

Tema 1. Estructuras de sistemas operativos

Introducción

Definición de sistema operativo

Un sistema operativo es un **programa** que controla la ejecución de aplicaciones y programas, que actúa como interfaz entre las aplicaciones y el hardware del computador. Un sistema operativo tiene tres **objetivos**:

1. Facilidad de uso: Se realiza mediante el uso de servicios.
2. Eficiencia: Permite que los recursos se utilicen de forma eficiente.
3. Capacidad para evolucionar: Se debe construir de tal forma que permita el añadido de nueva funciones sin interferir con los servicios.

Evolución de los sistemas operativos

El procesamiento en serie

Los primeros computadores **no disponían de un sistema operativo**, por lo que era el usuario el que debía realizar todas las tareas. Estos sistemas tenían dos grandes **problemas**: El usuario debía planificar el tiempo que debía reservar para una cierta operación, y un error durante la compilación o carga obligaba a volver a comenzar el proceso desde el principio.

Procesamiento en lotes sencillos

Este esquema tiene como idea central el **uso del monitor**, que transforma la entrada de trabajos en lotes sencillos y devuelve el control cuando ha acabado para cargar el siguiente programa. Este sistema tiene también grandes **fallos**, como el desperdicio de tiempo, además de que el monitor consume parte de la memoria y la CPU del computador.

Sistemas multiprogramados

La **multiprogramación** es la capacidad de un SO de realizar varios programas que residen en memoria principal al mismo tiempo, aunque necesita interrupciones para las operaciones de E/S.

En este tipo de estructuras, **el SO debe**:

- Proporcionar rutinas de gestión y control sobre las operaciones de E/S.
- Gestionar la memoria principal, ya que varios procesos deben estar listos para ser ejecutados.
- Planificar el uso del procesador, puesto que debe decidir qué proceso ejecutar.

Sistemas de tiempo compartido

Capacidad análoga a la de la multiprogramación, salvo que en este caso los programas que se ejecutan a la vez son **interactivos**, esto es, programas en los que el usuario interacciona con el computador.

Características de un SO

Un sistema operativo debe presentar las siguientes características:

- **Corrección**: El sistema operativo debe poder corregir fallos durante la ejecución (ej. interrupciones y excepciones)
 - **Eficiencia**: Debe permitir y gestionar el uso de los recursos de forma eficiente.
 - **Fiabilidad**: Debe dar respuestas predecibles a ciertas condiciones y proteger el computador y el software contra ataques o software erróneo.
-

-
- **Extensibilidad:** Es obligatorio que puedan añadirse nuevas funciones sin interferir en los servicios ya establecidos.
 - **Tolerancia a fallos:** Un fallo en la ejecución de un programa no debe repercutir ni conllevar a un fallo total del sistema.
 - **Reentrancia:** Un programa o proceso debe tener la capacidad de llamar a otra instancia de sí mismo durante la ejecución.

Sistemas operativos de propósito específico

Sistemas operativos de tiempo real

Un sistema de tiempo real es un sistema informático que no sólo requiere que los resultados calculados sean 'correctos', sino que esos resultados deben también producirse dentro de un periodo específico de tiempo. Los resultados producidos tras el periodo de tiempo especificado no tienen valor real, aunque sean correctos. Estos sistemas pueden clasificarse en dos tipos distintos en función de la importancia de la segunda condición antes descrita:

- **Sistemas de tiempo real estricto (Hard Real Time):** Tienen unos requisitos muy fuertes que garantizan que las tareas de tiempo real críticas se completen dentro del periodo especificado.
- **Sistemas de tiempo real no estricto (Soft Real Time):** Los requisitos en estos sistemas no son tan fuertes, limitándose a garantizar que las tareas críticas de tiempo real tengan prioridad sobre otras tareas y que retengan esa prioridad hasta complementarse. Muchos sistemas operativos comerciales, como GNU-Linux, entran dentro de este grupo de sistemas.

Sistemas operativos en red y sistemas operativos distribuidos

Sistema informático distribuido

Un sistema informático distribuido es un conjunto de ordenadores sin memoria ni reloj común interconectados a través de un sistema de comunicación que puede permitir que estén geográficamente dispersos. Este sistema de comunicación constituye la única posibilidad de comunicar información. La comunicación se realiza mediante operaciones de enviar/recibir mensajes, siendo la duración del envío de un mensaje grande respecto al tiempo de ejecución de una instrucción máquina. Un sistema distribuido tiene como **objetivo** compartir recursos o permitir la dispersión geográfica.

Sistema operativo de red

Un sistema operativo en red se trata de una **red de máquinas**, que suelen ser estaciones de trabajo, y una o dos máquinas que actúan como servidoras. Se trata de un añadido a los sistemas operativos locales, es decir, cada máquina tiene su propio sistema operativo. El usuario conoce la existencia de los múltiples computadores y debe trabajar con ellos de forma explícita, conociendo dónde se ubican los archivos y dónde se ejecutan los procesos.

Sistema operativo distribuido

Un sistema operativo distribuido es un **sistema operativo común compartido** por una red de computadores, pareciendo a los usuarios un sistema operativo normal centralizado. En un sistema operativo distribuido el usuario no conoce dónde se encuentran los archivos o dónde se ejecutan los procesos.

Sistema operativo paralelo

Un sistema operativo paralelo consiste en **una máquina** en la cual se disponen de **varios procesadores**, los cuales disponen de sus propias unidades de control, unidades aritmético-lógicas y registros pero que comparten una misma memoria principal y acceso a los dispositivos de E/S.

En un sistema operativo paralelo, puede dividirse una tarea en múltiples actividades realizadas por los distintos procesadores, lo que a su vez implica que la comunicación y sincronización en este tipo de sistemas es muy eficiente.

Estructura de un sistema operativo

Sistemas operativos monolíticos

Los sistemas operativos monolíticos los constituyen la gran mayoría de los primeros sistemas operativos, desarrollados en los años 50. Todo el sistema operativo se ejecuta como **un sólo ejecutable en modo kernel**. La falta de estructura de estos sistemas operativos tiene como consecuencia que se escriba como una colección de procedimientos enlazados entre sí en un único programa binario ejecutable extenso, además de dejar la libertad de que un proceso llame a cualquier otro. Normalmente, un núcleo monolítico se implementa como un único proceso con todos los elementos compartiendo el mismo espacio de direcciones.

El **núcleo/kernel** es una parte del SO que incluye las funciones más frecuentemente utilizadas y otras porciones del sistema operativo que se encuentren en uso. Gestiona la planificación de hilos, el intercambio de procesos, las excepciones... Esta parte, a diferencia del resto, no es paginable ni expulsable. Esta parte del sistema operativo no puede contener errores.

Este tipo de sistemas operativos presenta una **gran cantidad de inconvenientes**, entre los que se destacan:

- Ya que su falta de estructura provoca que un fallo en un programa de usuario haga que el sistema completo falle, podemos destacar la poca fiabilidad y tolerancia a fallos de estos sistemas operativos.
- Debido a su gran extensión y a su falta de consideración hacia las dependencias e interacciones, estos sistemas presentan una gran dificultad de comprensión y mantenimiento, además de ser poco extensibles.

Sistemas operativos por capas

En estos sistemas operativos, un poco más modernos que los anteriores, las funciones y procesos se agrupan jerárquicamente y sólo se producen interacciones entre capas o niveles adyacentes. Cada nivel dispone de estructuras de datos y rutinas que pueden utilizar los niveles superiores, y éste a su vez puede invocar operaciones de niveles inferiores. Este sistema proporciona una gran **simplicidad de construcción**, pero la mayoría de las capas o niveles siguen ejecutándose en modo núcleo.

Ejemplo: THE

En esta estructura se vieron implicados los siguientes problemas:

- Los sistemas de capas deben ser jerárquicos pero los sistemas reales son más complejos (no suelen ser "puros").
- Existe una **sobrecarga de comunicaciones** entre los procesos de distintas capas.

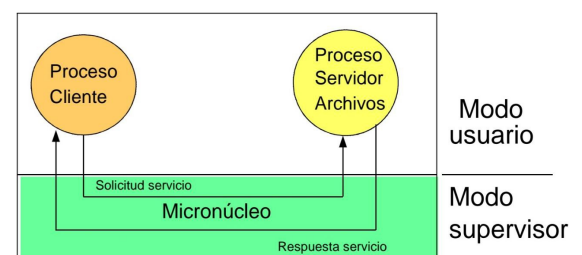
5:	Programas de Usuario
4:	Búfering para dispositivos de E/S
3:	Manejador de consola del operador
2:	Gestión de memoria
1:	Planificación de la CPU
Nivel 0:	Hardware

Estructura microkernel o micronúcleo

La **idea central** de la estructura de microkernel o micronúcleo es eliminar todos los componentes posibles del núcleo e implementarlos como módulos a nivel de usuario, resultando en un **núcleo más pequeño**, que actúa como intercambiador y validador de mensajes entre los distintos componentes. Por ejemplo, si una aplicación quiere abrir un archivo, manda un mensaje al servidor del sistema de archivos, el cual realiza el servicio y devuelve el mensaje a la aplicación, bajo previo consentimiento del núcleo.

Existen una **amplitud de ventajas** a la hora de usar este tipo de estructuras, entre las que destacaremos dos:

- Debido a la modularización, presenta una **mayor fiabilidad**, puesto que es mucho más difícil afectar negativamente a componentes del sistema ajenos al proceso con un error.



Además , la modularización permite una gran extensibilidad, ya que no es necesario construir un nuevo núcleo cuando se añade una nueva característica, simplemente ha de modificarse el módulo en cuestión.

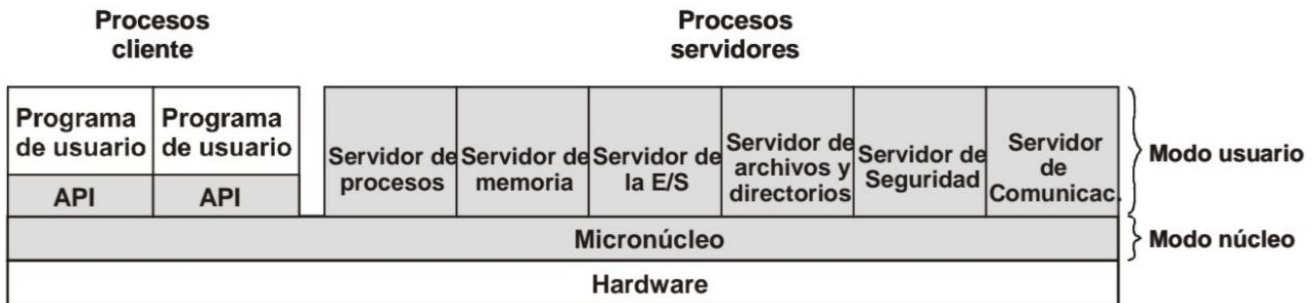
- Es **adaptable** a sistemas operativos distribuidos, puesto que un proceso puede enviar un mensaje sin saber dónde reside el servicio pedido.

La gran **desventaja**, sin embargo, de esta estructura, es el **rendimiento**, puesto que lleva más tiempo construir y enviar un mensaje a través del micronúcleo y aceptar y decodificar la respuesta que simplemente hacer una llamada a un servicio.

Máquinas virtuales

Tipos de software que simulan una copia exacta del software.

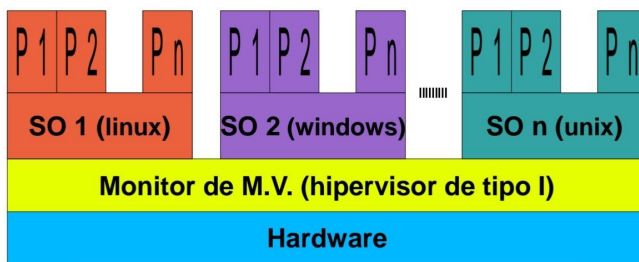
Es una técnica que actualmente está en auge → la capacidad de procesamiento actual mitiga la ineficiencia de las máquinas virtuales.



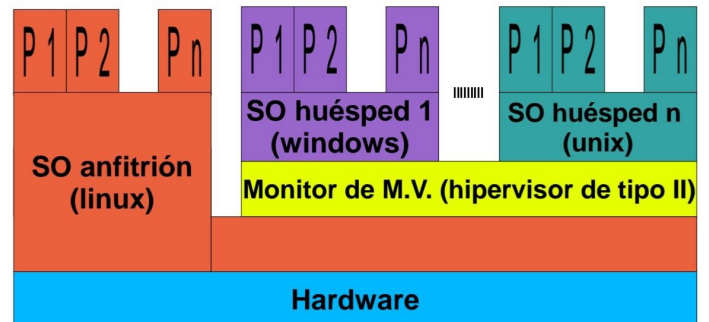
La principal **desventaja** es la ineficiencia.

Modelos

- El monitor solo abstrae el hardware
- Es más eficiente



- El monitor reduce la capa del SO anfitrión
- Pide reducir la eficiencia del SO anfitrión



SO's de Red

Un sistema operativo en red se trata de una red de máquinas, que suelen ser estaciones de trabajo, y una o dos máquinas que actúan como servidoras. Se trata de un añadido a los sistemas operativos locales, es decir, cada máquina tiene su propio sistema operativo y sus propios usuarios. El usuario conoce la existencia de los múltiples computadores y debe trabajar con ellos de forma explícita, conociendo dónde se ubican los archivos y dónde se ejecutan los procesos. Lo que les diferencia de los sistemas operativos de un solo procesador es la necesidad de software especial como: controlador de interfaz de la red, programas de conexión y acceso a archivos remoto.

SO's Paralelos

Un sistema operativo paralelo consiste en una máquina en la cual se disponen de varios procesadores, los cuales disponen de sus propias unidades de control, unidades aritmético-lógicas y registros pero que comparten una misma memoria principal y acceso a los dispositivos de E/S. En un sistema operativo paralelo, puede dividirse una tarea en múltiples actividades realizadas por los distintos procesadores, lo que a su vez implica que la comunicación y sincronización en este tipo de sistemas es muy eficiente.

Simétrico (SMP): cada procesador ejecuta una copia idéntica del SO (buen rendimiento).

Asimétrico (ASMP): un procesador maestro ejecuta el SO, los procesadores esclavos ejecutan procesos de usuario. Peor escalabilidad.