Curso de Ciência da Computação
Pontifícia Universidade Católica de Minas Gerais

Sistemas Operacionais

Capítulo X - Memória Virtual

Fundamentos

- Memória Virtual - separação da memória lógica da memória física.
  - Apenas parte do programa precisa estar na memória para execução.
  - O espaço de endereçamento lógico pode ser muito maior que o espaço de endereçamento físico.
  - Precisa permitir que as páginas sejam movidas de/e para a memória.
- A memória virtual pode ser implementada via:
  - paginação sob demanda
  - segmentação sob demanda

Paginação sob Demanda

- Trazer uma página à memória apenas quando for necessária.
  - Menos operações de I/O
  - Menos memória necessária
  - Resposta mais rápida
  - Mais usuários

- Página necessária $\Rightarrow$ referência a ela
  - referência inválida $\Rightarrow$ abortar
  - fora da memória $\Rightarrow$ trazer à memória

Bit Válido-Inválido

- A cada entrada da tabela de páginas é associado um bit válido-inválido (1 ⇒ em memória, 0 ⇒ fora da memória)
- Inicialmente, o bit é 0 em todas as entradas.
- Exemplo de tabela de página

Quadro #	Bit válido-inválido
	1
	1
	1
	1
	0
M	
	0
	0

Tabela de páginas

- Durante a tradução de endereço, se o bit na tabela é 0 ⇒ falta de página.

Falta de Página

- A primeira referência a uma página gera um trap para o S.O. $\Rightarrow$ page fault
- S.O. olha em outra tabela para decidir:
  - Referência inválida $\Rightarrow$ abortar.
  - Apenas não está na memória.
- Selecionar quadro vazio.
- Trazer página para o quadro.
- Alterar tabelas, bit de validação = 1.
- Instrução de recomeço

E se não houver quadros livres?

- Troca de página - encontrar alguma página na memória que não esteja em uso e a trocar.
  - algoritmo
  - performance - queremos um algoritmo que resulte em mínimo número de faltas de página.

- Mesma página pode ser trazida à memória várias vezes.

Performance de Paginação sob Demanda

- Taxa de falta de página $0 \leq p \leq 1.0$
  - se $p = 0$ não há falta de páginas
  - se $p = 1$, toda referência é uma falta
- Tempo de Acesso Efetivo (EAT)

$$\begin{aligned} EAT = {} & (1 - p) \times \text{acesso à memória} \\ & + p \ (\text{overhead de falta de página} \\ & + [\text{tirar a página}] \\ & + \text{trazer nova página} \\ & + \text{overhead de reinício}) \end{aligned}$$

Paginação sob Demanda - exemplo

- Tempo de acesso à memória = 1 microssegundo
- 50% do tempo, a página que está sendo trocada foi modificada e precisa ser removida.
- Tempo de Swap de Página = 10 ms = 15,000 microseg

$$EAT = (1 - p) \times 1 + p\ (15000)$$

$$1 + 14999p \qquad \text{(em microseg)}$$

Substituição de Página

- Usar um *bit de modificação* (*dirty bit*) para reduzir o overhead da transferência de página - apenas páginas modificadas são escritas em disco.

- Substituição de página completa a separação entre memória lógica e física - grande memória virtual pode ser disponibilizada em uma pequena memória física.

Algoritmos de Substituição de Página

- Desejam a menor taxa de falta de página.
- Avaliar os algoritmos, executando-os em uma seqüência particular de referências de memória e computando o número de faltas.
- Nos nossos exemplos, a seqüência é

  1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5.

Algoritmo First-In-First-Out (FIFO)

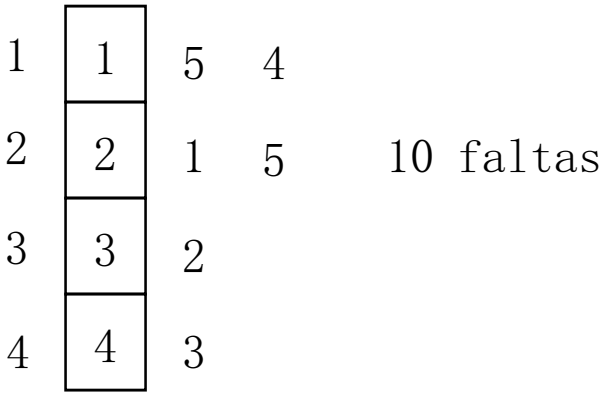
- Seqüência: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5
- 3 quadros (3 páginas na memória por processo em determinado instante)

9 faltas

Algoritmo First-In-First-Out (FIFO)

- anomalia de Belady:
  - mais quadros $\Rightarrow$ mais faltas

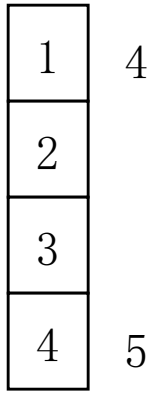
- Seqüência: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5
- 4 quadros



Algoritmo Ótimo

- Substituir páginas que não serão usadas por um maior período de tempo.
- 4 quadros

1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5



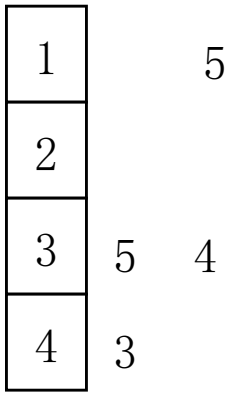
6 faltas

Como saber isto?

- Usado para medir a performance dos algoritmos

Algoritmo Least Recently Used (LRU)

- Seqüência: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5



- Implementação por contadores
  - Cada entrada de página tem um contador; cada vez que a página é referenciada, copiar o clock para o contador
  - Quando uma página precisa ser removida, os contadores decidem qual.

Algoritmo Least Recently Used (LRU)

- Implementação por Pilha - manter uma pilha de números de páginas duplamente encadeada:
  - Referência a página:
    - mover para o topo
    - requer mudança nos apontadores
  - Não é necessário busca para retirar a página LRU

Aproximações do LRU

- Bit de referência
  - Associar um bit a cada página, inicialmente = 0
  - Bit é 1 quando página é referenciada
  - Substituir a que tiver bit 0. Não se sabe a ordem, no entanto.
- Segunda chance
  - Precisa do bit de referência - fila circular.
  - Se a página a ser substituída (sentido horário) tem bit=1, então:
    - atribuir 0 ao bit.
    - Deixar a página na memória.
    - Substituir a próxima página (sentido horário), sujeito às mesmas regras.

Algoritmos de Contagem

- Manter um contador do número de referências feitas a cada página.
- Algoritmo LFU: substituir página com menor contador.
- Algoritmo MFU: baseia-se no argumento de que a página com menor contador provavelmente acabou de ser trazida à memória e ainda será usada.

Alocação de Quadros

- Cada processo precisa de um número mínimo de páginas.
- Dois esquemas básicos de alocação.
  - alocação fixa
  - alocação prioritária

Alocação Fixa

- Alocação igual - ex., se 100 quadros e 5 processos, dar a cada 20 páginas.
- Alocação Proporcional - Alocar de acordo com o tamanho do processo.
  - $s_i =$ size of process p_i
  - $S = \sum s_i$
  - $m =$ total number of frames
  - $a_i =$ allocation for $p_i = \frac{s_i}{S} \times m$

$m = 64$

$s_i = 10$

$s_2 = 127$

$a_1 = \frac{10}{137} \times 64 \approx 5$

$a_2 = \frac{127}{137} \times 64 \approx 59$

Alocação Prioritária

- Usar alocação proporcional com prioridades no lugar do tamanho.
- Se processo P_i gera uma falta de página,
  - selecione para substituição um de seus quadros
  - selecione para substituição um quadro de um processo com menor prioridade.

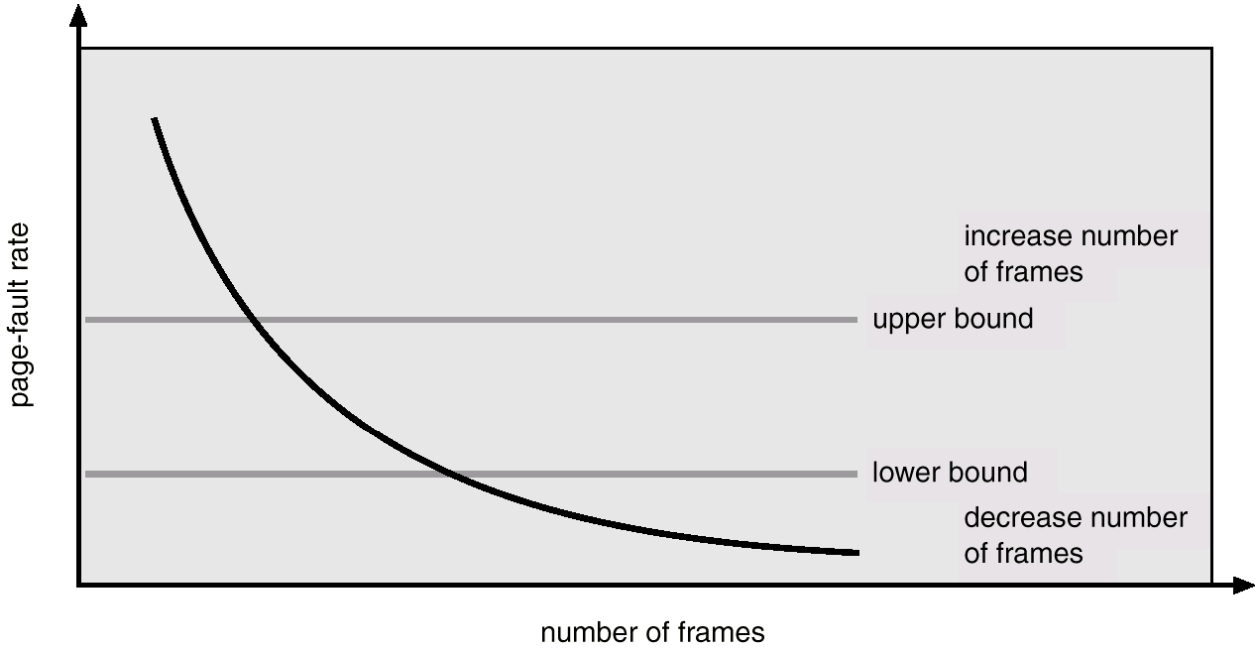
Alocação Global x Local

- Substituição Global - processo seleciona um quadro de substituição do conjunto de quadros; um processo pode tomar um quadro de outro.
- Substituição Local - cada processo seleciona um quadro dos seus próprios.

Thrashing

- Se um processo não tem páginas suficientes, a taxa de falta é muito alta. Isto leva a:
  - baixa utilização de CPU.
  - S.O. pensa que precisa aumentar o grau de multiprogramação.
  - Outro processo adicionado ao sistema.
- Thrashing ≡ um processo mais ocupado movendo páginas que executando

Frequência de Faltas de Páginas



- Estabelecer taxa aceitável
  - Se taxa muito baixa, processo perde quadros.
  - Se taxa muito alta, processo ganha quadros.

Outras Considerações

- Estrutura do programa
  - Arranjo A[1024, 1024] de inteiros
  - Cada linha é guardada em uma página:
    - um quadro para cada linha

  - Programa 1

```
for j := 1 to 1024 do
    for i := 1 to 1024 do
        A[i,j] := 0;
```

    1024 x 1024 faltas

  - Programa 2

```
for i := 1 to 1024 do
    for j := 1 to 1024 do
        A[i,j] := 0;
```

    1024 faltas