

# Sistemas Operacionais



Gerência de Memória e Memória Virtual

# Introdução

---

## □ Multiprogramação

- Vários processos prontos para execução, concorrem à CPU
- Para que a troca de contexto seja rápida, os processos devem estar na memória principal

## □ Gerência de Memória do SO

- deve prover mecanismos para que os processos utilizem a memória de forma segura e eficiente
- há várias técnicas de gerência
- cada SO emprega uma técnica em particular e isto depende fortemente da arquitetura do computador

## □ Memória

- Memória = vetor de palavras ou bytes com seus endereços
- A CPU busca instruções da memória, de acordo com o valor do Contador de Programa (podem ser de busca ou armazenamento em endereços de memória específicos)

# Espaço de Endereçamento Lógico x Físico

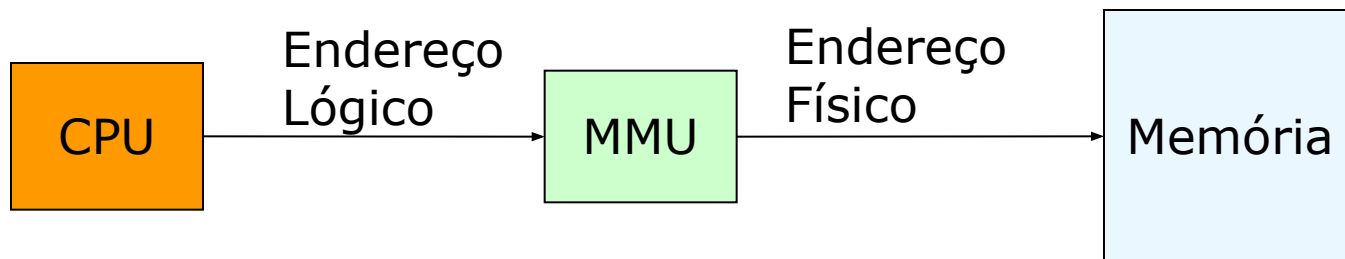
---

- **Memória Lógica** de um processo
  - é aquela que o processo enxerga
  - os endereços manipulados pelo processo são **Endereços Lógicos**
  - Por exemplo, as variáveis de um processo contém endereços lógicos.
  - Cada processo possui sua memória lógica, independente da memória lógica de outros processos
  - **Espaço de Endereçamento Lógico** (ou Virtual) de um processo é o conjunto de endereços lógicos que esse processo pode endereçar (enxergar). Há um espaço de endereçamento lógico por processo.
  
- **Memória Física**
  - é aquela implementada pelos circuitos integrados de memória
  - o **Endereço Físico** é usado para endereçar os circuitos integrados
  - **Espaço de Endereçamento Físico** é formado por todos os endereços aceitos pelos circuitos integrados de memória.

# Espaço de Endereçamento Lógico x Físico (2)

---

- Os mapeamentos de endereços em tempo de compilação e em tempo de carga resultam em endereçamentos lógicos e físicos iguais. O mapeamento de endereços em tempo de execução, não.
- O Mapeamento de tempo de execução dos endereços lógicos para físicos é feito pela **Unidade de Gerência de Memória (MMU – Memory Management Unit)**
  - A MMU é um componente do hardware



# Alocação Não Contígua



Paginação  
Segmentação

# Paginação

---

- Há o espaço de endereçamento lógico (processo) e o espaço de endereçamento físico.
- O espaço de endereçamento lógico é contíguo (visão do usuário)
- O espaço de endereçamento físico não precisa ser contíguo.
- A memória lógica é dividida em blocos de tamanho fixo e idênticos □ Páginas
- A memória física é dividida em blocos de tamanho fixo e idênticos □ Frames ou Quadros

# Paginação

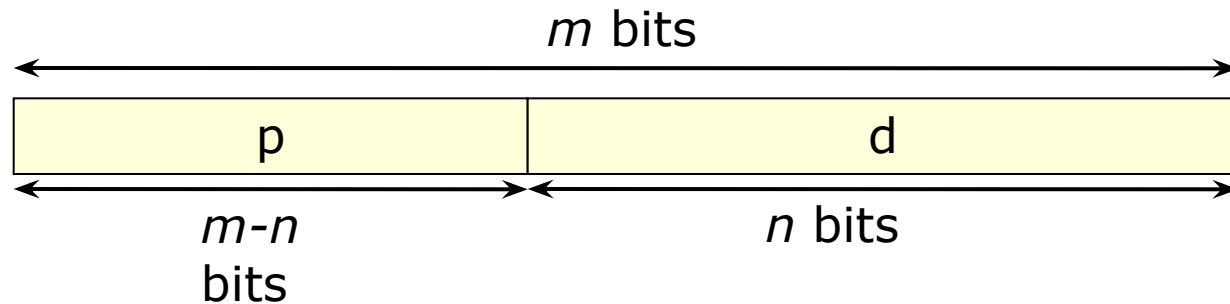
---

- Há necessidade de mapear o espaço lógico para o espaço físico
  - Endereço lógico □ Endereço Físico
- Fragmentação Interna
  - Restrita a última página ocupada pelo processo
- Cada página é alocada em um quadro
- Para o mapeamento de endereços, é utilizada a Tabela de Páginas
- A Tabela de Páginas é usada também para Proteção da Memória e Compartilhamento

# Paginação

---

- Endereço Lógico
  - $\langle n^{\circ} \text{ página}, \text{deslocamento} \rangle$
- Endereço Físico
  - $\langle n^{\circ} \text{ quadro}, \text{deslocamento} \rangle$





# Esquema Básico da Paginação

Processo	
Página	
0	1 A1
1	A2
2	A3
3	A4
4	A5

Tabela de Páginas

Processo 1

Página	Quadr	v/
a	o	i
0	2	v
1	3	v
2	8	v
3	9	v
4		i

Processo	
Página	
0	2 B1
1	B2
2	B3
3	B4

Tabela de Páginas

Processo 2

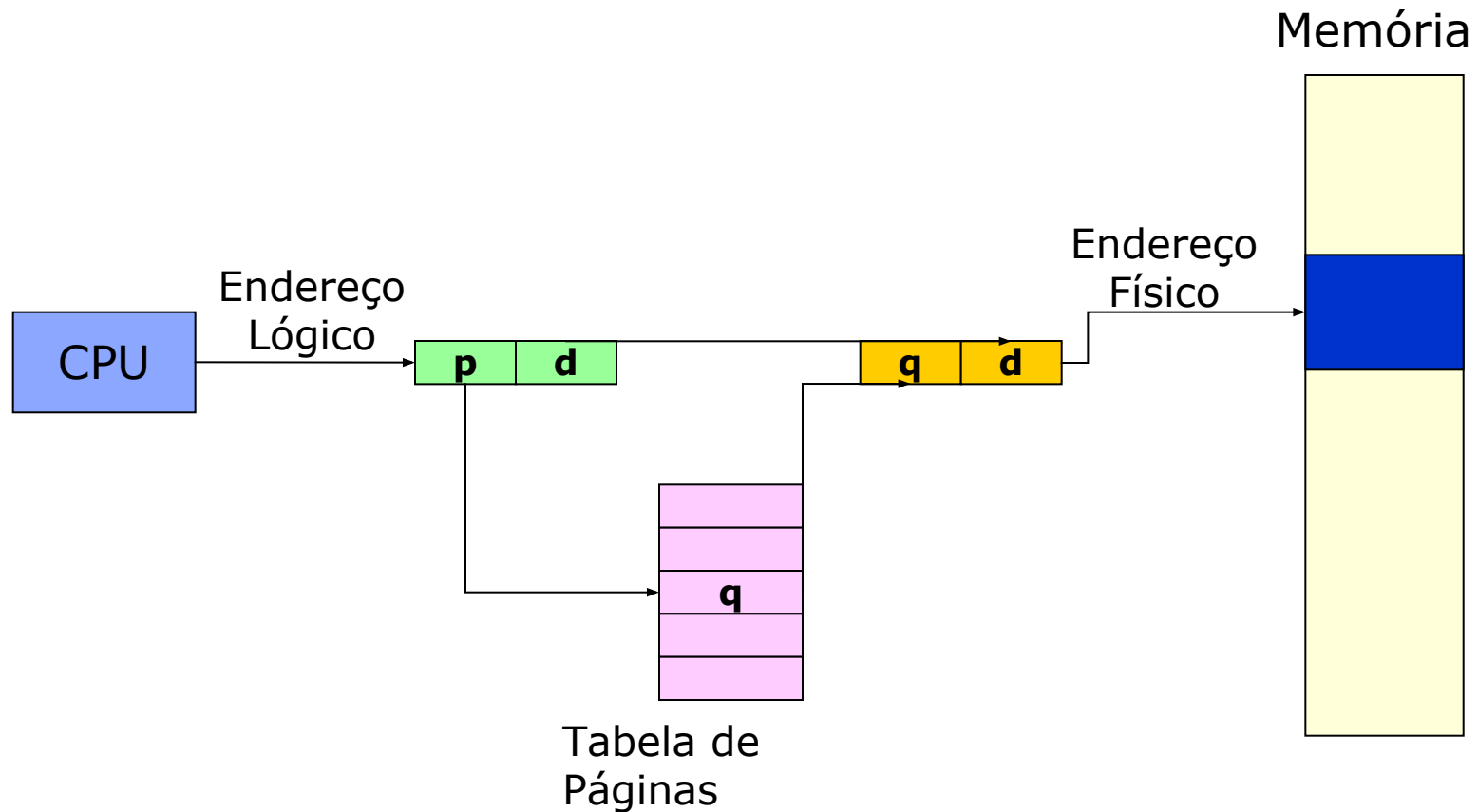
Página	Quadr	v/
a	o	i
0	4	v
1	5	v
2	6	v
3	7	v

Memória Principal

Quadro	
0	
1	
2	A1
3	A2
4	B1
5	B2
6	B3
7	B4
8	A3
9	A4
10	
11	
12	
13	
14	

# Paginação

## □ Mapeamento Endereço Lógico para Físico



# Paginação

---

- Tamanho da página: até 8 kbytes
- O SO mantém uma lista com os quadros livres
- Páginas pequenas implicam...
- Páginas grandes implicam...

# Paginação

---

## □ Proteção

- Processos acessam somente suas páginas
- Os registradores que controlam o acesso à tabela de páginas são acessados somente no modo supervisor
- Bits de controle
  - Para cada página da tabela
  - Indicam leitura, escrita ou executável
- Bit de Verificação (ou Validade)
  - Indica se a página pertence ao espaço de endereçamento lógico do processo

# Memória Virtual



# Memória Virtual

---

- Possibilita a execução de programas que não são carregados completamente na memória física
  
- Exemplos
  - Programa com funções raramente utilizadas
  - Rotinas de tratamento de exceções ou acesso a arquivos
  - Programas que para execução exigem maior espaço de memória

# Mecanismo básico de Paginação

---

## □ Tabela de páginas

- Há um bit de validação de cada página
  - V = página válida = página carregada na memória
  - I = página inválida = página não carregada na memória ou fora do espaço de endereçamento do processo

## □ Ao acontecer um acesso à memória...

- Busca tabela de páginas
- Bit v □ faz mapeamento para endereço físico
- Bit i □ MMU gera interrupção de proteção ao SO
  - No PCB há informação se
    - a página está fora do espaço de endereçamento de um processo (neste caso a operação é abortada)
    - ou
    - houve **FALTA DE PÁGINA – Page Fault** (página não carregada)

# Paginação por Demanda

Processo 1

M. Lógica

0	A
1	B
2	C
3	D

Tabela Páginas

Lógica	Física	Bit
0	1	v
1	5	v
2		i
3	0	v

M. Física

0	D
1	A
2	X
3	X
4	X
5	B

Disco

A
B
C
D

*Pager*: parte do SO responsável por carregar uma página específica de um processo do disco para a memória principal



# Substituição de Páginas na Memória

---

- Ao acontecer *Page Fault* (Falta de Página), há necessidade de uma página livre na memória física.
- Para tanto, o SO deve
  - escolher uma página lógica (*Página Vítima*)
  - copiar o conteúdo dessa página para o disco
  - marcar a página como inválida, na tabela de páginas

# Substituição de Páginas na Memória

## - Exemplo

Processo 1

M. Lógica

0	A
1	B
2	C
3	D

Tabela Páginas

Lógica	Física	Bit
0	1	v
1	5	v
2		i
3	0	v

M. Física

0	D
1	A
2	F
3	E
4	G
5	B

Disco

A
B
C
D
E
F
G
H

Processo 2

M. Lógica

0	E
1	F
2	G
3	H

Tabela Páginas

Lógica	Física	Bit
0	3	v
1	2	v
2	4	v
3		i

**Algoritmo de Substituição de Páginas**

**SO escolhe PÁGINA VÍTIMA**

**MMU - interrupção**

# Substituição de Páginas

- Bits auxiliares são adicionados à tabela de página para auxiliar a substituição:

**Bit de Sujeira (dirty bit):** indica quando a página foi alterada durante a execução do processo (é zerado quando a página é carregada; “ligado” quando acontece a escrita).

**Bit de Referência (reference bit):** indica quando a página foi acessada pelo processo. É inicializado com ZERO quando a página é carregada; recebe 1, quando a página é usada.

**Bit de Trança (lock bit):** é usado para trancar uma página lógica na memória física (a página bloqueada não pode ser escolhida como vítima).

**Tabela de Páginas**

Pág. Lógica	Pág. Física	Bit v / i	Bit sujeira	Bit referência	Bit tranca

# Algoritmos de Substituição de Página

## ou *Page-Replacement Algorithm*

---

- Algoritmo Ótimo
- FCFS ou FIFO
- LRU
  - Aproximações do LRU
    - Histórico de Bits de Referência ou Bits de Referência Adicionais
    - Segunda Chance ou Algoritmo do Relógio (*clock algorithm*)
    - Segunda Chance melhorado
- Substituição baseada em contagem

Objetivo é gerar a menor taxa de falta de páginas

# Algoritmos de Substituição de Página

---

- Para determinar o número de faltas de páginas:
  - **String de Referência**
    - Série de referências à memória
  - **Quadros**
    - Saber o número de quadros disponíveis

# Algoritmo Ótimo

---

## □ **Algoritmo Ótimo**

- O algoritmo de substituição de página ótimo é o que coloca a falta de página em um futuro mais longe

### □ **Menor taxa de falta de páginas**

- Conhecido com OPT ou MIN
- Consiste em substituir a página que não será usada pelo período mais longo
- Para tanto, é necessário conhecer as próximas referências às páginas (difícil de implementar)

# FIFO

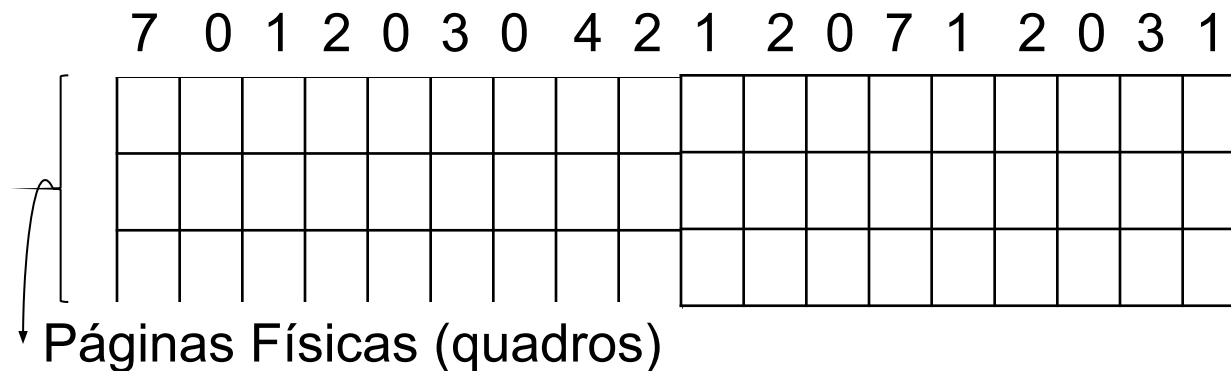
---

- FCFS (*First Come, First Served*) ou FIFO (*First In, First Out*)
  - A página escolhida para ser substituída (página vítima) é a página que está há mais tempo na memória
  - A MMU deve ter uma lista com os números das páginas lógicas. Quando uma página é carregada na memória, seu número é colocado no final da lista.
  - Desempenho ruim □ não considera se a página foi acessada
  - Exemplo
    - Rotina de Inicialização
    - Rotinas de Bibliotecas

# FIFO (Exemplo)

---

## String de Referência às Páginas:





# LRU

---

## □ LRU (*Least Recently Used*)

- Aproximação do Algoritmo Ótimo
- Escolhe a página que há mais tempo não é usada (acessada)
- Considera que páginas acessadas recentemente por um processo serão novamente acessadas por ele num futuro próximo
- Exige suporte de hardware (registradores adicionais)
  - **alto custo** para manter na tabela de páginas o momento exato do último acesso
- **Raramente encontrado** □ existem aproximações