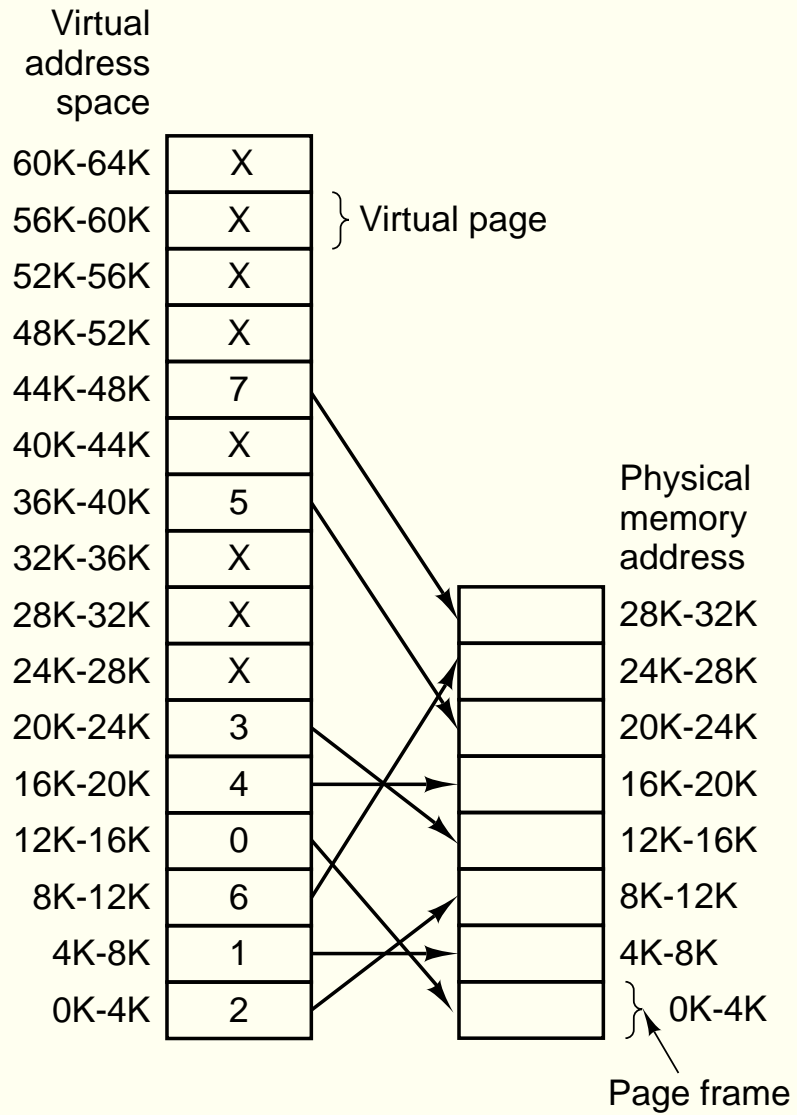


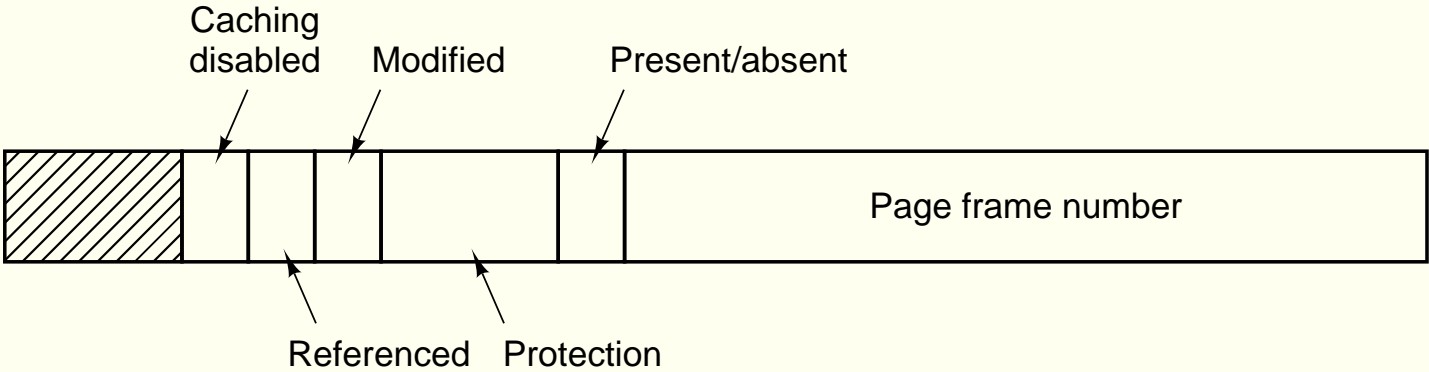
Gerenciamento de Memória

Paginação

Paginação



Entrada na tabela



Substituição de páginas

Algoritmo ótimo:

- Baseado no uso futuro de uma página
- Impossível de ser implementado
- Pode ser simulado (segunda execução do mesmo processo com a mesma entrada)
- Útil para medidas de desempenho

Não usada recentemente

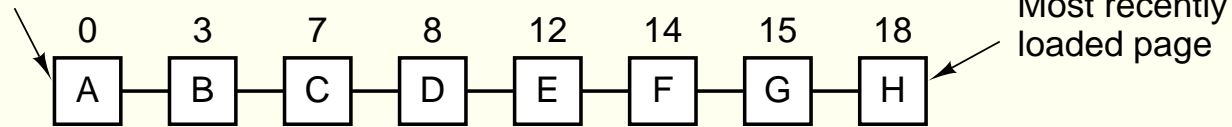
- Classe 0: não referenciada, não modificada
- Classe 1: não referenciada, mas modificada
- Classe 2: referenciada, mas não modificada
- Classe 3: referenciada e modificada

First In, First Out

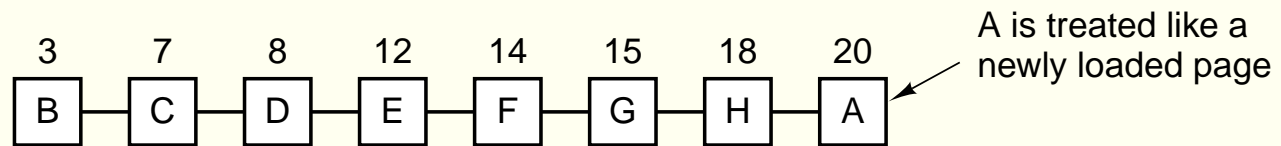
- Simplementes coloca as páginas em uma fila
- Pode remover páginas importantes

Segunda chance

Page loaded first



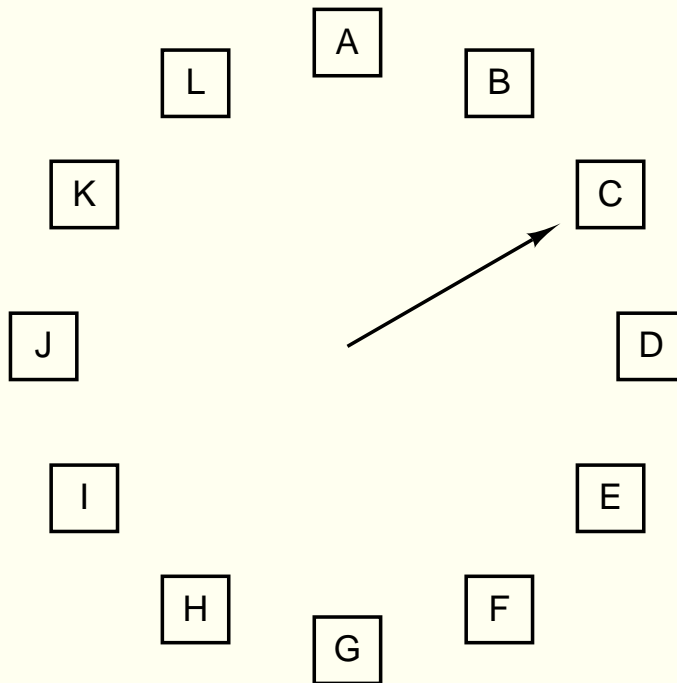
(a)



(b)

- Se o bit $R == 0$, a página é substituída, senão
- bit R é limpo e a página é colocada no final da fila

Relógio



When a page fault occurs, the page the hand is pointing to is inspected. The action taken depends on the R bit:

R = 0: Evict the page

R = 1: Clear R and advance hand

- Implementação circular da segunda chance

Menos recentemente usada

- LRU (Least Recently Used)
- Implementação utilizando lista ligada
- Implementação em hardware com contador
 - incrementado a cada instrução
 - entrada na tabela deve armazenar o contador
- Implementação com matriz $n \times n$

LRU em hardware

	Page			
	0	1	2	3
0	0	1	1	1
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0

(a)

	Page			
	0	1	2	3
0	0	0	1	1
1	1	0	1	1
2	0	0	0	0
3	0	0	0	0

(b)

	Page			
	0	1	2	3
0	0	0	0	1
1	1	0	0	1
2	1	1	0	1
3	0	0	0	0

(c)

	Page			
	0	1	2	3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0

(d)

	Page			
	0	1	2	3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	1
3	1	1	0	0

(e)

0	0	0	0
1	0	1	1
1	0	0	1
1	0	0	0

(f)

0	1	1	1
0	0	1	1
0	0	0	1
0	0	0	0

(g)

0	1	1	0
0	0	1	0
0	0	0	0
1	1	1	0

(h)

0	1	0	0
0	0	0	0
1	1	0	1
1	1	0	0

(i)

0	1	0	0
0	0	0	0
1	1	0	0
1	1	1	0

(j)

Acessos: 0 1 2 3 2 1 0 3 2 3

Simulando LRU em software

Página não usada frequentemente

- Contador de uso para cada página (soma o bit R a cada clock tick)
- Não esquece nada...
- Considere um compilador baseado em passos

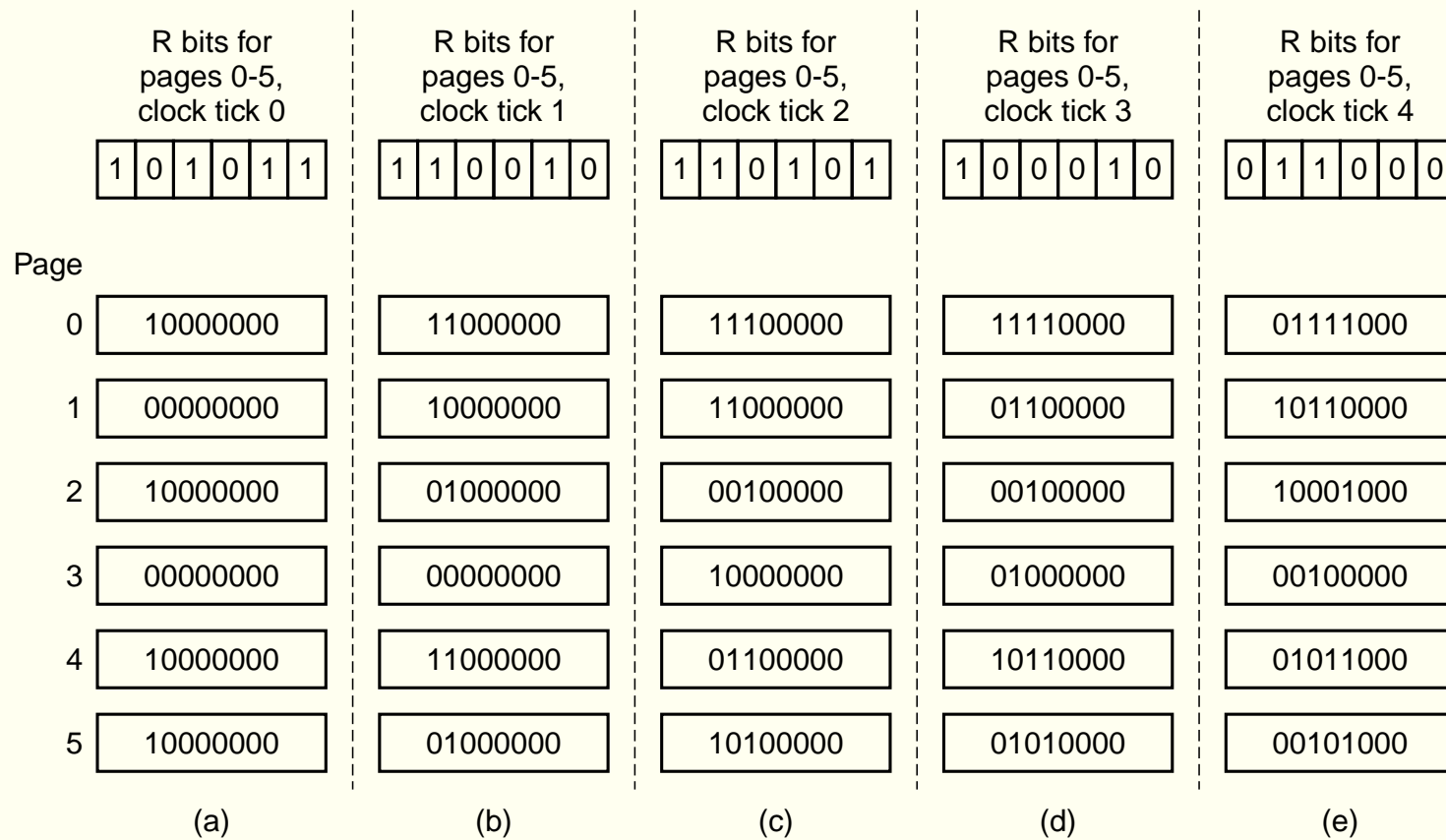
Menos recentemente usada

Aging

- O contador é deslocado à direita
- Bit R é adicionado à esquerda

		1 0 1 0 1 1			1 0 1 1 0 1
>>	1	1 0 1 0 1	>>	1	1 0 1 1 0
+	1	1 1 0 1 0 1	+	0	0 1 0 1 1 0

Aging

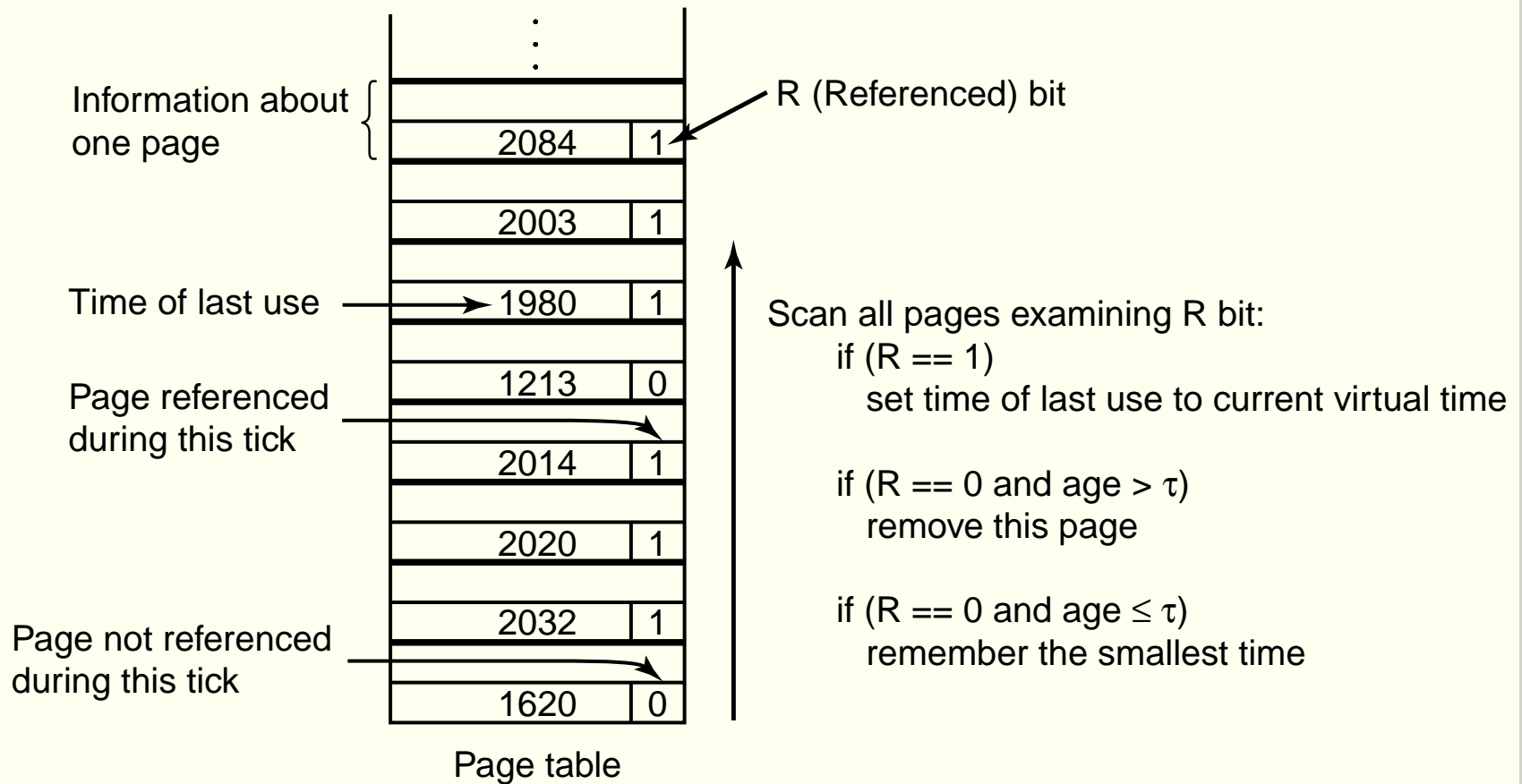


Working Set $w(k, t)$

- Conjunto de páginas utilizadas nas últimas k referências em relação ao instante t .
- Paginação sob demanda
- Prepaging
- Implementação exata é muito cara
 - Registro com k posições
- Conjunto passa a ser definido por tempo

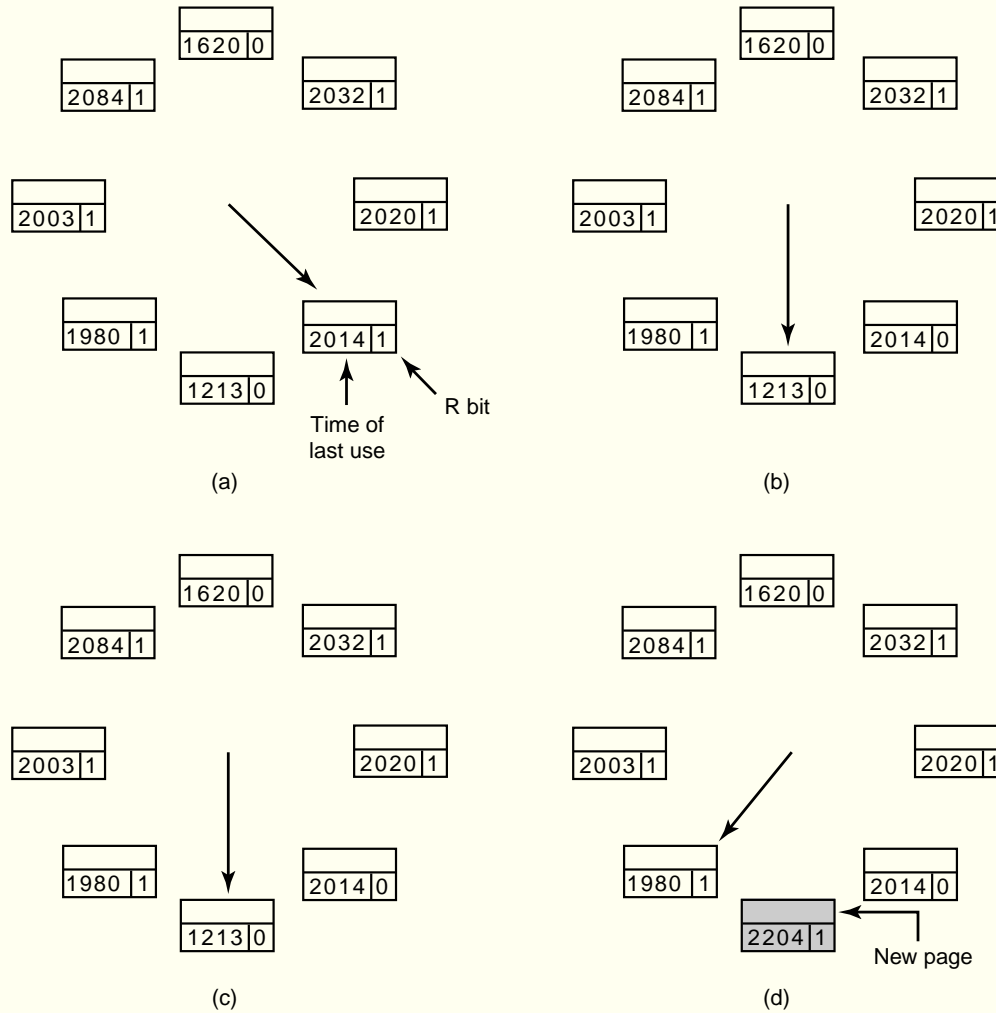
Working Set

2204 Current virtual time



WSClock (R==1 e R==0)

2204 | Current virtual time



Resumo

Algorithm	Comment
Optimal	Not implementable, but useful as a benchmark
NRU (Not Recently Used)	Very crude
FIFO (First-In, First-Out)	Might throw out important pages
Second chance	Big improvement over FIFO
Clock	Realistic
LRU (Least Recently Used)	Excellent, but difficult to implement exactly
NFU (Not Frequently Used)	Fairly crude approximation to LRU
Aging	Efficient algorithm that approximates LRU well
Working set	Somewhat expensive to implement
WSClock	Good efficient algorithm

Anomalia de Belady

Algoritmo FIFO e número de page faults

All pages frames initially empty

	0	1	2	3	0	1	4	0	1	2	3	4	
Youngest page		0	1	2	3	0	1	4	4	4	2	3	3
			0	1	2	3	0	1	1	1	4	2	2
Oldest page				0	1	2	3	0	0	0	1	4	4
		P	P	P	P	P	P				P	P	

9 Page faults

(a)

	0	1	2	3	0	1	4	0	1	2	3	4	
Youngest page		0	1	2	3	3	3	4	0	1	2	3	4
			0	1	2	2	2	3	4	0	1	2	3
Oldest page				0	1	1	1	2	3	4	0	1	2
					0	0	0	1	2	3	4	0	1
		P	P	P	P			P	P	P	P	P	P

10 Page faults

(b)

Algoritmos a pilha

Não apresentam a anomalia

Reference string 0 2 1 3 5 4 6 3 7 4 7 3 3 5 5 3 1 1 1 7 1 3 4 1

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

Page faults P P P P P P P P P P P P P P P

Distance string ∞ ∞ ∞ ∞ ∞ ∞ ∞ ∞ 4 ∞ 4 2 3 1 5 1 2 6 1 1 4 2 3 5 3

Exemplo: LRU

Política Local ou Global

	Age
A0	10
A1	7
A2	5
A3	4
A4	6
A5	3
B0	9
B1	4
B2	6
B3	2
B4	5
B5	6
B6	12
C1	3
C2	5
C3	6

(a)

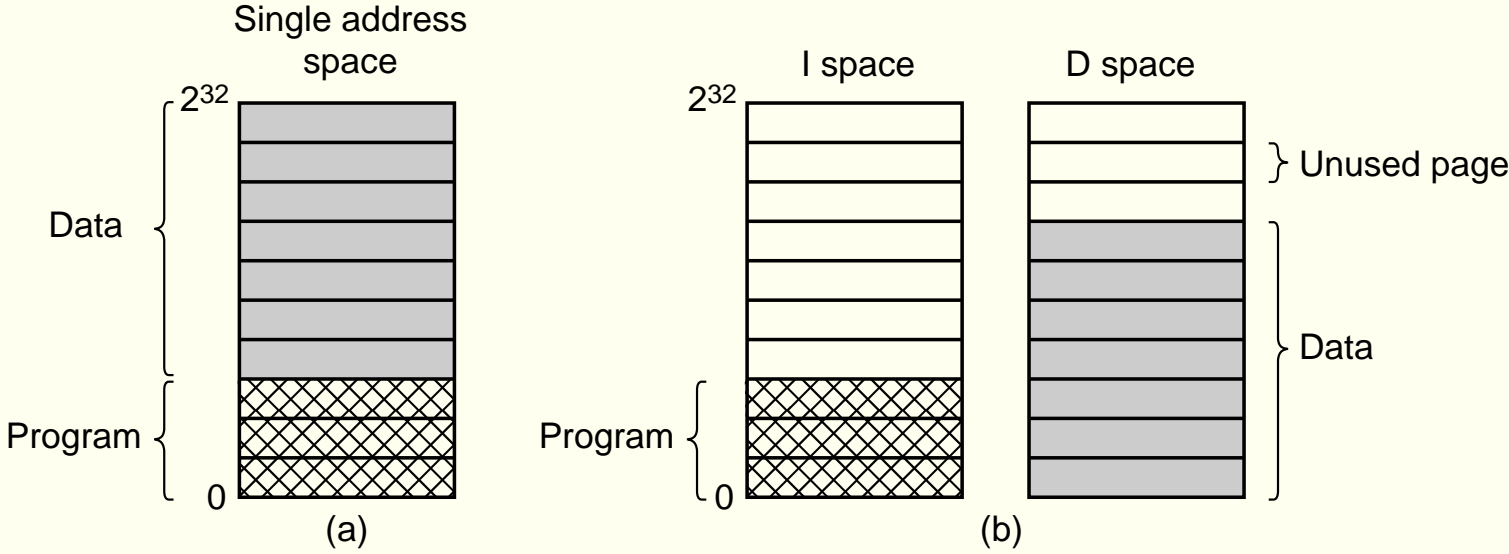
A0
A1
A2
A3
A4
A6
B0
B1
B2
B3
B4
B5
B6
C1
C2
C3

(b)

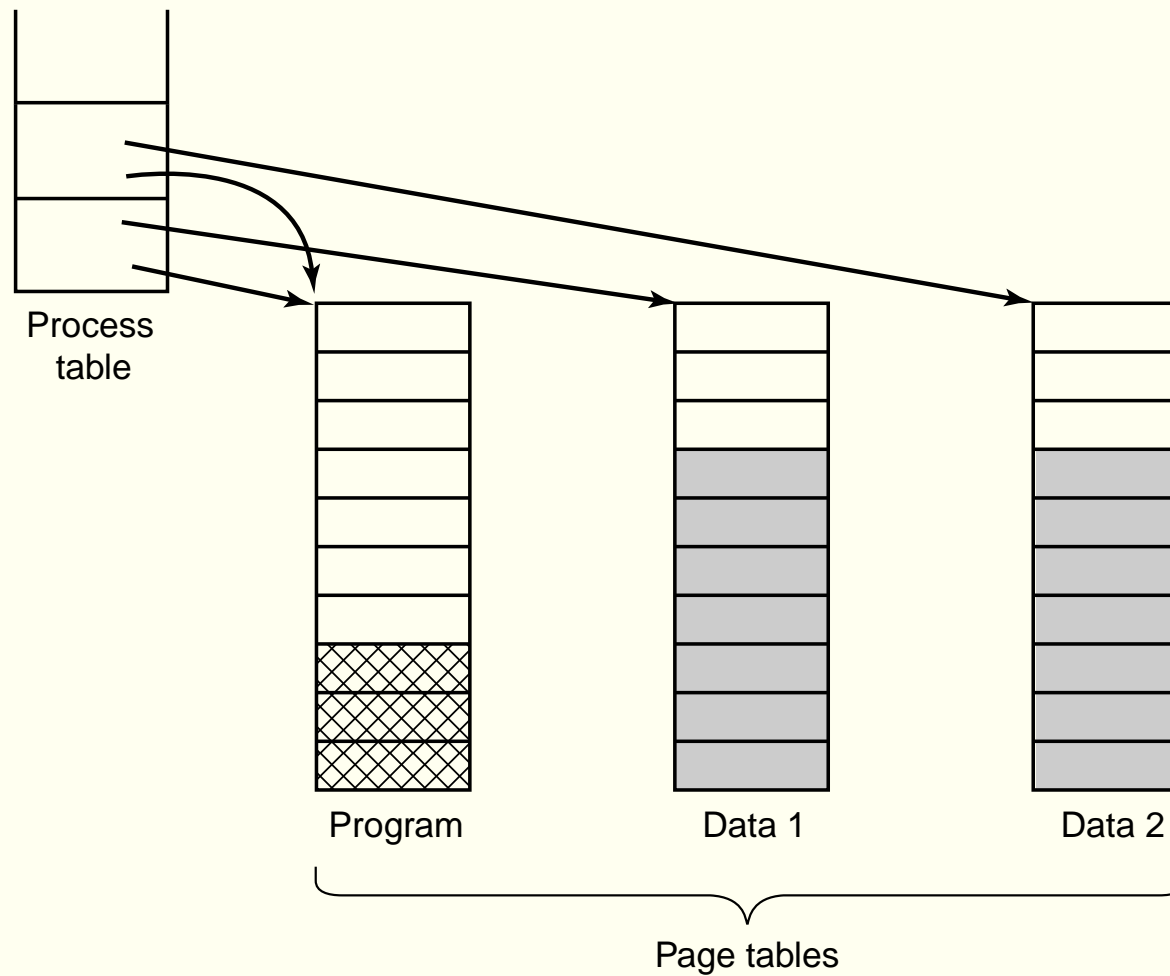
A0
A1
A2
A3
A4
A5
B0
B1
B2
A6
B4
B5
B6
C1
C2
C3

(c)

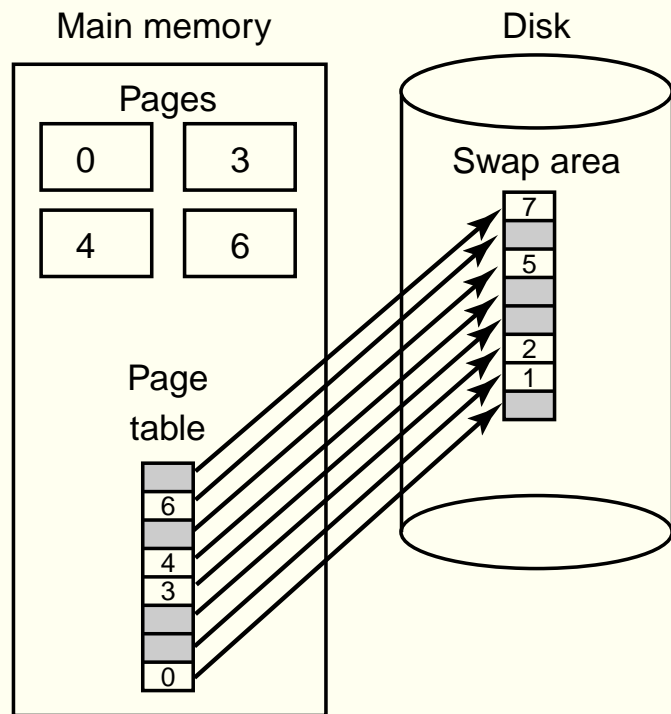
Espaços separados de endereçamento



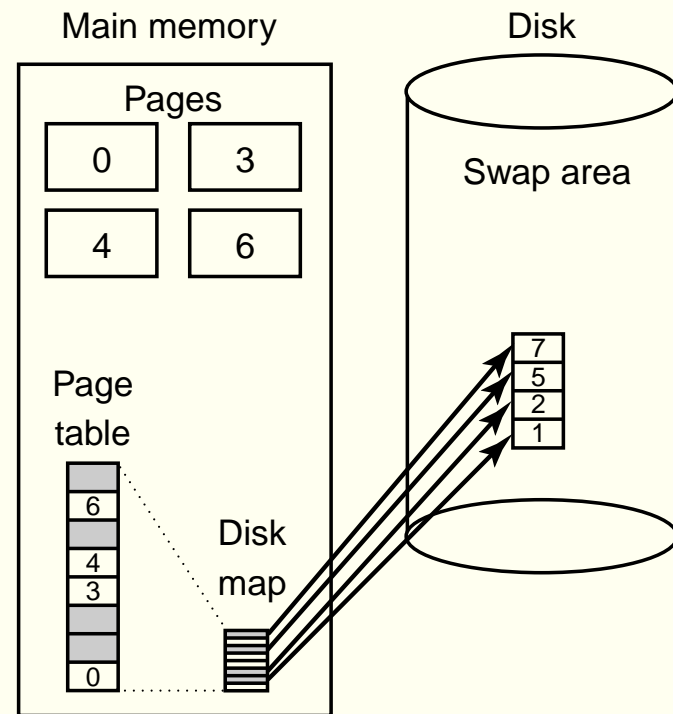
Páginas compartilhadas



Área de swap

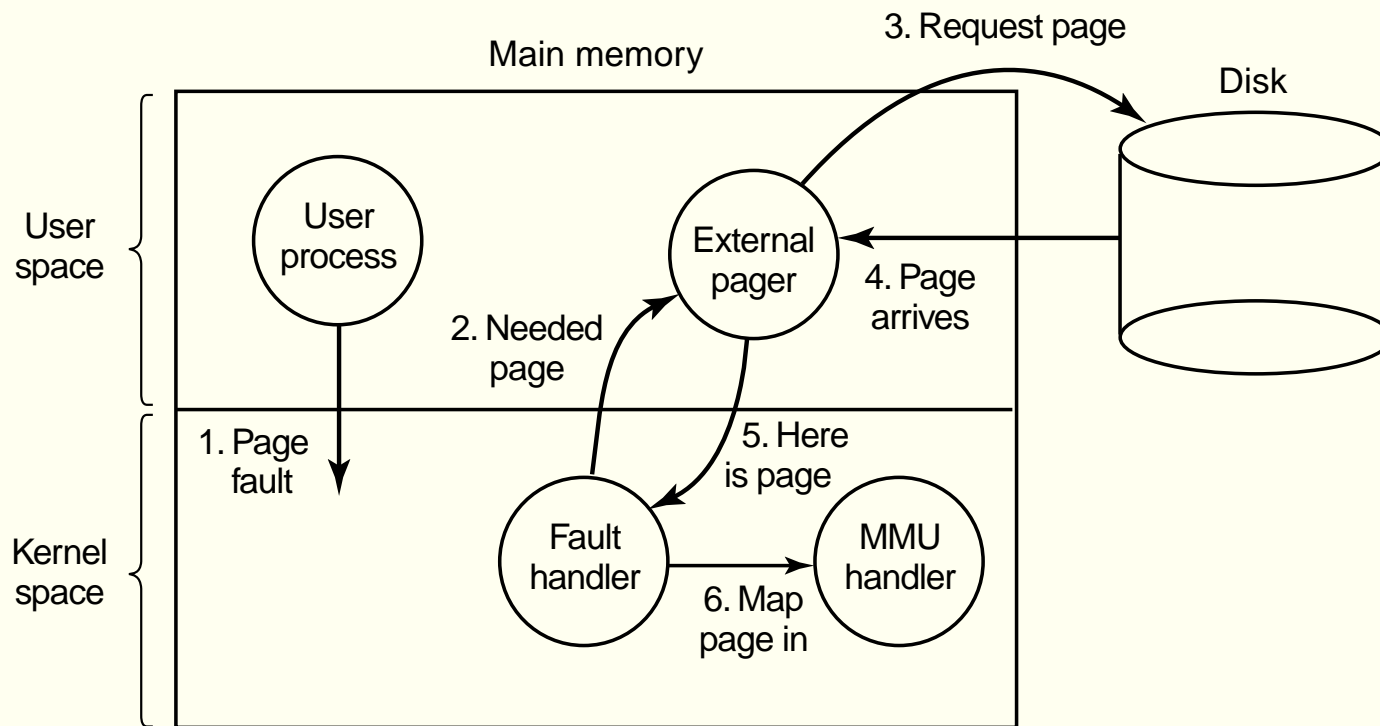


(a)

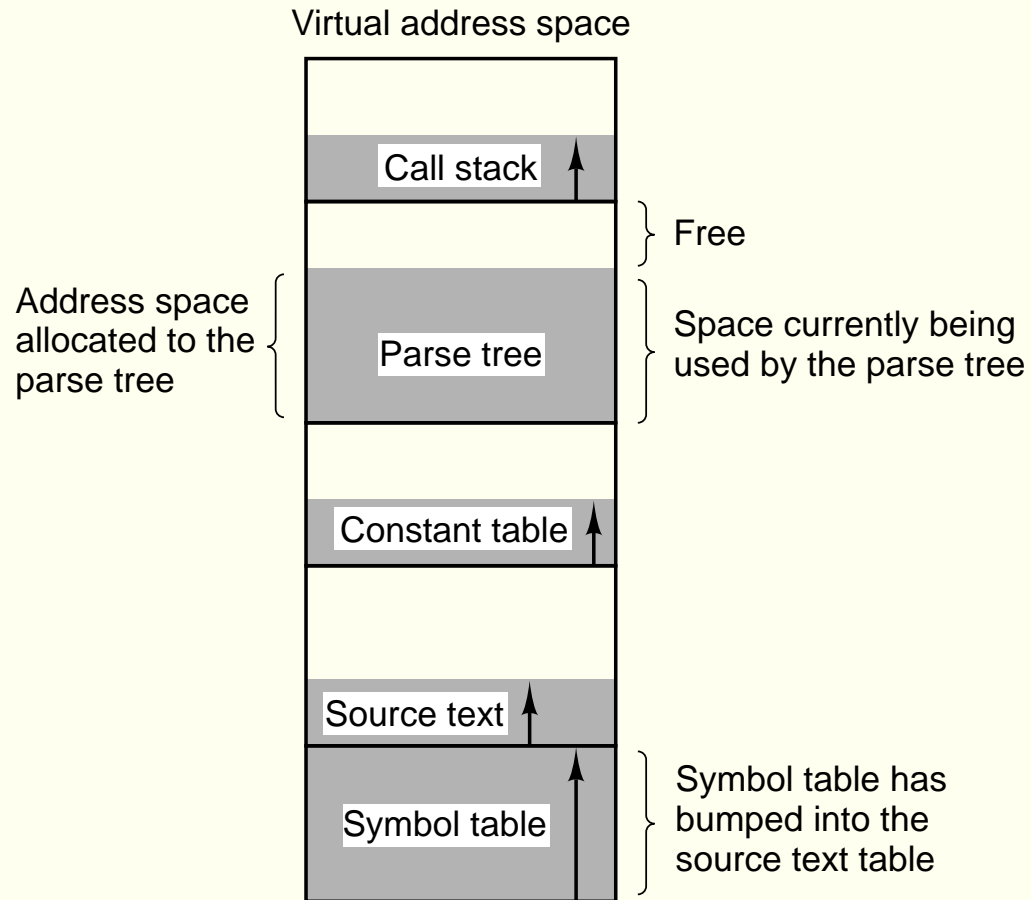


(b)

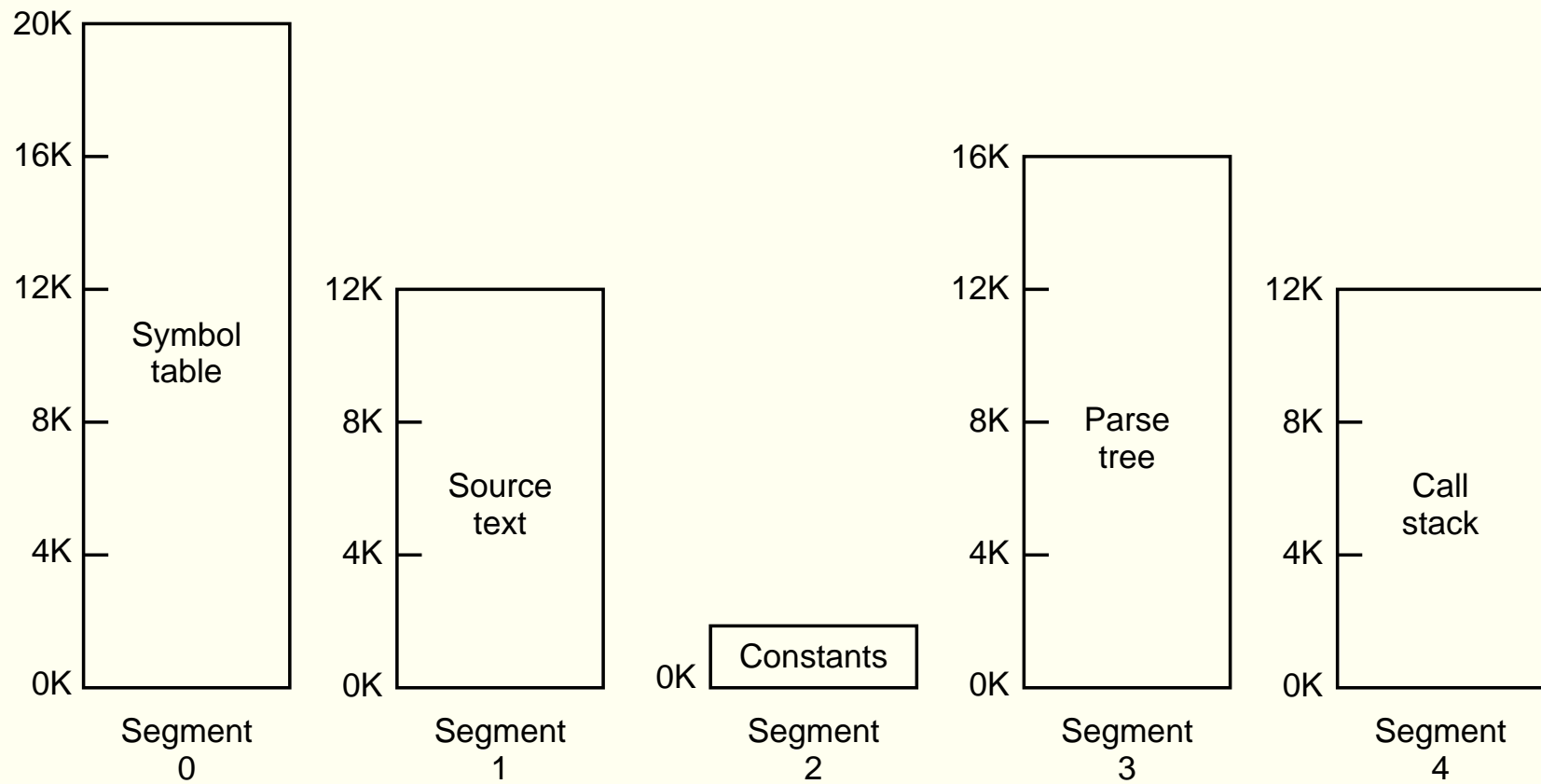
Paginador externo



Limites de um espaço único

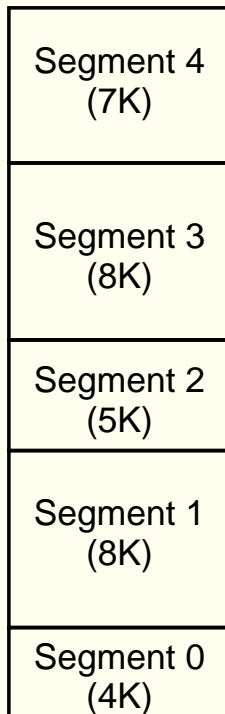


Segmentação

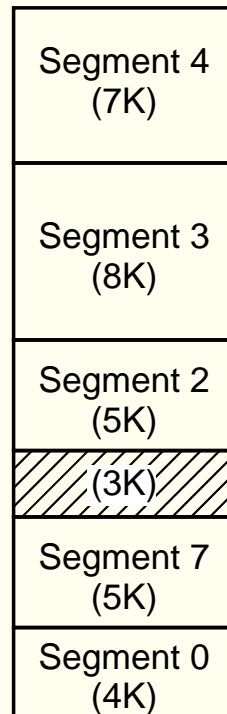


Consideration	Paging	Segmentation
Need the programmer be aware that this technique is being used?	No	Yes
How many linear address spaces are there?	1	Many
Can the total address space exceed the size of physical memory?	Yes	Yes
Can procedures and data be distinguished and separately protected?	No	Yes
Can tables whose size fluctuates be accommodated easily?	No	Yes
Is sharing of procedures between users facilitated?	No	Yes
Why was this technique invented?	To get a large linear address space without having to buy more physical memory	To allow programs and data to be broken up into logically independent address spaces and to aid sharing and protection

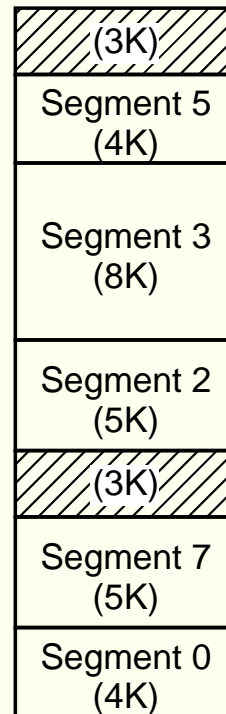
Segmentação pura



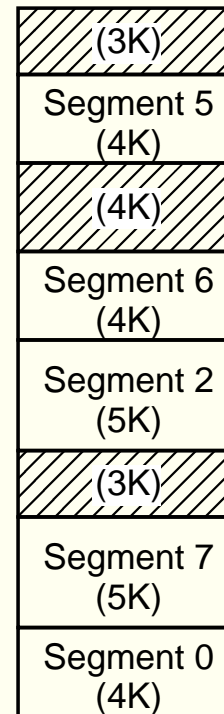
(a)



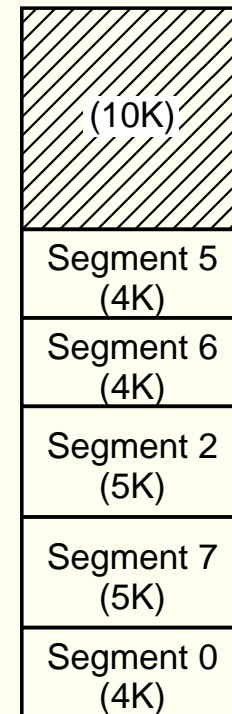
(b)



(c)

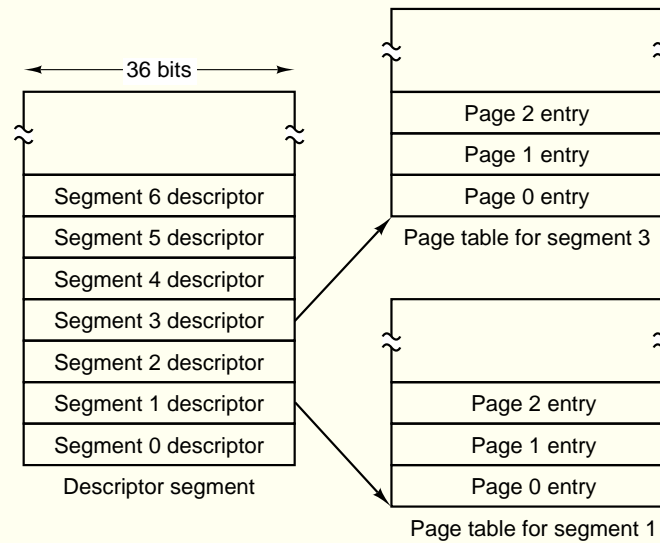


(d)

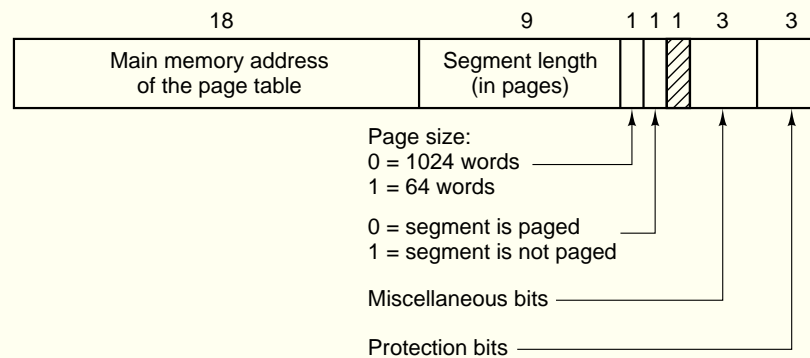


(e)

MULTICS: segmentação e paginação

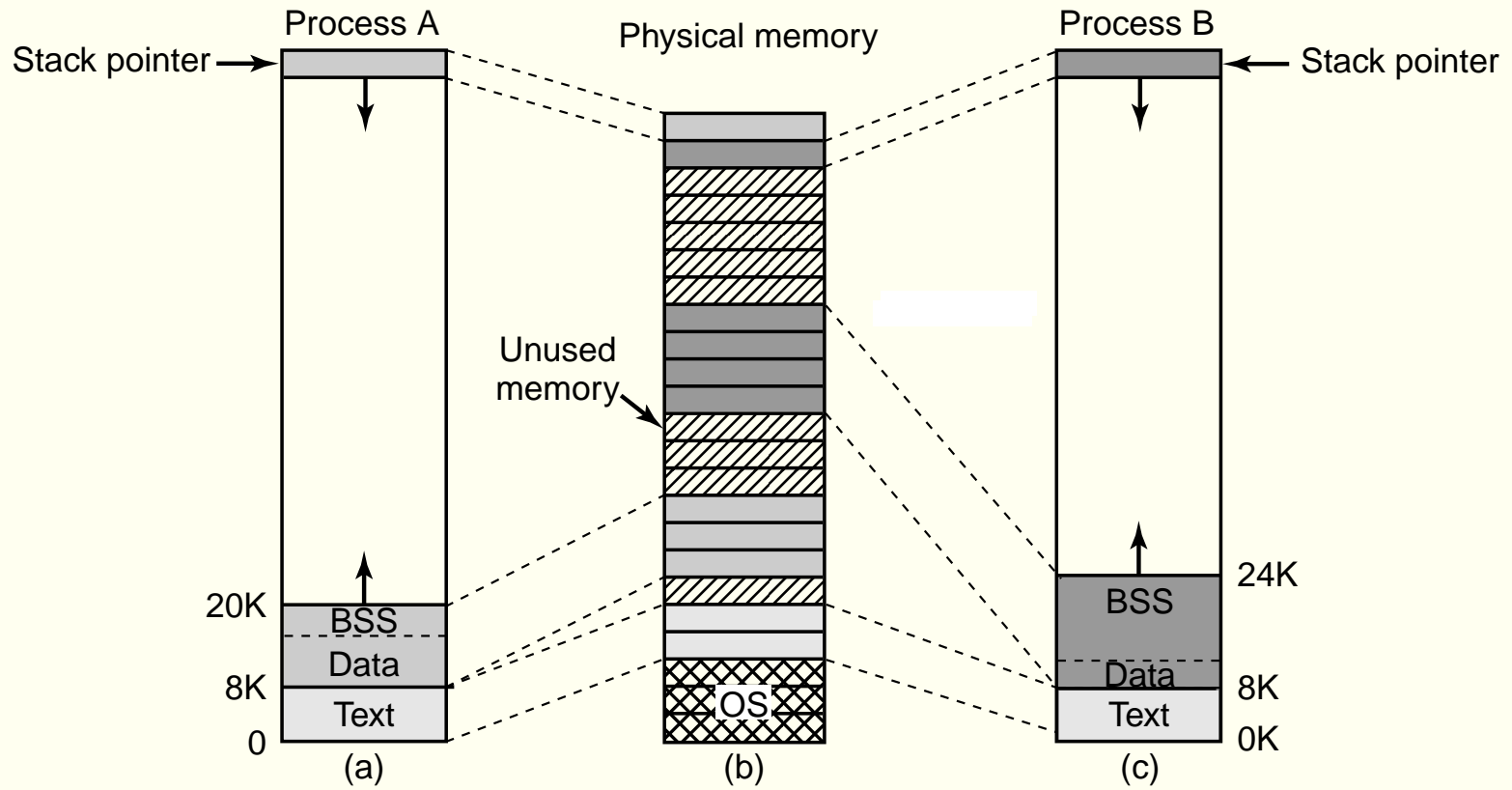


(a)



(b)

UNIX



Memória compartilhada

Processo A

Processo B

