

## Tema 1 – L1: Conceptos de Sistemas Operativos

### 1 ¿Por qué estudiar SSOO?

- El sistema operativo SO influye mucho en el **funcionamiento general**, en la **seguridad** y en el **rendimiento** del computador
- La importancia de la **elección** de un determinado SO para las empresas es cada día mayor, a nivel **estratégico**
- Conocer el funcionamiento del SO es fundamental para desarrollar aplicaciones que obtengan buenas **prestaciones** y para **comprender la causa de muchos problemas**.
- Un ingeniero debe ser agnóstico en sistemas operativos; es decir, no tener fijación especial por un solo SO, y tener un conocimiento polivalente y aplicable a todo SO.

### 2 Estructura y funcionamiento de un computador

\*\*El contenido a este apartado hace referencia a los conceptos vistos en la asignatura de Estructura de Computadores, vista en el cuatrimestre pasado. Para repasar cualquiera de los conceptos, puedes descargarlos los apuntes mediante este link: [\*\*¡Pulsa Aquí!\*\*](#)

### 3 Concepto de SO. Componentes y estructura del SO

#### Sistema operativo

El sistema operativo es un programa que actúa de intermediario entre el usuario y el hardware. **Objetivos:**

- Ejecutar programas
- Hacer uso eficiente de los recursos
- Proporcionar visión de máquina virtual extendida

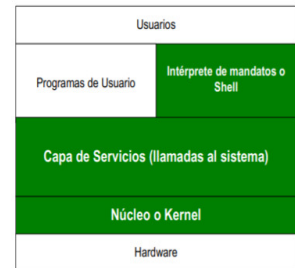
#### Funciones del sistema operativo

- **Gestionar los recursos** (UCP, memoria, ...)
  - Asignación y recuperación de recursos
  - Protección de los usuarios
  - Contabilidad/monitorización
  - Soporte de usuario
- **Máquina extendida / ofrecer servicios**
  - Ejecución de programas (procesos)
  - Órdenes de E/S
  - Operaciones sobre archivos
  - Detección y tratamiento de errores
- **Manejar la interfaz de usuario**
  - Shell

Niveles del sistema operativo

El SO está formado conceptualmente por 3 capas principales:

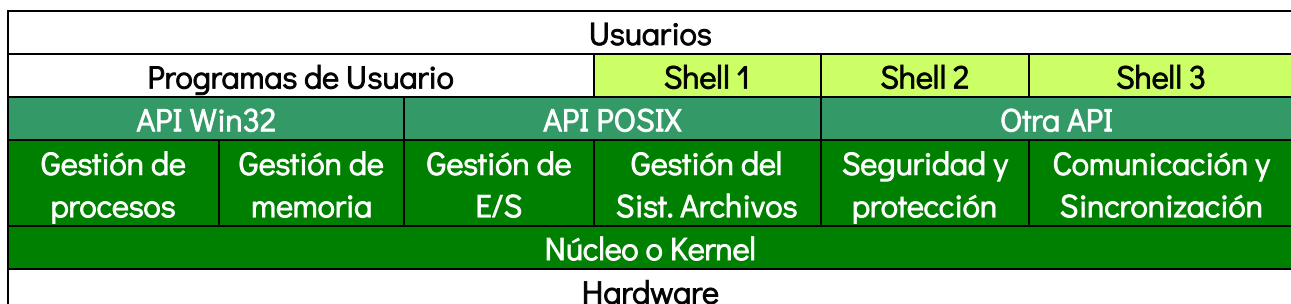
- Intérprete de mandatos o *shell*
- **Servicios o llamadas al sistema (API)** (guarda las instrucciones de tratamiento de las interrupciones o llamadas al sistema)
- **Núcleo o Kernel**

Estructura conceptual

- Modos de ejecución:
  - **Modo usuario:** Ejecución de procesos de usuario
  - **Modo supervisor/núcleo/kernel:** Ejecución del núcleo del SO
- Los procesos y el SO utilizan espacios de memoria separados
- Cuando un proceso necesita un servicio lo solicita al SO mediante una **llamada al sistema**: el sistema operativo entra en ejecución para realizar la función solicitada

Componentes del Sistema Operativo

- El SO puede ofrecer **varias API** simultáneamente
- Se pueden disponer de **varios intérpretes** de mandatos
- El **núcleo o kernel** interactúa directamente con el **hardware**. Ejecuta las **funciones básicas** de gestión del procesador, de la memoria y de las operaciones

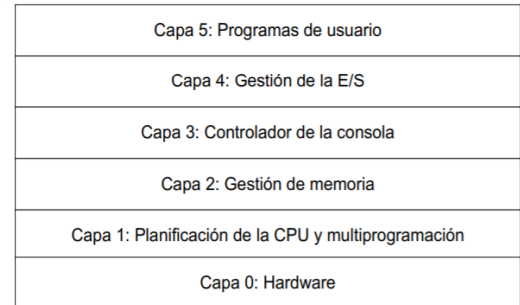
Alternativas de la estructura de sistemas operativos**Sistemas Operativos Monolíticos**

- No hay una **estructura clara** y bien definida
- **Todo el código** del SO está enlazado como un **único ejecutable** (un solo espacio de direcciones) que se ejecuta en modo “núcleo”
- El código presenta cierta organización pero internamente **no existe ocupación de información entre los distintos módulos**, pudiéndose llamar unos a otros sin restricciones
- Aunque es más eficiente en su funcionamiento, su **desarrollo y mantenimiento** es muy **complejo**
- Ejemplo: Los SO hasta los años 80, incluido UNIX



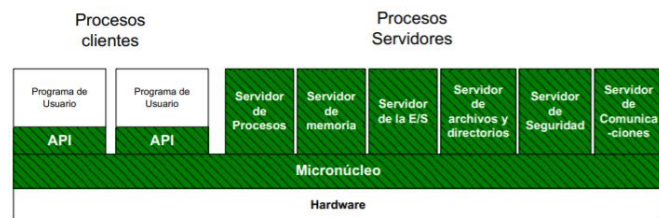
### Sistemas Operativos estructurados por capas

- El sistema se organiza como un **conjunto de capas superpuestas**, cada una con una **interfaz clara** y bien definida
- Cada capa se apoya en los servicios de la inmediatamente inferior para realizar sus funciones
- Las **ventajas** son la modularidad y la ocultación de la información, que facilita mucho el desarrollo y la depuración de cada capa por separado
- Esta estructura, sin embargo, **no resulta tan eficiente** porque una determinada operación en la capa superior implica realizar múltiples llamadas desde el nivel superior hasta el nivel inferior
- Hay **dificultad** a la hora de **distribuir funciones** del SO **entre** las distintas **capas**
- Ejemplos: THE, OS/2



### Sistemas Operativos estructurados por cliente/servidor (microkernel)

- Implementar la **mayor parte de los servicios del SO** como **procesos** de usuario, dejando solo una pequeña parte corriendo en modo núcleo denominada **micronúcleo** o **microkernel**
- Hay dudas sobre qué funciones debes implementar realmente en el microkernel pero al menos: interrupciones, gestión básica de procesos y memoria, y servicios básicos de comunicación
- **Ventajas:**
  - Muy flexible: cada servidor puede desarrollarse y depurarse más fácilmente al tratarse de programas pequeños y especializados
  - Es fácilmente extensible a un modelo distribuido
- **Desventajas:**
  - Sobrecarga en la ejecución de los servicios
- Ejemplos: Mac OS y Windows NT (aunque los servicios se ejecutan en kernel para mejorar el rendimiento)



### Clasificación de los Sistemas Operativos

- Número de procesos simultáneos:
  - Monotarea
  - Multitarea
- Modo de interacción:
  - Interactivo
  - Por lotes (batch)
- Número de usuarios simultáneos:
  - Monousuario
  - Multiusuario
- Número de procesadores:
  - Monoprocesador
  - Multiprocesador
- Número de hilos (threads):
  - Monothread
  - Multithread

- Tipos de usos:
  - Cliente
  - Servidor
  - Empotrado
  - Tiempo real

## 4 Arranque y activación del SO

### Arranque del sistema operativo

El sistema operativo se inicia al encender el computador

- Inicialmente se encuentra en el almacenamiento secundario
- ¿Cómo llega el SO a Memoria Principal?
- ¿Como se inicia su ejecución una vez está en memoria?

### Fases de arranque

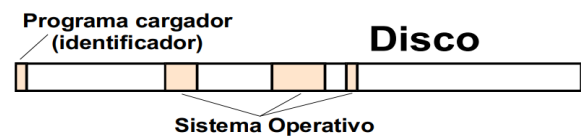
#### Iniciador ROM

- La señal RESET carga **valores predefinidos** en registros
- En el **PC** (contador de programa) se carga la **dirección** de arranque del cargador **ROM**
- Se ejecuta el **iniciador ROM** del sistema:
  - Se ejecuta un test del hardware del sistema
  - Trae a memoria el boot (iniciador) del sistema operativo



#### Cargador del sistema operativo

- El programa cargador se encuentra en el sector de inicio (boot) del disco
- Es responsable de cargar el sistema operativo
- Verifica la presencia del **identificador** en el sector de arranque



#### Parte residente del SO

Es la responsable de la iniciación del sistema operativo. Se encarga de:

- Verificación de consistencia del sistema de ficheros
- Creación de las estructuras de datos internas
- Activación de modo de memoria virtual
- Carga el resto del sistema operativo residente
- Habilita interrupciones
- Crea procesos iniciales

### Parada del computador

Para acelerar la ejecución el sistema operativo mantiene información en memoria no actualizada a disco

- Al apagar hay que volcar dicha información a disco y terminar la ejecución de todos los procesos

- Si no se hace volcado (apagando bruscamente) → Se pierde información y el sistema de ficheros se queda inconsistente

Otras alternativas en computadores personales:

- **Hibernación:** se guarda el estado de la memoria principal a disco
- **Apagado en espera (standby):** Parada del computador que mantiene alimentada a la memoria principal

### Generación del sistema operativo

- Los sistemas operativos suelen diseñarse para en una clase de máquinas que incluya diversas configuraciones y un amplia variedad de periféricos
- Es necesario generar una copia del sistema operativo basada en las características de configuración de la máquina
- La generación del sistema operativo se realiza durante la instalación inicial

### Parámetros de generación

- |                               |                                  |
|-------------------------------|----------------------------------|
| ➤ Modelo de CPU               | ➤ Cantidad de memoria disponible |
| ➤ Opiniones de CPU instaladas | ➤ Dispositivos instalados        |
| ➤ Número de CPUs              | ➤ Opciones del sistema operativo |

### Alternativas de generación

- Modificación del código fuente y compilación del sistema operativo
  - Ejecutable totalmente adaptado
  - Generación más lenta y arranque más rápido
- Creación de tablas y selección de módulos de biblioteca precompilada
  - El sistema operativo tiene controladores para todos los dispositivos
  - Solamente se montan los necesarios
  - Generación más rápida y arranque más lento
- Sistema totalmente controlado por tablas
  - Selección en tiempo de ejecución
  - Arranque más lento