**Resolucion practica 5**

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

**Direccion Logica o Virtual:**

Se refiere a las direcciones que un proceso utiliza en su espacio de direcciones.

Es la direccion que el proceso ve como si fuera la memoria total disponible para el.

Cada proceso tiene su propio espacio de direcciones virtuales.

Las direcciones logicas/virtuales deben ser convertidas por el sistema operativo, puede ser usando MMU.

Es necesario porque la memoria fisica del sistema (seria la memoria RAM) está dividida en multiples procesos y no se asignan direcciones de memoria contiguas/secuenciales.

**Direccion Fisica:**

La direccion fisica es la direccion real de un dato o instruccion en la memoria fisicca(memoria RAM).

La direccion es la direccion que el hardware reconoce y que utiliza para acceder a la memoria fisica.

La memoria fisica esta formada por marcos/frames de pagina, o particiones. Cada parte de la emoria fisica tiene una direccion unica.

**Conclusion:**

Direccion Logica/virtual: Es la direccion que usa un proceso, es abstracta y gestionada por el so.

Direccion Fisica: Es la direccion real en la memoria RAM, utilizada por el hardware para acceder a los datos.

2.- En la técnica de Particiones Múltiples, la memoria es divida 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 si):

Particiones Fijas

Particiones Dinámicas

**a) Explique como trabajan estos 2 métodos. Cite diferencias, ventajas y**

**desventajas.**

Particiones fijas: La memoria se divide en un numero fijo de particiones del mismo tamaño.

Cuando un proceso llega se asigna a una particion que tenga el espacio suficiente para contener el proceso.

Una de sus ventajas es que La implementacion es simple.

Otra de sus desventajas es que, las particiones al ser de tamaño fijo, si el proceso es mas pequeño que la particion asignada, se desperdicia espacio,generando fragmentacion interna.

**Particiones dinamicas:**

La particion se va ajustando de acuerdo al requerimiento del proceso, se van creando y desapareciendo de acuerdo a la demanda, cuando un proceso pide memoria se va a crear una nueva particion,

y cuando un proceso termina su ejecucion, esa particion se libera.

**ventajas y desventajas:**

Una de sus ventajas es que no ocurre fragmentacion interna, los procesos usan solo la memoria que necesitan.

Una de sus desventajas es que puede haber fragmentacion externa, ya que al asignarse y liberarse bloques de memoria de distintos tamaños, el espacio libre puede quedar fragmentado.

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

El SO debe disponer de:

Tabla de Particiones o Segmentos:

En particiones fijas, registra qué particiones están libres u ocupadas.

En particiones dinámicas, rastrea el tamaño y ubicación de los bloques de memoria libres y ocupados.

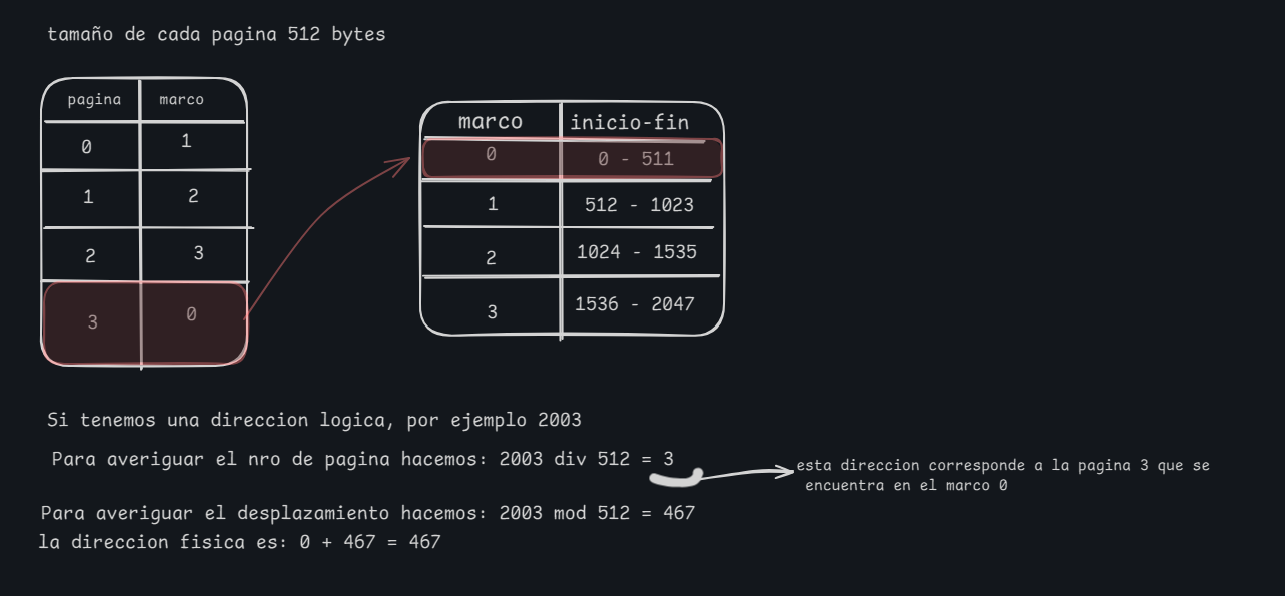
Tamaño de los procesos: Para determinar si entran en una particion o en un bloque dinamico.

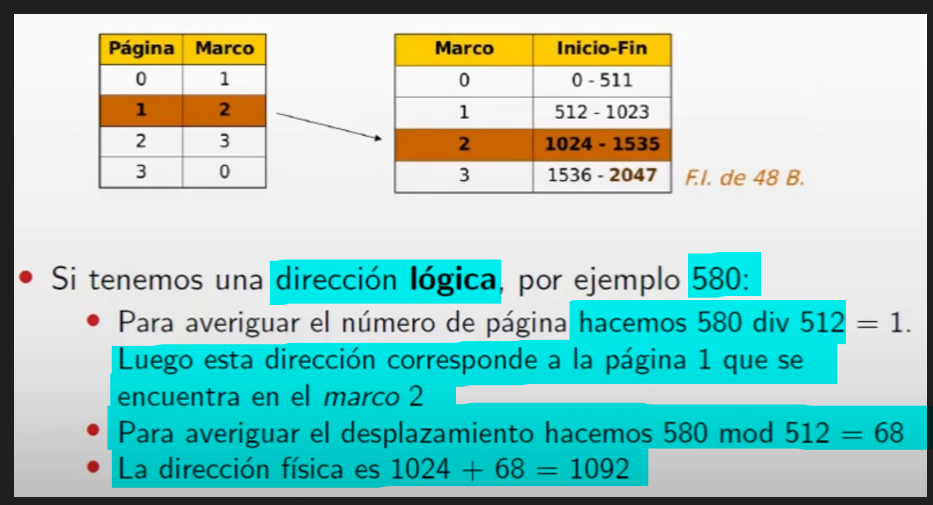
Algoritmos de asignacion: Para decidir como ubicar los procesos (First Fit, Best Fit, Worst Fit).

Tabla de procesos: Relaciona cada proceso con su ubicacion en memoria.

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

Lo hice siguiendo el ejemplo de la teoría.





**3.- Al trabajar con particiones fijas, los tamaños de las mismas se pueden considerar: Particiones de igual tamaño o Particiones de diferente tamaño.**

Cite ventajas y desventajas de estos 2 métodos.

Particiones de igual tamaño:

Facil de implementar y administrar, ya qye todas las particiones tienen el mismo tamaño.

La desventaja puede ser que si un proceso es mucho mas chico que la partición, se desperdicia espacio, generando fragmentación interna.

Particiones de diferente tamaño:

Los procesos se asignan a la partición que mas se ajuste a su tamaño, lo que reduce la fragmentación interna.

DesventajasL Complejidad para administrarlos, es necesario tener un registro detallado de las particiones, tamaños y estados (libres u ocupados). Ademas, aunque la fragmentación interna se reduce, pueden quedar espacios de memoria no utilizables entre particiones ocupadas

Conclusion: Las particiones de igual tamaño son mas simples pero menos eficientes en la utilización de memoria. En cambio las particiones de diferente tamaño son mas flexibles al momento de almacenar procesos, pero requieren mas recursos de gestión y pueden ser mas lentas en la asignación.

**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: Se produce en particiones físicas, es una porción/espacio de memoria que queda sin utilizar, y no pueden ser ocupadas por otro proceso, porque es una partición por proceso.

Fragmentacion externa: Se produce en particiones dinamicas, son huevos que van quedando en la memoria a medida que los procesos finalizan.

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

Una **solución** para la fragmentación externa es la compactación, pero es muy costosa de realizar.

**c) Que es la compactacion?**

Una técnica específica para reducir la **fragmentación externa**.  
Consiste en reorganizar los procesos y bloques libres en la memoria física para formar un bloque contiguo de memoria libre. Requiere mover los bloques de memoria en uso hacia un extremo de la memoria para liberar un bloque grande y continuo.

**5.- Paginación**

**a) Explique como trabaja este método de asignación de memoria.**

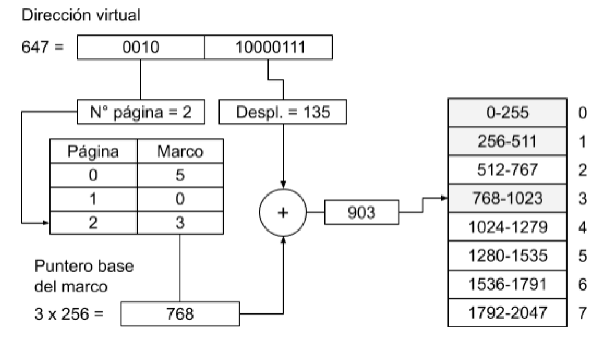
La **paginación** es un método de administración de memoria en el que:

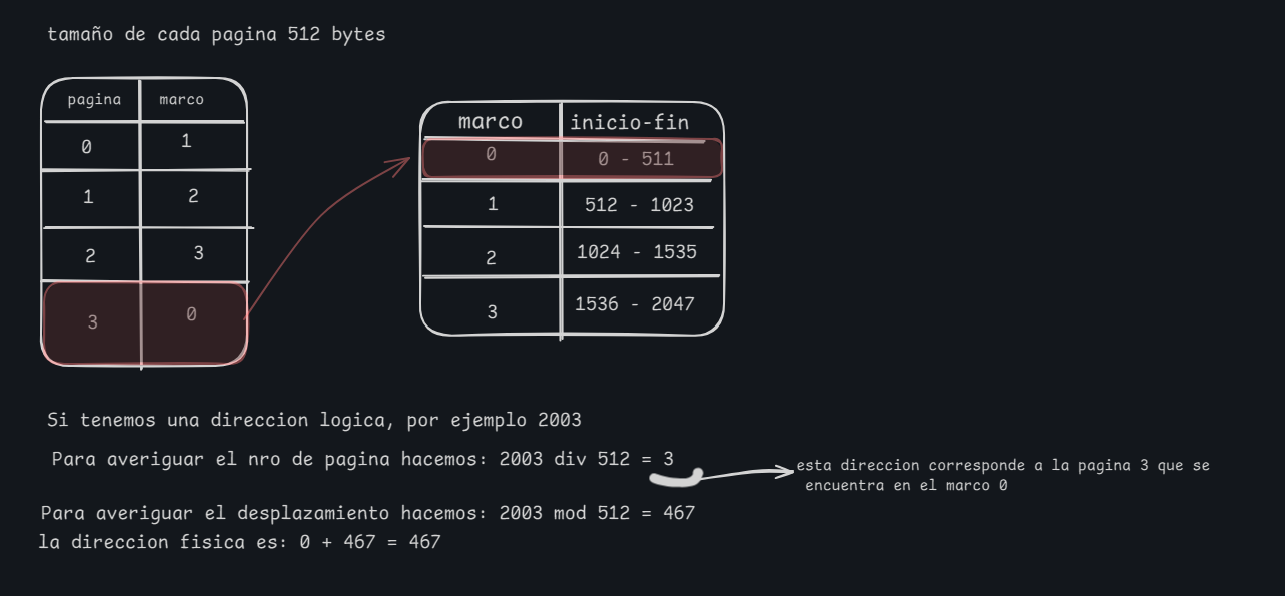
1. **División de Memoria Física**:  
   La memoria principal se divide en bloques de tamaño fijo llamados **marcos**.
2. **División de Procesos**:  
   Cada proceso se divide en bloques lógicos del mismo tamaño que los marcos, llamados **páginas**.
3. **Correspondencia Página-Marco**:  
   Las páginas de un proceso se asignan a marcos disponibles en la memoria, no necesariamente contiguos.
4. **Traducción de Direcciones**:  
   Una posición en la memoria se determina mediante:
   * El **número de página**, que indica qué página contiene la dirección lógica.
   * El **desplazamiento** dentro de la página, que identifica la posición exacta dentro de esa página.  
     Estas dos partes permiten calcular la **dirección física** en la memoria principal.

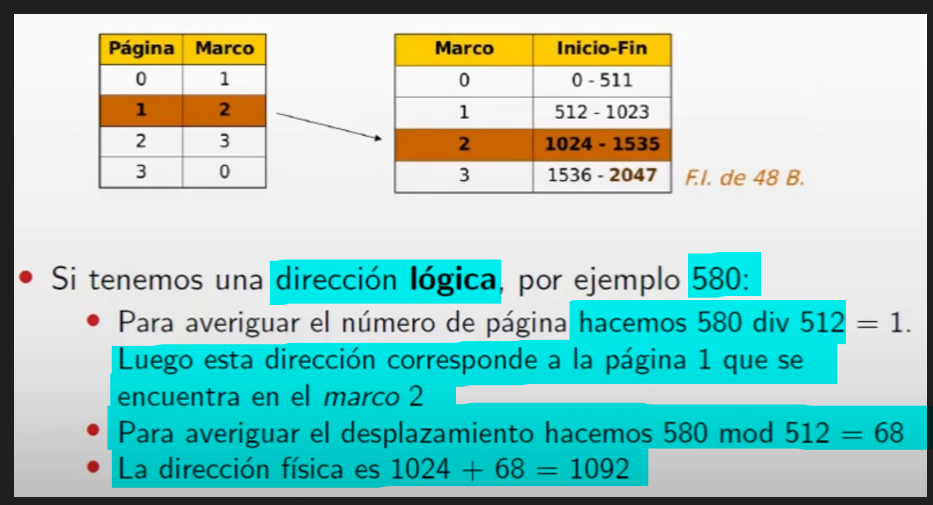
**b) ¿Qué estructuras adicionales debe poseer el SO para llevar a cabo su implementación?**

El SO debe mantener una tabla de paginas por cada proceso, donde cada entrada de dicha tabla contiene el marco en la que se coloca dicha pagina. Se guarda la base del marco que se cargo en la PCB.

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







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

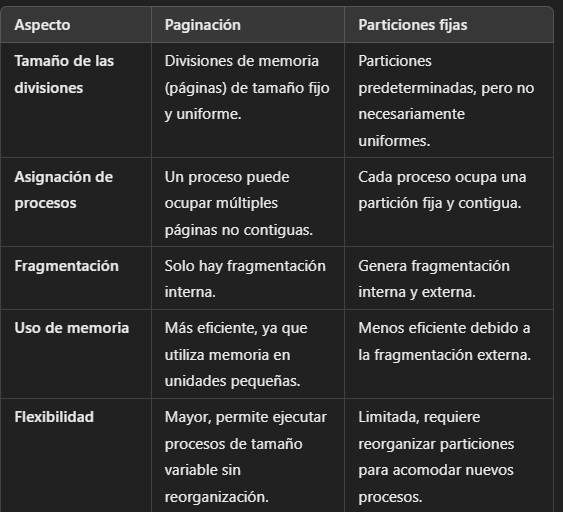
No ocurre fragmentación externa en la paginación, porque cualquier pagina de un proceso puede asignarse a cualquier marco disponible en la memoria física, sin importar su ubicación.

Fragmentacion interna si podría ocurrir en paginación, porque las paginas tienen un tamaño fijo. ***Por ende el espacio no utilizado en la ultima pagina del proceso genera desperdicio***

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

Similitudes: Ambas son técnicas utilizadas para gestionar la memoria principal en sistemas operativos. Ambas técnicas pueden causar fragmentación.

Diferencias:



**7) a)**

|  |  |
| --- | --- |
| Pagina | Marco |
| 0 | 3 |
| 1 | 5 |
| 2 | 2 |
| 3 | 6 |

Para calcular la dirección física de cada pagina hacemos = ( pagina \* tamaño pag ) - 1

Un proceso con 2000 bytes y con cada pagina de 512 bytes, va a usar 3 paginas completas y una pagina incompleta (la ultima pagina seria la incompleta).

La ultima pagina solo se van a usar = 2000 mod 512 = 464 bytes.

Por ende, la pagina 3 su dirección virtual seria: (1536 + 464) -1 = 1999

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Marco | Pagina |  | Direccion Virtual | Direccion Fisica |
| 0 | - |  |  | 1. - 511 |
| 1 | - |  |  | 512 – 1023 |
| 2 | 2 |  | 1024 …. 1535 | 1024 – 1535 |
| 3 | 0 |  | 0…. 511 | 1536 – 2047 |
| 4 | - |  |  | 2048 - 2559 |
| 5 | 1 |  | 512…. 1023 | 2560 – 3071 |
| 6 | 3 |  | 1536…1999 | 3072 - 3583 |

**Direccion Virtual = (Nro Pagina \* Tamaño Pagina) + Desplazamiento**

**Direccion Fisica = (Nro Marco \* Tamaño Pagina) + Desplazamiento**

**Duda: El desplazamiento en este caso seria 0? Porque nos interesa saber donde arranca y donde termina la dirección virtual en ese marco**

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Direcc Virtual | DIV 512  Nro Pagina | MOD 512  Desplazamiento | Marco  (base) | Direcc Fisica | Valida |
| 35 | 0 | 35 | M3 (1536) | 1536+35 =1571 | Si |
| 512 | 1 | 0 | M5 (2560) | 2560 + 0 = 2560 | Si |
| 2051 | 4 | 3 | Error |  | NO |
| 0 | 0 | 0 | M3(1536) | 1536 + 0 = 1536 | Si |
| 1335 | 2 | 301 | M2(1024) | 1024 + 301= 1325 | Si |
| 602 | 1 | 90 | M5(2560) | 2560 + 90= 2650 | Si |

**c) Indicar en caso de que se aposible, las direcciones ogicas que se correspondan con las siguientes direcciones físicas**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Direcc Fisica | DIV 512  Nro Marco | MOD 512  Desplazamiento | Pagina  (base) | Direcc Logica | Valida |
| 509 | 0 | 519 | Error | Error | NO |
| 1500 | 2 | 476 | P2 (1024) | 1024 + 476 = 1500 | SI |
| 0 | 0 | 0 | Error |  | NO |
| 3215 | 6 | 143 | P3(3072) | 3072 + 143 = 3215 | Sii |
| 2014 | 3 | 478 | P0 (1536) | 1536 + 478=2014 | SI |
| 2000 | 3 | 464 | P0(1536) | 1536+464=2000 | SI |

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

Fragmentacion externa, no se produce cuando usamos paginación.

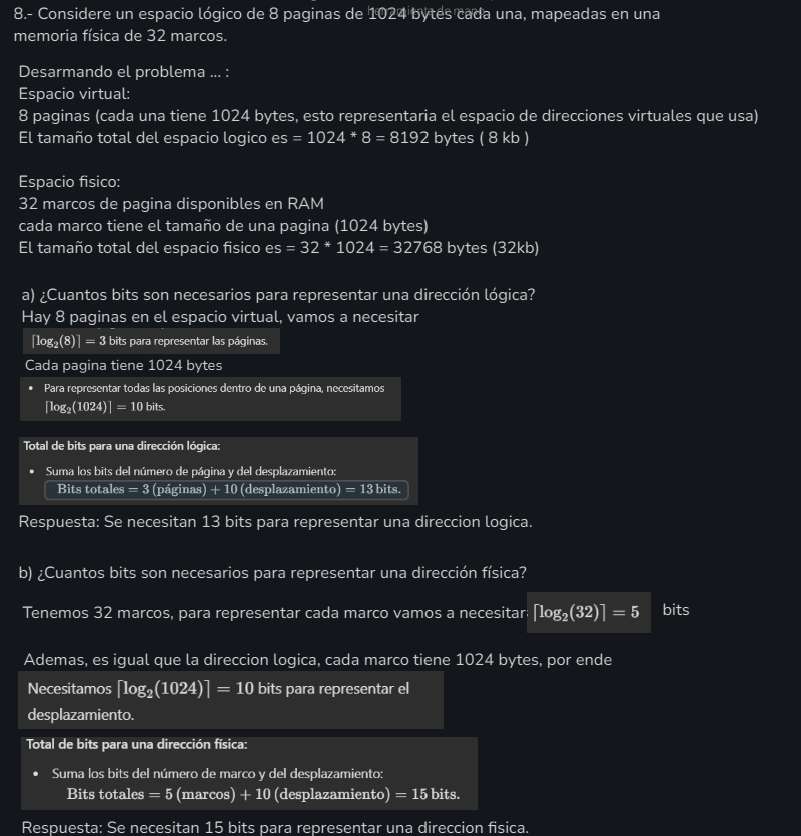
Se produce **fragmentación interna en la ultima pagina** , para calcularla hacemos

2000 mod 512 = 464 esta es la cantidad de bytes ocupados de la ultima pagina

Si en cada pagina podemos almacenar 512 bytes, y en la ultima almacenamos 464bytes

Tenemos una fragmentación interna de 512 – 464 = 48 bytes.

**8)**



**9.- Segmentación**

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

La segmentación es una técnica de gestión de memoria que divide el espacio de direcciones de un programa en segmentos de tamaño variable. (a diferencia de la paginación que divide la memoria en bloques de tamaño fijo). La segmentación permite que cada segmento tenga un tamaño dinamico y represente una unidad lógica del programa. La segmentación puede causar fragmentación externa.

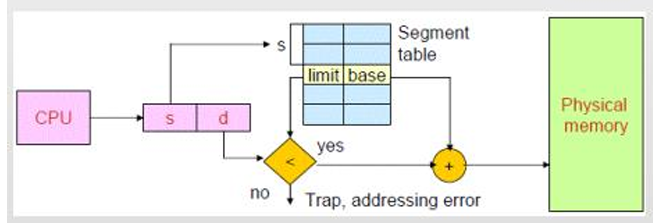
Las direcciones lógicas constan de 2 partes:

Selector de segmento y desplazamiento dentro del segmento (sobre le registro base y limite)

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

Una tabla de segmentos, donde por cada segmento hay dos valores: registro base y registro límite.

**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, debido a la forma en que se asigna y libera memoria a los segmentos, lo que puede generar espacios no contiguos de memoria libre en la memoria física.

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

Similitudes: Ambas técnicas permiten la asignación dinámica de memoria. Esto significa que la memoria se asigna de acuerdo a las necesidades de los procesos que se estén ejecutando en el sistema. Tanto las particiones dinamicas como la segmentación reducen la fragmentación interna, ya que la memoria se asigna en función de la cantidad de memoria necesaria para cada proceso. Ambas técnicas pueden sufrir segmentación externa si no se gestionan adecuadamente.

Diferencias: la segmentación asigna memoria a nivel de segmentos locicos, mientras que las particiones dinamicas asignan memoria en particiones visicas de tamaño variable.

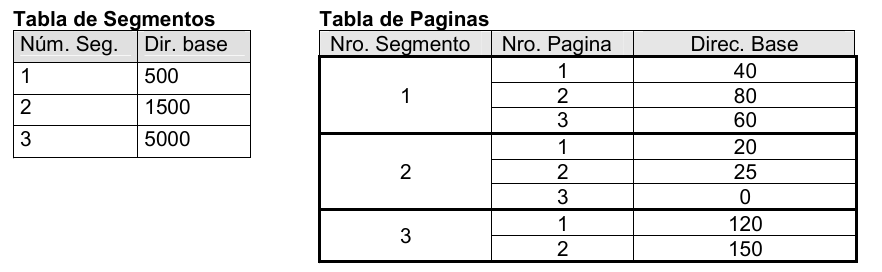
La segmentación permite una mayor protección y aislamiento entre diferentes tipos de datos, mientras que las particiones dinamicas asignan bloques contiguos de memoria.

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

Similitudes y diferencias:

Tanto en paginación como en segmentación, la memoria se divide en bloques mas pequeños para facilitar la asignación. En paginación, estos bloques se llaman paginas(logicas) y marcos(fisicos), mientras que en segmentación se llaman segmentos. Ambas técnicas buscan reducir la fragmentación interna. En la técnica de paginación podemos tener fragmentación interna, mientras que en la técnica de segmentación podemos tener fragmentación externa.

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



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

1. (2,1,1)
2. (1,3,15)
3. (3,1,10)
4. (2,3,5)

***Por lo que entendí, la dirección física en segmentación paginada, la podemos sacar con la siguiente fórmula:***

***Dirección base de tabla de segmentos + dirección base de tabla de paginas + desplazamiento***

1. 1500 + 20 + 1 = 1521
2. 500 + 60 + 15 = 575
3. 5000 + 120 + 10 = 5130
4. 1500 + 0 + 5 = 1505

**13) Memoria Virtual**

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

La memoria virtual permite ejecutar procesos que requieren más memoria que la disponible en el sistema, manteniendo en memoria principal solo aquella memoria que el proceso este utilizando y el resto en disco. De esta forma el usuario ya no debe preocuparse por las limitaciones de memoria física.

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

En el hardware, este debe ser capaz de detectar cuándo una instrucción está tratando de acceder a una dirección que no está en el momento cargada en memoria, y a partir de allí recién el SO podría generar las instrucciones necesarias para atender el fallo. Si esta tarea dependiera exclusivamente del SO, se debería emplear mucho tiempo para detectar si una dirección lógica hace referencia a una porción de programa ya cargada.

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

Se requiere como mínimo un bit que indique la presencia de la página en memoria, dado que sólo a partir de ese dato es que el hardware puede generar la interrupción necesaria para resolver el fallo de página. Por otro lado, debe contarse con información sobre la presencia de modificaciones o no sobre las páginas cargadas, dado que cualquier descarga de una página modificada implica la necesidad de actualizar los datos cargados en disco, para mantener la consistencia.

Otros bits de control pueden resultar útiles para definir el historial de uso del conjunto de páginas o segmentos de un proceso, o de todos los cargados, a fin de poder decidir con mayor grado de precisión cuál porción puede ser la mejor víctima para un reemplazo en caso de presentarse un fallo de página.

**14.- Fallos de Página (Page Faults):**

**a) ¿Cuándo se producen?**

Cuando una instrucción ejecutada hace referencia a una dirección lógica cuya página no está cargada en memoria.

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

El hardware, generando una interrupción. Al momento de resolver la dirección, cuando recupera la entrada en la tabla de páginas correspondiente, debe analizar el bit de control V, que indica si la página se encuentra o no cargada en memoria principal

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

1. Se genera el trap.

2. El SO bloquea al proceso (la CPU toma otro proceso).

3. El SO busca un marco 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. :

o Colocando el bit V en 1, en la página correspondiente.

o Coloca la dirección base del marco donde se colocó la página.

6. El proceso pasa del estado bloqueado a listo para ejecutar

7. Cuando vuelva a ser asignado al CPU, el proceso comenzará desde la instrucción que generó el fallo en primera instancia

**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”)?**

El espacio virtual de 32 bits significa que se pueden representar 2^32 direcciones,

2^32 direcciones máximas \* 1 byte = 2^32 bytes → 4.294.967.296 bytes→4 GB

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

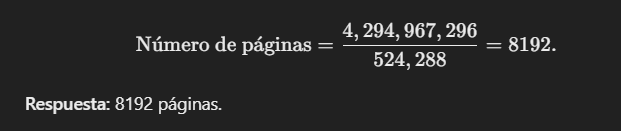
**Importante: Pasar todo a bytes asi trabajamos con la medida de bytes**

Para calcular el numero máximo de paginas, **dividimos** el espacio virtual total entre el tamaño de cada pagina.

Espacio virtual total = 4GB = 4.294.967.296bytes

Tamaño de cada pagina: 512 kb \* 1024 bytes = 524.288 bytes (pasamos a bytes)

Cantidad de paginas = espacio virtual / Tamaño de cada pagina



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

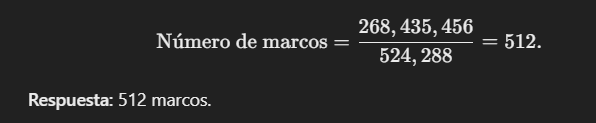
**Importante: Pasar todo a bytes asi trabajamos con la medida de bytes**

Para calcular el numero de marcos que puede haber en la memoria real, debemos **dividir** el tamaño total de la memoria real entre el tamaño de cada marco (equivale al tamaño de cada pagina)

Tamaño de la memoria real = 256 \* 1024 \* 1024 = 268.435.456 bytes

Tamaño de cada pagina = 512 \* 1024 = 524,288 bytes.

Numero de marcos : Tamaño memoria real / Tamaño de cada pagina



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

**El tamaño máximo de la tabla de paginas se calcula como el producto del numero máximo de paginas y el tamaño de cada entrada en la tabla.**

Tamaño máximo de la tabla = Numero de paginas \* Tamaño de cada entrada

Numero máximo de paginas: 8192 (calculado en ii)

Tamaño de cada entrada 2kb = 2 \* 1024 = 2048 bytes

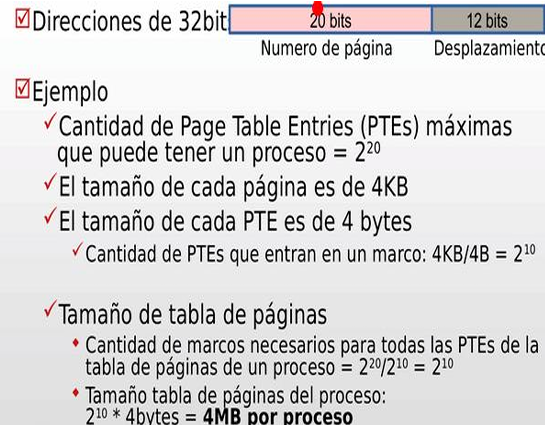
Tamaño máximo de la tabla = 8192 \* 2048 = 16,777,216bytes = 16 mg

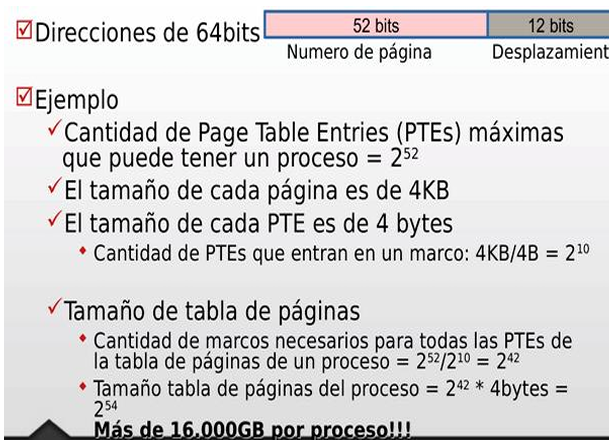
Respuesta 16 mb.

**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 paginas. Existen varios enfoques para administrar las tablas de páginas:**

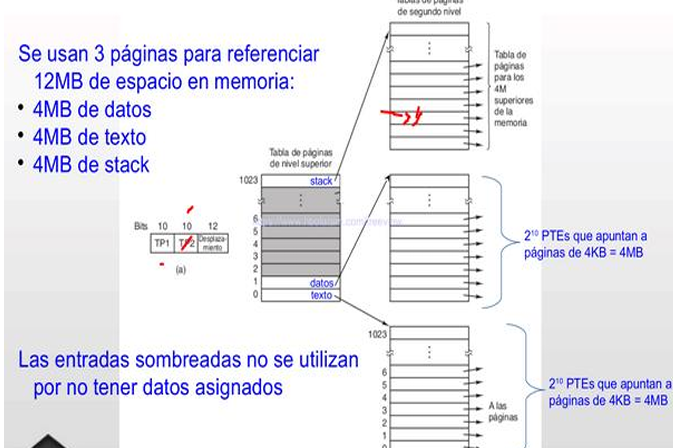
**Tablas de páginas de 1 nivel. Tablas de páginas de 2 niveles. Tablas de páginas invertidas.**

**Tabla de paginas**

* **Cada proceso tiene su propia tabla de paginas**
* **El tamaño de la tabla de páginas depende del espacio de direcciones del proceso.**
* **Tabla de 1 nivel:**
* ****

****

**Tabla de 2 niveles:**

****

**El propósito de la tabla de páginas multinivel es dividir la tabla de páginas lineal en múltiples tablas de páginas.**

**• Cada tabla de paginas suele tener el mismo tamaño pero se busca que tengan un menor numero de paginas por tabla.**

**• La idea general es que cada tabla sea más pequeña.**

**• Se busca que la tabla de paginas no ocupe demasiada RAM.**

**Además solo se carga una parcialidad de la tabla de paginas (la que se necesite usar).**

**• Existe un esquema de direccionamientos indirectos.**

**o Beneficios?**

**▪ Las tablas de segundo nivel (o más) se pueden llevar a memoria secundaria, liberando RAM.**

**o Desventaja?**

**▪ Más de un acceso a memoria para obtener un dato.**

**Tablas de paginas invertidas**

**Que son las tablas de paginas invertidas?**

Es una técnica utilizada para manejar espacios de direcciones virtuales muy grandes.

Contienen una entrada por cada marco de pagina en memoria real, en lugar de una entrada por cada pagina virtual. Es una única tabla para todo el sistema, cargada en memoria RAM.

El espacio de direcciones de la tabla corresponde al espacio físico (RAM) no al virtual de los procesos.

**Funcionamiento:**

El numero de pagina virtual se convierte en un valor de hash, que sirve como índice para buscar en la tabla.

Solo se almacenan las entradas correspondientes a paginas presentes en memoria RAM.

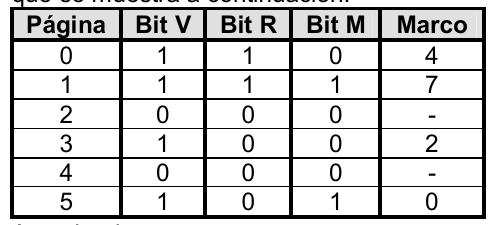
**Ventajas:**

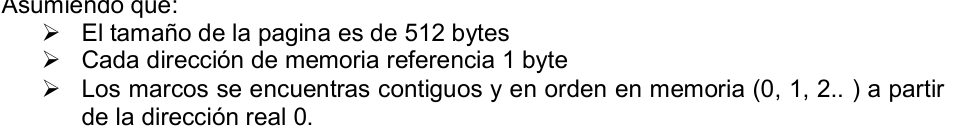
Reduce el tamaño de las tablas de paginas, ya que hay una entrada por marco físico y no por pagina virtual.

**Conclusión:**

Las tablas de páginas invertidas son una solución eficiente para sistemas con grandes espacios de direcciones, optimizando el uso de memoria y reduciendo la sobrecarga del sistema, pero dependen de una buena implementación de hash y manejo de colisiones.

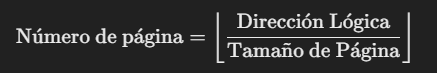
17:

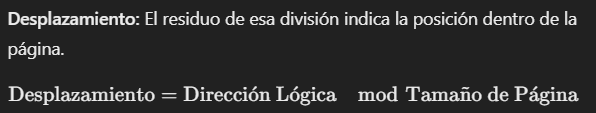


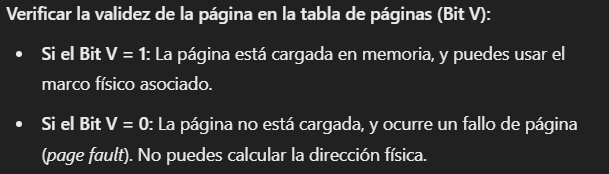


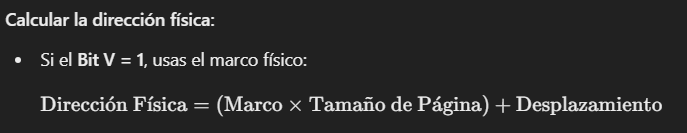
Para sacar la dirección física a partir de una dirección virtual hacemos:

Nro de pagina (sirve para conocer el bit V y saber si la pagina esta cargada o no)









**¿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

Nro de pagina = 1052 DIV 512 = 2

Desplazamiento = 1052 MOD 512 = 28

Pagina 2 Bit V = 0, indica que la pagina no esta cargada en memoria física.

Fallo de pagina.

b) 2221

Nro de pagina= 2221 DIV 512 = 4

Desplazamiento = 2221 MOD 512 = 173

Pagina 4 Bit V = 0, indica que la pagina no esta cargada en memoria física.

Fallo de pagina.

c) 5499

Nro de pagina = 5499 DIV 512 = 10

Despazamiento = 5499 MOD 512 = 379

Pagina 10 Bit V = 0 (no existe.) Por ende la dirección física no puede calcularse.

Direccion invalida.

d) 3101

Numero de pagina = 3101 DIV 512 = 6

Desplazamiento = 3101 MOD 512 = 29

Pagina 6 Bit V = 0 no existe.) Por ende la dirección física no puede calcularse.

Direccion invalida

**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.**

|  |  |  |
| --- | --- | --- |
|  | **Paginas pequeñas** | **Paginas grandes** |
| **Ventajas** | **Menos desperdicio de memoria** debido a la asignación de memoria mas precisa.  **Menos fragmentación externa** ya que cada pagina es mas fácilmente ajustable a las necesidades de los procesos. | **Menos fallo de pagina:** Con paginas mas grandes, se pueden cargar mas datos de una sola vez, lo que reduce la cantidad de fallos de pagina, ya que una sola pagina puede cubrir mas espacio de memoria virtual.  **Reduccion de overead de las tablas de paginas:** Al haber menos paginas, las tablas de paginas se hacen mas pequeñas y se requiere menos memoria para almacenarlas. |
| **Desventajas** | **Mayor overhoad de gestión de paginas:** Al tener mas paginas por proceso, el SO necesita gestionar mas entradas en la tabla de paginas.  **Mayor tasa de fallos de pagina:** Debido a que las paginas son mas pequeñas, es mas probable que se produzcan fallos de pagina cuando el proceso requiere mas memoria de la que puede cargarse en una sola pagina. | **Desperdicio de memoria (fragmentación interna):** Si un proceso no necesita toda la memoria de una pagina grande, habrá desperdicio dentro de esa pagina.  **Mayor latencia en caso de fallos de pagina:** Cuando ocurre un fallo de pagina, la cantidad de datos que deben ser cargados o descargados es mucho mayor.  **Mayor fragmentación interna:** Si un programa necesita cargar varias paginas grandes, podría haber suficiente espacio contiguo disponible. |

**Conclusion:**

Las paginas pequeñas son mas adecuadas para sistemas que requieren alta eficiencia en el uso de memoria y cuando los procesos tienden a ser pequeños o medianos.

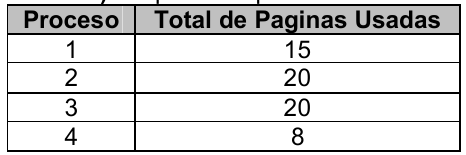
Las paginas grandes son mas adecuadas para sistemas que ejecutan aplicaciones grandes que requieren mucha memoria, minimizan la cantidad de fallos de pagina y la sobrecarga de gestión de memoria.

**19)**

**a) Descripción de las políticas de asignación de marcos**

|  |  |
| --- | --- |
| **Asignacion Fija** | **Asignacion Dinamica** |
| Se asigna un número fijo de marcos a cada proceso desde el momento en que se carga en memoria principal  **Ventaja:** Simplifica la gestión de memoria.  **Desventaja:** Puiede provocar ineficiencias, ya que algunos procesos pueden recibir mas marcos de los necesarios y otros menos, generando fragmentación o falta de paginas. | El numero de marcos asignados a un proceso puede variar durante su ejecución. Esto se hace en función de las necesidades del proceso.  **Ventaja:** Mejora el uso eficiente de la memoria.  **Desventaja:** Incrementa la complejidad del SO para monitorear y ajustar dinámicamente los marcos. |

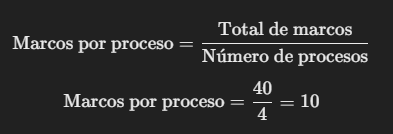
**b) Asignación de marcos con las técnicas: Reparto equitativo y Reparto proporcional.**

**Total de marcos = 40**

**Reparto Equitativo:**

**En esta técnica, los marcos se distribuyen de forma igual entre todos los procesos.**

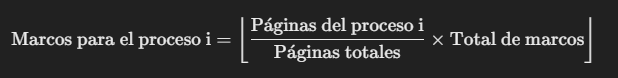
**Se divide el total de marcos por el numero de procesos.**



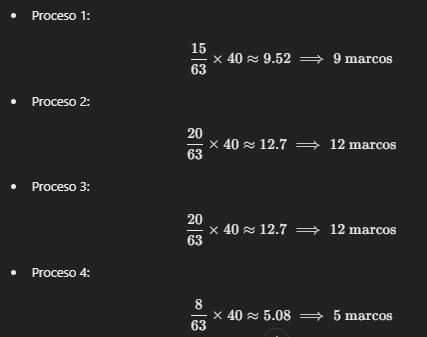
**Por ende, cada proceso va a tener 10 marcos.**

**Reparto Proporcional:**

**En esta técnica, los marcos se asignan en proporción al numero de paginas que usa cada proceso.**



**Total de paginas usadas: 15 + 20 + 20 + 8 = 63**

****

**c) Análisis de eficiencia de los repartos:**

**Reparto Equitativo:**

* + Este reparto **asigna el mismo número de marcos a todos los procesos,** independientemente de sus necesidades reales.
  + **Ineficiente porque procesos con más páginas (Proceso 2 y Proceso 3) podrían generar más fallos de página al tener menos marcos de los necesarios.**

**Reparto Proporcional:**

* + Este reparto **asigna marcos en función de las necesidades de cada proceso.**
  + **Más eficiente porque reduce los fallos de página al priorizar procesos que requieren más memoria**.

**Conclusión:** El reparto proporcional es más eficiente, ya que optimiza la asignación de marcos en función de las necesidades reales de cada proceso, minimizando la cantidad de fallos de página.

**20.- Reemplazo de páginas (selección de una victima):**

**¿Qué sucede cuando todos los marcos en la memoria principal están usados por las páginas de los procesos y se produce en fallo de página?**

**El SO debe seleccionar una de las páginas que se encuentra en memoria como victima, y ser reemplazada por la nueva página que produjo el fallo.**

**Considere los siguientes algoritmos de selección de victimas 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):** El algoritmo **reemplaza la pagina que lleva mas tiempo en memoria,** **sin considerar si sigue siendo utilizada**. Una desventaja es que al aumentar el numero de marcos, puede incrementar los fallos de pagina.

**Segunda Chance:** Similar a FIFO, pero brinda una oportunidad extra a las paginas que han sido referenciadas, mejorando la tasa de fallos en comparación con FIFO.

**LRU (Least Recently Used)**:Reemplaza la pagina menos recientemente usada. Tiende a ser mas eficiente porque utiliza el historial de accesos recientes para determinar que paginas probablemente no se necesitaran.

**OPT(Optimo):** Es el mejor algoritmo en termino de tasa de fallos, porque reemplaza la pagina que no será utilizada durante mas tiempo en el futuro. Es impráctico en la vida real ya que requiere conocimiento perfecto del futuro.

**b) Analice su funcionamiento. ¿Como los implementaría?**

**FIFO:**

Funcionamiento: Las paginas se manejan como una cola, la pagina que ingreso primero es la primera en salir cuando ocurre el fallo.

*Implementación*: Usar una cola circular para representar los marcos, reemplazar el elemento en la parte frontal al producirse un fallo.

**Segunda chance:**  Se da una “segunda oportunidad” a las paginas referenciadas**.** Utiliza un bit de referencia que indica si la pagina ha sido accedida recientemente.

Implementación: Usar un puntero circular, si el bit de referencia de una pagina es 1, restablecerlo a 0 y avanzar al siguiente marco hasta encontrar una pagina con bit 0 para reemplazar.

**LRU (Least Recently Used):**  Reemplaza la pagina que no ha sido usada por mas tiempo. Asume que las paginas mas antiguas probablemente no serán usadas pronto.

**OPT (Optimo):** Reemplaza la pagina que no será utilizada durante el mayor periodo en el futuro. Requiere conocer las futuras referencias de paginas.

Implementación: Identificar la pagina que no será accedida por mas tiempo y reemplazarla.

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

Cuando una pagina que tiene que ser reemplazada, esta modificada, el SO debe:

1. Escribir la pagina en el almacenamiento secundario.

**Si la pagina tiene el bit de modificación activado**: Sse debe escribir en el disco antes de ser reemplazada para evitar perdida de datos.

**Si la pagina tiene el bit de modificación desactivado**: Se puede descartar sin escribir en el disco.

1. Actualizar las tablas de paginas:

Marcar la pagina como no presente en memoria principal.

Actualizar la entrada de la nueva pagina, indicando su marco de memoria y los bits correspondientes (presencia,referencia,modificación,etc).

1. Gestionar la interrupcion de fallo de pagina:

Asignar un marco al proceso que solicito la nueva pagina

Leer la nueva pagina desde el disco hacia el marco asignado.

**21.- Alcance del reemplazo Al momento de tener que seleccionar una pagina victima, el SO puede optar por 2 políticas a utilizar:**

**Reemplazo local , o Reemplazo global**

1. **Describa como trabajan estas 2 políticas.**

**Reemplazo Local:**  Cada proceso tiene asignado un conjunto fijo de marcos en la memoria principal. Cuando ocurre un fallo de pagina, el reemplazo se realiza únicamente dentro de los marcos asignados a ese proceso. No afecta los marcos de otros procesos.

**Ventajas**: Previene que un proceso con alto índice de fallos, degrade el rendimiento de otros procesos.

**Reemplazo Global:** Los marcos de la memoria principal se consideran como un conjunto compartido entre todos los procesos. Cuando ocurre un fallo de pagina, el SO puede reemplazar cualquier marco en memoria, independientemente del proceso que lo este utilizando.

**Desventajas:** Puede causar “robo de marcos”, donde un proceso con muchos fallos, afecta el rendimiento de otros procesos.

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

**Asignacion fija de marcos:**

Asigna a cada proceso un numero fijo de marcos en memoria principal,determinado de antemano. La cantidad de marcos asignados a cada proceso no cambia durante su ejecución.

Por lo tanto, no es posible combinar la asignación fíja con reemplazo global, ya que la política de reemplazo global contradice el principio fundamental de asignación fija de marcos.

Por que? En reemplazo global, un proceso puede utilizar marcos asignados a otros procesos, mientras que la asignación fija, asegura que los marcos asignados a un proceso sean exclusivos para este.

Resumen: La política de reemplazo global requiere asignación variable de marcos, mientras que la asignación fija funciona en conjunto de políticas locales.

**25.- Hiperpaginación (Trashing)**

**a) ¿Qué es?**

La hiperpaginacion o trashing se da cuando el rendimiento del SO se degrada porque se pasa mas tiempo cargando paginas o atendiendo fallos de pagina, que en ejecutar los procesos.

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

El motivo principal puede ser la falta de marcos. Si el tamaño de la RAM no es suficiente para alojar las paginas de los procesos en ejecución, el SO necesita intercambiar constantemente paginas entre RAM y disco. Si hay varios procesos en ejecución simultanea y la demanda de memoria supera la capacidad disponible, es un motivo por el cual podemos entrar en hiperpaginacion.

**c) ¿Cómo la detecta el SO?**

Un incremento continuo en la tasa de fallos de pagina combinado con un bajo uso efectivo del procesador es un indicador de trashing.

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

Puede tomar varias acciones, pero lo ideal seria sacarle marcos a aquellos procesos que les sobren marcos, y asignarle los marcos sobrantes a los procesos que generen fallos de pagina, es decir, a los procesos que le falten marcos.

**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:**

|  |  |  |  |
| --- | --- | --- | --- |
| **USO** | **CPU %13 PAGINACION % 97** | **CPU % 87 PAGINACION % 3** | **CPU %13 PAGINACION % 3** |
| **Que sucede?** | El sistema esta en estado de hiperpaginacion. La mayoría de los recursos están utilizados para manejar fallos de pagina. | El sistema esta funcionando de manera eficiente. La CPU esta ocupada ejecutando instrucciones y los procesos tienen suficientes paginas en memoria para evitar fallos de pagina frecuentes.  Tiene un uso minimo de el dispositivo de paginación, lo cual es bueno. | El sistema esta en un estado de baja carga. La CPU no esta siendo aprovechada, y el dispositivo de paginación también tiene un uso minimo.  Tanto la CPU como el dispositivo de paginación tienen un uso minimo. (esto puede ocurrir si los procesos están en espera por ejemplo en operaciones e/s) |
| **Incrementar el nivel de multiprogramación aumentara el uso de la cpu?** | NO. Aumentar el nivel de multiprogramación solo empeoraría el problema, ya que habría mas procesos compitiendo por la misma cantidad de marcos, lo que incrementaría aun mas los fallos de pagina. | Si, se podría aumentar ligeramante el nivel de multiprogramación para aprovechar el tiempo de CPU no utilizado. | S, incrementar el nivel de multiprogramación implicaría aumentar la cantidad de procesos, por ende aumentaría el uso de la CPU, ya que actualmente estaría con un uso muy bajo.  Es el único caso que incrementar el nivel de multiprogramación podría ser beneficioso. |
| **La paginación esta siendo útil para mejorar el uso de la cpu?** | No, la tecnica de paginación estaría perjudicando el rendimiento. | Si, la paginación esta siendo útil, ya que permite manejar los procesos en memoria de manera eficiente sin degradar el rendimiento | Es difícil saber porque el uso de la CPU es bajo, pero actualmente la técnica de paginación no parece que impacte negativamente. |
| **Solucion?** | Una posible solución seria reducir el nivel de multiprogramación o aumentar la cantidad de memoria física disponible. | **…………………** | **……………………….** |

**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 mas rápido**

No es una solución efectiva. El problema principal es el alto numero de fallos de pagina que provocan que la CPU espere constantemente por la memoria. Es decir, importa la memoria, no la velocidad del procesador.

**b) Instalar un dispositivo de paginación mayor**

Podria acelerar el proceso de almacenamiento secundario(paginacion), pero no solucionaria el problema fundamental, que es que los procesos no tienen suficientes marcos en memoria principal, por eso se generan fallos de pagina excesivos.

**c) Incrementar el grado de multiprogramación**

Incrementar el grado de multiprogramación aumentaría la competencia por los marcos en memoria principal, por lo que aumentaría los fallos de pagina y empeorando la hiperpaginacion.

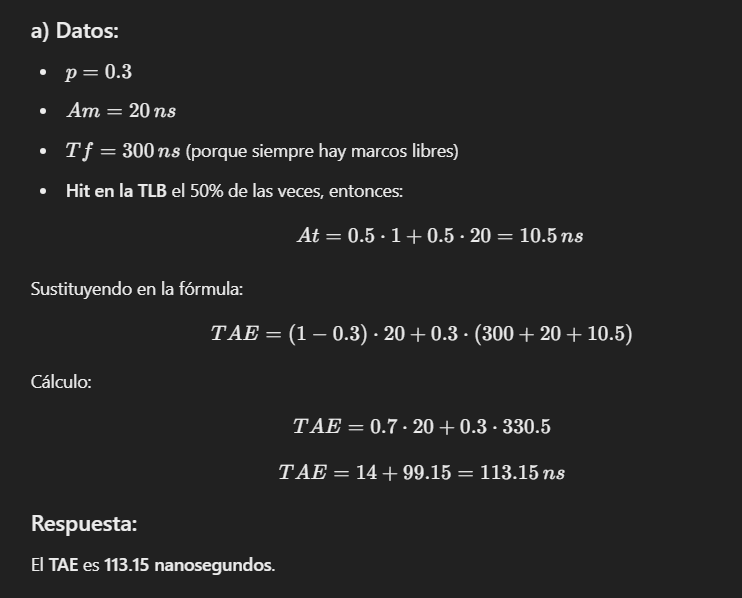
**d) Instalar mas memoria principal**

Es la mejor solución para este problema, mas memoria principal permite asignar mas marcos a cada proceso, por lo que se reduciría el numero de fallos por pagina, disminuyendo el uso del dispositivo de paginación. Ademas liberaría a la CPU de estar inactiva mientras espera que los fallos de pagina sean resueltos.

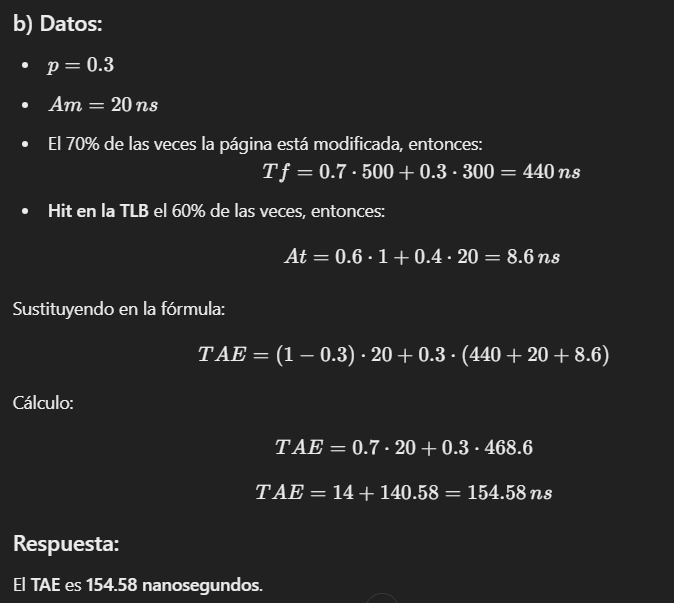
**e) Decrementar el quantum para cada proceso**

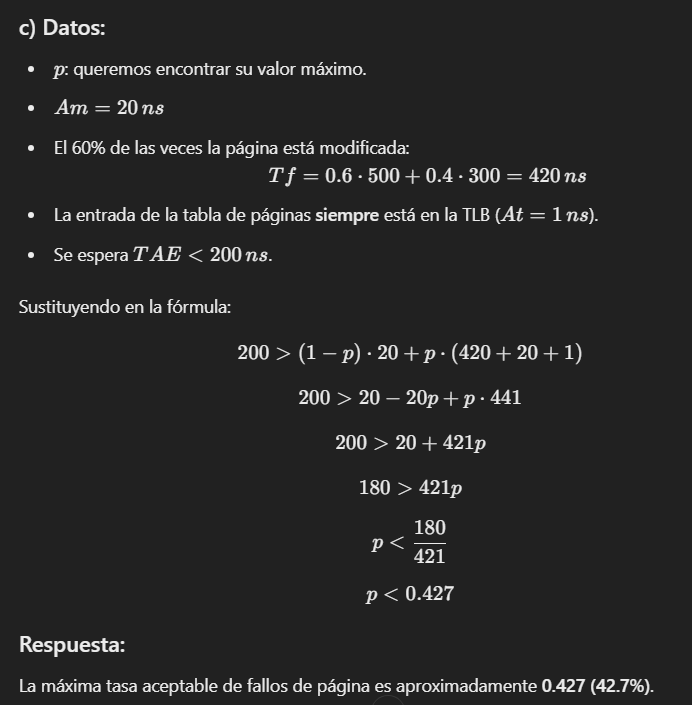
Reducir el quantum aumenta la frecuencia de los cambios de contexto. Por ende podría agravar la situación si los procesos necesitan mas paginas cada vez que se reanudan.

28 a)



b)





Puntos que no tengo ni idea como resolverlos porque no explican como encararlos ☺

Eje23: La parte donde terminan los procesos y queda espacio vacio en el grafico.

Eje24: No entiendo lo que pasa cuando hay paginas modificadas y marcos asignados a descarga asincrónica.