



Sistemas Operacionais

Aula 16 - Memória virtual

Professor: Wellington Franco

Introdução

“ O tamanho das memórias está aumentando rapidamente, mas o tamanho dos softwares está aumentando muito rápido. Durante a década de 1980, muitas universidades executavam um sistema de tempo compartilhado com dezenas de usuários simultâneos em um VAX 4MB. Agora, a Microsoft recomenda pelo menos 512MB para um único usuário do sistema Vista. ”

TANENBAUM, A. (2010)

Introdução

- Existem várias estratégias de gerenciamento de memória, em que deve manter simultaneamente muitos processos em memória para permitir multiprogramação.
- É uma técnica que permite a execução de processos que não estão completamente na memória
- Programas podem ser maiores do que a memória física
- Não é fácil de implementar e pode degradar substancialmente o desempenho se for empregada de forma descuidada.

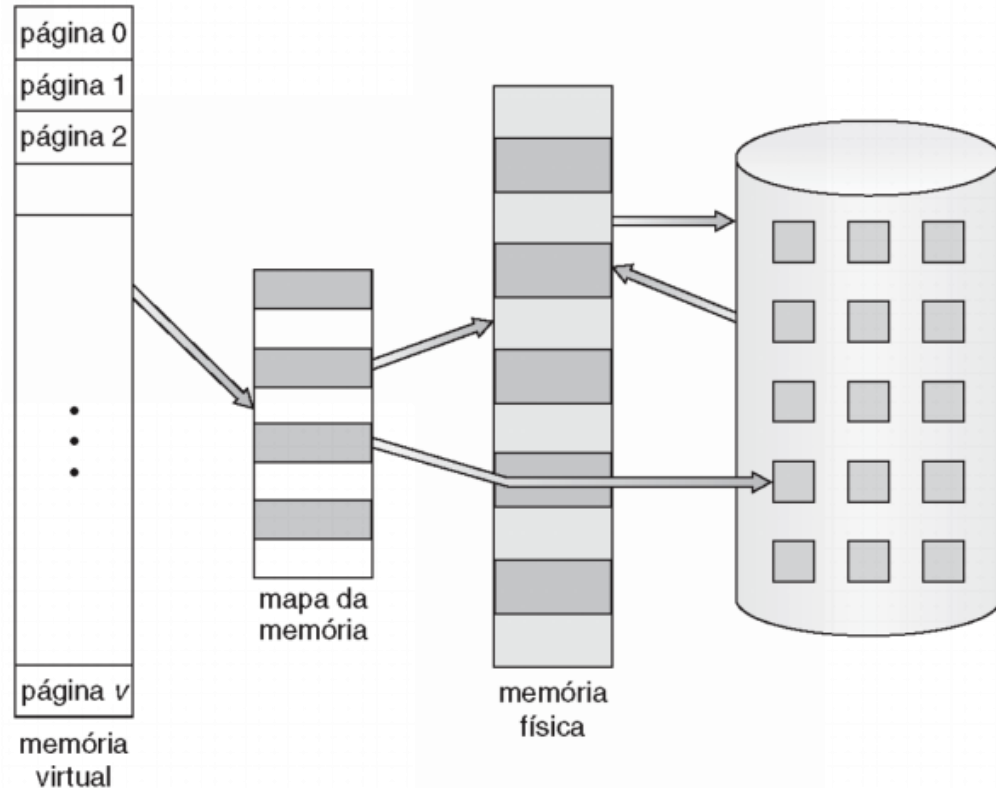
Introdução

- Memória virtual pode ser implementada de duas formas:
 - Paginação sob demanda x Segmentação sob demanda
- As instruções em execução devem estar na memória física. A ideia é que todo o espaço de endereçamento lógico esteja em algum momento no espaço de endereçamento físico
- Tarefa de programar muito mais fácil, porque o programador não precisa mais se preocupar com a quantidade de memória física disponível
 - Se concentra apenas no problema a ser programado

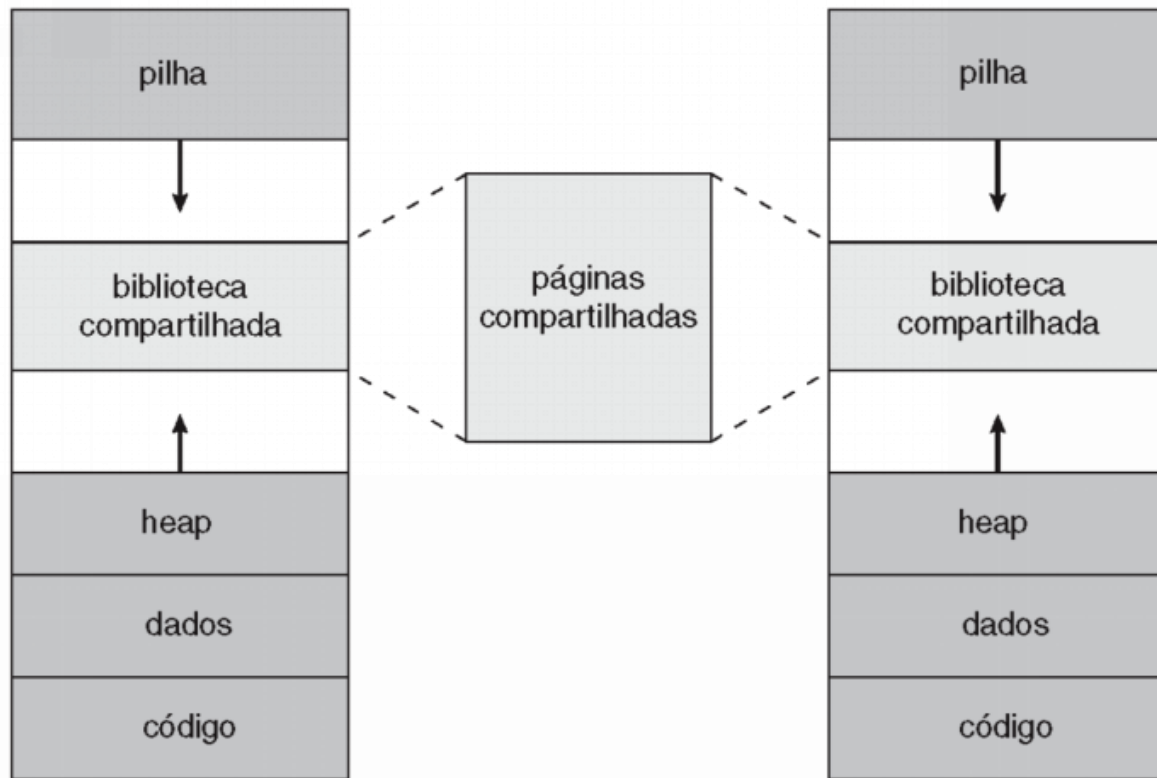
Introdução

- O espaço de endereçamento virtual de um processo diz respeito à visão lógica (ou virtual) de como um processo é armazenado em memória.
 - Começa no endereço 0 e permanece em memória contínua.
- É responsabilidade da MMU mapear páginas lógicas em quadros de páginas físicos em memória.

Memória virtual maior que a memória física



Compartilhamento utilizando memória virtual



Introdução

“ *Em qualquer computador existe um conjunto de endereços de memória que os programas podem gerar ao serem executados. Endereços podem ser gerados com o uso de registradores-base, de indexação, registradores de segmento ou outras técnicas. Esses endereços são chamados de **endereços virtuais**.* ”

TANENBAUM, A. (2010)

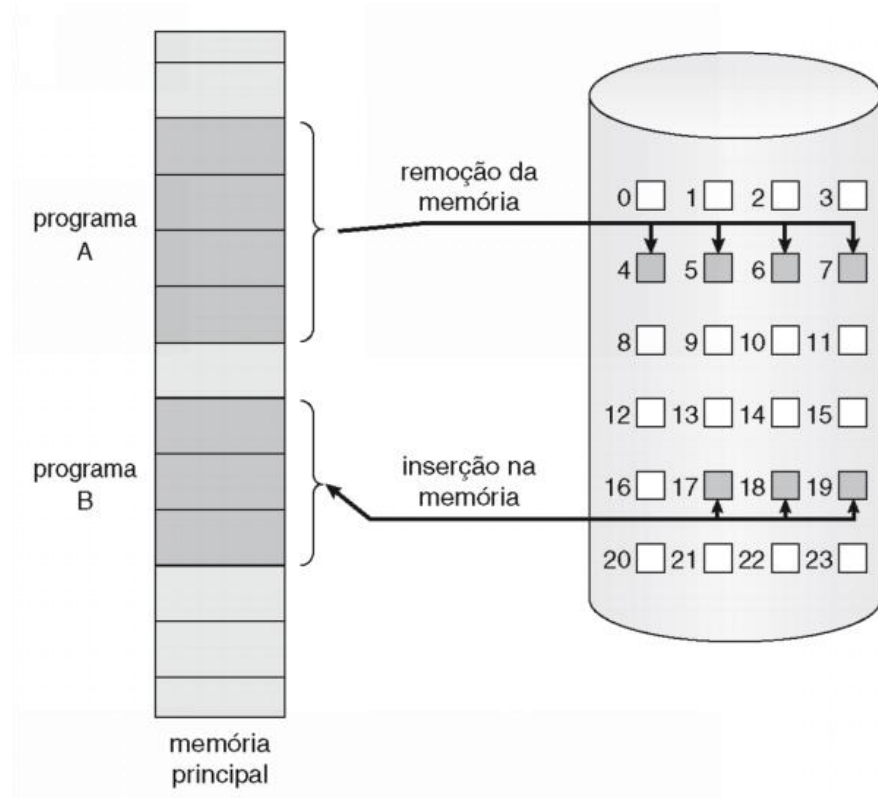
Paginação por demanda

- Como dito, para execução, é necessário que o programa seja transferido do disco para memória.
- Pode-se carregar todo o programa de uma vez
 - Pode não ser necessário ter todo o programa sendo executado no mesmo momento.
- **Uma estratégia é carregar páginas somente quando forem necessárias**
 - Páginas que jamais são acessadas jamais são carregadas na memória física.

Paginação por demanda

- Lazy swapper (mecanismo de permuta preguiçoso) - nunca traz uma página para memória a menos que ela seja necessária (processo)
 - Um Swapper que liga com páginas é chamada **paginador**

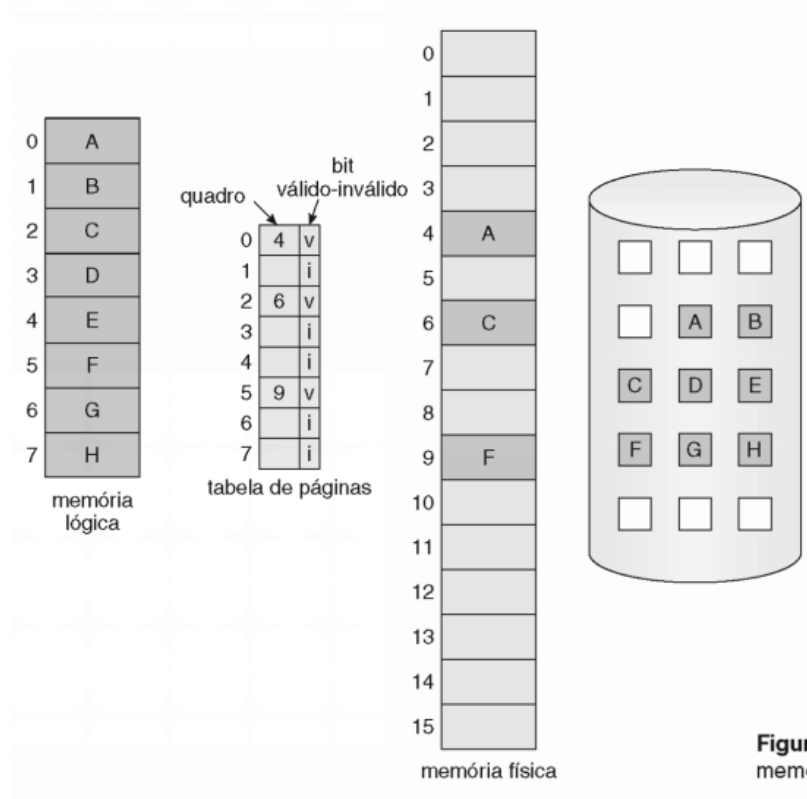
Transferência de memória paginada para disco



Paginação por demanda

- Para distinguir entre páginas que estão na memória e no disco é necessário um suporte de hardware
- Associa-se um bit de validade a cada entrada da tabela de paginação
 - **V** - na memória
 - **I** - fora da memória(ou não faz parte do espaço lógico)
- Página é necessária -> referenciada
 - Referência inválida -> aborta execução
 - Fora da memória -> página é carregada

Paginação por demanda



Paginação por demanda

- Inicialmente todas as entradas são setadas para 1
- Durante a tradução de um endereço, se o bit vale 1 falta de página!
 - Causa uma exceção para o S0

Qual o propósito desses algoritmos?

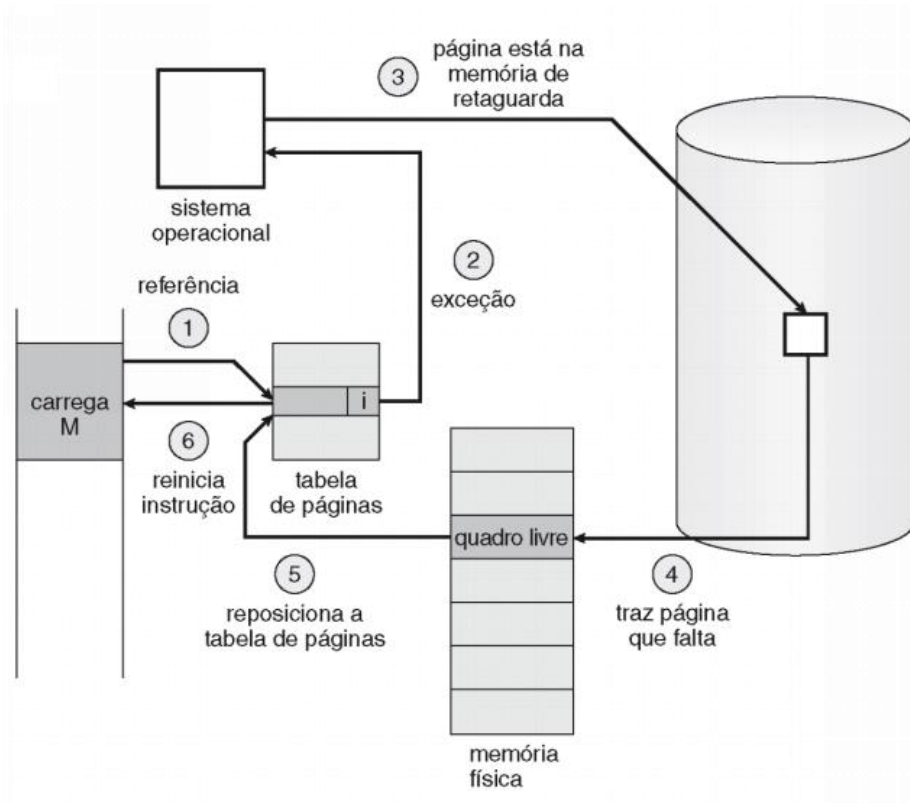
“ Quando ocorre uma falta de página, o sistema operacional precisa escolher uma página a ser removida da memória a fim de liberar espaço para uma nova página a ser trazida para a memória. Se uma página intensamente usada for removida, é provável que logo ela precisa ser trazida de volta, ocasionando custos extras. ”

TANENBAUM, A. (2010)

Falta de página

- A primeira referência a uma página fora da memória causa uma exceção:
 - Falta de página
- O Sistema Operacional usa outra tabela para decidir o que houve (PCB):
 - Referência inválida aborta o processo
 - Apenas uma referência a uma página fora da memória
- Obtém um quadro livre
- Carrega a página no quadro
- Reconfigura as tabelas
- Seta o bit de validade para **V**
- Re-executa a instrução que gerou a falta de página

Algoritmo de falta de página



Desempenho da paginação por demanda

- Pode afetar significativamente o desempenho de um sistema de computação
- É necessário calcular o **tempo de acesso efetivo**. O tempo de acesso à memória varia de 10 a 200 nanosegundos.
- Se não tiver erro de página, o tempo de acesso efetivo (TAE) será igual ao tempo de acesso à memória (TAM)

Desempenho da paginação por demanda

- Taxa de falta de páginas $0 \leq p \leq 1.0$ (probabilidade de ocorrência de erro)
 - se $p = 0$ não há falta de páginas
 - se $p = 1$, toda referência causa uma falta
- $TAE = (1 - p) \cdot TAM + p \cdot TEP$
- $TEP = \text{tempo de erro de página} = \text{tempo para tratar falta de página} +$
tempo de swap out +
tempo de carga da página +
tempo para reiniciar a execução

Desempenho da paginação por demanda

- Exemplo:
- Tempo de acesso à memória = 200 nanossegundos
- Tempo médio para tratamento de uma falta de página = 8 milissegundos
- $$\begin{aligned} \text{TAE} &= (1 - p) \cdot 200 + p \cdot (8 \text{ milissegundos}) \\ &= (1 - p) \cdot 200 + p \cdot 8.000.000 \\ &= 200 + p \cdot 7.999.800 \end{aligned}$$
- Se um acesso a cada 1.000 causa uma falta de página então:
TAE = 8.2 microssegundos. (40 vezes mais lento!!)

Desempenho da paginação por demanda

- Exemplo:
- Se quisermos uma degradação de desempenho menor que 10%?

Substituição de páginas

- **Superalocação:** quando aumenta o grau de multiprogramação do sistema. Tem o objetivo de preencher o máximo de quadros possíveis.
- Se um processo está em execução, ocorre um erro de página. O SO determina onde a página está no disco, mas então descobre que não há quadros disponíveis.

Substituição de páginas

- SO tem diversas opções:
 - Ele pode encerra o processo. Mas é a ideal?
 - Remover o processo da memória
 - Substituição de páginas

Substituição de páginas

- A substituição de páginas adota a seguinte abordagem:
 - Se nenhum quadro estiver livre, encontramos um que não esteja sendo utilizado no momento e o liberamos.
 - Pode-se liberar um quadro gravando seu conteúdo no espaço de permuta e modificando a tabela de páginas.
 - Pode-se então utilizar o quadro liberado para acolher a página pela qual o processo falhou

Substituição de páginas

- Algoritmo que altera o serviço de erros de página para incluir a substituição:
 - Localize a página desejada no disco
 - Localize um quadro livre:
 - Se há um quadro livre, utilize-o
 - Se não há tal quadro, utilizar um algoritmo de substituição para escolher um quadro vítima
 - Grave o quadro vítima no disco, atualize as tabelas de páginas e de quadros
 - Carregue a página desejada no quadro (recentemente) livre; atualize as tabelas de páginas e de quadros
 - Re-execute a instrução que causou a falta de página

Substituição de páginas

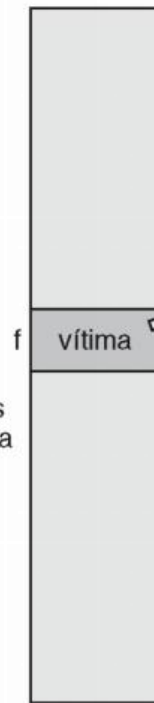
quadro bit válido-inválido

| | |
|---|---|
| | |
| 0 | i |
| f | v |
| | |
| | |

tabela de
páginas

(2) muda
para inválido

(4) reposiciona
tabela de páginas
para a nova página



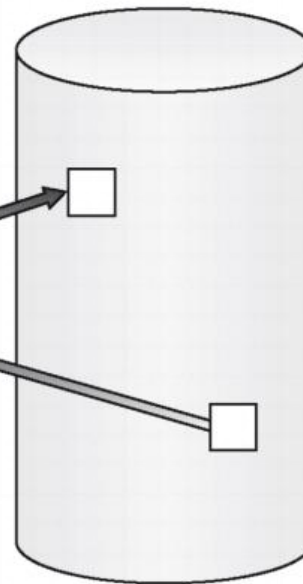
memória
física

remove
página
vítima

(1)

(3) insere
página
desejada

(3)



Substituição de páginas

- Avalia-se um algoritmo utilizando uma sequência de acessos a memória (sequência de referência) e contando a quantidade de faltas de páginas
- Queremos a menor taxa possível de faltas de páginas

Substituição de páginas

- Deve-se levar em conta apenas o número da página em vez do endereço inteiro
 - 0100, 0432, 0101, 0612, 0102, 0103, 0104, 0101, 0611, 0102, 0103, 0104, 0101, 0610, 0102, 0103, 0104, 0101, 0609, 0102, 105
 - 1, 4, 1, 6, 1, 6, 1, 6, 1, 6, 1
 - Para saber o número de erros de páginas, precisa saber o número de quadros disponíveis

Substituição de páginas

- Se tiver três ou mais quadro livres, quantos erros?
- Se tiver um quadro livre?

Substituição de páginas

- Se tiver três ou mais quadro livres, quantos erros?
 - Três erros, sendo um para a primeira referência a cada página
- Se tiver um quadro livre?
 - Onze erros, uma substituição a cada referência
- A medida que aumenta a quantidade de quadros, a quantidade de erros de página cai a um nível mínimo.

Substituição de páginas

- Exemplo:
 - Sequência de referências: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1
 - Quadros livres: 3

Substituição de páginas FIFO

sequência de referência

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

| | | | |
|---|---|---|---|
| 7 | 7 | 7 | 2 |
| | 0 | 0 | 0 |
| | | 1 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| 2 | 2 | 4 | 4 | 4 | 0 |
| 3 | 3 | 3 | 2 | 2 | 2 |
| 1 | 0 | 0 | 0 | 3 | 3 |

| | |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 3 | 2 |

| | | |
|---|---|---|
| 7 | 7 | 7 |
| 1 | 0 | 0 |
| 2 | 2 | 1 |

quadros de páginas

Substituição de páginas FIFO

- Fácil de compreender e programar
- Desempenho nem sempre é bom
 - A página substituída pode ser um módulo de inicialização utilizado há muito tempo e que não é mais necessário
 - Pode conter uma variável muito utilizada que foi inicializada cedo e está em uso constante

Substituição de páginas FIFO

- Exemplo:
 - 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5
 - Se três e quatro quadros disponíveis?

Substituição de páginas FIFO

- Exemplo:
 - 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5
 - Se três e quatro quadros disponíveis
 - Três: nove erros
 - Quatro: dez erros
 - **Anomalia de Belady:** a taxa de erros pode aumentar com o aumento de quadros

Substituição de páginas ótima

- Substitui a página que não será acessada pelo maior período de tempo
- Como obter essa informação?
- Difícil de implementar, mas útil para comparação com outros algoritmos

Substituição de páginas ótima

sequência de referência

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|--|---|--|---|--|---|--|---|--|--|--|--|--|---|--|--|
| 7 | 7 | 7 | 2 | | 2 | | 2 | | 2 | | 2 | | | | | | 7 | | |
| | 0 | 0 | 0 | | 0 | | 4 | | 0 | | 0 | | | | | | 0 | | |
| | | 1 | 1 | | 3 | | 3 | | 3 | | 1 | | | | | | 1 | | |

quadros de páginas

Substituição de páginas LRU

- **Menos recente utilizado** - Aproximação do ótimo
 - Associa a cada página a hora da última vez em que essa página foi utilizada.
 - Seleciona a página que não foi utilizada pelo período de tempo mais longo.
 - De trás ao invés de para frente no tempo

Substituição de páginas LRU

- Sequência de referência: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 5 |
| 2 | 2 | 2 | 2 | 2 |
| 3 | 5 | 5 | 4 | 4 |
| 4 | 4 | 3 | 3 | 3 |

Algoritmo LRU (Least Recently Used)

“ Para uma máquina com n molduras de página, há um hardware auxiliar que contém uma matriz $n \times n$ bits, inicialmente todos com o valor 0. Sempre que a moldura da página k for referenciada, o hardware marcará os bits da linha k com o valor 1 e, em seguida, todos os bits da coluna k com valor 0. A linha que possuir o menor valor binário será a página LRU. ”

TANENBAUM, A. (2010)

Substituição de páginas LRU

- **É usada com frequência e é considerada adequada**
- O problema consiste em determinar uma ordem para os quadros definida pela hora da última utilização

Substituição de páginas LRU

- **Contadores:**
 - Cada página possui um contador que é atualizado com o valor do relógio sempre que a página é referenciada
 - Quando uma página precisa ser substituída, consultar o contador para remover a que possuir o menor valor

Substituição de páginas LRU

sequência de referência

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|--|---|--|---|---|---|---|--|--|---|--|---|--|---|--|--|
| 7 | 7 | 7 | 2 | | 2 | | 4 | 4 | 4 | 0 | | | 1 | | 1 | | 1 | | |
| | 0 | 0 | 0 | | 0 | | 0 | 0 | 3 | 3 | | | 3 | | 0 | | 0 | | |
| | | 1 | 1 | | 3 | | 3 | 2 | 2 | 2 | | | 2 | | 2 | | 7 | | |

quadros de páginas

Substituição de páginas LRU

- **Pilha:**
 - Mantém uma pilha duplamente encadeada com os números das páginas:
 - Página referenciada
 - Move a página para o topo
 - Não requer algoritmo de busca

Substituição de páginas LRU

sequência de referência

4 7 0 7 1 0 1 2 1 2 7 1 2

| |
|---|
| 2 |
| 1 |
| 0 |
| 7 |
| 4 |

pilha
antes
de a

| |
|---|
| 7 |
| 2 |
| 1 |
| 0 |
| 4 |

pilha
após
b

↑
a

↑
b

Substituição de páginas baseada em contagem

- Mantém um contador para o número de referências a cada página
- **Algoritmo LFU:** substitui a página com o menor contador
 - Página muito utilizada deverá apresentar uma contagem alta
- **Algoritmo MFU:** substitui a página com o maior contador
 - Página com menor contagem provavelmente acabou de ser conduzida à memória e ainda será utilizada

Alocação de quadros

- Cada processo precisa de um número mínimo de quadros
- Dois esquemas principais de alocação
 - **Alocação fixa**
 - **Alocação por prioridade**

Alocação fixa

- **Alocação igualitária:** se há **100** quadros e **5** processos, alocar **20** quadros para cada um
- **Alocação proporcional:** quadros são alocados de acordo com o tamanho dos processos

s_i = tamanho do processo p_i

$$S = \sum s_i$$

m = número total de quadros

$$a_i = \text{alocação para } p_i = \frac{s_i}{S} \cdot m$$

$$m = 64$$

$$s_1 = 10$$

$$s_2 = 127$$

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

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

Alocação por prioridade

- Usa uma alocação proporcional baseada na prioridade ao invés do tamanho
- Se o processo P_i causa uma falta de página
 - Seleciona um de seus quadros para substituição
 - Seleciona um quadro de um processo de menor prioridade para substituição

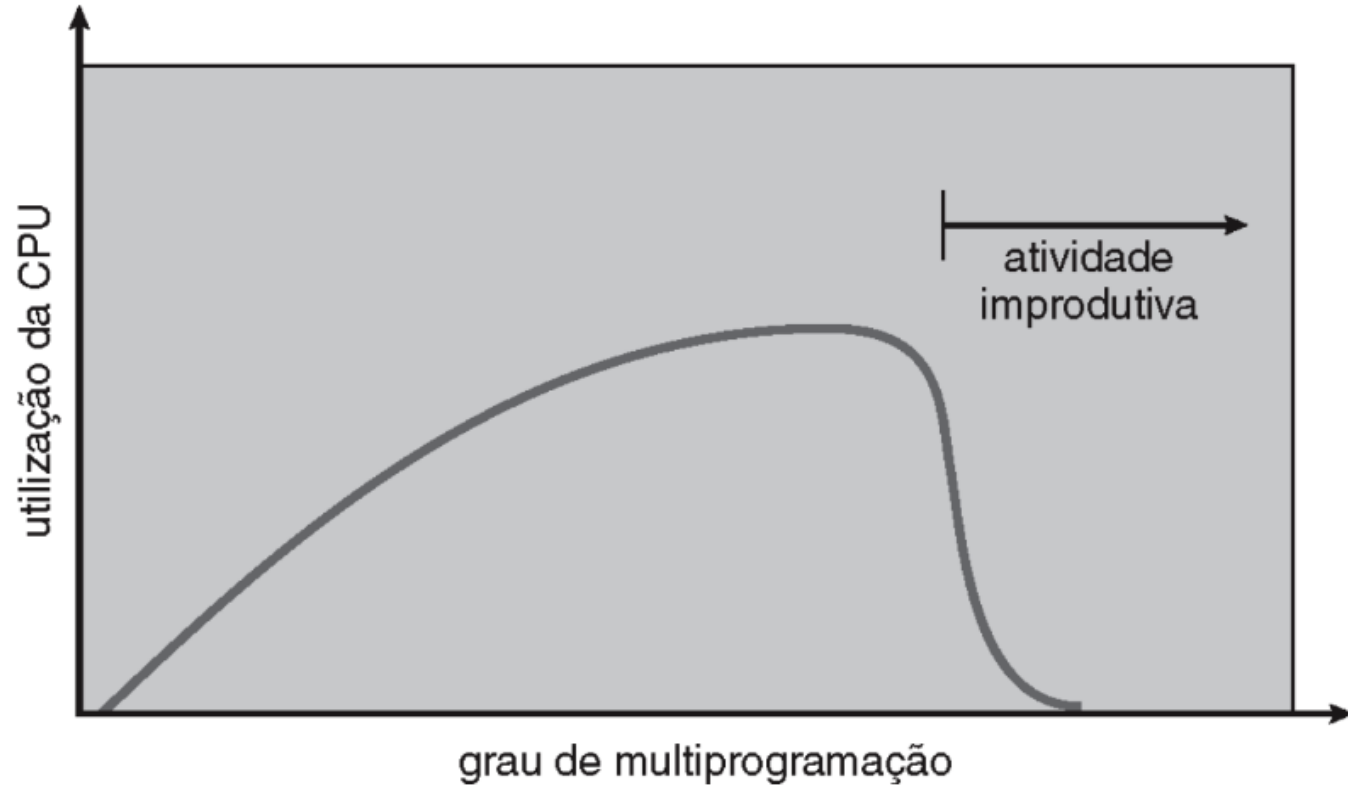
Alocação global x local

- **Substituição global:** o processo seleciona um quadro para substituição de uma lista com todos os quadros de memória
 - Um processo pode tomar um quadro de outro
- **Substituição local:** cada processo só seleciona para substituição os seus próprios quadros

Atividade improdutiva

- Se um processo não tem quadros “suficientes”, a taxa de falta de páginas pode ser muito alta. Isso leva a :
 - Baixa utilização da CPU
 - O Sistema Operacional conclui que precisa aumentar o grau de multiprogramação
 - Um novo processo é adicionado à memória
- **Atividade improdutiva (Thrashing):** um processo fica paralisado com operações de swap in e swap out (os processos estão gastando todo o seu tempo paginando)

Atividade improdutiva



Paginação X Segmentação

| Consideração | Paginação | Segmentação |
|---|--|---|
| O programador precisa saber que essa técnica está sendo usada? | Não | Sim |
| Há quantos espaços de endereçamento linear? | 1 | Muitos |
| O espaço de endereçamento total pode superar o tamanho da memória física? | Sim | Sim |
| Rotinas e dados podem ser distinguidos e protegidos separadamente? | Não | Sim |
| As tabelas cujo tamanho flutua podem ser facilmente acomodadas? | Não | Sim |
| O compartilhamento de rotinas entre os usuários é facilitado? | Não | Sim |
| Por que essa técnica foi inventada? | Para obter um grande espaço de endereçamento linear sem a necessidade de comprar mais memória física | Para permitir que programas e dados sejam divididos em espaços de endereçamento logicamente independentes e para auxiliar o compartilhamento e a proteção |



Dúvidas??

E-mail: wellington@crateus.ufc.br