

Abstrações de Memória: Projeto e Implementação de Paginação

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



Questões relativas ao projeto de paginação

- Política de alocação e de substituição
- Controle de carga
- Tamanho de páginas
- Compartilhamento de páginas

1

Política de alocação e de substituição

Política de alocação

- **Política de alocação global**

- Molduras de página são alocadas aos processos de forma **dinâmica**
- O **número de molduras de página** associadas a cada processo **varia com o tempo**

- **Política de alocação local**

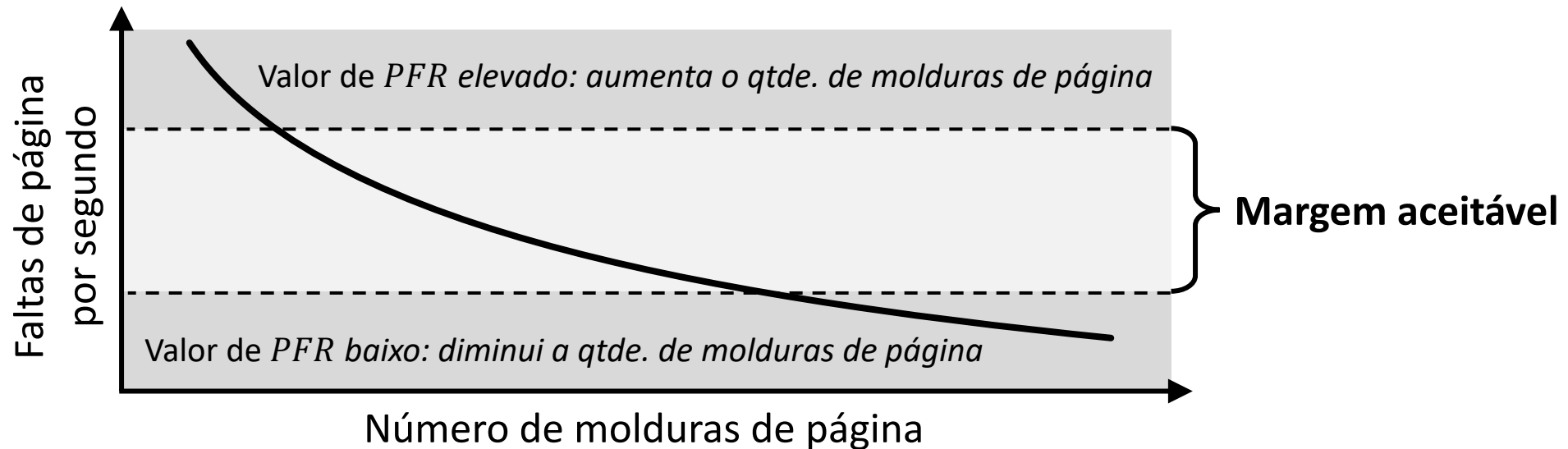
- Molduras de página são alocadas aos processos de forma **estática**
- O **número de molduras de página** associadas a cada processo é **fixo**

Política de alocação global

▪ Algoritmo Page Fault Frequency (PFF)

- Gerenciamento **dinâmico** de alocação de páginas
- Determina **quando aumentar/diminuir** o número de molduras de página

- $PFR = \frac{PF_{lsec} + PFR}{2}$, onde PFR é a taxa de faltas de página e PF_{lsec} é o número de faltas de página no último segundo



Política de substituição de páginas

- **Política de substituição global**

- Permite substituir molduras de página alocadas por **qualquer processo**

- **Política de substituição local**

- Somente pode substituir molduras de página **alocadas pelo próprio processo**

Política de substituição de páginas: exemplo com LRU

Após processo A requisitar acesso à página A6

Estado original da memória

Páginas do processo A	A0	10	Instante de último uso
	A1	7	
	A2	5	
	A3	4	
	A4	6	
	A5	3	
Páginas do processo C	B0	9	Política local
	B1	4	
	B2	6	
	B3	2	
	B4	5	
	B5	6	
	B6	12	
	C0	3	
	C1	5	
	C2	6	

Substituição local

A0
A1
A2
A3
A4
A6
B0
B1
B2
B3
B4
B5
B6
C0
C1
C2

Substituição global

A0
A1
A2
A3
A4
A5
A6
B0
B1
B2
B3
B4
B5
B6
C0
C1
C2

Política de substituição de páginas

- **Podem funcionar com políticas local e global**
 - FIFO, Clock, NRU, LRU, NFU e Aging
- **Somente com política local**
 - WSClock
 - Não existe um **working set global** do sistema

2 Controle de carga

O que acontece se os working sets de todos os processos em execução excederem o limite da memória RAM?

- Processos vão causar **faltas de página a todo momento** ➡ **thrashing**
- **PFF**: indicará que **vários processos precisam de mais memória**, porém **nenhum processo** precisará de **menos memória**
- Nesse caso, não há como alocar mais memória para um processo **sem prejudicar os outros!**

- **Solução:**

1. Swap-out de **alguns processos** para o disco para **liberar molduras de página**
2. **Alocar** as molduras de página aos **processos que precisam de mais memória**
3. **Repetir os passos 1 e 2 até que o problema de thrashing seja resolvido**

- Escolhas devem ser baseadas nos **perfis de processos**

- **CPU-bound vs. I/O-bound**
- Fazer swap-out de muitos **processos CPU-bound** pode deixar a CPU ociosa

3

Tamanho de páginas

Tamanho de páginas

- O tamanho das páginas é um **parâmetro** que pode ser escolhido pelo SO
 - Por exemplo, se o hardware foi desenvolvido para trabalhar com **páginas de 4 KB**, o SO pode considerar **pares de páginas como sendo páginas de 8 KB**
 - Para isso, o SO deverá sempre alocar **duas molduras de página consecutivas** na memória
- O tamanho das páginas tem um **impacto importante** em todo o sistema
 - **Não há um tamanho ideal único** para todos os tipos de sistema

Tamanho de páginas

Páginas grandes

Menos faltas de página

Mais fragmentação interna

Tabelas de páginas com **menos** entradas

Programas ocupam **mais** memória,
mesmo tendo um working set pequeno

Páginas pequenas

Mais faltas de página

Menos fragmentação interna

Tabelas de páginas com **mais** entradas

Maior ocupação de entradas na TLB

4 Compartilhamento

Compartilhamento de páginas

- **Em SOs multiprogramados é comum termos:**
 - Um usuário executando **diversos programas** que usam uma **mesma biblioteca**
 - **Múltiplos usuários** executando o **mesmo programa** ao mesmo tempo
 - Processos **multithreaded** ou que criam processos filhos via **fork ()**
- **Compartilhamento**
 - **Evita** a necessidade de **cópias** de páginas na memória
 - Melhora o **aproveitamento do espaço de armazenamento**

Compartilhamento de páginas

- Páginas com permissão de **somente leitura** (e.g., **segmento de texto** de processos) podem ser **facilmente compartilhadas**
- Páginas contendo **dados dos processos** também podem ser compartilhadas, apesar de ser necessário um **tratamento especial** nesses casos
- **Substituição de páginas compartilhadas** em um processo poderá gerar *page faults* em outros processos

Compartilhamento de páginas

Estratégia de copy on write utilizada no `fork()`

- Os processos pai e filho **compartilham** as molduras de página contendo os segmentos de **texto** e **dados**
- Cada processo possui a sua própria tabela de páginas, mas as entradas de ambas as tabelas **apontam para as mesmas molduras de página**
- As molduras de página são marcadas como **somente leitura**
- Na ocorrência de uma **escrita no segmento de dados**:
 - Uma **violação de proteção** de somente leitura gera uma sinalização ao SO
 - O SO realiza a **cópia da moldura de página**, atualiza a **tabela de páginas** e marca a moldura como **leitura/escrita**

! Obrigado pela atenção!



Dúvidas? Entre em contato:

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

