

SISTEMAS OPERATIVOS: INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS

Introducción y conceptos básicos

ADVERTENCIA

2

- 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.

Objetivos

SO: facilitar al usuario el uso
del computador.
Antes se encargaba de controlar
y comunicarse con el hardware

3

- 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.

Contenido

4

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.

¿Por qué hay que saber de S.S.O.O.?

El sistema Op. se encarga de ofrecernos todo, todo está sobre él.

5

- a) El SO, sus peculiaridades internas, influye de mucho en el funcionamiento general, en la seguridad y/o rendimiento del computador.
~ Por ello es importante conocer como funciona.
- b) La importancia de la elección de un determinado SO para una empresa es cada día mayor, casi estratégica.
↳ Para saber cual se adapta mejor a nuestras necesidades.
- c) Conocer el funcionamiento del SO es fundamental para desarrollar aplicaciones que obtengan buenas prestaciones y para comprender la causa de muchos problemas.
↳ Para saber como se comportará con nuestro trabajo.

Para comprender el funcionamiento del sistema

6

- ¿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

7

- ¿Es fácil encontrar administradores para este SO? ¿La administración es una tarea “oscura” y exclusiva de personal ultra-especializado?
Lo primero es proteger la inversión, hay que tener a alguien que sepa administrar, ofrece soporte, está actualizado y sea rentable
- ¿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

8

- 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. *necesito conocimiento de ello.*
 - ¿Cómo se desarrolla una aplicación multi-hilo para mi SO?

Un ingeniero debe ser

9



symbian
OS



redhat

No tener ideas
preconcebidas
El que más se
adapta a lo
que recibe



Windows
Mobile

Hacer una checklist
por cual se adapta
mejor.

AGNÓSTICO EN SISTEMAS OPERATIVOS



Mac OS



Contenido

10

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.

Tarea 1.1

11

- Lea el capítulo 1 del libro [Carretero 2007].
 - 1. Conceptos Arquitectónicos del Computador.

Contenido

12

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.

¿Qué es un sistema operativo?

Ocultar hardware y controlarlo para ofrecer servicios al usuario. Transforma unas cosas que son muy distintas.

13

- 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.

Funciones del sistema operativo

14

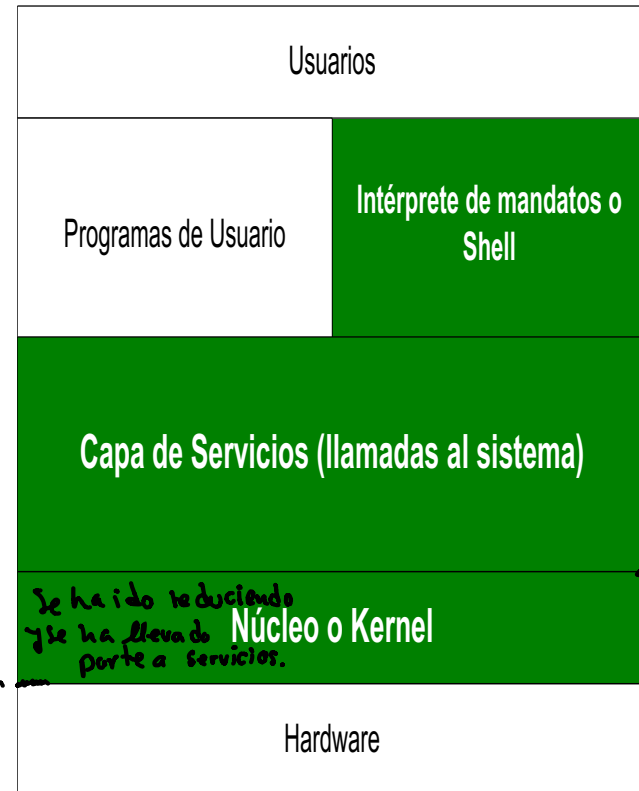
- **Gestor de recursos** (UCP, memoria, ...)
 - **Asignación y recuperación de recursos** ~ la memoria
 - **Protección de los usuarios** ~ El acceso, login y password, también los permisos ^{controla}
 - **Contabilidad/monitorización** ~ Para ver como está el sistema y conocer lo que pasa. ^{del usuario.}
 - Soporte de usuario
- **Máquina extendida (servicios)**
 - **Ejecución de programas** (procesos)
 - **Órdenes de E/S** ~ Leer/escibir ^{Activ. ...}
 - **Operaciones sobre archivos** ^{ficheros}
 - **Detección y tratamiento de errores** ~ errores que pueden ser reconocidos.
- **Interfaz de usuario**
 - **Shell** ~ Interfaz ^{externa al sistema operativo.} gráfica

Niveles del sistema operativo

15

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

Hardware
- Abstract
- Logic
→ LAL
o:
firmware
Entre el HW y SO



Para pasar entre núcleo y servicio
↓
Mediante un API

Estructura conceptual

16

□ Modos de ejecución:

□ Modo **usuario**: Ejecución de procesos de usuario.

→ Está restringido, no puede acceder a ciertas aplicaciones GIS, seguridad

□ Modo supervisor o **núcleo**: Ejecución del núcleo del SO.

↳ Puede ejecutar cualquier aplicación, sin restricción.

□ Los procesos y el SO utilizan espacios de memoria

separados.

Y el SO controla que acceda a su espacio.

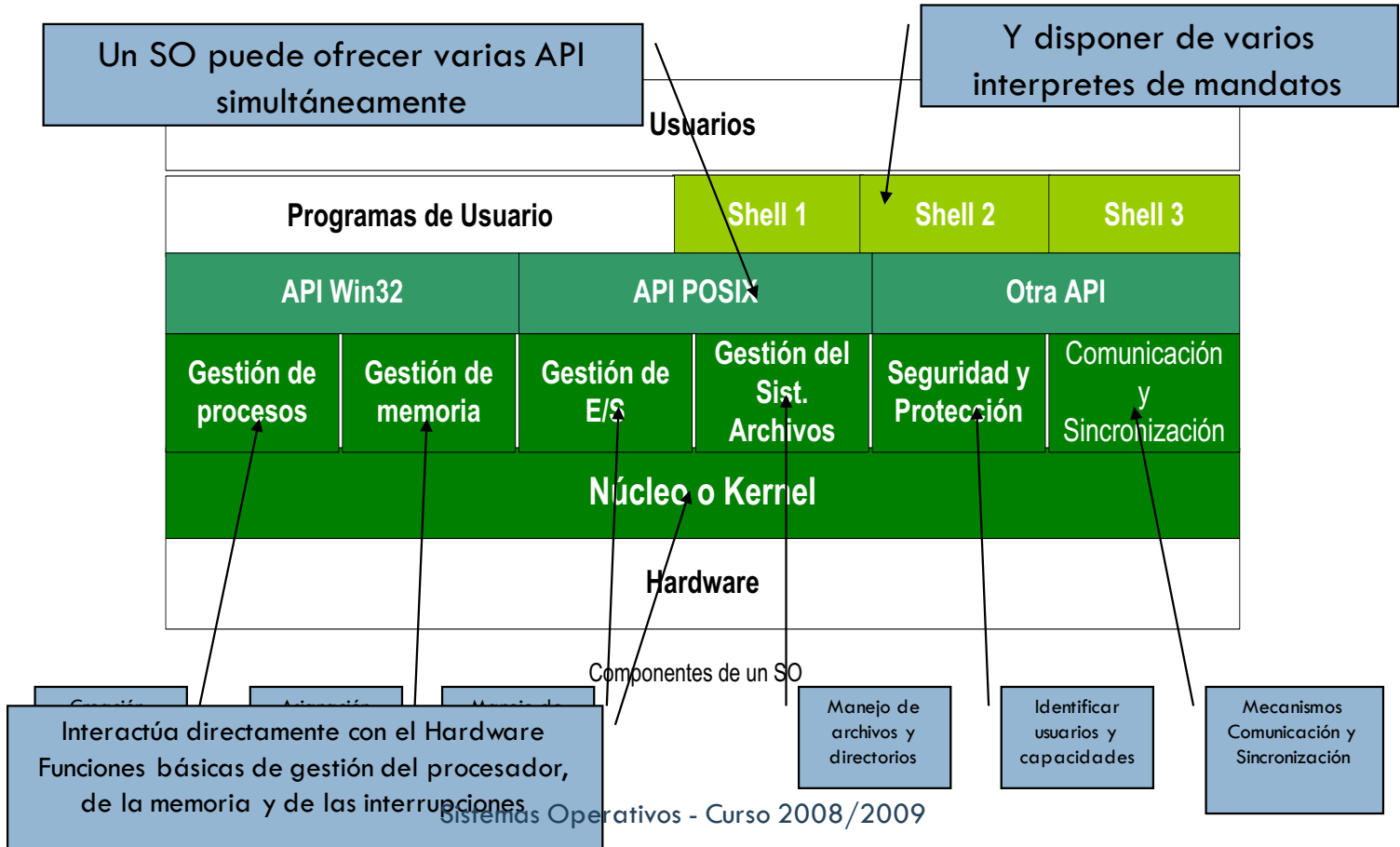
Cada aplicación tiene una parte de memoria RAM reservada, + y la del SO no puede modificarla a menos que tenga acceso privilegiado

□ 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

17



Alternativas de estructura

18

Principalmente se modifica el SO si: Añadir funcionalidades o Nuevo hardware.
Hay estándares para asegurar una base para que funcione correctamente aunque cambie estructura y siga siendo estándar.



Sistemas Operativos Monolíticos

19

- 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,...

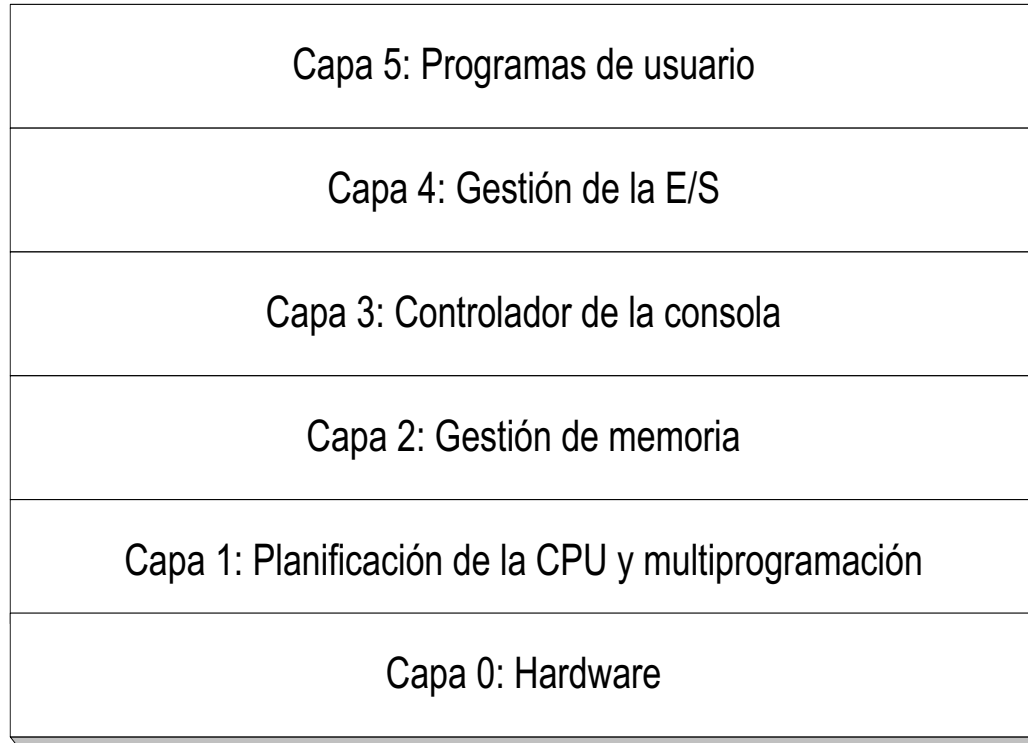
Sistemas Operativos estructurados por capas

20

- 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

21



Estructura por capas del sistema operativo THE

Sistemas Operativos - Curso 2008/2009

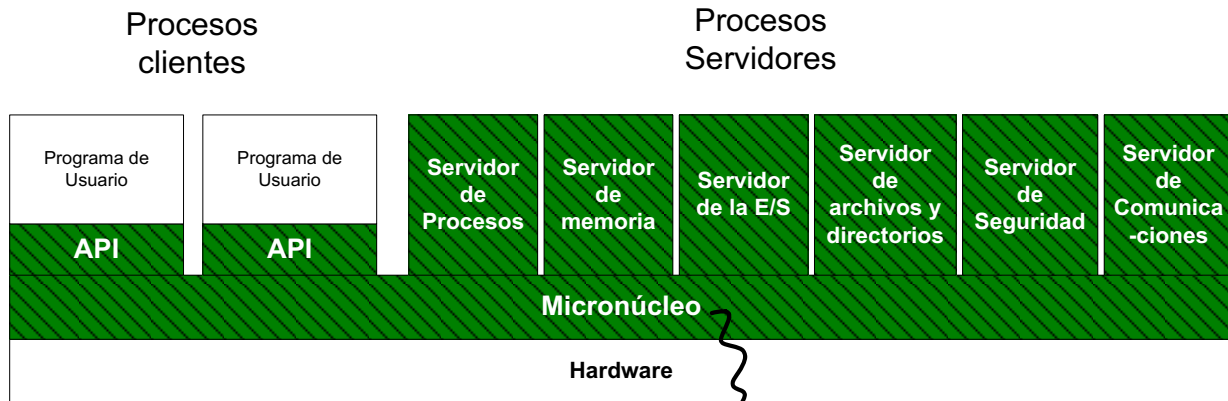
Sistemas Operativos estructurados: cliente/servidor

22

- 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

23



Estructura cliente-servidor en un sistema operativo

kernel reducido

Clasificación de Sistemas Operativos

- **Número de procesos simultáneos:**
 - **Monotarea.** *~ Monitorea solo 1 proceso a la vez*
 - **Multitarea.** *~ Varias procesos a la vez, necesita planificación.*
- **Modo de interacción:**
 - **Interactivo.**
 - **Por lotes (batch).** *~ tu pides y ya se hará en algún momento*
- **Número de usuarios simultáneos:**
 - **Monousuario.**
 - **Multiusuario.**
- **Número de procesadores:**
 - **Monoprocesador.**
 - **Multiprocesador.**
- **Número de hilos (threads):**
 - **Monothread.**
 - **Multithread.**
- **Tipo de uso:**
 - **Cliente.** *~ pide cosas*
 - **Servidor.** *~ proporciona cosas*
 - **Empotrado.**
 - **Tiempo real.** *~ se usan otros sistemas operativos.*

Contenido

25

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.**

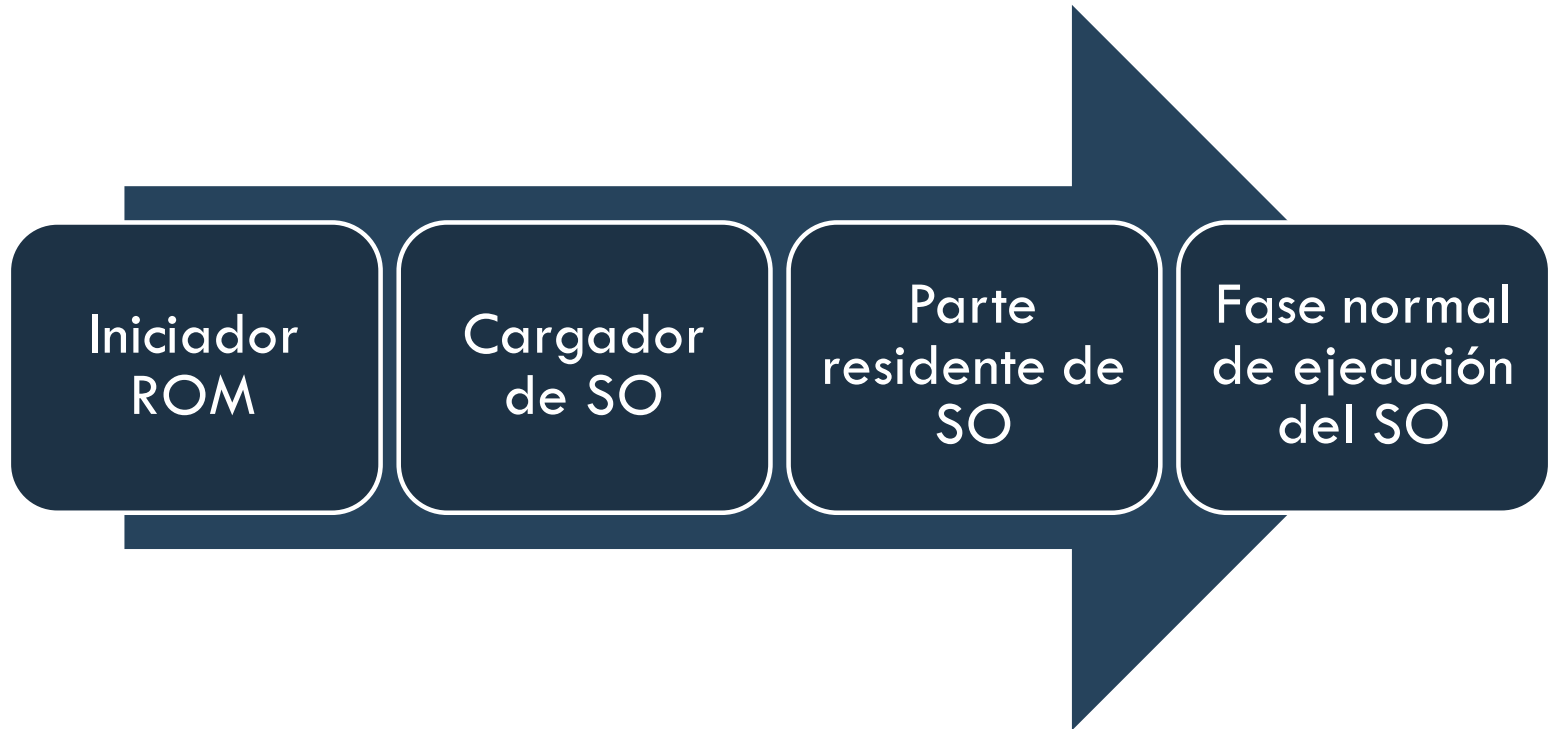
Arranque del sistema operativo

Al pulsar el botón se lleva a cabo la interrupción o (reset) que pone en PC la div. 0 que es el booter que carga la BIOS ^{-basic} que comprueba los dispositivos y la memoria, mira el disco ^{input} que contiene el SO y lo va cargando en memoria y cuando acaba lo empieza a ejecutar, para instalarlo.

- El sistema operativo se inicia al encender el computador. Cuando termina libera esa memoria.
Ya está arrancado.
- 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

27



Iniciador ROM

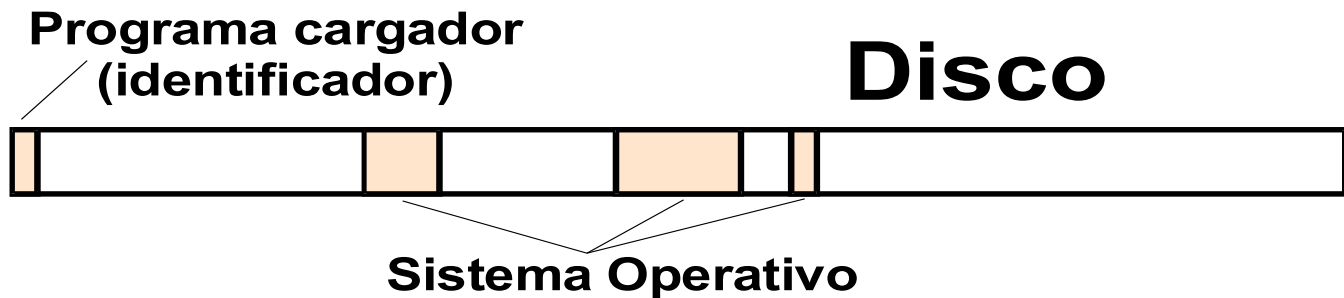
28

- 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

29

- 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.



Parte residente del SO

30

- 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

31

- Proceso *init.* ~ proceso de inicio
 - ▣ 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.*

Parada del computador

↳ Proceso inverso al arranque

32

- Para acelerar la ejecución el sistema operativo mantiene información en memoria no actualizada a disco.
Shut down
- Al apagar hay que volcar dicha información a disco y terminar la ejecución de todos los procesos.
- Si no se hace volcado (apagado brusco) } Hay una pequeña batería que permite que se lleve a cabo el volcado
 ■ Pérdida de información.
 ■ Sistema de ficheros en estado inconsistente. } Si el sistema es grande ⇒ SAI
- 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

33

- 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

34

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

Alternativas de generación

35

- 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.

¿Objetivo Cumplido?

36

- ❑ 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.

SISTEMAS OPERATIVOS: INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS

Introducción y conceptos básicos