

Práctica 9

1 ¿Que es MS-DOS? Da un pequeño resumen sobre que es MS-DOS, para que se usa y como funciona.

1.1 Respuesta

MS-DOS, siglas de Microsoft Disk Operating System, es un sistema operativo para computadoras basado en x86 que fue muy popular en los años 80 y principios de los 90. Fue desarrollado por Microsoft y es un sistema operativo de interfaz de línea de comandos. MS-DOS no utiliza una interfaz gráfica de usuario; en su lugar, opera a través de comandos de texto que el usuario debe escribir en la consola.

MS-DOS gestiona los archivos almacenados en discos, y permite ejecutar programas y gestionar la memoria del sistema. Es un sistema monousuario y monotarea, lo que significa que puede manejar un usuario a la vez y ejecutar un programa en cada momento. El sistema se carga desde el disco al iniciar la computadora y opera directamente sobre el hardware, ofreciendo comandos para la manipulación de archivos y directorios, gestión de procesos y más.

2 ¿Que es Shell? Da un pequeño resumen sobre que es Shell, para que se usa y como funciona.

2.1 Respuesta

El término "Shell" se refiere a la interfaz de usuario utilizada para acceder a los servicios de un sistema operativo. Funciona como un intérprete de comandos que permite a los usuarios comunicarse con el sistema operativo a través de comandos escritos, ofreciendo una capa de abstracción sobre las llamadas al sistema. En el contexto de MS-DOS, el shell más común es el Command.com, mientras que en sistemas Unix y Linux, ejemplos populares incluyen bash, zsh y fish.

El shell es esencialmente un programa que lee los comandos del usuario y los ejecuta, devolviendo los resultados al usuario. Los usuarios pueden realizar tareas como gestionar archivos, ejecutar software, y manipular el estado del sistema operativo directamente a través del shell. Además, los shells modernos

soportan scripting, lo que permite a los usuarios escribir secuencias de comandos para automatizar tareas repetitivas.

3 Si tuviéramos que implementar el interprete de comandos en una sistema operativo. ¿Porque es importante usar subrutinas y la convención de subrutinas?

3.1 Respuesta

Implementar subrutinas en el desarrollo de un intérprete de comandos de un sistema operativo es crucial por varias razones:

1. **Reutilización de código:** Las subrutinas permiten reutilizar código para tareas comunes, como el análisis de la entrada del usuario o la ejecución de comandos, lo cual reduce la redundancia y facilita el mantenimiento del código.
2. **Abstracción:** Las subrutinas ayudan a abstraer la complejidad operacional, dividiendo el programa en componentes más pequeños y manejables. Esto hace que el sistema sea más fácil de entender y modificar.
3. **Gestión de memoria:** Usar subrutinas puede ayudar a optimizar el uso de la memoria, ya que se puede controlar mejor la asignación y desasignación de recursos durante la ejecución de diferentes partes del intérprete.

4 Considerando los componentes básicos de una computadora moderna, además de las llamadas al sistema descritas en la práctica, ¿Qué otras llamadas al sistema resultarían necesarias?

4.1 Respuesta

Además de las llamadas al sistema para la ejecución de programas y gestión de archivos, otros tipos de llamadas al sistema son necesarias en una computadora moderna para:

1. **Gestión de dispositivos:** Para interactuar con el hardware como impresoras, redes y dispositivos de almacenamiento.
2. **Gestión de procesos:** Además de iniciar y terminar procesos, es crucial poder gestionar la prioridad y el uso de recursos de los procesos.

3. **Gestión de memoria:** Las llamadas al sistema que permiten asignar y liberar memoria son fundamentales para el manejo eficiente de los recursos del sistema.

5 Para que un programa pueda comunicarse con el hardware del equipo, se deben usar llamadas a sistema. Considerando esto, ¿Porque un programa cualquiera NO puede llamar directamente al Hardware de una PC?

5.1 Respuesta

Los programas se comunican con el hardware del equipo a través de llamadas al sistema porque proporcionan una interfaz controlada y segura para interactuar con los componentes del hardware. Directamente, los programas no pueden acceder al hardware debido a:

1. **Seguridad:** El acceso directo al hardware puede llevar a operaciones no seguras que podrían dañar el sistema.
2. **Estabilidad:** Limitar el acceso al hardware a través de llamadas al sistema ayuda a prevenir que los programas interfieran unos con otros o con el funcionamiento esencial del sistema.
3. **Abstracción:** Las llamadas al sistema ofrecen una capa de abstracción que simplifica la programación. Los desarrolladores de software pueden escribir programas sin necesidad de entender todos los detalles del hardware subyacente.

6 Investiga y describe con tus propias palabras ¿cómo resuelve una llamada al sistema el sistema operativo?

6.1 Respuesta

Cuando un programa realiza una llamada al sistema, el proceso típicamente incluye los siguientes pasos:

1. **Interrupción de software:** El programa ejecuta una instrucción especial que causa una interrupción de software, transfiriendo el control al sistema operativo.
2. **Contexto de seguridad:** El sistema operativo toma control, evaluando la solicitud para asegurarse de que el programa tiene permiso para realizar la llamada al sistema.

3. **Ejecución:** Si la llamada es válida, el sistema operativo ejecuta la operación correspondiente. Esto puede involucrar interactuar con el hardware, gestionar recursos, o realizar otras operaciones administrativas.
4. **Retorno:** Una vez completada la operación, el control se devuelve al programa, junto con los resultados de la operación de la llamada al sistema.