



SISTEMAS OPERATIVOS:

Lección 1: INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS

Jesús Carretero Pérez
Alejandro Calderón Mateos
José Daniel García Sánchez
Francisco Javier García Blas
José Manuel Pérez Lobato
María Gregoria Casares Andrés



ADVERTENCIA

- Este material es un simple guión de la clase: no son los apuntes de la asignatura.
- El conocimiento exclusivo de este material no garantiza que el alumno pueda alcanzar los objetivos de la asignatura.
- Se recomienda que el alumno utilice los materiales complementarios propuestos.



- Comprender de forma global la estructura y funcionamiento del computador.
- Recordar los elementos del computador que tienen impacto en el sistema operativo.
- Comprender qué es un Sistema Operativo.
- Conocer los principales servicios y componentes del Sistema Operativo.
- Comprender el proceso de arranque del sistema operativo.



1. ¿Por qué estudiar SSOO?
2. Estructura y funcionamiento de un computador.
3. Concepto de sistema operativo. Componentes y estructura del sistema operativo.
4. Arranque y activación del Sistema Operativo.
5. Evolución de los sistemas operativos.



¿Por qué hay que saber de SS.OO.?

- a) El SO, sus peculiaridades internas, influye de mucho en el funcionamiento general, en la seguridad y/o rendimiento del computador
- b) La importancia de la elección de un determinado SO para una empresa es cada día mayor, casi estratégica
- c) Conocer el funcionamiento del SO es fundamental para desarrollar aplicaciones que obtengan buenas prestaciones y para comprender la causa de muchos problemas.



Para comprender el funcionamiento del sistema

- ¿Qué SO aprovecha mejor las capacidades de mi sistema?
- ¿El SO soporta todos los dispositivos que pretendo conectar al computador? Si no lo hace, ¿qué se puede hacer?
- ¿Es lo suficientemente seguro para el entorno en el que ha de integrarse?
- ¿Mi/s aplicación/es correrá/n “suavemente” sobre el SO elegido? ¿Cómo se adaptará a mi carga de trabajo concreta?

Para poder elegir adecuadamente

- ¿Es fácil encontrar administradores para este SO? ¿La administración es una tarea “oscura” y exclusiva de personal ultra-especializado?
- ¿Qué soporte tiene el SO? ¿Con qué frecuencia se publican parches y mejoras?
- Aparte del coste ¿Qué expectativas de futuro tiene?



**Hay que
proteger la
inversión**



Para desarrollar software con buenas prestaciones

- Cuando se desarrolla software se debe recurrir a los servicios del SO para realizar muchas tareas.
 - ¿Qué servicios ofrece mi SO y cómo puedo invocarlos?
- Para aprovechar las nuevas arquitecturas es esencial el desarrollo de aplicaciones multi-hilo.
 - ¿Cómo se desarrolla una aplicación multi-hilo para mi SO?



symbian
OS



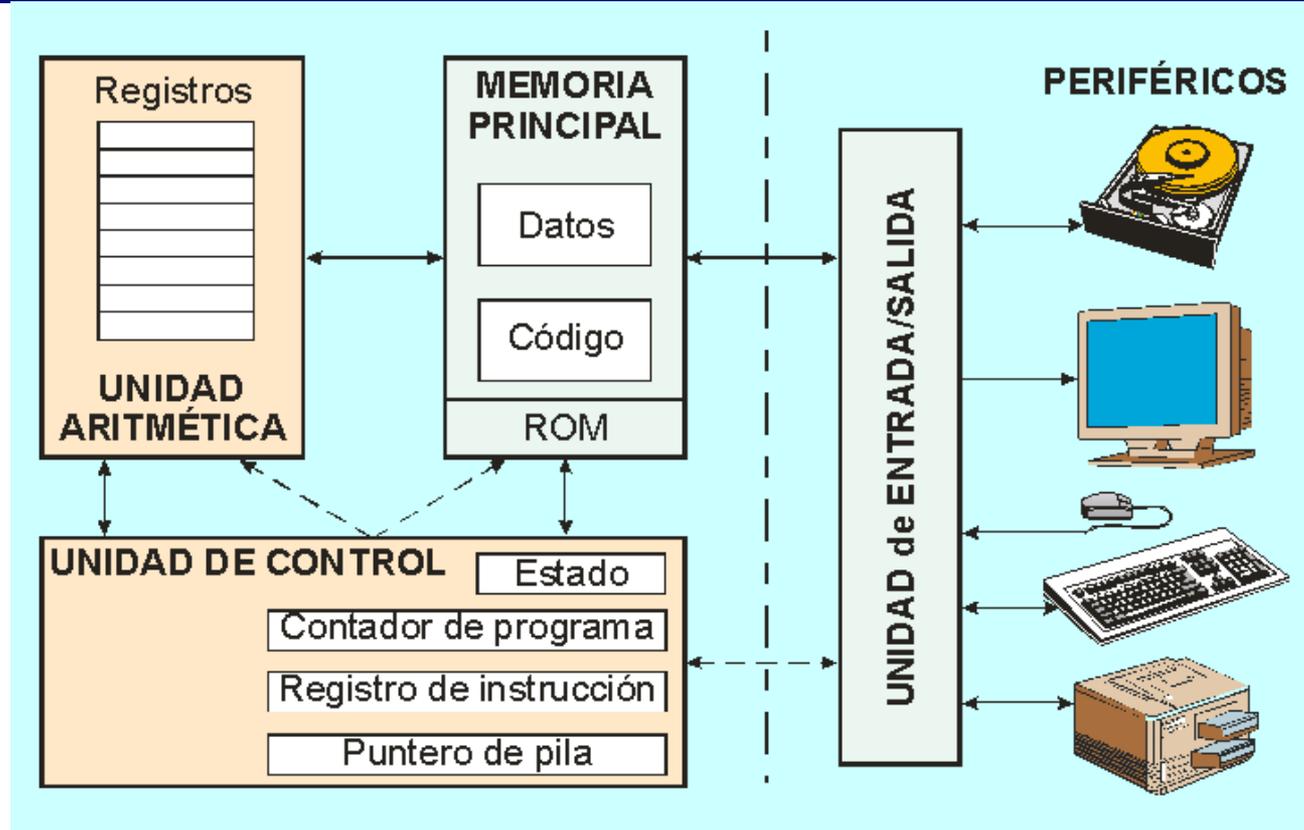
AGNÓSTICO EN SISTEMAS OPERATIVOS





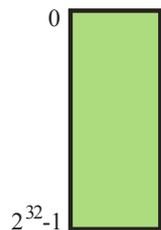
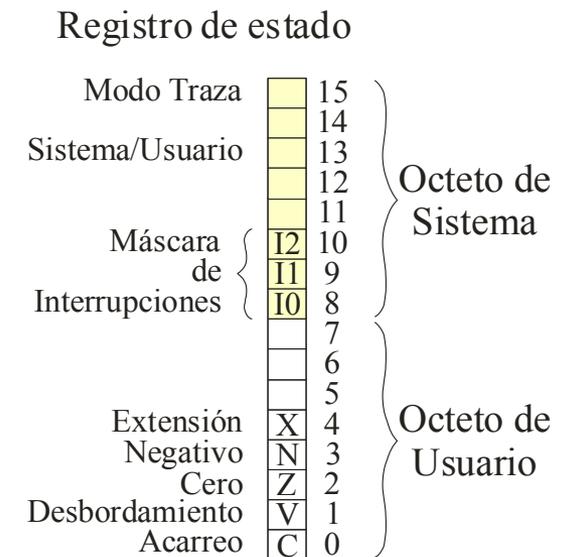
1. ¿Por qué estudiar SSOO?
- 2. Estructura y funcionamiento de un computador.**
3. Concepto de sistema operativo. Componentes y estructura del sistema operativo.
4. Arranque y activación del Sistema Operativo.
5. Evolución de los sistemas operativos.

Estructura del computador



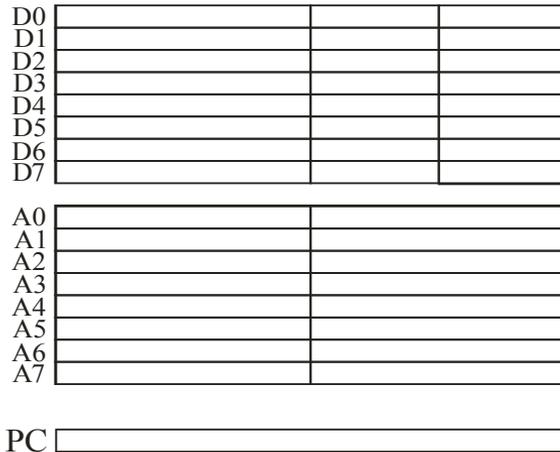
- **Monoprocesador**
- **Multiprocesador: MIMD: M. Distribuida vs. (UMA y NUMA)**

- Modelo de programación
- Juego de instrucciones (**U**usuario y **N**úcleo –ó **K** ó **S**--)

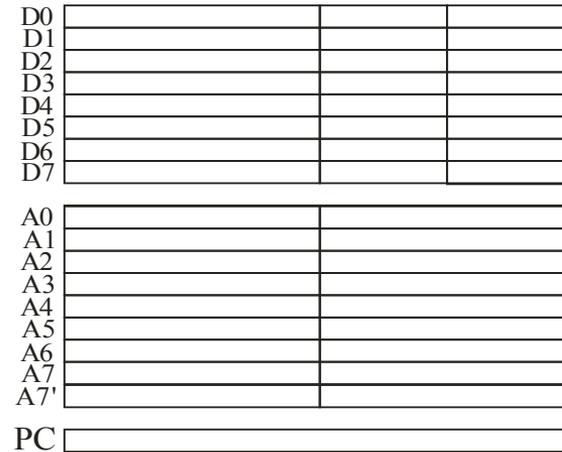
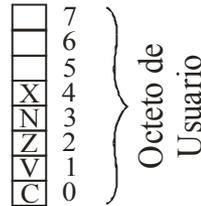


Juego de Instrucciones

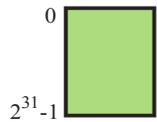
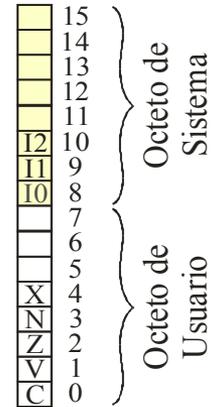
Niveles de ejecución



Registro de estado



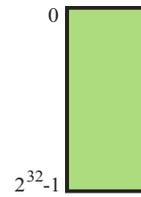
Registro de estado



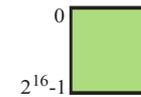
Mapa de memoria



Juego de Instrucciones



Mapa de memoria



Mapa de E/S

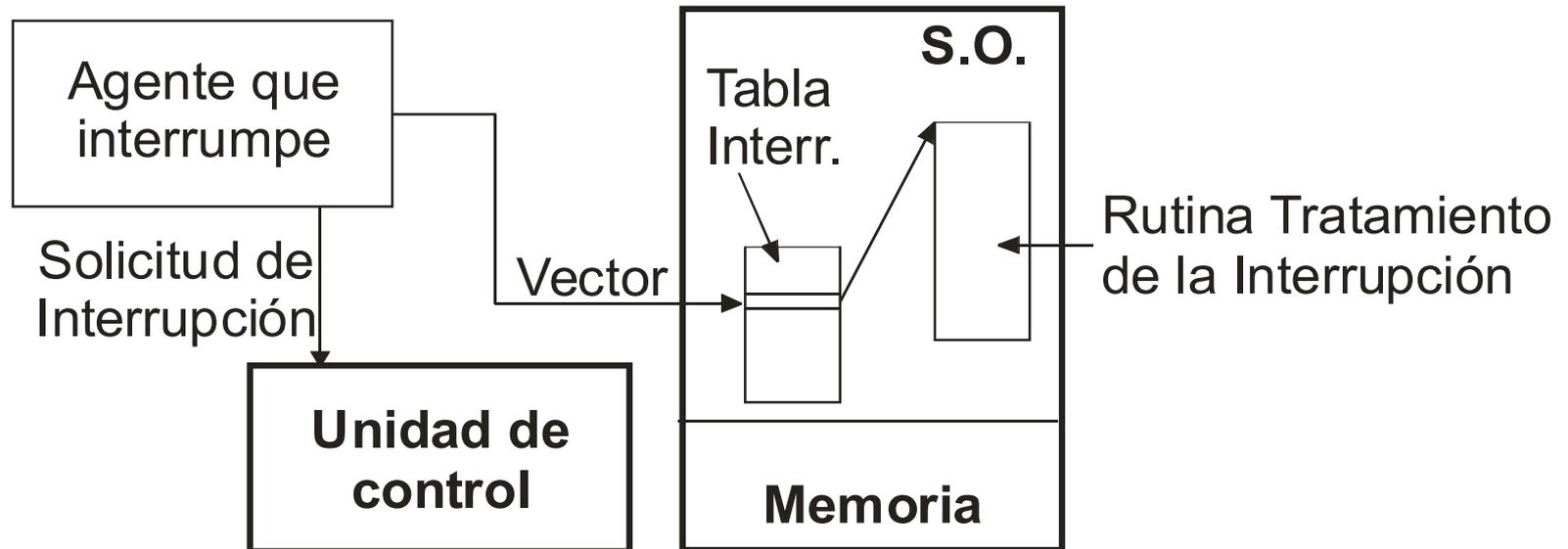


Juego de Instrucciones

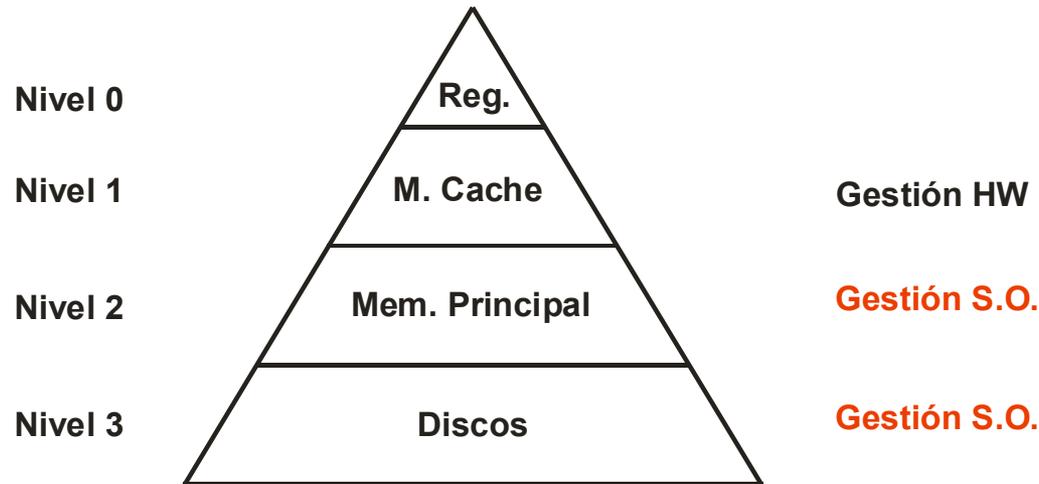
Modelo de programación de usuario

Modelo de programación de núcleo

Interrupciones

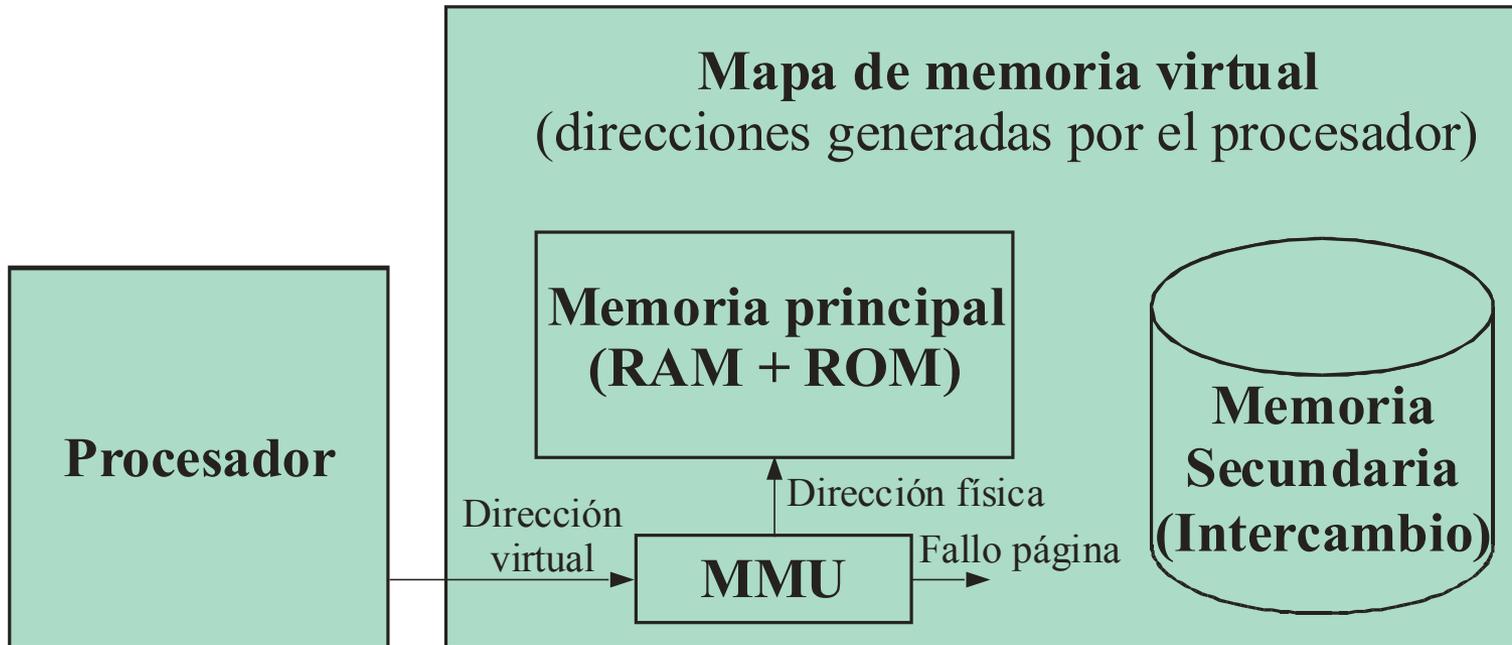


Jerarquía de Memoria

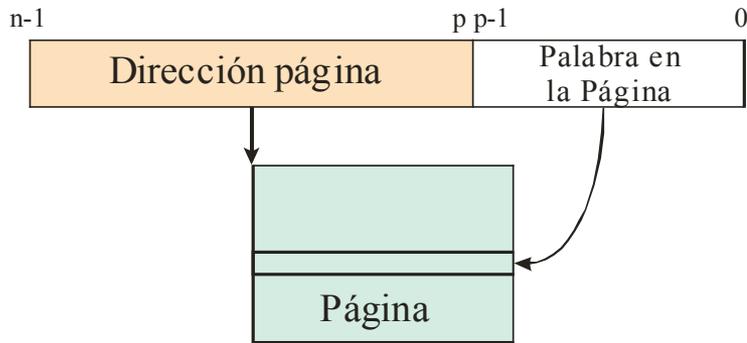
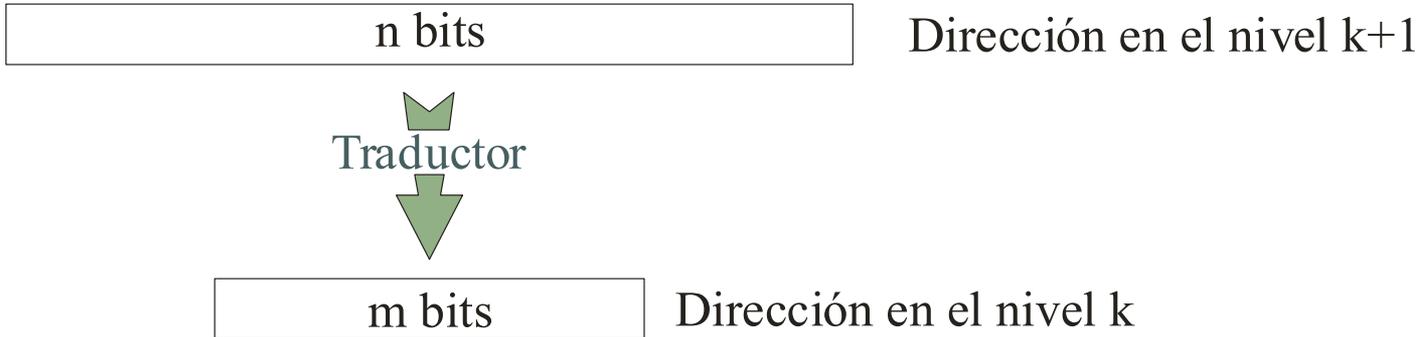


- Elementos de información replicados en varios niveles de la jerarquía
- Problema de coherencia
- Migración de la información: automática vs. bajo demanda
- Traducción de direcciones

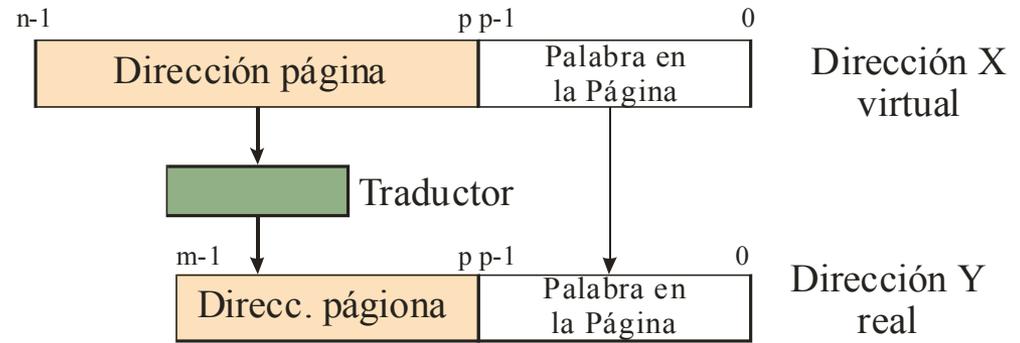
Memoria Virtual



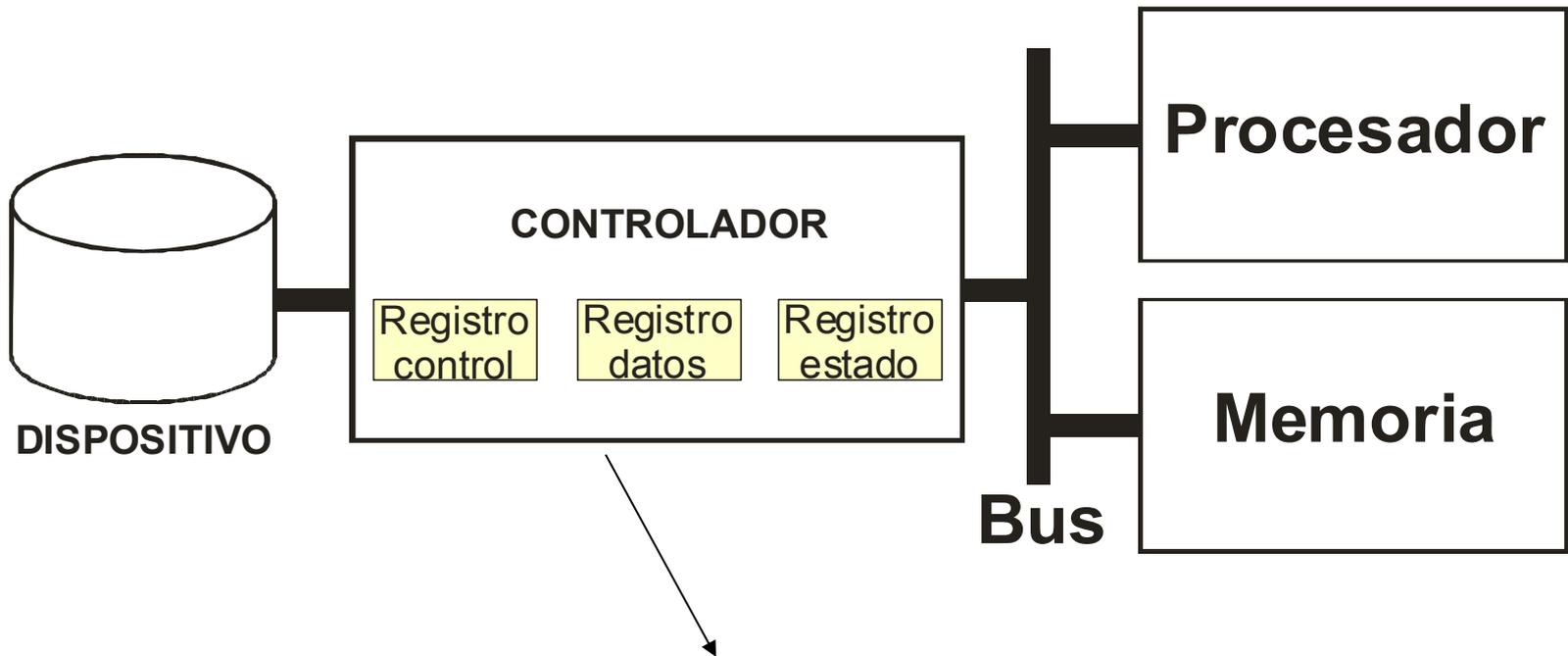
- La MMU realiza la traducción de PV a MP



División de la dirección



Traducción de la dirección



Módulo de E/S



1. ¿Por qué estudiar SSOO?
2. Estructura y funcionamiento de un computador.
- 3. Concepto de sistema operativo. Componentes y estructura del sistema operativo.**
4. Servicios del sistema operativo.
5. Arranque y activación del Sistema Operativo.
6. Evolución de los sistemas operativos.



¿Qué es un sistema operativo?

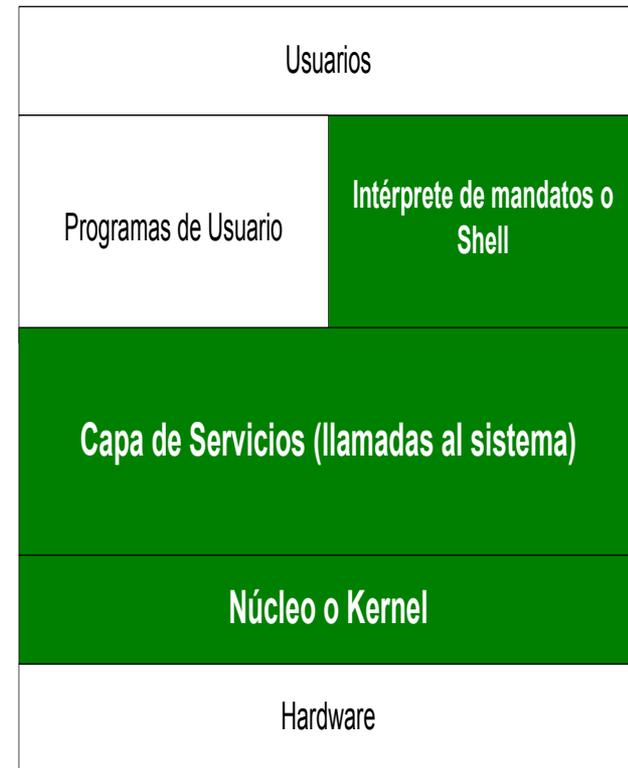
- Programa que actúa de intermediario entre el usuario del computador y el *hardware*.
- Objetivos:
 - Ejecutar programas.
 - Hacer un uso eficiente de los recursos.
 - Proporcionar visión de máquina virtual extendida.



- Gestor de recursos (UCP, memoria, ...)
 - Asignación y recuperación de recursos
 - Protección de los usuarios
 - Contabilidad/monitorización
 - Soporte de usuario
- Máquina extendida (servicios)
 - Ejecución de programas (procesos)
 - Órdenes de E/S
 - Operaciones sobre archivos
 - Detección y tratamiento de errores
- Interfaz de usuario
 - Shell

Niveles del sistema operativo

- El SO está formado conceptualmente por 3 capas principales:
 - Núcleo o *Kernel*
 - Servicios o llamadas al sistema
 - Intérprete de mandatos o *shell*

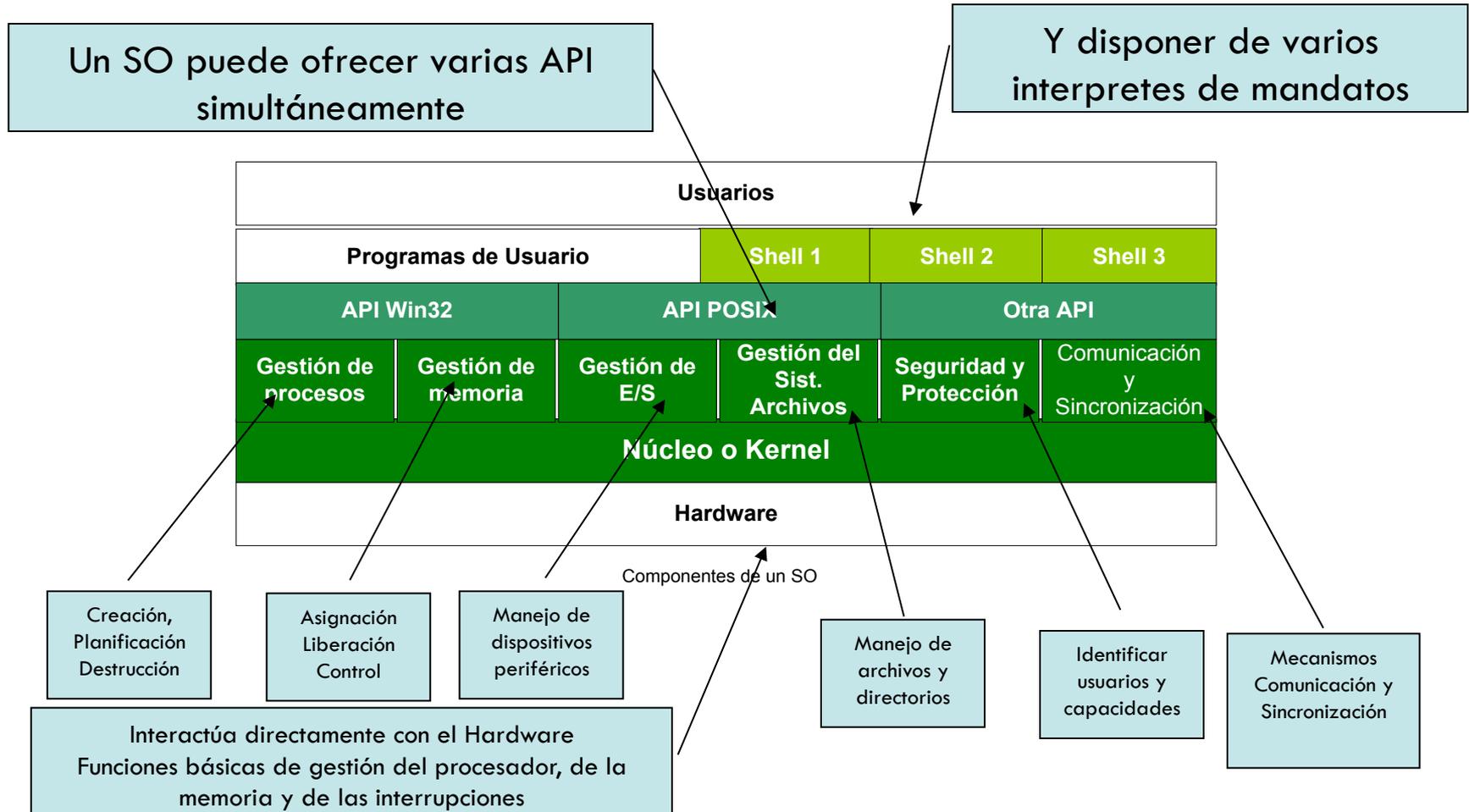


Estructura conceptual del SO



- Modos de ejecución:
 - Modo usuario: Ejecución de procesos de usuario.
 - Modo supervisor o núcleo: Ejecución del núcleo del SO.
- Los procesos y el SO utilizan espacios de memoria separados.
- Cuando un proceso necesita un servicio lo solicita al SO mediante una llamada al sistema.
 - El sistema operativo entra en ejecución para realizar la función solicitada.

Componentes del Sistema Operativo





Gestor de procesos

- Un proceso es un programa en ejecución.
- Un proceso necesita recursos, tales como UCP, memoria, ficheros, etc., para llevar a cabo su tarea.
- El SO tiene la responsabilidad de gestionar los siguientes aspectos de procesos:
 - Creación y destrucción.
 - Suspensión y reanudación.
 - Proporcionar mecanismos de sincronización y comunicación.
 - Asignación y mantenimiento de los recursos del proceso.



- Memoria: vector enorme de palabras o bytes, cada uno con su propia dirección.
 - Compartido por UCP y dispositivos E/S.
 - Volátil: pierde su contenido si el sistema falla.
- El SO tiene la responsabilidad de gestionar los siguientes aspectos de la memoria:
 - Mantener un mapa de las partes de memoria en uso y saber quién las está usando.
 - Decidir qué procesos se deben cargar, y dónde, cuando hay memoria disponible.
 - Asignar y liberar espacio de memoria cuando sea necesario.



Gestor de E/S y almacenamiento secundario

- El gestor de E/S está formado por:
 - Un sistema global de almacenamiento intermedio en memoria.
 - Manejadores genéricos, uno por cada clase, de dispositivos.
 - Manejadores específicos para cada dispositivo.
- Almacenamiento secundario *no volátil* en dispositivos rápidos de E/S (discos, NAD, etc.) como respaldo de la memoria.
- El SO tiene la responsabilidad de gestionar los siguientes aspectos de la E/S y el almacenamiento secundario:
 - Traducir peticiones a formato de manejador.
 - Copiar memoria de/a memoria a/de controlador.
 - Controlar operaciones por DMA.
 - Controlar dispositivos de E/S serie: teclado, ratón, etc.
 - Asignación y liberación de espacio.

Planificación de accesos a los dispositivos.



Servidor de ficheros y directorios

- *Fichero*: conjunto de información lógicamente relacionada y definida por su creador.
- *Directorio*: conjunto de identificadores lógicos de ficheros y su relación con identificadores internos del SO.
- Ficheros frecuentes: programas (fuentes y ejecutables) y datos.
- El SO tiene la responsabilidad de gestionar los siguientes aspectos del servicio de ficheros:
 - Creación y borrado de ficheros y directorios.
 - Primitivas para manipular ficheros y directorios.
 - Proyectar los ficheros sobre almacenamiento secundario.
 - Hacer respaldo de ficheros sobre dispositivos estables y no volátiles.



Servidor de protección y seguridad

- *Protección*: controlar accesos a los recursos de usuario y sistema.
- El servidor de protección debe:
 - Distinguir entre uso autorizado y no autorizado.
 - Especificar los controles de acceso a llevar a cabo.
 - Proporcionar métodos de control de acceso.
- *Seguridad*: proteger al sistema de un uso indebido.
- El servidor de seguridad debe:
 - Autenticar a los usuarios.
 - Evitar amenazas al sistema (gusanos, virus, piratas, fuego, etc.)
 - Evitar la interceptación de comunicaciones: cifrado, canales seguros, etc.

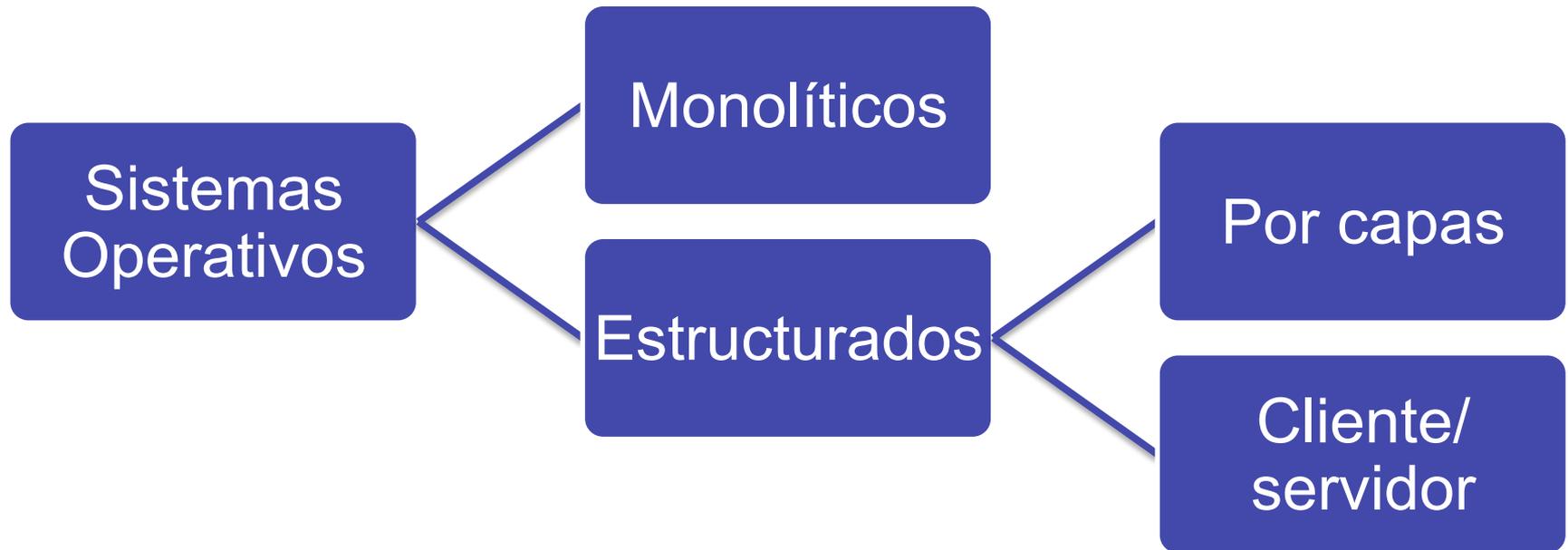


Servidor de comunicaciones

- Sistema en red o distribuido: conjunto de procesadores que no comparten memoria.
- Conectados mediante una *red de comunicaciones*.
- Proporciona mecanismos de comunicaciones locales y remotas para distintos tipos de redes: Ethernet, ATM, telefónica, etc.
- Proporciona acceso a recursos de la red, lo que permite:
 - Acelerar la computación.
 - Incrementar la disponibilidad de datos.
 - Mejorar la fiabilidad.
 - Abaratar el sistema.
- Responsabilidad del SO: resolución de nombres, enrutamiento, conexiones y control de flujo.



Alternativas de estructura





- No hay una estructura clara y bien definida.
- Todo el código del SO está enlazado como un único ejecutable (un solo espacio de direcciones) que se ejecuta en modo “núcleo”.
- El código presenta cierta organización pero internamente no existe ocultación de información entre los distintos módulos, pudiéndose llamar unos a otros sin restricciones
- Aunque es más eficiente en su funcionamiento, su desarrollo y mantenimiento es muy complejo.
- Ejemplos:
 - Todos los SO hasta los 80, incluido UNIX
 - MS-DOS y variantes actuales de UNIX: Solaris, Linux, AIX, HP-UX,...

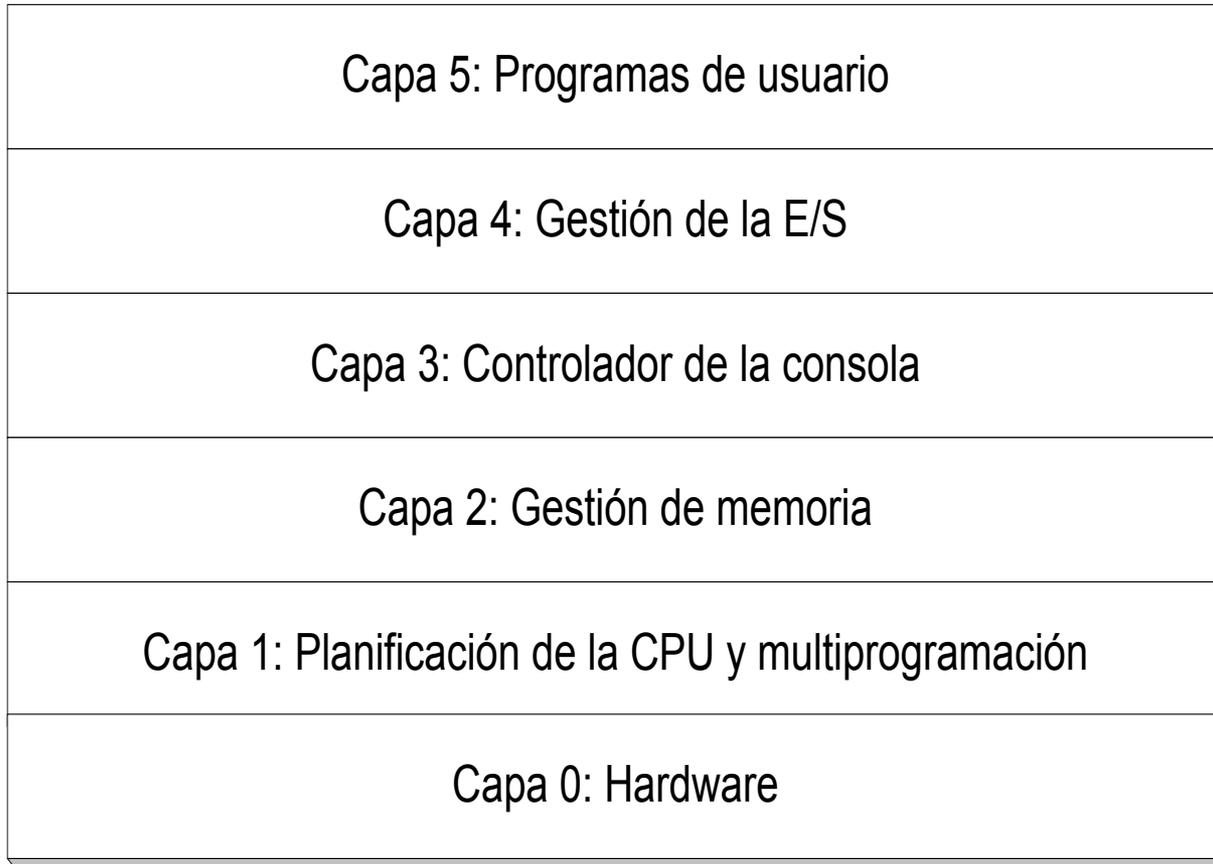


Sistemas Operativos estructurados por capas

- El sistema se organiza como un conjunto de capas superpuestas, cada una con una interfaz clara y bien definida
- Cada capa se apoya en los servicios de la inmediatamente inferior para realizar sus funciones
- Las ventajas son la modularidad y la ocultación de la información, que facilita mucho el desarrollo y la depuración de cada capa por separado.
- Esta estructura, sin embargo, no resulta tan eficiente porque una determinada operación en la capa superior implica realizar múltiples llamadas desde el nivel superior hasta el inferior.
- Dificultad a la hora de distribuir las distintas funciones del SO entre las distintas capas
- Ejemplos:
 - THE
 - OS/2



Sistemas Operativos estructurados por capas



Estructura por capas del sistema operativo THE



Sistemas Operativos estructurados: cliente/servidor

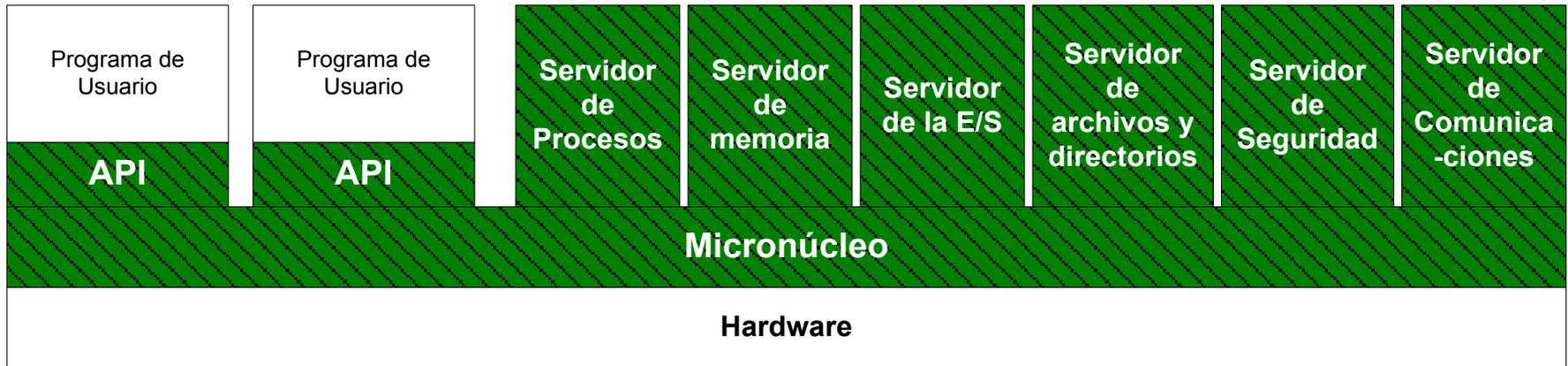
- Implementar la mayor parte de los servicios del SO como procesos de usuario, dejando solo una pequeña parte corriendo en modo núcleo denominada **micronúcleo** o **microkernel**
- Hay dudas sobre qué funciones debe implementar realmente el microkernel pero al menos: interrupciones, gestión básica de procesos y memoria y servicios básicos de comunicación
- Ventajas
 - Muy flexible. Cada servidor puede desarrollarse y depurarse más fácilmente al tratarse de programas pequeños y especializados.
 - Es fácilmente extensible a un modelo distribuido
- Desventajas
 - Sobrecarga en la ejecución de los servicios
- Ejemplos:
 - Minix y Amoeba (Tanenbaum)
 - Mac OS y Windows NT, aunque en realidad los servicios se ejecutan en espacio kernel para no penalizar el rendimiento → ¿Microkernel?



Sistemas Operativo estructurados: cliente/servidor

Procesos
clientes

Procesos
Servidores



Estructura cliente-servidor en un sistema operativo

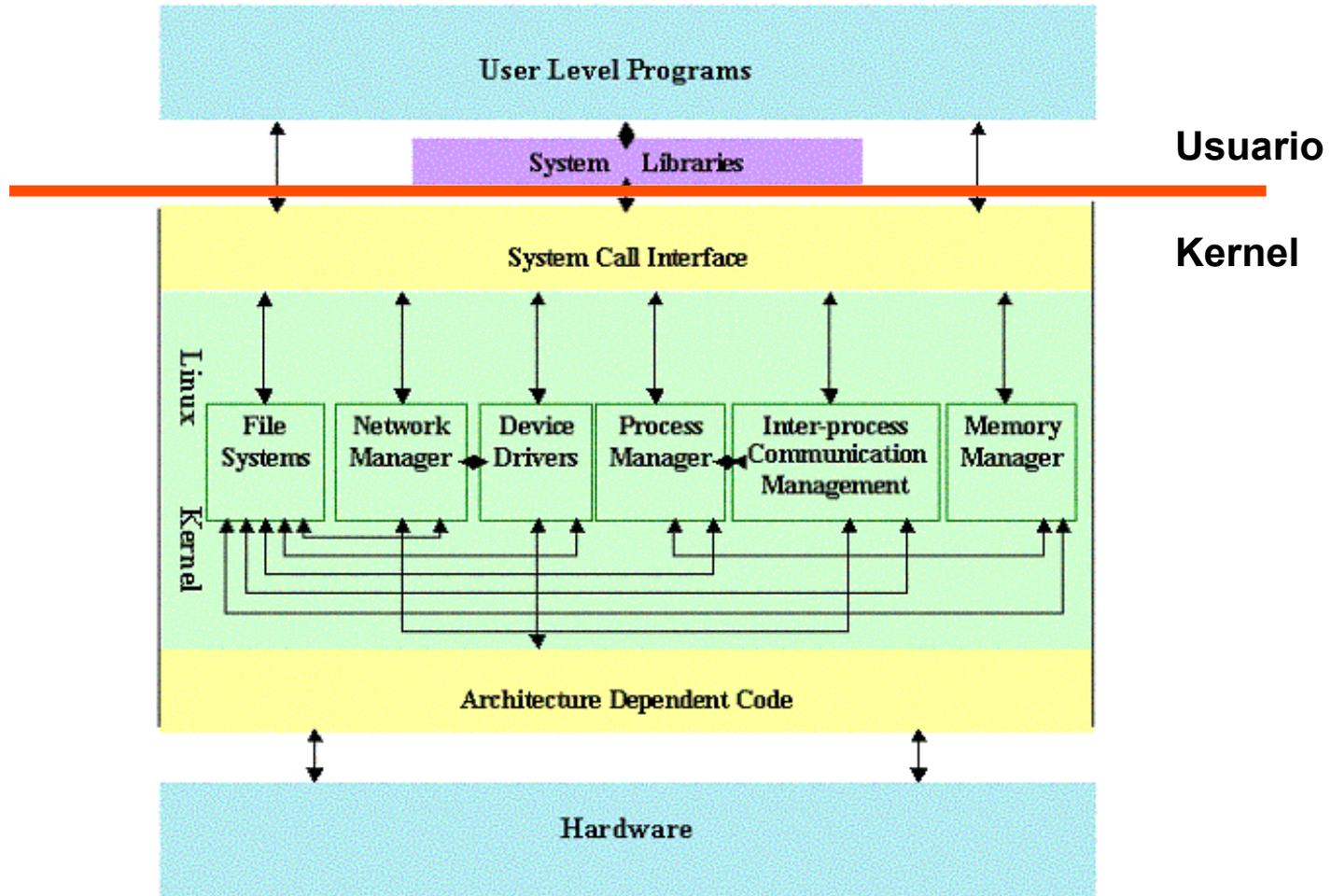


Clasificación de Sistemas Operativos

- Número de procesos simultáneos:
 - Monotarea.
 - Multitarea.
- Modo de interacción:
 - Interactivo.
 - Por lotes (batch).
- Número de usuarios simultáneos:
 - Monousuario.
 - Multiusuario.
- Número de procesadores:
 - Monoprocesador.
 - Multiprocesador.
- Número de hilos (*threads*):
 - *Monothread*.
 - *Multithread*.
- Tipo de uso:
 - Cliente.
 - Servidor.
 - Empotrado.
 - Tiempo real.

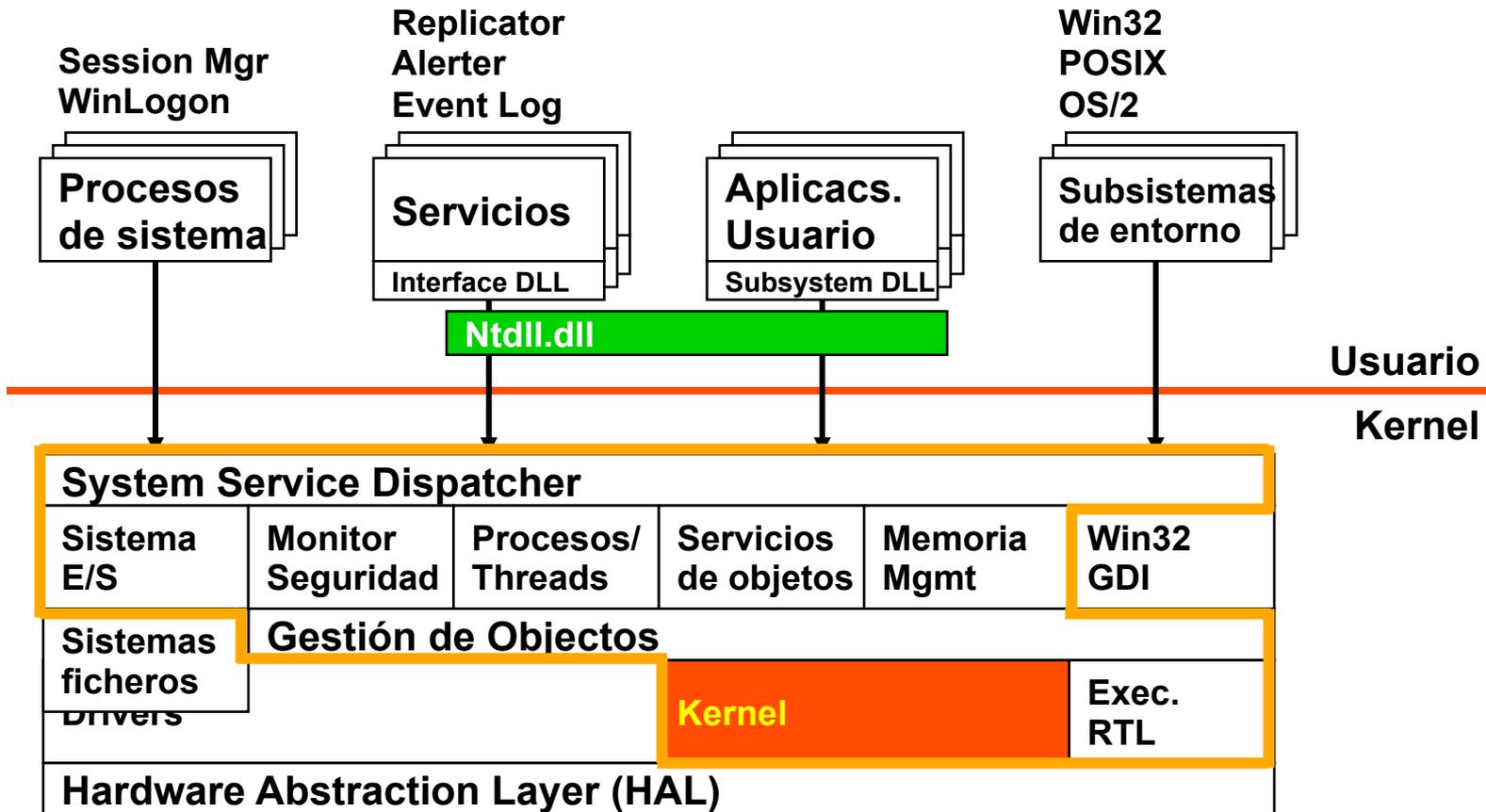


LINUX. Arquitectura





Windows NT. Arquitectura





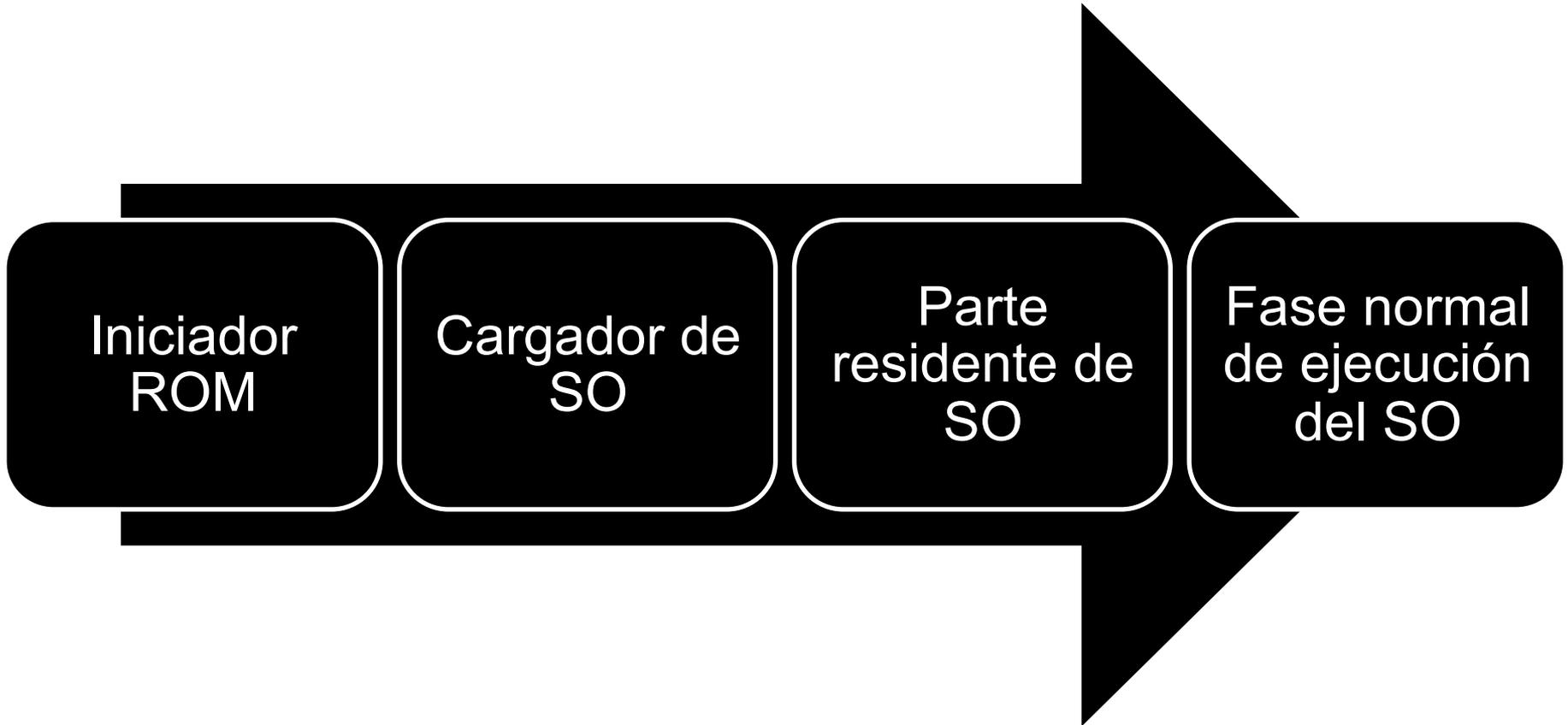
1. ¿Por qué estudiar SSOO?
2. Estructura y funcionamiento de un computador.
3. Concepto de sistema operativo. Componentes y estructura del sistema operativo.
- 4. Arranque del Sistema Operativo.**
5. Evolución de los sistemas operativos.



- El sistema operativo se inicia al encender el computador.
 - Inicialmente se encuentra en almacenamiento secundario.
 - ¿Cómo llega el sistema operativo a memoria principal?
 - ¿Cómo se hace para iniciar su ejecución una vez que se encuentra en memoria?



Fases en el arranque

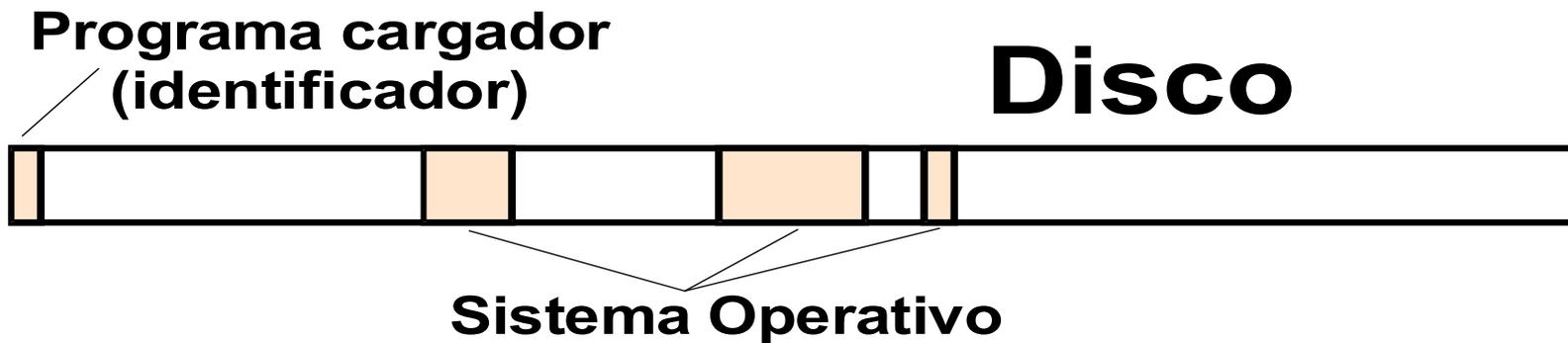




- La señal RESET carga valores predefinidos en registros.
 - CP ← dirección de arranque del cargador ROM
- Se ejecuta el iniciador ROM del sistema:
 - Test hardware del sistema
 - Trae a memoria el boot (iniciador) del SO

Cargador del sistema operativo

- El programa cargador se encuentra en el sector de inicio (boot) del disco.
- Es responsable de cargar el sistema operativo.
- Verifica la presencia de palabra mágica en sector de arranque.





- Responsable de la iniciación del sistema operativo.
 - Verificación de consistencia del sistema de ficheros.
 - Creación de las estructuras de datos internas.
 - Activación de modo de memoria virtual.
 - Carga el resto del sistema operativo residente.
 - Habilita interrupciones.
 - Crea procesos iniciales.



Ejemplo: Procesos iniciales en Linux

- Proceso *init*.
 - Proceso inicial ancestro de los demás procesos.
- Procesos de *login*.
 - Uno por terminal.
- Procesos demonio
 - Ejemplo *httpd*.
- Tras la autenticación el proceso de *login* se transforma en proceso *shell*.

- El SO se activa (*) por
 - Llamadas (“al Sistema”) realizadas por los programas
 - Interrupciones





Activación del SO

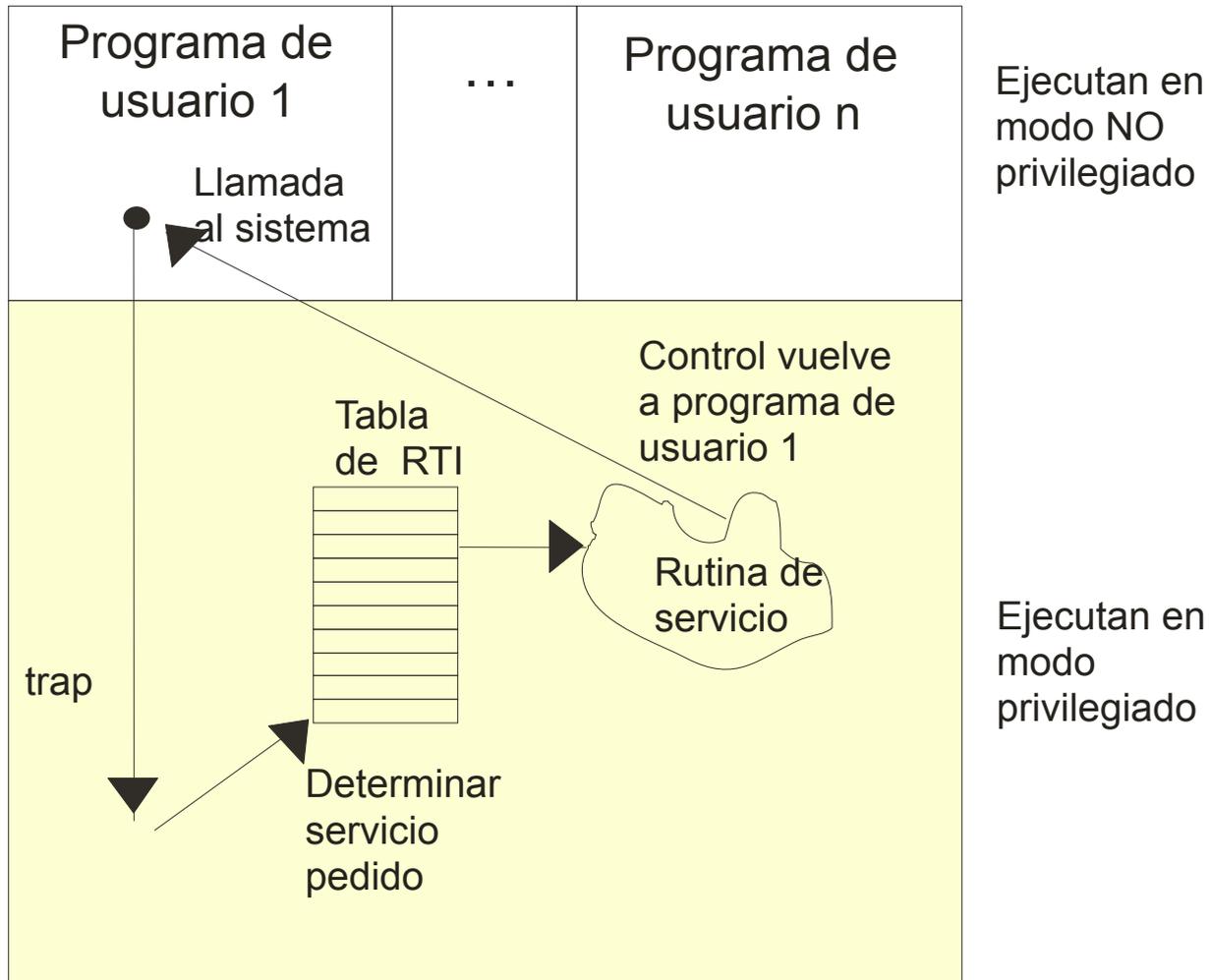
- El SO se activa por
 - Llamadas realizadas por los programas
 - Interrupciones producidas por los periféricos
 - Condiciones de excepción o error del hardware
- Forma de pedir
 - Por llamada → problema de seguridad
 - Por interrupción o *trap*
 - Rutina de biblioteca:
 - Instrucciones de máquina que preparan la llamada al SO



Llamadas al sistema

- Interfaz entre aplicaciones y SO.
 - Generalmente disponibles como funciones en ensamblador.
 - Actualmente en otros lenguajes de alto nivel (C, Bliss, ...).
- Tres métodos genéricos para pasar parámetros a las llamadas al sistema:
 - En registros.
 - En una tabla de memoria, cuya dirección se pasa al SO en un registro.
 - Poner los parámetros en la pila del programa y dejar que el SO los extraiga.
- Cada SO proporciona sus propias llamadas al sistema:
 - Estándar POSIX en UNIX y LINUX.
 - Win32 en Windows NT.

Paso de llamadas al sistema





- Proporcionan un entorno adecuado para el desarrollo y ejecución de programas.
- Categorías:
 - Manipulación de ficheros: copia, creación, mover, ...
 - Información de estado: monitores, ...
 - Apoyo a la programación: compiladores, depuradores, ...
 - Carga y ejecución de programas: cargador, ...
 - Comunicaciones: telnet, ftp, ...
 - Programas de aplicación: editores, servidores BD, ...
- La visión del SO que tienen muchos usuarios es la de los programas de sistema, no la de las llamadas al sistema.



Parada del computador

- Para acelerar la ejecución el sistema operativo mantiene información en memoria no actualizada a disco.
 - Al apagar hay que volcar dicha información a disco y terminar la ejecución de todos los procesos.
 - Si no se hace volcado (apagado brusco)
 - Pérdida de información.
 - Sistema de ficheros en estado inconsistente.
- Otras alternativas en computadores personales:
 - **Hibenación**: Se guarda estado de la memoria principal a disco.
 - **Apagado en espera** (standby): Parada del computador que mantiene alimentada la memoria principal.

Generación del sistema operativo

- Los sistemas operativos suelen diseñarse para en una clase de máquinas que incluya diversas configuraciones y una amplia variedad de periféricos.
- Es necesario generar una copia del sistema operativo basada en las características de configuración de la máquina.
- La generación del sistema operativo se realiza durante la instalación inicial.





Parámetros de generación

- Modelo de CPU.
- Opciones de CPU instaladas.
- Número de CPUs.
- Cantidad de memoria disponible.
- Dispositivos instalados.
- Opciones del sistema operativo.



Alternativas de generación

- **Modificación del código fuente y compilación del sistema operativo.**
 - Ejecutable totalmente adaptado.
 - Generación más lenta y arranque más rápido.
- **Creación de tablas y selección de módulos de biblioteca precompilada.**
 - El sistema operativo tiene controladores para todos los dispositivos.
 - Solamente se montan los necesarios.
 - Generación más rápida y arranque más lento.
- **Sistema totalmente controlado por tablas.**
 - Selección en tiempo de ejecución.
 - Arranque más lento.



1. ¿Por qué estudiar SSOO?
2. Estructura y funcionamiento de un computador.
3. Concepto de sistema operativo. Componentes y estructura del sistema operativo.
4. Arranque del Sistema Operativo.
- 5. Evolución de los sistemas operativos.**



Evolución: máquina desnuda (1950)

- Estructura
 - Grandes máquinas manejadas por consola
 - Sistemas monousuario y monoprogramados
 - Programador, usuario, operador
 - Cinta de papel o tarjetas perforadas
- Software
 - Ensambladores, compiladores, enlazadores y cargadores
 - Bibliotecas de subrutinas comunes
 - Manejadores de dispositivo
- Rendimiento
 - Uso ineficiente de recursos caros
 - Uso UCP muy pobre
 - Largo tiempo de activación
 - Ventaja: seguridad



Sistema sencillo por lotes

- Lectora de tarjetas como dispositivo de entrada.
- Reduce el tiempo de activación agrupando trabajos similares.
- Secuenciación automática de trabajos: rudimento de SO que transfiere control automáticamente de una tarea a otra.
- Monitor residente:
 - Control inicial en el monitor.
 - Se transfiere control a una tarea.
 - Cuando la tarea termina, el control vuelve al monitor.
- Usuario != operador. El operador es un prestador de servicios -> nace el centro de cálculo.
- ¿Cómo sabe el monitor que hacer? Mediante tarjetas de control.
- Monitor: intérprete de tarjetas de control, cargador y manejadores de dispositivos.

Sistema sencillo por lotes

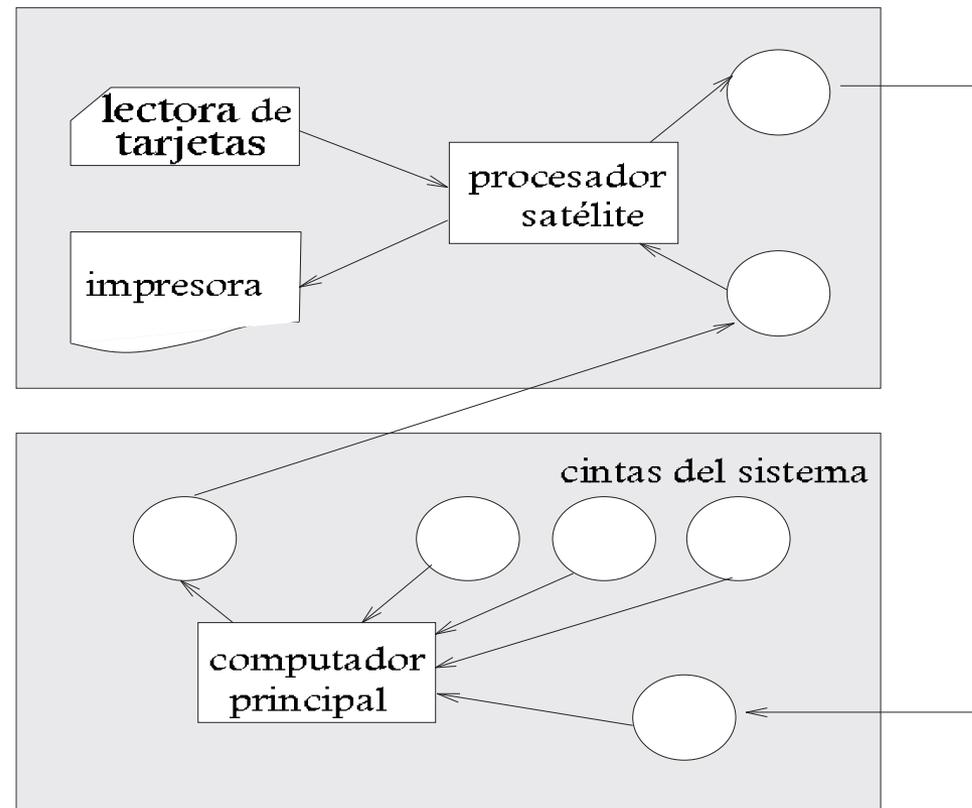
- Problema: E/S y UCP no solapan -> rendimiento bajo
- Solución: E/S lenta (tarjeta, impresora) off-line y rápida (carga) a cinta

Ventajas:

- UCP sólo limitada por cintas
- Los programas no cambian
- Múltiples procesadores satélites para una UCP

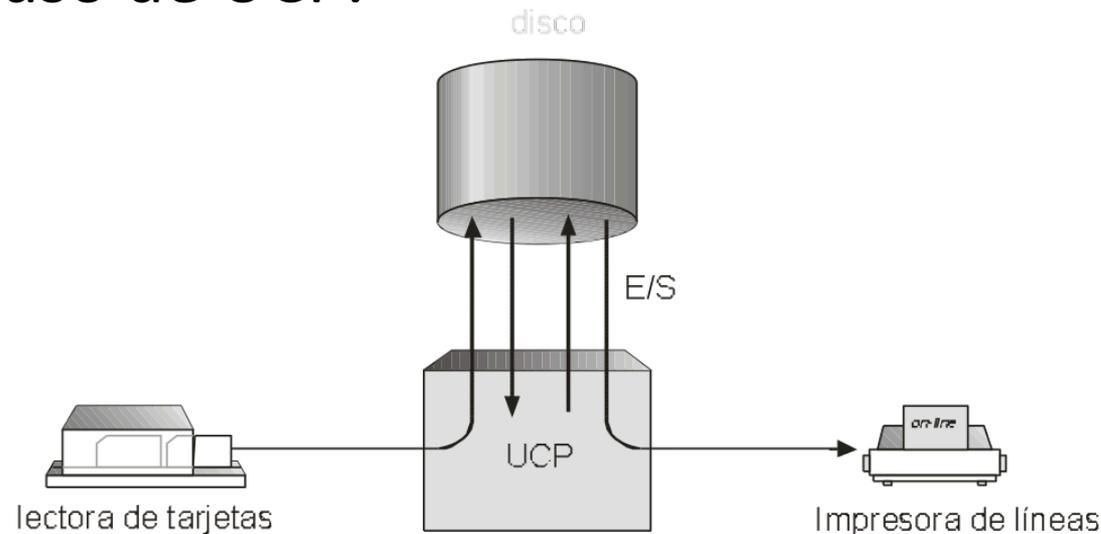
Problemas:

- E/S y UCP no solapan
- Sistema monoprogramado



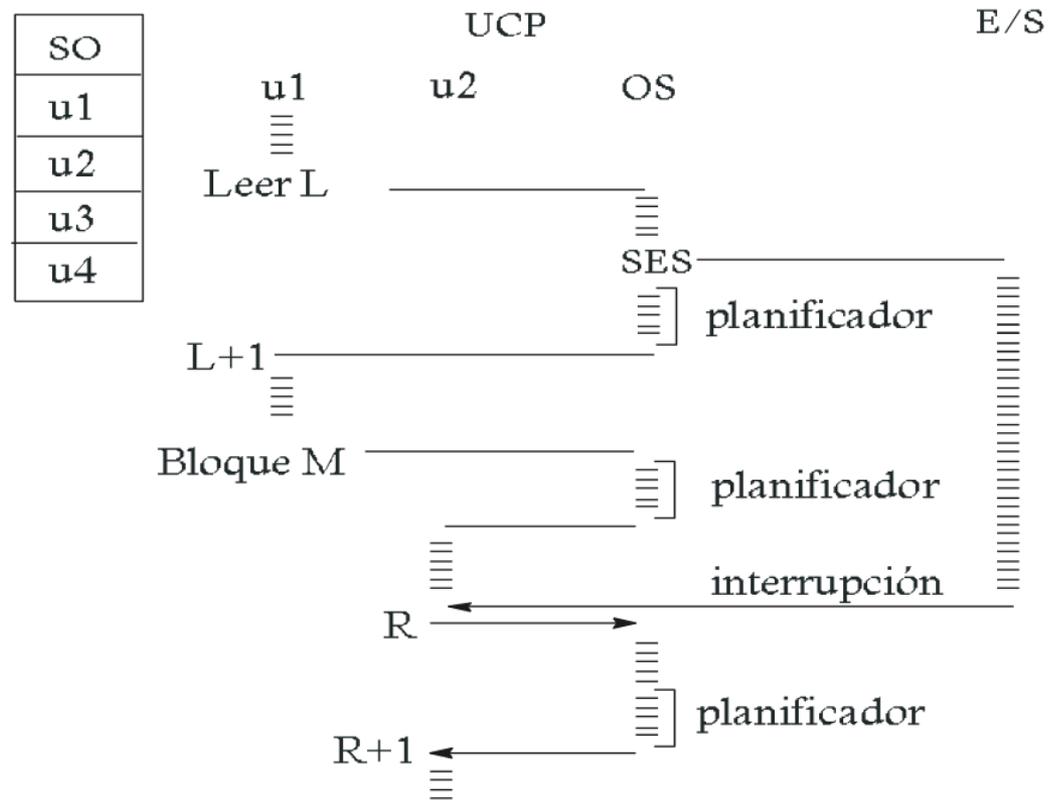
Spooling

- Usar UCP durante E/S -> mientras ejecuta un trabajo, el SO lee el siguiente e imprime la salida del anterior.
- Cola de tareas: estructura de datos que permite al SO seleccionar qué trabajo ejecutar el siguiente para optimizar uso de UCP.



Sistema por lotes multiprogramado

- Varios trabajos en memoria principal al mismo tiempo.
- Se multiplexa la UCP entre ellos.





Características de un SO multiprogramado

- Rutina de E/S propia del sistema.
- Gestión de memoria: el sistema debe asignar la memoria a varios trabajos.
- Planificación de la UCP: el sistema debe elegir entre varios trabajos listos para ejecutar.
- Asignación de dispositivos.
- ¡¡Todos los SO actuales son multiprogramados!!



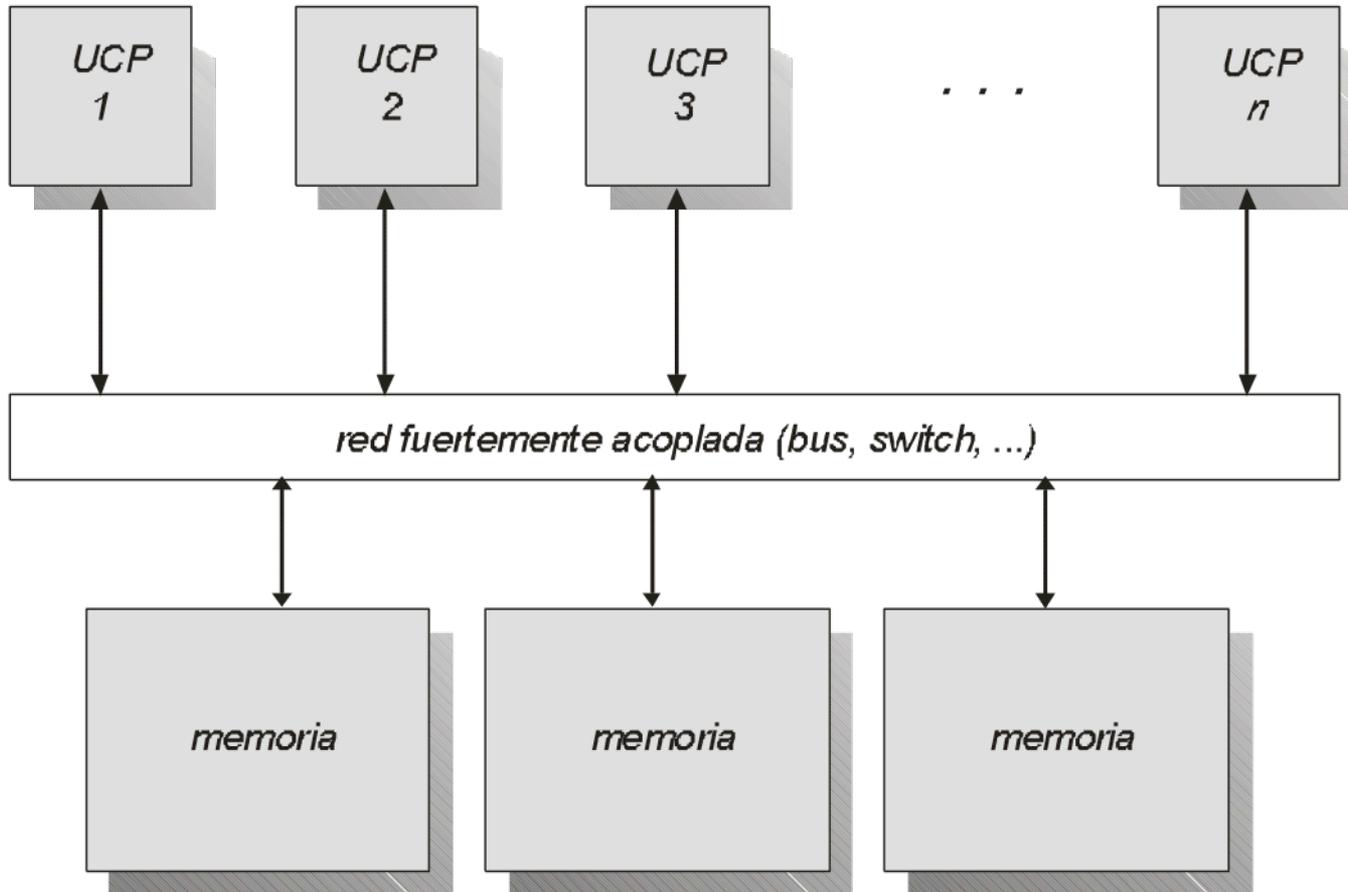
Sistemas de tiempo compartido

- La UCP se multiplexa entre varios trabajos residentes en memoria.
- La UCP se asigna sólo a trabajos residentes en memoria.
- Los trabajos se intercambian entre disco y memoria.
- Interactividad: comunicación *on-line* entre usuario y sistema.
- Aparece el intérprete de comandos, que sustituye a tarjetas de control.
- Sistema de ficheros *on-line* para permitir a los usuarios el acceso a datos y código.
- Multiusuario: los trabajos son de varios usuarios, cada uno de los cuales cree tener todo el computador.

- Sistemas dedicados a un único usuario (monousuario) y multiprogramados (generalmente ...)
- Dispositivos de E/S: teclado, ratón, pantallas gráficas, impresoras pequeñas, ...
- Convenientes para el usuario y muy interactivos.
- Adoptan tecnología de grandes sistemas operativos.
- Hasta ahora, sistemas aislados -> menos atención seguridad.
- Actualmente:
 - UCP muy potente (500 Mhz)
 - Mucha memoria (256 Mbytes)

- Multiprocesadores:
 - varias UCPs fuertemente acopladas -> reloj común.
 - Memoria compartida -> comunicación y sincronización.
 - Sistema operativo único para todas las UCPs.
- Ventajas:
 - Incremento de rendimiento.
 - Económicos
 - Incremento de fiabilidad: degradación parcial, tolerancia a fallos.
- Tipos:
 - Supercomputadores (64 hasta 4096 UCPs).
 - Multiprocesadores (2 hasta 64 UCPs).
 - Computadores personales paralelos (2 hasta 8 UCPs).

Sistemas paralelos II





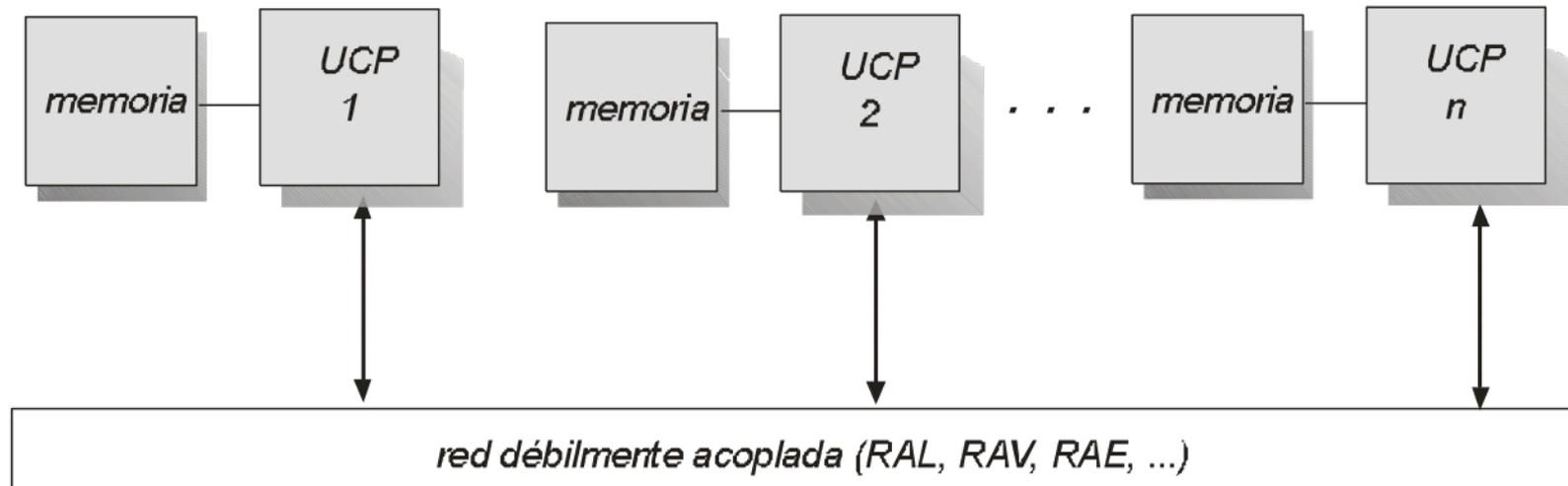
- Multiproceso simétrico:
 - Cada procesador ejecuta una copia idéntica del SO.
 - Muchos procesos simultáneos sin pérdida de rendimiento.
 - Cada procesador ejecuta el primer trabajo listo para ejecutar, no hay jerarquía.
- Multiproceso asimétrico:
 - A cada procesador se le asigna una tarea específica.
 - Un procesador maestro planifica el trabajo de los esclavos.
 - Muy frecuente en supercomputadores.
- Características SO:
 - Planificador más complejo.
 - Gestión de memoria más compleja.
 - Gestión de caches de memoria muy importante para rendimiento.



Sistemas distribuidos I

- Distribuyen la computación entre varios procesadores físicamente separados, que NO comparten memoria.
- Sistemas débilmente acoplados: comunicación mediante mensajes, no memoria -> paradigma programación nuevo.
- Ventajas:
 - Compartición de recursos: impresoras, discos, cintas, etc.
 - Reparto de carga -> acelerar la computación.
 - Fiabilidad: replicación y tolerancia a fallos .
 - Modelo de comunicación no limitado: el mismo programa vale para una red local o extendida.
- Tipos:
 - red de computadores conectados con una RAL.
 - Clusters conectados por una RAV.

- Cada nodo tiene su SO independiente.
- El usuario NO debería saber donde está trabajando.





- Restricciones de tiempo bien definidas.
- Clave: ejecutar tareas en el tiempo y orden fijado.
- STR críticos (avión, satélite, médicos, etc.):
 - No pueden perder ningún *deadline* de tarea.
 - Almacenamiento secundario limitado o ausente -> no MV.
 - Sistemas dedicados, no de propósito general.
- STR suave (multimedia, comunicaciones, etc.):
 - Pueden perder *deadlines* de tareas no críticas.
 - Uso limitado en sistemas de control: industria, robots, etc.
 - Aplicaciones que necesitan características avanzadas del SO.
- Uso: dispositivos de control, sistemas de control industrial, medicina, etc. ¡¡Muy extendidos!!



SISTEMAS OPERATIVOS:

INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS



¿Objetivo Cumplido?

- Comprender de forma global la estructura y funcionamiento del computador.
- Recordar los elementos del computador que tienen impacto en el sistema operativo.
- Comprender qué es un Sistema Operativo.
- Conocer los principales componentes del Sistema Operativo.
- Comprender el proceso de arranque del sistema operativo.
- Conocer la evolución de los sistemas operativos