

Abstrações de Memória: Segmentação e Segmentação com Paginação

Prof. Dr. Márcio Castro
marcio.castro@ufsc.br



1

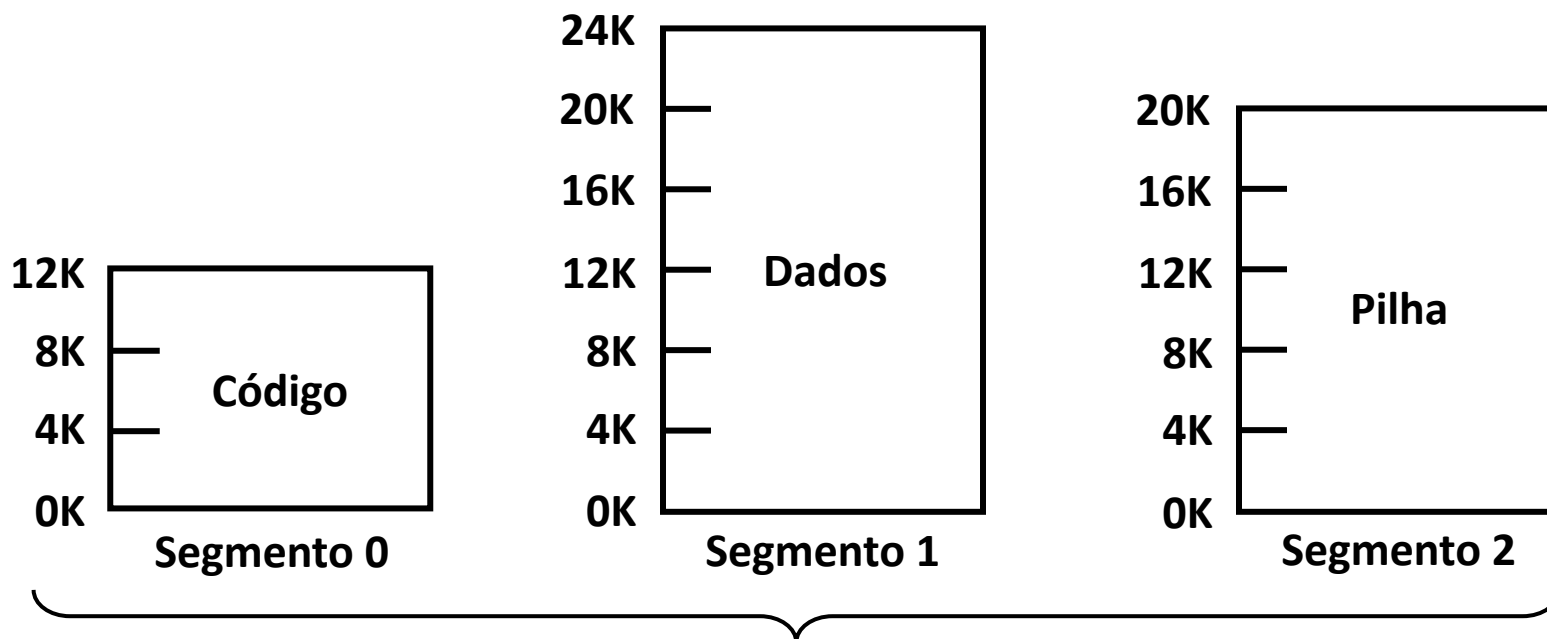
Introdução

Segmentação

- Na paginação a memória virtual é **unidimensional**
 - **Faixa de endereços virtuais:** 0 até $2^n - 1$, onde n é o número de bits utilizados para representar endereços virtuais
- Permitir mais de um espaço de endereçamento por processo traz **flexibilidade**
 - **Exemplo:** separar regiões de código, dados e pilha de processos

Segmentação

- O espaço de endereçamento do processo é dividido em **segmentos**
- Segmentos são blocos de endereços **contíguos** completamente **independentes** e de **tamanhos variáveis**



Exemplo de possíveis segmentos do espaço de endereçamento de um processo

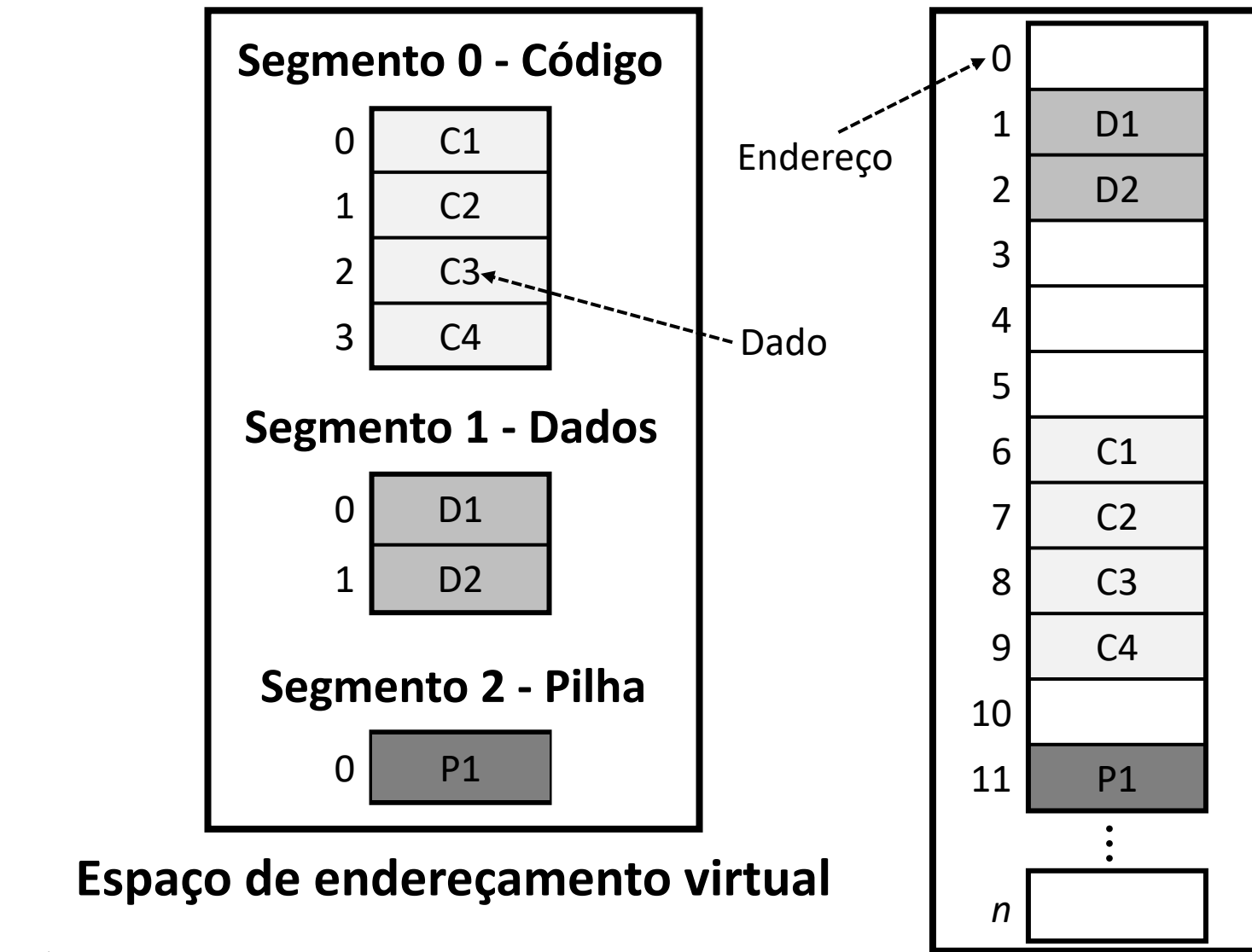
- **Duas formas de implementar o conceito de segmentação**
 - Segmentação pura
 - Segmentação com paginação

2 Segmentação pura

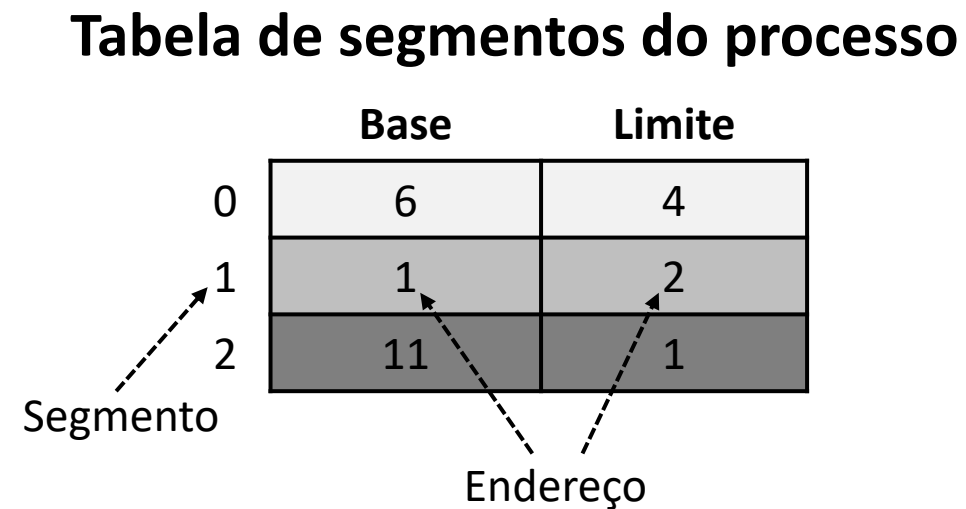
Segmentação pura

- Uma **tabela de segmentos** relaciona a memória virtual do processo com a memória física do computador
- **Cada entrada da tabela de segmentos contém:**
 - Número do segmento
 - Endereço inicial do segmento (**base**)
 - Tamanho do segmento (**limite**)

Segmentação pura



Espaço de endereçamento virtual

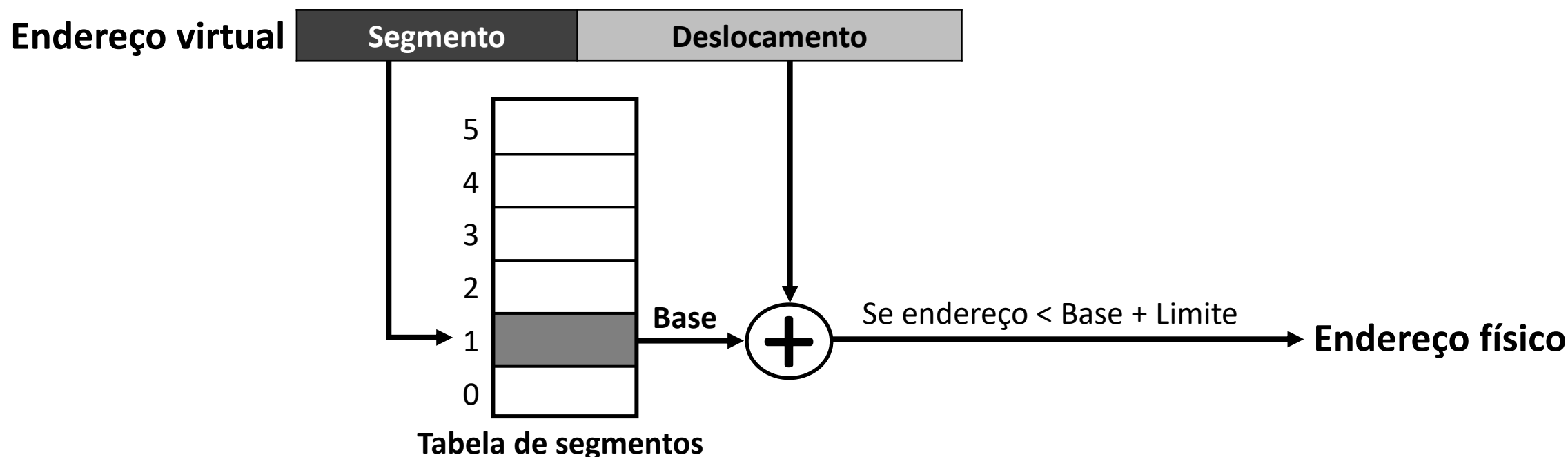


Memória física

Segmentação pura

- Endereços virtuais são divididos em 2 partes:
 - Número do segmento
 - Deslocamento dentro do segmento

Tradução de um endereço virtual em um endereço físico



Segmentação pura

- **A tabela de segmentos é armazenada na memória RAM**
 - Uma tabela de segmentos **para cada processo**
- **Registradores guardam as informações sobre o endereço da tabela**
 - Registradores são **atualizados** quando há **troca de contexto** entre processos

Segmentação pura

▪ Vantagens da segmentação

- Segmentos permitem acomodar estruturas de dados que **crecem/diminuem de tamanho**
- Mais fácil fazer **ligação de métodos/funções** compilados separadamente
- Livre de **fragmentação interna**
- Facilidade de **compartilhamento** de segmentos

- **Desvantagens**

- **Fragmentação externa:** pode haver espaço desperdiçado entre segmentos
- Gerenciamento de segmentos grandes é problemático

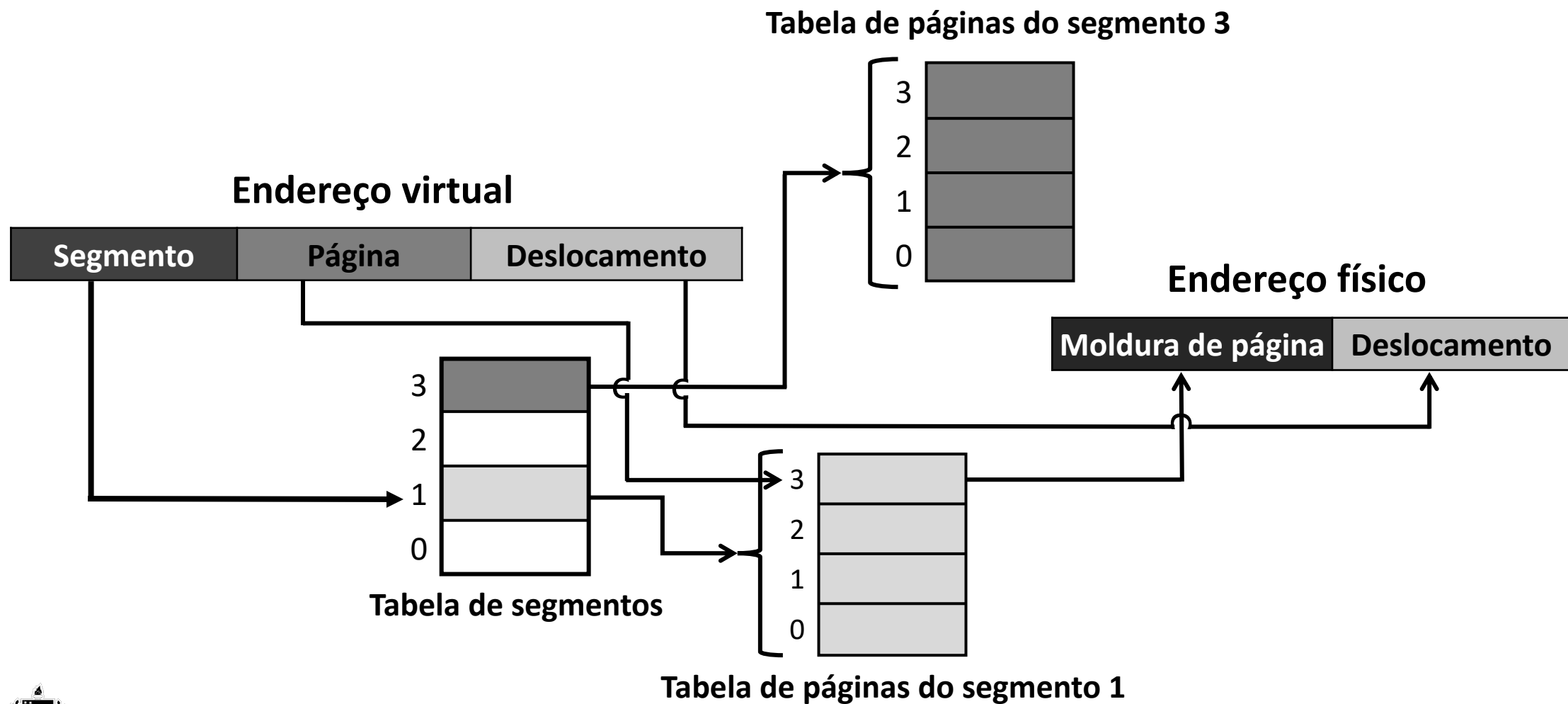
3

Segmentação com paginação

Segmentação com paginação

- É uma técnica que combina a **paginação** com a **segmentação pura**
- O espaço de endereçamento de **cada segmento** é dividido em **páginas** (**segmentos paginados**)
- Resolve o problema de gerenciamento de segmentos muito grandes

Segmentação com paginação



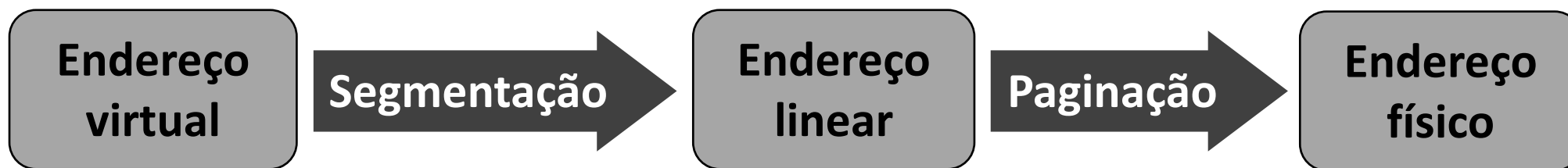
4

Segmentação com paginação no Intel x86-32

Segmentação com paginação no Intel x86-32

- A **segmentação pura** foi introduzida no Intel 8086 em 1978
 - Modelo conhecido por *real mode*
- Posteriormente, o suporte à **memória virtual (com paginação)** foi introduzido no Intel 80386 em 1985
 - Modelo conhecido por *protected mode*

Tradução de um endereço virtual em endereço físico (*protected mode*)



Segmentação com paginação no Intel x86-32

O sistema de memória virtual do x86 é baseado em duas tabelas

- ***Global Descriptor Table (GDT)***

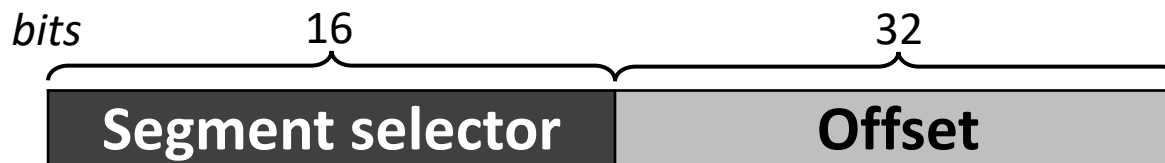
- Descreve segmentos de sistema (incluindo os do SO)
- Uma única GDT no sistema, compartilhada por todos os programas
- Registrador `gdtr`

- ***Local Descriptor Table (LDT)***

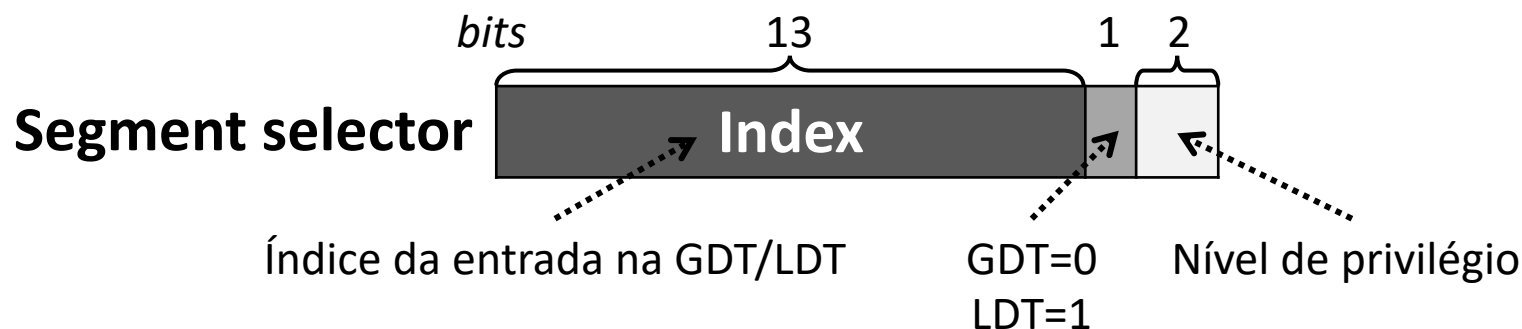
- Descreve segmentos dos processos
- Uma LDT para cada processo
- Registrador `ldtr`

Segmentação com paginação no Intel x86-32

- O endereço virtual é composto por um seletor de segmento (*segment selector*) e um deslocamento (*offset*)

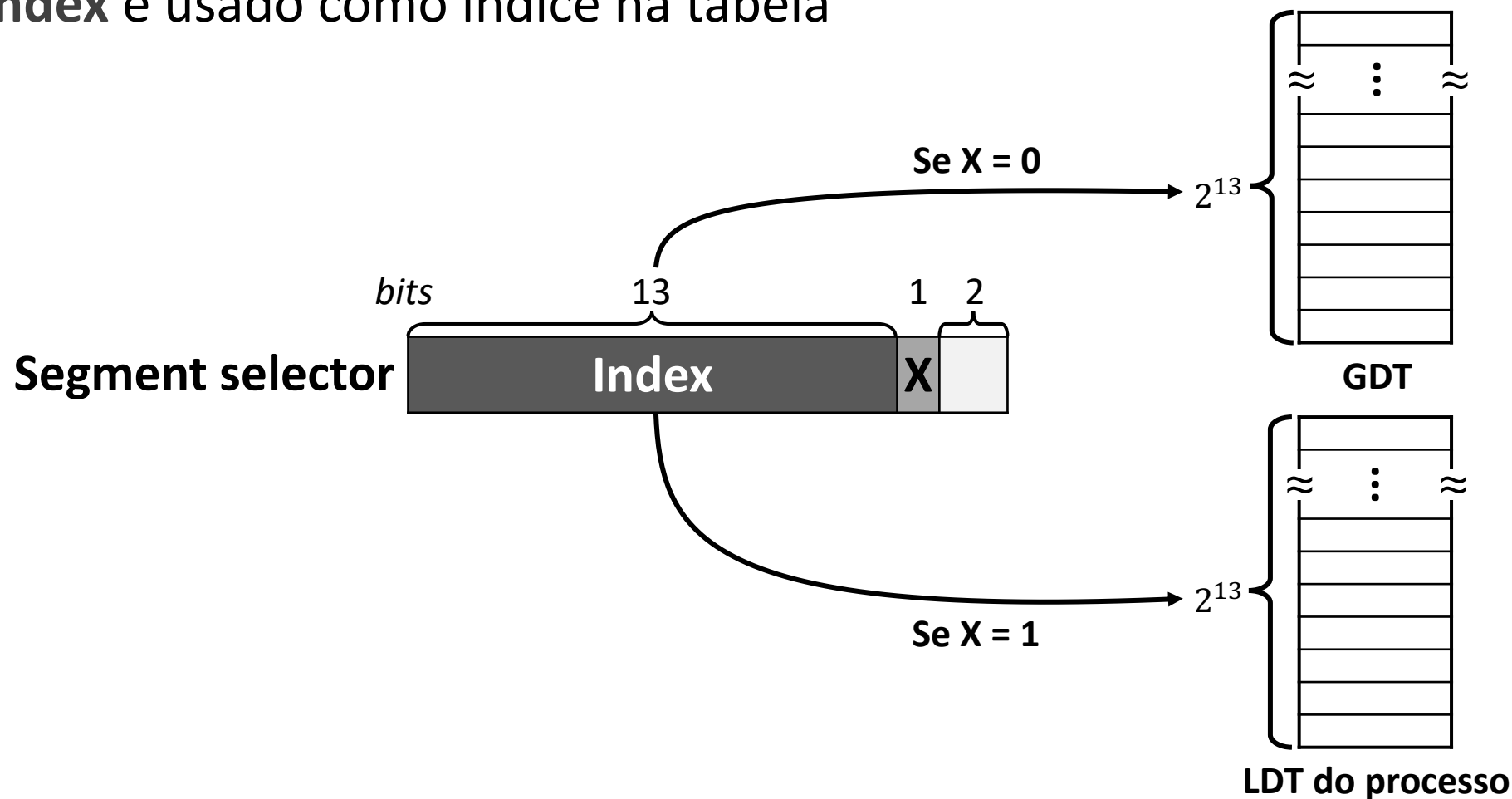


- *Selectors* são carregados em registradores de segmento
 - *code* (**cs**), *data* (**ds**), *stack* (**ss**) e extras (**es**, **fs** e **gs**)



Segmentação com paginação no Intel x86-32

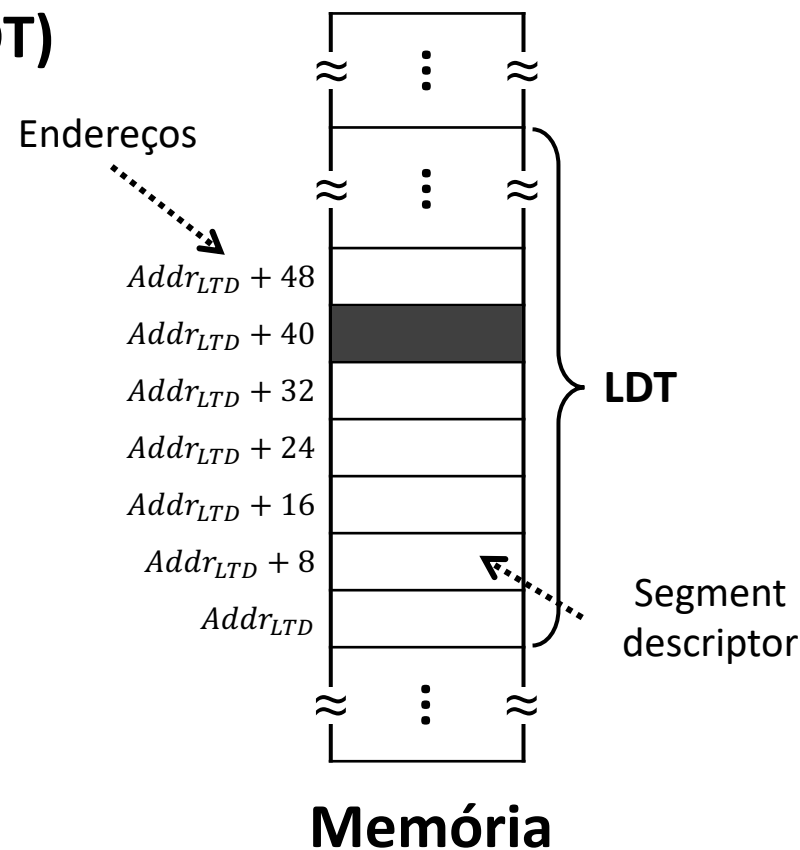
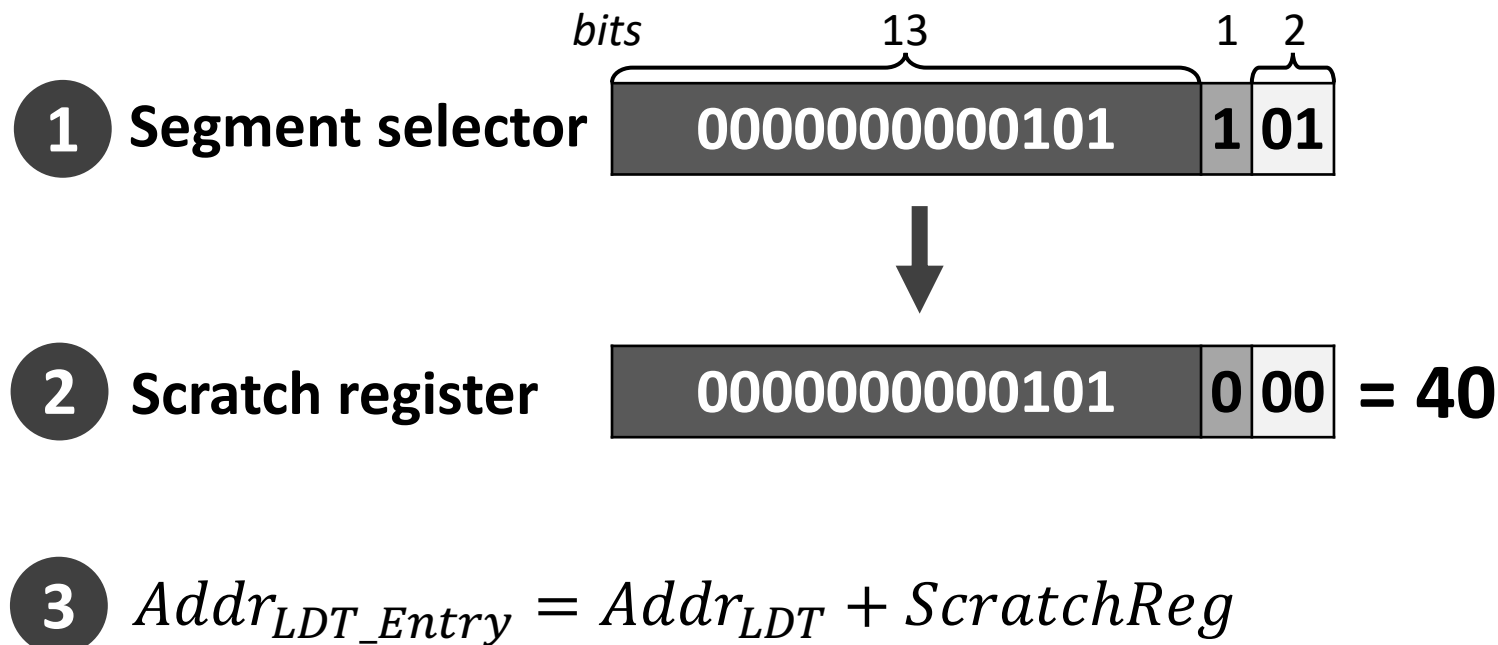
- O bit GDT/LDT indica qual tabela deve ser acessada
 - **Index** é usado como índice na tabela



Segmentação com paginação no Intel x86-32

- Cada entrada da GDT/LDT armazena um descritor de segmento (*segment descriptor*)

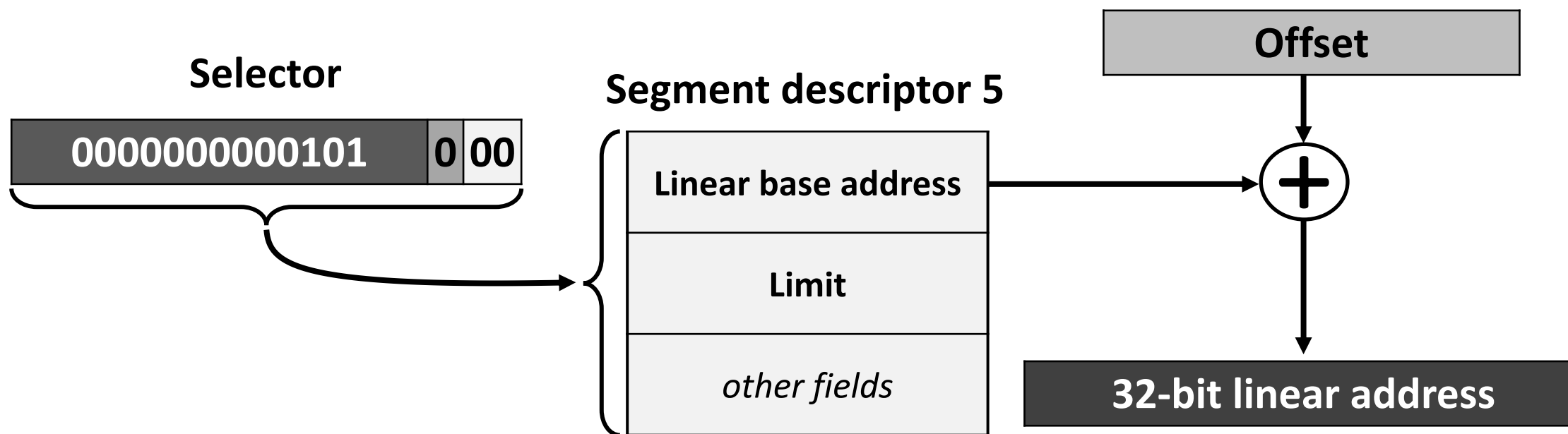
Passos para encontrar o descritor de segmento (exemplo LDT)



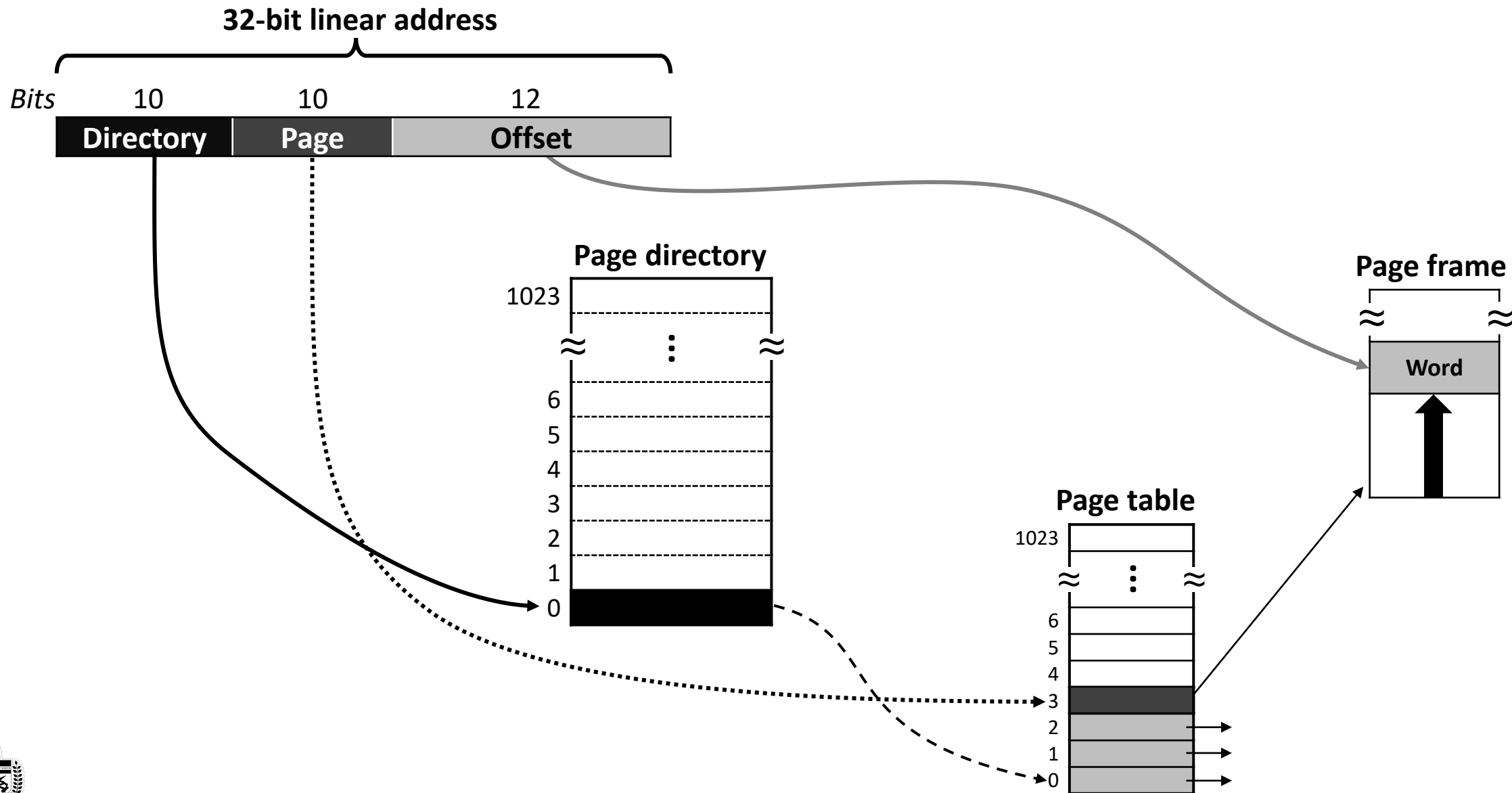
$$Addr_{LDT_Entry} = Addr_{LDT} + 40 \longrightarrow \text{Entrada 5 da LDT}$$

Segmentação com paginação no Intel x86-32

- O **descritor de segmento** contém endereços de **base** e **limite** do segmento dentro de um **espaço de endereçamento linear** de 32 bits



Segmentação com paginação no Intel x86-32





Obrigado pela atenção!



Dúvidas? Entre em contato:

- marcio.castro@ufsc.br
- www.marciocastro.com

