

# Bloque I: Principios de sistemas operativos



## **Tema 1. Principios básicos de los sistemas operativos**

Tema 2. Concurrencia

Tema 3. Ficheros

Tema 4. Sincronización y programación dirigida por eventos

Tema 5. Planificación y despacho

Tema 6. Sistemas de tiempo real y sistemas empujados

Tema 7. Gestión de memoria

Tema 8. Gestión de dispositivos de entrada-salida

## Notas:



### **Tema 1. Principios básicos de los sistemas operativos**

- Concepto de sistema operativo.
- Evolución de los sistemas operativos.
- Estructura de los sistemas operativos.
- Concepto de programa y proceso.
- Concurrencia: gestión de procesos y *threads*.
- Gestión de memoria.
- Ficheros y el sistema de ficheros.
- Entrada/salida.
- Sistemas multiusuario y protección.

# 1. Concepto de sistema operativo

Controla el uso por parte de los programas de aplicación de todos los recursos del computador: memoria, CPU, unidades de entrada y salida

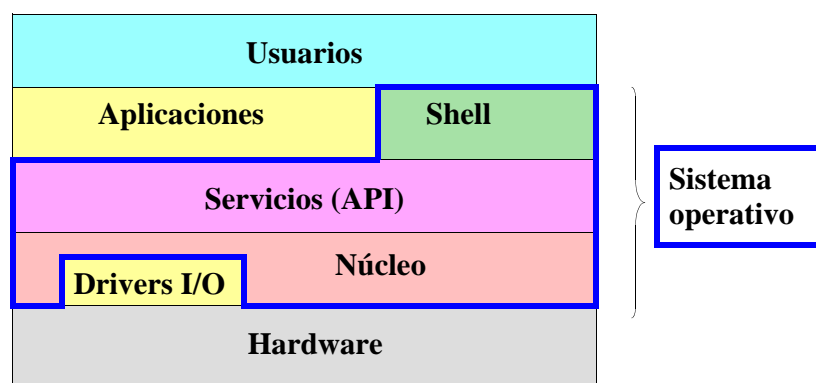
Independiza al programa de aplicación del hardware

Proporciona comunicación con otros computadores

Ejecuta servicios para los programas o aplicaciones

Ejecuta órdenes de los usuarios

# Niveles del sistema operativo



# Gestión de recursos

---

## Asignación de recursos

- memoria, disco, dispositivos de entrada salida
- CPU (planificación)

## Protección

- Entre usuarios y aplicaciones
- Entre cada aplicación y el sistema operativo
- Entre cada aplicación y el hardware

## Contabilidad

- A efectos de gestión de coste
- A efectos de calidad de servicio

# Servicios del sistema operativo

---

## Concurrencia

- procesos
- threads
- sincronización

## Ficheros

## Entrada/Salida (incluyendo comunicaciones)

## Gestión de la memoria

## Servicios de tiempo real

- planificación, relojes, ...

# Arranque del computador

Fase	Operación
Iniciador ROM	Test del hardware
	Carga del cargador del SO
Cargador del SO	Carga en memoria de componentes del SO
Inicialización SO	Test del sistema de archivos
	Creación de estructuras de datos internas
	Arranque de procesos del sistema (demonios)
Funcionamiento normal	

## 2. Evolución de los sistemas operativos

Los **primeros computadores** no tenían sistema operativo

- programados en lenguaje máquina
- el programador hacía todo el software

Sistemas de **procesamiento por lotes** (1955-1965)



- se agrupaban varios trabajos en un lote, para agilizar el proceso

# Evolución de los sistemas operativos (cont.)



## La **multiprogramación** (1965-1980)

- Varias aplicaciones cargadas a la vez en memoria
- Cuando una aplicación está en espera, las demás pueden continuar
- **Spooling**: leer tarjetas al mismo tiempo que se procesan otras
- El tiempo compartido entre múltiples usuarios es una evolución natural de la multiprogramación
  - ejemplo: **CTSS**, MIT;
  - reduce drásticamente el tiempo de espera a un trabajo
  - se populariza sólo al haber hardware de protección
  - desemboca en el sistema operativo **UNIX**

# Evolución de los sistemas operativos (cont.)



## Historia del **UNIX**

- A partir del CTSS se diseña el **MULTICS** (MIT, Bell Labs, General Electric)
  - pensado para cientos de usuarios
  - escrito en PL/I
  - resultó demasiado ambicioso

# Evolución de los sistemas operativos (cont.)



## Historia del **UNIX** (cont.)

- Ken Thompson, de Bell Labs, desarrolla una versión monousuario, en ensamblador: **UNICS**
  - se unen al proyecto Brian Kernighan y Denis Ritchie
  - se porta a un computador PDP-11, con hardware de protección de memoria
  - se reescribe en un lenguaje propio: B, que es una simplificación del BCPL
  - Ritchie rediseña el lenguaje y lo llama C
  - se reescribe Unix en C: se consigue la portabilidad
  - se distribuye gratuitamente a las universidades
  - se estandariza con el nombre POSIX

# Evolución de los sistemas operativos (cont.)



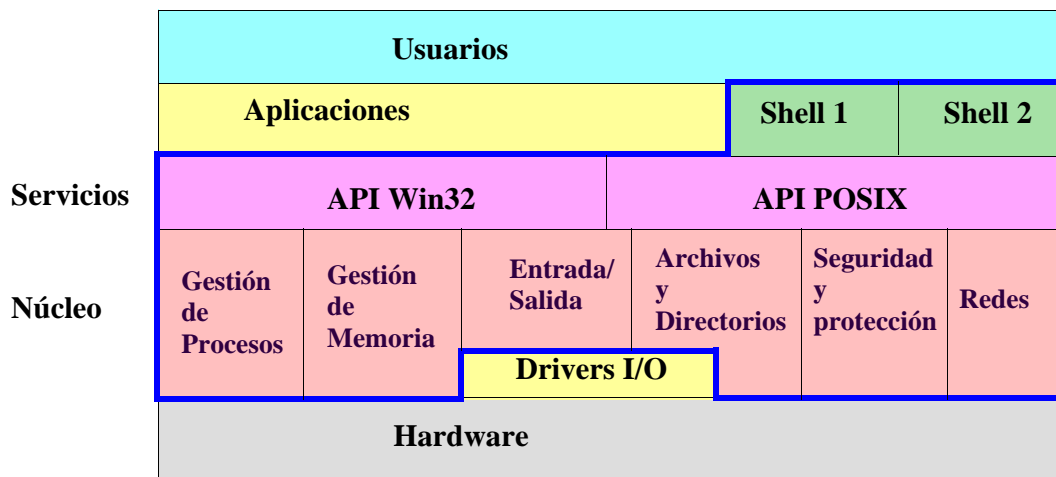
## **Computadores personales** (1980-1990)

- dos tipos:
  - estaciones de trabajo: UNIX y redes
  - PCs para oficinas y hogares: MS-DOS, luego Windows
- software fácil de usar: entornos gráficos de ventanas

## La **Internet** (1991-hoy)

- los sistemas operativos generales proporcionan conexión a la red
- se populariza el software libre (Gnu)
- dos grandes familias: Windows y Unix/Linux
- surgen otros niveles diferentes al SO: Java, .net

# 3. Estructura de los sistemas operativos



# Tipos de sistemas operativos por su estructura

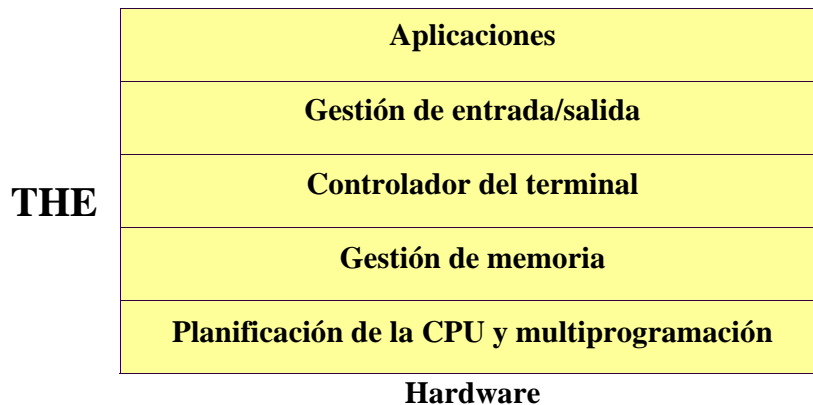
## a) *Monolíticos*

- Todos los componentes en un solo programa
- Un solo espacio de direcciones
- Un solo flujo de control (no hay concurrencia interna)
- El sistema operativo se ejecuta en modo privilegiado; las aplicaciones en modo usuario (restringido)
- Ejemplos: MS-DOS, UNIX
- Son complicados de modificar

# Tipos de sistemas operativos por su estructura (cont.)

## b) *A capas*

- Jerarquía de capas, cada una con una interfaz clara y que sólo usa los servicios de la capa inferior
- Ejemplos THE (Dijkstra 1968), OS/2 (Deitel, 1994)



# Tipos de sistemas operativos por su estructura (cont.)

## c) *Máquinas virtuales*

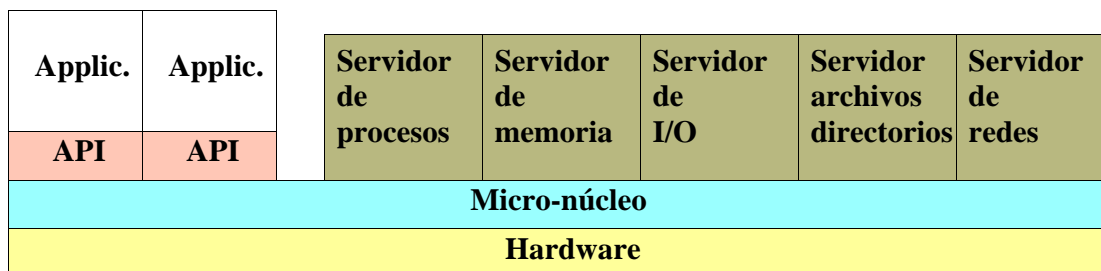
- Se proporciona un modelo de una máquina sobre la que se pueden ejecutar programas
- esta máquina se puede replicar por tiempo compartido
- se pueden ejecutar sistemas operativos completos sobre la máquina virtual
- Ejemplos:
  - JVM (Java virtual machine)
  - VM-Ware (emula un PC sobre Windows o Linux)
  - RT-Linux



# Tipos de sistemas operativos por su estructura (cont.)

## d) *Modelo cliente-servidor*

- Se mueven parte de los servicios del sistema operativo a una capa superior: servidores
- Los programas de aplicación son clientes
- Se comunican a través de un *micro-núcleo* (privilegiado)



# Tipos de sistemas operativos por su estructura (cont.)

## Los sistemas *cliente-servidor* (cont.)

- Son más flexibles
- Pero menos eficientes
- Ejemplo: Minix (Tanenbaum 1998), Mach (Accetta, 1986)
- Windows NT/2000/XP es mezcla de un sistema monolítico (por eficiencia) y cliente-servidor (por flexibilidad)

# Tipos de sistema operativo por sus servicios

Por el **número de usuarios**

- sistemas **empotrados**:
  - sin usuarios directos
  - forman parte de un sistema mayor
  - ejemplo: controlador de un televisor
- sistemas **monousuario**:
  - un solo usuario
  - ejemplo: Windows 98
- sistemas **multiusuario**
  - muchos usuarios
  - presenta protección entre usuarios
  - ejemplos: Unix, Windows XP profesional

# Tipos de sistema operativo por sus servicios (cont.)

Por el nivel de **protección** (se protegen datos de una aplicación y del sistema operativo frente a otras aplicaciones)

- **protegidos** (Unix, Windows XP profesional)
- **semi-protegidos** (Windows 98)
- **no protegidos** (Windows 3.1)

Por la capacidad de **conurrencia**

- **monoproceso y monotarea**: un solo programa con un solo flujo de control (MS-DOS)
- **multitarea**: capaz de ejecutar un solo programa, pero varias de sus partes a la vez (MaRTE OS)
- **multiproceso**: capaz de ejecutar varios programas a la vez

## 4. Concepto de programa y proceso

### Programa:

- es un fichero ejecutable, que reside en el disco
- generalmente creado mediante un compilador

### Proceso:

- es una instancia de un programa que está siendo ejecutada
- un mismo programa se puede ejecutar varias veces a la vez, dando lugar a varios procesos
- es la unidad de concurrencia básica
- tiene un espacio de direcciones propio
  - esto proporciona protección

## Servicios de procesos

### Crear un proceso

- a partir de un proceso padre, como una copia de él (UNIX)
- a partir de un fichero ejecutable en el disco (Win32)

### Ejecutar un proceso

- arrancado por el sistema: proceso por lotes o *batch*
- arrancado por el usuario: proceso interactivo

### Terminar un proceso

- terminación normal
- terminación por un error
- otro proceso o el usuario lo terminan

## Servicios de procesos (cont.)

**Cambiar** el programa de un proceso

- el proceso deja de ejecutar el programa actual
- y pasa a ejecutar otro que está en un fichero en disco

**Esperar** a que un proceso termine

- normalmente es un hijo
- se puede esperar a uno o varios

## 5. Concurrency: gestión de procesos y threads

**Thread:**

- un flujo de control simple perteneciente a un proceso
- todos los threads de un proceso comparten un único espacio de direccionamiento
- tienen un estado menos voluminoso que los procesos
- son más eficientes
- no tienen jerarquía padre-hijo

**Proceso** en una implementación multi-thread:

- un espacio de direccionamiento con uno o varios threads
- inicialmente contiene un solo thread: el thread principal
- luego se pueden crear más

# Servicios asociados a los threads

---

## **Crear** threads

- se especifica una función a ejecutar

## **Terminar** un thread

## **Esperar** a la terminación de un thread

- sólo a uno concreto

# Comunicación y sincronización entre procesos

---

## Comunicaciones locales

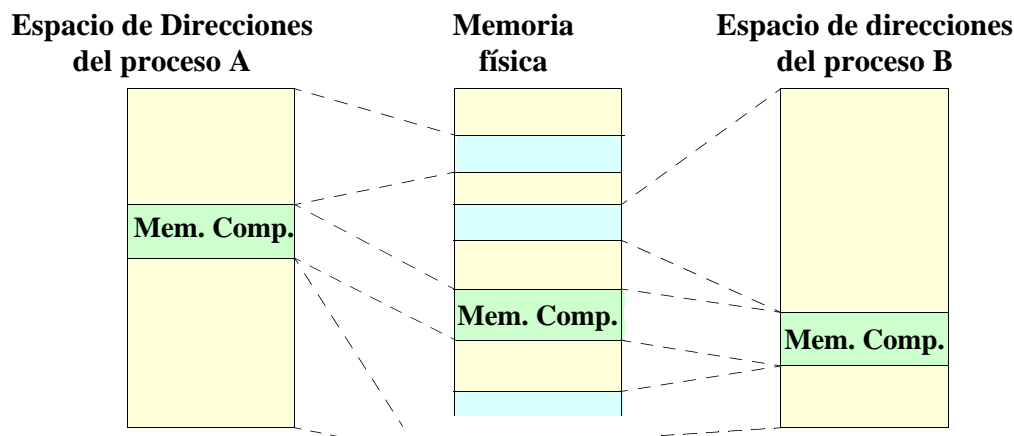
- eventos y variables condicionales: sincronización de espera
- señales: notificación de sucesos
- mensajes: envío de datos y posible sincronización de espera
- datos compartidos: sincronización de acceso mutuamente exclusivo

## Comunicaciones remotas

- mediante redes de comunicación y
- protocolos de comunicación

## 6. Gestión de memoria

### Gestión de memoria virtual



### Incluye los mecanismos de memoria compartida

## Gestión de espacios de direcciones virtuales

### Los procesos requieren estos servicios:

- asignarles memoria
- liberar memoria que ya no se necesita
- protección de accesos incorrectos
- compartir memoria
- desconectar la memoria virtual para procesos de tiempo real

## 7. Ficheros y el sistema de ficheros

---

Los ficheros son objetos abstractos que:

- se identifican mediante un nombre
- se pueden leer y escribir (excepto los directorios)
- representan:
  - ficheros normales: residen en memoria secundaria
  - directorios: contienen otros ficheros o directorios
  - dispositivos orientados al carácter
  - dispositivos orientados a bloque
  - tuberías (pipes) o ficheros especiales FIFO

## El sistema de ficheros

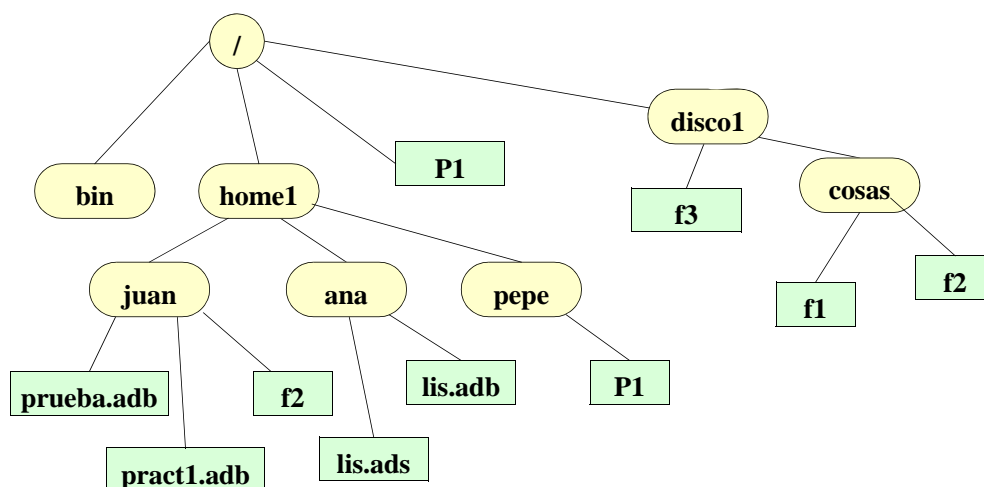
---

El sistema de ficheros es una colección de ficheros junto a los atributos que los caracterizan

- nombre
- tipo
- tamaño
- propietario
- fecha y hora de creación, modificación y acceso
- permisos de acceso (lectura, escritura y ejecución)
- enumeración de los bloques del disco donde se almacena el fichero (si es de tipo normal)

Tiene una estructura jerárquica de directorios

# El sistema de ficheros (cont.)



## Servicios relativos a los ficheros

### Crear:

- permite crear en el disco un fichero vacío en un directorio; generalmente también lo abre

### Abrir:

- permite crear una estructura de datos para manipular un fichero ya existente

### Cerrar:

- borra la estructura de datos creada al abrir el fichero

**Obtener los atributos** de un fichero: tamaño, tipo, permisos, ...

### Leer y escribir



# Servicios relativos a los ficheros (cont.)



## Uso de un fichero:

**Se abre o crea el fichero**

**Se usa para leer o escribir**

**Se cierra**

**Si es preciso, se borra**

Muchos de estos servicios están accesibles desde el lenguaje de programación

- pero otros no

# Servicios relativos a los directorios



**Crear** un directorio

**Borrar** un directorio

**Abrir** un directorio

- para operar con él

**Leer** un directorio

- recorriendo la lista de sus archivos

**Cerrar** un directorio

## 8. Entrada/salida

---

### Gestión de la entrada/salida:

- **facilita el uso de los dispositivos de entrada/salida**
  - optimizando el acceso según la velocidad del dispositivo
  - notificando a la aplicación sobre el estado del dispositivo
- **ofrece mecanismos de protección frente a los usuarios**
- **gestiona dispositivos de diversos tipos:**
  - de caracteres: por interrupciones o consulta
  - de bloques: por acceso directo a memoria

Los dispositivos de entrada/salida se asimilan a ficheros

## Servicios de entrada/salida

---

### **Leer** de un fichero o dispositivo

- depositando los datos en una variable en memoria

### **Escribir** en un fichero o dispositivo

- datos almacenados en una variable en memoria

### **Control** del funcionamiento del dispositivo

- depende por completo del dispositivo concreto
- por ejemplo:
  - cambiar la calidad de impresión (impresora)
  - cambiar la resolución (pantalla)
  - cambiar la velocidad de transmisión (línea de comunicación)

**Reposicionar** el puntero de lectura/escritura

**Sincronizar** la escritura de datos

- es decir, volcar físicamente los datos de las estructuras de datos intermedias al dispositivo final

## 9. Sistemas multiusuario y protección

La seguridad requiere dos tipos de servicios:

- **autenticación** de usuarios
  - por ejemplo, mediante nombre de usuario y contraseña
- **permisos**
  - asociados a los recursos: lista de control de accesos con sus modos de uso
- **privilegios**
  - asociados a los usuarios: lista de capacidades (cosas que un usuario puede hacer)
  - generalmente existen usuarios privilegiados: superusuario o administrador del sistema