

Introducción

Podemos definir a un sistema operativo como el **programa que controla la ejecución** de las aplicaciones permitiendo un **acceso eficiente a recursos compartidos limitados**.

El sistema operativo **actúa como una interfaz** entre las aplicaciones y el hardware, esto simplifica la labor del programador. La interfaz es un conjunto de comandos y/o métodos que permiten la intercomunicación del programa con otros programas o elementos.

Las características principales de un sistema operativo son:

- **Comodidad** – abstracción.
- **Eficiencia** – concurrencia.
- **Capacidad de evolución** – modularidad.

Las tareas principales de un sistema operativo son:

1. La creación de programas mediante editores, compiladores...
2. Acceso ordenado a los dispositivos de E/S.
3. Acceso controlado a los archivos.
4. Acceso al sistema.
5. Detección y respuesta de errores, estos errores pueden ser en el hardware, en el software o que el sistema operativo no es capaz de satisfacer la solicitud de una aplicación.
6. Contabilidad, recoger estadísticas, supervisar rendimiento...

El sistema operativo funciona de manera parecida a un software normal, es un programa ejecutado por el procesador por lo que el SO abandona el control del procesador para ejecutar otros programas. El SO no esta completamente en la memoria principal, solo las funciones mas usadas, a esto se le denomina **núcleo**, *kernel*, o *microkernel*.

Existen varios tipos de estructuras en los SO:

- **Sistemas monolíticos**: Su estructura es la más común, el SO es un conjunto de procedimientos que pueden llamarse mutuamente. No hay modo usuario/núcleo ya que el hardware no lo permite. El problema de esta estructura es que no hay ocultación de datos y es difícil de modificar y depurar.
- **Sistemas por capas**: El SO está organizada por una jerarquía de capas, cada una cimentada en la que está por debajo. El problema de este diseño es que baja la eficacia.
- **Maquinas virtuales**: Esta arquitectura es una extensión del sistema por capas, donde cada proceso tiene la ilusión de estar ejecutándose en un sistema mono programado. La maquina virtual proporciona copias exactas del hardware subyacente.

Ejemplos de maquinas virtuales son *emuladores*, *VMWare*, *VirtualBox*...

- **Ventajas**: El SO virtual no tiene problemas de seguridad y protección de recursos compartidos. Coexistencia de distintos sistemas operativos en un mismo HW.
- **Desventajas**: Perdida de eficiencia en las “traducciones”.
- **Cliente-Servidor**: Dentro del núcleo hay procesos que manejan recursos (servidores) y procesos que clientes que piden estos recursos, estos procesos se comunican mediante mensajes.
 - **Ventajas**: Se mantiene fácilmente debido a la modularización de los servicios, y es robusto. Es una extensión natural del modelo a sistemas distribuidos.

En un procesador existen registros que guardan información y son accesibles de manera muy rápida. Si los registros son **visibles al usuario**, permiten al programador minimizar referencias a memoria principal (ejemplos: *Registro de índice*, *puntero de segmento*, *puntero de pila*...), si son **control y estado**, son usados únicamente por el procesador para controlar operaciones y por rutinas del SO para controlar la ejecución de los programas (ejemplos: *PC - program counter*, *IR – Instruction register*, *PSW – Program state word*).

Otro elemento importante de los SO son las **interrupciones**. Son causadas cuando un evento externo al proceso ocurre, estas mejoran la eficiencia del procesamiento, permite al procesador ejecutar otras instrucciones durante E/S.

Existen varios tipos:

- **De programa**: División por cero, intento de ejecutar instrucción ilegal, referencia a memoria fuera del espacio asignado...
- **De reloj**: Para funciones periódicas.
- **De E/S**: Para indicar normalidad/error.
- **Fallo de HW**: Cortes de energía, errores de paridad de memoria...