

Práctica 5 – ISO

1.- Explique a que hacen referencia los siguientes términos:

- Dirección Lógica o Virtual: referencia a una localidad de memoria independiente de la asignación actual de los datos en la memoria. Si la CPU trabaja con direcciones lógicas, para acceder a memoria principal, se deben transformar en direcciones físicas.
- Dirección Física: Referencia una localidad en la Memoria Física (RAM). Es la dirección real o absoluta.

2.- 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

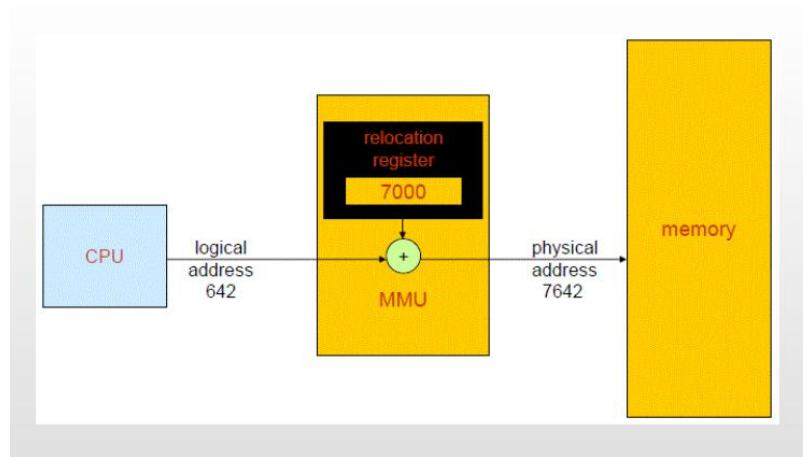
a) Explique cómo trabajan estos 2 métodos. Cite diferencias, ventajas y desventajas.

| | Particiones Fijas | Particiones Dinámicas |
|---------------------------|--|--|
| Diferencias y similitudes | <ul style="list-style-type: none"> - En estas la memoria se divide en particiones o regiones que pueden ser del mismo tamaño o no. - Alojan un proceso cada una. - Cada proceso se coloca en una de las particiones por algún criterio (First Fit, Best Fit, Worst Fit, Next Fit) - El mejor criterio para estas particiones es el Best Fit. | <ul style="list-style-type: none"> - En estas el tamaño y el número de particiones varía. - Alojan un proceso cada una. - Cada una de estas particiones se genera de forma dinámica y con el tamaño justo que necesita el proceso. - Cada proceso se coloca en una de las particiones por algún criterio (First Fit, Best Fit, Worst Fit, Next Fit) - El mejor criterio para estas particiones es el Worst Fit. |
| Ventajas | <ul style="list-style-type: none"> - Una de sus ventajas es el aislamiento de datos ya que estos están completamente separados de los datos en otras particiones. - Tienen fácil implementación. - Generan baja sobrecarga del CPU en el manejo de memoria y resolución de direcciones. | <ul style="list-style-type: none"> - Una de sus ventajas es que la asignación de memoria es más eficiente por proceso. - El grado de multiprogramación puede crecer o decrecer de forma dinámica. |
| Desventajas | - Generan Fragmentación interna. | - Generan Fragmentación externa. |

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

Para poder administrar la memoria con estos métodos el SO requiere del Registro Base (Dirección de comienzo del Espacio de Direcciones del proceso en la RAM) y del Registro Límite (Dirección final del proceso o medida del proceso. Es el tamaño de su espacio de direcciones).

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



3.- 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 | Particiones de diferente tamaño |
|-------------|--|--|
| Ventajas | <ul style="list-style-type: none"> - Son fáciles de implementar y administrar. - Permiten una distribución equitativa del espacio de memoria. | <ul style="list-style-type: none"> - Permiten un uso eficiente del espacio. - Son flexibles ya que se pueden ajustar a los tamaños de los procesos. |
| Desventajas | <ul style="list-style-type: none"> - Son pocos flexibles ya que no se puede ajustar el tamaño de las mismas si se requiere. - Se genera una ineficiencia si un proceso no utiliza completamente el tamaño asignado para las particiones. | <ul style="list-style-type: none"> - Requieren una mayor administración y planificación para el uso de las mismas. - Generan una mayor posibilidad de fragmentación. |

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.

Fragmentación Interna: hace referencia a la porción de la partición que queda sin utilizar.

Fragmentación Externa: Hace referencia a los huecos que van quedando en la memoria a medida que los procesos finalizan. Al no encontrarse en forma contigua puede darse el caso de que tengamos memoria libre para alojar un proceso, pero que no la podamos utilizar.

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

Para solucionar el problema se puede acudir a la compactación, pero resulta muy costosa en términos de eficiencia (memoria y ejecución).

5.- Paginación

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

- La Memoria Física es dividida lógicamente en pequeños trozos de igual tamaño llamados Marcos.

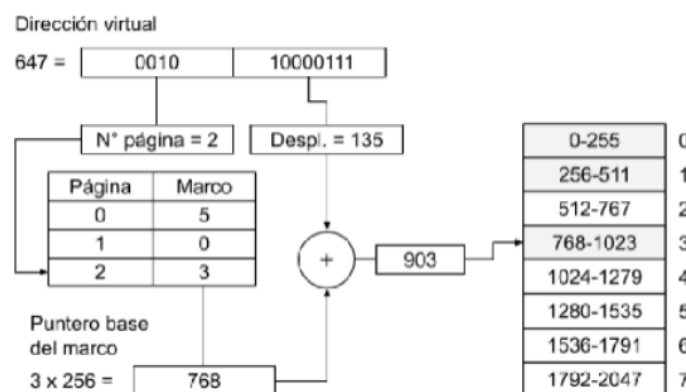
- La Memoria Lógica (espacio de direcciones) es dividida en trozos de igual tamaño que los Marcos llamados Páginas.

- La dirección lógica se interpreta como: Un número de página y un desplazamiento dentro de la misma.

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

El SO debe mantener una tabla de páginas por cada proceso, donde cada entrada contiene (entre otras) el Marco en la que se coloca cada página. Se guarda la base del marco que fue cargado en la PCB del proceso.

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



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

En este esquema, se puede producir fragmentación, pero es despreciable ya que únicamente quedarían los últimos espacios de memoria de los marcos sin ocupar.

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

Similitudes:

- Pueden generar fragmentación interna.
- Tanto la paginación como las particiones fijas dividen a la memoria en porciones con un tamaño fijo.
- Ambas técnicas permiten la ejecución simultánea de múltiples procesos.

Diferencias:

- Las Particiones Fijas dividen la memoria en bloques fijos y cada uno de estos es asociado a un proceso mientras que la Paginación divide la memoria en páginas de tamaño fijo, pero asigna memoria basada en la necesidad del proceso, por lo tanto, en una página puede haber más de un proceso.
- Las Particiones Fijas tienen un costo de implementación mucho más bajo que la Paginación.
- La Paginación permite la reubicación de un proceso durante la ejecución en pos de acomodar las demandas cambiantes mientras que en las Particiones Fijas esto no es posible.

7.- 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 páginas) en el que reflejen el estado descrito.

| Página | Marco | Marco | Inicio-Fin |
|--------|-------|-------|------------|
| 0 | 3 | 0 | 0-511 |
| 1 | 5 | 1 | 512-1023 |
| 2 | 2 | 2 | 1024-1535 |
| 3 | 6 | 3 | 1536-2047 |
| | | 4 | 2048-2559 |
| | | 5 | 2560-3071 |
| | | 6 | 3072-3583 |

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

- Número de Página = Dirección Lógica DIV Tamaño de Página.
- Desplazamiento = Dirección Lógica MOD Tamaño de Página.
- Dirección Física = Base del Frame + Desplazamiento.

i) 35: Es válida.

- $35 \text{ DIV } 512 = \text{Página } 0.$
- $35 \text{ MOD } 512 = \text{Desplazamiento } 35.$
- $1536 \text{ (Base del marco 3)} + 35 = \text{Dirección física } 1571.$

ii) 512: Es válida.

- $512 \text{ DIV } 512 = \text{Página } 1.$
- $512 \text{ MOD } 512 = \text{Desplazamiento } 0.$
- $2560 \text{ (Base del marco 5)} + 0 = \text{Dirección física } 2560.$

iii) 2051: No es válida.

- $2051 \text{ DIV } 512 = \text{Página } 4.$

iv) 0: Es válida.

- $0 \text{ DIV } 512 = \text{Página } 0.$
- $0 \text{ MOD } 512 = \text{Desplazamiento } 0.$
- $1536 \text{ (Base del marco 3)} + 0 = \text{Dirección física } 1536.$

v) 1325: Es válida.

- $1325 \text{ DIV } 512 = \text{Página } 2.$
- $1325 \text{ MOD } 512 = \text{Desplazamiento } 301.$

- $1024 \text{ (Base del marco 2)} + 301 = \text{Dirección física } 1325.$

vi) 602:

- $602 \text{ DIV } 512 = \text{Página } 1.$
- $602 \text{ DIV } 512 = \text{Desplazamiento } 90.$
- $2560 \text{ (Base del marco 5)} + 90 = \text{Dirección física } 2650.$

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

- $\text{Número de Marco} = \text{Dirección Física DIV Tamaño de Página}.$
- $\text{Desplazamiento} = \text{Dirección Física MOD Tamaño de Página}.$
- $\text{Dirección Virtual} = \text{Tamaño de Página} + \text{Desplazamiento}.$

i) 509: No es válida.

- $509 \text{ DIV } 512 = \text{Marco } 0.$
- $509 \text{ MOD } 512 = \text{Desplazamiento } 509.$

ii) 1500: Es válida.

- $1500 \text{ DIV } 512 = \text{Marco } 2.$
- $1500 \text{ MOD } 512 = \text{Desplazamiento } 476.$
- $512 + 476 = \text{Dirección virtual } 988.$

iii) 0: No es válida.

- $0 \text{ DIV } 512 = \text{Marco } 0.$
- $0 \text{ MOD } 512 = \text{Desplazamiento } 0.$

iv) 3215: Es válida

- $3215 \text{ DIV } 512 = \text{Marco } 6.$
- $3215 \text{ MOD } 512 = \text{Desplazamiento } 143.$
- $512 + 143 = \text{Dirección virtual } 655.$

v) 1024: Es válida.

- $1024 \text{ DIV } 512 = \text{Marco } 2.$
- $1024 \text{ MOD } 512 = \text{Desplazamiento } 0.$
- $512 + 0 = \text{Dirección virtual } 512.$

vi) 2000: Es válida.

- $2000 \text{ DIV } 512 = \text{Marco } 3.$
- $2000 \text{ MOD } 512 = \text{Desplazamiento } 464.$
- $512 + 464 = \text{Dirección virtual } 976.$

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

Se produce fragmentación interna.

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?

3 bits para el número de página ($0 \dots 7$) y 10 bits para el desplazamiento ($\log_2(1024) = 10 \Rightarrow 2^{10} = 1024$), es decir, 13 bits en total.

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

5 bits para la cantidad de marcos ($\log_2(32) = 5 \Rightarrow 2^5 = 32$) y 10 para el desplazamiento ($\log_2(1024) = 10 \Rightarrow 2^{10} = 1024$), es decir, 15 bits en total.

9.- Segmentación

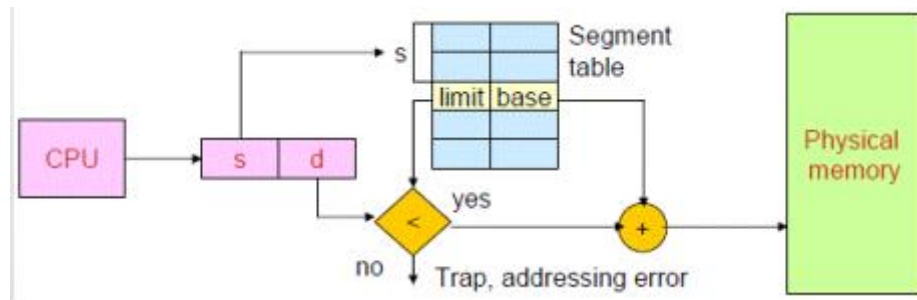
a) Explique cómo trabaja este método de asignación de memoria.

- Esquema que se asemeja a la “visión del usuario”. El programa se divide en partes/secciones.
- Un programa es una colección de segmentos. Un segmento es una unidad lógica.
- Todos los segmentos de un programa pueden no tener el mismo tamaño (código, datos, rutinas).
- Las direcciones Lógicas consisten en 2 partes: Selector de Segmento y Desplazamiento dentro del segmento ($0 < \text{desplazamiento} < \text{límite del segmento}$).
- Más de un proceso puede utilizar un mismo segmento, por lo tanto, comparten direcciones físicas. Con esta misma lógica se genera protección en la memoria.

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

El SO debe poseer una Tabla de Segmentos donde cada entrada contenga: Base (Dirección física del comienzo del segmento) y Limit (Longitud del segmento).

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.

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

| | Segmentación y Particiones dinámicas |
|-------------|---|
| Similitudes | <ul style="list-style-type: none"> - Ambas técnicas pueden causar fragmentación externa. - Las 2 técnicas se basan en una distribución de espacios variables en tamaño de forma dinámica, es decir, según se requiera. |
| Diferencias | <ul style="list-style-type: none"> - En Segmentación más de un proceso podría estar utilizando el mismo segmento mientras que en Particiones Dinámicas solo un proceso puede ocupar una partición. - Las divisiones de segmentos y distribución de los mismos se realizan de forma indiferente a su orden real mientras que en las Particiones esto no es así ya que todo el proceso se carga de manera contigua. |

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

| Paginación y Segmentación | |
|---------------------------|--|
| Similitudes | <ul style="list-style-type: none"> - Las dos estrategias hacen uso de estructuras adicionales. - Ambas técnicas pueden causar fragmentación. - Ambas permiten la distribución de memoria en partes no contiguas. - No requieren que las partes se asignen en el mismo orden que en el del programa principal. |
| Diferencias | <ul style="list-style-type: none"> - La Paginación elimina la fragmentación externa. - La Paginación es transparente al usuario mientras que la Segmentación si lo es. - La Segmentación da un mejor soporte a la compartición y protección de memoria. - La Segmentación facilita modularidad y estructuras de datos grandes. |

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

Tabla de Segmentos

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

Tabla de Paginas

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

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

- Dirección Física = Base de Segmento + Base de Página + Desplazamiento

i) (2,1,1)

ii) (1,3,15)

iii) (3,1,10)

iv) (2,3,5)

| Dirección Lógica | Núm. Segmento y Base | Núm. Página y Base | Desplazamiento | Dirección Física |
|------------------|----------------------|--------------------|----------------|------------------|
| (2, 1, 1) | 2 (1500) | 1 (20) | 1 | 1521 |
| (1, 3, 15) | 1 (500) | 3 (60) | 15 | 575 |
| (3, 1, 10) | 3 (5000) | 1 (120) | 10 | 5130 |
| (2, 3, 5) | 2 (1500) | 3 (0) | 5 | 1505 |

13.- Memoria Virtual

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

- Capacidad de ejecutar programas más grandes: esto lo logra a partir de la utilización de un espacio de direcciones virtuales que pueden ser mucho más grandes que la memoria RAM física, manteniendo en ella solo lo que el proceso esté utilizando y dejando lo demás en disco.

- Proporciona una abstracción entre la memoria física y la memoria virtual, lo que permite que los sistemas operativos gestionen eficientemente la asignación de memoria a los procesos para aprovechar al máximo los recursos disponibles.

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

Se debe apoyar en el Hardware, este debe de ser capaz de detectar si una instrucción está queriendo acceder a una dirección que no está cargada en memoria, y a partir de esa situación el SO resuelve el fallo. Es importante que el SO le brinde al Hardware la información necesaria para que pueda llevar a cabo esa tarea de detección.

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?

Al implementar esta técnica utilizando paginación por demanda se debe contar por lo menos con un “bit de presencia” / “bit de validez” que indique si la página está cargada en memoria o no. A partir de este bit es que el hardware puede generar la interrupción necesaria para resolver el fallo de página.

Otro bit necesario sería el “bit de modificación” / “bit de dirty” ya que le permite al sistema operativo saber si una página ha sido modificada desde que fue cargada en memoria, lo que puede ser crucial para decidir si la página necesita ser escrita nuevamente en el almacenamiento secundario antes de ser reemplazada.

14.- Fallos de Página (Page Faults):

a) ¿Cuándo se producen?

Los Fallos de Página se generan cuando una instrucción ejecutada hace referencia a una dirección lógica cuya página no está cargada actualmente en memoria.

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

El hardware es el responsable de detectar estos fallos, generando una interrupción para que el SO resuelva estos fallos de direcciones.

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

1. Se genera el trap.
2. El SO bloquea al proceso, es decir, la CPU toma otro proceso.
3. El SO busca un marco que esté libre en la memoria y genera una operación de E/S que le pide al disco, para copiar en dicho marco la página deseada.
4. La E/S avisa por interrupción cuando finaliza.
5. El SO actualiza la tabla de páginas del proceso (coloca el bit V en 1 y coloca la dirección base del marco donde se colocó la página)
6. El proceso pasa del estado bloqueado a ready.
7. Cuando vuelve a ser asignado al CPU, el proceso comienza desde la interrupción que generó el fallo en primera instancia.

15.- Direcciones:

a) Si se dispone de un 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”)?

Direcciones máximas = 2^{32}

Tamaño máximo => Direcciones máximas * 1 byte = 4.294.967.296 bytes = 4GB.

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

Tamaño máximo en Kb => Tamaño máximo (4.294.967.296 bytes) / 1024 = 4.194.304Kb

Número máximo de Páginas => Tamaño máximo en Kb / 512 = 8192 páginas por 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?

Tamaño de la memoria real en Kb => 256 Mb * 1024 = 262.144 Kb

Cantidad de Marcos => Tamaño de la memoria real en Kb / Tamaño de página (512 Kb) = 512 marcos.

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?

Tamaño máximo de la tabla de páginas => Número máximo de Páginas (8192) * Tamaño de cada entrada (2 Kb) = 16.384 Kb.

16.- 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 como trabajan estos enfoques e indique como se realiza la transformación de la dirección virtual en dirección física.

Tablas de Páginas de 1 nivel: en este enfoque solo hay una tabla de páginas que mapea las direcciones lógicas a direcciones físicas. Esta tabla contiene entradas que indican la correspondencia entre páginas virtuales y páginas físicas.

En este enfoque la transformación de direcciones se realiza mediante la tabla, descomponiendo a la dirección lógica en un índice que se utiliza para acceder a la entrada correspondiente en la tabla y un desplazamiento.

Tablas de páginas de 2 niveles: este enfoque tiene el propósito de dividir la tabla de páginas lineal en múltiples tablas de páginas. Cada tabla de páginas suele tener el mismo tamaño, pero se busca que tengan un menor número de páginas por tabla ya que la idea general es que cada tabla sea más pequeña para que no ocupe demasiada RAM.

Otro aspecto importante es que solo se carga una parcialidad de la tabla de páginas (solo lo que se necesite resolver) y presenta un esquema de direccionamientos indirectos, es decir, las tablas de segundo nivel se

pueden llevar a memoria secundaria liberando RAM pero hay más de un acceso a disco para obtener un dato.

Tablas Invertidas: hay una única tabla en todo el sistema que contiene todas las entradas de páginas para todos los procesos, el espacio de direcciones de esta tabla se refiere al espacio de la RAM y no al espacio de direcciones lógicas de un proceso. El número de página es transformado en un valor de HASH y este se usa como índice de la tabla para encontrar el marco asociado, además es necesario definir un mecanismo de encadenamiento para solucionar colisiones.

Para este mecanismo solo se mantienen las entradas de tablas de página (PTE) de las páginas presentes en RAM, organizando la tabla invertida como una tabla de hash en RAM, generando búsquedas indexadas por número de página virtual.

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

- Número de Página = Dirección Lógica DIV Tamaño de Página.
- Desplazamiento = Dirección Lógica MOD Tamaño de Página.
- Dirección Física = Base del Frame + Desplazamiento.
- Páginas 0, 1, 3 y 5 en memoria.

| Dirección Lógica | Número de Página | Desplazamiento | Marco y Base | Dirección Física | Fallo de Página |
|------------------|------------------|----------------|--------------|------------------|-----------------|
| 1052 | 2 | 28 | - | - | SI |
| 2221 | 4 | 173 | - | - | SI |
| 5499 | 10 | 379 | - | - | ERROR |
| 3101 | 6 | 29 | - | - | ERROR |

18.- 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 | Tamaño de página grande |
|-------------|---|---|
| Ventajas | <ul style="list-style-type: none"> - Con páginas pequeñas hay menos espacio no utilizado dentro de una página, reduciendo la fragmentación interna. - Más páginas pueden alocarse en memoria. - Si una página pequeña tiene errores, se pierde menos memoria en comparación con una página grande. | <ul style="list-style-type: none"> - Con páginas grandes se necesitan menos entradas en la tabla de páginas para mapear el mismo espacio de direcciones lógicas. - Con páginas más grandes hay menor fragmentación interna. - Se genera una menor sobrecarga del SO al gestionar con un número menor de páginas. - Es más rápido mover páginas grandes hacia la RAM ya que la memoria secundaria está diseñada para transferir bloques de datos más eficientemente. |
| Desventajas | <ul style="list-style-type: none"> - Con páginas pequeñas se necesitan más entradas en la tabla de páginas, por lo tanto esta se vuelve más grande, aumentando la sobrecarga y tiempo de búsqueda. - Genera mayor fragmentación externa. | <ul style="list-style-type: none"> - Si una página grande tiene errores, se pierde más memoria en comparación con una página pequeña. - Genera mayor fragmentación interna. |

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 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 como trabajan estas 2 políticas.

Asignación Fija: Con esta política a cada proceso se le asigna una cantidad arbitraria de marco. A su vez para el reparto de estas cantidades se puede usar:

- Reparto equitativo: Se asigna la misma cantidad de marcos a cada proceso -> marcos DIV procesos.
- Reparto proporcional: Se asignan marco en base a la necesidad de cada proceso -> $V_p \cdot m / V_t$. (Siendo V_p la cantidad de páginas que usa el proceso y V_t la sumatoria de las V_p de todos los procesos)

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

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

| Proceso | Total de Páginas Usadas |
|---------|-------------------------|
| 1 | 15 |
| 2 | 20 |
| 3 | 20 |
| 4 | 8 |

i) Reparto Equitativo

Con m (cantidad de marcos) = 40 y p (cantidad de procesos) = 4. Si usamos Reparto Equitativo $\rightarrow m \text{ DIV } p = 40 \text{ DIV } 4 = 10$. Total, de 10 marcos por procesos.

ii) Reparto Proporcional

Usando $V_p \cdot m / V_t$, ($m = 40$ y $V_t = 63$)

- Proceso 1: $\frac{15}{63} \times 40 = 9.52 \approx 10$
- Proceso 2: $\frac{20}{63} \times 40 = 12.7 \approx 13$
- Proceso 3: $\frac{20}{63} \times 40 = 12.7 \approx 13$
- Proceso 4: $\frac{8}{63} \times 40 = 5.08 \approx 5$

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

El Reparto Proporcional termina siendo más eficiente ya que si vemos por ejemplo al Proceso 4, podemos ver que con Reparto Equitativo se le asignó 2 marcos de más, algo que no queremos ya que esos marcos podrían estar siendo utilizados por procesos que los necesiten.

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.

De malo a bueno: FIFO, FIFO Segunda chance, LRU y Optimo.

b) Analice su funcionamiento. ¿Cómo los implementaría?

FIFO: elimina la página que ha estado en memoria por más tiempo. Cuando llega una nueva página, reemplaza a la más antigua en memoria. Se puede implementar el FIFO mediante una cola circular donde siempre que llegue una página se encola al final y cuando se necesita reemplazar se saca del inicio de la cola.

Segunda Chance: similar al FIFO, pero con una variante. La página más antigua se elimina a menos que haya sido referenciada recientemente, en cuyo caso se le da una “segunda oportunidad” y se coloca al final de la cola. Se puede implementar con una cola circular y un bit de referencia para la segunda oportunidad. Si es 1 hay segunda oportunidad, si es 0, no la hay.

LRU: reemplaza la página que no ha sido utilizada durante más tiempo. Mantiene un registro del tiempo de la última referencia a cada página. El registro se implementa con la ayuda del hardware generando timestamps.

Óptimo: selecciona la página que no será utilizada durante el período más largo. Es un algoritmo teórico ya que implica el conocimiento del futuro.

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

Cuando el SO se encuentra frente a esa situación reserva uno o varios marcos para realizar una descarga asincrónica de páginas, llevando a cabo las siguientes acciones:

1. La página que causó el fallo se coloca en un frame designado a la descarga asincrónica.
2. El SO envía la orden de descarga asincrónica mientras continúa ejecutando otro proceso.
3. El frame de descarga asincrónica pasa a ser el contenedor de la página víctima que fue descargada correctamente.

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 como trabajan estas 2 políticas.

Reemplazo local: El fallo de página de un proceso solo puede reemplazar sus propias páginas. No cambia la cantidad de frames asignado. El SO es capaz de determinar la tasa de fallo de página de cada proceso.

Reemplazo global: el fallo de página de un proceso puede reemplazar la página de cualquier proceso. El SO no controla la tasa de fallo de página de cada proceso. Un proceso se puede apropiar de frames de otro proceso.

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

A mi punto de vista, no. Con la Asignación Fija de marcos hacemos que cada proceso pueda usar un número fijo de marcos, si esto lo combinamos con Reemplazo Global puede ser que un proceso reemplace un marco de otro proceso y no uno de su propio conjunto, logrando que el número fijo de marcos que se asignó fijamente varíe, algo que no queremos si queremos llevar esa política adelante.

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 disponen de 5 marcos. ¿Cuántos fallos de página se producirán si se utilizan las siguientes técnicas de selección de víctima? (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 página requiere de 0,1 seg. Calcular el tiempo consumido por atención a los fallos de páginas para los algoritmos de a).

Solución en la carpeta de ejercicios resueltos Práctica-5.

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 | 8. C demanda 4 páginas |
| 2. A demanda 3 páginas | 9. A demanda 3 páginas |
| 3. C demanda 2 páginas | 10. B demanda 3 páginas |
| 4. B demanda 3 páginas | 11. C termina |
| 5. A demanda 3 páginas | 12. A demanda 3 páginas |
| 6. C demanda 2 páginas | 13. B termina |
| 7. B demanda 2 páginas | 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

Solución en la carpeta de ejercicios resueltos Práctica-5.

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 | 6. C modifica la página 1 |
| 2. A demanda 2 páginas | 7. B demanda 2 páginas |
| 3. C demanda 1 página | 8. A demanda 1 página |
| 4. B demanda 1 página | 9. C demanda 1 página |
| 5. A 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

Solución en la carpeta de ejercicios resueltos Práctica-5.

25.- *Hiperpaginación (Trashing)*

a) ¿Qué es?

Es la situación que se da cuando un sistema pasa más tiempo paginando que ejecutando procesos.

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

La hiperpaginación ocurre cuando los procesos en ejecución no tienen suficientes marcos asignados para mantener su conjunto de trabajo (working set). Esto provoca una serie de fallos de página consecutivos, ya que las páginas reemplazadas pertenecen al conjunto de trabajo de algún proceso, generando un ciclo continuo de fallos. Esta situación surge por la falta de memoria RAM suficiente para satisfacer la demanda de páginas de los procesos.

c) ¿Cómo la detecta el SO?

La hiperpaginación se detecta mediante el control de la frecuencia de fallos de página, definido por una tasa p que mide la proporción de fallos respecto a los accesos a memoria de un proceso ($0 < p \leq 1$). Si p se acerca a 1, el proceso no tiene suficientes marcos para mantener su localidad, indicando riesgo de hiperpaginación. Por otro lado, si p se acerca a 0, el proceso mantiene mejor su localidad. El sistema utiliza cotas mínima y máxima de p para evaluar el estado de los procesos y detectar posibles problemas de hiperpaginación.

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

Dependiendo del algoritmo de asignación y reemplazo, se puede reducir la cantidad de marcos asignados a un proceso con $p \rightarrow 0$, para reasignarlos a los procesos con $p \rightarrow 1$. En caso de tratarse de un reemplazo local, o de no haber procesos con p debajo de la cota mínima, se pueden seleccionar uno o más procesos para suspender, a fin de redistribuir sus marcos a los procesos que lo necesiten (es decir, reducir el grado de multiprogramación).

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?
- ¿Puede incrementarse el nivel de multiprogramación para aumentar el uso de la CPU?
- ¿La paginación está siendo útil para mejorar el rendimiento del sistema?

a) En este caso hay Hiperpaginación ya que el sistema se pasa más tiempo paginando que dándole uso a la CPU, por lo tanto, los procesos se quedan esperando las resoluciones de los fallos de página sin poder aprovechar al máximo el uso de la CPU.

No es recomendable aumentar el grado de multiprogramación ya que podría empeorar la situación. La paginación no está siendo útil para la mejoría del rendimiento del sistema.

b) En este caso no hay Hiperpaginación, los procesos están aprovechando la CPU sin tener que esperar a la resolución de fallos de página.

Se puede aumentar el grado de multiprogramación en este caso, pero hay que tener cuidado ya que la CPU se está utilizando casi en su totalidad por lo tanto un gran aumento del grado de multiprogramación podría generarnos una situación que empeore el rendimiento y aumente la tasa de fallos de página. La paginación está siendo útil para la mejoría del sistema.

c) En este caso no hay Hiperpaginación pero la CPU está bastante tiempo ociosa, quizás esto ocurre porque hay un lote de pequeños que se están ejecutando o quieren ejecutarse.

Es recomendable aumentar el grado de multiprogramación para hacer que la CPU no pase tanto tiempo ociosa. La paginación no está siendo muy útil para el sistema ya que no se está aprovechando del todo el uso de la CPU.

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%*

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

a) Instalar un procesador más rápido

Mejorar el procesador no resolvería el problema, ya que no se trata de velocidad, sino de hiperpaginación. Esto indica un problema relacionado con la cantidad de memoria RAM disponible y el rendimiento del dispositivo de paginación.

b) Instalar un dispositivo de paginación mayor

Esto si mejoraría la utilización del procesador ya que reduciría la Hiperpaginación.

c) Incrementar el grado de multiprogramación

No, lo único que se lograría es hacer que haya más procesos causando fallos de páginas.

d) Instalar más memoria principal

Esto si mejoraría la velocidad del procesador ya que se estaría dando más espacio para almacenar más páginas para los procesos logrando que estos disminuyan su tiempo de espera en la resolución de fallos de los mismos.

e) Decrementar el quantum para cada proceso

Esto no mejoraría la velocidad del procesador.

28.- La siguiente formula 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 pagina

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)?

$$TAE = 120,5 \text{ ns}$$

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)?

$$TAE = 160,6 \text{ ns}$$

c) Si suponemos que el 60% de las veces la página a reemplazar esta 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?

$$p \leq 0,42 \text{ ns}$$

29.- Anomalía de Belady

a) ¿Qué es?

Es un fenómeno que se refiere a un comportamiento no intuitivo en el contexto de la paginación de memoria en SO. Fue identificada por Leonard Belady en 1966. Ocurre en algoritmos de reemplazo de página

cuando se aumenta la cantidad de marcos de página disponibles, pero esto resulta en un aumento en la tasa de fallos de página en lugar de una disminución.

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

Solución en la carpeta de ejercicios resueltos Práctica-5.

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 almacenara en la primer 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 asumamos 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 cuantos fallos de páginas ocurren al ejecutar el programa (considere las veces que se ejecuta $C[i,j] = A[i,j] + B[i,j]$)

Si trabajamos con 3 matrices A, B y C, entonces, la cantidad de fallos de página totales es de $1024 * 3 = 3072$ veces.

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 páginas que ocurren.

Si se puede, si el for externo recorre por filas y el for interno recorre por columnas se reduce la cantidad de fallos ya que cada matriz va a generar 1 fallo cada 4 valores del for externo (porque cargamos en memoria 4 filas por matriz cada 4 iteraciones del for exterior). 64 veces se hace el for externo que recorre por filas.

Cada 4 filas (4 valores del for externo) una matriz generaría un fallo de página, es decir, $64 / 4 = 16$ es la cantidad de fallos por matriz. Si manipulamos 3 matrices A, B y C $\rightarrow 16 * 3 = 48$ fallos en total.

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 | 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 cual sería el conjunto de trabajo de los procesos A y B en el instante 24 (WSA(24) y WSB(24))

Se tiene que mirar desde el instante que nos dicen 24, 5 instantes hacia la izquierda contando el 24, es decir, el rango [20...24].

- WSA(24) = 4, 5, 6.
- WSB(24) = 2, 3, 5, 6.

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

- WSA(60) = 1, 2, 3, 4, 5.
- WSB(60) = 3, 4, 5.

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?

Con 8 frames no estamos en situación de trashing porque el WSA requiere de 3 frames y el WSB requiere 4, es decir, 7 en total. Pero si contamos con 6 si nos encontraríamos en situación de trashing.

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?

La tasa de fallos de páginas en el instante 38 es de 6

e) Para el valor obtenido en el inciso d), si suponemos que el S.O. utiliza como limites 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?

Al superar el límite superior, entramos en una situación de trashing. Para mejorar la tasa de fallos de página en el instante 38, que se calcula considerando los fallos de los últimos 10 instantes, asignar un marco adicional al proceso A reduciría la tasa a 5, lo que nos sacaría del trashing pero nos dejaría en el límite. Sin embargo, al aplicar esta solución a nivel de proceso, podría ocurrir la anomalía de Belady. Por lo tanto, una solución más segura sería reducir el grado de multiprogramación del sistema.