

## Preguntas de teoría SS00 by @victor.rgez SuperPreguntas✓



➤ ¿Cómo funciona el copy on write en la gestión de la memoria? ¿Qué ventajas tiene? Preguntada 4 veces (beneficios en la generación de procesos hijos a partir de un proceso padre (fork).)

El copy on write permite la creación rápida de nuevos procesos al compartir inicialmente las páginas de memoria entre el proceso padre y el proceso hijo.

El copy on write funciona de esta forma:

1\* crea el nuevo proceso ( con fork() ), este proceso hijo comparte las páginas de memoria con el proceso padre.

2\* Todas las páginas compartidas se marcan como copia en escritura, marcando todas las páginas como de solo lectura en la tabla de páginas de ambos procesos. Y una tabla interna en el PCB indica cuales son de lectura y cuales de copiar escritura.

3\* Si algún proceso intenta escribir en una página copy on write, la MMU genera una excepción, y notifica el suceso al SO.

4\* Después de ser copiado, el proceso puede escribir en la página asignada sin afectar a otros procesos. Pero otras páginas sí marcadas pueden seguir compartidas.

Algunas de las ventajas son:

Minimización de la cantidad de páginas y marcos que deben ser asignadas a los nuevos procesos.

Y permite ahorrar memoria y tiempo en la creación de procesos porque solo se copian las páginas que son modificadas por estos.

➤ Explica brevemente los tipos de fragmentación que conoces y pon un ejemplo de cada uno de ellos. - O -¿Qué diferencia existe entre la fragmentación que se ocasiona por el mecanismo de paginación y la que se produce como consecuencia de la asignación de memoria dinámica al montículo? Preguntado 2 veces

En la asignación contigua existen dos tipos de fragmentación:

- **La fragmentación externa** : que ocurre cuando hay suficiente espacio libre para la solicitud de memoria, pero este espacio no es contiguo. Por ejemplo , hay varios bloques libres dispersos en el espacio de direcciones, pero ninguno es suficientemente grande para el tamaño que requiere un proceso, aunque la memoria total sea insuficiente, la falta de continuidad impide asignarla. Algunas de las soluciones sería permitir la comparación( mover para que haya espacio y quepa), o usar paginación o segmentación.
- **La fragmentación interna** : que se produce por la diferencia entre el espacio solicitado y el espacio asignado finalmente. Hay más memoria de la que realmente se ha solicitado, y de la que se va a usar. Por ejemplo , un proceso solicita 50 kB se le asigna 64 KB pero solo usa 45kb al final. La diferencia de espacio es la Fragmentación interna, que es de 19. ( 64kB - 45kB)

➤ **Explica brevemente cómo funcionan las listas de control de acceso en un sistema de archivos y las diferencias entre la versión general y la versión condensada. Preguntado 2 veces**

Las listas de control de acceso ( ACL ) en un sistema de archivos son mecanismos que determinan los permisos de acceso a archivos y directorios para usuarios y grupos. Y hay dos versiones , la general y la condensada.

- En la **versión general**, cada archivo o directorio tiene asociada una lista de control de acceso que especifica los nombre de usuario o grupos y los tipos de acceso permitidos para cada. Cuando un usuario solicita un acceso a un archivo, el sistema operativo verifica la ACL asociada, para saber si tienen permisos de acceso. Es bastante difícil de implementar y gestionar. Y aumenta la complejidad del sistema.
- En la **versión condensada**, para solucionar el problema de las ACL generales, se enfoca en que los usuarios , se clasifican en 3 : propietario, grupo y todos. Se usa un campo para cada clase de usuario, donde cada campo es un bit que permite o deniega el acceso asociado. Son más simples de gestionar que las generales, pues tienen una longitud fija y simplifican la estructura de la lista de control de acceso.

➤ **Comenta brevemente las características de los directorios con estructura de árbol y los directorios en grafo acíclico. ¿Cuál es la principal diferencia entre ellos?**

En la estructura de directorios en árbol, se permite una jerarquía de niveles. Cada árbol con su propio directorio raíz. Cada entrada de directorio incluye un bit que indica si apunta a un archivo o a un subdirectorio. Los nombres de rutas pueden ser absolutos , comenzando en la raíz y descendiendo hacia el archivo, o relativo, definiendo una ruta a partir del directorio.

En la estructura de directorios en Grafo Acíclico, es una generalización de la estructura en árbol y permite que archivos y subdirectorios existan simultáneamente en distintos lugares de la estructura de directorios. Permite compartir archivos, y se puede acceder a ellos desde el directorio de distintos usuarios. Puede tener múltiples rutas de acceso para cada archivo o directorio. Y puede ser implementado por enlaces Simbólicos (archivos con ruta relativa o absoluta de otro archivo o subdirectorio) o mediante enlaces Duros( duplicar toda la información de los archivos compartidos en todos los directorios que lo comparten)

La principal diferencia entre ellos está en la capacidad de compartir archivos. Mientras que en la estructura de árbol, los archivos y subdirectorios están organizados jerárquicamente con una sola ruta de acceso, en el acíclico , pueden existir simultáneamente en diferentes lugares de la estructura y pueden tener múltiples rutas de acceso.

➤ **Comenta brevemente los algoritmos de buffering de páginas y sus ventajas**  
**Preguntado 2 veces**

Los algoritmos de buffering de páginas se usan en el contexto de la administración de la memoria virtual para mejorar la eficiencia del sistema. Mejoran la eficiencia de los algoritmos de reemplazo. Hay varias opciones:

★ Se mantiene una lista de marcos libres en memoria. Cuando se produce un fallo de página, se elige un marco de la lista y se carga la página en ese marco, al mismo tiempo que se selecciona otra como víctima, que se copia en el disco. Esto para que el proceso se reinicie lo antes posible, sin esperar a que la reemplazada sea escrita en el disco. Cuando finalice, el marco se incluye en la lista de marcos libres. La escritura se hace en segundo plano, mejorando la eficiencia.

★ Mediante registros de páginas en Marcos libres. Que mejora al procedimiento anterior, pues recuerda qué página estuvo en cada marco antes de pasar a la lista de marcos libres. Las páginas pueden recuperarse directamente desde la lista si falla alguna antes de que su marco sea usado por otra página. Reduce la necesidad de cargarlas desde el disco

★ También se puede mantener una lista de páginas modificadas e ir escribiéndose cuando el dispositivo del espacio de intercambio no esté ocupado. Aumenta la probabilidad de que una página esté limpia cuando sea seleccionada por el algoritmo de reemplazo, evitando la escritura en el disco. Minimiza las escrituras en el disco y aumenta las páginas ya listas para ser reemplazadas, lo cual mejora la eficiencia.

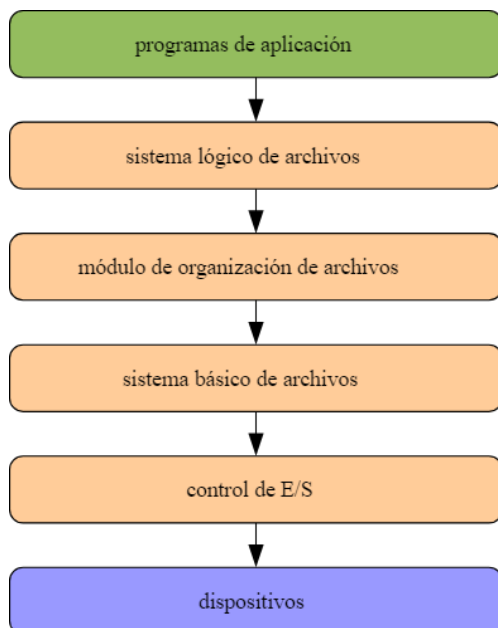
➤ **Explica qué es la semántica de coherencia y comenta dos ejemplos**  
**Preguntado 4 veces**

La semántica de coherencia se refiere a la manera en que las modificaciones realizadas por un proceso en archivos son observables por otros procesos. Se usa en sistemas operativos donde varios procesos pueden acceder y modificar los mismos archivos al mismo tiempo. La semántica de coherencia establece las reglas y garantías en cuanto a la visibilidad y consistencia de los cambios realizados en un archivo por diferentes procesos. Ejemplos:

**Semánticas POSIX:** Las escrituras son inmediatamente visibles para otros usuarios que hayan abierto el mismo archivo. Existe un modo de compartición que permite a los procesos compartir el **puntero de ubicación** en el archivo, afectando a todos los procesos que compartan el archivo.

**Semántica de sesión (AFS):** Las escrituras no son visibles inmediatamente para otros usuarios. Los cambios en un archivo son visibles solo en sesiones que comiencen después de cerrar el archivo. Permite múltiples accesos concurrentes, evitando retardos.

➤ **Explica brevemente los niveles que suelen componer la estructura de un sistema de archivos.**



La estructura de un sistema de archivos se compone de varios niveles. En primer lugar, el control de E/S incluye controladores de dispositivo que facilitan la transferencia de información entre la memoria principal y el disco en bloques. A continuación, el sistema básico de archivos envía comandos al controlador de dispositivo para leer y escribir bloques físicos en el disco.

El módulo de organización de archivos gestiona la traducción de direcciones lógicas a direcciones físicas de bloques, permitiendo la coherencia entre la organización lógica y física de los archivos. Por último, el sistema lógico de archivos administra los metadatos, como la estructura de directorios y los bloques de control de archivo (FCB), que contienen información sobre el archivo, incluyendo propietario, permisos y ubicación del contenido.

Este sistema también se encarga de tareas relacionadas con la protección y seguridad de los datos. Cada sistema operativo puede admitir uno o más sistemas de archivos para dispositivos de disco, lo que proporciona una gestión eficiente y segura de la información almacenada.

➤ **Explica brevemente los niveles RAID que conozcas.**

La tecnología RAID (Redundant Array of Inexpensive Disks) permite combinar discos para mejorar rendimiento o fiabilidad. Niveles comunes incluyen:

**RAID 0:** Distribuye datos entre discos sin redundancia. Mejora el rendimiento, pero la fiabilidad disminuye con más discos.

**RAID 1:** Crea copias exactas en espejo. Fiabilidad alta, mejora rendimiento linealmente con copias.

**RAID 5:** Distribuye datos y utiliza paridad para tolerancia a fallos. Escrituras costosas, pérdida total si fallan dos discos.

**RAID 6:** Similar a RAID 5, pero con doble paridad para mayor tolerancia a fallos.

Existen niveles anidados como RAID 0+1, RAID 1+0 y RAID 50. Pueden implementarse en hardware o software. Cada conjunto RAID se comporta como una unidad independiente, y su uso puede dividirse o combinarse según necesidades.

➤ **Explica qué es la hiperpaginación y por qué se produce.**

La hiperpaginación es cuando un proceso gasta más tiempo en realizar operaciones de paginación (trasladar datos entre la memoria principal y secundaria) que en ejecutarse. Se produce debido a la falta de suficientes marcos de página en la memoria principal para soportar las necesidades de los procesos en ejecución.

En sistemas multiprogramados más antiguos que implementaron la paginación bajo demanda, la hiperpaginación ocurre por un ciclo iterativo que involucra la monitorización del uso de la CPU. Si el uso de la CPU era bajo, el sistema operativo cargaba más procesos para aumentar la multiprogramación. Pero, si un proceso requería demasiada memoria, podía desplazar marcos de página de otros procesos, lo que incrementa la tasa de fallos de página del proceso afectado. A medida que aumentaban los fallos de página, disminuía el uso de la CPU. En respuesta, el sistema operativo cargaba más procesos para aumentar la multiprogramación, lo que llevaba a una reducción en la cantidad de memoria disponible para cada proceso y generaba un ciclo que baja el rendimiento.

En SO modernos, la hiperpaginación puede ocurrir de manera similar, aunque sin el efecto en cadena provocado por la optimización del uso de la CPU por parte del planificador a largo plazo. En ambos casos, la hiperpaginación se resuelve reduciendo el grado de multiprogramación, liberando así memoria para los procesos y mejorando el rendimiento global del sistema.

➤ **¿Cuáles son los requerimientos básicos para implementar un sistema de memoria virtual con paginación bajo demanda? (estructuras de datos necesarias, capacidades del hardware, etcétera) ¿Qué ventaja puede suponer en el algoritmo de reemplazo hacer uso del bit de página modificada?**

Para implementar un sistema de memoria virtual con paginación bajo demanda, se requieren:

**Tabla de Páginas:** Una estructura que rastree las páginas en memoria y su estado.

**Dispositivo de Almacenamiento Secundario:** Un disco para almacenar páginas no presentes en memoria (swap).

**Capacidad de Reiniciar Instrucciones:** La CPU debe manejar interrupciones por fallos de página y reiniciar la ejecución cuando la página necesaria está en memoria.

El uso del bit de "página modificada" en el algoritmo de reemplazo puede mejorar el rendimiento, indicando si una página ha sido modificada desde su carga. Esto optimiza el proceso de reemplazo al considerar la necesidad de escribir la página de vuelta al disco.

### ➤ **Explica por qué los Sistemas Unix tienen estructura de grafo acíclico**

En los Sistemas Unix, la estructura de directorios sigue un grafo acíclico para evitar enlaces simbólicos que generan ciclos. Esto ofrece ventajas como evitar la recursión infinita, simplificar la resolución de rutas, mejorar la seguridad y consistencia del sistema de archivos, y facilitar operaciones como la copia y el respaldo de directorios. La elección se basa en principios de simplicidad, eficiencia y coherencia en la gestión del sistema de archivos.

### ➤ **¿En la planificación de procesos, que es un cambio de Contexto? ¿Qué operaciones se realizan y que debe ser almacenado?**

El cambio de contexto en la planificación de procesos es la transición entre la ejecución de dos procesos en la CPU. En este proceso, se guardan los datos del estado del proceso saliente, como registros de la CPU y el contador de programa, y se cargan los datos del nuevo proceso entrante. Esta operación es esencial para la multitarea, permitiendo que múltiples procesos compartan el tiempo de la CPU. Durante el cambio de contexto, también se realizan ajustes en la tabla de páginas y otros recursos del sistema. La eficiencia en la ejecución de estas operaciones es crucial para el rendimiento general del sistema operativo. Aunque implica cierto costo en términos de tiempo y recursos, un cambio de contexto eficiente contribuye a la apariencia de ejecución simultánea de múltiples procesos.

### ➤ **Enumera y define brevemente los metadatos almacenados en disco que son requeridos por un sistema de archivos**

En un sistema de archivos, los metadatos almacenados en disco son cruciales para su funcionamiento. Algunas estructuras comunes son:

★**Bloque de Control de Arranque (BCA):** Contiene información para iniciar el sistema operativo desde un volumen.

★**Bloque de Control de Volumen (BCV o Superbloque):** Almacena detalles del volumen, como número máximo de bloques y espacio libre. Se llama superbloque en sistemas UNIX y Linux, y en NTFS, esta información está en la Tabla Maestra de Archivos (MFT).

★**Bloque de Control de Archivo (BCA o Inodo):** Guarda información detallada de cada archivo, como permisos y ubicación de bloques de datos. Se llama inodo en sistemas UNIX y Linux, y en NTFS, reside en la MFT.

★**Estructura de Directorios:** Organiza archivos. En sistemas UNIX y Linux, cada directorio es un archivo especial que almacena nombres e índices de archivos. En NTFS, toda la estructura de directorios está en la MFT.

Estas estructuras son clave para la gestión del sistema de archivos, desde el arranque hasta los detalles específicos de archivos y directorios.

**➤ Comenta ampliamente lo que sepas sobre el método básico de la paginación, soporte hardware de la tabla de páginas y protección.**

El método básico de paginación es una técnica de gestión de memoria que resuelve el problema de la fragmentación externa dividiendo tanto la memoria física como el espacio de direcciones virtuales en bloques de tamaño fijo llamados marcos y páginas. Cuando un proceso se ejecuta, sus páginas se cargan desde el almacenamiento secundario en marcos libres de la memoria física. Cada dirección virtual generada por la CPU se divide en un número de página y un desplazamiento, y mediante la tabla de páginas, se realiza la traducción a la dirección física correspondiente.

El soporte hardware de la tabla de páginas implica el uso de registros de la CPU o almacenamiento en memoria para mantener información sobre los marcos asignados, disponibles y su correspondencia con las páginas de los procesos. Además, la TLB actúa como una caché de traducciones rápidas para mejorar el rendimiento de la paginación.

En cuanto a la protección, se implementa mediante el establecimiento de permisos en las entradas de la tabla de páginas. Los bits de protección indican si una página es de lectura, escritura o ejecución, proporcionando control sobre el acceso a la memoria. La MMU comprueba que el tipo de acceso es válido. Si no, se genera una excepción. También se le añade un bit de válido, que indica si la página asociada está en el espacio de direcciones. Para no tener que almacenar una tabla de página completa con una entrada para cada página del espacio de direcciones, existe el PTLR (Registro de longitud de tabla de página) que se utiliza para indicar el tamaño real de la tabla de página.

**➤ Comenta ampliamente lo que sepas sobre los volúmenes de datos, incluyendo ejemplos de niveles RAID que conozcas. Preguntado 2 veces**

Los volúmenes de datos son estructuras de almacenamiento que contienen sistemas de archivos y pueden abarcar tanto partes específicas de un dispositivo de almacenamiento como estructuras más amplias que combinan múltiples dispositivos. Estos volúmenes permiten organizar y gestionar la información de manera eficiente, proporcionando una forma estructurada de acceder y almacenar datos.

La tecnología RAID (Redundant Array of Inexpensive Disks) permite combinar discos para mejorar rendimiento o fiabilidad. Niveles comunes incluyen:

**RAID 0:** Distribuye datos entre discos sin redundancia. Mejora el rendimiento, pero la fiabilidad disminuye con más discos.

**RAID 1:** Crea copias exactas en espejo. Fiabilidad alta, mejora rendimiento linealmente con copias.

**RAID 5:** Distribuye datos y utiliza paridad para tolerancia a fallos. Escrituras costosas, pérdida total si fallan dos discos.

**RAID 6:** Similar a RAID 5, pero con doble paridad para mayor tolerancia a fallos.

Existen niveles anidados como RAID 0+1, RAID 1+0 y RAID 50. Pueden implementarse en hardware o software. Cada conjunto RAID se comporta como una unidad independiente, y su uso puede dividirse o combinarse según necesidades

### ➤ **Explica las ventajas e inconvenientes del reemplazo global y local**

#### ★ **Reemplazo Local:**

##### **Ventajas:**

- El número de marcos asignados a un proceso permanece constante a lo largo del tiempo, lo que brinda estabilidad en el rendimiento y evita fluctuaciones inesperadas.
- Los procesos tienen control sobre sus propios marcos y no se ven afectados por el comportamiento de otros procesos. Pueden gestionar su memoria de manera predecible.

##### **Inconvenientes:**

- No se pueden liberar marcos poco utilizados para ser compartidos con otros procesos, lo que puede llevar a un mal uso de la memoria.
- Un proceso no puede ajustar dinámicamente la cantidad de memoria asignada según sus necesidades cambiantes.

#### ★ **Reemplazo Global:**

##### **Ventajas:**

- Permite la selección de marcos de memoria de cualquier proceso, lo que facilita una asignación más eficiente y puede mejorar el rendimiento general.
- Los procesos pueden beneficiarse de la disponibilidad de marcos de memoria independientemente de su asignación inicial, lo que permite adaptarse a situaciones de carga variable.

##### **Inconvenientes:**

- El número de marcos asignados a un proceso puede cambiar debido a los fallos de página de otros procesos, lo que puede introducir variabilidad en el rendimiento.
- Los procesos no tienen control directo sobre la cantidad de memoria asignada, ya que está sujeta a cambios impulsados por el sistema operativo y el comportamiento de otros procesos.



**➤ Explica brevemente cómo funcionan las listas de control de acceso en un sistema de archivos**

En un sistema de archivos, las listas de control de acceso (ACL) son mecanismos que permiten gestionar y regular el acceso a archivos y directorios por parte de usuarios y grupos. La implementación de ACL es crucial en entornos multiusuario para garantizar la seguridad y la protección de los datos. Hay diversas maneras de implementarlo: Cuando un usuario solicita acceder a un archivo concreto, el sistema operativo comprueba la ACL asociada a dicho archivo. Si el usuario, o alguno de sus grupos, está incluido en la lista para el tipo de acceso solicitado, se permite el acceso.

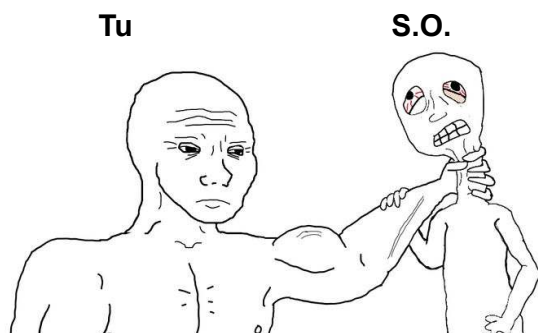
**➤ Explica el modelo del conjunto de trabajo y su relación con el problema de la hiperpaginación. Preguntado 3 veces**

El modelo del conjunto de trabajo es una estrategia utilizada para abordar el problema de la hiperpaginación en sistemas operativos. La hiperpaginación ocurre cuando un proceso gasta más tiempo paginando (moviendo páginas entre la memoria principal y el almacenamiento secundario) que ejecutándose, lo que resulta en un mal rendimiento del sistema.

El parámetro  $\Delta$  define el tamaño de la ventana del conjunto de trabajo. El sistema asigna marcos según el conjunto de trabajo de cada proceso, adaptándose a los cambios en su localidad a lo largo del tiempo. Esto evita la hiperpaginación, donde el gasto en paginación supera la ejecución.  $\Delta$  debe equilibrarse para garantizar eficiencia y evitar superposiciones. El sistema puede suspender procesos si la demanda total de marcos excede la disponibilidad, reiniciándolos más tarde. En resumen, el modelo gestiona eficientemente la memoria al considerar patrones de acceso de procesos.

**➤ Enumera los metadatos de un sistema de gestión de almacenamiento, el cometido de cada uno, y si son almacenados en el disco o no  
Preguntado 2 veces**

Los metadatos son datos que proporcionan información sobre otros datos. En los sistemas de gestión de almacenamiento, son esenciales para organizar, gestionar y acceder a la información almacenada. Estos datos ofrecen detalles sobre la estructura, atributos y ubicación de los archivos y directorios en un sistema.



★ **En el disco** hay distintas estructuras de metadatos:

**Bloque de control de arranque** que suele ocupar el primer bloque de cada volumen y que contiene la información necesaria para iniciar el SO

**El bloque de control de volumen** contiene todos los detalles acerca el número máximo de bloques, el tamaño de los bloques, el número de libres ...

**Bloque de control de archivo** Un FCB por cada archivo. Todos los FCB del sistema de archivos se almacenan en una tabla denominada directorio de dispositivo o tabla de contenidos del volumen.

**Estructura de directorios** Para organizar los archivos.

★ Y por otro lado los almacenados **en memoria**:

**Una tabla de montaje en memoria** que contiene información acerca de cada volumen montado.

**Una caché en memoria** de la estructura de directorios que almacena la información relativa a los directorios a los que se han accedido recientemente. Los directorios que actúan como puntos de montaje pueden contener un puntero a la entrada, en la tabla de montaje, del volumen montado en el directorio.

**La tabla global de archivos abiertos**, que contiene una copia del FCB de cada archivo abierto en el sistema.

**La tabla de archivos abiertos de cada proceso**, que contiene un puntero a la entrada apropiada de la entrada global de archivos abiertos, así como otras informaciones adicionales.

➤ **En un sistema de memoria virtual, ¿qué problema pretende resolver un algoritmo de reemplazo de páginas? ¿Por qué es crítico reducir la tasa de fallos de página?**

En un sistema de memoria virtual, un algoritmo de reemplazo de páginas aborda el problema de gestionar la escasez de memoria física cuando no hay suficientes marcos disponibles para alojar todas las páginas requeridas por los procesos en ejecución. La memoria virtual permite que los programas accedan a más memoria de la que está físicamente presente, utilizando espacio en disco como extensión de la memoria RAM.

Un fallo de página ocurre cuando un programa intenta acceder a una página que no está en la memoria principal (RAM) y debe cargarse desde el almacenamiento secundario (disco). Es crítico reducir las tasas de fallo de página porque este proceso es costoso en términos de tiempo y recursos, ya que implica acceder a dispositivos de almacenamiento más lentos, lo cual mejora la eficiencia y optimiza los recursos. Por lo tanto, un algoritmo de reemplazo de páginas eficiente se esfuerza por minimizar la frecuencia de fallos de página.

**➤ Explica las características de los directorios en grafo acíclico (comenta los enlaces hardlinks y simbólicos en Unix) y razona por qué es más habitual utilizar la estructura de grafo acíclico en lugar de un grafo genera. Preguntado 2 veces**

En la estructura de directorios en Grafo Acíclico, es una generalización de la estructura en árbol y permite que archivos y subdirectorios existan simultáneamente en distintos lugares de la estructura de directorios. Permite compartir archivos, y se puede acceder a ellos desde el directorio de distintos usuarios. Puede tener múltiples rutas de acceso para cada archivo o directorio. Y puede ser implementado por enlaces Simbólicos (archivos con ruta relativa o absoluta de otro archivo o subdirectorio) o mediante enlaces Duros( duplicar toda la información de los archivos compartidos en todos los directorios que lo comparten)

Esto ofrece ventajas como evitar la recursión infinita, simplificar la resolución de rutas, mejorar la seguridad y consistencia del sistema de archivos, y facilitar operaciones como la copia y el respaldo de directorios. La elección se basa en principios de simplicidad, eficiencia y coherencia en la gestión del sistema de archivos. Por eso es más usado.

**➤ Describe qué información se almacena en los denominados FCB (bloque de control de archivo)**

El Bloque de Control de Archivo ("File Control Block") es una estructura de metadatos utilizada por el sistema de archivos para gestionar y mantener información relacionada con un archivo específico. Los FCB contienen diversos detalles sobre el archivo, permitiendo al sistema operativo y a los programas acceder y manipular la información de manera eficiente. La información almacenada en un FCB típicamente incluye:

- Nombre simbólico del archivo
- Identificador del archivo dentro del sistema de archivos
- Tipo de archivo
- Ubicación (puntero a un dispositivo y a la ubicación de los bloques con los datos)
- Tamaño del archivo
- Protección (quién puede leerlo, escribirlo, ejecutarlo)
- Fecha, hora e identificación del usuario

➤ **Comenta ampliamente lo que sepas sobre paginación y el concepto de espacio de direcciones virtuales. Incluye en la explicación las principales ventajas de este sistema y los requerimientos en el hardware y las estructuras de datos necesarias**

El método básico de paginación es una técnica de gestión de memoria que resuelve el problema de la fragmentación externa dividiendo tanto la memoria física como el espacio de direcciones virtuales en bloques de tamaño fijo llamados marcos y páginas. Cuando un proceso se ejecuta, sus páginas se cargan desde el almacenamiento secundario en marcos libres de la memoria física. Cada dirección virtual generada por la CPU se divide en un número de página y un desplazamiento, y mediante la tabla de páginas, se realiza la traducción a la dirección física correspondiente.

En cuanto al espacio de direcciones virtuales, es el rango de direcciones que un proceso puede utilizar durante su ejecución. Cada proceso tiene su propio espacio de direcciones virtuales independiente. Se compone de secciones como el código, datos, montón y pila, cada una con su propio propósito y características. Es importante pues permite a los procesos tener la ilusión de contar con una memoria contigua y completa, independientemente de cómo esté distribuida realmente en la memoria física. Facilita la gestión de la memoria.

➤ **Comenta ampliamente lo que sepas sobre la evolución de la estructura de directorios, incluyendo cada uno de los tipos (directorios de un nivel, de dos niveles, en árbol, etc.).**

La evolución de la estructura de directorios ha sido un aspecto crucial en el desarrollo de los sistemas de archivos. Cada tipo de estructura ha buscado abordar desafíos específicos en la gestión de archivos y la colaboración entre usuarios.

★ En los **directorios de un nivel** : Todos los archivos están contenidos en un único directorio. Cuando el número de usuarios del sistema aumenta se hace más difícil que cada uno escoja nombres diferentes para sus archivos

★ En los **directorios de dos niveles**: cada **usuario** tiene su propio directorio de archivos de usuarios, esta estructura tiene el inconveniente de aislar a los usuarios.

★ **Directorios en árbol**: permite la creación de subdirectorios por parte de los usuarios. Cada árbol tiene un directorio raíz que puede contener archivos y otros directorios. Cada entrada de directorio indica si apunta a un archivo (0) o a un subdirectorio (1). Los nombres de ruta pueden ser absolutos (desde la raíz) o relativos (desde el directorio actual). Esta estructura posibilita el acceso de usuarios a los archivos de otros usuarios.

★ **directorios en Grafo Acíclico**, es una generalización de la estructura en árbol y permite que archivos y subdirectorios existan simultáneamente en distintos lugares de la estructura de directorios. Permite compartir archivos, y se puede acceder a ellos desde el directorio de distintos usuarios. Puede tener múltiples rutas de acceso para cada archivo o directorio. Y puede ser implementado por enlaces Simbólicos (archivos con ruta relativa o absoluta de otro archivo o subdirectorio) o mediante enlaces Duros( duplicar toda la información de los archivos compartidos en todos los directorios que lo comparten)

★**Directorios de grafo general**. Uno de los principales problemas de la estructura de directorios en grafo acíclico es garantizar que no exista ningún ciclo, esta estructura lo verifica mediante algoritmos. Sin embargo es caro, y lo mejor es evitar que puedan haber múltiples referencias a un mismo directorio y ya.

➤ **Explica brevemente en qué consiste el enlazado dinámico y qué ventajas e inconvenientes presenta**

El enlazado dinámico es un proceso que ocurre durante la carga o ejecución de un programa, en el cual las referencias a las librerías externas se resuelven en tiempo de ejecución en lugar de estar completamente resueltas durante la fase de compilación o enlazado estático. En este proceso, las librerías de enlace dinámico son cargadas dinámicamente en la memoria cuando son requeridas por el programa.

Algunas de las ventajas son el menor uso de memoria al cargar solo las librerías necesarias cuando se ejecutan. Permite actualizar las librerías compartidas sin tener que recompilar todo el programa. Acelera el tiempo de inicio de un programa. En cuanto a los inconvenientes, el programa depende de la existencia de las librerías en el sistema en tiempo de ejecución. Es más complejo, y puede empeorar el rendimiento por sobrecarga.



---

**Si estas preguntas te han servido para estudiar, agradecería un follow en mis redes:**

<https://www.instagram.com/victor.rgez/>

<https://github.com/alu0101540153>

**No dejes de seguir mirando mis apuntes, para apoyarme. Seguro que apruebas <3 Mucho ánimooo !**

---

