

# QXD0013 - Sistemas Operacionais

## Gerenciamento de Memória II

Marcos Dantas Ortiz<sup>1</sup>

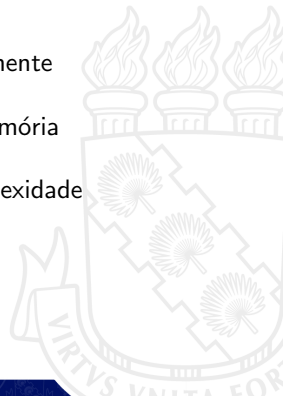
<sup>1</sup>Universidade Federal do Ceará, Brazil

10/10/2022

# Memória Virtual

---

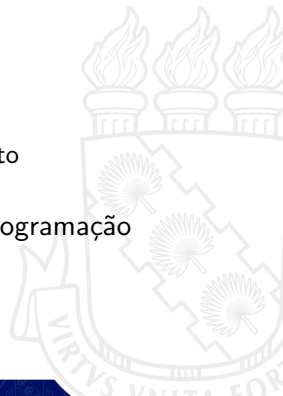
- Endereçamento → Abstração → Registradores-base/-limite
- Como gerenciar softwares que demandam muita memória?
- Possível solução: sobreposição (*overlay*)
  - Gerenciador de sobreposição → Carregado inicialmente
  - Programa dividido em módulos → Sobreposições
  - Troca e coexistência de sobreposições: HD ↔ Memória
  - Troca realizada pelo SO
  - Divisão do programa → Programador → + complexidade



# Memória Virtual

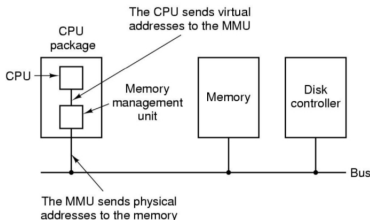
---

- Método concebido por Fotheringham em 1961: Memória Virtual
- Generalização do conceito de registradores-base/-limite
- Espaço de endereçamento dividido em blocos: Páginas
  - Cada página: série contínua de endereços
  - Mapeadas na memória física
  - Nem todas presentes na memória física
- Quando programa referencia endereçamento:
  - Página mapeada → Hardware executa mapeamento
  - Página não mapeada → SO alertado
- Também apropriado para ambientes com multiprogramação
  - Troca de CPU durante carregamento de página

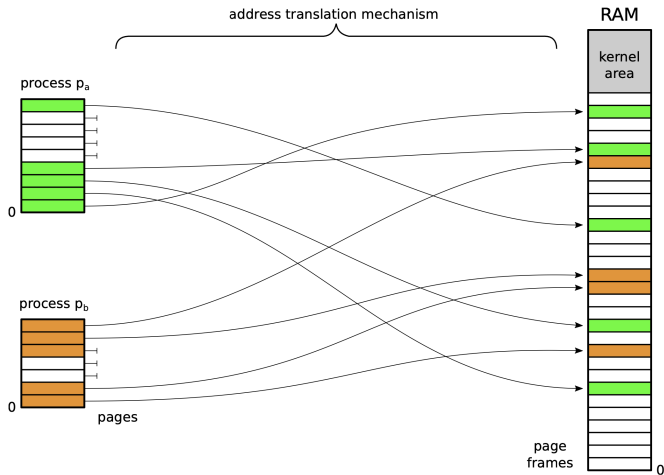


# Paginação

- Técnica utilizada pela maioria dos sistemas com memória virtual
- Cada programa
  - Conjunto de endereços virtuais = Espaço de endereçamento virtual
- Sem memória virtual: Virtual = Físico
  - Ler/Escrever → Direto no barramento de memória
- Com memória virtual
  - Endereço virtual → Memory Management Unit (MMU)
  - MMU: Mapeamento Virtual → Físico



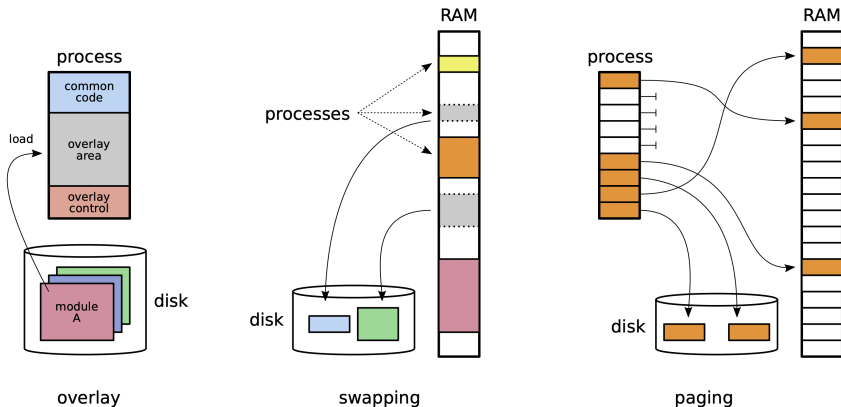
# Paginação



Fonte:

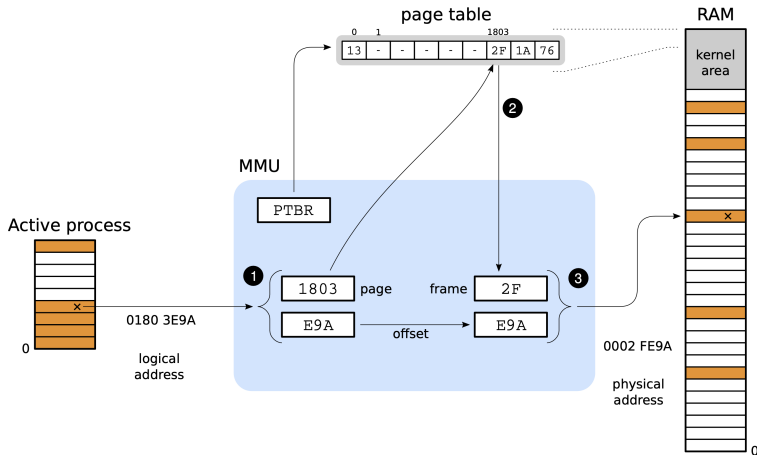
Maziero, C. A. (2014). Sistemas operacionais: conceitos e mecanismos. Livro aberto.

# Memória Virtual - Abordagens



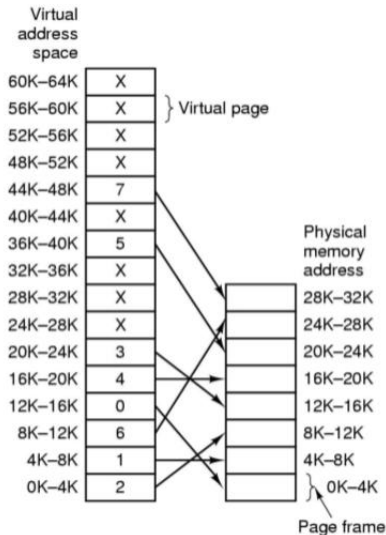
Fonte: Maziero, C. A. (2014). Sistemas operacionais: conceitos e mecanismos. Livro aberto.

# Paginação - Tradução de Endereços



Fonte: Maziero, C. A. (2014). Sistemas operacionais: conceitos e mecanismos. Livro aberto.

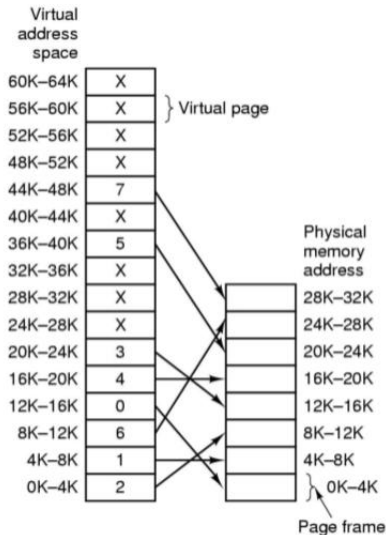
## Paginação: Exemplo



- Endereços virtuais de 16 bits (64KB)
- Apenas 32KB de memória física
- Memória física dividida em molduras
- Memória virtual
  - Divisão em páginas (*pages*)
  - Cada página = 4KB
- Memória física
  - Divisão em molduras (*page frames*)
  - Cada moldura de página = 4KB
- HD ↔ Memória: Páginas completas

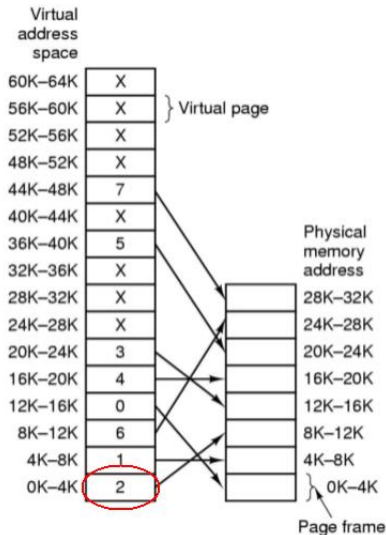


# Paginação: Exemplo



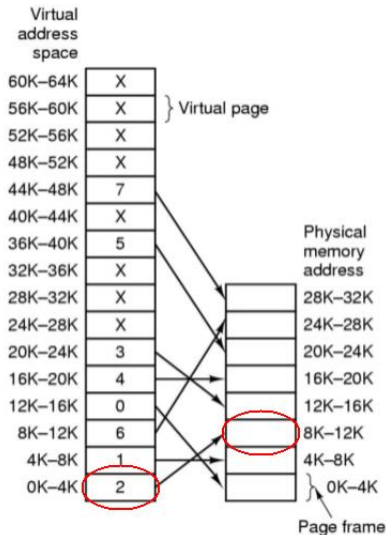
- Instrução: MOV REG,0
  - Endereço 0 → MMU
  - Endereço 0 → Moldura 2
  - Instrução efetiva:  
MOV REG,8192
- Instrução: MOV REG,32780
  - Endereço 32780 → MMU
  - Não mapeada
  - MMU: Falta de página → SO
  - SO salva uma moldura no HD
  - SO carrega página na moldura

# Paginação: Exemplo



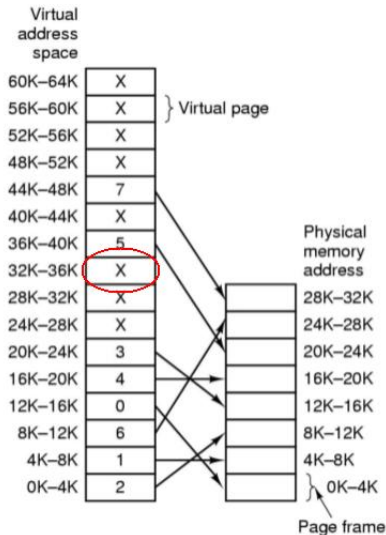
- Instrução: MOV REG,0
  - Endereço 0 → MMU
  - Endereço 0 → Moldura 2
  - Instrução efetiva:  
MOV REG,8192
- Instrução: MOV REG,32780
  - Endereço 32780 → MMU
  - Não mapeada
  - MMU: Falta de página → SO
  - SO salva uma moldura no HD
  - SO carrega página na moldura

## Paginação: Exemplo



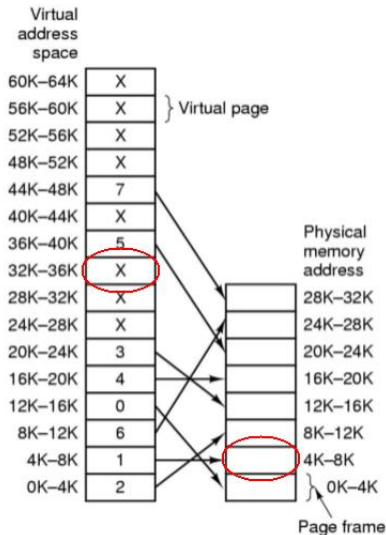
- Instrução: MOV REG,0
  - Endereço 0 → MMU
  - Endereço 0 → Moldura 2
  - Instrução efetiva:  
MOV REG,8192
- Instrução: MOV REG,32780
  - Endereço 32780 → MMU
  - Não mapeada
  - MMU: Falta de página → SO
  - SO salva uma moldura no HD
  - SO carrega página na moldura

## Paginação: Exemplo



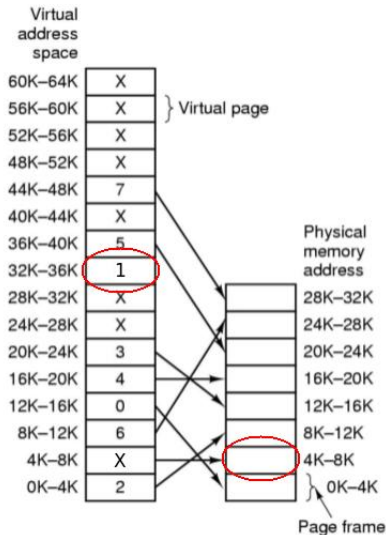
- Instrução: MOV REG,0
  - Endereço 0 → MMU
  - Endereço 0 → Moldura 2
  - Instrução efetiva:  
MOV REG,8192
- Instrução: MOV REG,32780
  - Endereço 32780 → MMU
  - Não mapeada
  - MMU: Falta de página → SO
  - SO salva uma moldura no HD
  - SO carrega página na moldura

## Paginação: Exemplo



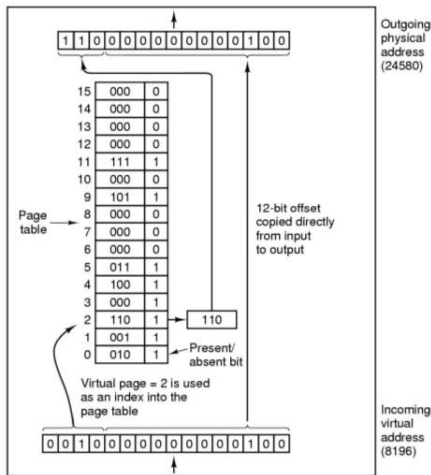
- Instrução: MOV REG,0
  - Endereço 0 → MMU
  - Endereço 0 → Moldura 2
  - Instrução efetiva:  
MOV REG,8192
- Instrução: MOV REG,32780
  - Endereço 32780 → MMU
  - Não mapeada
  - MMU: Falta de página → SO
  - SO salva uma moldura no HD
  - SO carrega página na moldura

## Paginação: Exemplo



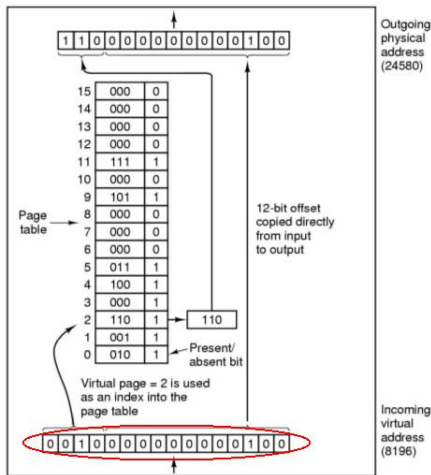
- Instrução: MOV REG,0
  - Endereço 0 → MMU
  - Endereço 0 → Moldura 2
  - Instrução efetiva:  
MOV REG,8192
- Instrução: MOV REG,32780
  - Endereço 32780 → MMU
  - Não mapeada
  - MMU: Falta de página → SO
  - SO salva uma moldura no HD
  - SO carrega página na moldura

# Funcionamento MMU



- Endereço de 16 bits chega à MMU
- Duas partes:
  - 4 bits → Página (16 páginas)
  - 12 bits → Deslocamento ( $2^{12}$  posições)
- Tabela de páginas:
  - Posição da página
  - Mapeado/Não Mapeado
- Endereço de 15 bits da memória física

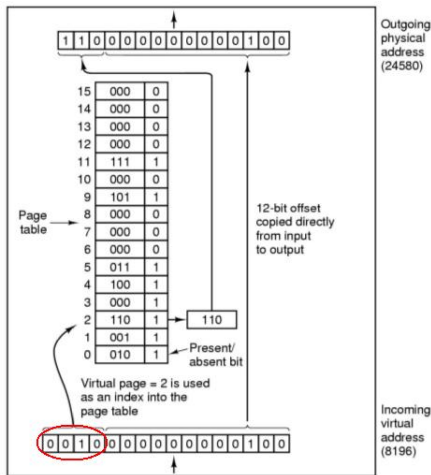
# Funcionamento MMU



- Endereço de 16 bits chega à MMU
- Duas partes:
  - 4 bits → Página (16 páginas)
  - 12 bits → Deslocamento ( $2^{12}$  posições)
- Tabela de páginas:
  - Posição da página
  - Mapeado/Não Mapeado
- Endereço de 15 bits da memória física

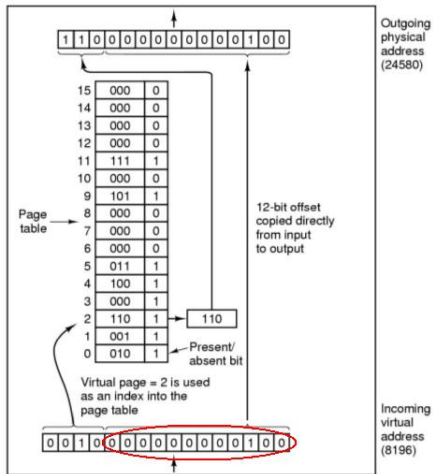


# Funcionamento MMU



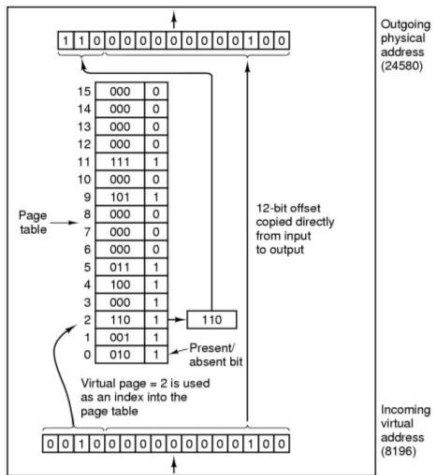
- Endereço de 16 bits chega à MMU
- Duas partes:
  - 4 bits → Página (16 páginas)
  - 12 bits → Deslocamento ( $2^{12}$  posições)
- Tabela de páginas:
  - Posição da página
  - Mapeado/Não Mapeado
- Endereço de 15 bits da memória física

# Funcionamento MMU



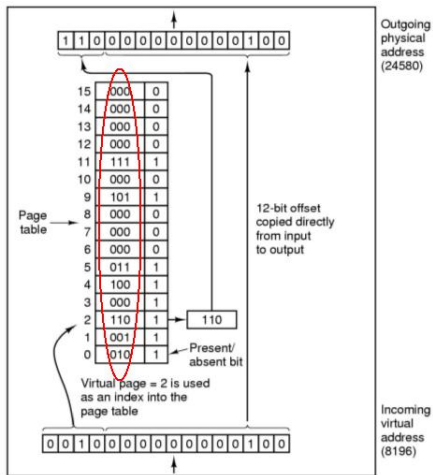
- Endereço de 16 bits chega à MMU
- Duas partes:
  - 4 bits → Página (16 páginas)
  - 12 bits → Deslocamento ( $2^{12}$  posições)
- Tabela de páginas:
  - Posição da página
  - Mapeado/Não Mapeado
- Endereço de 15 bits da memória física

# Funcionamento MMU



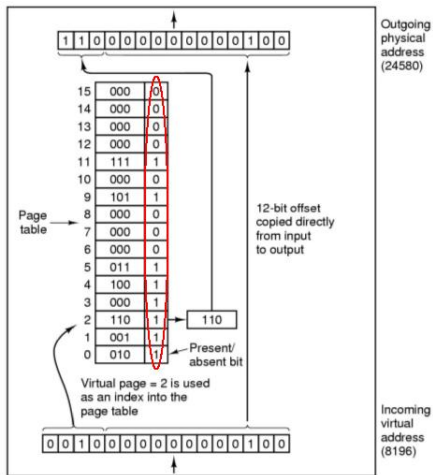
- Endereço de 16 bits chega à MMU
- Duas partes:
  - 4 bits → Página (16 páginas)
  - 12 bits → Deslocamento ( $2^{12}$  posições)
- Tabela de páginas:
  - Posição da página
  - Mapeado/Não Mapeado
- Endereço de 15 bits da memória física

# Funcionamento MMU



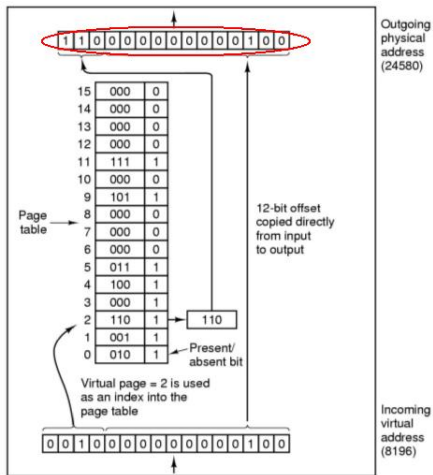
- Endereço de 16 bits chega à MMU
- Duas partes:
  - 4 bits → Página (16 páginas)
  - 12 bits → Deslocamento ( $2^{12}$  posições)
- Tabela de páginas:
  - Posição da página
  - Mapeado/Não Mapeado
- Endereço de 15 bits da memória física

# Funcionamento MMU



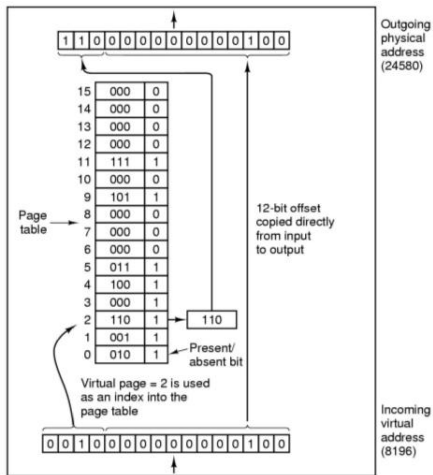
- Endereço de 16 bits chega à MMU
- Duas partes:
  - 4 bits → Página (16 páginas)
  - 12 bits → Deslocamento ( $2^{12}$  posições)
- Tabela de páginas:
  - Posição da página
  - Mapeado/Não Mapeado
- Endereço de 15 bits da memória física

# Funcionamento MMU



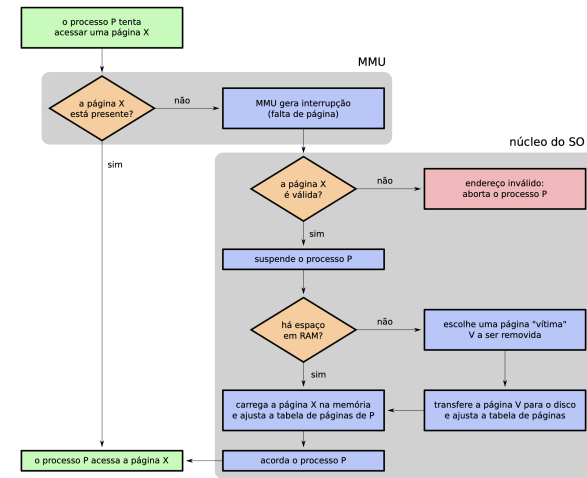
- Endereço de 16 bits chega à MMU
- Duas partes:
  - 4 bits → Página (16 páginas)
  - 12 bits → Deslocamento ( $2^{12}$  posições)
- Tabela de páginas:
  - Posição da página
  - Mapeado/Não Mapeado
- Endereço de 15 bits da memória física

# Funcionamento MMU



- Endereço de 16 bits chega à MMU
- Duas partes:
  - 4 bits → Página (16 páginas)
  - 12 bits → Deslocamento ( $2^{12}$  posições)
- Tabela de páginas:
  - Posição da página
  - Mapeado/Não Mapeado
- Endereço de 15 bits da memória física

# Resumo - Paginação



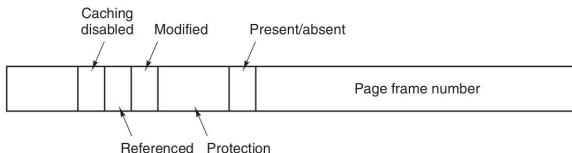
Fonte: Maziero, C.

A. (2014). Sistemas operacionais: conceitos e mecanismos. Livro aberto.



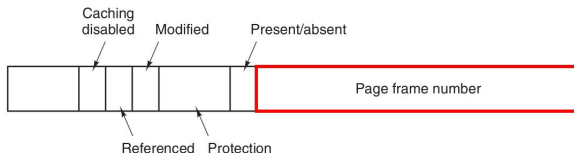
# Estrutura Detalhada de uma Página

- Tamanho comum: 32 bits
- Campo mais importante: número da moldura de página
- Bit de mapeamento: Se igual a zero → interrupção
- Quais acessos permitidos: 2 ou 3 bits
- Indica se a página foi modificada
  - Escrita na página → Igual a 1
  - Sincronização Memória ↔ HD
- Indica se a página está sendo referenciada
  - Escolha da página a ser substituída em caso de falta
- Controla uso do mecanismo de cache
  - Importante para E/S mapeada em memória



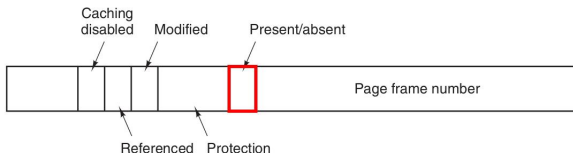
# Estrutura Detalhada de uma Página

- Tamanho comum: 32 bits
- Campo mais importante: número da moldura de página
- Bit de mapeamento: Se igual a zero → interrupção
- Quais acessos permitidos: 2 ou 3 bits
- Indica se a página foi modificada
  - Escrita na página → Igual a 1
  - Sincronização Memória ↔ HD
- Indica se a página está sendo referenciada
  - Escolha da página a ser substituída em caso de falta
- Controla uso do mecanismo de cache
  - Importante para E/S mapeada em memória



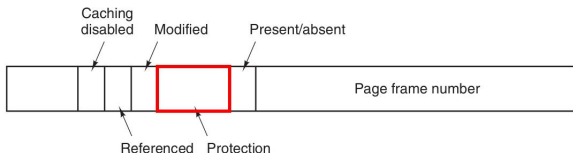
# Estrutura Detalhada de uma Página

- Tamanho comum: 32 bits
- Campo mais importante: número da moldura de página
- Bit de mapeamento: Se igual a zero → interrupção
- Quais acessos permitidos: 2 ou 3 bits
- Indica se a página foi modificada
  - Escrita na página → Igual a 1
  - Sincronização Memória ↔ HD
- Indica se a página está sendo referenciada
  - Escolha da página a ser substituída em caso de falta
- Controla uso do mecanismo de cache
  - Importante para E/S mapeada em memória



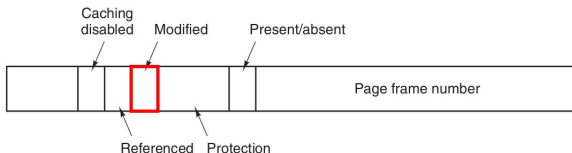
# Estrutura Detalhada de uma Página

- Tamanho comum: 32 bits
- Campo mais importante: número da moldura de página
- Bit de mapeamento: Se igual a zero → interrupção
- Quais acessos permitidos: 2 ou 3 bits
- Indica se a página foi modificada
  - Escrita na página → Igual a 1
  - Sincronização Memória ↔ HD
- Indica se a página está sendo referenciada
  - Escolha da página a ser substituída em caso de falta
- Controla uso do mecanismo de cache
  - Importante para E/S mapeada em memória



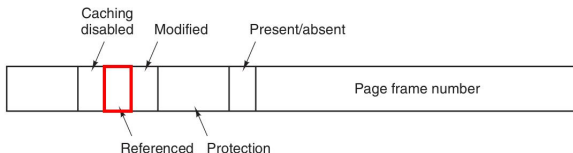
# Estrutura Detalhada de uma Página

- Tamanho comum: 32 bits
- Campo mais importante: número da moldura de página
- Bit de mapeamento: Se igual a zero → interrupção
- Quais acessos permitidos: 2 ou 3 bits
- Indica se a página foi modificada
  - Escrita na página → Igual a 1
  - Sincronização Memória ↔ HD
- Indica se a página está sendo referenciada
  - Escolha da página a ser substituída em caso de falta
- Controla uso do mecanismo de cache
  - Importante para E/S mapeada em memória



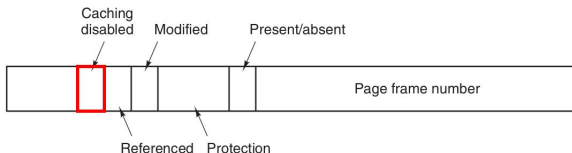
# Estrutura Detalhada de uma Página

- Tamanho comum: 32 bits
- Campo mais importante: número da moldura de página
- Bit de mapeamento: Se igual a zero → interrupção
- Quais acessos permitidos: 2 ou 3 bits
- Indica se a página foi modificada
  - Escrita na página → Igual a 1
  - Sincronização Memória ↔ HD
- Indica se a página está sendo referenciada
  - Escolha da página a ser substituída em caso de falta
- Controla uso do mecanismo de cache
  - Importante para E/S mapeada em memória



# Estrutura Detalhada de uma Página

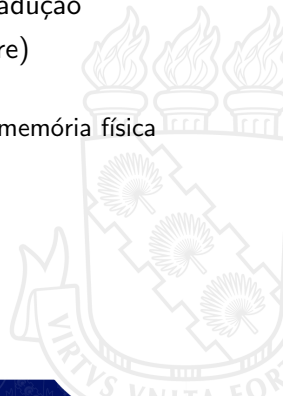
- Tamanho comum: 32 bits
- Campo mais importante: número da moldura de página
- Bit de mapeamento: Se igual a zero → interrupção
- Quais acessos permitidos: 2 ou 3 bits
- Indica se a página foi modificada
  - Escrita na página → Igual a 1
  - Sincronização Memória ↔ HD
- Indica se a página está sendo referenciada
  - Escolha da página a ser substituída em caso de falta
- Controla uso do mecanismo de cache
  - Importante para E/S mapeada em memória



# Paginação

---

- Tabela não possui informação dos endereços em disco
- Somente informações para o hardware realizar tradução
- Outras informações controladas pelo SO (software)
- Memória Virtual → Cria abstração
  - Espaço de endereçamento virtual = abstração da memória física
  - Assim como processo é para CPU





# Acelerando Paginação

---

- Acabamos de ver os princípios básicos da **memória virtual** e da **paginação**.
- Em qualquer sistema de paginação, duas questões fundamentais precisam ser abordadas:
  - O mapeamento do endereço virtual para o endereço físico precisa ser rápido.
  - Se o espaço do endereço virtual for grande, a tabela de páginas será grande.

# Acelerando Paginação

---

- Mapeamento deve ser rápido
  - É uma consequência do fato de que o **mapeamento virtual-físico** precisa ser feito em cada referência de memória.
- Tabela proporcional ao endereçamento virtual
  - Decorre do fato de que todos os computadores modernos usam endereços virtuais de pelo menos 32 bits, com 64 bits tornando-se a norma para computadores de mesa e laptops.
  - 4KB de página + 32 bits endereçamento → 1 milhão de páginas
  - Uma tabela por processo (lembre-se de que cada processo precisa da sua própria tabela de páginas. **Por que?**)

# Acelerando Paginação

---

- Mapeamento deve ser rápido
  - É uma consequência do fato de que o **mapeamento virtual-físico** precisa ser feito em cada referência de memória.
- Tabela proporcional ao endereçamento virtual
  - Decorre do fato de que todos os computadores modernos usam endereços virtuais de pelo menos 32 bits, com 64 bits tornando-se a norma para computadores de mesa e laptops.
  - 4KB de página + 32 bits endereçamento → 1 milhão de páginas
  - Uma tabela por processo (lembre-se de que cada processo precisa da sua própria tabela de páginas. (**Porque ele tem seu próprio espaço de endereço virtual**)).

# TLB (Translation Lookaside Buffers) ou memória associativa

---

- Vamos examinar agora esquemas amplamente implementados para acelerar a paginação e lidar com grandes espaços de endereços virtuais
- O ponto de partida da maioria das técnicas de otimização é o fato de a tabela de páginas estar na memória.
- Na memória
  - 2x mais lento
- Sob essas condições, ninguém usaria a paginação.
- **Qual a solução?**

# TLB (Translation Lookaside Buffers) ou memória associativa

---

- A solução que foi concebida é equipar os computadores com um pequeno dispositivo de hardware para mapear **endereços virtuais** para **endereços físicos** sem ter de passar pela tabela de páginas.

# TLB (Translation Lookaside Buffers) ou memória associativa

---

- Qual é o propósito do translation lookaside buffer (TLB)?

# TLB (Translation Lookaside Buffers) ou memória associativa

---

- Qual é o propósito do **translation lookaside buffer (TLB)**?
  - O TLB é um cache que contém as entradas da tabela de página que têm sido mais usados recentemente.
  - O objetivo é evitar que, na maioria das vezes, ter de ir para o disco para recuperar uma entrada da tabela de página

# Acelerando Paginação - TLB

---

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



# TLB (Translation Lookaside Buffers) ou memória associativa

---

- Vamos ver agora como a **TLB funciona**.
  - Quando um endereço virtual é apresentado para a MMU para tradução, o hardware primeiro confere para ver se o seu número de página virtual está presente na TLB comparando-o com todas as entradas simultaneamente (isto é, em paralelo).
  - É necessário um hardware especial para realizar isso, que todas as MMUs com TLBs têm.

# Acelerando Paginação - SW TLB

---

- Vantagem quando a TLB é grande → MMU mais simples
- Estratégias para melhora do desempenho:
  - Reduzir número e custo de *page miss*
  - Antecipar página a serem carregadas
  - Cache maior com entradas do tipo TLB
- *soft miss* x *hard miss*

