

Introducción a la Informática

Ingeniería en Informática

Departamentos de Automática y Ciencias de la Computación

Tema 6

Sistemas Operativos



Introducción

- ¿Qué es un sistema operativo?
- Ubicación de un sistema operativo en un computador
- Descripción de un sistema operativo:
 - Funcional
 - Estructural
 - Realización
- Funciones de los sistemas operativos a través de la evolución histórica

Tema 6

© Sebastián Sánchez y Óscar García

2

Introducción

- Tipos de sistemas operativos
 - Sobremesa
 - Servidores
 - Empotrados
 - Distribuidos
 - Sistemas de Tiempo Real
- Modelos de desarrollo de software
- Windows2K vs Linux
- Los sistemas operativos en la vida real
 - Usuarios
 - Administradores
 - Desarrolladores

Tema 6

© Sebastián Sánchez y Óscar García

3

¿Qué es un SO?

- Un SO es un conjunto de programas que por medio de abstracciones ponen el hardware del ordenador, de modo seguro, a disposición del usuario
- H. Deitel: un SO es un programa que actúa como interfaz entre el usuario de un ordenador y el HW del mismo, ofreciendo el entorno necesario para que el usuario pueda ejecutar programas
- Katzan: conjunto de programas y datos que ayudan a crear otros programas y a controlar su ejecución
- Madnik y Donovan: conjunto de programas que gestionan los recursos del sistema, optimizan su uso y resuelven conflictos

Tema 6

© Sebastián Sánchez y Óscar García

4

¿Por qué son interesantes los SO?

- Combinan aspectos referentes a distintas áreas relacionadas con los computadores: lenguajes, hardware, estructuras de datos, algoritmos, etc.
- Actúan como coordinadores entre los usuarios y los programas resolviendo posibles conflictos
- Hacen que los computadores parezcan ser más de lo que son en realidad
- Son sistemas software grandes y complejos, por lo tanto es necesario abordarlos de una forma estructurada
- Las ideas utilizadas son aplicables a otras parcelas del software

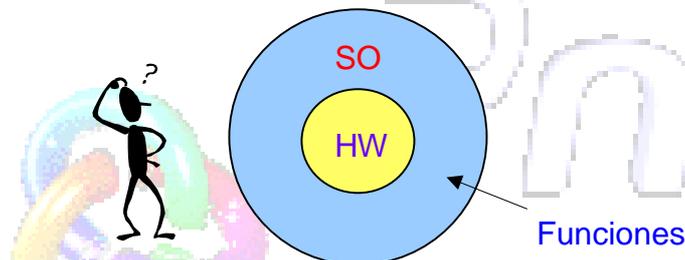
Tema 6

© Sebastián Sánchez y Óscar García

5

Objetivo básico de un SO

- “Crear una máquina virtual para la que sea sencillo trabajar”



- Dependiendo de las funciones incluidas y de la relación entre ellas, tendremos diferentes tipos de Sistemas Operativos

Tema 6

© Sebastián Sánchez y Óscar García

6

Funciones del SO

- Un SO es un administrador de recursos y una interfaz con los programas, los usuarios y el HW. Sus funciones son:
 - Compartir el hardware entre usuarios
 - Facilitar la entrada salida
 - Planificar recursos entre usuarios
 - Definir la "interfaz de usuario"
 - Permitir a los usuarios compartir los datos
 - Recuperarse de los errores
- Los recursos claves que un SO administra son:
 - los procesadores y los dispositivos de E/S
 - el almacenamiento y los datos

Tema 6

© Sebastián Sánchez y Óscar García

7

Visiones de un SO

- Visión amplia
 - Disponer de un entorno para crear y mantener programas
 - Disponer de una interfaz sofisticada para las operaciones del programador
 - Gestionar los recursos del sistema
- Visión restringida
 - Básicamente el núcleo del sistema con las funciones y estructuras de datos necesarios para gestionar los recursos
 - CPU, memoria, dispositivos de E/S, procesos ...

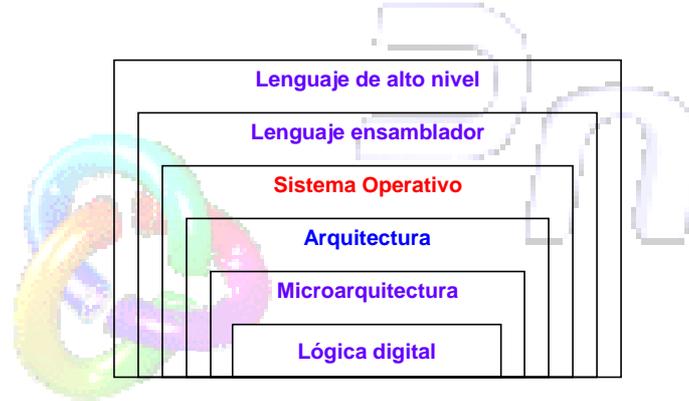
Tema 6

© Sebastián Sánchez y Óscar García

8

¿Qué lugar ocupa?

- Estructura clásica de niveles de un computador



Tema 6

© Sebastián Sánchez y Óscar García

9

Modelos de cada nivel

- Modelo funcional o de caja negra: describe la función del sistema, sus acciones sobre el entorno y sus reacciones a éste
- Modelo estructural: describe la organización del sistema como conjunto de partes relacionadas
- A un mismo modelo funcional pueden corresponder distintos modelos estructurales
- Los modelos se emplean con el fin de aplicarlos en la práctica y resolver un determinado problema
- La realización práctica de cada modelo puede llevarse a cabo de modos muy diferentes

Tema 6

© Sebastián Sánchez y Óscar García

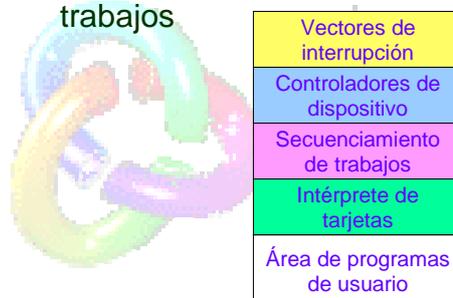
10

Evolución histórica

- En resumen:
- 1ª etapa:
 - Ordenadores caros y operarios baratos
- 2ª etapa:
 - Ordenadores baratos y operarios caros
- Veamos la cosa con más detalle

Al principio

- Monitor simple residente
 - Objetivo: Reducir el tiempo de lanzamiento de los trabajos
 - Método: Secuenciamiento automático de los trabajos



Sistemas batch (*off-line*)

- Objetivo: Reducir los tiempos de espera de E/S
 - Método: Utilizar dispositivos de E/S más rápidos (cintas o discos) y dejar que las lectoras y las impresoras sean manejadas más tarde por otras CPU's (dispositivos lógicos)
- Ejemplo:
 - IBM 1401 (ordenador "barato") se encarga de manipular la lectora de tarjetas y la impresora
 - IBM 7094 (ordenador caro) se encarga del cálculo. Lee y deposita los resultados en cintas

Tema 6

© Sebastián Sánchez y Óscar García

13

Arquitectura de entrada-salida

- Objetivo: solapar el funcionamiento de la CPU y la entrada-salida una sola máquina
- Método:
 - Interrupciones
 - Acceso directo a memoria (DMA)
- SPOOLING (Simultaneous Peripheral Operation On-line)
 - El disco es empleado como un buffer
 - Consecuencia: el SO puede escoger del conjunto de trabajos el trabajo a realizar desde el disco

Tema 6

© Sebastián Sánchez y Óscar García

14

Multiprogramación

- Objetivo: aprovechar los tiempos de espera de un trabajo en la CPU para ejecutar instrucciones de otro trabajo
- Método: mantener los trabajos simultáneamente en memoria y elegir el trabajo a conmutar
- Consecuencias:
 - Gestión de memoria
 - Planificación de dispositivos
 - Gestión de abrazos mortales
 - Control de la concurrencia
 - Protección

Tema 6

Tiempo compartido

- Objetivo: permitir la interacción entre el usuario y el trabajo que se está ejecutando. Los sistemas *batch* carecen de esta interactividad
- Método: utilizar las técnicas de multiprogramación y planificación de la CPU para proporcionar a cada usuario una pequeña proporción del tiempo de CPU
- Consecuencias:
 - Protección
 - Gestión de disco
 - Mecanismo para la ejecución concurrente

Tema 6

Sistemas de tiempo real

- Son muy utilizados en sistemas empujados: UCIs, automóviles, teléfonos móviles, sistemas de control, etc.
- El sistema operativo debe responder siempre dentro de los márgenes requeridos por el sistema controlado, si nos salimos de esos márgenes, tendremos que buscar otras soluciones
- Existen actualmente lenguajes de programación (Ada) que nos dan grandes posibilidades para controlar procesos en tiempo real

Tema 6

© Sebastián Sánchez y Óscar García

17

Sistemas distribuidos

- Funciona como uno centralizado, pero se ejecuta en múltiples procesadores independientes
- Es distribuido si está formado por varios elementos que cooperan para dar un servicio único
- Un SOD no es un sistema simplemente replicado. Hay un particionado, una cooperación, ningún componente sobrevive solo, hay mentalidad de grupo
- El SW el determina si el sistema es o no distribuido.
- Si puedes decir qué máquinas estás usando, dónde corren los programas, dónde están los archivos, entonces NO es SOD

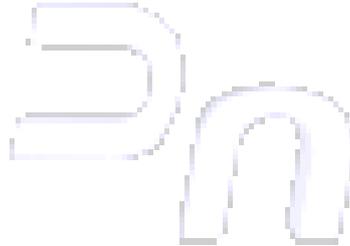
Tema 6

© Sebastián Sánchez y Óscar García

18

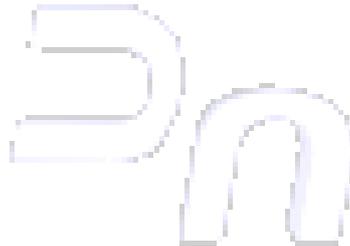
Modelos de desarrollo de SW

- Software de libre distribución
 - Shareware
 - Freeware
 - Open Source
 - Licencia GNU
- Software propietario
 - Microsoft
 - HP
 - Compaq
 - SGI



Windows NT/2K vs. Linux

- Filosofía de desarrollo de ambos sistemas
- Comparativa
 - Servicio técnico
 - Precio
 - Fiabilidad
 - Seguridad
 - Rendimiento
 - Portabilidad
 - Participación



SO's en la vida real

■ Tipos de usuarios

● "Ordinarios"

- ✓ Facilidad de uso
- ✓ Interfaces gráficas: Gnome, MacOS, Windows, Neutrino, etc.

● Administradores

- ✓ Hacer la vida fácil a los demás (o imposible...)

● Desarrolladores

- ✓ De aplicaciones
- ✓ De sistemas
- ✓ De interfaces
- ✓ etc.