

Tema 6.1

Sistema Operativos (SSOO)

Memoria Virtual

Índice

- Características de la memoria virtual
- Paginación
- Segmentación

Recordemos...

Los dos logros principales de la paginación y segmentación son:

- Todas las referencias dentro de un proceso son direcciones lógicas.
- Un proceso puede dividirse en varias partes (páginas o segmentos) y no es necesario que estas partes estén contiguas en memoria.

Memoria virtual

- Al añadir la memoria virtual conseguimos que no sea necesario cargar todas las páginas de un proceso en memoria.
- Es decir, parte de esas páginas pueden estar en memoria virtual.
- Según esto se puede considerar tener sólo algunos fragmentos del proceso cargado en memoria.
- A este conjunto de páginas se le llama conjunto residente.

Memoria virtual

El funcionamiento utilizando **el conjunto residente + memoria virtual** sería:

1. El procesador ejecuta las páginas de los programas que estén en memoria.
2. Al detectar (con las tablas de páginas o segmentos) una dirección lógica no cargada, el procesador indica un fallo de acceso a memoria principal (fallo de página).
3. El sistema operativo bloquea el proceso y emite una solicitud de E/S.
4. Mientras, el SO elige a otro proceso que está en espera (cola de listos) para ejecutar (recordad los algoritmos de planificación de procesos).
5. Al recibir la interrupción de E/S que indica que el fragmento deseado ya está en memoria, reanuda la ejecución del proceso bloqueado (cuando le toque según el algoritmo, pasa a estado listo).

Memoria virtual: ventajas y desventajas

Ventajas de este nuevo esquema:

- Habrá más procesos (sus conjuntos residentes) en memoria principal.
- Un proceso puede ser más grande que toda la memoria principal.

Desventajas:

- **Hiperpaginación:** se producen demasiados fallos de página.
- El procesador **consume más tiempo** intercambiando páginas que ejecutando dicho proceso

Memoria virtual: principio de cercanía

Principio de cercanía: las referencias a los datos y al programa dentro de un proceso, tienden a agruparse (espacial y temporalmente).

- Por lo anterior, es válido suponer que, durante cortos intervalos de tiempo, se necesitarán sólo unos pocos fragmentos de un proceso, y no todo él cargado por completo en memoria

En este principio se basa el uso de la memoria virtual.

Memoria virtual

A continuación se describen los algoritmos que se utilizan en la gestión de memoria virtual.

- El criterio principal de estos algoritmos, es mejorar el rendimiento.
- En memoria, cuando se habla de rendimiento, se hace referencia a minimizar el número de fallos de página.

¿Qué era un fallo de página?

Fallo de página

Fallo de página

- Se produce cuando un proceso hace referencia a un dato que está en una página que no está cargada en la memoria principal (porque está en memoria virtual).
- Este dato está en la MV porque no hay hueco libre para cargarla en memoria.
- Recordad: las páginas son del proceso, los marcos de la memoria

Marco: son propias de la memoria.

Página: son propias del proceso.

Fallo de página

Lo que se pretende es organizar las cosas para que, durante el tiempo en que un proceso se está ejecutando, la probabilidad de que haga referencia a una página ausente sea mínima.

Es decir, que haya el menor número de fallos de página posibles.

Recordemos... A este proceso se le llama paginación o intercambio (**swapping**).

Memoria virtual: políticas de optimización

Decisiones en el diseño de un sistema de gestión de memoria:

- 1) Políticas de lectura
- 2) Políticas de ubicación
- 3) Políticas de reemplazo

Política de lectura

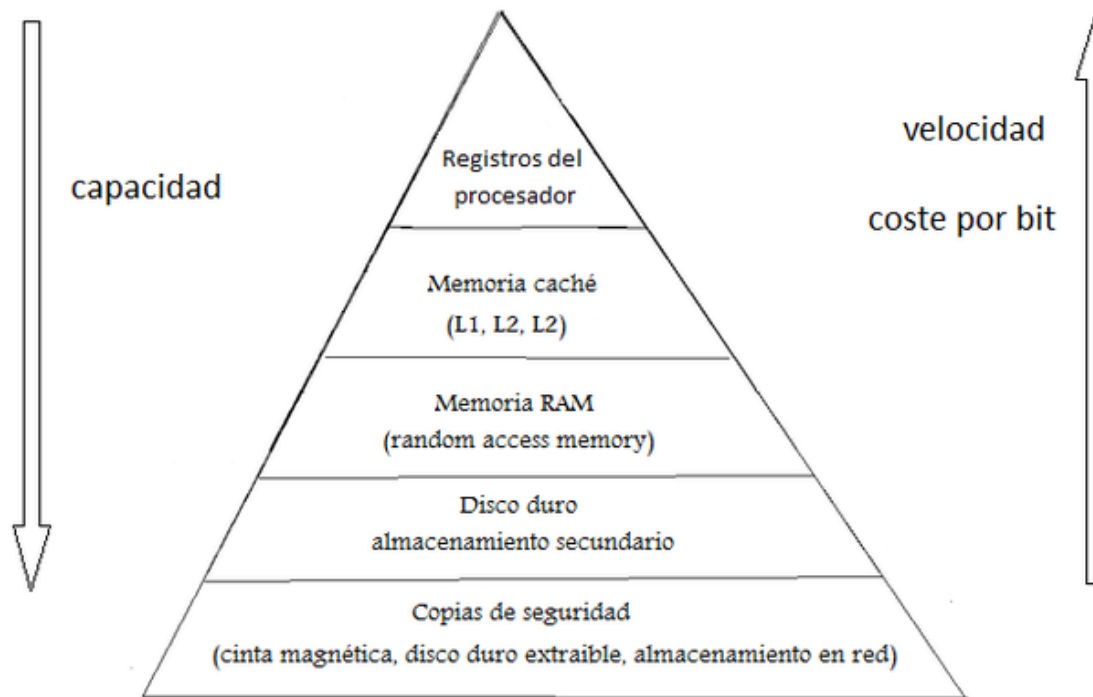
- Está relacionada con la decisión de cuándo se debe cargar una página en memoria principal.
- Hay dos tipos:
 - paginación por demanda y
 - paginación previa.

Paginación por demanda: sólo se trae la página a memoria principal cuando se hace referencia a una posición de esa página.

Paginación previa: se cargan otras páginas distintas a la demandada en cada fallo de página.

Política de ubicación

- Está relacionada con la decisión de dónde va a residir una parte de un proceso en memoria principal.



Memoria caché (i)

La memoria caché es una memoria en la que se almacena una serie de datos para su rápido acceso.

La memoria caché de un procesador es un tipo de memoria volátil (tipo RAM) con una gran velocidad. En la actualidad esta memoria está integrada en el procesador, y su cometido es almacenar una serie de instrucciones y datos a los que el procesador accede continuamente, con la finalidad de que estos accesos sean instantáneos.

Hay tres tipos de memoria caché para procesadores:

- Caché de 1er nivel (L1): esta caché está integrada en el núcleo del procesador, trabajando a su misma velocidad. La cantidad de memoria caché L1 varía entre los 64KB y los 256KB. Esta memoria a su vez, suele estar dividida en dos partes dedicadas, una para instrucciones y otra para datos.
- Caché de 2º nivel (L2): está integrada también en el procesador, tiene las mismas ventajas que la caché L1, aunque es algo más lenta que esta. La L2 suele ser de mayor tamaño que la caché L1, pudiendo llegar a superar los 2MB. A diferencia de la L1, esta no está dividida y su utilización está más enfocada a programas que al propio sistema.

Memoria caché (ii)

- En cuanto a la utilización de la caché L2 en procesadores multinúcleos, existen dos tecnologías claramente diferenciadas. Por un lado, está la habitualmente utilizada por Intel, que consiste en que el total de la caché L2 está accesible para ambos núcleos y por otro está la utilizada por AMD, en la que cada núcleo tiene su propia caché L2 dedicada solo para ese núcleo.
- Caché de 3er nivel (L3): es un tipo de memoria caché más lenta que la L2 y muy poco utilizada en la actualidad. En un principio esta caché estaba incorporada a la placa base, no junto al procesador, y su velocidad de acceso era bastante más lenta que una caché de nivel 2 o 1.

Aunque esta última es una memoria con una gran velocidad (muy superior a la RAM, especialmente en la época que se utilizaba), dicha velocidad depende de la comunicación entre el procesador y la placa base.

Las memorias caché son extremadamente rápidas (su velocidad aproximadamente 5 veces superior a la RAM). Como valor añadido no tiene latencia y como principal inconveniente, es un tipo de memoria muy costosa.

Política de reemplazo

- Está relacionada con la decisión de seleccionar la página a reemplazar en memoria principal al cargar una nueva.
- El objetivo de estas políticas es seleccionar la página a reemplazar con menos probabilidad de ser referenciada en el futuro.
- Intentan predecir el comportamiento futuro en función del comportamiento pasado.

Algoritmos para el reemplazo de páginas:

- 1) **Óptimo**
- 2) **Usado hace más tiempo**
- 3) **FIFO**
- 4) **De reloj**

Algoritmos de gestión de MV

Óptimo

Selecciona la página que tiene que esperar más tiempo antes de ser referenciada.

- Es el que genera menor número de fallo.
- Imposible de implementar. El SO debería conocer las referencias futuras.
- Es más un algoritmo teórico que sirve como referencia para comparar.

Algoritmos de gestión de MV

Usada hace más tiempo (LRU)

Reemplaza la página de memoria que no haya sido referenciada desde hace más tiempo.

- Debido al principio de cercanía, ésta debería ser la página con menor probabilidad de ser referenciada en un futuro próximo
- Resultados buenos
- Implementación complicada. Requiere ordenación cronológica de acceso a todas las páginas de todos los procesos.

Algoritmos de gestión de MV

Primera en entrar primera en salir (FIFO)

Implementación sencilla según técnica de espera circular (round-robin).

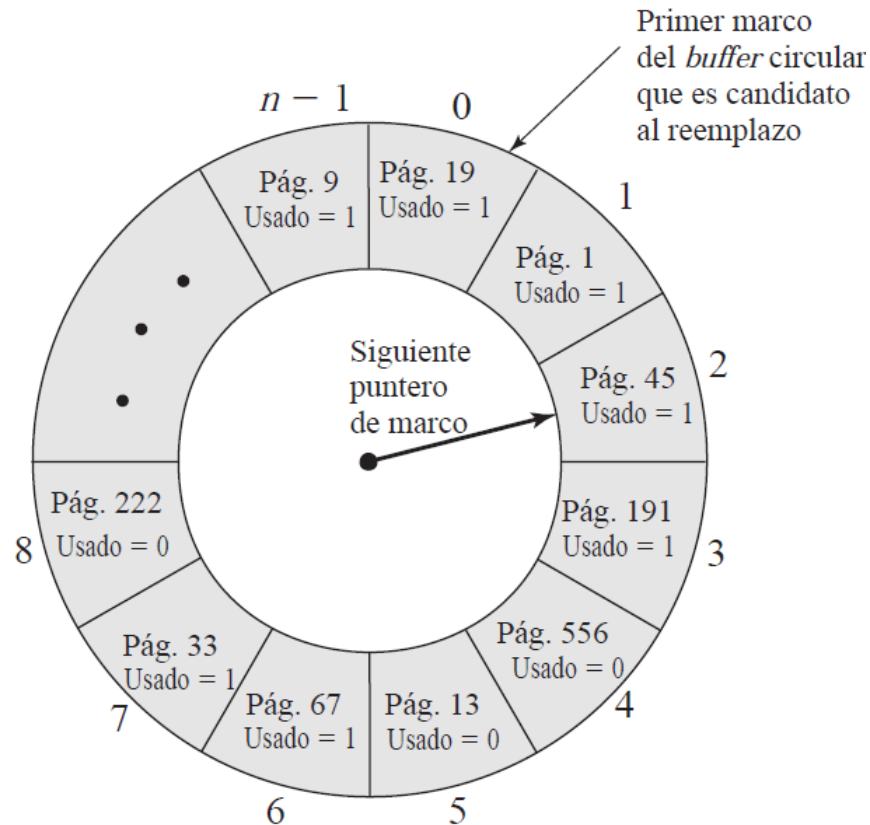
- Genera más fallos de página que las anteriores.

De reloj

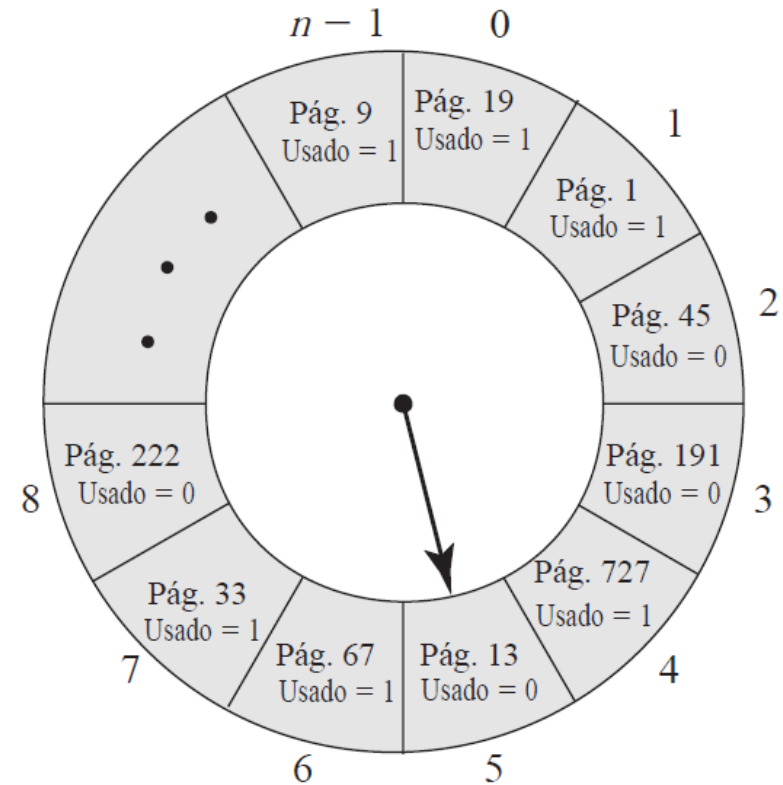
Asocia un bit de uso adicional a cada marco, que se pone a 1 al cargar una página nueva y aumenta con cada referencia a la página.

- Al solicitarse un reemplazo, un puntero recorre todos los bits de uso hasta encontrar el primer bit de uso a 0.
- Con cada bit encontrado a 1 lo pone a 0
- Es como un FIFO optimizado

Algoritmos de gestión de MV



(a) Estado del *buffer* justo ante del reemplazo de una página



(b) Estado del *buffer* justo después del siguiente reemplazo de página

Algoritmos de gestión de MV

Un buffer circular con n marcos de memoria principal que se encuentran disponibles para reemplazo de la página.

Antes del comienzo del reemplazo de una página del buffer por la página entrante 727, el puntero al siguiente marco apunta al marco número 2, que contiene la página 45. En este momento la política del reloj comienza a ejecutarse.

Debido a que el bit usado de la página 45 del marco 2 es igual a 1, esta página no se reemplaza. En vez de eso, el bit de usado se pone a 0 y el puntero avanza. De forma similar la página 191 en el marco 3 tampoco se reemplazará; y su bit de usado se pondrá a 0, avanzando de nuevo el puntero.

En el siguiente marco, el marco número 4, el bit de usado está a 0. Por tanto, la página 556 se reemplazará por la página 727. El bit de usado se pone a 1 para este marco y el puntero avanza hasta el marco 5, completando el procedimiento de reemplazo de página.

Ejemplos

Un proceso tiene asignado 3 marcos (tamaño del conjunto residente fijo), pero la ejecución del proceso hace referencia a 5 páginas diferentes.

La cadena de referencias a las páginas durante la ejecución del proceso es la siguiente:

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

Ejemplos - Solución

Flujo de
páginas
referenciales

2	3	2	1	5	2	4	5	3	2	5	2																																				
<table><tr><td>2</td></tr><tr><td></td></tr><tr><td></td></tr></table>	2			<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table>	2	3	1	<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	4	3	5	<table><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	4	3	5	<table><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	4	3	5	<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5
2																																															
2																																															
3																																															
2																																															
3																																															
2																																															
3																																															
1																																															
2																																															
3																																															
5																																															
2																																															
3																																															
5																																															
4																																															
3																																															
5																																															
4																																															
3																																															
5																																															
4																																															
3																																															
5																																															
2																																															
3																																															
5																																															
2																																															
3																																															
5																																															
2																																															
3																																															
5																																															

F

F

F

LRU

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

F

F

F

F

FIFO

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

F

F

F

F

F

F

F

RELOJ

→	<table><tr><td>2*</td></tr><tr><td></td></tr><tr><td></td></tr></table>	2*			→	<table><tr><td>2*</td></tr><tr><td>3*</td></tr><tr><td></td></tr></table>	2*	3*		→	<table><tr><td>2*</td></tr><tr><td>3*</td></tr><tr><td></td></tr></table>	2*	3*		→	<table><tr><td>2*</td></tr><tr><td>3*</td></tr><tr><td>1*</td></tr></table>	2*	3*	1*	→	<table><tr><td>5*</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table>	5*	3	1	→	<table><tr><td>5*</td></tr><tr><td>2*</td></tr><tr><td>1</td></tr></table>	5*	2*	1	→	<table><tr><td>5*</td></tr><tr><td>2*</td></tr><tr><td>4*</td></tr></table>	5*	2*	4*	→	<table><tr><td>5*</td></tr><tr><td>2*</td></tr><tr><td>4*</td></tr></table>	5*	2*	4*	→	<table><tr><td>3*</td></tr><tr><td>2</td></tr><tr><td>4</td></tr></table>	3*	2	4	→	<table><tr><td>3*</td></tr><tr><td>2*</td></tr><tr><td>4</td></tr></table>	3*	2*	4	→	<table><tr><td>3*</td></tr><tr><td>2</td></tr><tr><td>5*</td></tr></table>	3*	2	5*	→	<table><tr><td>3*</td></tr><tr><td>2*</td></tr><tr><td>5*</td></tr></table>	3*	2*	5*
2*																																																											
2*																																																											
3*																																																											
2*																																																											
3*																																																											
2*																																																											
3*																																																											
1*																																																											
5*																																																											
3																																																											
1																																																											
5*																																																											
2*																																																											
1																																																											
5*																																																											
2*																																																											
4*																																																											
5*																																																											
2*																																																											
4*																																																											
3*																																																											
2																																																											
4																																																											
3*																																																											
2*																																																											
4																																																											
3*																																																											
2																																																											
5*																																																											
3*																																																											
2*																																																											
5*																																																											

F

F

F

F

F

F

Gestión de memoria

Otros aspectos que hay que gestionar:

- 1) Gestión del conjunto residente
- 2) Políticas de vaciado
- 3) Control de carga

1) Gestión del conjunto residente

Los aspectos que hay que gestionar son:

- 1.1) Tamaño del conjunto residente: Fijo y Variable
- 1.2) Alcance del reemplazo: Local y Global

Gestión de memoria

1.1) Tamaño del conjunto residente

Dilema:

- A menor memoria asignada a un proceso, mayor el número de procesos en memoria.
- A menor número de páginas del proceso cargadas mayor tasa de fallos de página.

Existen dos políticas de asignación.

- **Asignación fija:** Otorga a un proceso un número fijo de marcos páginas.
- **Asignación variable:** Permite que el número de marcos cambie a lo largo de la vida del proceso.

Gestión de memoria

1.2) Alcance del reemplazo:

- **Local:** si la política escoge únicamente entre las páginas del proceso que causó fallo.
- **Global:** si las candidatas son cualquier página residente en la memoria, sea o no el proceso que provocó el fallo de página.

Estos dos aspectos, tamaño del conjunto residente y alcance del reemplazo se pueden combinar y mezclar y así tendremos:

- Asignación fija y alcance local.
- Asignación variable y alcance local.
- Asignación variable y alcance local.

Gestión de memoria

Asignación fija y alcance local:

- Cada proceso cargado tiene un número fijo de páginas en memoria.
- Con un fallo de página se debe elegir una a reemplazar de entre las del proceso cargadas en memoria.

Asignación fija y alcance global

- Cada proceso cargado tiene un número fijo de páginas en memoria.
- Cuando no hay marcos libres se debe elegir la página a reemplazar de entre todas las cargadas en memoria principal.

Gestión de memoria

Asignación variable y alcance local

- Al cargar un proceso en memoria se le asigna un número fijo de marcos de páginas según las necesidades del programa
- Con fallo de página se selecciona una a reemplazar del conjunto residente del proceso

Asignación variable y alcance global

- Al cargar un proceso en memoria se le asigna un número fijo de marcos de páginas según las necesidades del programa
- Cuando no hay marcos libres se debe elegir la página a reemplazar de entre todas las cargadas en memoria principal.

Gestión de memoria

2) Políticas de vaciado

- Está relacionada con la decisión de en qué momento hay que escribir en memoria secundaria una página modificada.
- Vaciado por demanda: Una página se escribe en memoria secundaria sólo cuando haya sido elegida para reemplazarse.
- Vaciado previo: Escribe las páginas modificadas por lotes antes de que se necesiten sus marcos.

Gestión de memoria

3) Control de carga

- Consiste en determinar el número de procesos que pueden estar en memoria principal (grado de multiprogramación)
- Pocos procesos residentes -> Esto no es lo que quiero. Quiero aprovechar bien la memoria.
- Demasiados procesos residentes -> el tamaño medio del conjunto residente no será el adecuado y habrá frecuentes fallos de página -> alto tiempo en intercambio.

Bibliografía

- **CARRETERO**, Jesús, **GARCÍA**, Félix, **DE MIGUEL**, Pedro, **PÉREZ**, Fernando. Sistemas Operativos: una visión aplicada. McGraw-Hill, 2001.
- **STALLINGS**, William. **Sistemas operativos: aspectos internos y principios de diseño. 5ª Edición. Editorial Pearson Educación. 2005. ISBN: 978-84-205-4462-5.**
- **TANENBAUM**, Andrew S. Sistemas operativos modernos. 3ª Edición. Editorial Prentice Hall. 2009. ISBN: 978-607- 442-046-3.



Universidad
Francisco de Vitoria
UFV Madrid

Grado en Ingeniería Informática
Escuela Politécnica Superior