

Resumen Extendido - Sistemas Operativos

Este resumen abarca los temas fundamentales de sistemas operativos con un enfoque especial en la gestión de procesos y memoria, dos de los pilares del funcionamiento de cualquier SO moderno.

1. ¿Qué es un Sistema Operativo?

Es un software que actúa como intermediario entre el hardware de la computadora y los programas del usuario. Sus funciones principales incluyen la gestión de procesos, memoria, dispositivos, archivos y la seguridad del sistema.

2. Proceso de Arranque (Boot)

Comienza cuando se enciende la computadora:

- BIOS/UEFI inicializa el hardware y busca un disco de arranque.
- El MBR/GPT contiene información de particiones y el bootloader.
- El bootloader (como GRUB) carga el kernel del sistema operativo.
- El kernel inicializa componentes y monta el sistema de archivos raíz.
- Se ejecuta init o systemd, que lanza servicios y el entorno gráfico o línea de comandos.

3. Tipos de Kernel

- Monolítico: todo el SO corre en modo kernel, incluyendo drivers. Ej: Linux.
- Microkernel: solo funciones esenciales están en modo kernel. Ej: Minix.
- Híbrido: combina ambos, busca un balance entre rendimiento y estabilidad. Ej: Windows.

4. Gestión de Procesos

Un proceso es un programa en ejecución. Cada proceso tiene su propio espacio de memoria, registros y estado.

El Sistema Operativo utiliza una estructura llamada PCB (Process Control Block) para almacenar información del proceso como:

- Estado (nuevo, listo, ejecutando, esperando, terminado)
- Contador de programa
- Registros de CPU
- Información de planificación

El cambio de contexto ocurre cuando el SO guarda el estado de un proceso y carga el de otro. Las llamadas comunes para gestión de procesos son:

- fork(): crea un nuevo proceso hijo

- exec(): reemplaza el contenido de un proceso
- wait(): espera a que termine un hijo
- exit(): termina el proceso actual

5. Gestión de Memoria

El SO debe controlar cómo se asigna, usa y libera la memoria RAM.

- Memoria principal (RAM): se usa para ejecutar programas.
- Paginación: divide la memoria en bloques (páginas). Esto permite que el SO asigne memoria de forma flexible.
- Memoria virtual: permite que los programas crean tener más memoria de la que realmente hay, usando el disco como respaldo.
- Swapping: cuando la RAM se llena, se pueden mover procesos inactivos al disco (área de intercambio o swap).

Estas técnicas evitan que un proceso acceda a la memoria de otro y mejoran el uso eficiente del recurso.

6. Llamadas al Sistema

Son funciones que permiten que los programas accedan a servicios del sistema operativo. Se ejecutan en modo kernel y permiten:

- Crear y controlar procesos
- Leer y escribir archivos
- Comunicar procesos entre sí
- Administrar dispositivos y memoria

7. Modo Usuario vs. Modo Kernel

El modo usuario restringe el acceso a hardware y memoria para proteger el sistema.

El modo kernel tiene control total del hardware.

Cuando un programa necesita un servicio del SO, se hace una transición al modo kernel mediante una llamada al sistema.

8. Sistema de Archivos

Es la forma en que el SO organiza y almacena archivos en un disco.

- Estructura jerárquica (directorios y subdirectorios).
- Permisos: en Linux (rwx para usuario, grupo y otros), en Windows ACLs.

- Sistemas de archivos: ext4 (Linux), NTFS (Windows).

9. Linux vs. Windows

- Linux: kernel monolítico, código abierto, uso común en servidores y programación.
- Windows: kernel híbrido, cerrado, ampliamente usado en escritorio.
- Linux usa Bash; Windows, PowerShell o CMD.
- Linux soporta ext4, Btrfs; Windows usa NTFS.