

Practica 5 - ISO

- 1) **Dirección lógica o virtual:** Es una dirección que enmascara o abstrae una dirección física, referencia a una localidad en memoria, por lo que se debe traducir a una dirección física.

Dirección Física: Es la dirección real, es con la que se accede efectivamente a Memoria, Representa la dirección absoluta en memoria principal.

- 2) **a) Particiones fijas:** La memoria se divide en particiones o regiones de tamaño Fijo (pueden ser todas del mismo tamaño o no), alojan un proceso cada una y cada uno de estos procesos se coloca dependiendo del criterio de colocación (First fit, best fit, worst fit, next fit).

Ventajas: no hay que tener en cuenta el manejo de tamaños, por lo que la performance de ubicación de un proceso en memoria no se ve afectada, hacer un First fit no es costoso.

Desventajas: Se puede producir fragmentación interna, que es el espacio de la partición que no se usa ya que el proceso es mas chico que si tamaño.

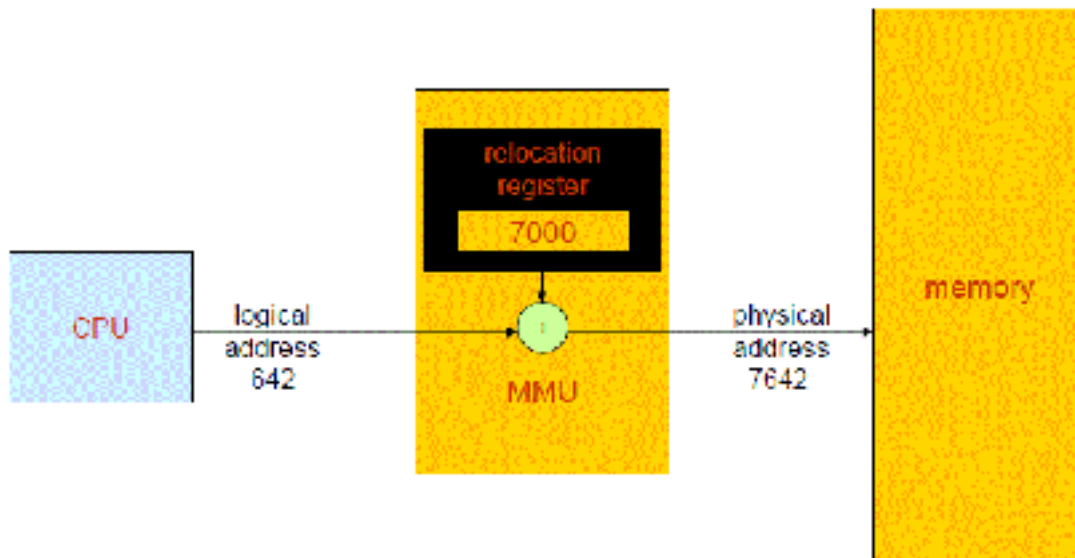
Particiones dinámicas: Las particiones varían en tamaño y en numero, también alojan un proceso cada una y cada partición se genera de forma dinámica del tamaño justo del proceso.

Ventajas: No hay fragmentación interna.

Desventajas: Es costoso el manejo de ubicación de procesos, además, podría presentarse fragmentación externa, que es el espacio de memoria entre procesos que queda libre cuando estos finalizan y que al ser muy chicas no se puede utilizar para la ubicación de uno nuevo. Para solucionar este problema se podría utilizar la compactación, pero esta es muy costosa.

- b) El SO debe tener una tabla de paginas (las paginas son las porciones en que es dividida el espacio de direcciones de un proceso) la cual contiene el marco (porción en la que es dividida la memoria) donde se encuentra la pagina.

c)



- 3) **Particiones Fijas**

Ventajas: No hay necesidad de tener criterios de colocación ya que todos las particiones son del mismo tamaño, por lo que basta con encontrar alguna libre para almacenar el proceso.

Desventajas: Si entra un proceso de tamaño mas grande que el de a partición, no se puede cargar en memoria.

Particiones Dinámica

Ventajas: Se puede aprovechar mejor el espacio en memoria al darle a un proceso chico una partición mas pequeña (se reduce la fragmentación interna).

Desventajas: Se debe tener un algoritmo que analice los tamaños y verifique que partición es mas conveniente para cada archivo, por lo que la performance se ve degradada y se produce fragmentación externa.

4) a) Fragmentación interna:

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

Fragmentación Externa:

- Se produce en el formato de particiones dinámicas.
- Son huecos que van quedando en la memoria a medida que los procesos finalizan.
- Al no encontrarse contigua puede darse el caso de que tengamos memoria libre para ubicar el proceso pero que no la podamos utilizar.

b) Para remediar el problema de la fragmentación externa esta la técnica de compactación, que junta todos los procesos de forma contigua para que el espacio libre pequeño se junte con los grandes y sea utilizable para futuros procesos. Sin embargo, esta técnica es muy costosa, ya que debe mover los espacios de direcciones de los procesos que estén generando la fragmentación.

5) a) Paginación

- La memoria se divide en porciones de igual tamaño llamadas marco
- El espacio de direcciones de los procesos se divide en porciones de igual tamaño denominadas páginas.
- El tamaño de la página es igual al tamaño del marco (suele ser de 512 bytes)
- El sistema operativo mantiene una tabla de páginas para cada proceso, la cual contiene el marco donde se encuentra cada pagina.
- La paginación bajo demanda es una técnica eficiente de manejar esta estrategia.

b) Como estructura adicional el Sistema Operativo necesita una tabla de páginas.

c) Lógica -> Física

Página	Marco		Marco	Inicio-Fin
0	1		0	0 - 511
1	2		1	512 - 1023
2	3		2	1024 - 1535
3	0		3	1536 - 2047

F.L. de 48 B.

Si tenemos una dirección lógica, por ejemplo 580:

- Para averiguar el número de página hacemos $580 \text{ DIV } 512 = 1$.
- Luego esta dirección corresponde a la página 1 que se encuentra en el marco 2
- Para averiguar el desplazamiento hacemos $580 \text{ MOD } 512 = 68$
- La dirección física es $1024 + 68 = 1092$

Lógica -> Física

Página	Marco		Marco	Inicio-Fin
0	1		0	0 - 511
1	2	←	1	512 - 1023
2	3		2	1024 - 1535
3	0		3	1536 - 2047

El. de 48 B.

Si tenemos una dirección física, por ejemplo 1092:

- Para averiguar el número de marco hacemos $1092 \text{ DIV } 512 = 2$.
- Luego en el marco numero 2 tenemos a pagina numero 1.
- Para averiguar el desplazamiento hacemos $1092 \text{ MOD } 512 = 68$.
- La dirección lógica es $512 + 68 = 580$.

d) No genera fragmentación externa ya que la memoria esta dividida en marcos de igual tamaño, por lo que no es posible que no haya espacio sin utilizar entre ellos. Puede generar fragmentación interna en la ultima pagina del proceso que este cargado ya que puede utilizar menos espacio de la totalidad del marco.

6) Las similitudes entre la paginación y las particiones fijas son que ambas tienden a generar fragmentación interna, las porciones de memoria son todas de igual tamaño, y la asignación de información se da de forma simple. Por otra parte, se diferencian en que la paginación requiere de una estructura adicional denominada tabla de páginas.

7) Practica

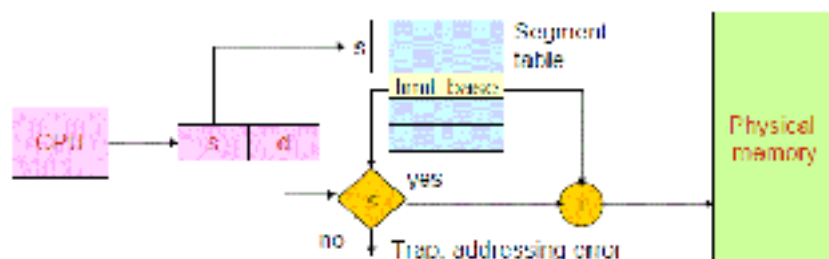
8) Practica

9) a) Segmentación

- Se puede ver como una mejora de la paginación. No hay fragmentación interna, sino externa.
- La tabla de segmentos, ademas de tener la dirección de inicio del mismo, tiene la longitud o el limite.
- Las direcciones lógicas constan de dos partes: un numero de segmento S y un desplazamiento D dentro del segmento ($0 < D < \text{Limite}$)

b) Como estructura adicional el Sistema Operativo necesita una tabla de segmentos.

c)



d) No genera fragmentación interna ya que cada proceso tiene el tamaño del segmento. Puede generar fragmentación externa porque al terminar un proceso y agregar un nuevo segmento para un nuevo proceso, ocurra que haya espacio entre este segmento y el siguiente.

10) Las Similitudes entre segmentación y particiones dinámicas es que en ambas el tamaño se ajusta al tamaño del proceso, generan fragmentación externa y no generan interna. Por otra parte se diferencian en que la segmentación requiere de una estructura adicional denominada tabla de segmentos, la cual posee a su vez un limite del cual la dirección no puede sobrepasarse; mientras que la dinámica no posee esa estructura adicional, por lo que no se realiza ese control.

11) Las similitudes entre paginación y segmentación son que ambas constan de particiones en memoria, además de la dirección base a la cual se le aplica el desplazamiento para encontrar el dato al que se quiere acceder y ambas requieren de una estructura adicional. Por otra parte, se diferencian en que la paginación genera fragmentación interna y no externa, mientras que la segmentación genera fragmentación externa y no interna.

12) Practica

13) Memoria Virtual

a) El mayor beneficio que tiene es el manejo apropiado de paginas en memoria para evitar los fallos de paginas (o por lo menos la gran mayoría).

b) En el comportamiento pasado para predecir que paginas son convenientes tener en memoria porque es mas probable que vuelvan a necesitarse.

c) Se requiere un bit de referencia para saber cuales paginas fueron referenciadas y cuales no.

14) Fallo de Paginas (Page Faults)

a) Cuando una pagina que un proceso necesita no esta en memoria, por lo que tiene que irse a buscar a memoria secundaria y pasarla a memoria principal para que este disponible.

b) Quien detecta el fallo de pagina es el Hardware, y el responsable de atenderlo es el Sistema Operativo.

c) Las acciones que realiza el Sistema Operativo cuando ocurre un fallo de pagina son:

- Reservar uno o varios marcos para la descarga asincrónica de paginas

- Cuando es necesario descargar la pagina modificada:

 - 1 - La pagina que provoco el fallo se coloca en un frame de descarga asincrónica.

 - 2 - El SO envía la orden de descargar asincrónicamente la pagina modificada mientras continua la ejecución de otro proceso.

 - 3 - El frame de descarga asincrónica pasa a ser el que contenía a la pagina víctima que ya se descargo correctamente.

15) a) Se pueden referenciar 2^{32} direcciones, si cada una referencia un byte, se obtienen $2^{32} \times 1 \text{ byte} = 4\text{gb}$, por lo que el tamaño máximo de un proceso seria de 4gb, aunque va a depender de la arquitectura del Sistema Operativo.

b) $(2^{32} \text{ direcciones}) / (512 * 1024) = 8.192 \text{ páginas}$

Tengo 2^{32} direcciones.

Cada pagina ocupa 512kb que es igual a $512 * 1024 = 524.288 \text{ bytes}$

c) *0,5 Mb ocupa una pagina*

256 Mb de memoria real

podría tener 128 marcos

d) El tamaño máximo de la tabla de páginas de cada proceso es 2^{14} .

16) tabla de paginas de nivel 1: Contiene la tabla de paginas del proceso.

tabla de paginas de nivel 2: Los primeros 10 bits direccionan a la primer tabla de paginas, los segundos 10 bits juntos con el resultado de la primer tabla te direcciona a la 2 tabla, con la cual obtenemos el frame y junto con los 12 bits restantes se obtiene la dirección física.

tabla de paginas invertida: Hay una entrada por cada frame, hay una sola tabla para todo el sistema. El numero de pagina es transformado en un valor de hash, y este se usa como índice para encontrar el marco asociado.

17) Practica

18) tamaño de paginas pequeño: Hay menor fragmentación interna, sin embargo, se requieren mas paginas para el proceso, por ende las tablas de paginas se tornan mas grandes. Ademas de esta forma mas paginas pueden residir en memoria.

Tamaño de paginas grandes: Mayor fragmentación interna, sin embargo la memoria secundaria esta diseñada para transferir grandes bloques de datos, por lo que la performance en transferencia de datos se ve optimizada (es mas rápido mover paginas hacia la memoria principal).

19) a) Asignación fija: A cada proceso se la asigna una cantidad arbitraria de marcos. A su vez para el reparto se puede usar reparto equitativo (se asigna la misma cantidad de marcos a cada proceso, $M \div P$) y reparto proporcional (se asignan marcos a la necesidad que tiene cada proceso).

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

b) i) Reparto equitativo: Serian $40 \text{ marcos} / 4 \text{ procesos} = 10 \text{ marcos}$ para cada proceso.

ii) Reparto Proporcional: $M = 40$ $S1 = 15$ $S2 = 20$ $S3 = 20$ $S4 = 8$ $S = 63$

Proceso S1 = $15 * 40 / 63 = 9$ marcos

Proceso S2 = $20 * 40 / 63 = 12$ marcos

Proceso S3 = $20 * 40 / 63 = 12$ marcos

Proceso S4 = $8 * 40 / 63 = 5$ marcos

c) Ninguno de los dos repartos

20) Opt: Algoritmo optimo, saca a la pagina que no se va a referenciar en un futuro, es imposible de implementar ya que no se conocen los futuros eventos.

LRU (least recently used/ ultima recientemente usada): Saca de la memoria a la pagina que hace mayor tiempo que no se referencia. Cada pagina debe contener información del instante de su ultima referencia.

FIFO (First in first out/ primero en entrar, primero en salir): Saca a la pagina mas vieja, es una cola circular. Muy fácil de implementar pero genera muchos fallos de paginas.

FIFO con segunda chance: Utiliza el bit de referencia R, por lo que saca la pagina mas vieja que no haya sido referenciada recientemente.

FIFO Segunda Chance																															
pa	2	4	1	3	1	1	2	6	2	4	2	4	1	4	8	1	8	6	1	4	1	5	1	4	3	1	8	7	9	4	
pr	b	b	a	a	a	c	c	b	b	b	a	a	a	c	c	b	b	c	c	c	c	a	a	a	b	b	b	a	a	a	
f1	2 b	2 b	2 b	2 b	2 b	2 b	2 b	2 b	2 b*	2 b*	2 b	2 b	2 b	2 b	1 b	1 b	1 b	1 b	1 b	1 b	1 b	1 b	1 b	4 a	4 a	4 a	4 a	7a	7 a	7 a	
f2		4 b	4 b	4 b	4 b	4 b	4 b	4 b	4 b	4 b*	4 b	4 b	4 b	4 b	4 b	8 b	8 b	8 b	8 b	8 b	8 b	8 b	8 b	8 b	3 b	3 b	3 b	3 b	3b	9 a	9 a
f3			1 a	1 a	1 a*	1 a	1 a	1 a	1 a	1 a	1 a	1 a*	1 a*	1 a*	1 a*	1 a*	1 a	1 a	1 a	1 a	1 a	1 a	1 a*	1 a*	1 a*	1 a	1 a	1a	1 a	4 a	
f4				3 a	3 a	3 a	3 a	3 a	3 a	3 a	2 a	2 a	2 a	2 a	2 a	2 a	2 a	6 c	6 c	6 c	6c	6 c	6c	6 c	6 c	1 b	1 b	1b	1 b	1 b	
f5						1 c	1 c	1 c	1 c	1 c	1 c	4 a	4 a	4 a	4 a	4 a	4 a	4 a	1 c	1 c	1c *	1 c*	1c *	1 c*	1 c*	1 c*	1 c*	1c	1 c	1 c	
f6							2 c	2 c	2 c	2 c	2 c	2 c	2 c	4 c	4 c	4 c	4 c	4 c	4 c	4 c	4 c*	4c *	4 c*	4c *	4 c	4 c	4 c	8 b	8b	8 b	8 b
f7								6 b	6 b	6 b	6 b	6 b	6 b	6 b	8 c	8 c	8 c	8 c	8 c	8 c	8 c	8c	5 a	5 a	5 a	5 a	5 a	5 a	5a	5 a	5 a
PF	X	X	X	X		X	X	X			X	X		X	X	X	X	X	X			X		X	X	X	X	X	X	X	X

24)

25) a) Un sistema esta en hiperpaginacion cuando pasa mas tiempo paginando que ejecutando procesos.

b) El motivo que causa el trashing es que cada vez que el SO busca una pagina en memoria, esta se encuentra en disco, y debe ser swapeada. Esto puede suceder cuando se posee poca memoria RAM o el working set de los procesos es muy grande.

c)

d) El sistema operativo intenta adivinar, en función de la historia reciente, qué fragmentos se usarán con menor probabilidad en un futuro próximo.

26) a) En el caso a se utiliza mucho tiempo en la paginación lo cual seria un típico caso de *trashing*, en el caso b y c son mas óptimos en términos de tiempo empleado en paginación, pero en el c la CPU permanece ociosa mas tiempo.

b) Asumiendo que el tiempo de CPU que no se indica permanece ociosa, se podría aumentar la multiprogramación y así aumentar el uso de la CPU en el caso b y c.

c) En el caso a no estaría siendo útil la paginación debido a que la CPa0U tiene poco tiempo de uso y no es optimo. En los casos b seria útil la paginación ya que aumenta el timepo de CPU.

27)

28) $TAE = At + (1-p) \cdot Am + p \cdot (Tf + Am)$

Donde:

TAE = tiempo de acceso efectivo

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

Am = tiempo de acceso a la memoria real

Tf = tiempo se atención de una fallo de pagina

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

a) $p = 0.3$ $Am = 20$ nanosegundos $Tf = 300$ nanosegundos $At =$

29) a) La anomalía de Belady es un efecto por el cual es posible tener más fallos de página al aumentar el número de marcos en la memoria física utilizando el método FIFO como algoritmo de reemplazo de páginas en sistemas de gestión de memoria virtual con paginación.

b) https://es.wikipedia.org/wiki/Anomal%C3%ADa_de_Belady

30) a) Habra 1.024 fallos de pagina.

b) Si, se podría modificar cambiando de orden los indices, en la instrucción $C[i; j] = A[i; j] + B[i; j]$; por la instrucción $C[j; i] = A[j; i] + B[j; i]$; reduciendo los fallos de pagina a 16 fallos de pagina.

31) a) Proceso A:

Proceso B:

b) Proceso A:

Proceso B:

c) m = cantidad de frames disponibles

0	1	2	3	4
4	5	6		

D = demanda total de frames

WSS_i = medida del working set del proceso_i

0	1	2	3	4
2	3	5	6	

$\sum WSS_i = D$

Si $D > m$ habra thrashing

0	1	2	3	4
1	2	3	4	5

No habría trashing para 8 frases pero si para 6 frames.

d) La tasa de fallos de pagina en el instante 38 seria 3.

0	1	2	3	4
3	4	5		

e) Al obtener en el inciso d) una tasa de fallos igual a 3 y el SO utiliza como limite inferior 2 y superior 5, no debería tomar ninguna acción respecto a la cantidad de marcos asignados.