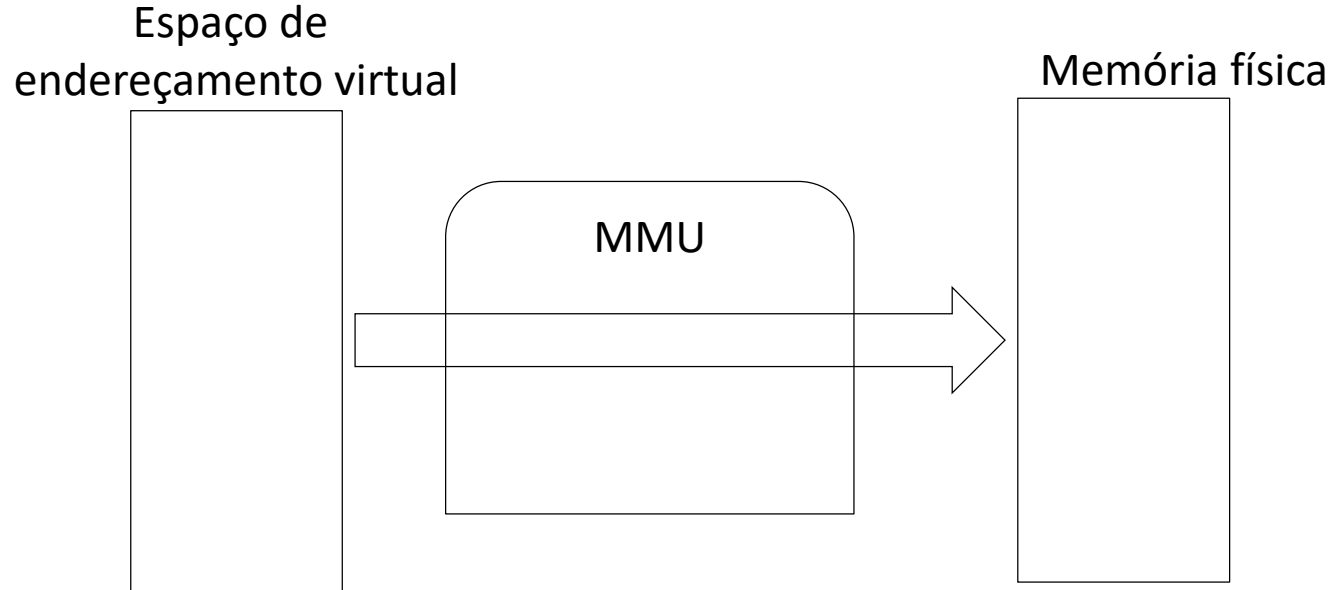


# Arquitecturas de MMU (Memory Management Unit)

---

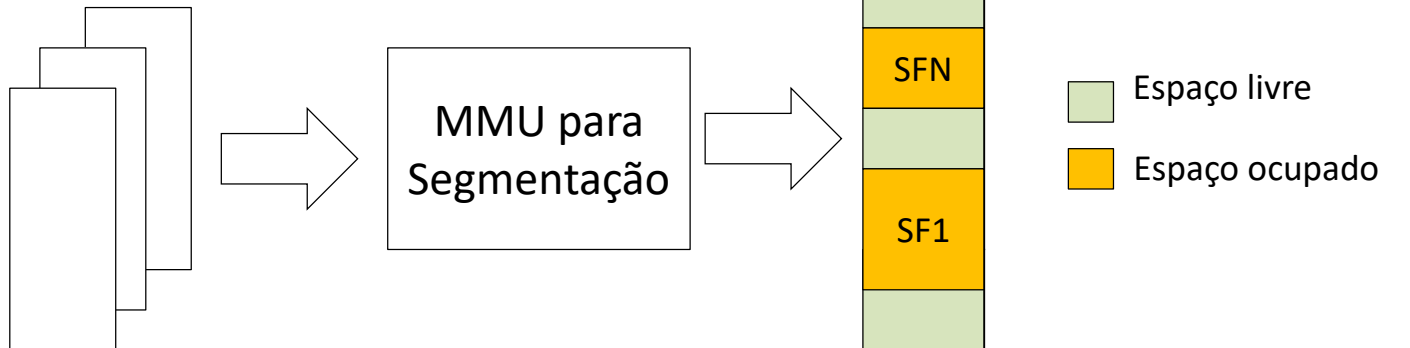
- Suporte hardware à construção de espaços de endereçamento virtual
  - Arquitectura Segmentada
  - Arquitectura Paginada
  - Arquitectura Segmentada/Paginada



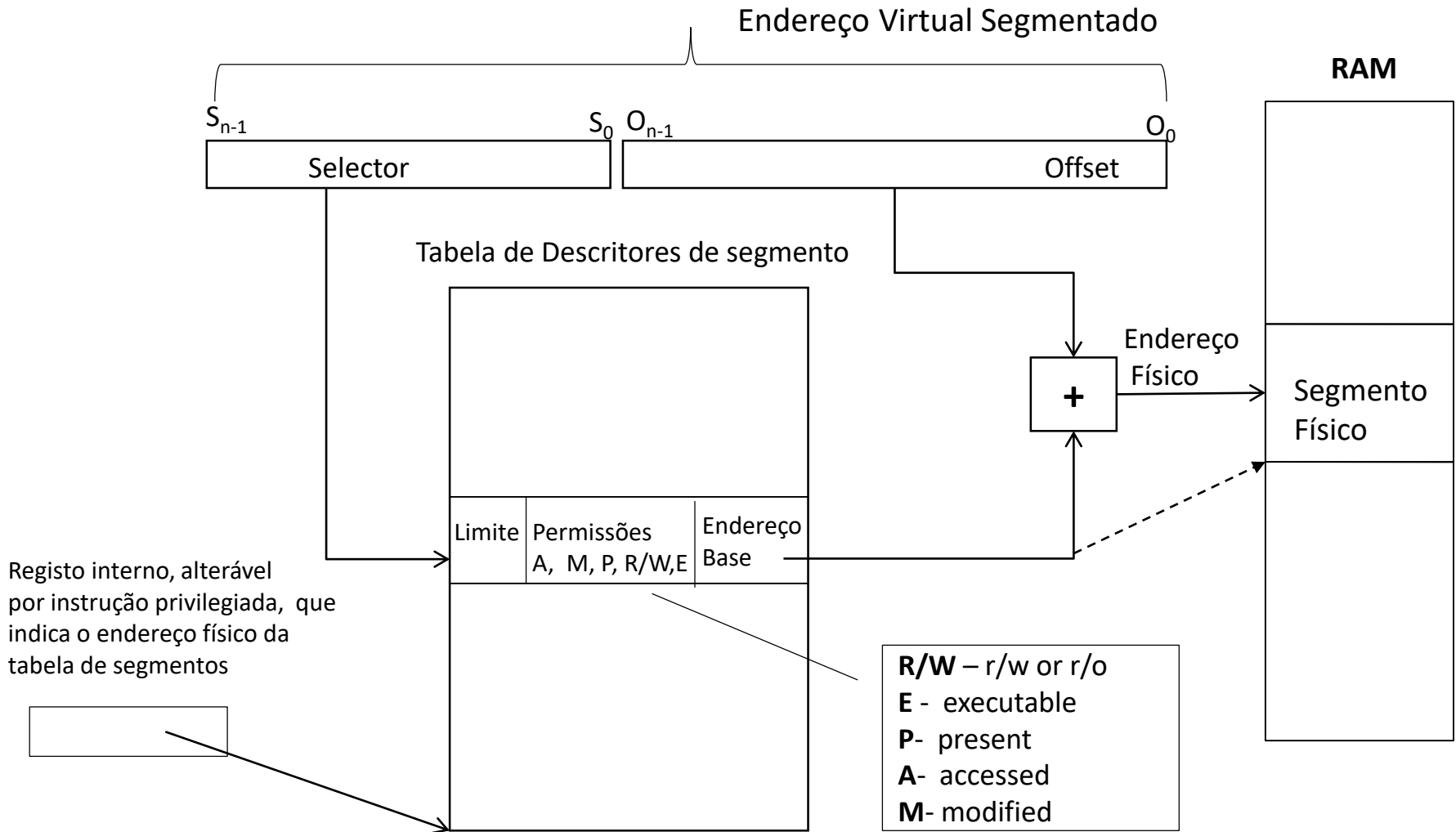
# Visão aplicacional de arquitectura segmentada

- Na arquitectura segmentada os segmentos são visíveis às aplicações e resultam de um modelo de programação orientada à organização do código e dados em módulos, ocupando segmentos distintos.
  - Um endereço virtual é especificado por duas componentes. O `id` do segmento ((tipicamente o índice para uma tabela de descritores de segmento) e o `offset` dentro do segmento

Segmentos aplicacionais dentro de um processo. O espaço virtual é segmentado



# Arquitectura Segmentada (Segmentação)



# Características da Segmentação

---

- **Vantagens**

- Mapeamento simples o que reduz a complexidade do *hardware*. A associação de uma cache de segmentos em uso evita o acesso constante à tabela de descritores.
- Facilita a recolocação de código e dados.
- Facilita a criação de regiões de memória partilhada entre processos.

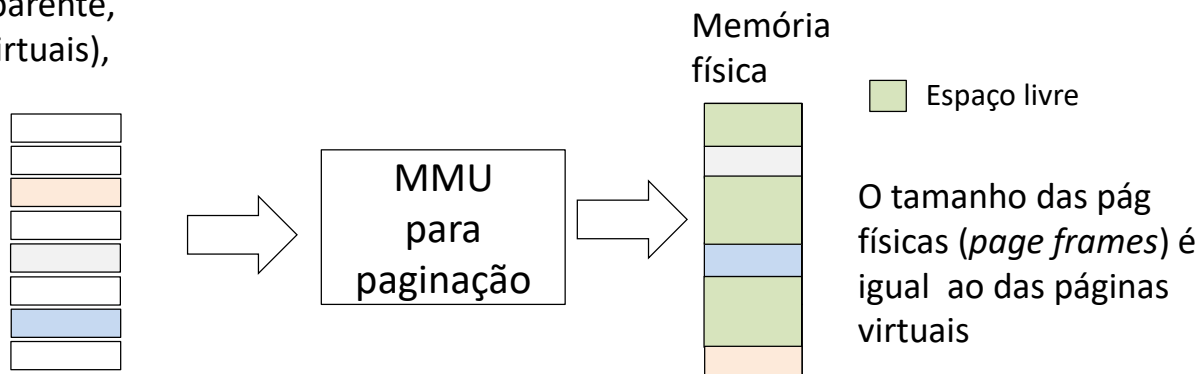
- **Desvantagens**

- Modelo de ponteiros complexo.
- Gestão complexa da memória disponível. O facto dos blocos a gerir terem dimensões variáveis gera problemas já vistos na implementação de *heaps*, nomeadamente a fragmentação externa.
- Implementação de memória virtual pouco eficiente, devido ao tempo dispendido em escritas/leituras do disco para segmentos grandes (> 64KB)

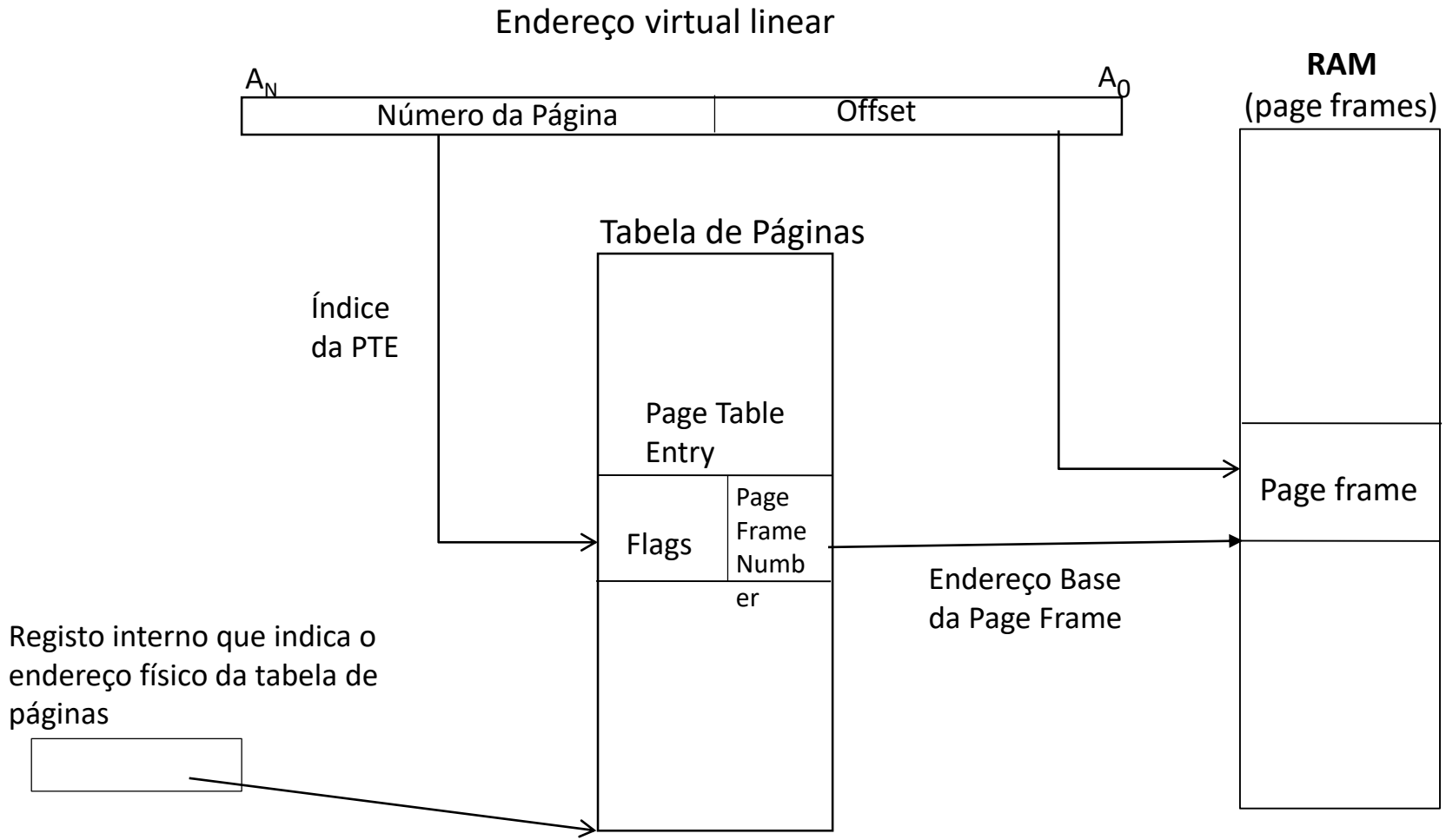
# Visão aplicacional de arquitectura paginada

- Na arquitectura paginada as páginas são transparentes às aplicações:
  - O espaço de endereçamento que as aplicações veem é sempre linear, tal como o espaço de endereçamento físico.
  - A organização lógica nada tem a ver com a organização física (ex: uma instrução ou uma variável global podem começar no final de uma página e terminar na página seguinte ocupando endereços físicos não adjacentes).
  - Páginas lógicas e páginas físicas têm sempre a mesma dimensão.

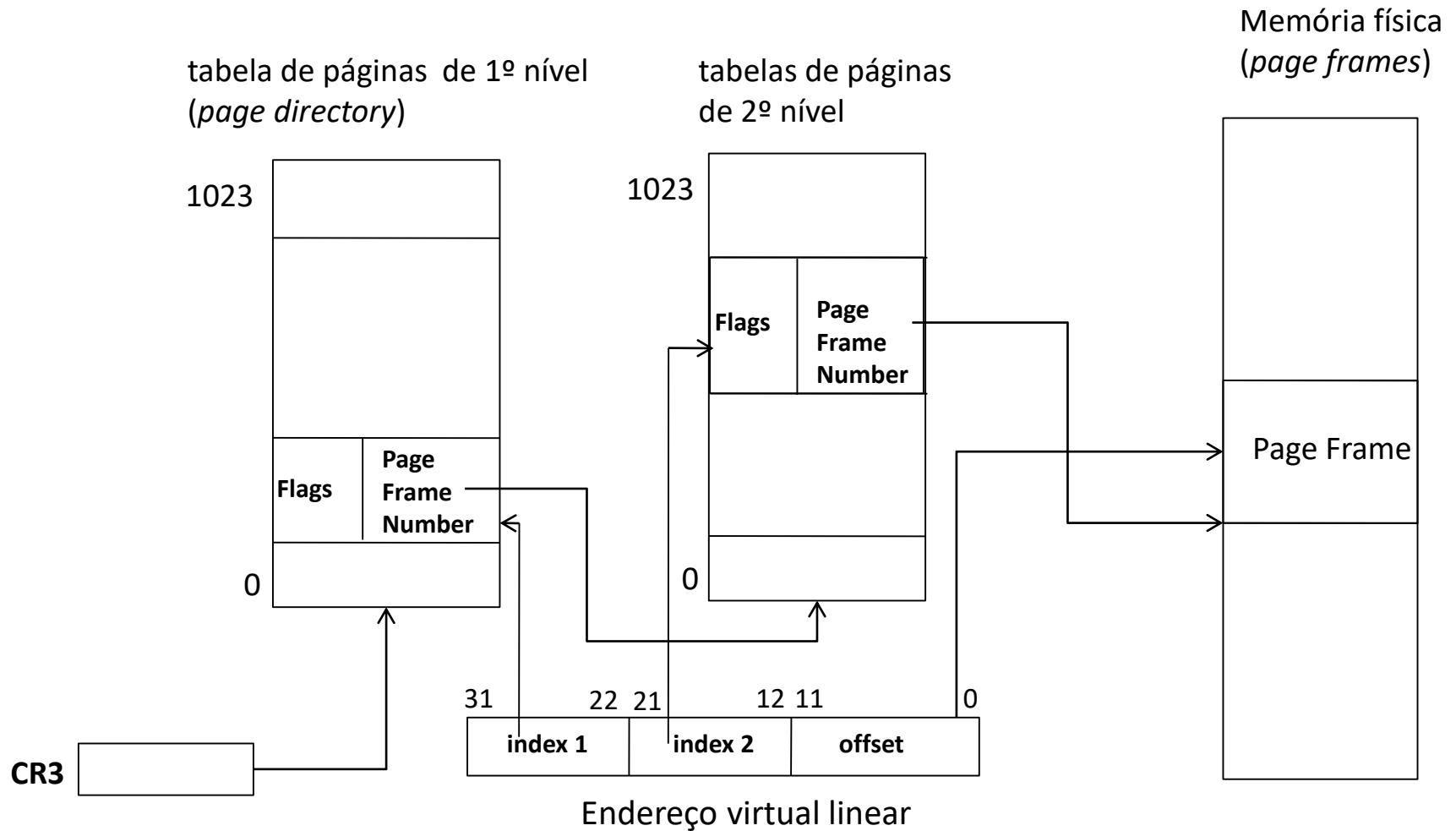
O espaço virtual é linear, dividido, de forma transparente, em páginas lógicas (ou virtuais), de dimensão constante



# Arquitectura Paginada (Paginação)

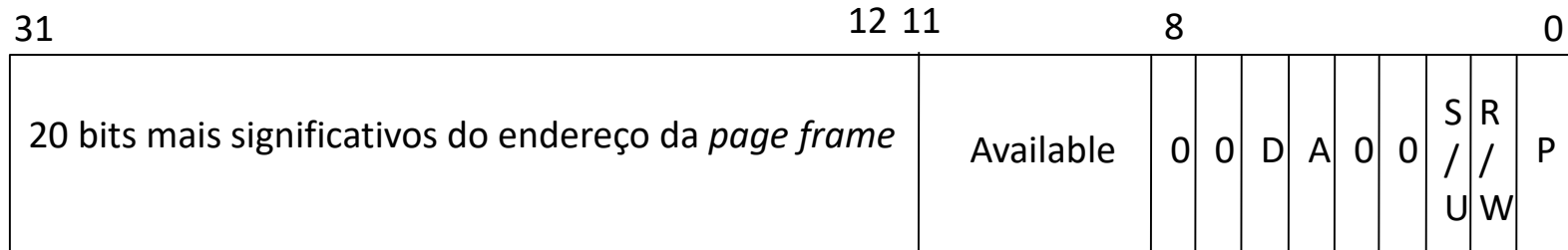


# Arquitetura Paginada (exemplo para x86)



# Entrada de tabela de páginas – PTE (Page Table Entry) (exemplo para i386)

---



- P** – Present (indica que a página está associada a uma page frame)
- R/W** – Read/Write (indica se a página pode ser modificada)
- S/U** – Supervisor/User (indique que a página pode ser usada em user mode)
- 0** – bits reservados (já utilizados em versões mais recentes do CPU)

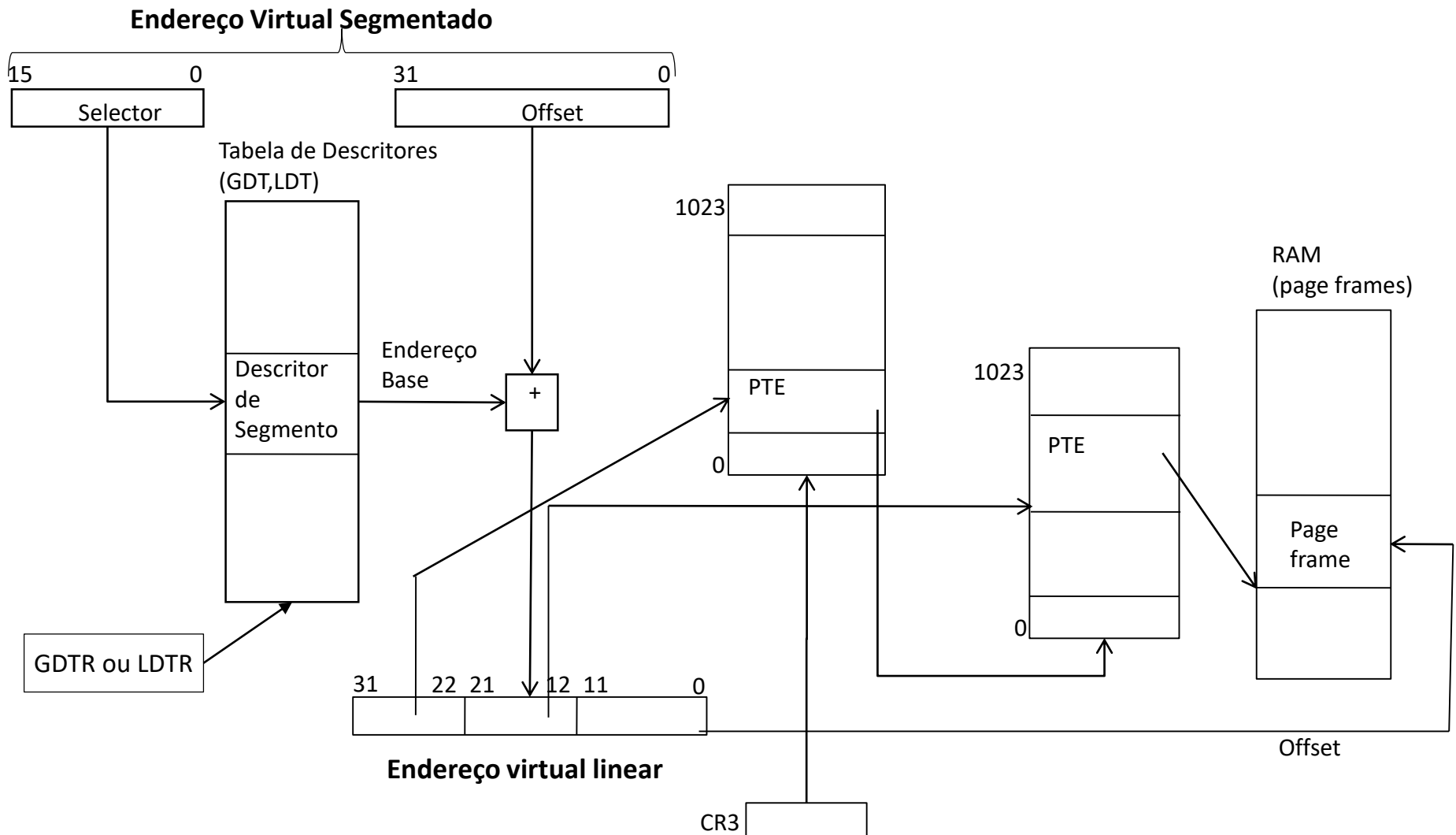
Bits alterados automaticamente pela MMU:

- A** – Accessed (indica que a página foi lida)
- D** – Dirty (indica que a página foi modificada)

**Available** – Disponíveis para uso pelo Sistema Operativo



# Arquitetura Segmentada/Paginada (exemplo para x86)

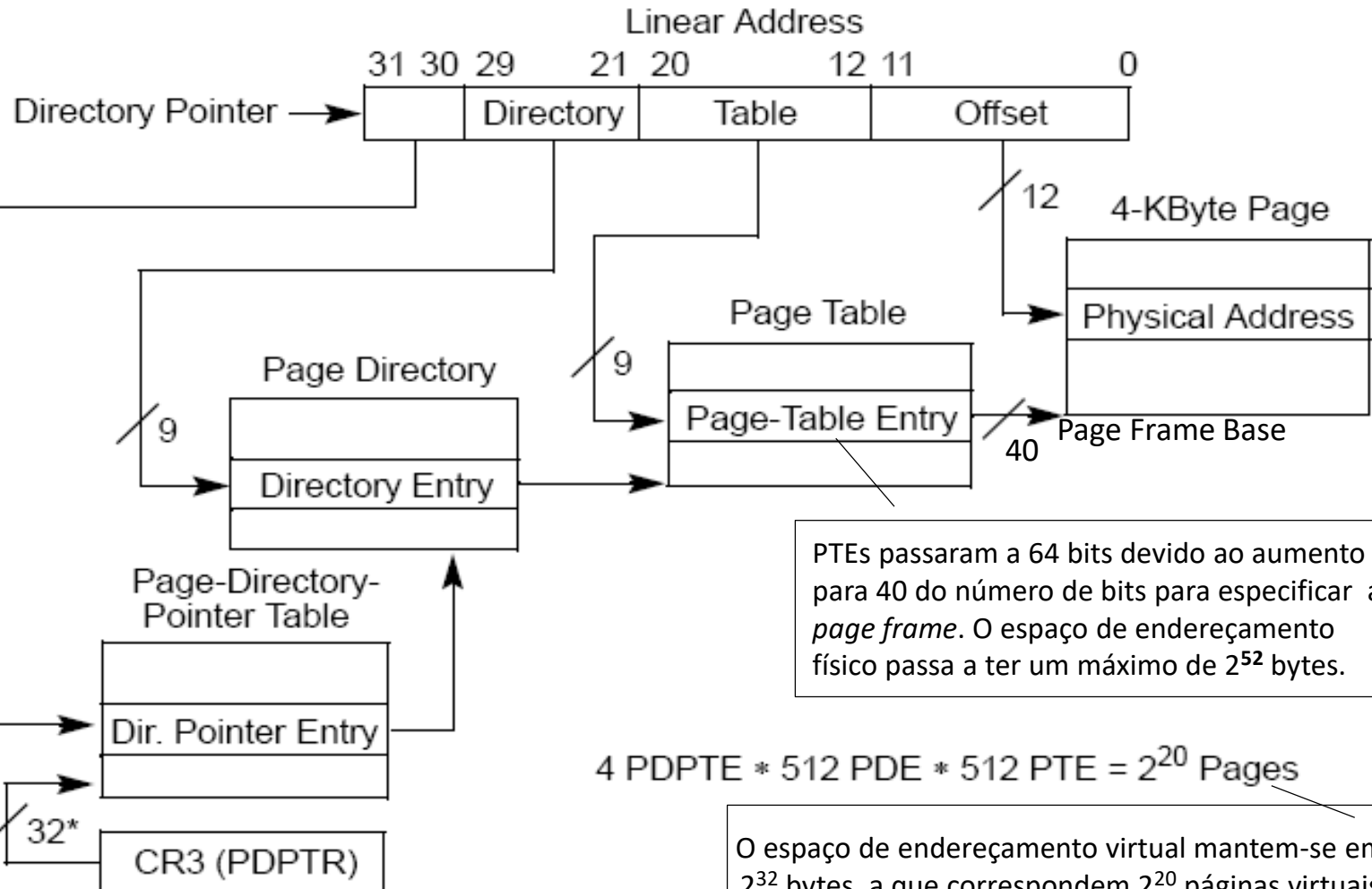


# Exercício

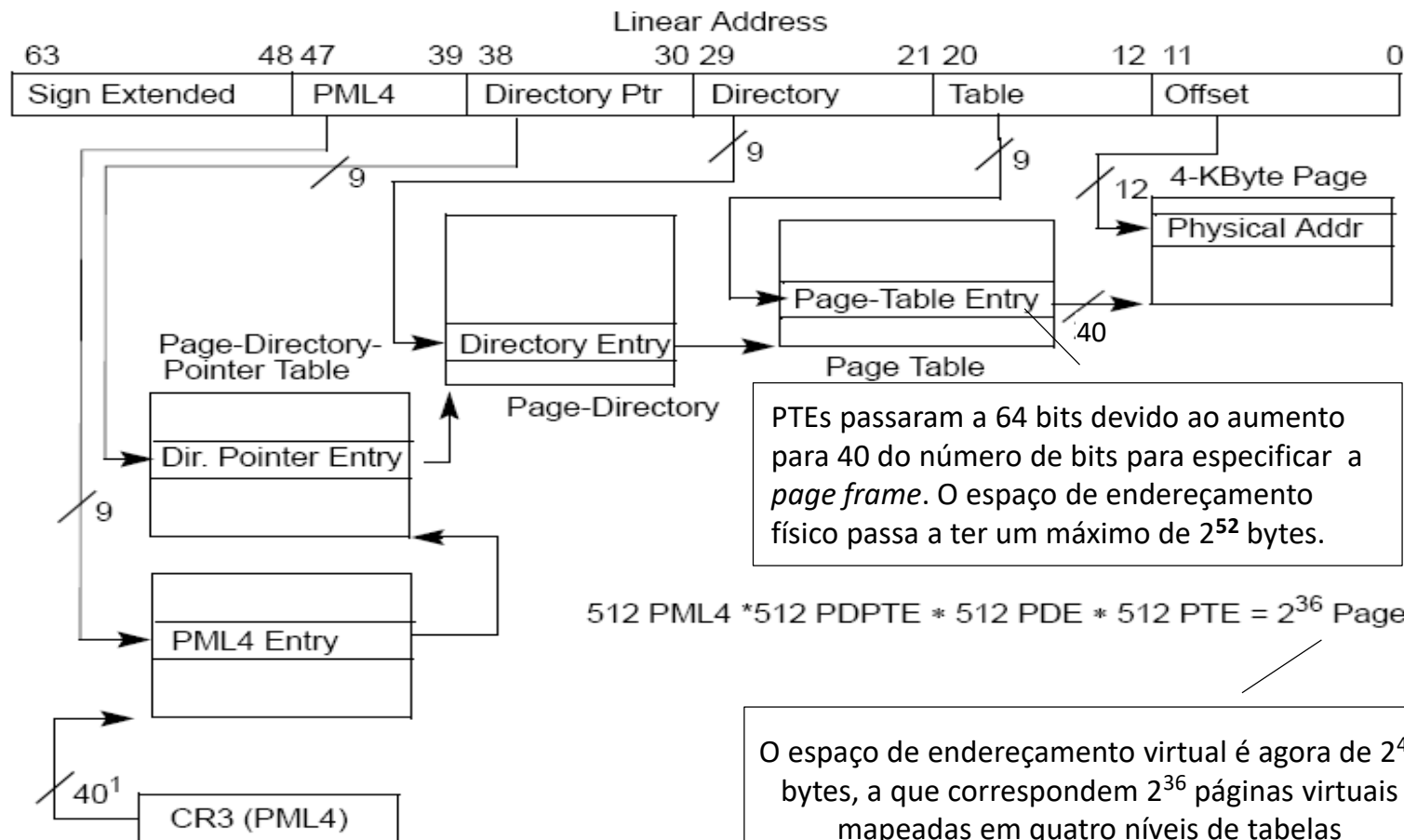
---

- Considere um processador com endereços virtuais de 36 bits que utiliza uma estrutura de paginação de dois níveis. Os endereços virtuais são divididos em: 11 bits para a diretoria de páginas(tabela de 1º nível) e 12 bits para as tabelas de 2º nível.  
Sabendo que as tabelas de páginas ocupam sempre uma página, indique:
  - qual a dimensão das páginas e de cada PTE?
  - quantas páginas existem no espaço de endereçamento virtual?
  - Sabendo que existem 21 bits na PTE para especificar a page frame qual o espaço de endereçamento físico?

## Variantes em CPU's modernos da família x86 - **Physical Address Extensions (PAE)**



# Paginação na arquitectura x64

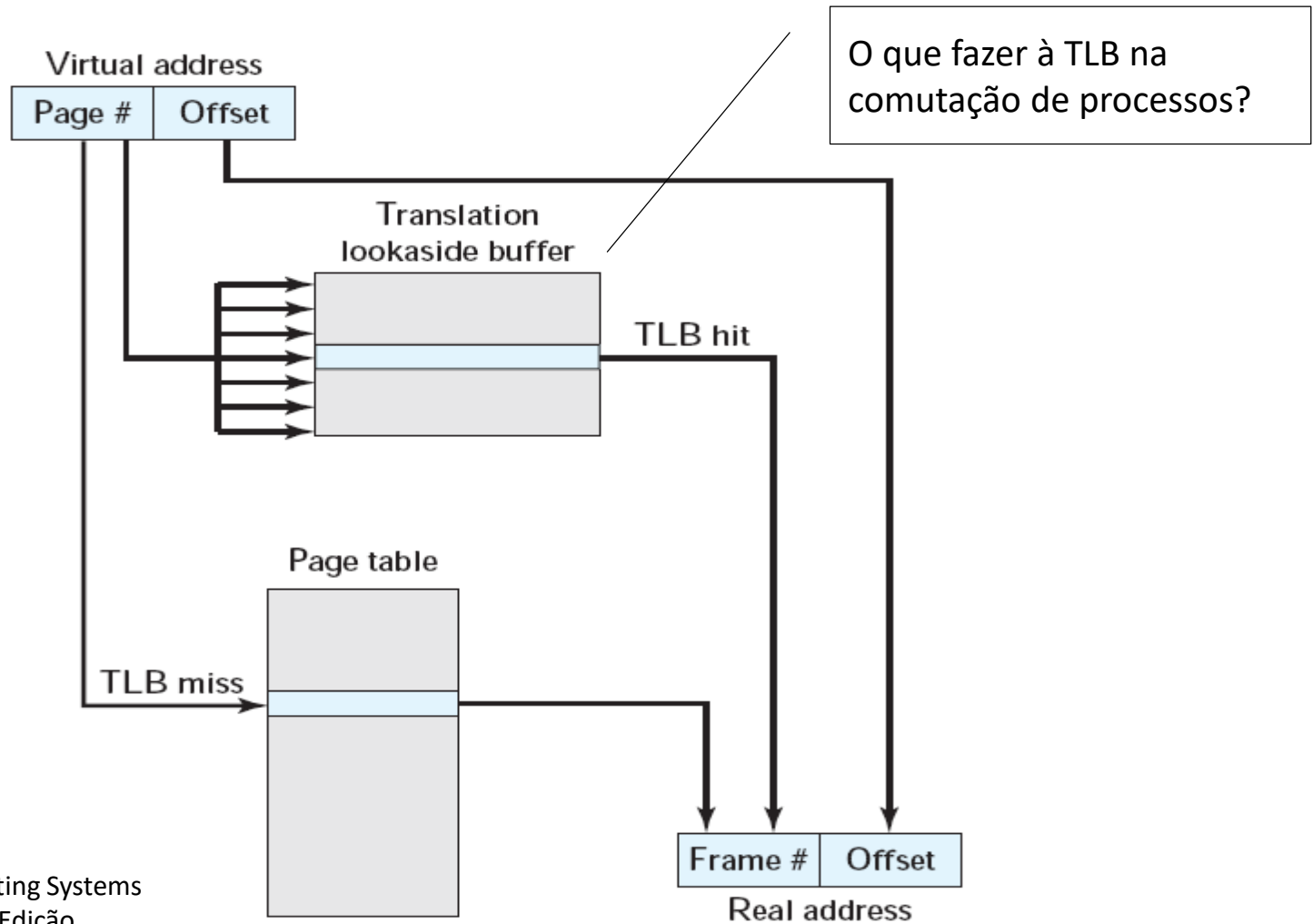


## NOTES:

1. 40 bits aligned onto a 4-KByte boundary

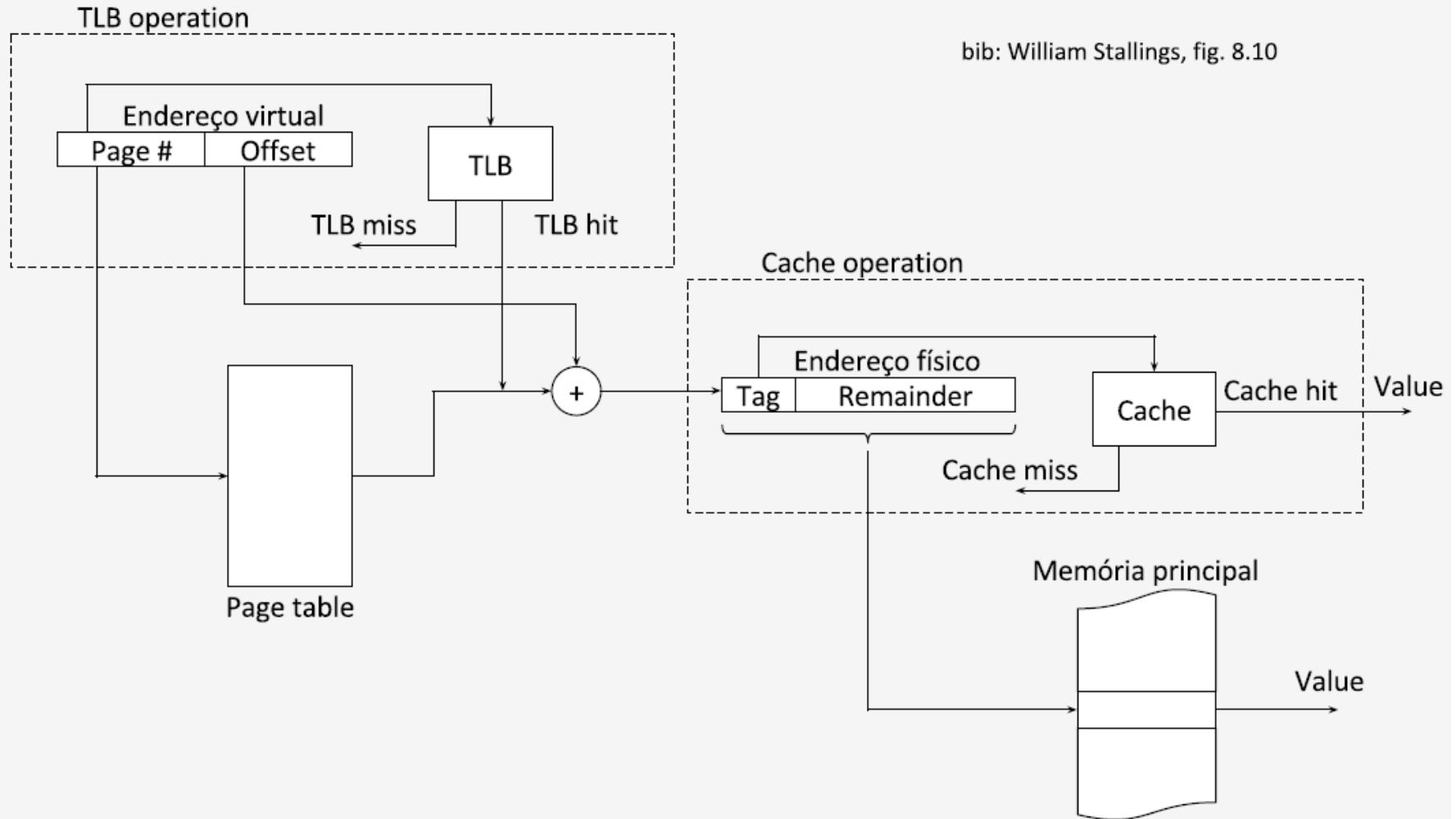
# Paginação – A necessidade da cache TLB

(Translation Lookaside Buffer)



Retirado do livro Operating Systems  
de William Stallings, 6ª Edição

# Cache (de PTE's e de dados) – visão global



# Objectivos de aprendizagem

---

- Compreender e enunciar as principais características das arquitecturas segmentada e paginada.
- Identificar vantagens/desvantagens de cada um dos modelos.
- Nas arquitecturas paginadas:
  - explicar a necessidade de existência de arquitecturas multi-nível
  - explicar a necessidade da cache TLB
  - sintetizar características da arquitectura (dimensões de PTE, de página (lógica e física), de espaços de endereçamento virtual e memória física, número de níveis, etc.), a partir de informações parcelares sobre a arquitectura.
  - Enunciar e compreender as *flags* típicas presentes num PTE
  - justificar as dimensões típicas das páginas nas arquitecturas actuais.

# Bibliografia

---

- Stallings, Operating Systems, 7ª Ed.
  - Cap. 8, Memory Management Strategies
- Solomon, Russinovich- Windows Internal, 5ª Ed
  - Cap. 9, Memory Management
    - Secção 9.6, Address Translation