Sistemas Operacionais

Memória Virtual

Eduardo Furlan Miranda
2025-05-03

Baseado em: BARBOSA, C. da S. Sistemas Operacionais.
Londrina: EDE SA, 2018. ISBN 978-85-522-1177-8.

Swapping vs. Memória Virtual

- Swapping: move o processo inteiro entre disco e memória principal
  - Um programa completo é carregado para ser executado
  - O processo inteiro é retornado ao disco quando sai da memória
- Memória Virtual (paginação): move partes (páginas) dos processos
  - Programa dividido em unidades menores chamadas páginas
  - Apenas partes (páginas) ativas são mantidas na RAM

Memória Virtual

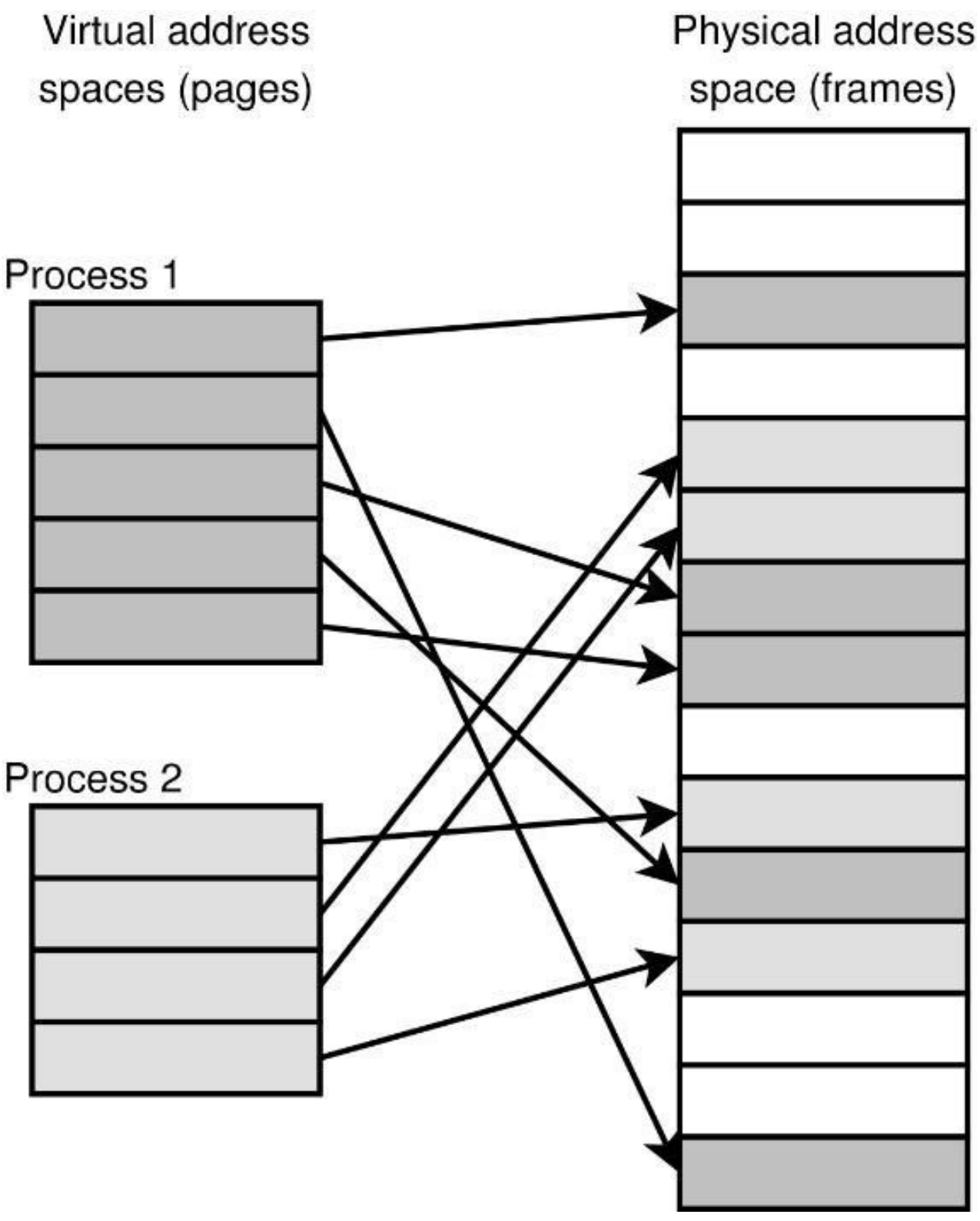
- A memória virtual é um espaço reservado no disco rígido
  - Utilizado quando a memória RAM é insuficiente para executar processos
  - Permite que partes dos processos fiquem em disco
- Lidar com programas maiores que a memória física era um desafio para programadores
  - A técnica de overlay (sobreposição) era usada, dividindo programas em módulos
  - A divisão por overlay era manual, feita pelo programador
- A memória virtual foi desenvolvida para que a divisão do programa fosse feita pelo sistema operacional

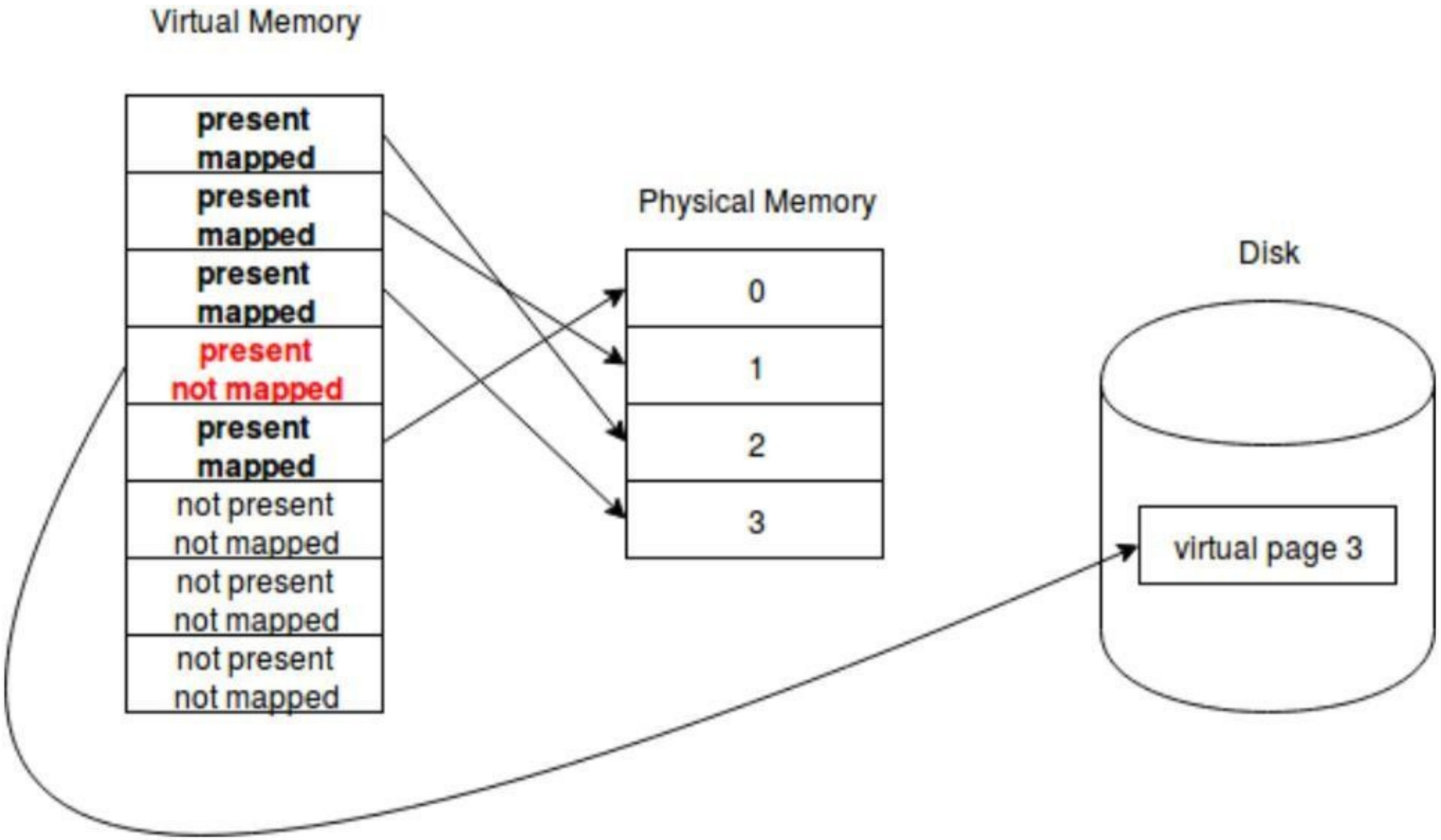
Definindo Memória Virtual

- Permite que o volume de informações de um programa ultrapasse a quantidade total de memória física disponível
  - Mantém as partes ativas na memória principal
  - Mantém as demais partes no disco rígido
- Pode ser um arquivo dinâmico e de tamanho variável, ou uma partição no disco de tamanho fixo
- Permite que vários processos compartilhem a memória principal
  - Somente algumas partes dos processos ficam ativas na memória
  - Possibilita uma utilização eficiente do processador
  - Reduz a fragmentação da memória principal

Memória Virtual vs. Memória Física

- Em sistemas sem memória virtual, o endereço virtual e o endereço físico são os mesmos
- Em sistemas com memória virtual, o endereço virtual é mapeado para o endereço físico
  - Este mapeamento é feito pela MMU (Unidade de Gerenciamento de Memória)
- Programas podem ser maiores que o tamanho da memória física
  - Mas não podem ser totalmente carregados nela, devendo permanecer em disco


Virtual address spaces (pages)
Physical address space (frames)
Process 1
Process 2


Virtual Memory
present
mapped
present
mapped
present
mapped
present
not mapped
present
mapped
not present
not mapped
not present
not mapped
not present
not mapped
Physical Memory
0
1
2
3
Disk
virtual page 3

Paginação: Conceito Básico

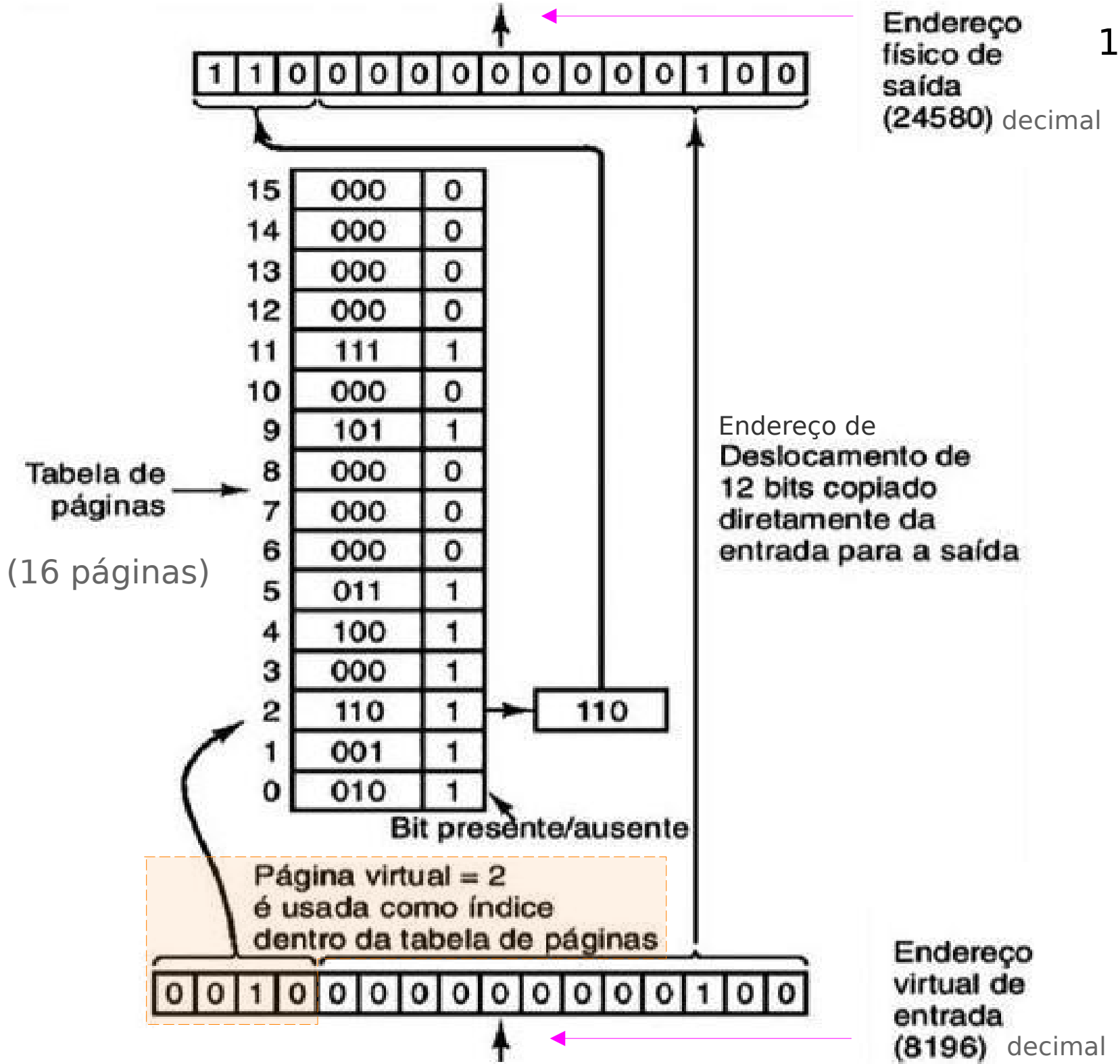
- A paginação é uma técnica de gerência de memória utilizada em sistemas com memória virtual
- Divide o espaço de endereçamento virtual e real em blocos de mesmo tamanho
  - Estes blocos são chamados páginas
- Foi criada para fornecer um espaço de endereçamento linear
  - Sem a necessidade de adquirir mais memória física

Paginação: Espaço de Endereçamento

- Os programas geram endereços virtuais
  - Estes endereços virtuais constituem o espaço de endereçamento virtual
- O endereço virtual divide-se em unidades conhecidas como páginas
- As páginas são referenciadas na memória física pelas molduras de página (frames)
  - Páginas e molduras de página têm o mesmo tamanho
  - A movimentação entre disco e memória é sempre realizada em unidades de página

MMU (Memory Management Unit): Função e Localização

- A MMU é responsável por mapear endereços virtuais em endereços físicos
- É um chip localizado na CPU (Unidade Central de Processamento)
- A CPU gera os endereços virtuais e os envia à MMU
- A MMU, por sua vez, envia os endereços físicos para a memória
- Se um programa acessar uma página que não esteja mapeada, a MMU faz esta validação e desvia para o sistema operacional (falta de página)


Endereço físico de saída (24580) decimal
1 1 0 0 0 0 0 0 0 0 0 0 1 0 0
15 000 0
14 000 0
13 000 0
12 000 0
11 111 1
10 000 0
9 101 1
8 000 0
7 000 0
6 000 0
5 011 1
4 100 1
3 000 1
2 110 1
1 001 1
0 010 1
110
Tabela de páginas
(16 páginas)
Endereço de Deslocamento de 12 bits copiado diretamente da entrada para a saída
Bit presente/ausente
Página virtual = 2 é usada como índice dentro da tabela de páginas
0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0
Endereço virtual de entrada (8196) decimal

MMU: Mapeamento de Endereços

- O mapeamento de endereços virtuais para físicos é realizado pela MMU
- O endereço virtual de entrada é dividido em um número de página e um deslocamento dentro da página
  - Por exemplo, um endereço virtual de 16 bits pode ser dividido em um número de página de 4 bits e um deslocamento de 12 bits
- O número da página é usado como índice na tabela de páginas
- O deslocamento não sofre alteração e é concatenado ao número da moldura de página para formar o endereço físico

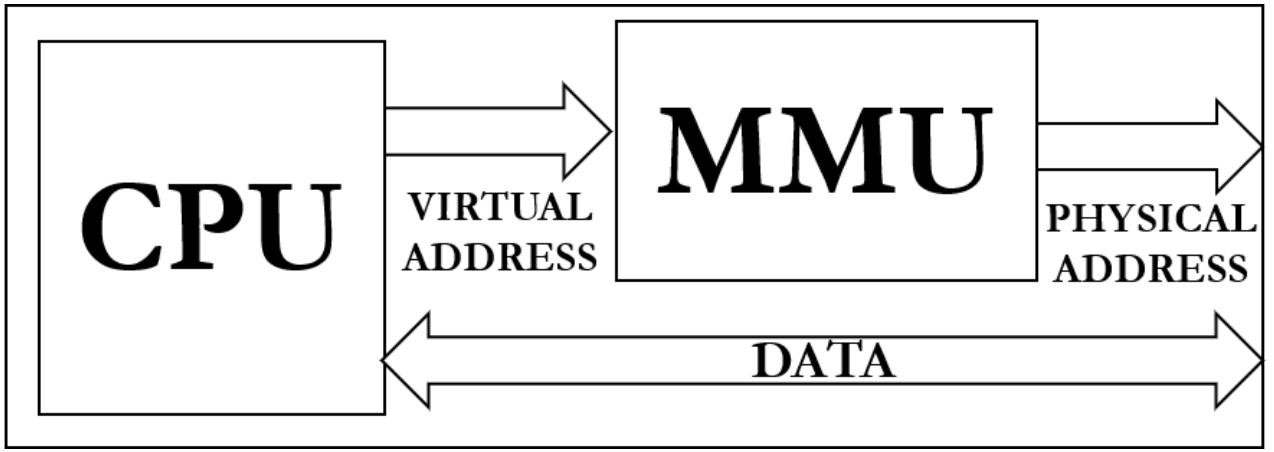


Tabela de Páginas: Propósito e Conteúdo

- O objetivo da tabela de páginas é mapear páginas virtuais em molduras de página física
- Cada processo possui sua tabela de páginas própria
- Cada página possui uma entrada na tabela de páginas
- A tabela de páginas contém o endereço virtual de cada moldura de página na memória física

Tabela de Páginas: Bit Presente/Ausente

- Cada página na tabela possui um bit presente/ausente
- Se o bit for 0 (zero), indica uma interrupção por falta de página
- Se o bit tiver o valor 1, a página está mapeada na memória
- O número da moldura de página encontrado na tabela é usado para formar o endereço físico

Falta de Página (Page Fault): Quando Ocorre

- Uma falta de página ocorre quando um programa tenta acessar uma página que não está mapeada na memória RAM
- A MMU valida o acesso e desvia para o sistema operacional
- Esta interrupção (trap) é chamada de falta de página

Falta de Página: Tratamento

- Quando ocorre uma falta de página, o sistema operacional precisa escolher uma página a ser removida da memória
  - A página removida libera espaço para uma nova página ser trazida
- Se a página na memória teve alteração, ela deve retornar ao disco rígido,
  - para atualizar a cópia da página virtual que está lá
- Se não houve alteração, não é preciso retornar com a página para o disco
- O sistema operacional seleciona uma moldura de página, salva seu conteúdo (se modificado) no disco, marca a entrada na tabela de páginas como não mapeada, carrega a nova página e atualiza a tabela

Algoritmos de Substituição de Páginas: Introdução

- Um dos maiores desafios na gerência de memória virtual por paginação é decidir quais páginas devem ser liberadas
- Quando uma falta de página ocorre, o sistema operacional precisa escolher uma página a ser removida
- O objetivo é selecionar as páginas com as menores chances de serem referenciadas (utilizadas) no futuro próximo
- Quanto menor for o tempo gasto com as recargas de páginas, mais eficiente será o algoritmo

Algoritmo Ótimo

- Seleciona uma página que não será referenciada no futuro ou aquela que demorará mais para ser utilizada novamente
- Garante uma menor paginação
- Impossível de ser implementado
  - O sistema operacional não consegue prever o futuro das aplicações
  - Não sabe quando cada página será referenciada novamente
- Possui um bom desempenho teórico

Algoritmo NUR (Não Recentemente Utilizada)

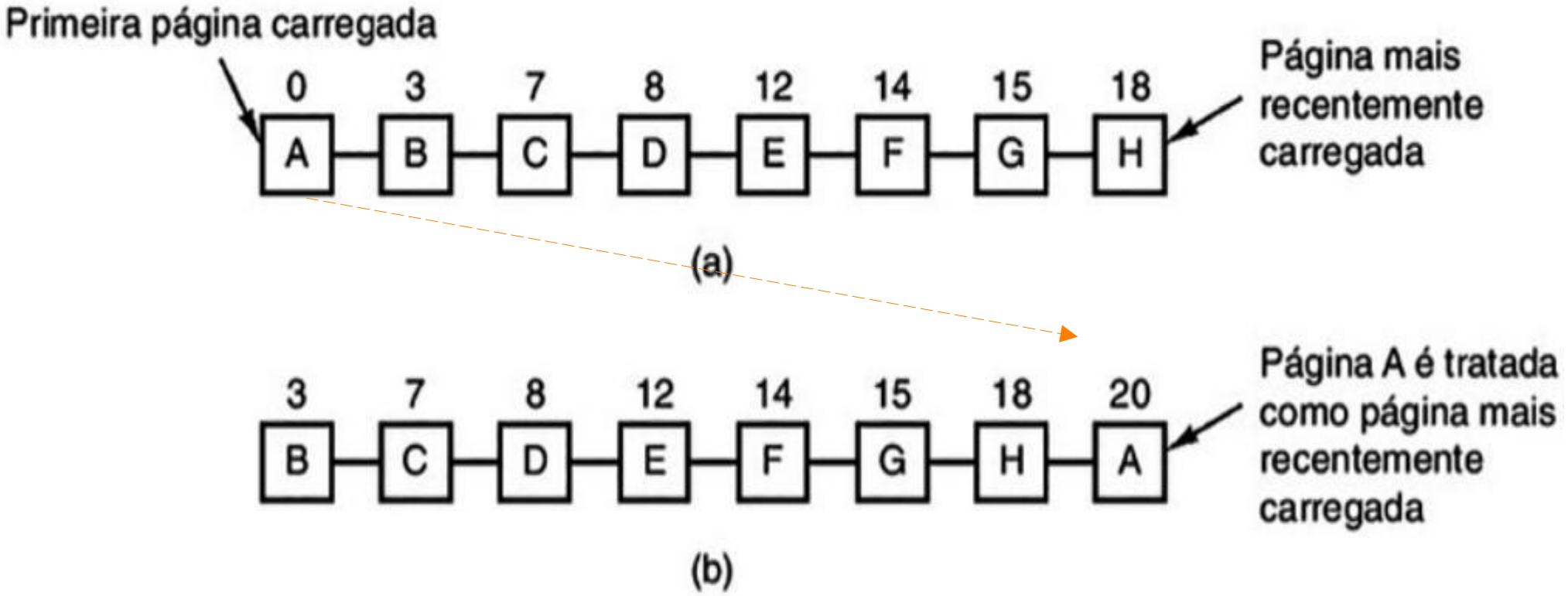
- Utiliza dois bits associados a cada página virtual
  - R (referenciada) e M (modificada)
    - Identificam quais páginas físicas estão sendo usadas
- Quando um processo inicia, bits R e M são 0 para todas as páginas
- Periodicamente, o bit R é limpo para diferenciar páginas
  - Páginas não referenciadas recentemente vs. páginas referenciadas
- Separa páginas em quatro categorias com base nos bits R e M (Classe 0: !R,!M; Classe 1: !R,M; Classe 2: R,!M; Classe 3: R,M)
- Remove aleatoriamente uma página da classe de ordem mais baixa que não esteja vazia
  - Por exemplo, pode remover uma página modificada, mas não referenciada
- É simples e de fácil implementação

Algoritmo FIFO (First In First Out)

- A primeira página utilizada será a primeira a ser escolhida para ser removida
- As páginas são inseridas em uma fila
  - As páginas mais antigas estão no início da fila
  - As páginas mais recentes estão no final da fila
- É um algoritmo de baixo custo
- Quase não é implementado
  - Não considera se a página é constantemente utilizada
  - Pode remover páginas antigas que são referenciadas frequentemente

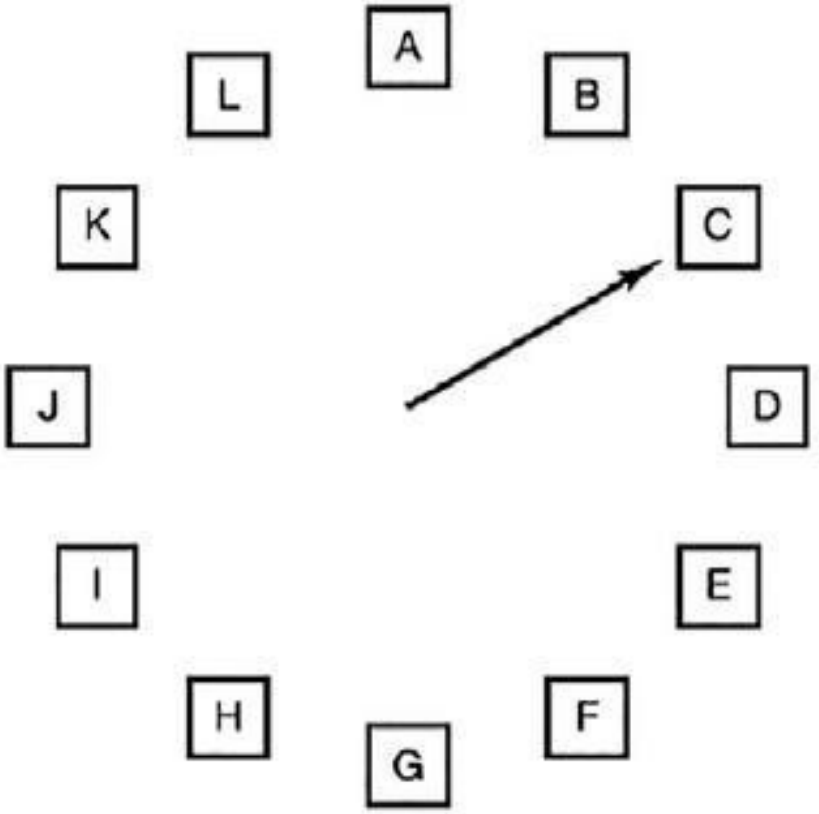
Algoritmo Segunda Chance (SC)

- Uma alteração simples no algoritmo FIFO para evitar a eliminação de uma página muito usada
- Verifica o bit de referência (R) da página mais antiga
- Se o bit R for 0, a página é substituída
- Se o bit R for 1, o bit R é colocado em 0, e essa página é inserida no final da lista
  - É tratada como se tivesse acabado de ser carregada na memória
- O algoritmo continua a percorrer a lista a partir da próxima página
- É o algoritmo FIFO melhorado


Primeira página carregada
0
3
7
8
12
14
15
18
A
B
C
D
E
F
G
H
Página mais recentemente carregada
(a)
3
7
8
12
14
15
18
20
B
C
D
E
F
G
H
A
Página A é tratada como página mais recentemente carregada
(b)

Algoritmo Relógio

- Mantém todas as páginas em uma lista circular em forma de relógio
- Um ponteiro aponta para a "cabeça" da lista (página mais antiga)
- Quando ocorre uma falta de página, o ponteiro verifica o bit R da página atual (apontada)
- Se o bit R for 0, a página é retirada e o ponteiro avança
- Se o bit R for 1, o bit R é colocado em 0 e o ponteiro avança para a próxima página mais antiga
- A diferença para o SC está na implementação (fila vs. lista circular)
- É um algoritmo realista


A
B
C
D
E
F
G
H
I
J
K
L

Algoritmo MRU (Mais Recentemente Utilizada)

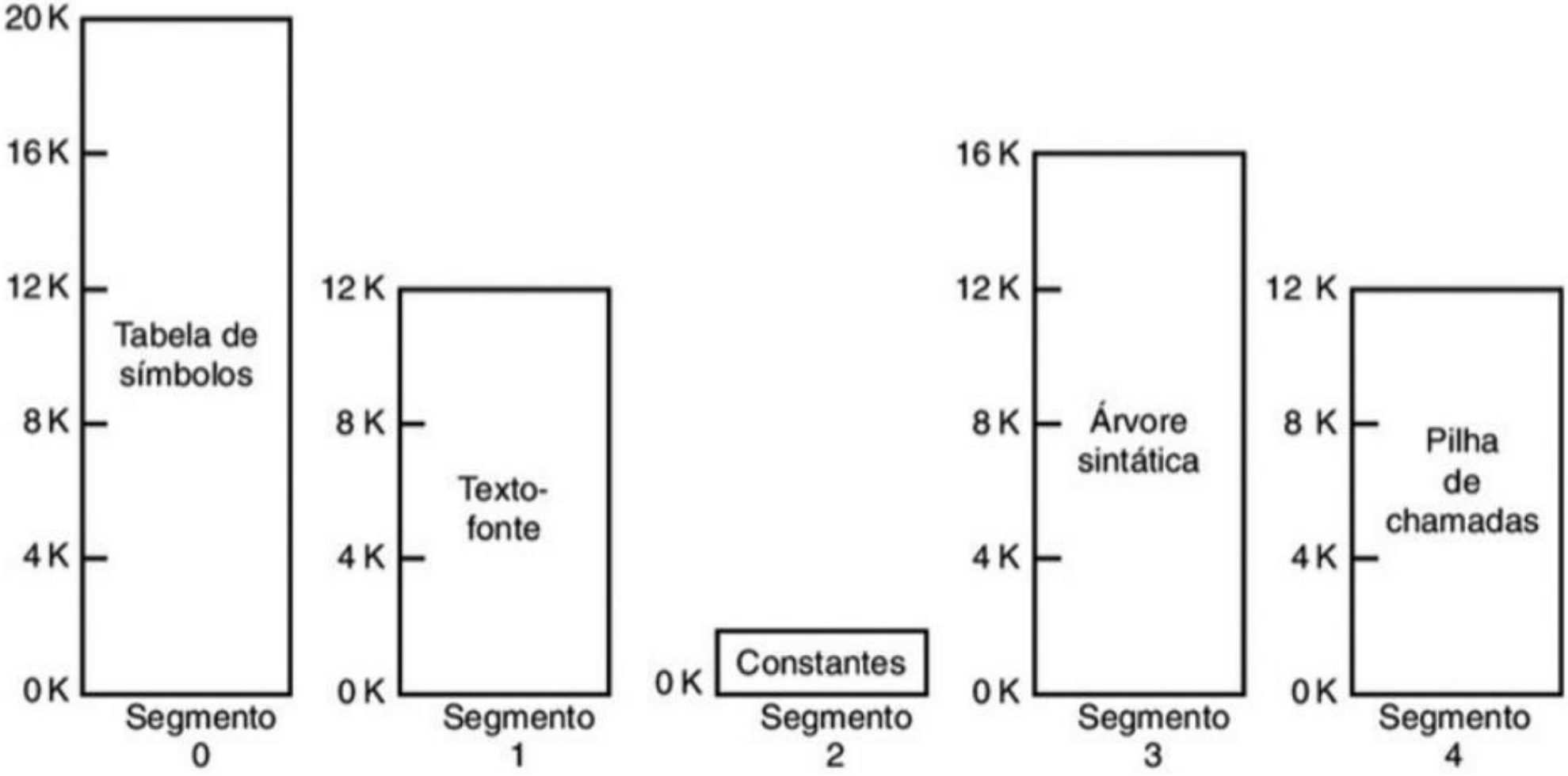
- Baseado na observação de que páginas referenciadas intensamente nas últimas instruções provavelmente serão usadas novamente
  - Páginas não referenciadas recentemente podem não ser usadas na próxima instrução
- Aproxima-se do desempenho do algoritmo ótimo
- Possui uma implementação onerosa
  - Mantém uma lista encadeada na memória com páginas mais usadas no início
  - Exige atualização a cada referência de memória
- É considerado um bom algoritmo, mas difícil de implementar

Segmentação: Conceito e Características

- A memória virtual é unidimensional na paginação
  - Inicia na posição 0 e vai até um endereço máximo
- A segmentação fornece vários espaços de endereçamento independentes, chamados segmentos
- Um segmento é uma unidade lógica, de conhecimento do programador
  - Pode ser um vetor, uma pilha, etc.
- Cada segmento tem um tamanho dinâmico e independente
  - Pode aumentar ou diminuir durante a execução
- Endereços são compostos pelo número do segmento e um deslocamento dentro do segmento
- Segmentos podem ter diferentes proteções (execução, leitura, leitura/escrita)

Exemplo:
Registradores x86:
CS (Code Segment)
CS (Code Segment)
SS (Stack Segment)
ES (Extra Segment)
FS e GS (usados pelo sistema operacional)

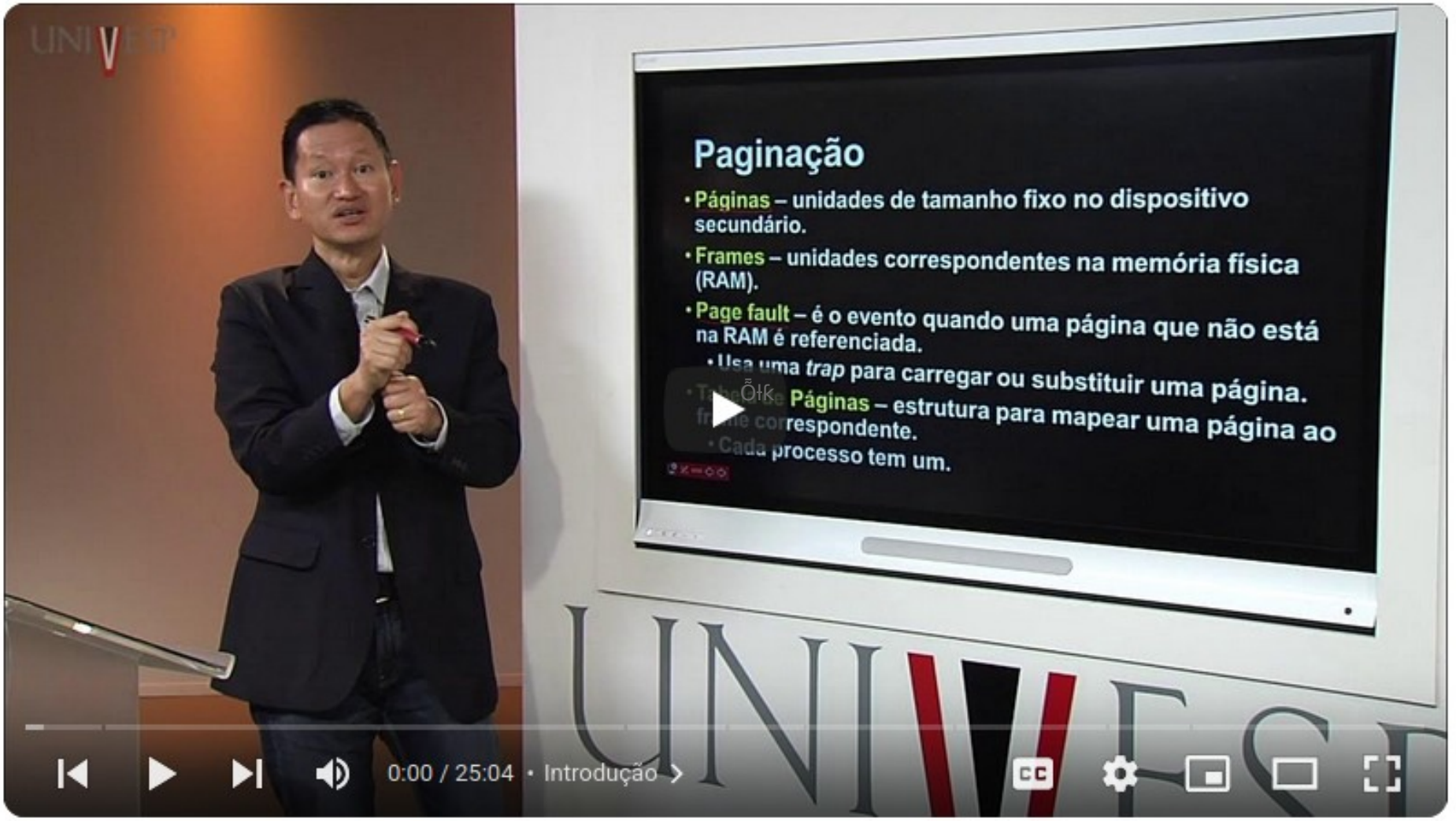
Figura 4.13 | Memória segmentada



Segmentação vs. Paginação

Quadro 4.1 | Critérios de segmentação e paginação

	Segmentação	Paginação
O programador sabe do seu uso?	Sim	Não
Endereços lineares existentes?	Vários	Um
O tamanho dos dados podem exceder a quantidade de memória física?	Sim	Sim
Facilidade de compartilhamento entre usuários?	Sim	Não



Sistemas Operacionais - Aula 18 - Técnicas de Memória Virtual