
# Monoprogramación

- Antes de continuar, el procesador debe esperar hasta que la instrucción de E/S termine.

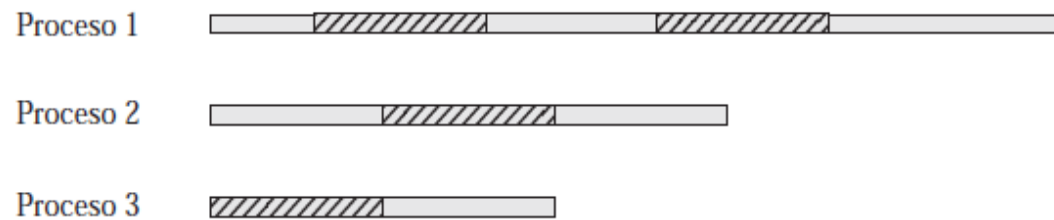


(a) Monoprogramación

# Multiprogramación y multiproceso



(a) Intercalado (multiprogramación, un procesador)



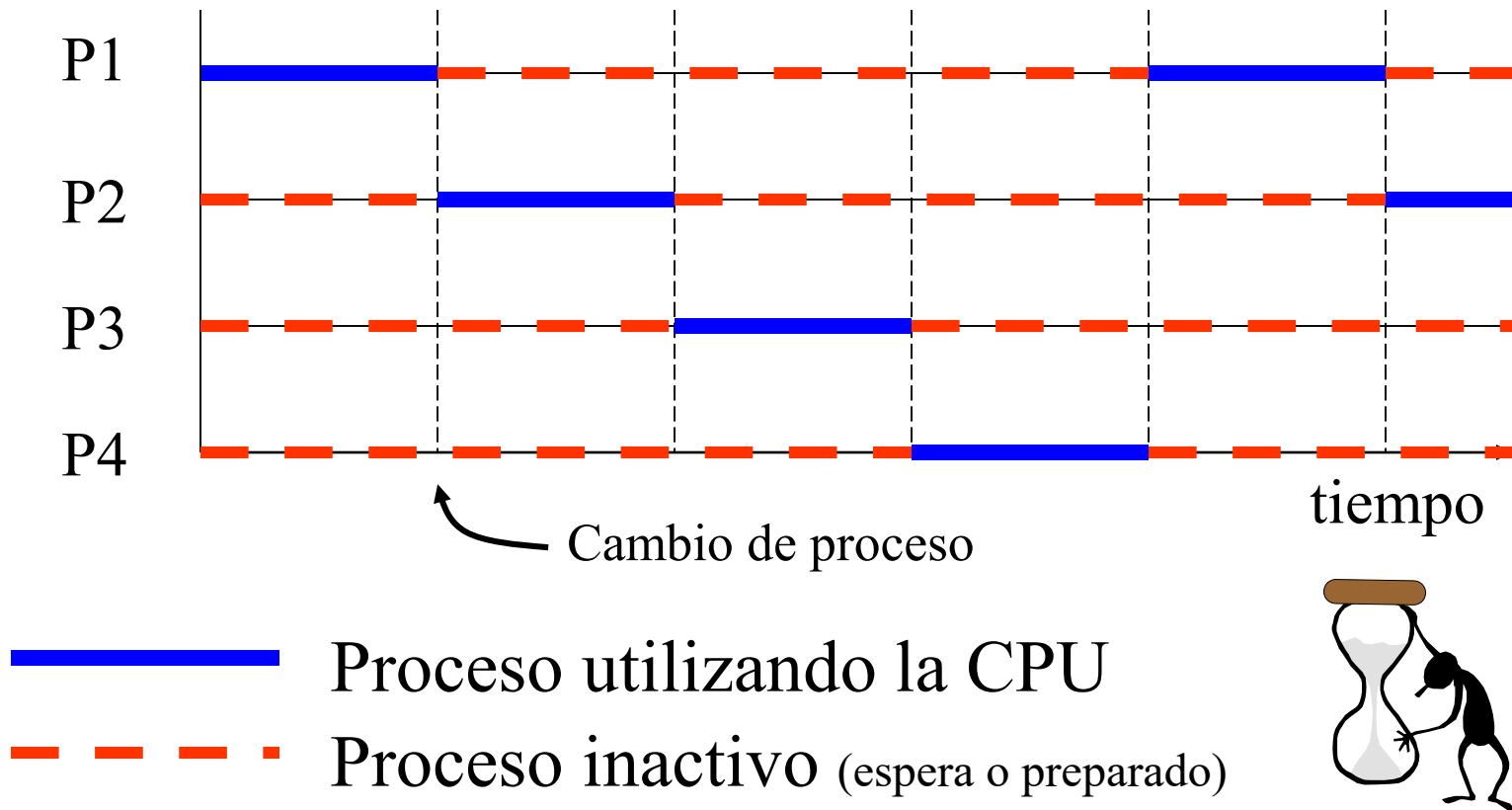
(b) Intercalado y solapamiento (multiproceso; dos procesadores)

▨ Bloqueado    □ Ejecutando

Figura 2.12. Multiprogramación y multiproceso.

# Evolución – 3. Multiprogramados

Se comparte tiempo de procesador:



# Ejemplo

	TRABAJO1	TRABAJO2	TRABAJO3
Tipo de trabajo	Cálculo intensivo	E/S intensiva	E/S intensiva
Duración	5 min.	15 min.	10 min.
Memoria exigida	50 K	100 K	80 K
¿Necesita disco?	No	No	Sí
¿Necesita terminal?	No	Sí	No
¿Necesita impresora?	No	No	Sí

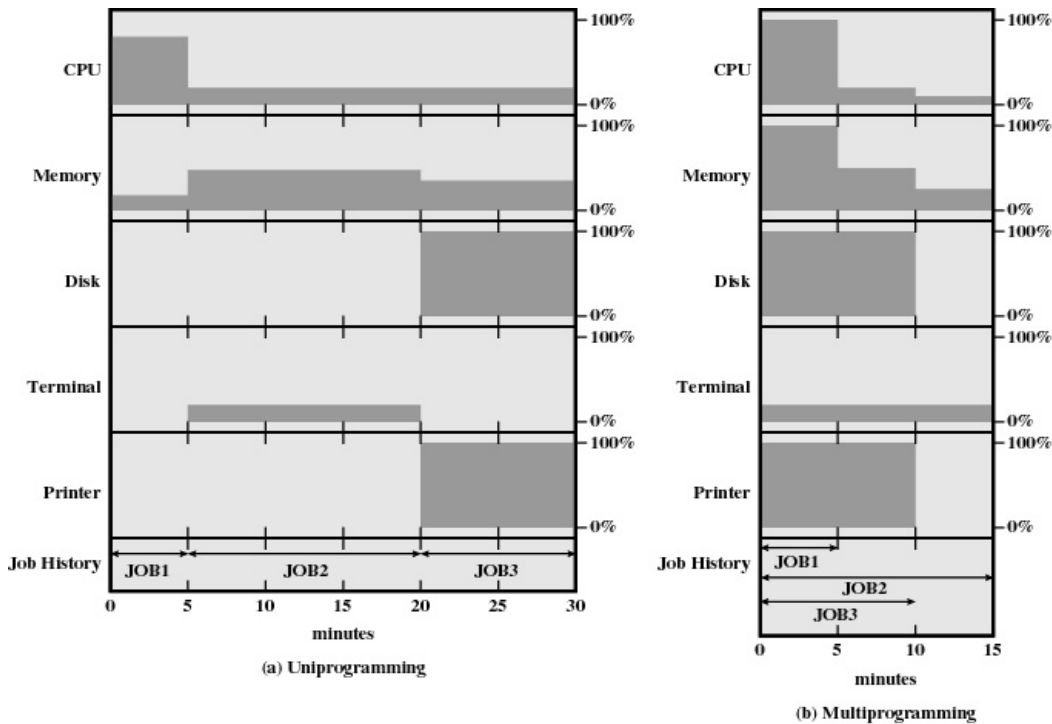


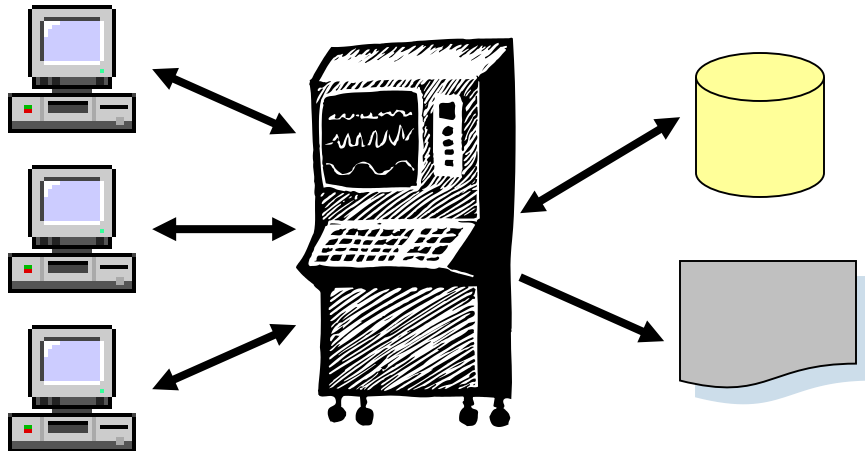
Figure 2.6 Utilization Histograms

	Monoprogramación	Multiprogramación
Uso del procesador	22%	43%
Uso de la memoria	30%	67%
Uso del disco	33%	67%
Uso de la impresora	33%	67%
Tiempo transcurrido	30 min.	15 min.
Tasa de productividad	6 trabajos/hora	12 trabajos/hora
Tiempo medio de respuesta	18'3 min.	10 min.

# Evolución – 3. Multiprogramados

Se utiliza para:

- ¡Incremento de la utilización de la CPU y recursos!
- Soportar múltiples usuarios activos (servidor)
- Permitir programas interactivos (terminal)



# Evolución – 3. Multiprogramados

Se requiere:

- Gestión de recursos (memoria, dispositivos E/S)
  - ⇒ Mecanismos de protección (evitar interferencias)
- Planificación de procesos

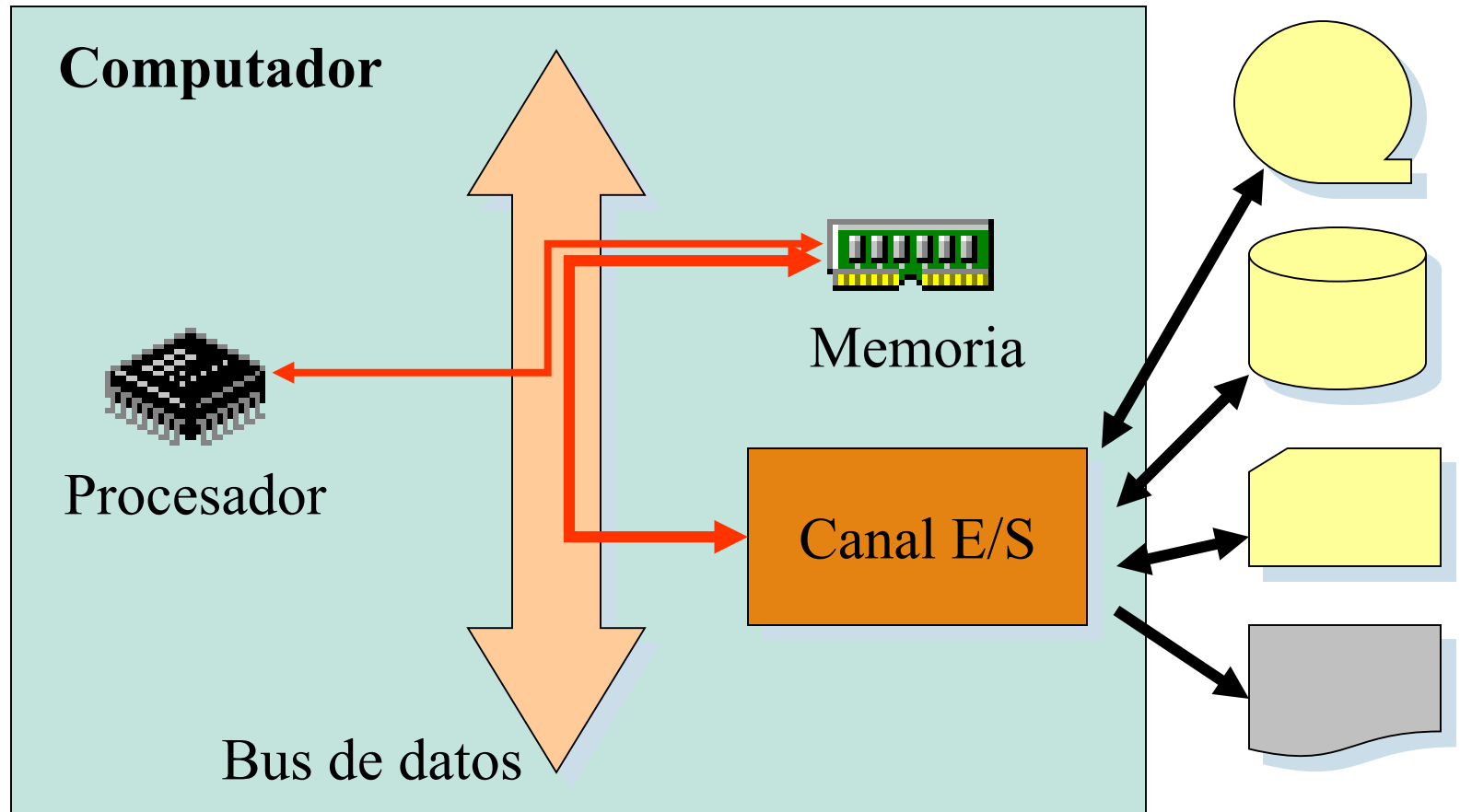
Características HW que lo favorecen:

- Interrupciones E/S
- Canales E/S autónomos (DMA): el procesador envía orden E/S y continúa con otro programa, mientras la E/S la efectúa el controlador de dispositivo



# Evolución – 3. Multiprogramados

Canales E/S autónomos (DMA)



# Evolución – 3. Multiprogramados

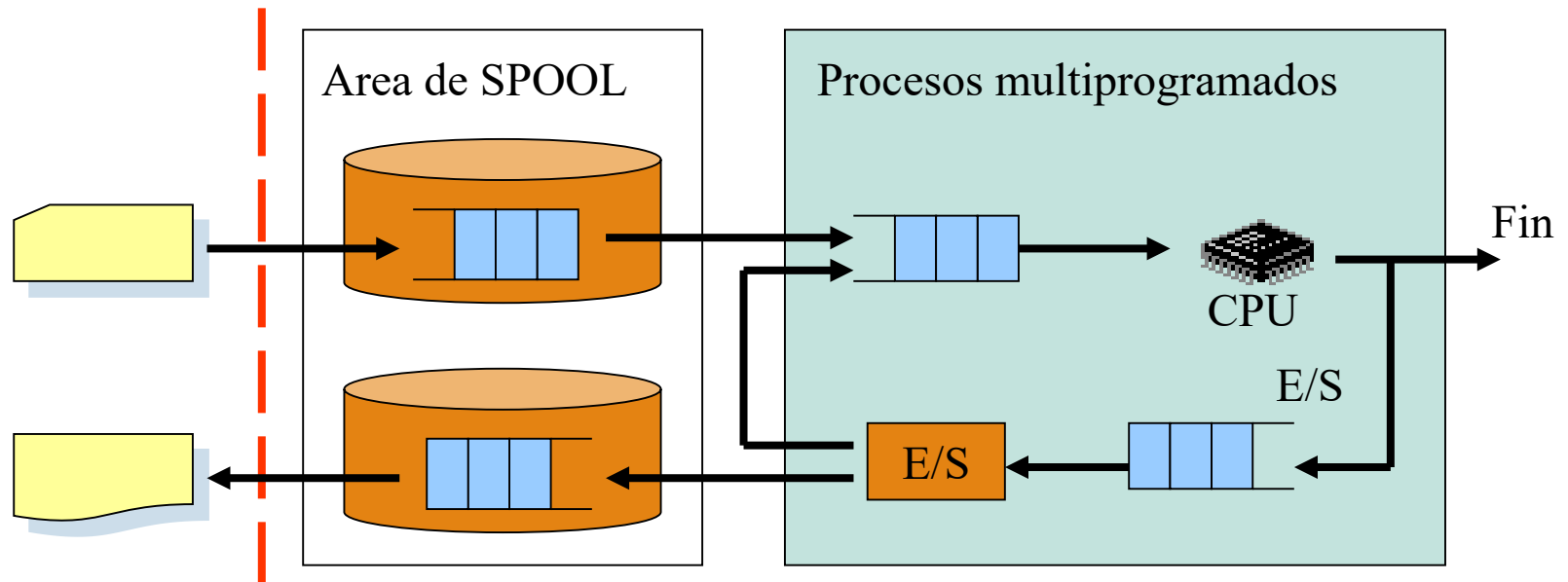
Tipos de sistemas multiprogramados:

- A) Sistemas por lotes multiprogramados
- B) Sistemas de tiempo compartido
- C) Sistemas operativos combinados
- D) Sistemas paralelos

# Evolución – 3. Multiprogramados

## A) Sistemas por lotes multiprogramados

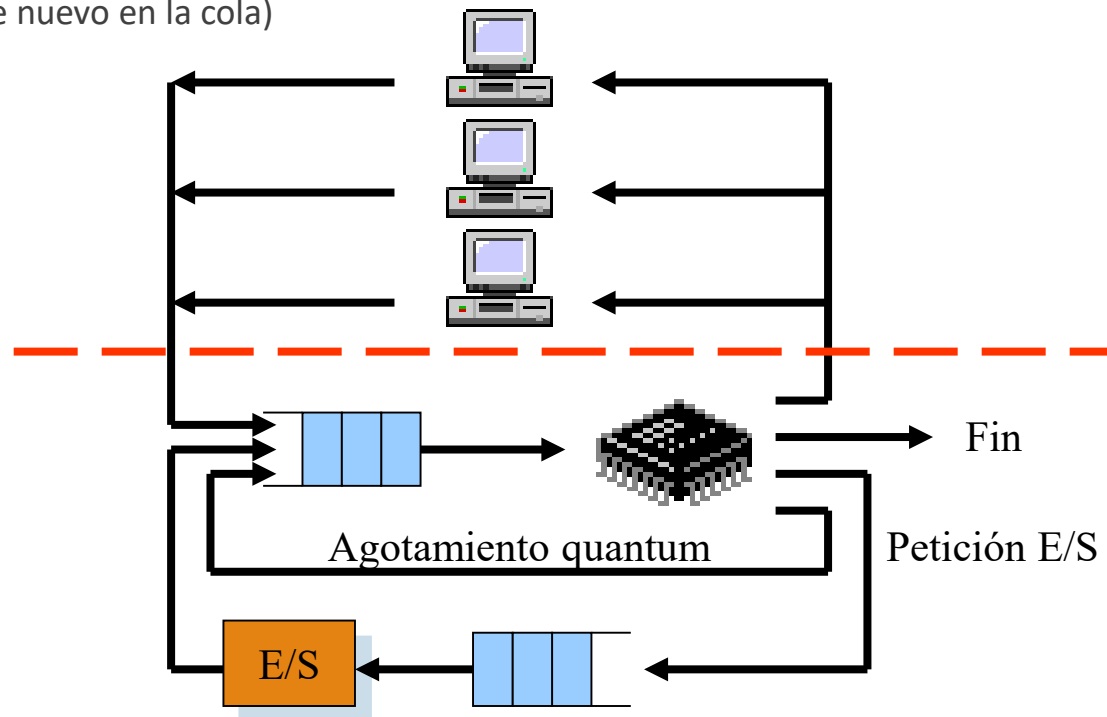
- Planificación de trabajos: mientras quede memoria libre se aceptan trabajos por lotes
- Planificación de CPU: el S.O. decide por el usuario cual de los procesos en cola ejecuta
- Se requiere mecanismos de protección



# Evolución – 3. Multiprogramados

## B) Sistemas de tiempo compartido

- Ejecución de varias tareas interactivas (shell)
- Multitarea y multiusuario (el S.O. intercala su ejecución => ráfagas o quantums)
- El S.O. cede el control a otro proceso cuando:
  - Un proceso termina su ejecución
  - Un proceso necesita una operación E/S
  - El quantum se agota (entra de nuevo en la cola)



# Multiprogramación por lotes vs. Tiempo compartido

	<b>Multiprogramación por lotes</b>	<b>Tiempo compartido</b>
Objetivo principal	Maximizar la utilización del procesador	Minimizar tiempo de respuesta
Origen de las instrucciones al sistema operativo	Instrucciones de un lenguaje de control de trabajos incluidas en el trabajo	Órdenes dadas en el terminal

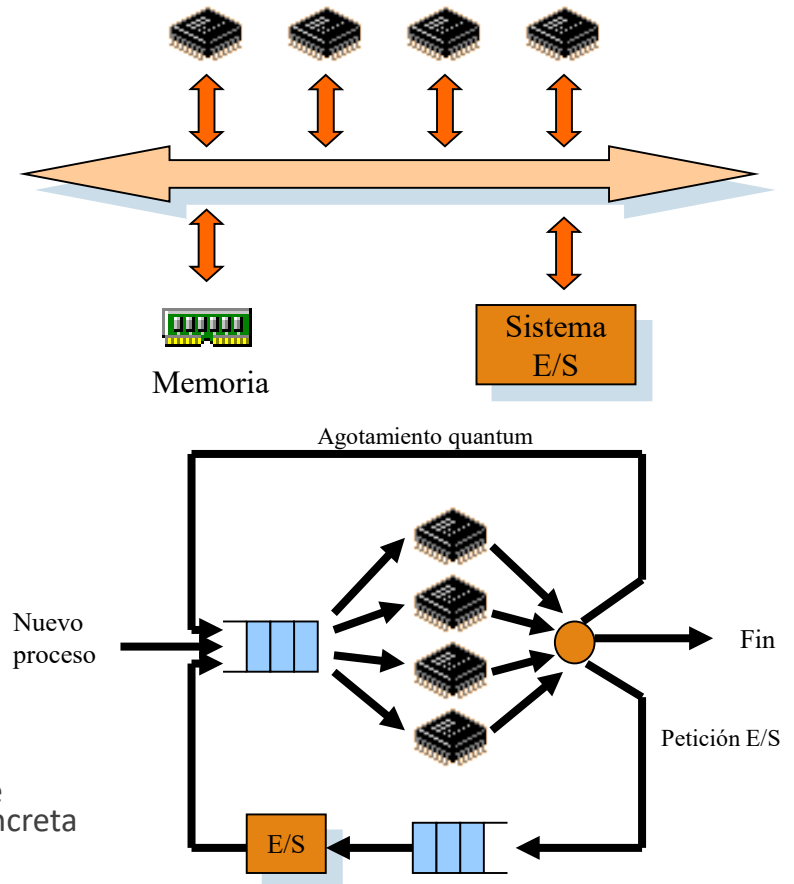
# Evolución – 3. Multiprogramados

## C) Sistemas operativos combinados

- Puede que no encaje en un solo molde
- Ejemplo: Podría permitir ejecución de lotes con menor prioridad concurrentemente con tareas interactivas.

## D) Sistemas paralelos (multiproceso)

- Son sistemas fuertemente acoplados, basados en computadores con varios procesadores comunicados (multiprocesamiento)
- Se comparten ciertos recursos como reloj, memoria y otros dispositivos
- Ventajas:
  - Se aumenta el rendimiento (throughput)
  - Se aumenta la fiabilidad
  - Son económicos
- Multiproceso simétrico (SMP)
  - Cada procesador ejecuta una copia idéntica del SO
  - Ejecución de muchos procesos sin bajar rendimiento
  - La mayoría de los SO actuales lo soportan (Solaris, Windows NT, ...)
- Multiproceso asimétrico
  - Uno de los procesadores es el maestro, y controla al resto de procesadores esclavos, cada uno encargado de una tarea concreta
  - En grandes SO: SunOS



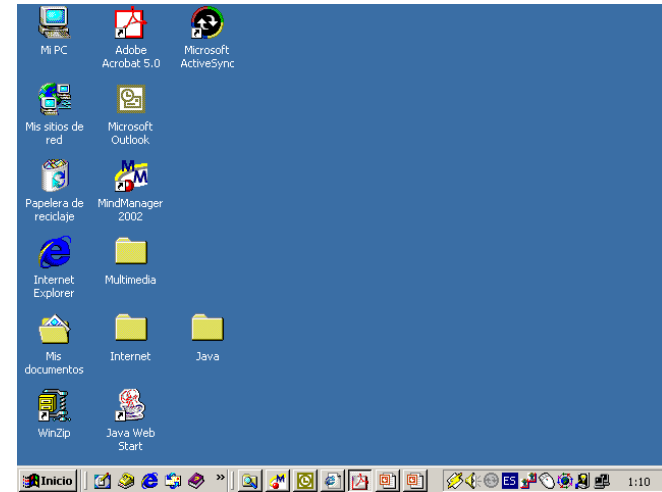
# Evolución – 4. Personales y red

Sistemas de ordenador personal y red:

- Ordenador personal (PC): son sistemas de computadores dedicados a un único usuario, de bajo coste
- Énfasis para facilitar su uso por el usuario y mejorar tiempos de respuesta (interactividad)
- Incorporan interfaces gráficas y dispositivos de E/S amigables (WIMP: Windows Icons Mouse Pulldown)

Ejemplos

- CP/M, MS-DOS: monousuario, monoprogramación, no protegido
- Macintosh, OS/2, Windows: multitarea, memoria virtual



# Evolución – 4. Personales y red

Estaciones de trabajo en red:

- Desarrollo de LAN (Ethernet y Token Ring)
- Servicios de comunicación integrados en el SO (protocolo TCP/IP: redes locales e Internet)
- Computación cliente/servidor
- Estaciones sin disco

Convergencia PC y estaciones en red

Los usuarios son conscientes de la existencia de otras máquinas y recursos



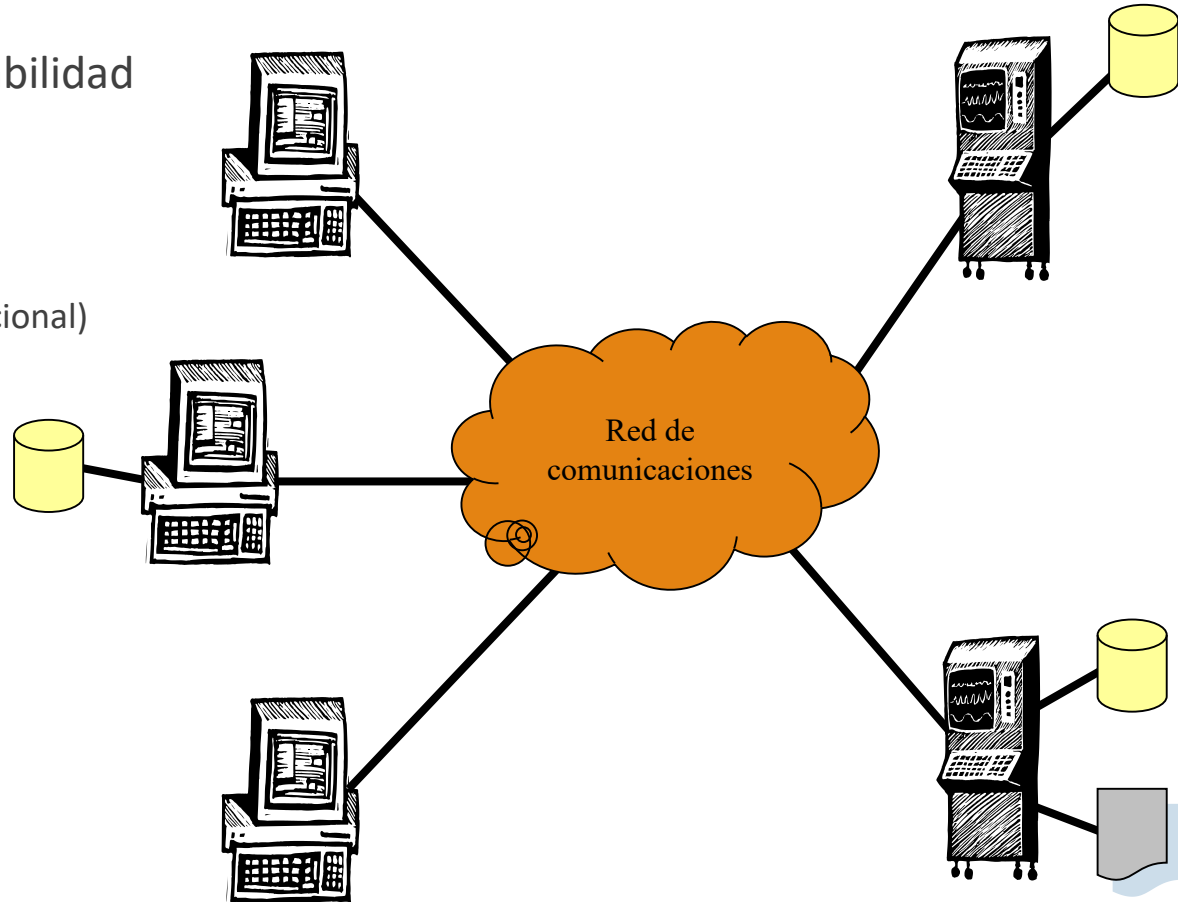


# Evolución – 5. Otros desarrollos

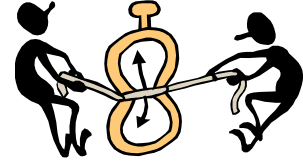


## Sistemas distribuidos

- Sistemas débilmente acoplados, son autónomos pero están bien comunicados, de forma transparente para el usuario
- Compartición de recursos
- Aumento del rendimiento y fiabilidad
- Servicios distribuidos:
  - Archivos
  - Bases de datos
  - Procesamiento (capacidad computacional)
  - Comunicaciones



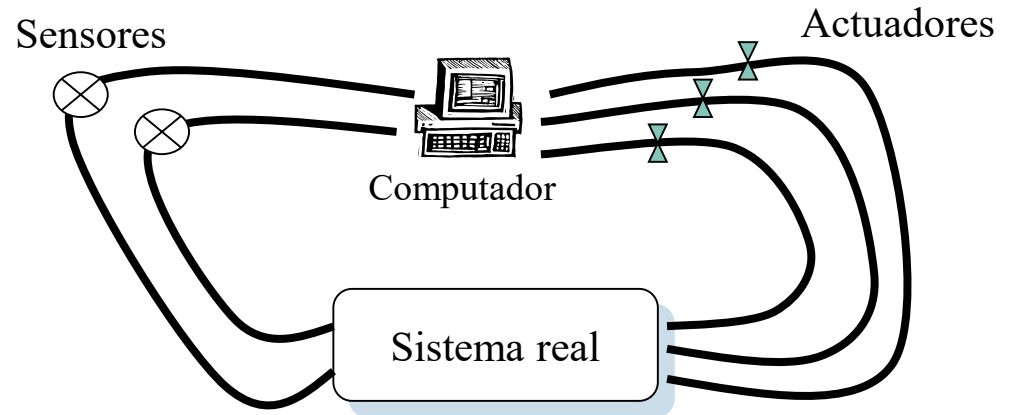
# Evolución – 5. Otros desarrollos



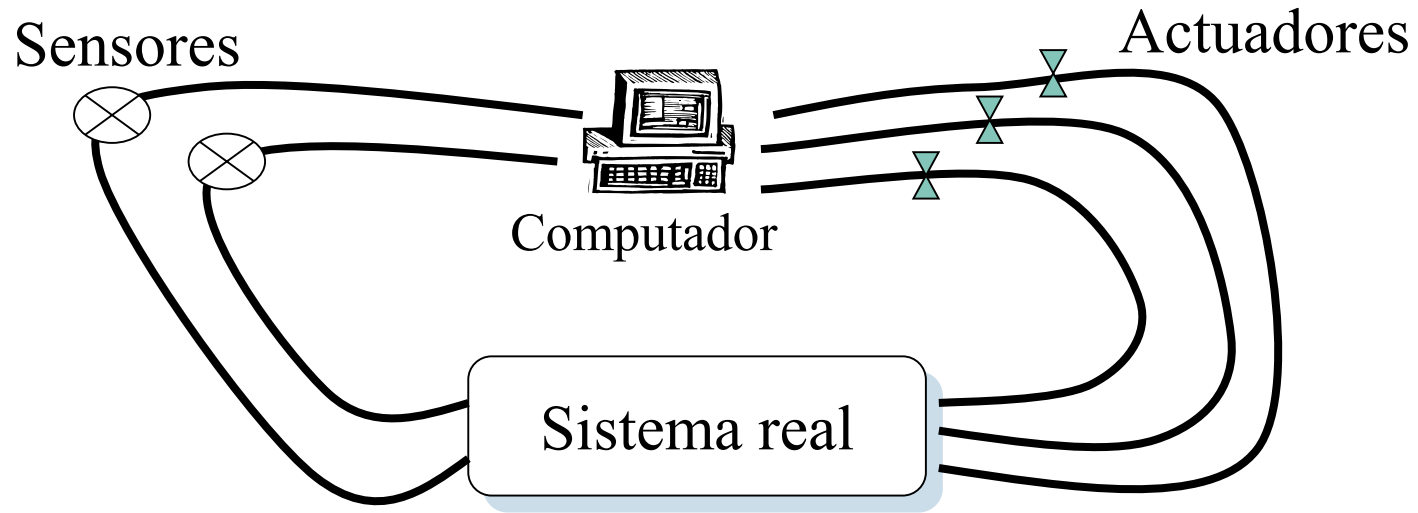
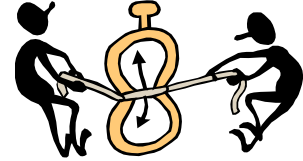
## Sistemas en tiempo real

- Existe un requerimiento fuerte de tiempo sobre las operaciones y el flujo de datos
- Se usa en sistemas de propósitos especiales:
  - Problemas de control de procesos industriales
  - Control de vuelo
  - Aplicaciones militares
- Cada proceso se encarga de controlar un único suceso externo  $\Rightarrow$  planificación expropiativa basada en prioridades

- La gestión de la memoria es más sencilla:
- Los procesos están continuamente en memoria
- Número de procesos no varía mucho



# Evolución – 5. Otros desarrollos



- La gestión de la memoria es más sencilla:
  - Los procesos están continuamente en memoria
  - Número de procesos no varía mucho

Ante los datos de la tabla adjunta, representar en un eje temporal la utilización de:

CPU, Memoria, Disco, Terminal e Impresora

Calcular:

La duración total,

El % de utilización de la CPU, Disco, e Impresora

El % medio de consumo de memoria.

La tasa de productividad

Y el tiempo medio de respuesta

Para los casos de monoprogramación y multiprogramación

Nota: los porcentajes de utilización de CPU de los procesos no tienen en cuenta la sobrecarga que supone realizar E/S a Disco

**Tipo de SO : Por Lotes Multitarea sin memoria virtual ni DMA**

	Trabajo 1	Trabajo2	Trabajo3
Tipo	Calculo	Calculo	E/S
<b>CPU (%)</b>	<b>30</b>	<b>50</b>	<b>15</b>
<b>Duracion (Min)</b>	<b>5</b>	<b>20</b>	<b>25</b>
<b>Memoria Kb</b>	<b>50</b>	<b>125</b>	<b>12</b>
<b>Disco</b>	<b>Si</b>	<b>No</b>	<b>Si</b>
<b>Impresora</b>	<b>No</b>	<b>No</b>	<b>Si</b>
<b>Memoria Monitor Kb</b>	<b>100</b>		
<b>Memoria Principal Kb</b>	<b>640</b>		
<b>Sobrecarga CPU (%) Gestion E/S a Disco</b>	<b>5</b>		
<b>Sobrecarga Cambio Procesos</b>	<b>0</b>		

## Puntos a recordar

---

CAPÍTULO 1: FUNDAMENTOS DE LOS SISTEMAS  
OPERATIVOS

# Puntos a recordar

Un S.O. es un programa que se encarga de simplificar el manejo y la utilización de la computadora haciéndola segura y eficiente

Las funciones del sistema operativo se agrupan en tres categorías:

- Gestión de los recursos de la computadora
- Ejecución de servicios para los programas
- Ejecución de programas de usuario

Un sistema operativo está formado por tres grandes bloques:

- El núcleo
- La capa de servicios o llamadas al sistema
- El intérprete de comandos o shell

Los servicios se suelen agrupar según su funcionalidad en varios componentes:

- Gestión de procesos
- Gestión de memoria
- Gestión de E/S
- Gestión de archivos y directorios
- Comunicación y sincronización de procesos
- Seguridad y Protección

El iniciador ROM es el programa de arranque que se encuentra en memoria ROM y cuyas funciones son:

- Comprobar el sistema
- Leer y Almacenar en la memoria el cargador del S.O.
- Ceder el control al cargador del S.O.

El cargador del S.O. tiene por misión traer a memoria algunos componentes del S.O.

En un sistema monolítico todos sus componentes se encuentran integrados en un único programa que ejecuta en un único espacio de direcciones. Todas las funciones que ofrece se ejecutan en modo kernel.

Un sistema por capas se organiza como una jerarquía de capas donde cada una ofrece un interfaz clara y bien definida a la superior y solamente utiliza los servicios de la inferior y los suyos propios.

En un sistema cliente/servidor la mayor parte de los servicios y funciones del sistema se implementan en procesos de usuario dejando solo una pequeña parte en modo núcleo.

# Puntos a recordar

Un proceso es un programa en ejecución. La información que compone un proceso es la siguiente:

- La imagen en memoria
- El estado del procesador
- El contenido del PCB (Process Control Block)

El PCB es una estructura de datos que almacena distinta información del proceso

Los sistemas operativos ofrecen servicios para crear, ejecutar y matar procesos.

Un S.O. monotarea o monoproceso solo permite que exista un proceso en cada instante.

Un S.O. multitarea o multiproceso permite que coexistan varios procesos activos simultáneamente.

Un sistema monousuario solo permite dar soporte a un usuario

Un sistema multiusuario da soporte a varios usuarios simultáneamente desde terminales distintos o vía red. Es obligatorio que sea multitarea.

El planificador o scheduler es el elemento del sistema operativo que se encarga de seleccionar el proceso que se ha de ejecutar a continuación.

El activador o dispatcher es el elemento que se encarga de poner en ejecución el proceso elegido por el scheduler

El gestor de memoria se encarga de asignar memoria a los procesos para crear su imagen en memoria, proporcionar memoria cuando un proceso lo solicita y liberarla cuando no la quieran. Trata los errores de acceso a memoria, permite la compartición de memoria y gestiona las tablas de páginas y fallos de página en memoria virtual.

El gestor de E/S controla el funcionamiento de los dispositivos de E/S proporcionando una interfaz que permite utilizar los periféricos de forma sencilla y uniforme ofreciendo mecanismos de protección para impedir acceso sin control.

El servidor de archivos ofrece al usuario una visión lógica compuesta por una serie de objetos (archivos y directorios) identificados por nombre lógico sobre los que se puede realizar una serie de operaciones. La visión física incluye los detalles de cómo están almacenados estos objetos en los periféricos

El sistema de archivos es el conjunto de archivos incluidos en una unidad de disco. Está compuesto por los datos de los archivos y metainformación. La metainformación es toda aquella que necesita el sistema de archivos para representar los archivos y directorios.

# Puntos a recordar

El S.O. ofrece mecanismos de comunicación que permiten transferir cadenas de bytes entre procesos. También ofrece mecanismos de sincronización que permiten que los procesos se bloqueen y se despierten.

La seguridad en un sistema operativo se encarga de garantizar la identidad de los usuarios (autenticación) de definir lo que pueden hacer (por privilegios)

Los privilegios en un S.O. especifican los recursos a los que puede acceder cada usuario.

Una lista de control de acceso especifica los usuarios y grupos que pueden acceder a un recurso.

Una capacidad asocia a cada usuario o grupo la lista de recursos a los que puede acceder.

El S.O. es un servidor que se activa con las llamadas al sistema emitidas por los procesos mediante funciones de librería, por las interrupciones emitidas por periféricos y hardware y con las condiciones de excepción o error del hardware.

La activación del sistema operativo sólo puede realizarse mediante el mecanismo de las interrupciones, por tanto las llamadas al sistema deben implementarse mediante una instrucción TRAP.

La principal función del interfaz de usuario es permitir acceder y manipular los objetos y recursos del sistema.

Existen dos tipos de interfaces de usuario del sistema: alfanuméricas y gráficas.

POSIX es el estándar de interfaz de usuario para sistemas portables definido por IEEE

Win32 define los servicios ofrecidos por los sistemas operativos de Microsoft.

Los primeros S.O. (años 50) se caracterizaban por la ejecución de trabajos en lotes.

Un sistema operativo de tiempo compartido se encarga de repartir el tiempo de CPU entre los distintos usuarios asignando de forma rotativa intervalos de tiempo (quantums) a los procesos de los usuarios.

Un sistema operativo distribuido aparece al usuario como un único sistema centralizado.

Un middleware es una capa de software que se ejecuta sobre un sistema operativo y existente y que se encarga de gestionar un sistema distribuido de la misma forma que lo hace un sistema operativo distribuido.



Fin

---

CAPÍTULO 1: INTRODUCCIÓN A LOS SISTEMAS  
OPERATIVOS