

INTRODUCCION A SISTEMAS OPERATIVOS



Objetivos de un Sistema Operativo

- Proveer la interfase entre el usuario y la máquina para lograr un uso conveniente y eficiente
- Administrar los recursos (concepto de “programa de control”)

Evolución de los sistemas operativos

Primeros sistemas:

- El programador es el operador (hands-on)
- Existe mucha interacción pero la CPU estaba desperdiciada
- Aparecen las bibliotecas de drivers
- Aparecen los compiladores

Monitor simple o batch sencillo

- Preparación previa de los trabajos a ejecutar (secuenciamiento de los trabajos)
- Existe un monitor residente que se encarga de realizar la transición al próximo trabajo
- Aparecen las instrucciones privilegiadas (modo Maestro/esclavo o usuario/supervisor) para evitar que los programas interfieran con, por ejemplo, la E/S de otros programas

Batch sofisticado

- Multiprogramación
- Se busca mejorar los tiempos desperdiciados de CPU:
 - técnica de buffering (adelantar las lecturas o escrituras mientras la CPU procesa el dato anterior)
 - operación off-line: se traslada la información de periféricos lentos a aquellos más veloces para optimizar los tiempos de ingreso al sistema
 - spooling: reemplaza los sistemas off-line por la virtualización de los periféricos lentos sobre periféricos rápidos

Time Sharing (tiempo compartido)

- Se utiliza la multiprogramación y la planificación de la CPU para que cada usuario tenga una parte pequeña de tiempo de CPU (intercambio de contexto)
- Cada usuario cree que tiene el sistema para él solo
- Se pierde la interacción entre el usuario y la máquina y por lo tanto aparecen los problemas de prever todos los errores de antemano y el debug de los programas es estático y se realiza a través del análisis de los vuelcos de memoria (dumps)

Sistemas de Tiempo Real

- Utilizan un dispositivo de control (un sensor) para una aplicación dedicada
- En base a los datos captados por el sensor el sistema realiza los ajustes
- Es **crítico** el tiempo de respuesta!!

Multiprocesamiento

- Existen varias CPUs interconectadas
- Usualmente una realiza las funciones de coordinador y las otras son de tipo dedicado
- La CPU coordinadora suele ser un equipo principal o Mainframe (host)
- Cuando se trata de diversas computadoras interconectadas por una red en las que se distribuyen las funciones coordinandolas a través de un único sistema operativo estamos en presencia de un **Sistema Operativo Distribuido**

Servicios que brinda un S.O.

- Ejecución de los procesos
- Realización de las operaciones de E/S
- Manejo del sistema de Archivos
- Detección de errores
- Administración y asignación de los recursos a los procesos (memoria, tiempo de CPU, periféricos, archivos)
- Accounting: contabilización de lo que realiza
- Protección entre procesos

Estructura de Sistemas Operativos

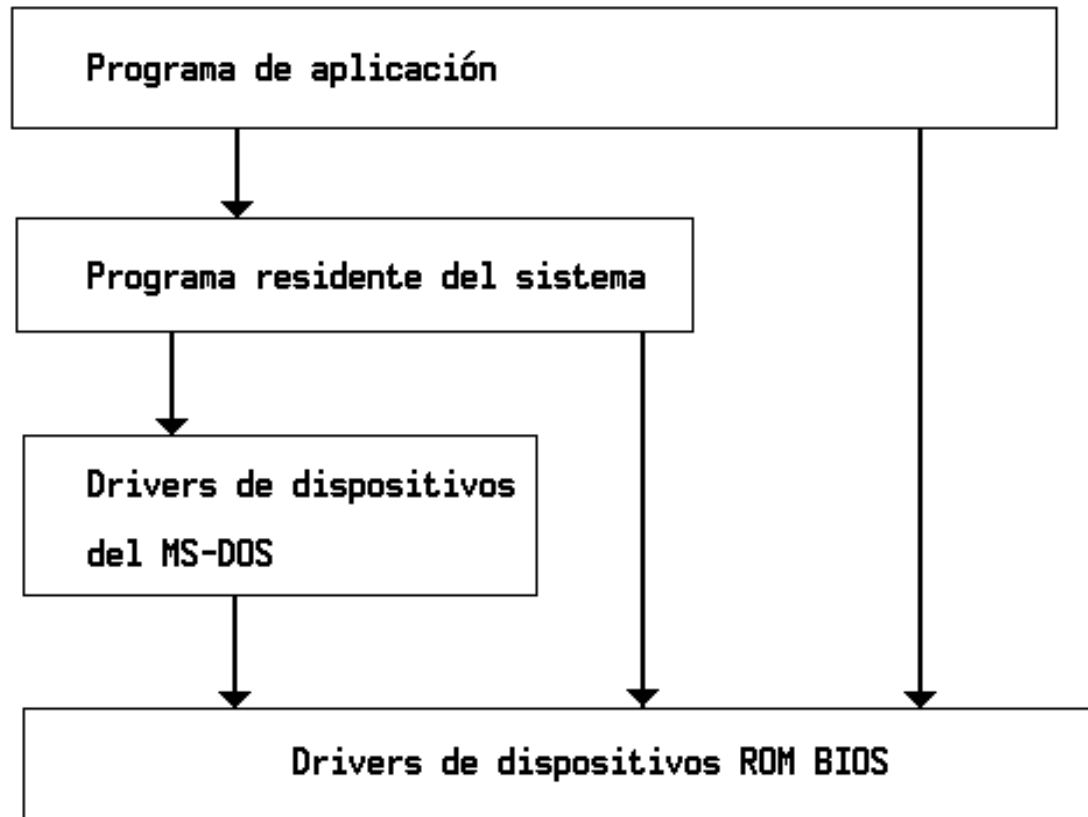
Se divide en módulos con interfases bien definidas

Cada módulo tiene su función, inputs y outputs cuidadosamente definidos

Pero hay S.O. que no están bien diseñados.

Veamos algunos ejemplos

DOS (disk operating system)



Permite el acceso de programas de aplicación a las rutinas básicas de E/S

UNIX (primeras versiones)

usuarios		
shells y comandos. compiladores e intérpretes		
bibliotecas del sistema		
interfaz de llamadas del sistema al kernel		
señales	sist. de archivos	administración de CPU
manejo de terminales	swapping	reemplazo de páginas
sist. de E/S de caracteres	sist. de E/S de bloques	pedido de páginas
drivers de terminales	drivers de cintas y discos	memoria virtual
interfaz del kernel al hardware		
controladores de terminales	controladores de discos y cintas	controladores de memoria
terminales	discos y cintas	memoria física

K
E
R
N
E
L

El kernel junta demasiada funcionalidad (drivers e interfases)

Mejor es AIX (S.O. De IBM) que divide el kernel en dos

Diseño en capas

Cada capa se construye sobre la anterior y tiene una interfase bien definida.

La capa 0 es el hardware y la N es la interfase de usuario.

Una capa solo puede invocar las funciones de las capas inferiores.

Ventajas:

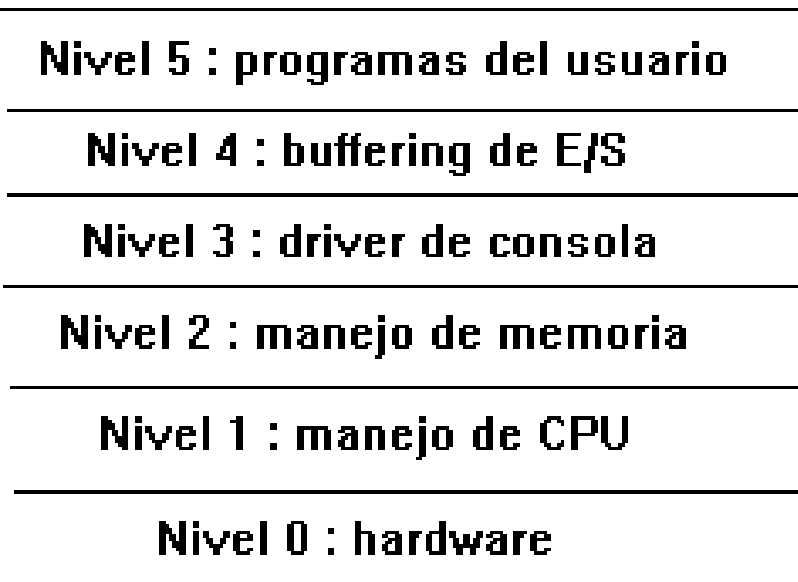
- Facilita la depuración de cada capa
- Cada capa no necesita conocer cómo están implementadas las funciones en las otras capas del modelo solo conoce la interfase de comunicación

Ejemplo:

Usuarios	7
Shell (comandos + JCL)	6
Planificador de Trabajos	5
Administración de archivos	4
Administración de Periféricos	3
Administración de Memoria	2
Núcleo (Adm. Procesador + Semáforos)	1
Hardware	0

THE (Technische Hogeschool Eindhoven)

Fue el primer S.O. diseñado en capas



Otro sistema operativo diseñado en capas: el VENUS con 7 capas

Sistema Operativo OS/2 (IBM)

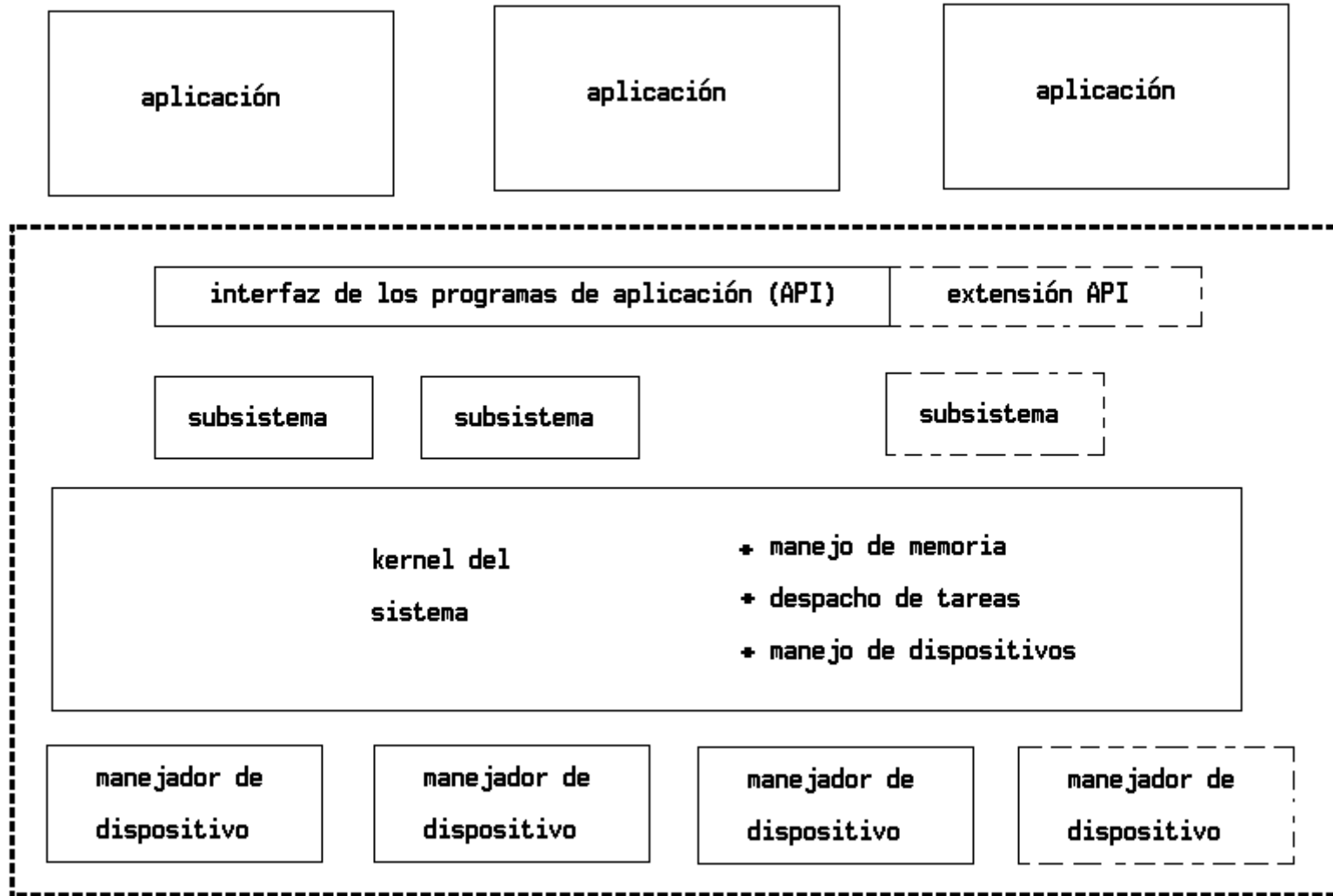


Diagrama de Estados de un S.O. (preview)

