



QXD0013 - Sistemas Operacionais

Gerenciamento de Memória III

Marcos Dantas Ortiz¹

¹Universidade Federal do Ceará, Brazil

26/09/2019

Tabelas de páginas para memórias grandes

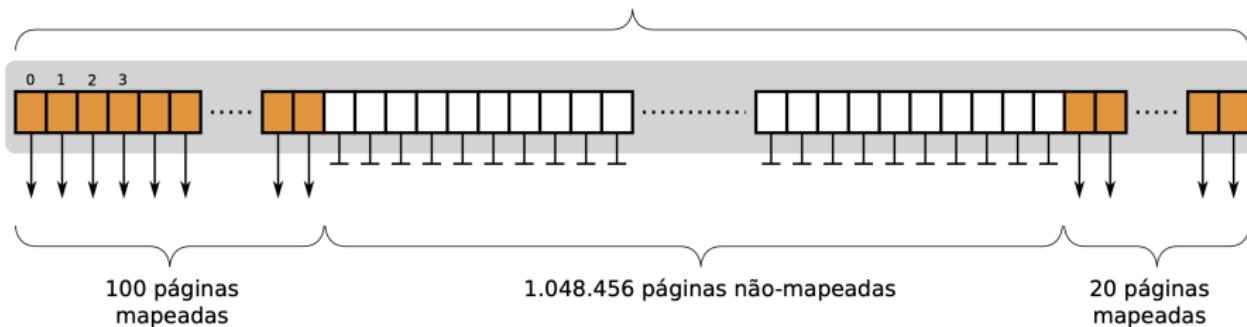
- As TLBs podem ser **usadas para acelerar a tradução de endereços virtuais para endereços físicos** em relação ao esquema de tabela de páginas na memória original.
- **Mas esse não é o único problema que precisamos combater.** Outro problema é como lidar com espaços de endereços virtuais muito grandes.

Tamanho da tabela de páginas

Em uma CPU de 32 bits com páginas de 4 Kbytes:

- Uma entrada na tabela: 32 bits (no do frame + flags)
 - Tabela com 2^{20} entradas: 4 Mbytes de RAM
 - Desperdício de RAM em processos pequenos

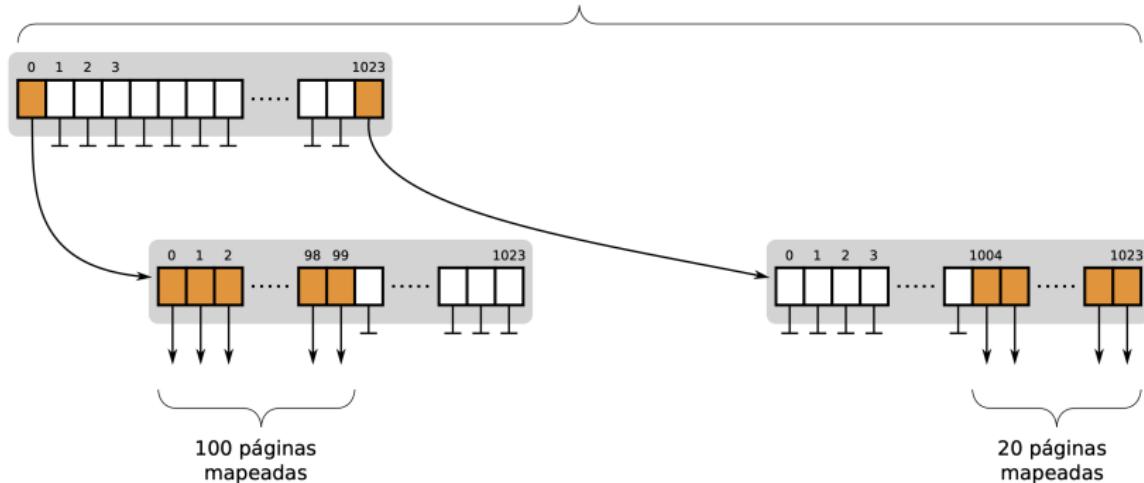
1.048.576 ponteiros de 32 bits = 4 MBytes



Fonte: Maziero, C. A. (2014). Sistemas operacionais: conceitos e mecanismos. Livro aberto.

Tabela de páginas com 2 níveis

3×1.024 ponteiros de 32 bits = 12 KBytes



Fonte: Maziero, C. A. (2014). Sistemas operacionais: conceitos e mecanismos. Livro aberto.

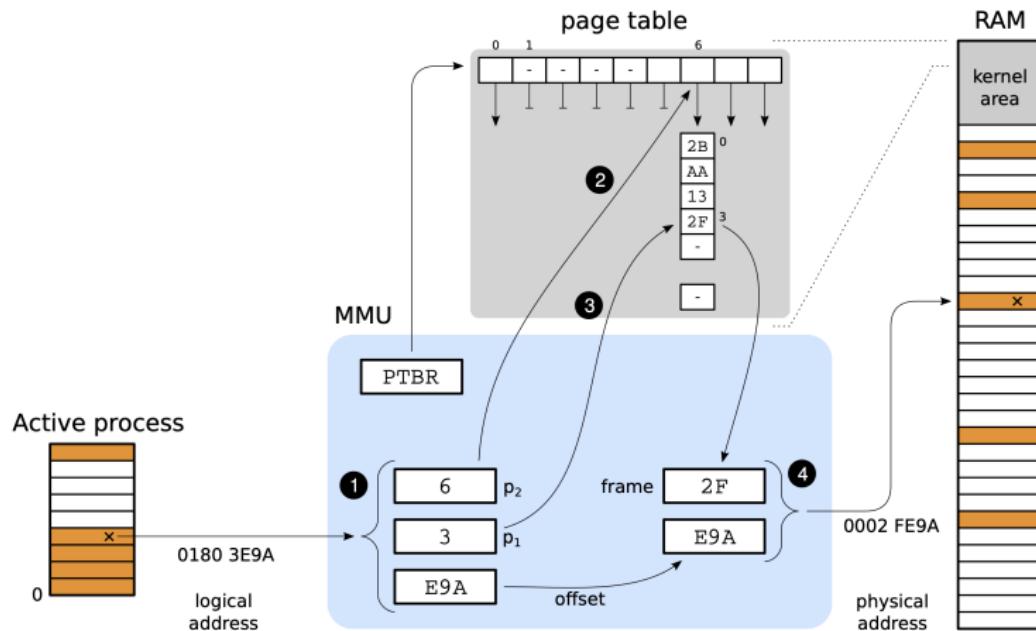
Tabelas de páginas multinível - Exemplo

- Endereçamento virtual de 32 bits (4GB)
 - campo PT1 (10 bits)
 - campo PT2 (10 bits)
 - campo Deslocamento (12 bits) → Página de 4KB
 - Dado que os deslocamentos são de 12 bits, as páginas são de 4KB e há um total de 2^{12} delas.

Bits	10	10	12
	PT1	PT2	Offset

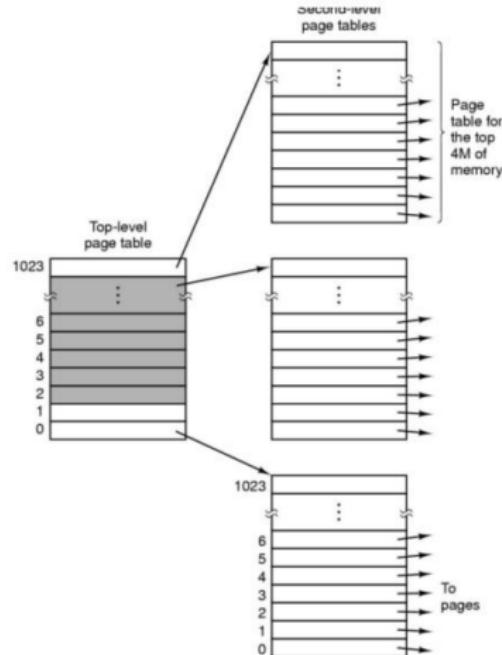
- O segredo para o uso do método da tabela de páginas multinível é evitar manter todas as **tabelas de páginas na memória o tempo inteiro**

Tradução - Tabela de páginas com 2 níveis



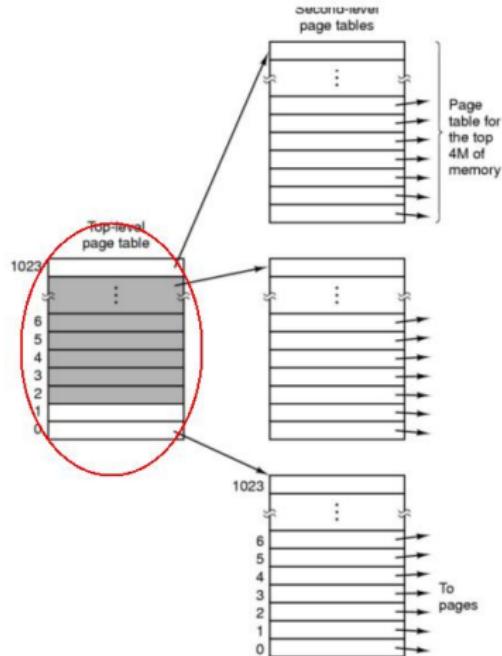
Fonte: Maziero, C. A. (2014). Sistemas operacionais: conceitos e mecanismos. Livro aberto.

Tabelas de páginas multinível - Exemplo



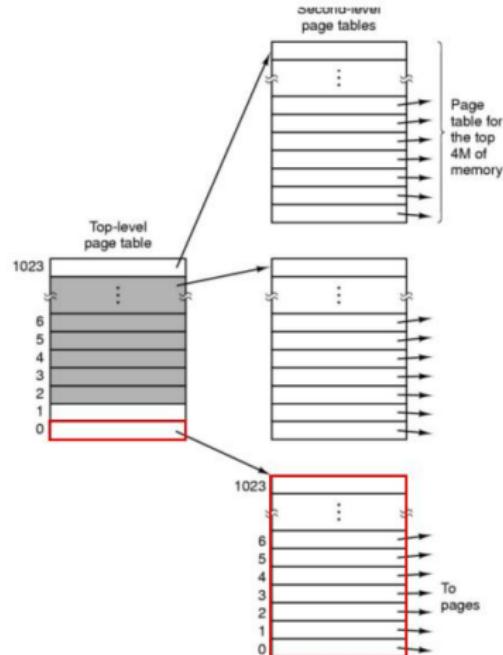
- Campo PT1 → Índice nível 1
 - Cada um representa 4MB
 - Entrada 0 → Código
 - Entrada 1 → Dados
 - Entrada 1023 → Pilha
 - Área não mapeada
- Campo PT2 → Índice da moldura
- Endereço 0x00403004
 - 0000 0000 0100 0000 0011
 - 0000 0000 0100
 - PT1 = 1
 - PT2 = 3
 - Deslocamento = 4

Tabelas de páginas multinível - Exemplo



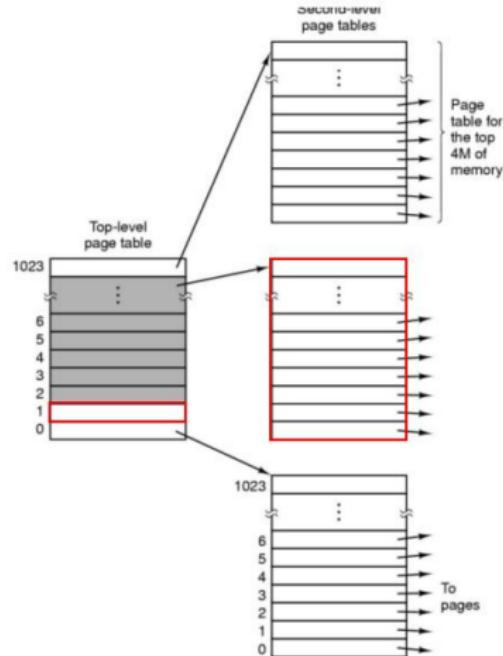
- Campo PT1 → Índice nível 1
 - Cada um representa 4MB
 - Entrada 0 → Código
 - Entrada 1 → Dados
 - Entrada 1023 → Pilha
 - Área não mapeada
- Campo PT2 → Índice da moldura
- Endereço 0x00403004
 - 0000 0000 0100 0000 0011
 - 0000 0000 0100
 - PT1 = 1
 - PT2 = 3
 - Deslocamento = 4

Tabelas de páginas multinível - Exemplo



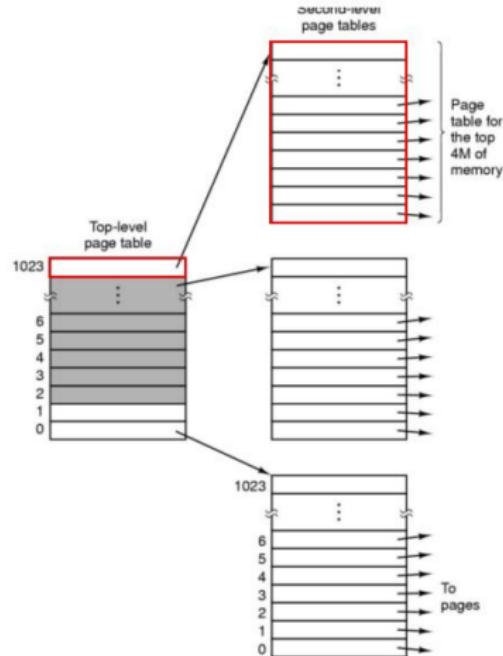
- Campo PT1 → Índice nível 1
 - Cada um representa 4MB
 - Entrada 0 → Código
 - Entrada 1 → Dados
 - Entrada 1023 → Pilha
 - Área não mapeada
- Campo PT2 → Índice da moldura
- Endereço 0x00403004
 - 0000 0000 0100 0000 0011
 - 0000 0000 0100
 - PT1 = 1
 - PT2 = 3
 - Deslocamento = 4

Tabelas de páginas multinível - Exemplo



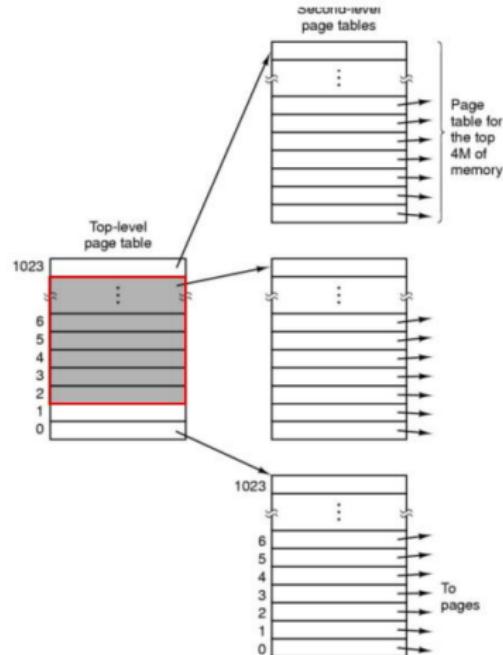
- Campo PT1 → Índice nível 1
 - Cada um representa 4MB
 - Entrada 0 → Código
 - Entrada 1 → Dados
 - Entrada 1023 → Pilha
 - Área não mapeada
- Campo PT2 → Índice da moldura
- Endereço 0x00403004
 - 0000 0000 0100 0000 0011
 - 0000 0000 0100
 - PT1 = 1
 - PT2 = 3
 - Deslocamento = 4

Tabelas de páginas multinível - Exemplo



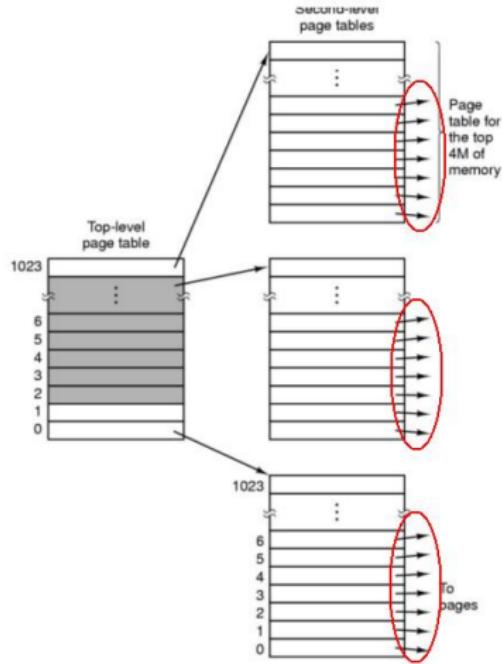
- Campo PT1 → Índice nível 1
 - Cada um representa 4MB
 - Entrada 0 → Código
 - Entrada 1 → Dados
 - Entrada 1023 → Pilha
 - Área não mapeada
- Campo PT2 → Índice da moldura
- Endereço 0x00403004
 - 0000 0000 0100 0000 0011
 - 0000 0000 0100
 - PT1 = 1
 - PT2 = 3
 - Deslocamento = 4

Tabelas de páginas multinível - Exemplo



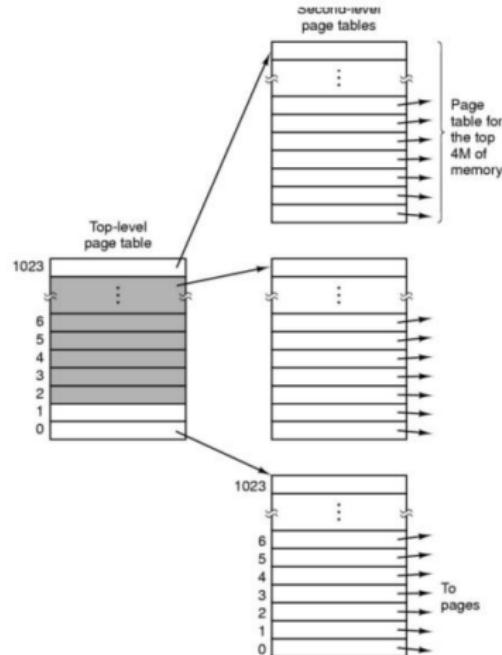
- Campo PT1 → Índice nível 1
 - Cada um representa 4MB
 - Entrada 0 → Código
 - Entrada 1 → Dados
 - Entrada 1023 → Pilha
 - Área não mapeada
- Campo PT2 → Índice da moldura
- Endereço 0x00403004
 - 0000 0000 0100 0000 0011
 - 0000 0000 0100
 - PT1 = 1
 - PT2 = 3
 - Deslocamento = 4

Tabelas de páginas multinível - Exemplo



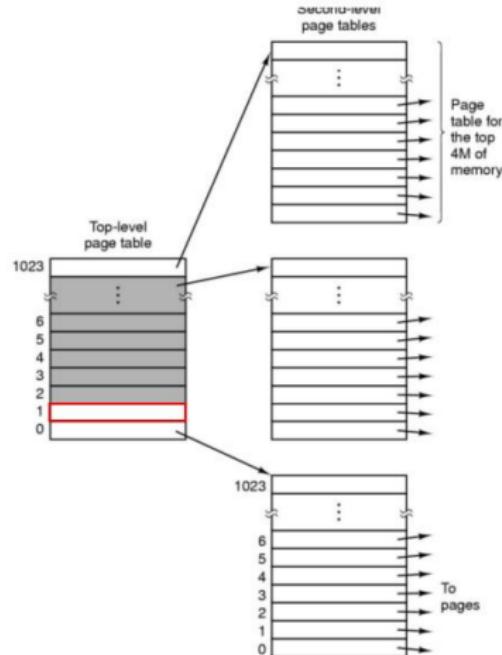
- Campo PT1 → Índice nível 1
 - Cada um representa 4MB
 - Entrada 0 → Código
 - Entrada 1 → Dados
 - Entrada 1023 → Pilha
 - Área não mapeada
- Campo PT2 → Índice da moldura
- Endereço 0x00403004
 - 0000 0000 0100 0000 0011
 - 0000 0000 0100
 - PT1 = 1
 - PT2 = 3
 - Deslocamento = 4

Tabelas de páginas multinível - Exemplo



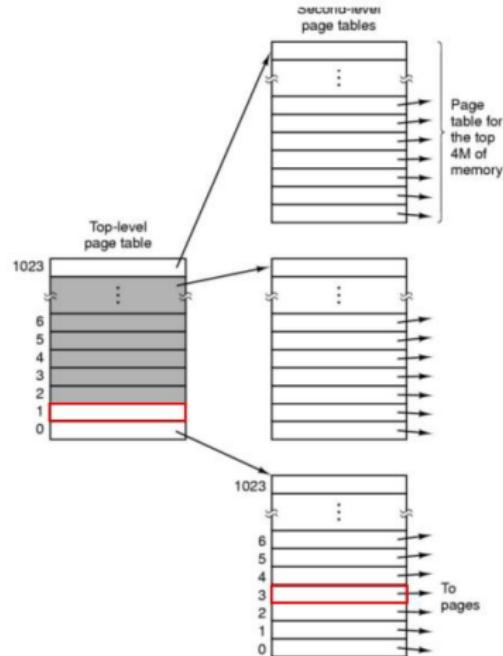
- Campo PT1 → Índice nível 1
 - Cada um representa 4MB
 - Entrada 0 → Código
 - Entrada 1 → Dados
 - Entrada 1023 → Pilha
 - Área não mapeada
- Campo PT2 → Índice da moldura
- Endereço 0x00403004
 - 0000 0000 0100 0000 0011
 - 0000 0000 0100
 - PT1 = 1
 - PT2 = 3
 - Deslocamento = 4

Tabelas de páginas multinível - Exemplo



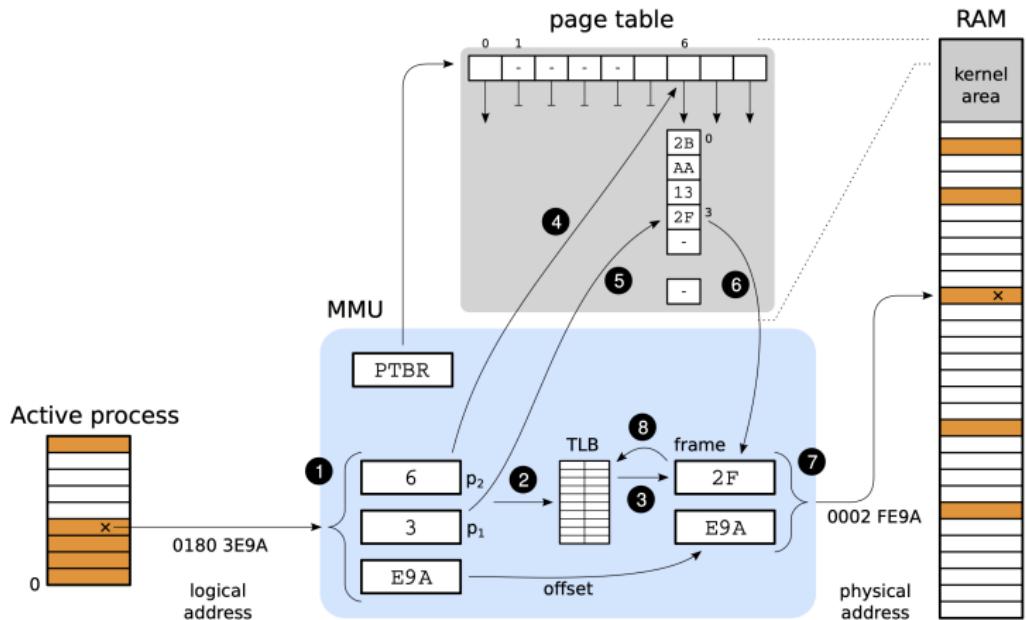
- Campo PT1 → Índice nível 1
 - Cada um representa 4MB
 - Entrada 0 → Código
 - Entrada 1 → Dados
 - Entrada 1023 → Pilha
 - Área não mapeada
- Campo PT2 → Índice da moldura
- Endereço 0x00403004
 - 0000 0000 0100 0000 0011
 - 0000 0000 0100
 - PT1 = 1
 - PT2 = 3
 - Deslocamento = 4

Tabelas de páginas multinível - Exemplo



- Campo PT1 → Índice nível 1
 - Cada um representa 4MB
 - Entrada 0 → Código
 - Entrada 1 → Dados
 - Entrada 1023 → Pilha
 - Área não mapeada
- Campo PT2 → Índice da moldura
- Endereço 0x00403004
 - 0000 0000 0100 0000 0011
 - 0000 0000 0100
 - PT1 = 1
 - PT2 = 3
 - Deslocamento = 4

Tradução - Tabela de páginas com 2 níveis com TLB



Fonte: Maziero, C. A. (2014). Sistemas operacionais: conceitos e mecanismos. Livro aberto.

Tabelas de páginas multinível - Exemplo

- Como exemplo, considere o endereço virtual de 32 bits 0x00403004 (4.206.596 em decimal), que corresponde a 12.292 bytes dentro do trecho dos dados.
- $4 * 1024 * 1024 = 4194304$
- $4206596 - 4194304 = 12.292$ bytes dentro do trecho dos dados

Tabela de Páginas Invertida

- Uma alternativa para os níveis cada vez maiores em uma hierarquia de paginação é conhecida como **tabela de páginas invertidas**.
- Elas foram usadas pela primeira vez por processadores como o **PowerPC, o UltraSPARC e o Itanium**.

Tabela de Páginas Invertida - Exemplo

- Exemplo:
 - Endereçamento de 64 bits
 - Tabela normal e página de 4KB → 2^{52} entradas
 - Tabela invertida e RAM de 1GB → $1\text{GB}/4\text{KB} = 262144$ entradas
 - Cada entrada → par (processo,página virtual)
- Embora tabelas de páginas invertidas poupem muito espaço, pelo menos quando o espaço de endereço virtual é muito maior do que a memória física, elas têm um sério problema: **a tradução virtual-física torna-se muito mais difícil.**
- **A saída desse dilema é fazer uso da TLB.** Se ela conseguir conter todas as páginas intensamente usadas, a tradução pode acontecer tão rápido quanto com as tabelas de páginas regulares.

Tabela de Páginas Invertida - Exemplo

Traditional page table with an entry for each of the 2^{52} pages

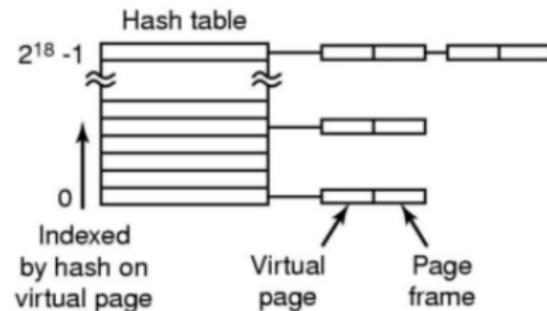
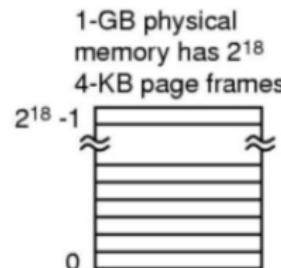


Tabela de Páginas Invertida - Exemplo

- As tabelas de páginas invertidas são comuns em máquinas de 64 bits porque mesmo com um tamanho de página muito grande, o número de entradas de tabela de páginas é gigantesco.
- Por exemplo, com páginas de 4 MB e endereços virtuais de 64 bits.
Quantas entradas de tabelas de páginas são necessárias?

Tabela de Páginas Invertida - Exemplo

- As tabelas de páginas invertidas são comuns em máquinas de 64 bits porque mesmo com um tamanho de página muito grande, o número de entradas de tabela de páginas é gigantesco.
- Por exemplo, com páginas de 4 MB e endereços virtuais de 64 bits.
Quantas entradas de tabelas de páginas são necessárias?
- → são necessárias 2^{42} entradas de tabelas de páginas.

Algoritmos de Substituição de Página

- Quando ocorre uma **falta de página**, o sistema operacional tem de escolher uma página para remover da memória a fim de abrir espaço para a que está chegando.
- Falta de página → Escolha de página a ser removida
- Caso modificada → Atualização do disco

Algoritmo Ótimo

- O algoritmo de substituição de página **melhor possível** é fácil de descrever
- Mas impossível de implementar de fato.
- **Ele funciona deste modo:** no momento em que ocorre uma falta de página, há um determinado conjunto de páginas na memória.
 - Uma dessas páginas será referenciada na próxima instrução.
 - Outras páginas talvez não sejam referenciadas até 10, 100 ou talvez 1.000 instruções mais tarde.
 - Cada página pode ser rotulada com o número de instruções que serão executadas antes de aquela página ser referenciada pela primeira vez.

Algoritmo Ótimo

- **Exemplo:** considere as referências de página 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, com quadro de 4 páginas. Encontre o número de falhas de página.

Algoritmo Ótimo

Page
reference

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

No. of Page frame - 4

7	0	1	2	0	3	0	4	2	3	0	3	2	3
	0	0	1	0	0	1	0	0	0	0	0	0	0
7	7	7	7	7	3	3	3	3	3	3	3	3	3
Miss	Miss	Miss	Miss	Miss	Hit	Miss	Hit	Miss	Hit	Hit	Hit	Hit	Hit

Total Page Fault = 6

Algoritmo Não Usado Recentemente (Not Recently Used (NRU))

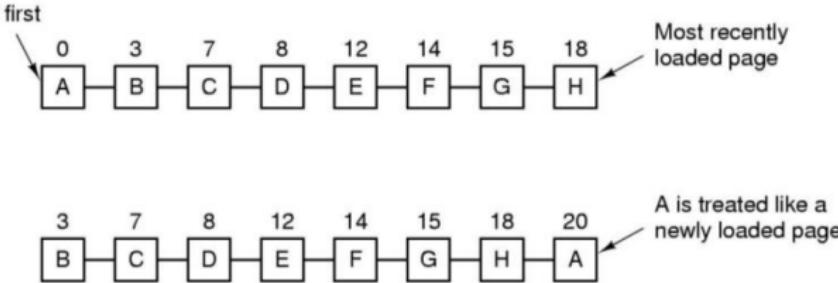
- Algoritmo será substituída a usada menos recentemente.
- 2 bits de status
 - Referenciado (R)
 - Modificado (M)
- Atualizados em todas referências
- Algoritmo de paginação pode usar R e M
 - Processo inicializado → R=0 e M=0
 - Periodicamente R é limpo
 - **Quando ocorre uma falta de página**, o sistema operacional inspeciona todas as páginas e as divide em quatro categorias baseadas nos valores atuais de seus bits R e M:
 - Classe 0: não referenciada , não modificada
 - Classe 1: não referenciada , modificada
 - Classe 2: referenciada , não modificada
 - Classe 3: referenciada , modificada
 - Algoritmo **remove aleatoriamente** página de classe mais baixa

Algoritmo First In, First Out (FIFO)

- Algoritmo de baixo custo
- SO mantém lista com páginas em ordem de inserção
- Falta de página → Mais antiga removida
- Não necessariamente menos acessada
- Raramente utilizado na forma pura

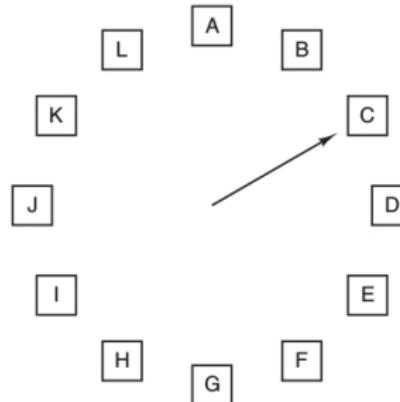
Algoritmo Second Chance

- Modificação do algoritmo FIFO
- Inspeção do bit R da página mais antiga
 - $R=0 \rightarrow$ Mais antiga e não utilizada \rightarrow Removida
 - $R=1$
 - R é limpo
 - Página colocada no final da fila
 - Continua pesquisa
- Se todas páginas tem $R=1 \rightarrow$ FIFO puro



Algoritmo do Relógio

- Algoritmo segunda chance é ineficaz
 - Reinsere constantemente páginas na lista
- Maneira mais eficiente seria utilizar uma fila circular (relógio)



Algoritmo Usado Menos Recentemente (Least Recently Used (LRU))

- Aproximação do algoritmo ótimo
- **páginas que foram usadas intensamente** nas últimas instruções provavelmente o serão em seguida de novo
- **páginas que não foram usadas há eras** provavelmente seguirão sem ser utilizadas por um longo tempo
- Falta de página → Remover não utilizada por mais tempo

Algoritmo Least Recently Used (LRU)

- Outra Implementação:
 - n molduras →
 - Hardware auxiliar com matriz de $n \times n$ bits
 - Matriz inicializada com todos os bits 0
 - Moldura k referenciada
 - Bits da linha $k \rightarrow 1$
 - Bits da coluna $k \rightarrow 0$
 - Página LRU → Menor valor binário

Algoritmo Least Recently Used (LRU)

- Outra Implementação (Exemplo):

- $n = 3$
- Sequência de referenciamento: 0-1-2-3-2-1-0-3-2-3

	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

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

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

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

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

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

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

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

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

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

Algoritmo Least Recently Used (LRU)

- Outra Implementação (Exemplo):

- $n = 3$
- Sequência de referenciamento: 0-1-2-3-2-1-0-3-2-3

	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

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

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

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

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

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

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

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

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

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

Algoritmo Least Recently Used (LRU)

- Outra Implementação (Exemplo):
 - $n = 3$
 - Sequência de referenciamento: 0-1-2-3-2-1-0-3-2-3

	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

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

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

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

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

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

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

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

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

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

Algoritmo Least Recently Used (LRU)

- Outra Implementação (Exemplo):

- $n = 3$
- Sequência de referenciamento: 0-1-2-3-2-1-0-3-2-3

	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

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

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

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

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

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

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

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

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

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

Algoritmo Least Recently Used (LRU)

- Outra Implementação (Exemplo):

- $n = 3$

- Sequência de referenciamento: 0-1-2-3-2-1-0-3-2-3

	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

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

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

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

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

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

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

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

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

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

Algoritmo Least Recently Used (LRU)

- Outra Implementação (Exemplo):
 - $n = 3$
 - Sequência de referenciamento: 0-1-2-3-2-1-0-3-2-3

	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

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

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

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

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

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

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

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

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

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

Algoritmo Least Recently Used (LRU)

- Outra Implementação (Exemplo):
 - $n = 3$
 - Sequência de referenciamento: 0-1-2-3-2-1-0-3-2-3

	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

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

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

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

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

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

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

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

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

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

Algoritmo Least Recently Used (LRU)

- Outra Implementação (Exemplo):

- $n = 3$
- Sequência de referenciamento: 0-1-2-3-2-1-0-3-2-3

	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

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

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

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

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

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

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

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

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

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

Algoritmo Least Recently Used (LRU)

- Outra Implementação (Exemplo):

- $n = 3$
- Sequência de referenciamento: 0-1-2-3-2-1-0-3-2-3

	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

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

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

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

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

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

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

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

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

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

Algoritmo Least Recently Used (LRU)

- Outra Implementação (Exemplo):
 - $n = 3$
 - Sequência de referenciamento: 0-1-2-3-2-1-0-3-2-3

	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

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

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

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

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

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

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

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

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

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

Algoritmo Least Recently Used (LRU)

- Outra Implementação (Exemplo):
 - $n = 3$
 - Sequência de referenciamento: 0-1-2-3-2-1-0-3-2-3

	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

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

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

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

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

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

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

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

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

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

Not Frequently Used (NFU)

- Embora o algoritmo de LRU anterior seja (em princípio) realizável, poucas máquinas, se é que existe alguma, têm o hardware necessário.
- Em vez disso, é necessária uma solução que possa ser implementada em software.
- Uma possibilidade é o algoritmo de substituição de páginas **não usadas frequentemente (NFU — Not Frequently Used)**.

Simulação do LRU em software

- Modificação do NFU → Aproximação do LRU em software
- Dois passos:
 - Forma binária do contador tem os bits deslocados para direita
 - Bit mais a esquerda do contador é substituído por R
- Também chamado de Algoritmo de Envelhecimento

Simulação do LRU em software

R bits for pages 0-5, clock tick 0	R bits for pages 0-5, clock tick 1	R bits for pages 0-5, clock tick 2	R bits for pages 0-5, clock tick 3	R bits for pages 0-5, clock tick 4	
1 0 1 0 1 1	1 1 0 0 1 0	1 1 0 1 0 1	1 0 0 0 1 0	0 1 1 0 0 0	
Page	0	1	2	3	
0	10000000	11000000	11100000	11110000	01111000
1	00000000	10000000	11000000	01100000	10110000
2	10000000	01000000	00100000	00100000	10010000
3	00000000	00000000	10000000	01000000	00100000
4	10000000	11000000	01100000	10110000	01011000
5	10000000	01000000	10100000	01010000	00101000

Simulação do LRU em software

R bits for pages 0-5, clock tick 0	R bits for pages 0-5, clock tick 1	R bits for pages 0-5, clock tick 2	R bits for pages 0-5, clock tick 3	R bits for pages 0-5, clock tick 4	
1 0 1 0 1 1	1 1 0 0 1 0	1 1 0 1 0 1	1 0 0 0 1 0	0 1 1 0 0 0	
Page	0	1	2	3	
0	10000000	11000000	11100000	11110000	01111000
1	00000000	10000000	11000000	01100000	10110000
2	10000000	01000000	00100000	00100000	10010000
3	00000000	00000000	10000000	01000000	00100000
4	10000000	11000000	01100000	10110000	01011000
5	10000000	01000000	10100000	01010000	00101000

Simulação do LRU em software

R bits for pages 0-5, clock tick 0	R bits for pages 0-5, clock tick 1	R bits for pages 0-5, clock tick 2	R bits for pages 0-5, clock tick 3	R bits for pages 0-5, clock tick 4	
1 0 1 0 1 1	1 1 0 0 1 0	1 1 0 1 0 1	1 0 0 0 1 0	0 1 1 0 0 0	
Page	0	1	2	3	
0	10000000	11000000	11100000	11110000	01111000
1	00000000	10000000	11000000	01100000	10110000
2	10000000	01000000	00100000	00100000	10010000
3	00000000	00000000	10000000	01000000	00100000
4	10000000	11000000	01100000	10110000	01011000
5	10000000	01000000	10100000	01010000	00101000

Simulação do LRU em software

R bits for pages 0-5, clock tick 0	R bits for pages 0-5, clock tick 1	R bits for pages 0-5, clock tick 2	R bits for pages 0-5, clock tick 3	R bits for pages 0-5, clock tick 4	
1 0 1 0 1 1	1 1 0 0 1 0	1 1 0 1 0 1	1 0 0 0 1 0	0 1 1 0 0 0	
Page	0	1	2	3	
0	10000000	11000000	11100000	11110000	01111000
1	00000000	10000000	11000000	01100000	10110000
2	10000000	01000000	00100000	00100000	10010000
3	00000000	00000000	10000000	01000000	00100000
4	10000000	11000000	01100000	10110000	01011000
5	10000000	01000000	10100000	01010000	00101000

Simulação do LRU em software

R bits for pages 0-5, clock tick 0	R bits for pages 0-5, clock tick 1	R bits for pages 0-5, clock tick 2	R bits for pages 0-5, clock tick 3	R bits for pages 0-5, clock tick 4
1 0 1 0 1 1	1 1 0 0 1 0	1 1 0 1 0 1	1 0 0 0 1 0	0 1 1 0 0 0
Page	0	1	2	3
0 1000000	11000000	11100000	11110000	01111000
1 00000000	10000000	11000000	01100000	10110000
2 10000000	01000000	00100000	00100000	10010000
3 00000000	00000000	10000000	01000000	00100000
4 10000000	11000000	01100000	10110000	01011000
5 10000000	01000000	10100000	01010000	00101000

Simulação do LRU em software

R bits for pages 0-5, clock tick 0	R bits for pages 0-5, clock tick 1	R bits for pages 0-5, clock tick 2	R bits for pages 0-5, clock tick 3	R bits for pages 0-5, clock tick 4	
1 0 1 0 1 1	1 1 0 0 1 0	1 1 0 1 0 1	1 0 0 0 1 0	0 1 1 0 0 0	
Page	0	1	2	3	
0	10000000	11000000	11100000	11110000	01111000
1	00000000	10000000	11000000	01100000	10110000
2	10000000	01000000	00100000	00100000	10010000
3	00000000	00000000	10000000	01000000	00100000
4	10000000	11000000	01100000	10110000	01011000
5	10000000	01000000	10100000	01010000	00101000

Simulação do LRU em software

R bits for pages 0-5, clock tick 0	R bits for pages 0-5, clock tick 1	R bits for pages 0-5, clock tick 2	R bits for pages 0-5, clock tick 3	R bits for pages 0-5, clock tick 4
1 0 1 0 1 1	1 1 0 0 1 0	1 1 0 1 0 1	1 0 0 0 1 0	0 1 1 0 0 0
Page				
0	10000000	11000000	11100000	11110000
1	00000000	10000000	11000000	01100000
2	10000000	01000000	00100000	00100000
3	00000000	00000000	10000000	01000000
4	10000000	11000000	01100000	10110000
5	10000000	01000000	10100000	01010000

Simulação do LRU em software

R bits for pages 0-5, clock tick 0	R bits for pages 0-5, clock tick 1	R bits for pages 0-5, clock tick 2	R bits for pages 0-5, clock tick 3	R bits for pages 0-5, clock tick 4
1 0 1 0 1 1	1 1 0 0 1 0	1 1 0 1 0 1	1 0 0 0 1 0	0 1 1 0 0 0
Page				
0	10000000	11000000	11100000	11110000
1	00000000	10000000	11000000	01100000
2	10000000	01000000	00100000	00100000
3	00000000	00000000	10000000	01000000
4	10000000	11000000	01100000	10110000
5	10000000	01000000	10100000	01010000

Simulação do LRU em software

R bits for pages 0-5, clock tick 0	R bits for pages 0-5, clock tick 1	R bits for pages 0-5, clock tick 2	R bits for pages 0-5, clock tick 3	R bits for pages 0-5, clock tick 4
1 0 1 0 1 1	1 1 0 0 1 0	1 1 0 1 0 1	1 0 0 0 1 0	0 1 1 0 0 0
Page				
0	10000000	11000000	11100000	11110000
1	00000000	10000000	11000000	01100000
2	10000000	01000000	00100000	00100000
3	00000000	00000000	10000000	01000000
4	10000000	11000000	01100000	10110000
5	10000000	01000000	10100000	01010000

Simulação do LRU em software

R bits for pages 0-5, clock tick 0	R bits for pages 0-5, clock tick 1	R bits for pages 0-5, clock tick 2	R bits for pages 0-5, clock tick 3	R bits for pages 0-5, clock tick 4
1 0 1 0 1 1	1 1 0 0 1 0	1 1 0 1 0 1	1 0 0 0 1 0	0 1 1 0 0 0
Page				
0	10000000	11000000	11100000	11110000
1	00000000	10000000	11000000	01100000
2	10000000	01000000	00100000	00100000
3	00000000	00000000	10000000	01000000
4	10000000	11000000	01100000	10110000
5	10000000	01000000	10100000	01010000

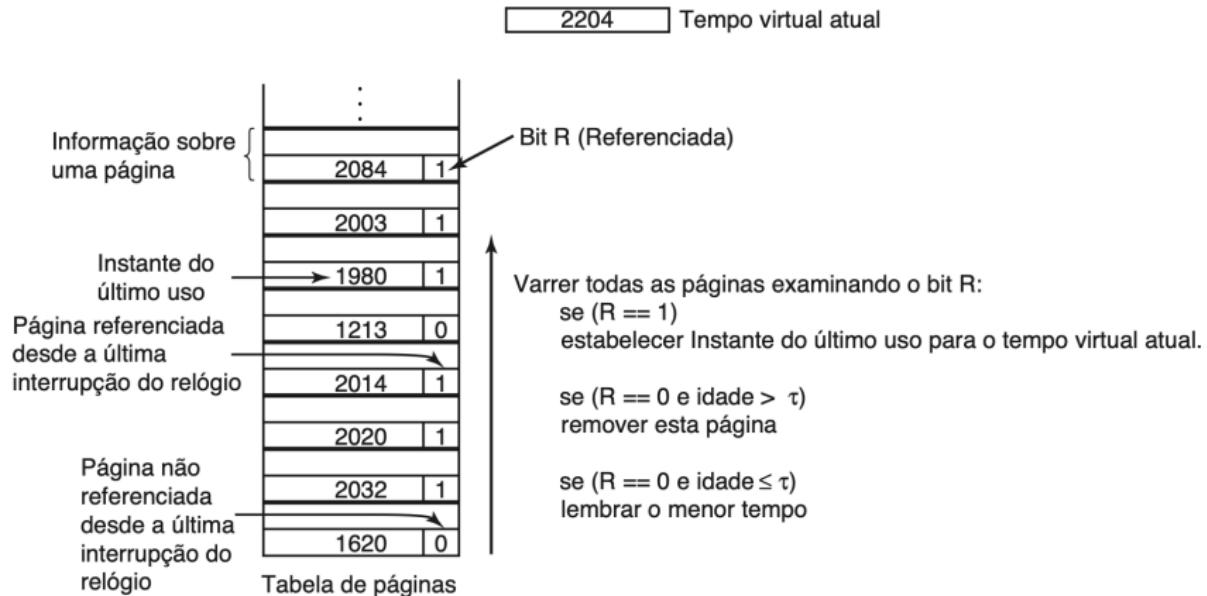
Simulação do LRU em software

R bits for pages 0-5, clock tick 0	R bits for pages 0-5, clock tick 1	R bits for pages 0-5, clock tick 2	R bits for pages 0-5, clock tick 3	R bits for pages 0-5, clock tick 4	
1 0 1 0 1 1	1 1 0 0 1 0	1 1 0 1 0 1	1 0 0 0 1 0	0 1 1 0 0 0	
Page	0	1	2	3	
0	10000000	11000000	11100000	11110000	01111000
1	00000000	10000000	11000000	01100000	10110000
2	10000000	01000000	00100000	00100000	10010000
3	00000000	00000000	10000000	01000000	00100000
4	10000000	11000000	01100000	10110000	01011000
5	10000000	01000000	10100000	01010000	00101000

O algoritmo de substituição de páginas do conjunto de trabalho

- O conjunto de páginas que um processo está atualmente usando é o seu **conjunto de trabalho**.
- Se todo o conjunto de trabalho está na memória, o processo será executado sem causar muitas faltas até passar para outra fase de execução
- Após um tempo, o processo tem a maior parte das páginas que ele precisa para ser executado com relativamente poucas faltas de páginas.

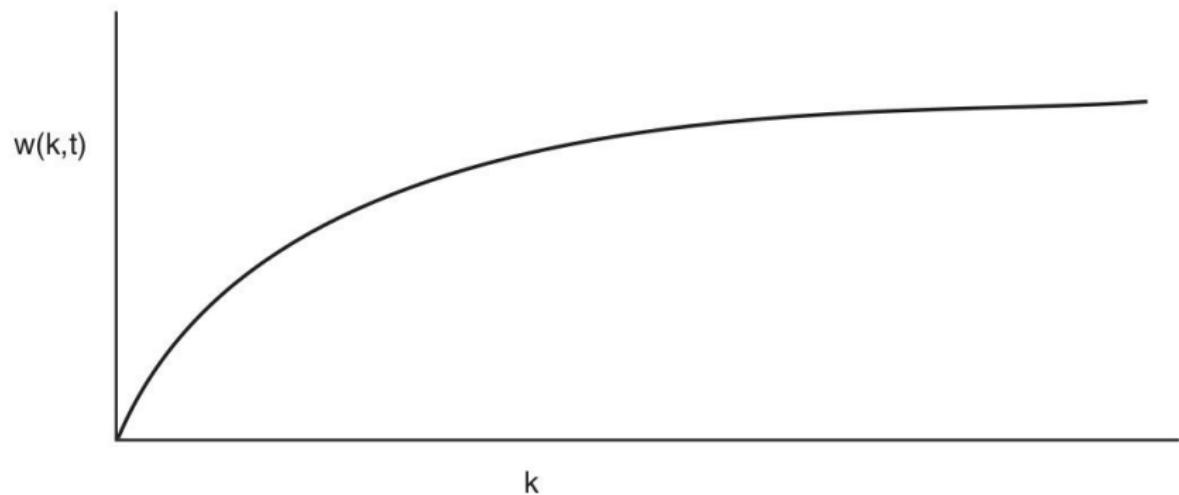
O algoritmo do conjunto de trabalho



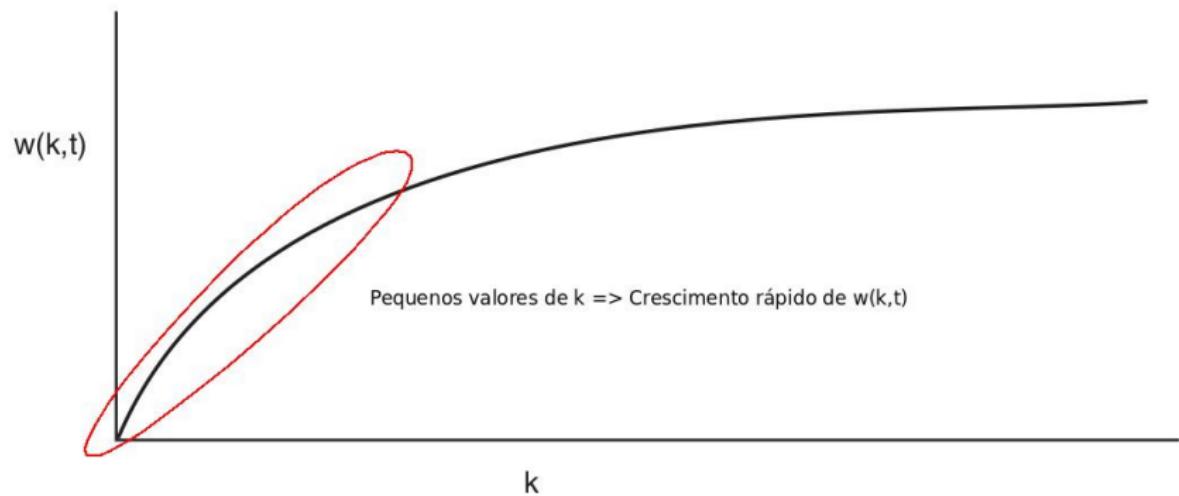
Modelo do Conjunto de Trabalho

- Portanto, muitos sistemas de paginação tentam controlar o conjunto de trabalho de cada processo e certificar-se de que ele está na memória antes de deixar o processo ser executado.
- Essa abordagem é chamada de **modelo do conjunto de trabalho**
- Ele foi projetado para **reduzir substancialmente o índice de faltas de páginas**.
- Carregar as páginas antes de deixar um processo ser executado também é chamado de **pré-paginação**.

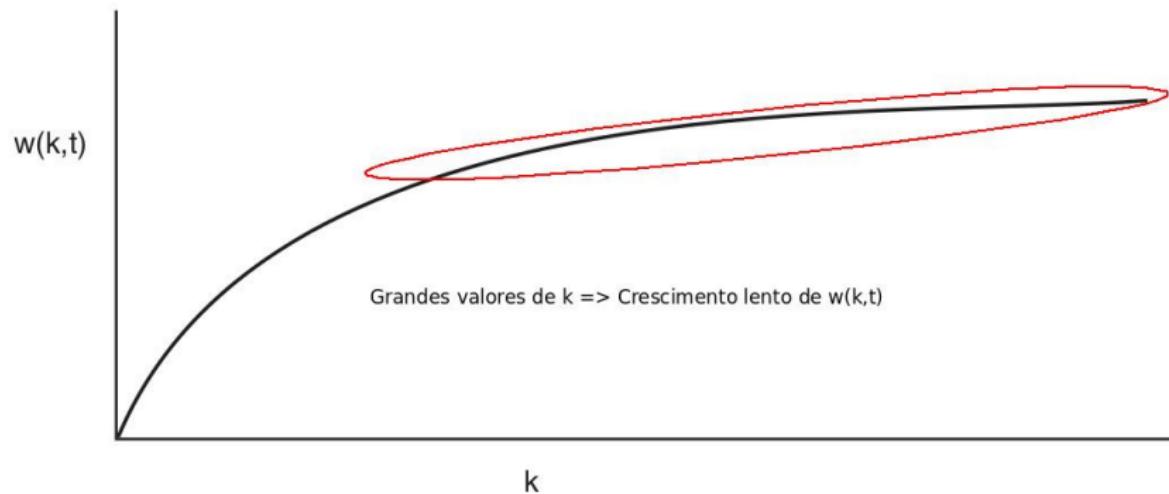
Modelo do Conjunto de Trabalho



Modelo do Conjunto de Trabalho



Modelo do Conjunto de Trabalho

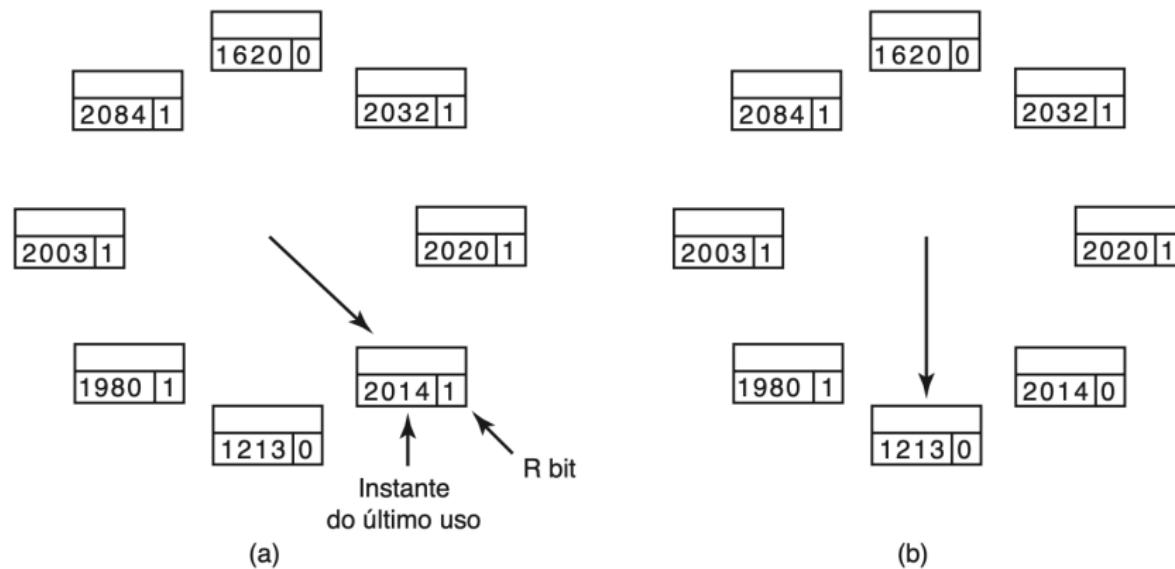


Algoritmo WSClock

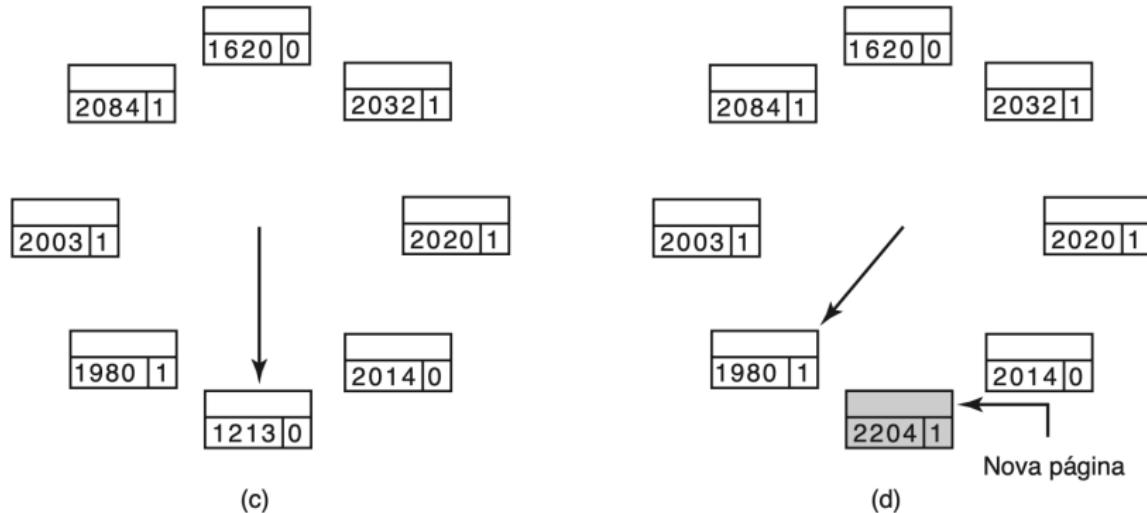
- O algoritmo básico do conjunto de trabalho é enfadonho, já que a tabela de páginas inteira precisa ser varrida a cada falta de página até que uma candidata adequada seja localizada.
- Um algoritmo melhorado, que é baseado no algoritmo de relógio mas também usa a informação do conjunto de trabalho, é chamado de **WSClock**
- Por sua simplicidade de implementação e bom desempenho, ele é amplamente usado na prática.

Algoritmo WSClock

2204 Tempo virtual atual



Algoritmo WSClock



Resumo - Algoritmos de Substituição

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

Exercício

- Um sistema de memória possui três frames e oito páginas virutais. Considerar a sequência de referência **01232304523143263212**.
 - Quantas falhas de página ocorrerão quando FIFO, LRU e Ótimo algoritmos são usados respectivamente?