INE5412 Sistemas Operacionais I

L. F. Friedrich

Capítulo 3

Memoria Virtual – Algoritmos de troca

Tratamento da falta de página

- O hardware cria uma cilada para o núcleo, salvando o contador do programa na pilha.
- Uma rotina em código é iniciada para salvar o conteúdo dos registradores de uso geral e outras informações voláteis.
- O sistema operacional descobre a ocorrência de uma falta de página e tenta descobrir qual página virtual é necessária.
- Uma vez conhecido o endereço virtual que causou a falta da página, o sistema verifica se esse endereço é válido e se a proteção é consistente com o acesso.

- Se existe alguma moldura livre aloca. Senão seleciona uma moldura para realizar a **troca de página**.
- Se a moldura da página selecionada estiver suja, a página é escalonada para ser transferida para o disco e será realizado um chaveamento de contexto.
- Quando a moldura da página estiver limpa, o sistema operacional buscará o endereço em disco onde está a página virtual solicitada e escalonará uma operação para trazê-la.
- Quando a interrupção de disco indicar que a página chegou na memória, as tabelas de páginas serão atualizadas para refletir sua posição, e será indicado que a moldura de página está normal.

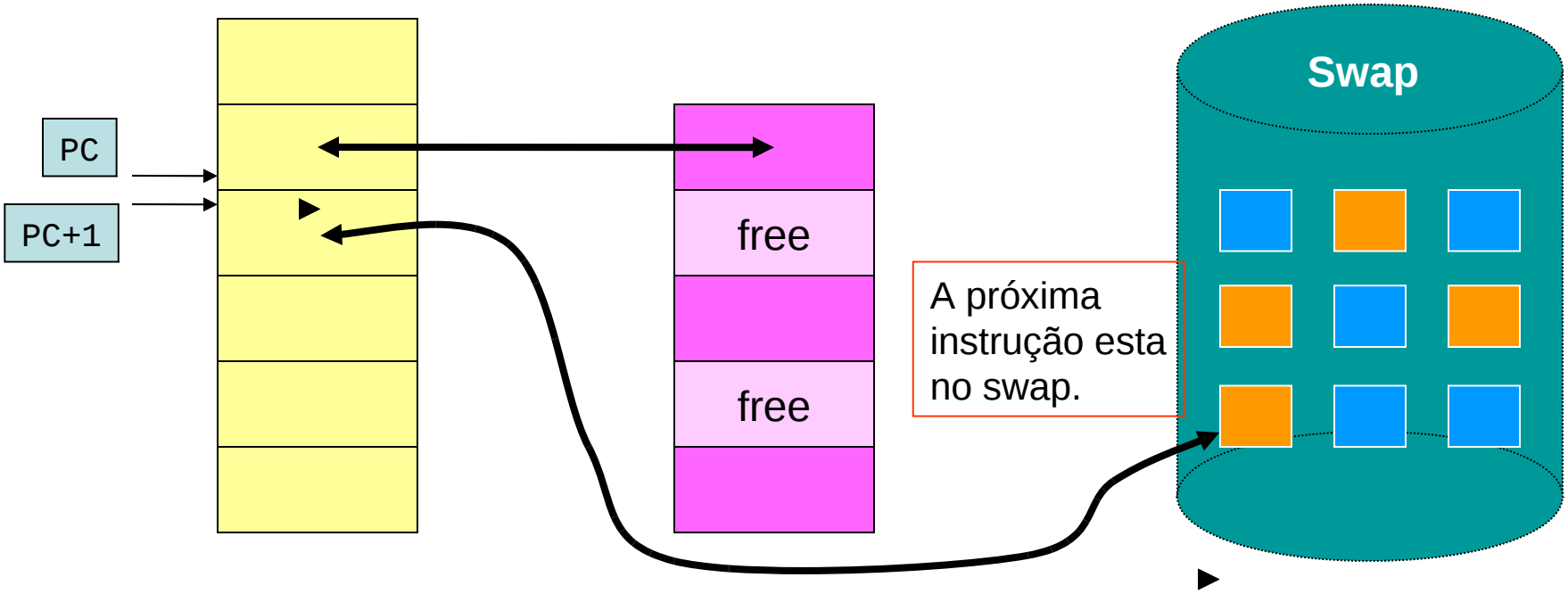
- A instrução que estava faltando é recuperada para o estado em que se encontrava quando começou, e o contador de programa é reiniciado a fim de apontar para aquela instrução.
- O processo em falta é escalonado, o sistema operacional retorna para a rotina, em linguagem de máquina, que o chamou.
- Esta rotina recarrega os registradores e outras informações de estado e retorna ao espaço de usuário para continuar a execução como se nada tivesse ocorrido.

Troca de páginas

- Troca de páginas acontece quando as duas condições a seguir acontecem.
  - O processo solicita uma página que não existe na memória física, ocorrência de uma falta de página.
  - Não existe frames livres disponíveis na memória física, o SO precisa escolher uma página a ser removida para liberar espaço.

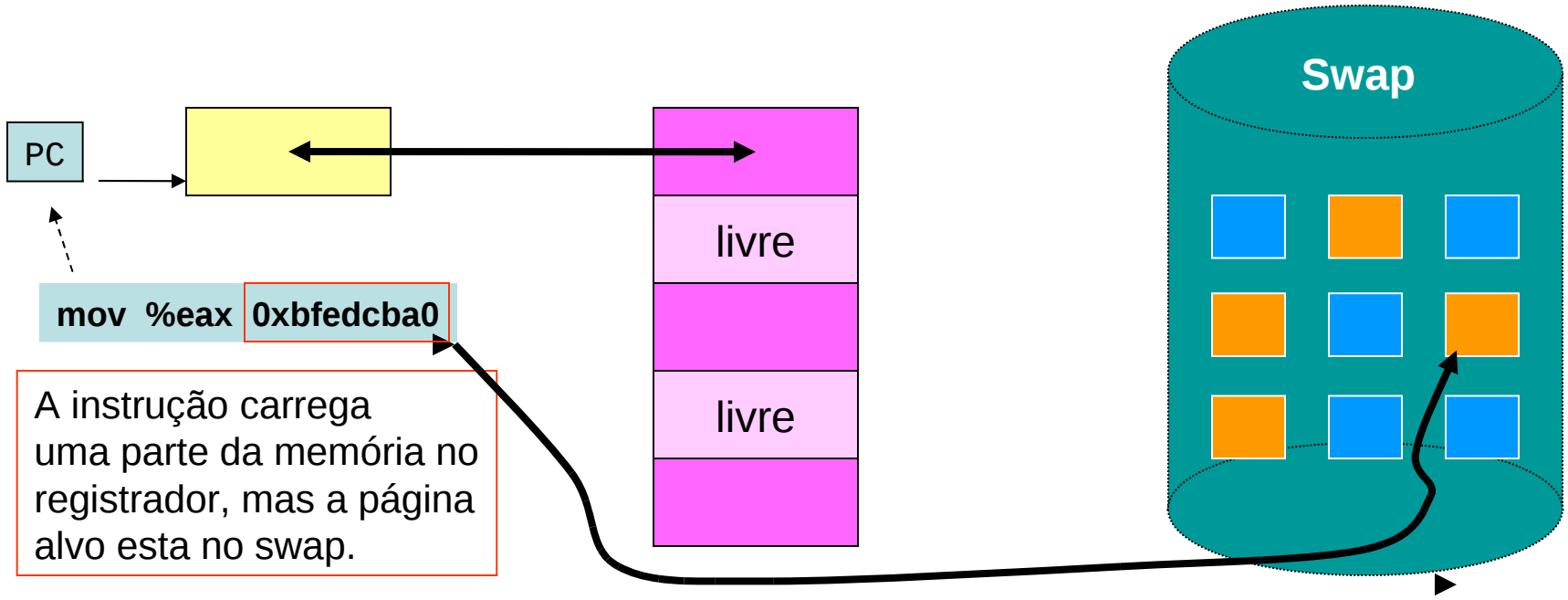
Condição (1)

- Uma nova página pode ser necessária quando:
  (a) buscando uma nova instrução.



Condição (1)

- Uma nova página pode ser necessária quando:
  (b) Uma instrução acessa uma parte da memória.



Condição (2)

- Os frames podem ser cheios facilmente.
  - Ativando mais processos.
  - Alocando mais memória.

Condição (2)

- OOM gerador versão 2!

```
#define ONE_MEG  1024 * 1024

int main(void) {
    void *ptr;
    int counter = 0;

    while(1) {
        ptr = malloc(ONE_MEG);
        if(!ptr)
            break;
        memset(ptr, 0, ONE_MEG);
        counter++;
        printf("Allocated %d MB\n", counter);
    }

    return 0;
}
```

Muita atividade de swapping acontece na execução desta função!

Condição (2)

$$
\begin{aligned}
1MB &= 2^{20} \\
&= 2^{8} \times 2^{12} \\
&= 256 \times 4KB
\end{aligned}
$$

Uma chamada malloc() em OOM-v2 gera aproximadamente 256 novas páginas!

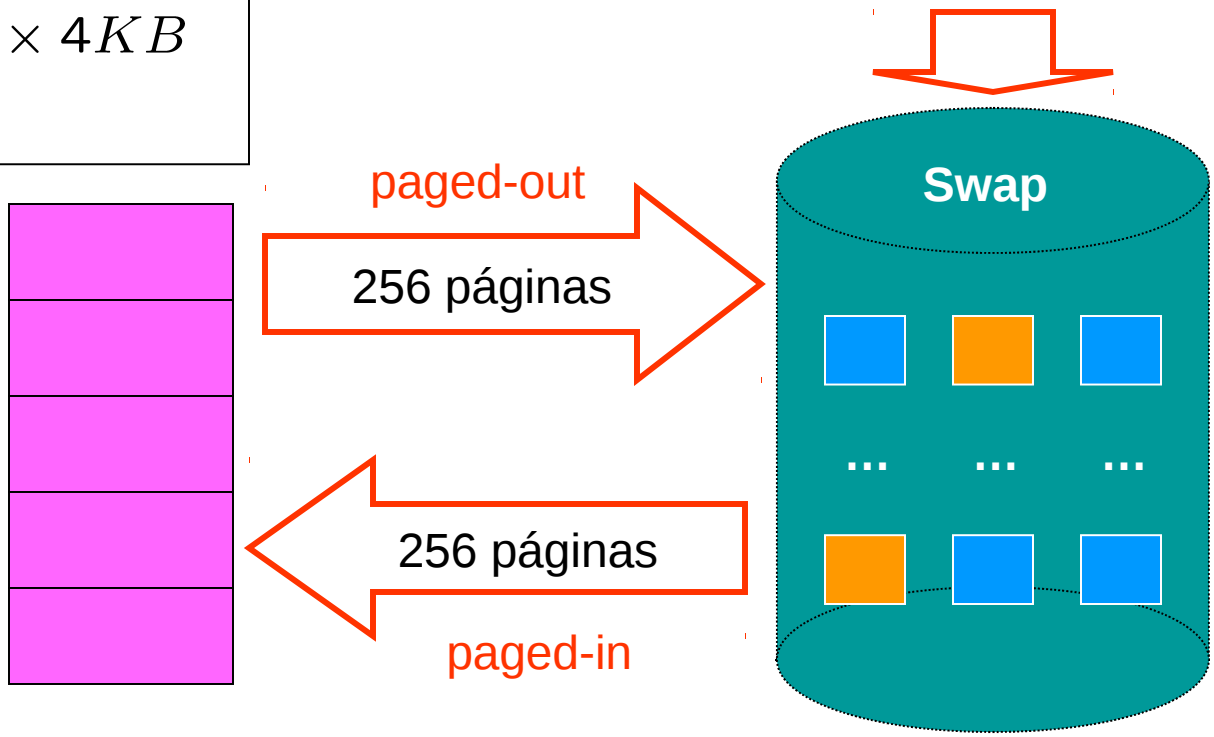
Uma chamada memset() em OOM-v2 precisa 256 páginas para ser paged in p/ memória física!

Condição (2)

$$
\begin{aligned}
1MB &= 2^{20} \\
&= 2^{8} \times 2^{12} \\
&= 256 \times 4KB
\end{aligned}
$$

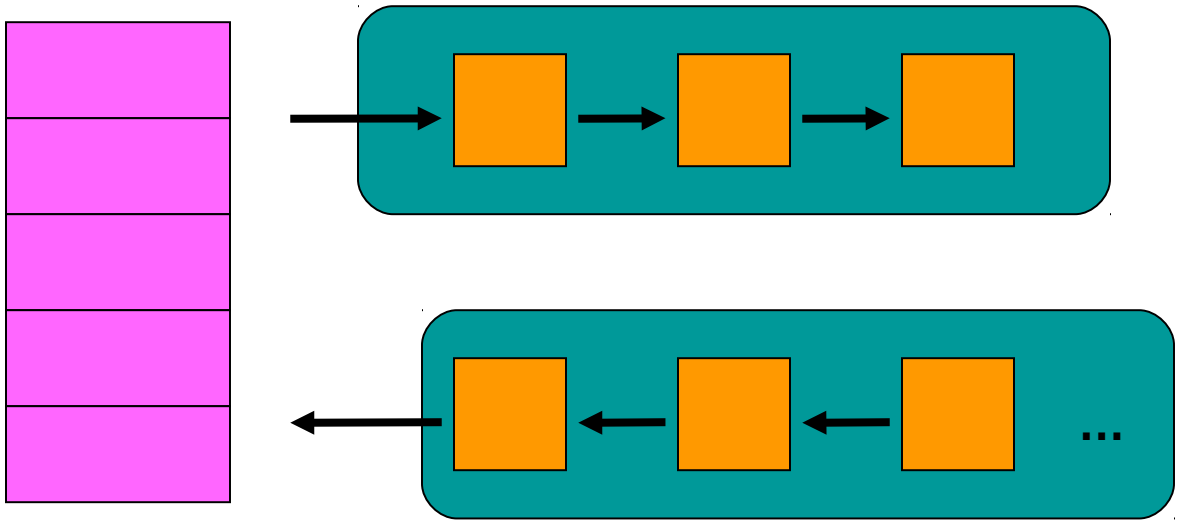
Uma chamada malloc() em OOM-v2 gera aproximadamente 256 novas páginas!

Outra chamada memset() em OOM-v2 pode precisar 256 páginas para ser paged out porque pode não existir páginas livres na memória física.



Algoritmo de troca de páginas

- Como o kernel toma decisões com relação a quais páginas paged out.

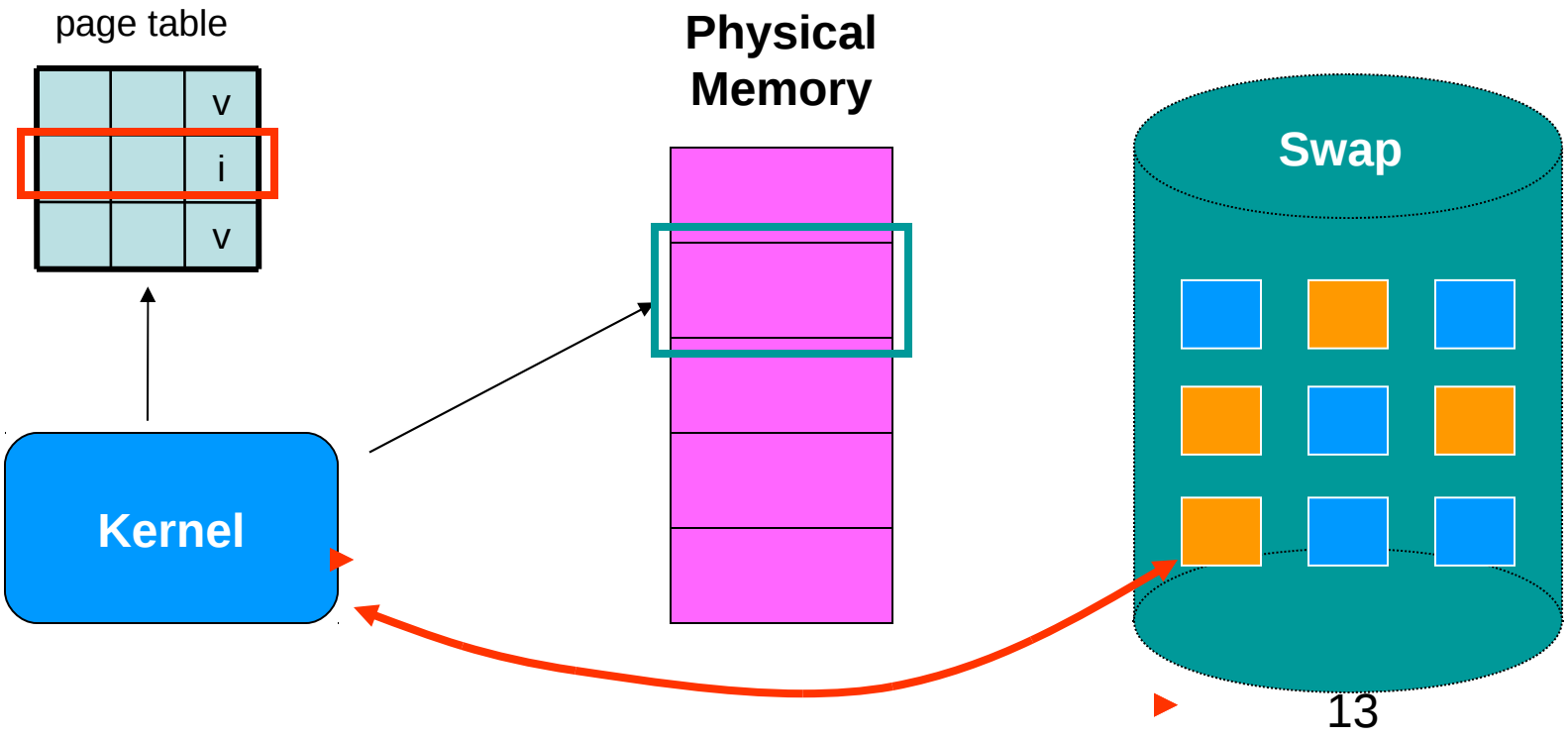


As páginas estão saíndo porque o numero de frames é limitado.

As páginas voltam por causa das referências de memória.

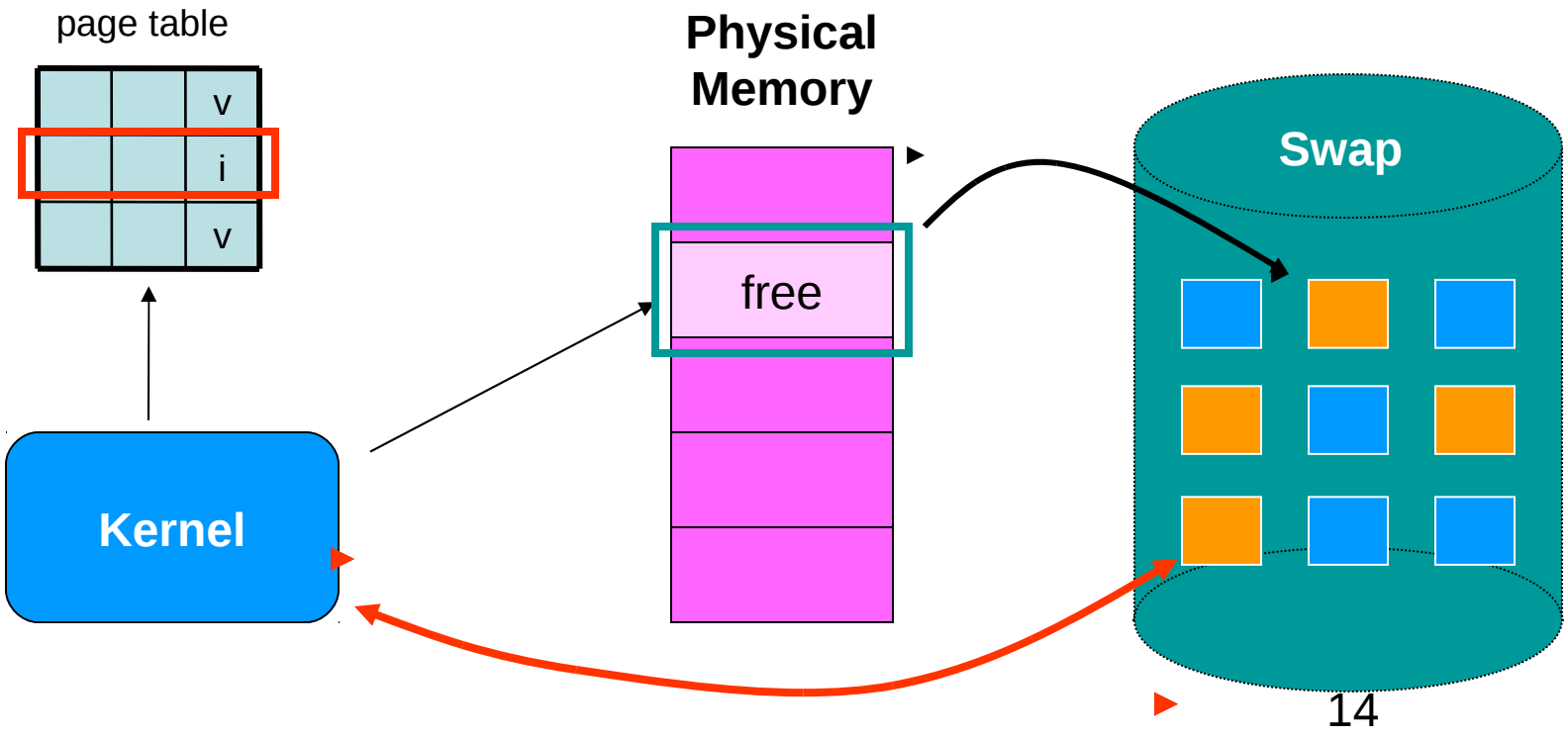
Algoritmo de troca de páginas

- Passos em um algoritmo de troca de páginas:
  (1) Decidir a página vítima na memória física;
  - O kernel mantém dados para tomada de decisão.
  - Desempenho melhor se página escolhida não for muito usada



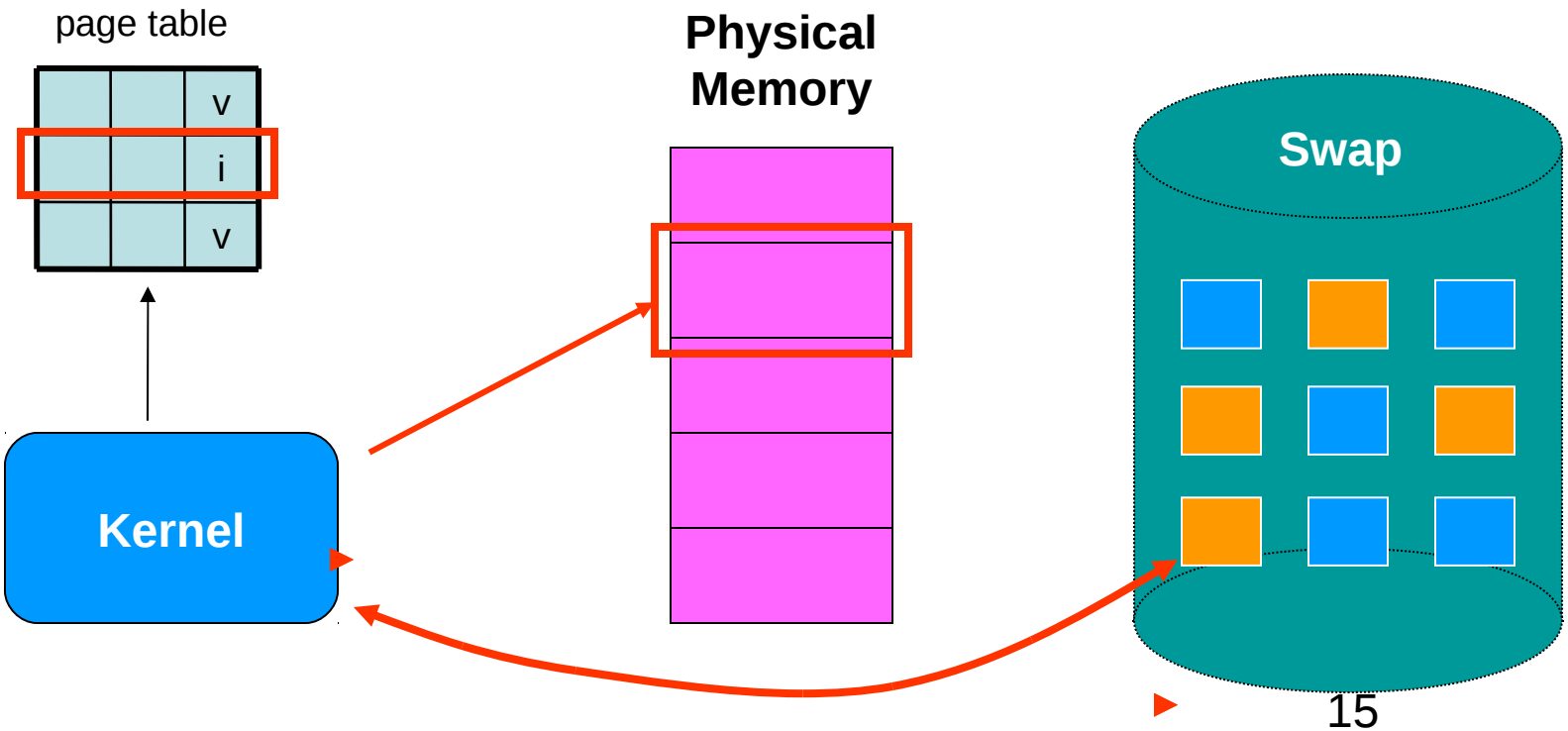
Algoritmo de troca de páginas

- Passos em um algoritmo de troca de páginas:
  (2) Page out a vitima e escreve a mesma de volta para o swap;
  - A página na memória física sobre-escreve a página no swap..



Algoritmo de troca de páginas

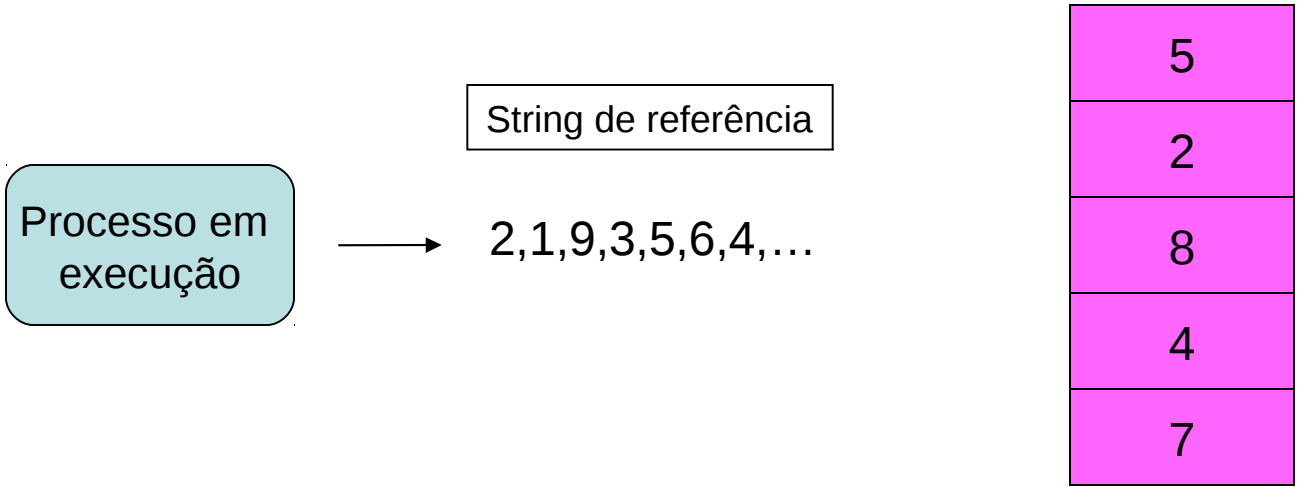
- Passos em um algoritmo de troca de páginas:
  (3) page in o frame desejado;
  - Este passo é similar ao caso quando existe frames livres.



Algoritmo de troca de páginas

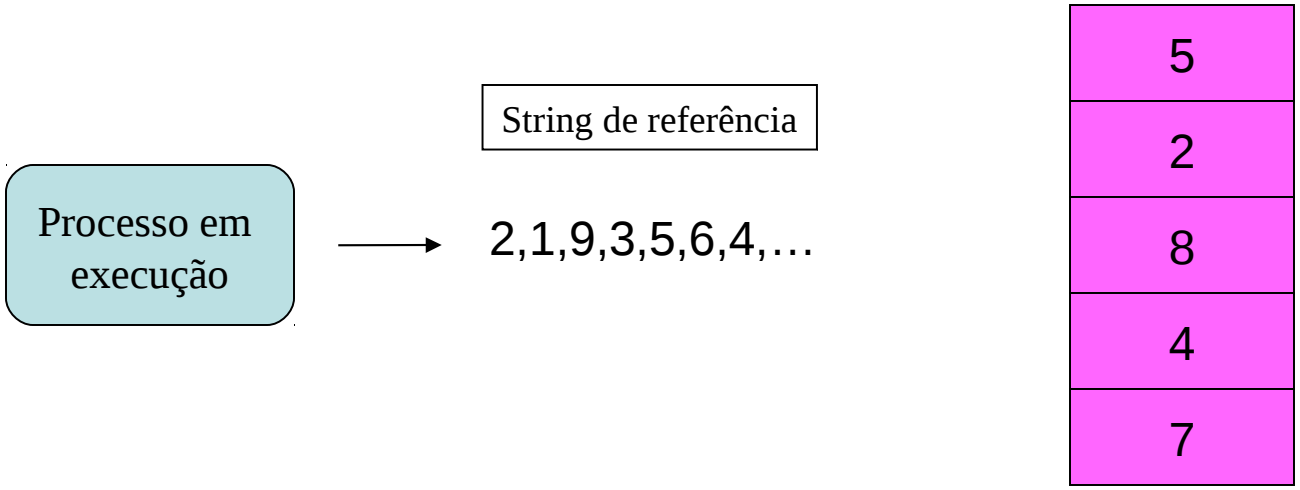
- O kernel precisa saber o seguinte:
  - O estado de alocação atual dos frames.
  - A ordem de referência das páginas, ou string de referência.

- Para simplificar, todas as páginas tem um número.



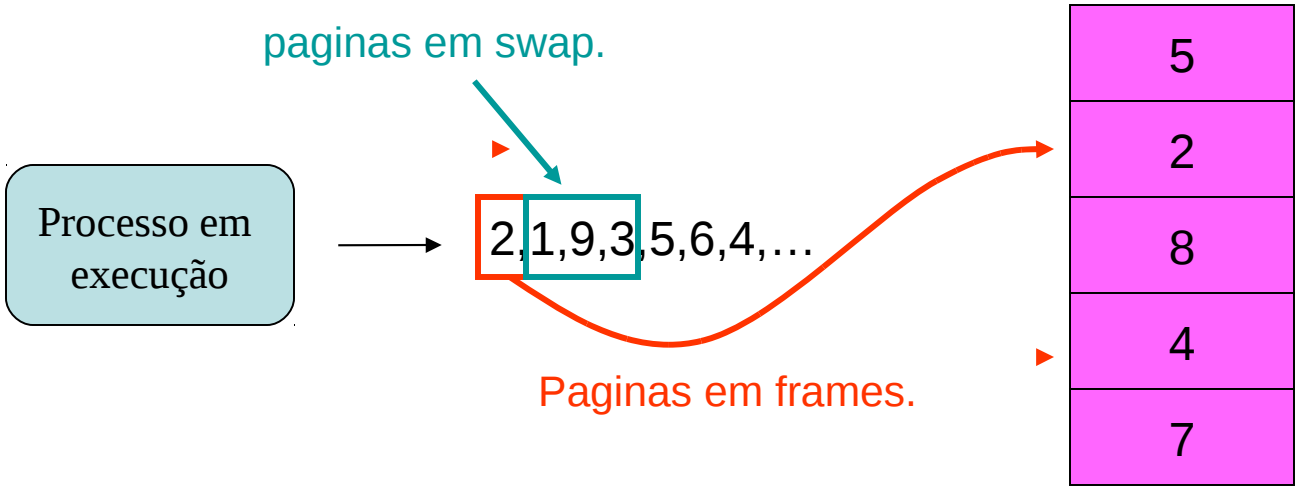
Algoritmo de troca de páginas

- O string de referência é
  - A ordem das páginas que estão sendo referenciadas pelo processo executando.
  - O string de referência tem relação direta com o estado atual dos frames.



Algoritmo de troca de páginas

- Quando o string de referência refere-se a uma página nos frames, nenhuma troca de página é necessária.
- Senão, um algoritmo de troca de página é necessário para encontrar uma página vítima nos frames..



Algoritmo de troca de páginas

- O principal do algoritmo é localizar qual página é a página vítima.

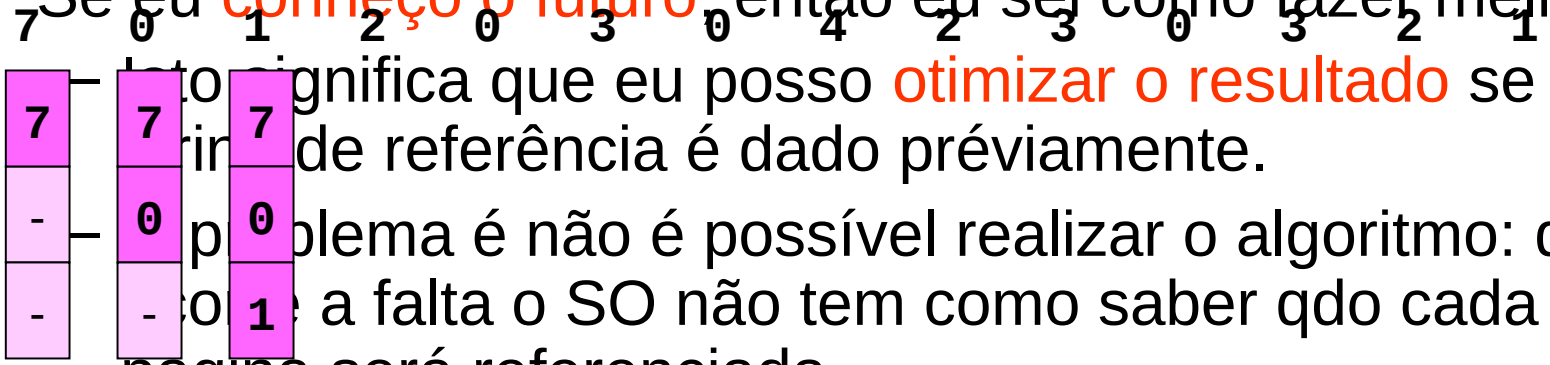
- Em todos os algoritmos de substituição, surge a seguinte questão: remover do processo ou de outros?
  - Duas possibilidades veremos mais tarde
  - Antes os algoritmos de substituição

Algoritmos de substituição de páginas

- Algoritmo ótimo de substituição de página.
- Algoritmo de substituição de página não usada recentemente(NRU).
- Algoritmo de substituição de página primeiro a entrar, primeiro a sair(FIFO).
- Algoritmo de substituição de página segunda chance.
- Algoritmo de substituição de página de relógio(Clock).
- Algoritmo de substituição de página usada menos recentemente(LRU).
- Algoritmo de substituição de página de conjunto de trabalho.
- Algoritmo de substituição de página WSClock.

Algoritmo de troca Ótimo

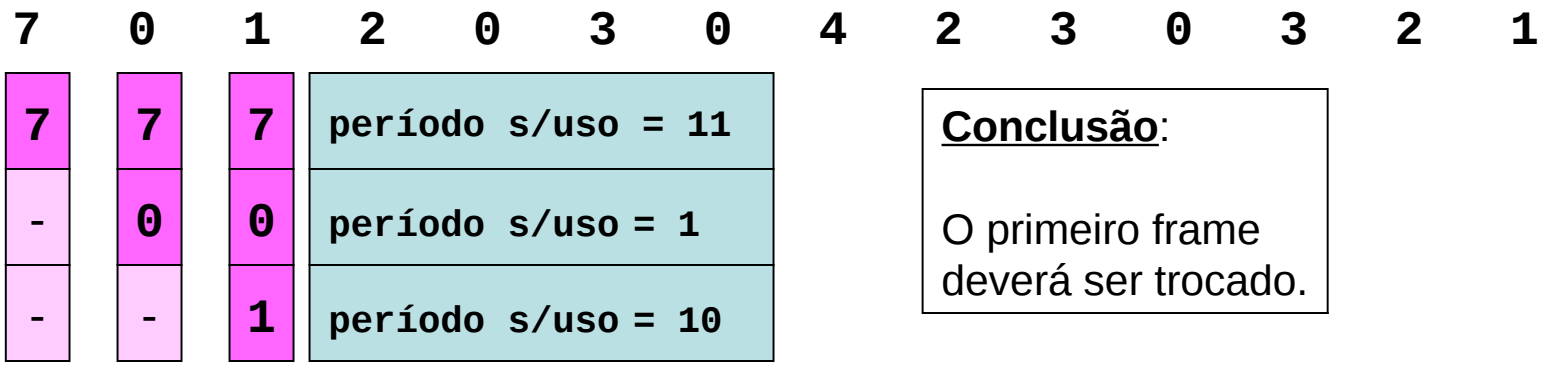
- O melhor algoritmo é fácil de descrever, mas é impossível de implementar.
- Na ocorrência de falta uma página será referenciada, as outras podem ser referenciadas 10, 100, ou até 1000 instruções depois. Rotular cada página com o número de instruções executadas antes da mesma ser referenciada. Remover a página com o maior rótulo.
- Se eu conheço o futuro, então eu sei como fazer melhor.
  - Isto significa que eu posso otimizar o resultado se o padrão de referência é dado préviamente.
  - O problema é não é possível realizar o algoritmo: qdo ocorre a falta o SO não tem como saber qdo cada página será referenciada.



page faults	3

Algoritmo de troca Ótimo

- Estratégia :
  - Trocar a página que não vai ser usada pelo maior período de tempo.



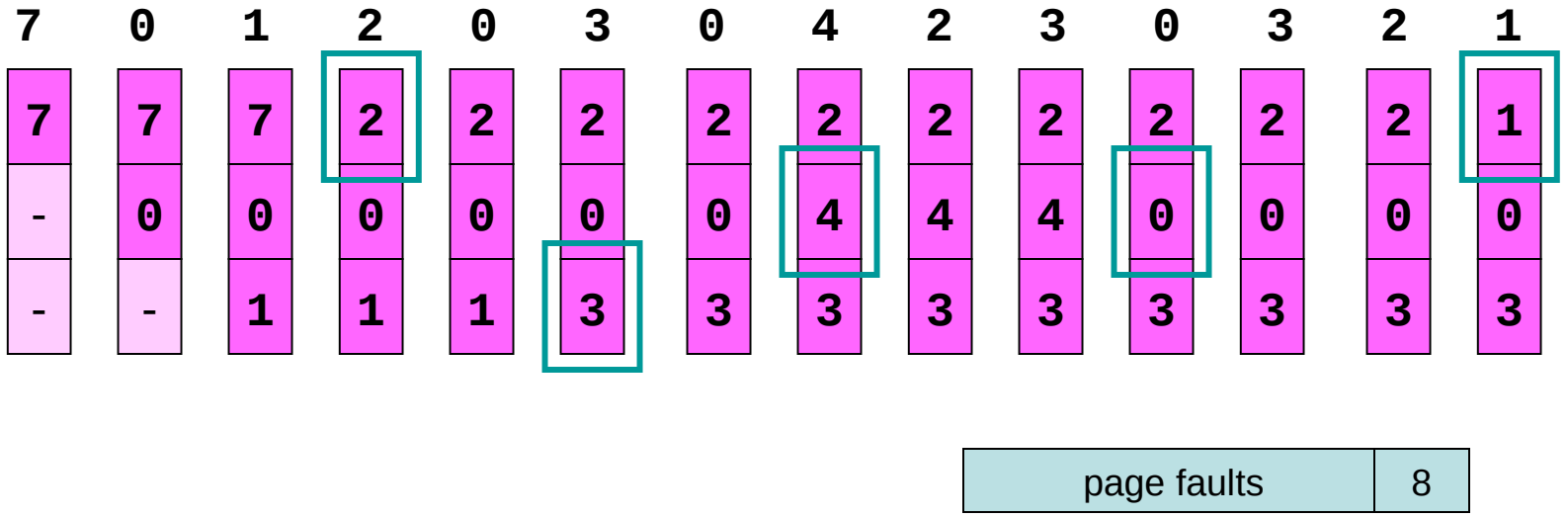
Conclusão:

O primeiro frame deverá ser trocado.

page faults	3

Algoritmo de troca Ótimo

- Estratégia:
  - Trocar a página que não vai ser usada pelo maior período de tempo.



Troca de páginas NRU

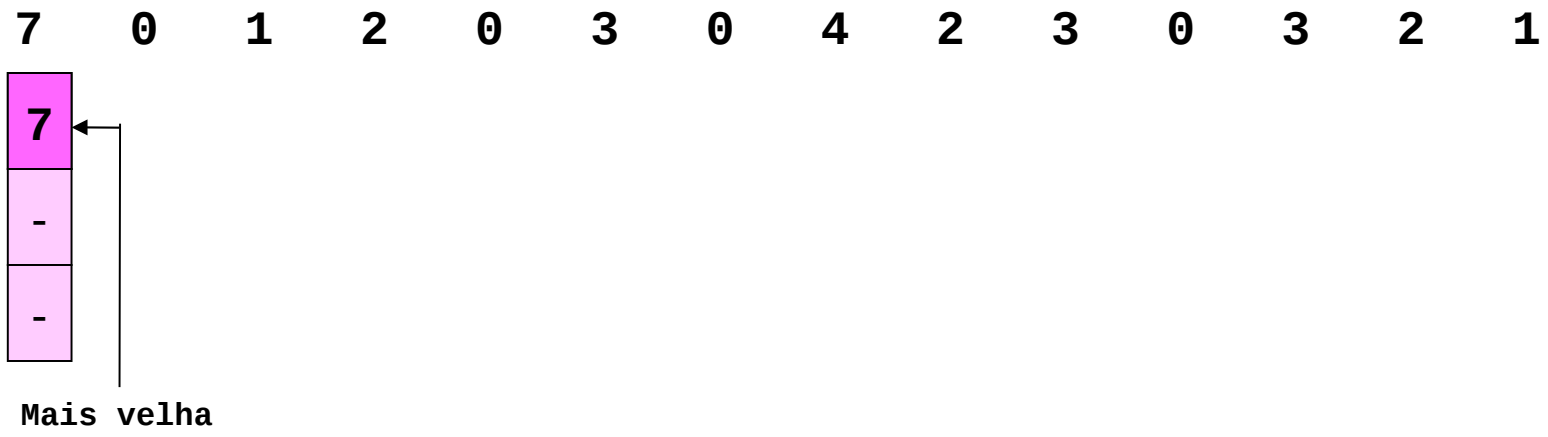
- Usualmente máquinas com MV tem 2 bits de status:
  - Bit referenciado(R), bit modificado(M)
- Se hardware não tem, podemos simular: qdo processo é inicializado as entradas na TP são não presentes na memória. Uma referência ocasiona falta de página:
  - O SO coloca o bit R em 1, altera a entrada da TP para apontar para o frame correto(modo READ ONLY) e inicializa a instrução.
  - Se a página for modificada, outra falta ocorre e o SO pode colocar o bit M em 1, alterando o modo para (READ/WRITE)
- Os bits R e M são usados da seguinte forma: qdo processo é inicializado, os bits são colocados em 0 pelo SO. Periodicamente, o bit R é limpo, diferencia páginas referenciadas/não referenciadas recentemente. Qdo ocorre falta de página SO divide páginas em 4:
  - R=0 M=0, R=0 M=1, R=1 M=0, R=1 M=1
- O algoritmo NRU Not Recently Used remove aleatóriamente uma página da classe de ordem mais baixa que não esteja vazia.
  - Principal vantagem é fácil de entender e de implementar e oferece um desempenho adequado.

Troca de páginas FIFO

- Algoritmo FIFO é de baixo custo.
- A primeira página a ser paged in para os frames será a primeira página a ser paged out dos frames.
  - A página vitima será a página mais velha entre os frames.
  - A idade da página é contada pelo tempo que ela foi armazenada na memória.

Troca de páginas FIFO

- Inicialmente, os frames estão vazios.
  - A primeira página requisitada causará uma falta de página.
  - Porque existe frames livres, nenhuma troca é necessária.



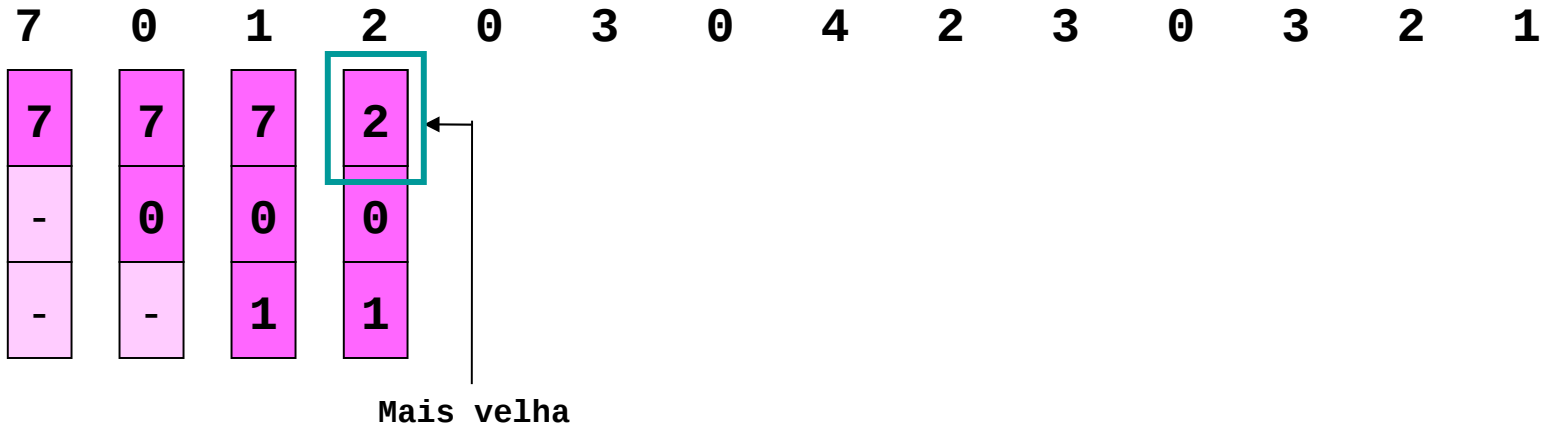
Troca de páginas FIFO

- Quando não tem frames livres,
  - O algoritmo de troca de páginas será executado durante a execução da rotina de tratamento de falta de páginas.

Number of page faults	3

Troca de páginas FIFO

- Quando não existe frames livres,
  - O algoritmo FIFO escolherá a página mais velha para ser a página vítima.
  - A página requisitada tomará o lugar da página vítima.



page faults	3

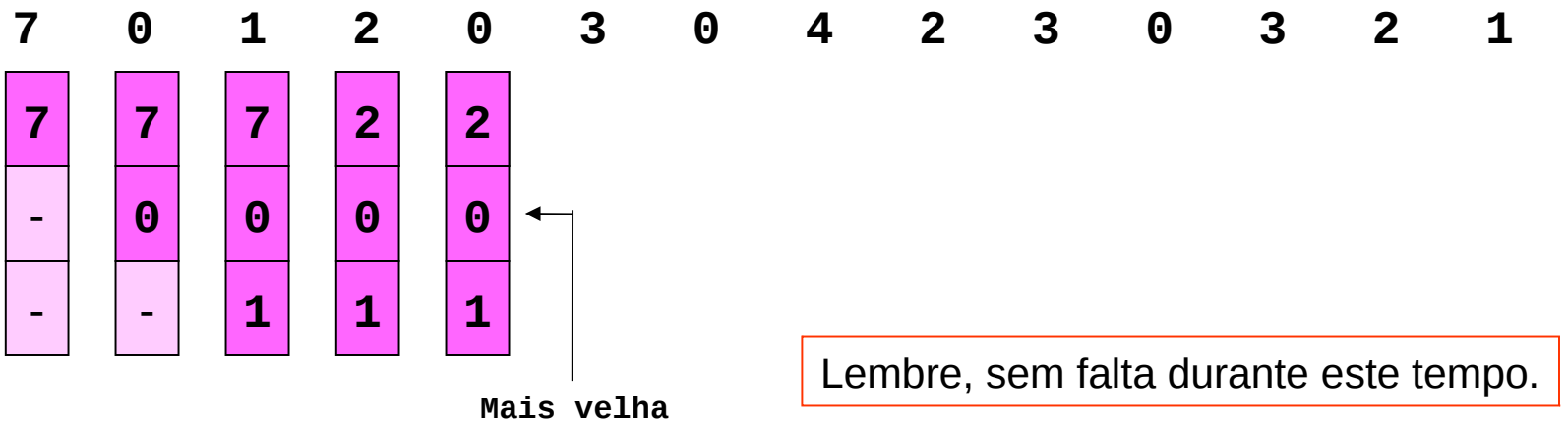
Troca de páginas FIFO

- Quando não existe frames livres,
  - O frame mais velho muda para o segundo mais-velho.
  - O contador de faltas é incrementado.

page faults	4

Troca de páginas FIFO

- Quando a página referenciada é encontrada na memória,
  - A idade do frame não muda.
  - Isto é, o frame com a página 0 continua sendo o mais velho.

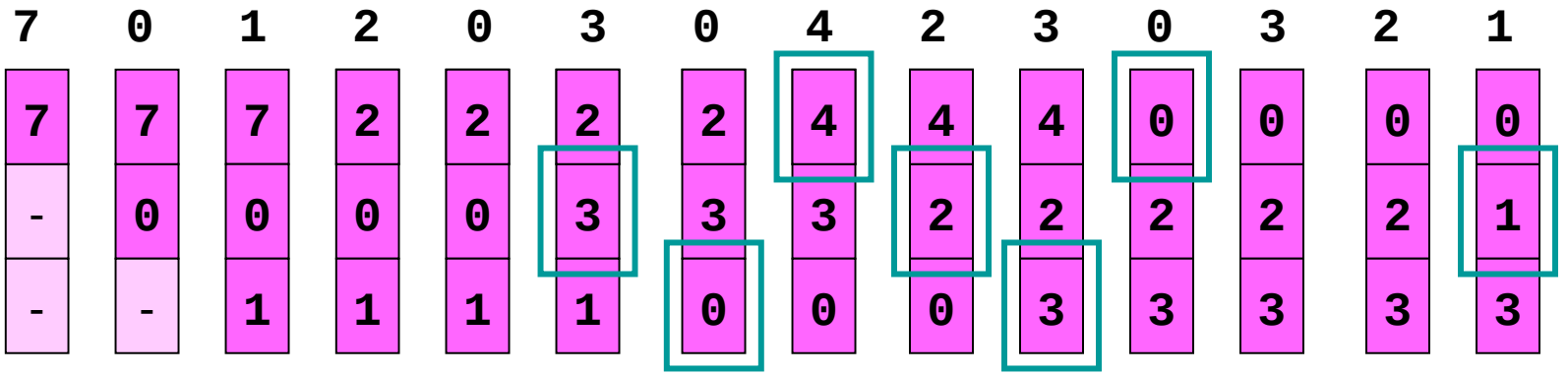


Lembre, sem falta durante este tempo.

Number of page faults	4

Troca de páginas FIFO

- E assim continua…
  - Parece que não existe inteligência neste método…
  - Tanto pode remover página pouco usada quanto página muito usada.
  - FIFO puro, raramente utilizado.



Algoritmo segunda chance

- Uma modificação simples no FIFO, evita remover uma página muito usada, verificando o bit R da página na frente da fila.
- Metodologia:
  - Se o bit R= 0, a página é a mais antiga e não esta sendo usada, sendo substituída.
  - Se o bit R= 1, colocar o bit R em 0, colocar a página no final da lista atualizando o tempo de carga, como se ela tivesse sido carregada agora. A pesquisa continua.
  - Figura a seguir mostra funcionamento

Algoritmo segunda chance

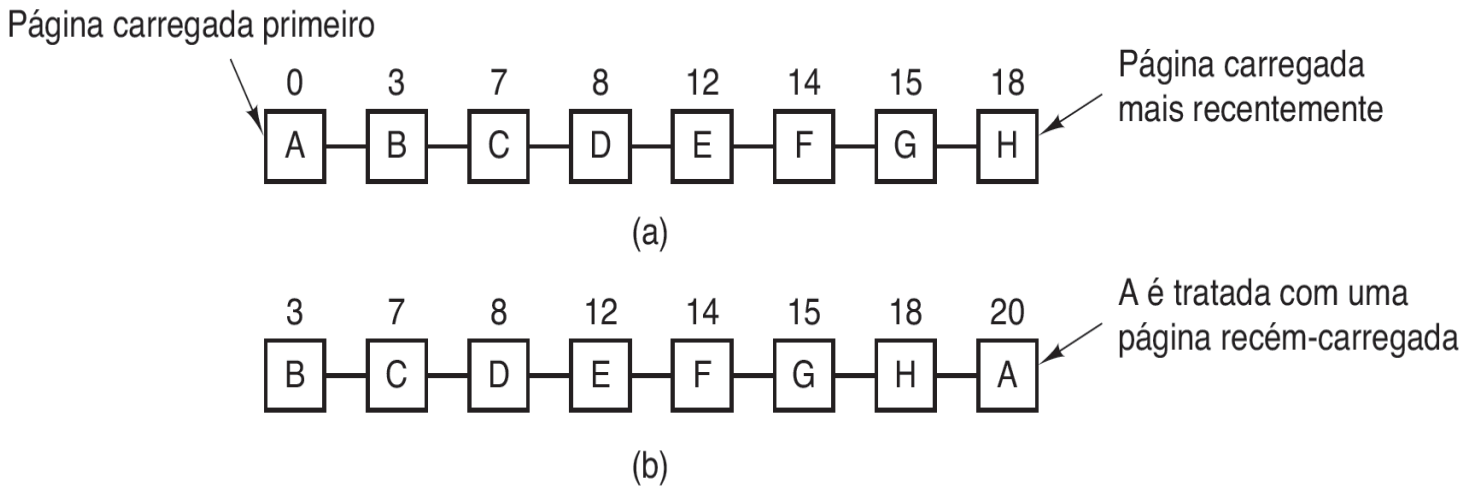
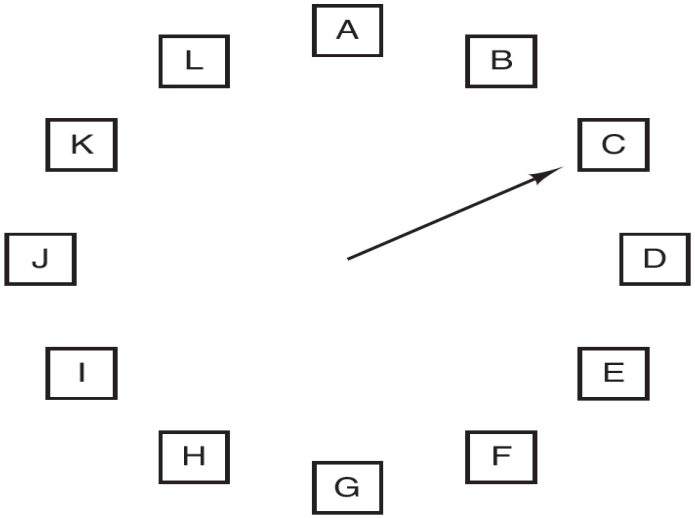


Figura 3.14 Operação de segunda chance. (a) Páginas na ordem FIFO. (b) Lista de páginas se uma falta de página ocorre no tempo 20 e o bit *R* de *A* possui o valor 1. Os números acima das páginas são seus tempos de carregamento.

Algoritmo do relógio (Clock)

- Algoritmo segunda chance é ineficaz porque fica reinserindo páginas no final da fila sem necessidade.
- Uma melhoria na estratégia é fazer a lista circular em forma de relógio, figura a seguir.
- Um ponteiro aponta para a página mais antiga, a cabeça da lista.
- Qdo ocorre uma falta, examina a página indicada pelo ponteiro:
  - Se o bit R=0, a página é removida, a nova página é inserida no seu lugar e o ponteiro avança.
  - Se o bit R=1, o mesmo é zerado e o ponteiro avança.
  - Esse processo é repetido até que seja encontrada uma página com bit R=0.

Algoritmo de substituição de página de relógio



Quando ocorre uma falta de página, a página indicada pelo ponteiro é inspecionada. A ação executada depende do bit R:

R = 0: Remover a página
R = 1: Zerar R e avançar o ponteiro

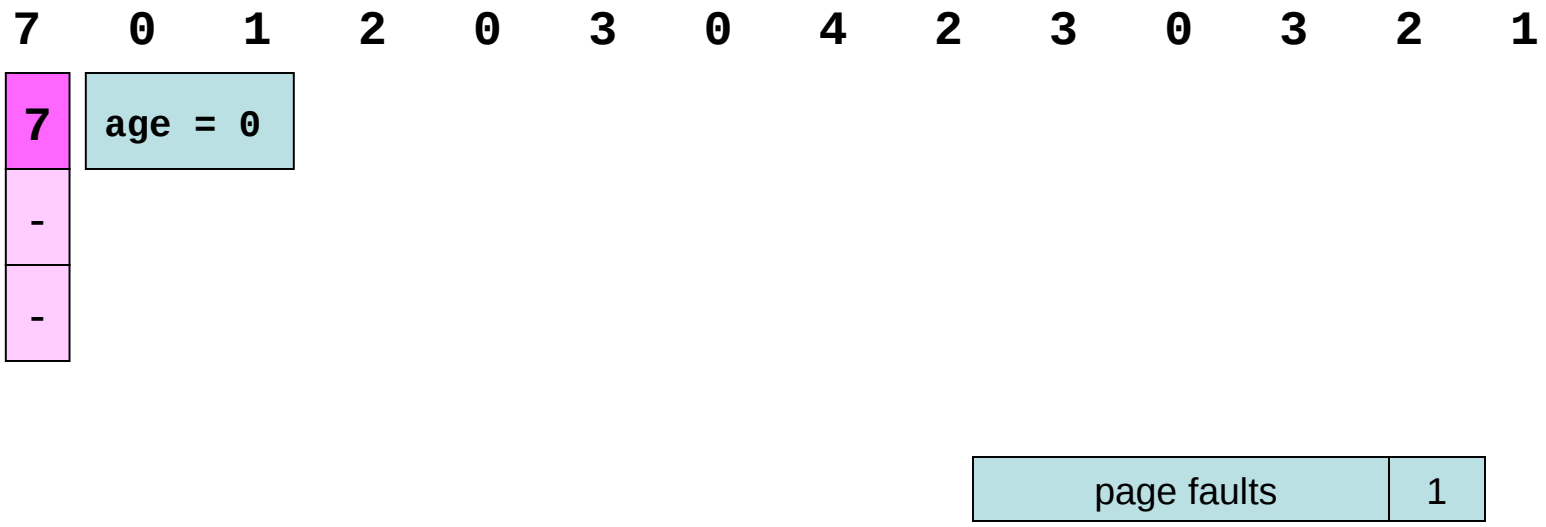
Figura 3.15 Algoritmo de substituição de página relógio.

Algoritmo de troca LRU

- Infelizmente, não conhecemos o futuro… Uma boa aproximação do algoritmo de troca ótimo é baseada na observação de que as páginas muito utilizadas ultimamente, serão muito utilizadas nas próximamente. Ao contrário, páginas não sendo utilizadas por um período longo de tempo permanecerão assim.
- Assim, dqo ocorrer falta eliminar a página não utilizada pelo período de tempo mais longo.
  - estratégia menos recentemente usado (least-recently-used - LRU).
- Metodologia:
  - Atribui a cada frame uma idade.
  - Se o frame é usado, seta a idade do frame em 0.
  - A idade de outros frames é incrementada de 1.

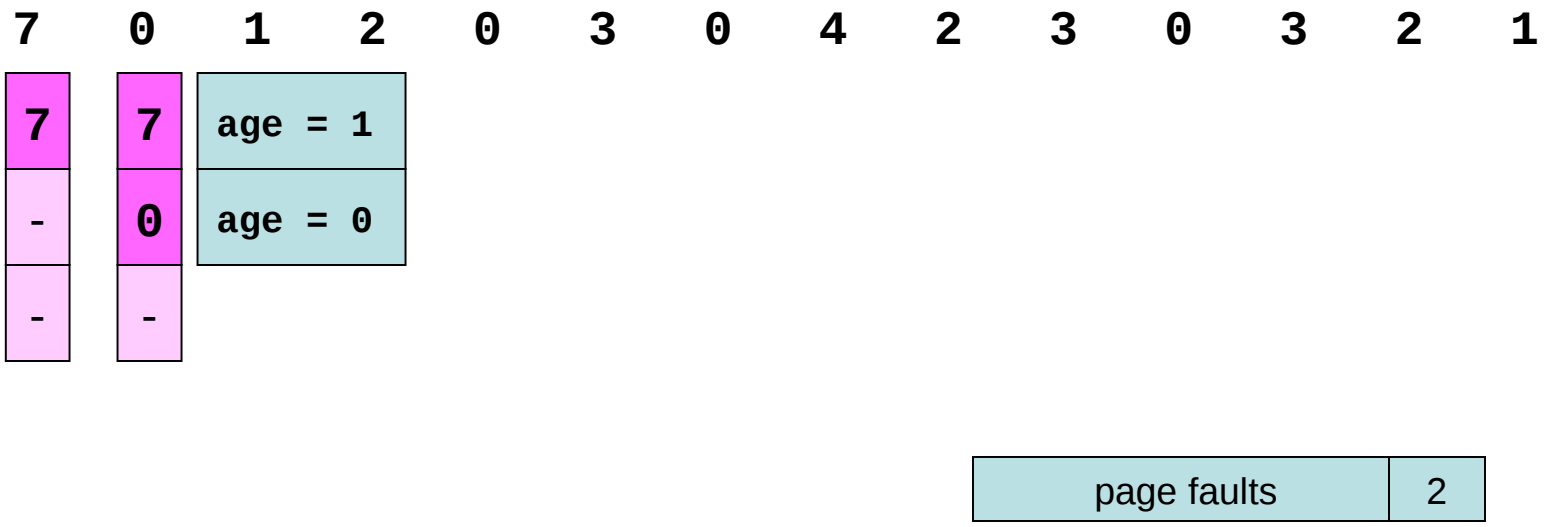
Algoritmo de troca LRU

- Estratégia:
  - Trocar a página que será menos recentemente usada, ou seja, o frame mais velho.



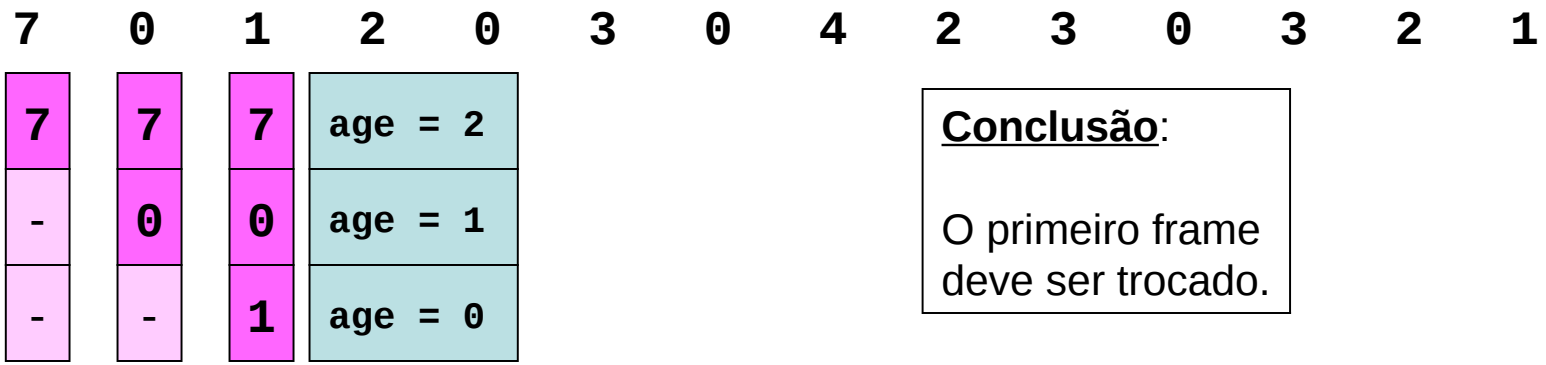
Algoritmo de troca LRU

- Estratégia :
  - Trocar a página que será menos recentemente usada, ou seja, o frame mais velho.



Algoritmo de troca LRU

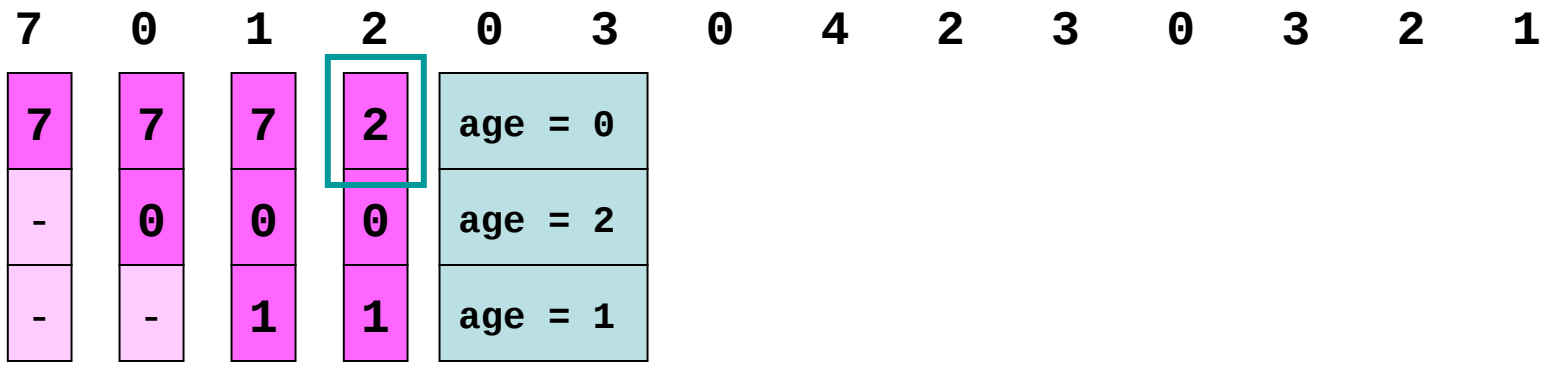
- Estratégia :
  - Trocar a página que será menos recentemente usada, ou seja, o frame mais velho.



page faults	3

Algoritmo de troca LRU

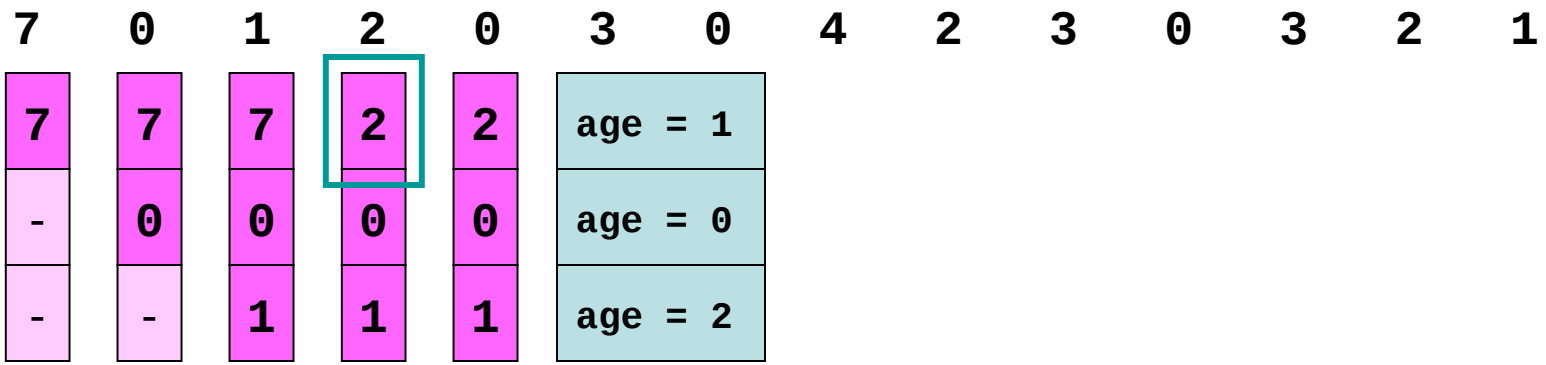
- Estratégia :
  - Trocar a página que será menos recentemente usada, ou seja, o frame mais velho.



page faults	4

Algoritmo de troca LRU

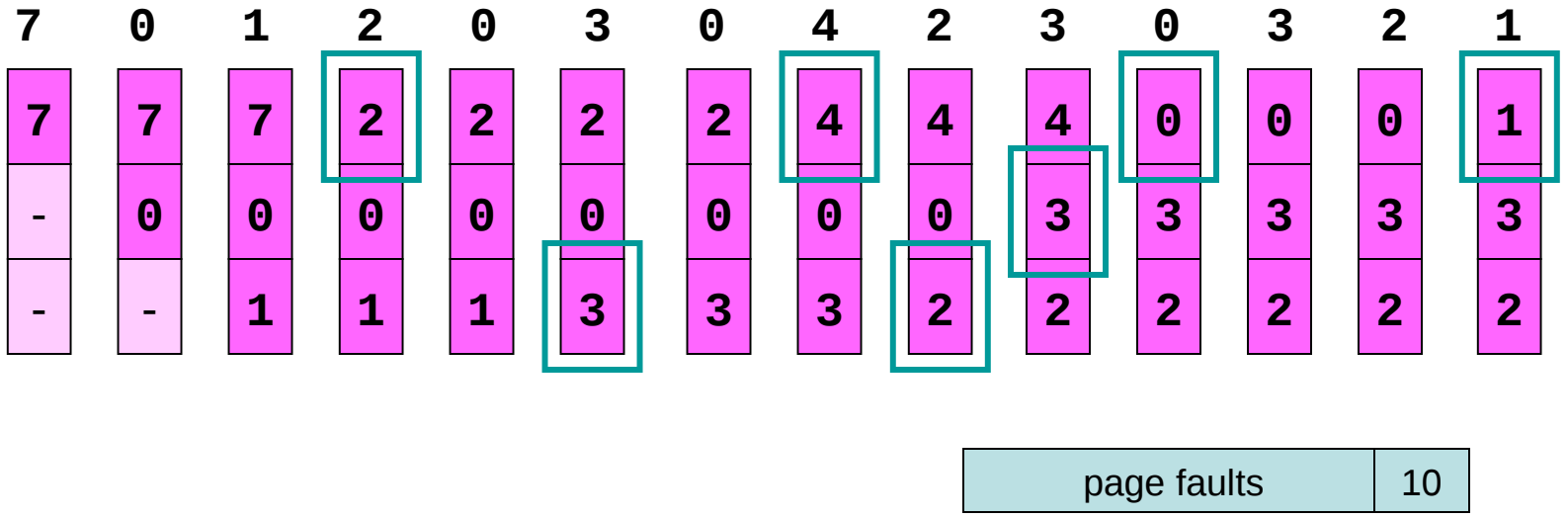
- Estratégia :
  - Trocar a página que será menos recentemente usada, ou seja, o frame mais velho.



page faults	4

Algoritmo de troca LRU

- Estratégia :
  - Trocar a página que será menos recentemente usada, ou seja, o frame mais velho.



Dificuldades do LRU

- Embora realizável teoricamente, o LRU é difícil de implementar
  - Manter uma lista de páginas na memória:
    - frente. mais recentemente usada
    - Trás menos recentemente usada
  - Atualizar a lista em cada referência; encontrar página na lista, deletar e posicionar na frente é caro (mesmo com harware).
- Usualmente, outros modos de implementar com hardware especial:
  - Requer hardware: *C*ontador de 64bits incrementado automaticamente. Cada entrada na TP deve ter o contador, após referência o valor de C é colocado na entrada da referenciada. Na falta o SO busca na TP o menor – usada menos recentemente.
  - Hardware para matriz *n* x *n* bits, iniciados em 0; qdo frame *k* for referenciado marca todos os bits da coluna *k* em 0 e da linha *k* em 1; a linha com o menor valor binário é a LRU – figura a seguir.

Algoritmo de substituição de página LRU

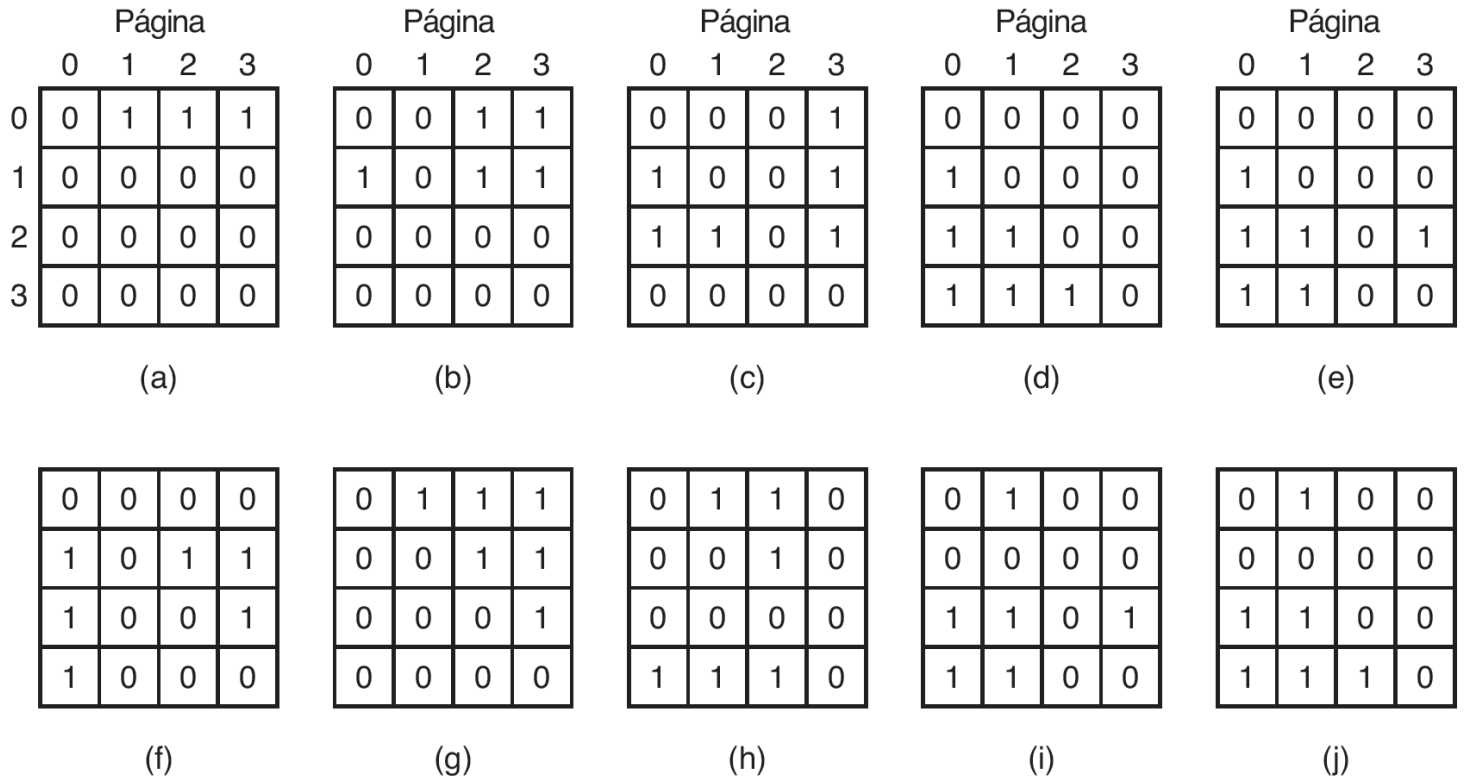


Figura 3.16 LRU usando uma matriz em que as páginas são referenciadas na ordem 0, 1, 2, 3, 2, 1, 0, 3, 2, 3.

Simulação do LRU - software

- Embora realizáveis as implementações anteriores poucas máquinas oferecem o hardware necessário. Buscar alternativas em software como: página não usada frequentemente (**NFU** – *not frequently used*)
- Requer contadores associados as páginas, inicialmente em 0; interrupção de relógio → o SO percorre as páginas na memóra: é difícil de implementar
  - O bit R (0,1) é somado ao contador da página
    - Tentativa de saber quantas vezes cada página já foi referenciada
  - Na falta, a página com a menor contagem será escolhida para substituição.
- Problema principal: NFU nunca se esquece de nada. Exemplo: compilador/montador de múltiplos passos.
- Modificação em NFU → simulação do LRU
  - Primeiro: contadores são deslocados um bit a direita.
  - Em seguida: o bit R de cada página é somado ao bit mais a esquerda do contador
  - Figura a seguir (diferenças do LRU: páginas 3 e 5 (e) – contador 0)

Simulando LRU em software

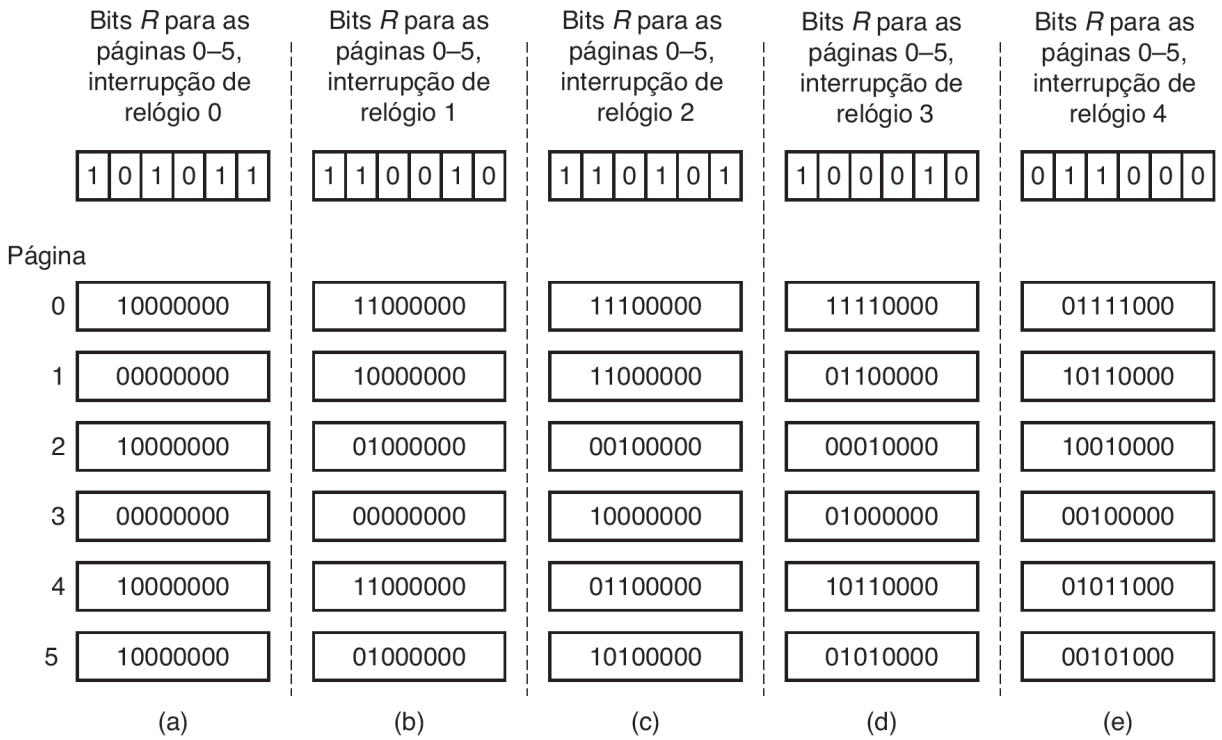


Figura 3.17 O algoritmo de envelhecimento simula o LRU em software. São mostradas seis páginas para cinco interrupções de relógio. As cinco interrupções de relógio são representadas de (a) até (e).

Algoritmo de substituição de página de conjunto de trabalho

- No modo mais puro da paginação, os processos são inicializados sem qualquer de suas páginas presentes na memória – Paginação por Demanda.
- Os processos na sua execução apresentam uma propriedade chamada de localidade de referência.
- O conjunto de páginas que um processo está usando atualmente é denominado conjunto de trabalho (*working set*). Se o WS esta na memória o processo executa com poucas faltas. Se não tiver memória suficiente para o WS o processo terá muitas faltas e executará lentamente.
- Um programa que gere faltas de página frequente e continuamente provoca Ultrapaginação (Thrashing).
- Sistema multiprogramado transferências memória—disco—memória são comuns, o que fazer qdo as páginas relativas a um processoa são trazidas de volta?
- Muitos sistemas tentam gerenciar o WS – *working set model* – reduzir as faltas, usando pré-paginação.

Algoritmo de substituição de página de conjunto de trabalho

Em qualquer instante de tempo, *t*, existe um conjunto que é constituído de todas as páginas usadas pelas *k* referências à memória – conjunto de trabalho.
Para implementar o modelo do conjunto de trabalho, é necessário que o SO saiba quais são as páginas pertencentes ao conjunto de trabalho.

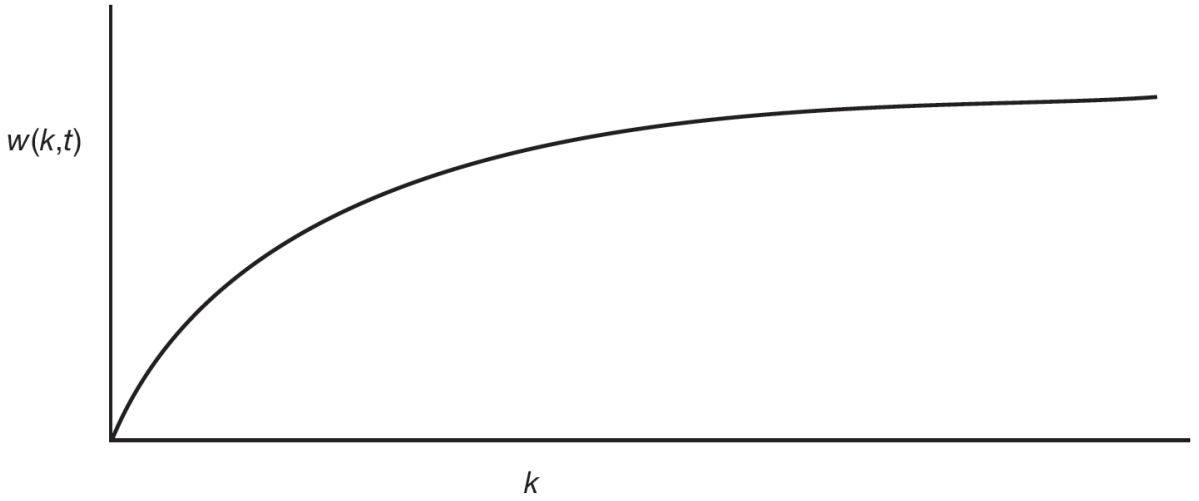
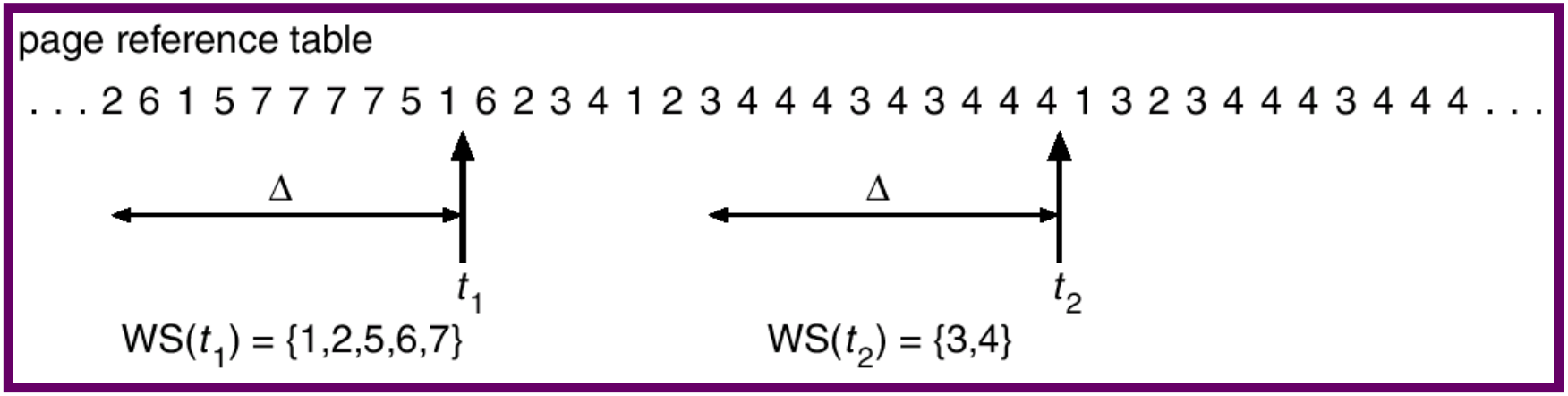


Figura 3.18 O conjunto de trabalho é o conjunto das páginas usadas pelas *k* referências mais recentes à memória. A função *w*(*k*, *t*) é o tamanho do conjunto de trabalho no instante *t*.

Algoritmo de substituição de página de conjunto de trabalho

Um possível algoritmo seria: quando ocorrer falta encontrar página que não pertence ao conjunto de trabalho para remover. É preciso determinar quais páginas pertencem ou não – gerenciar em tempo real. Por exemplo: um registrador de deslocamento com tamanho *k*, cada referência desloca o registrador.

Aproximação muito comum: usar tempo de execução e não numero de referências. Páginas usadas nos ultimos 100ms constituem o conjunto de trabalho: se o processo utilizou no intervalo T, T+100 apenas 40ms de CPU este será considerado – tempo virtual atual.



Algoritmo de substituição de página de conjunto de trabalho

Apenas páginas presentes são consideradas. Cada página contém informação de ultima referência e o bit R. R e M inicializados por hardware. Cada interrupção limpa R. Idade= tempo virtual atual – instante do último uso.

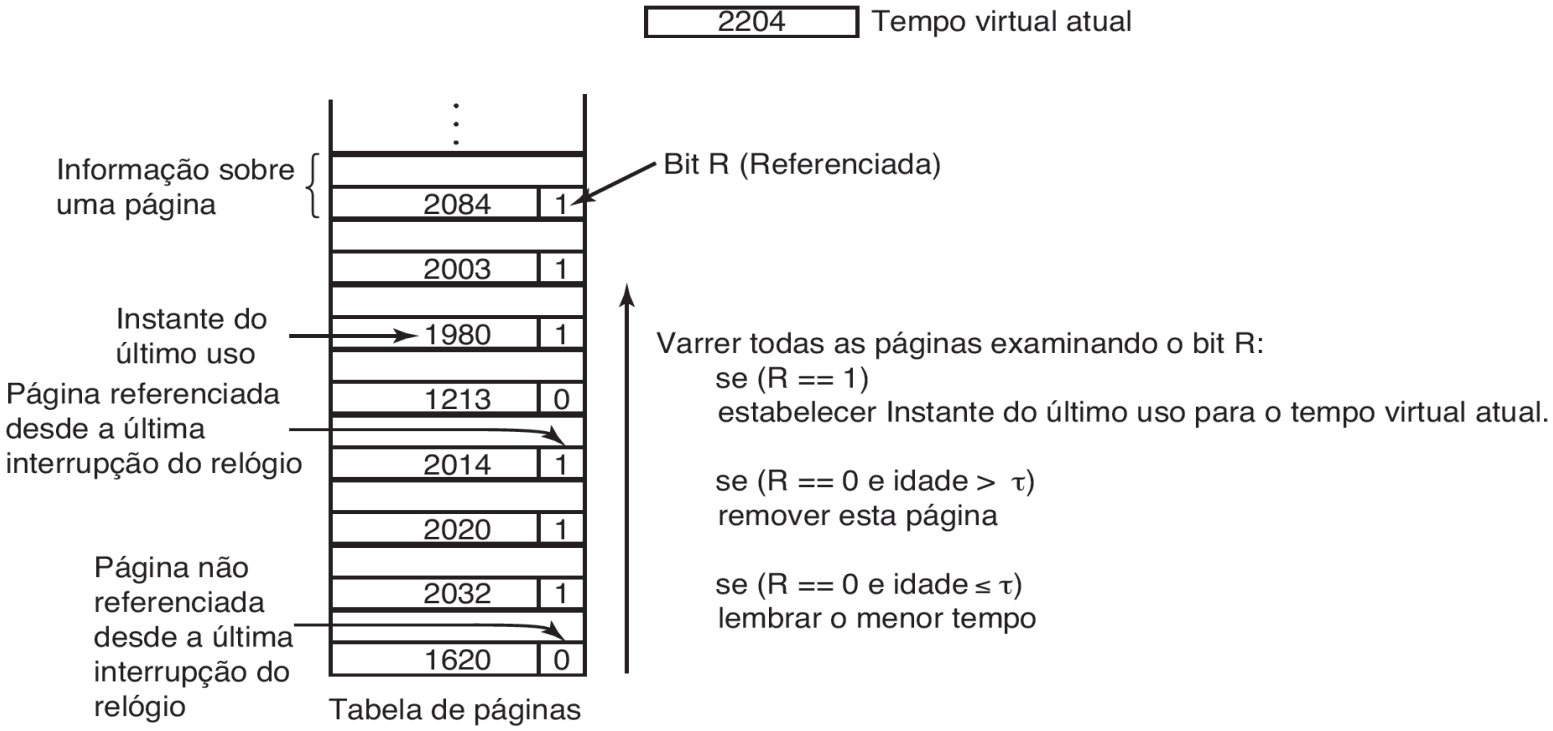
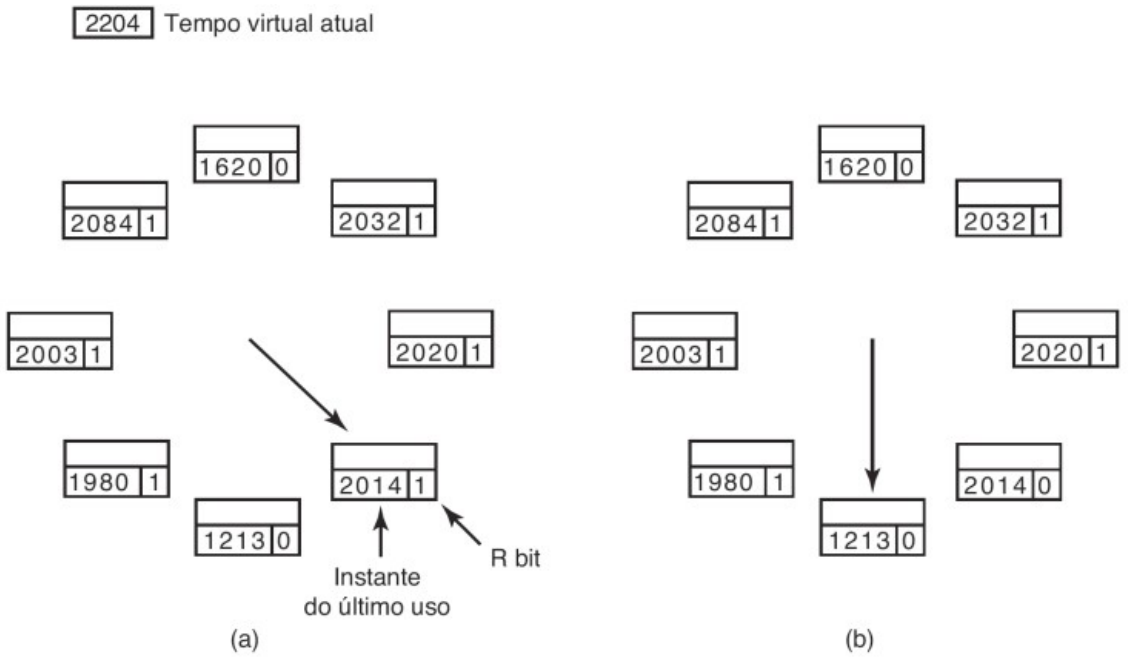


Figura 3.19 Algoritmo do conjunto de trabalho.

Algoritmo WSClock

- WS básico pequisa a tabela em cada falta
- Algoritmo melhorado, baseado no algoritmo do relógio usando informações do WS => bastante utilizado
- Implementação
  - Estrutura de dados = lista circular de frames, vazia. Cada entrada contém **instante do ultimo uso**, bit R.
  - Relógio => cada falta, a página apontada é examinada
    - R==1 => página referenciada, R=0 avança ponteiro
    - R==0 =>
      - Idade > intervalo **p** e página limpa, substitui página suja => escalona escrita, avança
  - O que acontece volta completa
    - Pelo menos 1 escrita escalonada
    - Nenhuma escrita escalonada

Algoritmo de substituição de página WSClock



(***Continua***)

Algoritmo de substituição de página WSClock

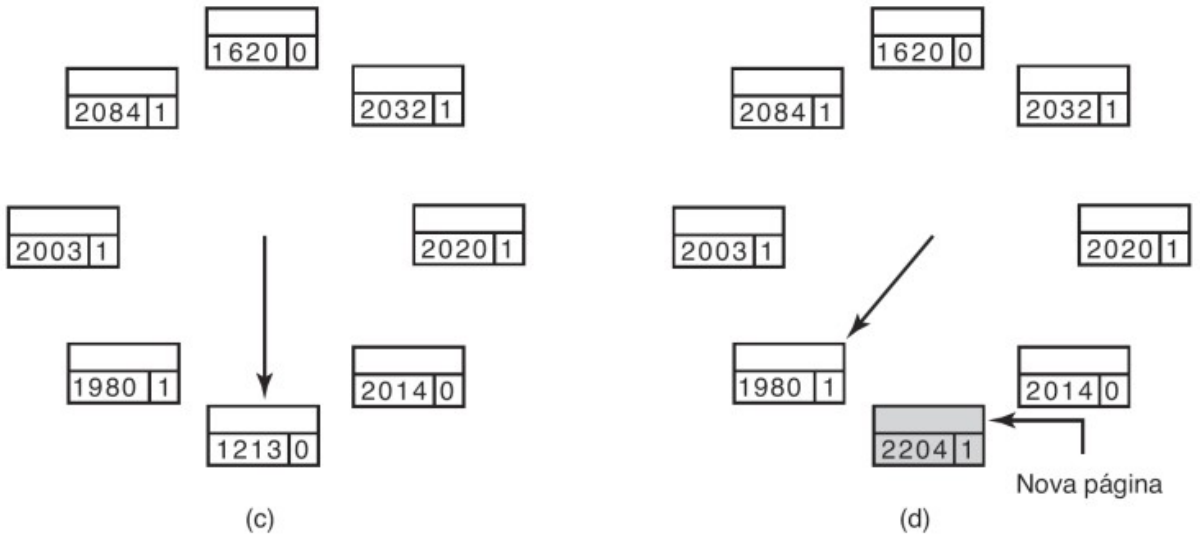


Figura 3.20 Funcionamento do algoritmo WSClock. (a) e (b) exemplificam o que acontece quando $R = 1$. (c) e (d) exemplificam a situação $R = 0$.

Resumo dos algoritmos de substituição de página

Algoritmo	Comentário
Ótimo	Não implementável, mas útil como um padrão de desempenho
NRU (não usada 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 usada)	Aproximação bastante rudimentar do LRU
Envelhecimento (*aging*)	Algoritmo eficiente que aproxima bem o LRU
Conjunto de trabalho	Implementação um tanto cara
WSClock	Algoritmo bom e eficiente

Tabela 3.2 Algoritmos de substituição de página discutidos no texto.