

Diagnóstico

SISTEMAS CONCURRENTES Y DISTRIBUIDOS

Ignacio David Vázquez Pérez

218292866

Tarea 1. Cuestionario de Repaso de Sistemas Operativos

Objetivo

El alumno conocerá la relación entre un sistema operativo centralizado y un sistema operativo distribuido así como las características básicas de los sistemas distribuidos en comparación con los sistemas operativos personales.

Introducción:

Los sistemas operativos son una parte esencial de la informática moderna y desempeñan un papel fundamental en la gestión de recursos, la ejecución de programas y la interacción con las computadoras y dispositivos electrónicos que utilizamos en nuestra vida diaria. Si bien a menudo pasan desapercibidos para el usuario promedio, los sistemas operativos son el "cerebro" detrás de cada dispositivo digital, permitiendo que funcione de manera coherente y eficiente.

En esta tarea, exploraremos en detalle qué son los sistemas operativos, cuál es su importancia en la informática actual y cómo desempeñan un papel crucial en la administración de recursos, la seguridad y la interacción usuario-máquina. También analizaremos las diferentes categorías de sistemas operativos y cómo han evolucionado a lo largo del tiempo para adaptarse a las cambiantes demandas de la tecnología.

Actividades a realizar

1. Historia del sistema operativo (acontecimientos mas importantes)

- Década de 1940: Aparece la primera generación de computadoras en el mundo. Los programas necesitaban la más detallada especificación del hardware para ejecutarse correctamente.
- 1956: General Motors desarrolla lo que es hoy considerado el primer sistema operativo, el GM-NAA I/O, para su IBM 7041.
- Década de 1950 (Sistema Batch): A principios de los años 1950, los sistemas operativos hacen una aparición discreta y bastante simple, con conceptos tales como el monitor residente y el almacenamiento temporal.

- Década de 1960: En los años 60 se produjeron cambios notorios en varios campos de la informática, con la aparición del circuito integrado¹. Se introdujo el sistema operativo multiprogramación, donde la memoria principal alberga a más de un programa de usuario¹.
- 1963: Un equipo del Massachusetts Institute of Technology dirigido por Fernando Corbato desarrolló el sistema CTSS (Compatible Time Sharing System), que fue el primer sistema operativo práctico que permitió a varios usuarios ejecutar varios programas diferentes desde terminales.
- 1969: Los programadores Ken Thompson y Dennis Ritchie comenzaron a desarrollar su propia versión reducida de Multics, a la que llamaron Unix².
- 1974: Los sistemas operativos modernos comenzaron en 1974 para computadoras personales con Microsoft Windows y Mac OS X que se actualizan y utilizan continuamente en la actualidad.

2. ¿Qué es un sistema operativo?

Un sistema operativo es un programa o conjunto de programas que actúa como intermediario entre los usuarios y el hardware de un sistema informático. Su objetivo principal es proporcionar un entorno en el que el usuario pueda ejecutar aplicaciones de manera conveniente y eficiente. Además, los sistemas operativos aseguran que las aplicaciones diferentes que se ejecutan al mismo tiempo no interfieran entre sí.

3. ¿Cuáles son las funciones y objetivos de un sistema operativo?

- **Gestión de recursos:** El sistema operativo gestiona los recursos de hardware y software del sistema informático, incluyendo la CPU, memoria, almacenamiento y dispositivos de entrada/salida.
- **Abstracción de hardware:** Proporciona una interfaz uniforme y simplificada para interactuar con el hardware, ocultando la complejidad y las diferencias entre distintos tipos de hardware.
- **Ejecución de programas:** El sistema operativo es responsable de cargar programas en la memoria y ejecutarlos. También proporciona servicios como acceso a archivos y manejo de errores durante la ejecución del programa.
- **Gestión de archivos:** El sistema operativo gestiona el sistema de archivos, incluyendo la creación, borrado, lectura y escritura de archivos.
- **Seguridad y protección:** El sistema operativo proporciona mecanismos para proteger los datos y recursos del sistema contra accesos no autorizados o malintencionados.
- **Comunicación entre procesos:** El sistema operativo proporciona mecanismos para la comunicación e interacción entre procesos.
- **Gestión de usuarios:** El sistema operativo gestiona las cuentas de usuario y sus privilegios.

El objetivo principal de un sistema operativo es proporcionar un entorno en el que el usuario pueda ejecutar aplicaciones de manera eficiente y conveniente.

4. Tipos de servicios que brinda un sistema operativo.

1. **Gestión de memoria:** El sistema operativo es responsable de administrar la memoria del computador, lo que incluye conocer qué zonas de memoria están libres y cuáles están ocupadas, decidir qué procesos cargar cuando haya memoria libre, y reservar y liberar zonas de memoria según se solicite.
2. **Gestión de procesos:** Un proceso es un programa en ejecución que necesita tiempo de CPU, una porción de memoria, archivos, E/S y demás recursos. El sistema operativo es responsable de la creación y eliminación de procesos, la planificación de procesos, la sincronización entre procesos y la comunicación entre procesos.
3. **Gestión de la E/S:** El sistema operativo proporciona una interfaz uniforme para el acceso a los dispositivos, proporciona manejadores para los dispositivos concretos y trata automáticamente los errores más típicos.
4. **Sistema de archivos:** El sistema operativo es responsable de la manipulación de archivos y directorios, ubicar los archivos y directorios en los dispositivos de almacenamiento secundario y automatizar ciertos servicios como copia de seguridad, versiones, etc.
5. **Redes:** En un sistema distribuido, el sistema operativo proporciona primitivas para conectarse con equipos remotos y acceder de forma controlada a sus recursos.
6. **Sistema de protección:** La protección abarca los mecanismos destinados a controlar el acceso de los usuarios a los recursos, de acuerdo con los privilegios que se definan.
7. **Ejecución de programas:** El sistema operativo es responsable de gestionar la memoria de acceso aleatorio y ejecutar las aplicaciones, designando los recursos necesarios.
8. **Administración del CPU:** El sistema operativo administra al CPU gracias a un algoritmo de programación.
9. **Dirección de entradas y salidas de datos:** A través de drivers, el sistema operativo direcciona las entradas y salidas de datos por medio de los periféricos de entrada o salida.
10. **Administración de la información:** Para el buen funcionamiento del PC, el sistema operativo administra la información.

5. Características deseables de un sistema operativo.

- Eficiencia: Un buen SO debe ser capaz de gestionar los recursos del sistema de manera eficiente, minimizando el tiempo de inactividad y maximizando el rendimiento.
- Facilidad de uso: Debe ser fácil de usar, con una interfaz de usuario intuitiva y documentación clara.
- Portabilidad: Idealmente, un SO debe ser portátil, lo que significa que puede ser fácilmente transferido de una máquina a otra.

6. ¿Cuál es el modo usuario y el modo Kernel o supervisor?

En un sistema operativo, el modo usuario y el modo kernel (también conocido como modo supervisor) son dos niveles distintos de privilegios de ejecución.

- Modo Usuario: En este modo, las aplicaciones se ejecutan con privilegios limitados para proteger el sistema operativo y otros recursos del sistema. Las aplicaciones no pueden acceder

directamente a los recursos de hardware o a áreas de memoria protegidas. Si una aplicación necesita acceder a estos recursos, debe hacer una solicitud al sistema operativo a través de una interfaz de programación de aplicaciones (API).

- **Modo Kernel o Supervisor:** Este es el modo de mayor privilegio en el que se ejecuta el núcleo del sistema operativo. En este modo, el código tiene acceso directo a todos los recursos de hardware y puede ejecutar cualquier instrucción de la máquina. El sistema operativo utiliza este modo para realizar tareas críticas del sistema, como la gestión de la memoria, la programación de tareas y la entrada/salida de dispositivos.

7. Definir procesamiento en serie, procesamiento por lotes, multiprogramación, monoprogramación, sistema operativo paralelo, sistema operativo distribuido.

- **Procesamiento en serie:** Es un tipo de procesamiento de datos donde las tareas se ejecutan una después de la otra. Cada tarea debe completarse antes de que pueda comenzar la siguiente.
- **Procesamiento por lotes:** En este tipo de procesamiento, las tareas se agrupan en lotes y se procesan juntas. Esto puede ser más eficiente que el procesamiento en serie porque el sistema operativo puede optimizar el procesamiento de las tareas en función de su similitud.
- **Multiprogramación:** Es una técnica que permite a varias tareas compartir el tiempo de procesamiento del sistema. El sistema operativo cambia rápidamente entre tareas, dando la apariencia de que todas las tareas se están ejecutando simultáneamente.
- **Monoprogramación:** Es un enfoque de procesamiento en el que solo se ejecuta una tarea a la vez. Una vez que se completa una tarea, se inicia la siguiente.
- **Sistema operativo paralelo:** Es un tipo de sistema operativo que permite la ejecución simultánea de múltiples tareas en múltiples procesadores. Esto puede mejorar significativamente el rendimiento y la eficiencia del sistema.
- **Sistema operativo distribuido:** Es un sistema operativo que se ejecuta en múltiples máquinas y permite que las tareas se distribuyan entre ellas. Esto puede proporcionar una alta disponibilidad, escalabilidad y tolerancia a fallos.

8. Definir estructura del sistema operativo monolítico, por capas, micro kernell, cliente servidor y máquina virtual.

1. Sistema Operativo Monolítico:

- **Definición:** En un sistema operativo monolítico, todo el núcleo del sistema operativo y los servicios se ejecutan en un solo espacio de direcciones y como una única imagen de código en el espacio del kernel. Esto significa que todas las funciones del sistema operativo, como la gestión de procesos, el sistema de archivos y los controladores de dispositivos, se ejecutan en un solo programa.
- **Características:** Este enfoque es simple y eficiente en términos de rendimiento, pero puede ser menos flexible y más propenso a fallos en comparación con otras arquitecturas.

2. Sistema Operativo por Capas:

- **Definición:** En un sistema operativo por capas, el sistema operativo se divide en capas distintas, donde cada capa proporciona un conjunto específico de servicios. Cada capa se comunica solo con las capas adyacentes, y las capas más altas utilizan los servicios de las capas más bajas.
- **Características:** Esta estructura permite una mayor modularidad y mantenimiento, ya que las capas pueden actualizarse o reemplazarse sin afectar a las demás. Sin embargo, puede introducir cierta sobrecarga debido a la comunicación entre capas.

3. Microkernel:

- **Definición:** En un sistema operativo con microkernel, el núcleo del sistema operativo contiene solo las funciones esenciales, como la gestión de procesos y la comunicación interprocesos. Los servicios del sistema operativo se implementan como procesos independientes que se ejecutan en el espacio de usuario.
- **Características:** Esta arquitectura es altamente modular y favorece la estabilidad y la flexibilidad. Los servicios pueden agregarse o eliminarse fácilmente. Sin embargo, puede tener un impacto en el rendimiento debido a la comunicación entre procesos.

4. Cliente-Servidor:

- **Definición:** En un sistema operativo cliente-servidor, el sistema operativo se organiza en torno al principio de que los servicios del sistema operativo son proporcionados por servidores dedicados que atienden a solicitudes de clientes. Los clientes pueden ser procesos o aplicaciones que solicitan servicios específicos.
- **Características:** Esta estructura permite una gran flexibilidad y escalabilidad, ya que los servidores pueden ejecutarse en máquinas separadas y servir a múltiples clientes. Se utiliza comúnmente en sistemas distribuidos.

5. Máquina Virtual:

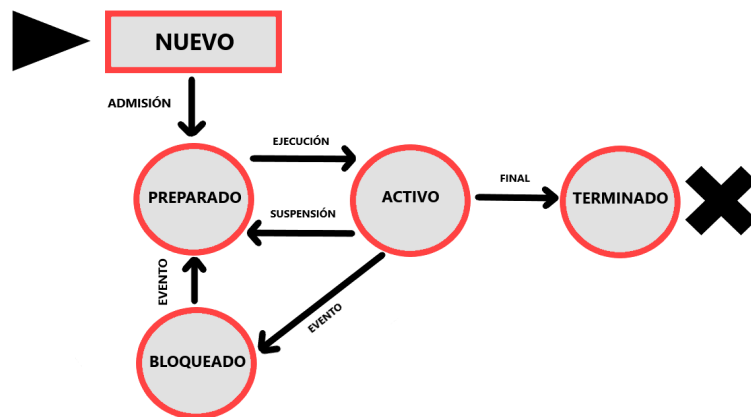
- **Definición:** En una máquina virtual, el sistema operativo se ejecuta como una capa de software sobre el hardware físico. Cada instancia del sistema operativo virtualizado, conocida como máquina virtual (VM), se comporta como una máquina independiente con su propio sistema operativo.
- **Características:** Esta arquitectura permite la consolidación de múltiples sistemas operativos en una sola máquina física, lo que es útil para la virtualización de servidores y pruebas de software. Cada VM se ejecuta de forma aislada, lo que aumenta la seguridad y la flexibilidad.

9. ¿Qué es un proceso?

En un sistema operativo, un proceso es una entidad fundamental que representa la ejecución de un programa o una tarea en una computadora. Un proceso es una unidad de trabajo independiente que consta de varios componentes, incluyendo el código del programa, datos, estado de ejecución y recursos del sistema asignados.

10. Definir los estados de un proceso y dibujar el diagrama de 5 estados

1. **Nuevo (New):** En este estado, el proceso se ha creado, pero aún no se ha asignado tiempo de CPU ni recursos. Está en espera de ser admitido en la cola de procesos listos para su ejecución.
2. **Listo (Ready):** Un proceso en el estado "Listo" está preparado para ejecutarse y ha sido admitido en la cola de procesos listos. Espera su turno para obtener tiempo de CPU. Los procesos en este estado pueden cambiar rápidamente a "En Ejecución" cuando se les asigna la CPU.
3. **En Ejecución (Running):** En este estado, el proceso está ejecutándose activamente y utiliza la CPU. Solo puede haber un proceso en ejecución en un núcleo de CPU en un momento dado (en sistemas monoprocesador), o varios procesos en sistemas multiprocesador.
4. **En Espera (Blocked o Waiting):** Cuando un proceso se encuentra en este estado, significa que está esperando que ocurra un evento o una condición antes de poder continuar su ejecución. Esto puede deberse a una solicitud de entrada/salida, como la lectura de datos desde un disco o la espera de entrada del usuario.
5. **Terminado (Terminated):** El estado final de un proceso. Indica que el proceso ha completado su ejecución y ha liberado todos los recursos que tenía asignados. En este punto, el proceso se elimina de la lista de procesos activos.



11. ¿Cuáles son las clases de interrupciones?

Las interrupciones en un sistema operativo se pueden clasificar en tres categorías principales:

- **Interrupciones de Hardware:** Son generadas por dispositivos de hardware, como temporizadores, operaciones de entrada/salida, fallos de hardware y controladores de dispositivos.
- **Interrupciones de Software:** Son generadas por programas y se utilizan para solicitar servicios del sistema operativo (llamadas al sistema) o manejar errores de programación y situaciones excepcionales (excepciones de software).
- **Interrupciones de Sincronización:** Se utilizan para coordinar la ejecución de múltiples procesos, permitiendo que los procesos esperen o se despierten cuando se cumplen ciertas condiciones.

Cada tipo de interrupción tiene un propósito específico y ayuda al sistema operativo a gestionar eficientemente los recursos y las tareas del sistema.

12. Tipos de planificadores de procesos

- **Planificador a Corto Plazo (Short-Term Scheduler):** También conocido como planificador de CPU, se encarga de seleccionar qué proceso en la cola de procesos listos obtendrá acceso a la CPU de manera inmediata. Su objetivo principal es maximizar la utilización de la CPU y minimizar el tiempo de espera de los procesos.
- **Planificador a Largo Plazo (Long-Term Scheduler):** Este planificador decide cuándo admitir nuevos procesos desde la cola de procesos en espera al sistema. Su objetivo es mantener un equilibrio entre la carga del sistema y la capacidad de recursos, evitando la sobrecarga de la CPU.
- **Planificador a Mediano Plazo (Medium-Term Scheduler):** Su función es la gestión de procesos que están en espera y que no están listos para ser ejecutados inmediatamente. Puede decidir mover procesos de la memoria principal a almacenamiento secundario (swapping) para liberar espacio en la memoria RAM.
- **Planificador de Tiempo Real (Real-Time Scheduler):** Utilizado en sistemas operativos de tiempo real, asigna tiempo de CPU a procesos en función de sus prioridades y restricciones temporales estrictas. Garantiza que los procesos críticos cumplan sus plazos de ejecución.
- **Planificador de Lote (Batch Scheduler):** Se utiliza en sistemas donde los usuarios envían trabajos en lotes y no interactúan directamente con la computadora. El planificador de lotes organiza y ejecuta los trabajos de manera secuencial o en paralelo.
- **Planificador de Prioridad (Priority Scheduler):** Asigna tiempo de CPU a los procesos en función de sus niveles de prioridad. Los procesos con prioridades más altas obtienen acceso a la CPU antes que los de prioridad más baja. Este tipo de planificador se utiliza en sistemas multiprogramados donde es importante la asignación de recursos en función de la importancia de los procesos.
- **Planificador de Retroalimentación (Feedback Scheduler):** Utiliza múltiples colas de procesos con diferentes prioridades. Los procesos pueden moverse entre colas en función de su comportamiento y tiempo de ejecución. Este enfoque permite adaptarse a la variabilidad de los procesos y promueve la equidad en la asignación de CPU.

13. ¿Cuáles son los algoritmos de planificación de procesos? Incluir breve descripción

- **FIFO (First-Come, First-Served):** Este es uno de los algoritmos más simples, donde los procesos se ejecutan en el orden en que llegan a la cola de listos. El proceso que llega primero es el primero en recibir tiempo de CPU. Sin embargo, puede llevar a problemas de inanición para procesos de baja prioridad y no optimiza la utilización de la CPU.
- **SJF (Shortest Job First):** En este algoritmo, se selecciona el proceso con la duración más corta (tiempo de CPU requerido) en la cola de listos. Esto minimiza el tiempo de espera promedio, pero puede llevar a la inanición de procesos largos si llegan continuamente procesos cortos.
- **Prioridad:** Los procesos se ejecutan en función de su prioridad, donde el proceso con la prioridad más alta obtiene tiempo de CPU primero. Puede ser preemptivo o no preemptivo. El problema principal es la posible inanición de procesos de baja prioridad si los procesos de alta prioridad llegan continuamente.
- **Round Robin:** En este algoritmo, los procesos se ejecutan en un orden circular y se les asigna un tiempo de ejecución fijo (quantum). Cuando un proceso agota su quantum, se coloca nuevamente

al final de la cola. Es justo, pero puede tener un alto costo de cambio de contexto si el quantum es demasiado pequeño.

- **Planificación por Retroalimentación (Feedback Scheduling):** Este algoritmo permite que los procesos cambien de una cola a otra en función de su comportamiento y tiempo de ejecución. Los procesos que necesitan más tiempo pueden moverse a colas de menor prioridad, y los procesos de alta prioridad pueden avanzar a colas de mayor prioridad.
- **Multinivel (Multilevel Queue):** Divide la cola de procesos en varias colas con prioridades diferentes. Los procesos se asignan a colas en función de su prioridad y tipo. Por ejemplo, pueden haber colas para procesos de usuario y para procesos del sistema. Cada cola puede utilizar un algoritmo de planificación diferente.
- **Lotería (Lottery Scheduling):** Asigna a cada proceso un número de lotería y, en cada asignación de tiempo de CPU, se elige aleatoriamente un boleto ganador. El proceso asociado con el boleto ganador obtiene la CPU. Este enfoque es útil para sistemas con múltiples usuarios y procesos interactivos.
- **Planificación en Tiempo Real (Real-Time Scheduling):** Se utiliza en sistemas de tiempo real, donde se establecen restricciones estrictas en los plazos de ejecución de los procesos. Los procesos se programan de acuerdo con sus prioridades y requisitos temporales.

14. ¿Qué es un hilo?

Un hilo, en el contexto de la programación y los sistemas operativos, es la unidad más pequeña de ejecución dentro de un proceso. Un hilo comparte el mismo espacio de memoria y recursos del proceso al que pertenece, lo que significa que múltiples hilos pueden ejecutarse simultáneamente en un proceso y compartir datos y código. Los hilos permiten la ejecución concurrente y paralela de tareas dentro de un proceso, lo que puede mejorar la eficiencia y la capacidad de respuesta de las aplicaciones. Los hilos pueden utilizarse para dividir una aplicación en tareas más pequeñas y realizar operaciones concurrentes, como procesamiento en segundo plano, sin la necesidad de crear procesos separados.

15. ¿Qué significa exclusión mutua?

La exclusión mutua, en informática y programación concurrente, es un concepto que se refiere a la condición en la que solo un proceso o hilo tiene acceso a un recurso compartido o una sección crítica en un momento dado. El objetivo de la exclusión mutua es prevenir conflictos y garantizar que múltiples procesos o hilos no accedan simultáneamente a un recurso compartido, lo que podría causar condiciones de carrera y resultados no deterministas en un programa.

Para lograr la exclusión mutua, se utilizan mecanismos como semáforos, mutex (mutual exclusion locks), cerrojos, o cualquier otro método de sincronización que permita a los procesos o hilos adquirir y liberar el acceso al recurso de manera controlada. La exclusión mutua es fundamental para garantizar la coherencia y la integridad de los datos en sistemas concurrentes y evitar problemas de concurrencia.

16. ¿Cómo se da la comunicación entre procesos?

- Variables compartidas: Compartir datos en la memoria, pero requerirá sincronización para evitar problemas.
- Pipes: Para comunicación unidireccional entre procesos.
- Sockets: Usados para comunicación en redes, locales o remotas.
- Colas de mensajes: Para enviar y recibir mensajes de manera asíncrona.
- Memoria compartida: Compartir una región de memoria para datos, pero con necesidad de sincronización.
- Comunicación por señales: Envío de señales para notificar eventos o excepciones.
- RPC (Remote Procedure Call): Llamar funciones o procedimientos en otros procesos.
- Corrutinas o hilos: Comunicación dentro del mismo proceso utilizando variables compartidas o estructuras de datos.

17. Dibujar esquema de jerarquía de memoria



18. ¿Cuales son las operaciones sobre archivos?

- Creación: Crear un nuevo archivo en el sistema de archivos.
- Apertura: Abrir un archivo existente para lectura, escritura o ambas.
- Lectura: Leer datos desde un archivo abierto.
- Escritura: Escribir datos en un archivo abierto.
- Cierre: Cerrar un archivo después de realizar operaciones de lectura o escritura.
- Borrado: Eliminar un archivo del sistema de archivos.
- Renombrado: Cambiar el nombre de un archivo existente.
- Desplazamiento: Mover el puntero de lectura/escritura a una posición específica dentro de un archivo.
- Truncamiento: Reducir el tamaño de un archivo, eliminando parte de su contenido.
- Duplicación: Crear una copia de un archivo existente.
- Verificación de existencia: Comprobar si un archivo existe en el sistema de archivos.

- Cambio de atributos y permisos: Modificar metadatos como permisos de acceso, propietario y fechas de creación/modificación.

19. ¿Que es Cifrado? Incluir un ejemplo

El cifrado es un proceso mediante el cual la información se convierte en un formato ilegible o incomprensible para cualquiera que no tenga la clave o el algoritmo de descifrado adecuado. El propósito principal del cifrado es proteger la confidencialidad de los datos y garantizar que solo las partes autorizadas puedan acceder y comprender la información.

Un ejemplo de encriptación breve podría ser el siguiente:

Mensaje original: "SECRETO"

Clave de encriptación: "12345"

Aplicando un cifrado simple de sustitución, podríamos desplazar cada letra del mensaje original cinco posiciones hacia adelante en el alfabeto:

Mensaje encriptado: "XJHWJYT"

20. Generalidades de seguridad del sistema operativo

- Control de Acceso: Los sistemas operativos tienen políticas de control de acceso que determinan quién puede acceder a qué recursos y con qué permisos. Esto ayuda a evitar accesos no autorizados a archivos y configuraciones del sistema.
- Autenticación y Autorización: Se utiliza autenticación para verificar la identidad de los usuarios y luego se aplica la autorización para determinar qué acciones pueden realizar. Esto garantiza que solo los usuarios autorizados tengan acceso a los recursos.
- Gestión de Usuarios y Contraseñas: Los sistemas operativos permiten la creación y gestión de cuentas de usuario, así como la configuración de contraseñas seguras para proteger las cuentas contra ataques de fuerza bruta.
- Actualizaciones y Parches: Mantener el sistema operativo actualizado con las últimas actualizaciones y parches de seguridad es esencial para corregir vulnerabilidades conocidas y mejorar la seguridad.
- Firewalls: Los sistemas operativos pueden incluir firewalls integrados para controlar el tráfico de red entrante y saliente, protegiendo así el sistema contra ataques en línea.
- Auditoría y Registro: Los sistemas operativos pueden registrar eventos importantes, como accesos a archivos o cambios en la configuración, para rastrear actividades sospechosas y ayudar en investigaciones de seguridad.
- Gestión de Procesos y Permisos: Los sistemas operativos tienen mecanismos para limitar los permisos y capacidades de los procesos en ejecución, asegurando que no realicen acciones dañinas.
- Cifrado de Datos: Los sistemas operativos pueden proporcionar herramientas y bibliotecas para cifrar datos confidenciales, lo que ayuda a proteger la información almacenada y transmitida.

- Políticas de Contraseñas y Bloqueo de Cuentas: Establecer políticas de contraseñas fuertes y activar bloqueos de cuentas temporales después de múltiples intentos fallidos de inicio de sesión refuerza la seguridad.
- Seguridad en la Red: Los sistemas operativos también pueden incorporar medidas de seguridad de red, como la autenticación en redes inalámbricas y la configuración de VPN (Virtual Private Network) para conexiones seguras.
- Eliminación Segura de Datos: Proporcionan métodos para eliminar de manera segura los datos sensibles, evitando la recuperación no autorizada de información.

Conclusión

Los sistemas operativos son el núcleo de cualquier dispositivo informático y desempeñan un papel crítico en la gestión de recursos, la ejecución de aplicaciones y la facilitación de la interacción entre usuarios y hardware. Su importancia radica en que permiten la operación eficiente y segura de computadoras y dispositivos móviles, facilitando la ejecución de tareas complejas y proporcionando una interfaz de usuario intuitiva. Los sistemas operativos son la base de la informática moderna y siguen siendo esenciales en la actualidad para impulsar la tecnología y la innovación en una variedad de campos, desde la informática personal hasta la nube y la inteligencia artificial.

Bibliografía

- Tanenbaum Andrew. (1995). Sistemas Operativos Distribuidos. España. Prentice-Hall Hisp.
- McIver Ann. (2011). Sistemas Operativos. México. Cengage Learning.
- Tanenbaum, A., & Van Steen M. (2008). Sistemas Distribuidos, Principios y Paradigmas. (Segunda ed.). Prentice Hall.
- Tanenbaum, A. (2011). Redes de Computadoras. (Quinta ed.). Prentice Hall.
- Elmasri, R., Gil Carrick, A., & Levine, D. (2010). Sistemas Operativos, Un enfoque en espiral. McGraw-Hill.
- Estudiando. (2020, 20 de septiembre). Historia de los sistemas operativos: cronología y evolución. [Estudiando](#)
- Universitat Politècnica de Catalunya. (n.d.). Historia de los sistemas operativos. [UPC Universitat Politècnica de Catalunya](#)
- ComputerWorld. (2003, 11 de julio). Historia de los sistemas operativos. [ComputerWorld](#)
- Tipos de sistemas operativos y sus características. <https://www.universitatcarlemany.com/actualidad/blog/tipos-de-sistemas-operativos/>
- Sistema Operativo - Concepto, usos, tipos, funciones y ejemplos. <https://concepto.de/sistema-operativo/>
- Estructura y servicios del sistema operativo. <http://sopa.dis.ulpgc.es/fso/teoria/pdf/Old/so-01-3-Estructura del so-6x1.pdf>