Sistemas Operacionais

Memória Virtual

Lesandro Ponciano

2024

Objetivos da Aula

- Analisar
  - Características da memória virtual
  - Benefícios da memória virtual
- Explicar
  - Conceitos de paginação por demanda
  - Algoritmos de substituição de páginas

Contextualização

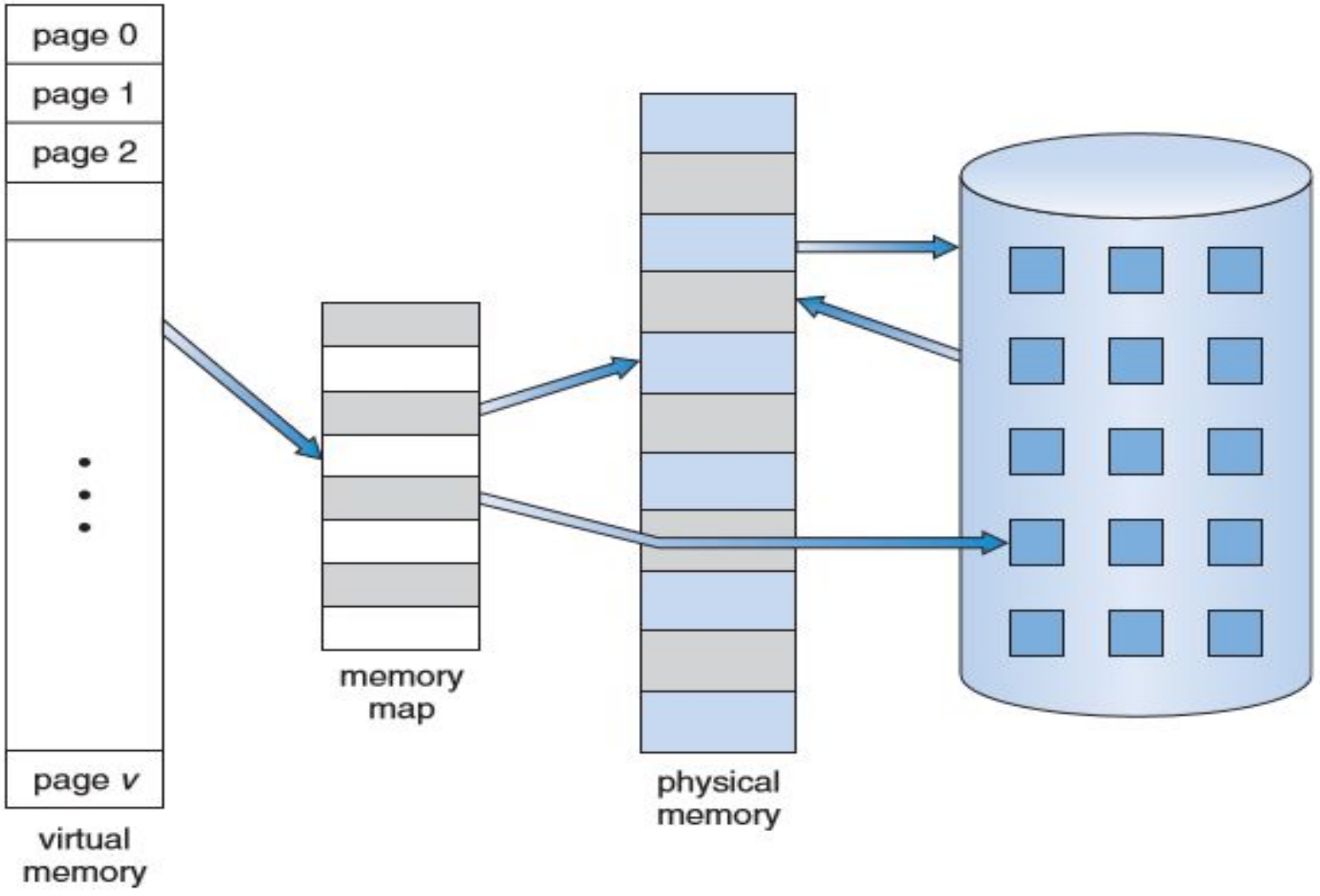
- Suponha um computador com 32 kbytes de memória principal

- Como podemos:
  - rodar programas que usam mais do que 32 kbytes?
  - executar vários programas ao mesmo tempo que juntos usem mais de 32 kbytes?

- Solução: Memória virtual

Memória Virtual

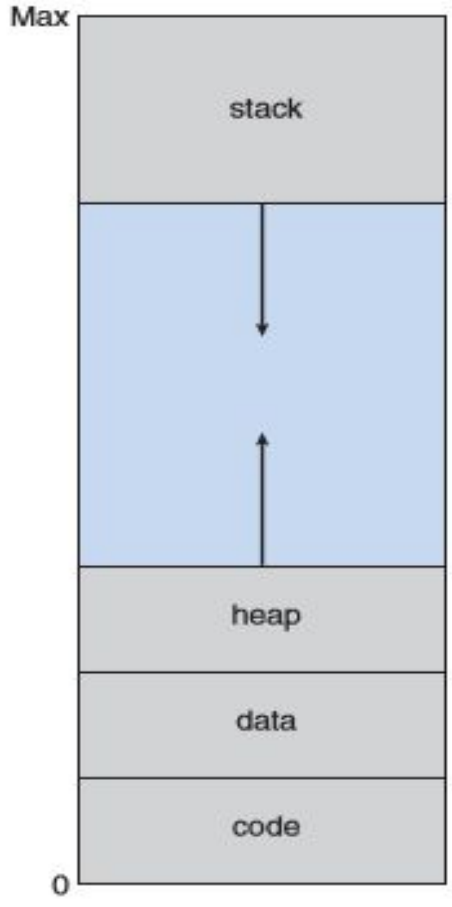
- Envolve a separação entre
  - memória lógica, percebida pelos processos dos usuários
  - memória física, disponível no computador
- Permite que uma memória lógica (virtual) extremamente grande seja fornecida para os programadores quando apenas uma pequena quantidade de memória física está disponível
- Permite ver a memória principal como uma cache de grande capacidade de armazenamento


page 0
page 1
page 2
page v
virtual memory
memory map
physical memory

Espaço de Endereçamento Virtual

- Diz respeito à visão lógica (ou virtual) de como um processo é armazenado na memória

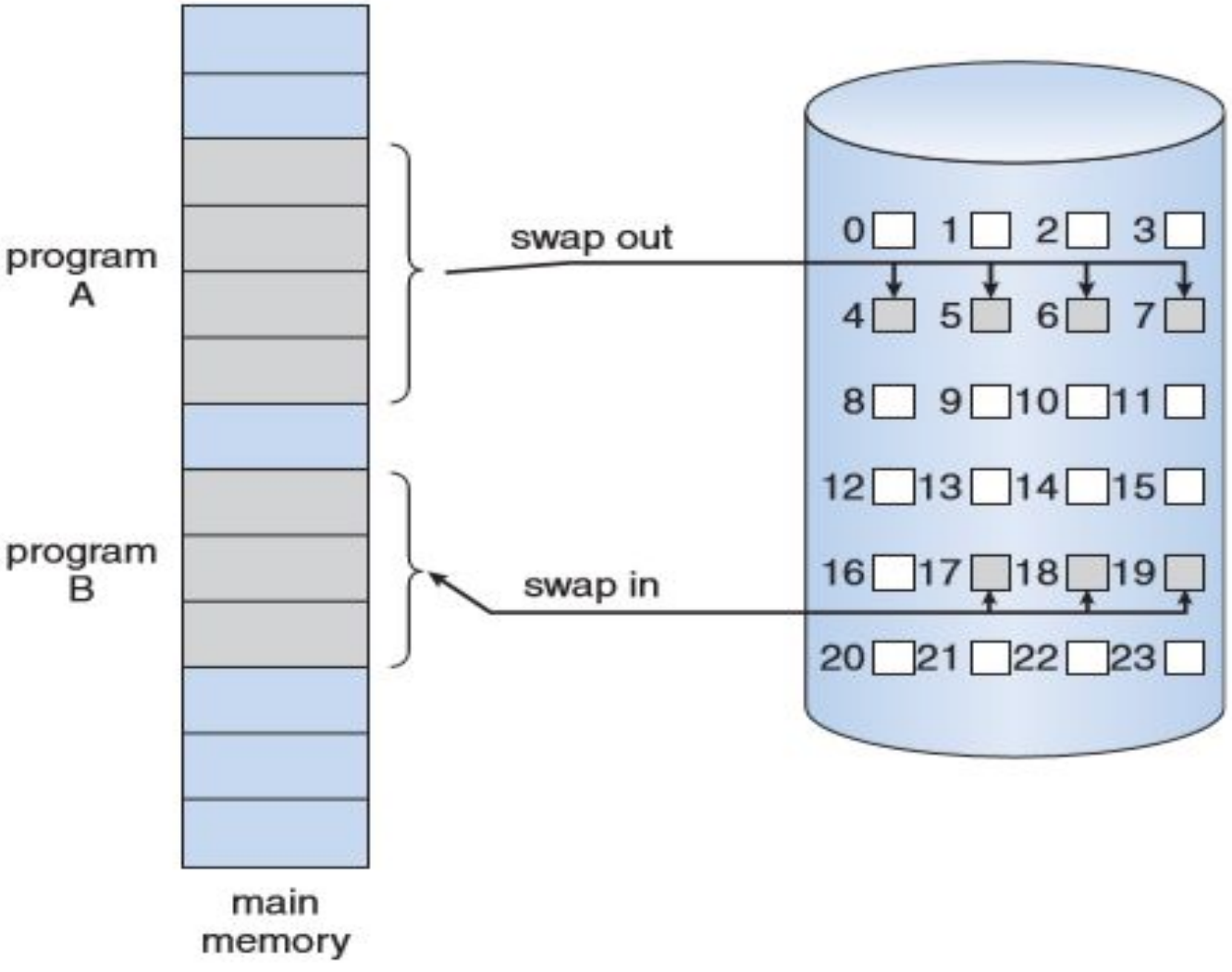
- Espaço da pilha (*stack*) e do *heap* podem crescer virtualmente



Paginação por Demanda

- Em paginação
  - processos são vistos como um conjunto de páginas em vez de um grande espaço de endereçamento contíguo

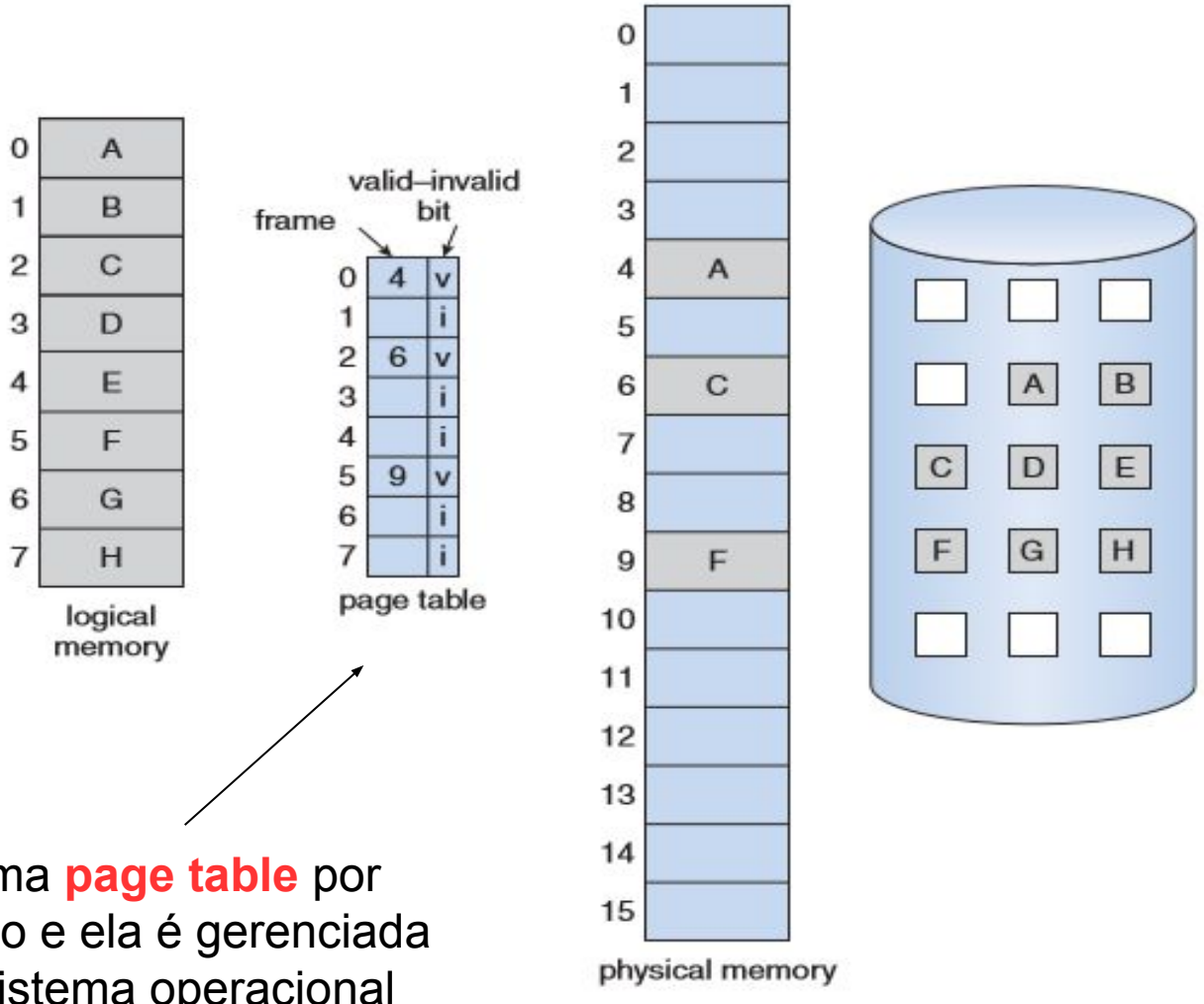
- Na memória virtual paginada por demanda
  - as páginas são carregadas apenas quando necessárias durante a execução do programa
  - Páginas que nunca são acessadas, nunca são carregadas na memória


program A
program B
swap out
swap in
0 1 2 3
4 5 6 7
8 9 10 11
12 13 14 15
16 17 18 19
20 21 22 23
main memory

Bit “Válido - Inválido”

- Para saber quais páginas do processo foram carregadas do disco para a memória, utiliza-se o bit “válido - inválido”
  - Válido: a página é válida e está na memória
  - Inválido: a página é inválida ou encontra-se no disco

- Quando um processo tenta acessar uma página inválida, ocorre um erro de página (*page fault*)
  - O erro gera uma interrupção ao sistema operacional que deve tratá-lo (carregar a página para a memória)
  - Semelhante ao "cache miss" em *caching*



Há uma **page table** por processo e ela é gerenciada pelo sistema operacional

Tratamento do Erro de Página

- Verificar se a referência é um acesso válido à memória
  - Se a referência é um acesso inválido, encerra o processo
  - Se a referência é um acesso válido, a página que ele está tentando acessar deve ser trazida para a memória

- Para carregar uma página para a memória
  1) Encontrar um quadro livre na memória
  2) Trazer a página desejada para o quadro
  3) Quando for trazida, modificar a tabela de páginas para indicar que agora a página está na memória (válido)
  4) Reiniciar a instrução que foi interrompida após o erro de página. Desta vez, o processo será capaz de acessar a página desejada

Substituição de Páginas

- O que acontece se o sistema operacional não encontrar um quadro livre na memória física?
  - Usa-se um algoritmo de substituição de páginas para selecionar um quadro-alvo
  - Grava-se o quadro alvo no disco e altera-se a tabela de páginas de acordo
    - Atualizar o bit “válido – inválido"
- O algoritmo de substituição deve evitar substituir uma página que voltará a ser acessada
  - geraria um novo erro de página

Avaliação de Algoritmos

- Há vários algoritmos de substituição de páginas
  - Geralmente implementados no Sistema Operacional

- O critério de avaliação de um algoritmo é a taxa de erros de páginas
  - Erros de página têm um custo de tempo elevado
  - Para cada erro, será feito um acesso à memória secundária
  - Um algoritmo é melhor quanto menor sua taxa de erros de páginas

Avaliação de Algoritmos

- Dada uma sequência de referência que indica as páginas requisitadas e a ordem em que as requisições ocorrem
  - Executa-se o algoritmo
  - Calcula-se a taxa de erros de páginas

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

Algoritmos

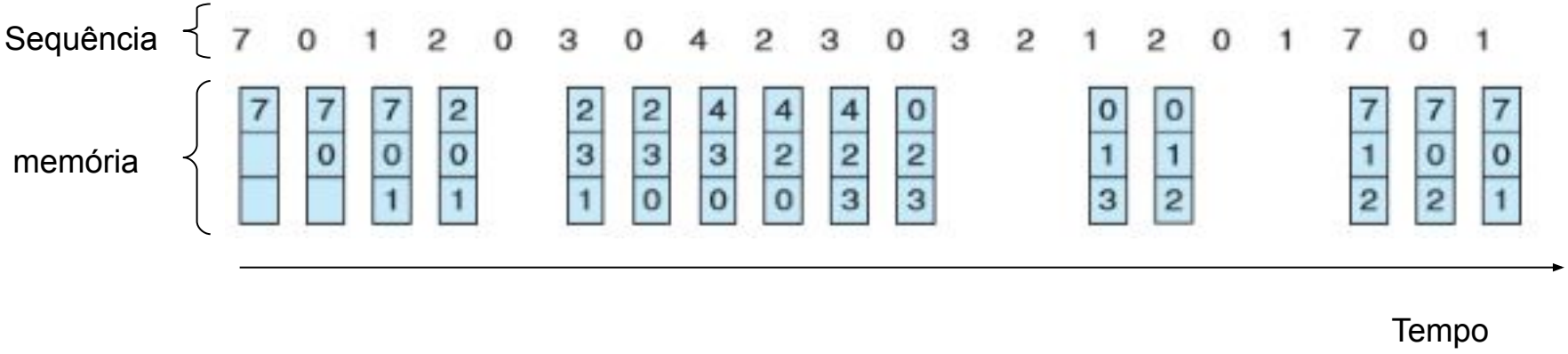
- Substituição de Páginas FIFO
- Substituição Ótima de Páginas (OPT)
- Substituição de Páginas LRU
- Substituição de Páginas com Base em Contagem
  - Substituição de páginas menos frequentemente usadas (Least-frequently-used, LFU)
  - Substituição de páginas mais frequentemente usadas (Most-frequently-used, MFU)

First In, First Out - FIFO

- Primeiro a chegar é o primeiro a sair (*First In, First Out*, FIFO)
- Mantém a ordem em que as páginas foram trazidas para a memória
- Quando uma página tem que ser substituída, a mais antiga é selecionada para deixar a memória
  - Uma nova página assumirá o quadro que ficará livre

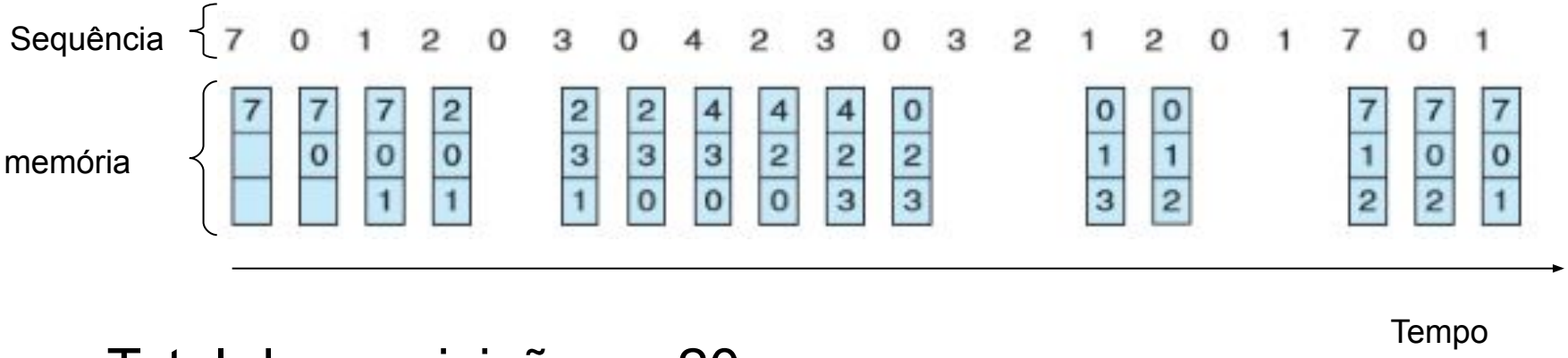
Exemplo FIFO

Memória com três quadros



Exemplo FIFO

Memória com três quadros



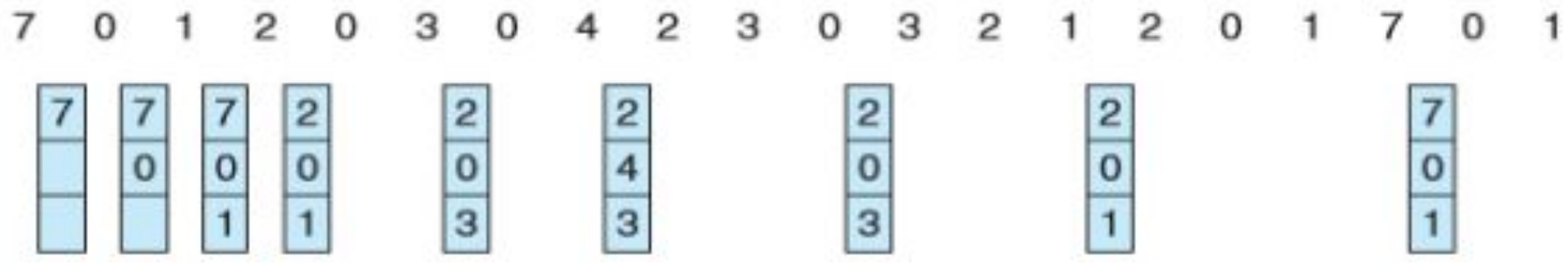
Total de requisições = 20
Total de erros = 15
Taxa de erro = 15/20 = 0.75

Optimum - OPT

- Substitui a página que não será usada pelo período de tempo mais longo
- Garante a menor taxa de erros de páginas possível para uma quantidade fixa de quadros
- É difícil de implementar, pois requer o conhecimento antecipado da sequência de referência

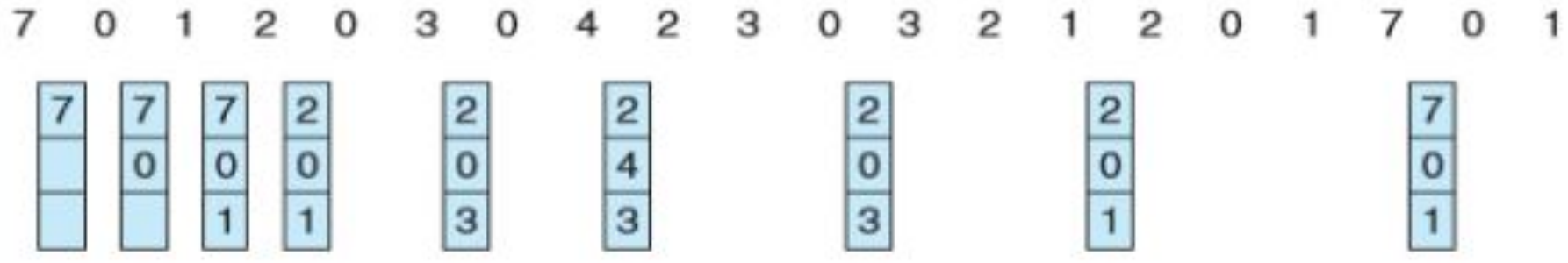
Exemplo do OPT

Memória com três quadros



Exemplo do OPT

Memória com três quadros



Total de requisições = 20
Total de erros = 9
Taxa de erro = 9/20 = 0.45

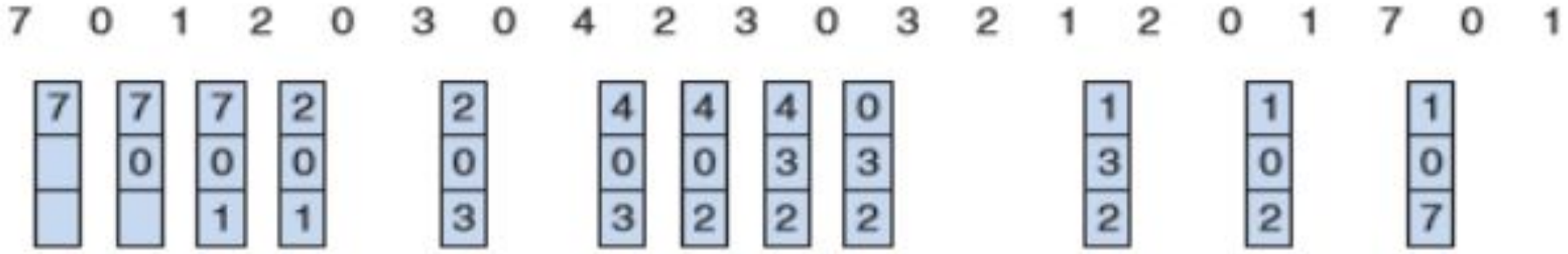
Nenhum algoritmo é capaz de gerar menos que 9 erros de página na sequência dada

.*Least Recently Used - LRU*

- Usa o passado recente como uma aproximação do futuro
- Remove primeiro a página menos recentemente usada
  - *Least-recently-used*, LRU
  - Associa a cada página o tempo do último acesso
- Quando uma página deve ser substituída, o algoritmo LRU seleciona a página que não foi usada pelo período de tempo mais longo

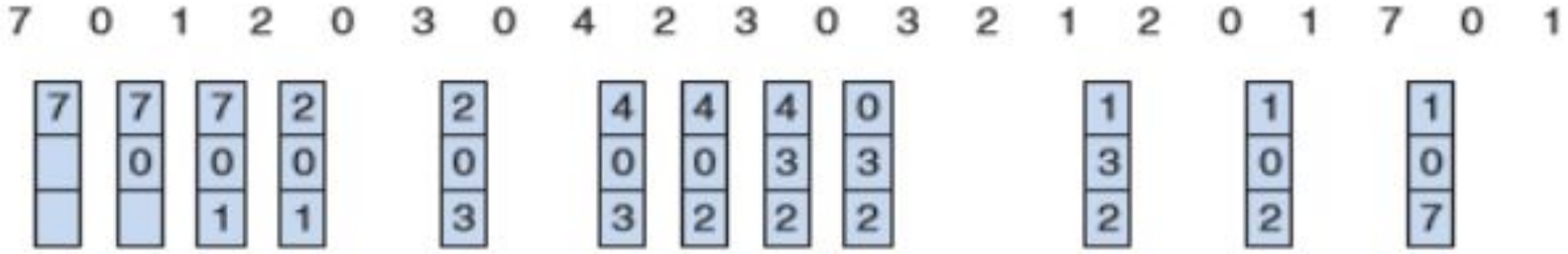
Exemplo do LRU

Memória com três quadros



Exemplo do LRU

Memória com três quadros



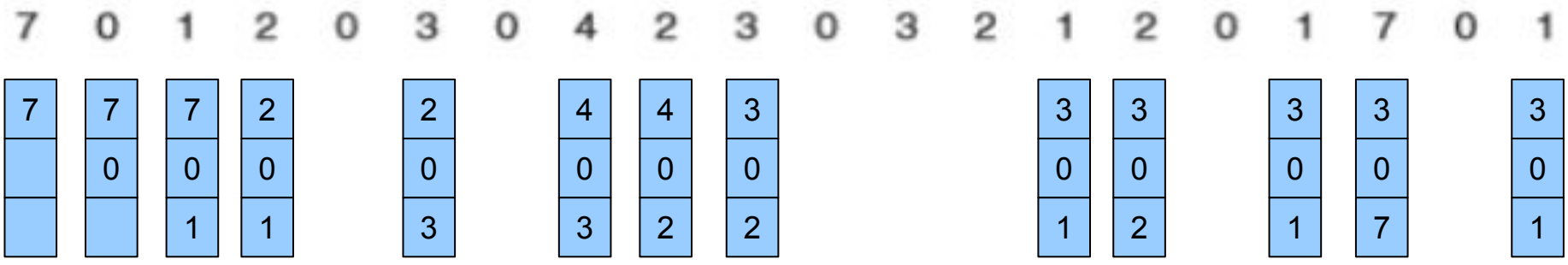
Total de requisições = 20
Total de erros = 12
Taxa de erro = 12/20 = 0.6

Least Frequently Used - LFU

- Baseia-se na contagem de uso das páginas
- Substitui primeiro as páginas menos frequentemente usadas
  - *Least-frequently-used*, LFU
  - O argumento é que uma página usada ativamente deve ter uma alta contagem de referências futuras
  - Pode-se usar FIFO ou LRU como critério de desempate
- A implementação desse algoritmo é cara
- O desempenho pode se aproximar ao do OPT

Substituição de Páginas LFU
usando FIFO como critério de desempate

Memória com três quadros



Total de requisições = 20
Total de erros = 13
Taxa de erro = 13/20 = 0.65

Most Frequently Used - MFU

- Baseia-se na contagem de utilização
- Substitui primeiro as páginas mais frequentemente usadas
  - *Most-frequently-used*, MFU
  - O argumento é que a página com menor contagem provavelmente acabou de ser trazida para a memória e ainda deve ser usada
- A implementação desse algoritmo é cara

Atividade Improdutiva (*Thrashing*)

- Excessiva transferência de páginas e/ou segmentos entre a memória principal e memória secundária
- Um processo está em atividade improdutiva quando está gastando mais tempo em transferência de páginas do que em execução
  - Ocorre apenas em sistemas que implementam memória virtual ou outros mecanismos baseados em acesso à memória secundária

Atividade de Fixação 1

- Explique os seguintes conceitos
  - Memória virtual
  - Erro de página
  - Bit válido-inválido na paginação por demanda
  - Algoritmo de substituição de páginas
  - Taxa de erro
  - Atividade improdutiva

Atividade de Fixação 2

- Mostre a evolução da memória e calcule a taxa de erros de página
  - Tamanho da memória: 3 quadros
  - Sequência de referência: 1, 4, 4, 2, 3, 5, 1, 3, 1, 4, 3, 5, 6, 7, 6
  - Algoritmos OPT, LRU, LFU, MFU, quando necessário, use FIFO como critério de desempate

Referências

SILBERSCHATZ, Abraham; GALVIN, Peter B.; GAGNE, Greg. Fundamentos de sistemas operacionais: princípios básicos. Rio de Janeiro, RJ: LTC, 2013. xvi, 432 p. ISBN 9788521622055

STALLINGS, William. Arquitetura e Organização de Computadores: projeto para o desempenho - 8ª edição.

P. Weisberg and Y. Wiseman, "Using 4KB page size for Virtual Memory is obsolete," 2009 IEEE International Conference on Information Reuse & Integration, Las Vegas, NV, 2009, pp. 262-265.

PONCIANO, L; Andrade, Nazareno ; Brasileiro, Francisco ; Brasileiro, Francisco . BitTorrent traffic from a caching perspective. Journal of the Brazilian Computer Society (Impresso), v. 19, p. 475-491, 2013.

Sistemas Operacionais

Prof. Dr. Lesandro Ponciano

https://orcid.org/0000-0002-5724-0094