

Infra-Estrutura de Software

IF677

Gerenciamento de Memória

Tópicos

- Gerenciamento básico de memória
- Troca de processos
- Memória virtual
- Paginação

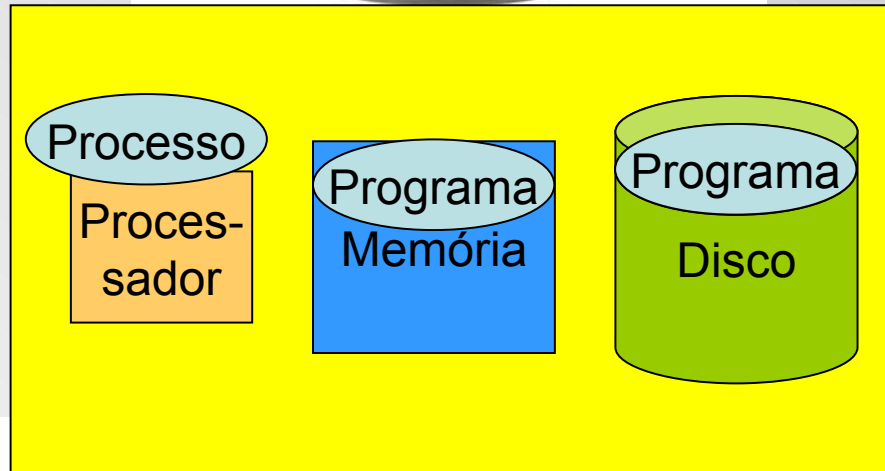
Gerenciamento de Memória

- Idealmente, o que todo programador deseja é dispor de uma memória que seja
 - grande
 - rápida
 - não volátil
- **Hierarquia de memórias**
 - pequena quantidade de memória rápida, de alto custo - **cache**
 - quantidade considerável de **memória principal** de velocidade média, custo médio
 - gigabytes de armazenamento em **disco** de velocidade e custo baixos
- O gerenciador de memória trata a hierarquia de memórias

Software

Como rodar um programa?

1. PC aponta para o endereço de memória onde o programa foi escrito
2. Processador executa instruções do programa trazidas da memória



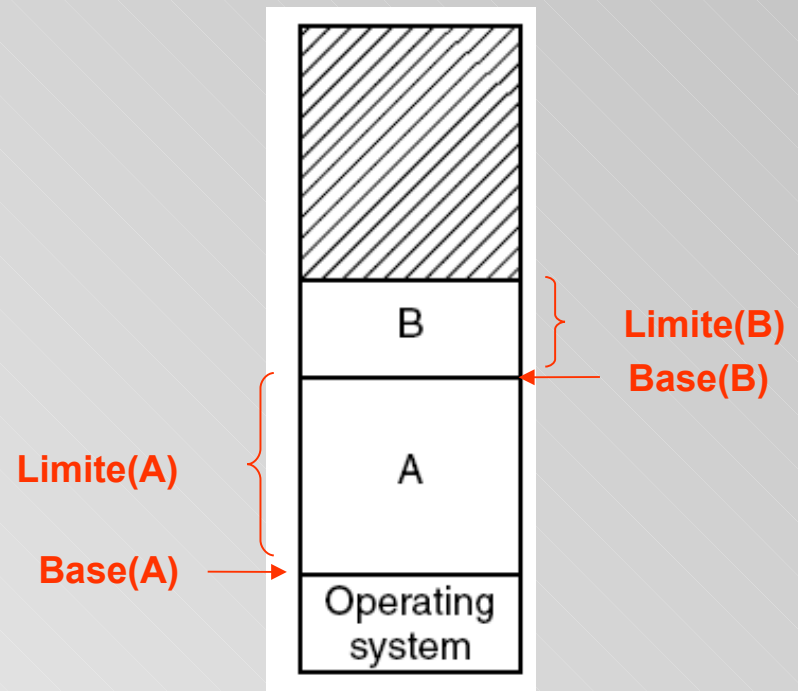
1. Dado o comando para executar um programa, é realizada uma seqüência de instruções para copiar código e dados do programa objeto do disco para a memória principal

Relocação e Proteção

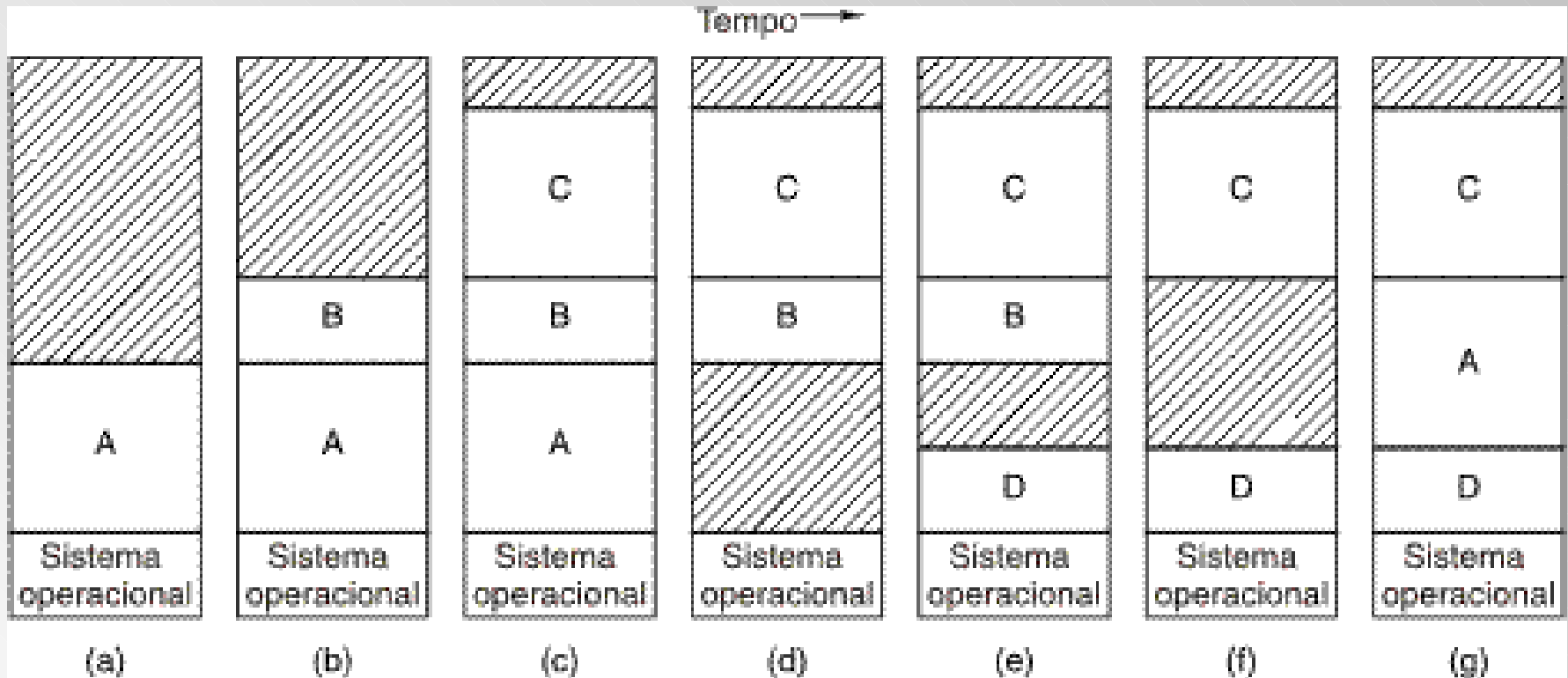
- Não se sabe com certeza onde o programa será carregado na memória
 - Localizações de endereços de variáveis e de código de rotinas não podem ser absolutos
- Uma solução para relocação e proteção: uso de valores base e limite

Registradores Base e Limite

- Usados para dar a cada processo um espaço de endereçamento separado (protegido) – partição
- Base = início da partição
- Limite = tamanho da partição

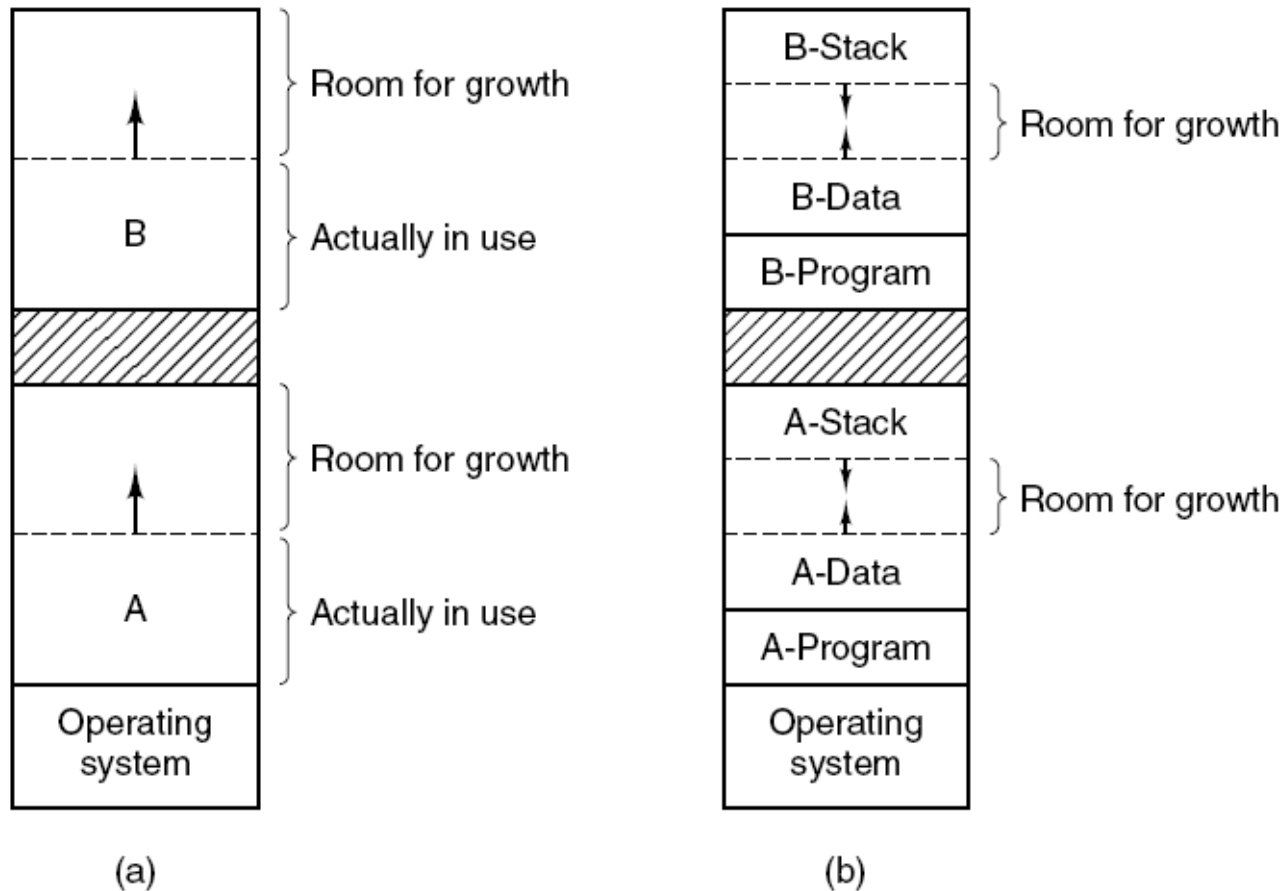


Swapping: Troca de Processos (1)



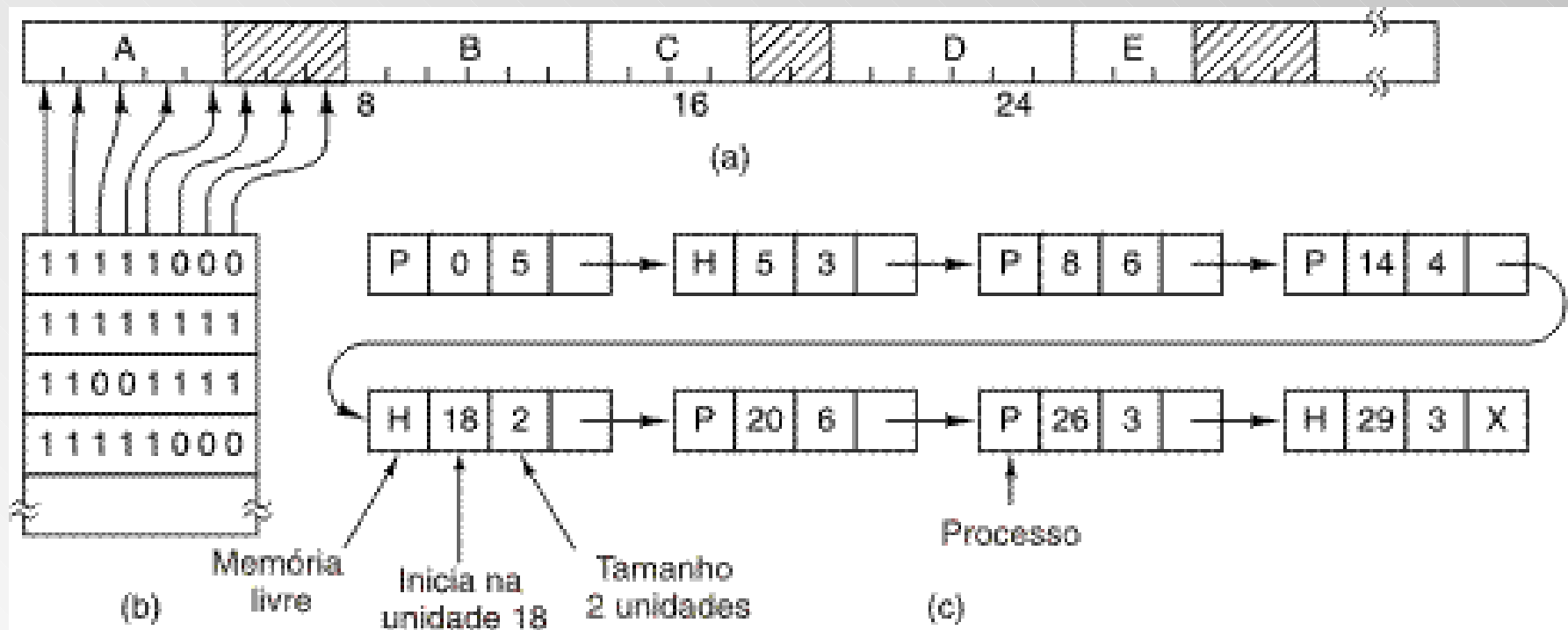
- Alterações na alocação de memória à medida que processos entram e saem da memória
- Regiões sombreadas correspondem a regiões de memória não utilizadas naquele instante

Troca de Processos (2)



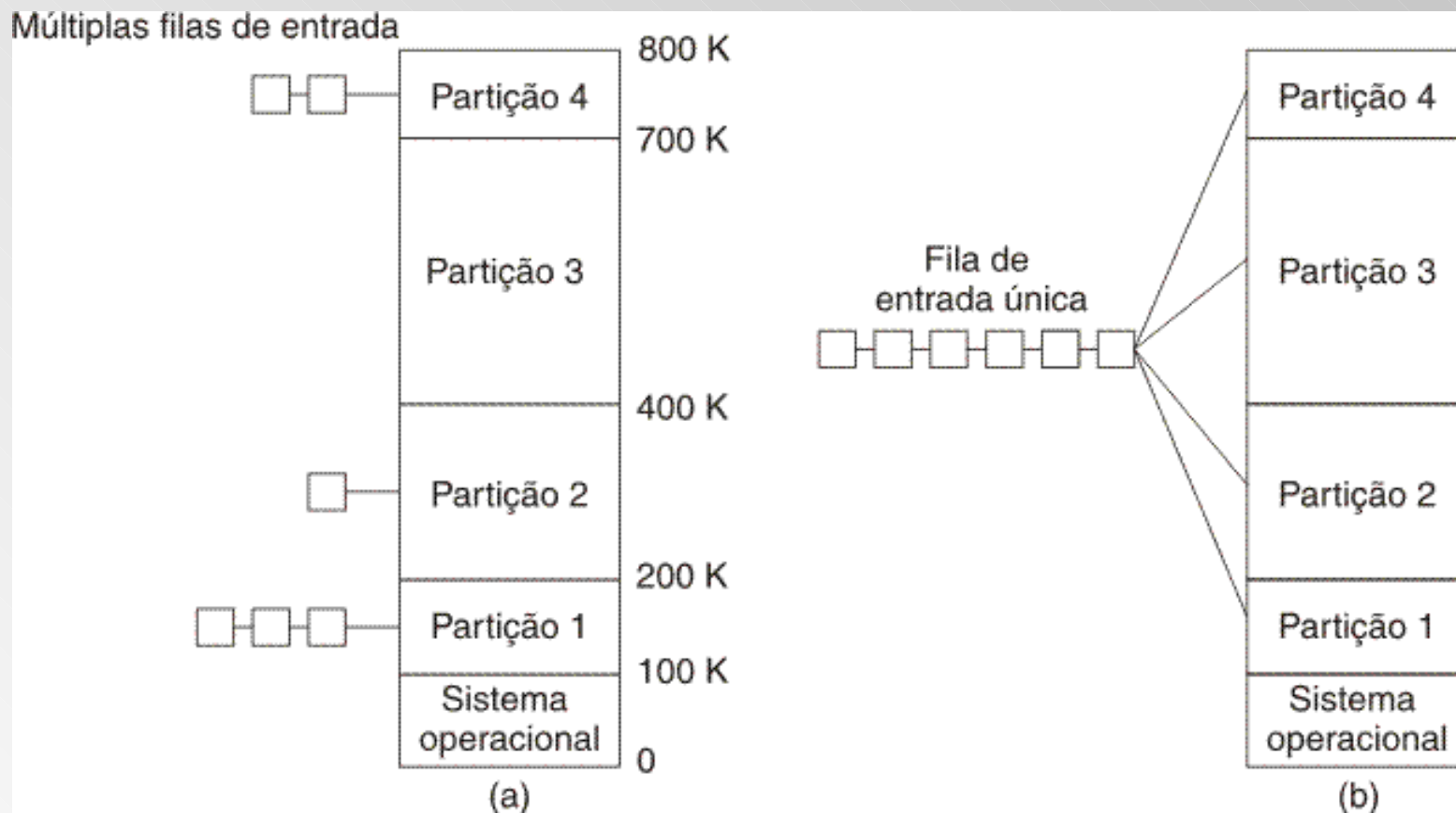
- a) Alocação de espaço para uma área de dados em expansão
- b) Alocação de espaço para uma pilha e uma área de dados, ambos em expansão

Gerenciamento de Memória Livre



- a) Parte da memória com 5 segmentos de processos (P) e 3 segmentos de memória livre (Hole – H)
- b) Mapa de bits correspondente
- c) Mesmas informações em uma lista encadeada

Multiprogramação com Partições Fixas

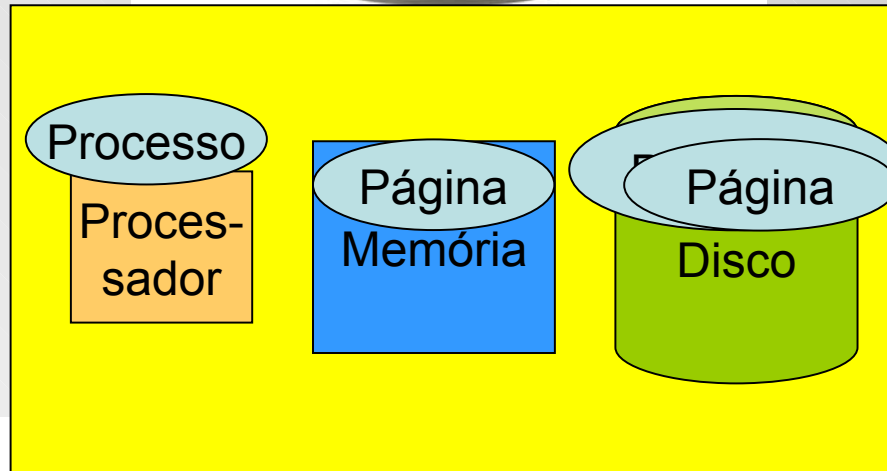


- Partições fixas de memória
 - a) filas de entrada separadas para cada partição
 - b) fila única de entrada

Software

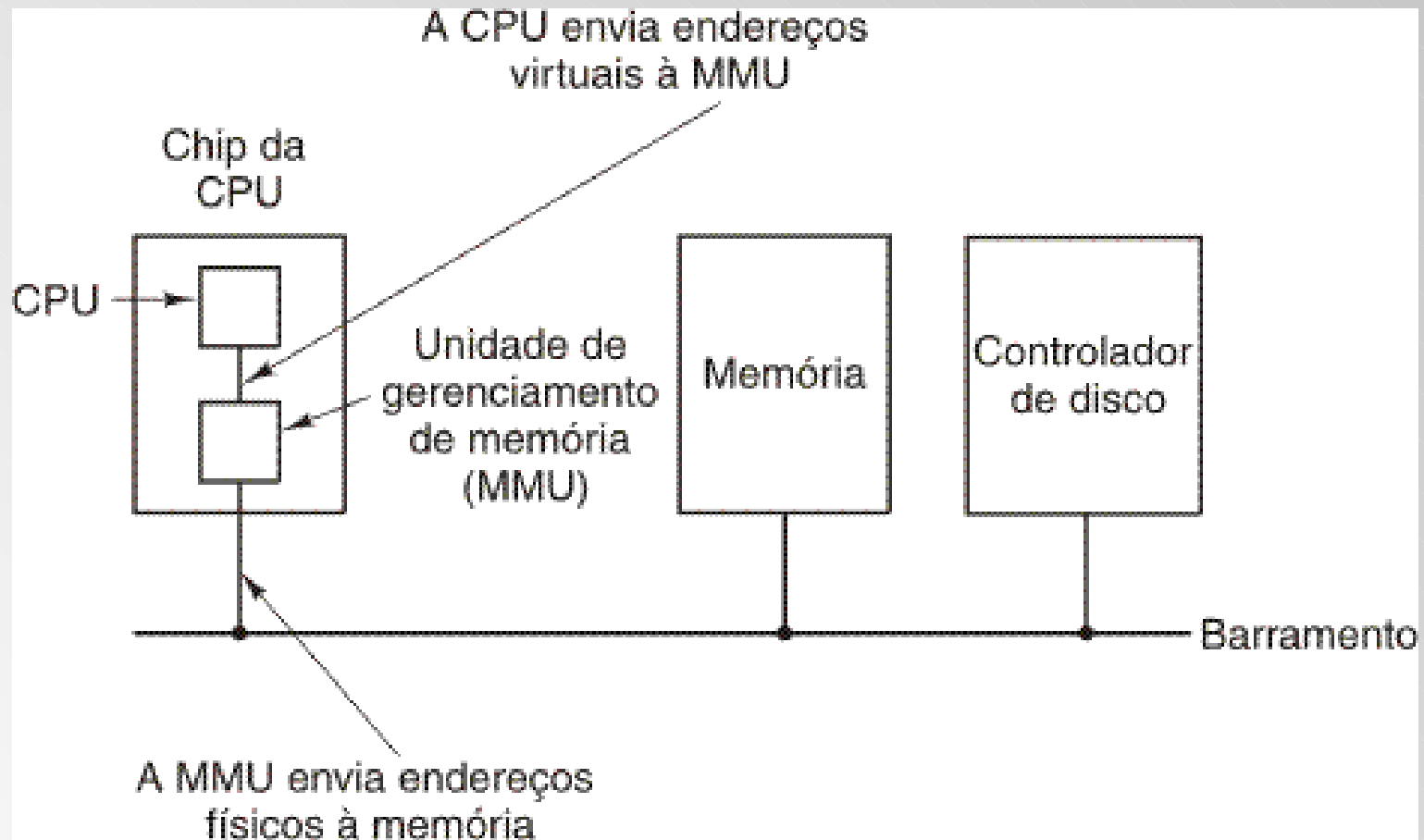
Como rodar um programa se ele for maior do que o espaço de memória disponível?

Os conceitos de “**Página**” e “**Memória Virtual**”



Memória Virtual

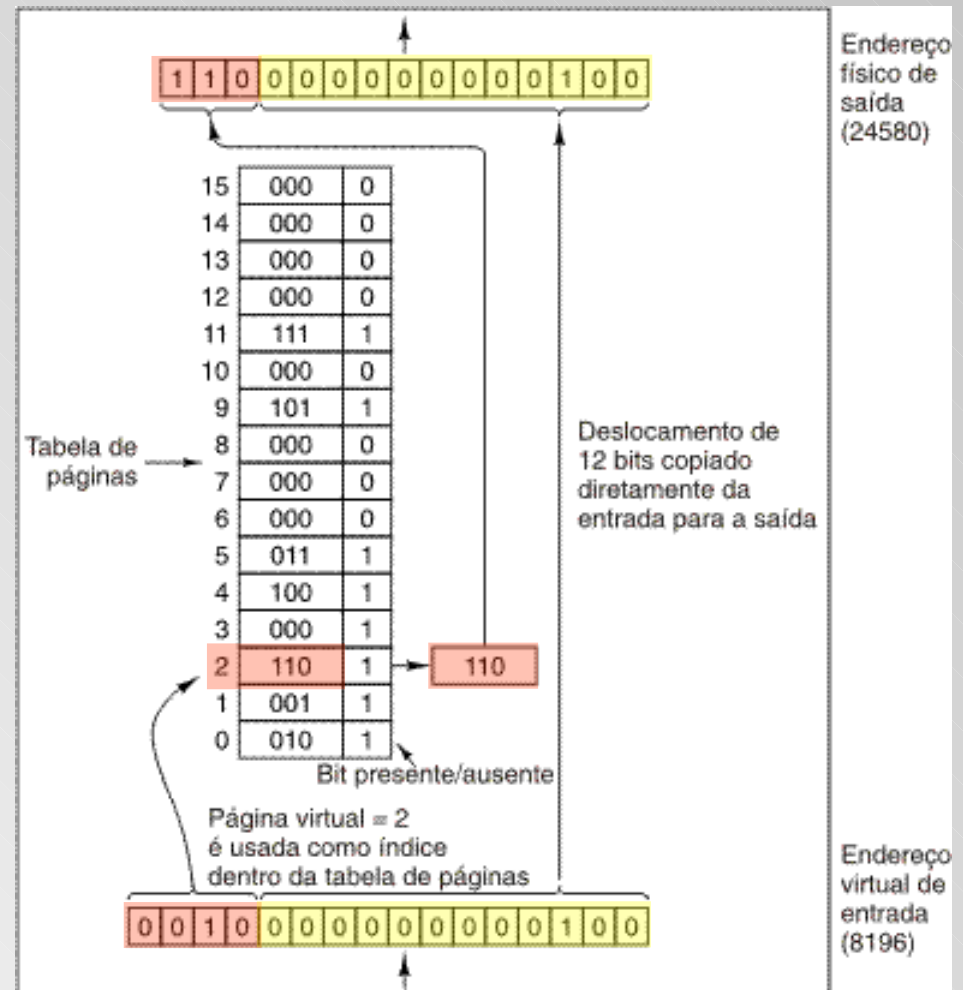
Paginação (1)



Localização e função da MMU (Memory Management Unit): nos dias de hoje é comum se localizar no chip da CPU

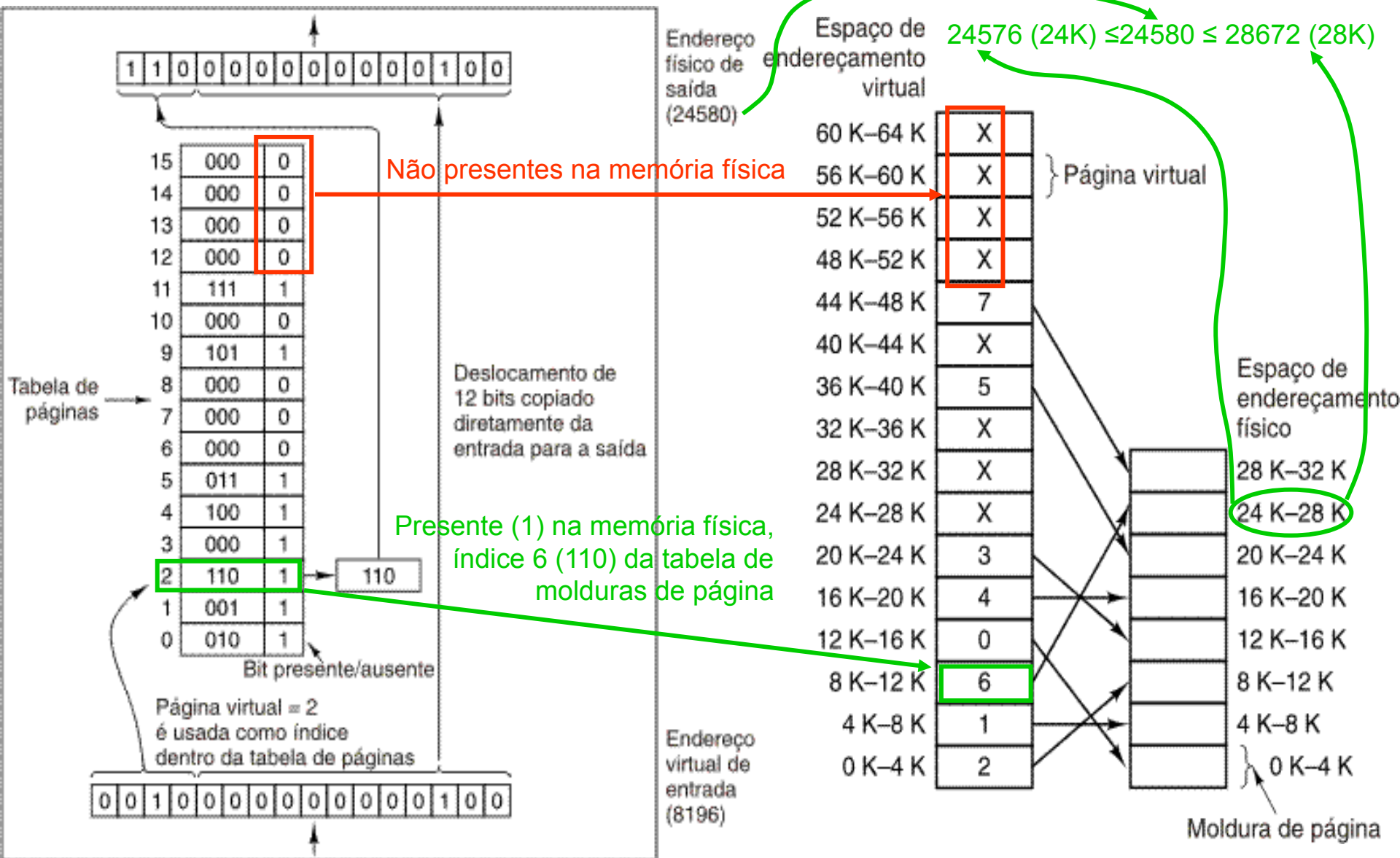
Paginação (2)

- Operação interna da MMU com 16 páginas de 4KB: determinação do endereço físico de memória em função do endereço virtual vindo da CPU (PC) e da tabela de páginas
- Tabela de páginas fica dentro da MMU

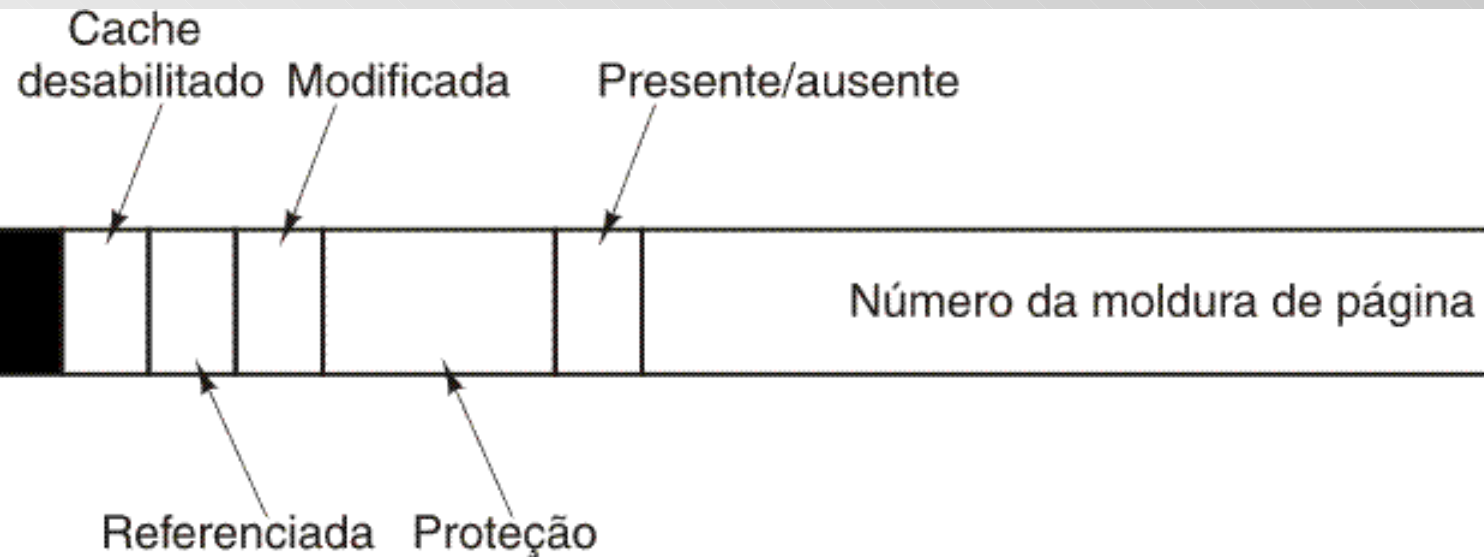


Paginação (3): Relação entre endereços virtuais e endereços de memória física dada pela tabela de páginas

Memória virtual 64K > Memória real 32K



Entrada típica de uma tabela de páginas



Tópicos

- ✓ Gerenciamento básico de memória
- ✓ Troca de processos
- ✓ Memória virtual
- ✓ Paginação
 - Aceleração da paginação
 - Substituição de páginas
 - Segmentação

Acelerando a Paginação

1. O mapeamento de endereço virtual para endereço físico deve ser rápido
2. Se o espaço de endereçamento virtual for grande, a tabela de páginas será grande

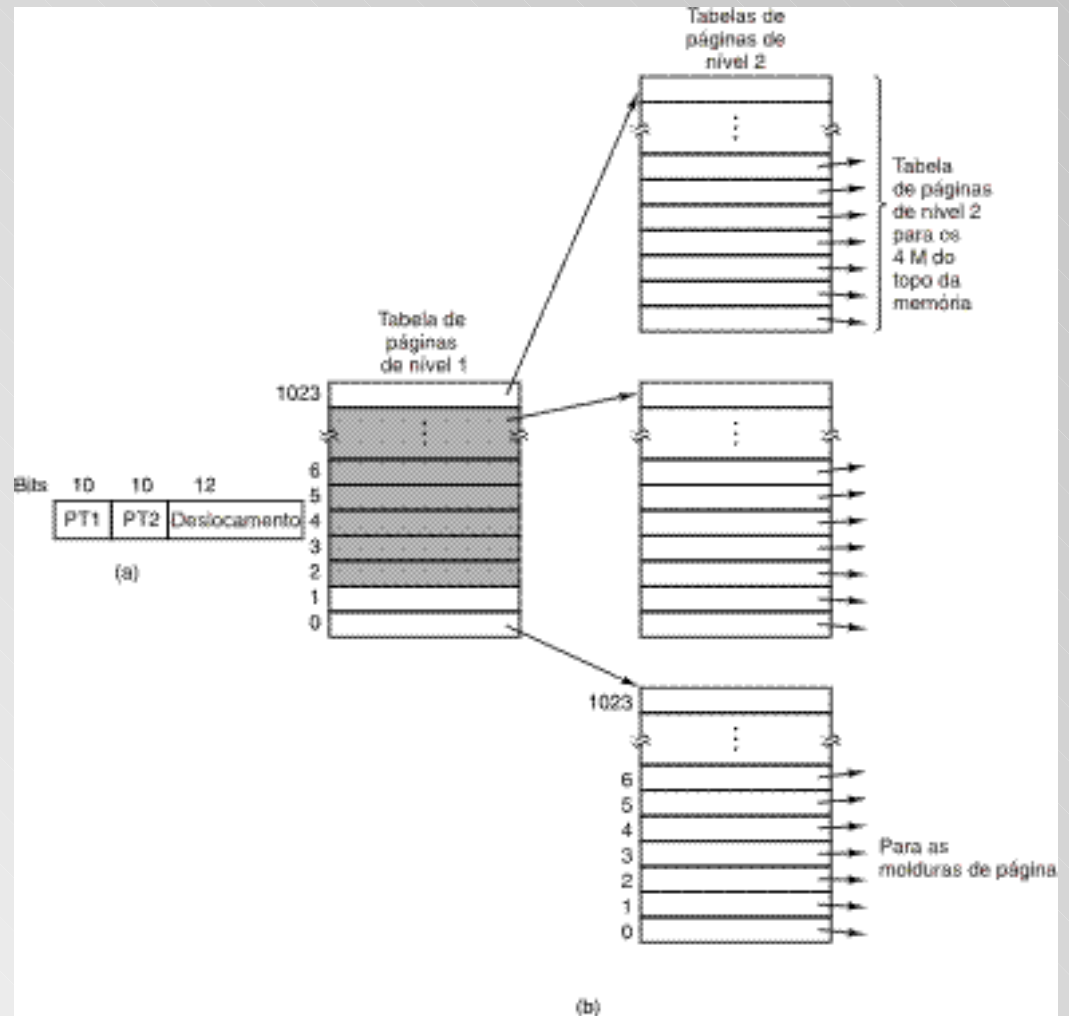
Memória Associativa ou TLB (Translation Lookaside Buffers)

- Tabela das traduções de endereços mais recentes
- Funciona como uma *cache* para tabelas de página

Valid	Virtual page	Modified	Protection	Page frame
1	140	1	RW	31
1	20	0	R X	38
1	130	1	RW	29
1	129	1	RW	62
1	19	0	R X	50
1	21	0	R X	45
1	860	1	RW	14
1	861	1	RW	75

Tabelas de Páginas Multi-Níveis

- Para minimizar o problema de continuamente armazenar tabelas de páginas muito grandes na memória



- a) Endereço de 32 bits com 2 campos (Page Table - PT1, PT2) para endereçamento de tabelas de páginas
- b) Tabelas de páginas com 2 níveis

Tópicos

- ✓ Gerenciamento básico de memória
- ✓ Troca de processos
- ✓ Memória virtual
- ✓ Paginação
- ✓ Aceleração da paginação
 - Substituição de páginas
 - Segmentação

Substituição de Páginas

- Falta de página (*page-fault*) na memória:
 - qual página deve ser removida?
 - alocação de espaço para a página a ser trazida para a memória
- A página modificada deve primeiro ser salva
 - se não tiver sido modificada é apenas sobreposta
- Melhor não escolher uma página que está sendo muito usada
 - provavelmente precisará ser trazida de volta logo

Substituição de Páginas: Algoritmos

- Ótimo: procura substituir o mais tarde possível - impraticável
- First-In, First-Out (FIFO)
- Not Recently Used (NRU)
- Segunda chance (SC)
- Least Recently Used (LRU)
- Conjunto de trabalho (Working Set - WS)
- Relógio (Clock)
- WSClock

Primeira a Entrar, Primeira a Sair (FIFO)

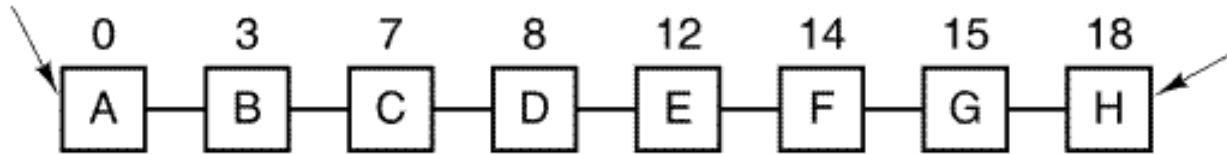
- Mantém uma lista encadeada de todas as páginas
 - página mais antiga na cabeça da lista
 - página que chegou por último na memória no final da lista
- Na ocorrência de falta de página
 - página na cabeça da lista é removida
 - nova página adicionada no final da lista
- Desvantagem
 - página há mais tempo na memória pode ser usada com muita frequência

Não Usada Recentemente (NUR/NRU)

- Cada página tem os bits Referenciada (R) e Modificada (M)
 - Bits são colocados em 1 quando a página é referenciada e modificada
- As páginas são classificadas
 - Classe 0: não referenciada (0), não modificada (0)
 - Classe 1: não referenciada (0), modificada (1)
 - Classe 2: referenciada (1), não modificada (0)
 - Classe 3: referenciada (1), modificada (1)
- NUR remove página aleatoriamente
 - da classe de ordem mais baixa que não esteja vazia

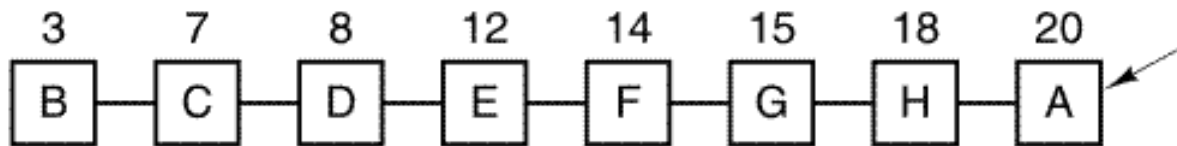
Segunda Chance (SC)

Primeira página carregada



Página mais recentemente carregada

(a)

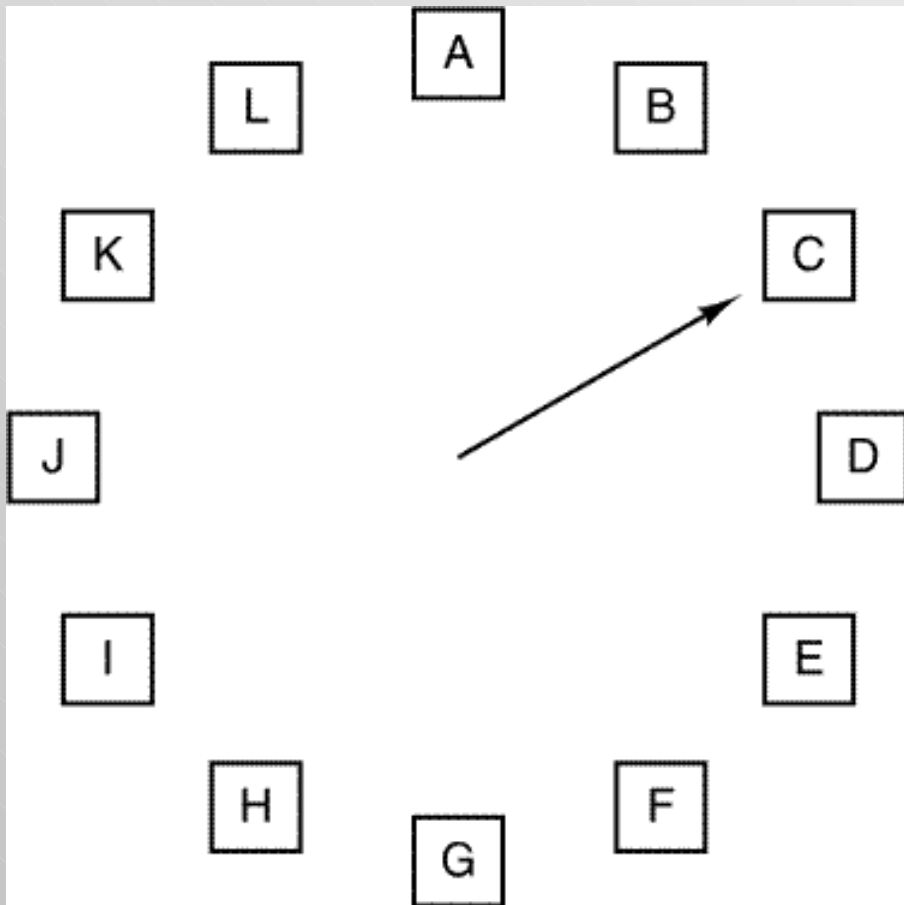


Página A é tratada como página mais recentemente carregada

(b)

- Operação do algoritmo segunda chance
 - a) lista de páginas em ordem FIFO
 - b) estado da lista em situação de **falta de página no instante 20**, com o bit R da página A em 1
 - números representam instantes de carregamento das páginas na memória

Relógio



Quando ocorre uma falta de página, a página apontada é examinada.

A atitude a ser tomada depende do bit R:

R = 0: Retira a página,

R = 1: Faz R = 0 e avança o ponteiro.

Menos Recentemente Usada (MRU/LRU)

- Considera que páginas usadas recentemente logo serão usadas novamente
 - retira da memória a página que há mais tempo não é usada
- Uma lista encadeada de páginas deve ser mantida
 - página mais recentemente usada no início da lista, menos usada no final da lista
 - atualização da lista a cada referência à memória
 - **hardware especial** pode ser usado

LRU em Software

- NFU (Not Frequently Used) → cada página tem um contador que é incrementado quando é usada
 - Nunca esquece nada
- Envelhecimento (*Aging*)
 - Próximo ao **algoritmo ótimo**
 - Cada página mantém um contador (por ex., com 8 bits)
 - A cada interrupção de relógio
 - Desloca o contador para a direita
 - Acrescenta o valor de R ao bit mais significativo

O algoritmo de envelhecimento

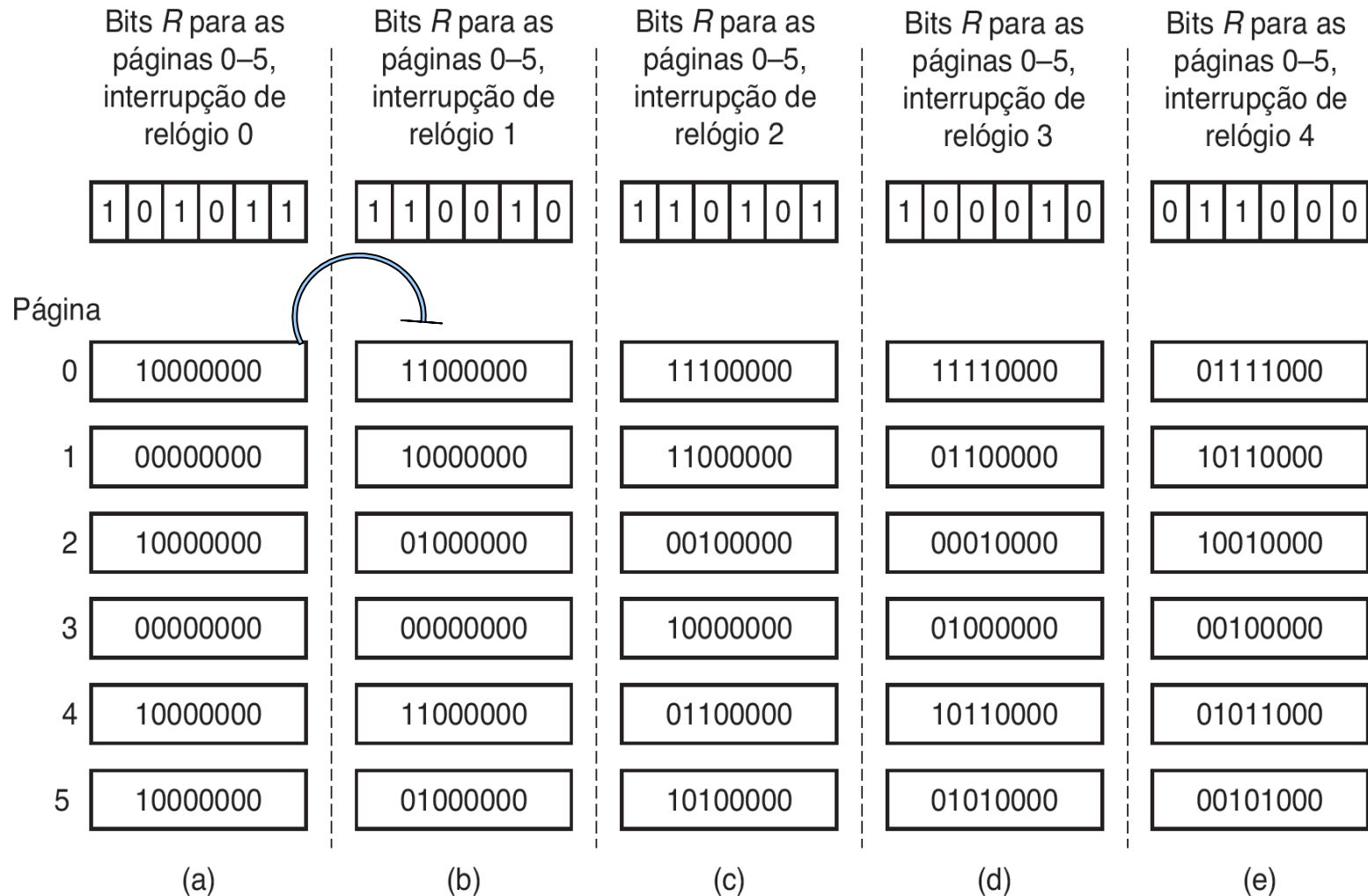
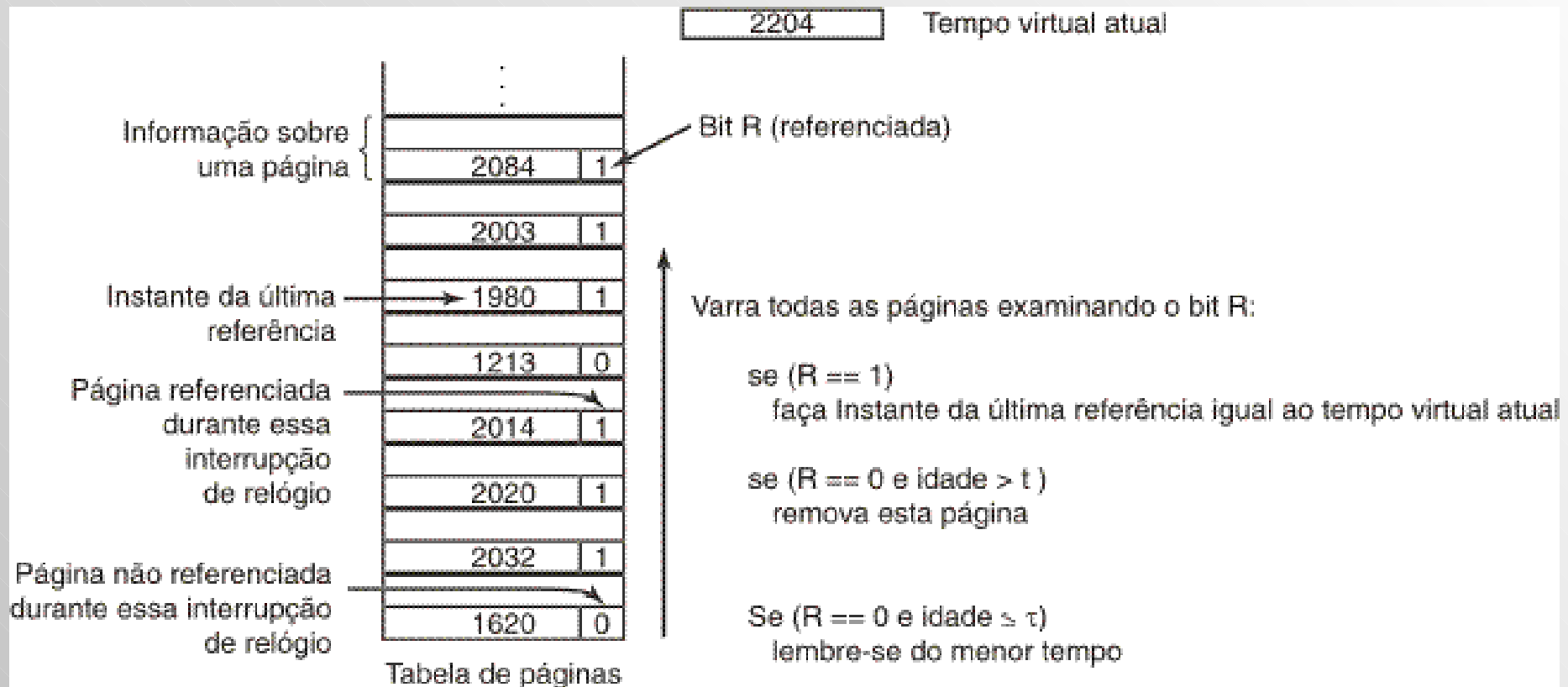


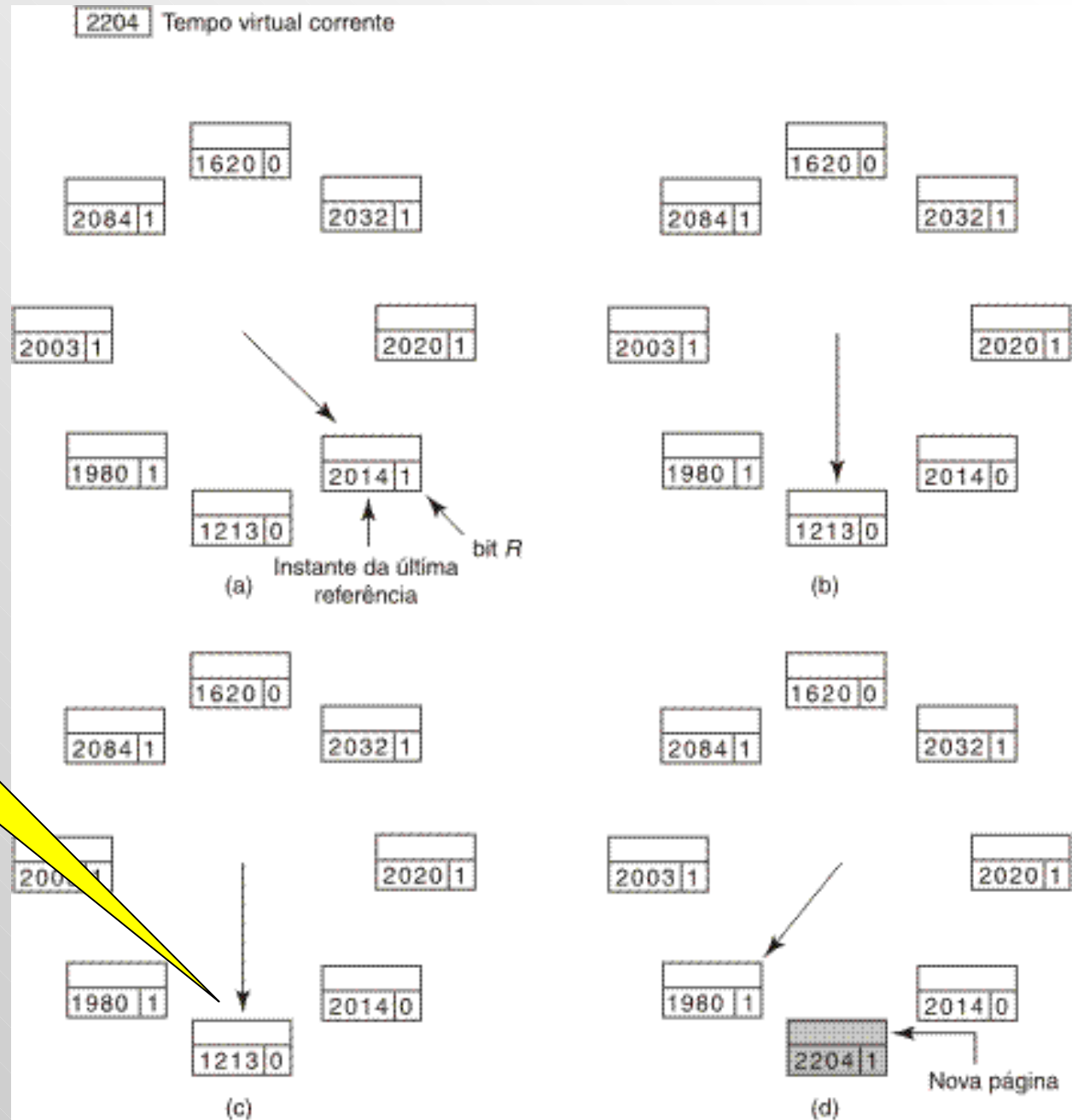
Figura 3.17 O algoritmo de envelhecimento simula o LRU em software. São mostradas seis páginas para cinco interrupções de relógio. As cinco interrupções de relógio são representadas de (a) até (e).

Conjunto de Trabalho (WS)



WSClock

Prioriza
páginas
limpas!



O Algoritmo Ótimo!

- O algoritmo FIFO sempre seleciona a **página mais antiga** para ser trocada – First-In, First-Out
- O algoritmo LRU sempre seleciona a **página que não vem sendo usada há mais tempo** – Least Recently Used (Menos Recentemente Usada - MRU)
- O algoritmo ótimo sempre seleciona a **página que não será usada por mais tempo...**
 - Mas como o SO pode determinar quando cada uma das páginas será referenciada? Daqui a 10 instruções, 100 instruções, 1000 instruções...
 - **IMPOSSÍVEL!!!**

Revisão dos Algoritmos de Substituição de Página

Algoritmo	Comentário
Ótimo	Não implementável, mas útil como um padrão de desempenho
NUR (não usada recentemente)	Muito rudimentar
FIFO (primeira a entrar, primeira a sair)	Pode descartar páginas importantes
Segunda chance	Algoritmo FIFO bastante melhorado
Relógio	Realista
MRU (menos recentemente usada)	Excelente algoritmo, porém difícil de ser implementado de maneira exata
NFU (não freqüentemente usada)	Aproximação bastante rudimentar do MRU
Envelhecimento (<i>aging</i>)	Algoritmo bastante eficiente que se aproxima bem do MRU
Conjunto de trabalho	Implementação um tanto cara
WSClock	Algoritmo bom e eficiente

Comparação

- FIFO

Procura página
na memória

Memória
com 3
molduras
(frames) de
páginas



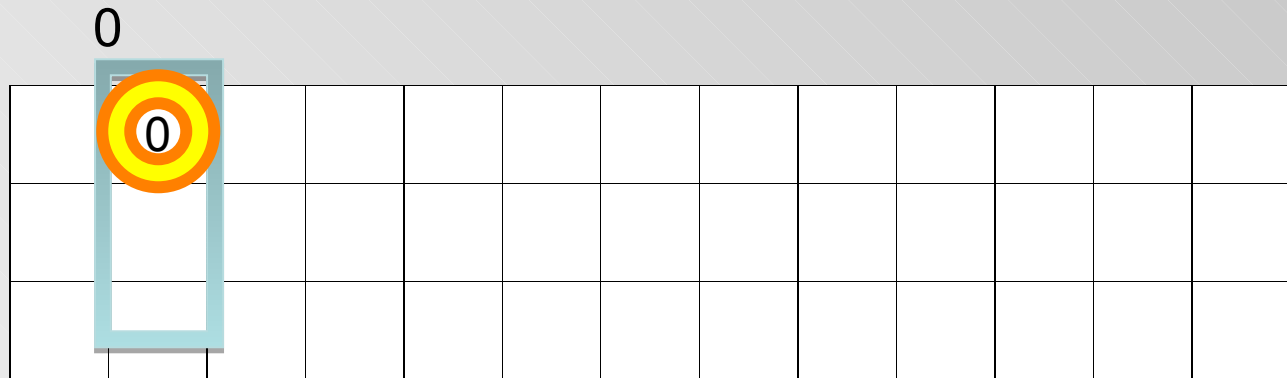
Memória
com 4
molduras
(frames) de
páginas



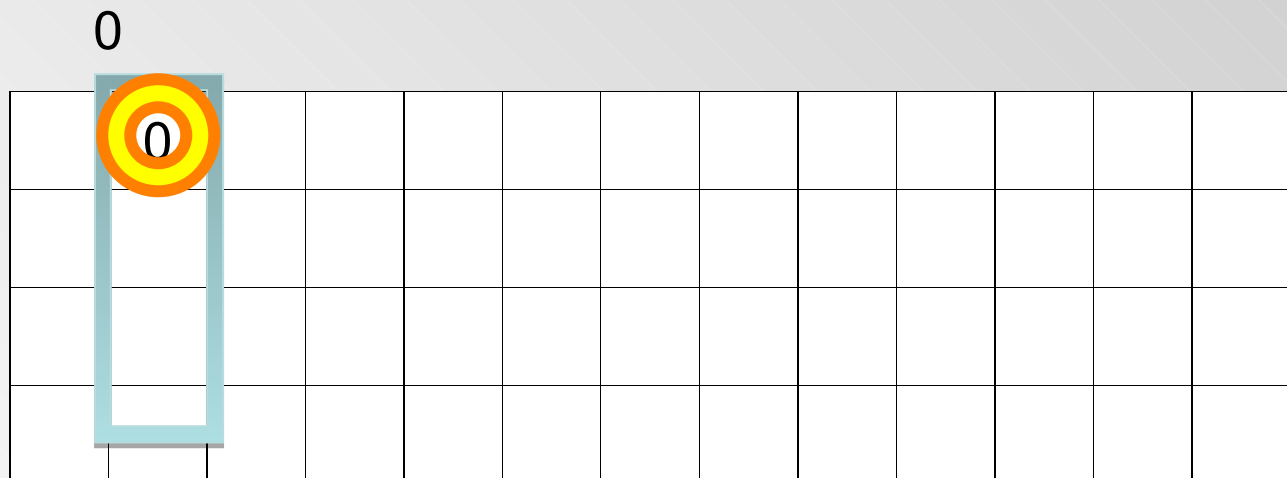
Comparação

- FIFO

3 molduras
(frames) de
páginas



4 molduras
(frames) de
páginas

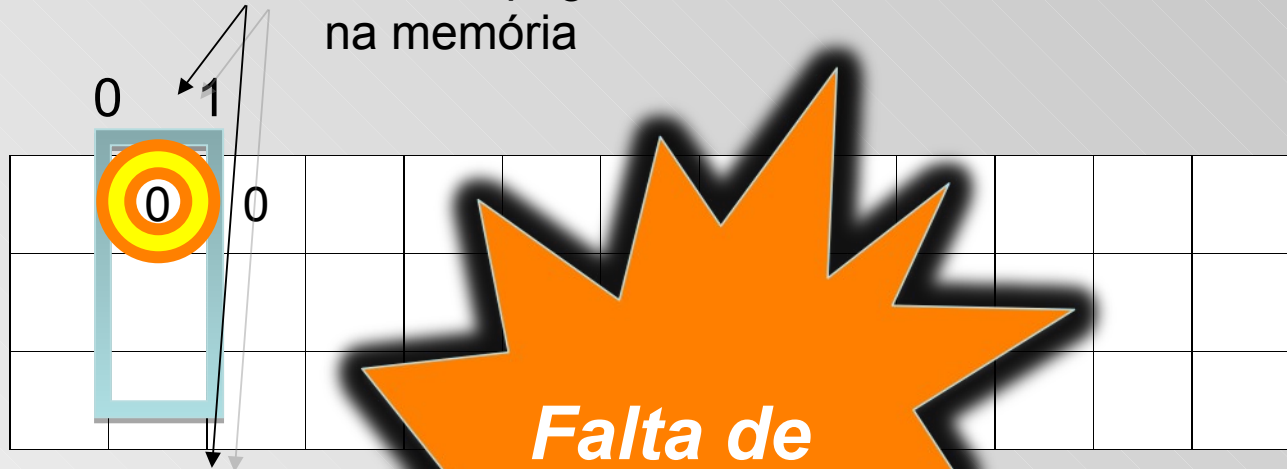


Comparação

- FIFO

Procura página
na memória

3 molduras
(frames) de
páginas



4 molduras
(frames) de
páginas



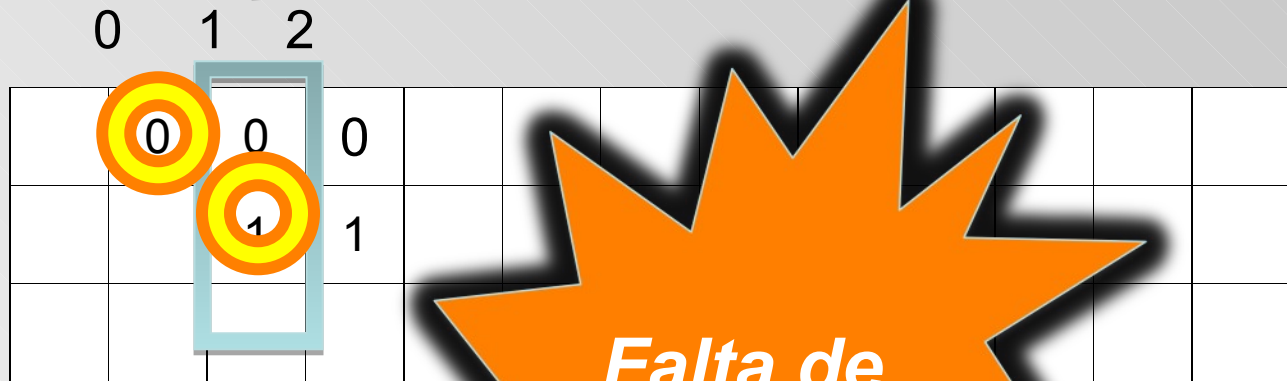
*Falta de
página*

Comparação

- FIFO

Procura página
na memória

3 molduras
(frames) de
páginas



4 molduras
(frames) de
páginas



Comparação

- FIFO

Procura página na memória

3 molduras (frames) de páginas



4 molduras (frames) de páginas

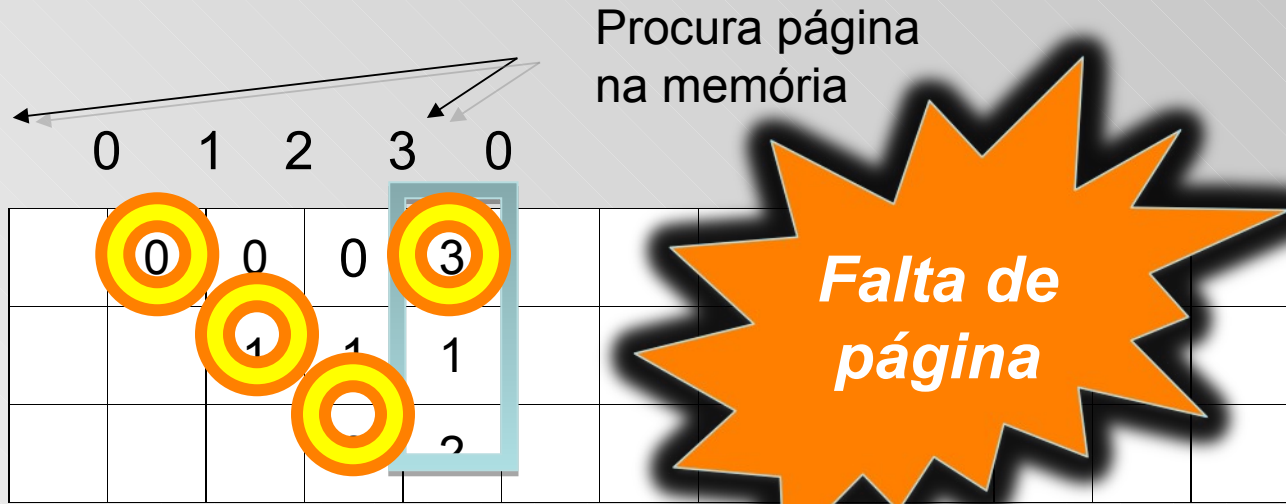


Falta de página

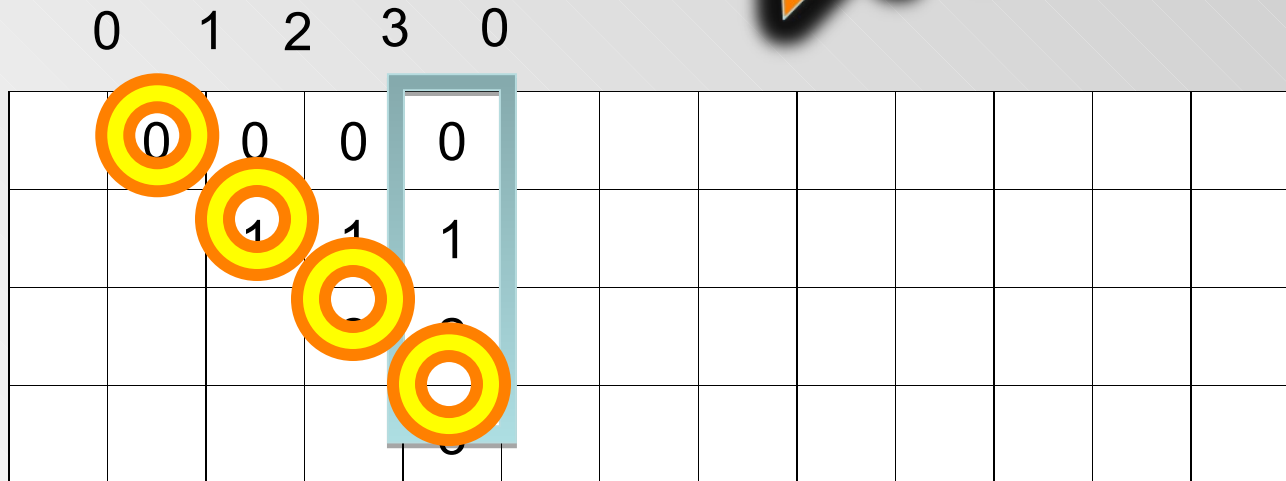
Comparação

- FIFO

3 molduras
(frames) de
páginas



4 molduras
(frames) de
páginas

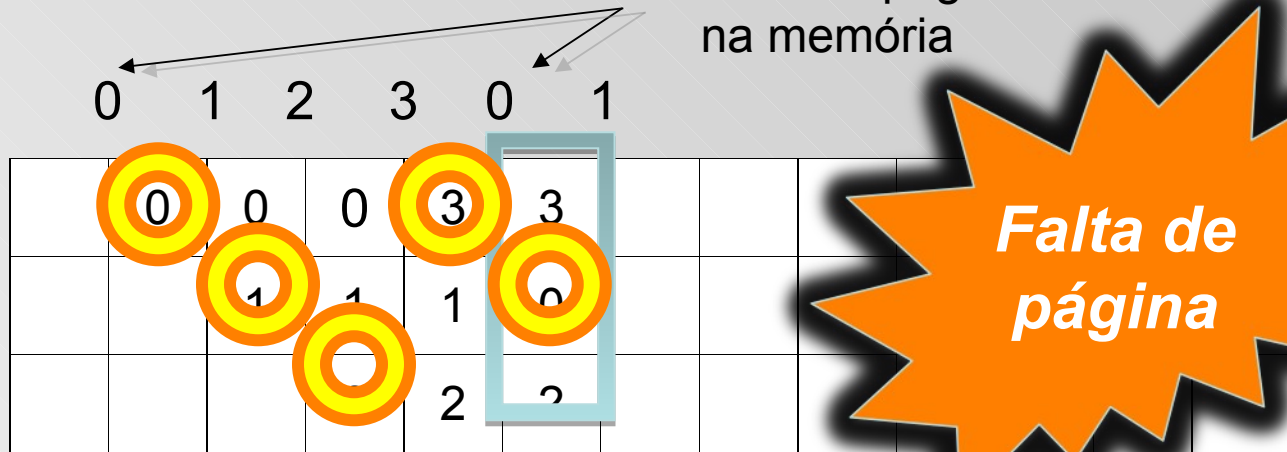


Comparação

- FIFO

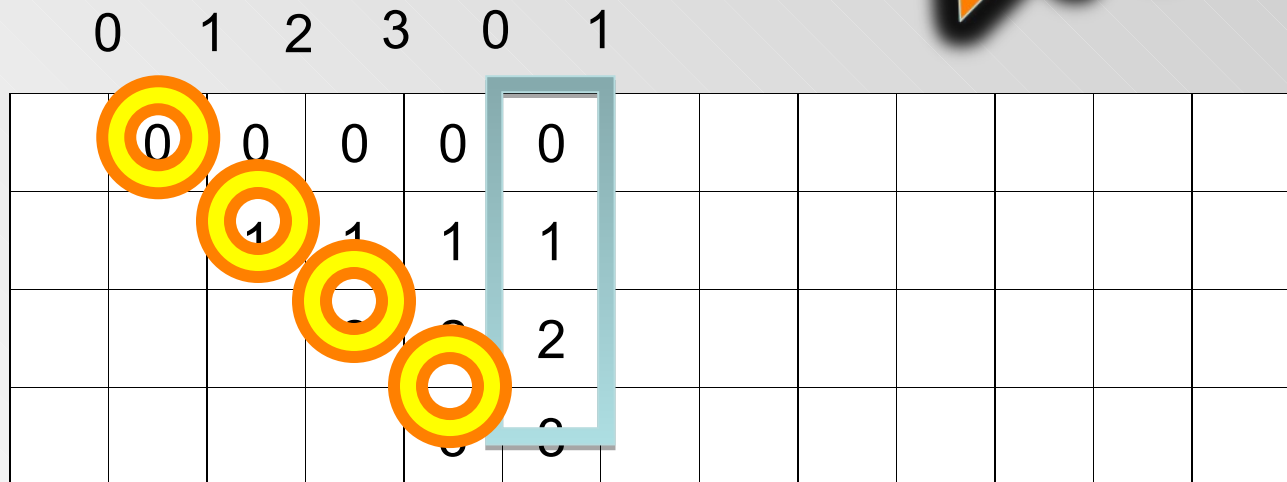
Procura página na memória

3 molduras (frames) de páginas



Falta de página

4 molduras (frames) de páginas



Comparação

- FIFO

Procura página na memória

3 molduras (frames) de páginas

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

4 molduras (frames) de páginas

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

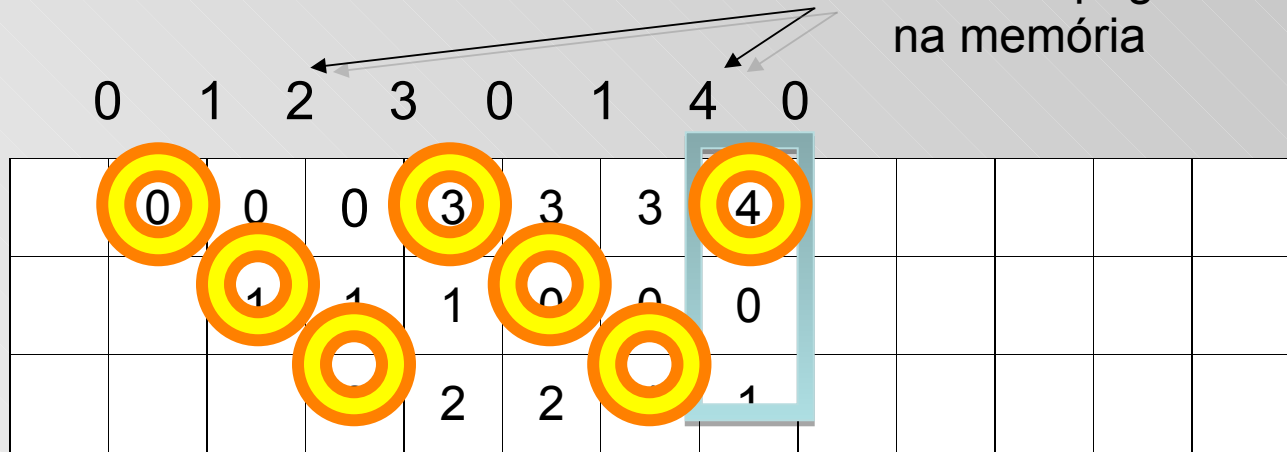
Falta de página

Comparação

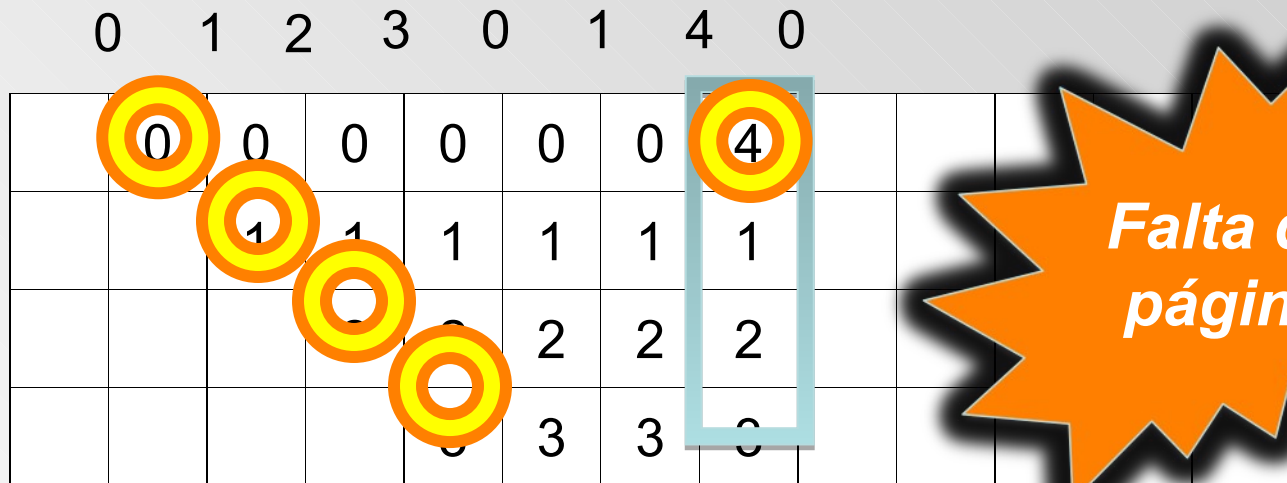
- FIFO

Procura página na memória

3 molduras (frames) de páginas



4 molduras (frames) de páginas



Falta de página

Comparação

- FIFO

Procura página na memória

3 molduras (frames) de páginas

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

4 molduras (frames) de páginas

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

Falta de página

Comparação

- FIFO

Procura página na memória

3 molduras (frames) de páginas

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

4 molduras (frames) de páginas

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

Falta de página

Comparação

- FIFO

Procura página na memória

3 molduras (frames) de páginas



4 molduras (frames) de páginas



Comparação

- FIFO

Procura página na memória

3 molduras (frames) de páginas

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

4 molduras (frames) de páginas

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

Falta de página

Comparação

- FIFO

Solicitações de página=12

Faltas de página=9; taxa de falta=9/12=75%; taxa de sucesso=3/12=25%

3 molduras
(frames) de
páginas

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

4 molduras
(frames) de
páginas

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

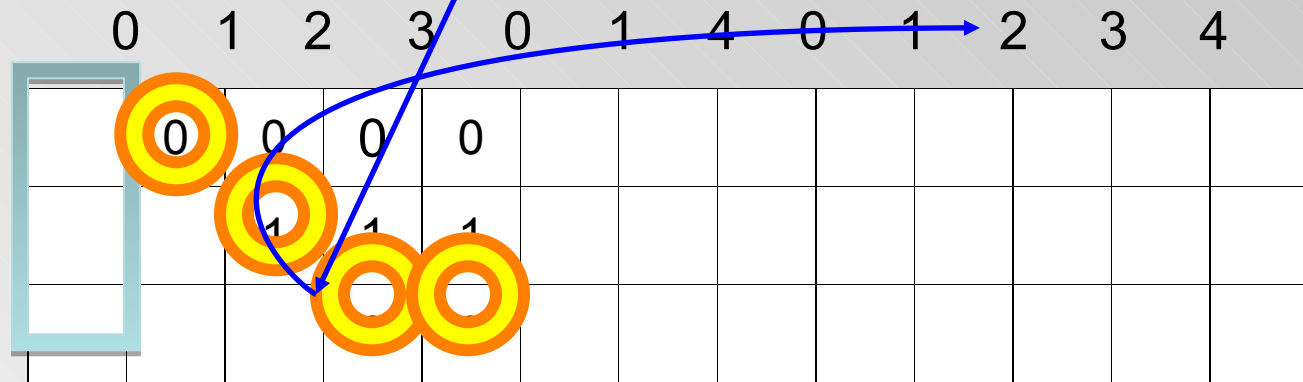
Faltas de página=10; taxa de falta=10/12=83,3%; taxa de sucesso=2/12=16,7%

Comparação

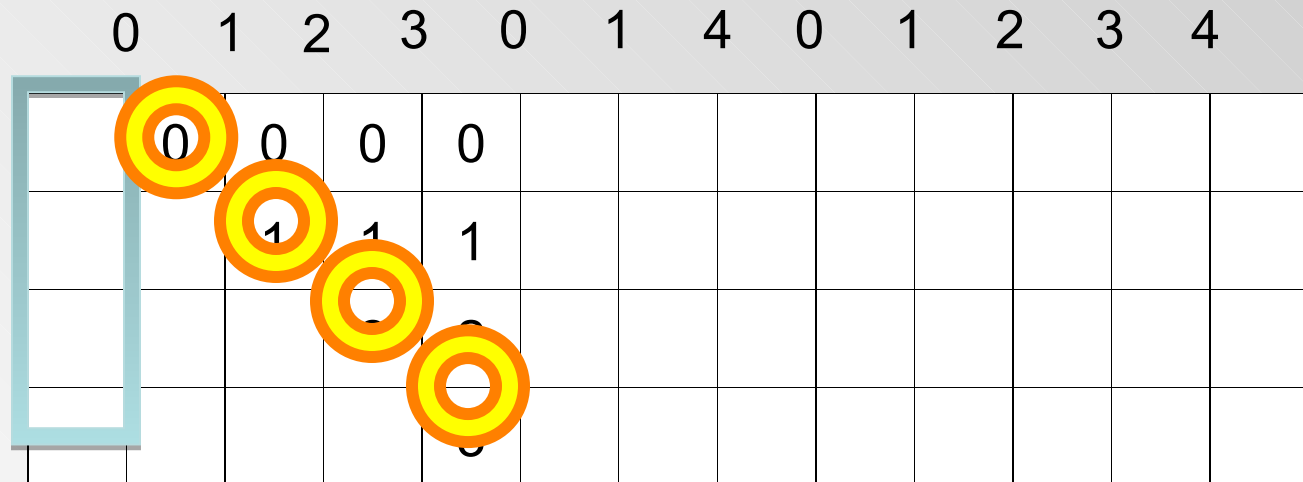
- Algoritmo Ótimo

Sabendo que 2 só será usada mais tarde....

3 molduras
(frames) de
páginas



4 molduras
(frames) de
páginas



Comparação

- Algoritmo Ótimo

Faltas de página=7; taxa de falta=7/12=58,3%; taxa de sucesso=5/12=41,7%

3 molduras
(frames) de
páginas

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

4 molduras
(frames) de
páginas

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

Faltas de página=6; taxa de falta=6/12=50%; taxa de sucesso=6/12=50%

Comparação

- LRU

Faltas de página=10; taxa de falta=10/12=83,3%; taxa de sucesso=2/12=16,7%

3 molduras
(frames) de
páginas

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

4 molduras
(frames) de
páginas

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

Faltas de página=8; taxa de falta=8/12=66,7%; taxa de sucesso=4/12=33,3%

Comparação

- FIFO
 - 3 molduras: sucesso = 25%
 - 4 molduras: sucesso = 16,7%
- Ótimo
 - 3 molduras: sucesso = 41,7%
 - 4 molduras: sucesso = 50%
- LRU
 - 3 molduras: sucesso = 16,7%
 - 4 molduras: sucesso = 33,3%

É de se esperar que quanto mais molduras, maior o sucesso, ou seja, menor a chance de "falta de página"

Anomalia de Belady

Todas as molduras de páginas estão vazias inicialmente

		0	1	2	3	0	1	4	0	1	2	3	4	
Página mais nova		0	1	2	3	0	1	4	4	4	2	3	3	
			0	1	2	3	0	1	1	1	4	2	2	
Página mais velha				0	1	2	3	0	0	0	1	4	4	
		P	P	P	P	P	P	P			P	P		9 faltas de página

(a)

		0	1	2	3	0	1	4	0	1	2	3	4	
Página mais nova		0	1	2	3	3	3	4	0	1	2	3	4	
			0	1	2	2	2	3	4	0	1	2	3	
Página mais velha				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 faltas de página

(b)

Esperado: quanto mais molduras de página a memória possuir, menos faltas de página o programa terá.

Anomalia: neste exemplo, o algoritmo de substituição FIFO tem mais faltas de página (10P) para mais molduras (4)...

- FIFO com 3 molduras de página
- FIFO com 4 molduras de página
- *P* mostra quais referências de página causaram faltas de página

Controle de Carga

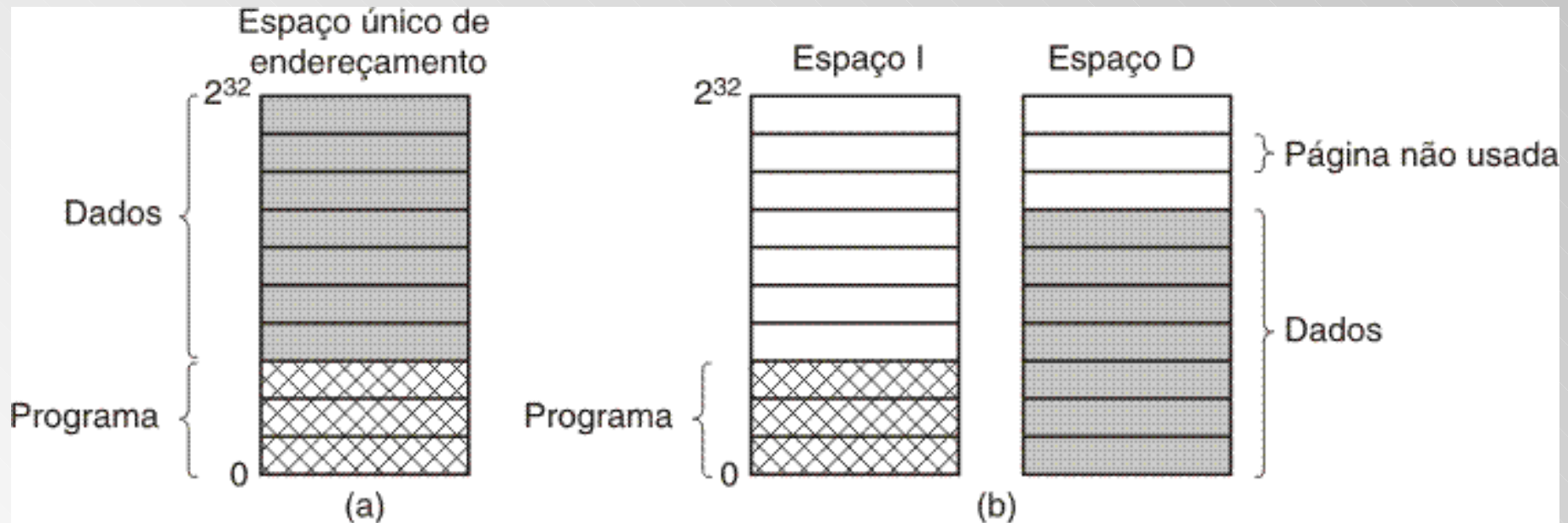
- Mesmo com um bom projeto, o sistema ainda pode sofrer **paginação excessiva** (*thrashing*)
- Quando
 - alguns processos precisam de mais memória
 - mas nenhum processo precisa de menos (ou seja, nenhum pode ceder páginas)
- **Solução:**
Reduzir o número de processos que competem pela memória
 - levar alguns deles para disco (swap) e liberar a memória a eles alocada
 - **reconsiderar grau de multiprogramação**

Tamanho de Página

Tamanho de página pequeno

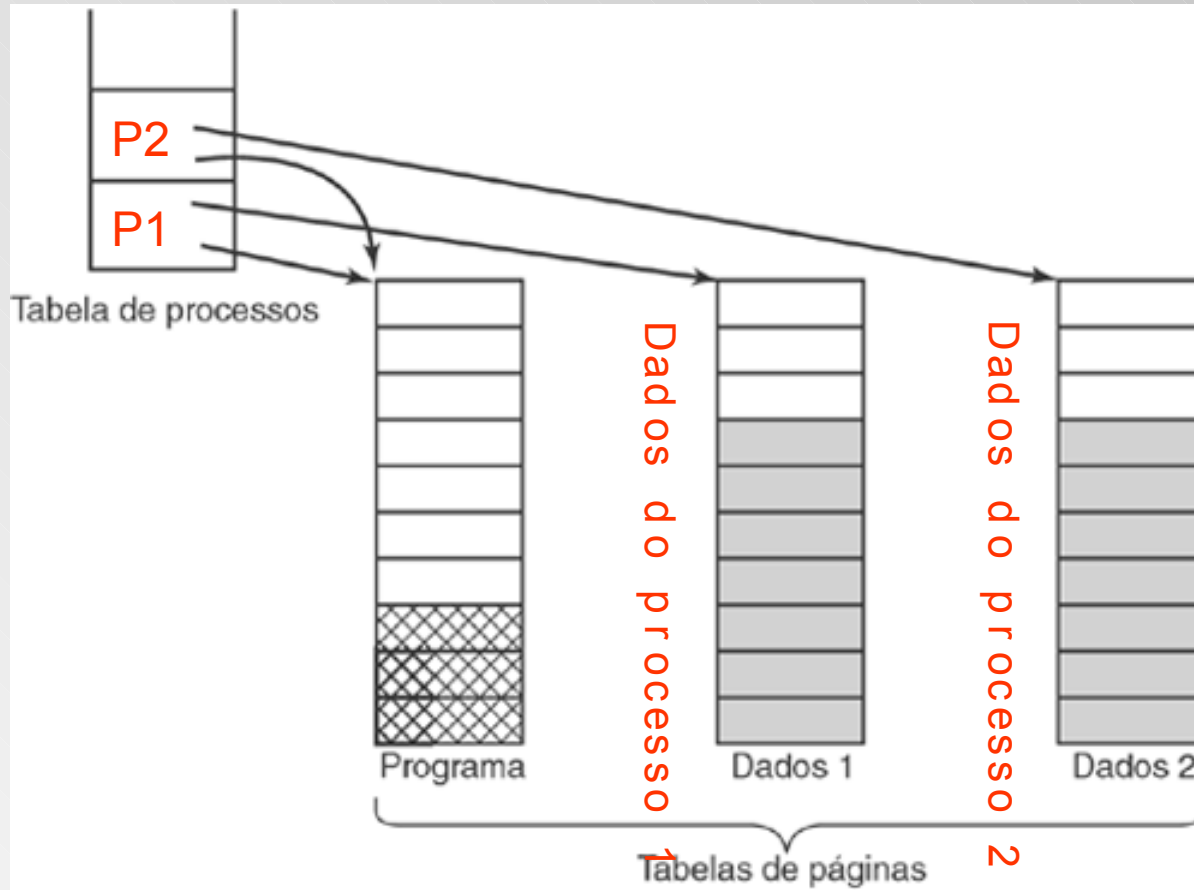
- Vantagens
 - menos fragmentação interna
 - menos programa não usado na memória
- Desvantagens
 - programas precisam de mais páginas, tabelas de página maiores

Espaços Separados de Instruções e Dados



- a) Espaço de endereçamento único
- b) Espaços separados de instruções (I) e dados (D)

Páginas Compartilhadas



Dois processos que compartilham o mesmo código de programa e, por consequência, a mesma tabela de páginas para **instruções**

Envolvimento do S.O. com Paginação

1. Criação de processo

- determina tamanho do programa
- cria **tabela de página**
- cria **área de troca**

1. Execução de processo

- **inicia** MMU (Unidade de Gerenciamento de Memória) para novos processos
- **limpa** a TLB

Envolvimento do S.O. com Paginação

1. Ocorrência de falta de página
 - **determina endereço virtual** que causou a falta
 - **escolhe** e descarta, se necessário, página antiga
 - carrega página requisitada para a memória (swap)
1. Terminação de processo
 - Libera tabela de páginas, páginas e espaço em disco que as páginas ocupam

Lidando com uma falta de página (1)

1. Hardware desvia (trap) a execução para o **núcleo**, salvando o PC na pilha
2. Uma rotina de código Assembly salva registradores de uso geral e outras informações voláteis
3. S.O. determina página virtual requerida
4. S.O. **valida endereço** e **escalona remoção da página** da memória
5. Se página modificada, **escreve** para o disco
 1. Também envolve **escalonamento**

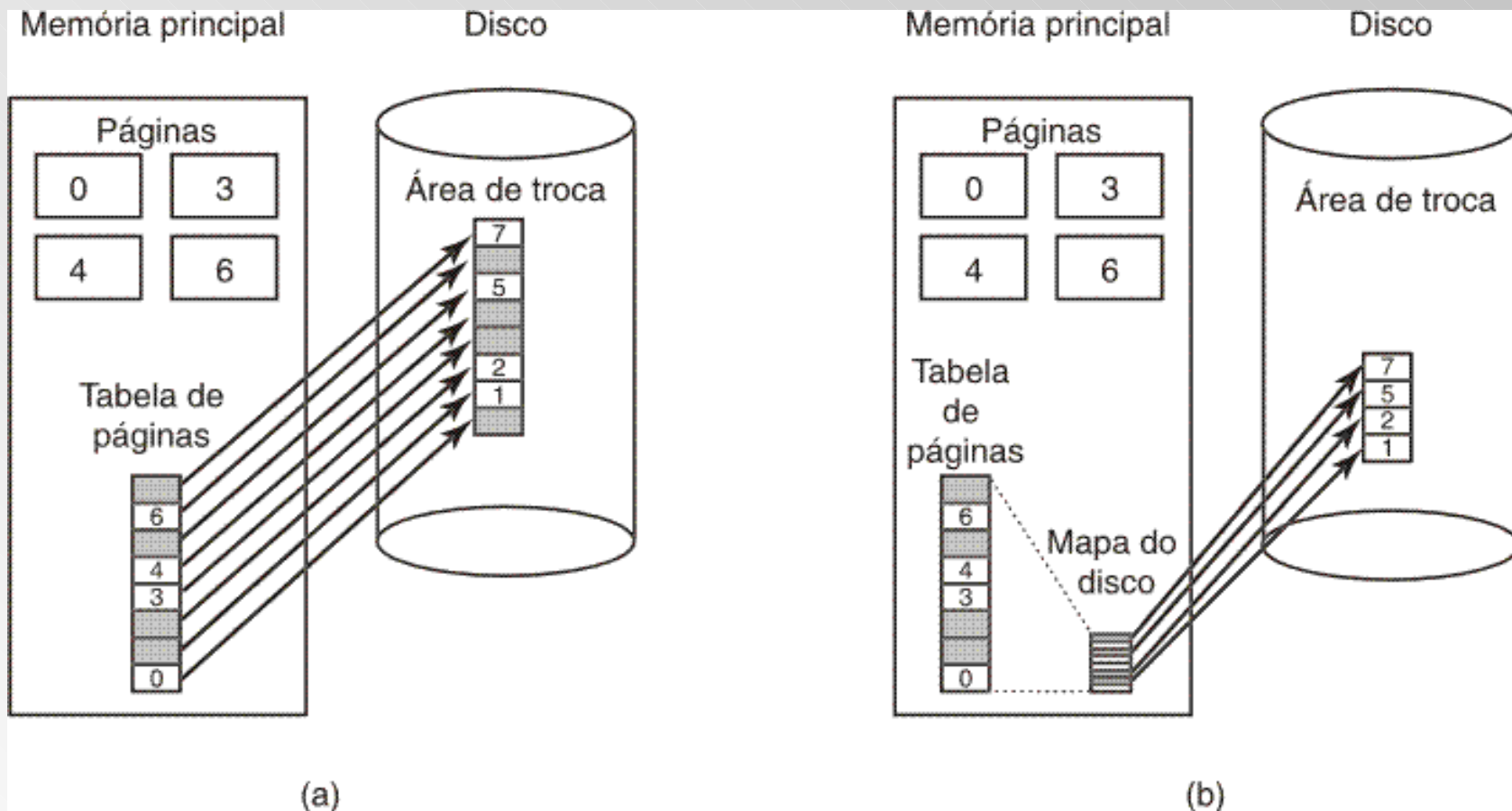
Lidando com uma falta de página (2)

1. Quando a **moldura** está **liberada**, o SO captura a página requerida no endereço definido no disco (**operação de E/S**)
2. Quando a **interrupção de disco** indica que a página chegou, as tabelas de página são atualizadas
3. PC aponta para a instrução que provocou a falta
4. O processo que sentiu a falta de página é escalonado, o SO retorna para a rotina Assembly que o chamou
5. Demais registradores e outras informações são carregados
6. Retorna para o espaço de usuário para continuar a execução, como se **nenhuma falta tivesse ocorrido**

Fixação de Páginas na Memória

- Memória virtual e E/S interagem ocasionalmente
- Processo 1 emite chamada ao sistema para ler do disco para o *buffer*
 - enquanto espera pela E/S, outro processo (2) inicia
 - ocorre uma falta de página para o processo 2
 - *buffer* do processo 1 pode ser escolhido para ser levado para disco – **problema!**
- Solução possível
 - Fixação de páginas envolvidas com E/S na memória

Memória Secundária



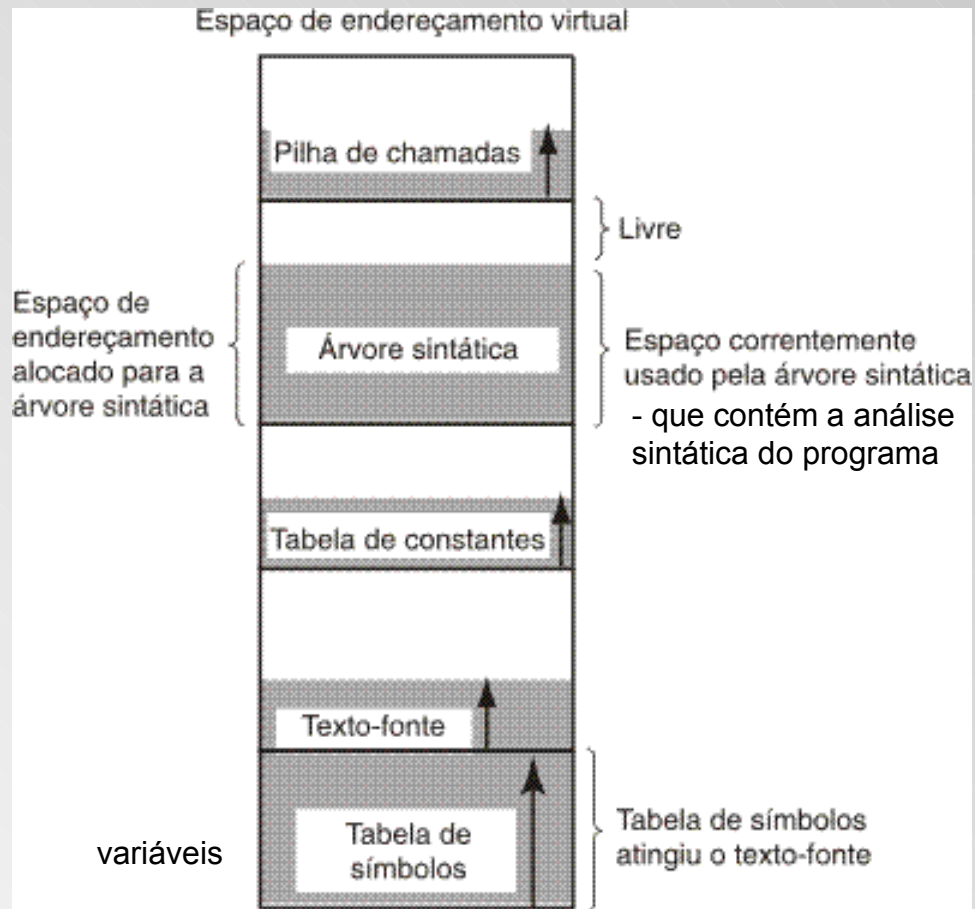
(a) Paginação para uma área de **troca estática**

(b) Páginas alocadas **dinamicamente** em disco

Tópicos

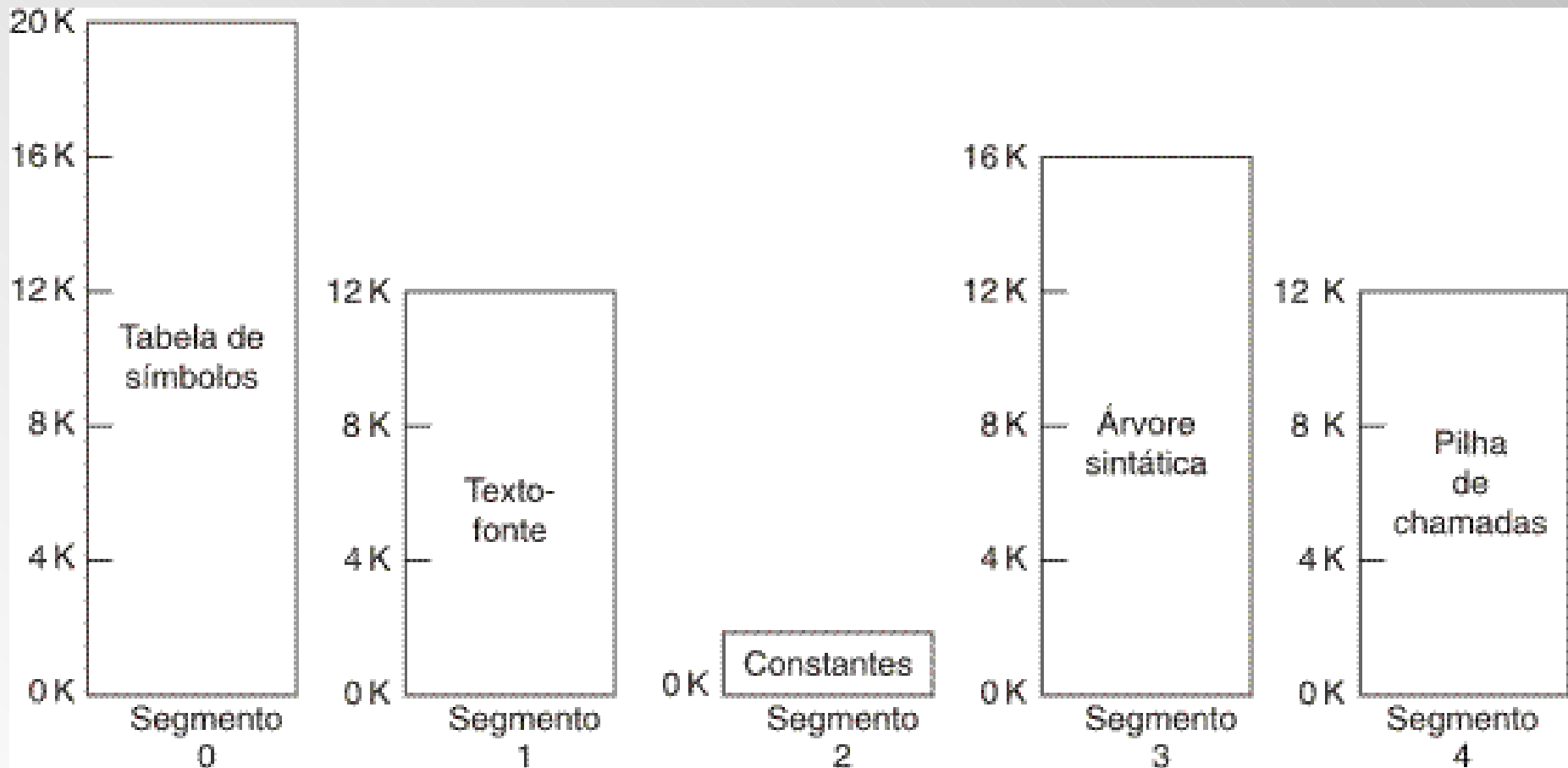
- ✓ Gerenciamento básico de memória
- ✓ Troca de processos
- ✓ Memória virtual
- ✓ Paginação
- ✓ Aceleração da paginação
- ✓ Substituição de páginas
- Segmentação

Segmentação (1)



- Espaço de endereçamento unidimensional com tabelas crescentes
- Uma tabela pode atingir outra...

Segmentação (2)



Permite que cada tabela cresça ou encolha, independentemente

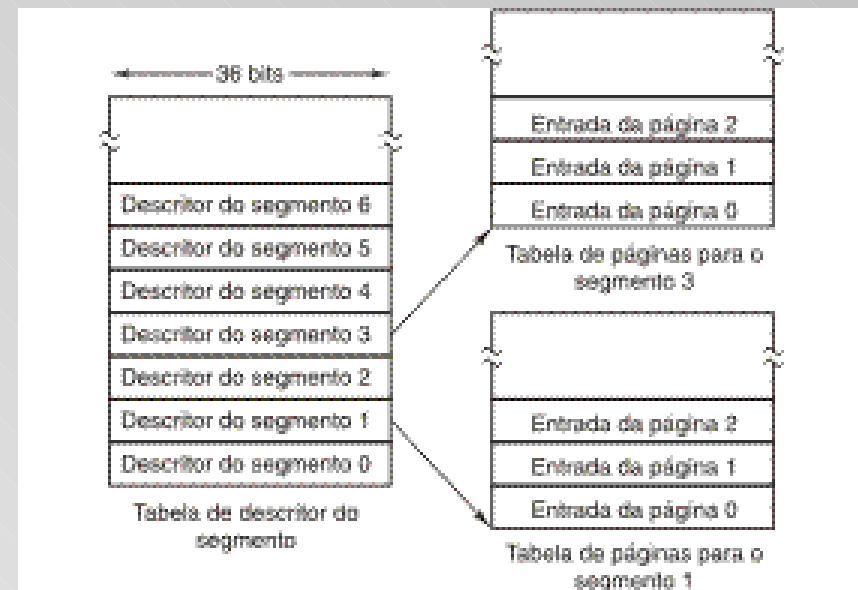
Comparação entre paginação e segmentação

Consideração	Paginação	Segmentação
O programador precisa estar ciente de que essa técnica está sendo usada?	Não	Sim
Quantos espaços de endereçamentos lineares existem?	Um	Muitos
O espaço de endereçamento total pode exceder o tamanho da memória física?	Sim	Sim
Os procedimentos e os dados podem ser diferenciados e protegidos separadamente?	Não	Sim
As tabelas com tamanhos variáveis podem ser acomodadas facilmente?	Não	Sim
O compartilhamento de procedimentos entre usuários é facilitado?	Não	Sim
Por que essa técnica foi inventada?	Para fornecer um grande espaço de endereçamento linear sem a necessidade de comprar mais memória física	Para permitir que programas e dados sejam quebrados em espaços de endereçamento logicamente independentes e para auxiliar o compartilhamento e a proteção

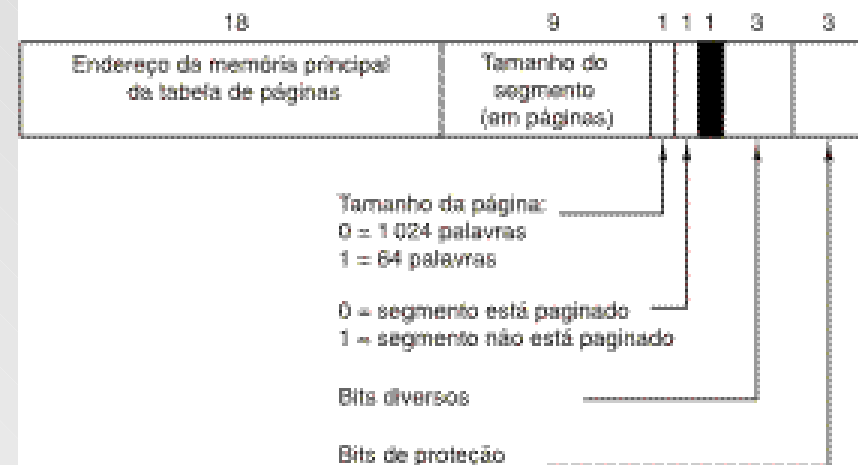
O compilador precisa!

Segmentação com Paginação: MULTICS (1)

- a) **Descritores de segmentos apontam para tabelas de páginas**
- b) **Descritor de segmento**

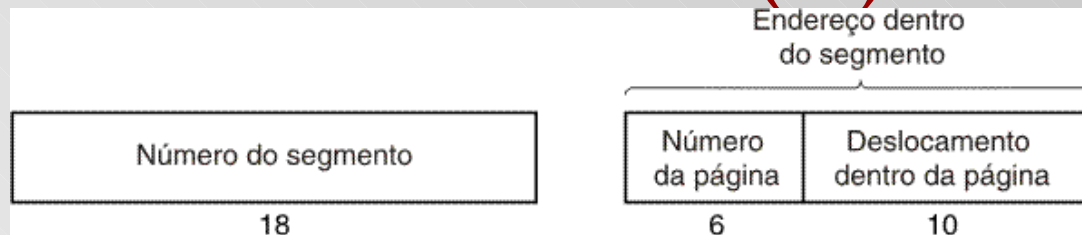


(a)

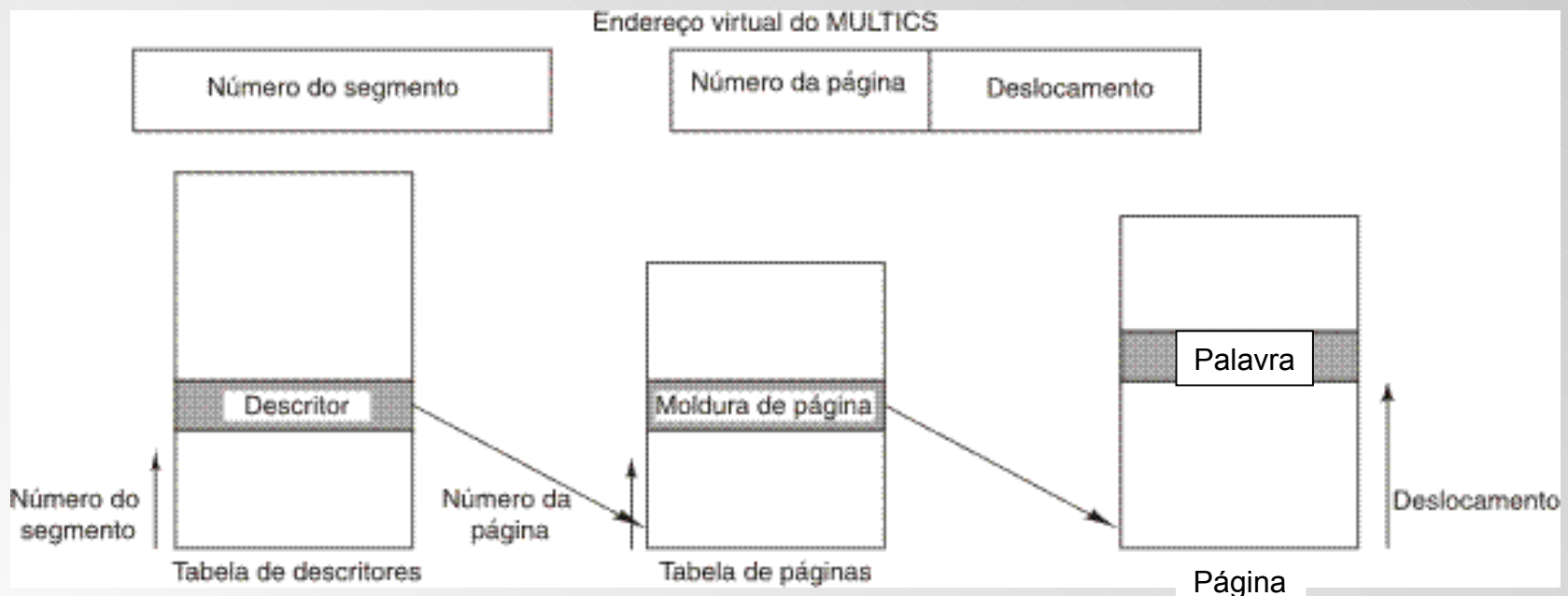


(b)

Segmentação com Paginação: MULTICS (2)



Um endereço virtual de 34 bits no MULTICS



Conversão de um endereço MULTICS de duas partes em um endereço da memória principal

Conclusões

- Na forma mais simples de memória virtual, cada espaço de endereçamento de um processo é dividido em páginas de tamanho uniforme
 - podem ser colocadas em qualquer moldura de página disponível na memória
- Dois dos melhores algoritmos de substituição de páginas são o Envelhecimento (aging) e o WSClock

Conclusões

- No projeto de sistemas de paginação, a escolha de um algoritmo não é suficiente

Outras considerações:

- Política de alocação
- Tamanho de página
- Segmentação ajuda a lidar com estruturas de dados que mudam de tamanho durante a execução e
 - simplifica o compartilhamento;
 - permite proteção diferente para segmentos diferentes
- Segmentação e paginação podem ser combinados para fornecer uma memória virtual de duas dimensões