



Instituto Politécnico Nacional

ESCOM

Sistemas Operativos  
David Araujo Diaz

Islas Osorio Enrique  
Numero de lista: 20  
Boleta:2021630409

4SCM1

Tarea 3: Administración de memoria

### Tarea 3: Administración de Memoria

#### 1. ¿Qué características debe de tener una memoria ideal?

La memoria RAM es un componente imprescindible para el buen funcionamiento del ordenador. El sistema operativo utiliza este espacio de memoria para cargar los procesos mientras los ejecuta, por lo que contar con la suficiente cantidad de memoria RAM aporta fluidez al funcionamiento del ordenador.

Si el sistema operativo no cuenta con suficiente espacio en la memoria RAM como para almacenar en ella todos los datos que necesita, lo que hace es reservar un espacio adicional en el disco duro o SSD del ordenador para trabajar sobre él.

#### 2. ¿Qué es el administrador de memoria?

El único espacio de almacenamiento que el procesador puede utilizar directamente, más allá de los registros (que si bien le son internos y sumamente rápidos, pero de capacidad muy escasa) es la memoria física. Todas las arquitecturas de procesador tienen instrucciones para interactuar con la memoria, pero ninguna lo tiene para hacerlo con medios persistentes de almacenamiento, como las unidades de disco<sup>1</sup>. Cabe mencionar que cuando veamos en un texto referencia al almacenamiento primario siempre se referirá a la memoria, mientras que el almacenamiento secundario se refiere a los discos u otros medios de almacenamiento persistente.

Todos los programas que deseemos ejecutar deben cargarse a la memoria del sistema antes de ser utilizados. En esta unidad veremos cómo administra el sistema operativo a la memoria para permitir que varios procesos la compartan — Esta tarea debe preverse desde el proceso de compilación de nuestros programas (en particular, la fase de ligado). Hoy en día, además, casi todos los sistemas operativos emplean implementaciones que requieren de hardware especializado — La Unidad de Manejo de Memoria (MMU). Hablaremos de cómo se manejaban los sistemas multitarea antes de la universalización de los MMU, y qué rol juegan hoy en día.

#### 3. ¿Cuál es el trabajo del administrador de memoria?

La gestión de memoria o administración de memoria es el acto de gestionar la memoria de un dispositivo informático. El proceso de asignación de memoria a los programas que la solicitan.<sup>1</sup> La gestión de la memoria principal de una computadora es una tarea de suma importancia para el funcionamiento de la misma.

La administración de memoria se refiere a los distintos métodos y operaciones que se encargan de obtener la máxima utilidad de la memoria, organizando los procesos y programas que se ejecutan de manera tal que se aproveche de la mejor manera posible el espacio disponible.

#### 4. ¿Cuáles son las dos clases en la que es posible dividir los sistemas de administración de memoria?

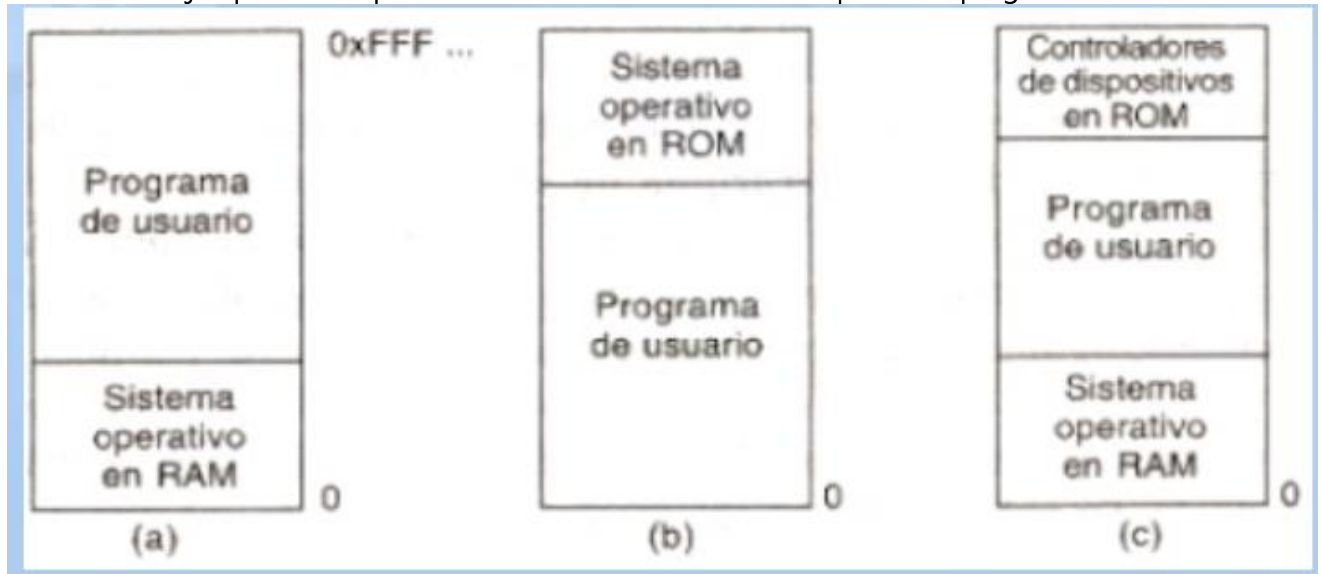
Los sistemas de administración de memoria se pueden clasificar en dos tipos. Los que desplazan los procesos de la memoria principal al disco y viceversa durante la ejecución (intercambio y paginación) y aquellos que no

#### 5. En que consiste el esquema de administración de memoria por monoprogramación.

El esquema de administración de memoria más sencillo posible es ejecutar sólo un programa a la vez, compartiendo la memoria entre ese programa y el sistema operativo. En la Figura 5 se muestran

tres variaciones sobre este tema.

6. Describa tres ejemplos del esquema de administración de memoria por monoprogamación.

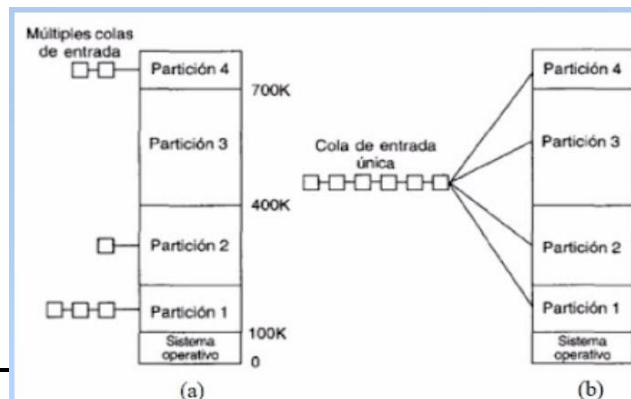


7. En que consiste el esquema de administración de memoria por multiprogramación con particiones fijas.

Aunque a veces se usa la monoprogamación en computadoras pequeñas con sistemas operativos sencillos, hay muchos casos en que es deseable permitir la ejecución de múltiples procesos a la vez. En los sistemas de tiempo compartido, tener varios procesos en la memoria a la vez implica que cuando un proceso está bloqueado esperando que termine una E/S, otro puede usar el CPU. Así, la multiprogramación aumenta el aprovechamiento del CPU. Sin embargo, incluso en computadoras personales hay ocasiones en las que resulta útil poder ejecutar dos o más programas a la vez.

8. ¿Cuáles son las ventajas y desventajas de usar una sola cola de entrada?

La desventaja de repartir los trabajos entrantes en colas distintas se hace evidente cuando la cola de una partición grande está vacía pero la cola de una partición pequeña está llena, como es el caso de las particiones 1 y 3 en la Figura 6(a). Una organización alternativa sería mantener una sola cola, como en la Figura 6(b). Cada vez que se libera una partición, se selecciona el trabajo más cercano a la cabeza de la cola que cabe en esa partición, se carga en dicha partición y ejecuta. Puesto que no es deseable



desperdiciar una partición grande en un trabajo pequeño, una estrategia diferente consiste en examinar toda la cola de entrada cada vez que se libera una partición y escoger el trabajo más grande que cabe en ella.

**9. En que consiste el esquema de administración de memoria por intercambio. Cuál es la diferencia en usar particiones fijas y particiones variables.**

En un sistema por lotes la organización de la memoria en particiones fijas es adecuado pero en un ambiente multiusuario la situación es distinta con el tiempo compartido, ya que existen mas usuarios de los que puede albergar la memoria, por lo que es conveniente albergar el exceso de los procesos en disco., por supuesto para ser ejecutados estos procesos deben ser trasladados a la memoria principal. Al traslado de procesos de disco a memoria y viceversa se le llama intercambio.

**10. ¿Qué es la compactación de memoria y por que casi nunca se emplea?**

Se denomina compresión o compactación de almacenamiento. Esta técnica consiste en trasladar todas las áreas ocupadas del almacenamiento hacia algún extremo de la memoria. Este Proceso es utilizado cuando se tiene una memoria con partición variable en el que se tiene 5 procesos y digamos que el 3 termino su trabajo lo que hace el sistema operativo es coger el proceso 4 y el 5 hacia arriba para ocupar el lugar del 3 y así no tener huecos entre ellos, aunque es tedioso para el sistema operativo porque a cada momento tiene que mover los procesos. Su objetivo consiste en desplazar el contenido de la Memoria libre en un sólo bloque de gran tamaño.

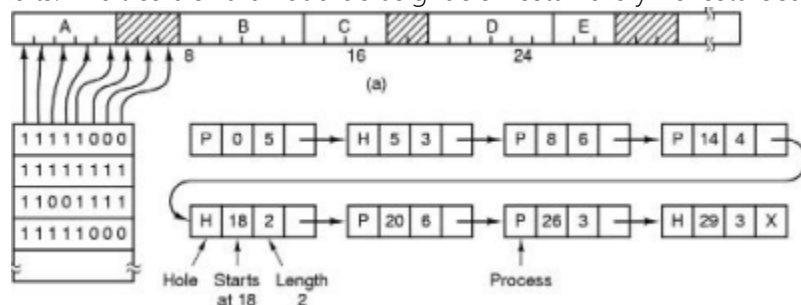
La compactación no siempre es posible, sólo es posible si la relocalización es dinámica y se efectúa en el momento de la ejecución. Este esquema puede ser bastante costoso.

Algunas desventajas de esta técnica de compresión:

- 1) Consume recursos del sistema.
- 2) El sistema debe detener todo mientras efectúa la compresión, lo que puede afectar los tiempos de respuesta.
- 3) Implica la relocalización (reubicación) de los procesos que se encuentran en la memoria.

**11. En que consiste el esquema de administración de memoria con mapas de bits. Cuál debe de ser la relación entre el mapa de bits, el tamaño de la memoria y el tamaño de la unidad de asignación.**

Con un mapa de bits, la memoria se divide en unidades de asignación, que pueden ser desde unas cuantas palabras hasta varios kilobytes. A cada unidad de asignación le corresponde un bit del mapa de bits. El bit es 0 si la unidad de asignación está libre y 1 si está ocupada.



El tamaño de la unidad de asignación es una cuestión de diseño importante. Cuanto más pequeña sea la unidad, mayor será el mapa de bits. Si se escoge una unidad de asignación grande, el mapa de bits será pequeño, pero podría desperdiciarse una unidad de memoria apreciable en la última unidad de asignación del proceso si el tamaño del proceso no es un múltiplo exacto de la unidad de asignación.

- 12.** En que consiste el esquema de administración de memoria con listas enlazadas. Describa los algoritmos de primer ajuste, siguiente ajuste, mejor ajuste, peor ajuste y ajuste rápido; diga como es posible hacer estos algoritmos mas eficientes. ¿Cuál de estos algoritmos es el más eficiente?

Esta lista puede estar ordenada por direcciones en cuyo caso la actualización de la misma al terminar un proceso es casi que inmediata. Un proceso que termina tiene por lo general dos vecinos (uno por arriba y uno por abajo), a menos que se encuentre en algún extremo de la memoria.

Cuando los procesos y los huecos se mantienen en una lista ordenada por direcciones, se pueden utilizar diversos algoritmos para asignar la memoria para un proceso de reciente creación. Se supone que el administrador de memoria conoce la cantidad de memoria a asignar.

First Fit o el Primero en Ajustarse: El administrador de memoria revisa toda la lista de segmentos hasta encontrar un espacio lo suficientemente grande. El espacio se divide entonces en dos partes, una para el proceso y otro para la memoria no utilizada. Este algoritmo es rápido, puesto que busca lo menos posible.

- Siguiente en Ajustarse: Funciona de la misma forma que el anterior, con la diferencia que mantiene un registro del lugar dónde se encuentra un hueco adecuado. La siguiente vez que se le llama, comienza a buscar desde el punto donde se detuvo, en lugar de comenzar a buscar siempre desde el inicio.
- Best-Fit o el Mejor en Ajustarse: Busca en toda la lista y toma el mínimo hueco adecuado. En lugar de asignar un hueco grande, intenta encontrar un hueco más cercano al tamaño necesario.
- El peor ajuste: Toma siempre el hueco más grande disponible, de forma que el hueco resultante sea lo suficientemente grande para ser útil.

Estos cuatro algoritmos pueden agilizarse si se tienen dos listas independientes, una para los procesos y otra para los huecos. De esta forma todos ellos, pueden limitarse a inspeccionar la lista de huecos y no la de los procesos. Sin embargo, el aumentar la velocidad implica que la complejidad se aumenta y la velocidad al liberar la memoria se aumenta, ya que un segmento liberado, debe ser liberado de la lista de procesos y adicionado a la lista de huecos.

- 13.** ¿Cuál fue la primera solución al problema que se presenta cuando los programas ya no caben en memoria?

Otro algoritmo de asignación es el de ajuste rápido, que consiste en tener listas independientes para algunos de los tamaños que se solicitan con mayor frecuencia. Si se solicita un espacio de tamaño típico (de los más frecuentes) se busca primero en tales listas. Igual sucede si se libera, se adiciona a una de ellas. Esto facilita la administración de la memoria puesto que se van a tener espacios libres de tamaños comunes en una lista, lo que facilita su búsqueda. Sin embargo el mantenimiento de las listas también es costoso en tiempo.

- 14.** ¿Cuál es la idea básica de la memoria virtual?

En informática, la memoria virtual es una técnica de gestión de la memoria que se encarga de que el sistema operativo disponga, tanto para el software de usuario como para sí mismo, de mayor cantidad de memoria que esté disponible físicamente. La mayoría de los ordenadores tienen cuatro

tipos de memoria: registros en la CPU, la memoria caché (tanto dentro como fuera del CPU), la memoria RAM y el disco duro. En ese orden, van de menor capacidad y mayor velocidad a mayor capacidad y menor velocidad.

**15. Describa detalladamente la técnica de administración de memoria virtual por paginación.**

La memoria virtual usualmente (pero no necesariamente) es implementada usando paginación. En paginación, los bits menos significativos de la dirección de memoria virtual son preservados y usados directamente como los bits de orden menos significativos de la dirección de memoria física. Los bits más significativos son usados como una clave en una o más tablas de traducción de direcciones (llamadas tablas de paginación), para encontrar la parte restante de la dirección física buscada.

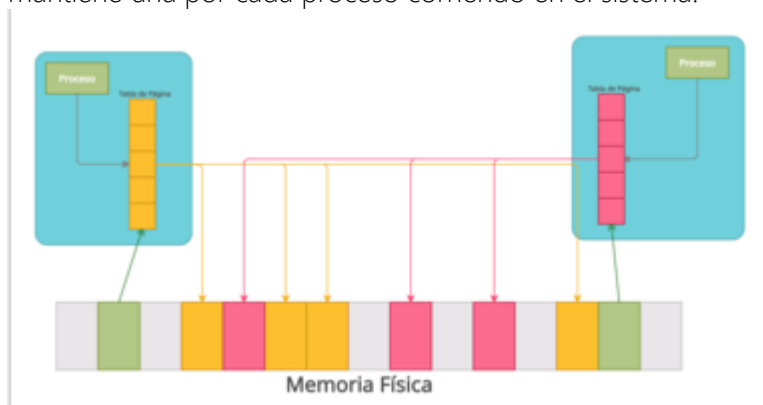
**16. ¿Cuál es el trabajo de la unidad de administración de memoria (MMU)?**

La unidad de gestión de memoria,<sup>1</sup> unidad de administración de memoria o unidad de manejo de memoria<sup>2</sup> (del inglés Memory Management Unit, MMU) es un dispositivo de hardware formado por un grupo de circuitos integrados, responsable del manejo de los accesos a la memoria por parte de la Unidad de Procesamiento Central (CPU) o procesador.

Entre las funciones de este dispositivo se encuentran la traducción de las direcciones lógicas (o virtuales) a direcciones físicas (o reales), la protección de la memoria, el control de caché y, en arquitecturas de computadoras más simples (especialmente en sistemas de 8 bits), bank switching.

**17. ¿Cuál es el propósito de las tablas de página?**

Las tablas de paginación o tablas de páginas son una parte integral del Sistema de Memoria Virtual en sistemas operativos, cuando se utiliza paginación. Son usadas para realizar las traducciones de direcciones de memoria virtual (o lógica) a memoria real (o física) y en general el sistema operativo mantiene una por cada proceso corriendo en el sistema.



**18. ¿Qué problemas se presentan las tablas de página?**

Los sistemas que utilizan tablas de páginas invertidas tienen problemas para implementar el concepto de memoria compartida, ya que, cada entrada de la tabla de páginas invertida corresponde a solo un marco en memoria.

Para crear una estructura de tabla de páginas que contenga asignaciones para cada página virtual en el espacio de direcciones virtuales se podría generar un desperdicio de espacio, podemos solucionar este problema de espacio excesivo colocando la tabla de páginas en la memoria virtual y dejando

que el sistema de memoria virtual administre la memoria que requiere la tabla de páginas. Sin embargo, parte de esta estructura de tabla de páginas lineal siempre debe permanecer residente en la memoria física para evitar errores de páginas circulares y buscar una página que ya no se encuentre en la memoria.

**19. ¿Por qué se presenta un fallo de página?**

En un sistema de memoria virtual paginada, un fallo de página (del inglés page fault) es una excepción arrojada cuando un programa informático requiere una dirección que no se encuentra en la memoria principal actualmente. Aunque el término sugiere un mal funcionamiento, se trata de un procedimiento normal dentro de la marcha del programa.

**20. ¿Qué bits componen una entrada de tabla? Describa cada uno de ellos.**

El formato de una entrada en la tabla de segmentos es: Base (24 bits) | Límite (16 bits). Sea un sistema con gestión de memoria paginada. La capacidad máxima de direccionamiento virtual es de 4 GB y la memoria principal es de 256 MB. El desplazamiento dentro de una página y de un marco se realiza con 12 bits.

**21. ¿Qué dice el algoritmo de reemplazo de página óptimo? ¿Cual es su principal problema?**

En sistemas operativos que utilizan paginación para el manejo de memoria, los algoritmos de reemplazo de páginas son usados para decidir qué páginas pueden ser sacadas de memoria cuando se necesita cargar una nueva y ya no hay marcos de páginas libres.

Este algoritmo debe de tener el menor índice de fallos de página de todos los algoritmos. En teoría, este algoritmo debe de reemplazar la página que no va a ser usada por el periodo más largo de tiempo. Desafortunadamente, el algoritmo de reemplazo óptimo es fácil en teoría, pero prácticamente imposible de implementar, dado que requiere conocer a futuro las necesidades del sistema. Tal algoritmo existe y ha sido llamado OPT o MIN, pero se usa únicamente para estudios de comparaciones

Características:

Es el que produce menos fallos de pagina para cualquier cantidad de marcos

Se reemplaza la pagina que va a tardar más tiempo en ser referenciada

No puede implementarse pues implicaría conocer de antemano las paginas que va a referenciar el proceso

· Aunque no puede implementarse se utiliza como referencia para los demás algoritmos

**22. En que consiste el algoritmo de sustitución de páginas no usadas recientemente.**

La mayoría de los ordenadores presentan los bits R (de referencia) y M (de modificación) en las entradas de la tabla de páginas, siendo estos bits actualizados vía hardware. Si el hardware no proporciona dichos bits, éstos pueden ser simulados mediante el software. Para ello, cuando se inicia un proceso se señalan todas sus entradas en la tabla de páginas como si no estuvieran dentro de la memoria. Si se hace referencia a cualquier página, ocurre un fallo de página. El sistema operativo activa entonces el bit R (en sus propias tablas) y cambia la entrada de la tabla de páginas para que apunte hacia la página correcta, poniendo dicha entrada en modo sólo lectura (recordar los bits de permisos). El proceso retoma su ejecución; si se escribe en la página, ocurre otra interrupción por violación de permisos, lo que permite al

sistema operativo activar el bit M en sus tablas, cambiando los permisos de la página a lectura y escritura.

Disponiendo de los citados bits R y M se puede construir el siguiente algoritmo: al iniciar un proceso, el sistema operativo asigna un valor 0 a los bits R y M de todas sus páginas. De manera periódica (cada interrupción del reloj) se pone a cero el bit R, para distinguir las páginas que no tienen referencias recientes de las que sí.

Ante un fallo de página, el sistema operativo inspecciona todas las páginas y las divide en cuatro categorías, según los valores actuales de los bits R y M:

Clase 0: no referenciada, ni modificada (0,0).

Clase 1: no referenciada, pero modificada (0,1).

Clase 2: referenciada, pero no modificada (1,0).

Clase 3: referenciada y modificada (1,1).

**23. Describa el algoritmo de sustitución de páginas de primera que entra, primera que sale (FIFO).**

El algoritmo de reemplazo más sencillo es el primero en entrar, primero en salir. Este algoritmo asocia a cada página el instante en el que se trajo a memoria. Cuando hay que reemplazar una página, se elige la más antigua. Para la cadena de referencias de ejemplo, y suponiendo un total de tres marcos, aplicamos este algoritmo con el resultado de 15 fallos de página.

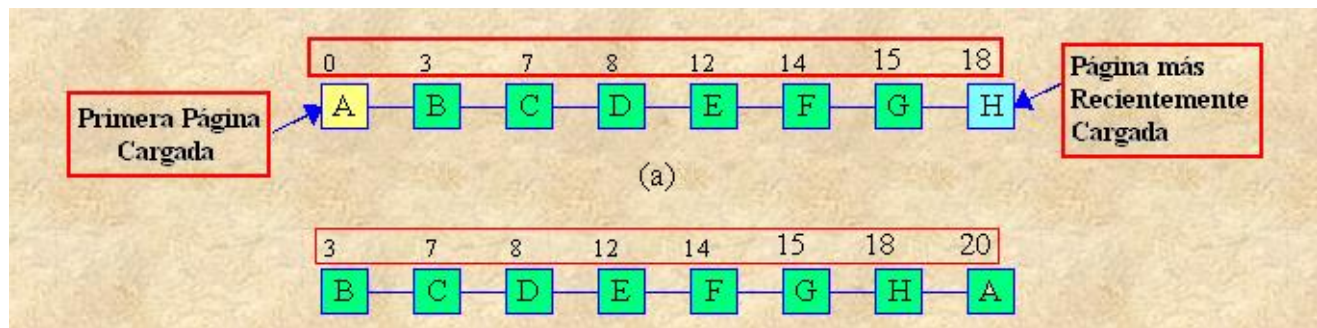
El algoritmo FIFO es fácil de comprender y programar, pero su rendimiento no siempre es bueno. La página reemplazada puede ser un módulo de asignación de valores iniciales que se utilizó hace mucho tiempo y que ya no se necesita. Pero también puede contener una variable cuyo valor inicial se asignó hace tiempo pero que se utiliza constantemente. Además, este algoritmo presenta una irregularidad denominada anomalía de Belady. Para ilustrarlo suponga que ahora tenemos la siguiente serie de referencias 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5. La figura 7.5 muestra la curva de fallos de páginas frente al número de marcos disponibles. Sorprendentemente el número de fallos para cuatro marcos (diez) es mayor que para tres marcos (nueve). Esto significa que con el algoritmo FIFO la tasa de fallos pueden aumentar al incrementar el número de marcos asignados.

**24. Describa en que consiste el algoritmo de sustitución de páginas de segunda oportunidad.**

Una modificación simple de FIFO, que evita deshacerse de una página de uso frecuente, inspecciona el bit R de la página más antigua. Si es 0, la página es antigua y no utilizada, por lo que se reemplaza de manera inmediata. Si el bit es 1, R se pone a cero, la página se coloca al final de la lista de páginas, como si hubiera llegado en ese momento a la memoria. Después continúa la búsqueda siguiendo la lista.

La operación de este algoritmo, llamado de la segunda oportunidad, se muestra en la figura 7.7. En la figura 7.7 (a), vemos las páginas de la A a la H, en una lista enlazada ordenada según el tiempo de llegada a la memoria.



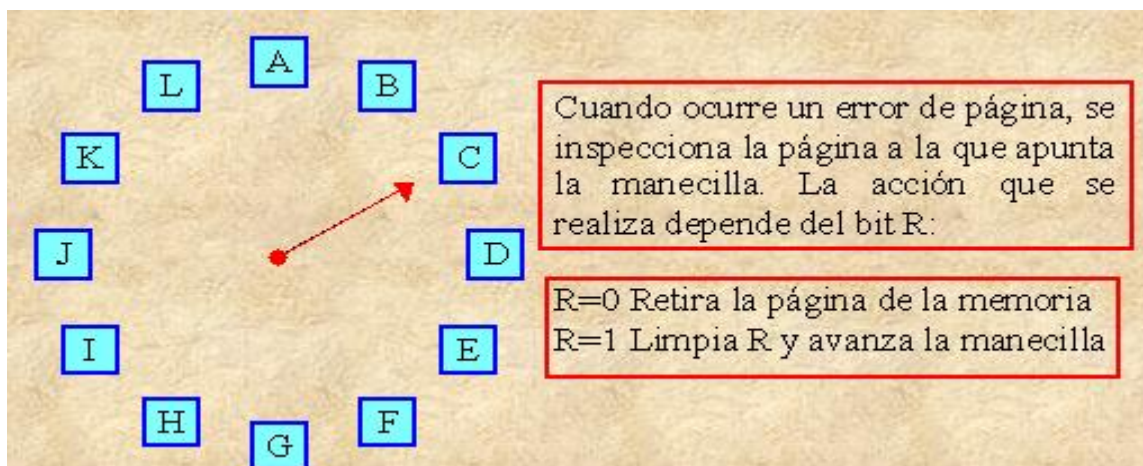


Supongamos que ocurre un fallo de página. La página más antigua es A, si A tiene el bit R a cero, se saca de la memoria. Por contra, si R vale 1, A se coloca al final de la lista, poniéndose a cero el bit R. La búsqueda de una página adecuada prosigue con B.

Lo que hace el algoritmo de la segunda oportunidad es buscar una página antigua sin referencias, si todas las páginas tienen alguna referencia deviene en un simple FIFO. En efecto, supongamos que todas las páginas de la figura 7.7 (a) tienen R a 1. Una a una, el sistema operativo traslada las páginas al final de la lista y pone su bit R a cero. Al final, se vuelve a la página A, la cual tiene R a cero. En este momento, A se retira de la memoria. Por lo tanto, el algoritmo siempre termina.

## 25. En que consiste el algoritmo de sustitución de páginas por reloj.

Aunque la segunda oportunidad es un algoritmo razonable, es ineficiente e innecesario, puesto que desplaza las páginas en una lista. Un mejor diseño consiste en mantener las páginas en una lista circular,



con la forma de un reloj, como se muestra en la figura 7.8. Una manecilla apunta hacia la página más antigua.

Al ocurrir un fallo de página, se inspecciona la página a la que apunta la manecilla. Si su bit R vale 0, la página se retira de la memoria, se inserta la nueva página en su lugar en el reloj, y la manecilla avanza una posición. Si R vale 1, este bit se pone a 0 y la manecilla avanza a la página siguiente. Este proceso continúa hasta encontrar una página con R a cero. Este algoritmo sólo difiere del anterior en su implementación.

**26. Describa en que consiste el algoritmo de sustitución de páginas menos recientemente usadas.**

La mayoría de los ordenadores presentan los bits R (de referencia) y M (de modificación) en las entradas de la tabla de páginas, siendo estos bits actualizados vía hardware. Si el hardware no proporciona dichos bits, éstos pueden ser simulados mediante el software. Para ello, cuando se inicia un proceso se señalan todas sus entradas en la tabla de páginas como si no estuvieran dentro de la memoria. Si se hace referencia a cualquier página, ocurre un fallo de página. El sistema operativo activa entonces el bit R (en sus propias tablas) y cambia la entrada de la tabla de páginas para que apunte hacia la página correcta, poniendo dicha entrada en modo sólo lectura (recordar los bits de permisos). El proceso retoma su ejecución; si se escribe en la página, ocurre otra interrupción por violación de permisos, lo que permite al sistema operativo activar el bit M en sus tablas, cambiando los permisos de la página a lectura y escritura.

Disponiendo de los citados bits R y M se puede construir el siguiente algoritmo: al iniciar un proceso, el sistema operativo asigna un valor 0 a los bits R y M de todas sus páginas. De manera periódica (cada interrupción del reloj) se pone a cero el bit R, para distinguir las páginas que no tienen referencias recientes de las que sí.

**27. ¿Qué son los segmentos y por que se emplean?**

Un aspecto importante de la gestión de la memoria que la paginación convierte en inevitable es la separación de la visión que el usuario tiene de la memoria y la memoria física real. La visión del usuario no coincide con la memoria física real. La visión del usuario se transforma en la memoria física. La traducción de direcciones permite esta diferencia entre la memoria lógica y la física.

La segmentación es un esquema de administración de la memoria que soporta la visión que el usuario tiene de la misma. Un espacio de direcciones lógicas es una colección de segmentos. Cada segmento tiene un nombre y una longitud. Las direcciones especifican tanto el nombre del segmento como el desplazamiento dentro del segmento. Por lo tanto, el usuario especifica cada dirección mediante dos cantidades: un nombre de segmento y un desplazamiento. (Compárese este esquema con la paginación, donde el usuario especificaba solamente una única dirección, que el hardware particionaba en número de página y desplazamiento, siendo todo ello invisible al programador).

Por simplicidad de implementación, los segmentos están numerados y se referencian por un número de segmento en lugar de por un nombre. Normalmente el programa de usuario se ensambla (o compila), y el ensamblador (o el compilador) construye automáticamente segmentos que reflejan el programa de entrada. Un compilador de Pascal podría crear segmentos separados para (1) las variables globales, (2) la pila de llamada de procedimientos, para almacenar parámetros y devolver direcciones, (3) el código de cada procedimiento o función, y (4) las variables locales de cada procedimiento y función. El cargador tomaría todos esos segmentos y les asignaría números de segmento.

**28. Como se especifica una dirección en memoria segmentada o bidimensional.**

Al igual que la paginación, la segmentación es también un esquema de administración de memoria. Es compatible con la vista del usuario de la memoria. El proceso se divide en segmentos de tamaño variable

y se carga en el espacio de direcciones de la memoria lógica.

El espacio de direcciones lógicas es la colección de segmentos de tamaño variable. Cada segmento tiene su nombre y longitud. Para la ejecución, los segmentos del espacio de la memoria lógica se cargan en el espacio de la memoria física.

Para poder acceder a una ubicación específica de la memoria, la CPU genera señales en el bus de dirección, que habitualmente tiene una longitud de 32 bits en la mayoría de máquinas actuales. Un bus de dirección de 32 bits permite especificar hasta  $2^{32} = 4.294.967.296$  direcciones de memoria distintas.

Las direcciones de memoria se expresan a menudo en código hexadecimal. Por ejemplo, para expresar el valor binario 111110101000000000010101100 se escribe 0x3F5000AC en hexadecimal.

**29. Realice una tabla comparativa entre paginación y segmentación.**

Bases para la comparación	Paginación	Segmentación
BASIC	Una página es de tamaño de bloque fijo.	Un segmento es de tamaño variable.
Fragmentación	La paginación puede conducir a la fragmentación interna.	La segmentación puede conducir a una fragmentación externa.
Dirección	La dirección especificada por el usuario se divide por CPU en un número de página y desplazamiento.	El usuario especifica cada dirección por dos cantidades, un número de segmento y el desplazamiento (límite de segmento).
tamaño	El hardware decide el tamaño de la página.	El tamaño del segmento es especificado por el usuario.
Mesa	La paginación implica una tabla de páginas que contiene la dirección base de cada página.	La segmentación implica la tabla de segmentos que contiene el número y el desplazamiento del segmento (longitud del segmento).

**30. En que difiere la implementación de la segmentación respecto a la paginación.**

La paginación y la segmentación son ambos esquemas de gestión de memoria. La paginación permite que la memoria se divida en bloques de tamaño fijo, mientras que la segmentación divide el espacio de memoria en segmentos del tamaño de bloque variable. Cuando la paginación conduce a una fragmentación interna, la segmentación conduce a una fragmentación externa.

Andrés, R. (2021, 28 septiembre). Esta es la cantidad de memoria RAM ideal que debe tener tu ordenador.

ComputerHoy. <https://computerhoy.com/reportajes/tecnologia/cantidad-memoria-ideal-ordenador-414879>

Wolf, G. (s. f.). Sistemas Operativos — Administración de memoria. sistop. Recuperado 24 de abril de 2022, de [http://sistop.gwolf.org/html/04\\_administracion\\_de\\_memoria.html](http://sistop.gwolf.org/html/04_administracion_de_memoria.html)

colaboradores de Wikipedia. (2022, 10 marzo). Gestión de memoria. Wikipedia, la enciclopedia libre. Recuperado 24 de abril de 2022, de [https://es.wikipedia.org/wiki/Gesti%C3%B3n\\_de\\_memoria#:~:text=La%20administraci%C3%B3n%20de%20memoria%20se,manera%20posible%20el%20espacio%20disponible](https://es.wikipedia.org/wiki/Gesti%C3%B3n_de_memoria#:~:text=La%20administraci%C3%B3n%20de%20memoria%20se,manera%20posible%20el%20espacio%20disponible).

Multiprogramación con particiones fijas - Sistemas Operativos. (s. f.). google sites. <https://sites.google.com/site/osupaep2010/administracion-de-memoria-1/multiprogramacion-con-particiones-fijas>

C. (2016, 18 mayo). Compresión / Compactación de Memoria. chsosuc201611701220057. <https://chsosuc201611701220057.wordpress.com/2016/04/30/compresion-compactacion-de-memoria/>