

Introducción a los Sistemas Operativos

Práctica 5

1.-Dirección Lógica y Física

Explique a que hacen referencia los siguientes términos:

- Dirección Lógica o Virtual
- Dirección Física
-

Dirección Lógica o Virtual

La **dirección lógica**, también llamada **dirección virtual**, es la **dirección generada por un proceso cuando accede a memoria**. Este término pertenece al espacio de direcciones del programa y es **gestionado** por la **Unidad de Gestión de Memoria (MMU)** del sistema operativo. La dirección lógica **no corresponde directamente a una ubicación física en la memoria RAM**.

- **Contexto:** Cada proceso tiene su propio espacio de direcciones lógicas, lo que significa que los programas pueden operar como si fueran los únicos en la memoria.
- **Uso:** La dirección lógica se utiliza en la **memoria virtual**, que permite **abstraer la memoria física disponible para el sistema**.
- **Conversión:** Las **direcciones lógicas son traducidas a direcciones físicas** mediante una tabla de páginas o un mecanismo equivalente, lo que permite implementar técnicas como la protección de memoria y el intercambio entre la memoria principal y el almacenamiento secundario.

Dirección Física

La **dirección física** se refiere a una **ubicación real en la memoria RAM del sistema**. Es la dirección que el hardware utiliza para acceder directamente a los datos almacenados.

- **Contexto:** Esta dirección está en el **espacio de direcciones físicas** de la memoria principal y es gestionada por el hardware y el sistema operativo.
- **Uso:** Las direcciones físicas son las que utilizan los dispositivos de hardware, como la CPU y los controladores de memoria, para acceder directamente a las celdas de memoria.
- **Relación con la dirección lógica:** La **MMU traduce** las **direcciones lógicas** generadas por los procesos en **direcciones físicas** utilizando estructuras como:
 - **Tablas de páginas:** Para sistemas con paginación.
 - **Segmentos:** Para sistemas con segmentación.

2.- Particiones

En la técnica de Particiones Múltiples, la memoria es dividida en varias particiones y los procesos son ubicados en estas, siempre que el tamaño del mismo sea menor o igual que el tamaño de la partición.

Al trabajar con particiones se pueden considerar 2 métodos (independientes entre sí):

Particiones Fijas

Particiones Dinámicas

a) Explique cómo trabajan estos 2 métodos. Cite diferencias, ventajas y desventajas.

Particiones Fijas

La **memoria principal** se divide en un número **fijo de particiones** de tamaño predeterminado al inicio del sistema.

- Cada partición puede contener un solo proceso.
- Los **tamaños de las particiones son fijos** y no cambian durante la ejecución.
- Si un proceso **requiere menos memoria** que la partición, se desperdicia el espacio restante (**fragmentación interna**).
- **Ventajas:**
 - **Simplicidad de implementación.**
 - Permite un acceso rápido y fácil gestión de la asignación de memoria.
- **Desventajas:**
 - Genera **fragmentación interna**: espacios desperdiciados en las particiones. Entre partición y partición puede quedar espacio inutilizado.
 - El número de particiones fija un límite al número de procesos que pueden ejecutarse simultáneamente.
 - **No es flexible** para procesos con necesidades de **memoria variables**.

Particiones Dinámicas

La memoria se divide en **particiones creadas dinámicamente** según las necesidades de cada proceso.

- Cada proceso recibe una **partición del tamaño exacto de su requerimiento de memoria**.
- A medida que los procesos terminan, se liberan las particiones asignadas, pero esto puede generar **fragmentación externa** (**espacios libres dispersos en la memoria**).
- **Ventajas:**
 - **Uso eficiente de la memoria**, ya que las particiones se ajustan al tamaño del proceso.
 - Permite una mayor **flexibilidad** para ejecutar procesos de diferentes tamaños.
- **Desventajas:**
 - Genera **fragmentación externa**: pequeños bloques de memoria libres que no pueden ser usados por nuevos procesos.

- La gestión es más compleja y puede requerir técnicas como la **compactación para reorganizar los bloques libres**.

b) ¿Qué información debe disponer el SO para poder administrar la memoria con estos métodos?

Particiones disponibles:

- Tamaño y ubicación (dirección de inicio y fin). (creo que seria lo mas importante para hacer la traduccion)
- Estado (ocupada/libre).

Procesos en memoria:

- Identificador del proceso.
- Tamaño requerido por el proceso.
- Ubicación de la partición asignada.

Fragmentación:

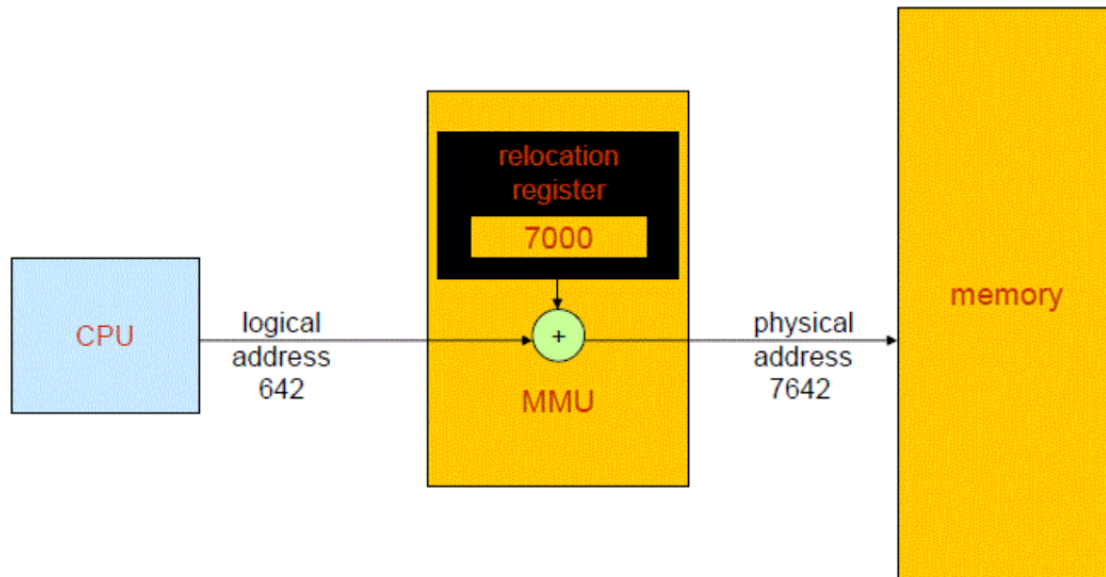
- Para particiones dinámicas, debe saber dónde están los bloques libres y su tamaño.

Tabla de asignación:

- Mapeo entre procesos y particiones para la traducción de direcciones lógicas a físicas.

c) Realice un gráfico indicando cómo realiza el SO la transformación de direcciones lógicas a direcciones físicas.

La MMU es la que hace la traducción de virtual a física. La lógica tiene una base y un desplazamiento para traducir se suman los 2 valores.



3.- Tipos de Particiones Fijas

Al trabajar con particiones fijas, los tamaños de las mismas se pueden considerar:

Particiones de igual tamaño.

Particiones de diferente tamaño.

Cite ventajas y desventajas de estos 2 métodos.

Particiones de igual tamaño

Ventajas:

- **Simplicidad de implementación:**
 - o Es más sencillo para el sistema operativo gestionar particiones de tamaño fijo e igual.
- **Asignación rápida:**
 - o Cualquier partición libre puede ser asignada al proceso **sin necesidad de evaluar su tamaño**.
- **Predecibilidad:**
 - o Se puede **estimar fácilmente cuántos procesos caben en la memoria**.

Desventajas:

- **Fragmentación interna:**
 - o Si el tamaño de un proceso es menor que la partición, el espacio restante se desperdicia.
- **Falta de flexibilidad:**

- Procesos más grandes que el tamaño de la partición no pueden ejecutarse, incluso si hay memoria suficiente distribuida entre varias particiones.
- **Ineficiencia para cargas heterogéneas:**
 - En sistemas con procesos de tamaños variados, muchas particiones podrían quedar parcialmente utilizadas.

Particiones de diferente tamaño

Ventajas:

- **Mejor ajuste:**
 - Los procesos se pueden asignar a particiones cuyo tamaño se aproxime más a sus necesidades, reduciendo la fragmentación interna.
- **Flexibilidad:**
 - Permite acomodar procesos de diferentes tamaños, siempre que exista una partición disponible con suficiente espacio.
- **Optimización del uso de memoria:**
 - Se aprovecha mejor la memoria disponible al asignar particiones de tamaño más adecuado.

Desventajas:

- **Complejidad en la gestión:**
 - El sistema operativo debe mantener un registro detallado de las particiones y encontrar la más adecuada para cada proceso.
- **Fragmentación externa:**
 - Pueden quedar bloques libres dispersos que no se pueden usar para procesos más grandes.
- **Decisión inicial crítica:**
 - Si los tamaños de las particiones no se distribuyen adecuadamente, pueden surgir problemas de desbalanceo (procesos esperando una partición grande mientras otras pequeñas permanecen libres).

4.- Fragmentación

Ambos métodos de particiones presentan el problema de la fragmentación:

Fragmentación Interna (Para el caso de Particiones Fijas)

Fragmentación Externa (Para el caso de Particiones Dinámicas)

a) Explique a que hacen referencia estos 2 problemas

Fragmentacion interna

Ocurre cuando quedan **espacios de memoria inutilizados** por haber **asignado las particiones antes de ejecutar el proceso**. Pasa en las particiones fijas. Es una **porción de memoria** que queda **sin utilizar**.

Fragmentacion Externa

Son **huecos** que van **quedando en la memoria** a medida que los **procesos finalizan**.

Es un **desperdicio entre bloques no asignados**.

b) El problema de la Fragmentación Externa es posible de subsanar. Explique una técnica que evite este problema.

Compactar memoria (reorganizar bloques).

5.- Paginación

a) Explique cómo trabaja este método de asignación de memoria.

Se **reparte la memoria en marcos** que son porciones pequeñas de memoria. El espacio de direcciones de memoria se divide en porciones de igual tamaño llamadas páginas.

Se mantienen una tabla de páginas para cada proceso que contiene marcos donde se contiene cada página.

Cuando un proceso necesita memoria, sus páginas se asignan a marcos disponibles en la memoria principal. Las páginas pueden estar dispersas físicamente, pero se presentan al proceso como un espacio contiguo gracias a la traducción lógica-física.

b) ¿Qué estructuras adicionales debe poseer el SO para llevar a cabo su implementación?

Tabla de páginas:

- Una estructura que almacena el mapeo entre páginas lógicas y marcos físicos.
- Cada proceso tiene su propia tabla.

Registro base de la tabla de páginas:

- Apunta al inicio de la tabla de páginas del proceso activo.

TLB (Translation Lookaside Buffer):

- Cache especial que acelera el acceso a la tabla de páginas al almacenar entradas recientemente usadas.

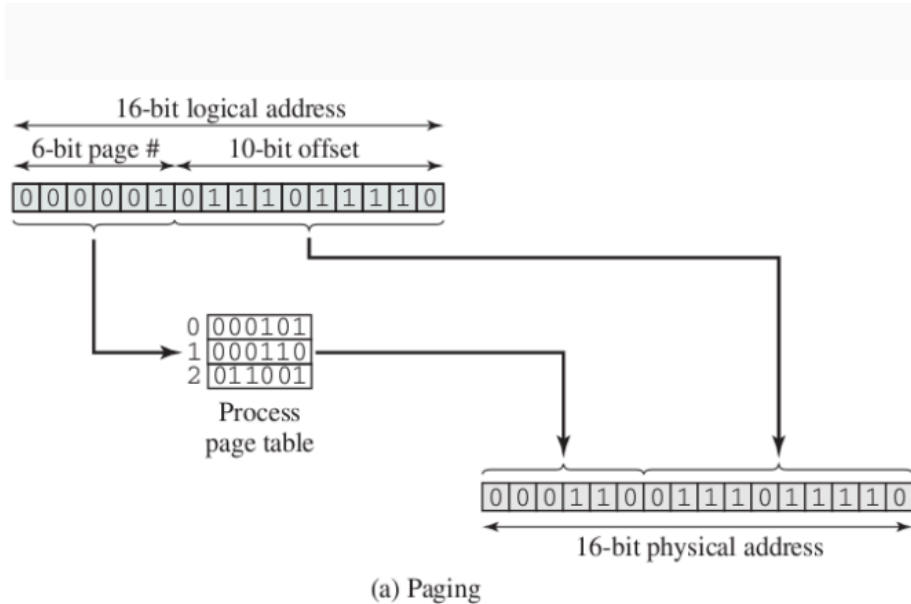
Bit de validez:

- Indica si una página está presente en la memoria física o debe ser cargada desde disco.

Algoritmo de reemplazo de páginas (lo agregue xq me gusta aunque no es necesario)

- Necesario si no hay marcos libres y se requiere liberar espacio en memoria.

c) Explique, utilizando gráficos, como son transformadas las direcciones lógicas en físicas.



d) En este esquema: ¿Se puede producir fragmentación (interna y/o externa)?

Sí puede ocurrir la fragmentación interna, si un proceso no utiliza todo el espacio de una página, el espacio restante dentro del marco asignado se desperdicia. Lo bueno de la paginación es que elimina la fragmentación externa.

6.- Paginación y Particiones Fijas

Cite similitudes y diferencias entre la técnica de paginación y la de particiones fijas.

Similitudes

- La paginación y las particiones fijas eliminan la fragmentación externa pero las 2 pueden tener fragmentación interna.
- Dividen la memoria en bloques
- Pero pueden tener fragmentación interna si queda memoria sin utilizar en los bloques asignados.

Diferencias

- La **paginación** requiere la **tabla de páginas** para mapear donde se encuentra realmente su página

- La **paginación** es más **eficiente** ya que no asigna toda la memoria si no la va a usar
- La **paginación** permite que se **expandan dinámicamente** los procesos si necesitan más memoria.
- En las **particiones fijas** los procesos están en **porciones contiguas de memoria**, mientras que en la paginación están dispersos en la memoria.
- En la paginación un proceso puede ocupar diferentes páginas, mientras que en las particiones fijas solo puede estar en una partición.

7.- Calculos de memoria (paginación)

Suponga un sistema donde la memoria es administrada mediante la técnica de paginación, y donde:

El tamaño de la página es de 512 bytes

Cada dirección de memoria referencia 1 byte.

Los marcos en memoria principal se encuentran desde la dirección física 0.

Suponga además un proceso con un tamaño 2000 bytes y con la siguiente tabla de páginas:

Página	Marco
0	3
1	5
2	2
3	6

a) Realice los gráficos necesarios (de la memoria, proceso y tabla de paginas) en el que reflejen el estado descrito.

Memoria PPal		
Marcos	Paginas	Dir Logica
0	-	0..511
1	-	512..1023
2	2	1024..1535
3	0	1536..2047
4	-	2048..2559
5	1	2560..3071
6	3	3072..

b) Indicar si las siguientes direcciones lógicas son correctas y en caso afirmativo indicar la dirección física a la que corresponden:

i) 35

ii) 512

iii) 2051

iv) 0

v) 1325

vi) 602

$n^{\circ}\text{página} = \text{dir lógica} \div \text{tam de pag}$

$\text{Desplazamiento} = \text{dir lógica} \bmod \text{tam de página}$

DIR Fisica= (marco log \times tam de c pagina)+Desplazamiento.

Dir Logica	Marco	Pagina	Desplazamiento	Dir Fisica	Válida
35	3	0	35	$1536 + 35 = 1571$	si
512	5	1	0	$2560 + 0 = 2560$	si
2051	??	4	3	??	no
0	3	0	0	$1536 + 0 = 1536$	si
1325	2	2	301	$1024 + 301 = 1325$	si
602	5	1	90	$2560 + 90 = 2650$	si

c) Indicar, en caso de ser posible, las direcciones lógicas del proceso que se corresponden si las siguientes direcciones físicas:

i) 509

ii) 1500

iii) 0

iv) 3215

v) 1024

vi) 2000

marco= dir fisica div 512

desplazamiento= dir lógica mod 512

DIR LOGICA= (Pagina logica× tam de c pag)+Desplazamiento.

Dir Fisica	Marco	Pagina	Desplazamiento	Dir Logica	Valida
509	0	??	509	??	No PF
1500	2	2	476	$(512*2)+476=1500$	si
0	0	??	0	??	no
3215	6	3	143	$(512*3)+143=3215$	si
1024	2	2	0	$(512*2)+0=1024$	si
2000	3	0	464	$(0*2000)+464=464$	si

d) ¿Indique, en caso que se produzca, la fragmentación (interna y/o externa)?

8.-

Considere un espacio lógico de 8 páginas de 1024 bytes cada una, mapeadas en una memoria física de 32 marcos.

a) ¿Cuántos bits son necesarios para representar una dirección lógica?

Son necesarios 3 bits para las páginas ($2^3=8$) y 10 bits porque cada página tiene 1024 bytes (2^{10}) en total 13 bits

b) ¿Cuántos bits son necesarios para representar una dirección física?

Se necesitan para los marcos 5 bits ($2^5=32$) y se necesitan 10 bits para el desplazamiento porque es el mismo que las páginas, en total 15 bits.

9.- Segmentación

a) Explique cómo trabaja este método de asignación de memoria.

La **segmentación** es una técnica de gestión de memoria en la que un **proceso se divide en segmentos de tamaño variable**, donde **cada segmento** puede contener **una parte diferente** del programa.

En lugar de tratar toda la memoria como una secuencia lineal de direcciones (como en la paginación), la segmentación permite que las **direcciones lógicas** se compongan de dos partes:

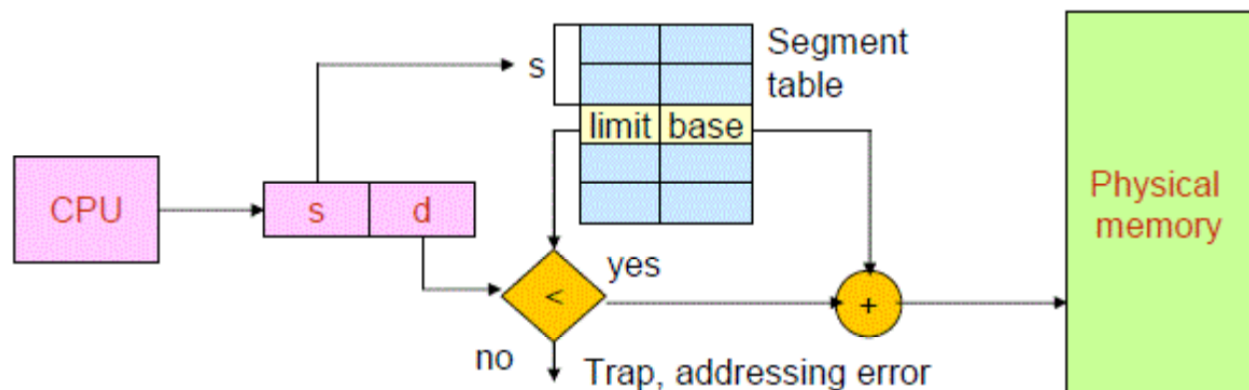
1. **Número de segmento**: Identifica qué parte del proceso está siendo referenciada.
2. **Desplazamiento dentro del segmento**: Especifica la posición dentro del segmento.

El sistema operativo mantiene una **tabla de segmentos** que almacena la **base (dirección de inicio)** y el **límite (tamaño)** de cada segmento. Cada vez que un proceso intenta acceder a una dirección, la dirección lógica es convertida a una dirección física usando esta información.

b) ¿Qué estructuras adicionales debe poseer el SO para llevar a cabo su implementación?

El SO tiene que tener una tabla de segmentos que indique base y límite, el registro de segmento y el bit de validación

c) Explique, utilizando gráficos, como son transformadas las direcciones lógicas en físicas.



d) En este esquema: ¿Se puede producir fragmentación (interna y/o externa)?

Se puede producir **fragmentación externa**, se produce cuando hay **espacio libre en la memoria**, pero **no es suficiente para acomodar un segmento completo**. Aunque hay espacio, no está contiguo, lo que impide que nuevos segmentos se ubiquen allí.

Y también se puede producir **fragmentación interna**, puede ocurrir en situaciones donde los **segmentos son más grandes que los datos que almacenan**.

10.- Segmentación y Particiones Dinámicas

Cite similitudes y diferencias entre la técnica de segmentación y la de particiones dinámicas.

Similitudes

- Permiten la asignación de memoria en bloques de tamaño variable.
- Son susceptibles a la **fragmentación externa**. Esto ocurre cuando se producen huecos de memoria no contiguos debido a la asignación y liberación dinámica de bloques de memoria, lo que impide que la memoria se utilice de manera eficiente.

- El sistema operativo necesita mantener tablas que sigan el estado de la memoria.

Diferencias

- En segmentación la memoria está dividida en segmentos y en particiones está dividida en particiones
- En segmentación la dirección lógica se divide en dos partes: el número de segmento y el desplazamiento dentro del segmento. Mientras que en particiones dinámicas se dirige directamente a la memoria.
- En segmentación los segmentos no son contiguos y pueden generar fragmentación externa.

11.- Paginacion y Segmentacion Diferencias y Similitudes

Cite similitudes y diferencias entre la técnica de paginación y segmentación.

Similitudes

Gestión de memoria no contigua:

- Tanto la paginación como la segmentación permiten que un **proceso se divida en partes no contiguas dentro de la memoria física**, lo que ayuda a optimizar el uso de la memoria y a evitar problemas de memoria insuficiente para procesos grandes.

Uso de tablas para la traducción:

- Ambos métodos requieren estructuras de datos adicionales (**tablas**) para mapear las **direcciones lógicas a direcciones físicas**.
 - **Paginación:** Tabla de páginas.
 - **Segmentación:** Tabla de segmentos.

Traducción de direcciones:

- En ambos casos, la **dirección lógica** debe **transformarse** en una **dirección física** mediante el uso de las tablas correspondientes.

Tema	Paginación	Segmentación
Unidad de división	Se divide la memoria en páginas	Se divide la memoria en segmentos
Tamaño	Las páginas todas tienen el mismo tamaño	Cada segmento tiene un tamaño diferente, dependiendo las necesidades del proceso
Fragmentacion	Produce Fragmentación interna	Produce Fragmentación externa por los

	si un proceso no llena toda la página que se le asignó	espacios entre los segmentos asignados
Dirección lógica	Se divide en el número de la página y un desplazamiento	Se divide en el número del segmento y un desplazamiento dentro del segmento.
Tabla	Tiene una tabla de páginas que cada página tiene entradas donde cada una apunta al marco físico correspondiente	Tiene una tabla de segmentos que cada entrada tiene el límite y la base del segmento
Flexibilidad	Poco flexible ya que las páginas tienen el mismo tamaño	Flexible ya que los segmentos pueden adaptarse al tamaño lógico del programa
Propósito	Para administrar la memoria de manera más eficiente sin importar la estructura del sistema	Para reflejar la estructura lógica del programa
Traducción de direcciones	Simple ya que solo implica el mapeo entre los marcos y las páginas	Compleja porque implica verificar límites y realizar cálculos de desplazamiento

12.- Cálculos de Segmentación Paginada

Dado un S.O. que administra la memoria por medio de segmentación paginada, y teniéndose disponibles las siguientes tablas:

Tabla de Páginas

Nro. Segmento	Nro. Página	Direc. Base
1	1	40
	2	80
	3	60
2	1	20
	2	25
	3	0
3	1	120
	2	150

Tabla de Segmentos

Núm. Seg.	Dir. base
1	500
2	1500
3	5000

Indicar las direcciones físicas correspondientes a las siguientes direcciones lógicas (segmento, página, desplazamiento):

i) (2,1,1)

ii) (1,3,15)

iii) (3,1,10)

iv) (2,3,5)

Dirección base del segmento+Dirección base de la página+Desplazamiento= dirección lógica

- i) $1500 + 20 + 1 = 1521$
- ii) $500 + 60 + 15 = 575$
- iii) $5000 + 120 + 10 = 5130$
- iv) $1500 + 0 + 5 = 1505$

13.- Memoria Virtual

a) Describa qué beneficios introduce este esquema de administración de la memoria.

Permite trabajar con programas más grandes que la memoria física:

- Un proceso puede utilizar más espacio lógico del que realmente está disponible en memoria principal.

Aislamiento y seguridad entre procesos:

- Cada proceso tiene su propio espacio de direcciones virtuales, evitando accesos indebidos a la memoria de otros procesos.

Eficiencia en la utilización de memoria:

- Solo las páginas necesarias se cargan en memoria, dejando más espacio disponible para otros procesos.

Facilita la multitarea:

- Permite que varios procesos se ejecuten simultáneamente, aprovechando al máximo los recursos del sistema.

Simplifica la programación:

- Los desarrolladores no necesitan preocuparse por la administración de memoria física, ya que el sistema operativo se encarga de mapear las direcciones virtuales a las físicas.

b) ¿En qué se debe apoyar el SO para su implementación?

Hardware:

- **Unidad de Gestión de Memoria (MMU):** Transforma direcciones virtuales en direcciones físicas en tiempo real.
- **Tablas de páginas:** Para almacenar los mapeos entre páginas virtuales y marcos físicos.

- **Bits adicionales:** Bits como el *bit de presente/ausente* (P) y el *bit de modificado* (M) que indican el estado de las páginas.

Software:

- **Algoritmos de reemplazo de páginas:** Como *FIFO*, *LRU*, *NRU*, etc., para decidir qué página reemplazar cuando no hay marcos libres.
- **Gestor de interrupciones:** Para *manejar los page faults* (fallos de página) cuando un proceso intenta acceder a una página que no está en memoria.

Dispositivo de almacenamiento secundario:

- **Área de intercambio (swap):** Parte del disco duro utilizada para *almacenar* temporalmente las *páginas que no están en memoria*.

c) Al implementar esta técnica utilizando paginación por demanda, las tablas de páginas de un proceso deben contar con información adicional además del marco donde se encuentra la página. ¿Cuál es esta información? ¿Por qué es necesaria?

- **Bit de presente/ausente (P):**
 - Indica si la *página está en memoria principal (1) o en disco (0)*.
 - **Necesidad:** Dice si el *acceso a la página requiere un fallo de página* y de su carga futura.
- **Bit de acceso (A):**
 - Indica si la *página ha sido leída o escrita recientemente*.
 - **Necesidad:** Utilizado por *algoritmos de reemplazo para decidir qué página retirar*.
- **Bit de modificado o dirty (M):**
 - Indica si la *página ha sido modificada desde que se cargó*.
 - **Necesidad:** *Evita escribir páginas no modificadas* en disco durante un reemplazo.
- **Referencia al marco (si está en memoria):**
 - **Necesidad:** Para *identificar la ubicación física de la página* en caso de estar cargada.
- **Información de permisos:**
 - Define si la *página es de solo lectura, lectura/escritura, o ejecutable*.
 - **Necesidad:** *Previene errores* como escritura en áreas de código o ejecución de datos.
- **Bit de protección o validación:**
 - Indica si el *proceso tiene permiso* para acceder a la página.
 - **Necesidad:** *Refuerza la seguridad* y evita violaciones de acceso.

14.- Fallos de Página (Page Faults):

a) ¿Cuándo se producen?

Se producen cuando un proceso intenta acceder a una página que:

1. No está cargada en la memoria principal (marcos de memoria).
2. Está presente únicamente en el almacenamiento secundario (swap o disco).

b) ¿Quién es responsable de detectar un fallo de página?

La MMU esta al intentar acceder a la tabla de páginas verifica el bit de ausente o presente y si esta no esta en memoria (el bit está en ausente) se hace una trap(una interrupción) para notificar al so.

c) Describa las acciones que emprende el SO cuando se produce un fallo de página.

1. Se genera el trap
2. Se interrumpe y se bloquea el proceso
3. Se identifica la página a ser cargada a memoria
4. Se comprueba si hay espacio en memoria
 - Si hay marcos disponibles se carga la pagina directamente
 - Si no hay marcos disponibles se usa algún algoritmo de reemplazo de página
5. La página se carga desde el almacenamiento secundario al marco seleccionado en memoria principal.
6. Se actualiza la tabla de páginas bit de presente/ausente =1 y se registra el número de marco donde se encuentra la página
7. El proceso pasa de bloqueado a ready to run
8. Se actualizan los registros del CPU para continuar la ejecución del proceso desde la instrucción interrumpida.

15.- Direcciones:

a) Si se dispone de una espacio de direcciones virtuales de 32 bits, donde cada dirección referencia 1 byte:

i) ¿Cuál es el tamaño máximo de un proceso (recordar “espacio virtual”)?

ii) Si el tamaño de página es de 512Kb. ¿Cuál es el número máximo de páginas que puede tener un proceso?

iii) Si el tamaño de página es de 512Kb. y se disponen de 256 Mb. de memoria real ¿Cuál es el número de marcos que puede haber?

iv) Si se utilizaran 2 Kb. para cada entrada en la tabla de páginas de un proceso: ¿Cuál sería el tamaño máximo de la tabla de páginas de cada proceso?

i)

$(\text{cant de direcciones}) * (\text{tam. de referencia}) = \text{tamaño máx de proceso}$

$2^{32} * 1 \text{ byte} = 4.294.967.296 * 1 \text{ byte}$

ii)

Tamaño máx. del proceso / tamaño de página = número máximo de páginas

$4.294.967.296 \text{ bytes} / 1024 = 4194304 \text{ kb}$

$4194304 \text{ kb} / 512 \text{ kb} = 8192$

iii)

Numero de paginas / cantidad de memoria = cantidad de marcos

Página = 512 KB Memoria = 262144 KB

$262144 \text{ KiB} / 512 \text{ KiB} = 512 \text{ marcos}$

iv) $4.294.967.296 * 2 \text{ kb} = 8,589,934,592 \text{ kb}$

16.- Tablas de Paginas

Como se vio en el ejercicio anterior, la tabla de páginas de un proceso puede alcanzar un tamaño considerablemente grande, que incluso, no podría almacenarse de manera completa en la memoria real. Es por esto que el SO también realiza paginación sobre las tablas de páginas.

Existen varios enfoques para administrar las tablas de páginas:

- Tablas de páginas de 1 nivel.
- Tablas de páginas de 2 niveles.
- Tablas de páginas invertidas.

Explique brevemente cómo trabajan estos enfoques e indique cómo se realiza la transformación de la dirección virtual en dirección física.

Tablas de Páginas de 1 Nivel

Este esquema utiliza una **única tabla de páginas** para mapear cada página del espacio de direcciones virtuales a un marco en la memoria física.

- **Cómo funciona:**

- La **dirección virtual** se divide en dos partes:
 1. **Número de página:** Se usa como índice para buscar en la tabla de páginas.
 2. **Desplazamiento dentro de la página:** Se agrega al marco recuperado de la tabla para calcular la dirección física.
- La **tabla de páginas** contiene una **entrada por página virtual**, con información como el número de marco y bits de control (presente/ausente, acceso, dirty).

- **Transformación de dirección virtual a física:**

- **Extraer el número de página y el desplazamiento.**

- Consultar la tabla de páginas para encontrar el marco correspondiente.
- Combinar el número de marco con el desplazamiento para obtener la dirección física.
- **Ventajas:**
 - Sencilla de implementar.
- **Desventajas:**
 - Puede ser ineficiente si el espacio de direcciones virtuales es muy grande, ya que la tabla de páginas puede ocupar mucho espacio.

Tablas de Páginas de 2 Niveles

Diseñadas para gestionar mejor espacios de direcciones virtuales grandes, dividiendo la tabla de páginas en dos niveles jerárquicos.

- **Cómo funciona:**
 - La dirección virtual se divide en tres partes:
 1. **Índice del directorio de páginas (primer nivel):** Señala una tabla secundaria.
 2. **Índice en la tabla secundaria (segundo nivel):** Apunta al marco correspondiente.
 3. **Desplazamiento dentro de la página.**
 - El primer nivel (directorio de páginas) contiene punteros a tablas secundarias que almacenan el mapeo real de páginas virtuales a marcos físicos.
- **Transformación de dirección virtual a física:**
 - Extraer el índice del directorio y buscar la tabla secundaria correspondiente.
 - Usar el índice en la tabla secundaria para encontrar el marco físico.
 - Combinar el marco físico con el desplazamiento para obtener la dirección física.
- **Ventajas:**
 - Reduce el espacio necesario para las tablas, ya que solo se crean tablas secundarias para las páginas usadas.
- **Desventajas:**
 - Incrementa la complejidad y el tiempo de acceso debido a las búsquedas en dos niveles.

Tablas de Páginas Invertidas

En lugar de mantener una entrada por página virtual, mantienen una entrada por marco físico. Este esquema se utiliza para ahorrar memoria en sistemas con grandes espacios de direcciones virtuales.

- **Cómo funciona:**
 - Cada entrada en la tabla de páginas invertida corresponde a un marco físico e incluye:
 1. El número de página virtual almacenada en ese marco.
 2. El identificador del proceso dueño de la página (si aplica).

- Para **traducir** direcciones, se debe **buscar la página virtual en la tabla invertida** (generalmente usando una tabla hash para acelerar la búsqueda).
- **Transformación de dirección virtual a física:**
 - Utilizar el **número de página virtual** como clave para **buscar en la tabla hash**.
 - **Encontrar el marco físico** correspondiente a través de la **tabla invertida**.
 - Combinar el **marco** con el **desplazamiento** para obtener la **dirección física**.
- **Ventajas:**
 - **Ahorra espacio** al eliminar la necesidad de una entrada por página virtual.
- **Desventajas:**
 - Las **búsquedas pueden ser más lentas** si no se optimizan con estructuras adicionales como tablas hash.

17.-

Suponga que la tabla de páginas para un proceso que se está ejecutando es la que se muestra a continuación:

Página	Bit V	Bit R	Bit M	Marco
0	1	1	0	4
1	1	1	1	7
2	0	0	0	-
3	1	0	0	2
4	0	0	0	-
5	1	0	1	0

Asumiendo que:

El tamaño de la página es de 512 bytes

Cada dirección de memoria referencia 1 byte

Los marcos se encuentran contiguos y en orden en memoria (0, 1, 2..) a partir de la dirección real 0.

¿Qué dirección física, si existe, correspondería a cada una de las siguientes direcciones virtuales? (No gestione ningún fallo de página, si se produce)

a) 1052 b) 2221 c) 5499 d) 3101

Dir virtual div tamaño de pag= n ° de página

Dir virtual mod tamaño de página = desplazamiento

Dir física = base + desplazamiento

a) $1052 \div 512 = 2$

$1052 \bmod 512 = 28$

20+ ?? fallo de página

b) $2221 \div 512 = 4$

$2221 \bmod 512 = 173$

173+ ?? fallo de página

c) $5499 \div 512 = 10$

$5499 \bmod 512 = 379$

379+ ??? inválido

d) $3101 \div 512 = 6$

$3101 \bmod 512 = 29$

29 + ??? inválido?

18.- Tamaño de la página

Tamaño de la Página: La selección del tamaño de la página influye de manera directa sobre el funcionamiento de la memoria virtual.

Compare las siguientes situaciones con respecto al tamaño de página, indicando ventajas y desventajas:

- Un tamaño de página pequeño.
- Un tamaño de página grande.

Tamaño de Página Pequeño

- **Ventajas:**
 1. **Menor fragmentación interna:** Como las páginas son pequeñas, el desperdicio de memoria dentro de las páginas no completamente ocupadas (fragmentación interna) es menor.
 2. **Uso eficiente de la memoria:** Al asignar páginas más pequeñas, se puede ajustar más finamente la memoria a las necesidades de los procesos.
 3. **Mayor flexibilidad:** Permite acomodar más procesos en memoria al dividir la memoria física en bloques más pequeños.

4. **Caché más eficiente:** Las páginas pequeñas tienen menos probabilidad de sobrepasar la capacidad de las cachés.
- **Desventajas:**
 1. **Tamaño mayor de las tablas de páginas:** Con **más páginas** para gestionar, las **tablas** de páginas **crecen**, ocupando más memoria.
 2. **Mayor sobrecarga del sistema:** **Más referencias** a la tabla de páginas y **más interrupciones** (TLB miss) pueden reducir el rendimiento.
 3. **Mayor tiempo de búsqueda:** Se requieren **más consultas** para localizar páginas en memoria, lo que puede incrementar el tiempo de acceso.

Tamaño de Página Grande

- **Ventajas:**
 1. **Menor tamaño de las tablas de páginas:** Al necesitar menos páginas para el mismo espacio, las tablas son más pequeñas, reduciendo la sobrecarga de almacenamiento.
 2. **Menor cantidad de fallos de página:** Las **páginas más grandes** contienen **más datos**, lo que aumenta la probabilidad de que los datos requeridos estén ya cargados en memoria.
 3. **Eficiencia en transferencias de disco:** Los **sistemas** de memoria virtual cargan y escriben **páginas completas**; cargar páginas grandes es más eficiente para operaciones de E/S.
- **Desventajas:**
 1. **Mayor fragmentación interna:** Es probable que las páginas más grandes no se llenen completamente, lo que genera desperdicio de memoria.
 2. **Menor flexibilidad:** Puede ser más difícil acomodar varios procesos en memoria debido al **tamaño de los bloques**.
 3. **Efecto negativo en la caché:** Las **páginas grandes** pueden ocupar **más espacio** en la memoria caché, reduciendo su efectividad para otros procesos o datos.

19.- Asignación de marcos a un proceso (Conjunto de trabajo o Working Set):

Con la memoria virtual paginada, no se requiere que todas las páginas de un proceso se encuentren en la memoria. El SO debe controlar cuantas páginas de un proceso puede tener en la memoria principal. Existen 2 políticas que se pueden utilizar:

Asignación Fija

Asignación Dinámica.

a) Describa cómo trabajan estas 2 políticas.

b) Dada la siguiente tabla de procesos y las páginas que ellos ocupan, y teniéndose 40 marcos en la memoria principal, cuántos marcos le corresponden a cada proceso si se usa la técnica de Asignación Fija:

i) Reparto Equitativo

ii) Reparto Proporcional

Proceso	Total de Paginas Usadas
1	15
2	20
3	20
4	8

c) ¿Cuál de los 2 repartos usados en b) resultó más eficiente? ¿Por qué?

a)

Asignación Fija: a cada proceso se le asigna una cantidad arbitraria de marco.

A su vez para el reparto se puede usar:

- Reparto equitativo: se asigna la misma cantidad de marcos a cada proceso $\rightarrow m \text{ div } p$
- Reparto proporcional: se asignan marco en base a la necesidad que tiene cada proceso $\rightarrow V_p \cdot m / V_t$
 v_p = páginas usadas por proceso
 V_t = sumatoria de las v_p

Asignación dinámica: los procesos se van cargando en forma dinámica de acuerdo a la cantidad de marcos que necesiten

b) Si se usa el reparto equitativo a cada proceso se le asigna 10 marcos

Si se usa el reparto proporcional son

$$V_t = 15 + 20 + 20 + 8 = 63$$

$$\text{Proceso 1} \rightarrow 15 \cdot 40 / 63 = 9,52 \sim 10$$

$$\text{Proceso 2} \rightarrow 20 \cdot 40 / 63 \sim 13$$

$$\text{Proceso 3} \rightarrow 20 \cdot 40 / 63 \sim 13$$

$$\text{Proceso 4} \rightarrow 8 \cdot 40 / 63 \sim 5$$

c) Resultó más eficiente el reparto proporcional ya que en este quedan marcos que se pueden utilizar si otro proceso necesita marcos.

20.- Reemplazo de páginas (selección de una víctima):

¿Qué sucede cuando todos los marcos en la memoria principal están usados por las páginas de los procesos y se produce un fallo de página? El SO debe seleccionar una de las páginas que se encuentra en memoria como víctima, y ser reemplazada por la nueva página que produjo el fallo.

Considere los siguientes algoritmos de selección de víctimas básicos:

LRU

FIFO

OPT (Optimo)

Segunda Chance

a) Clasifique estos algoritmos de malo a bueno de acuerdo a la tasa de fallos de página que se obtienen al utilizarlos.

FIFO (First In, First Out):

Este algoritmo tiende a ser menos eficiente debido a que no considera el uso reciente de las

páginas. Puede reemplazar páginas que aún son útiles, generando más fallos. Es vulnerable al *anomalía de Belady* (mayor cantidad de marcos no garantiza menor tasa de fallos).

Segunda Oportunidad (Second Chance):

Mejora sobre FIFO al introducir un bit de referencia, lo que permite evitar reemplazar páginas que se han utilizado recientemente. Sin embargo, aún puede tener un desempeño intermedio.

LRU (Least Recently Used):

Este algoritmo elige reemplazar la página que no se ha utilizado durante más tiempo. Su rendimiento suele ser mejor porque aprovecha el principio de *localidad temporal*, aunque puede ser costoso en términos de implementación.

OPT (Óptimo):

Este es el mejor algoritmo en términos de tasa de fallos porque reemplaza la página que no se necesitará por más tiempo en el futuro. Sin embargo, no es práctico porque requiere conocimiento previo del acceso futuro (se utiliza como referencia teórica).

b) Analice su funcionamiento. ¿Cómo los implementaría?

FIFO:

- **Funcionamiento:** Se reemplaza la página más antigua (la primera en ingresar).
- **Implementación:**
Usar una cola para rastrear el orden de llegada de las páginas. Cuando ocurre un fallo, se elimina la página al frente de la cola y se agrega la nueva al final.

Segunda Oportunidad (Second Chance):

- **Funcionamiento:** Es una variante de FIFO que verifica si la página al frente de la cola ha sido referenciada recientemente (bit de referencia = 1). Si lo ha sido, se le da una "segunda oportunidad" y se mueve al final de la cola. Si no, se reemplaza.
- **Implementación:**
 - Mantener una cola circular de páginas.
 - Asociar un bit de referencia con cada página.
 - Avanzar un puntero hasta encontrar una página con bit de referencia = 0 para reemplazarla.
 - Si el bit es 1, se resetea a 0 y se mueve al final.

LRU (Least Recently Used):

- **Funcionamiento:** Se reemplaza la página menos utilizada recientemente.
 - **Implementación:**
 - **Opción 1:** Usar una pila donde las páginas usadas recientemente se mueven a la cima.
 - **Opción 2:** Usar una tabla de tiempos donde cada entrada registre el último acceso de cada página.
- Costo:** Implementaciones precisas pueden ser costosas en términos de tiempo y memoria.

OPT (Óptimo):

- **Funcionamiento:** Reemplaza la página que no será utilizada por más tiempo en el futuro.
- **Implementación:**
Este algoritmo requiere información previa sobre las futuras referencias, lo que lo hace no práctico en sistemas reales.
- **Simulación:** Se utiliza para análisis teórico y comparación con otros algoritmos.

c) Sabemos que la página a ser reemplazada puede estar modificada. ¿Qué acciones debe llevar el SO cuando se encuentra ante esta situación?

Cuando la página a ser reemplazada ha sido modificada (bit de "dirty" = 1), el sistema operativo debe realizar las siguientes acciones:

1. **Escribir la página en disco:**
 - Se guarda el contenido modificado en el almacenamiento secundario para preservar la coherencia de los datos.
 - Este proceso puede ralentizar el sistema debido a las operaciones de E/S.
2. **Actualizar estructuras de datos:**
 - Marcar el marco como disponible en la tabla de marcos.
 - Actualizar la tabla de páginas del proceso para invalidar la entrada correspondiente.
3. **Cargar la nueva página:**
 - Recuperar la página requerida desde el disco y cargarla en el marco libre.
 - Actualizar la tabla de páginas con la nueva entrada.

21.- Alcance del reemplazo

Al momento de tener que seleccionar una página víctima, el SO puede optar por 2 políticas a utilizar:

Reemplazo local

Reemplazo global

a) Describa cómo trabajan estas 2 políticas.

Reemplazo local: en esta política cada proceso tiene una cantidad fija asignada de marcos, cuando pasa un fallo de página se elige una página víctima solo de esa cantidad de marcos asignada al proceso. No afecta las páginas de otros procesos.

Reemplazo Global: en esta política se pueden elegir las páginas víctimas en cualquier proceso. Es más difícil de gestionar.

b) ¿Es posible utilizar la política de “Asignación Fija” de marcos junto con la política de “Reemplazo Global”? Justifique.

No es compatible porque la asignación fija contradice el reemplazo global que dice que se pueden tomar páginas de cualquier espacio de memoria. Si se quiere usar el reemplazo global se debería usar la asignación dinámica.

22.-

Considere la siguiente secuencia de referencias de páginas:

1, 2, 15, 4, 6, 2, 1, 5, 6, 10, 4, 6, 7, 9, 1, 6, 12, 11, 12, 2, 3, 1, 8, 1, 13, 14, 15, 3, 8

a) Si se dispone de 5 marcos. ¿Cuántos fallos de página se producirán si se utilizan las siguientes técnicas de selección de víctimas? (Considere una política de Asignación Dinámica y Reemplazo Global)

i) Segunda Chance

ii) FIFO

iii) LRU

iv) OPT

b) Suponiendo que cada atención de un fallo se pagina requiere de 0,1 seg. Calcular el tiempo consumido por atención a los fallos de páginas para los algoritmos de a).

23.-

Sean los procesos A, B y C tales que necesitan para su ejecución las siguientes páginas:

A: 1, 3, 1, 2, 4, 1, 5, 1, 4, 7, 9, 4

B: 2, 4, 6, 2, 4, 1, 8, 3, 1, 8

C: 1, 2, 4, 8, 6, 1, 4, 1

Si la secuencia de ejecución es tal que los procesos se ejecutan en la siguiente secuencia:

1. **B** demanda 2 páginas
2. **A** demanda 3 páginas
3. **C** demanda 2 páginas
4. **B** demanda 3 páginas
5. **A** demanda 3 páginas
6. **C** demanda 2 páginas
7. **B** demanda 2 páginas

8. **C** demanda 4 páginas
9. **A** demanda 3 páginas
10. **B** demanda 3 páginas
11. **C** termina
12. **A** demanda 3 páginas
13. **B** termina
14. **A** termina

a) Considerando una política de Asignación Dinámica y Reemplazo Global y disponiéndose de 7 marcos. ¿Cuántos fallos de página se producirán si se utiliza la técnica de selección de víctimas:

i) LRU ii) Segunda Chance

b) Considerando una política de Asignación Fija con reparto equitativo y Reemplazo Local y disponiéndose de 9 marcos. ¿Cuántos fallos de página se producirán si se utiliza la técnica de selección de víctimas:

i) LRU ii) Segunda Chance

24.-

Sean los procesos A, B y C tales que necesitan para su ejecución las siguientes páginas:

A: 1, 2, 1, 7, 2, 7, 3, 2

B: 1, 2, 5, 2, 1, 4, 5

C: 1, 3, 5, 1, 4, 2, 3

Si la secuencia de ejecución es tal que los procesos se ejecutan en la siguiente

manera:

1. **C** demanda 1 página
2. **A** demanda 2 páginas
3. **C** demanda 1 página
4. **B** demanda 1 página
5. **A** demanda 1 página
6. **C** modifica la página 1
7. **B** demanda 2 páginas

8. **A** demanda 1 página
9. **C** demanda 1 página
10. **B** modifica la página 2
11. **A** modifica la página 2
12. **B** demanda 2 páginas
13. **A** demanda 1 página
14. **B** demanda 2 páginas

15. **C** demanda 2 páginas
16. **C** demanda 1 página
17. **A** demanda 1 página
18. **B** termina

19. **A** demanda 2 páginas
20. **C** demanda 1 página
21. **A** termina
22. **C** termina

Considerando una política de Asignación Dinámica y Reemplazo Global y disponiéndose de 7 marcos, debiéndose guardar 1 marco para la gestión de descarga asincrónica de páginas modificadas ¿Cuántos fallos de página se producirán si se utiliza la técnica de selección de víctima:

- a) Segunda Chance
- b) FIFO
- c) LRU

25.- Hiperpaginación (Trashing)

- a) ¿Qué es?

Decimos que un sistema está en thrashing cuando pasa más tiempo paginando que ejecutando procesos .

- b) ¿Cuáles pueden ser los motivos que la causan?

Insuficiencia de memoria física: Hay más procesos activos de los que la RAM puede manejar, causando un uso excesivo de la memoria virtual.

Tamaño de conjunto de trabajo insuficiente: El proceso no tiene suficientes páginas en memoria para continuar ejecutándose de manera eficiente.

Demasiados procesos concurrentes: El sistema intenta mantener muchos procesos en memoria al mismo tiempo, lo que provoca una competencia excesiva por los recursos.

Configuración incorrecta del sistema: Parámetros como el tamaño del archivo de intercambio (swap) pueden ser inadecuados.

Carga de trabajo no optimizada: Aplicaciones mal diseñadas que hacen un acceso intensivo y aleatorio a la memoria.

- c) ¿Cómo la detecta el SO?

Alta actividad de E/S (entrada/salida): Incremento desproporcionado en la transferencia de páginas entre RAM y disco.

Bajo uso de CPU: La CPU está inactiva mientras espera que se completen las operaciones de E/S.

Latencia alta: Los **tiempos de respuesta** de los procesos se vuelven significativamente **lentos**.

Disminución del rendimiento global: Monitorización de métricas de rendimiento, como la frecuencia de fallos de página (*page faults*).

d) Una vez que lo detecta, ¿qué acciones puede tomar el SO para eliminar este problema?

Existen técnicas para evitarlo → estrategia de Working Set

26.-

Considere un sistema cuya memoria principal se administra mediante la técnica de paginación por demanda que utiliza un dispositivo de paginación, algoritmo de reemplazo global LRU y una política de asignación que reparte marcos equitativamente entre los procesos. El nivel de multiprogramación es actualmente, de 4.

Ante las siguientes mediciones:

a) Uso de CPU del 13%, uso del dispositivo de paginación del 97%.

b) Uso de CPU del 87%, uso del dispositivo de paginación del 3%.

c) Uso de CPU del 13%, uso del dispositivo de paginación del 3%.

Analizar:

¿Qué sucede en cada caso?

a) **Situación:** El sistema está en un estado de **hiperpaginación (trashing)**. Los **procesos pasan la mayor parte del tiempo esperando por páginas que no están en memoria**, lo que satura el dispositivo de paginación y deja la CPU subutilizada.

Conclusión: La **paginación está afectando negativamente al rendimiento**. Incrementar el nivel de multiprogramación **empeoraría la situación**, ya que aumentaría la competencia por los marcos de memoria.

b) **Situación:** El sistema está funcionando de manera **eficiente**. Los procesos tienen suficientes marcos de memoria para mantener su conjunto de trabajo, y la CPU está ocupada procesando tareas.

Conclusión: La **paginación es efectiva** y no está causando cuellos de botella. Incrementar el nivel de multiprogramación podría reducir el rendimiento si los procesos comienzan a competir por marcos

c) Situación: Ni la CPU ni el dispositivo de paginación están siendo utilizados significativamente. Esto sugiere que los procesos están bloqueados esperando eventos externos (como E/S o entrada de usuario), lo que provoca una baja utilización de recursos.

Conclusión: Incrementar el nivel de multiprogramación podría ser beneficioso en este caso, ya que introducir más procesos podría aumentar el uso de la CPU.

¿Puede incrementarse el nivel de multiprogramación para aumentar el uso de la CPU?

a) No. Incrementar el nivel de multiprogramación agravaría la hiperpaginación, reduciendo aún más el rendimiento.

b) No. El sistema ya está funcionando de manera eficiente. Incrementar la multiprogramación podría causar saturación de memoria y desencadenar hiperpaginación.

c) Sí. Dado que el sistema está infrautilizado, añadir más procesos podría aumentar el uso de la CPU sin un impacto significativo en la memoria.

¿La paginación está siendo útil para mejorar el rendimiento del sistema?

a) No. La paginación está generando una sobrecarga debido a la hiperpaginación, lo que degrada el rendimiento.

b) Sí. La paginación permite a los procesos ejecutar eficientemente, aprovechando la memoria disponible.

c) Parcialmente. Aunque la paginación no es problemática, su impacto es limitado porque los recursos no están siendo plenamente utilizados.

27.-

Considere un sistema cuya memoria principal se administra mediante la técnica de paginación por demanda. Considere las siguientes medidas de utilización:

- Utilización del procesador: 20%
- Utilización del dispositivo de paginación: 97,7%
- Utilización de otros dispositivos de E/S: 5%

Cuales de las siguientes acciones pueden mejorar la utilización del procesador:

a) Instalar un procesador más rápido

b) Instalar un dispositivo de paginación mayor. **Esto ya que reduciría el tiempo necesario para manejar fallos de página.**

c) Incrementar el grado de multiprogramación

d) Instalar más memoria principal. Esto ya que al aumentar la memoria principal, **los procesos tendrán un mayor número de marcos disponibles**, reduciendo la frecuencia de fallos de página y, por ende, la dependencia del dispositivo de paginación.

e) Decrementar el quantum para cada proceso

28.-

La siguiente fórmula describe el tiempo de acceso efectivo a la memoria al utilizar paginación para la implementación de la memoria virtual:

$$TAE = At + (1 - p) * Am + p * (Tf + Am)$$

Donde:

TAE = tiempo de acceso efectivo

p = tasa de fallo de página ($0 \leq p \leq 1$)

Am = tiempo de acceso a la memoria real

Tf = tiempo de atención de un fallo de página

At = tiempo de acceso a la tabla de páginas. Es igual al tiempo de acceso a la memoria (Am) si la entrada de la tabla de páginas no se encuentra en la TLB.

Suponga que tenemos una memoria virtual paginada, con tabla de páginas de 1 nivel, y donde la tabla de páginas se encuentra completamente en la memoria.

Servir una falla de página tarda 300 nanosegundos si hay disponible un marco vacío o si la página reemplazada no se ha modificado, y 500 nanosegundos si se ha modificado. El tiempo de acceso a memoria es de 20 nanosegundos y el de acceso a la TLB es de 1 nanosegundo

a) Si suponemos una tasa de fallos de página de 0,3 y que siempre contamos con un marco libre para atender el fallo ¿Cuál será el TAE si el 50% de las veces la entrada de la tabla de páginas se encuentra en la TLB (hit)?

$$At = 0.5 * 1ns + 0.5 * 20ns = 0.5 + 10 = 10.5ns$$

$$10.5ns + (1 - 0.3) * 20ns + 0.3 * (300ns + 20ns) = 120.5$$

b) Si suponemos una tasa de fallos de página de 0,3; que el 70% de las ocasiones la página a reemplazar se encuentra modificada. ¿Cuál será el TAE si el 60% de las veces la entrada de la tabla de páginas se encuentra en la TLB (hit)?

$$Tf = 0.3 \cdot 300 + 0.7 \cdot 500 = 90 + 350 = 440$$

$$At = 0.6 \cdot 1 + 0.4 \cdot 20 = 0.6 + 8 = 8.6$$

$$8.6 + (1 - 0.3) \cdot 20 + 0.3 \cdot (440 + 20) = 160.6$$

c) Si suponemos que el 60% de las veces la página a reemplazar está modificada, el 100% de las veces la entrada de la tabla de páginas requerida se encuentra en la TLB (hit) y se espera un TAE menor a 200 nanosegundos. ¿Cuál es la máxima tasa aceptable de fallas de página?

$$at = 1$$

$$Tf = 300 \cdot 0.4 + 500 \cdot 0.6 = 120 + 300 = 420$$

$$200ns > 1ns + (1-p) \cdot 20 + p \cdot (420 + 20)$$

$$200ns > 1ns + 20 - 20p + p440$$

$$200ns > 21ns + 420p$$

$$179ns > 420p$$

$$179ns / 420 > p$$

$$0.426 > p$$

$$p < 0.426$$

La máxima tasa aceptables de fallo de página es aproximadamente 0.426

29.- Anomalía de Belady

a) ¿Qué es?

La **anomalía de Belady** es un fenómeno en el que **incrementar el número de marcos de página asignados a un proceso en un sistema de memoria virtual puede provocar un aumento en la cantidad de fallos de página**, en lugar de reducirlos como se esperaría.

Este comportamiento ocurre con ciertos algoritmos de reemplazo de páginas, como **FIFO (First In, First Out)**, y no con algoritmos óptimos o basados en el pasado reciente, como LRU.

b) Dada la siguiente secuencia de referencias a páginas:

3, 2, 1, 0, 3, 2, 4, 3, 2, 1, 0, 4

I. Calcule la cantidad de fallos de páginas si se cuentan con 3 marcos y se utiliza el algoritmo de reemplazo FIFO

II. Calcule la cantidad de fallos de páginas si se cuentan con 4 marcos y se utiliza el algoritmo de reemplazo FIFO

Analice la situación

Ocorre la anomalía de Belady.

30.-

Considere el siguiente programa:

```
#define Size 64
int A[Size; Size], B[Size; Size], C[Size; Size];
int register i, j;
for (j = 0; j < Size; j++)
    for (i = 0; i < Size; i++)
        C[i; j] = A[i; j] + B[i; j];
```

Si asumimos que el programa se ejecuta en un sistema que utiliza paginación por demanda para administrar la memoria, donde cada página es de 1Kb. Cada número entero (int) ocupa 4 bytes. Es claro que cada matriz requiere de 16 páginas para almacenarse. Por ejemplo: $A[0,0]..A[0,63]$, $A[1,0]..A[1,63]$, $A[2,0]..A[2,63]$ y $A[3,0]..A[3,63]$ se almacenará en la primera página.

Asumamos que el sistema utiliza un working set de 4 marcos para este proceso. Uno de los 4 marcos es utilizado por el programa y los otros 3 se utilizan para datos (las matrices). También asumimos que para los índices "i" y "j" se utilizan 2 registros, por lo que no es necesario el acceso a la memoria para estas 2 variables.

a) Analizar cuántos fallos de páginas ocurren al ejecutar el programa (considere las veces que se ejecuta $C[i,j] = A[i,j] + B[i,j]$)

b) Puede ser modificado el programa para minimizar el número de fallos de páginas. En caso de ser posible indicar la cantidad de fallos de fallos de páginas que ocurren.

31.-

Considere las siguientes secuencias de referencias a páginas de los procesos A y B, donde se muestra en instante de tiempo en el que ocurrió cada una (1 a 78):

Proceso A																																							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	
1	2	1	3	4	3	3	2	1	1	3	4	5	6	1	2	1	3	3	4	5	6	6	6	5	4	3	1	1	2	3	4	5	4	3	2	1	1	1	
40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	
2	2	1	2	1	1	2	3	4	5	6	7	8	6	5	3	1	2	3	4	5	6	5	1	1	1	2	3	4	5	4	2	1	1	2	3	4	5	1	1

Proceso B																																						
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
1	2	3	3	3	4	4	5	1	1	1	2	3	4	4	4	4	1	1	2	3	6	5	6	5	4	6	1	1	1	1	4	5	1	3	2	1	1	2
40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78
2	2	3	3	4	4	4	3	4	5	4	6	1	1	1	2	3	3	3	4	5	5	5	1	6	2	3	1	2	3	1	1	1	2	2	2	2	3	3

a) Considerando una ventana $\Delta=5$, indique cuál sería el conjunto de trabajo de los procesos A y B en el instante 24 (WSA(24) y WSB(24))

WSA(24)={ 6,5,4}

WSB(24)={ 6,5,3,2}

b) Considerando una ventana $\Delta=5$, indique cuál sería el conjunto de trabajo de los procesos A y B en el instante 60 (WSA(60) y WSB(60))

WSA(60)={5,4,3,2,1}

WSB(60)={5,4,3}

c) Para el los WS obtenidos en el inciso a), si contamos con 8 frames en el sistema ¿Se puede indicar que estamos ante una situación de trashing? ¿Y si contáramos con 6 frames?

WSA(24)+WSB(24)= 3+4=7 → no porque tenemos mas frames que los que necesitamos

Con 6 sí porque nos faltan marcos

d) Considerando únicamente el proceso A, y suponiendo que al mismo se le asignaron inicialmente 4 marcos, donde el de reemplazo de páginas es realizado considerando el algoritmo FIFO. ¿Cuál será la tasa de fallos en el instante 38 de páginas suponiendo que la misma se calcula contando los fallos de páginas que ocurrieron en las últimas 10 unidades de tiempo?

Pf =6

e) Para el valor obtenido en el inciso d), si suponemos que el S.O. utiliza como límites superior e inferior de tasa de fallos de páginas los valores 2 y 5 respectivamente ¿Qué acción podría tomar el S.O. respecto a la cantidad de marcos asignados al proceso?

Como la tasa de fallos **igualó el límite superior**, el sistema operativo podría **asignar más marcos** al proceso para reducir la tasa de fallos.