

1. ¿Qué tareas realiza el Administrador de Memoria del Sistema Operativo?

El Administrador de Memoria del Sistema Operativo realiza las siguientes tareas:

Asignación de memoria a procesos.

Liberación de memoria cuando un proceso ya no la necesita.

Protección de la memoria para evitar accesos no autorizados.

Administración de la memoria virtual, incluida la gestión de la paginación y segmentación.

Resolución de conflictos de asignación de memoria cuando múltiples procesos compiten por recursos limitados.

2. ¿Por qué es necesario administrar la memoria?

Es necesario administrar la memoria en un sistema operativo por varias razones:

Para garantizar un uso eficiente de la memoria física.

Para proporcionar aislamiento y protección entre los procesos.

Para permitir que múltiples procesos compartan la memoria de manera segura.

Para gestionar la memoria virtual y hacer un uso eficiente de la memoria secundaria (como el disco duro).

Para prevenir fallos de página y trashing.

3. ¿Qué se tiene el Sistema Operativo para realizar una óptima administración de memoria?

Para realizar una óptima administración de memoria, el Sistema Operativo dispone de:

Algoritmos de asignación de memoria que determinan cómo se asigna la memoria a los procesos.

Tablas de administración de memoria que mantienen un registro de qué partes de la memoria están ocupadas y por quién.

Técnicas de protección de memoria para garantizar que los procesos no accedan a la memoria de otros procesos de manera no autorizada.

Mecanismos de paginación, segmentación y memoria virtual para extender la memoria física utilizando almacenamiento secundario.

Algoritmos de sustitución de páginas y segmentos para manejar la memoria virtual de manera eficiente.

4. ¿Cómo se pueden clasificar a los sistemas operativos de acuerdo a cómo administran la memoria?

Los sistemas operativos se pueden clasificar de acuerdo a cómo administran la memoria en:

Sistemas de asignación contigua: Asignan bloques contiguos de memoria a procesos.

Sistemas de asignación no contigua: Asignan memoria de manera fragmentada utilizando paginación, segmentación o una combinación de ambas.

5. ¿Qué es la Fragmentación Interna? ¿Qué es la Fragmentación Externa?

La Fragmentación Interna es el espacio no utilizado dentro de un bloque asignado a un proceso debido a que el tamaño del bloque es mayor que el tamaño necesario para almacenar la información.

La Fragmentación Externa es el espacio no utilizado entre bloques de memoria asignados a procesos, lo que puede dificultar la asignación de memoria a procesos nuevos.

6. ¿Cuál es el esquema de administración de memoria más sencillo? Indique características, ventajas y desventajas.

El esquema de administración de memoria más sencillo es la asignación contigua, donde los procesos se cargan en bloques de memoria contiguos.

Características: simplicidad.

Ventajas: fácil de implementar y baja sobrecarga.

Desventajas: puede generar fragmentación interna y limita la flexibilidad en la asignación de memoria.

7. ¿Por qué es necesario tener que manejar la memoria de múltiples procesos a la vez?

Es necesario manejar la memoria de múltiples procesos a la vez porque en un sistema operativo multitarea, varios procesos se ejecutan simultáneamente y compiten por los recursos de memoria. La administración adecuada de la memoria es esencial para evitar conflictos y garantizar el rendimiento y la seguridad del sistema.

8. ¿Qué problemas introduce la multiprogramación? ¿cómo fueron solucionados inicialmente?

La multiprogramación introduce problemas como la fragmentación, la asignación eficiente de memoria y la gestión de conflictos de memoria. Estos problemas se abordaron inicialmente mediante técnicas de asignación de particiones fijas y particiones variables.

9. ¿En qué consiste el esquema de Multiprogramación con Particiones Fijas? Indique ventajas y desventajas.

El esquema de Multiprogramación con Particiones Fijas asigna particiones de memoria de tamaño predeterminado a los procesos. Ventajas: simplicidad y baja sobrecarga. Desventajas: puede llevar a fragmentación interna y no es eficiente en la utilización de la memoria.

10. ¿Cómo se podría solucionar el problema de trabajos pequeños con Multiprogramación con Particiones Fijas? ¿Por qué es poco recomendable?

El problema de trabajos pequeños en Multiprogramación con Particiones Fijas se puede solucionar dividiendo las particiones en tamaños más pequeños. Sin embargo, esto no es recomendable porque aumenta la sobrecarga de administración.

11. ¿Para qué tipos de procesos sería recomendable el esquema de Multiprogramación con Particiones Fijas? ¿Por qué?

El esquema de Multiprogramación con Particiones Fijas es recomendable para procesos con requisitos de memoria similares y cuando se conoce de antemano el tamaño de los procesos. Es eficiente cuando los procesos tienen un tamaño uniforme.

12. ¿Para qué tipos de procesos no sería recomendable el esquema de Multiprogramación con Particiones Fijas? ¿Por qué?

No sería recomendable el esquema de Multiprogramación con Particiones Fijas para procesos con tamaños muy diferentes, ya que puede llevar a una asignación ineficiente de memoria y fragmentación interna.

13. ¿Qué estrategias se definieron para solucionar los problemas encontrados en Multiprogramación con Particiones Fijas?

Se definieron estrategias como la compactación de memoria, la reubicación dinámica y la partición dinámica para solucionar los problemas de asignación de memoria en Multiprogramación con Particiones Fijas.

14. ¿En qué consiste el esquema de Intercambio? Indique ventajas y desventajas.

El esquema de Intercambio (Swapping) consiste en mover un proceso completo entre la memoria principal y el almacenamiento secundario cuando se produce un cambio de contexto. Ventajas: permite ejecutar procesos más grandes, sencillez en la administración. Desventajas: alto tiempo de E/S, ineficiente en sistemas multitarea.

15. ¿Qué es la compactación de memoria? ¿Por qué es importante?

La compactación de memoria es el proceso de reorganizar las particiones de memoria para eliminar la fragmentación interna. Es importante para aprovechar mejor la memoria disponible y evitar problemas de fragmentación.

16. ¿Qué problemas traen los procesos que utilizan memoria dinámica frente a los procesos con memoria fija? ¿Cómo se solucionaría ese problema?

Los procesos que utilizan memoria dinámica pueden fragmentar la memoria, lo que dificulta la asignación de memoria a otros procesos. Este problema se puede solucionar mediante técnicas de asignación no contigua, como la paginación o la segmentación.

17. ¿Cuáles son los dos métodos propuestos para administrar la memoria dinámica? Indique sus características, ventajas y desventajas de cada uno.

Los dos métodos propuestos para administrar la memoria dinámica son la paginación y la segmentación:

Paginación: Divide la memoria en páginas de tamaño fijo y asigna páginas a procesos.

Ventajas: elimina la fragmentación externa, permite un uso eficiente de la memoria física.

Desventajas: puede generar fragmentación interna.

Segmentación: Divide el espacio de direcciones en segmentos lógicos y asigna segmentos a procesos.

Ventajas: facilita la protección y el aislamiento, permite una asignación más flexible.

Desventajas: puede generar fragmentación externa.

18. ¿Qué algoritmos se proponen para asignar memoria dinámica con listas enlazadas? Indique características, ventajas y desventajas.

Los algoritmos propuestos para asignar memoria dinámica con listas enlazadas incluyen:

Primer ajuste: Asigna el primer bloque libre que es lo suficientemente grande para el proceso. Ventaja: simple y rápido. Desventaja: puede generar fragmentación.

Mejor ajuste: Asigna el bloque libre más pequeño que sea suficiente para el proceso. Ventaja: reduce la fragmentación. Desventaja: puede ser lento y generar fragmentación interna.

Peor ajuste: Asigna el bloque libre más grande disponible. Ventaja: puede reducir la fragmentación externa. Desventaja: ineficiente en términos de uso de la memoria.

19. En qué consiste el método de Memoria Virtual?

El método de Memoria Virtual permite que un sistema operativo utilice más memoria de la que realmente tiene físicamente disponible, mediante la combinación de memoria principal y almacenamiento secundario (como disco duro) para respaldar el espacio de direcciones de los procesos. Los procesos creen que tienen acceso a una cantidad de memoria mayor que la físicamente presente.

20. ¿Qué son las direcciones virtuales? Indique cómo se relacionan con el Hardware, el Sistema Operativo y los Procesos

Las direcciones virtuales son direcciones de memoria utilizadas por un proceso, que son generadas por el proceso mismo. Estas direcciones se relacionan con el hardware a través de la unidad de gestión de memoria (MMU), que las traduce en direcciones físicas correspondientes en la memoria RAM. El sistema operativo administra esta traducción y asigna la memoria física adecuada a cada dirección virtual.

21. ¿Qué es la Paginación? ¿Qué ventajas presenta?

La paginación es un esquema de administración de memoria virtual en el que la memoria física y la memoria virtual se dividen en páginas de tamaño fijo. Ventajas de la paginación incluyen la gestión eficiente de la memoria y la eliminación de fragmentación externa.

22. ¿Qué es la Tabla de Páginas?

La Tabla de Páginas es una estructura de datos utilizada en sistemas de paginación para realizar la traducción de direcciones virtuales a direcciones físicas. Consiste en una tabla que almacena la

relación entre las páginas de memoria virtual y las páginas correspondientes en la memoria física.

23. ¿Cómo se realiza la traducción de una dirección virtual a una dirección física utilizando la Tabla de Páginas?

La traducción de una dirección virtual a una dirección física utilizando la Tabla de Páginas implica dividir la dirección virtual en una parte de número de página y una parte de desplazamiento dentro de la página. Luego, el número de página se utiliza como índice en la Tabla de Páginas para obtener la dirección física correspondiente.

24. ¿Qué problemas hay que considerar para implementar la Tabla de Páginas?

Los problemas a considerar para implementar la Tabla de Páginas incluyen la gestión de la memoria ocupada por la tabla misma, el tiempo de búsqueda en la tabla y la sincronización para garantizar que las traducciones sean precisas y actualizadas.

25. ¿Qué son las Tablas de Páginas Multinivel? ¿Qué problema soluciona?

Las Tablas de Páginas Multinivel son una extensión de la Tabla de Páginas que solucionan el problema de tener una tabla de gran tamaño. Divide la tabla en múltiples niveles de tablas, lo que mejora la eficiencia y la administración de la memoria.

26. ¿Cómo se realiza la traducción de una dirección virtual a una dirección física utilizando la Tabla de Páginas Multinivel?

Para realizar la traducción de una dirección virtual a una dirección física utilizando Tablas de Páginas Multinivel, se divide la dirección virtual en partes de índices correspondientes a los niveles de tablas. Cada índice se utiliza para acceder a una tabla específica en el nivel correspondiente, y finalmente se obtiene la dirección física.

27. ¿Qué ventajas brinda utilizar las Tablas de Páginas Multinivel? ¿Cuántos niveles son recomendables en las Tablas de Páginas Multinivel?

Las ventajas de utilizar Tablas de Páginas Multinivel incluyen una mayor eficiencia en la gestión de la memoria y la capacidad de administrar sistemas con una gran cantidad de memoria virtual. El número de niveles recomendables depende de la arquitectura del sistema y los requisitos de rendimiento.

28. ¿Cómo se pueden implementar las entradas de la Tablas de Páginas Multinivel? ¿De qué depende?

Las entradas de las Tablas de Páginas Multinivel se pueden implementar de diversas formas, como tablas o árboles, y depende de la arquitectura específica del hardware y del diseño del sistema operativo.

29. ¿Cómo se solucionaría el otro problema de implementar la Tabla de Páginas? ¿De qué depende (Hardware o Sistema Operativo) y por qué?

El problema de implementar las Tablas de Páginas, como la sincronización y la gestión de la memoria ocupada, puede depender tanto del hardware como del sistema operativo. El hardware debe proporcionar soporte para la traducción de direcciones virtuales, mientras que el sistema operativo se encarga de la administración y actualización de las tablas.

30. ¿Cómo se usa la TLB para traducir las direcciones virtuales?

La TLB (Translation Lookaside Buffer) se utiliza para acelerar la traducción de direcciones virtuales a físicas. Almacena las traducciones más recientes, lo que evita buscar en la Tabla de Páginas cada vez.

31. ¿Qué sucede cuando no se encuentra la página en la TLB? ¿Quién es el encargado de resolverlo?

Cuando no se encuentra una página en la TLB, se produce un fallo en la TLB y es responsabilidad del sistema operativo buscar la traducción correcta en la Tabla de Páginas y actualizar la TLB con la entrada correspondiente.

32. ¿Cuál es la mayor desventaja de utilizar Tablas de Páginas Multinivel?

La mayor desventaja de utilizar Tablas de Páginas Multinivel es que pueden aumentar la latencia en la traducción de direcciones debido a la necesidad de acceder a múltiples niveles de tablas. Esto puede afectar el rendimiento en aplicaciones que requieren un acceso rápido a la memoria.

33. ¿Qué son las Tablas de Páginas Invertidas? Indique ventajas, desventajas y su relación con la TLB.

Las Tablas de Páginas Invertidas son una variante de la administración de la memoria virtual en la que cada entrada de la tabla de páginas contiene información sobre qué proceso ocupa cada página en la memoria física. Las ventajas incluyen un ahorro de memoria en comparación con las

tablas de páginas regulares, pero las desventajas incluyen una búsqueda más lenta para encontrar la entrada de la tabla de páginas invertidas correspondiente a una dirección virtual. La TLB (Translation Lookaside Buffer) se utiliza para mitigar la desventaja de la búsqueda lenta al almacenar algunas de las traducciones más recientes.

34. ¿Qué es el algoritmo de sustitución de páginas? ¿Por qué es importante?

El algoritmo de sustitución de páginas decide qué página de memoria debe ser reemplazada por una nueva página cuando no hay espacio suficiente en la memoria física para cargarla. Es importante porque afecta directamente al rendimiento del sistema operativo y puede influir en la eficiencia del acceso a la memoria.

35. Describa el algoritmo de sustitución de páginas óptimo ¿Por qué es difícil de implementar?

El algoritmo de sustitución de páginas óptimo reemplaza la página que no se utilizará durante el mayor tiempo posible en el futuro. Es difícil de implementar en la práctica porque requiere conocer el futuro comportamiento de los procesos, lo cual es imposible en la mayoría de los casos.

36. Describa el algoritmo de sustitución de páginas no utilizada recientemente (en inglés No Recently Used, o NRU).

El algoritmo de sustitución de páginas no utilizada recientemente (NRU) clasifica las páginas en cuatro categorías según su uso reciente y decide reemplazar una página de la categoría menos utilizada. Las categorías suelen ser modificada-no modificada y referenciada-no referenciada.

37. Describa el algoritmo de sustitución de páginas de primera que entra, primera que sale (en inglés First In - First Out, o FIFO)

El algoritmo de sustitución de páginas de primera que entra, primera que sale (FIFO) reemplaza la página que ha estado en la memoria durante más tiempo. Es simple de implementar pero no tiene en cuenta el uso reciente de las páginas, lo que puede llevar a un rendimiento deficiente.

38. Describa el algoritmo de sustitución de páginas de segunda oportunidad.

El algoritmo de sustitución de páginas de segunda oportunidad (Second Chance) es una mejora de FIFO que considera si una página ha sido referenciada recientemente antes de reemplazarla. Si ha sido referenciada, se le da una "segunda oportunidad" y no se reemplaza de inmediato.

39. Describa el algoritmo de sustitución de páginas por Reloj.

El algoritmo de sustitución de páginas por Reloj (Clock) es una variante de segunda oportunidad que utiliza un puntero para recorrer las páginas en la memoria. Las páginas que no han sido referenciadas recientemente se reemplazan cuando el puntero las alcanza.

40. Describa el algoritmo de sustitución de páginas menos recientemente usadas (en inglés Least Recently Used, o LRU).

El algoritmo de sustitución de páginas menos recientemente usadas (LRU) reemplaza la página que no se ha utilizado durante el período más largo de tiempo. Es efectivo pero puede ser costoso de implementar y mantener debido a la necesidad de llevar un registro del uso de cada página.

41. ¿Por qué la implementación del algoritmo LRU dependería del Hardware?

La implementación del algoritmo LRU puede depender del hardware debido a la necesidad de mantener registros de acceso a cada página. Algunos sistemas pueden utilizar hardware especializado, como contadores de tiempo de acceso, para implementar LRU de manera eficiente.

42. Describa el algoritmo de sustitución de páginas no utilizado frecuentemente (en inglés No Frequently Used, o NFU) e indique cómo soluciona el problema de LRU.

El algoritmo de sustitución de páginas no utilizado frecuentemente (NFU) se basa en un contador que se incrementa con cada acceso a la página. Cuando se necesita reemplazar una página, se selecciona la página con el contador más bajo. NFU resuelve el problema de LRU al no requerir un seguimiento detallado del historial de acceso, pero aún puede no ser óptimo en algunas situaciones.

43. ¿Cuál son las principales diferencias entre los algoritmos LRU y NFU?

Las principales diferencias entre los algoritmos LRU y NFU son que LRU se basa en el historial de acceso exacto, mientras que NFU se basa en un contador de acceso. LRU es más preciso pero puede ser costoso de implementar y mantener, mientras que NFU es más simple pero menos preciso.

44. ¿Qué es la paginación por demanda?

La paginación por demanda es un enfoque en el que las páginas de un proceso no se cargan en la memoria física hasta que son necesarias. Esto permite ahorrar espacio en la memoria física al cargar solo las páginas requeridas para la ejecución actual del proceso.

45. ¿Qué es el conjunto de trabajo de un proceso? ¿Cómo se relaciona con el tamaño de la memoria física?

El conjunto de trabajo de un proceso es el conjunto de páginas que un proceso utiliza en un momento dado.

Se relaciona con el tamaño de la memoria física porque si el conjunto de trabajo de un proceso es más grande que la memoria física disponible, puede provocar trashing, lo que afecta negativamente al rendimiento.

46. ¿Qué es el “trashing”? ¿Por qué se quiere evitar?

El "trashing" es una situación en la que el sistema operativo pasa una cantidad significativa de tiempo intercambiando páginas entre la memoria principal y el almacenamiento secundario debido a una falta de memoria física. Esto ralentiza el sistema y reduce la eficiencia.

47. ¿En qué consiste la pre-paginación?

La pre-paginación es una estrategia en la que el sistema operativo carga anticipadamente páginas adicionales en la memoria antes de que sean estrictamente necesarias. El objetivo es evitar la situación de trashing al anticipar las necesidades del proceso y cargar páginas antes de que se produzca un fallo de página.

48. ¿Cuáles son las dos políticas de asignación de memoria? Descríbalas indicando ventajas y desventajas.

Las dos políticas de asignación de memoria son la asignación contigua y la asignación no contigua.

Asignación contigua: Asigna un bloque de memoria contiguo a un proceso. Ventajas: simple y eficiente en términos de acceso a la memoria. Desventajas: puede generar fragmentación interna y limita la flexibilidad en la administración de la memoria.

Asignación no contigua: Permite asignar memoria no contigua al proceso, utilizando páginas o segmentos. Ventajas: evita la fragmentación interna y permite una asignación más flexible. Desventajas: puede requerir una mayor gestión de la tabla de páginas o segmentos.

49. ¿Cómo se podría resolver el problema de trashing usando el algoritmo de asignación de frecuencia de fallos de página (en inglés page fault frequency, o PFF)?

El problema de trashing se podría resolver utilizando el algoritmo de asignación de frecuencia de fallos de página (PFF) para determinar cuándo cargar o liberar páginas adicionales en la memoria. PFF mide con qué frecuencia se producen fallos de página y ajusta dinámicamente el número de páginas en memoria para mantener un equilibrio entre la carga y la liberación de páginas.

50. ¿Qué se debe considerar para definir el tamaño de página?

Para definir el tamaño de página, se deben considerar factores como la granularidad adecuada para el sistema y la cantidad de memoria disponible. Un tamaño de página más grande puede reducir la sobrecarga de administración de tablas de páginas pero puede aumentar la fragmentación interna.

51. ¿Cuál es la relación entre el tamaño de página y el tamaño de bloque de disco (cluster)?

La relación entre el tamaño de página y el tamaño de bloque de disco (cluster) es importante porque puede influir en la eficiencia de la lectura y escritura en disco. Si el tamaño de página es mucho mayor que el tamaño del bloque de disco, puede haber una mayor cantidad de lecturas/escrituras innecesarias en disco.

52. ¿En qué casos sería necesario permitir a los procesos controlar la memoria virtual?

Sería necesario permitir a los procesos controlar la memoria virtual en casos en los que se requiere un mayor grado de personalización y control sobre la asignación y gestión de memoria. Esto puede ser útil en aplicaciones de tiempo real, sistemas integrados o entornos donde los procesos tienen requisitos de memoria específicos.

53. ¿Qué son los Segmentos?

Los segmentos son divisiones lógicas de un espacio de direcciones virtuales independientes utilizadas en sistemas de segmentación. Cada segmento puede contener información específica, como código, datos, pila, etc., y se puede tratar como una unidad lógica en lugar de una página.

54. ¿Por qué sería útil tener más de un espacio de direcciones virtuales independientes?

Tener más de un espacio de direcciones virtuales independientes puede ser útil para separar procesos o componentes lógicos en un sistema, lo que proporciona aislamiento y seguridad.

También puede permitir la carga y descarga dinámica de módulos de software en espacios de direcciones separados.

55. ¿Qué ventajas tiene el uso de los segmentos?

Las ventajas del uso de segmentos incluyen el aislamiento entre segmentos, facilitando la protección y el control de acceso. Además, los segmentos permiten una administración de memoria más flexible al asignar y liberar segmentos independientemente.

56. ¿Cómo se definen las direcciones virtuales utilizando segmentos?

Las direcciones virtuales utilizando segmentos se definen mediante un par (segmento, desplazamiento). El segmento indica qué parte del espacio de direcciones se está utilizando, y el desplazamiento es la posición relativa dentro del segmento.

57. Compare las principales características de la Paginación con la Segmentación.

Comparación de Paginación y Segmentación:

Paginación: Divide la memoria en páginas de tamaño fijo. Las páginas se asignan a procesos. Elimina la fragmentación externa, pero puede generar fragmentación interna.

Segmentación: Divide el espacio de direcciones en segmentos lógicos. Los segmentos pueden variar en tamaño. Facilita la protección y el aislamiento, pero puede generar fragmentación externa.

58. ¿Por qué es más fácil manejar la protección utilizando segmentos que utilizando páginas?

Es más fácil manejar la protección utilizando segmentos que utilizando páginas porque los segmentos son unidades lógicas más grandes y pueden asignarse permisos de acceso a nivel de segmento. Esto permite un control más granular sobre qué partes de un espacio de direcciones pueden ser accedidas por un proceso.

59. Describa la implementación de segmentación pura.

La implementación de segmentación pura divide la memoria en segmentos de diferentes tamaños y permite que los procesos ocupen múltiples segmentos de forma independiente. No se utiliza paginación en este caso. Cada segmento se identifica por un número de segmento y se puede proteger individualmente.

60. ¿Qué problema presenta la segmentación pura? ¿Cómo se evitaría?

La segmentación pura presenta el problema de fragmentación externa, ya que los segmentos pueden dejar espacios vacíos entre ellos. Para evitarlo, se puede utilizar la segmentación paginada, que combina las ventajas de ambas técnicas y reduce la fragmentación.

61. ¿Cuáles son los principales objetivos de la administración de la memoria?

Simplificar (y abstraer) el uso de la memoria a los usuarios.

Optimizar el uso de la memoria principal de la computadora.

Permitir:

- Proteger los datos.
- Compartir datos.

62. ¿Quién se encarga de la administración de marcos y páginas, traducción de direcciones, y la carga de páginas no presentes?

Sistema operativo, Hardware, Sistema operativo respectivamente.

63. ¿Cuáles son las políticas de manejo de memoria virtual?

Política de búsqueda:

Por demanda: cuando el proceso solicita una página no presente, el SO la busca en el disco rígido y la carga en la RAM.

Pre paginación: cuando el proceso solicita una página no presente, el SO la busca en el disco rígido y la carga en la RAM.

Política de limpieza:

Por demanda: los datos del frame se guardan en el disco rígido sólo cuando es necesario liberar ese frame para cargar otra página

Pre limpieza: el SO anticipa que algunas frames van a tener que ser liberados, y antes de ser necesario guarda los datos de esos frames en el disco rígido (en este caso, guarda los datos, pero no libera los frames)

Usando buffer: política por demanda, pero la diferencia es que en vez de esperar a que los datos se guarden en el disco rigido para liberar el frame, guarda los datos en una memoria temporal (buffer) y de ahí luego lo guarda en el disco rigido (se gana velocidad porque no

tiene que esperar que se guarde en el disco, pero, es poco seguro porque si hay un problema al guardar en el disco, los datos se pueden perder)

Política de reemplazo:

Alcance local: siempre cada proceso va a poder usar solamente los frames que tiene asignados

Global: el proceso podría usar cualquier frame de la memoria RAM que esté disponible

Principales algoritmos:

Optimo o Ideal: solo funciona en la teoría, porque considera los futuros pedidos de páginas.

FIFO: First In First Out (página que se cargó hace más tiempo, es la que va a ser reemplazada)

LRU: Last Recently Used (menos usado recientemente, es decir que va a reemplazar el frame que no haya sido usado hace más tiempo)

LFU: Last Frequently Used (menos frecuentemente usado, es decir que va a reemplazar el frame que fue usado menos cantidad de veces)