

Memoria Virtual

Memoria Virtual es una técnica que permite la ejecución de los procesos, sin que residan completamente en memoria.

Definiciones

Memoria Virtual

Memoria Virtual es una técnica que permite la ejecución de los procesos, sin que residan completamente en memoria.

Espacio Virtual

Cada proceso posee su propio **espacio virtual**, el cual es como percibe la memoria ese proceso. En cuanto al proceso se refiere, este no es almacenamiento en memoria principal.

Definiciones

Memoria Virtual

Memoria Virtual es una técnica que permite la ejecución de los procesos, sin que residan completamente en memoria.

Espacio Virtual

Cada proceso posee su propio **espacio virtual**, el cual es como percibe la memoria ese proceso. En cuanto al proceso se refiere, este no es almacenamiento en memoria principal.

Conjunto Residente (*Resident Set*)

Es la porción de un proceso que se encuentra en la memoria principal en un momento determinado.

- **Se pueden mantener más procesos en memoria principal.** Puesto que se van a cargar sólo algunos fragmentos de un proceso particular, habrá espacio para más procesos.
- **Es posible que un proceso sea más grande que toda la memoria principal.** Se elimina así una de las limitaciones más notorias de la programación.

- **Se pueden mantener más procesos en memoria principal.** Puesto que se van a cargar sólo algunos fragmentos de un proceso particular, habrá espacio para más procesos.
- **Es posible que un proceso sea más grande que toda la memoria principal.** Se elimina así una de las limitaciones más notorias de la programación.

Cómo funciona

Locality En un momento determinado, solo unas pocas piezas de un proceso determinado son usadas (y mantenidas en memoria principal)

Soporte del Hardware Es necesario para soportar los esquemas de paginación o segmentación. La traducción de direcciones virtuales a direcciones físicas la realiza la Unidad de Administración de Memoria o MMU (*Memory Management Unit*) del procesador.

Soporte del S.O. Se necesita código para administrar el movimiento de páginas o segmentos entre la memoria secundaria (disco) y la memoria principal.

Cómo funciona

Locality En un momento determinado, solo unas pocas piezas de un proceso determinado son usadas (y mantenidas en memoria principal)

Soporte del Hardware Es necesario para soportar los esquemas de paginación o segmentación. La traducción de direcciones virtuales a direcciones físicas la realiza la Unidad de Administración de Memoria o MMU (*Memory Management Unit*) del procesador.

Soporte del S.O. Se necesita código para administrar el movimiento de páginas o segmentos entre la memoria secundaria (disco) y la memoria principal.

Cómo funciona

Locality En un momento determinado, solo unas pocas piezas de un proceso determinado son usadas (y mantenidas en memoria principal)

Soporte del Hardware Es necesario para soportar los esquemas de paginación o segmentación. La traducción de direcciones virtuales a direcciones físicas la realiza la Unidad de Administración de Memoria o MMU (*Memory Management Unit*) del procesador.

Soporte del S.O. Se necesita código para administrar el movimiento de páginas o segmentos entre la memoria secundaria (disco) y la memoria principal.

Elementos

- **Tabla de Páginas:** al igual que en la paginación simple, se necesita una MPT (*Memory Page Table*) por proceso. Cada entrada en esta tabla (PTE, *Page Table Entry*) contiene el número de frame de memoria en el que se encuentra la página.
- **Bit de presencia:** como solamente algunas páginas están en memoria principal, se necesita un bit para indicarlo (*Present Bit*).
- **Bit de modificación:** indica si la página se modificó desde que fue cargada en memoria (*Dirty Bit*).

Ejemplo: Formato de una PTE de 32Bits (IA-32)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Address of 4KB page frame																				Ignored		G	P A T	D	A	P C D	P W T	U / S	R / W	1	PTE: 4KB page	
Ignored																										0	PTE: not present					

Elementos

- **Tabla de Páginas:** al igual que en la paginación simple, se necesita una MPT (*Memory Page Table*) por proceso. Cada entrada en esta tabla (PTE, *Page Table Entry*) contiene el número de frame de memoria en el que se encuentra la página.
- **Bit de presencia:** como solamente algunas páginas están en memoria principal, se necesita un bit para indicarlo (*Present Bit*).
- **Bit de modificación:** indica si la página se modificó desde que fue cargada en memoria (*Dirty Bit*).

Ejemplo: Formato de una PTE de 32Bits (IA-32)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Address of 4KB page frame																				Ignored		G	P A T	D	A	P C D	P W T	U / S	R / W	1	PTE: 4KB page	
Ignored																													0	PTE: not present		

Elementos

- **Tabla de Páginas:** al igual que en la paginación simple, se necesita una MPT (*Memory Page Table*) por proceso. Cada entrada en esta tabla (PTE, *Page Table Entry*) contiene el número de frame de memoria en el que se encuentra la página.
- **Bit de presencia:** como solamente algunas páginas están en memoria principal, se necesita un bit para indicarlo (*Present Bit*).
- **Bit de modificación:** indica si la página se modificó desde que fue cargada en memoria (*Dirty Bit*).

Ejemplo: Formato de una PTE de 32Bits (IA-32)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Address of 4KB page frame																Ignored		G	P A T	D	A	P C D	P W T	U / S	R / W	1	PTE: 4KB page					
Ignored																										0	PTE: not present					

Paginación

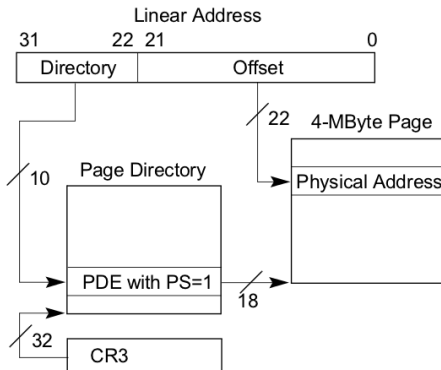


Figura: Traducción de direcciones virtuales a físicas con páginas de 4-MByte. Paginación de un solo nivel (Arquitectura IA-32).

Paginación

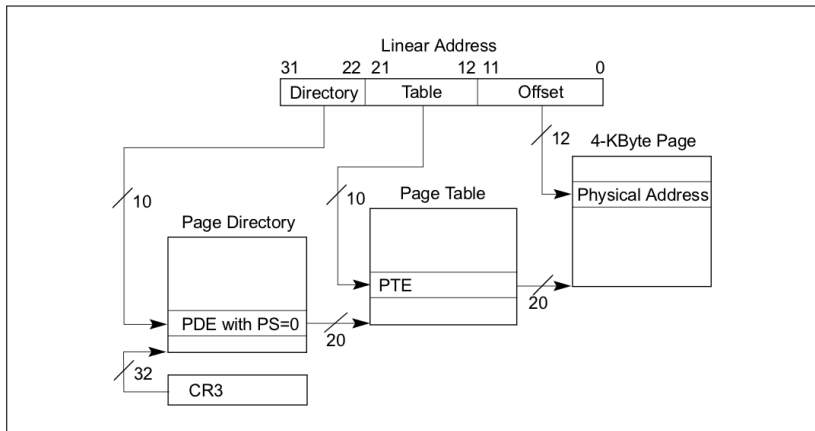


Figura: Traducción de direcciones virtuales a físicas con páginas de 4-KByte. Paginación de dos niveles (Arquitectura IA-32).

Fallo de página (Page Fault)

Cuando el proceso intenta acceder a un dato o instrucción ubicados en una página que no está presente en la memoria principal (Present Bit = 0), el hardware de paginación al tratar de traducir la dirección detecta que la página no se encuentra en MP y genera un trap para invocar al Sistema Operativo. El Sistema Operativo cargará la página faltante y actualizará la tabla de paginas correspondiente.

Paginación

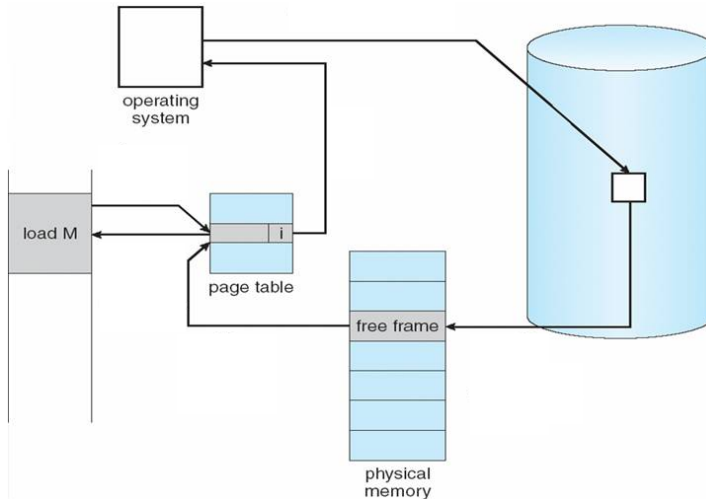


Figura: Gestión de un fallo de página

Paginación

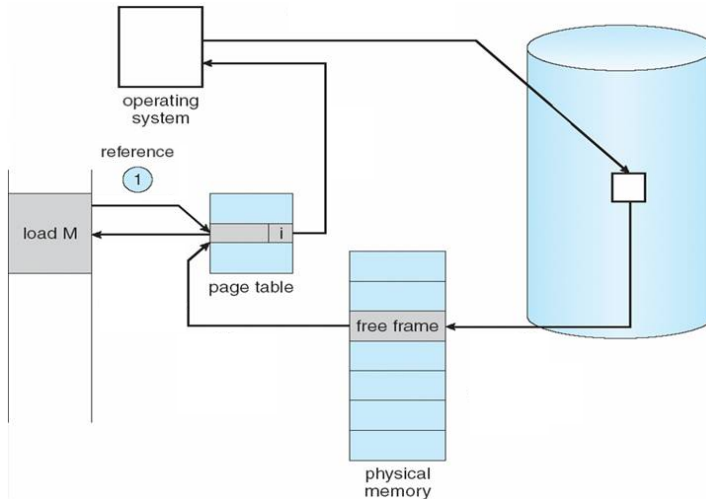


Figura: Gestión de un fallo de página

Paginación

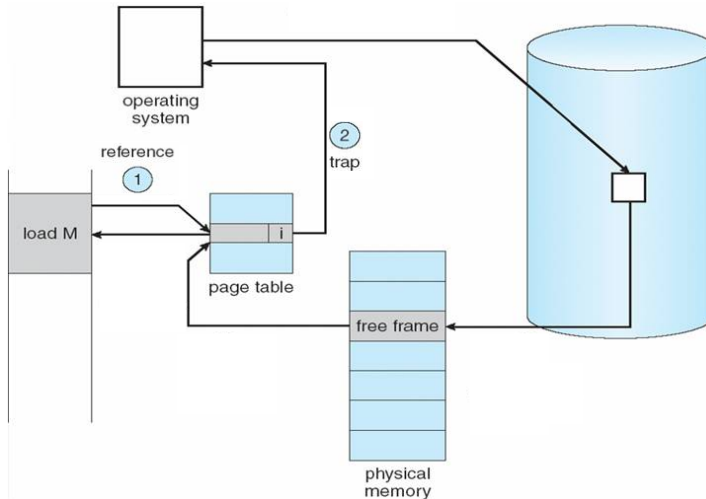


Figura: Gestión de un fallo de página

Paginación

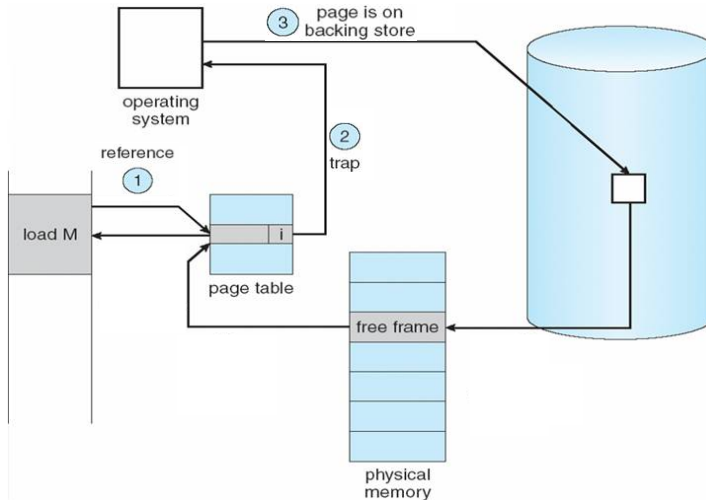


Figura: Gestión de un fallo de página

Paginación

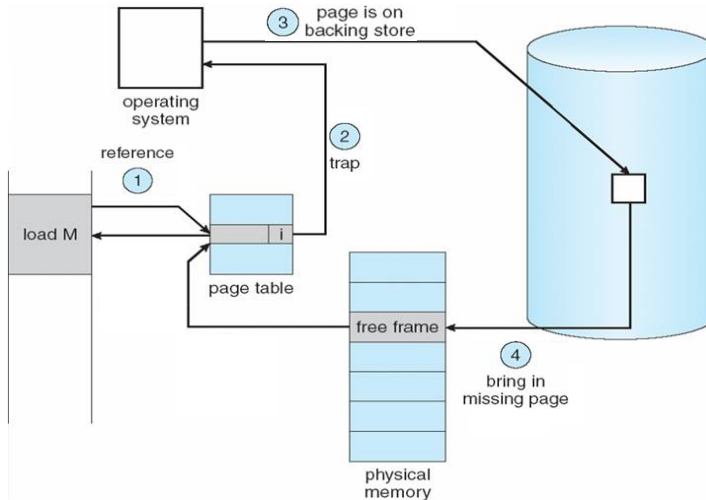


Figura: Gestión de un fallo de página

Paginación

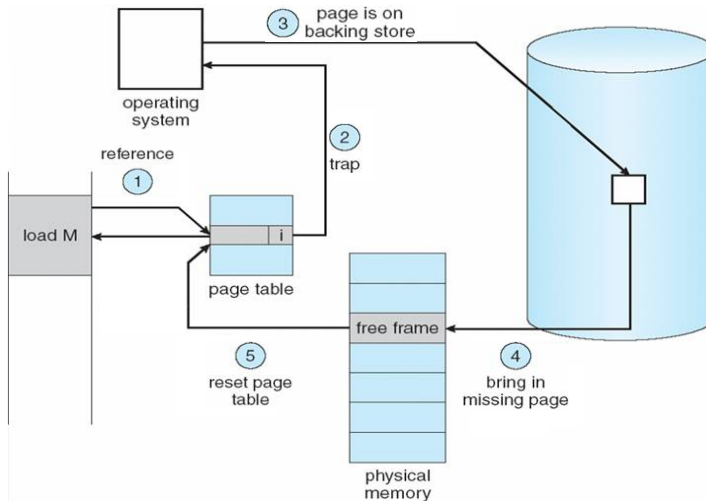


Figura: Gestión de un fallo de página

Paginación

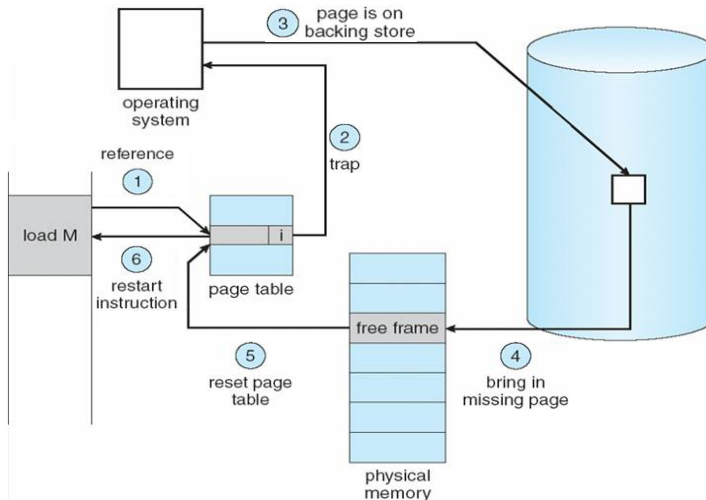


Figura: Gestión de un fallo de página

Tabla de Páginas Invertidas

En este esquema, la porción del número de página en una dirección virtual se mapea en una tabla de hash utilizando una función de hash. La tabla de hash contiene el puntero a la tabla de páginas invertidas, que a su vez contiene la PTE correspondiente.

Paginación

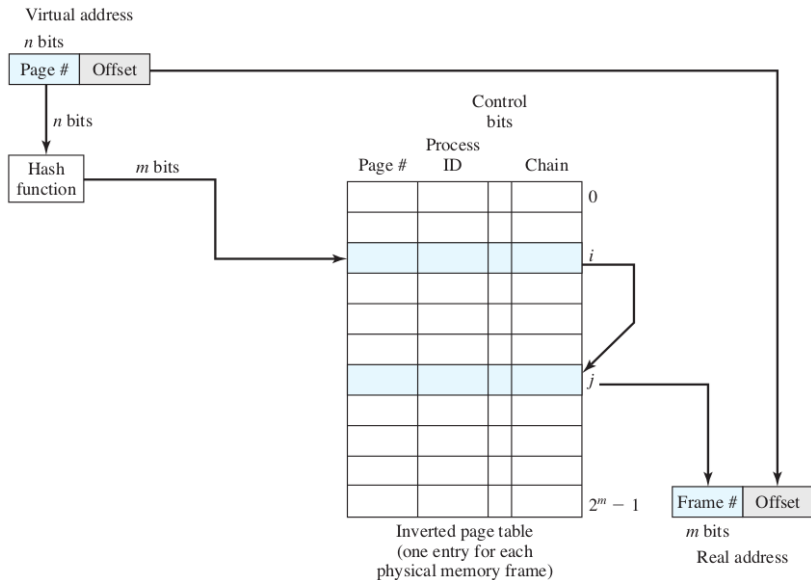
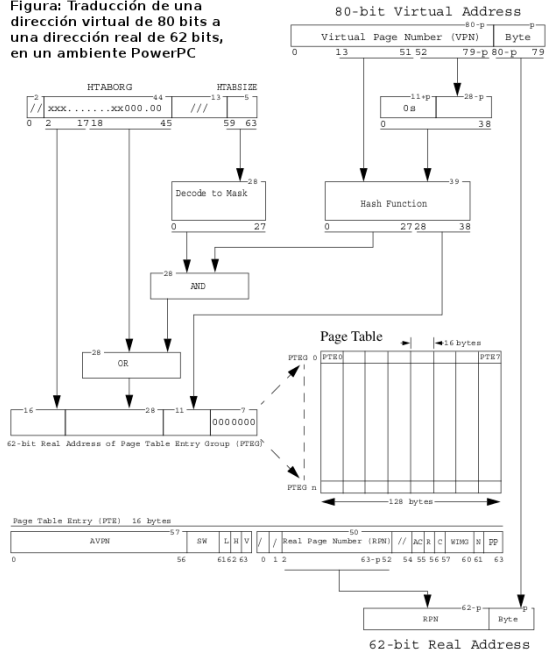


Figura: Traducción de una dirección virtual de 80 bits a una dirección real de 62 bits, en un ambiente PowerPC



TLB: *Translation Lookaside Buffer*

Un procesador puede mantener información relativa a la traducción de direcciones virtuales en Buffers de Traducción Anticipada (TLBs). En general, los TLBs contienen entradas que mapean los números de páginas con los números de frames.

- Cada entrada en un TLB es una traducción individual.
- Cada entrada en un TLB es referenciada por un número de página.
- Cada entrada en un TLB creada por un procesador se encuentra asociada a un identificador de contexto del proceso (PCID: *Process-context identifier*). En la arquitectura IA-32 este identificador cuenta con 12 bits.

TLB: *Translation Lookaside Buffer*

Un procesador puede mantener información relativa a la traducción de direcciones virtuales en Buffers de Traducción Anticipada (TLBs). En general, los TLBs contienen entradas que mapean los números de páginas con los números de frames.

- Cada entrada en un TLB es una traducción individual.
- Cada entrada en un TLB es referenciada por un número de página.
- Cada entrada en un TLB creada por un procesador se encuentra asociada a un identificador de contexto del proceso (PCID: *Process-context identifier*). En la arquitectura IA-32 este identificador cuenta con 12 bits.

TLB: *Translation Lookaside Buffer*

Un procesador puede mantener información relativa a la traducción de direcciones virtuales en Buffers de Traducción Anticipada (TLBs). En general, los TLBs contienen entradas que mapean los números de páginas con los números de frames.

- Cada entrada en un TLB es una traducción individual.
- Cada entrada en un TLB es referenciada por un número de página.
- Cada entrada en un TLB creada por un procesador se encuentra asociada a un identificador de contexto del proceso (PCID: *Process-context identifier*). En la arquitectura IA-32 este identificador cuenta con 12 bits.

Paginación

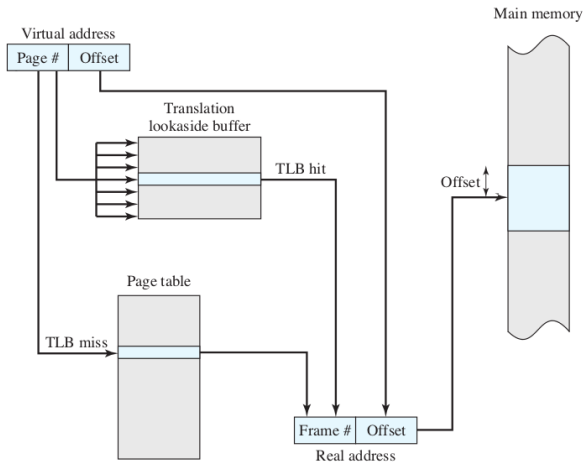


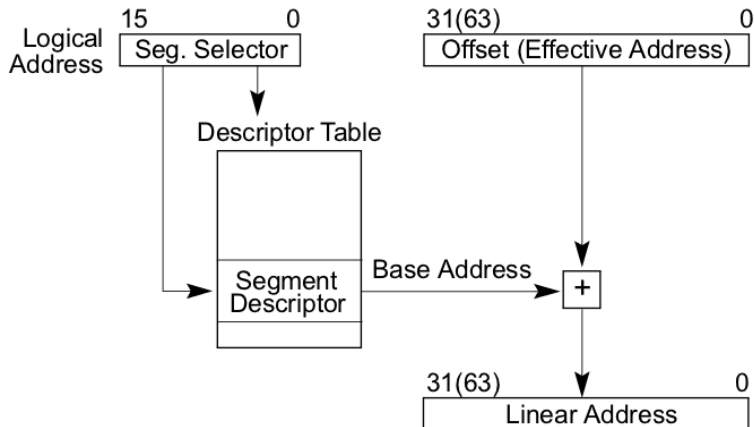
Figura: Uso del TLB

Elementos

- **Tabla de Segmentos:** al igual que en la segmentación simple, se necesitan tablas de segmento (*Segment Descriptor Table*) para los procesos. Cada entrada en esta tabla es un descriptor de segmento, que contiene la información relacionada con el segmento (Dirección base, Límite, Tipo, etc).
- **Bit de presencia:** como solamente algunos segmentos están en memoria principal se necesita un bit para indicarlo (*Present Bit*).
- **Bit de modificación:** indica si el segmento se modificó desde que fue cargado en memoria (*Dirty Bit*).

Segmentación

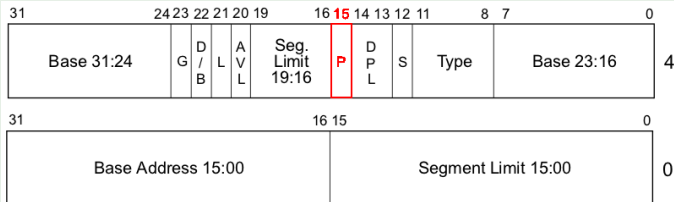
Ejemplo: Traducción de direcciones virtuales a físicas en IA-32



Elementos

- **Tabla de Segmentos:** al igual que en la segmentación simple, se necesitan tablas de segmento (*Segment Descriptor Table*) para los procesos. Cada entrada en esta tabla es un descriptor de segmento, que contiene la información relacionada con el segmento (Dirección base, Límite, Tipo, etc).
- **Bit de presencia:** como solamente algunos segmentos están en memoria principal se necesita un bit para indicarlo (*Present Bit*).
- **Bit de modificación:** indica si el segmento se modificó desde que fue cargado en memoria (*Dirty Bit*).

Ejemplo: Descriptor de Segmento en IA-32



- L — 64-bit code segment (IA-32e mode only)
- AVL — Available for use by system software
- BASE — Segment base address
- D/B — Default operation size (0 = 16-bit segment; 1 = 32-bit segment)
- DPL — Descriptor privilege level
- G — Granularity
- LIMIT — Segment Limit
- P — Segment present
- S — Descriptor type (0 = system; 1 = code or data)
- TYPE — Segment type

Elementos

- **Tabla de Segmentos:** al igual que en la segmentación simple, se necesitan tablas de segmento (*Segment Descriptor Table*) para los procesos. Cada entrada en esta tabla es un descriptor de segmento, que contiene la información relacionada con el segmento (Dirección base, Límite, Tipo, etc).
- **Bit de presencia:** como solamente algunos segmentos están en memoria principal se necesita un bit para indicarlo (*Present Bit*).
- **Bit de modificación:** indica si el segmento se modificó desde que fue cargado en memoria (*Dirty Bit*).

Paginación con Segmentación

- Combina las ventajas del esquema segmentado puro y del esquema paginado puro.
- El esquema paginado, que es transparente al programador, elimina la fragmentación externa y proporciona un uso eficiente de la memoria.
- El esquema segmentado, que es visible para el programador, tiene la capacidad de administrar estructuras de datos que pueden crecer, modularidad y soporte para particiones y protección.

Paginación con Segmentación

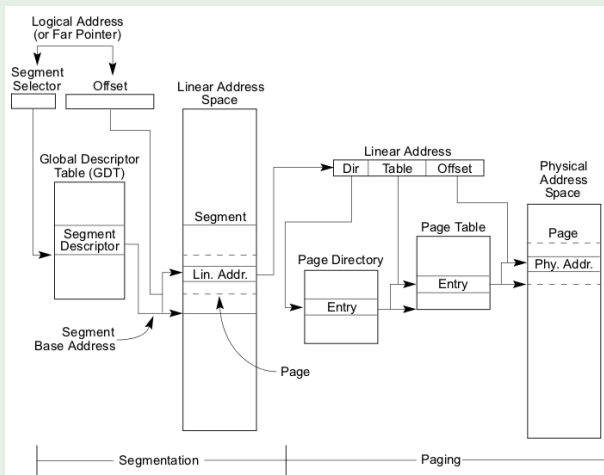
- Combina las ventajas del esquema segmentado puro y del esquema paginado puro.
- El esquema paginado, que es transparente al programador, elimina la fragmentación externa y proporciona un uso eficiente de la memoria.
- El esquema segmentado, que es visible para el programador, tiene la capacidad de administrar estructuras de datos que pueden crecer, modularidad y soporte para particiones y protección.

Paginación con Segmentación

- Combina las ventajas del esquema segmentado puro y del esquema paginado puro.
- El esquema paginado, que es transparente al programador, elimina la fragmentación externa y proporciona un uso eficiente de la memoria.
- El esquema segmentado, que es visible para el programador, tiene la capacidad de administrar estructuras de datos que pueden crecer, modularidad y soporte para particiones y protección.

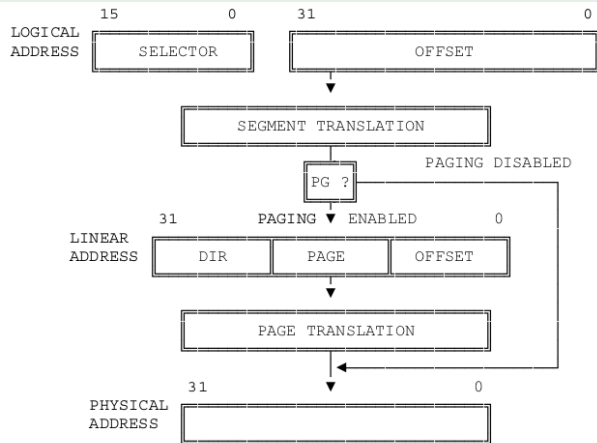
Paginación con Segmentación

Ejemplo: Segmentación y paginación en IA-32



Paginación con Segmentación

Ejemplo: Proceso de traducción en IA-32



Administrador de Memoria

Elementos de diseño en la implementación del software del administrador de memoria del sistema operativo:

- **Política de Carga**
- Política de Ubicación
- Política de Reemplazo
- Administración del Resident Set
- Política de Limpieza
- Control de Carga

Administrador de Memoria

Elementos de diseño en la implementación del software del administrador de memoria del sistema operativo:

- Política de Carga
- **Política de Ubicación**
- Política de Reemplazo
- Administración del Resident Set
- Política de Limpieza
- Control de Carga

Administrador de Memoria

Elementos de diseño en la implementación del software del administrador de memoria del sistema operativo:

- Política de Carga
- Política de Ubicación
- **Política de Reemplazo**
- Administración del Resident Set
- Política de Limpieza
- Control de Carga

Administrador de Memoria

Elementos de diseño en la implementación del software del administrador de memoria del sistema operativo:

- Política de Carga
- Política de Ubicación
- Política de Reemplazo
- Administración del Resident Set
- Política de Limpieza
- Control de Carga

Administrador de Memoria

Elementos de diseño en la implementación del software del administrador de memoria del sistema operativo:

- Política de Carga
- Política de Ubicación
- Política de Reemplazo
- Administración del Resident Set
- Política de Limpieza
- Control de Carga

Administrador de Memoria

Elementos de diseño en la implementación del software del administrador de memoria del sistema operativo:

- Política de Carga
- Política de Ubicación
- Política de Reemplazo
- Administración del Resident Set
- Política de Limpieza
- **Control de Carga**

Política de Carga

Esta política define cuando debe traerse a memoria una página. Existen dos alternativas comunes: paginación por demanda (*Demand Paging*) y pre-paginación (*Prepaging*).

Paginación por demanda Una página se trae a memoria solamente cuando se la referencia.

Pre-paginación Además de la página referenciada (y que produjo un fallo de página) se cargan otras páginas, aprovechando las características de los dispositivos de memoria secundaria.

Esta política define cuando debe traerse a memoria una página. Existen dos alternativas comunes: paginación por demanda (*Demand Paging*) y pre-paginación (*Prepaging*).

Paginación por demanda Una página se trae a memoria solamente cuando se la referencia.

Pre-paginación Además de la página referenciada (y que produjo un fallo de página) se cargan otras páginas, aprovechando las características de los dispositivos de memoria secundaria.

Política de Carga

Esta política define cuando debe traerse a memoria una página. Existen dos alternativas comunes: paginación por demanda (*Demand Paging*) y pre-paginación (*Prepaging*).

Paginación por demanda Una página se trae a memoria solamente cuando se la referencia.

Pre-paginación Además de la página referenciada (y que produjo un fallo de página) se cargan otras páginas, aprovechando las características de los dispositivos de memoria secundaria.

Política de Ubicación

Esta política determina en que lugar de la memoria real se debe cargar la porción del programa faltante. En el caso de usar paginación pura o segmentación con paginación se vuelve irrelevante.

En un sistema que utiliza segmentación pura la política de ubicación es un elemento importante de diseño. Se pueden utilizar los algoritmos best-fit, first-fit, next-fit o worst-fit.

Política de Reemplazo

Entra en funcionamiento cuando ocurre un fallo de página y se necesita reemplazar la página de un marco por la faltante. Involucra diferentes conceptos:

- Cuántos marcos de página se le asigna a un proceso.
- Cuáles páginas se considerarán para realizar el reemplazo. Solamente las del proceso que produjo el fallo o las de todos los procesos.
- Finalmente, de entre todas las páginas candidatas, cuál se reemplazará.

Política de Reemplazo

Entra en funcionamiento cuando ocurre un fallo de página y se necesita reemplazar la página de un marco por la faltante. Involucra diferentes conceptos:

- Cuántos marcos de página se le asigna a un proceso.
- Cuáles páginas se considerarán para realizar el reemplazo. Solamente las del proceso que produjo el fallo o las de todos los procesos.
- Finalmente, de entre todas las páginas candidatas, cuál se reemplazará.

Política de Reemplazo

Entra en funcionamiento cuando ocurre un fallo de página y se necesita reemplazar la página de un marco por la faltante. Involucra diferentes conceptos:

- Cuántos marcos de página se le asigna a un proceso.
- Cuáles páginas se considerarán para realizar el reemplazo. Solamente las del proceso que produjo el fallo o las de todos los procesos.
- Finalmente, de entre todas las páginas candidatas, cuál se reemplazará.

Política de Reemplazo

Entra en funcionamiento cuando ocurre un fallo de página y se necesita reemplazar la página de un marco por la faltante. Involucra diferentes conceptos:

- Cuántos marcos de página se le asigna a un proceso.
- Cuáles páginas se considerarán para realizar el reemplazo. Solamente las del proceso que produjo el fallo o las de todos los procesos.
- Finalmente, de entre todos las páginas candidatas, cuál se reemplazará.

Algoritmos de reemplazo de páginas

- Óptimo (OPT: *Optimal*)
- Menos recientemente usado (LRU: *Least Recently Used*)
- Primero en entrar primero en salir (FIFO: *First In First Out*)
- Reloj (*Clock* ó *Second-Chance*)
- Reloj Mejorado (*Enhanced Clock* ó *Enhanced Second-Chance*)

Óptimo (OPT)

Este algoritmo selecciona para reemplazar la página cuya próxima referencia será la más alejada en el tiempo. Es el que menos fallos de página produce, pero no es implementable.

	0	1	2	3	0	1	4	0	1	2	3	4
M1												
M2												
M3												

Óptimo (OPT)

Este algoritmo selecciona para reemplazar la página cuya próxima referencia será la más alejada en el tiempo. Es el que menos fallos de página produce, pero no es implementable.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0											
M2												
M3												
	F											

Óptimo (OPT)

Este algoritmo selecciona para reemplazar la página cuya próxima referencia será la más alejada en el tiempo. Es el que menos fallos de página produce, pero no es implementable.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0										
M2		1										
M3												
	F	F										

Óptimo (OPT)

Este algoritmo selecciona para reemplazar la página cuya próxima referencia será la más alejada en el tiempo. Es el que menos fallos de página produce, pero no es implementable.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0									
M2		1	1									
M3			2									
	F	F	F									

Óptimo (OPT)

Este algoritmo selecciona para reemplazar la página cuya próxima referencia será la más alejada en el tiempo. Es el que menos fallos de página produce, pero no es implementable.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	0								
M2		1	1	1								
M3			2	3								
	F	F	F	F								

Óptimo (OPT)

Este algoritmo selecciona para reemplazar la página cuya próxima referencia será la más alejada en el tiempo. Es el que menos fallos de página produce, pero no es implementable.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	0	0							
M2		1	1	1	1							
M3			2	3	3							
	F	F	F	F								

Óptimo (OPT)

Este algoritmo selecciona para reemplazar la página cuya próxima referencia será la más alejada en el tiempo. Es el que menos fallos de página produce, pero no es implementable.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	0	0	0						
M2		1	1	1	1	1						
M3			2	3	3	3						
	F	F	F	F								

Óptimo (OPT)

Este algoritmo selecciona para reemplazar la página cuya próxima referencia será la más alejada en el tiempo. Es el que menos fallos de página produce, pero no es implementable.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	0	0	0	0					
M2		1	1	1	1	1	1					
M3			2	3	3	3	4					
	F	F	F	F			F					

Óptimo (OPT)

Este algoritmo selecciona para reemplazar la página cuya próxima referencia será la más alejada en el tiempo. Es el que menos fallos de página produce, pero no es implementable.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	0	0	0	0	0				
M2		1	1	1	1	1	1	1				
M3			2	3	3	3	4	4				
	F	F	F	F			F					

Óptimo (OPT)

Este algoritmo selecciona para reemplazar la página cuya próxima referencia será la más alejada en el tiempo. Es el que menos fallos de página produce, pero no es implementable.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	0	0	0	0	0	0			
M2		1	1	1	1	1	1	1	1			
M3			2	3	3	3	4	4	4			
	F	F	F	F			F					

Óptimo (OPT)

Este algoritmo selecciona para reemplazar la página cuya próxima referencia será la más alejada en el tiempo. Es el que menos fallos de página produce, pero no es implementable.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	0	0	0	0	0	0	2		
M2		1	1	1	1	1	1	1	1	1		
M3			2	3	3	3	4	4	4	4		
	F	F	F	F			F			F		

Óptimo (OPT)

Este algoritmo selecciona para reemplazar la página cuya próxima referencia será la más alejada en el tiempo. Es el que menos fallos de página produce, pero no es implementable.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	0	0	0	0	0	0	2	3	
M2		1	1	1	1	1	1	1	1	1	1	
M3			2	3	3	3	4	4	4	4	4	
	F	F	F	F			F			F	F	

Óptimo (OPT)

Este algoritmo selecciona para reemplazar la página cuya próxima referencia será la más alejada en el tiempo. Es el que menos fallos de página produce, pero no es implementable.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	0	0	0	0	0	0	2	3	3
M2		1	1	1	1	1	1	1	1	1	1	1
M3			2	3	3	3	4	4	4	4	4	4
	F	F	F	F			F			F	F	

Óptimo (OPT)

Este algoritmo selecciona para reemplazar la página cuya próxima referencia será la más alejada en el tiempo. Es el que menos fallos de página produce, pero no es implementable.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	0	0	0	0	0	0	2	3	3
M2		1	1	1	1	1	1	1	1	1	1	1
M3			2	3	3	3	4	4	4	4	4	4
	F	F	F	F			F			F	F	

Total de fallos de página: 7

Menos Recientemente Usado (LRU)

Este algoritmo selecciona para reemplazar la página cuya última referencia fue la más alejada en el tiempo. Se basa en el principio de localidad referencial, pero es difícil de implementar.

	0	1	2	3	0	1	4	0	1	2	3	4
M1												
M2												
M3												

Menos Recientemente Usado (LRU)

Este algoritmo selecciona para reemplazar la página cuya última referencia fue la más alejada en el tiempo. Se basa en el principio de localidad referencial, pero es difícil de implementar.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0											
M2												
M3												
	F											

Menos Recientemente Usado (LRU)

Este algoritmo selecciona para reemplazar la página cuya última referencia fue la más alejada en el tiempo. Se basa en el principio de localidad referencial, pero es difícil de implementar.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0										
M2		1										
M3												
	F	F										

Menos Recientemente Usado (LRU)

Este algoritmo selecciona para reemplazar la página cuya última referencia fue la más alejada en el tiempo. Se basa en el principio de localidad referencial, pero es difícil de implementar.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0									
M2		1	1									
M3			2									
	F	F	F									

Menos Recientemente Usado (LRU)

Este algoritmo selecciona para reemplazar la página cuya última referencia fue la más alejada en el tiempo. Se basa en el principio de localidad referencial, pero es difícil de implementar.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	3								
M2		1	1	1								
M3			2	2								
	F	F	F	F								

Menos Recientemente Usado (LRU)

Este algoritmo selecciona para reemplazar la página cuya última referencia fue la más alejada en el tiempo. Se basa en el principio de localidad referencial, pero es difícil de implementar.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	3	3							
M2		1	1	1	0							
M3			2	2	2							
	F	F	F	F	F							

Menos Recientemente Usado (LRU)

Este algoritmo selecciona para reemplazar la página cuya última referencia fue la más alejada en el tiempo. Se basa en el principio de localidad referencial, pero es difícil de implementar.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	3	3	3						
M2		1	1	1	0	0						
M3			2	2	2	1						
	F	F	F	F	F	F						

Menos Recientemente Usado (LRU)

Este algoritmo selecciona para reemplazar la página cuya última referencia fue la más alejada en el tiempo. Se basa en el principio de localidad referencial, pero es difícil de implementar.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	3	3	3	4					
M2		1	1	1	0	0	0					
M3			2	2	2	1	1					
	F	F	F	F	F	F	F					

Menos Recientemente Usado (LRU)

Este algoritmo selecciona para reemplazar la página cuya última referencia fue la más alejada en el tiempo. Se basa en el principio de localidad referencial, pero es difícil de implementar.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	3	3	3	4	4				
M2		1	1	1	0	0	0	0				
M3			2	2	2	1	1	1				
	F	F	F	F	F	F	F					

Menos Recientemente Usado (LRU)

Este algoritmo selecciona para reemplazar la página cuya última referencia fue la más alejada en el tiempo. Se basa en el principio de localidad referencial, pero es difícil de implementar.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	3	3	3	4	4	4			
M2		1	1	1	0	0	0	0	0			
M3			2	2	2	1	1	1	1			
	F	F	F	F	F	F	F					

Menos Recientemente Usado (LRU)

Este algoritmo selecciona para reemplazar la página cuya última referencia fue la más alejada en el tiempo. Se basa en el principio de localidad referencial, pero es difícil de implementar.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	3	3	3	4	4	4	2		
M2		1	1	1	0	0	0	0	0	0		
M3			2	2	2	1	1	1	1	1		
	F	F	F	F	F	F	F			F		

Menos Recientemente Usado (LRU)

Este algoritmo selecciona para reemplazar la página cuya última referencia fue la más alejada en el tiempo. Se basa en el principio de localidad referencial, pero es difícil de implementar.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	3	3	3	4	4	4	2	2	
M2		1	1	1	0	0	0	0	0	0	3	
M3			2	2	2	1	1	1	1	1	1	
	F	F	F	F	F	F	F			F	F	

Menos Recientemente Usado (LRU)

Este algoritmo selecciona para reemplazar la página cuya última referencia fue la más alejada en el tiempo. Se basa en el principio de localidad referencial, pero es difícil de implementar.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	3	3	3	4	4	4	2	2	2
M2		1	1	1	0	0	0	0	0	0	3	3
M3			2	2	2	1	1	1	1	1	1	4
	F	F	F	F	F	F	F			F	F	F

Menos Recientemente Usado (LRU)

Este algoritmo selecciona para reemplazar la página cuya última referencia fue la más alejada en el tiempo. Se basa en el principio de localidad referencial, pero es difícil de implementar.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	3	3	3	4	4	4	2	2	2
M2		1	1	1	0	0	0	0	0	0	3	3
M3			2	2	2	1	1	1	1	1	1	4
	F	F	F	F	F	F	F			F	F	F

Total de fallos de página: 10

Primero en Entrar Primero en Salir (FIFO)

Este algoritmo selecciona para reemplazar la página ingresó primero a la memoria. Es facil de implementar ya que trata los marcos como una cola circular. No tiene buen desempeño y además sufre la Anomalía de Belady.

	0	1	2	3	0	1	4	0	1	2	3	4
M1												
M2												
M3												

Primero en Entrar Primero en Salir (FIFO)

Este algoritmo selecciona para reemplazar la página ingresó primero a la memoria. Es facil de implementar ya que trata los marcos como una cola circular. No tiene buen desempeño y además sufre la Anomalía de Belady.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0											
M2												
M3												
	F											

Primero en Entrar Primero en Salir (FIFO)

Este algoritmo selecciona para reemplazar la página ingresó primero a la memoria. Es facil de implementar ya que trata los marcos como una cola circular. No tiene buen desempeño y además sufre la Anomalía de Belady.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0										
M2		1										
M3												
	F	F										

Primero en Entrar Primero en Salir (FIFO)

Este algoritmo selecciona para reemplazar la página ingresó primero a la memoria. Es facil de implementar ya que trata los marcos como una cola circular. No tiene buen desempeño y además sufre la Anomalía de Belady.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0									
M2		1	1									
M3			2									
	F	F	F									

Primero en Entrar Primero en Salir (FIFO)

Este algoritmo selecciona para reemplazar la página ingresó primero a la memoria. Es facil de implementar ya que trata los marcos como una cola circular. No tiene buen desempeño y además sufre la Anomalía de Belady.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	3								
M2		1	1	1								
M3			2	2								
	F	F	F	F								

Primero en Entrar Primero en Salir (FIFO)

Este algoritmo selecciona para reemplazar la página ingresó primero a la memoria. Es facil de implementar ya que trata los marcos como una cola circular. No tiene buen desempeño y además sufre la Anomalía de Belady.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	3	3							
M2		1	1	1	0							
M3			2	2	2							
	F	F	F	F	F							

Primero en Entrar Primero en Salir (FIFO)

Este algoritmo selecciona para reemplazar la página ingresó primero a la memoria. Es facil de implementar ya que trata los marcos como una cola circular. No tiene buen desempeño y además sufre la Anomalía de Belady.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	3	3	3						
M2		1	1	1	0	0						
M3			2	2	2	1						
	F	F	F	F	F	F						

Primero en Entrar Primero en Salir (FIFO)

Este algoritmo selecciona para reemplazar la página ingresó primero a la memoria. Es facil de implementar ya que trata los marcos como una cola circular. No tiene buen desempeño y además sufre la Anomalía de Belady.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	3	3	3	4					
M2		1	1	1	0	0	0					
M3			2	2	2	1	1					
	F	F	F	F	F	F	F					

Primero en Entrar Primero en Salir (FIFO)

Este algoritmo selecciona para reemplazar la página ingresó primero a la memoria. Es facil de implementar ya que trata los marcos como una cola circular. No tiene buen desempeño y además sufre la Anomalía de Belady.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	3	3	3	4	4				
M2		1	1	1	0	0	0	0				
M3			2	2	2	1	1	1				
	F	F	F	F	F	F	F					

Primero en Entrar Primero en Salir (FIFO)

Este algoritmo selecciona para reemplazar la página ingresó primero a la memoria. Es facil de implementar ya que trata los marcos como una cola circular. No tiene buen desempeño y además sufre la Anomalía de Belady.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	3	3	3	4	4	4			
M2		1	1	1	0	0	0	0	0			
M3			2	2	2	1	1	1	1			
	F	F	F	F	F	F	F					

Primero en Entrar Primero en Salir (FIFO)

Este algoritmo selecciona para reemplazar la página ingresó primero a la memoria. Es facil de implementar ya que trata los marcos como una cola circular. No tiene buen desempeño y además sufre la Anomalía de Belady.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	3	3	3	4	4	4	4		
M2		1	1	1	0	0	0	0	0	2		
M3			2	2	2	1	1	1	1	1		
	F	F	F	F	F	F	F			F		

Primero en Entrar Primero en Salir (FIFO)

Este algoritmo selecciona para reemplazar la página ingresó primero a la memoria. Es facil de implementar ya que trata los marcos como una cola circular. No tiene buen desempeño y además sufre la Anomalía de Belady.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	3	3	3	4	4	4	4	4	
M2		1	1	1	0	0	0	0	0	2	2	
M3			2	2	2	1	1	1	1	1	3	
	F	F	F	F	F	F	F			F	F	

Primero en Entrar Primero en Salir (FIFO)

Este algoritmo selecciona para reemplazar la página ingresó primero a la memoria. Es facil de implementar ya que trata los marcos como una cola circular. No tiene buen desempeño y además sufre la Anomalía de Belady.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	3	3	3	4	4	4	4	4	4
M2		1	1	1	0	0	0	0	0	2	2	2
M3			2	2	2	1	1	1	1	1	3	3
	F	F	F	F	F	F	F			F	F	

Primero en Entrar Primero en Salir (FIFO)

Este algoritmo selecciona para reemplazar la página ingresó primero a la memoria. Es facil de implementar ya que trata los marcos como una cola circular. No tiene buen desempeño y además sufre la Anomalía de Belady.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	3	3	3	4	4	4	4	4	4
M2		1	1	1	0	0	0	0	0	2	2	2
M3			2	2	2	1	1	1	1	1	3	3
	F	F	F	F	F	F	F			F	F	

Total de fallos de página: 9

Anomalía de Belady

Situación en la que un algoritmo de reemplazo de páginas produce más fallos de páginas cuando el proceso posee más marcos de página asignados.

	0	1	2	3	0	1	4	0	1	2	3	4
M1												
M2												
M3												
M4												

Anomalía de Belady

Situación en la que un algoritmo de reemplazo de páginas produce más fallos de páginas cuando el proceso posee más marcos de página asignados.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0											
M2												
M3												
M4												
	F											

Anomalía de Belady

Situación en la que un algoritmo de reemplazo de páginas produce más fallos de páginas cuando el proceso posee más marcos de página asignados.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0										
M2		1										
M3												
M4												
	F	F										

Anomalía de Belady

Situación en la que un algoritmo de reemplazo de páginas produce más fallos de páginas cuando el proceso posee más marcos de página asignados.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0									
M2		1	1									
M3			2									
M4												
	F	F	F									

Anomalía de Belady

Situación en la que un algoritmo de reemplazo de páginas produce más fallos de páginas cuando el proceso posee más marcos de página asignados.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	0								
M2		1	1	1								
M3			2	2								
M4				3								
	F	F	F	F								

Anomalía de Belady

Situación en la que un algoritmo de reemplazo de páginas produce más fallos de páginas cuando el proceso posee más marcos de página asignados.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	0	0							
M2		1	1	1	1							
M3			2	2	2							
M4				3	3							
	F	F	F	F								

Anomalía de Belady

Situación en la que un algoritmo de reemplazo de páginas produce más fallos de páginas cuando el proceso posee más marcos de página asignados.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	0	0	0						
M2		1	1	1	1	1						
M3			2	2	2	2						
M4				3	3	3						
	F	F	F	F								

Anomalía de Belady

Situación en la que un algoritmo de reemplazo de páginas produce más fallos de páginas cuando el proceso posee más marcos de página asignados.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	0	0	0	4					
M2		1	1	1	1	1	1					
M3			2	2	2	2	2					
M4				3	3	3	3					
	F	F	F	F			F					

Anomalía de Belady

Situación en la que un algoritmo de reemplazo de páginas produce más fallos de páginas cuando el proceso posee más marcos de página asignados.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	0	0	0	4	4				
M2		1	1	1	1	1	1	0				
M3			2	2	2	2	2	2				
M4				3	3	3	3	3				
	F	F	F	F			F	F				

Anomalía de Belady

Situación en la que un algoritmo de reemplazo de páginas produce más fallos de páginas cuando el proceso posee más marcos de página asignados.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	0	0	0	4	4	4			
M2		1	1	1	1	1	1	0	0			
M3			2	2	2	2	2	2	1			
M4				3	3	3	3	3	3			
	F	F	F	F			F	F	F			

Anomalía de Belady

Situación en la que un algoritmo de reemplazo de páginas produce más fallos de páginas cuando el proceso posee más marcos de página asignados.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	0	0	0	4	4	4	4		
M2		1	1	1	1	1	1	0	0	0		
M3			2	2	2	2	2	2	1	1		
M4				3	3	3	3	3	3	2		
	F	F	F	F			F	F	F	F		

Anomalía de Belady

Situación en la que un algoritmo de reemplazo de páginas produce más fallos de páginas cuando el proceso posee más marcos de página asignados.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	0	0	0	4	4	4	4	3	
M2		1	1	1	1	1	1	0	0	0	0	
M3			2	2	2	2	2	2	1	1	1	
M4				3	3	3	3	3	3	2	2	
	F	F	F	F			F	F	F	F	F	

Anomalía de Belady

Situación en la que un algoritmo de reemplazo de páginas produce más fallos de páginas cuando el proceso posee más marcos de página asignados.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	0	0	0	4	4	4	4	3	3
M2		1	1	1	1	1	1	0	0	0	0	4
M3			2	2	2	2	2	2	1	1	1	1
M4				3	3	3	3	3	3	2	2	2
	F	F	F	F			F	F	F	F	F	F

Anomalía de Belady

Situación en la que un algoritmo de reemplazo de páginas produce más fallos de páginas cuando el proceso posee más marcos de página asignados.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0	0	0	0	0	0	4	4	4	4	3	3
M2		1	1	1	1	1	1	0	0	0	0	4
M3			2	2	2	2	2	2	1	1	1	1
M4				3	3	3	3	3	3	2	2	2
	F	F	F	F			F	F	F	F	F	F

Total de fallos de página: 10

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	▶											
M2												
M3												

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁											
M2	▶											
M3												
	F											

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁										
M2		1 ₁										
M3		▶										
	F	F										

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	▶ 0 ₁									
M2		1 ₁	1 ₁									
M3			2 ₁									
	F	F	F									

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	▶ 0 ₁									
M2		1 ₁	1 ₁									
M3			2 ₁									
	F	F	F	F								

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	0 ₀									
M2		1 ₁	▶ 1 ₁									
M3			2 ₁									
	F	F	F	F								

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	0 ₀									
M2		1 ₁	1 ₀									
M3			▶ 2 ₁									
	F	F	F	F								

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	▶ 0 ₀									
M2		1 ₁	1 ₀									
M3			2 ₀									
	F	F	F	F								

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	0 ₀	3 ₁								
M2		1 ₁	1 ₀	▶ 1 ₀								
M3			2 ₀	2 ₀								
	F	F	F	F								

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	0 ₀	3 ₁	3							
M2		1 ₁	1 ₀	1 ₀	0 ₁							
M3			2 ₀	2 ₀	▶ 2 ₀							
	F	F	F	F	F							

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	0 ₀	3 ₁	3 ₁	▶ 3 ₁						
M2		1 ₁	1 ₀	1 ₀	0 ₁	0 ₁						
M3			2 ₀	2 ₀	2 ₀	1 ₁						
	F	F	F	F	F	F						

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	0 ₀	3 ₁	3 ₁	▶ 3 ₁						
M2		1 ₁	1 ₀	1 ₀	0 ₁	0 ₁						
M3			2 ₀	2 ₀	2 ₀	1 ₁						
	F	F	F	F	F	F	F					

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	0 ₀	3 ₁	3 ₁	3 ₀						
M2		1 ₁	1 ₀	1 ₀	0 ₁	▶ 0 ₁						
M3			2 ₀	2 ₀	2 ₀	1 ₁						
	F	F	F	F	F	F	F					

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	0 ₀	3 ₁	3 ₁	3 ₀						
M2		1 ₁	1 ₀	1 ₀	0 ₁	0 ₀						
M3			2 ₀	2 ₀	2 ₀	▶ 1 ₁						
	F	F	F	F	F	F	F					

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	0 ₀	3 ₁	3 ₁	▶ 3 ₀						
M2		1 ₁	1 ₀	1 ₀	0 ₁	0 ₀						
M3			2 ₀	2 ₀	2 ₀	1 ₀						
	F	F	F	F	F	F	F					

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	0 ₀	3 ₁	3 ₁	3 ₀	4 ₁					
M2		1 ₁	1 ₀	1 ₀	0 ₁	0 ₀	▶ 0 ₀					
M3			2 ₀	2 ₀	2 ₀	1 ₀	1 ₀					
	F	F	F	F	F	F	F					

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	0 ₀	3 ₁	3 ₁	3 ₀	4 ₁	4 ₁				
M2		1 ₁	1 ₀	1 ₀	0 ₁	0 ₀	0 ₀	▶ 0 ₁				
M3			2 ₀	2 ₀	2 ₀	1 ₀	1 ₀	1 ₀				
	F	F	F	F	F	F	F					

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	0 ₀	3 ₁	3 ₁	3 ₀	4 ₁	4 ₁	4 ₁			
M2		1 ₁	1 ₀	1 ₀	0 ₁	0 ₀	0 ₀	0 ₁	▶ 0 ₁			
M3			2 ₀	2 ₀	2 ₀	1 ₀	1 ₀	1 ₀	1 ₁			
	F	F	F	F	F	F	F					

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	0 ₀	3 ₁	3 ₁	3 ₀	4 ₁	4 ₁	4 ₁			
M2		1 ₁	1 ₀	1 ₀	0 ₁	0 ₀	0 ₀	0 ₁	▶ 0 ₁			
M3			2 ₀	2 ₀	2 ₀	1 ₀	1 ₀	1 ₀	1 ₁			
	F	F	F	F	F	F	F			F		

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	0 ₀	3 ₁	3 ₁	3 ₀	4 ₁	4 ₁	4 ₁			
M2		1 ₁	1 ₀	1 ₀	0 ₁	0 ₀	0 ₀	0 ₁	0 ₀			
M3			2 ₀	2 ₀	2 ₀	1 ₀	1 ₀	1 ₀	▶ 1 ₁			
	F	F	F	F	F	F	F			F		

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	0 ₀	3 ₁	3 ₁	3 ₀	4 ₁	4 ₁	▶ 4 ₁			
M2		1 ₁	1 ₀	1 ₀	0 ₁	0 ₀	0 ₀	0 ₁	0 ₀			
M3			2 ₀	2 ₀	2 ₀	1 ₀	1 ₀	1 ₀	1 ₀			
	F	F	F	F	F	F	F			F		

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	0 ₀	3 ₁	3 ₁	3 ₀	4 ₁	4 ₁	4 ₀			
M2		1 ₁	1 ₀	1 ₀	0 ₁	0 ₀	0 ₀	0 ₁	▶ 0 ₀			
M3			2 ₀	2 ₀	2 ₀	1 ₀	1 ₀	1 ₀	1 ₀			
	F	F	F	F	F	F	F			F		

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	0 ₀	3 ₁	3 ₁	3 ₀	4 ₁	4 ₁	4 ₀	4 ₀		
M2		1 ₁	1 ₀	1 ₀	0 ₁	0 ₀	0 ₀	0 ₁	0 ₀	2 ₁		
M3			2 ₀	2 ₀	2 ₀	1 ₀	1 ₀	1 ₀	1 ₀	▶ 1 ₀		
	F	F	F	F	F	F	F			F		

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	0 ₀	3 ₁	3 ₁	3 ₀	4 ₁	4 ₁	4 ₀	4 ₀	▶ 4 ₀	
M2		1 ₁	1 ₀	1 ₀	0 ₁	0 ₀	0 ₀	0 ₁	0 ₀	2 ₁	2 ₁	
M3			2 ₀	2 ₀	2 ₀	1 ₀	1 ₀	1 ₀	1 ₀	1 ₀	3 ₁	
	F	F	F	F	F	F	F			F	F	

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	0 ₀	3 ₁	3 ₁	3 ₀	4 ₁	4 ₁	4 ₀	4 ₀	4 ₀	▶ 4 ₁
M2		1 ₁	1 ₀	1 ₀	0 ₁	0 ₀	0 ₀	0 ₁	0 ₀	2 ₁	2 ₁	2 ₁
M3			2 ₀	2 ₀	2 ₀	1 ₀	1 ₀	1 ₀	1 ₀	1 ₀	3 ₁	3 ₁
	F	F	F	F	F	F	F			F	F	

Clock (Second-Chance)

Utiliza una cola circular de frames y el bit de Acceso. Si el frame tiene el bit de acceso en 1 lo actualiza a 0 y avanza el puntero. Cuando encuentra un frame en 0 reemplaza la página que contiene por la que produjo el fallo de página.

	0	1	2	3	0	1	4	0	1	2	3	4
M1	0 ₁	0 ₁	0 ₀	3 ₁	3 ₁	3 ₀	4 ₁	4 ₁	4 ₀	4 ₀	4 ₀	4 ₁
M2		1 ₁	1 ₀	1 ₀	0 ₁	0 ₀	0 ₀	0 ₁	0 ₀	2 ₁	2 ₁	2 ₁
M3			2 ₀	2 ₀	2 ₀	1 ₀	1 ₀	1 ₀	1 ₀	1 ₀	3 ₁	3 ₁
	F	F	F	F	F	F	F			F	F	

Total de fallos de página: 9

Clock Mejorado I

El algoritmo Clock puede mejorarse agregandole más bits. En este caso usando el bit de modificación (*Dirty Bit*) asociado con cada página. Las páginas modificadas, no pueden reemplazarse hasta que se escriban en memoria secundaria (Disco). El Clock Mejorado explota el uso del bit de modificación generando cuatro clases de frames, de acuerdo a la situación de la página contenida:

- No accedida, no modificada ($A=0;D=0$).
- Accedida, no modificada ($A=1;D=0$).
- No accedida, modificada ($A=0;D=1$).
- Accedida, modificada ($A=1;D=1$).

Clock Mejorado I

El algoritmo Clock puede mejorarse agregandole más bits. En este caso usando el bit de modificación (*Dirty Bit*) asociado con cada página. Las páginas modificadas, no pueden reemplazarse hasta que se escriban en memoria secundaria (Disco). El Clock Mejorado explota el uso del bit de modificación generando cuatro clases de frames, de acuerdo a la situación de la página contenida:

- No accedida, no modificada ($A=0;D=0$).
- Accedida, no modificada ($A=1;D=0$).
- No accedida, modificada ($A=0;D=1$).
- Accedida, modificada ($A=1;D=1$).

Clock Mejorado I

El algoritmo Clock puede mejorarse agregandole más bits. En este caso usando el bit de modificación (*Dirty Bit*) asociado con cada página. Las páginas modificadas, no pueden reemplazarse hasta que se escriban en memoria secundaria (Disco). El Clock Mejorado explota el uso del bit de modificación generando cuatro clases de frames, de acuerdo a la situación de la página contenida:

- No accedida, no modificada ($A=0;D=0$).
- Accedida, no modificada ($A=1;D=0$).
- No accedida, modificada ($A=0;D=1$).
- Accedida, modificada ($A=1;D=1$).

Clock Mejorado I

El algoritmo Clock puede mejorarse agregandole más bits. En este caso usando el bit de modificación (*Dirty Bit*) asociado con cada página. Las páginas modificadas, no pueden reemplazarse hasta que se escriban en memoria secundaria (Disco). El Clock Mejorado explota el uso del bit de modificación generando cuatro clases de frames, de acuerdo a la situación de la página contenida:

- No accedida, no modificada ($A=0;D=0$).
- Accedida, no modificada ($A=1;D=0$).
- No accedida, modificada ($A=0;D=1$).
- Accedida, modificada ($A=1;D=1$).

Clock Mejorado I

El algoritmo Clock puede mejorarse agregandole más bits. En este caso usando el bit de modificación (*Dirty Bit*) asociado con cada página. Las páginas modificadas, no pueden reemplazarse hasta que se escriban en memoria secundaria (Disco). El Clock Mejorado explota el uso del bit de modificación generando cuatro clases de frames, de acuerdo a la situación de la página contenida:

- No accedida, no modificada ($A=0;D=0$).
- Accedida, no modificada ($A=1;D=0$).
- No accedida, modificada ($A=0;D=1$).
- Accedida, modificada ($A=1;D=1$).

Funcionamiento

- 1 Recorra la lista de frames buscando uno con los bits ($A=0;D=0$). En este recorrido NO cambie los bits de uso.

Funcionamiento

- 1 Recorra la lista de frames buscando uno con los bits ($A=0;D=0$). En este recorrido NO cambie los bits de uso.
- 2 Si en el paso 1 no encontró ningún frame, recorra la lista de frames buscando uno con los bits ($A=0;D=1$). Durante este recorrido ponga el bit de uso (A) en 0 a los frames que saltea.

Funcionamiento

- 1 Recorra la lista de frames buscando uno con los bits ($A=0;D=0$). En este recorrido NO cambie los bits de uso.
- 2 Si en el paso 1 no encontró ningún frame, recorra la lista de frames buscando uno con los bits ($A=0;D=1$). Durante este recorrido ponga el bit de uso (A) en 0 a los frames que saltea.
- 3 Si en el paso 2 no encontró ningún frame, habrá regresado a la posición de comienzo y todos los frames tienen el bit de uso (A) con el valor 0. Prosiga con el paso 1 y de ser necesario el paso 2.

Administración del Resident Set

Se basa en dos características:

Tamaño del Resident Set Para esta característica se utilizan dos políticas: Asignarle al proceso una cantidad fija de frames durante su existencia en el sistema o asignarle una cantidad de frames que pueda variar durante la vida del proceso.

Alcance del reemplazo Para esta característica se utilizan dos políticas: alcance global, en donde la página a reemplazar se elige de todos los procesos en memoria; y alcance local, en la que la página a reemplazar se elige de entre todas las pertenecientes al proceso que produjo el fallo de página.

Administración del Resident Set

Se basa en dos características:

Tamaño del Resident Set Para esta característica se utilizan dos políticas: Asignarle al proceso una cantidad fija de frames durante su existencia en el sistema o asignarle una cantidad de frames que pueda variar durante la vida del proceso.

Alcance del reemplazo Para esta característica se utilizan dos políticas: alcance global, en donde la página a reemplazar se elige de todos los procesos en memoria; y alcance local, en la que la página a reemplazar se elige de entre todas las pertenecientes al proceso que produjo el fallo de página.

Administración del Resident Set

Se basa en dos características:

Tamaño del Resident Set Para esta característica se utilizan dos políticas: Asignarle al proceso una cantidad fija de frames durante su existencia en el sistema o asignarle una cantidad de frames que pueda variar durante la vida del proceso.

Alcance del reemplazo Para esta característica se utilizan dos políticas: alcance global, en donde la página a reemplazar se elige de todos los procesos en memoria; y alcance local, en la que la página a reemplazar se elige de entre todas las pertenecientes al proceso que produjo el fallo de página.

Administración del Resident Set

	Reemplazo Local	Reemplazo Global
Asignación Fija	<ul style="list-style-type: none">● La cantidad de frames asignados al proceso es fija● La página a reemplazar es elegida de entre los frames asignados al proceso	<ul style="list-style-type: none">● No es posible
Asignación Variable	<ul style="list-style-type: none">● La cantidad de frames asignados al proceso se modifica cada cierta cantidad de tiempo● La página a reemplazar es elegida de entre los frames asignados al proceso	<ul style="list-style-type: none">● La página a reemplazar es elegida de entre todos frames de memoria, sin importar que proceso produjo el fallo de página. Esto produce que el tamaño del Resident Set de los procesos varíe constantemente

Administración del Resident Set

	Reemplazo Local	Reemplazo Global
Asignación Fija	<ul style="list-style-type: none">● La cantidad de frames asignados al proceso es fija● La página a reemplazar es elegida de entre los frames asignados al proceso	<ul style="list-style-type: none">● No es posible
Asignación Variable	<ul style="list-style-type: none">● La cantidad de frames asignados al proceso se modifica cada cierta cantidad de tiempo● La página a reemplazar es elegida de entre los frames asignados al proceso	<ul style="list-style-type: none">● La página a reemplazar es elegida de entre todos frames de memoria, sin importar que proceso produjo el fallo de página. Esto produce que el tamaño del Resident Set de los procesos varíe constantemente

Administración del Resident Set

	Reemplazo Local	Reemplazo Global
Asignación Fija	<ul style="list-style-type: none">● La cantidad de frames asignados al proceso es fija● La página a reemplazar es elegida de entre los frames asignados al proceso	<ul style="list-style-type: none">● No es posible
Asignación Variable	<ul style="list-style-type: none">● La cantidad de frames asignados al proceso se modifica cada cierta cantidad de tiempo● La página a reemplazar es elegida de entre los frames asignados al proceso	<ul style="list-style-type: none">● La página a reemplazar es elegida de entre todos frames de memoria, sin importar que proceso produjo el fallo de página. Esto produce que el tamaño del Resident Set de los procesos varíe constantemente

Administración del Resident Set

	Reemplazo Local	Reemplazo Global
Asignación Fija	<ul style="list-style-type: none">● La cantidad de frames asignados al proceso es fija● La página a reemplazar es elegida de entre los frames asignados al proceso	<ul style="list-style-type: none">● No es posible
Asignación Variable	<ul style="list-style-type: none">● La cantidad de frames asignados al proceso se modifica cada cierta cantidad de tiempo● La página a reemplazar es elegida de entre los frames asignados al proceso	<ul style="list-style-type: none">● La página a reemplazar es elegida de entre todos frames de memoria, sin importar que proceso produjo el fallo de página. Esto produce que el tamaño del Resident Set de los procesos varíe constantemente

Administración del Resident Set

	Reemplazo Local	Reemplazo Global
Asignación Fija	<ul style="list-style-type: none">● La cantidad de frames asignados al proceso es fija● La página a reemplazar es elegida de entre los frames asignados al proceso	<ul style="list-style-type: none">● No es posible
Asignación Variable	<ul style="list-style-type: none">● La cantidad de frames asignados al proceso se modifica cada cierta cantidad de tiempo● La página a reemplazar es elegida de entre los frames asignados al proceso	<ul style="list-style-type: none">● La página a reemplazar es elegida de entre todos frames de memoria, sin importar que proceso produjo el fallo de página. Esto produce que el tamaño del Resident Set de los procesos varíe constantemente

Política de Limpieza

Esta política es la opuesta a la política de carga y define cuando una página modificada ($D=1$) debe escribirse en la memoria secundaria (disco). Básicamente existen dos aproximaciones: limpieza por demanda (*Demand Cleaning*) y limpieza adelantada o pre-limpieza (*Pre-Cleaning*).

Limpieza por demanda Una página modificada solo será escrita en la memoria secundaria cuando sea seleccionada para su reemplazo.

Pre-Limpieza Una política de limpieza adelantada escribirá las páginas modificadas en la memoria secundaria antes de que sean seleccionadas para su reemplazo.

Política de Limpieza

Esta política es la opuesta a la política de carga y define cuando una página modificada ($D=1$) debe escribirse en la memoria secundaria (disco). Básicamente existen dos aproximaciones: limpieza por demanda (*Demand Cleaning*) y limpieza adelantada o pre-limpieza (*Pre-Cleaning*).

Limpieza por demanda Una página modificada solo será escrita en la memoria secundaria cuando sea seleccionada para su reemplazo.

Pre-Limpieza Una política de limpieza adelantada escribirá las páginas modificadas en la memoria secundaria antes de que sean seleccionadas para su reemplazo.

Política de Limpieza

Esta política es la opuesta a la política de carga y define cuando una página modificada ($D=1$) debe escribirse en la memoria secundaria (disco). Básicamente existen dos aproximaciones: limpieza por demanda (*Demand Cleaning*) y limpieza adelantada o pre-limpieza (*Pre-Cleaning*).

Limpieza por demanda Una página modificada solo será escrita en la memoria secundaria cuando sea seleccionada para su reemplazo.

Pre-Limpieza Una política de limpieza adelantada escribirá las páginas modificadas en la memoria secundaria antes de que sean seleccionadas para su reemplazo.

Definición

Se refiere a la determinación de cuantos procesos residen en la memoria principal. También se lo denomina nivel de multiprogramación.

- Si hay pocos procesos en memoria principal, entonces podrían existir situaciones en las que todos se encuentren bloqueados y se gastará mucho tiempo en swapping.
- Si hay muchos procesos en memoria principal, entonces, en promedio, el tamaño del Resident Set de cada uno será inadecuado y ocurrirán fallos de página muy frecuentemente.

Definición

Se refiere a la determinación de cuantos procesos residen en la memoria principal. También se lo denomina nivel de multiprogramación.

- Si hay pocos procesos en memoria principal, entonces podrían existir situaciones en las que todos se encuentren bloqueados y se gastará mucho tiempo en swapping.
- Si hay muchos procesos en memoria principal, entonces, en promedio, el tamaño del Resident Set de cada uno será inadecuado y ocurrirán fallos de página muy frecuentemente.

Definición

Se refiere a la determinación de cuantos procesos residen en la memoria principal. También se lo denomina nivel de multiprogramación.

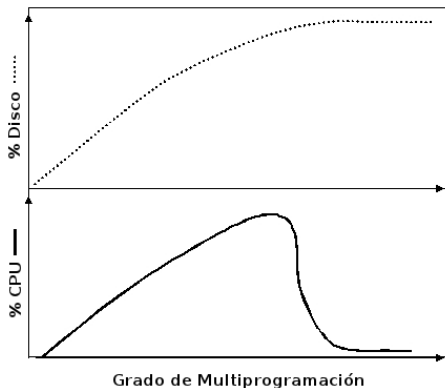
- Si hay pocos procesos en memoria principal, entonces podrían existir situaciones en las que todos se encuentren bloqueados y se gastará mucho tiempo en swapping.
- Si hay muchos procesos en memoria principal, entonces, en promedio, el tamaño del Resident Set de cada uno será inadecuado y ocurrirán fallos de página muy frecuentemente.

Hiperpaginación (*Thrashing*)

Definición

Situación en la que el CPU gasta más tiempo llevando y trayendo páginas hacia y desde la memoria secundaria, que ejecutando instrucciones de los procesos.

- Los procesos poseen un resident set demasiado pequeño
- Generan fallos de página constantemente
- Provocado por un nivel de multiprogramación excesivamente alto



Muchas Gracias!