

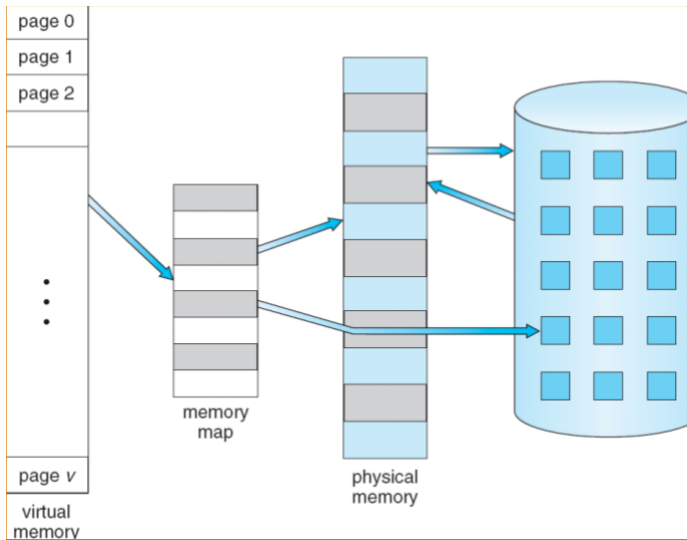
# Sistemas Operativos

## Paginación bajo demanda

Departamento de Ingeniería en Sistemas y Computación  
Universidad Católica del Norte, Antofagasta.

- Separación de memoria lógica de usuario de la memoria física
- Sólo una parte del programa debe estar en memoria para ejecución
- Espacio de direcciones lógico puede ser más grande que la memoria física

# Memoria virtual

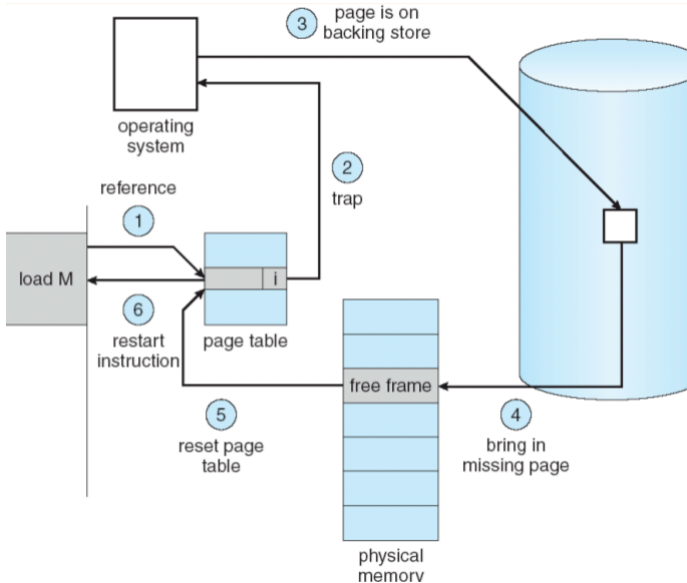


Puede implementarse a través de

- Paginación bajo demanda
- Segmentación bajo demanda

- Trae una página a memoria cuando es requerida
- El código que maneja páginas se conoce como **paginador**
- Cuando se requiere una página:
  - referencia inválida: se aborta
  - no está en memoria: se carga

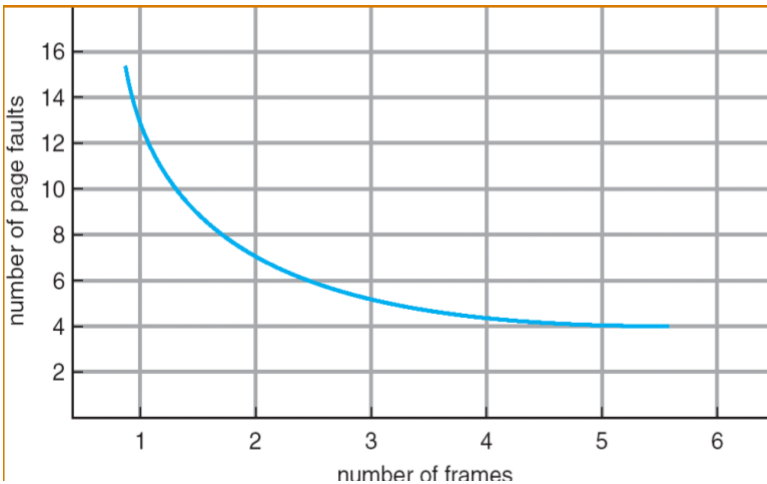
# Fallo de página



Qué sucede si no hay marcos disponibles?

- se busca una página en memoria que **no** esté en uso y se saca
- una misma página puede ser traída a memoria muchas veces
- se desea un algoritmo que permita obtener el menor número de fallos de página posibles

# Fallos de página versus número de marcos





# Algoritmo First-In First-Out

Serie: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

3 frames (3 páginas pueden estar en memoria)

4 frames

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

9 faltas de página

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

10 faltas de página

**Anomalía de Belady:** más frames  $\Rightarrow$  más faltas de página

# Ejemplo

reference string

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

7	7	7	2
	0	0	0
		1	1

2	2	4	4	4	0
3	3	3	2	2	2
1	0	0	0	3	3

0	0
1	1
3	2

7	7	7
1	0	0
2	2	1

page frames

# Algoritmo Less-Recently-Used

Serie: 1, 2, 3, 4, 1, 2, **5**, 1, 2, **3**, **4**, **5**

1	1	1	1	<b>5</b>
2	2	2	2	2
3	<b>5</b>	5	4	4
4	4	<b>3</b>	3	3

Implementación con contador

# Ejemplo

reference string

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

7	7	7	2		2		4	4	4	0			1		1		1
	0	0	0		0		0	0	3	3			3		0		0
		1	1		3		3	2	2	2			2		2		7

page frames

Reemplazar página que NO será utilizada en el periodo más largo

Ejemplo con 4

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

1
2
3
4

4

6 faltas de página

5

# Ejemplo

reference string

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

7	7	7	2		2		2			2			2				7		
	0	0	0		0		4			0			0				0		
		1	1		3		3			3			1				1		

page frames

Considere la serie de referencias:

4, 3, 2, 1, 2, 5, 1, 2, 4, 3, 5.

De los siguientes algoritmos, determine la cantidad de fallas de página que se obtiene con cada uno, considerando que sólo 3 páginas pueden estar cargadas en memoria.

- FIFO
- LRU
- Óptimo