

Sistemas Operativos

Material de estudio universitario — Taller 2: NotebookLM

Curso de Computadoras | Nivel Universitario

Este documento fue preparado para ser subido a NotebookLM durante el taller. Contiene teoría, tablas, casos prácticos y preguntas de reflexión sobre Sistemas Operativos.

Contenido de este documento

- 1. Introducción a los Sistemas Operativos
- 2. Las 5 Funciones Principales del SO
- 3. Tipos de Sistemas Operativos
- 4. Gestión de Procesos y Memoria
- 5. Sistemas de Archivos
- 6. Seguridad y Protección
- 7. Sistemas Operativos Modernos: Comparativa
- 8. Casos Prácticos de Aplicación Docente
- 9. Preguntas de Reflexión y Autoevaluación

1. Introducción a los Sistemas Operativos

Un sistema operativo (SO) es el software fundamental que actúa como intermediario entre el usuario, las aplicaciones y el hardware de una computadora. Sin el sistema operativo, ninguna otra aplicación podría ejecutarse: es la capa que traduce las instrucciones de alto nivel del usuario en señales que el hardware puede entender y ejecutar.

El primer SO reconocible fue el GM-NAA I/O, desarrollado en 1956 para la computadora IBM 704. Desde entonces, los sistemas operativos han evolucionado desde simples monitores de proceso por lotes hasta

complejos sistemas multitarea, multiusuario y distribuidos que gestionan millones de procesos simultáneos en servidores en la nube.

Definición formal

- Un sistema operativo es un conjunto de programas que gestionan los recursos de hardware y software de un sistema informático, proporcionando servicios comunes a los programas de aplicación y actuando como interfaz entre el usuario y el hardware.
- También se define como el programa más importante que se ejecuta en una computadora en todo momento, frecuentemente denominado el 'kernel' o núcleo del sistema.

Componentes principales de un SO

Componente	Descripción y función
Kernel (Núcleo)	Componente central que gestiona directamente el hardware. Controla CPU, memoria y dispositivos de E/S.
Shell	Interfaz entre el usuario y el kernel. Puede ser gráfica (GUI) o de línea de comandos (CLI).
Sistema de archivos	Organiza y controla cómo se almacenan y recuperan los datos en los dispositivos de almacenamiento.
Gestor de procesos	Controla la creación, planificación y terminación de procesos y hilos.
Gestor de memoria	Administra la asignación y liberación de memoria RAM entre los procesos.
Controladores (Drivers)	Software especializado que permite al SO comunicarse con dispositivos de hardware específicos.
Interfaz de usuario	Entorno visual o textual mediante el cual el usuario interactúa con el sistema.

2. Las 5 Funciones Principales del Sistema Operativo

Las funciones de un sistema operativo pueden agruparse en cinco grandes categorías que cubren todos los aspectos de la gestión del sistema informático. Comprender estas funciones es fundamental para cualquier profesional del área de computación.

1. Gestión de Procesos

Un proceso es un programa en ejecución. El SO es responsable de crear, planificar, suspender, reanudar y terminar procesos. Utiliza algoritmos de planificación como Round Robin, FCFS (First Come First Served) y Shortest Job Next para asignar tiempo de CPU de manera eficiente entre los procesos activos.

- Asignación de tiempo de CPU mediante algoritmos de planificación
- Manejo de comunicación entre procesos (IPC)
- Sincronización de procesos y manejo de condiciones de carrera
- Control de estados: nuevo, listo, ejecutando, bloqueado, terminado

2. Gestión de Memoria

La memoria RAM es un recurso limitado y crítico. El SO asigna bloques de memoria a cada proceso, garantizando el aislamiento entre ellos (ningún proceso puede acceder directamente a la memoria de otro). Técnicas avanzadas como la memoria virtual permiten ejecutar programas más grandes que la RAM disponible usando el disco duro.

- Asignación y liberación dinámica de memoria
- Memoria virtual mediante paginación y segmentación
- Protección de memoria entre procesos
- Manejo del espacio de intercambio (swap)

3. Gestión de Almacenamiento

El SO proporciona una abstracción sobre el almacenamiento físico mediante el sistema de archivos. Gestiona como se organizan los datos en discos duros, SSDs y otros medios, proporcionando operaciones de lectura, escritura, búsqueda y eliminación de archivos y directorios.

- Organización jerárquica en directorios y subdirectorios
- Control de acceso y permisos (lectura, escritura, ejecución)
- Gestión de espacio libre y fragmentación
- Soporte para múltiples sistemas de archivos: NTFS, ext4, APFS, FAT32

4. Gestion de Dispositivos de E/S

El SO gestiona todos los dispositivos de entrada y salida mediante controladores (drivers). Proporciona una interfaz uniforme para que las aplicaciones puedan comunicarse con cualquier dispositivo sin necesidad de conocer sus detalles técnicos específicos.

- Manejo de interrupciones de hardware
- Buffering y spooling para optimizar el uso de dispositivos lentos
- Abstracción de dispositivos mediante API estandarizada
- Gestión de plug-and-play para dispositivos USB

5. Seguridad y Protección

El SO implementa mecanismos para proteger los datos y recursos del sistema contra accesos no autorizados, tanto de usuarios externos como de errores de programas. Esto incluye autenticación de usuarios, control de acceso a archivos y recursos, y aislamiento entre procesos.

- Autenticación mediante contraseñas, biometría o certificados
- Control de acceso basado en roles (RBAC)
- Cifrado de datos en reposo y en tránsito
- Registro de auditorías (logs del sistema)

3. Tipos de Sistemas Operativos

Los sistemas operativos se clasifican según su arquitectura, modo de procesamiento, cantidad de usuarios que pueden atender simultáneamente y el tipo de hardware sobre el que operan. Conocer estas clasificaciones permite seleccionar el SO adecuado para cada caso de uso.

Tipo de SO	Características clave	Ejemplos
Monotarea / Monousuario	Ejecuta un solo programa a la vez. Un único usuario. Simple y confiable.	MS-DOS, CP/M
Multitarea / Multiusuario	Múltiples programas y usuarios simultáneos. Compartición de recursos.	Linux, macOS, Windows Server
Tiempo real (RTOS)	Respuesta garantizada dentro de plazos estrictos. Crítico para sistemas industriales.	VxWorks, FreeRTOS, QNX
Distribuido	Múltiples computadoras coordinadas que aparecen como un sistema único.	Google Fuchsia, Plan 9
Móvil	Optimizado para pantallas táctiles, batería y conectividad móvil.	Android, iOS, HarmonyOS
Embebido	Integrado en hardware específico. Recursos muy limitados.	Linux embebido, ThreadX
En la nube / Virtualizado	Abstrae el hardware físico. Permite máquinas virtuales y contenedores.	VMware ESXi, Hyper-V, KVM

4. Gestión de Procesos y Memoria

4.1 Ciclo de vida de un proceso

Un proceso pasa por diferentes estados durante su existencia en el sistema. Comprender este ciclo es esencial para el diagnóstico de problemas de rendimiento y para el diseño de software eficiente.

Estado	Descripción
NUEVO	El proceso acaba de ser creado pero aún no ha sido admitido por el planificador.
LISTO	El proceso está esperando que se le asigne tiempo de CPU.
EJECUTANDO	Las instrucciones del proceso se están ejecutando en la CPU actualmente.

BLOQUEADO	El proceso espera un evento externo (E/S, un recurso) y no puede avanzar.
SUSPENDIDO	El proceso fue sacado de la RAM al disco por el SO para liberar memoria.
TERMINADO	El proceso completo su ejecucion o fue terminado por el SO o el usuario.

4.2 Algoritmos de planificacion de CPU

El planificador de CPU decide que proceso obtiene acceso al procesador en cada momento. La eleccion del algoritmo tiene un impacto directo en el rendimiento, la equidad y los tiempos de respuesta del sistema.

Algoritmo	Descripcion y uso tipico
FCFS (First Come First Served)	El primer proceso en llegar es el primero en ejecutarse. Simple pero puede producir efecto convoy.
Round Robin (RR)	Cada proceso recibe un quantum fijo de CPU en turnos. Equitativo. Usado en sistemas de tiempo compartido.
Shortest Job Next (SJN)	Se ejecuta primero el proceso con el menor tiempo estimado de ejecucion. Optimo pero requiere conocer los tiempos.
Priority Scheduling	Procesos con mayor prioridad se ejecutan primero. Riesgo de inanicion para procesos de baja prioridad.
Multilevel Queue	Multiples colas con diferentes prioridades y algoritmos. Combina las ventajas de los metodos anteriores.

5. Sistemas de Archivos

El sistema de archivos es la estructura lógica que el SO usa para organizar y gestionar los datos almacenados en dispositivos de almacenamiento. Define como se nombran los archivos, como se organizan en directorios y como se controla el acceso.

Sistema de archivos	Sistema operativo / Uso principal
NTFS (NT File System)	Windows. Soporta permisos avanzados, cifrado EFS, journaling y archivos de gran tamaño.
FAT32	Compatible con todos los sistemas. Limitado a archivos de max. 4 GB. Usado en memorias USB.
exFAT	Extension de FAT32. Sin límite de tamaño de archivo. Popular en memorias flash y SD.
ext4	Linux. Journaling, alta performance, soporta volúmenes hasta 1 exabyte.
APFS (Apple File System)	macOS e iOS. Optimizado para SSD, cifrado nativo, snapshots instantáneos.
ZFS	Servidores. Protección contra corrupción de datos, RAID integrado, snapshots.
BTRFS	Linux moderno. Copy-on-write, snapshots, compresión transparente.

6. Seguridad y Protección en el SO

La seguridad es una responsabilidad compartida entre el hardware, el sistema operativo y las aplicaciones. El SO implementa la primera capa de defensa con mecanismos que protegen tanto a los usuarios entre sí como al sistema completo frente a amenazas externas.

Mecanismo de seguridad	Descripción
Autenticación	Verificar la identidad del usuario mediante contraseñas, biometría (huella, rostro) o tokens.
Autorización (RBAC)	Asignar permisos específicos a roles. Un estudiante no tiene los mismos permisos que un administrador.
Sandboxing	Ejecutar aplicaciones en entornos aislados para que no puedan afectar al resto del sistema.
Cifrado	Proteger datos sensibles en disco o en transmisión con algoritmos como AES-256 o RSA.

Firewall del SO	Filtrar conexiones de red no autorizadas a nivel del sistema operativo.
Actualizaciones	Parchear vulnerabilidades conocidas. El SO notifica y gestiona las actualizaciones de seguridad.
Logs de auditoria	Registrar eventos del sistema para detectar actividades sospechosas o diagnosticar fallos.

7. Sistemas Operativos Modernos: Comparativa

En el panorama actual conviven multiples sistemas operativos, cada uno con fortalezas particulares segun el caso de uso. Esta comparativa ayuda a entender por que diferentes organizaciones e individuos eligen sistemas distintos.

Criterio	Windows 11	macOS Sequoia	Linux (Ubuntu)	Android 15
Cuota de mercado	72% escritorio	15% escritorio	4% escritorio	74% movil
Licencia	Propietaria (pago)	Propietaria (gratis con Apple)	Open Source (gratis)	Open Source / propietaria
Kernel	Windows NT	XNU (Darwin)	Linux	Linux (modificado)
Multitarea	Si	Si	Si	Si (limitado)
Seguridad	Windows Defender, TPM	Gatekeeper, T2/M-chip	SELinux, AppArmor	SELinux, sandboxing
Entorno grafico	Windows Shell	Aqua / Cocoa	GNOME, KDE, etc.	Material You
Uso principal	Oficinas, gaming, empresariales	Creativos, desarrollo Apple	Servidores, programadores	Smartphones, tablets
Fortaleza	Compatibilidad de software	Integracion hardware-software	Estabilidad y personaliz	Ecosistema de apps moviles

8. Casos Practicos de Aplicacion Docente

Los siguientes casos practicos estan disenados para conectar los conceptos teoricos del sistema operativo con situaciones reales que los docentes universitarios pueden usar en clase. Cada caso incluye el problema, el concepto de SO aplicado y posibles actividades de aprendizaje.

Caso 1: La computadora universitaria que se congela

Situacion:

En el laboratorio de computo, la computadora de un estudiante se congela completamente durante un examen. El estudiante no puede cerrar ningun programa.

Concepto de SO aplicado:

Gestion de procesos — cuando un proceso entra en un estado de bloqueo infinito (deadlock) y el planificador no puede recuperarlo, el sistema operativo puede necesitar una interrupcion forzada (Ctrl+Alt+Del en Windows, kill -9 en Linux).

Actividad sugerida:

- Practicar en clase: abrir el Administrador de Tareas, identificar procesos con alto uso de CPU o memoria, forzar el cierre de un proceso no responsivo.

Caso 2: El disco duro lleno de un servidor educativo

Situacion:

El servidor del campus que aloja la plataforma Moodle llega al 95% de capacidad de almacenamiento, lo que hace que los estudiantes no puedan entregar tareas.

Concepto de SO aplicado:

Gestion de almacenamiento — el SO monitorea el espacio disponible y puede enviar alertas. Los sistemas de archivos modernos como ZFS o BTRFS permiten ver exactamente que carpetas o archivos consumen mas espacio.

Actividad sugerida:

- Actividad: usar comandos du y df en Linux para identificar los directorios que consumen mas espacio y determinar cual puede limpiarse de forma segura.

Caso 3: Múltiples estudiantes conectados a un servidor

Situación:

Un servidor Linux del departamento necesita atender a 30 estudiantes que simultáneamente están compilando proyectos de programación.

Concepto de SO aplicado:

Multiusuario y gestión de recursos — el SO asigna prioridades y cuotas de CPU y memoria a cada usuario. El planificador Round Robin garantiza que ninguno monopolice el procesador, aunque los tiempos de respuesta aumentan con más usuarios.

Actividad sugerida:

- Discusión: cómo configurarían los permisos para que cada estudiante solo pueda acceder a su propio directorio, sin poder ver ni modificar el trabajo de sus compañeros.

Caso 4: Instalación de software en el campus

Situación:

El departamento de IT necesita instalar el mismo software en 200 computadoras del campus de forma eficiente y segura.

Concepto de SO aplicado:

Gestión de procesos y seguridad — las herramientas de despliegue centralizado (como SCCM en Windows o Ansible en Linux) usan las APIs del SO para instalar software de forma remota con privilegios de administrador.

Actividad sugerida:

- Proyecto: investigar cómo funciona una imagen de disco (imagen ISO o clon) y cómo el SO permite clonar una instalación completa para replicarla en múltiples equipos.

9. Preguntas de Reflexion y Autoevaluacion

Las siguientes preguntas estan disenadas para generar conversacion profunda sobre los conceptos de sistemas operativos. En el taller, NotebookLM podra responderlas basandose en el contenido de este documento, permitiendote comparar la respuesta de la IA con tu propio conocimiento.

Comprension basica

- 1. Explica con tus propias palabras la diferencia entre el kernel y el shell de un SO.
- 2. Un estudiante dice que Windows es 'mas rapido' que Linux. Como lo refutarías usando conceptos del SO?
- 3. Por que un sistema de archivos FAT32 no puede almacenar un archivo de video de 8 GB?
- 4. Describe el ciclo de vida completo de un proceso desde que el usuario hace doble clic en un icono hasta que el programa se cierra.

Analisis y aplicacion

- 1. Imagina que eres el planificador de CPU con solo 1 nucleo y 5 procesos esperando. Uno es un juego, otro es un antivirus, otro es una actualizacion, otro es el navegador y otro es un compilador. Cual ejecutarías primero y por que?
- 2. Un sistema operativo para una estacion espacial necesita procesar datos de sensores cada 50 milisegundos exactos. Cual tipo de SO elegirías y que funcion del SO es mas critica en este caso?
- 3. Compara la seguridad de Windows con la de Linux usando al menos 3 mecanismos de seguridad especificos de cada uno.
- 4. Como afecta a la experiencia del usuario cuando el SO tiene que usar memoria de intercambio (swap) intensamente?

Sintesis y evaluacion

- 1. Diseña la arquitectura de un SO minimo para una tableta educativa escolar. Que funciones incluirías obligatoriamente y cuales omitirías para reducir el consumo de bateria?
- 2. Si tuvieras que migrar una universidad de Windows a Linux, cuales son los 3 mayores desafios desde el punto de vista del sistema operativo y como los resolverías?
- 3. Los contenedores Docker comparten el kernel del SO anfitrión en lugar de virtualizarlo completamente. Explica que ventajas y que riesgos de seguridad implica esto.

Nota para el taller

- Sube este PDF a NotebookLM como fuente principal del notebook.

- Luego prueba haciendo preguntas del nivel 'Análisis' al chat de NotebookLM y compara las respuestas con las tuyas.
- Usa las preguntas de 'Síntesis' como prompt para generar tu Slide Deck final.