

T2-SISTEMAS OPERATIVOS

Un **sistema operativo** es un programa o conjunto de programas que actúa como intermediario entre el usuario y el hardware del ordenador, gestionando los recursos del sistema y optimizando.

FUNCIONES

-Debe gestionar HW y proporcionar una interfaz:

1. Gestión del procesador: Control de la ejecución de los programas. Control de concurrencia: Prioridades entre procesos que solicitan el mismo recurso.

2.Gestión de E/S: Administración de periféricos. Proporciona el interfaz entre usuario y dispositivo.

3.Gestión de la memoria: Reserva y liberación de la memoria.

4.Gestión de archivos y directorios.

5.Gestión de permisos y de usuarios.

6.Control de errores y pérdida de datos.

7.Control de seguridad.

8.Interfaz de usuario /GUI.

COMPONENTES

1.El núcleo o Kernel: interactúa directamente con el HW y gestiona los recursos como procesador o memoria.

2.Servicios: Interfaces de comunicación entre las aplicaciones y el componente HW que necesitamos usar. Ej: Impresión de fichero. **Tipos:**

1.Gestor de memoria

2.Gestor de procesos: Encargado de la gestión del procesador. Procesos simultáneamente precisan del microprocesador.

3.Gestor de E/S: Periféricos

4.Gestor de archivos y directorios: Gestión de almacenamiento secundario.

5.Comunicación y sincronización entre procesos

6.Seguridad: Qué puede hacer cada usuario, indicando al núcleo qué recursos puede usar.

-Planificador/administrador de memoria: es el encargado de mantener el control y la forma en la que se asignan espacios en las diferentes memorias.

-Scheduler/planificador de la CPU: Su función consiste en repartir el tiempo disponible de un microprocesador entre todos los procesos que están disponibles para su ejecución.

-Servicios: es un conjunto de actividades que buscan responder las necesidades de un cliente.

3.Shell o interprete de comandos (ordenes):

- Proporciona al usuario por donde “hablar” con el PC.
- El usuario dispone de una serie de comandos que ejecutará y Shell interpretará para realizar la operación correcta.
- Shell varía según el SO y las ordenes.

Clasificación del SO (SW complejo y grande):

1.Por estructura:

1.SO Monolíticos: un único programa donde se integran todos los componentes. V: Están hechos a medida. D: Dificultad para agregar funcionalidad o modificar la existente. Poca flexibilidad. Ej: MS-DOS, Unix.

2.SO Estructura concreta: Cada capa recoge una función concreta y ofrece servicios a la que está justo por encima. V: Cada parte menos compleja.

3.SO Cliente –Servidor: Minimizar el núcleo (Fun. Básicas) trasladando el código de todos sus servicios a las capas superiores. El núcleo solo deberá controlar la comunicación entre cliente y servidor. V: Modular y propósito general. Ej: Windows Xp, vista, 7...

2.Evolucion histórica (está asociada a la evolución HW):

Primera Etapa:

Proceso en serie

- o No existía SO.
- o El programador desarrollaba en lenguaje máquina (Tarjetas perforadas)
- o Trabajos ejecutados en serie.
- o Reserva de “hora” por usuario.

Segunda Etapa:

Sistemas operativos por lotes

- o Trabajos ejecutados en serie.
- o Programador con tareas similares.
- o Diferencia: Las tareas se agrupaban por tipo, formando lotes (*batch*).
- o El programador dejaba el conjunto de tarjetas perforadas (lote de programas) sobre la bandeja de entrada y el SO se encargaba de ir procesando la salida de forma secuencial.

Tercera Etapa:

1. Sistemas operativos por lotes con multiprogramación/multitarea:
 - Varios programas simultáneamente en la memoria principal.
 - Ejecución de varios programas a la vez.
 - Clave en los SO modernos.
2. Sistemas operativos de tiempo compartido:
 - El SO comparte sus recursos con todos los usuarios del sistema, y reparte el tiempo de uso de los recursos entre ellos.
 - Transparente para el usuario.
 - También común en arquitecturas actuales.
3. Sistemas operativos en tiempo real:
 - Se construyen para tareas muy específicas y en sistemas que necesitan procesar multitud de eventos.
 - Objetivos: Rápidos tiempos de respuesta.
 - Ej.: Control aéreo, trenes, etc.
4. Sistemas operativos de propósito general o multimodo:
 - Capaces de operar por lotes, multiprogramación tiempo real o tiempo compartido.
 - Muy complejos.

Cuarta etapa:

Intentan centrarse más en el fácil uso de los SO a costa del rendimiento.

- Sistemas operativos orientados a usuarios finales
 - Sin comandos complejos.
 - Ej.: Windows 8 o últimas distribuciones de Linux.
- Sistemas operativos distribuidos:
 - Definición: Conjunto de ordenadores independientes que aparecen ante los usuarios del sistema como una única máquina.
 - Conectados en red.
 - Ej.: Red de sucursales de un banco.

(Lo contrario sería SO centralizados.)

SO según los servicios que ofrece:

SOFTWARE MONOUSUARIO. Un sistema operativo monousuario (de mono: 'uno'; y usuario) es un sistema operativo que sólo puede ser ocupado por un único usuario en un determinado tiempo.

SOFTWARE MULTIUSUARIO. Multiusuario: de multi: varios; y usuarios, "apto para ser utilizado por muchos usuarios".

SOFTWARE MONOTAREA. Son aquellos que solo permiten una tarea a la vez por el usuario. Puede darse el caso de un sistema multiusuario y monotarea, en el cual se admiten varios usuarios, pero cada uno de ellos puede estar haciendo una tarea.

MULTITAREA. es aquel sistema que permite al usuario realizar varios trabajos a la vez.

MONOPROCESO: son los que únicamente permiten realizar un proceso a la vez.

MULTIPROCESO: son los que permiten realizar varios procesos al mismo tiempo.

SO Propietarios:

- La propiedad intelectual pertenece a una empresa.
- Licencias de uso.
- Sin acceso al código fuente, o sin derecho a modificarlo ni distribuirlo.
- Ej. Windows.

SO libres:

Las cuatro libertades para que un software sea libre:

1. **Libertad 0 – De ejecución:** La libertad de usar el programa, con cualquier propósito.
2. **Libertad 1 – De inspección:** La libertad de estudiar el funcionamiento del programa, y adaptarlo a las necesidades.
3. **Libertad 2 - De redistribución:** La libertad de distribuir copias, con lo que puede ayudar a otros.
4. **Libertad 3 – Para mejorar el programa:** La libertad de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie.

+ 1 y 3 requieren acceso al código fuente → SW de código abierto.

+ Puede ser gratis o no → distinguir de software gratuito.

Tipos de Aplicaciones

Gratuitas (freeware) o comerciales: Importante tenerlo en consideración en cualquier proyecto, de cara a la planificación (coste) del mismo y al mantenimiento posterior.

Libres o propietarias:

1. *Licencias de SW libre:* distribución del código fuente+ el programa+4 libertades anteriores.
2. *Licencias de SW propietario:* Los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo (con o sin modificaciones).

Aplicaciones *opensource* o privativas: referidas a la posibilidad de acceso al código fuente o no. Repercusión en cualquier implantación.

En función del tipo de licencia de distribución, el SW comercial puede ser:

OEM (Original equipment manufacturer): Habitualmente, se vende un SW supeditado a la condición de que también se compre un equipo nuevo. Alianzas entre empresas de SW y HW.

- Suelen añadir más restricciones al SW, como:
 - # de reinstalaciones.
 - No vender ni ceder a terceros, salvo en las mismas condiciones en las que se compraron (como parte de un equipo).

Retail: Programa totalmente del usuario, pudiendo cederlo o venderlo.

Licencias por volumen:

- Destinado a empresas.
- Condiciona el # de equipos que pueden utilizar el SW bajo la misma licencia.
- Se suelen vender por paquetes de X número de licencias. No se puede ceder a terceros.

En caso de tener varios SO instalados en un equipo, permite seleccionar el SO que se desea iniciar.

NTLDR (*NT Loader*): Windows NT, Windows 2000, Windows XP y Windows Server 2003.

Administrador de arranque de Windows (*Bootmgr*): Windows Vista, Windows 7, etc.

Lilo (*Linux Loader*) y su evolución Grub.