Memoria Virtual

UNIVERSIDAD DEL BÍO-BÍO

Agenda

1. Objetivos
2. Fundamentos
3. Paginación bajo Demanda

UNIVERSIDAD DEL BÍO-BÍO

Objetivos

- Describir las ventajas de un sistema de memoria virtual.
- Explicar los conceptos de paginación bajo demanda, algoritmos de sustitución de páginas y asignación de marcos de página.

Fundamentos

- Sabemos que podemos ejecutar programas que se encuentren parcialmente en la memoria principal (carga dinámica), lo que proporciona muchas ventajas:
  - Programas ya no estarían restringidos a la cantidad de MP disponible. Programas podrían escribirse para espacio de direcciones virtual extremadamente grande.
  - Como cada programa podría ocupar menos MP, se podrían ejecutar más programas al mismo tiempo, incrementando la tasa de utilización del procesador.
  - Se necesitarían menos operaciones de E/S para cargar o intercambiar cada programa de usuario con el fin de almacenarlo en memoria, de manera que cada programa de usuario se ejecutaría mas rápido.
- Son ventajas tanto para el sistema como para el usuario.

UNIVERSIDAD DEL BÍO-BÍO

Memoria Virtual

- Técnica que permite la ejecución de un proceso que no requiere estar completamente en memoria principal, siendo esta característica transparente para el programador.
- Proporcionar a los programadores una memoria virtual extremadamente grande, cuando sólo está disponible una memoria física de menor tamaño.
- Facilita la programación, ya no hay que preocuparse de la cantidad de memoria física, puede concentrarse en el problema a resolver.
- Solamente parte del programa necesita estar en memoria para ejecutar.
- Permite compartir espacios de direcciones por varios procesos.
- **Puede ser implementada vía Paginación bajo demanda.**

UNIVERSIDAD DEL BÍO-BÍO

Memoria Virtual

- Estrategia para la gestión de la Memoria Virtual:
  - **Búsqueda**, ¿Cuándo ir a buscar de la memoria secundaria una parte de un programa? (Anticipación o por Demanda).
  - **Ubicación**, ¿Dónde cargar una parte de un programa en la memoria principal? (en paginación es trivial).
  - **Reemplazo**, ¿Qué sacar a memoria secundaria si no existe espacio disponible?.

UNIVERSIDAD DEL BÍO-BÍO

Paginación bajo Demanda

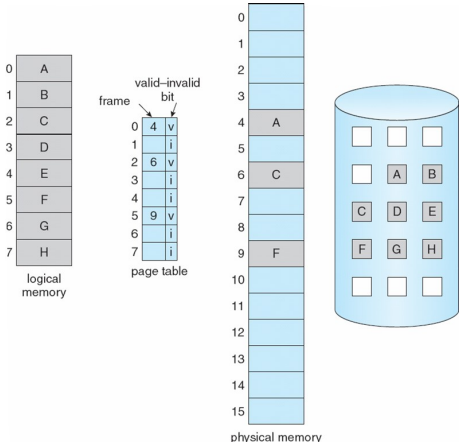
- **Búsqueda**, Una página se trae a memoria principal cuando se referencia y no está en MP.
- **Ubicación**, Una página puede ser cargada en cualquier marco de página libre.
- **Reemplazo**, En caso de no existir espacio, la página que está en el marco a asignar debe ser respaldada (si es necesario) y sobrecargarla con la página traída.

Bit de Validación

- Campo de información en Tabla de Páginas que determina si la página asociada se encuentra en MP o MS.
- Valor **v** indica que acceso es legal y además está en MP
- Valor **i** indica que acceso es ilegal (página no se encuentra en espacio lógico del proceso) o es válida pero NO está en MP.
- Acceso a una página no presente en MP causa un **Fallo o Falta de página** (Durante la traducción de la dirección).

UNIVERSIDAD DEL BÍO-BÍO

Ejemplo


0 A
1 B
2 C
3 D
4 E
5 F
6 G
7 H
logical memory
valid–invalid bit
frame
0 4 v
1 i
2 6 v
3 i
4 i
5 9 v
6 i
7 i
page table
0
1
2
3
4 A
5
6 C
7
8
9 F
10
11
12
13
14
15
physical memory
A B
C D E
F G H

Fallos de página

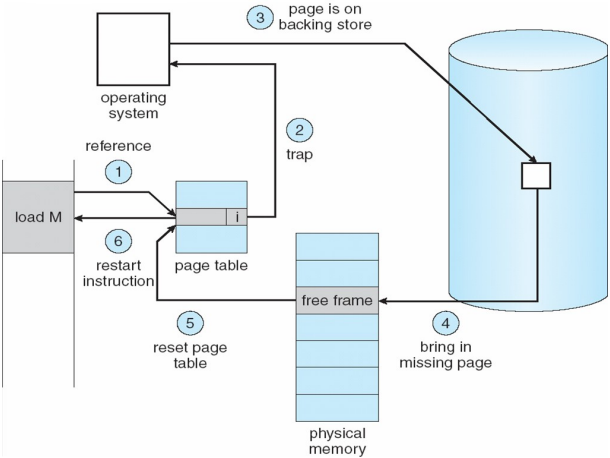
¿Qué sucede si se trata de acceder a página que no está en MP?

- SO mira en tabla de páginas (que se mantiene con el PCB del proceso) para decidir:
  - Si referencia es inválida, entonces aborta (termina el proceso).
  - Si referencia es válida:
    - Busca un marco libre.
    - Ordena operación de disco para leer la página en el marco asignado.
    - Se modifica la tabla de páginas (bit de validación a **v**).
    - Se reinicia la instrucción que fue interrumpida.

Si hay una referencia a una página, la primera referencia a esa página siempre será una interrupción (trap) al SO: **page fault**.

UNIVERSIDAD DEL BÍO-BÍO

Fallos de página



Reemplazo de páginas

- ¿Cómo crear espacio libre?
  - Comprar mas memoria.
  - Swappear un proceso a disco (reduciendo el grado de multiprogramación).
  - Reemplazar una página.
- Reemplazo:
  - Permite mantener más procesos en memoria, no necesita tener todas las páginas en MP (aumenta grado de multiprogramación)
  - Aumenta al doble el tiempo de acceso cuando hay de fallo de página (descarga + carga de página), pero se puede reducir usando bit de modificación.
  - Probabilidad de falla de página aumenta si la fracción de programa en MP es menor.

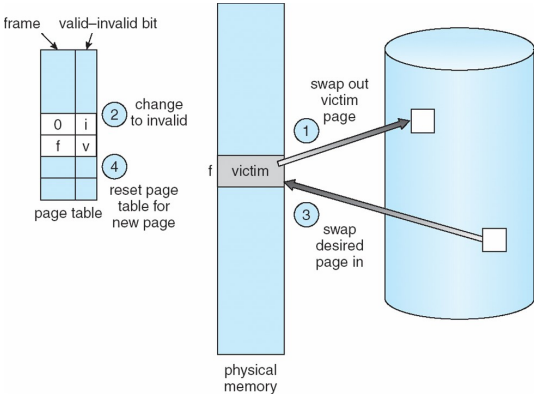
UNIVERSIDAD DEL BÍO-BÍO

Reemplazo de páginas

- Método de reemplazo:
  - Si no existe marco de página libre, se debe reemplazar.
  - Elegir una víctima usando un **Algoritmo de reemplazo**.
  - Si la página saliente ha sido modificada (bit de modificación), debe ser respaldada/copiada a MP, de lo contrario, se ahorra este paso.
  - Cargar la nueva página en el marco liberado.

Reemplazo de páginas

Víctima **f**



Algoritmos de Reemplazo

- Diferentes políticas de reemplazo:
  - Aleatorio
  - Óptimo
  - FIFO
  - El menos usado,
  - etc.
- Algoritmo elegido tiene fuerte impacto en el desempeño del sistema.
- Evaluación de algoritmos es utilizando un string particular de referencias a memoria y calculando el número de fallos de página para tal string.
  - ejemplo string: A,B,C,D,A,B,E,A,B,C,D,E

UNIVERSIDAD DEL BÍO-BÍO

Algoritmos de Reemplazo - Aleatorio

- Reemplaza aleatoriamente cualquier página de MP, sin hacer ningún esfuerzo de predicción.
- Este algoritmo es el más simple dado que no requiere mantener información sobre el comportamiento del proceso.
- Por lo anterior, no logra un buen desempeño.

Algoritmos de Reemplazo - Óptimo

- Se llega a un desempeño óptimo reemplazando aquella página que va a ser nuevamente usada en el futuro más lejano.
- Mira hacia adelante en el tiempo.
- Logra el mejor desempeño posible.
- Tiene conocimiento total sobre el desarrollo de la computación.
- No es, en la práctica, realizable. Sin embargo, permite comparar qué tan bien lo hacen otros algoritmos respecto del óptimo.

UNIVERSIDAD DEL BÍO-BÍO

Algoritmos de Reemplazo - Óptimo

A	B	C	D	A	B	E	A	B	C	D	E
A*	A	A	A	A	A	A	A	A	C*	C	C
	B*	B	B	B	B	B	B	B	B	D*	D
		C*	D*	D	D	E*	E	E	E	E	E
X	X	X	X			X			X	X	

- X = Falta de página. Página no está en MP, se va a buscar a MS.

- Si no aparece referencia futura a páginas, se debe considerar reemplazar la que lleva más tiempo en MP.

Algoritmos de Reemplazo - FIFO

- Asocia a cada página **el instante** en que dicha página fue cargada en memoria.
- Mira hacia atrás en el tiempo.
- Cuando hay que sustituir una página, se elige la más antigua.

Algoritmos de Reemplazo - FIFO

A	B	C	D	A	B	E	A	B	C	D	E
A*	B*	C*	D*	A*	B*	E*	E	E	C*	D*	D
	A	B	C	D	A	B	B	B	E	C	C
		A	B	C	D	A	A	A	B	E	E
X	X	X	X	X	X	X			X	X	

- 9 faltas de página.

- Por comodidad, mover la página más antigua a los marcos inferiores y así reemplazar siempre la página del último marco.

Algoritmos de Reemplazo - FIFO

A	B	C	D	A	B	E	A	B	C	D	E
A*	B*	C*	D*	D	D	E*	A*	B*	C*	D*	E*
	A	B	C	C	C	D	E	A	B	C	D
		A	B	B	B	C	D	E	A	B	C
			A	A	A	B	C	D	E	A	B
X	X	X	X			X	X	X	X	X	X

- 10 faltas de página.

Algoritmos de Reemplazo - FIFO

- **Anomalía de Belady**
  - Parece natural que contra más marcos de página que se asignen a un proceso, habrá menos fallas de página.
  - Sin embargo, Laszlo Belady (1969) observó que en el algoritmo FIFO puede suceder lo contrario (ver ejemplos anteriores).

Algoritmos de Reemplazo - LRU (Least Recently Used)

- **Menos recientemente utilizada.**
- Asocia a cada página el **instante en que se usó por última vez**.
- Usar un stack, ordenando páginas por antigüedad de referencia.

Algoritmos de Reemplazo - LRU (Least Recently Used)

A	B	C	D	A	B	E	A	B	C	D	E
A*	B*	C*	D*	A*	B*	E*	A	B	C*	D*	E*
	A	B	C	D	A	B	E	A	B	C	D
		A	B	C	D	A	B	E	A	B	C
X	X	X	X	X	X	X			X	X	X

- 10 faltas de página.

- Por comodidad, dejar en el tope la página que se referenció recientemente.

Algoritmos de Reemplazo - LRU (Least Recently Used)

A	B	C	D	A	B	E	A	B	C	D	E
A*	B*	C*	D*	A	B	E*	A	B	C*	D*	E*
	A	B	C	D	A	B	E	A	B	C	D
		A	B	C	D	A	B	E	A	B	C
			A	B	C	D	D	D	E	A	B
X	X	X	X			X			X	X	X

- 8 faltas de página.