

Administración de la Memoria



Integrantes:
Anyelys Escalona
Cesar Manrique
Bárbara Lozada
Ricardo Quero
Fabiola Alvarado

Paginación

Fallo de Página

Espacio de
dirección virtual

60K-64K

X

56K-60K

X

52K-56K

X

48K-52K

X

44K-48K

7

40K-44K

X

36K-40K

5

32K-36K

X

28K-32K

X

24K-28K

X

20K-24K

3

16K-20K

4

12K-16K

0

8K-12K

6

4K-8K

1

0K-4K

2

} Página virtual

Dirección
de memoria
física

28K-32K

24K-28K

20K-24K

16K-20K

12K-16K

8K-12K

4K-8K

0K-4K

Marco de página

Cada página empieza en un múltiplo de 4096 y termina 4095 direcciones más arriba, por lo que de 4 K a 8 K en realidad significa de 4096 a 8191 y de 8 K a 12 K significa de 8192 a 12287.

Paginación

Fallo de Página

Espacio de
dirección virtual

60K-64K
56K-60K
52K-56K
48K-52K
44K-48K
40K-44K
36K-40K
32K-36K
28K-32K
24K-28K
20K-24K
16K-20K
12K-16K
8K-12K
4K-8K
0K-4K

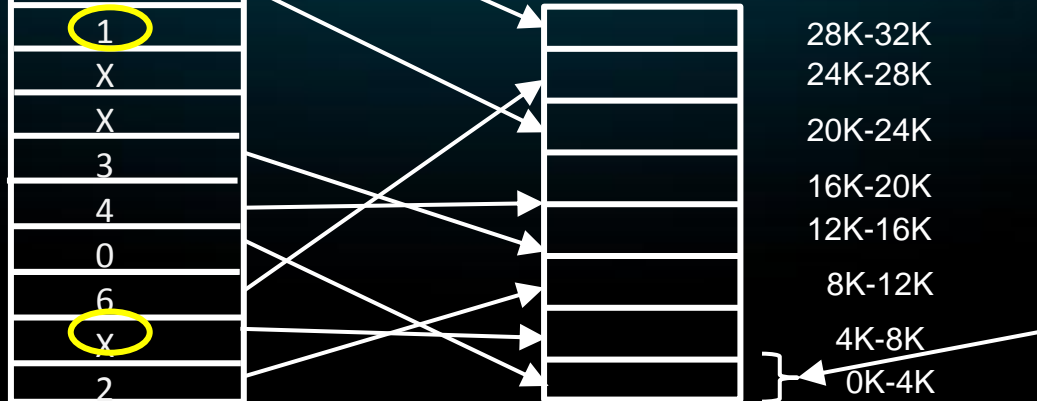
X
X
X
X
7
X
5
1
X
X
3
4
0
6
X
2

} Página virtual

Dirección
de memoria
física

28K-32K
24K-28K
20K-24K
16K-20K
12K-16K
8K-12K
4K-8K
0K-4K

Marco de página



Manejo de Fallos de Página

¿Qué ocurre en un fallo de página?

1. El Hardware salta al kernel.
1. Se inicia una rutina en código ensamblador.
1. El SO descubre que ha ocurrido un fallo de página.
1. Una vez que se conoce la dirección virtual que produjo el fallo, el sistema comprueba si esta dirección es válida y si la protección es consistente con el acceso.
1. Si el marco de página seleccionado está sucio, la página se planifica para transferirla al disco.
1. El sistema operativo busca la dirección de disco en donde se encuentra la página necesaria.

Manejo de Fallos de Página

¿Qué ocurre en un fallo de página?

7. Cuando la interrupción de disco indica que la página ha llegado, las tablas de páginas se actualizan para reflejar su posición y el marco se marca como en estado normal.
7. La instrucción fallida se respalda al estado en que tenía cuando empezó.
7. El proceso fallido se planifica y el sistema operativo regresa a la rutina (en lenguaje ensamblador) que lo llamó.
7. Esta rutina recarga los registros y demás información de estado, regresando al espacio de usuario para continuar la ejecución, como si no hubiera ocurrido el fallo.

Algoritmos de Reemplazo de Páginas

Cuando ocurre un fallo de página, el sistema operativo tiene que elegir una página para desalojarla y hacer espacio para la página entrante.

- Si la página a eliminar se modificó mientras estaba en memoria, debe volver a escribirse en el disco para actualizar la copia del mismo.
- Si la página no se ha modificado (por ejemplo, si contiene el texto del programa), la copia ya está actualizada y no se necesita reescribir. La página que se va a leer sólo sobrescribe en la página que se va a desalojar.

Aunque sería posible elegir una página al azar para desalojarla en cada fallo de página, el rendimiento del sistema es mucho mayor si se selecciona una página que no sea de uso frecuente. Si se elimina una página de uso frecuente, tal vez tenga que traerse de vuelta rápidamente, lo cual produce una sobrecarga adicional.

Algoritmos de Reemplazo de Páginas

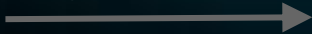
Algoritmo Óptimo

Un algoritmo óptimo debe generar el mínimo número de fallos de página.

Por ello, la página que se debe reemplazar es aquella que tardará más tiempo en volverse a usar.

Página 5	122, 1000, 1099,4409
----------	----------------------

Página entrante



Página 1	123, 1002, 2099,10000
----------	-----------------------

Página 2	11, 2000, 22444, 22443
----------	------------------------

Página 3	1000, 1001, 1002,1003
----------	-----------------------

Página 4	500, 1100, 11001,23000
----------	------------------------

Página a desalojar



Evidentemente, este algoritmo es irrealizable, ya que no se puede predecir cuáles serán las siguientes páginas accedidas.

El interés de este algoritmo es que sirve para comparar el rendimiento de otros algoritmos realizables.

Algoritmos de Reemplazo de Páginas

Algoritmo Óptimo

Se reemplaza la página 1 pues nunca se volverá a hacer referencia a ésta.

Se reemplaza la página 4 o la 3 pues nunca se volverá a hacer referencia a éstas.

Cadena de Referencias

2 3 2 1 5 2 4 5 3 2 5 2

2	2	2	2	2	2	4	4	4	2	2	2
	3	3	3	3	3	3	3	3	3	3	3
			1	5	5	5	5	5	5	5	5

Faltas de Página

F F F F F F F

Se reemplaza la página 2 dado que es la que tardará más tiempo en volverse a referenciar.

Algoritmos de Reemplazo de Páginas

Algoritmo LRU (*Least Recently Used*)

Las páginas que no se han utilizado durante siglos probablemente seguirán sin utilizarse durante mucho tiempo. Esta idea sugiere un algoritmo realizable: cuando se produzca una falta de página, sustituir la página que lleve más tiempo sin utilizarse.

Aunque LRU es teóricamente realizable, tiene un coste elevado. Para implementarlo fielmente es necesario mantener una lista enlazada de todas las páginas que están en la memoria, con la página que se utilizó más recientemente al principio y la página menos recientemente utilizada al final. La dificultad consiste en que la lista debe actualizarse en cada referencia a la memoria. Encontrar una página en la lista, sacarla de la lista y reinsertarla al frente es una operación muy lenta, incluso realizándose por hardware (suponiendo que pudiera construirse tal hardware).

Algoritmos de Reemplazo de Páginas

Algoritmo LRU (*Least Recently Used*)

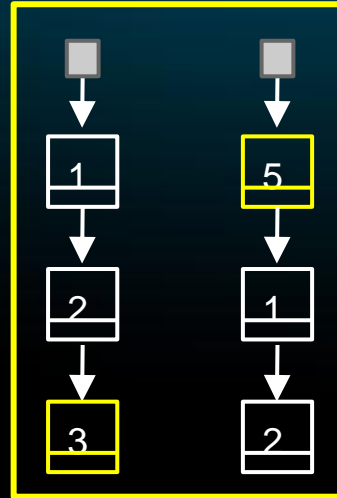
Cadena de Referencias

2	3	2	1	5	2	4	5	3	2	5	
2	3	2	1	5	2	4	5	3	2	5	2
	2	3	2	1	5	2	4	5	3	2	5
			3	2	1	5	2	4	5	3	3

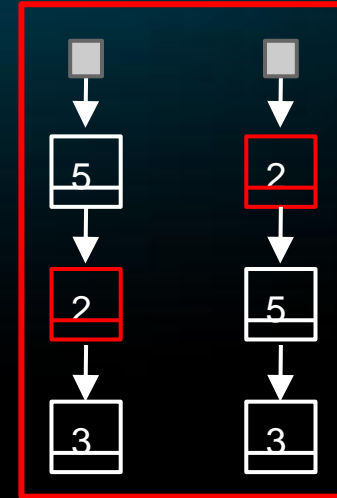
Faltas de Página

F F F F F F F

Reemplazo



Actualización



Algoritmos de Reemplazo de Páginas

Algoritmo FIFO (*First In, First Out*)

El sistema operativo mantiene una lista de todas las páginas que están actualmente en la memoria, con la más antigua al principio de la lista y la más nueva al final.

Al producirse una falta de página, se sustituye la página que está al principio de la lista y la nueva se añade al final. FIFO en algunos casos podría sustituir una página que no tiene uso frecuente, pero también es posible que sustituya páginas que son muy usadas a lo largo de la vida de un programa, por lo tanto, estas páginas se cargarán y expulsarán repetidas veces.

Por esta razón, raramente se utiliza FIFO en su forma pura.

Algoritmos de Reemplazo de Páginas

Algoritmo FIFO (*First In, First Out*)

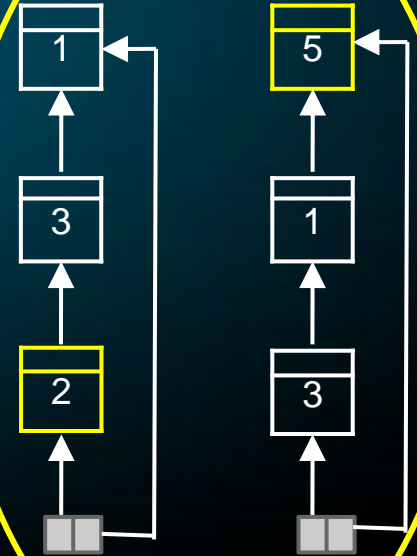
Cadena de Referencias

2	3	2	1	5	2	4	5	3	2	5	
2	3	3	1	5	2	4	4	3	3	5	2
	2	2	3	1	5	2	2	4	4	3	5
			2	3	1	5	5	2	2	4	3

Faltas de Página

F F F F F F F F F F

La página 5
reemplaza a
la página 2



Algoritmos de Reemplazo de Páginas

Algoritmo de la Segunda Oportunidad

Una modificación sencilla del algoritmo FIFO que evita el problema de sustituir una página que se usa mucho consiste en examinar el bit R de la página más antigua.

Si es 0, quiere decir que la página es antigua y además no se usa. Por lo tanto, se la sustituye de inmediato.

Si el bit R es 1, se pone a 0, se coloca la página al final de lista de páginas y su instante de carga se actualiza como si acabara de cargarse en la memoria. Luego se continúa la búsqueda.

Nota: Para hacer posible que el sistema operativo pueda recoger estadísticas útiles sobre qué páginas se están usando y cuáles no, casi todos los ordenadores con memoria virtual asocian a cada página dos bits de estado R (*Referenced* o Referenciada) y M (*Modified* o Modificada). R se activa cada vez que se referencia la página (para leer o escribir) y M se activa cada vez que se escribe en la página. Una vez activado un bit, conserva el valor 1 hasta que el sistema operativo lo desactiva a 0 por software.

Algoritmos de Reemplazo de Páginas

Algoritmo de la Segunda Oportunidad

Página
entrante:

Pág6	1
------	---

Pág1	1
------	---

Pág2	1
------	---

Pág3	0
------	---

Pág4	0
------	---

Pág5	0
------	---

Se examina la primera página.

Tiene su bit de uso en uno, por lo tanto se le da una segunda oportunidad. Se desactiva R y se pasa al final de la lista.

Pág2	1
------	---

Pág3	0
------	---

Pág4	0
------	---

Pág5	0
------	---

Pág1	0
------	---

Igual que en el primer caso.

Pág3	0
------	---

Pág4	1
------	---

Pág5	0
------	---

Pág1	0
------	---

Pág2	0
------	---

Al encontrarse una página con el bit R en 0, se escoge para el desalojo y se agrega la página entrante.

Pág4	1
------	---

Pág5	1
------	---

Pág1	0
------	---

Pág2	0
------	---

Pág6	0
------	---

Algoritmos de Reemplazo de Páginas

Algoritmo del Reloj (*Clock*)

Es una mejora a nivel de implementación del algoritmo de la segunda oportunidad.

Se basa en mantener todos los marcos de página sobre una lista circular en la forma de un reloj. Una manecilla apunta a la página más antigua.

Cuando se presenta una falta de página, se examina la página a la que apunta la manecilla. Si su bit R es 0, dicha página se sustituye, insertando la nueva en su lugar y adelantando una posición la manecilla. Si R es 1, se cambia a 0 y la manecilla se adelanta a la siguiente página. Este proceso se repite hasta hallar una página con $R=0$.

No es sorprendente que a este algoritmo se le llame el algoritmo del reloj.

Algoritmos de Reemplazo de Páginas

Algoritmo del Reloj (Clock)

Cadena de Referencias	2	3	2	1	5	2	4	5	3	2	5	2
	2	2	2*	→ 2*	2	2*	→ 2*	→ 2*	→ 2	→ 2*	→ 2*	→ 2*
	→	3	3	3	5	5	5	5*	5	5	5*	5*
Faltas de Página	F	F		F	F		F		F			

Pág1 | 0



Pág2 | 1

Si la página tiene el bit R en 1, lo pone en 0 y avanza la manecilla.

Pág3 | 0

Algoritmos de Reemplazo de Páginas

Algoritmo del Reloj (Clock)

Cadena de Referencias	2	3	2	1	5	2	4	5	3	2	5	2
	2	2	2*	→ 2*	2	2*	→ 2*	→ 2*	→ 2	→ 2*	→ 2*	→ 2*
	→	3	3	3	5	5	5	5*	5	5	5*	5*
		→	→	1	→ 1	→ 1	4	4	3	3	3	3
Faltas de Página	F	F		F	F		F		F			

Pág1 | 0

Pág2 | 0

●
↓
Pág5 | 0

Al conseguir una página con el bit R en 0 la sustituye.

Algoritmos de Reemplazo de Páginas

Algoritmo del Reloj (Clock)

Cadena de Referencias	2	3	2	1	5	2	4	5	3	2	5	2
	2	2	2*	→ 2*	2	2*	→ 2*	→ 2*	→ 2	→ 2*	→ 2*	→ 2*
	→	3	3	3	5	5	5	5*	5	5	5*	5*
		→	→	1	→ 1	→ 1	4	4	3	3	3	3
Faltas de Página	F	F		F	F		F		F			

La manecilla queda apuntando a la página 1.



Algoritmos de Reemplazo de Páginas

Algoritmo NRU (*Not Recently Used*)

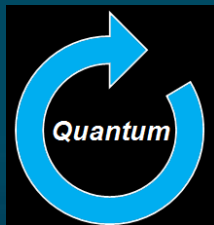
- En este método cada página contiene 2 bits que permite decidir cual página desalojar:
 - R (bit referenciado): Se enciende cuando se realiza un acceso a esta página
 - M (bit modificado): Se enciende cuando se escribe en la página.
- Se establecen así 4 clases de acuerdo al valor de los bits:
 - Clase 0: No referenciada, No Modificado
 - Clase 1: No referenciada, Modificado
 - Clase 2: Referenciada, No Modificado
 - Clase 3: Referenciada, Modificado
- El algoritmo escoge una página partiendo desde las con clase menor.
- El sistema operativo en cada interrupción de reloj cambia el estado del bit R, para poder identificar cuáles páginas son referenciadas constantemente durante el quantum.

Algoritmos de Reemplazo de Páginas

Algoritmo NRU (*Not Recently Used*)

Pág.	Bit R	Bit M
0	0	0
1	0	0
2	0	0
3	0	0

Estado de la tabla de páginas en t_0



Durante el quantum se acceden las páginas 0, 1 y 3; y se escribe en las 0 y 3

Pág.	Bit R	Bit M
0	1	1
1	1	0
2	0	0
3	1	1

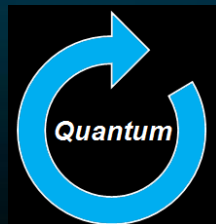
Estado de la tabla de páginas en t_1



El SO apaga todos los bits R

Pág.	Bit R	Bit M
0	1	1
1	1	0
2	0	0
3	0	1

Estado de la tabla de páginas en t_2



Durante el siguiente quantum se hacen accesos a las páginas 0 y 1, luego se pide la página 4 que no está presente y se produce una falta de página.

Pág.	Bit R	Bit M
0	0	1
1	0	0
2	0	0
3	0	1

Estado de la tabla de páginas en t_3

Algoritmos de Reemplazo de Páginas

Algoritmo de Envejecimiento

- ❑ Cada página tiene un contador asociado.
- ❑ El algoritmo de envejecimiento tiene en cuenta tanto en que momento fue referenciada frecuentemente una página, y no solamente cuántas veces fue referenciada.
- ❑ Con cada pulso del reloj primero se desplazan a la derecha los contadores (se dividen entre 2) y después el bit R correspondiente a cada página se añade al bit más significativo.
- ❑ Cuando sucede una falta de página, la página cuyo contador es el más bajo se reemplaza (página que menos ha sido referenciada en los últimos ticks de reloj).

Algoritmos de Reemplazo de Páginas

Algoritmo de Envejecimiento

Bits R para
las páginas 0 a 5,
pulso de reloj 0

1	0	1	0	1	1
---	---	---	---	---	---

Bits R para
las páginas 0 a 5,
pulso de reloj 1

1	1	0	0	1	0
---	---	---	---	---	---

Bits R para
las páginas 0 a 5,
pulso de reloj 2

1	1	0	1	0	1
---	---	---	---	---	---

Bits R para
las páginas 0 a 5,
pulso de reloj 3

1	0	0	0	1	0
---	---	---	---	---	---

Bits R para
las páginas 0 a 5,
pulso de reloj 4

0	1	1	0	0	0
---	---	---	---	---	---

1	0	0	0	0	0
---	---	---	---	---	---

1	1	0	0	0	0
---	---	---	---	---	---

1	1	0	0	0	0
---	---	---	---	---	---

1	1	1	0	0	0
---	---	---	---	---	---

0	1	1	1	0	0
---	---	---	---	---	---

0	0	0	0	0	0
---	---	---	---	---	---

1	0	0	0	0	0
---	---	---	---	---	---

1	1	0	0	0	0
---	---	---	---	---	---

0	0	1	1	0	0
---	---	---	---	---	---

1	0	1	1	0	0
---	---	---	---	---	---

1	0	0	0	0	0
---	---	---	---	---	---

0	1	0	0	0	0
---	---	---	---	---	---

0	0	1	0	0	0
---	---	---	---	---	---

0	0	1	0	0	0
---	---	---	---	---	---

1	0	0	1	0	0
---	---	---	---	---	---

0	0	0	0	0	0
---	---	---	---	---	---

0	0	0	0	0	0
---	---	---	---	---	---

1	0	0	0	0	0
---	---	---	---	---	---

0	1	0	0	0	0
---	---	---	---	---	---

0	0	1	0	0	0
---	---	---	---	---	---

1	0	0	0	0	0
---	---	---	---	---	---

1	1	0	0	0	0
---	---	---	---	---	---

0	1	1	0	0	0
---	---	---	---	---	---

1	0	1	1	0	0
---	---	---	---	---	---

0	1	0	1	1	0
---	---	---	---	---	---

1	0	0	0	0	0
---	---	---	---	---	---

0	1	0	0	0	0
---	---	---	---	---	---

1	0	1	0	0	0
---	---	---	---	---	---

0	1	0	1	0	0
---	---	---	---	---	---

0	0	1	0	1	0
---	---	---	---	---	---

Algoritmos de Reemplazo de Páginas

Conjunto de Trabajo (*Working Set*)

- Empíricamente los procesos al ejecutarse no hacen referencia a todas sus páginas constantemente. Sucede que en cualquier fase de la ejecución el proceso exhibe una localidad de referencia, es decir, accede a un subconjunto pequeño de la totalidad de sus páginas
- Formalmente se denomina Conjunto de Trabajo al conjunto de páginas que está utilizando en un momento dado. Este no es fijo, y cambia periódicamente a medida que el proceso se ejecuta.
- Si no existe memoria disponible en el sistema para contener el conjunto de trabajo de un proceso, este generará muchos fallos de páginas en un número reducido de instrucciones. A este fenómeno se le denomina hiperpaginación.

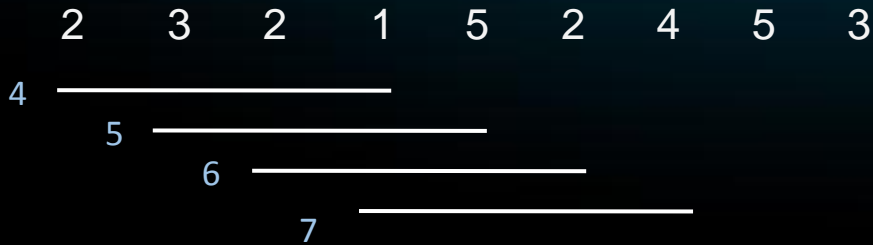
Algoritmos de Reemplazo de Páginas

Algoritmo del Conjunto de Trabajo (*Working Set*)

- El algoritmo Working Set se basa en lograr mantener en memoria las páginas que se consideran dentro del conjunto de trabajo para así reducir al máximo la cantidad de faltas de páginas.

- Para establecer el conjunto se puede usar uno de los siguientes métodos:

1. Páginas referenciadas en los k últimos accesos a memoria, aquellas que no están en el conjunto pueden ser desalojadas



T	Working set
4	2, 3, 1
5	3, 2, 1, 5
6	2, 1, 5
7	1, 5, 2, 4

Algoritmos de Reemplazo de Páginas

Algoritmo del Conjunto de Trabajo (*Working Set*)

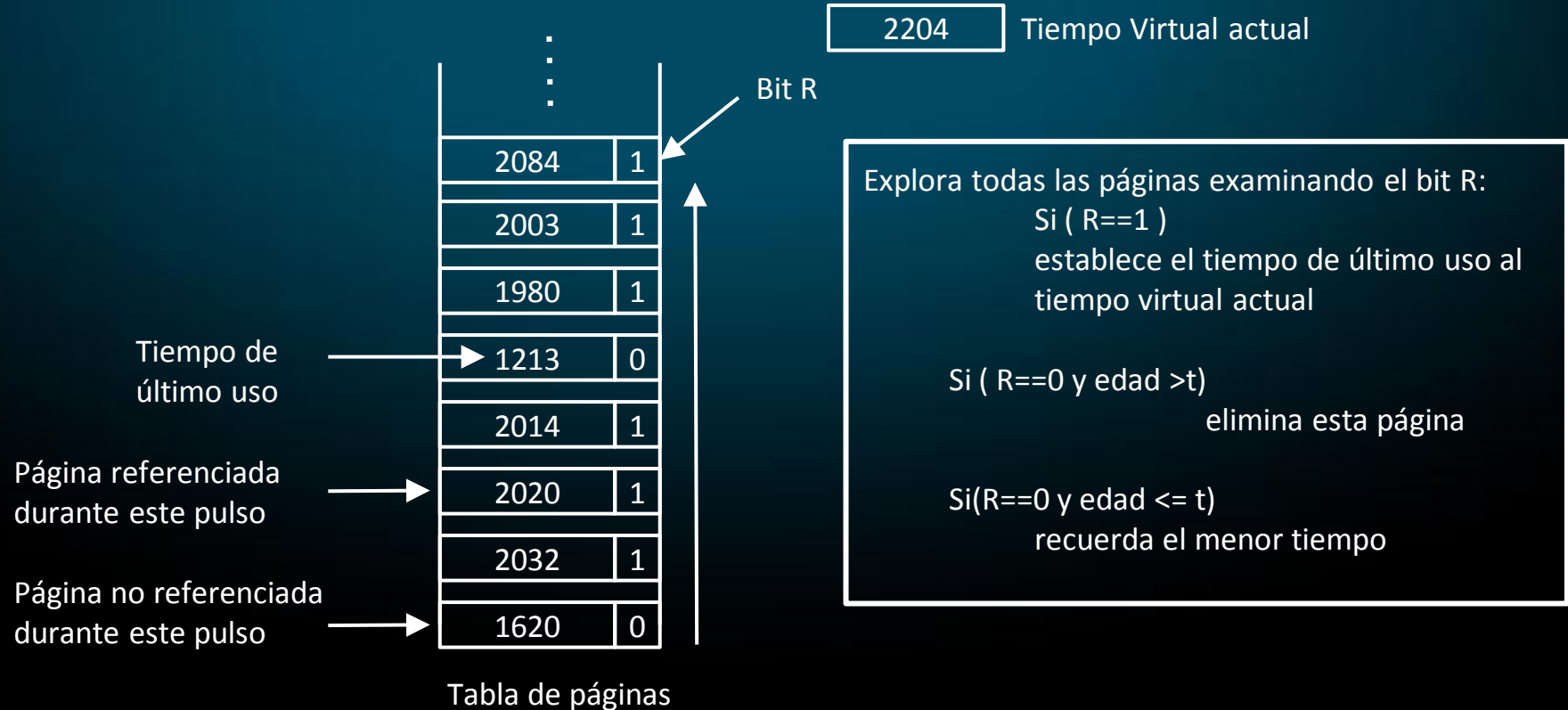
2. Usar el tiempo virtual de ejecución (tiempo real que ha ocupado el proceso en CPU) para determinar la edad de la página. Aquellas que sobrepasan una edad máxima T se consideran fuera del conjunto y pueden ser desalojadas.

El bit R se apaga en cada interrupción de reloj, actualizando el tiempo de acceso a aquellas páginas cuyo bit R es 1.

En caso de presentarse una falta de página se procede de la siguiente manera:

Algoritmos de Reemplazo de Páginas

Algoritmo del Conjunto de Trabajo (*Working Set*)



Algoritmos de Reemplazo de Páginas

Algoritmo del Conjunto de Trabajo (*Working Set*)

- Si se explora toda la tabla de páginas sin encontrar una candidata para ser sustituida, significa que todas las páginas están en el conjunto de trabajo. En tal caso, si se encontró una o más páginas con $R = 0$, se sustituye la de mayor edad. En el peor caso, todas las páginas se habrán referenciado durante el tic de reloj actual (y, por lo tanto, todas tienen $R = 1$), así que se escoge una al azar para sustituirla, preferiblemente una que esté limpia.
- Siempre la exploración termina de actualizar las entradas restantes de la tabla de páginas.

Algoritmos de Reemplazo de Páginas

Algoritmo del Conjunto de Trabajo (*Working Set*)

Ejemplo: ocurre un fallo de página en la siguiente situación:

Se elimina esta página

Igual al segundo caso

Igual al primer caso

No es una candidata ideal para la sustitución, su última referencia fue después de $T = 1204$

⋮	
2084	1
2003	1
1980	0
1203	0
2014	1
1500	0
2032	1
1620	0

Tabla de páginas

$t = 1000$

Tiempo Virtual = 2204

Se termina de actualizar la tabla

Se actualiza su tiempo de último uso

Estado final:

⋮	
2204	1
2204	1
1980	0
2204	1
2204	1
1500	0
2204	1
1620	0

Nueva página

Algoritmos de Reemplazo de Páginas

Algoritmo WSClock

- Se usa conjuntamente las ideas del algoritmo del reloj y la información del conjunto de trabajo.
- Es necesario usar una lista circular tal como en el algoritmo del reloj.
- Cada entrada contiene el campo Tiempo de último uso del algoritmo básico del conjunto de trabajo, así como el bit R y el bit M .
- En cada tic del reloj se actualizan los tiempos de uso de las páginas que poseen en bit R en 1.

Algoritmos de Reemplazo de Páginas

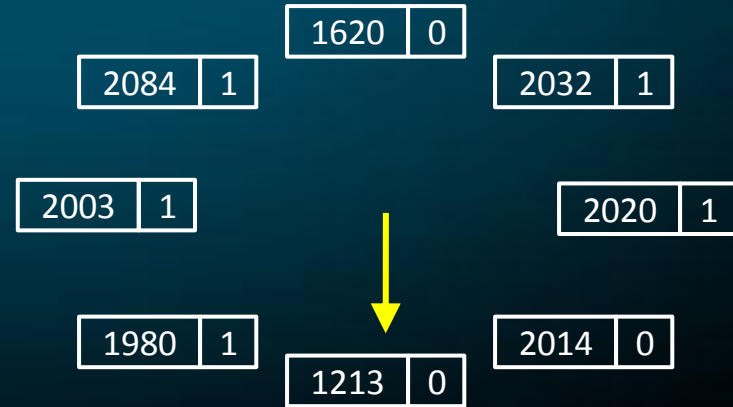
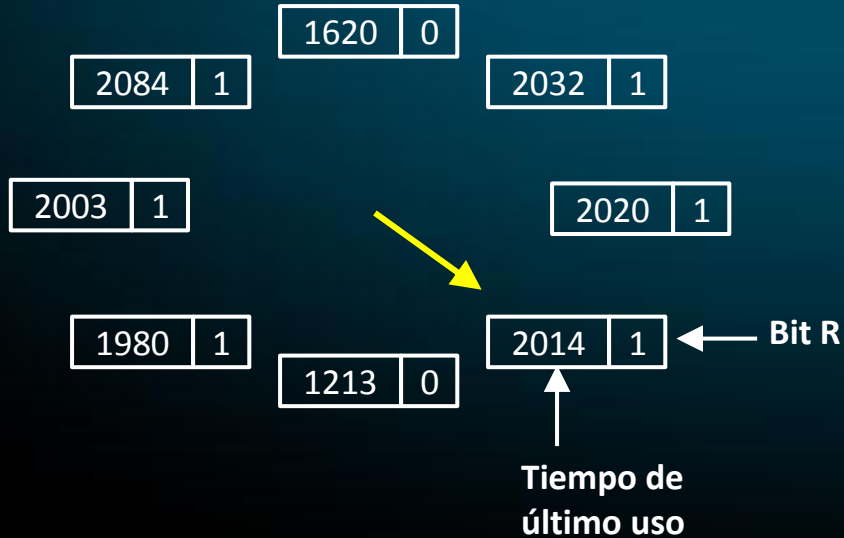
Algoritmo WSClock

- Cada vez que ocurre un fallo de página se revisa primero la página a la que apunta la manecilla.
- Si el bit R es 1, entonces no es una candidata ideal para sustituir, pues se referenció durante el tic actual. Por lo tanto, el bit R se pone a 0, se adelanta la manecilla a la siguiente página y se repite el algoritmo.
- Si el bit R es 0:
 - Si su edad es mayor que t y la página está limpia, la nueva página se coloca en ese marco de página.
 - Si la página está sucia, no podrá sustituirse de inmediato. Para evitar el cambio de página, el proceso se planifica la escritura en disco, pero la manecilla se adelanta y el algoritmo continúa con la siguiente página.
- Para reducir el tráfico del disco se puede fijar un límite de escritura de n páginas.

Algoritmos de Reemplazo de Páginas

Algoritmo WSClock

2204 Tiempo virtual actual



Ejemplo de lo que ocurre cuando $R = 1$

Algoritmos de Reemplazo de Páginas

Algoritmo WSClock

2204 Tiempo virtual actual



Ejemplo de lo que ocurre cuando $R = 0$

Algoritmos de Reemplazo de Páginas

Algoritmo WSClock

¿Qué sucede si la manecilla da toda la vuelta y regresa a su punto de partida?

Debemos distinguir dos casos:

1. Se planificó al menos una escritura: la manecilla tan solo se sigue adelantando, en busca de una página limpia. Pues se ha planificado alguna escritura y eventualmente conseguirá un página limpia.
2. No se planificó ninguna escritura: en este caso todas las páginas pertenecen al conjunto de trabajo, si encontró alguna página limpia durante el recorrido se sustituye. Si no hay páginas limpias se escoge como víctima la página actual y se escribe en el disco y se desaloja.

Algoritmos de Reemplazo de Páginas

Óptimo: No se puede implementar, pero es útil como punto de comparación.

LRU (menos usadas recientemente): Excelente, pero difícil de implementar con exactitud.

FIFO (primera en entrar, primera en salir): Podría descartar páginas importantes.

Segunda oportunidad: Gran mejora sobre FIFO.

Reloj: Realista, mejora de implementación de la segunda oportunidad.

NRU (No usadas recientemente): Una pobre aproximación del LRU.

Envejecimiento: Algoritmo eficiente que se aproxima bien a LRU.

Conjunto de trabajo: Muy costoso de implementar.

WSClock: Algoritmo eficientemente bueno.

Modelación de Algoritmos de Paginación

Anomalía de Belady

Caso donde agregar marcos de página a la memoria de un proceso, aumenta la cantidad de faltas de página.

Ejemplo: Sean dos sistemas con reemplazo de páginas FIFO, uno con 3 y el otro con 4 marcos de página, con la siguiente cadena de referencias: 0 1 2 3 0 1 4 0 1 2 3 4

Cadena de Referencias	0	1	2	3	0	1	4	0	1	2	3	
Más reciente		4	1	2	3	0	1	4	4	4	2	3
			0	1	2	3	0	1	1	1	4	2
Más antigua				0	1	2	3	0	0	0	1	4

F F F F F F F F F F

9 Faltas de Página

Cadena de Referencias	0	1	2	3	0	1	4	0	1	2	3	4
Más reciente		0	1	2	3	3	3	4	0	1	2	3
			0	1	2	2	2	3	4	0	1	2
				0	1	1	1	2	3	4	0	1
Más antigua					0	0	0	1	2	3	4	0

F F F F F F F F F F F F

10 Faltas de

Página

Modelación de Algoritmos de Paginación

Algoritmos de Pila

Presentan la propiedad de que teniendo una secuencia de referencias, una memoria con más marcos de página y un número determinado de referencias, mantiene el conjunto de páginas que contendría con un tamaño menor en la misma referencia. En otras palabras, la tasa de faltas nunca aumenta si se incrementa el tamaño de memoria. Por lo tanto evitan la anomalía de Belady.

Ejemplo: Sean dos sistemas con reemplazo de páginas LRU, con 3 y 4 marcos de página respectivamente, con la siguiente cadena de referencias: **0 2 1 3 5 4 3 1 4**

		0	2	1	3	5	4	3	1	4
Páginas en Memoria		0	2	1	3	5	4	3	1	4
			0	2	1	3	5	4	3	1
				0	2	1	3	5	4	3
Páginas en Disco					0	2	1	1	5	5
						0	2	2	2	2
							0	0	0	0
7 Faltas de Pg.		F	F	F	F	F	F		F	

		0	2	1	3	5	4	3	1	4	
Páginas en Memoria			0	2	1	3	5	4	3	1	4
				0	2	1	3	5	4	3	1
					0	2	1	3	5	4	3
						0	2	1	1	5	5
Páginas en Disco							0	2	2	2	2
								0	0	0	0
6 Faltas de Pg.		F	F	F	F	F	F				

Modelación de Algoritmos de Paginación

La Cadena de Distancias

Puede decirse que es la distancia que hay desde la parte superior de la pila hasta donde se encuentra la página antes de ser referenciada. Tomando el reemplazo de páginas LRU, tenemos:

Como la página 3 no había sido referenciada, tiene distancia infinita.

Ahora la página 3 antes de su referencia estaba en la posición 3, por lo tanto su distancia es 3.

La página 1 antes de su referencia estaba en la posición 4, por lo tanto su distancia es 4.

		0	2	1	3	5	4	3	1	4	4	1	2	0
Páginas en Memoria		0	2	1	3	5	4	3	1	4	4	1	2	0
			0	2	1	3	5	4	3	1	1	4	1	2
				0	2	1	3	5	4	3	3	3	4	1
Páginas en Disco					0	2	1	1	5	5	5	5	3	4
						0	2	2	2	2	2	2	5	3
							0	0	0	0	0	0	0	5
		F	F	F	F	F	F		F			F	F	
		∞	∞	∞	∞	∞	∞	3	4	3	1	2	5	6

9 Falta de Página
Cadena de Distancias

Vector C

$$C_1 = 1$$

$$C_2 = 1$$

$$C_3 = 2$$

$$C_4 = 1$$

$$C_5 = 1$$

$$C_6 = 1$$

$$C_{\infty} = 6$$

Modelación de Algoritmos de Paginación

Predicción de la Tasa de Faltas de Página

Utilizando en el vector C que se calcula a través de la cadena de distancias, podemos obtener la predicción de la tasa de faltas con la siguiente fórmula:

$$F_m = \sum_{k=m+1}^n C_k + C_\infty$$

m = marcos de página.

n = cantidad de páginas.

Vector C

$C_1 = 1$
$C_2 = 1$
$C_3 = 2$
$C_4 = 1$
$C_5 = 1$
$C_6 = 1$
$C_\infty = 6$

Vector F

$F_1 = 12$
$F_2 = 11$
$F_3 = 9$
$F_4 = 8$
$F_5 = 7$
$F_6 = 6$
$F_\infty = 6$

$$C_2 + C_3 + C_4 + C_5 + C_6 + C_\infty$$

$$C_3 + C_4 + C_5 + C_6 + C_\infty$$

$$C_4 + C_5 + C_6 + C_\infty$$

.

.

.

Con una memoria de 5 marcos de página se producirían 7 faltas de página de las 13 referencias.

Modelación de Algoritmos de Paginación

		0	2	1	3	5	4	3	1	4	4	1	2	0
Páginas en Memoria		0	2	1	3	5	4	3	1	4	4	1	2	0
Páginas en Disco			0	2	1	3	5	4	3	1	1	4	1	2
				0	2	1	3	5	4	3	3	3	4	1
					0	2	1	1	5	5	5	5	3	4
						0	2	2	2	2	2	2	5	3
							0	0	0	0	0	0	0	5
	F	F	F	F	F	F	F	F	F	F	F		F	

**1 Marco de Página
12 Faltas de Página**

		0	2	1	3	5	4	3	1	4	4	1	2	0
Páginas en Memoria		0	2	1	3	5	4	3	1	4	4	1	2	0
			0	2	1	3	5	4	3	1	1	4	1	2
				0	2	1	3	5	4	3	3	3	4	1
					0	2	1	1	5	5	5	5	3	4
						0	2	2	2	2	2	2	5	3
Páginas en Disco							0	0	0	0	0	0	0	5
	F	F	F	F	F	F	F							F

**5 Marco de Página
7 Faltas de Página**

Aspectos de Diseño para los Sistemas de Paginación

Tamaño de Página

La elección de un tamaño de página ideal va a depender de varios factores:

- Con páginas de pequeño tamaño se evita la fragmentación interna, pero tiene ciertos inconvenientes que son:
 - ❑ Aumenta de forma considerable la tabla de páginas.
 - ❑ Las transferencias a y desde el disco se incrementan de forma alarmante, esto produce pérdidas de tiempo, sobre todo por localización y demora rotacional.
- La tabla de páginas se carga en los registros del hardware cuando se pasa de un proceso a otro.

Aspectos de Diseño para los Sistemas de Paginación

Desde un punto de vista matemático, podemos analizar el problema considerando:

s es el tamaño promedio de los procesos

p el tamaño de la página

Cada entrada a la tabla ocupa e bytes

tenemos

s/p número de páginas que se necesitan por proceso

se/p espacio ocupado en la tabla de páginas (bytes)

$p/2$ memoria que se desperdicia en la última página

Así que el costo general debido a ambos factores será:

$$\text{Costo general} = se/p + p/2$$

- Si la página es reducida, el primer término crece.
- Si la página tiene un tamaño grande, aumenta un segundo término.

El tamaño adecuado debe quedar entre ambos. Para ello calculamos la derivada y la igualamos a cero.

$$-se/p + \frac{1}{2} = 0$$

Por lo tanto:

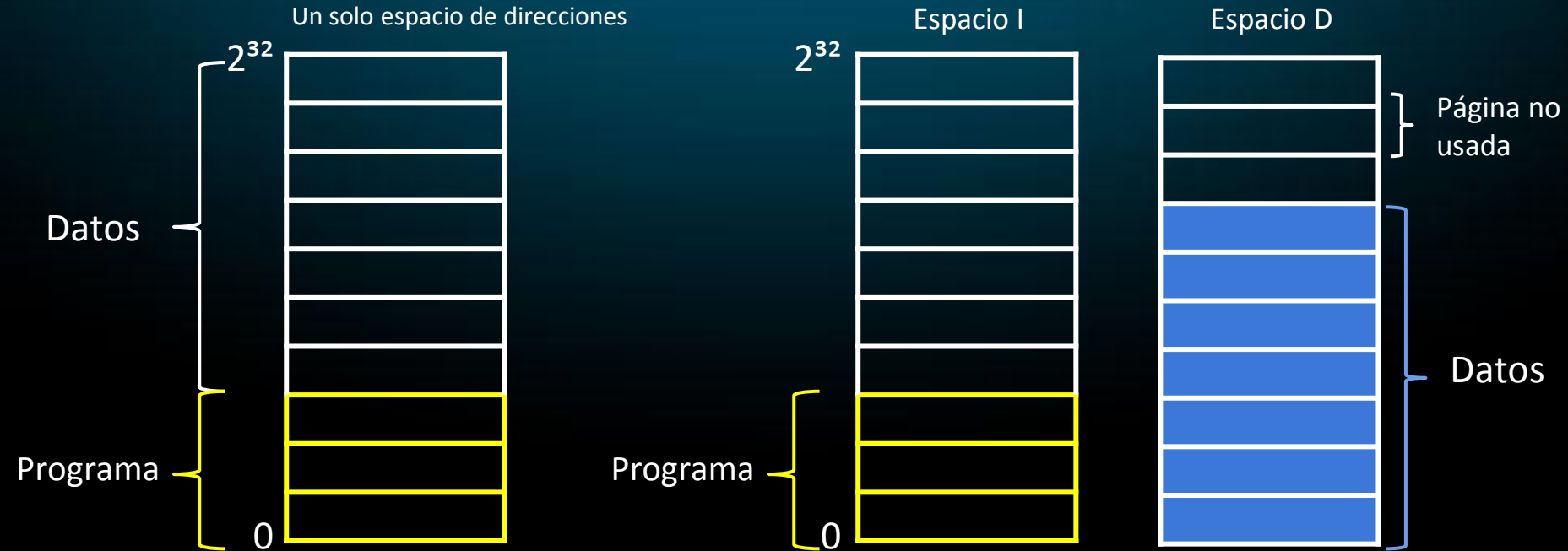
$$p = \sqrt{2se}$$

Aspectos de Diseño para los Sistemas de Paginación

Espacios Separados de Instrucciones y de Datos

Como la mayoría de las computadoras poseen un espacio de direcciones pequeño se planteó la idea de tener dos espacios de direcciones distintos, uno para las instrucciones y otro para los datos.

El enlazador debe estar enterado del uso de espacios I y D separados, porque en tal caso los datos se reubican en la dirección virtual 0 en vez de comenzar después del programa.



Aspectos de Diseño para los Sistemas de Paginación

Control de Carga

Determina el número de procesos que pueden residir en memoria principal, conocido como grado de multiprogramación.

- Si, en un instante dado, hay pocos procesos residentes en memoria principal, habrán muchas ocasiones en las que todos los procesos estén bloqueados.
- Por otro lado, si hay demasiados procesos residentes en memoria principal, el tamaño medio del conjunto residente de proceso no será el adecuado y se producirán frecuentes fallos de páginas. El resultado es una situación conocida como hiperpaginación.

Aspectos de Diseño para los Sistemas de Paginación

❖ Suspensión de Procesos

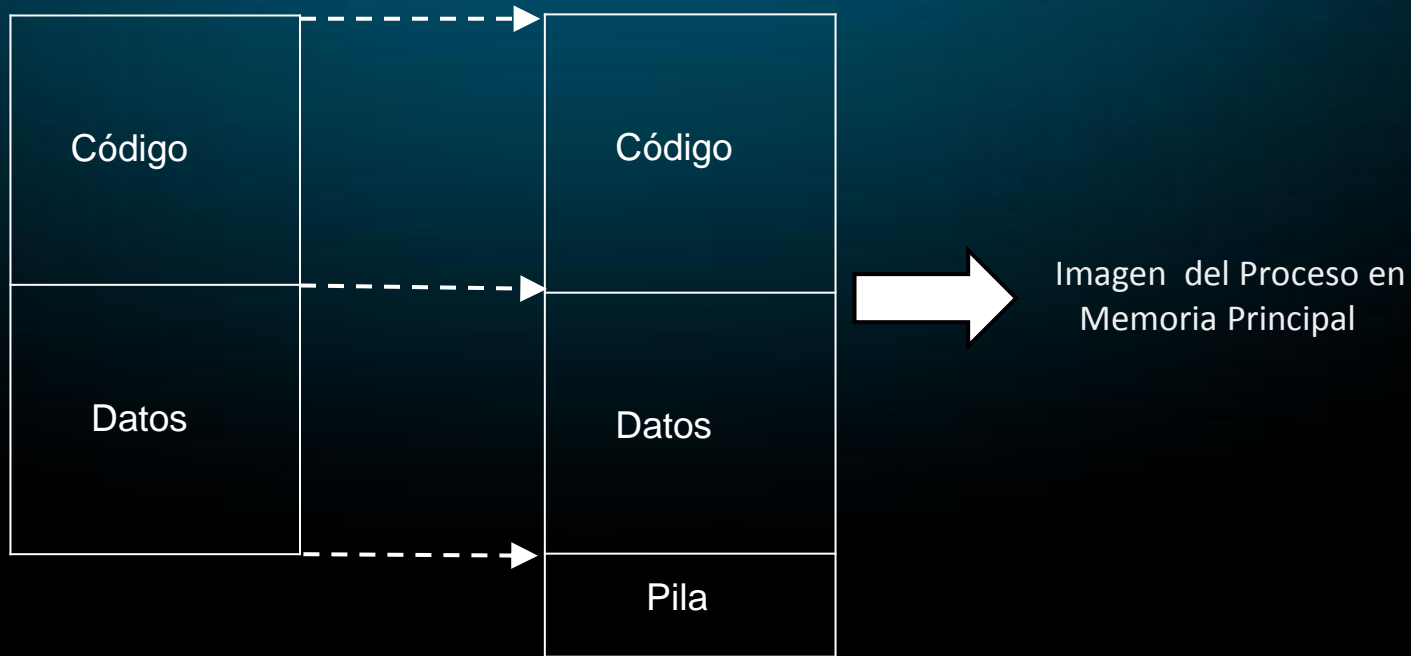
➤ ¿Qué procesos deben suspenderse para disminuir el grado de multiprogramación?

- ❑ Procesos con la prioridad más baja.
- ❑ Procesos con fallos de páginas.
- ❑ Último proceso activado.
- ❑ Proceso con el conjunto residente más pequeño.
- ❑ El proceso mayor.
- ❑ Procesos con la mayor ventana de ejecución restante.

Aspectos de Diseño para los Sistemas de Paginación

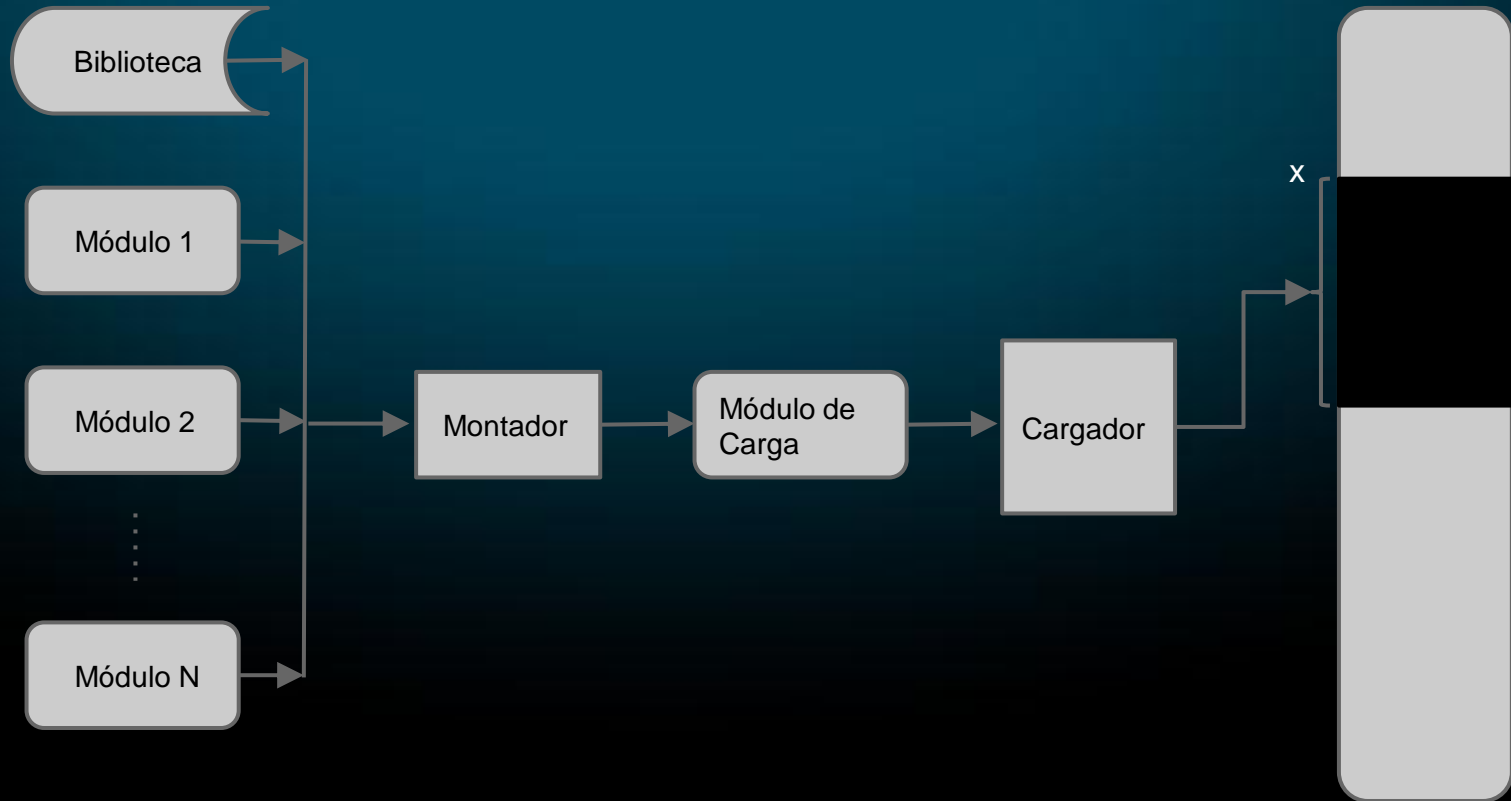
❖ Carga

Consiste en la carga del programa de un proceso en memoria principal y crear una imagen del proceso. Esto es llevado a cabo por medio de un cargador que sitúa el módulo de carga en una posición en memoria.



Aspectos de Diseño para los Sistemas de Paginación

ESQUEMA DE CARGA



Aspectos de Diseño para los Sistemas de Paginación

❖ Carga Absoluta

La carga absoluta necesita que el módulo de carga ocupe siempre la misma posición de memoria principal. Así pues, todas las referencias del módulo de carga para el cargador debe ser direcciones específicas o absolutas en memoria principal.

➤ Desventaja

- ❑ El módulo de carga resultante sólo puede situarse en una región de memoria principal.

Aspectos de Diseño para los Sistemas de Paginación

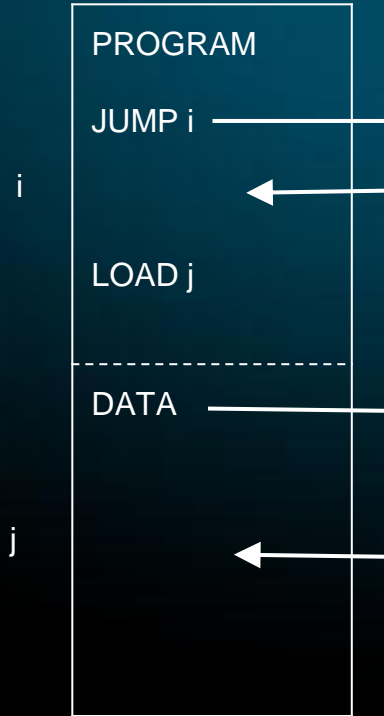
❖ Carga Reubicable

En la carga reubicable, el ensamblador o el compilador no generará direcciones reales de memoria principal (direcciones absolutas) sino direcciones relativas a algún punto conocido, tal como el comienzo del programa.

Si el módulo va a ser cargado comenzando por la posición x , el cargador simplemente sumará x a cada referencia a memoria a medida que cargue el módulo en memoria.

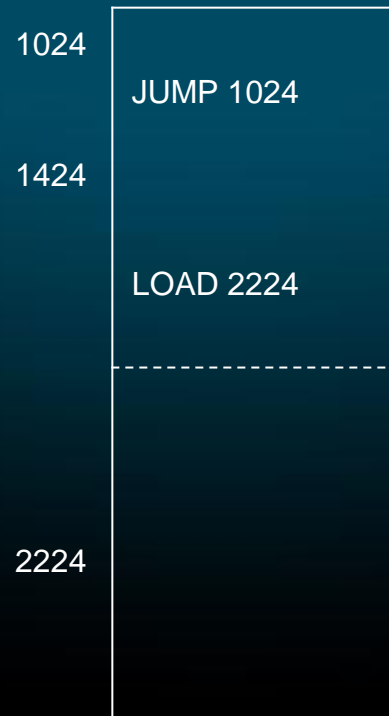
Políticas de Administración de la Memoria

Direcciones
simbólicas



Módulo objeto

Direcciones
Absolutas



Módulo de carga absoluta

Direcciones
Relativas



Módulo de carga relativa

Aspectos de Implantación para los Sistemas de Paginación

Participación del Sistema Operativo en la Paginación

Hay cuatro ocasiones en las que el sistema operativo tiene que realizar trabajo relacionado con la paginación:

- ❖ Cuando se crea un proceso
- ❖ Cuando se ejecuta un proceso
- ❖ Cuando ocurre un fallo de página
- ❖ Cuando un proceso termina

Aspectos de Implantación para los Sistemas de Paginación

Respaldo de Instrucción

Cuando un programa hace referencia a una página que no está en memoria, la instrucción que produjo el fallo se detiene antes de terminar de ejecutarse y se efectúa un salto al sistema operativo.

Bloqueo de Páginas en Memoria

Si un dispositivo de E/S se encuentra en el proceso de realizar una transferencia por DMA a una página, al eliminarla parte de los datos se escribirán el búfer al que pertenecen y parte sobre la página que se vaya a cargar.

Protección de la Memoria

Protección de la Memoria

La protección de memoria evita que un proceso acceda a la memoria que no le ha sido asignada. Así pueden evitarse problemas durante la ejecución del proceso, y también se evita que proceso maligno acceda a los recursos del sistema.

❖ Aspectos de la Protección

- Cada dirección física generada por la CPU es controlada para comprobar si es una dirección válida.
- En caso de un acceso inválido se genera una interrupción al sistema operativo.
- La unidad que convierte direcciones lógicas a físicas es la MMU (Memory Management Unit), y es la que controla el acceso a memoria.
- La unidad MMU únicamente debe ser administrada en modo supervisor.

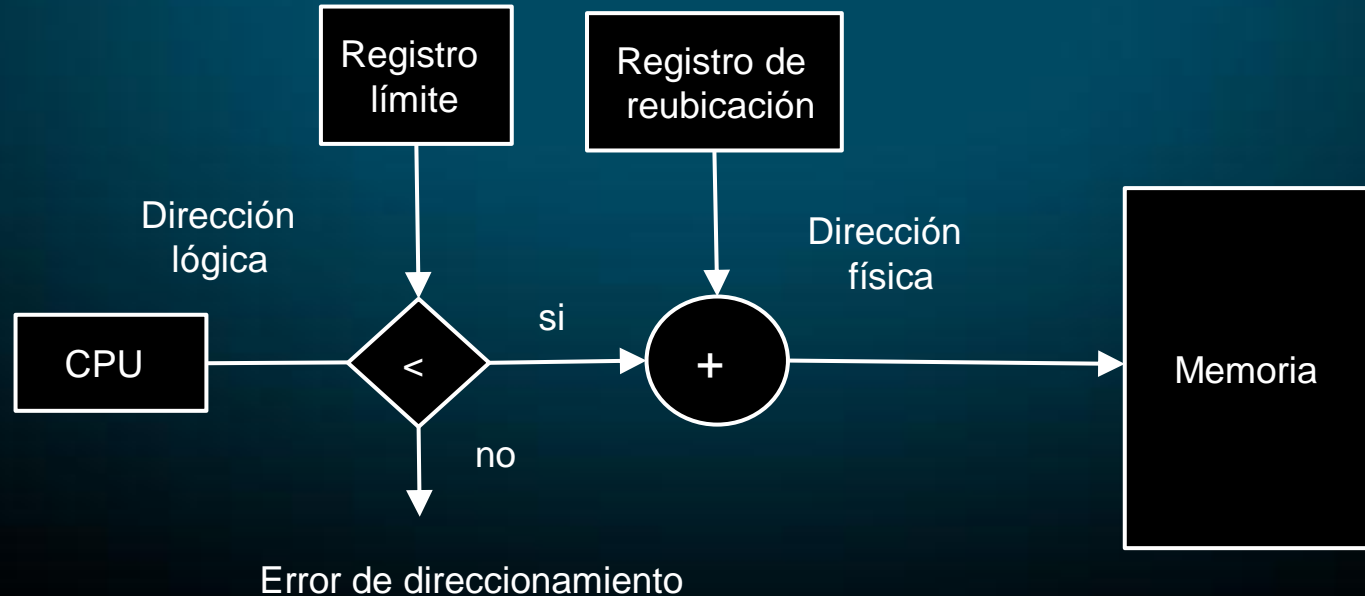
Protección de la Memoria

¿Cómo llevar a cabo la protección de la memoria?

❖ Registro Base y Registro Límite.

- Cada dirección generada por la CPU se compara si es menor que el registro límite. En caso de serlo, la dirección física se obtiene sumando el valor del registro base.
- Si la dirección es mayor que el registro límite se genera una excepción.
- El hardware ha de proporcionar dichos registros.
- El cambio de contexto actualizará el valor de dichos registros.

Protección de la Memoria



Protección de la Memoria

❖ Llave de Protección

La memoria física es dividida en bloques de un tamaño particular (ej. 2KB), donde cada bloque tiene asociado con una llave de protección.

➤ ¿Cómo acceder al bloque correspondiente ?

- ❑ El hardware comprueba que la llave del proceso actual coincida con la del bloque de memoria al que se accede.
- ❑ Si coincide, se le permite llevar a cabo la operación.
- ❑ En caso contrario, este genera una excepción.

Protección de la Memoria

❖ Protección de memoria en entorno con Paginación

- En la Tabla de páginas pueden encontrarse unos bits de protección asociado a cada marco.
- Indican si la página es de sólo lectura o lectura o escritura.
- Cuando se consulta el número de marco, se consulta además los bits de protección.
- Se debe controlar que el número de páginas no supere el total de páginas usada por el proceso (sería una dirección incorrecta).

Compartición de la Memoria

Compartición

La compartición de la memoria permite comunicar procesos que se ejecutan. Esto llevaría a que las direcciones lógicas de dos o más procesos, posiblemente distintas entre sí, se correspondan con la misma dirección física. La posibilidad que dos o más procesos compartan una zona de memoria implica que el sistema de gestión de memoria debe permitir que la memoria asignada a un proceso no sea contigua.

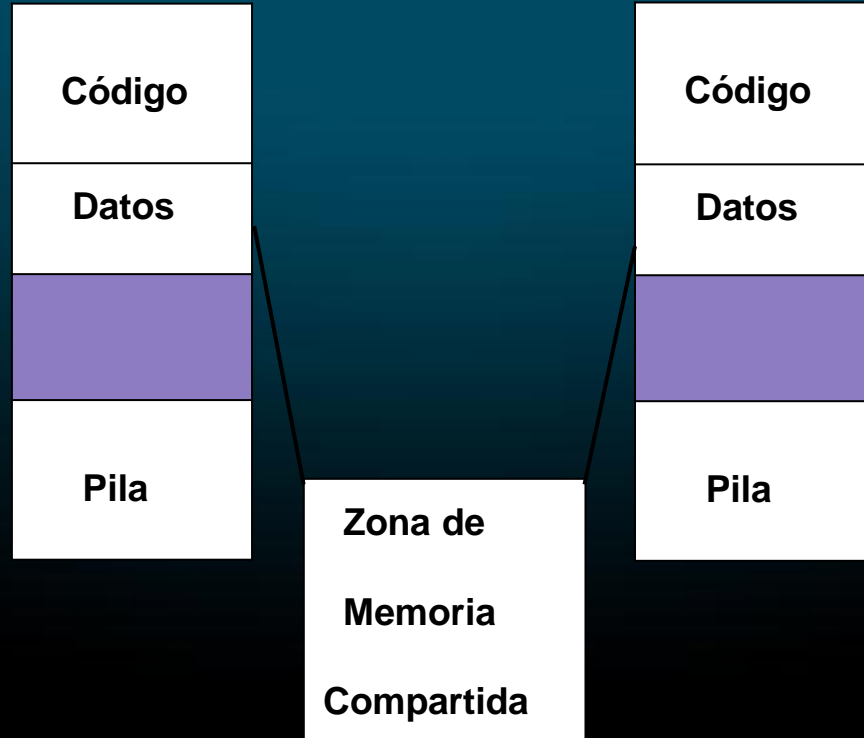
Compartición de la Memoria

❖ Aspectos de la Compartición

- Debe haber flexibilidad para permitir que varios procesos accedan a una misma zona de memoria:
 - ❑ dos procesos ejecutando el mismo programa comparten código.
 - ❑ varios procesos pueden utilizar una misma estructura de datos.
- Se trata de permitir que direcciones lógicas de dos o más procesos posiblemente distintas entre sí, se correspondan con la misma dirección física.
- La compartición de memoria no debe comprometer la protección básica.
- Los mecanismos de paginación y segmentación resuelven adecuada y simultáneamente los problemas de reubicación, protección y compartición.

Compartición de la Memoria

Procesos compartiendo
zona de memoria



Compartición de la Memoria

❖ Páginas Compartidas

- La paginación permite compartir código común entre varios procesos:
 - ❑ Sólo si el código es reentrante (no se modifica durante ejecución).
 - ❑ El área de datos de los procesos sería diferente .
 - ❑ Ejemplo: Varios procesos ejecutan el mismo editor de textos.

Compartición de la Memoria

**Una Única copia
Del Editor en
Memoria física**

Proceso 1

Memoria
Lógica

Editor 1
Editor 2
Editor 3
Datos 1

0

1

2

3

Tabla de
Páginas

3
4
5
1

Proceso 2

Memoria
Lógica

Editor 1
Editor 2
Editor 3
Datos 2

0

1

2

3

Tabla de
Páginas

3
4
5
7

Memoria
Física

0

1

2

3

4

5

6

7

Datos 1
Editor 1
Editor 2
Editor 3
Datos 2

Compartición de la Memoria

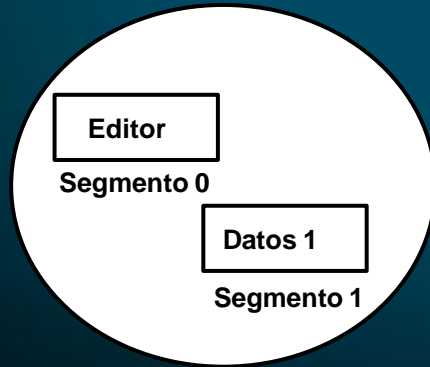
❖ Segmentos Compartidos

➤ La segmentación permite compartir código común entre varios procesos:

- ❑ Pueden realizarse a nivel de segmento (código o datos).
- ❑ Cada proceso tendrá una tabla de segmento.
- ❑ Compartir un segmento significa que una entrada de la tabla de segmentos coincide en varios procesos (igual posición física).
- ❑ Si compartimos un segmento todos los procesos que lo comparten deben definir dicho segmento con el mismo código.

Compartición de la Memoria

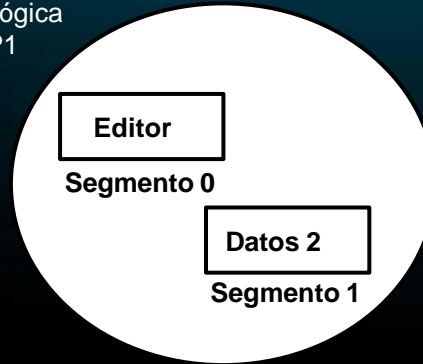
❑ Ejemplo compartición Editor :



Memoria Lógica
Proceso P1

Tabla de Segmentos
Proceso P1

	Límite	Base
0	25286	43062
1	4425	68348

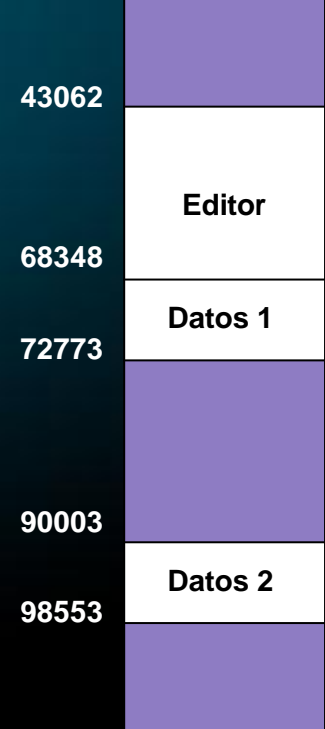


Memoria Lógica
Proceso P2

Tabla de Segmentos
Proceso P2

	Límite	Base
0	25286	43062
1	8550	90003

Memoria física



Políticas de Administración de la Memoria

Políticas de Lectura

Está relacionada con la decisión de cuándo se debe cargar una página en memoria principal. Las dos alternativas más comunes son la paginación por demanda y la paginación previa.

❖ Paginación por Demanda

Se trae una página a memoria principal sólo cuando se hace referencia a una posición en dicha página.

Ventaja: Mejora el grado de multiprogramación.

Desventaja: Cada fallo de página requiere cargar a memoria principal una página a leer.

❖ Paginación Previa

Se cargan otras páginas distintas a las demandadas debido a un fallo de página.

Ventaja: Cuando se hace referencia a una página ya puede estar cargada.

Desventaja: Se pueden cargar páginas innecesarias.

Políticas de Administración de la Memoria

Políticas de Ubicación

Tiene que ver con determinar dónde va a residir una parte de un proceso en memoria principal.

- **Posibles alternativas:**

Mejor ajuste

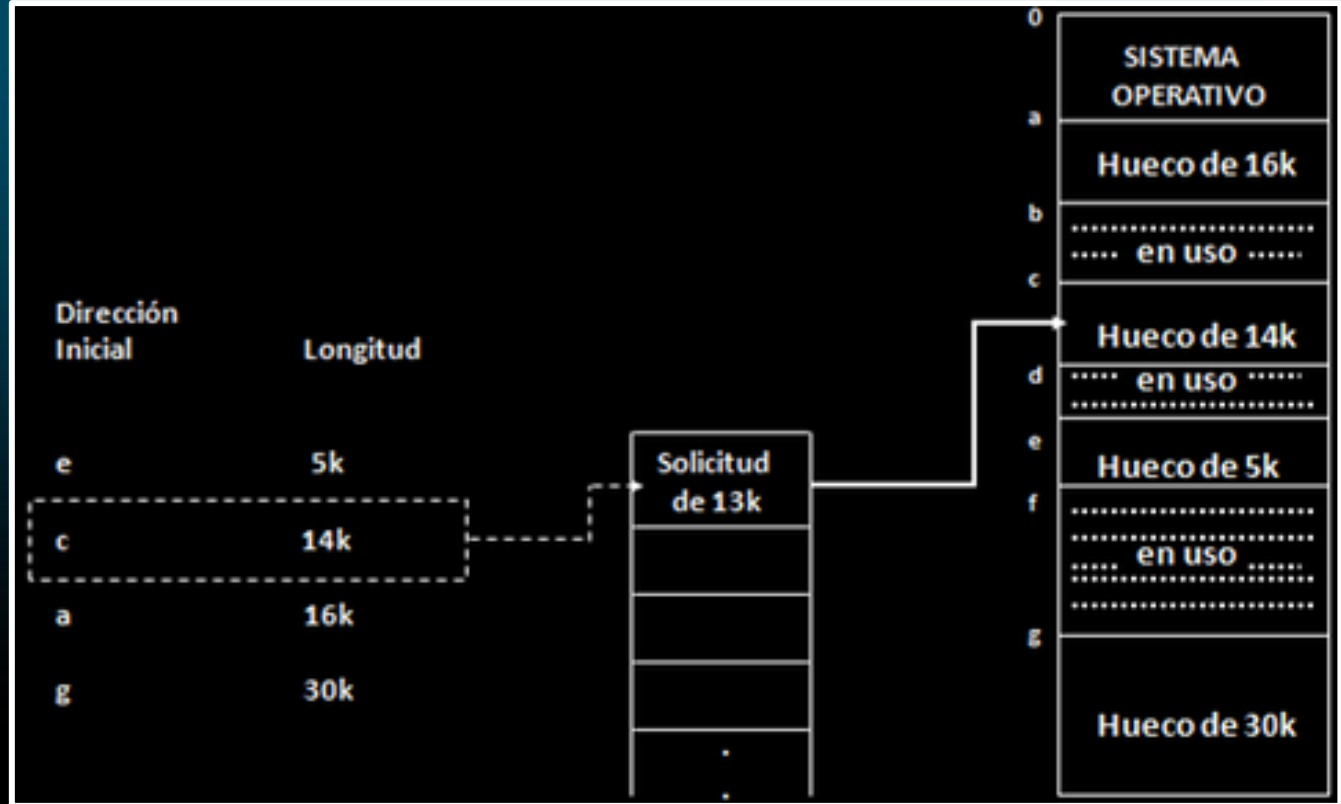
Peor ajuste

Primer ajuste

Políticas de Administración de la Memoria

Políticas de Ubicación

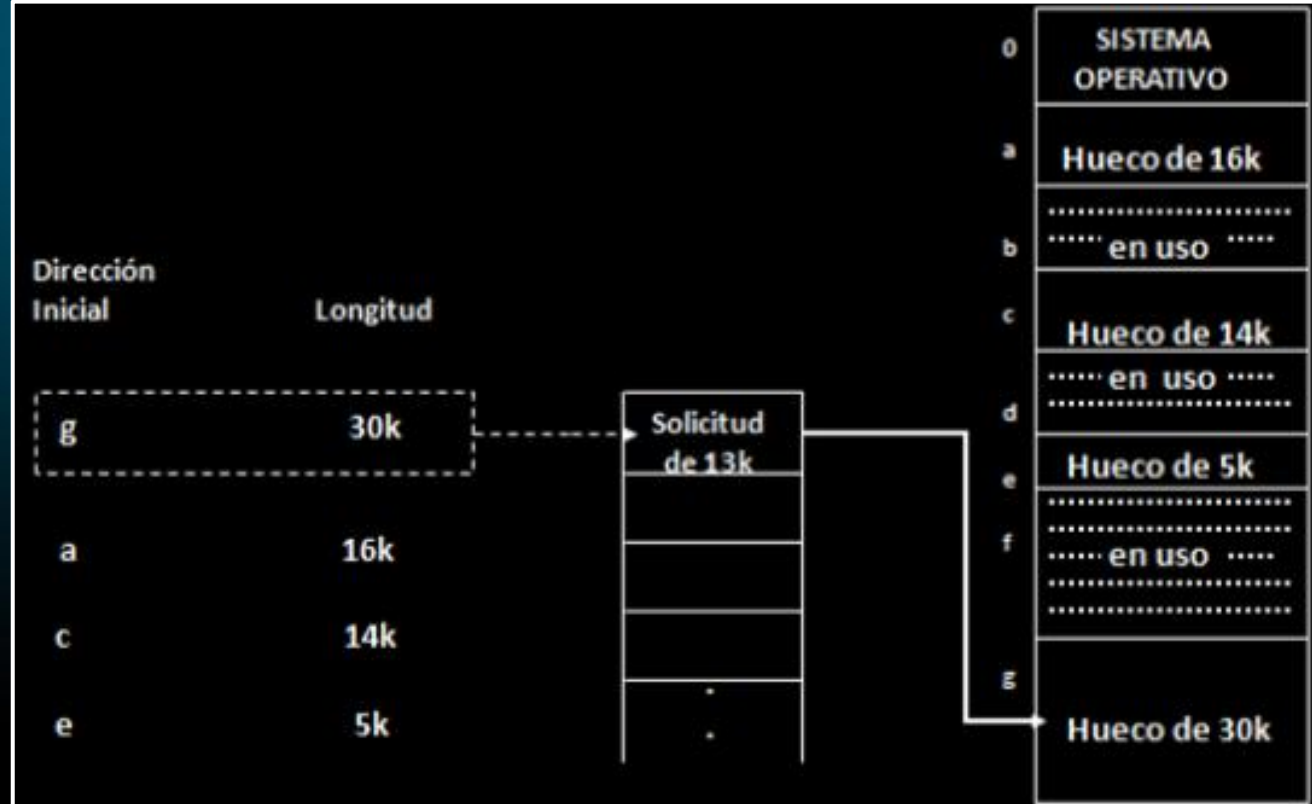
❖ **Mejor Ajuste:**
Colocar el
proceso en el
menor bloque en
el que quepa.



Políticas de Administración de la Memoria

Políticas de Ubicación

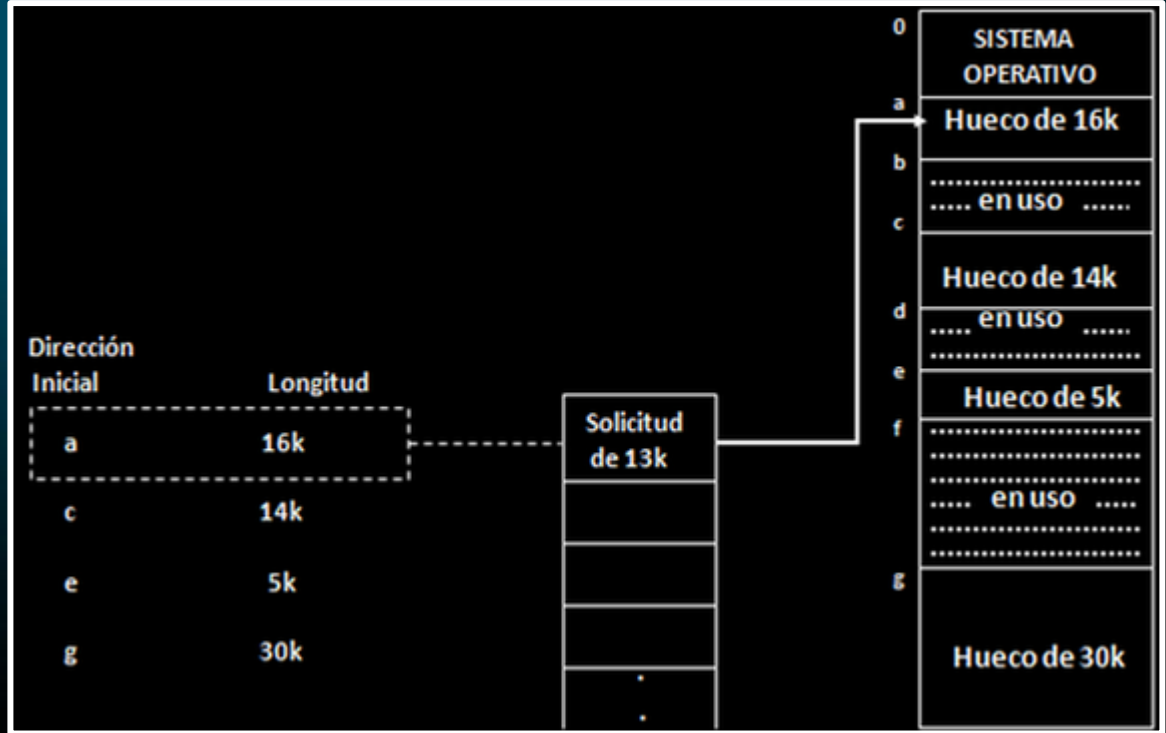
- ❖ Peor Ajuste:
Colocar el proceso en el espacio más grande en el que quepa.



Políticas de Administración de la Memoria

Políticas de Ubicación

- ❖ **Primer Ajuste:** Colocar el proceso en el primer espacio de la lista de almacenamiento libre en el que quepa.



Políticas de Administración de la Memoria

Políticas de Reemplazo

- Local
- Global

A0
A1
A2
A3
A4
A5
B0
B1
B2
B3
B4
B5
B6
C1
C2
C3

Reemplazo de
Página Local

A0
A1
A2
A3
A4
A5
B0
B1
B2
B3
B4
B5
B6
C1
C2
C3

Reemplazo de
Página Global

A0
A1
A2
A3
A4
A5
B0
B1
B2
B3
B4
B5
B6
C1
C2
C3

Políticas de Administración de la Memoria

Políticas de Reemplazo

- Local
- Global

A0
A1
A2
A3
A4
A5
B0
B1
B2
B3
B4
B5
B6
C1
C2
C3



Reemplazo de
Página Local

A0
A1
A2
A3
A6
A5
B0
B1
B2
B3
B4
B5
B6
C1
C2
C3

Reemplazo de
Página Global

A0
A1
A2
A3
A4
A5
B0
B1
B2
A6
B4
B5
B6
C1
C2
C3

Algoritmos de Reemplazo de Páginas

Óptimo: No se puede implementar, pero es útil como punto de comparación.

LRU (menos usadas recientemente): Excelente, pero difícil de implementar con exactitud.

FIFO (primera en entrar, primera en salir): Podría descartar páginas importantes.

Segunda oportunidad: Gran mejora sobre FIFO.

Reloj: Realista, mejora de implementación de la segunda oportunidad.

NRU (No usadas recientemente): Una pobre aproximación del LRU.

Envejecimiento: Algoritmo eficiente que se aproxima bien a LRU.

Conjunto de trabajo: Muy costoso de implementar.

WSClock: Algoritmo eficientemente bueno.

Políticas de Administración de la Memoria

Gestión de Conjunto Residente

❖ Tamaño del Conjunto Residente

El S.O debe decidir cuantas páginas traer, es decir, cuanta memoria principal asignar a un determinado proceso.

- **Asignación Fija:** Otorga a cada proceso un número fijo de páginas en las que ejecutar
- **Asignación Variable:** Permite que el número de marcos asignados a un proceso cambie a lo largo de su vida

Políticas de Administración de la Memoria

Gestión de Conjunto Residente

❖ Alcance del Reemplazo

	Reemplazo Local	Reemplazo Global
Asignación Fija	El número de marcos asignados a un proceso es fijo. La página a reemplazar se elige de entre los marcos asignados al proceso.	No es posible
Asignación Variable	El número de marcos asignados a un proceso puede cambiar de un momento a otro para mantener su conjunto de trabajo. La página a reemplazar se elige de entre las páginas asignadas al proceso	La página a reemplazar se elige de entre todos los marcos disponibles en memoria principal; esto hace que cambie el tamaño del conjunto residente de los procesos

Políticas de Administración de la Memoria

Gestión de Conjunto Residente

❖ Alcance del Reemplazo

FIJA

LOCAL



GLOBAL



Nº marcos
asignado
pueden crecer

VARIABLE



La página a
reemplazar se
elige de entre
todos los
marcos

Políticas de Administración de la Memoria

Política de Vaciado

Son las encargadas de tomar la decisión de escribir en memoria principal una página que ha sido modificada. Las dos alternativas más habituales son el vaciado por demanda y el vaciado previo.

❖ Vaciado por Demanda

Una página será escrita en memoria principal sólo cuando haya sido elegida para ser modificada.

Ventaja: Minimiza la cantidad de páginas escritas en memoria secundaria.

Desventaja: Cada fallo de página produce 2 transferencias de página.

❖ Vaciado Previo

Las páginas modificadas, serán escritas en memoria secundaria antes de que se necesiten sus marcos de páginas.

Ventaja: Las páginas modificadas podrán ser escritas por lotes.

Desventaja: Podría haber escrituras innecesarias.

Políticas de Administración de la Memoria

Alternativa: Existe una mejor solución, incorporar un almacenamiento intermedio de páginas, que permita la adopción de la siguiente política:

- Se cuenta con dos listas, una de páginas modificadas y otra de no modificadas.
- Las páginas de la lista de modificadas pueden escribirse periódicamente por lotes y trasladarse a la lista de no modificadas.
- Las páginas de la lista de no modificadas pueden reclamarse, si ha sido de nuevo referenciada o perderse, cuando se asigna su marco a otra página.