

Parcial 2 - Investigación

Deadlocks

1. ¿Qué es un deadlock y bajo qué condiciones puede ocurrir?

Los procesos de un sistema operativo multi-procesos suelen competir por los recursos disponibles. Cuando un proceso solicita un recurso, si este no está disponible, el proceso entra a un estado de espera hasta que el recurso esté disponible para su uso. A veces el proceso nunca sale del estado de espera, debido a que el recurso requerido está siendo retenido por otro proceso en estado de espera. A esta situación se le llama deadlock.

Un deadblock, entonces, se produce si más de un proceso retiene un recurso y solicita el de otro. Provocando que un proceso A dependa de los recursos de otro proceso B y, a su vez, el proceso B requiera de los recursos del proceso A. Puede parecer que es un error de programación en lugar de un error del diseño del sistema operativo, pero no lo es. Una posible solución a los deadlocks es imponer restricciones en el diseño del sistema sobre el orden en el que se solicitan los recursos.

2. Liste las condiciones que se pueden atacar para prevenir deadlocks (deadlock prevention) y explique los protocolos que se proponen para lograrlo. En sus explicaciones incluya alternativas, descripciones breves, ventajas y desventajas. No incluya ejemplos.

Las condiciones necesarias para que un deadlock ocurra son 4: *Mutual Exclusion, Hold and Wait, No Preemption y Circular Wait*. Por lo que cada una de estas situaciones debe ser manejada de la siguiente manera:

- Mutual Exclusion: sucede cuando al menos un recurso debe ser retenido de manera no compartible, por lo que solo un proceso puede usar el recurso al mismo tiempo. Si otro proceso solicita el recurso, el proceso debe esperar a que sea liberado. Esta condición debe manejarse con que la condición de exclusión mutua debe mantenerse. Es decir, que al menos un recurso no se debe compartir. Debido a que los recursos compartidos, no requieren de acceso mutuo exclusivo y por lo tanto no implican deadlock. Una de la ventaja de prevenir Mutual Exclusion, es que se hace uso de recursos compartibles que evitan los deadlocks. Mientras que una desventaja es que existen recursos que no pueden ser compartibles, y que son más propensos a generar deadlocks
- Hold and Wait: esto ocurre cuando un proceso debe retener al menos un recurso y esperar para adquirir un nuevo recurso que está siendo utilizado por otro proceso. Para controlar esto, el sistema operativo debe garantizar que un proceso realice una solicitud del recurso cada vez que lo necesite y que este no este utilizando otro recurso. Un protocolo que es utilizado consiste en que cada proceso solicite y asigne todos sus recursos antes de empezar su ejecución. Otro protocolo, consiste en que un proceso solicite un recurso solo cuando no tiene ninguno. Una desventaja de estos protocolos es *starvation*, debido a que, si un proceso requiere varios recursos, este debe esperar mucho para ejecutarse. Mientras que una ventaja es que los procesos no retendrían recursos que no están siendo utilizados, dejando los recursos libres para otros procesos.
- No preemption: sucede cuando los recursos no pueden ser *preempted*, lo que significa que el recurso solo puede ser liberados voluntariamente mediante el proceso que lo está utilizando, lo que sucede cuando el proceso completa su ejecución. Para evitar este suceso, podemos aplicar el siguiente protocolo, en donde si un proceso que está reteniendo un recurso hace la solicitud de un nuevo recurso, y el nuevo recurso está disponible, el sistema operativo puede realizar la liberación del recurso retenido antes de otorgarle el nuevo recurso al proceso. Así el recurso liberado es agregado a la lista de los recursos que el proceso está esperando y puede ser reiniciado con los recursos antiguamente solicitados, así como los nuevos. Entre las ventajas encontramos que los procesos podrán utilizar todos los recursos, al tener una lista con los recursos necesarios, incluso con los que ya utilizaron. Mientras que la desventaja es que algunos recursos no son capaces de guardar el estado de utilización, para que pueden ser continuados después.
- Circular wait: sucede cuando en un conjunto de procesos, el primer proceso está esperando por los recursos del segundo proceso, el segundo espera por el tercero, así sucesivamente, hasta que el último proceso espera

por los recursos del primer proceso. Creando básicamente un círculo de dependencia que nunca termina. Para evitar que esto suceda, se puede implementar un orden total de todos los recursos disponibles y solicitar que cada proceso realice solamente peticiones de recursos de orden ascendente en el orden de los recursos. De esta manera, cada proceso solo puede pedir recursos de diferente orden. Una desventaja de esta implementación es que el orden asignado a los recursos puede causar deadlock si se realiza incorrectamente. Mientras que la ventaja es que se tiene un orden para realizar la petición de recursos y de esta manera de organizan los recursos.

3. ¿Cuál es la motivación para emplear deadlock avoidance en lugar de deadlock prevention?

Los algoritmos de deadlock prevention evitan los deadlock al limitar la manera en que se solicitan los recursos. Lo que implica que al menos una de las condiciones necesarias no pueda ocurrir. Sin embargo, la prevención de los deadlock causa efectos secundarios que causan bajas en el rendimiento del sistema. Por lo que la solución propuesta es la utilización de deadlock avoidance, debido a que un método de deadlock avoidance es menos estricto que un algoritmo de deadlock prevention y este requiere más información sobre cómo se utiliza el recurso. Aunque deadlock avoidance presenta mayor complejidad, presenta ser más seguro en la detección de deadlock sin afectar al sistema y solamente solicitando más información.

a. Describe los conceptos de secuencia segura y estado seguro.

Cuando un sistema se encuentra en estado seguro, se está diciendo que el sistema es capaz de asignar todos sus recursos a los procesos, con algún orden, y aun así no causar algún deadlock. Formalmente, un sistema está en estado seguro solo si existe una secuencia segura. Siendo una secuencia segura para un proceso la solicitud de recursos en donde esta puede ser satisfecha con los recursos disponibles, más los recursos que ya están siendo utilizados por el proceso (no causa deadlock).

b. Lea sobre el algoritmo del banquero para asignación de recursos. Luego, resuelva este ejercicio:

i. ¿Qué contiene la matriz Need?

Los valores de Need son:

A, B, C, D

P0 (0, 0, 0, 0)

P1 (0, 7, 5, 0)

P2 (1, 0, 0, 2)

P3 (0, 0, 2, 0)

P4 (0, 6, 4, 2)

ii. ¿Estaría este sistema en un estado seguro, en este instante?

Si está en un estado seguro, con Avialable en (1,5,2,0) el proceso P3 o P0 podría ejecutarse. Cuando el proceso termine de ejecutarse, los recursos se liberan y los otros procesos podrían ejecutarse.

iii. ¿Si el proceso P1 solicita 0 instancias del recurso A, 4 del recurso B, 2 del recurso C y 0 del recurso D, ¿podrían asignársele los recursos solicitados inmediatamente?

Si la solicitud de P1 cambia a (0,4,2,0), si se le puede asignar recursos inmediatamente. Dejando el orden de finalización como: P0, P2, P3, P1 Y P4.

4. Si, en un sistema, no se emplean métodos de deadlock prevention ni deadlock avoidance, ¿qué dos cosas quedan por hacer? Describa brevemente un acercamiento a cada una.

Si un sistema no emplea deadlock prevention ni deadlock avoidance, entonces puede ocurrir un deadlock. Por esta razón el sistema debe proveer alguno de los siguientes:

- Un algoritmo que examina el estado del sistema para determinar si se ha producido deadlock: si todos los recursos tienen una sola instancia, es posible implementar una instancia del grafo *wait-for*. El cual ayuda a identificar si existe un deadlock identificando si existen ciclos dentro del grafo y los evita construyendo el grafo *wait-for*. Sin embargo, una desventaja es que este esquema no funciona en sistemas con múltiples instancias de cada recurso. En donde el algoritmo que se emplea debe ser parecido al algoritmo del Banquero.

- Un algoritmo para recuperarse del deadlock: cuando un algoritmo determina que se ha provocado un deadlock, se pueden realizar varias acciones. Una opción es permitir al sistema recuperarse automáticamente del deadlock. Hay dos maneras de romper el deadlock:
 - Abortar todos los procesos del deadlock: este método claramente eliminara el deadlock, pero a un alto precio. Debido a que el proceso posiblemente se ejecutó por bastante tiempo y se descarta sus resultados. Y se deba computar en el futuro.
 - Abortar un proceso a la vez hasta que el deadlock es eliminado: este método induce al overhead, debido a que cada vez que se aborta un proceso, se debe ejecutar un algoritmo que determine si existe deadlock y luego de la ejecución de algoritmo se puede continuar con la computación del proceso o se aborta un proceso.

Manejo de memoria

5. ¿Cuál es el objetivo principal de la MMU/MCC (Memory Management Unit/Memory Controller Chip)?

El objetivo de la MMU es realizar la traducción de memoria lógica a memoria física. Esto se debe a que, aunque en los métodos de tiempo de compilación y tiempo de carga se generan direcciones físicas y lógicas idénticas. Sin embargo, en el tiempo de ejecución existe una diferencia o desfase entre las direcciones físicas y lógicas. Ya que la asignación del espacio lógico no corresponde a la asignación del espacio físico. Y es en esta asignación de tiempo de ejecución en donde se realiza el mapeo de la memoria lógica a la memoria física por el hardware del MMU.

¿Cuáles son las diferentes formas de apoyo con hardware que se emplean en cada esquema de manejo de memoria?

- Swapping: es cuando un proceso se encuentra en memoria, es sacado de ella temporalmente para almacenarlo en memoria secundaria y eventualmente es devuelto a memoria para continuar su ejecución. Esto provoca el aumento de direcciones físicas del sistema aumentando a su vez el grado de multiprogramación.
- Contiguous Memory Allocation: la memoria debe poder alojar tanto al sistema operativo y a los procesos del usuario. Por ende, la memoria se suele particionar en una parte para el sistema operativo y otra para albergar los procesos del usuario. Por lo tanto, se debe considerar como se realizará la asignación de la memoria disponible a los procesos de la manera más eficiente. En Contiguous Memory Allocation cada proceso está contenido en una sola sección de memoria contigua a la sección que contiene el siguiente proceso.
- Segmentation: es un esquema de administración de memoria que permite al programador ver la memoria como direcciones físicas. En donde un espacio de direcciones lógicas es representado por una colección de segmentos, y a cada segmento se le asigna un nombre y su longitud. Las direcciones especifican tanto el nombre del segmento como el desfase dentro del segmento. Lo que permite al programador especificar cada dirección por su nombre y el desfase.
- Paging: es un esquema de administración de memoria que realiza una división de la memoria física en bloques de tamaño fijo llamados frames y divide la memoria lógica en bloques del mismo tamaño llamados pages. Así, cuando un proceso va a ser ejecutado sus pages son cargadas en cualquiera de los frames disponibles. Además, este esquema evita la fragmentación externa y la necesidad de compactación.

6. Describa la relación y diferencias entre los conceptos de segmentación y pagineo (paging). Asegúrese de dejar clara la diferencia entre direcciones de memoria lógicas o virtuales y direcciones físicas.

La relación existente entre estos dos esquemas de administración de memoria es que ambos permiten la asignación de memoria no contigua. También se caracterizan por hacer la división de la memoria física y lógica en bloques, aunque lo realicen de forma diferentes. En segmentación las direcciones lógicas se organizan como una pareja de datos: segmento offset. Mientras que en paging las páginas, que equivalen a las direcciones lógicas, se organizan con una asignación de base dos. Una diferencia es que el paging no incurre en fragmentación, mientras que segmentación si lo hace y evita la necesidad de compactación.

7. ¿Qué es swapping?

Es cuando un proceso se encuentra en memoria, es sacado de ella temporalmente para almacenarlo en memoria secundaria y eventualmente es devuelto a memoria para continuar su ejecución. Esto provoca el aumento de

direcciones físicas del sistema aumentando a su vez el grado de multiprogramación. El swapping no es un esquema viable para los sistemas operativos móviles, debido a que estos tienden a tener memoria secundaria escasa. Por lo que utilizar la memoria secundaria para almacenar procesos puede causar que los dispositivos se queden sin memoria secundaria.

¿Cuál es la diferencia entre swapping y paging?

La diferencia entre estos dos esquemas de administración de memoria recae en que swapping hace uso de la memoria secundaria para almacenar procesos que estaban en memoria, creando más direcciones físicas. Por su parte paging no crea memorias físicas adicionales, solo utilizando las existentes. Otra diferencia es que swapping no realiza ninguna división de la memoria física ni lógica, debido a que generalmente se copia el proceso completo o toda la memoria a un espacio disponible dentro del disco duro.

8. ¿Qué es y cómo se da la fragmentación? ¿Qué acercamientos para asignación de memoria son susceptibles a cada tipo?

La fragmentación es un problema que se da debido a la forma en que se almacenan los datos dentro de la memoria. Se da cuando los procesos son cargados y removidos de la memoria constantemente, el espacio libre restante se convierte en pequeños trozos repartidos en toda la memoria. Estos pequeños espacios resultan ser tan pequeños que solo datos específicos pueden darles uso, provocando pérdida del espacio total disponible. Este caso es lo que se conoce como fragmentación. La fragmentación externa existe cuando hay suficiente espacio libre para satisfacer la necesidad, pero este espacio no es contiguo. Este es un serio problema y en el peor de los casos este espacio se da por perdido.

Las estrategias first-fit y best-fit de alojamiento de memoria sufren de fragmentación externa. Esto depende del tamaño promedio de los procesos y de la cantidad total de memoria que se tiene disponible. Además, hemos visto que paging evita la fragmentation externa y que segmentation sufre de este problema. Si tomamos en cuenta a swapping, podemos decir que puede causar fragmentación en la memoria secundaria, aunque realmente depende del sistema de archivos utilizado por el sistema operativo.

9. ¿Cuáles son las implicaciones principales de variar el tamaño de página en un sistema que emplea paging? ¿Cuáles son las motivaciones para las diferentes estructuras de tabla de paginación?

Si el tamaño de las pages es muy grande, se puede tener el problema desperdicio de memoria, debido a que los programas pequeños no utilizaran este espacio, causando fragmentación interna. Mientras que si las pages son demasiado pequeñas se tendrían muchas entradas en la page table, provocando que los tiempos de búsqueda de una page específica aumente. Esto motiva a la utilización de diferentes estructuras de page tables, en donde cada estructura puede representar pages de diferentes tamaños, y cada proceso se le asigna las pages más adecuadas a su tamaño. Lo que resuelve el problema de tener que definir un solo tamaño de pages y seleccionar un tamaño inadecuado. También el tiempo de búsqueda se ve beneficiado, ya que la búsqueda se puede reducir a menos elemento, al saber a cuál tabla se tiene que ir a buscar, en vez de tener que buscar en una tabla enorme.

10. ¿Por qué se pueden encontrar dificultades para implementar dynamic linking en un sistema que emplea una tabla invertida de paginación?

Una ddl (dynamic-link library) es un archivo con código ejecutable que se carga cuando un programa la necesita. Aunque esta definición es exclusiva para los sistemas operativos Windows, prácticamente el concepto también existe en los sistemas operativos modernos. Mientras que la paginación invertida hace referencia es una técnica en donde hay una entrada por cada frame de la memoria, la cual además incluye información del proceso que posee dicha página. Por lo tanto, en el sistema solo habrá una tabla de páginas invertida y esta solo tendrá una entrada por cada frame. Aunque esta técnica reduce la memoria física ocupada por la tabla de páginas, incrementa el tiempo de búsqueda de una página.

Es por esta razón que implementar dynamic linking en un sistema que emplea una tabla invertida de paginación presenta dificultades. En donde la creación de una librería dinámica implica que se debe mapear en memoria y que

cuando quiera ser utilizada debe realizarse una búsqueda dentro de la tabla invertida de paginación para identificar donde se encuentra en memoria. Y luego ya se puede continuar con la ejecución del proceso. Esto puede implicar starvation en el peor de los casos, debido a que, si la búsqueda toma mucho tiempo, el proceso se encuentra esperando a los resultados de la búsqueda sin realizar ningún trabajo.

Almacenamiento secundario

11. Explique el mecanismo de cabezales y discos magnéticos con el que funcionan los discos duros, incluyendo la organización de los discos en pistas, sectores y cilindros. ¿Qué pasa si un cabezal entra en contacto físico con uno de los platos magnéticos?

Los discos magnéticos son los componentes esenciales en un disco duro, debido a que es aquí donde se almacena la información. Están fabricados de cerámica, vidrio o aluminio, revestidos con una aleación metálica. Generalmente los discos duros están compuestos por discos magnéticos, denominados platos, montados sobre un eje central. Cada plato puede detener dos caras, las cuales se utilizan para almacenar datos.

Los cabezales son partes móviles que leen y escriben la información guardada en los discos magnéticos. Estos son los componentes más sensibles dentro de un disco duro y funcionan variando la posición dentro del disco duro para poder acceder a la información solicitada. Esta pieza lleva en la punta piezas de cerámica, que son los dispositivos sensibles a los campos magnéticos dentro del disco magnético. Debido a la debilidad de la señal que generan estos campos magnéticos, el cabezal dispone de un amplificador de señal.

En los discos magnéticos, la información se organiza en pistas, cilindros y sectores. Una pista es un camino circular en la superficie del disco magnético donde la información es grabada y leída magnéticamente. La pista es una división física de los datos.

Una pista de unidad de disco es un camino circular en la superficie del disco magnético donde la información es grabada magnéticamente y de la cual también se lee la información registrada. Mientras que un cilindro es el conjunto de pistas que se encuentran en diferentes discos magnéticos y que están en la misma posición dentro de cada uno de los discos magnéticos (tienen el mismo diámetro). Es decir que forman un cilindro imaginario en los discos magnéticos. Un sector es una división formada entre dos líneas radiales de una pista, en donde se puede almacenar y leer información.

Si un cabezal entra en contacto físico con un disco duro magnético este puede causar daños al recubrimiento magnético del disco. Lo que puede ocasionar pérdida de información en el sector dañado o incluso dejar inservible la parte dañada, por lo que no sería posible realizar escrituras o lecturas.

12. ¿Qué miden los tiempos de búsqueda (seek time) y la latencia rotacional (rotational latency) sobre el funcionamiento de un disco duro?

El *Seek Time* mide el tiempo en que el conjunto de cabezales o brazo del disco se mueve a un cilindro especificado. Mientras que el *Rotational Latency* mide el tiempo para que el sector deseado rote al cabezal del disco, es decir, en posicionar un sector en el cabezal. Cabe resaltar que existen 2 partes que contribuyen a la velocidad del disco: Transfer rate y Positioning time. Seek Time y Rotational Latency son parte del Positioning time.

13. Describa y distinga los conceptos de SAN (storage area network) y NAS (network attached storage).

SAN es una red privada que conecta servidores y unidades de almacenamiento, en la que se usa protocolos de almacenamiento en lugar de protocolos de red. La ventaja de utilizar una SAN es su flexibilidad, debido a que se pueden conectar múltiples hosts y múltiples arreglos de almacenamiento a la misma SAN, y el almacenamiento se puede asignar dinámicamente a los hosts. Las SAN hacen posible que los clústeres de servidores compartan el mismo almacenamiento y que las matrices de almacenamiento incluyan varias conexiones directas de host.

NAS es un dispositivo de almacenamiento conectado a la red, el cual tiene como principal propósito que la información almacenada pueda ser accedida por medio de una red de datos. Los clientes acceden a la información por medio de una llamada a un procedimiento remoto, la cual se realiza a través de TCP o UDP en una red IP, generalmente la

misma red de área local (LAN) que lleva todo el tráfico de datos a los clientes. El concepto de NAS se puede abstraer como un protocolo más de acceso de almacenamiento.

Entre los problemas más comunes de almacenamiento remoto es que las operaciones de I/O consumen el ancho de banda de las redes, provocando aumento en la latencia de la comunicación entre el servidor y el cliente. Las SANs evitan este inconveniente utilizando otros protocolos y conexiones directas con los hosts, en comparación de las NASs. Mientras que la ventaja de las NASs es que brindan acceso a archivos completos a todos los hosts que lo soliciten, pero no es tan flexible como las SANs.

14. ¿Cómo se organiza lógicamente el almacenamiento secundario? ¿Cómo se mapea esta organización a un disco duro típico (i.e., qué contienen las direcciones a las que se traducen las referencias lógicas)? ¿Cuál es la diferencia entre un block y un cluster?

El almacenamiento secundario se organiza mediante una unidad de transferencia, que son denominados bloques lógicos. El tamaño de estos bloques puede depender de cada disco, en general suelen ser de 512 bytes. Para este caso el disco se trata como si fuera una lista de una dimensión, en donde a cada bloque se le asigna a un sector del disco. El mapeo de cada sector se maneja al seleccionar el primer sector de cada pista, empezando por la más alejada del centro y moviéndose hacia las pistas más cercanas. La diferencia entre un bloque y un cluster es que el bloque es una pequeña parte de cada pista, mientras que un cluster es el conjunto de dos o más bloques.

Posteriormente se aplica una partición, la cual es un espacio en donde pueden ser almacenados los archivos. Específicamente, una partición debe ser formateada con un sistema de archivos para que puede ser utilizada por el sistema operativo. Al aplicar un sistema de archivos, una partición puede almacenar directorios y/o archivos según el usuario requiera.

15. Explique las diferencias entre los siguientes conceptos:

Bloque y sector.

La diferencia en los sistemas operativos más modernos entre los bloques y sectores es el tamaño de cada uno. Debido a que los sectores tienen un tamaño específico dentro del disco y los bloques pueden tener tamaños arbitrarios, que no necesariamente sean iguales al tamaño de un sector.

Formateo a bajo nivel y formateo lógico.

La diferencia radica en la forma en que se elimina la información dentro del almacenamiento. Cuando se hace un formateo lógico, realmente se hacen cambios sobre el sistema de archivos, mientras que cuando se realiza un formateo a bajo nivel se está cambiando físicamente los sectores del disco duro. Es por esta razón, que al realizar un formateo lógico es posible recuperar la información formateada, debido a que en realidad aún existe, pero ya no se tiene la estructura ni referencia para ser leída por el sistema operativo. Pero un formateo a bajo nivel no se puede recuperar la información debido a que se cambió la forma de los sectores y se escribió sobre ellos.

Error suave y error duro durante el manejo de un disco duro.

La diferencia radica en la forma en que se manifiesta el error en el disco duro. Siendo un error duro en el disco un error que es físico en el disco, lo que significa que los clusters de almacenamiento del disco duro tienen un daño físico. Mientras que un error suave en el disco es una falla de sistema, en la cual no es posible acceder a un sector específico del disco duro. Por lo que se puede restaurar la información en ese bloque o con un formateo se puede corregir. Mientras que un error duro, al ser físico, no se puede reparar con ninguna herramienta de formateo.

16. Identifique los algoritmos de calendarización de disco descritos por las gráficas a continuación. Incluya una interpretación para cada gráfica y las ventajas principales de cada algoritmo. En cada gráfica se supone que el cabezal se ubica en el cilindro 53 y que la secuencia de solicitudes por cilindros es la siguiente: 98, 183, 37, 122, 14, 124, 65, 67.

En la gráfica podemos observar que el cabezal tiene que moverse de un lado a otro, sin importar donde se encuentre, lo que puede provocar starvation, ya que se debe esperar hasta que el cabezal llegue al cilindro que se quiere leer. La ventaja de este algoritmo es que todas las peticiones llegarán a ser procesadas eventualmente, solo tienen que esperar por su turno.

b. SSTF

En la gráfica podemos observar que el cabezal no tiende a recorrer de un lado para otro para procesar las solicitudes, y más bien forma un camino para llegar al cilindro más lejano en donde pasa por las otras solicitudes. La ventaja de este algoritmo es que se reduce el tiempo en el que el cabezal tiene que moverse para llegar al cilindro deseado.

c. LOOK

En la gráfica podemos observar que el cabezal recorre los cilindros que están hacia la izquierda, y cuando no quedan más a la izquierda cambia su movimiento a la derecha. La ventaja de este algoritmo es que no necesita llegar hasta el final del disco, como el SCAN, para cambiar de dirección, lo que disminuye el tiempo de espera para las lecturas en la otra dirección.

Filesystems

17. ¿Qué es un archivo? ¿Cuál es su propósito como abstracción? ¿Cómo se relaciona con el concepto de volumen?

Un archivo es la abstracción de las propiedades físicas del almacenamiento en un sistema operativo hacia el almacenamiento lógico. Es decir, es la representación lógica del almacenamiento físico. El propósito de esta abstracción es que gracias a que las computadoras pueden almacenar información en varios medios de almacenamiento, como discos magnéticos, cintas magnéticas y discos ópticos, se necesita una representación para esta información dentro del sistema operativo. Para que el sistema informático sea fácil de usar, el sistema operativo proporciona una vista lógica uniforme de la información almacenada. Los archivos son asignados por el sistema operativo a los dispositivos físicos. Estos dispositivos de almacenamiento generalmente son no volátiles, por lo que el contenido es persistente entre los reinicios del sistema.

Debido a que es posible la creación de diferentes sistemas de archivos en diferentes particiones, cualquier identidad que contiene un sistema de archivos es denominada volumen. El volumen puede ser un subconjunto de un dispositivo, un dispositivo completo o varios dispositivos vinculados entre sí. Los volúmenes también pueden almacenar múltiples sistemas operativos, permitiendo que un sistema arranque y ejecute más de un sistema operativo a la vez.

18. Explique los siguientes conceptos:

Sistema de archivos virtual.

Un sistema de archivos virtual es una abstracción del sistema de archivos normal, que permite a las aplicaciones cliente acceder a la información de diversos tipos de sistemas de archivos. Una aplicación de VFS es crear un puente sobre las diferencias en los sistemas de archivos de Windows, de Mac OS y Unix, de modo que las aplicaciones pudieran tener acceso a archivos en los sistemas de archivos locales sin tener que realizar una traducción entre los diferentes tipos de sistemas de archivos y así tener compatibilidad entre ellos.

Sistema de archivos temporal.

Los sistemas de archivos temporal hacen referencia a la creación de archivos temporales, que se utilizan para hacer uso de memoria o para prevenir pérdida de información. Estos archivos son creados por un programa y son eliminados cuando ya no se necesitan. Algunos programas crean archivos y no los eliminan. Esto puede ocurrir porque el programa se bloquea o porque el desarrollador no realizó esta acción. Los archivos temporales dejados a un lado acumulan con el tiempo mucho espacio en el disco, aunque se puede recurrir a una serie de utilidades de sistema, llamadas limpiadores de archivos temporales o limpiadores de disco, que sirven para remover archivos temporales del sistema operativo.

Es un sistema de archivos en el que se puede montar cualquier dispositivo y ejecutar de manera virtual. Una de las ventajas de este sistema de archivos es que no es necesario grabar un DVD para montar el sistema, puede ser montado en un espacio de memoria. Es utilizado muchas veces para crear particiones booteables.

19. ¿Qué es un directorio? ¿Cómo se relaciona con el concepto de punto de montaje (mount point) para un sistema de archivos? Describa al menos tres estructuras lógicas para directorios.

Un directorio es un contenedor virtual en donde se almacenan y organizan archivos, estos pueden tener otros directorios llamados subdirectorios. Técnicamente, el directorio almacena información acerca de los archivos que contiene: como los atributos de los archivos o dónde se encuentran físicamente en el dispositivo de almacenamiento. Al considerar una estructura de directorio, debemos tener en cuenta que se deben realizar las siguientes operaciones dentro del directorio:

- Buscar un archivo: es necesario poder buscar un archivo en particular, debido a que los archivos tienen nombres simbólicos, se debe poder buscar los archivos que correspondan a un patrón en particular.
- Crear un archivo: se puede crear y agregar un archivo al directorio.
- Eliminar un archivo: si un archivo ya no es necesitado, se debe poder remover del directorio.
- Listar un directorio: se debe poder listar los elementos de un directorio y el contenido de los archivos.
- Renombrar un archivo: debido a que el nombre representa el contenido del archivo para un usuario, el usuario debe ser capaz de cambiar el nombre cuando el contenido del archivo cambia.

Directorios Single-Level

Es la estructura lógica para directorios más simple, la cual consiste en que todos los archivos son contenido en el mismo directorio, lo que es fácil de soportar y entender. Un directorio de un solo nivel tiene limitaciones significativas, como la organización, si el número de archivos es grande. Como todos los archivos están en el mismo directorio, deben tener nombres únicos. Y si existen más usuarios, deben compartir el mismo directorio.

Directorios Two-Level

En esta estructura de directorios cada usuario tiene su propio directorio (UFD). Las UFD tienen estructuras similares, pero cada una enumera solo los archivos de un solo usuario. Cuando el usuario inicia sesión, el sistema se busca el directorio correspondiente al nombre de usuario o el número de cuenta, y cada entrada apunta a su identificador de usuario. Cuando un usuario hace referencia a un archivo en particular, solo se busca su propia UFD, de esta manera diferentes usuarios pueden tener archivos con el mismo nombre.

Directorios Tree-Structured

Esta estructura comparte las características de como cada usuario tiene su propio directorio, con agregación de que cada usuario puede crear subdirectorios. De esta manera, existe un directorio raíz que contiene a los otros directorios, y este directorio raíz es el que se asocia a cada usuario. Realmente lo que se está creando un árbol de directorios, en donde cada nodo representa un directorio y las aristas son enlaces a archivos o a subdirectorios.

20. Describa los conceptos en dos de los siguientes incisos:

File control block (FCB), inode y vnode.

El File control block es una estructura del sistema de archivos en la que se mantiene el estado de un archivo abierto, de esta manera el sistema operativo puede administrar el FCB mientras el archivo está en memoria siendo utilizado por su proceso. El inode es una estructura de datos en un sistema de archivos de Unix que describe un objeto del sistema de archivos, como un archivo o un directorio. Cada inode almacena los atributos y las ubicaciones de bloque de disco de los datos del objeto. La representación en memoria del núcleo del sistema operativo de estos datos se llama estructura inode en Linux. Los sistemas derivados de BSD utilizan el término vnode, que se refiere a la capa del sistema de archivos virtual del kernel.

Cuando se realiza una asignación encadenada de archivos en el disco, se refiere a que los archivos están siendo colocados en espacios encadenados o enlazados, por lo que se obtiene:

- En lo que respecta a la administración del espacio libre, no hay desperdicio de espacio.
- No hay acceso aleatorio.
- No se genera fragmentación extra.
- Se adapta mejor a archivos secuenciales.

Tablas de archivos y file descriptors/handles.

Una tabla de archivos es una matriz de punteros a las estructuras de archivos del kernel, la cual representa archivos abiertos. Cuando los procesos realizan llamadas I/O se refieren a los archivos por un entero, que es conocido como descriptor de archivos, que utiliza el kernel para indexar en esta tabla.

Extras

21. Explique qué es un RAID y cómo funcionan sus niveles 0, 1 y 5.

RAID es una técnica de organización de discos, utilizada normalmente para abordar los problemas de rendimiento y fiabilidad. Gracias a que los discos duros, es viable conectar muchos discos a un sistema informático. Tener una gran cantidad de discos en un sistema presenta oportunidades para mejorar la tasa en la que los datos se pueden leer o escribir, al colocar los discos a operar en paralelo. Además, esta configuración ofrece el potencial para mejorar la confiabilidad de los datos almacenados, debido a que se crean redundancias de la información, y a su vez se puede almacenar en varios discos. Así, la falla de un disco no conduce a la pérdida completa de los datos.

RAID 0

Es un RAID en donde los datos se distribuyen en 2 o más discos. Esta organización no tiene redundancia, paridad o tolerancia a fallos. Debido a esto, si un dispositivo falla dentro del RAID, todo el arreglo dejará de funcionar. Esto se debe a que la información está distribuida a lo largo del sistema y la falta de una porción provoca errores de interpretación. Se debe aplicar este esquema si lo que se busca tener es mayores velocidades de escritura y lectura o si se quiere implementar un volumen de mayor tamaño.

RAID 1

Este RAID consiste en un espejo de uno o más discos. Este esquema presenta redundancia, pero no presenta paridad o aumento del volumen. Es viable aplicar este RAID cuando los tiempos de lectura y la importancia de los datos es más importante que las velocidades de escritura o el aumento del tamaño del volumen. Cuando ocurre un fallo en un disco duro, solo se debe efectuar una copia del disco espejo y la información se podrá recuperar.

RAID 5

Consiste en la división de bloques a nivel con paridad distribuida. Requiere que todas las unidades, excepto una, estén presentes para funcionar. Si falla una sola unidad, las lecturas faltantes se pueden calcular a partir de la paridad distribuida, de modo que no se pierdan datos. RAID 5 requiere al menos tres discos. Al aplicar este RAID, las velocidades de escritura aumentan, ya que la información está distribuida entre todos los discos.

22. ¿Cuáles son las particularidades de funcionamiento de un Network Filesystem (NFS)?

Un sistema de archivos NFS distribuido utilizado en redes, en donde las computadoras conectadas a la red de datos pueden acceder al almacenamiento. La característica principal de este sistema de archivos es que los usuarios pueden acceder y utilizarlo de manera concurrente, lo que permite la eliminación de redundancia de archivos. En donde un archivo puede ser accedido desde la red por todos los usuarios y no tiene que ser necesariamente descargado por todos.

23. ¿Qué es sector sparing y para qué sirve?

Un sector sparing es un sector adicional del disco duro y tiene la utilidad de servir de “repuesto” para un sector dañado. Al ocurrir un error en algún sector, el disco duro marca ese sector como dañado y asigna un sector spare como remplazo del dañado. De esta manera el disco duro presenta cierta tolerancia a fallos, aunque no es la mejor opción. Esto se debe a que los sectores de sparing solo pueden ser escritos una cierta cantidad de veces y presentan un mayor tiempo de búsqueda al ser leídos o escritos.