

Aplicación

Conceptos básicos de sistemas operativos



Fundamentos esenciales para comprender el funcionamiento de los sistemas informáticos modernos

Sistema Operativo

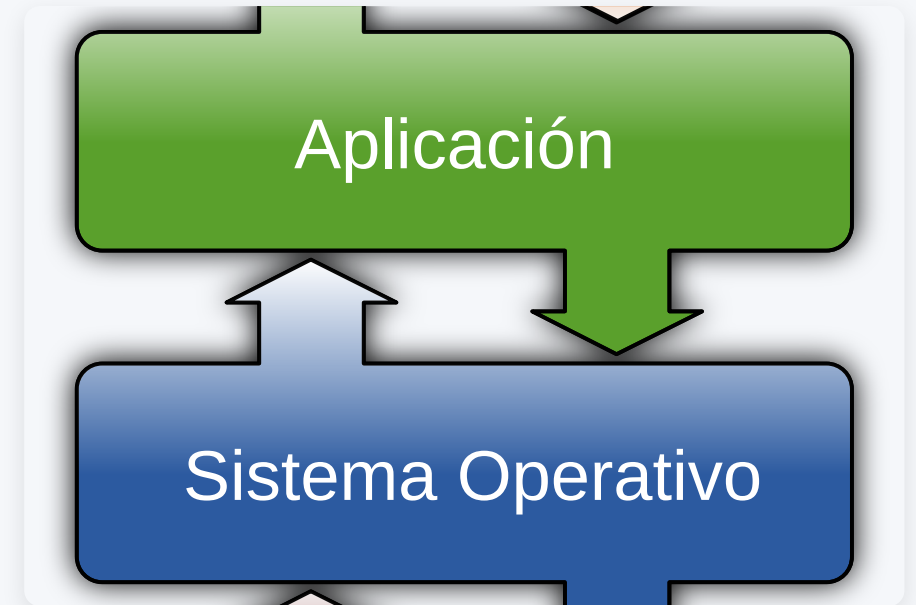
Introducción

💡 Fundamentos esenciales

- ✓ Base para comprender sistemas informáticos **modernos**
- ✓ Cruciales para profesionales de TI, administradores y desarrolladores

⚙️ Más que un simple programa

- ✓ **Ecosistema complejo** que gestiona recursos
- ✓ Proporciona servicios y actúa como **intermediario**
- ✓ Entre hardware y aplicaciones



Componentes fundamentales

Los conceptos básicos de un sistema operativo se agrupan en cinco elementos esenciales:



Los procesos

Programa en ejecución que requiere recursos específicos. Unidad básica de trabajo en un sistema operativo.



Los archivos

Objetos que contienen datos en un sistema informático. Mecanismo de abstracción para almacenamiento permanente.



Llamadas al sistema

Interfaz con el sistema operativo para solicitar servicios. Puente entre espacio de usuario y kernel.



El núcleo

Parte central y crítica del sistema operativo. Intermediario entre aplicaciones y hardware.



El intérprete de comandos

Interfaz entre usuarios y sistema operativo. Traduce órdenes del usuario en acciones ejecutables.

Los procesos

Un proceso es **un programa en ejecución** que requiere recursos específicos del sistema para funcionar.

↔ Estados de un proceso

▶ **Ejecución**: usando CPU

⌚ **Preparado**: esperando CPU

|| **Espera**: esperando E/S

■ **Terminado**: finalizado

⚙ Conceptos relacionados

🕒 **Planificación** y despacho

↔ **Comunicación** entre procesos

🔄 **Sincronización** de recursos

🔲 **Concurrencia** y paralelismo



! Importancia

✓ Unidad básica de **trabajo**

✓ Permite **multitarea**


✓ Facilita **multiprocesamiento**

✓ Optimiza uso de **recursos**

Los archivos


Los archivos son **objetos encargados de contener los datos** en un sistema informático, permitiendo almacenar información de forma permanente.

Atributos y propiedades

 **Nombre** y extensión

 **Tamaño** en bytes

 **Permisos** de acceso

 **Fechas** de creación/modificación

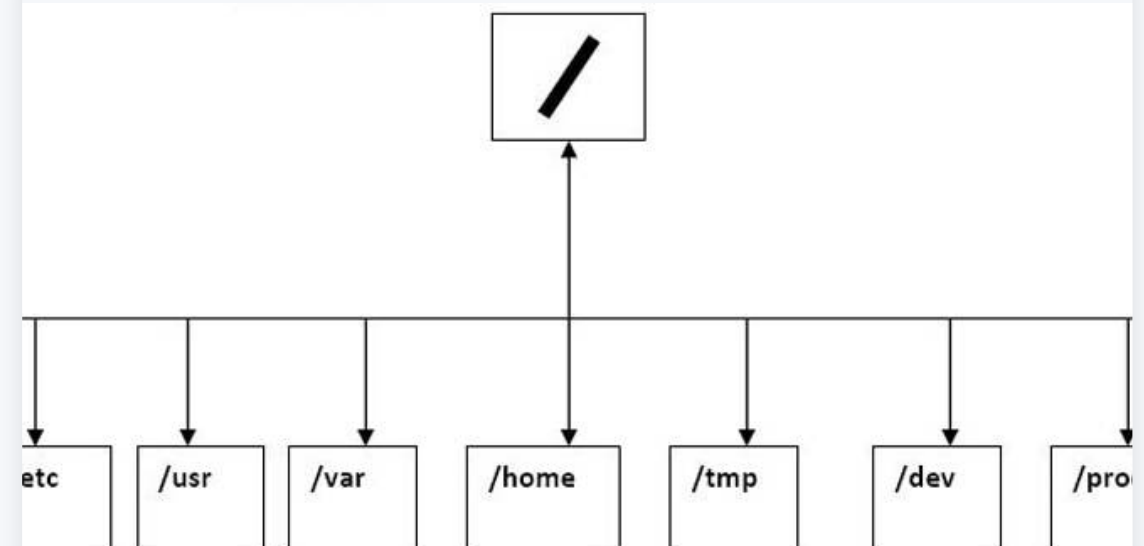
Tipos de archivos

 Archivos de **datos**

 Archivos **ejecutables**

 Archivos de **sistema**

 **Directorios** (carpetas)



Operaciones básicas

 **Creación** de archivos

 **Lectura** de contenido

 **Escritura** de datos

 **Eliminación** de archivos

Sistema de archivos

 Organización **jerárquica**

 Directorio **raíz**

 **Subdirectorios** anidados

Rutas absolutas y relativas

Las llamadas al sistema

Las llamadas al sistema son **el conjunto de instrucciones que sirven como interfaz** con el sistema operativo para solicitar los servicios que ofrece.

→ Función principal

↕ **Puente** entre espacio de usuario y kernel

🛡️ Acceso **controlado** a recursos

✅ Proporciona **seguridad** y protección

🔍 **Abstracción** del hardware

🏠 Categorías principales

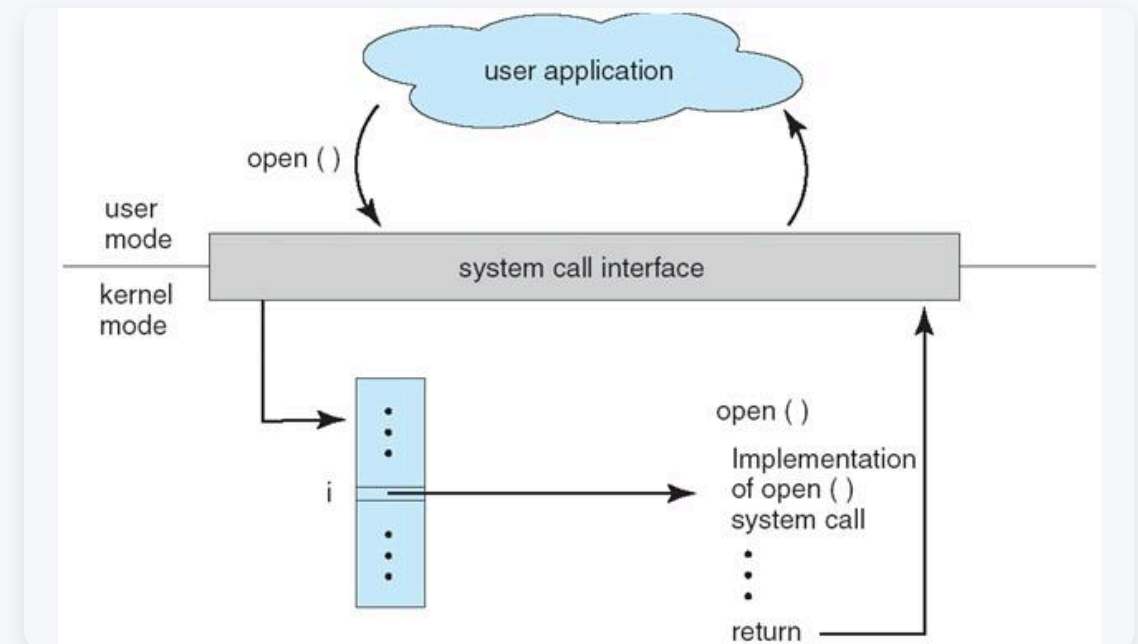
⚙️ Gestión de **procesos**

📁 Gestión de **memoria**

📁 Gestión de **archivos**

🛡️ **Protección** y seguridad

💻 Gestión de **dispositivos** E/S



🔧 Mecanismo de funcionamiento

📞 La aplicación **solicita** un servicio

↔ Se produce un **cambio de modo** (usuario → kernel)

⚙️ El **kernel ejecuta** la operación

← Devuelve el **resultado** a la aplicación

! Importancia

⏏ Base para **desarrollo** de aplicaciones

🔧 Optimización del **rendimiento**

📁 **Interfaz estándar** para programadores

🛡️ Control de **privilegios** de acceso

El núcleo del sistema operativo

El núcleo (kernel) es **la parte central y más crítica** del sistema operativo, que actúa como intermediario entre las aplicaciones y el hardware.

Tipos principales de núcleos

🏠 **Monolíticos**: todo en un solo bloque

🏠 **Microkernel**: mínimo y modular

🔗 **Híbridos**: combinan ambos enfoques

🏠 **Exokernel**: mínima abstracción

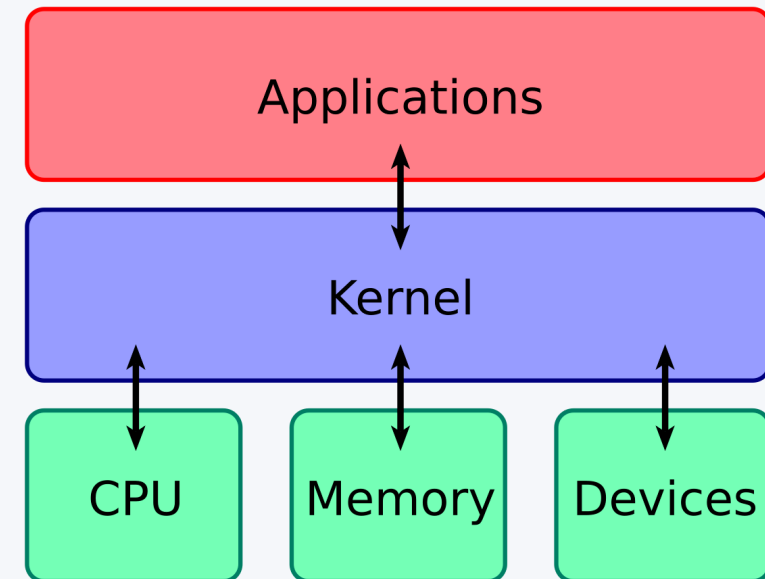
Funciones críticas

⚙️ Gestión de **procesos**

📄 Gestión de **memoria**

💻 Gestión de **dispositivos**

🛡️ **Seguridad** y protección



Características principales

! Ejecución en **modo privilegiado**

📄 **Residente** en memoria

🔌 Cargado durante el **arranque**

🕒 Activo mientras el **sistema opera**

Comparación de tipos

📈 **Monolíticos**: mayor rendimiento

🛡️ **Microkernel**: mayor seguridad

⚖️ **Híbridos**: equilibrio

🏠 **Exokernel**: flexibilidad

El intérprete de comandos

El intérprete de comandos (shell) es **un programa especial que proporciona una interfaz** entre los usuarios interactivos y el sistema operativo.

↔ Función principal

🗨️ Traduce órdenes del usuario

🔗 Intermediario usuario-sistema

🔧 Permite **automatización** de tareas

⚙️ Ejecuta **programas** y comandos

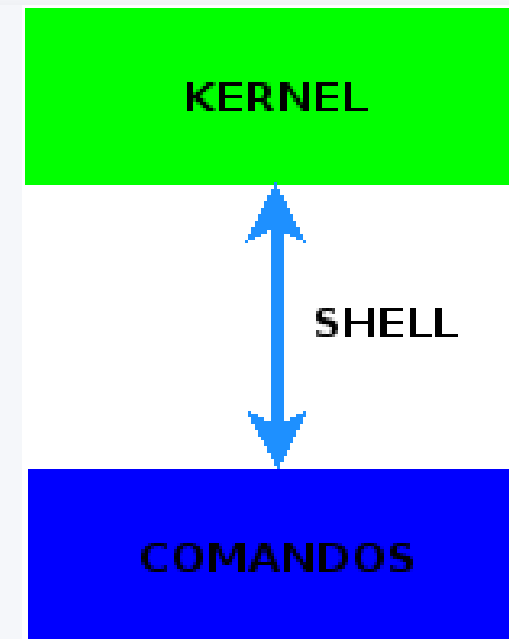
🏠 Tipos de intérpretes modernos

💻 Línea de comandos tradicionales

🔧 Shells avanzados (PowerShell, Zsh)

💻 Entornos integrados (Terminal)

↔ Sistemas híbridos (WSL)



🕒 Ejemplos populares

💻 cmd.exe (Windows)

💻 Bash (Linux/macOS)

💻 PowerShell (Windows)

📱 Zsh (macOS/Linux)

🔑 Características avanzadas

📖 Scripting y automatización

⚙️ Personalización del entorno

🕒 Historial de comandos

📁 Autocompletado de comandos

Relación entre los conceptos básicos

Los cinco componentes fundamentales no funcionan de manera aislada, sino que están profundamente interconectados:



Los **procesos** utilizan **llamadas al sistema** para acceder a los servicios del **núcleo**



Los **archivos** son gestionados por el **núcleo** y accedidos a través de **llamadas al sistema**



El **intérprete de comandos** crea **procesos** que manipulan **archivos** utilizando **llamadas al sistema**



El **núcleo** gestiona la ejecución de **procesos** y la asignación de recursos a los mismos

Esta interconexión forma un sistema cohesivo donde cada componente depende de los demás para proporcionar una experiencia de usuario fluida y eficiente.

Importancia en el funcionamiento del sistema

Entender estos conceptos básicos es esencial porque:



Administración eficiente

Conocer cómo funcionan los **procesos**, **archivos** y el **núcleo** ayuda a optimizar el rendimiento y resolver problemas



Desarrollo de software

Los desarrolladores deben comprender las **llamadas al sistema** y la gestión de **procesos** para crear aplicaciones eficientes



Mejora de la seguridad

Comprender cómo se gestionan los **permisos de archivos** y la **protección del núcleo** es fundamental para mantener un sistema seguro



Automatización de tareas

El conocimiento del **intérprete de comandos** y los **procesos** permite crear scripts para automatizar operaciones repetitivas



Base para aprendizaje avanzado

Estos conceptos son la base para entender tecnologías más avanzadas como **virtualización**, **contenedores** y **computación en la nube**

El dominio de estos conceptos básicos proporciona las herramientas necesarias para comprender, administrar y optimizar cualquier sistema operativo moderno.

Evolución de los conceptos básicos

Aunque estos conceptos fundamentales se mantienen estables, han evolucionado significativamente con el tiempo:



Procesos

Multitarea simple → Multiprogramación avanzada →
Gestión en entornos distribuidos



Núcleo

Monolíticos simples → Arquitecturas híbridas →
Microkernels optimizados



Intérprete de comandos

Interfaces de texto básicas → Entornos ricos →
Integración con IA y scripting avanzado



Archivos

Sistemas básicos → Journaling → Compresión y
cifrado integrado



Llamadas al sistema

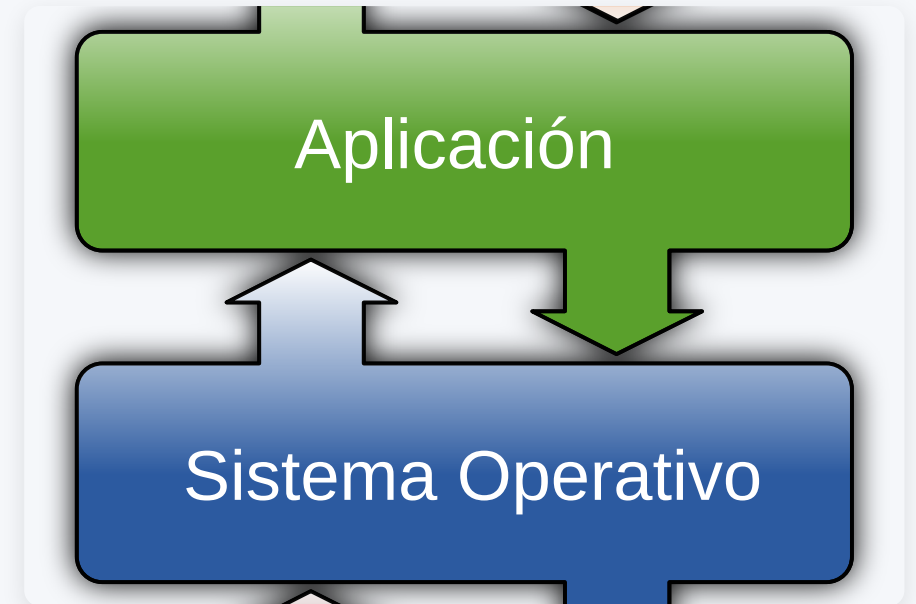
Interfaces limitadas → API completas → Soporte para
virtualización y nube



Conclusión

💡 Base esencial para sistemas informáticos

- 🎓 Proporciona **conocimientos teóricos** y **herramientas prácticas** para trabajar eficientemente
- 💻 Comprensión profunda de **procesos**, **archivos**, **llamadas al sistema**, **núcleo** e **intérprete de comandos**
- 📈 Permite **aprovechar al máximo** los recursos del sistema y **resolver problemas complejos**
- 🔄 Conceptos **fundamentales pero en evolución constante** para adaptarse a nuevos desafíos tecnológicos



Dominar estos conceptos es esencial para cualquier profesional que desee comprender, administrar o desarrollar sobre sistemas informáticos modernos.