

Tema 1. Conceptos fundamentales de los Sistemas Operativos

Competencias:

- ✓ Conocer la ubicación del sistema operativo dentro de un sistema informático y cuál es su cometido dentro de dicho sistema.
- ✓ Clasificar e identificar las posibles estructuras internas de un sistema operativo.
- ✓ Saber elegir el modelo de diseño más adecuado para un tipo de sistema operativo a partir de sus requisitos y restricciones.
- ✓ Tomar un papel más activo en el aprendizaje.
- ✓ Capacidad dialéctica, siendo capaz de argumentar y defender sus decisiones.

Tema 1. Conceptos fundamentales de los Sistemas Operativos

1. Introducción a los Sistemas Operativos.

1. Niveles del software. Concepto de Sistema Operativo.
2. Funciones principales de un Sistema Operativo.

2. Evolución y clasificación de los Sistemas Operativos. **(Actividad 1)**

1. Evolución histórica de los Sistemas Operativos.
2. Clasificación de los Sistemas Operativos.

3. Estructuras de los Sistemas Operativos.

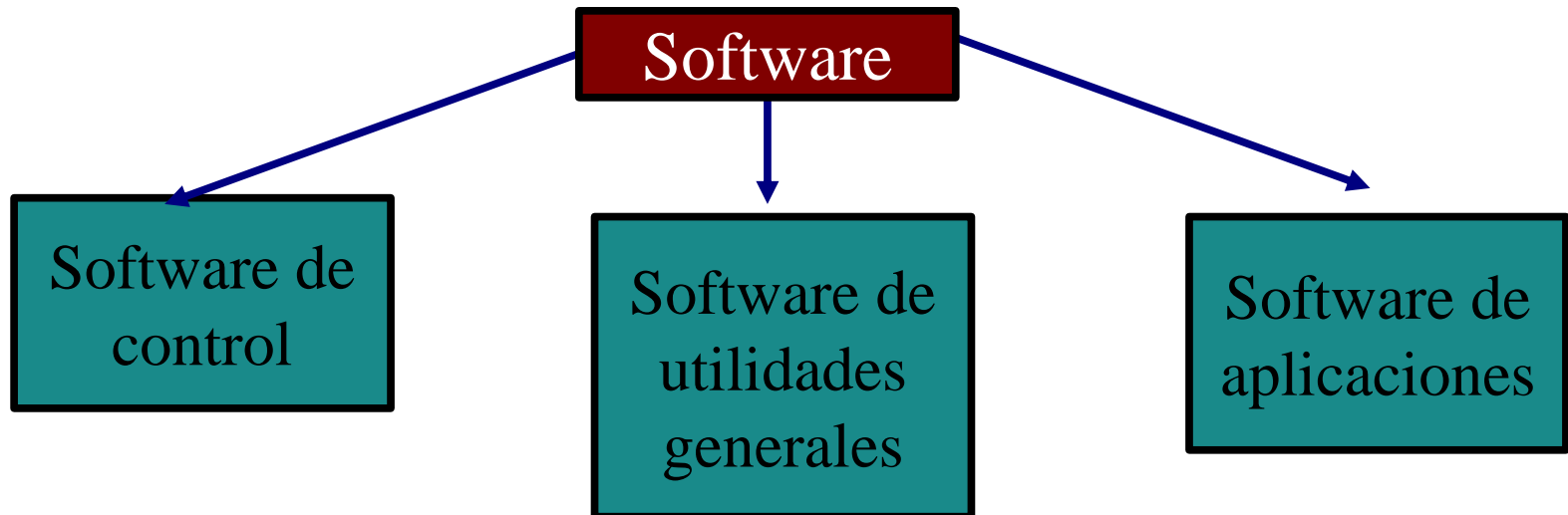
1. Sistemas monolíticos.
2. Sistemas en estratos.
3. Máquinas virtuales.
4. Modelo cliente-servidor.
5. Estructura orientada al objeto.
6. Sistemas híbridos.

1. Introducción a los Sistemas Operativos (I)

1.1. Niveles del software. Concepto de Sistema Operativo.

Soporte lógico o software: conjunto de programas asociados a un computador (los suministrados por el constructor, los adquiridos en empresas especializadas y los redactados por los propios usuarios).

Niveles del software.



1. Introducción a los Sistemas Operativos (I)

1.1. Niveles del software. Concepto de Sistema Operativo.

¿Que es un Sistema Operativo? **Software.**

¿Para que se usa un Sistema Operativo? **Para comunicarse con la máquina física.**

Definición: Un *Sistema Operativo* es un programa (o conjunto de subprogramas o módulos) de control que tiene por objeto facilitar el uso del ordenador y conseguir que este se utilice eficientemente.

- Actúa como interfase entre el usuario y la máquina física,
- Controla la ejecución de otros programas,
- Gestiona y asigna directamente los recursos hardware a los programas,
- Contabiliza la utilización de los recursos,
- Protege los datos y los programas,
- Gestiona y mantiene ficheros (memoria secundaria),
- etc.

1. Introducción a los Sistemas Operativos (II)

1.2. Funciones principales de un Sistema Operativo. (I)

1. **Inicializar la máquina:** preparar el ordenador para su funcionamiento.

a) *Inicialización total* (Initial Program Loading (IPL), Bootstrapping).

b) *Inicialización parcial.*

1. Introducción a los Sistemas Operativos (II)

1.2. Funciones principales de un Sistema Operativo. (II)

2. Servir de máquina extendida (virtual): ocultar los detalles del hardware al usuario y proporcionar un entorno más cómodo.

Objetivos:

- a) *Seguridad:* el S. O. debe evitar que la ejecución de los programas se interfieran unos entre otros.

Instrucciones

No peligrosas

Peligrosas

Modos de operación del Hardware

Modo no privilegiado
(modo usuario)

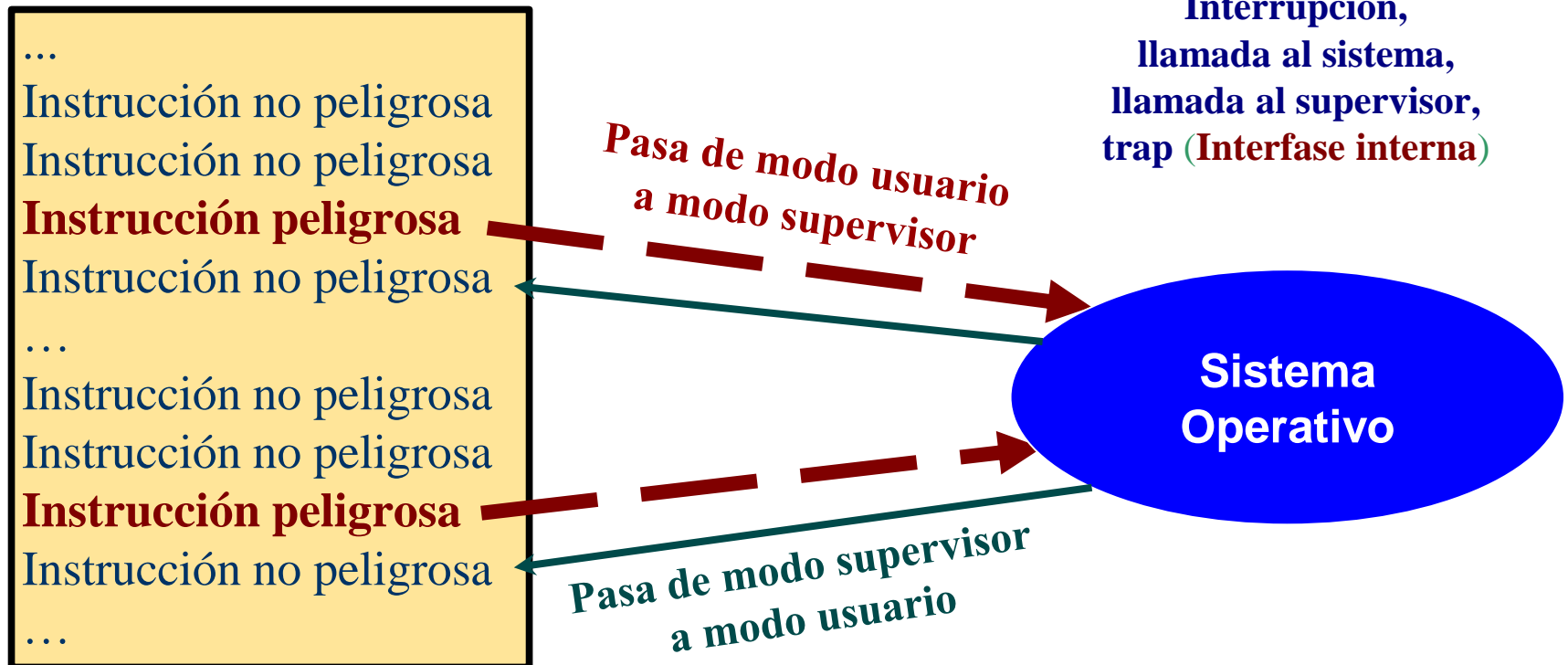
Modo privilegiado
(modo supervisor)



1. Introducción a los Sistemas Operativos (II)

1.2. Funciones principales de un Sistema Operativo. (III)

Programa en ejecución



1. Introducción a los Sistemas Operativos (III)

1.2. Funciones principales de un Sistema Operativo. (IV)

b) Abstracción: los S.O. construyen recursos (virtuales) de alto nivel a partir de los recursos de más bajo nivel (físicos). La máquina física se transforma en una *máquina virtual*. Con el lenguaje de comandos del S. O. (shell, **Interfase externa**) se invocan a esos servicios.

3. Administrar los recursos para su funcionamiento: el S.O. es el responsable de:

- a) asignar* a un programa todos los recursos que necesite. Para ello, debe ser justo en el reparto y en el tiempo asignado, impidiendo que no se favorezca a determinados programas;
- b) controlar* el uso correcto de los recursos de forma que los programas no se interfieran.

1. Introducción a los Sistemas Operativos (IV)

Característica de un Sistema Operativo

Un S.O. debe ser:

- **Determinista:** el mismo programa ejecutado con los mismos datos debe dar los mismos resultados en cualquier momento y en cualquier ejecución.
- **In determinista:** el S. O. debe responder a circunstancias que pueden ocurrir en un orden impredecible.

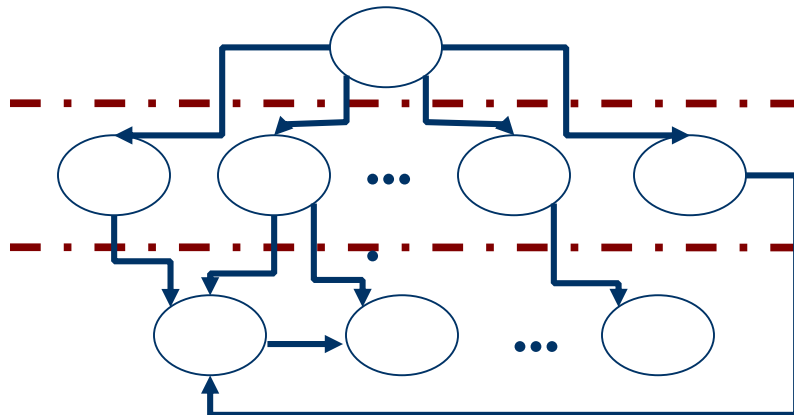
3. Estructura de los Sistemas Operativos (I)

3.1. Sistemas monolíticos.

Descripción: sin estructura definida. Se componen de un conjunto de procedimientos, donde cada uno de ellos puede llamar a todos los demás.

Proceso para requerir un servicio del S. O.: el programa de usuario coloca los parámetros necesarios en los lugares previamente definidos, y seguidamente ejecuta una llamada al supervisor (se cambia de modo usuario a modo supervisor), transfiriendo el control al S. O. que examina dichos parámetros y determina el procedimiento que debe invocar. Cuando se complete la llamada devuelve el control al programa de usuario.

S. O. con una pequeña estructura de niveles:



Procedimiento Principal

Procedimientos de Servicio

Procedimientos Auxiliares

3. Estructura de los Sistemas Operativos (II)

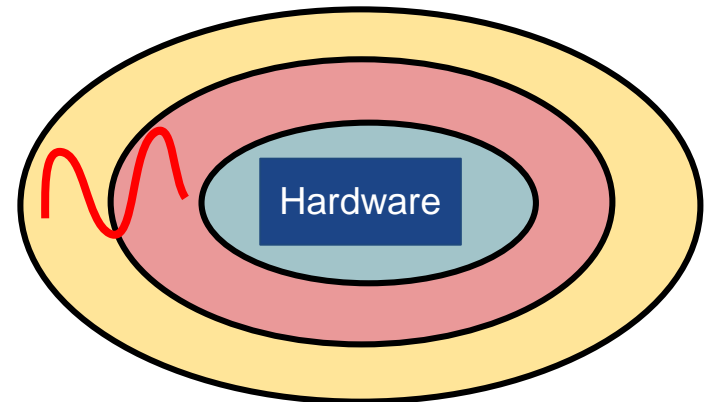
3.2. Sistemas en estratos.

Descripción: se organiza en una jerarquía de estratos, estando construido cada uno de ellos sobre el otro que tiene menor jerarquía que él.

Ejemplos:

- **THE** (Dijkstra, 1968): sistema de procesamiento por lotes.
- **MULTICS:** anillos concéntricos alrededor del hardware, donde los internos son más privilegiados que los externos.

Operador de THE
Programas de usuario
Administración de E/S
Comunicación entre operador y procesos
Administración de memoria y tambor
Distribución del procesador y multiprogramación

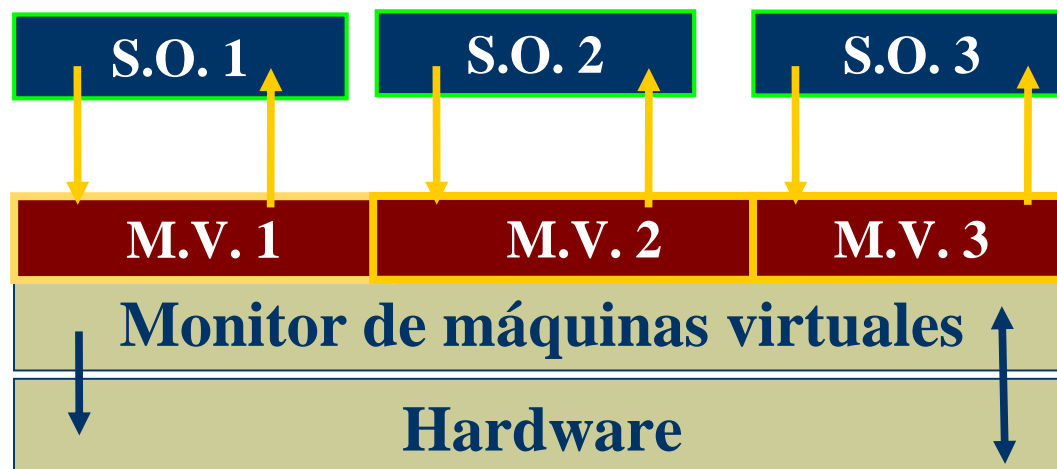


3. Estructura de los Sistemas Operativos (III)

3.3. Máquinas virtuales.

Descripción: crea ilusiones (máquinas virtuales) de la máquina real, permitiendo que en cada máquina virtual se ejecute un S. O. distinto.

El programa de control es el que se ejecuta directamente sobre el propio hardware y ofrece al nivel inmediatamente superior varias máquinas virtuales.

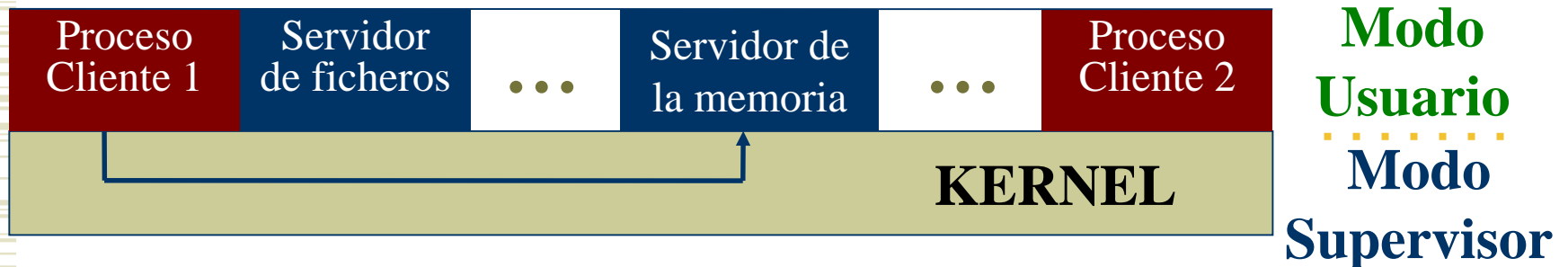


3. Estructura de los Sistemas Operativos (IV)

3.4. Modelo Cliente-Servidor.

Descripción: su objetivo es minimizar el kernel desplazando el código de todos sus servicios a estratos lo más superiores posibles. Para ello, la mayoría de sus funciones se implementan como procesos de usuario, denominados *procesos servidores*, de forma que cuando un proceso de usuario llamado *proceso cliente*, necesita un servicio del S.O. lo que hace es enviar un mensaje al proceso servidor correspondiente, que realiza el trabajo y devuelve la respuesta.

El kernel lo único que hace es implementar la comunicación entre clientes y servidores y entre servidores y el hardware.

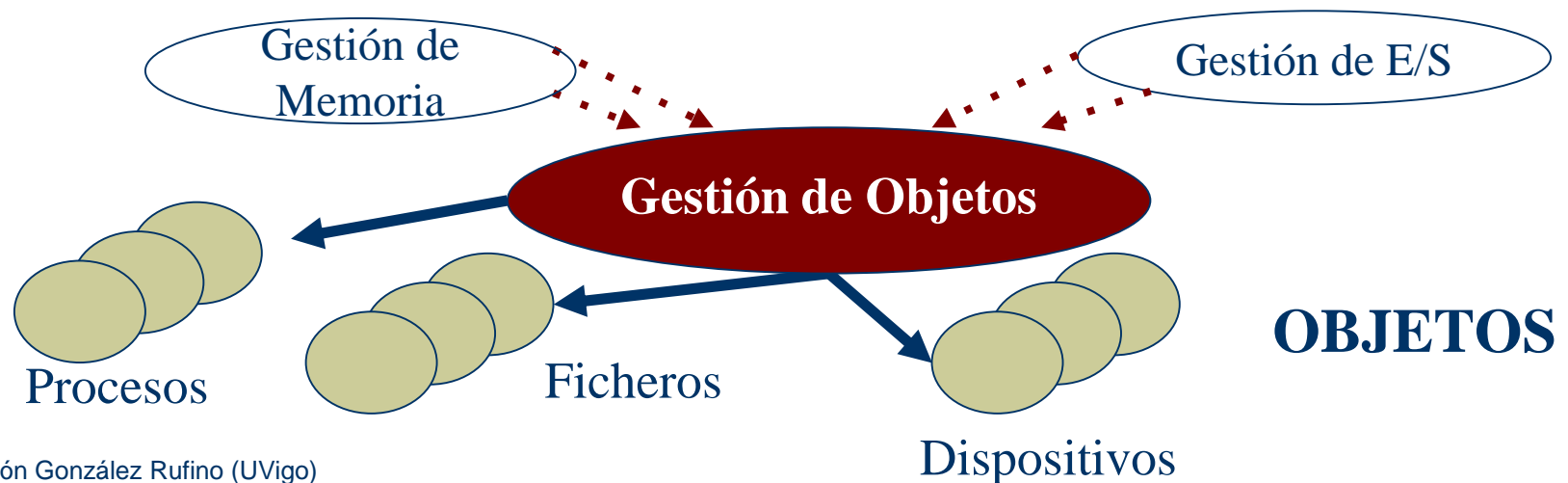


3. Estructura de los Sistemas Operativos (V)

3.5. Estructura orientada al objeto.

Descripción: se basan en una colección de objetos, donde las funciones del sistema son un tipo de objeto (ficheros, dispositivos, etc). La interacción entre dichos objetos viene determinada por las *capacidades* que cada uno tenga para actuar con el otro.

El kernel es el responsable de mantener las definiciones de los tipos de objetos soportados y de controlar los privilegios de acceso a los mismos. Cuando un programa desee realizar una operación sobre un objeto determinado, deberá ejecutar una llamada al sistema, indicando qué derechos tiene para poder utilizarlo y qué operación intenta llevar a cabo. Como resultado de dicha llamada, el sistema validará la petición y, si puede ser aceptada, permitirá la realización de dicha operación.

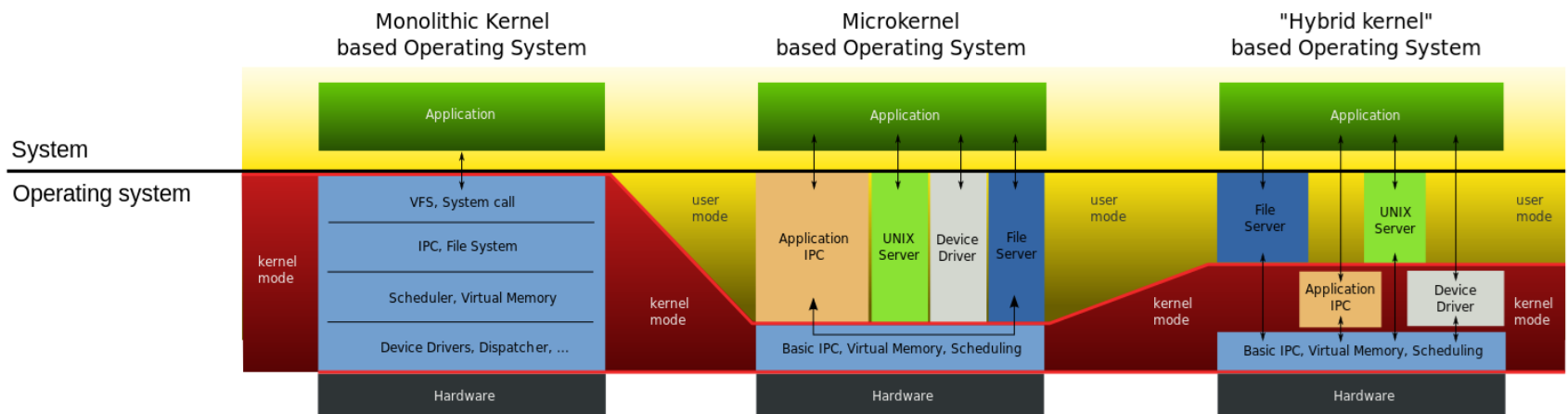


3. Estructura de los Sistemas Operativos (VI)

3.6. Sistemas híbridos.

Descripción: son similares a los sistemas cliente-servidor (o microkernel) aunque añaden ciertas funcionalidades al kernel para que se ejecuten más rápido que si permanecen en el espacio de usuario.

Se les llama híbridos porque usan mecanismos o conceptos de arquitectura de los sistemas monolíticos y de los sistemas cliente-servidor.



Fuente: <https://commons.wikimedia.org/wiki/File:OS-structure2.svg>