

GP4 Arquitectura y Sistemas Operativos

Alumno: Cardozo Benjamín Emanuel, Benítez Santiago Miguel, Ortiz Marcos Andrés Ramón

Organización y Arquitectura de Computadoras

1. Defina el concepto de computadora elemental.

Una computadora elemental es un dispositivo electrónico capaz de recibir datos de entrada, procesarlos siguiendo un conjunto de instrucciones

2. Explique qué es la arquitectura de Von Neumann y describa sus principales componentes.

La arquitectura de Von Neumann es un modelo de diseño de computadoras propuesto por el matemático John von Neumann. Este modelo estableció las bases para el funcionamiento de la mayoría de las computadoras modernas

Principales características:

Programa y datos se almacenan en la misma memoria.

La CPU sigue un ciclo de instrucción: buscar (fetch) la instrucción desde la memoria, decodificarla y ejecutarla.

Procesamiento secuencial: las instrucciones se procesan una después de otra

Principales componentes:

1. Unidad de Entrada/Salida:

Permite la comunicación entre la computadora y el usuario o dispositivos externos (ej: teclado, impresora).

2. Memoria Principal:

Almacena tanto los datos como el programa que debe ejecutarse. La misma memoria contiene instrucciones y datos, lo que simplifica la arquitectura

3. Unidad de Control:

Coordina y dirige el flujo de datos e instrucciones dentro de la CPU y entre los componentes del sistema

4. Unidad Aritmético-Lógica (ALU):

Se encarga de realizar operaciones matemáticas y lógicas

5. Bus de Datos:

Conjunto de líneas que transporta datos entre los distintos componentes de la computadora

La mayoría de las computadoras actuales utilizan una arquitectura basada en Von Neumann o variaciones de ella

3. Arquitectura Harvard, diferencias con Von Neumann

La arquitectura Harvard un modelo de diseño de computadoras, desarrollado en los años 1940 para el proyecto Harvard Mark I. A diferencia de Von Neumann, separaba la memoria de datos y la memoria de instrucciones.

Principales características de la Arquitectura Harvard:

Memoria separada para datos y para instrucciones.

Buses separados para datos y para instrucciones, permitiendo que ambos fluyan al mismo tiempo (acceso paralelo).

Mayor velocidad y eficiencia, ya que no hay "colisión" entre la lectura de datos y la lectura de instrucciones.

Diferencias principales entre Von Neumann y Harvard:

En la arquitectura Von Neumann, los datos y las instrucciones comparten la misma memoria y utilizan el mismo bus de comunicación. En la arquitectura Harvard, los datos y las instrucciones se almacenan en memorias separadas y se usan buses independientes

La arquitectura Von Neumann es más sencilla y más barata de implementar, pero puede ser más lenta. La arquitectura Harvard es más rápida y eficiente en términos de ejecución, pero su diseño es más complejo

Arquitectura Von Neumann: Computadoras personales (PCs, notebooks).

Arquitectura Harvard: Microcontroladores como Arduino o PIC, donde se requiere alta velocidad y bajo consumo de energía.

4. Arquitecturas RISC y CISC. Conceptos y diferencias.

RISC (Reduced Instruction Set Computer) y CISC (Complex Instruction Set Computer) son dos enfoques diferentes para diseñar la arquitectura interna de los procesadores.

RISC

RISC utiliza un conjunto reducido y simple de instrucciones. Cada instrucción realiza una operación muy básica, lo que permite que todas las instrucciones tomen el mismo tiempo para ejecutarse (normalmente un solo ciclo de reloj). Fomenta la simplicidad y eficiencia: los procesadores RISC suelen ser más rápidos y consumen menos energía.

Ejemplos de procesadores RISC: ARM, MIPS, PowerPC.

CISC

CISC emplea un conjunto complejo y amplio de instrucciones. Cada instrucción puede realizar operaciones más sofisticadas, como multiplicaciones o el manejo directo de memoria, en una sola instrucción. El objetivo de CISC era reducir la cantidad de instrucciones necesarias para ejecutar un programa, optimizando el uso de la memoria.

Ejemplos de procesadores CISC: Intel x86, AMD Ryzen.

5. Defina qué es una CPU y explique su importancia en una computadora.

La CPU es el "cerebro" de la computadora. Se encarga de ejecutar instrucciones, procesar datos y coordinar todas las actividades del sistema. Es un circuito electrónico que interpreta y ejecuta las instrucciones de los programas almacenados en la memoria. Realiza operaciones aritméticas, lógicas, de control y de entrada/salida.

¿Porque es importante?

La CPU transforma los datos de entrada en resultados de salida útiles.

Supervisa y dirige el funcionamiento de los diferentes componentes (memoria, dispositivos de entrada/salida, etc.).

El rendimiento general de una computadora depende en gran medida de la potencia y eficiencia de su CPU.

A través de operaciones lógicas, permite que los programas sigan diferentes caminos de ejecución según condiciones específicas.

Ejemplo:

Un procesador Intel Core i7 es una CPU moderna capaz de realizar miles de millones de instrucciones por segundo

6. Describa qué son la Unidad de Control (UC) y la Unidad Aritmético-Lógica (UAL) y la función de cada una dentro de la CPU.

Unidad de Control (UC)

Se encarga de interpretar las instrucciones de los programas y de coordinar el trabajo de todos los componentes internos de la computadora.

Funciones principales:

Lee las instrucciones almacenadas en la memoria.

Decodifica qué acción debe realizarse.

Envía señales de control a otras partes de la CPU (como la ALU y registros) para ejecutar la instrucción.

Controla el flujo de datos dentro de la CPU y entre la CPU y el resto del sistema (memoria y dispositivos de entrada/salida)

Unidad Aritmético-Lógica (UAL o ALU)

Realiza las operaciones matemáticas y lógicas básicas necesarias para procesar datos.

Funciones principales:

Efectúa operaciones aritméticas como suma, resta

Realiza operaciones lógicas como comparaciones (igualdad, mayor que, menor que) y operaciones de tipo AND, OR, NOT.

Determina resultados de condiciones necesarias para la ejecución de instrucciones (por ejemplo, saber si un número es igual a cero para un salto condicional)

7. Explique el concepto de programas e instrucciones desde la perspectiva del funcionamiento de la CPU.

Programas:

Un programa es un conjunto ordenado de instrucciones almacenados en memoria que le indican a la CPU qué tareas debe realizar. Cada programa está formado por pasos lógicos y específicos que la computadora debe seguir para completar una tarea

Instrucciones:

Una instrucción es una orden individual que la CPU puede entender y ejecutar.

Cada instrucción generalmente indica:

Qué operación realizar (por ejemplo, sumar dos valores). Sobre qué datos actuar (por ejemplo, valores almacenados en ciertas posiciones de memoria)

8. ¿Cómo procesa la CPU los datos para realizar operaciones?

La CPU procesa los datos siguiendo un ciclo de ejecución que se repite constantemente mientras la computadora está encendida. Este ciclo tiene varias etapas bien definidas:

Etapas principales del procesamiento de datos:

1. Captura o Búsqueda (Fetch): La CPU lee una instrucción desde la memoria principal, en la dirección que indica el contador de programa.
2. Decodificación (Decode): La CPU interpreta la instrucción: entiende qué operación debe realizar y con qué datos.
3. Ejecución (Execute): La CPU realiza la operación indicada por la instrucción: puede ser una suma, una comparación, un movimiento de datos, etc.
4. Acceso a Memoria (si es necesario): Si la instrucción requiere leer o escribir datos, la CPU accede a la memoria RAM o a registros internos.
5. Escritura del Resultado El resultado de la operación se guarda en un registro, en la memoria o se envía a un dispositivo de salida.
6. Actualización del contador de programa: La CPU avanza al siguiente paso, incrementando el contador de programa para procesar la próxima instrucción.

9. Jerarquía de memorias. Diferencie entre memoria principal y memorias secundarias, proporcionando ejemplos de cada tipo.

La jerarquía de memorias es una organización de los distintos tipos de memoria en un sistema, ordenados según su:

Velocidad (más rápida arriba, más lenta abajo).

Costo (más cara arriba, más barata abajo).

Capacidad (menos capacidad arriba, más capacidad abajo).

El objetivo de la jerarquía es optimizar el rendimiento y el costo del sistema.

Memoria Principal:

Definición:

Es la memoria que la CPU puede acceder directamente y de manera rápida para ejecutar instrucciones y procesar datos.

Características:

Rápida.

Volátil (pierde su contenido al apagar la computadora).

Ejemplos:

RAM (Memoria de Acceso Aleatorio): almacena datos y programas en uso inmediato.

ROM (Memoria de Solo Lectura): contiene instrucciones básicas de arranque del sistema (como el BIOS).

Memorias Secundarias:

Definición:

Son dispositivos de almacenamiento masivo que guardan datos de forma permanente o semipermanente.

Características:

Más lentas que la memoria principal.

No volátiles (conservan la información aunque se apague el equipo).

Mayor capacidad de almacenamiento.

Ejemplos:

Discos duros (HDD).

Unidades de estado sólido (SSD).

Memorias USB (pendrives).

Tarjetas SD.

Discos ópticos como CD, DVD, Blu-ray.

Resumen de la jerarquía típica (de arriba hacia abajo):

1. Registros (en la CPU) más rápidos y pequeños.
2. Cache (L1, L2, L3) memoria intermedia de alta velocidad.
3. RAM memoria principal.
4. SSD
- / HDD almacenamiento masivo.
5. Medios externos USB, CD, backups en la nube.

10. Explique qué es la RAM y su rol dentro de la computadora.

¿Qué es la RAM?

La RAM o Memoria de Acceso Aleatorio es un tipo de memoria volátil que se utiliza como memoria principal en una computadora.

Rol de la RAM dentro de la computadora:

Almacena datos e instrucciones temporalmente mientras la CPU los está utilizando. Proporciona acceso rápido a los programas activos y los archivos en uso, mucho más rápido que el acceso a un disco duro o SSD.

Permite la multitarea: cuantos más datos puede guardar la RAM, más programas pueden ejecutarse simultáneamente sin afectar demasiado el rendimiento.

Cuando se apaga la computadora, todo el contenido de la RAM se pierde (es memoria volátil).

11. Describa el propósito y funcionamiento de los dispositivos de almacenamiento fuera de línea y los dispositivos extraíbles.

Su propósito principal es guardar información de forma segura, hacer copias de respaldo (backups), transportar datos o liberar espacio en el almacenamiento principal.

Para acceder a los datos, es necesario conectar el dispositivo al sistema o cargarlo de forma manual.

Ejemplos de dispositivos de almacenamiento fuera de línea: Discos duros externos (HDD portátiles).

Unidades de estado sólido portátiles (SSD externos).

CDs, DVDs, Blu-ray

¿Qué son los dispositivos extraíbles?

Son un tipo especial de almacenamiento que puede ser conectado y desconectado fácilmente de la computadora mientras esta está en funcionamiento. Sirven para transferir información entre equipos o aumentar temporalmente el almacenamiento disponible.

Ejemplos de dispositivos extraíbles: Memorias USB (pendrives).

Tarjetas de memoria SD/microSD.

Diferencias entre ambos conceptos:

Ambos son removibles físicamente, pero el almacenamiento fuera de línea destaca la idea de que no está siempre conectado y su función muchas veces es la seguridad o el resguardo de datos importantes. Los dispositivos extraíbles se centran más en facilidad de uso y movilidad de datos.

12. Defina el concepto de periféricos y clasifíquelos en categorías según su función.

Un periférico es cualquier dispositivo externo conectado a una computadora que permite la entrada, salida o almacenamiento de información, o la comunicación con otros sistemas. Son los encargados de expandir las capacidades de la computadora, permitiendo que interactúe con el usuario o con otros dispositivos.

Clasificación de periféricos según su función:

1. Periféricos de entrada: Permiten introducir datos y comandos en la computadora.

Ejemplos: Teclado, Mouse

2. Periféricos de salida:

Permiten que la computadora devuelva información procesada al usuario.

Ejemplos: Monitor, Impresora

3. Periféricos de entrada/salida (mixtos):

Permiten tanto ingresar como obtener información de la computadora.

Ejemplos: Pantallas táctiles, Pendrives (USB)

13. Explique los mecanismos básicos de procesamiento de entrada/salida (E/S).

El procesamiento de entrada/salida (E/S) es el conjunto de métodos que permiten que la computadora interactúe con los dispositivos periféricos para recibir (entrada) o enviar (salida) datos. Cuando un programa necesita comunicarse con un dispositivo (por ejemplo, leer un archivo de un pendrive o imprimir un documento), el sistema operativo y la CPU utilizan distintos mecanismos para gestionar esa transferencia de información.

Ejemplo mecanismos de procesamiento de E/S:

1. E/S programada: La CPU controla directamente cada transferencia de datos. El procesador debe esperar a que cada operación de E/S se complete antes de continuar.

Desventaja: Ineficiencia, ya que la CPU pierde tiempo valioso esperando.

2. E/S mediante interrupciones: El dispositivo envía una interrupción a la CPU cuando está listo para enviar o recibir datos. La CPU puede seguir ejecutando otras tareas mientras tanto, mejorando la eficiencia.

3. Acceso Directo a Memoria (DMA - Direct Memory Access): Un controlador especial (el controlador DMA) se encarga de transferir datos entre un dispositivo y la memoria sin intervención directa de la CPU.

14. ¿Cómo gestiona un sistema operativo los dispositivos de E/S y qué problemas puede solucionar al hacerlo?

El sistema operativo (SO) se encarga de administrar el acceso a los dispositivos de E/S para que los programas y usuarios puedan utilizarlos de forma segura, ordenada y eficiente. La gestión de E/S incluye tareas como:

Detección de dispositivos: Identificar qué periféricos están conectados.

Instalación y uso de controladores (drivers): Utilizar programas especiales que permiten al SO comunicarse con el hardware.

Planificación y control de accesos: Decidir qué proceso accede a qué dispositivo y cuándo.

Cola de operaciones: Organizar varias solicitudes de E/S y atenderlas en orden o según prioridad.

Protección y seguridad: Evitar que un dispositivo sea usado incorrectamente o de manera simultánea por dos procesos que puedan generar conflicto.

15. ¿Qué son los buses, cuáles son sus funciones y tipos? Grafique y explique una estructura tipo.

Un bus es un sistema de comunicación que transfiere datos entre los distintos componentes de una computadora (CPU, memoria, dispositivos de E/S). Básicamente, es un conjunto de cables o pistas que permite que la información viaje de un lugar a otro dentro del sistema.

Funciones principales de los buses:

Transferir datos entre la CPU, la memoria, y los dispositivos de entrada/salida.

Transmitir direcciones de memoria donde deben ser leídos o escritos los datos.

Transportar señales de control para coordinar las operaciones (como señales de lectura o escritura).

Tipos de buses:

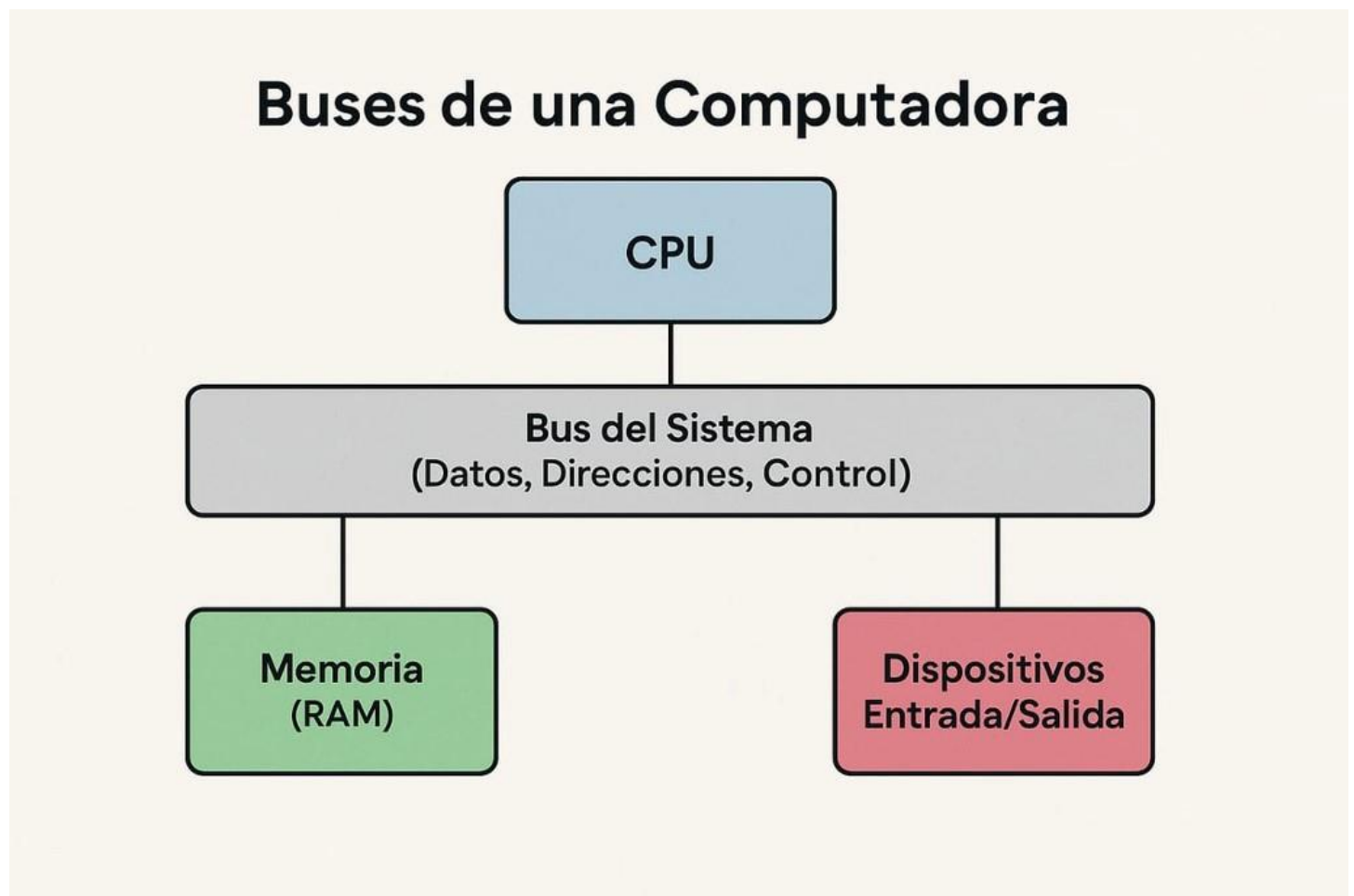
1. Bus de datos:

Transporta los datos propiamente dichos.

Su ancho (número de líneas) determina cuánta información puede transferir simultáneamente (por ejemplo, un bus de 32 bits puede mover 32 bits de datos de una vez).

2. Bus de direcciones:Transporta la dirección de memoria donde deben ser leídos o escritos los datos.Define cuánta memoria puede direccionar la CPU (por ejemplo, un bus de 16 bits puede direccionar $2^{16} = 65.536$ ubicaciones de memoria).
3. Bus de control:Transporta señales de control que coordinan y gestionan las operaciones (como órdenes de lectura, escritura, interrupciones, etc.).

Esquema de Buses



16. ¿Cuáles buses se utilizan en la actualidad?

Hoy en día, los sistemas de computadoras modernas utilizan varios tipos de buses especializados para diferentes funciones. Los principales son:

Buses internos:

Bus de datos interno: Conecta los diferentes componentes internos de la CPU (como registros, ALU, UC).

Bus de memoria: Conecta la CPU con la memoria RAM.

Bus de control: Transporta señales de control que coordinan las operaciones de lectura/escritura.

Buses externos / de expansión:

PCI Express (PCIe): Es el bus de expansión más usado actualmente. Conecta tarjetas gráficas, SSDs, tarjetas de red, etc. Es muy rápido y permite comunicaciones paralelas.

USB (Universal Serial Bus): Es uno de los buses más populares para conectar dispositivos externos como teclados, pendrives, impresoras, celulares, etc.

SATA (Serial ATA): Se utiliza para conectar unidades de almacenamiento como discos duros (HDD), unidades SSD y unidades ópticas (DVD, Blu-Ray).

Thunderbolt: Una tecnología de alta velocidad que combina datos, video y energía en un solo cable, usada sobre todo en computadoras Apple.

I2C y SPI (en sistemas embebidos): Son buses usados en dispositivos pequeños como sensores, microcontroladores y placas de desarrollo (Arduino, Raspberry Pi).

17. Introduzca el concepto de sistemas operativos y explique su importancia en la gestión de recursos informáticos. Ejemplos.

¿Qué es un sistema operativo?

Un sistema operativo (SO) es un software fundamental que administra y controla el hardware y otros programas en una computadora o dispositivo. Actúa como un intermediario entre el usuario y el hardware, permitiendo que los recursos de la máquina se utilicen de manera organizada y eficiente.

Importancia en la gestión de recursos informáticos:

El sistema operativo tiene varias funciones clave: Gestión de procesos: Controla la ejecución de programas y asigna recursos (tiempo de CPU, memoria) a cada uno.

Gestión de memoria: Administra el uso de la memoria principal, asignando espacio a los programas en ejecución.

Gestión de dispositivos de entrada/salida: Controla la comunicación entre la CPU y los periféricos (impresoras, discos, teclado, etc.).

Gestión de archivos: Organiza, almacena, recupera y protege los archivos en dispositivos de almacenamiento.

Seguridad y protección: Protege los datos y recursos de accesos no autorizados o errores accidentales.

Ejemplos de sistemas operativos:

Para computadoras de escritorio y portátiles: Microsoft Windows 10, Windows 11. macOS (Apple).
Linux (distribuciones como Ubuntu, Fedora).

Para dispositivos móviles: Android (Google). iOS (Apple).

Para servidores: Windows Server. Red Hat Enterprise Linux. Ubuntu Server.

18. Defina qué es un kernel y describa los tipos de kernels que existen. (Basado en la estructura de los SO según Tanenbaum)

¿Qué es un kernel?

El kernel es el núcleo del sistema operativo, la parte central y más importante.

Se encarga de: Gestionar la comunicación entre el hardware y el software.

Coordinar el uso de la CPU, la memoria, los dispositivos de entrada/salida y los archivos.

Asegurar que múltiples procesos puedan ejecutarse de manera controlada y segura

Tipos de kernels según Tanenbaum:

1. Monolítico:

Todo el sistema operativo corre en un solo espacio de memoria llamado modo kernel.

Todos los servicios (manejo de archivos, dispositivos, redes, etc.) están integrados directamente en el kernel.

Ventaja: Mayor velocidad de ejecución porque todo está junto. Desventaja: Más difícil de mantener, cualquier error puede colapsar todo el sistema.

Ejemplo: Linux tradicional, Unix.

2. Microkernel:

Solo se incluyen las funciones más básicas dentro del kernel (como la comunicación entre procesos y el manejo básico de hardware).

Los servicios del sistema operativo (archivos, redes, etc.) corren fuera del kernel como procesos independientes (llamados servidores).

Ventaja: Mayor estabilidad y seguridad; si un servidor falla, no colapsa todo el sistema. Desventaja: Más sobrecarga de comunicación entre procesos, lo que puede hacer que sea más lento.

Ejemplo: Minix, QNX.

3. Kernel híbrido:

Combina características de los kernels monolíticos y microkernels. Intenta lograr un buen balance entre eficiencia y modularidad.

Ejemplo: Windows NT (y sus sucesores como Windows 10/11), macOS.

4. Exokernel:

Es una arquitectura experimental donde el kernel ofrece lo mínimo indispensable para dar control directo a las aplicaciones sobre el hardware.

Ventaja: Máximo rendimiento posible.

Desventaja: Mucha complejidad para los programadores.

Ejemplo: Proyecto Exokernel del MIT.

19. Explique cómo el diseño y la implementación de un kernel afectan al rendimiento y seguridad de un sistema operativo.

¿Cómo afecta el diseño del kernel al rendimiento?

Velocidad de ejecución: Un kernel monolítico puede ofrecer mayor velocidad, ya que todos los servicios (manejo de archivos, redes, dispositivos) están integrados en un solo bloque de código, eliminando la necesidad de pasar constantemente mensajes entre procesos.

Tiempo de respuesta: En un microkernel, debido a que los servicios operan como procesos separados y deben comunicarse mediante mensajes, hay mayor sobrecarga, lo que puede retrasar ligeramente el procesamiento de operaciones simples.

Eficiencia del hardware: Kernels como los híbridos intentan equilibrar la eficiencia (manteniendo servicios críticos cerca del núcleo) con la estabilidad de un microkernel, buscando optimizar el uso de CPU, memoria y dispositivos.

¿Cómo afecta el diseño del kernel a la seguridad?

Superficie de ataque: En un kernel monolítico, como todo corre en el mismo espacio de memoria (modo kernel), un error o vulnerabilidad en cualquier parte puede comprometer todo el sistema operativo.

Si un driver o servicio falla, puede provocar desde bloqueos hasta vulnerabilidades de seguridad graves.

Aislamiento de procesos: En un microkernel, como los servicios funcionan separados en modo usuario, es más difícil que un fallo afecte a todo el sistema. Si un servicio falla, solo ese servicio puede detenerse o reiniciarse, no todo el sistema operativo.

Facilidad de actualización: En arquitecturas tipo microkernel o híbridas, es más fácil actualizar o reemplazar partes específicas (como un servidor de archivos o de red) sin necesidad de reiniciar todo el sistema, reduciendo los riesgos de seguridad.

20. Describa un proceso tipo de arranque de una computadora.

Pasos principales del proceso de arranque:

1. Encendido y suministro de energía (Power On):

Al presionar el botón de encendido, se activa la fuente de alimentación y se distribuye corriente a todos los componentes.

2. Ejecución del POST (Power-On Self-Test):

La computadora realiza un autochequeo de hardware: verifica que memoria RAM, teclado, procesador, tarjeta gráfica y otros componentes esenciales estén presentes y funcionen correctamente.

3. Inicialización del BIOS/UEFI:

El BIOS (Basic Input/Output System) o el más moderno UEFI (Unified Extensible Firmware Interface) toma el control. Configura los componentes básicos de hardware (discos, puertos USB, etc.) y prepara el sistema para cargar el sistema operativo.

4. Búsqueda del dispositivo de arranque:

El BIOS/UEFI busca en una lista de dispositivos (disco duro, USB, CD/DVD, red) el sector de arranque (boot sector).

El orden de búsqueda se puede configurar (por ejemplo, arrancar primero desde un pendrive).

5. Carga del cargador de arranque (Bootloader): El bootloader (como GRUB en Linux o el gestor de arranque de Windows) es un pequeño programa cuya misión es cargar el sistema operativo en la memoria RAM.

6. Carga del sistema operativo: El sistema operativo comienza a cargarse en la RAM: primero el kernel, luego los controladores de dispositivos y, finalmente, los servicios y programas necesarios.

7. Inicio de sesión del usuario: Una vez que el sistema operativo está listo, se muestra la pantalla de inicio de sesión o el escritorio para que el usuario pueda empezar a trabajar.