

# **Aula 5**

## **Memória Virtual**

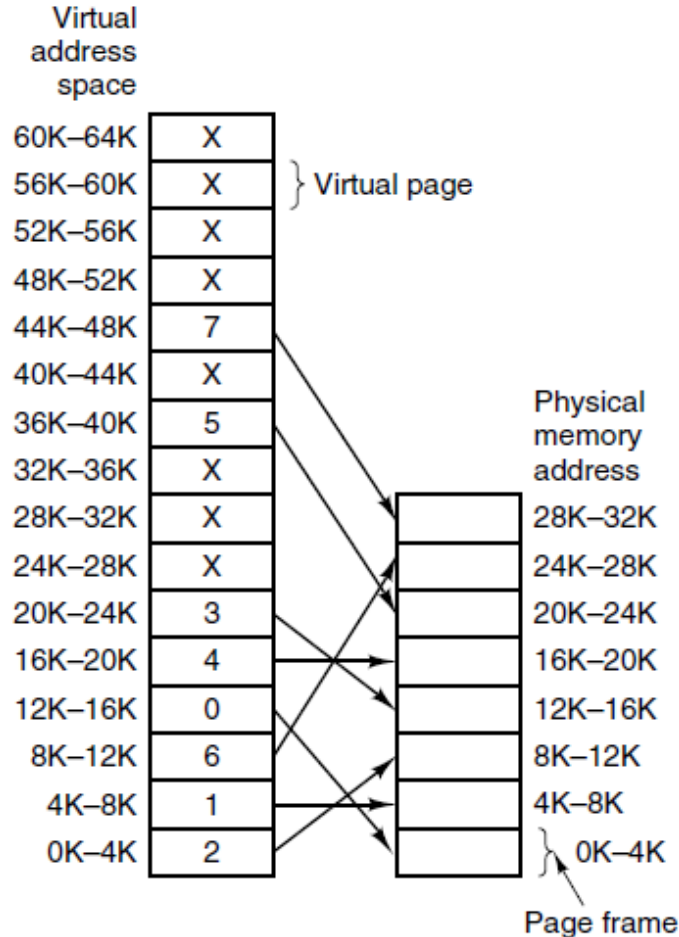
### **pt.1**

# Introdução

- Programas maiores que a memória disponível?
  - Programa dividido em módulos → Overlays
  - Antes: divisão a cargo do programador
  - Hoje: memória virtual

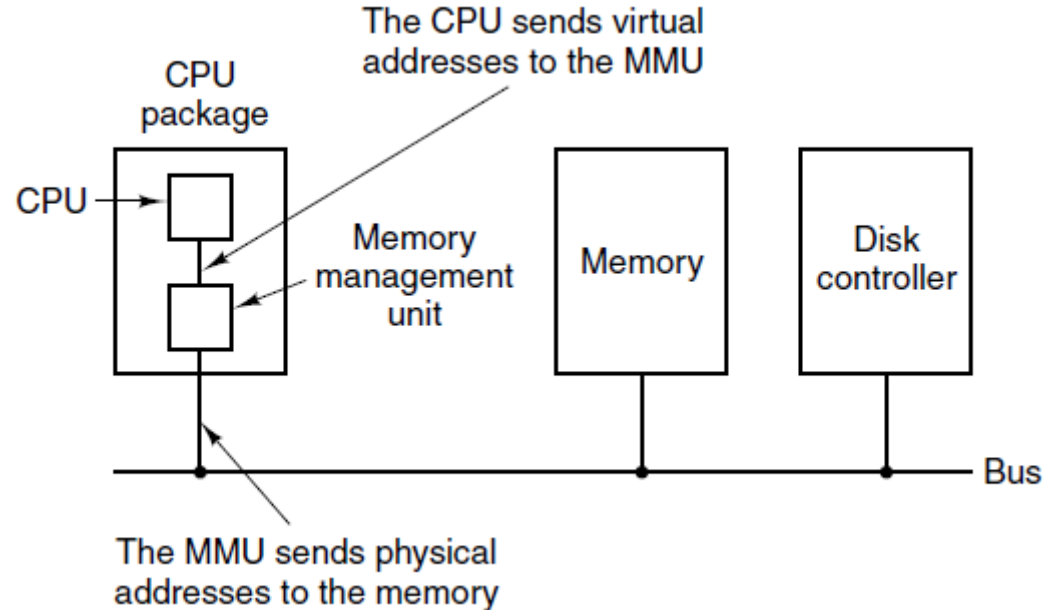
# Paginação

- Página
  - Espaço de endereçamento virtual
- Moldura de página
  - Unidades correspondentes na memória física



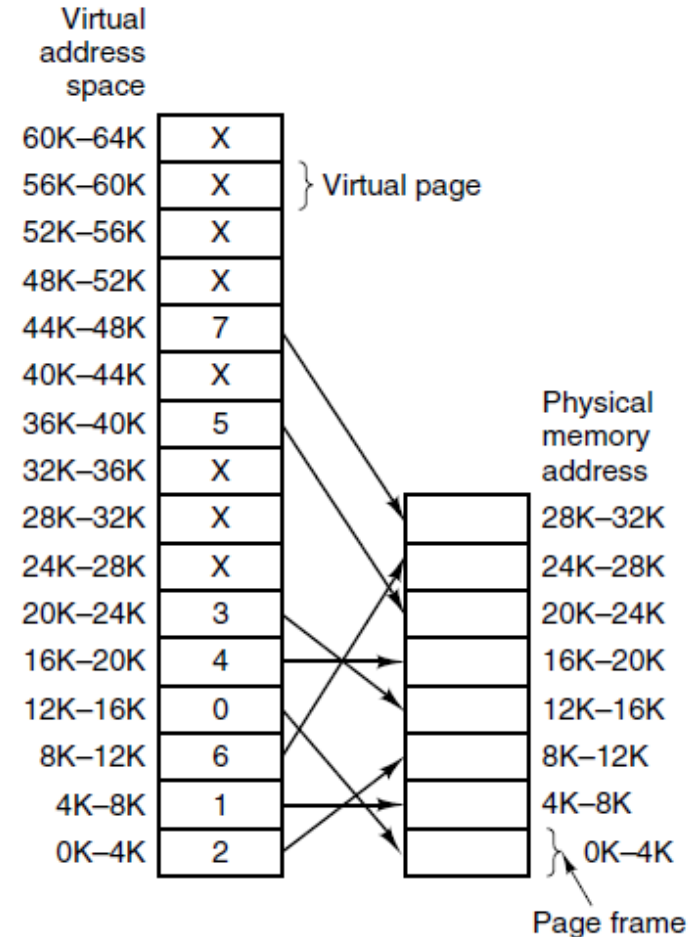
# Paginação

- MMU
  - Unidade de gerenciamento de memória
  - Mapeia endereços virtuais em endereço físicos



# Paginação

- Página não mapeada
  - Page fault (interrupção)
  - SO → escolhe moldura pouco usada para a substituição

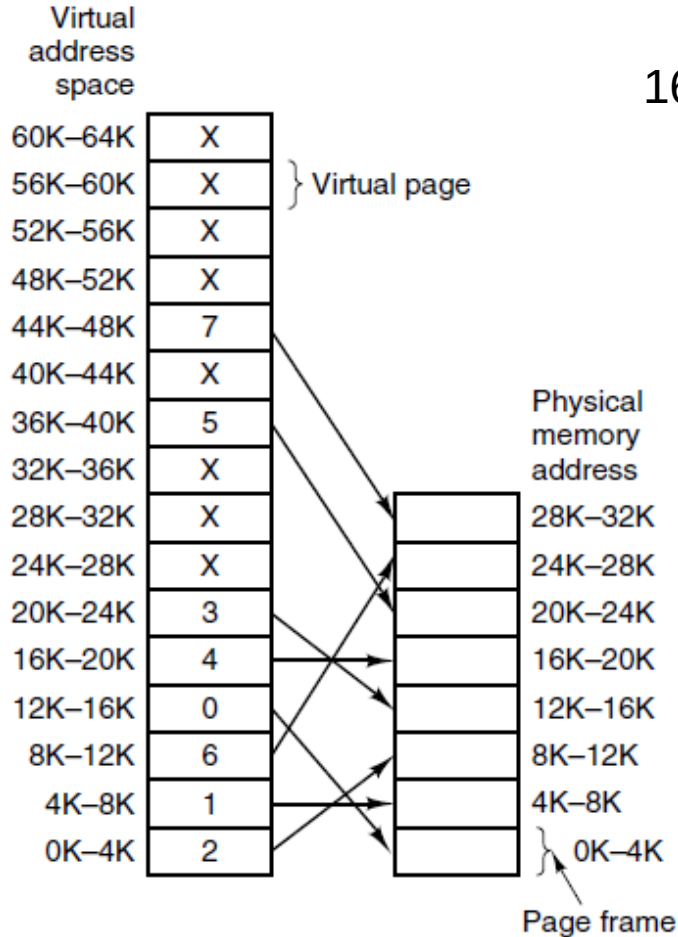


$$2^{16} = 65536$$

$$2^{12} = 4096$$

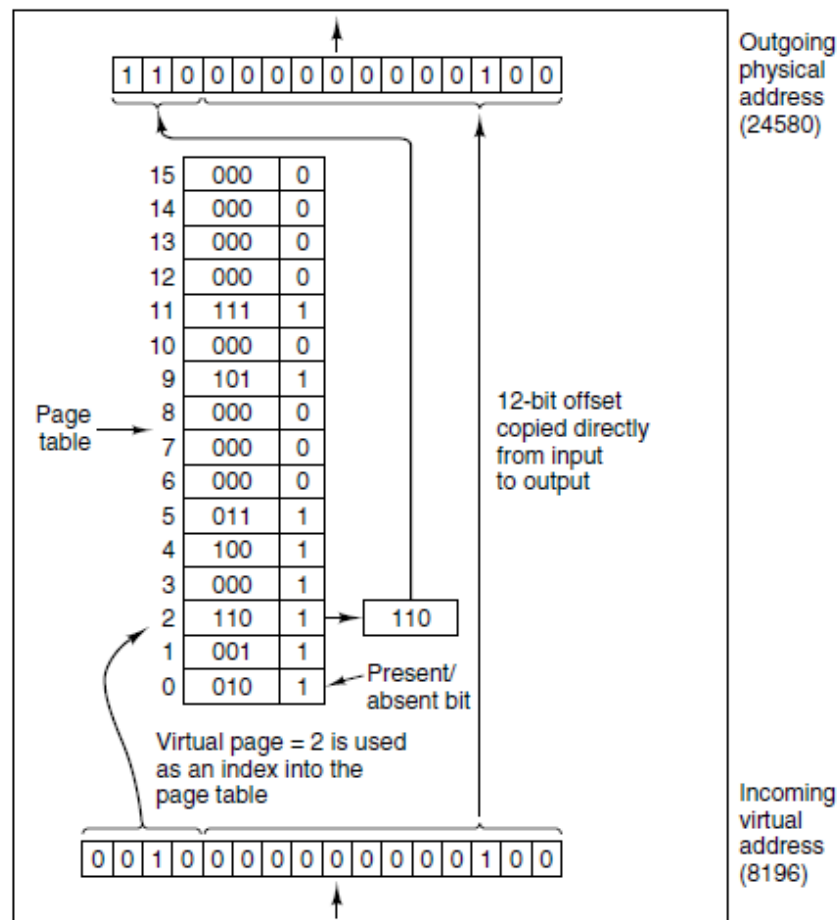
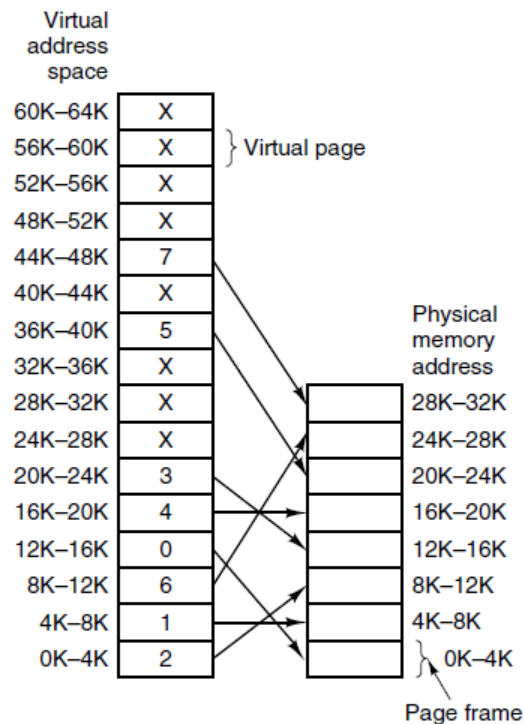
# Tabela de Páginas

16 bits: 4bits página + 12 bits endereço na página



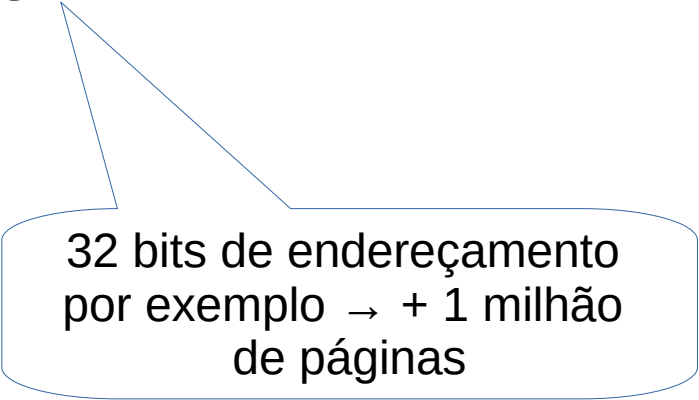
# Tabela de Páginas

- Forma mais simples de mapear



# Tabela de Páginas

- Problemas
  - Tabela pode ser extremamente grande
  - O mapeamento deve ser rápido

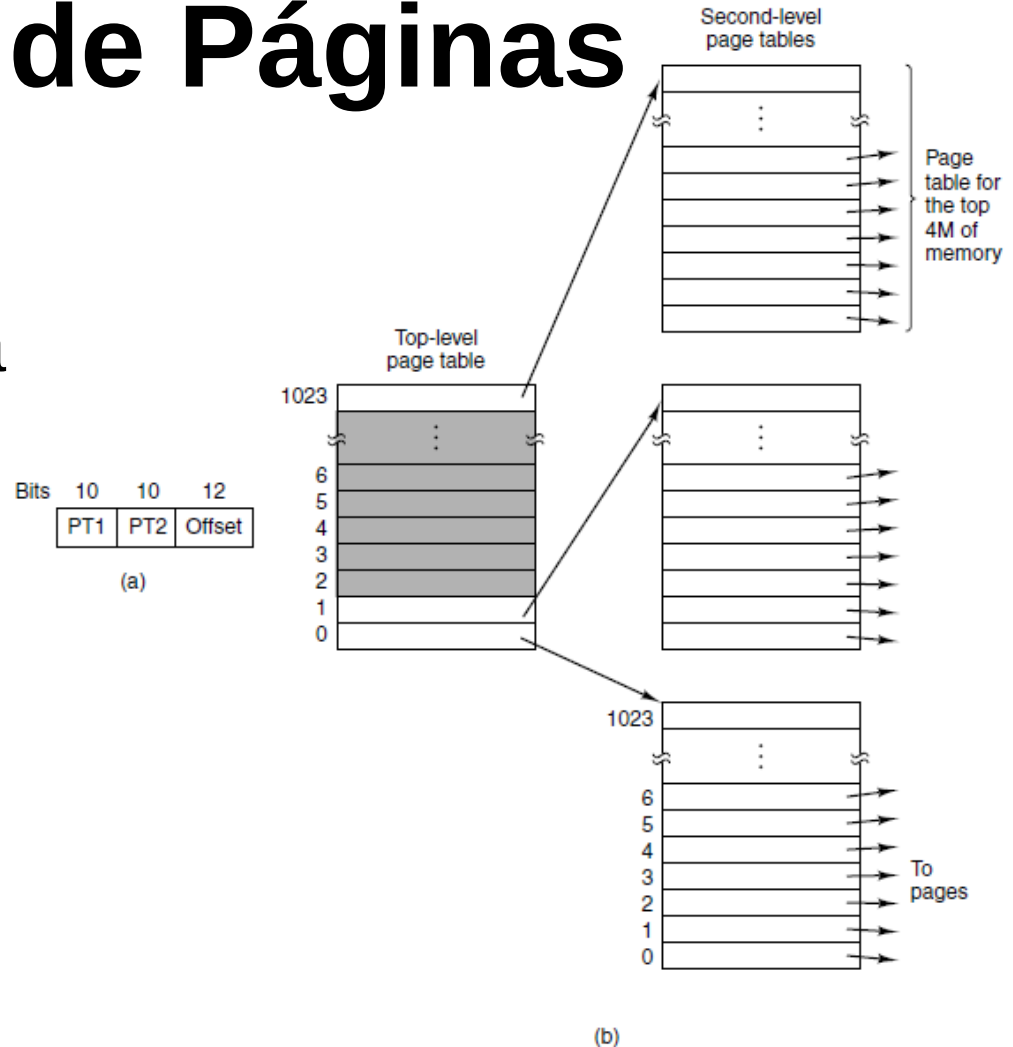


32 bits de endereçamento  
por exemplo → + 1 milhão  
de páginas



# Tabelas de Páginas

- Multinível
  - Minimiza o problema
  - Evitando que todas as tabelas sejam mantidas na memória



# Memória associativa

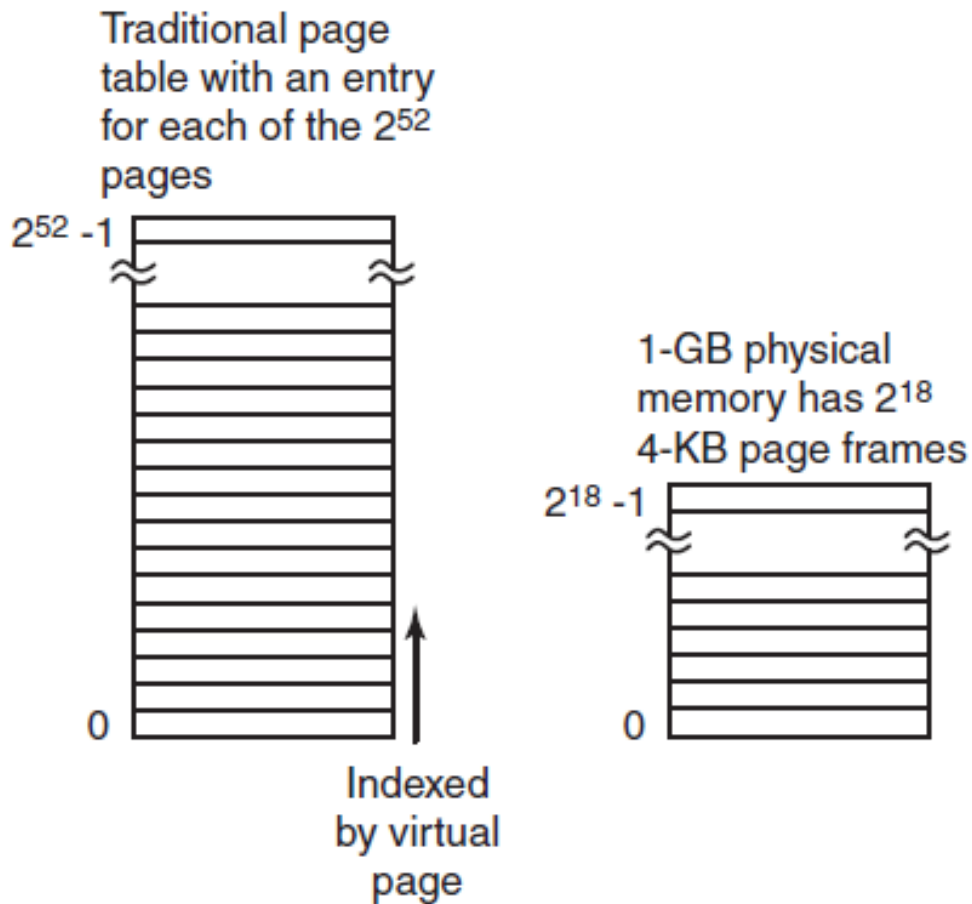
- TLB (Translation lookaside buffer)
  - Dispositivo dentro da MMU
  - Contém pequeno número de entradas que agiliza o mapeamento das páginas

Valid	Virtual page	Modified	Protection	Page frame
1	140	1	RW	31
1	20	0	R X	38
1	130	1	RW	29
1	129	1	RW	62
1	19	0	R X	50
1	21	0	R X	45
1	860	1	RW	14
1	861	1	RW	75

# Tabela de Páginas Invertida

- Uma entrada por moldura de página
  - Ao invés de uma entrada por página do endereçamento virtual
- Economiza muito espaço
  - Tradução de endereços é mais lenta

# Tabela de Páginas Invertida



# Tabela de Páginas Invertida

- Usar o valor de hash do endereço

