

Tema 15

Sistemas operativos

Componentes – Estructura – Funciones – Tipos

 Clave en programación, redes, sistemas y ciberseguridad.

1. Introducción

 El sistema operativo (SO) es el software fundamental que:

- Actúa como intermediario entre hardware y software de aplicación
- Gestiona los recursos del sistema: CPU, memoria, dispositivos, archivos
- Coordina la ejecución de programas de forma segura y concurrente

Conocerlo es esencial para el desarrollo de software y administración de sistemas.

2. Funciones principales del sistema operativo

Un SO moderno gestiona:

- Procesos
- Memoria
- Entrada/salida
- Archivos
- Seguridad y acceso

2.1 Gestión de procesos

- Planificación, ejecución, sincronización y terminación
- Estados de proceso:

Nuevo → Listo → Ejecutando → Bloqueado → Terminado

Algoritmos:

- FIFO
- Round Robin
- SJF
- Colas multinivel



2.2 Gestión de memoria

- Asignación dinámica de memoria
- Técnicas:
 - Paginación
 - Segmentación
 - Swapping
- **Memoria virtual:** simula más memoria de la disponible físicamente
- Protección entre procesos

2.3 Gestión de E/S

- Control de dispositivos a través de **drivers**
- Técnicas:
 - Búferes y cachés
 - Interrupciones
 - Polling

 Interfaces comunes: USB, PCIe, SATA

2.4 Gestión de archivos

- Organización jerárquica: carpetas y ficheros
- Operaciones: crear, leer, escribir, montar, borrar
- Permisos: lectura, escritura, ejecución

Sistemas de archivos comunes:

- ext4
- NTFS
- FAT32
- APFS
- XFS

2.5 Seguridad y control de acceso

- Gestión de usuarios y grupos
- Mecanismos:
 - Autenticación (login)
 - Listas de control de acceso (ACL)
 - Cifrado
 - SELinux, AppArmor

 Prevención de accesos no autorizados

3. Componentes del sistema operativo

3.1 Núcleo (Kernel)

- Parte central del sistema operativo
- Gestiona procesos, memoria, archivos y dispositivos

Tipos de kernel:

- **Monolítico:** todo en uno (Linux clásico)
- **Microkernel:** servicios separados (Minix, QNX)
- **Híbrido:** mezcla (Windows NT, macOS)

3.2 Módulos y gestores

- Subcomponentes del kernel encargados de:
 - Memoria
 - Archivos
 - Red
 - Seguridad
- ✓ Pueden cargarse o descargarse en tiempo de ejecución (modularidad)

3.3 Shell: CLI / GUI

- **Shell:** interfaz entre usuario y sistema operativo
- CLI: `bash` , `sh` , `PowerShell`
- GUI: GNOME, KDE, Windows Explorer

 Permite lanzar comandos o aplicaciones

3.4 Controladores (drivers)

- Traducen instrucciones del SO para el hardware
- Pueden cargarse dinámicamente:
 - `modprobe` (Linux)
 - `.sys` o `.inf` (Windows)

 Sin drivers, el hardware no puede funcionar correctamente

4. Estructura del sistema operativo

Modelo	Características
Monolítico	Todo en un único bloque (Linux clásico)
Microkernel	Servicios mínimos en el núcleo (Minix, QNX)
Modular	Carga dinámica de módulos (Linux moderno)
Cliente-servidor	Servicios como procesos separados (TS Windows)
Híbrido	Combina varios modelos (Windows NT, macOS)

5. Tipos de sistemas operativos

5.1 Según el dispositivo

Tipo	Ejemplos
Escritorio	Windows, macOS, Ubuntu
Móvil	Android, iOS
Servidor	Debian, CentOS, Windows Server
Embebido	RTOS, Raspbian, OpenWRT
Tiempo real	FreeRTOS, QNX

5.2 Según la arquitectura

Tipo	Descripción
Monousuario	Un único usuario simultáneo
Multiusuario	Varios usuarios gestionados por el SO
Monotarea	Una tarea a la vez
Multitarea	Varias tareas concurrentes
Distribuidos	Varios sistemas trabajando en red
Virtualizados	Sistemas en máquinas virtuales (VMware)
Para contenedores	Ligeros: Alpine, Distroless


6. Tendencias actuales

 Nuevas líneas de evolución:

- **Contenerización** y virtualización nativa
- **Seguridad reforzada:**
 - TPM (Trusted Platform Module)
 - SELinux, AppArmor
- **Automatización de tareas:** scripting, DevOps
- **SO minimalistas** para entornos cloud e IoT
- **Kernel Live Patching**, actualizaciones sin reiniciar

Conclusión

El sistema operativo:

- ✓ Es el **corazón del sistema informático**
- ✓ Gestiona recursos, usuarios, seguridad y dispositivos
- ✓ Se adapta a múltiples plataformas: escritorio, servidores, móviles, cloud
-  Comprender sus componentes y estructura es clave para cualquier profesional TIC.