

INVESTIGACION 02

PRESENTADO POR:

SEBASTIÁN DAVID MENDOZA ALVARADO

PRESENTADO A:

PABLO ROJAS

Docente

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE FISICOMECHANICAS

ESCUELA DE INGENIERIA DE SISTEMAS

SISTEMAS OPERACIONALES

BUCARAMANGA

2023

1. En entornos compartidos, los usuarios comparten el sistema. Esto puede dar lugar a varios problemas de seguridad.

- **Mencione dos problemas y explíquelos.**

A) **Acceso no autorizado:** En un sistema compartido, cualquier persona con acceso al sistema puede tener acceso a los archivos y datos de otros usuarios, lo que aumenta el riesgo de que alguien obtenga información confidencial o realice cambios malintencionados en los archivos de otros usuarios.

B) **Vulnerabilidades de software:** Los sistemas compartidos a menudo ejecutan software común, como sistemas operativos, aplicaciones y servicios, que pueden contener vulnerabilidades de seguridad. Si un atacante aprovecha una vulnerabilidad en el software compartido, puede comprometer el sistema y acceder a los datos de todos los usuarios.

- **Es posible asegurar el mismo grado de seguridad en un sistema compartido que en un sistema dedicado. Explique su respuesta**

No es posible asegurar el mismo grado de seguridad en un sistema compartido que en un sistema dedicado. En un sistema compartido, varios usuarios tienen acceso a los recursos y servicios del sistema, lo que significa que existe un mayor riesgo de acceso no autorizado, brechas de seguridad y violaciones de datos. Además, en un entorno compartido, los usuarios pueden tener diferentes niveles de habilidades y conocimientos técnicos, lo que puede aumentar aún más el riesgo de errores de seguridad.

En contraste, un sistema dedicado está diseñado para ser utilizado por un solo usuario o grupo de usuarios con privilegios limitados, lo que reduce el riesgo de acceso no autorizado y limita la exposición a vulnerabilidades de seguridad. Además, los sistemas dedicados suelen tener un mayor control de acceso y medidas de seguridad más estrictas, lo que los hace más seguros que los sistemas compartidos.

En resumen, aunque se pueden tomar medidas para aumentar la seguridad en un sistema compartido, nunca se podrá garantizar el mismo grado de seguridad que en un sistema dedicado. Por lo tanto, es importante evaluar cuidadosamente los riesgos de seguridad y elegir la opción de sistema adecuada en función de las necesidades y el uso previsto.

2. Un problema común en los OS es la utilización de recursos. Enumere los recursos que deben gestionarse en las siguientes maquina (explique porqué):

- **Sistemas embebidos:** en sistemas embebidos, que son sistemas informáticos integrados en dispositivos o equipos más grandes, los recursos que deben gestionarse incluyen:

**Memoria:** la memoria es limitada en los sistemas embebidos, por lo que se debe administrar cuidadosamente para evitar que el sistema se quede sin memoria y se produzcan errores o fallas. Además, los sistemas embebidos suelen utilizar memoria no volátil, como EEPROM o flash, lo que significa que se debe tener cuidado de no sobrescribir esta memoria de forma incorrecta.

**CPU:** la CPU es el procesador del sistema y es responsable de realizar las operaciones del sistema. En los sistemas embebidos, la CPU suele ser menos potente que en otros sistemas, lo que significa que se deben optimizar las operaciones del sistema para evitar la sobrecarga de la CPU y mejorar el rendimiento.

**Almacenamiento:** en los sistemas embebidos, el almacenamiento suele ser limitado y no es posible agregar más espacio de almacenamiento. Por lo tanto, se debe administrar el almacenamiento cuidadosamente para evitar que se llene y se produzcan errores.

**Energía:** los sistemas embebidos suelen tener una fuente de alimentación limitada, como baterías o paneles solares. Por lo tanto, se debe gestionar cuidadosamente el uso de energía para evitar el agotamiento de la batería o la pérdida de energía en el sistema.

En resumen, en los sistemas embebidos se deben gestionar cuidadosamente los recursos de memoria, CPU, almacenamiento y energía para garantizar que el sistema funcione de manera eficiente y sin errores.

- **Mainframe:** en un mainframe, que es un tipo de sistema informático de alta capacidad de procesamiento y almacenamiento utilizado por grandes empresas e instituciones, los recursos que deben gestionarse incluyen: }

**Procesamiento de CPU:** los mainframes están diseñados para manejar grandes volúmenes de datos y procesamiento de transacciones. Por lo tanto, la administración del procesamiento de CPU es crítica para garantizar que el sistema funcione de manera eficiente y que los usuarios tengan acceso a los recursos de procesamiento necesarios.

**Almacenamiento:** los mainframes tienen una gran cantidad de almacenamiento, pero este almacenamiento también debe administrarse cuidadosamente. La administración del almacenamiento incluye la asignación de espacio de almacenamiento, la eliminación de datos obsoletos y la optimización del uso de almacenamiento.

**Memoria:** la administración de memoria en un mainframe es importante para garantizar un rendimiento óptimo y la eficiencia del sistema. La gestión de memoria incluye la asignación de memoria a los programas y la eliminación de programas que ya no se están utilizando para liberar espacio en la memoria.

**Redes y E/S:** los mainframes suelen estar conectados a una red y utilizan una gran cantidad de E/S para comunicarse con otros dispositivos. La gestión de redes y E/S incluye la asignación de recursos de red y E/S a los programas y la optimización del uso de la red y los dispositivos E/S para garantizar un rendimiento óptimo.

En resumen, en un mainframe se deben gestionar cuidadosamente los recursos de CPU, almacenamiento, memoria, redes y E/S para garantizar que el sistema funcione de manera eficiente y que los usuarios tengan acceso a los recursos necesarios. La gestión de estos recursos también es crítica para garantizar la seguridad y la disponibilidad del sistema.

- **Workstation:** en una Workstation, que es un tipo de ordenador de alto rendimiento utilizado para trabajos técnicos y de diseño, los recursos que deben gestionarse incluyen:

**Procesamiento de CPU:** las Workstation suelen tener procesadores potentes que se utilizan para procesamiento de gráficos, simulaciones y cálculos complejos. La administración del procesamiento de CPU es importante para garantizar un rendimiento óptimo y evitar la sobrecarga del procesador.

**Memoria:** las Workstation suelen tener una gran cantidad de memoria para manejar grandes volúmenes de datos y aplicaciones complejas. La gestión de memoria incluye la asignación de memoria a los programas y la eliminación de programas que ya no se están utilizando para liberar espacio en la memoria.

**Almacenamiento:** las Workstation suelen tener discos duros de alta capacidad y unidades de estado sólido (SSD) para almacenar grandes cantidades de datos. La gestión de almacenamiento incluye la asignación de espacio de almacenamiento, la eliminación de datos obsoletos y la optimización del uso de almacenamiento.

**Redes y E/S:** las Workstation suelen estar conectadas a redes y dispositivos externos, como impresoras y escáneres. La gestión de redes y E/S incluye la asignación de recursos de red y E/S a los programas y la optimización del uso de la red y los dispositivos E/S para garantizar un rendimiento óptimo.

En resumen, en una Workstation se deben gestionar cuidadosamente los recursos de CPU, memoria, almacenamiento, redes y E/S para garantizar un rendimiento óptimo y que los usuarios tengan acceso a los recursos necesarios. La gestión de estos recursos también es importante para garantizar la seguridad y la disponibilidad del sistema.

- **Server:** en un servidor, que es un ordenador utilizado para proporcionar servicios y recursos a otros dispositivos conectados en una red, los recursos que deben gestionarse incluyen:

**Procesamiento de CPU:** los servidores suelen tener procesadores potentes que se utilizan para realizar múltiples tareas simultáneamente, como ejecutar aplicaciones, gestionar bases de datos y manejar conexiones de red. La gestión del procesamiento de CPU es importante para evitar la sobrecarga del procesador y garantizar un rendimiento óptimo.

**Memoria:** los servidores suelen tener una gran cantidad de memoria para manejar grandes volúmenes de datos y múltiples aplicaciones. La gestión de memoria incluye la asignación de memoria a los programas y la eliminación de programas que ya no se están utilizando para liberar espacio en la memoria.

**Almacenamiento:** los servidores suelen tener grandes capacidades de almacenamiento para almacenar datos y archivos compartidos por múltiples usuarios. La gestión de almacenamiento incluye la asignación de espacio de almacenamiento, la eliminación de datos obsoletos y la optimización del uso de almacenamiento.

**Redes y E/S:** los servidores suelen estar conectados a redes y dispositivos externos, como impresoras y escáneres. La gestión de redes y E/S incluye la asignación de recursos de red y E/S a los programas y la optimización del uso de la red y los dispositivos E/S para garantizar un rendimiento óptimo.

**Seguridad:** la gestión de la seguridad es especialmente importante en un servidor, ya que los datos y recursos almacenados pueden ser sensibles y críticos. La gestión de la seguridad incluye la implementación de medidas de seguridad, como autenticación, autorización y cifrado, para proteger los datos y prevenir el acceso no autorizado.

En resumen, en un servidor se deben gestionar cuidadosamente los recursos de CPU, memoria, almacenamiento, redes y E/S para garantizar un rendimiento óptimo y que los usuarios tengan acceso a los recursos necesarios. Además, la gestión de la seguridad es crucial para proteger los datos y recursos almacenados en el servidor.

- **Mobile:** En los sistemas operativos móviles, algunos de los recursos que deben gestionarse son los siguientes:

**Batería:** Las baterías de los dispositivos móviles son limitadas y deben ser gestionadas cuidadosamente. El sistema operativo debe administrar el consumo de energía de las aplicaciones y procesos, limitando el uso de recursos como la CPU, la red y el GPS, por ejemplo, para prolongar la duración de la batería.

**Red:** El uso de redes móviles puede ser costoso en términos de consumo de energía y de datos móviles. El sistema operativo debe administrar el uso de la red para garantizar la eficiencia en el consumo de la batería y la asignación de ancho de banda de forma justa entre las aplicaciones.

**CPU y memoria:** En dispositivos móviles, la cantidad de memoria y el rendimiento de la CPU son limitados. El sistema operativo debe gestionar eficazmente la asignación de memoria y el uso de la CPU para garantizar un rendimiento óptimo y evitar cuellos de botella.

**Sensores:** Los dispositivos móviles modernos vienen con varios sensores, como GPS, acelerómetros, giroscopios, cámaras, etc. El sistema operativo debe gestionar el uso de los sensores para garantizar su eficiencia y evitar un consumo innecesario de la batería.

En resumen, en los sistemas operativos móviles, la gestión de los recursos de hardware, como la batería, la CPU, la memoria, los sensores y la red, es fundamental para garantizar un rendimiento óptimo y la duración de la batería. El sistema operativo debe gestionar eficazmente estos recursos para mejorar la experiencia del usuario en términos de eficiencia, duración de la batería y rendimiento.

### 3. Caracterice dos casos de uso para implementar un OS para servidor y PC.

Aquí se presentan dos casos de uso para implementar un sistema operativo (OS) tanto para servidores como para PC:

**Servicio de hosting web:** Un servidor dedicado que ofrece servicios de alojamiento web a clientes necesitaría un sistema operativo diseñado para manejar múltiples conexiones de red y procesamiento de aplicaciones en tiempo real. El sistema operativo debe estar diseñado para permitir la ejecución de múltiples aplicaciones en un servidor dedicado, lo que garantiza la escalabilidad y la capacidad de alojar varios sitios web en el mismo servidor. También es importante que el OS cuente con una gestión adecuada de recursos como la CPU, la memoria y el almacenamiento para asegurar una buena experiencia del

usuario en el sitio web y un alto nivel de disponibilidad y rendimiento del servicio. Por otro lado, para una PC utilizada para el desarrollo de sitios web, el OS debe ser fácil de usar y contar con una gran variedad de herramientas de desarrollo, para que el desarrollador pueda trabajar de manera eficiente y con una alta productividad.

**Gestión de bases de datos:** Un sistema operativo diseñado para servidores de bases de datos debe ser capaz de gestionar múltiples conexiones de red y la ejecución de consultas complejas de manera simultánea. El sistema operativo debe estar diseñado para ofrecer una alta disponibilidad, escalabilidad y seguridad. Por ejemplo, una empresa que gestiona grandes volúmenes de datos y depende de un servidor de base de datos para sus operaciones diarias requeriría un OS confiable y seguro, capaz de ofrecer redundancia y recuperación de datos en caso de fallos del hardware. Para la PC de un administrador de bases de datos, el OS debe ser fácil de usar y contar con una gran cantidad de herramientas de gestión de bases de datos, para permitir una gestión y mantenimiento eficiente y efectivo de la base de datos.

#### 4. Compare las diferencias entre multiprocesamiento simétrico y asimétrico.

El multiprocesamiento simétrico (SMP) y el multiprocesamiento asimétrico (AMP) son dos enfoques diferentes para el diseño de sistemas de multiprocesamiento, y se diferencian principalmente en cómo se comparten los recursos del sistema entre los procesadores.

En el multiprocesamiento simétrico (SMP), todos los procesadores comparten la misma memoria principal y los mismos dispositivos de entrada/salida, y pueden acceder a ellos simultáneamente. Cada procesador es igual y tiene la misma capacidad para realizar cualquier tarea, lo que significa que los procesos se pueden asignar a cualquier procesador disponible sin que haya una diferencia significativa en su capacidad de procesamiento. En un sistema SMP, cada procesador puede ejecutar un conjunto de tareas diferentes al mismo tiempo, y se utiliza una técnica llamada balanceo de carga para asegurar que el trabajo se distribuya uniformemente entre los procesadores. Los sistemas SMP son muy adecuados para aplicaciones intensivas en procesamiento, como servidores de bases de datos y aplicaciones de modelado y simulación.

En el multiprocesamiento asimétrico (AMP), los procesadores tienen diferentes capacidades y roles en el sistema. Por lo general, hay un procesador principal (también llamado "maestro") que se encarga de las tareas críticas del sistema, mientras que los demás procesadores (también llamados "esclavos") se utilizan para realizar tareas secundarias. Cada procesador en un sistema AMP tiene acceso a su propia memoria local, lo que significa que la comunicación y la sincronización entre los procesadores son más complejas que en un sistema SMP. Los sistemas AMP son adecuados para aplicaciones en las que se requiere un alto grado de especialización, como en sistemas de control de tiempo real o en dispositivos móviles donde se busca maximizar la eficiencia energética.

En resumen, la principal diferencia entre SMP y AMP es que en el primero, los procesadores son iguales y comparten recursos uniformemente, mientras que, en el segundo, los procesadores tienen roles y capacidades diferentes y tienen acceso a su propia memoria local.

#### 5. Enumere los requerimientos para que dos máquinas se junten en un cluster y provean un servicio de alta disponibilidad (HA).

Para que dos o más máquinas se junten en un cluster y provean un servicio de alta disponibilidad (HA), es necesario cumplir con los siguientes requerimientos:

**Hardware compatible:** Todas las máquinas que formarán parte del cluster deben tener hardware compatible y ser capaces de ejecutar el mismo sistema operativo.

**Conexión de red confiable:** Las máquinas deben estar conectadas por una red confiable que les permita comunicarse entre sí de manera rápida y confiable.

**Sistema operativo compatible:** Todas las máquinas que formarán parte del cluster deben ejecutar el mismo sistema operativo y tener la misma configuración.

**Software de cluster:** Se debe instalar y configurar el software de cluster apropiado en cada máquina que formará parte del cluster.

**Almacenamiento compartido:** Todas las máquinas del cluster deben tener acceso a un almacenamiento compartido que les permita acceder a los mismos datos.

**Configuración de tolerancia a fallos:** El software de cluster debe estar configurado para permitir la tolerancia a fallos y la recuperación automática en caso de que una máquina falle.

**Pruebas y monitoreo:** El cluster debe ser probado y monitoreado regularmente para asegurarse de que esté funcionando correctamente y proporcionando el nivel de disponibilidad esperado.

**Redundancia de recursos:** El cluster debe contar con recursos redundantes, como alimentación eléctrica y sistemas de enfriamiento, para asegurar que los servicios estén disponibles en todo momento.

## 6. Compare las diferencias entre una excepción y una interrupción.

Tanto las excepciones como las interrupciones son eventos que ocurren en un sistema informático y que pueden afectar su funcionamiento normal. Sin embargo, existen diferencias significativas entre ambas:

**Causa:** Una interrupción se produce cuando un dispositivo externo o una tarea del sistema solicita la atención inmediata de la CPU. Por otro lado, una excepción es generada por el propio programa en ejecución debido a un error en el código o una condición inesperada.

**Momento de ocurrencia:** Las interrupciones son eventos asincrónicos, es decir, que pueden ocurrir en cualquier momento durante la ejecución del sistema. En cambio, las excepciones son eventos síncronos, que ocurren en el momento en que se está ejecutando una instrucción específica.

**Control:** Cuando ocurre una interrupción, la CPU detiene temporalmente la tarea actual para manejar la solicitud del dispositivo o tarea externa. Después de completar la tarea solicitada, la CPU reanuda la tarea original. En cambio, cuando ocurre una excepción, el control del sistema es transferido al manejador de excepciones, quien determina cómo



manejar la situación y cómo continuar con la ejecución del programa.

**Tipos:** Las interrupciones pueden ser de diferentes tipos, como interrupciones de hardware, interrupciones de software o excepciones generadas por hardware. Las excepciones, por otro lado, se generan dentro del programa y pueden ser de diferentes tipos, como excepciones de división por cero, excepciones de desbordamiento de pila, excepciones de acceso a memoria no permitida, entre otras.

En resumen, las interrupciones son eventos externos que pueden ser generados por dispositivos o tareas del sistema, mientras que las excepciones son generadas por el propio programa debido a errores en el código o situaciones inesperadas. Además, las interrupciones ocurren de manera asincrónica, mientras que las excepciones son síncronas y ocurren durante la ejecución del programa. La forma en que se manejan también es diferente, ya que las interrupciones son manejadas por la CPU, mientras que las excepciones son manejadas por el programa y su manejador correspondiente.

## **7. El DMA (acceso directo a memoria) se usa en dispositivos I/O para evitar uso innecesario de la CPU.**

- **¿Como interactúa la CPU con el dispositivo para coordinar la transferencia?**

Cuando se utiliza DMA, la CPU se encarga de configurar el dispositivo de E/S (entrada/salida) y la memoria. La CPU es responsable de indicar al dispositivo de E/S la ubicación de la memoria donde debe leer o escribir los datos y el tamaño de la transferencia. Una vez que se ha configurado el dispositivo de E/S y se ha iniciado la transferencia de datos, la CPU no tiene que estar involucrada en la transferencia de datos en sí. El dispositivo de E/S utiliza el DMA para acceder directamente a la memoria y transferir los datos entre la memoria y el dispositivo.

La CPU solo se involucra en la transferencia cuando se produce una interrupción para notificar que la transferencia ha sido completada o que se ha producido algún error.

En resumen, la CPU se encarga de configurar la transferencia de datos utilizando DMA y luego permite que el dispositivo de E/S acceda directamente a la memoria para realizar la transferencia sin la necesidad de involucrar a la CPU en la transferencia en sí.

- **¿Como sabe la CPU que las operaciones de memoria se han completado?**

La CPU utiliza interrupciones para saber cuándo se han completado las operaciones de memoria realizadas mediante DMA. Cuando se inicia una transferencia de datos utilizando DMA, se configura una interrupción que se activa cuando se completa la transferencia. Una vez que se completa la transferencia, el dispositivo de E/S genera una interrupción que es recibida por la CPU.

La CPU, al recibir la interrupción, sabe que la transferencia de datos se ha completado y puede continuar con otras tareas. La interrupción también puede llevar información adicional, como si se produjo algún error durante la transferencia. En este caso, la CPU puede tomar medidas para corregir el error o notificar al usuario.

En resumen, la CPU utiliza interrupciones para saber cuándo se han completado las



operaciones de memoria realizadas mediante DMA. La interrupción es generada por el dispositivo de E/S una vez que se completa la transferencia y lleva información adicional sobre la operación realizada.

**8. Identifique dos razones por las que la cache es útil. ¿Qué problemas resuelve y causa?**

La caché es una memoria de acceso rápido que se utiliza para almacenar datos o instrucciones que se utilizan con frecuencia. Aquí están dos razones por las que la caché es útil:

**Mejora el rendimiento:** La caché puede mejorar el rendimiento del sistema al reducir el tiempo de acceso a los datos. En lugar de tener que buscar datos en la memoria principal, el procesador puede acceder rápidamente a los datos que se encuentran en la caché. Esto puede mejorar significativamente la velocidad de las operaciones del sistema.

**Reduce la carga de la memoria principal:** Almacenar datos en la caché reduce la cantidad de acceso a la memoria principal, lo que puede reducir la carga en la memoria principal y mejorar la eficiencia del sistema. Esto es particularmente útil en sistemas con una gran cantidad de procesos y/o con una cantidad limitada de memoria.

Sin embargo, la caché también puede causar algunos problemas. Dos de los principales son:

**Consistencia de los datos:** Si un dato se almacena en la caché y se modifica en la memoria principal, puede haber inconsistencia de los datos. Esto puede causar problemas si el procesador utiliza los datos almacenados en la caché en lugar de los datos actualizados en la memoria principal.

**Sobrecarga de la caché:** Si la caché se llena con datos que no se utilizan con frecuencia, puede reducir la eficiencia del sistema al limitar la cantidad de espacio disponible para datos más importantes.

**9. Explique con un ejemplo, como se manifiesta el problema de mantener la coherencia de los datos de cache en los siguientes entornos:**

- **Sistema distribuido**

La coherencia de los datos en un sistema distribuido se refiere a mantener los datos actualizados y consistentes en todos los nodos del sistema. En un sistema distribuido con múltiples caches, cada nodo tiene su propia cache local para acelerar el acceso a los datos.

El problema surge cuando varios nodos intentan acceder y modificar los mismos datos. Si un nodo modifica los datos, las copias de las caches de los otros nodos quedan desactualizadas, lo que puede llevar a inconsistencias en los datos. Por ejemplo, supongamos que en un sistema distribuido hay dos nodos que comparten una variable X en sus caches. El nodo A modifica la variable X y actualiza su cache, pero el nodo B aún tiene una copia de la variable desactualizada en su cache. Si el nodo B intenta acceder a la

variable, obtendrá la versión desactualizada, lo que puede causar errores y comportamientos inesperados.

Para evitar este problema, los sistemas distribuidos utilizan diferentes técnicas para mantener la coherencia de los datos de cache, como el protocolo de invalidación o el protocolo de actualización. En el protocolo de invalidación, cuando un nodo modifica una variable, envía un mensaje a los demás nodos para invalidar sus copias de la variable en sus caches. En el protocolo de actualización, cuando un nodo modifica una variable, envía un mensaje a los demás nodos para actualizar sus copias de la variable en sus caches.

En resumen, en un sistema distribuido con múltiples caches, el problema de mantener la coherencia de los datos de cache surge cuando varios nodos intentan acceder y modificar los mismos datos. Para evitar este problema, se utilizan diferentes técnicas para mantener la coherencia de los datos de cache, como el protocolo de invalidación o el protocolo de actualización.

- **Sistema multiprocesador**

En un sistema multiprocesador, cada procesador tiene su propia caché de datos. Si dos o más procesadores acceden a la misma ubicación de memoria, pueden almacenar copias de la misma ubicación en sus respectivas cachés. Si uno de los procesadores modifica el valor almacenado en su caché, los otros procesadores no lo saben y pueden seguir trabajando con la versión anterior de los datos almacenados en sus cachés. Esto puede causar problemas de coherencia de datos, ya que el sistema no tiene una visión global coherente de los datos.

Por ejemplo, si dos procesadores en un sistema multiprocesador intentan actualizar el mismo valor de una variable compartida al mismo tiempo, puede ocurrir un problema de coherencia de datos. Si ambos procesadores almacenan una copia de la variable en sus cachés, ambos podrían actualizar su copia local y luego escribir los nuevos valores en la memoria principal. Dependiendo del orden en que ocurran estas operaciones, puede que solo una de las actualizaciones se refleje en la memoria principal y la otra se pierda. Esto resultaría en una inconsistencia en los datos y podría llevar a resultados incorrectos en el sistema. Para evitar este problema, los sistemas multiprocesador utilizan protocolos de coherencia de caché que coordinan el acceso a las ubicaciones de memoria compartidas y aseguran que todas las copias de una ubicación de memoria sean coherentes entre sí. Los protocolos de coherencia de caché utilizan señales de control de hardware para asegurar que los procesadores estén al tanto de los cambios realizados por otros procesadores en la memoria compartida.

- **Sistema de un solo procesador**

En un sistema de un solo procesador, el problema de la coherencia de cache surge cuando hay varias copias de los mismos datos en diferentes niveles de la jerarquía de cache. Por ejemplo, supongamos que una aplicación se está ejecutando y necesita acceder a una variable llamada "X". El procesador primero buscará la variable en su cache L1 y, si no está allí, buscará en la cache L2 y así sucesivamente hasta que la encuentre en la memoria principal.

Si el valor de la variable "X" se actualiza en un nivel de la jerarquía de cache, la actualización no se reflejará automáticamente en los demás niveles de cache que contienen

copias de "X". Por lo tanto, si el procesador accede a "X" desde otro nivel de cache que no ha sido actualizado, la variable no será coherente y la aplicación podría experimentar resultados impredecibles o incorrectos.

Por ejemplo, si un procesador tiene una copia de la variable "X" en su cache L1 y otro procesador modifica el valor de "X" en su propia cache, la copia en el primer procesador ya no será válida. Para resolver este problema, se utiliza el protocolo de coherencia de cache, que garantiza que todas las copias de "X" en la jerarquía de cache sean actualizadas y sean coherentes entre sí.