

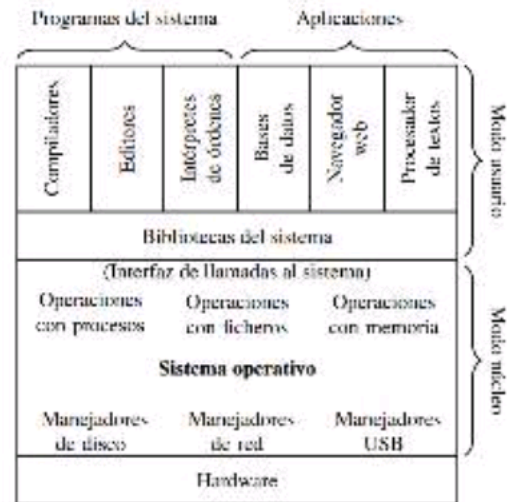
# TEMA I: INTRODUCCIÓN

## CONCEPTO DE SISTEMA OPERATIVO

El **sistema operativo como una máquina extendida o virtual** es un programa que **oculta el funcionamiento del hardware** al programador (manejo de interrupciones, relojes, control de memoria y otras características de bajo nivel) y presenta una **interfaz sencilla**. Además, añade el almacenamiento y organización de la información por medio de un sistema de ficheros, la capacidad de **ejecutar varios programas a la misma vez**, etc...

El **sistema operativo como un controlador de recursos** es un programa que controla todas las piezas de un sistema complejo y proporciona una **asignación ordenada y controlada de los procesadores, memoria, dispositivos E/S**, etc..., entre los distintos programas que compiten por ellos.

Sistemas operativos hay muchos, cada uno con sus ventajas e inconvenientes en función de las tareas a realizar, entorno de ejecución, precio, tiempo, etc... Una idea que debe quedar clara es que las características de un sistema operativo dependen muchas veces de las propiedades del hardware sobre el que se ejecuta.



## PROTECCIÓN DEL SISTEMA OPERATIVO

La **protección** del S.O. solo se consigue con la ayuda del hardware y de tres mecanismos básicos:

- **Modos de ejecución del procesador**: el procesador en **modo núcleo** es capaz de **ejecutar cualquier instrucción** disponible en el procesador. En cambio, el procesador en **modo usuario** **no puede ejecutar ciertas instrucciones** de la CPU; si lo hace, provocará una **excepción** que hará que el programa termine su ejecución.
- **Protección de la memoria**: el hardware debe tener algún **mecanismo** que impida que un problema acceda a zonas de memoria RAM que no le pertenecen. De esta forma se evita que un programa pueda leer o modificar el código, los datos, la pila o el sistema operativo.
- **Interrupciones periódicas**: para que un programa no se haga con el control de la CPU, es **necesario** que el hardware disponga de algún reloj que produzca **interrupciones periódicas** que permitan al sistema operativo hacerse con el control de la máquina.

Un aspecto importante es que la protección solo funcionará si los tres mecanismos mencionados existen y se usan de forma conjunta.

## HISTORIA Y EVOLUCIÓN

### PRIMERA GENERACIÓN (1945-1955): VÁLVULAS Y CONEXIONES

Las grandes máquinas que ocupaban habitaciones enteras estaban construidas mediante miles de válvulas de vacío y cables conectados a mano. Se desconocían los lenguajes de programación y, por supuesto, los sistemas operativos. La programación se realizaba totalmente en lenguaje máquina y con frecuencia se utilizaban conexiones para controlar las funciones básicas del sistema.

### SEGUNDA GENERACIÓN (1955-1965): TRANSISTORES Y SISTEMAS DE PROCESAMIENTO POR LOTES

Lectores de tarjetas, impresoras de líneas, cintas magnéticas, ensambladores, cargadores y enlazadores facilitaron la programación. La introducción del transistor fue fundamental, ya que hizo que los ordenadores se volvieran fiables.

Surgieron las primeras bibliotecas de funciones de E/S, que facilitaban la labor al programador ya que se seguía permitiendo la interacción del programador con la máquina; pero, si el programador no tenía experiencia podía producir importantes pérdidas de tiempo. Para evitar el problema contratan a un operador profesional.

Otra medida fue planificar los trabajos para evitar las pérdidas de tiempo por cambios de cinta. Pero cuando se detenía un trabajo el operador tenía que percatarse de la situación observando la consola. Durante la transición de cargar el lector de tarjetas o de cinta para el siguiente trabajo, la CPU permanecía inactiva, algo inaceptable. Para superar esa inactividad se desarrolló el sistema de procesamiento por lotes, lo que dio lugar a la creación de los primeros sistemas operativos rudimentarios (como el monitor residente).

El monitor residente ocupaba de forma permanente una parte de la memoria principal, dejando el resto libre para la ejecución de programas. Este sistema operativo se componía de un intérprete de tarjetas de control, un cargador de programas y aplicaciones y manejadores de dispositivos de E/S.

Los dispositivos de E/S eran lentos en comparación con la velocidad del procesamiento de la CPU. Una solución fue pasar de operaciones en línea (on-line), donde la CPU interactuaba directamente con los dispositivos de E/S lentos, a operaciones fuera de línea (off-line), donde la CPU interactúa con dispositivos de E/S. En esta última podemos distinguir tres etapas:

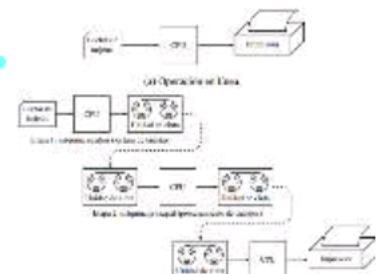
**1ª etapa:** los datos se leen de las tarjetas y se guardan en una cinta (esta se debe llenar).

**2ª etapa:** los datos de la cinta se procesan y los resultados se guardan en otra cinta.

**3ª etapa:** los datos de la cinta de salida se imprimen.

La operación fuera de línea dio lugar a la idea de independencia de dispositivos de E/S. El acceso a los dispositivos lógicos siempre se hace de la misma manera, independientemente del dispositivo físico que haya por debajo. Se consiguió que la E/S y la CPU funcionasen en paralelo. El coste del paralelismo era bastante alto, y la solución la dieron los buffers y los spoolers.

Un buffer es una zona de memoria principal de un cierto tamaño que se utiliza para el almacenamiento temporal de datos. La CPU no puede intervenir en la E/S del dispositivo; como mucho la CPU puede transferir los datos, aunque lo ideal es que no intervenga. El mecanismo de spooler permite la superposición de la E/S de unos programas con el cómputo de otro. Con esto conseguimos que la CPU y los dispositivos de E/S estén más ocupados. Además, la posibilidad de tener varios trabajos distintos en disco hace que sea posible cambiar el orden del procesamiento de estos.



La E/S progresa simultáneamente con su cómputo.

### TERCERA GENERACIÓN (1965-1980): CIRCUITOS INTEGRADOS Y MULTIPROGRAMACIÓN

La aparición de los circuitos integrados hace que los ordenadores se vuelvan más pequeños y fiables. La multiprogramación consiste en tener varios programas en memoria. Cuando el programa que se usa la CPU no pueda continuar su ejecución porque tiene que esperar, se pasara a otro programa que esté listo para ejecutarse.

El tiempo compartido o multitarea es una variante de la multiprogramación. Consiste en realizar un cambio rápido entre tareas, lo que permite que cada usuario pueda interactuar con el programa que está ejecutando y tener la sensación que es el único usuario de la máquina.

### CUARTA GENERACIÓN (1980-1995): ORDENADORES PERSONALES

La mejora de los circuitos integrados los vuelve más pequeños, dando lugar a la aparición de circuitos LSI y VLSI. Ahora en un chip relativamente pequeño, es posible integrar millones de transistores y otros componentes; con un menor precio, haciéndolos más accesibles. Por lo que, aparece el ordenador personal o PC.

También aparecieron las estaciones de trabajo, que son ordenadores muy potentes conectados por medio de una red Ethernet o similar. Los sistemas operativos que dominan son MS-DOS y Unix; pero a mediados de los 80 empiezan a proliferar la red de ordenadores personales, lo que da lugar a la aparición de nuevos tipos de sistemas operativos.

### QUINTA GENERACIÓN (1995-ACTUALIDAD): INTERNET Y DISPOSITIVOS MÓVILES

Si algo caracteriza a esta generación es la explosión del uso de Internet y la proliferación de los teléfonos móviles. En lo que a sistemas operativos se refiere, aparecen Windows 2000 (destinado a los servidores empresariales) y Windows XP (destinado a usuarios). Otros sistemas operativos que alcanzan gran popularidad en esta generación son Linux, Apple Mac OS X, Android e iOS.

## TIPOS DE SISTEMAS OPERATIVOS

### SISTEMAS OPERATIVOS DE PROPÓSITO GENERAL (flexibilidad y adaptación)

Los sistemas operativos diseñados para realizar una gran variedad de tareas computacionales en entornos de diferente naturaleza. Estos sistemas se caracterizan por su flexibilidad y su capacidad para adaptarse tanto al hardware como a los trabajos, Ejemplos: Unix, Microsoft Windows y Apple macOS.

En los supercomputadores, donde el sistema operativo más usado es Unix, son sistemas formados por un gran número de elementos conectados entre si por mediante una red, que son capaces de procesar grandes cantidades de datos.

Unix también es el sistema operativo más común en los mainframes, estos sistemas son capaces de procesar a la vez muchos trabajos que requieren enormes cantidades de E/S,

Cuando pasamos a sistemas de cómputo menos potentes como los servidores podemos encontrar tanto Unix, como Windows o macOS; al igual que en los ordenadores personales. Y finalmente encontramos versiones especiales de estas tres adaptados a los teléfonos inteligentes, tabletas y diferentes sistemas integrados.



## SISTEMAS OPERATIVOS DE RED

Unix, Windows y macOS son también sistemas operativos de red, aunque, rara vez se denominan así. En los sistemas operativos de red, los usuarios son conscientes de la existencia de varios ordenadores conectados mediante una red. Cada máquina es independiente de las demás, ya que ejecuta su propio sistema operativo local y tiene sus propios usuarios. Sin embargo, gracias a la red, puede interactuar con el resto de las máquinas siempre que se hayan concedido las autorizaciones oportunas entre las distintas máquinas que participan.

## SISTEMAS OPERATIVOS DISTRIBUIDOS

Son un conjunto de computadores conectados entre sí mediante una red, que son vistos por los usuarios como un sistema tradicional, con un único computador y sistema operativo (*imagen única del sistema*). Estos sistemas son capaces de manejar varios procesadores y en ellos el usuario no es consciente del lugar donde se ejecutan sus programas ni dónde se encuentran sus ficheros (*transparencia de localización*). Ejemplos: Amoeba, Plan 9 e Inferno

Se puede utilizar una impresora o cualquier otro recurso conectado en otra máquina (compartición de recursos); al disponer de varios procesadores se puede hacer que un trabajo use varios y así terminar antes; y confiable ya que, si una máquina falla, el resto seguirá realizando todo el trabajo.

## SISTEMAS OPERATIVOS DE TIEMPO REAL

Ejemplos: VxWorks y QNX. El parámetro clave son las restricciones temporales, las podemos encontrar de dos tipos: **rigurosos y no rigurosos**. En los **primeros**, una acción se debe realizar necesariamente en cierto momento o intervalo, sino se puede poner el sistema en peligro. En los **segundos**, es aceptable no cumplir de vez en cuando un plazo, siempre y cuando el comportamiento general del sistema se ajuste a unos ciertos parámetros.

## SISTEMAS OPERATIVOS PARA TARJETAS INTELIGENTES

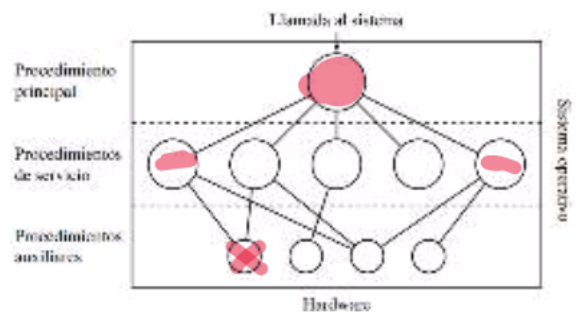
Por las propias características de estas tarjetas, que tienen grandes limitaciones de potencia y memoria, estos sistemas operativos solo disponen de una máquina virtual de Java (JVM) que se encarga de ejecutar los applets que se cargan en la tarjeta.



## ESTRUCTURA DE LOS SISTEMAS OPERATIVOS

### SISTEMAS MONOLÍTICOS

Este tipo de organización es la **más común** (Unix y Windows). Se caracteriza por estar formada por una **colección de procedimientos** que **se llaman unos a otro**, tener cada procedimiento definido en una **interfaz muy clara** en cuanto a lo que hace el procedimiento, los parámetros que recibe y los resultados que produce; además no existe la ocultación, ya que todo procedimiento es visible a los demás. Y tampoco tiene una estructura bien definida de los procedimientos existentes, aunque algo de estructura tiene que haber. Su estructura es:

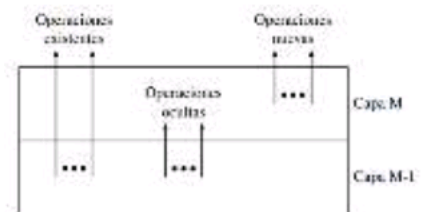


- **Procedimiento principal:** recibe las peticiones de llamadas al sistema y tras analizarlas llama a los procedimientos de servicio correspondientes.
- **Procedimientos de servicio:** llevan a cabo las llamadas al sistema.
- **Procedimientos auxiliares:** ayuda a los procedimientos de servicio a realizar su trabajo.

### SISTEMAS CON CAPAS

Los avances han permitido realizar un **diseño más modular**. En un **diseño modular por capas** el sistema operativo se divide en capas, **cada una construida sobre la anterior**. La capa más baja (0) corresponde al hardware y la capa más alta es la interfaz con el usuario.

**Cada capa solo utiliza las funciones y servicios de la capa inmediatamente inferior**, una vez se ha implementado, verificado y depurado una capa se asciende a la siguiente, haciéndolo mucho más fácil. **Y cada capa oculta los detalles de bajo nivel a las capas superiores**. Esta permite modificar una capa en cualquier momento sin tener que hacer cambios en otras, excepto que se altere la interfaz de la capa.



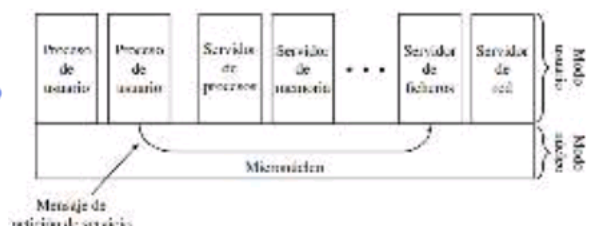
La mayor dificultad de este enfoque es el propio diseño de capas ya que solo puede utilizar los servicios de las capas inferiores; **esto hace que surjan problemas de dependencia entre capas** y que sea difícil saber dónde colocar una determinada función.

### MODELO CLIENTE-SERVIDOR

**Explotan la idea de mover parte del sistema operativo a capas superiores y mantener un núcleo mínimo (micronúcleo o microkernel, controlan las comunicaciones cliente-servidor)**. Lo usual es implantar la mayoría de las **funciones del sistema operativo en los procesos usuario**, por lo que se distinguen dos tipos de procesos:

- **Cliente:** solicitan servicios.
- **Servicios:** realizan los trabajos solicitados por los procesos clientes y les devuelven las respuestas.

Los **servidores se ejecutan en modo usuario**, por lo que **no hay acceso directo al hardware**. Otra ventaja del modelo cliente-servidor es su **capacidad de adaptación a los sistemas distribuidos**. Un cliente se comunica con el servidor por medio de mensajes (se envía una solicitud y se recibe una respuesta).



Ya que los servidores se ejecutan en modo usuario, hay funciones que no pueden realizar directamente ellos mismos. Para solucionar este problema, existen dos opciones:

- **Incluir el servidor en el núcleo**, esto hace que para el servidor incluido desaparezca la ventaja de protección frente a los fallos en el propio servidor.
- **Existencia de mensajes especiales** enviados por los servidores que captura el micronúcleo para procesarlos el mismo. En este caso, el micronúcleo también verificará si el servidor que le envía una solicitud tiene permiso para hacerlo (la más preferible).