

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

- **Dirección Lógica o Virtual**
- **Dirección Física**
- Dirección Lógica:
 - Es una dirección que enmascara o abstrae una dirección física
 - Referencia a una localidad en memoria
 - Se la debe traducir a una dirección física
 - La CPU trabaja con direcciones logicas(virtual), que deben ser decodificadas a memoria fisica, para ello existe un Hardware llamado MMU. De esta manera no se pierde tiempo de cpu en realizar esa tarea.
- Dirección Física:
 - Es la dirección real. Es con la que se accede efectivamente a memoria
 - Representa la dirección absoluta en memoria principal

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.

TODAS ESTAS LAS VIMOS CON BERTONE

- Particiones Fijas (Causan fragmentación interna):
 - La memoria se divide en particiones o regiones de tamaño fijo
→ tamaños iguales o diferentes
 - Alojan un único proceso
 - Cada proceso se coloca en alguna partición de acuerdo a algún criterio:
 - First Fit (primer ajuste): Tenemos una lista de particiones, la cual recorreremos hasta encontrar el primer lugar donde nuestro proceso entre.
 - Best Fit (Mejor ajuste): Tenemos una lista de particiones en la cual tenemos espacios vacíos. Nuestro requerimiento se colocará en el espacio vacío que menos fragmentación cause.
 - Worst Fit (peor ajuste): Inverso al anterior. Se pone en el lugar mas grande que haya. Generando así un espacio amplio desperdiciado.
 - Next Fit : no lo explican en la explicación
- Particiones Dinámicas:
 - Las particiones varían en tamaño y número. Causando fragmentación EXTERNA.
 - Alojan un proceso cada una. (1 único proceso para cada Partición)
 - Cada partición se genera en forma dinámica del tamaño justo

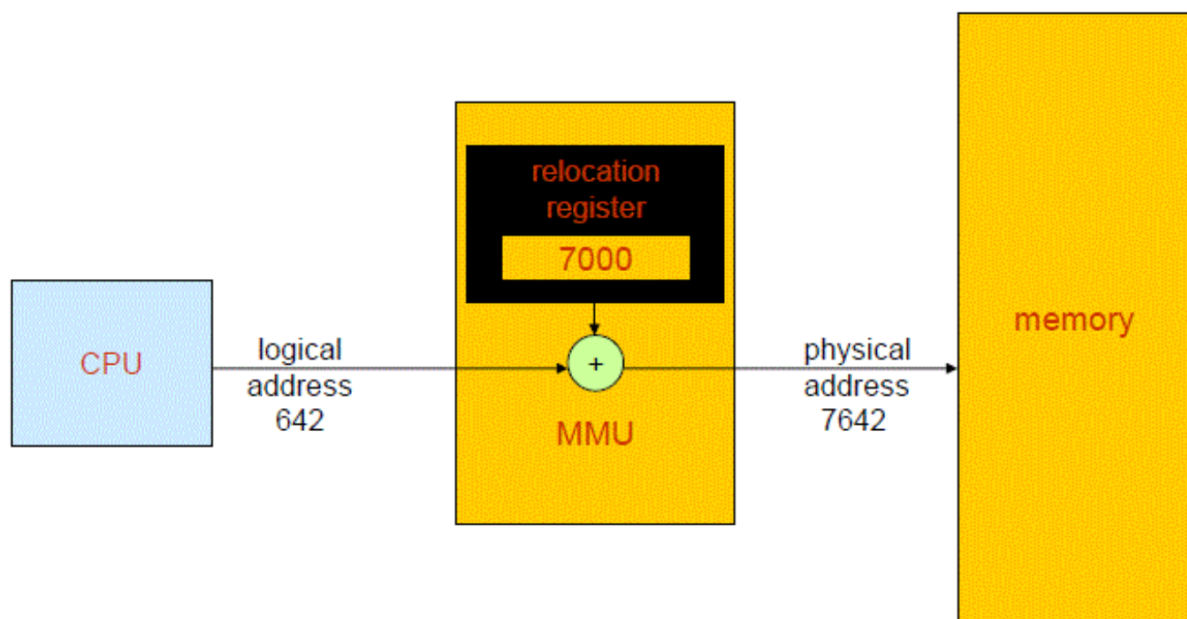
que necesita el proceso. Cuando se libera la memoria el espacio queda hueco pudiendo aprovechar por otro reasignándole el espacio disponible, generando fragmentación Externa.

- Si se liberan dos espacios de memoria contiguos se pueden unir generando una mayor. Si se quiera aprovechar realmente la memoria, habría que generar una compactación de memoria para unir todos los espacios libres y hacer uno mayor. ESTO ES MUY COSTOSO.

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

El Sistema Operativo (SO) necesita información sobre el tamaño de las particiones y su estado para administrar la memoria en los métodos de Particiones Fijas. Para Particiones Dinámicas, el SO debe mantener una lista de particiones disponibles, una tabla de asignación de particiones, información sobre fragmentación y políticas de asignación y liberación. En ambos casos, esta información es esencial para gestionar eficazmente la memoria y asignar procesos a las particiones correspondientes.

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:

Ventajas:

Simplicidad: Es más fácil de implementar, ya que todas las particiones tienen el mismo tamaño.

Fragmentación interna reducida: La fragmentación interna (espacio no utilizado dentro de una partición) se reduce al mínimo, ya que cada partición es del mismo tamaño.

Desventajas:

Fragmentación externa: Puede haber fragmentación externa, ya que si un proceso no encaja en una partición, esa partición se desperdicia y no se puede utilizar por otros procesos.

Ineficiencia en el uso de la memoria: Puede llevar a un uso ineficiente de la memoria, ya que los procesos pequeños ocupan el mismo tamaño de partición que los procesos grandes.

Particiones de diferente tamaño:

Ventajas:

Mayor flexibilidad: Permite acomodar procesos de diversos tamaños de manera más eficiente.

Reducción de fragmentación externa: Al asignar particiones de tamaño adecuado para cada proceso, se reduce la fragmentación externa en comparación con particiones de igual tamaño.

Desventajas:

Implementación más compleja: Requiere un algoritmo de asignación de particiones más sofisticado para elegir la partición adecuada para un proceso entrante.

Posible fragmentación interna: Aunque reduce la fragmentación externa, puede dar lugar a fragmentación interna si un proceso no ocupa toda la partición asignada.

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

- La fragmentación se produce cuando una localidad de memoria no puede ser utilizada por no encontrarse en forma contigua

- Fragmentación Interna:

- Se produce en el esquema de particiones fijas, por ejemplo
- Es interna a la localidad asignada
- Es la porción de la localidad que queda sin utilizar

- Fragmentación Externa:

- Se produce en el esquema de particiones dinámicas, por ejemplo
- son 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
- Solución → compactación → muy costosa

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

Compactación de Memoria:

La compactación de memoria es una técnica que consiste en reorganizar las particiones de memoria para eliminar la fragmentación externa. Esto se hace moviendo los procesos y sus datos para que estén contiguos en la memoria. La compactación sólo es posible en sistemas con particiones dinámicas y requiere la reubicación de procesos. Aunque es efectiva, puede ser costosa en términos de tiempo y recursos, ya que implica mover datos y procesos en la memoria.

Paginación:

La paginación es una técnica de gestión de memoria que divide la memoria física en marcos de tamaño fijo y divide los procesos en páginas del mismo tamaño. Las páginas no necesitan estar contiguas en memoria física, lo que evita la fragmentación externa. Cuando un proceso necesita memoria, el sistema operativo asigna páginas no contiguas según la disponibilidad. Esto reduce la fragmentación externa, pero introduce una pequeña fragmentación interna debido a que el último marco de una página puede no estar completamente lleno.

5.- Paginación

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

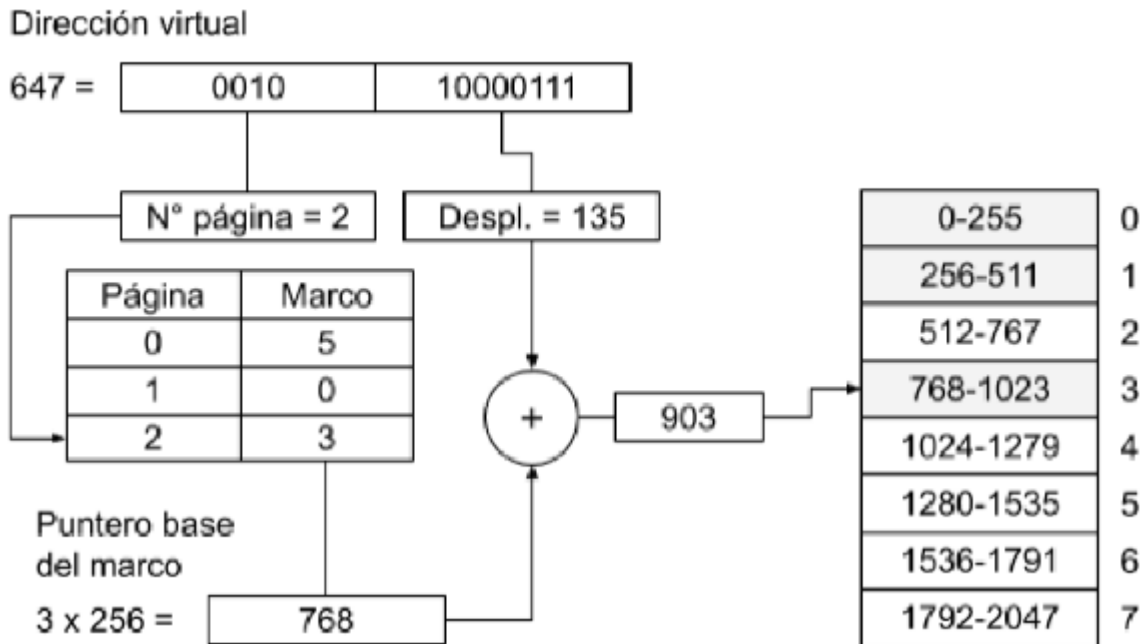
En la paginación, la memoria se divide en pequeños bloques de tamaño fijo llamados "marcos"(frames), y los programas y datos se dividen en unidades del mismo tamaño, llamadas "páginas".

El sistema operativo mantiene una tabla de páginas que mapea las páginas virtuales utilizadas por un proceso a las ubicaciones físicas de los marcos de página en memoria.

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

El sistema operativo debe mantener una tabla de páginas por cada proceso (guardado en la PCB), donde cada entrada de la tabla tiene el marco donde va dicha página.

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



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

Es posible que se produzca fragmentación interna pero no tan agresiva como en particiones fijas. El programa o proceso al ser dividido en páginas de igual tamaño a lo sumo se genera fragmentación interna en la última página (más grande que el resto que quedaba) asegurando que el resto calce justo.

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

Similitudes:

- Proporcionan protección de procesos.
- Pueden sufrir de fragmentación interna.
- Tamaño de unidades de asignación: La paginación utiliza páginas de tamaño fijo, y las particiones fijas tienen particiones de tamaño fijo.

Diferencias:

- Reubicación: La paginación facilita la reubicación de procesos, mientras que las particiones fijas pueden complicarla (Todo el proceso está en una partición, de forma continuada). La paginación permite la ubicación de las páginas en distintos lugares de la memoria (no adyacentes).
- Implementación de la tabla de páginas o de particiones: La paginación usa una tabla de páginas, mientras que las particiones fijas gestionan listas de particiones disponibles y asignadas.

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:

<i>Página</i>	<i>Marco</i>
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.

Para hacer este ejercicio primero debemos poner todos los marcos, asignarles a cada uno 512 bytes de espacio, y luego asignarle sus correspondientes páginas, en orden ascendente a las páginas le asignamos sus 512 bytes de direccionamiento virtual.

Memoria principal			
Marco	Página	Inicio-Fin virtual	Inicio-Fin físico
0	-		0-511
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	1536-2047	3072-3583

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

lógica -> físico:

i) 35

para saber a qué **PÁGINA** pertenece se le hace: $35 \div 512 = 0$ (marco 3, página 0)

El desplazamiento se realiza haciendo $35 \bmod 512 = 35$

la dirección física es = 35 (desplazamiento) + 1536 (inicio de físico) = 1571

ii) 512

página : $512 \div 512 = 1$;

desplazamiento : $512 \bmod 512 = 0$

dirección física = 0 (desplazamiento) + 2560 (inicio de pag 1) = 2560

iii) 2051

pagina = $2051 \div 512 = 4$ (La cual no existe en nuestro proceso)

No existe 2051 en las direcciones virtuales (Error).

iv) 0

pagina = $0 \div 512 = 0$

desplazamiento = $0 \bmod 512 = 0$

dir fisica = $0 + 1536 = 1536$

otra forma de hacerlo:

[$0 + (3 \cdot 512) = 1536$] donde 3 es el marco, 512 el tamaño del marco y 0 la dir virtual dada

v) 1325

pagina: $1325 \div 512 = 2$

desplazamiento: $1325 \bmod 512 = 301$

dir fisica = $301 + 1024 = 1325$

vi) 602

pagina: $602 \div 512 = 1$

desplazamiento: $602 \bmod 512 = 90$

dir fisica = $1024 + 90 = 1924$

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

Ahora es al revés, tenemos que fijarnos en qué dirección lógica se traducen las físicas, así que:

i) 509

para hallar el **Marco**: $509 \div 512 = 0$ (marco 0, X)

desplazamiento: $509 \bmod 512 = 509$

Error: en el marco 0 no hay paginas que corresponden al proceso

ii) 1500

marco: $1500 \div 512 = 2$ (marco 2, pagina 2)

desplazamiento: $1500 \bmod 512 = 476$

dir logica: $476 + 1024 = 1500$

iii) 0

marco : $0 \div 512 = 0$ (marco 0, X)

Error: en el marco 0 no hay paginas que corresponden al proceso

iv) 3215

marco: $3215 \div 512 = 6$ (marco 6, pagina 3)

desplazamiento: $3215 \bmod 512 = 143$

dir logica: $1536 + 143 = 1679$

v) 1024

marco: $1024 \div 512 = 2$ (marco 2, pagina 2)

desplazamiento : $1024 \bmod 512 = 0$

dir fisica: $0 + 1024 = 1024$

vi) 2000

marco: $2000 \div 512 = 3$ (marco 3, pagina 0)

desplazamiento: $2000 \bmod 512 = 464$

dir logica: $0 + 464 = 464$

[$2000 - 1536 = 464$]

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

2048 -> tamaño asignado.

2000 -> Tamaño del proceso.

2048 - 2000 = fragmentación interna 48 bytes.

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

TRANSFORMAR LOS NUMEROS A POTENCIA DE 2 y luego sumar.

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

para calcular esto debemos saber primero que:

$$8 = 2^3 = 3 \text{ bits}$$

$$1024 = 2^{10} = 10 \text{ bits}$$

Por lo tanto para poder realizar representar una memoria lógica hace falta **13 bits**.

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

El n de marcos -> $32 = 2^5 = 5$ bits para direccionamiento,

desplazamiento -> $1024 = 2^{10} = 10$ bits para direccionamiento.

El número de bits necesario es 15 bits

9.- Segmentación

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

La segmentación es una mejora de la paginación, nos permite dividir el programa en partes para evitar tener cargado todo un proceso en la memoria. Pudiendo dividir las partes de un programa (segmentos) que no sufren modificaciones (como códigos ejecutables) de aquellos segmentos que sufren de alguna modificación(aquellos que guardan variables de sistema). Esto nos hace que cuando carguemos o descarguemos un segmento de un programa actuemos de forma diferentes:

- Los segmentos sin modificar no hace falta guardar, solo se descargan.

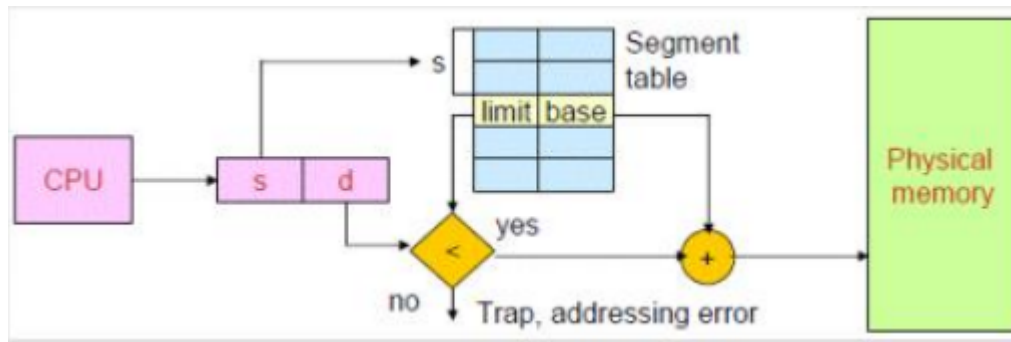
- Mientras que los segmentos modificados antes de descargarlos de la memoria debemos preservar los datos modificados, eso se logra usando la memoria SWAP que no deja de ser un espacio en nuestra memoria preservado para este fin.

Los segmentos están compuestos por páginas. Que a diferencia del proceso de paginación normal, en este no es necesario tener todo el programa en memoria.

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

Se debe tener una tabla de segmentos(que reemplaza la tabla de páginas), 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

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

- Similitudes:
 - Ambas pueden generar fragmentación externa
 - Ambas se basan en la idea de distribuir espacios de tamaño variable a los procesos en forma dinámica (según la necesidad particular)
- Diferencias:
 - El particionamiento dinámico requiere que todo el proceso se encuentre cargado de forma continua en memoria.
 - La segmentación subdivide y distribuye las divisiones de forma indiferente a su orden real, aunque sí mantiene la secuencia dentro de cada segmento

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

- Similitudes:
 - Ambas dividen un proceso en partes que serán almacenadas en memoria en lugares no contiguos, ni en orden.
 - Ambas requieren de tablas auxiliares que posibilitan su implementación (tabla de página y tabla de segmento)
- Diferencias:
 - La paginación transparente al programador.
 - La paginación elimina fragmentación externa. • La segmentación es visible al programador.
 - La segmentación facilita modularidad, estructuras de datos grandes y da mejor soporte a la compartición y protección

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 Páginas

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):

Para realizar esto hay que tener en cuenta que el número que nos dan corresponde a **(número de segmento, nro página, desplazamiento)**; comprendiendo esto deberemos sumar la dirección base (que corresponde al segmento en la tabla de segmentos) + dir base (que apunta el número de página de la tabla de páginas) + el desplazamiento dado.

i) **(2,1,1)** = 1500 (dirección donde inicia el segmento 2 en la tabla) + 20 (dir base apuntada por la pag 1) + 1 (desplazamiento dado) = 1521.

ii) **(1,3,15)** = 500 + 60 + 15 = 575

iii) **(3,1,10)** = 5000 + 120 + 10 = 5130

iv) **(2,3,5)** = 1500 + 0 + 5 = 1505

13.- Memoria Virtual

a) Describa qué 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 qué se debe apoyar el SO para su implementación?

En el hardware, pues 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. Aun así, el SO debe dar al hardware la información requerida para llevar a cabo su tarea.

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?

Se requiere como mínimo un bit que indique la presencia de la página en memoria, puesto 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, puesto 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?

Los fallos de página se producen cuando un proceso cargado en sistema requiera para ejecutarse una página que no se encuentra en memoria, para ello el sistema debe buscar, en memoria secundaria la página faltante para cargarla en sistema.

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

El encargado de detectar que hubo un fallo de página es el hardware (MMU), quien genera una interrupción. Al momento de generar la interrupción verifica la dirección de memoria en la tabla de páginas y analiza el bit V (1 = está en memoria, 0 = no está en memoria).

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

1. se genera la interrupción
2. El SO bloquea el 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 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")?

$2^{32} * 1 \text{ bytes} = 4.294.967.296 \text{ bytes} \rightarrow 4.294.967.296 \text{ bytes} / 1024 = 4194304 \text{ kB} \rightarrow 4194304 \text{ kB} / 1000 = 4096 \text{ mB} = 4 \text{ GB}$

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

En este caso lo que debemos hacer es dividir el total por 512 (el espacio que consume cada página).

$4194304 \text{ kB} / 512 \text{ kB} = 8192$ páginas totales x 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?

1- convertimos el mB en kB (*1024).

$256 \text{ mB} * 1024 = 262.144 \text{ kb}$

2- Luego se asigna al total dividido el tamaño de cada pagina.

$262.144 \text{ kB} / 512 \text{ kB} = 512$ marcos en toda la memoria

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?

Dado que 8192 es la cantidad máxima de páginas por proceso, entonces habría 8192 entradas máximas en la página, y cada uno pesa 2 kB:

2kB por entrada de la tabla.

$2\text{kB} * 8192 = 16384 \text{ kB} = 16\text{mB}$ por cada tabla

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

Tabla de páginas

- Cada proceso tiene su tabla de páginas y el tamaño de la tabla de páginas dependerá del tamaño de espacio de direcciones virtuales, y el tamaño de las páginas
ej: $32 \text{ entradas de páginas} * 512\text{kB c/u} = 16384 \text{ kB espacio de direcciones}$

- Puede alcanzar un tamaño muy grande tanto que no puedan entrar en memoria. para ello se dividen estas tablas de acuerdo a niveles.

- Formas de organizar:

- Tabla de 1 nivel: tabla única lineal

- Tabla de 2 niveles: o más, multinivel

- Tabla invertida: Hashing

Las **tablas de página de 1 nivel** funcionan como el sistema de paginación sin memoria virtual. Todas las páginas se encuentran en memoria, y las direcciones se resuelven asociando una parte de la dirección virtual con un número de marco, siendo el valor restante el desplazamiento dentro de dicho marco. La entrada en la tabla de páginas indica si la página está presente en memoria, y el número de marco es lo que el hardware utiliza para resolver la base de la dirección física (y luego sumarle el desplazamiento).

Luego de mucho analisis llegamos a la conclusiones que:

1 - la ppt es una cagada mentirosa.

2 - Yo creo q no nos toman esto

3 - Las tablas de paginas de 1 nivel requieren q todas las paginas entren en la tabla, a la cual se le suma posterior mente el desplazamiento para convertirla en memoria fisica (ejercicios que hicimos antes). Teniendo en cuenta que el tamaño de la tabla va a ser proporcional al espacio requerido para cada pagina q compone el proceso, dando paso a los bits que la direccionan.

Dado que tenemos una dirección de la cual 20bits son direcciones a paginas (n^* de páginas), de que tamaño seria la tabla de paginas?

en cada marco entra 4 Kb, cada entrada de la tabla (referencias a las páginas) son 4 bytes , entonces por marco entran $4kb/4bytes = 1024 (2^{10})$ entradas.

Necesitamos entonces saber cuantos marcos necesitamos para referenciar 2^{20} páginas.

$2^{20} / 2^{10} = 2^{10} (1024)$ marcos.

Entonces para almacenar una tabla de tamaño completo necesitamos $1024 \text{ marcos} * 4kb (4096 \text{ bytes} = 2^{12})$ tamaño de marco= 4mb.

Las **tablas de página de 2 niveles**, funcionan de manera similar, pero agregan un nivel de indirección con tablas paginadas. El primer nivel es equivalente a las tablas de página de 1 nivel, aunque cuenta con menos entradas, ya que el segmento de dirección virtual que antes identificaba a una entrada en la tabla de páginas, se encuentra subdividido en dos. La primera parte de la dirección lógica indica una entrada en la tabla de páginas de primer nivel. Dicha entrada refiere a una tabla paginada que, de la misma manera que cualquier página, puede o no estar cargada en memoria. Si la tabla de segundo nivel está cargada en memoria, la entrada en la tabla de primer nivel tendrá un indicador de la base del marco en el que se encuentra la tabla de segundo nivel. La segunda parte de la dirección lógica se utiliza para identificar la entrada de página efectivamente del proceso que se está buscando, la cual contiene finalmente la dirección base del marco al que la dirección lógica apunta, a la cual se agregan los bits de desplazamiento para concretar la resolución de la dirección física

En este caso cabe aclarar que se busca una mejoría en cuanto al espacio requerido en la memoria ram para ejecutar la tabla de paginas. Esto se logra a partir de dividir la tabla original a otras mas chicas. Siendo la 1ra la que guarde la direccion que te lleva a la 2da y esta + un desplazamiento genera la direccion de memoria efectiva que estara en la memoria fisica.

Las **tablas de páginas invertidas** buscan solucionar el problema invirtiendo el proceso de conversión de direcciones (de ahí su nombre). La tabla de páginas invertida se encuentra siempre cargada en memoria, y contiene una entrada por cada marco de la memoria (en lugar de por cada página de cada proceso). El número de página de cada proceso se vincula con un marco a través de una función de hash relativamente sencilla (de otra forma los costos de procesamiento serían demasiado altos). Como varias páginas pueden dar como resultado de la función de hash un mismo valor, si un marco está asociado con más de una página, las entradas se encadenan, de modo que el punto de entrada siempre es la primera entrada en la tabla de páginas. La entrada de página, a su vez, contiene un identificador del proceso al que corresponde, a fin de poder ser reconocida de manera unívoca. Una vez resuelta la asociación entre página y tabla, se puede resolver la dirección física agregando el desplazamiento como en los demás métodos

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)

MARCO	PAGINA	MEM VIR	MEM FIS
0	5	1536-2047	0-511
1			512- 1023
2	3	1024-1535	1024 - 1535
3			1536 - 2047
4	0	0-511	2048 - 2559

5			2560 - 3071
6			3072 - 3583
7	1	512-1023	3584 - 4095

a) $1052 \text{ div } 512 = 2$ pagina

$1052 \text{ mod } 512 = 28$ desplazamiento

Error (PAge fault)

b) $2221 \text{ div } 512 = 4$

$2221 \text{ MOD } 512 = 173$

Error (PAge fault)

c) 5499

$5499 \text{ div } 512 = 10$

Error Direccion invalida. (no existe la pagina 10 en el proceso)

d) $3101 \text{ div } 512 = 6$

Error Direccion invalida. (no existe la pagina 6 en el proceso)

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.

Ventajas:

Fragmentación interna reducida.

Uso eficiente de la memoria.

Desventajas:

Mayor sobrecarga en las tablas de páginas.

Mayor tiempo de búsqueda.

Posible aumento de fallos de página.

Un tamaño de página grande.

Ventajas:

Menos sobrecarga en las tablas de páginas.

Menor sobrecarga del sistema.

Desventajas:

Fragmentación interna potencial.

Uso ineficiente de la memoria.

Mayor probabilidad de pérdida de datos en caso de fallo de página.

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.

- 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 → $m \div p$ (m = marcos, p = cant procesos)
 - Reparto proporcional: se asignan marco en base a la necesidad que tiene cada proceso → $V_p \cdot m / V_t$ (V_p = páginas que pide el proceso, m = marcos, V_t = total de páginas)
- 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, cuántos marcos le corresponden a cada proceso si se usa la técnica de Asignación Fija:

i) Reparto Equitativo

ii) Reparto Proporcional

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

i) 10 a cada uno.

ii)

1- $(15 \cdot 40) / 63 = 10$

2- $(20 \cdot 40) / 63 = 13$

3- $(20 \cdot 40) / 63 = 13$

4- $(8 \cdot 40) / 63 = 5$

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

El más eficiente es la asignación proporcional ya que no asigna marcos de más a procesos que nunca lo van a usar.

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 (Óptimo)

Segunda Chance

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

OPT -> Mejor pero no se puede implementar

LRU -> El menos recientemente usado

FIFO segunda chance -> *FIFO* pero mejor

FIFO -> Primero en entrar primero en salir

b) Analice su funcionamiento. ¿Cómo los implementaría?

OPT: no se puede

LRU: Cada página debe tener información del instante de su última referencia.

FIFO: La página más vieja en la memoria es reemplazada.

FIFO con segunda chance:

- Cuando la página se carga en memoria, el bit R se pone a 0
- Cuando la página es referenciada el bit R se pone en 1
- La víctima se busca en orden *FIFO*. Se selecciona la primer página cuyo bit R está en 0
- Mientras se busca la víctima cada bit R que tiene el valor 1, se cambia a 0

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

1-Escribir la Página Modificada al Almacenamiento Secundario.

2-Actualización del Mapa de Páginas o Tabla de Páginas.

3-Marcado de la Página como No Modificada.

4-Liberación de Recursos Asociados.

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.

El reemplazo local solamente permite que un proceso reemplace sus propias páginas, no cambiando la cantidad de frames asignados al mismo. Esto permite determinar la tasa de page faults de cada proceso. También produce que un proceso tenga frames asignados sin usar que no son reemplazados por otros procesos.

El reemplazo global permite que se reemplace una página de cualquier proceso. Los procesos pueden entonces tomar frames de otros y aumentar su cantidad. No se puede medir los page-faults.

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

No, son ideas contrarias ya que si un proceso toma un marco de otro se rompe con la idea de que cada proceso tiene unos marcos predefinidos.

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 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 de página requiere de 0,1 seg. Calcular el tiempo consumido por atención a los fallos de páginas para los algoritmos de a).