

Desarrollo Tarea #1 – SO

Respuestas Teóricas:

1. La evolución de los sistemas informáticos ha sido un proceso continuo y acelerado, marcado por una serie de hitos importantes que han permitido mejorar su rendimiento, eficiencia y funcionalidad.

Según los libros *Operating System Concepts 9th Silberschatz and Galvin* y *Sistemas Operativos Modernos de Tanenbaum*, los hitos más importantes de la evolución de los sistemas informáticos son los siguientes:

Primera generación (1945-1955):

- 1945: Construcción del ENIAC, el primer ordenador digital electrónico.
- 1951: Construcción del UNIVAC I, el primer ordenador comercial.
- 1955: Desarrollo de la programación estructurada.

Segunda generación (1955-1965):

- 1956: Introducción de los transistores en los ordenadores.
- 1957: Desarrollo del lenguaje COBOL.
- 1959: Desarrollo del lenguaje FORTRAN.

Tercera generación (1965-1980):

- 1965: Introducción de los circuitos integrados en los ordenadores.
- 1969: Desarrollo del lenguaje Pascal.
- 1969: Desarrollo de la ARPANET, la primera red de ordenadores.

Cuarta generación (1980-1990):

- 1971: Introducción del microprocesador Intel 4004.
- 1975: Desarrollo del lenguaje C.
- 1981: Introducción del IBM PC, el primer ordenador personal.

Quinta generación (1990-presente):

- 1991: Introducción del sistema operativo Windows 3.1.
- 1993: Introducción del sistema operativo Linux.
- 1995: Introducción del sistema operativo Windows 95.
- 2001: Introducción del sistema operativo Windows XP.
- 2007: Introducción del sistema operativo Windows Vista.
- 2009: Introducción del sistema operativo Mac OS X Snow Leopard.
- 2010: Introducción del sistema operativo Windows 7.
- 2012: Introducción del sistema operativo Mac OS X Mountain Lion.
- 2013: Introducción del sistema operativo Windows 8.
- 2014: Introducción del sistema operativo Windows Server 2012.
- 2015: Introducción del sistema operativo macOS Sierra.
- 2016: Introducción del sistema operativo Windows 10.
- 2017: Introducción del sistema operativo macOS High Sierra.
- 2018: Introducción del sistema operativo Windows Server 2019.
- 2019: Introducción del sistema operativo macOS Catalina.
- 2020: Introducción del sistema operativo Windows 11.
- 2021: Introducción del sistema operativo macOS Monterey.
- 2022: Introducción del sistema operativo Windows 11 Sun Valley 2.

Estos hitos han permitido desarrollar sistemas informáticos más potentes, eficientes y accesibles. Han impulsado la adopción de ordenadores en los hogares, las empresas y las instituciones educativas. Han dado lugar al desarrollo de nuevas tecnologías, como la Internet, la computación en la nube y la inteligencia artificial.

Algunos de los hitos más recientes de la evolución de los sistemas informáticos incluyen:

- El desarrollo de los sistemas operativos multitarea, que permiten que varios programas se ejecuten al mismo tiempo.
- El desarrollo de los sistemas operativos multiusuario, que permiten que varios usuarios accedan a un mismo ordenador de forma simultánea.
- El desarrollo de los sistemas operativos distribuidos, que permiten que varios ordenadores se conecten entre sí para compartir recursos.
- El desarrollo de los sistemas operativos en la nube, que permiten que los usuarios accedan a los recursos de un ordenador remoto a través de Internet.
- El desarrollo de los sistemas operativos móviles, que están diseñados para funcionar en dispositivos portátiles como teléfonos inteligentes y tabletas.

La evolución de los sistemas informáticos continuará en el futuro, a medida que se desarrollen nuevas tecnologías y se identifiquen nuevas necesidades.

2. Según Silberschatz, Galvin, un sistema informático puede dividirse en cuatro componentes:

1. Hardware: Este componente proporciona los recursos informáticos básicos, como la CPU, la memoria y los dispositivos de I/O. El hardware es la parte física del sistema informático que se puede tocar y ver. Incluye todos los componentes electrónicos que se utilizan para realizar las funciones básicas del computador.

2. Sistema operativo: Este componente controla y coordina el uso del hardware entre varias aplicaciones y usuarios. El sistema operativo es el software que gestiona el hardware del computador y proporciona servicios a los programas de aplicación. Actúa como intermediario entre el usuario y el hardware, facilitando al usuario la interacción con el ordenador.

3. Programas de aplicación: Este componente define las formas en que se utilizan los recursos del sistema para resolver los problemas informáticos de los usuarios, como procesadores de texto, compiladores, navegadores web y sistemas de bases de datos. Los programas de aplicación son software diseñado para realizar tareas específicas para el usuario. Se instalan sobre el sistema operativo y utilizan sus servicios para interactuar con el hardware.

4. Usuarios: Este componente consta de personas, máquinas y otros equipos. Los usuarios son las personas o máquinas que interactúan con el sistema informático. Pueden ser usuarios finales que utilizan programas de aplicación para realizar tareas u otras máquinas que se comunican con el sistema a través de una red.

3. Un núcleo monolítico es un tipo de núcleo de sistema operativo en el que todo el sistema operativo se ejecuta en modo kernel como un único programa en un espacio de direcciones único. Los servicios del usuario y del kernel se implementan en el mismo espacio de direcciones. Un ejemplo de sistemas operativos basados en núcleos monolíticos son Unix y Linux.

Por otro lado, un micronúcleo es un tipo de núcleo de sistema operativo en el que solo se incluyen las funcionalidades básicas, mientras que el resto de los servicios se ejecutan en el espacio de usuario. En un micronúcleo, los servicios del usuario y del kernel se mantienen en espacios de direcciones separados. Los servidores invocan “servicios” entre sí enviando mensajes a través de IPC (Comunicación entre procesos). Esta separación tiene la ventaja de que, si un servidor falla, otros servidores aún pueden funcionar eficientemente. Un ejemplo de sistemas operativos basados en micronúcleos son Mac OS X y Windows NT.

4. Un sistema operativo es un programa de software que gestiona los recursos de un ordenador, como la memoria, el procesador, los dispositivos de entrada/salida y el almacenamiento. Su objetivo es proporcionar a los usuarios y a las aplicaciones un entorno para ejecutarse y realizar sus tareas.

Desde una perspectiva funcional, un sistema operativo proporciona las siguientes funciones:

- Gestión de procesos: El sistema operativo coordina la ejecución de los procesos de forma que se ejecuten de forma correcta y eficiente.
- Gestión de memoria: El sistema operativo asigna memoria a las aplicaciones de forma que no se solapen entre sí y que ninguna aplicación utilice más memoria de la que necesita.
- Gestión de dispositivos: El sistema operativo proporciona una interfaz entre las aplicaciones y los dispositivos físicos, como el teclado, el ratón, la pantalla y la impresora.
- Gestión de archivos: El sistema operativo proporciona un sistema de archivos para almacenar y organizar los datos de las aplicaciones.
- Gestión de seguridad: El sistema operativo protege los datos de las aplicaciones y los usuarios de accesos no autorizados.

Desde una perspectiva de servicios, un sistema operativo proporciona los siguientes servicios:

- Servicios de entrada/salida: El sistema operativo proporciona un conjunto de rutinas para acceder a los dispositivos de entrada/salida, como el teclado, el ratón, la pantalla y la impresora.
- Servicios de gestión de memoria: El sistema operativo proporciona un conjunto de rutinas para asignar memoria a las aplicaciones y para gestionar la memoria que ya está en uso.
- Servicios de gestión de procesos: El sistema operativo proporciona un conjunto de rutinas para crear, ejecutar, suspender y finalizar procesos.
- Servicios de gestión de archivos: El sistema operativo proporciona un conjunto de rutinas para crear, abrir, leer, escribir y cerrar archivos.
- Servicios de gestión de seguridad: El sistema operativo proporciona un conjunto de rutinas para proteger los datos de las aplicaciones y de los usuarios de accesos no autorizados.

Las dos perspectivas son complementarias y proporcionan una visión completa de lo que es un sistema operativo.

Ejemplos de sistemas operativos

Los sistemas operativos más comunes son:

- Windows: El sistema operativo más popular del mundo, desarrollado por Microsoft.
- macOS: El sistema operativo de Apple, diseñado para ordenadores Mac.
- Linux: Un sistema operativo de código abierto, disponible para una amplia gama de dispositivos.
- Android: Un sistema operativo móvil, desarrollado por Google.
- iOS: El sistema operativo móvil de Apple, disponible para los iPhone y iPad.

Estos sistemas operativos proporcionan a los usuarios una interfaz gráfica de usuario (GUI) que les permite interactuar con el ordenador de forma intuitiva. También proporcionan una amplia gama de aplicaciones y herramientas que permiten a los usuarios realizar sus tareas diarias.

5. Para que sirve los llamados al sistema

Las llamadas al sistema son una forma de que las aplicaciones de usuario interactúen con el núcleo del sistema operativo. Permiten a las aplicaciones realizar tareas que requieren acceso al hardware o a otros recursos del sistema, como acceder a un archivo, crear un proceso o enviar una señal a otro proceso.

Las llamadas al sistema son esenciales para el funcionamiento de cualquier sistema operativo. Sin ellas, las aplicaciones no podrían realizar tareas básicas como leer y escribir datos en la memoria, acceder a dispositivos de entrada/salida o comunicarse entre sí.

Algunos ejemplos de llamadas al sistema incluyen:

- `write()`: Permite a una aplicación escribir datos en un dispositivo de salida, como una pantalla o un archivo.
- `read()`: Permite a una aplicación leer datos de un dispositivo de entrada, como un teclado o un archivo.
- `open()`: Permite a una aplicación abrir un archivo para lectura, escritura o ambos.
- `close()`: Permite a una aplicación cerrar un archivo que ya no está en uso.
- `fork()`: Permite a una aplicación crear un nuevo proceso.
- `exec()`: Permite a una aplicación cargar un nuevo programa en memoria y ejecutarlo.
- `signal()`: Permite a una aplicación enviar una señal a otro proceso.

Las llamadas al sistema se implementan de forma diferente en cada sistema operativo. Sin embargo, en general, siguen un patrón similar. Cuando una aplicación realiza una llamada al sistema, el sistema operativo transfiere el control al núcleo del sistema. El núcleo ejecuta la llamada al sistema y luego devuelve el control a la aplicación.

Las llamadas al sistema son una parte fundamental de la arquitectura de cualquier sistema operativo. Sin ellas, los sistemas operativos no serían capaces de proporcionar los servicios que los usuarios esperan.

6.

De acuerdo con los *Operating System Concepts 9th Silberschatz and Galvin* y *Sistemas Operativos Modernos de Tanenbaum*, un sistema operativo multiprogramado es un sistema operativo que permite la ejecución simultánea de varios procesos en un mismo procesador. Esto se logra mediante la técnica de intercalación, que consiste en asignar el procesador a un proceso durante un breve periodo de tiempo, y luego pasarlo a otro proceso.

En los capítulos 1, 2 y 3 de *Operating System Concepts 9th Silberschatz and Galvin*, se define un sistema operativo como un programa de software que controla los recursos de hardware de una computadora y proporciona servicios a los programas de aplicación. Los sistemas operativos multiprogramados se introdujeron por primera vez en los años 60 y rápidamente se convirtieron en la norma.

Los sistemas operativos multiprogramados ofrecen una serie de ventajas, entre las que se incluyen:

- Permiten que los usuarios puedan ejecutar varias aplicaciones simultáneamente, lo que mejora la productividad.
- Permiten que los recursos del sistema, como el procesador y la memoria, se utilicen de manera más eficiente.
- Permiten que los sistemas operativos sean más tolerantes a los fallos, ya que si un proceso se bloquea, no afecta al resto de procesos.

Los sistemas operativos multiprogramados se utilizan en una gran variedad de sistemas, desde ordenadores personales hasta grandes sistemas centrales. Algunos ejemplos de sistemas operativos multiprogramados incluyen:

- Windows
- macOS
- Linux
- Unix

- Solaris
- IBM z/OS

Los sistemas operativos multiprogramados requieren de una serie de mecanismos para gestionar los procesos, tales como:

- Planificación: El sistema operativo debe decidir qué proceso ejecutar en cada momento.
- Almacenamiento: El sistema operativo debe almacenar los procesos en la memoria de forma eficiente.
- Protección: El sistema operativo debe proteger los recursos de cada proceso de los demás procesos.

La planificación de los procesos es una tarea compleja, ya que el sistema operativo debe tener en cuenta una serie de factores, como la prioridad de los procesos, el tiempo de espera de los procesos y el uso de los recursos.

El almacenamiento de los procesos en la memoria también es una tarea compleja, ya que el sistema operativo debe asegurarse de que hay suficiente memoria para todos los procesos.

La protección de los recursos de los procesos es esencial para evitar que un proceso acceda a los recursos de otro proceso.

En los capítulos 1 y 2 de "Sistemas Operativos Modernos de Tanenbaum", se describe en detalle cómo funcionan los sistemas operativos multiprogramados. Se explican los conceptos de proceso, planificación de procesos, almacenamiento de procesos y protección de procesos.

Los sistemas operativos multiprogramados son una parte fundamental de la informática moderna. Permiten que los usuarios puedan utilizar sus ordenadores de forma más eficiente y productiva.

7.

De acuerdo con los libros "Operating System Concepts" de 9th Silberschatz & Galvin y "Sistemas Operativos Modernos" de Tanenbaum, un proceso es un programa de computadora en ejecución. Está formado por un conjunto de instrucciones que se ejecutan de forma secuencial y que pueden acceder a los recursos del sistema, como el procesador, la memoria y los dispositivos de entrada/salida.

En los capítulos 1, 2 y 3 de *Operating System Concepts 9th Silberschatz and Galvin*, se define un proceso como una unidad de ejecución, que es una unidad de trabajo que el sistema operativo puede programar y ejecutar. Los procesos se pueden crear, destruir, cambiar de estado y suspender.

Los procesos se pueden clasificar en dos tipos principales:

- Procesos de usuario: Son los procesos que se ejecutan en nombre de un usuario. Los ejemplos típicos de procesos de usuario son los navegadores web, los procesadores de texto y los reproductores multimedia.
- Procesos del sistema: Son los procesos que se ejecutan en nombre del sistema operativo. Los ejemplos típicos de procesos del sistema son los procesos de gestión de memoria, los procesos de gestión de archivos y los procesos de gestión de dispositivos.

Cada proceso tiene su propio espacio de memoria virtual, que está aislado de los espacios de memoria de otros procesos. Esto significa que un proceso no puede acceder a la memoria de otro proceso, lo que ayuda a evitar errores y fallos.

Los procesos también tienen su propio conjunto de registros, que almacenan información sobre el estado del proceso. Los registros incluyen el contador de programa, que indica la siguiente instrucción que se va a ejecutar, y los registros de datos, que almacenan los datos que se están utilizando en la ejecución actual del proceso.

Los procesos se pueden comunicar entre sí mediante un mecanismo llamado llamadas al sistema. Las llamadas al sistema permiten a un proceso solicitar servicios del sistema operativo, como la lectura de un archivo, la escritura de un archivo o la comunicación con otro proceso.

En los capítulos 1 y 2 de "Sistemas Operativos Modernos de Tanenbaum", se describe en detalle cómo funcionan los procesos en los sistemas operativos modernos. Se explican los conceptos de estado de proceso, ciclo de vida de proceso y comunicación entre procesos.

Los procesos son una parte fundamental de los sistemas operativos modernos. Permiten que los usuarios puedan ejecutar varias aplicaciones simultáneamente y que los sistemas operativos puedan gestionar los recursos del sistema de forma eficiente.

8. Según *Operating System Concepts 9th Silberschatz and Galvin* y "Sistemas Operativos Modernos de Tanenbaum", los estados de un proceso son los siguientes:

- Nuevo: Un proceso se encuentra en estado nuevo cuando se crea, pero aún no se ha ejecutado.
- Listo: Un proceso se encuentra en estado listo cuando está listo para ejecutarse, pero no está usando el procesador.
- Ejecutando: Un proceso se encuentra en estado ejecutando cuando está usando el procesador.
- Bloqueado: Un proceso se encuentra en estado bloqueado cuando no puede ejecutarse porque está esperando algún evento, como que un recurso esté disponible.

- Terminado: Un proceso se encuentra en estado terminado cuando ha completado su ejecución.

El estado de un proceso puede cambiar a lo largo de su ciclo de vida. Por ejemplo, un proceso se mueve del estado nuevo al estado listo cuando se le asigna memoria y otros recursos. Se mueve del estado listo al estado ejecutando cuando el sistema operativo lo selecciona para ejecutar. Se mueve del estado ejecutando al estado bloqueado cuando está esperando un evento, como que un dispositivo de I/O esté disponible. Se mueve del estado bloqueado al estado listo cuando el evento esperado ocurre. Se mueve del estado listo al estado terminado cuando ha completado su ejecución.

Los libros mencionados proporcionan información sobre los siguientes aspectos de los estados de un proceso:

- Definición de los estados de un proceso
- Cambios de estado de un proceso
- Ciclo de vida de un proceso

Esta información es esencial para comprender el funcionamiento de los sistemas operativos modernos.

En particular, los libros mencionados explican que los estados de un proceso son importantes para el sistema operativo porque permiten que este controle la ejecución de los procesos de forma eficiente. Por ejemplo, el sistema operativo puede utilizar los estados de los procesos para decidir qué proceso ejecutar en cada momento.

9.

De acuerdo con los libros *Operating System Concepts 9th Silberschatz and Galvin* y "Sistemas Operativos Modernos de Tanenbaum", la información que se guarda en el PCB asociado a un proceso incluye lo siguiente:

- Estado del proceso: El estado actual del proceso, como nuevo, listo, ejecutando, bloqueado o terminado.
- Registros del proceso: Los registros del procesador que el proceso está utilizando, como el contador de programa, los registros de datos y los registros de estado.
- Pila del proceso: La pila de llamadas al sistema que el proceso está utilizando.
- Dirección del proceso: La dirección de memoria virtual del proceso.
- Prioridad del proceso: La prioridad del proceso, que se utiliza para decidir qué proceso ejecutar en cada momento.
- Referencias de archivos del proceso: Las referencias de archivos que el proceso ha abierto.
- Señales del proceso: Las señales que el proceso ha recibido.

- Datos del proceso: Otros datos asociados al proceso, como el nombre del proceso, el usuario que lo creó y la fecha en que se creó.

El PCB es una estructura de datos que almacena información sobre el estado y los recursos de un proceso. El sistema operativo utiliza el PCB para controlar la ejecución de los procesos.

Los libros mencionados proporcionan información sobre los siguientes aspectos del PCB:

- Definición del PCB
- Información almacenada en el PCB
- Uso del PCB por el sistema operativo

En particular, los libros mencionados explican que el PCB es importante para el sistema operativo porque permite que este controle la ejecución de los procesos de forma eficiente. Por ejemplo, el sistema operativo puede utilizar el PCB para determinar el estado actual de un proceso, los registros que está utilizando y los recursos a los que tiene acceso.

10.

De acuerdo con los libros *Operating System Concepts 9th Silberschatz and Galvin* y *Sistemas Operativos Modernos de Tanenbaum*, las actividades principales de un sistema operativo en relación con la gestión de procesos son las siguientes:

- Creación y terminación de procesos: El sistema operativo debe crear nuevos procesos cuando se inician aplicaciones o se ejecutan tareas. También debe terminar los procesos que ya no son necesarios.
- Planificación de procesos: El sistema operativo debe decidir qué proceso ejecutar en cada momento. Esto se llama planificación de procesos.
- Almacenamiento de procesos: El sistema operativo debe almacenar los procesos en la memoria de forma eficiente. Esto se llama almacenamiento de procesos.
- Protección de procesos: El sistema operativo debe proteger los recursos de cada proceso de los demás procesos. Esto se llama protección de procesos.

Las actividades de gestión de procesos son esenciales para que los sistemas operativos puedan ejecutar múltiples procesos de forma simultánea.

En particular, los libros mencionados explican que las actividades de gestión de procesos son importantes para el sistema operativo porque permiten que este controle la ejecución de los procesos de forma eficiente. Por ejemplo, el sistema operativo puede utilizar la planificación de

procesos para garantizar que los procesos más importantes se ejecuten primero. También puede utilizar la protección de procesos para evitar que un proceso acceda a los recursos de otro proceso.

Aquí hay una descripción más detallada de cada actividad:

- Creación y terminación de procesos: Cuando se inicia una aplicación o se ejecuta una tarea, el sistema operativo debe crear un nuevo proceso. El proceso se crea asignando un PCB y asignando memoria al proceso. Cuando el proceso ya no es necesario, el sistema operativo debe terminarlo liberando la memoria y otros recursos que el proceso estaba utilizando.
- Planificación de procesos: El sistema operativo debe decidir qué proceso ejecutar en cada momento. Esto se hace teniendo en cuenta factores como la prioridad del proceso, el tiempo de espera del proceso y el uso de los recursos del proceso.
- Almacenamiento de procesos: Los procesos se almacenan en la memoria principal del sistema. El sistema operativo debe asegurarse de que los procesos se almacenan de forma eficiente para que se puedan ejecutar de forma eficaz.
- Protección de procesos: Los procesos deben estar protegidos entre sí para evitar que un proceso acceda a los recursos de otro proceso. Esto puede hacerse mediante la asignación de direcciones de memoria virtuales únicas a cada proceso y mediante el uso de mecanismos de protección de memoria.

Los sistemas operativos modernos suelen utilizar técnicas avanzadas de gestión de procesos para mejorar el rendimiento y la eficiencia. Estas técnicas pueden incluir la multiprogramación, la hibernación de procesos y la preemption.

Desarrollo Practica:

Nuestro código recopila instrucciones de un archivo de texto, línea por línea. Luego, descompone el contenido de cada línea en instrucciones, registros y datos. Utilizamos estructuras y funciones previamente creadas para realizar comprobaciones utilizando enums, funciones y estructuras. También diseñamos funcionalidades adicionales para mejorar la eficiencia y la precisión del código, la cual fue la detección y correcta ejecución de la instrucción SUB. Esto nos permite procesar las instrucciones de manera más efectiva y obtener resultados precisos.

El Código de nuestro código se encuentra en el siguiente link de GitHub: <https://github.com/Miguel-Angel-Nivia/Sistemas-Operativos.git>

Aquí se encuentra el código main, un .h que se usa para el main y los .txt para los textos planos donde son diversos ejemplos de usabilidad de nuestro código (Todo se encuentra documentado).

Esperamos que le guste nuestro trabajo y quedamos atentos a cualquier comentario.