

# Fundamentos de Sistemas Operativos

## Trabajo Semana #3

Universidad Politécnica Internacional

### Estudiantes:

Jefferson Villalobos López

Andrew Barrantes James

Isaac Ruiz Jiménez

Joshua Mena Mendoza

Diego Loria Soto

El Capítulo 1, titulado Introducción, establece las bases conceptuales de un sistema operativo (SO) y la arquitectura de una computadora.

## 1.1 ¿Qué hace un sistema operativo?

El sistema operativo (SO) es un programa esencial que actúa como intermediario entre el usuario y el hardware de la computadora. Su doble objetivo principal es:

- \* Ejecutar programas de usuario de forma práctica y eficiente.
- \* Gestionar el hardware de la computadora para asegurar el correcto funcionamiento del sistema e impedir el uso inapropiado.

### Vista del Usuario (1.1.1)

La perspectiva del SO depende del entorno informático:

- \* Sistemas de un solo usuario (PC, Estación de trabajo): El enfoque está en la facilidad de uso y el rendimiento, priorizando que el usuario obtenga el máximo trabajo.
- \* Sistemas multiusuario (Mainframe, Servidores): El enfoque cambia a maximizar la utilización de recursos (CPU, memoria, E/S) para asegurar que todos los usuarios tengan acceso equitativo.
- \* Sistemas de dispositivos portátiles (PDA, smartphones): El diseño se centra en gestionar recursos limitados, como la potencia, la velocidad y la interfaz.

### Vista del Sistema (1.1.2)

Desde el punto de vista del sistema, el SO se define como:

- \* Asignador de recursos (Resource Allocator): Administra todos los recursos del sistema (CPU, memoria, almacenamiento, E/S) y debe decidir cómo asignarlos a los programas y usuarios de la forma más eficiente y justa.
- \* Programa de control (Control Program): Controla la ejecución de los programas para prevenir errores y el uso indebido del hardware.

### Definición de Sistemas Operativos (1.1.3)

No existe una definición única. Sin embargo, la definición más aceptada es que el SO es el kernel (núcleo), el programa que se ejecuta en todo momento en la computadora. El kernel es la única parte del sistema que se mantiene residente en la memoria.

## 1.2 Organización de una computadora

Una computadora moderna consta de uno o más CPU (unidad central de procesamiento) y varios controladores de dispositivos (para discos, audio, video, etc.), todos conectados a través de un bus común que proporciona acceso a la memoria compartida.

### Funcionamiento de una computadora (1.2.1)

El funcionamiento se basa en un ciclo de ejecución de instrucciones:

- \* Programa de arranque (Bootstrap): Cuando se enciende la computadora, se ejecuta un programa de arranque (almacenado en la ROM o EPROM) que inicializa el sistema y carga el kernel del SO en la memoria para iniciar su ejecución.

- \* Interrupciones: Las operaciones están impulsadas por interrupciones. Cuando un evento (de hardware o software) ocurre, se genera una interrupción, que detiene la CPU y transfiere el control a una rutina de servicio de interrupción específica. Este mecanismo permite al SO gestionar las operaciones de forma concurrente.

- \* Acceso directo a memoria (DMA): Para grandes transferencias de datos (por ejemplo, de disco a memoria), el DMA permite al controlador del dispositivo transferir bloques de datos directamente a la memoria sin la intervención de la CPU, liberándola para otros trabajos.

### Estructura de almacenamiento (1.2.2)

Los programas deben estar en la memoria principal (RAM) para ser ejecutados. El almacenamiento se organiza en una jerarquía que equilibra velocidad y coste:

| Nivel | Velocidad | Característica |

|---|---|---|

| 1. Registros | Más rápida | Almacenamiento dentro de la CPU |

| 2. Caché | Muy rápida | Copia de datos temporales de la memoria principal |

| 3. Memoria principal (RAM) | Rápida | Volátil, se pierde al apagar |

| 4. Almacenamiento secundario | Lento | No volátil, discos magnéticos (HDD), discos electrónicos (SSD/Flash) |

El sistema operativo se encarga de gestionar esta jerarquía y la migración de datos entre los niveles para mejorar el rendimiento.

### Estructura de E/S (1.2.3)

La entrada/salida (E/S) es fundamental para la fiabilidad y el rendimiento.

- \* Cada dispositivo de E/S es gestionado por un controlador de dispositivo, que tiene un búfer local y registros.

- \* El SO utiliza un controlador de dispositivo lógico (device driver) para comunicarse con el controlador físico y presentar una interfaz uniforme al sistema operativo.

## 1.3 Arquitectura de un sistema informático

El SO está diseñado para la arquitectura específica de la computadora:

#### Sistemas de un solo procesador (1.3.1)

Son la mayoría, con una CPU principal que ejecuta instrucciones de propósito general y procesadores especiales para tareas específicas (como los controladores de E/S).

#### Sistemas multiprocesador (1.3.2)

También llamados sistemas paralelos o sistemas de acoplamiento fuerte (tightly coupled), tienen dos o más procesadores que comparten memoria, bus, reloj y periféricos.

Ventajas:

- \* Mayor rendimiento: Más trabajo realizado en menos tiempo.
- \* Economía de escala: Comparten periféricos y almacenamiento, siendo potencialmente más baratos que múltiples sistemas de un solo procesador.
- \* Mayor fiabilidad: Si un procesador falla, el sistema puede seguir funcionando a velocidad reducida (degradación elegante).

Tipos:

- \* Multiprocesamiento Simétrico (SMP): Cada procesador ejecuta una copia idéntica del SO y las tareas se reparten dinámicamente. Es el más común.
- \* Multiprocesamiento Asimétrico: Un procesador maestro controla el sistema, y el resto son procesadores esclavos.

#### Sistemas en cluster (1.3.3)

Consisten en dos o más sistemas informáticos independientes (nodos) interconectados por una red (LAN o más rápida). A diferencia de los multiprocesadores, no comparten memoria ni reloj y están acoplados de forma débil (loosely coupled). Se utilizan principalmente para:

- \* Alta disponibilidad: Si un nodo falla, otro toma el control del servicio.
- \* Alto rendimiento: Permite que las aplicaciones se ejecuten en paralelo a través de múltiples nodos.

Espero que este resumen te sea de gran utilidad para tu estudio.

## 1.5 Operaciones del sistema operativo

Esta sección describe cómo funciona un sistema operativo, basándose en el concepto de interrupciones y la necesidad de una operación en modo dual para garantizar la protección del sistema.

### Mecanismo de Interrupciones y Excepciones

- El sistema operativo se activa por eventos que son manejados por interrupciones (generadas por el hardware) o excepciones/trampas (traps, generadas por el software).
- Una interrupción transfiere el control a una rutina de servicio de interrupción (interrupt service routine), que se encuentra a través de una tabla de direcciones llamada vector de interrupciones.

### Operación en Modo Dual (Doble Modo)

- Para proteger el sistema operativo y otros procesos de fallos o actividades maliciosas, el hardware del sistema proporciona una operación en modo dual.
- El hardware tiene un bit de modo (o modo kernel) que indica el estado actual de la CPU:
  - o Modo kernel (o modo supervisor): Bit de modo = 0. Es el modo más privilegiado y donde se ejecuta el código del sistema operativo.
  - o Modo usuario: Bit de modo = 1. Es el modo menos privilegiado y donde se ejecuta el código del usuario.
- Ciertas instrucciones (como las de E/S o la gestión del temporizador) son privilegiadas y solo pueden ejecutarse en modo kernel.
- Cuando un programa de usuario necesita un servicio del sistema operativo, ejecuta una llamada al sistema (system call), que es una interrupción de software que cambia el modo de la CPU a modo kernel, permitiendo al sistema operativo atender la solicitud.
- El Temporizador es esencial para el modo dual, ya que previene que un programa de usuario se quede en un bucle infinito o monopolice el CPU. El temporizador genera una interrupción después de un período de tiempo definido para devolver el control al sistema operativo.

---

## 1.6 Gestión de procesos

Esta sección introduce el concepto fundamental de proceso y las tareas que el sistema operativo debe realizar para gestionarlos.

### El Concepto de Proceso

- Un proceso es un programa en ejecución y es la unidad de trabajo fundamental en el sistema.
- El sistema informático consta de una colección de procesos que se ejecutan concurrentemente, los cuales se dividen en:
  - o Procesos del sistema: Ejecutan código del sistema operativo.
  - o Procesos de usuario: Ejecutan código del usuario.
- Para ejecutar su tarea, un proceso necesita recursos como tiempo de CPU, memoria, archivos y dispositivos de E/S.

### Funciones de la Gestión de Procesos

El sistema operativo es responsable de las siguientes actividades en lo que se refiere a la gestión de procesos:

- Crear y borrar procesos de usuario y del sistema.
- Suspende y reanuda procesos.
- Proporcionar mecanismos para la sincronización de procesos.
- Proporcionar mecanismos para la comunicación entre procesos.
- Proporcionar mecanismos para el tratamiento de los interbloqueos (deadlocks).

## Resumen: Componentes del Sistema Operativo (Capítulo 1, Secciones 1.7 - 1.9)

### 1.7. Gestión de Memoria

La gestión de memoria se encarga de administrar y coordinar la memoria principal (RAM), la cual es un almacenamiento de acceso rápido compartido por la CPU y los dispositivos de E/S. Es crucial porque la memoria es el único almacén grande de datos al que la CPU puede acceder directamente.

Objetivo: Optimizar el uso de la memoria disponible, tanto física como virtual.

#### Funciones Clave del Sistema Operativo:

Mantener el registro de qué partes de la memoria están actualmente en uso y por quién.

Decidir qué procesos deben cargarse en memoria cuando hay espacio (y desasignarlo cuando terminan).

Asignar dinámicamente el espacio de memoria y garantizar la protección para que los procesos no puedan acceder al espacio de memoria que no les pertenece.

### 1.8. Gestión de Almacenamiento

El sistema operativo gestiona el almacenamiento de datos persistente a través de dos componentes principales: el sistema de archivos y la gestión de almacenamiento secundario.

#### A. Sistema de Archivos

El sistema de archivos proporciona el mecanismo para el almacenamiento y acceso en línea a los datos y programas que residen en los discos.

La unidad lógica de almacenamiento de datos más pequeña que se puede manipular es el archivo. Los archivos se agrupan en directorios para facilitar la organización.

El SO debe gestionar la asignación de espacio, el acceso, la protección de los archivos y la creación y eliminación de archivos y directorios.

### B. Gestión de Almacenamiento Secundario

Dado que la memoria principal es volátil (pierde su contenido al apagar la energía) y relativamente pequeña, los sistemas informáticos deben disponer de almacenamiento secundario (como discos duros y SSD) para almacenar de forma permanente grandes cantidades de datos.

El SO debe gestionar el espacio libre del almacenamiento, la asignación de ese espacio y la planificación de disco (el orden en el que se atienden las solicitudes de E/S del disco) para asegurar una alta velocidad de transferencia.

## 1.9. Protección y Seguridad

La protección y la seguridad son aspectos fundamentales para garantizar el uso adecuado de los recursos del sistema y evitar el acceso no autorizado.

**Protección:** Mecanismos que controlan el acceso de los procesos o usuarios a los recursos definidos del sistema (CPU, memoria, archivos). Su principal función es evitar que un proceso interfiera con otro o con la operación del sistema.

**Seguridad:** Mecanismos que defienden el sistema contra amenazas externas e internas, como virus, denegación de servicio (DoS) o acceso no autorizado. Para lograr la seguridad, el sistema operativo utiliza la autenticación (verificar la identidad de un usuario) y la autorización (otorgar permisos para usar recursos).

**Diferencia Clave:** La protección se enfoca en los mecanismos internos para controlar el acceso, mientras que la seguridad se centra en la defensa del sistema contra todos los ataques externos o fallos que puedan comprometer su integridad y datos.



### 1.10 – Computación distribuida

La computación distribuida representa un cambio fundamental en la forma en que concebimos el procesamiento de datos. En lugar de depender de una única máquina centralizada, este modelo propone que múltiples computadoras independientes —cada una con su propio sistema operativo— trabajen en conjunto a través de una red para ejecutar tareas compartidas. Esta colaboración permite distribuir la carga de trabajo, mejorar la eficiencia y aumentar la tolerancia a fallos. Si una de las máquinas falla, las demás pueden continuar operando sin que el sistema colapse.

Este enfoque ha sido clave en el desarrollo de tecnologías modernas como la computación en la nube, los clústeres de servidores y las. Además de sus ventajas técnicas, la computación distribuida plantea desafíos importantes, como la sincronización entre procesos, la seguridad en la transmisión de datos y la consistencia de la información compartida. A medida que las aplicaciones se vuelven más complejas y demandantes, este modelo se consolida como una solución flexible y escalable para enfrentar los retos del procesamiento masivo de información.

### 1.11 – Computación paralela

La computación paralela se enfoca en acelerar el procesamiento de tareas mediante la ejecución simultánea de múltiples procesos. A diferencia de la computación distribuida, que opera entre distintas máquinas, la paralela se desarrolla dentro de una misma unidad física que cuenta con varios núcleos o procesadores. Este tipo de arquitectura permite dividir una tarea compleja en subtareas que se ejecutan al mismo tiempo, reduciendo significativamente el tiempo total de procesamiento.

Este modelo es especialmente útil en áreas como la simulación científica, el procesamiento de imágenes, el modelado tridimensional y la inteligencia artificial, donde se requiere una gran capacidad de cálculo. Sin embargo, implementar computación paralela no es sencillo: los procesos deben coordinarse cuidadosamente para evitar conflictos en el acceso a recursos compartidos, y el diseño de algoritmos eficientes es crucial para aprovechar al máximo el potencial de esta arquitectura. En definitiva, la computación paralela no solo mejora el rendimiento, sino que también permite abordar problemas que serían imposibles de resolver con un enfoque secuencial.

### 1.12 – Sistemas operativos en tiempo real

Los sistemas operativos en tiempo real están diseñados para responder a eventos externos dentro de límites temporales estrictos y predecibles. En ciertos entornos, como el control de maquinaria industrial, los sistemas médicos o los vehículos autónomos, una demora en la respuesta puede tener consecuencias graves. Por eso, estos sistemas deben garantizar una reacción inmediata y confiable ante cualquier estímulo.

Existen dos tipos principales de sistemas en tiempo real: los duros y los suaves. En los sistemas duros, el incumplimiento de los tiempos de respuesta puede provocar fallos críticos, mientras que en los suaves se permite cierta flexibilidad sin comprometer la funcionalidad general. Lo esencial en ambos casos es la capacidad del sistema operativo para gestionar interrupciones, priorizar tareas y mantener un comportamiento determinista. Estos sistemas no se valoran solo por su velocidad, sino por su precisión y consistencia, cualidades indispensables en aplicaciones donde la seguridad y la estabilidad son prioritarias.

### 1.13 – Sistemas operativos embebidos

Los sistemas operativos embebidos se integran en dispositivos con funciones específicas y recursos limitados. A diferencia de los sistemas operativos generales, que gestionan múltiples aplicaciones y usuarios, los embebidos están diseñados para cumplir una tarea concreta de manera eficiente y confiable. Se encuentran en una amplia variedad de dispositivos cotidianos, como electrodomésticos, relojes inteligentes, sistemas de navegación, sensores industriales y equipos médicos.

Su diseño suele ser compacto, con un núcleo adaptado a las necesidades del dispositivo, y muchas veces operan sin intervención humana durante largos periodos. Estos sistemas deben ser altamente confiables, consumir poca energía y funcionar con una cantidad reducida de memoria. En un mundo cada vez más interconectado, los sistemas embebidos son fundamentales para el desarrollo de tecnologías como el Internet de las cosas, donde cada dispositivo cumple un rol específico dentro de una red más amplia. Su simplicidad y estabilidad los convierten en una solución ideal para entornos donde la eficiencia y la durabilidad son esenciales.