

# TEMA-1.pdf



**Anónimo**



**Sistemas Operativos**



**2º Grado en Ingeniería Informática**



**Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación  
Universidad de Granada**

De la imprenta a la nube, de las descargas a Netflix.  
Fuera los temarios infinitos, bienvenido Wuolah.

¿Y para tu futuro? ¿A quién le vas a confiar tu formación?

**The Globe, cursos del siglo XXI**

Descúbrelo  
en estos QR



# Estudiar **sin publi** es posible.

Compra Wuolah Coins y que nada te distraiga durante el estudio.



## TEMA 1 ESTRUCTURAS DE SO's

### ÍNDICE

1. Arquitecturas monolíticas, en capas, microkernel y maquinas virtuales
  - 1.1 Sistema monolítico
  - 1.2 Sistema de capas. Modelo THE
  - 1.3 Microkernel
  - 1.4 Modelo cliente/servidor
  - 1.5 Maquinas virtuales
2. Sistemas operativos de propósito específico
  - 2.1 SO de tiempo real
  - 2.2 Multicomputadores
  - 2.3 Multiprocesador



WUOLAH

# 1. Arquitectura

## 1.1 Sistema monolítico

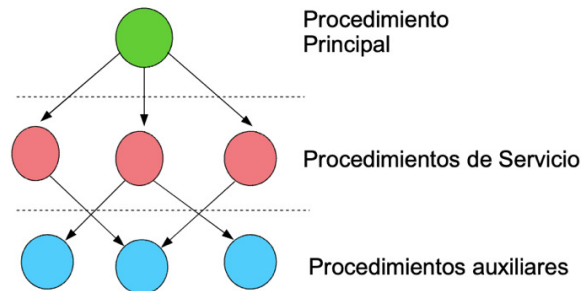
No tienen una estructura bien definida

El SO está formado por un conjunto de procedimientos de forma que cada uno puede llamar a los demás cuando lo necesite

Todas las funciones que ofrece el sistema operativo se ejecutan en un modo supervisor

SO -> ejecuta en un único espacio de direcciones

En este tipo de sistemas no se aplica el principio de ocultación de información



Modelo simple de estructura de un SO monolítico

### --PROBLEMAS--

- Difíciles de comprender y modificar
- No son fiables (un error en alguna parte puede provocar la caída del sistema)
- Difíciles de mantener

## 1.2 Sistema de capas. Modelo THE

En un sistema de capas, el SO se organiza como una jerarquía de capas donde cada capa es una máquina más abstracta para la capa superior (cada capa ofrece una interfaz clara y bien definida a la capa superior y solo utiliza servicios que le ofrece la capa inferior)



- Modularidad y ocultamiento de información (una capa no necesita conocer cómo se ha implementado otra capa cuando le solicita información, únicamente necesita conocer la interfaz)

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

Esto facilita la depuración, verificación del sistema ya que permite ir construyendo y depurando por separado. El sistema estaba compuesto de una serie de procesos secuenciales. Estos procesos se sincronizan con declaraciones explícitas de sincronización. Se puede probar y verificar de forma independiente cada proceso.

### --PROBLEMAS--

- Es difícil construir seguridad debido a que hay muchas interacciones entre capas adyacentes

# Exámenes, preguntas, apuntes.



- Los sistemas de capas deben ser jerárquicos pero los sistemas reales son mas complejos
- Sobrecarga de comunicaciones entre procesos de distintas capas
- Las capas tienen demasiada funcionalidad y cambios en una capa pueden tener numerosos efectos. Como resultado es difícil implementar versiones a medida del SO básico con algunas funciones añadidas o eliminadas

A menudo los sistemas están modelados con esta estructura, pero no están así contruidos

### 1.3 Microkernel

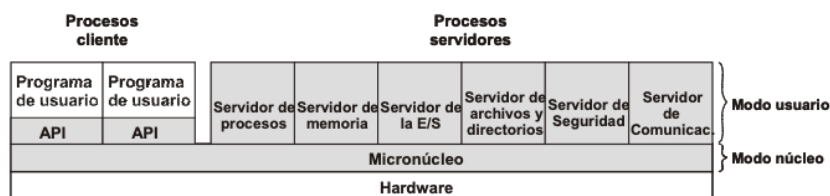
Solamente las funciones absolutamente esenciales del SO están en el núcleo (mayor flexibilidad)

Los servicios y aplicaciones menos esenciales se construyen sobre el micronúcleo y se ejecutan en modo usuario.

Para solicitar un servicio, el proceso de usuario envía un mensaje al proceso servidor, que realiza el servicio y devuelve al cliente una respuesta. (Mayor sobrecarga por envío/recepción de mensajes)

La característica general es que muchos servicios que tradicionalmente habían formado parte del SO ahora son subsistemas externos que interactúan con el núcleo y entre ellos mismos. (Ej: servidores de archivos, gestores de memoria virtual...)

La arquitectura del micronúcleo reemplaza la tradicional estructura vertical y estratificada en capas por una horizontal. Los componentes del SO externos al micronúcleo se implementan como servidores de procesos; interactúan entre ellos dos a dos, normalmente por paso de mensajes a través del micronúcleo



El micronúcleo también realiza una función de protección, previene el paso de mensajes a no ser que el intercambio este permitido

Por ejemplo, si una aplicación quiere abrir un archivo, manda un mensaje al servidor del sistema de archivos. Si quiere crear un proceso o hilo, manda un mensaje al servidor de procesos. Cada uno de los servidores puede mandar mensajes al resto de los servidores y puede invocar funciones primitivas del micronúcleo. Es decir, es una arquitectura cliente/servidor dentro de un solo computador.

Algunos sistemas micronúcleo permiten servidores en modo sistema

- Mas eficiente, pero rompe la filosofía micronúcleo

# Estudiar **sin publi** es posible.

Compra Wuolah Coins y que nada te distraiga durante el estudio.



- Servidores son programas independientes, pero se ejecutan en mismo espacio de direcciones del micronúcleo. No usan IPCs para comunicarse

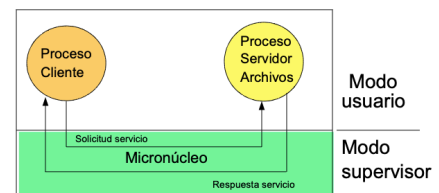
## --VENTAJAS--

- Interfaces uniformes. No hay que diferenciar entre servicios a nivel de núcleo y a nivel de usuario, todos se proporcionan a través del paso de mensajes
- Extensibilidad. Permite agregar nuevos servicios, así como la realización de múltiples servicios en la misma área funcional
- Flexibilidad. No solo se pueden añadir nuevas características al SO, además las características existentes se pueden eliminar para realizar una implementación mas pequeña y mas eficiente
- Portabilidad. En la arquitectura micronúcleo todo o gran parte del código específico del procesador esta en el micronúcleo, por tanto, los cambios necesarios para transferir el sistema a un nuevo procesador son menores y tienden a estar unidos en grupos lógicos.
- Fiabilidad. Un micronúcleo
- Soporte de sistemas distribuidos. Cuando se envía un mensaje desde un cliente hasta un proceso servidor, el mensaje debe incluir un identificador del servicio pedido. Si se configura un sistema distribuido (un clúster por ej) de tal forma que todos los procesos y servicios tengan identificadores únicos, entonces habrá una sola imagen del sistema a nivel de micronúcleo

## 1.4 Modelo Cliente/Servidor

Este tipo de modelo consiste en implementar la mayor parte de los servicios y funciones del SO en procesos de usuario dejando solo una pequeña parte del SO ejecutando en modo núcleo.

A esa parte se le denomina micronúcleo y a los procesos que ejecutan el resto de las funciones se les denomina servidores.



- Cuando un proceso realiza una llamada a un servicio hay un cambio de modo (no de contexto). Si el servicio tardara mucho, el SO decide si retirarle la CPU y bloquearlo -> si hay un cambio de contexto
- Cuando cambiamos entre "Proceso cliente" y "Proceso servidor" se producen cambios de modo y contexto. Exactamente: 2 cambios de modo y 4 cambios de contexto.



WUOLAH



Cambios de modo <b>vs</b> contexto. Implicaciones
<b>Cambios de contexto:</b> hay un cambio de proceso y hay que guardar toda la información asociada al proceso que va a ser cambiado
<b>Cambio de modo:</b> es simplemente un cambio de privilegio. Producido por una interrupción, excepción o llamada al sistema.
Cambio de contexto (dispatcher) → cambio de modo (SO)

### 1.5 Maquinas virtuales

Software que implementa una maquina virtual (= o ≠ maquina real). Cada copia es una replica exacta del hardware.

El SO crea una maquina virtual pero extendida (abstracción del hardware)

La capacidad de procesamiento actual mitiga la ineficiencia de las

Maquinas Virtuales

Los procesadores mas actuales incluyen soporte para la misma.

Las maquinas virtuales permiten mejorar el aislamiento entre los usuarios, si bien incrementan la sobrecarga en las llamadas al sistema.

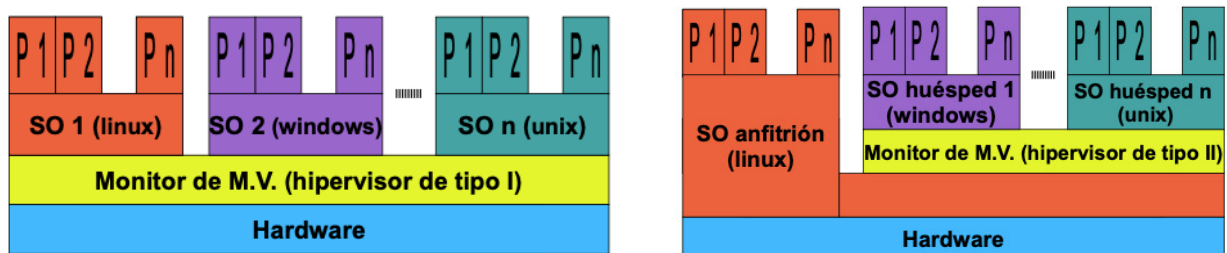
Un buen ejemplo de uso de maquinas virtuales se puede encontrar en el sistema operativo MVS de IBM.

Cada maquina virtual dota a cada usuario de un subconjunto, mas o menos completo, de características del hardware, que puede ser muy distinto de la implementación física real del mismo.

Cuando un usuario accede al sistema operativo MVS se le asigna una maquina virtual.

Este entorno incluye una emulación completa hardware que, posteriormente se proyecta en el hardware real. De esta forma cada usuario solo puede acceder a su maquina virtual.

Además de proporcionar aislamiento, las maquinas virtuales permiten limitar el acceso de los usuarios a los recursos que realmente puedan necesitar. De esta forma se puede reducir sensiblemente el rango de los objetos a proteger por el SO.



## 2. Sistemas operativos de propósito específico

### 2.1 SO de tiempo real

Computación de tiempo real: aquella en la que la corrección del sistema depende no solo del resultado lógico de la computación sino también del momento en el que se producen los resultados.

En general, en un sistema de tiempo real, algunas de las tareas son tareas de tiempo real, las cuales tienen cierto grado de urgencia. Tales tareas intentan controlar o reaccionar a eventos que tienen lugar en el mundo exterior. La **idea básica** es que el SO debe garantizar la respuesta a sucesos físicos en intervalos de tiempo fijos.

Así normalmente es posible asociar un plazo de tiempo límite con una tarea concreta, donde tal plazo especifica el instante de comienzo o de finalización.

Los SO de tiempo real se utilizan para aplicaciones especializadas, como por ejemplo sistemas de control.

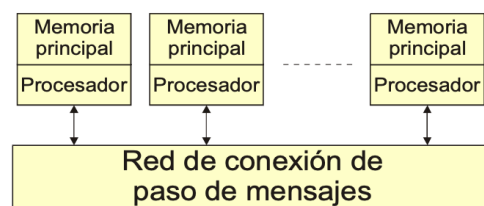
Un SO de estas características se obtendría modificando la planificación de procesos de un SO con carácter general.

- SO de tiempo real duro: aquella que debe cumplir su plazo límite; de otro modo se producirá un daño inaceptable o error fatal del sistema.
- SO de tiempo real blando: tiene asociado un plazo límite deseable pero no obligatorio; sigue teniendo sentido planificar y completar la tarea incluso cuando su plazo límite ya haya vencido

#### --PROBLEMAS--

- Planificar las actividades con el fin de satisfacer todos los requisitos críticos

### 2.2 Multicomputadores



PC<sub>1</sub> puede acceder remotamente a los ficheros de PC<sub>N</sub> con el uso de programas como telnet o ftp.

Lo interesante sería un SO completamente distribuido que o se vea la red y que o tengo que hacer llamadas al sistema que depende del SO que monte para obtener la información que necesita.

SO 100% distribuidos no existen por lo difícil que supone igualar tantas máquinas (se suele lidiar con un sistema de archivos distribuido)



# Estudiar **sin publi** es posible.

Compra Wuolah Coins y que nada te distraiga durante el estudio.



Un SO de estas características mejora el uso de recursos y aumenta la fiabilidad del sistema (si un ordenador falla, no se cae el sistema pues hay mas ejecutando código del SO)

- SO de Red:

Los usuarios son conscientes de la existencia de varias computadoras y debe trabajar con ellas de manera explícita. Cada nodo ejecuta su propio SO local y tiene sus propios usuarios. El SO de red es un añadido al SO local, que permite a las máquinas interactuar con los servidores. Normalmente se utiliza una arquitectura de comunicaciones común para dar soporte a estas aplicaciones de red.

Lo que les diferencia de los SO de un solo procesador es la necesidad de software especial como:

- Controlador de interfaz de la red
- Programas de conexión y acceso a archivos remotos

- SO distribuidos:

Es un sistema operativo común compartido por una red de computadores.

A los usuarios les parece un SO normal centralizado, pero les proporciona **acceso transparente** a los recursos de diversas máquinas.

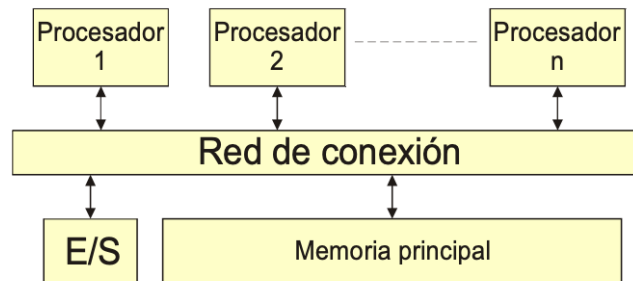
Son sistemas débilmente acoplados (sistemas sin memoria común)

Permiten la compartición de recursos distribuidos, hardware o software y permiten algún paralelismo (pero el aumento de velocidad no es el objetivo).

Aumenta la fiabilidad del sistema y normalmente incorporan un conjunto de funciones de comunicación mas sencillas para proporcionar mayor eficiencia

## 2.3 Multiprocesador

Los procesadores comparten memoria y reloj.



### --VENTAJAS--

- Aumenta el rendimiento de las aplicaciones

Simétrico (SMP): cada procesador puede usar el SO y ejecuta una copia idéntica del SO, cada procesador lleva su planificación, ofrece buen rendimiento. Es el mas usado



### --PROBLEMAS--

- Hay que tener cuidado con el paralelismo, varios procesadores acceden a la tabla de registros del SO simultáneamente.

### --VENTAJAS--

- Aumenta la velocidad de ejecución de las aplicaciones. Hay que indicar (mediante llamadas al sistema) que partes del código se pueden ejecutar concurrentemente (hebras en lugar de procesos)
- Tolerantes a fallos (si un procesador falla, no se cae el sistema pues hay mas corriendo el SO)

Asimétrico (ASMP): uno de los procesadores es el "maestro" que es el que ejecuta el código del SO, el resto "esclavos" ejecutan las aplicaciones. Este multiprocesamiento esta en desuso ya que si el maestro falla, se cae todo el sistema provocando cuello de botella ya que caen los otros procesadores con las aplicaciones que ejecutasen pues el SO no esta disponible para continuar con la planificación de procesos. Peor escalabilidad.

- SO paralelos:

Son sistemas que tienen mas de un procesador compartiendo el bus de la computadora, el reloj y en algunas ocasiones la memoria y dispositivos periféricos. Son sistemas fuertemente acoplados.

Características:

- Soportan aplicaciones paralelas que desean obtener aumento de velocidad de tareas computacionalmente complejas (realizan mas trabajo en menos tiempo)
- Necesitan primitivas básicas para dividir una tarea en múltiples actividades paralelas
- Proporcionan comunicación y sincronización eficiente entre esas actividades
- Ejecutan programas atendiendo de manera concurrente varios procesos de un mismo usuario
- Cada procesador cuenta con su propia memoria local (se mantienen copias de cada proceso)
- Permiten compartir de manera dinámica procesos y recursos entre los diferentes procesadores y hay dos tipos ([SMP](#) y [ASMP](#))