



---

---

Sistemas Operativos I

## Tema 1

# Introducción a los sistemas operativos



Equipo de sistemas operativos DISCA / DSIC

# Objetivos

---



- 1.- Presentar el concepto de sistema operativo.
- 2.- Describir las funciones que debe llevar a cabo cualquier sistema operativo actual.
- 3.- Dar una visión general de la historia de los sistemas operativos que ayudará a entender cuándo y por qué se introdujeron estas funciones en el sistema.
- 4.- Presentar el concepto de llamada al sistema como mecanismo necesario para obtener los servicios de éste.
- 5.- Presentar el intérprete de órdenes como ejemplo más importante de programa del sistema.

# Introducción a los sistemas operativos



## Índice

### 1.- Concepto de sistema operativo

### 2.- Funciones del s.o.

- ✍ Gestión de procesos
- ✍ Gestión de la memoria
- ✍ Gestión de ficheros
- ✍ Gestión de la red
- ✍ Protección

### 3.- Evolución de los sistemas operativos

- ✍ Primeros sistemas
- ✍ Sistemas por lotes (batch)
- ✍ Sistemas multiprogramados
- ✍ Sistemas de tiempo compartido
- ✍ Sistemas Unix
- ✍ Sistemas para PC
- ✍ Tendencias actuales

### 4.- Llamadas al sistema

### 5.- Programas del sistema

## Bibliografía

✍ *W. Stallings.*

Operating Systems, 3<sup>a</sup> ed.

✍ Tema 2.

✍ *A. Silberschatz, P. B. Galvin.*

Sistemas Operativos. 5<sup>a</sup> ed.

✍ Temas 1 y 3



## 1.- Concepto de sistema operativo

## 2.- Funciones del s.o.

- ✍ Gestión de procesos
- ✍ Gestión de la memoria
- ✍ Gestión de ficheros
- ✍ Gestión de la red
- ✍ Protección

## 3.- Evolución de los sistemas operativos

- ✍ Primeros sistemas
- ✍ Sistemas por lotes (batch)
- ✍ Sistemas multiprogramados
- ✍ Sistemas de tiempo compartido
- ✍ Sistemas Unix
- ✍ Sistemas para PC
- ✍ Tendencias actuales

## 4.- Llamadas al sistema

## 5.- Programas del sistema

# 1.- Concepto de sistema operativo



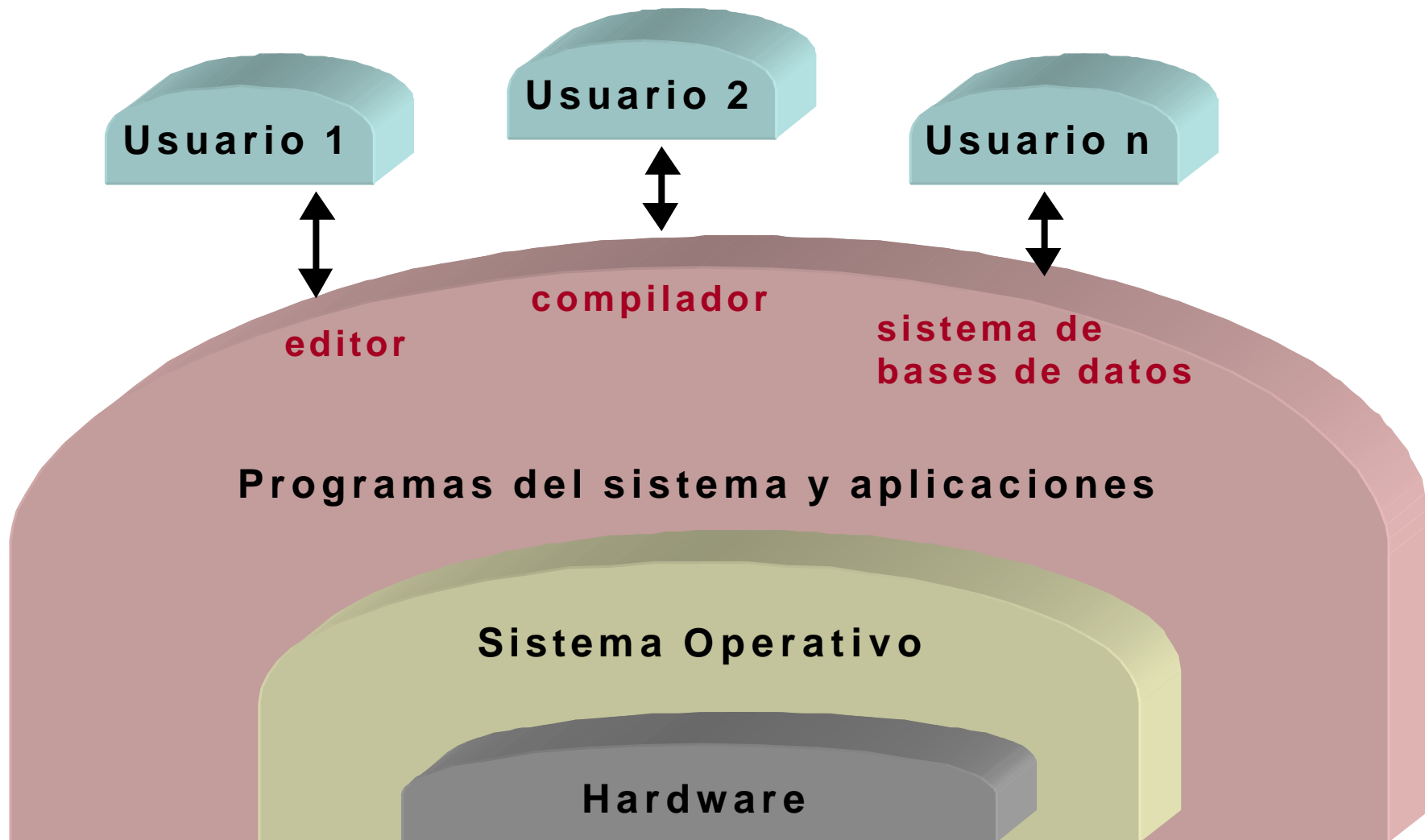
## Sistema operativo

Programa que actúa como interfaz entre los usuarios y el hw del computador.

### Objetivos:

- ✎ *Comodidad:* Facilitar el uso de la máquina a los usuarios, proporcionando un entorno cómodo para el desarrollo y ejecución de programas y para el acceso a los recursos de la máquina.
- ✎ *Eficiencia:* El s.o. debe administrar los recursos (hw y sw) de la máquina de forma que se aprovechen de la manera más conveniente.

# 1.- Concepto de sistema operativo



# 1.- Concepto de sistema operativo



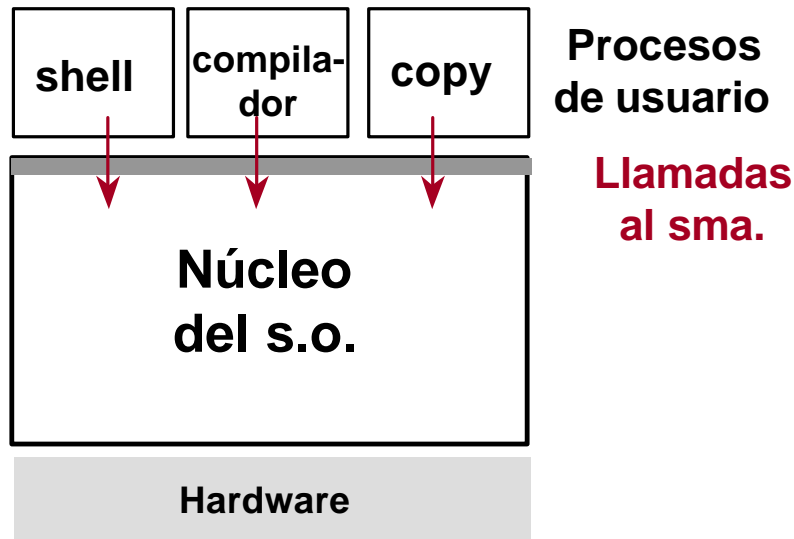
## Conceptos de sistema operativo

- 1.- **Máquina extendida:** Los recursos hw son difíciles de manejar y programar (especialmente la E/S). El s.o. debe *ocultar la complejidad* del hw y proporcionar una *interfaz sencilla* de utilizar.
- 2.- **Gestor de recursos:** En un computador los programas en ejecución deben *compartir los recursos* disponibles que son limitados (CPU, memoria, dispositivos de E/S, etc.). El s.o. establece la *política* que determina a quién, cuándo, cuánto tiempo y la cantidad de recurso que asigna.

# 1.- Concepto de sistema operativo



## Conceptos básicos



- ✎ **Núcleo del s.o.:** Parte residente del s.o.: se carga al arrancar la máquina y se mantiene siempre en memoria.
- ✎ **Procesos de usuario:** Aplicaciones hechas por programadores que se ejecutan sobre el s.o. Piden los servicios del s.o. invocando “llamadas al sistema”.
- ✎ **Llamadas al sistema:** Interfaz formada por un conjunto de servicios que el s.o. ofrece a los procesos de usuario.
- ✎ **Programas del sistema:** Utilidades del s.o. que se ejecutan como procesos de usuario (fuera del núcleo). *Ejemplos:*
  - ✎ Editores, compiladores, montadores, ...
  - ✎ Programas o ventanas para manipulación de ficheros y directorios (copy, mkdir, ...)
  - ✎ Programas de acceso a la red



# 1.- Concepto de sistema operativo

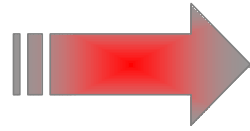


## ✎ Modos de ejecución

- ✎ La mayor parte de los procesadores actuales tienen como mínimo dos modos de funcionamiento o modos de ejecución:
  - ✎ **Supervisor o privilegiado:** En este modo es posible acceder a todo el juego de instrucciones del procesador. Con ello se pueden utilizar todos los recursos, y realizar ciertas acciones tales como programar los controladores de dispositivo, atender interrupciones, programar la unidad de gestión de memoria (MMU), etc.
  - ✎ **Usuario:** En este modo no se pueden utilizar las instrucciones del procesador relacionadas con la E/S, la gestión del modo de ejecución, la gestión de la memoria y otras. Estas instrucciones se catalogan como *privilegiadas*.
- ✎ En el modo supervisor únicamente se ejecuta el sistema operativo. Por tanto es el único programa que tiene control y acceso a todos los recursos de un equipo.
- ✎ En el modo usuario se ejecutarán todos los demás programas. Para utilizar los recursos no accesibles en modo usuario (aquellos que requieran instrucciones privilegiadas), se deberán realizar llamadas al sistema.



## 1.- Concepto de sistema operativo



## 2.- Funciones del s.o.

- ✍ Gestión de procesos
- ✍ Gestión de la memoria
- ✍ Gestión de ficheros
- ✍ Gestión de la red
- ✍ Protección

## 3.- Evolución de los sistemas operativos

- ✍ Primeros sistemas
- ✍ Sistemas por lotes (batch)
- ✍ Sistemas multiprogramados
- ✍ Sistemas de tiempo compartido
- ✍ Sistemas Unix
- ✍ Sistemas para PC
- ✍ Tendencias actuales

## 4.- Llamadas al sistema

## 5.- Programas del sistema

## 2.- Funciones del sistema operativo

---



### ✎ Funciones del s.o.

Las funciones más importantes que lleva a cabo un s.o. se pueden clasificar en:

- ✎ Gestión de procesos.
- ✎ Gestión de memoria.
- ✎ Gestión del sistema de ficheros.
- ✎ Gestión de dispositivos de E/S.
- ✎ Gestión de la red.
- ✎ Protección.

— cada una de ellas viene caracterizada por una serie de abstracciones que se han desarrollado para resolver las dificultades de los problemas prácticos.

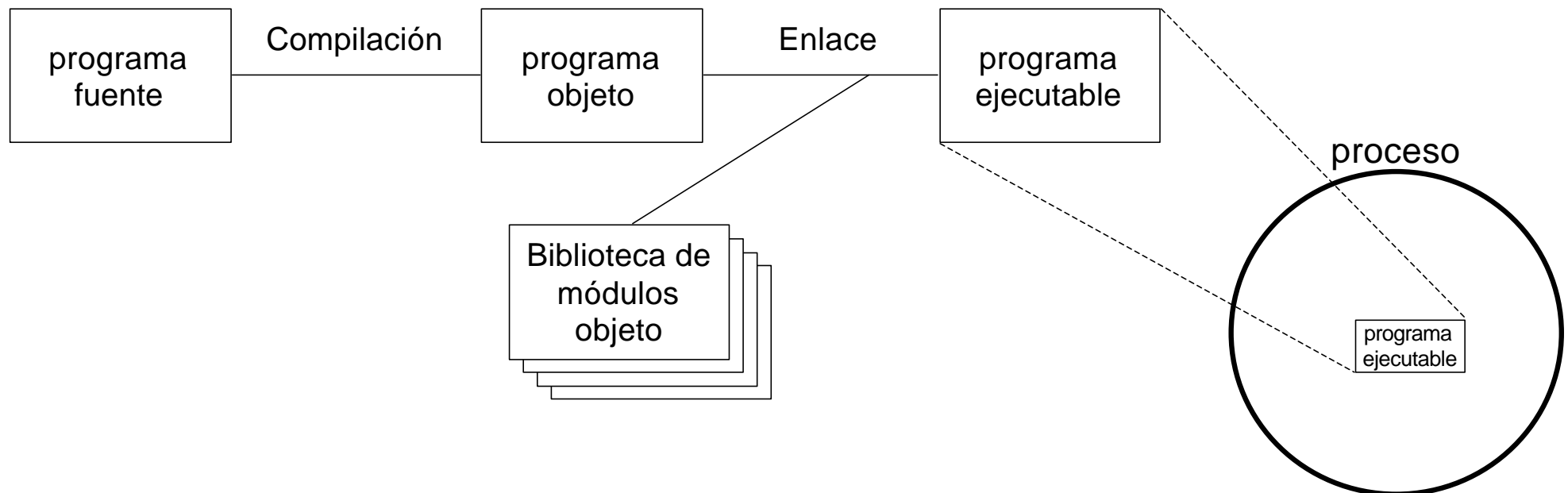
## 2.- Funciones del sistema operativo



### Gestión de procesos

Es importante diferenciar entre proceso y programa:

- Programa: Lista de instrucciones. Ente pasivo. Se almacena en ficheros.
- Proceso: Ejecución de un programa. Ente activo. La ejecución de un programa requiere que éste se cargue, previamente, en memoria.



## 2.- Funciones del sistema operativo



### Gestión de procesos (ii)

#### Problemática de implementación de procesos

Tema  
3

- ✍ **Ejecución concurrente:** Simular la ejecución paralela de varios programas sobre un único procesador (repartiendo para ello el tiempo del procesador entre los procesos).
- ✍ **Planificación de procesos:** Determinar a qué proceso se le asigna el procesador en cada instante.

#### Funciones para la gestión de procesos:

- ✍ Creación y eliminación de procesos.
- ✍ Comunicación y sincronización de procesos.

Tema  
4

## 2.- Funciones del sistema operativo



### Gestión de la memoria

S02

La gestión de memoria se encuentra normalmente asociada a la gestión de procesos: para ejecutar un proceso es necesario asignarle memoria y cargarlo en ella; cuando finaliza su ejecución, la memoria debe ser liberada.

#### ✎ **Problemática para la gestión de memoria**

✎ **Aislamiento del proceso:** Cada proceso debe tener una zona de memoria independiente que no puede ser interferida por otros procesos.

✎ **Asignación de la memoria:** Llevar el control de las zonas de memoria libres y ocupadas y la zona de memoria asignada a cada proceso.

✎ **Memoria virtual:** Técnica que permite direccionar la memoria desde un punto de vista lógico, sin depender del tamaño o la ubicación de la zona de memoria que finalmente se asigne al proceso.

✎ **Funciones de gestión de memoria:** La gestión de memoria es transparente al programador: la asignación y la liberación se hacen automáticamente con la creación y eliminación de procesos.

## 2.- Funciones del sistema operativo



### Gestión de ficheros

S02

✎ **Fichero:** Abstracción para definir una colección de información no volátil. Su objetivo es proporcionar un modelo de trabajo sencillo con los discos.

### ✎ **Problemática de la implementación de ficheros**

✎ Organización del sistema de ficheros: directorios

✎ Asignación de espacio en disco a la información (de manera no necesariamente contigua).

✎ Gestión del espacio libre y ocupado en disco.

✎ Protección.

### ✎ **Funciones para la gestión de ficheros:**

✎ Creación y eliminación de ficheros y directorios.

✎ Manipulación de ficheros y directorios (lectura, escritura, copia, ...).

## 2.- Funciones del sistema operativo



### Gestión de dispositivos de E/S

S02

La gestión de la E/S tiene como objetivo proporcionar una interfaz de alto nivel de los dispositivos de E/S sencilla de utilizar. En algunos sistemas esta interfaz es semejante a la de los ficheros.

### Problemática de la gestión de dispositivos de E/S

**Manejadores de dispositivos (*drivers*):** Componentes del s.o. que ocultan la complejidad y las peculiaridades del hw de E/S y ofrecen una interfaz independiente del dispositivo. El manejador (sw) se encarga de realizar la programación de los *controladores de dispositivos* (hw).

**Gestionar el uso de los dispositivos:** Proporcionar *acceso exclusivo, spooling, buffering, caching, ...*



## 2.- Funciones del sistema operativo



### Gestión de la red TIN, RAL, RC1, RC2, SRD

Comprende varios niveles:

- ✎ **Manejadores (drivers) de tarjetas de red.**
- ✎ **Protocolos de comunicación** (TCP/IP, Netbios, ...): Resuelven el acceso a la red y proporcionan una API (*Application Programming Interface*) para comunicación entre procesos remotos.
- ✎ **Aplicaciones para uso de la red** (WWW, FTP, ...). Son programas de aplicación contruidos sobre la API de comunicación que facilitan el acceso a recursos remotos (ficheros, información,...)

## 2.- Funciones del sistema operativo



### ☞ Protección y seguridad — SO2, CSO, AUW, SSO

Mecanismo para permitir o denegar el acceso de un *proceso* de un usuario determinado a un *recurso* (ficheros, dispositivos de E/S).

### ☞ **Problemática de protección**

- ☞ Diseño de un modelo y una *política* de protección para definir qué accesos son legales y cuáles son ilegales.
- ☞ Implementación de un mecanismo ejecutivo que vigile el cumplimiento de las reglas de protección definidas.
- ☞ *Seguridad*: Mecanismo para garantizar la privacidad de la información frente a ataques intencionados por parte de intrusos. Muchos de ellos se basan en técnicas para autenticar usuarios y codificación criptográfica.

### ☞ **Funciones de protección:**

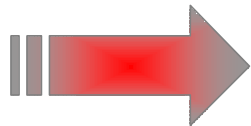
- ☞ Funciones para definir “propietarios” de recursos o niveles de protección.



## 1.- Concepto de sistema operativo

## 2.- Funciones del s.o.

- ✍ Gestión de procesos
- ✍ Gestión de la memoria
- ✍ Gestión de ficheros
- ✍ Gestión de la red
- ✍ Protección



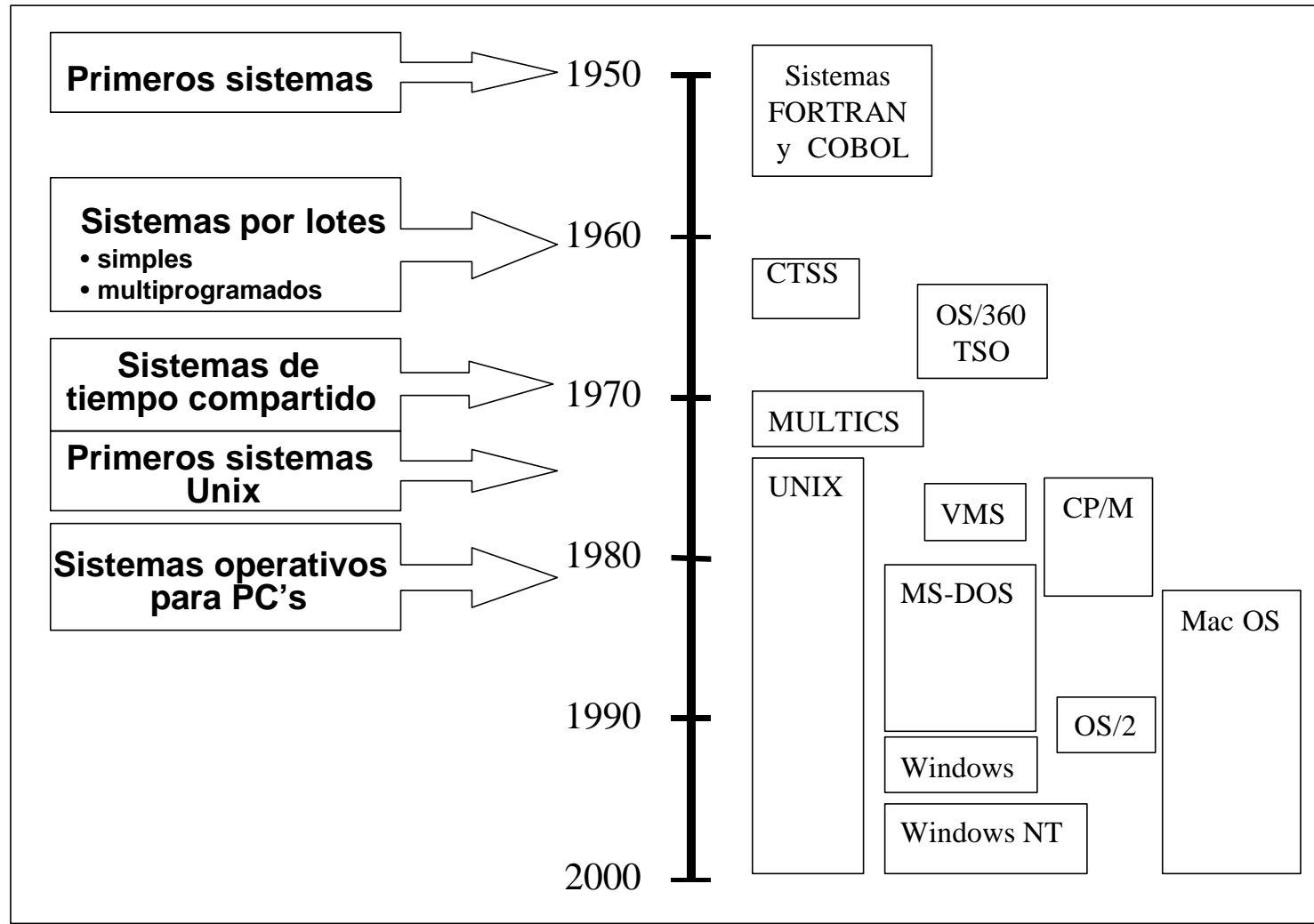
## 3.- Evolución de los sistemas operativos

- ✍ Primeros sistemas
- ✍ Sistemas por lotes (batch)
- ✍ Sistemas multiprogramados
- ✍ Sistemas de tiempo compartido
- ✍ Sistemas Unix
- ✍ Sistemas para PC
- ✍ Tendencias actuales

## 4.- Llamadas al sistema

## 5.- Programas del sistema

# 3.- Evolución de los sistemas operativos



# 3.- Evolución de los sistemas operativos

---



## 3.1.- Primeros sistemas

### Tipos de sistema

- ✍ Máquinas voluminosas operadas desde una consola
- ✍ Teletipos, tarjetas y cintas perforadas
- ✍ Sin sistema operativo

### Modo de operación

- ✍ “*Hands on*”: El usuario hace las veces de operador delante de la consola y tiene el control total sobre la máquina.

# 3.- Evolución de los sistemas operativos

---



## 3.1.- Primeros sistemas (ii)

### Software del sistema

No existe un s.o. como tal. El sw del cual se dispone es:

✍ Cargadores

✍ Compiladores / ensambladores

✍ Bibliotecas con “manejadores de dispositivos”

### Problemática

✍ Mala utilización de recursos caros (CPU, E/S) debido al elevado tiempo de desarrollo y preparación delante de la consola.

✍ Interfaz muy rudimentaria con el usuario.

# 3.- Evolución de los sistemas operativos



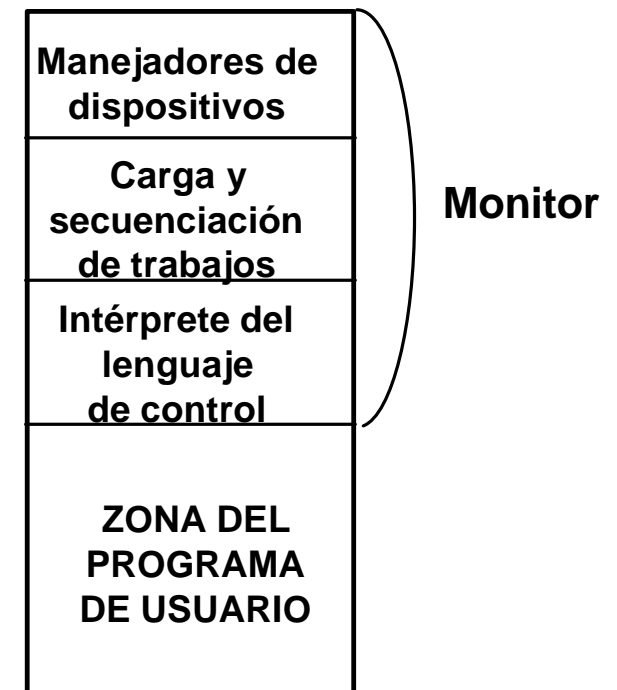
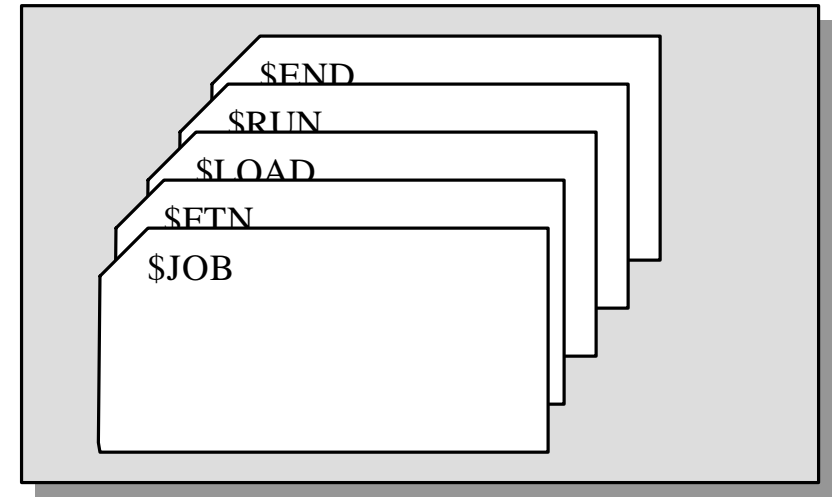
## 3.2.- Sistemas por lotes (batch) sencillos

### ✎ Modo de operación:

- ✎ Contratación de un *operador* profesional.
- ✎ Ausencia de interacción directa usuario - máquina.
- ✎ Agrupación *por lotes* de trabajos similares.
- ✎ Secuenciación automática de trabajos con *tarjetas de control*.

### ✎ Funciones del s.o.

Aparece el primer esbozo de s.o.: el *monitor residente*.



# 3.- Evolución de los sistemas operativos



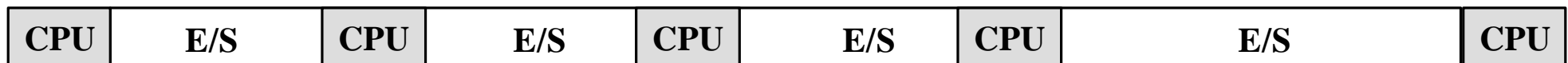
## 3.2.- Sistemas por lotes (batch) sencillos (ii)

### Problemática

Baja utilización de la CPU debido a los elevados tiempos de E/S.

✍ **Tipo de carga:** *trabajos* con secuencia de ráfagas de CPU y E/S. La diferencia de velocidad entre la CPU y la E/S es de tres órdenes de magnitud o más.

✍ **Utilización de la CPU** =  $\text{tiempo\_CPU\_ocupada} / \text{tiempo\_total}$





# 3.- Evolución de los sistemas operativos

---



## 3.2.- Sistemas por lotes sencillos (iii)

### Soluciones

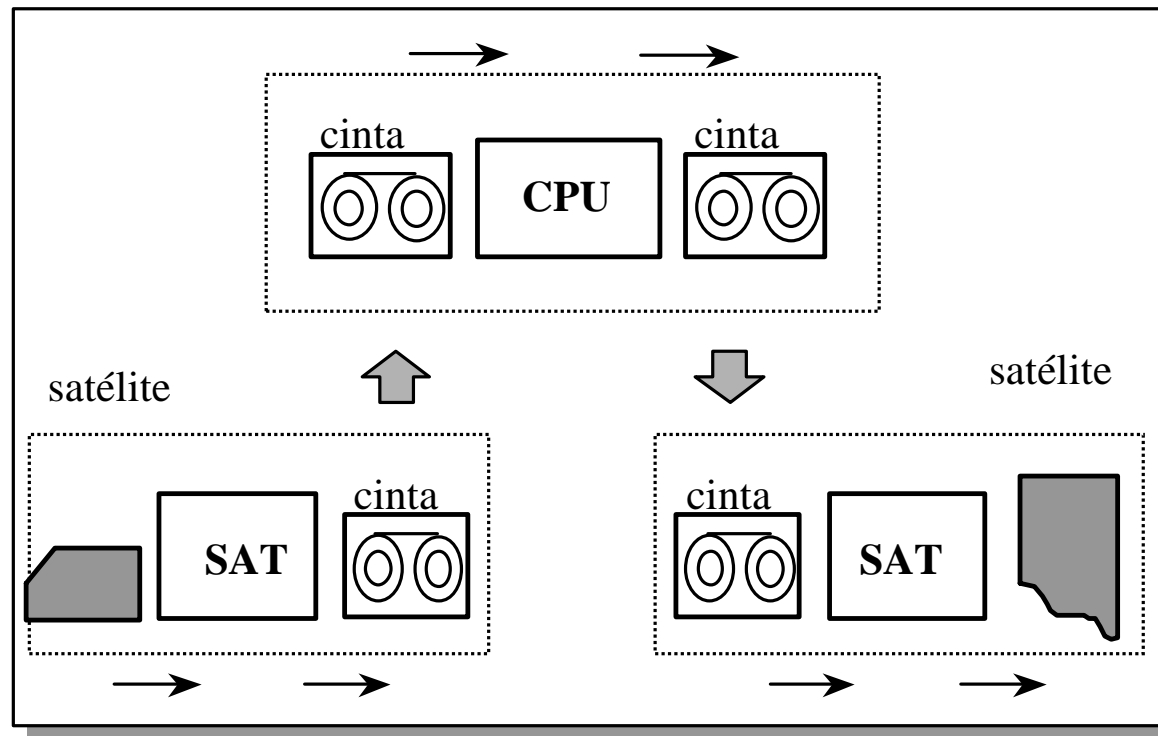
- ✍ **Operación fuera de línea:** reducir los tiempos de E/S utilizando computadores satélite para ejecutar las operaciones de E/S sobre dispositivos lentos.
- ✍ **Spooling:** reducir los tiempos de E/S utilizando el disco como un buffer para compensar la diferencia de velocidades entre la E/S y los periféricos lentos.
- ✍ **Multiprogramación:** Aprovechar las ráfagas de espera (de la E/S) de un trabajo para ejecutar las ráfagas de CPU de otros trabajos.

# 3.- Evolución de los sistemas operativos



## Operación fuera de línea

- La CPU trabaja con periféricos de E/S “rápidos”: cintas
- Sistemas satélites se encargan de efectuar la E/S sobre periféricos lentos (tarjetas, impresoras) y de transvasar los datos entre periféricos lentos y rápidos.

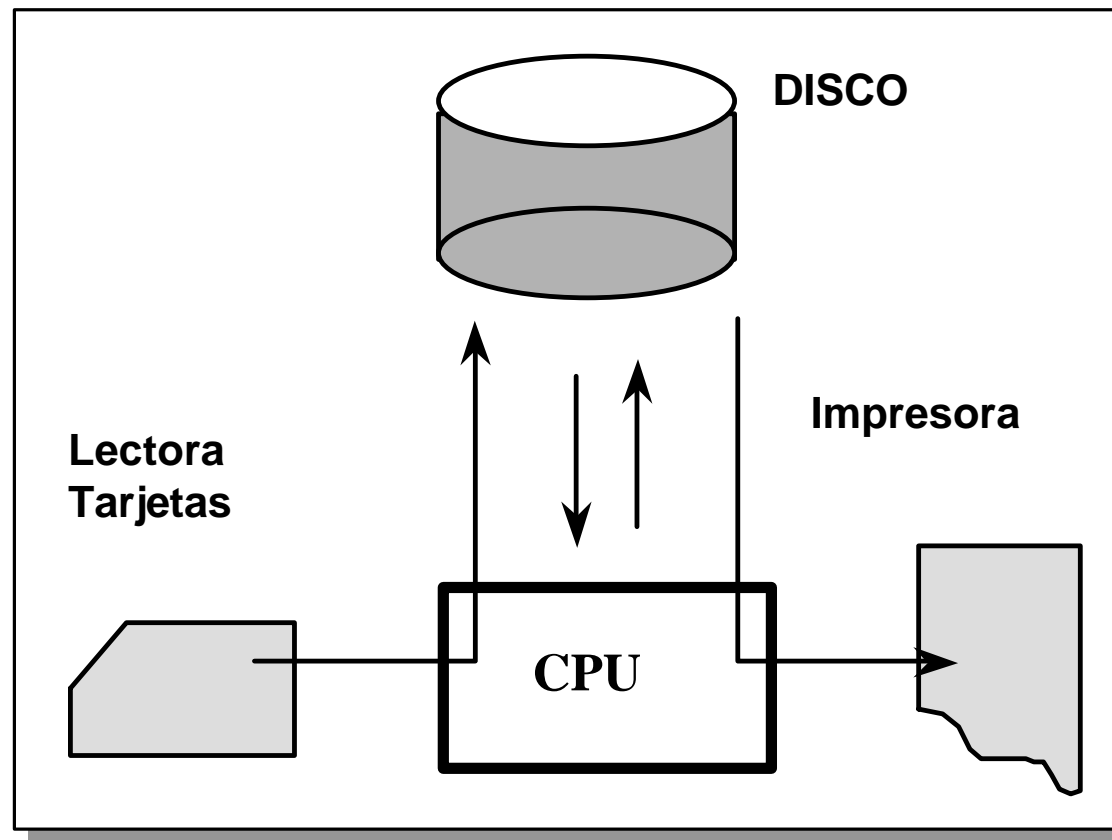


# 3.- Evolución de los sistemas operativos



## Spooling

- ✗ El disco actúa como un buffer de E/S.
- ✗ Mientras el periférico está ocupado, la CPU vuelve a disco.
- ✗ Cuando el periférico está listo, se realiza la transferencia disco - periférico.



## 3.- Evolución de los sistemas operativos



### 3.3.- Sistemas por lotes multiprogramados

Son sistemas por lotes en los que se introduce la técnica de multiprogramación.

En los sistemas por lotes *sin spooling*, no podía haber multiprogramación puesto que se dependía del lector de tarjetas.

No se podía intercalar la ejecución de dos trabajos diferentes, puesto que sólo las tarjetas de uno de ellos podían estar en el lector.

#### Nuevas funciones del s.o.

\_El s.o. crece considerablemente en funciones y tamaño:

✍ **Gestión de procesos:** planificación de procesos y cambios de contexto, ...

✍ **Gestión de memoria:** reparto y protección de la memoria, ...

#### Problemática

✍ La ausencia de interacción usuario - máquina hace difícil la elaboración y depuración de programas

# 3.- Evolución de los sistemas operativos



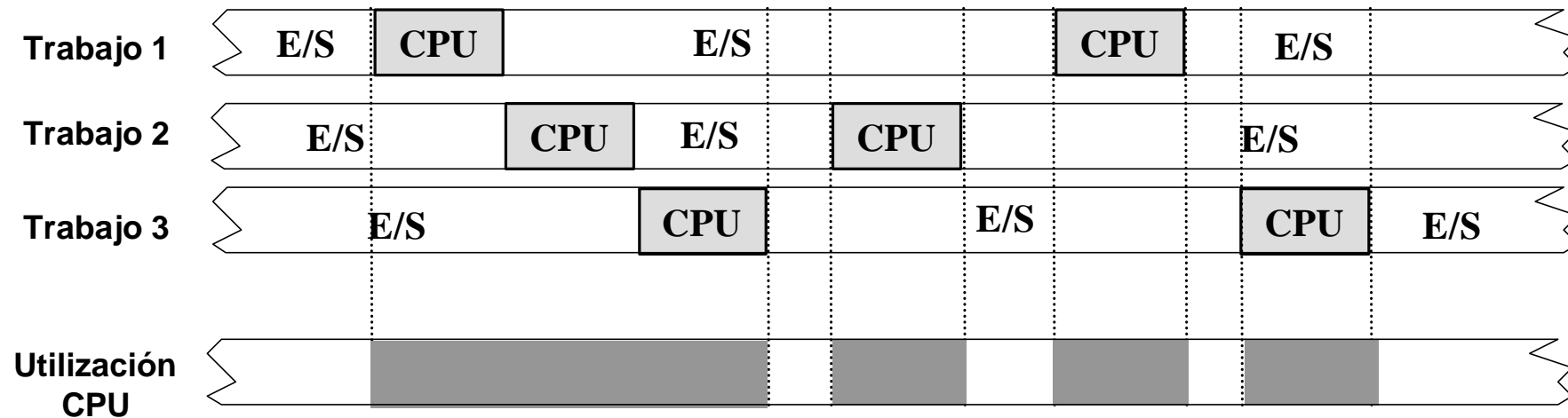
## 3.3.- Sistemas por lotes multiprogramados (ii)

### ↳ Multiprogramación

↳ **Ejecución concurrente:** Ejecución paralela de varios trabajos.

- Con un solo procesador se lleva a cabo alternando la CPU entre procesos: las ráfagas de espera (de E/S) de un trabajo se aprovechan para ejecutar las ráfagas de CPU de otros trabajos
- Implica mantener varios trabajos en memoria simultáneamente.

↳ **Objetivo:** Aumentar la utilización de la CPU



# 3.- Evolución de los sistemas operativos



## 3.4.- Sistemas de tiempo compartido

### Tipos de sistema

- ✍ *Mainframes* con muchas terminales que distribuyen potencia de cálculo como una especie de “central eléctrica”.
- ✍ Sistemas multiusuario que combinan la multiprogramación y la interacción usuario - máquina a través de terminales.

### Objetivos

- ✍ Tiempos de respuesta aceptables
- ✍ **Mejora de la interfaz usuario - máquina**
- ✍ *Multitarea*: un usuario puede ejecutar varios *procesos* concurrentemente.

## 3.- Evolución de los sistemas operativos



### 3.4.- Sistemas de tiempo compartido (ii)

#### Funciones del s.o:

✍ **Gestión de procesos:** Madura el concepto de *proceso* y la planificación.

✍ **Gestión de memoria:** Nacen y se desarrollan las técnicas de *memoria virtual*.

✍ **Gestión de ficheros:** Nace la necesidad de sistemas de ficheros “en línea” y comienzan a desarrollarse los precedentes de los sistemas de ficheros actuales.

✍ Necesidad de protección entre los diferentes usuarios y procesos existentes.

✍ **Crisis del sw:** La envergadura que alcanza el s.o. pone en tela de juicio las metodologías existentes para desarrollar sw.

## 3.- Evolución de los sistemas operativos



### 3.5.- Los primeros sistemas Unix

#### ✎ Tipo de sistema

✎ *Minicomputadores*: máquinas de arquitectura más sencilla que los *mainframes* pero con mayor potencia de cálculo y menos usuarios.

✎ *Ejemplos*: PDP-1 ... PDP-7 ... PDP-11.

#### ✎ Objetivos

✎ Adaptar los objetivos de los sistemas de tiempo compartido (MULTICS) a arquitecturas minicomputador.

✎ Utilización de lenguajes de alto nivel para el diseño del s.o.



## 3.- Evolución de los sistemas operativos



### 3.5.- Los primeros sistemas Unix (ii)

#### ✎ Aportaciones de Unix

- ✎ Diseño sencillo y potente. Dos únicas abstracciones: *proceso* y *fichero*.
- ✎ Introduce el modelo de *sistema de ficheros* actual.
- ✎ Introduce un potente *intérprete de órdenes*.
- ✎ *Entorno de programación* muy completo.
- ✎ Base para el desarrollo de la primera *red de computadores* y TCP/IP.
- ✎ Introduce el concepto de *sistema abierto*.

# 3.- Evolución de los sistemas operativos



## 3.6.- Sistemas operativos para PCs

### Tipo de sistema

- ✍ Máquinas basadas en microprocesadores. Inicialmente de arquitectura muy sencilla y limitada, pero con una continua evolución en potencia.
- ✍ Interfaces usuario-máquina muy sofisticadas: ratones, pantallas gráficas, etc.
- ✍ Pensadas para un solo usuario.

### Objetivos

- ✍ **Sencillez y comodidad del usuario:** contrasta con los primeros sistemas, donde los altos precios imponían la optimización en la utilización de recursos.

# 3.- Evolución de los sistemas operativos



## 3.7.- Sistemas operativos para PCs (ii)

### Características primeros sistemas (MS-DOS, CP/M, ...)

✍ Poco más que un sistema de ficheros y un cargador.

### Características sistemas actuales (Windows, Unix, MacOS, OS/2, ...)

✍ Interfaz basada en sistema de ventanas y ratón (introducida por Macintosh)

✍ Sistemas de ficheros tipo árbol (introducidos por Unix)

✍ Capacidad multitarea

✍ Acceso a la red

✍ Progresiva introducción de mecanismos de protección (al estilo UNIX)

## 3.- Evolución de los sistemas operativos



### 3.8.- Tendencias actuales

- ✍ **Sistemas operativos en red:** Permiten el *acceso a recursos remotos*. Proporcionan protocolos de comunicación y gran variedad de aplicaciones para acceso a la red.
- ✍ **Sistemas distribuidos:** Permiten el acceso a recursos remotos *de manera transparente* (obviando el hecho de la distribución). Proporcionan entornos de programación distribuida y acceso transparente a bases de datos remotas.
- ✍ **Sistemas multimedia:** Sistemas que incorporan interfaces audiovisuales.
  - ✍ Desarrollo de nuevas interfaces usuario-máquina.
  - ✍ Desarrollo de protocolos de transmisión en tiempo real.

# 3.- Evolución de los sistemas operativos

---



## ✎ Tendencias de diseño actuales

- ✎ Tecnología orientada a *objetos*
- ✎ Modelo *cliente-servidor* para aplicaciones distribuidas
- ✎ *Micronúcleos*: núcleos de s.o. con una funcionalidad mínima sobre los que desarrollar el s.o. como una aplicación de usuario.



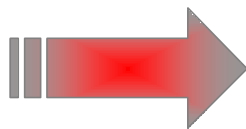
## 1.- Concepto de sistema operativo

## 2.- Funciones del s.o.

- ✍ Gestión de procesos
- ✍ Gestión de la memoria
- ✍ Gestión de ficheros
- ✍ Gestión de la red
- ✍ Protección

## 3.- Evolución de los sistemas operativos

- ✍ Primeros sistemas
- ✍ Sistemas por lotes (batch)
- ✍ Sistemas multiprogramados
- ✍ Sistemas de tiempo compartido
- ✍ Sistemas Unix
- ✍ Sistemas para PC
- ✍ Tendencias actuales



## 4.- Llamadas al sistema

## 5.- Programas del sistema

## 4.- Llamadas al sistema

---



### ✎ Llamadas al sistema

- ✎ Conjunto de servicios básicos que proporciona el sistema operativo
- ✎ Definen la interfaz entre un proceso y el sistema operativo.

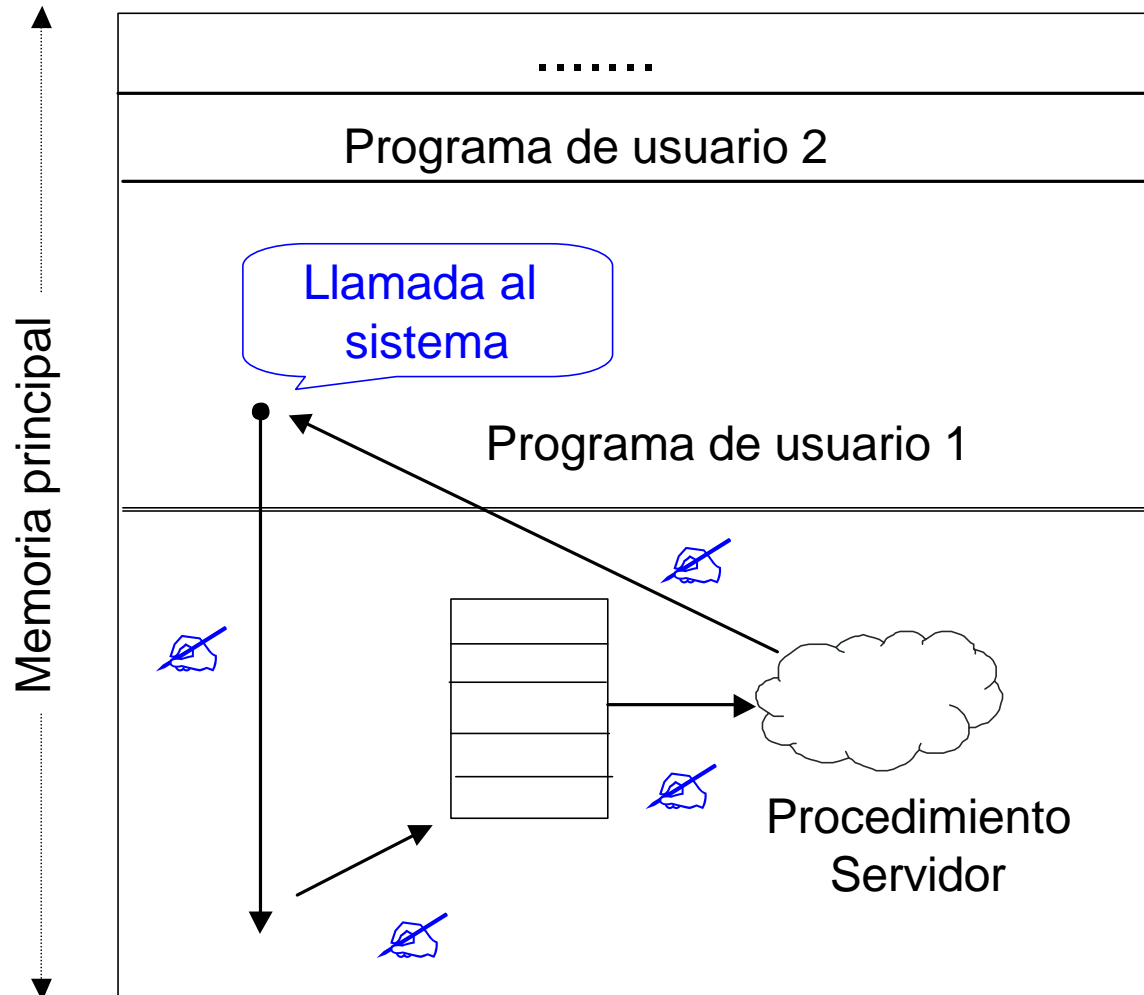
### ✎ Implementación de las llamadas al sistema

- ✎ Se implementan como interrupciones software o *traps*.
- ✎ Normalmente se proporciona una interfaz como funciones de biblioteca de lenguajes de programación que ocultan el *trap* y dan la apariencia de llamadas a procedimientos o funciones predefinidas.

# 4.- Llamadas al sistema



## Implementación de las llamadas al sistema



Programas de usuario  
**Modo Usuario**

- Trap o int. software
- Determinación del servicio
- Llamada al p.servidor
- Retorno del control

Sistema Operativo  
**Modo Supervisor**





## 1.- Concepto de sistema operativo

## 2.- Funciones del s.o.

- ✎ Gestión de procesos
- ✎ Gestión de la memoria
- ✎ Gestión de ficheros
- ✎ Gestión de la red
- ✎ Protección

## 3.- Evolución de los sistemas operativos

- ✎ Primeros sistemas
- ✎ Sistemas por lotes (batch)
- ✎ Sistemas multiprogramados
- ✎ Sistemas de tiempo compartido
- ✎ Sistemas Unix
- ✎ Sistemas para PC
- ✎ Tendencias actuales

## 4.- Llamadas al sistema



## 5.- Programas del sistema

# 5.- Programas del sistema

---



## Concepto

- ✍ Utilidades del sistema operativo que se ejecutan como procesos de usuario y proporcionan un entorno más cómodo.
- ✍ Son programas escritos en un lenguaje de programación (como C) que realizan llamadas al sistema.

## Clasificación

- ✍ Tratamiento de ficheros y directorios: mkdir, cp, mv, ls ...
- ✍ Filtros: grep, sort, head, tail ...
- ✍ Desarrollo de programas: Editores, compiladores, ensambladores, editores de enlace, depuradores ...
- ✍ Sistemas de ventanas: X11
- ✍ Comunicaciones: mail, ftp, rlogin ...
- ✍ Intérpretes de órdenes: sh, ksh, bash

# 5.- Programas del sistema

---



## Intérprete de órdenes

- ✍ Es la interfaz primaria entre el usuario y el sistema operativo.
- ✍ Es un programa que lee de su entrada estándar una orden introducida por un usuario, la analiza y la ejecuta.
- ✍ En la mayoría de sistemas el intérprete de órdenes es un programa que se ejecuta como un proceso de usuario.
- ✍ Ejemplos: El Shell de Unix, COMMAND.COM de MS-DOS
- ✍ Dos tipos de órdenes: externas e internas

# 5.- Programas del sistema

---



## Órdenes externas

- ✎ El intérprete de órdenes crea un proceso para ejecutar la orden. La orden es cualquier fichero ejecutable.
- ✎ Ejemplo: `cp f1 f2`

## Órdenes internas

- ✎ El intérprete de órdenes ejecuta él mismo la orden. Su código contempla las acciones necesarias para ejecutarla.
- ✎ Ejemplo: `cd`