



Memória Virtual

Prof. Tiago Gonçalves Botelho



Motivação

- Problemas da gerência de memória:
 - Todo espaço lógico mapeado no espaço físico;
 - O tamanho do programa é limitado pelo tamanho da memória;
 - Desperdício de memória por manter armazenado código não utilizado frequentemente:
 - Ex.: rotinas de tratamento de erro.
- Um programa ocupando apenas um espaço de memória que realmente necessitaria permite uma economia substancial de espaço de memória.



Memória Virtual – Conceito

- É a técnica que permite a execução de um processo sem que ele esteja completamente em memória:
 - Separação de vínculo: endereço lógico e endereço físico.
- Princípios básicos:
 - Carregar uma página/segmento na memória principal apenas quando ela for necessária;
 - Paginação por demanda;
 - Segmentação por demanda.
 - Manter na memória apenas páginas/segmentos necessários.



Memória Virtual - Vantagens

- Aumento do grau de multiprogramação;
- Reduz o número de operações de E/S para carga/*swap* do programa;
- Capacidade de executar programas maiores que a capacidade disponível de memória.



Memória Virtual – Princípio da Localidade

- Base de funcionamento da memória virtual:
 - Existe a probabilidade que acessos a instruções e a dados sejam limitados a um trecho.
- Em um determinado instante de tempo apenas esses trechos do processo necessitam estar em memória.

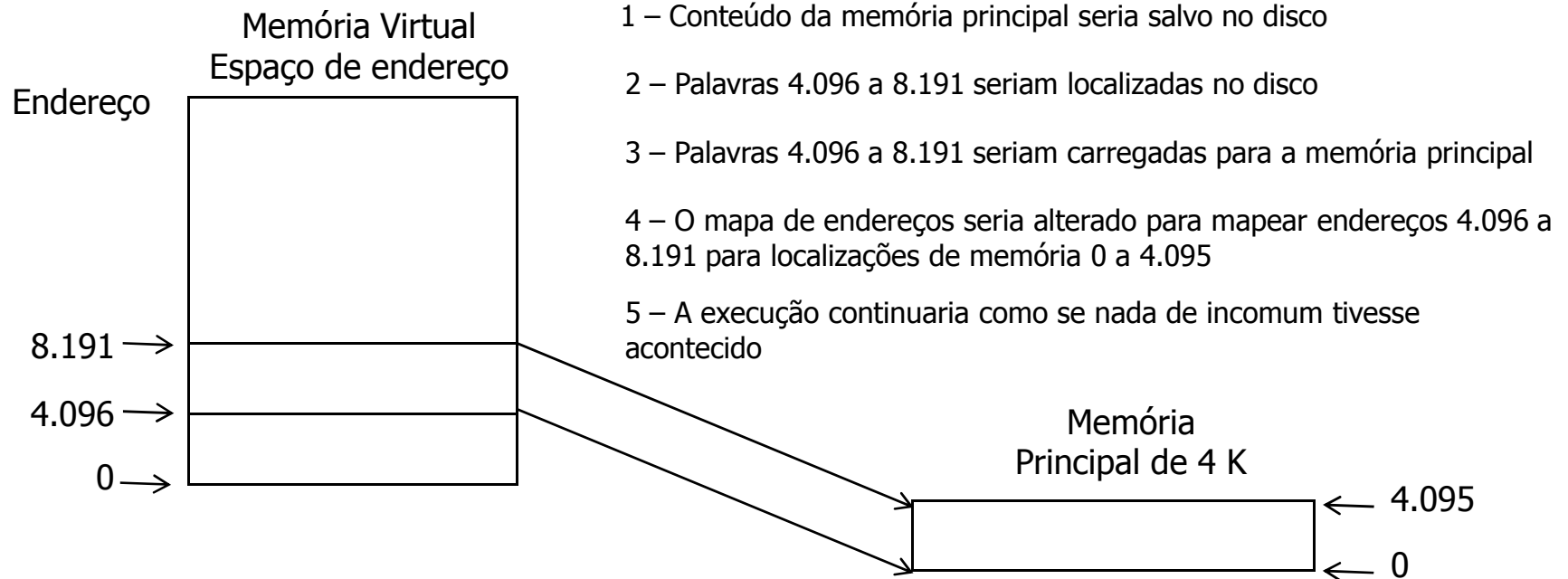


Necessidades para implementação de memória virtual

- Hardware deve suportar paginação e/ou segmentação;
- Sistema operacional deve controlar o fluxo de páginas/segmentos entre a memória secundária (disco) e a memória principal.
- Necessidade de gerenciar:
 - Áreas livres e ocupadas;
 - Mapeamento da memória lógica em memória física;
 - Substituição de páginas/segmentos.

Memória Virtual - Paginação

Mapeamento no qual endereços virtuais 4.096 a 8.191 são mapeados para endereços da memória principal 0 a 4.095





Memória Virtual - Paginação

- A técnica de sobreposição automática é denominada **paginação** e os trechos de programa lidos do disco são chamados **páginas**.
- Uma **tabela de páginas** especifica o endereço físico correspondente para cada endereço virtual.
- Programas são escritos como se houvesse memória principal suficiente para todo espaço de endereço virtual, ainda que não seja esse o caso.
- Geralmente o programador usa algumas características não existentes sem se preocupar como elas funcionam.

Tabela de páginas

Página Endereços Virtuais

| | |
|----|-----------------|
| | |
| 15 | 61.440 – 65.535 |
| 14 | 57.344 – 61.439 |
| 13 | 53.248 – 57.343 |
| 12 | 49.152 – 53.247 |
| 11 | 45.056 – 49.151 |
| 10 | 40.960 – 45.055 |
| 9 | 36.864 – 40.959 |
| 8 | 32.768 – 36.863 |
| 7 | 28.672 – 32.767 |
| 6 | 24.576 – 28.671 |
| 5 | 20.480 – 24.575 |
| 4 | 16.384 – 20.479 |
| 3 | 12.288 – 16.383 |
| 2 | 8.192 – 12.287 |
| 1 | 4.096 – 8.191 |
| 0 | 0 – 4.095 |

(A)

Quadro de Memória principal – 32 K
página Endereços Físicos

| | |
|---|-----------------|
| 7 | 28.672 – 32.767 |
| 6 | 24.576 – 28.671 |
| 5 | 20.480 – 24.575 |
| 4 | 16.384 – 20.479 |
| 3 | 12.288 – 16.383 |
| 2 | 8.192 – 12.287 |
| 1 | 4.096 – 8.191 |
| 0 | 0 – 4.095 |

(B)

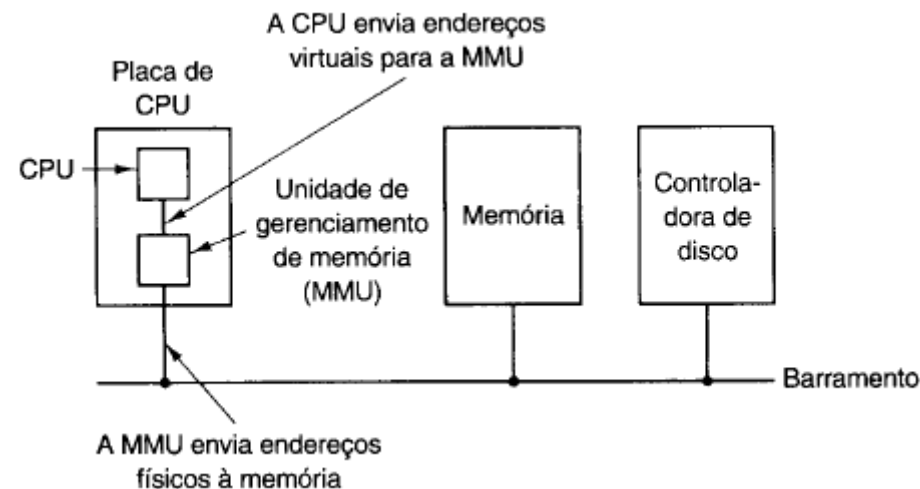


Implementação de paginação

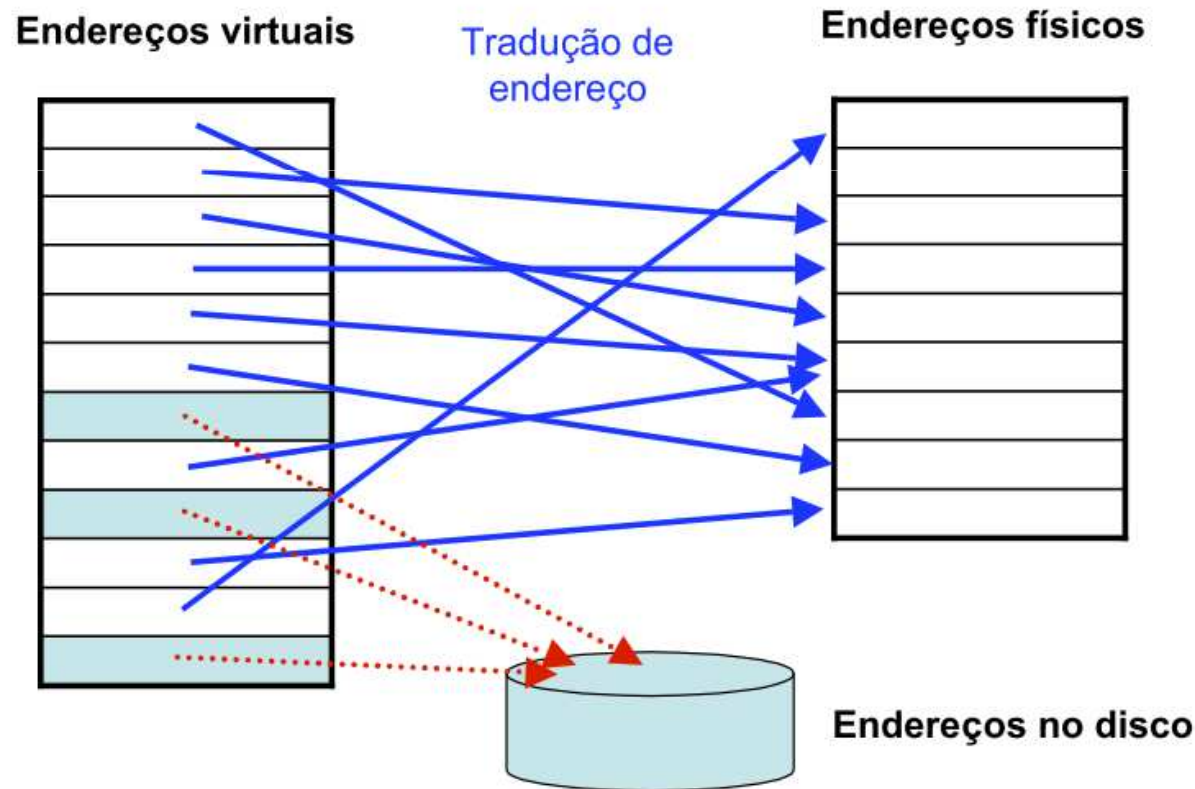
- (A) = Os primeiros 64 KB do espaço de endereço virtual divididos em 16 páginas, cada uma com 4 KB.
- (B) = Memória principal de 32 KB dividida em 8 quadros de página de 4KB cada.

Mapeamento virtual para físico

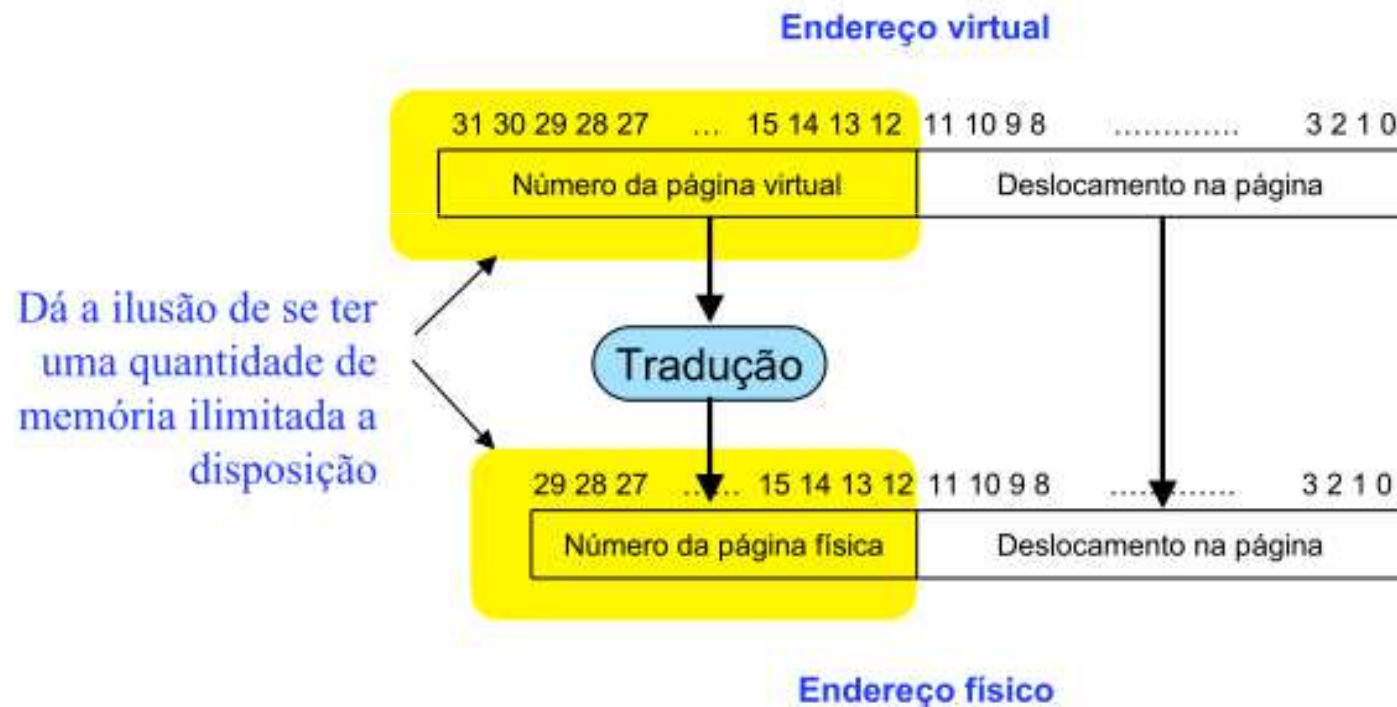
- MMU (*Memory Management Unit*);
- Geralmente está presente no chip da CPU;
- Verificação realizada examinando o bit presente /ausente na entrada da tabela de páginas.



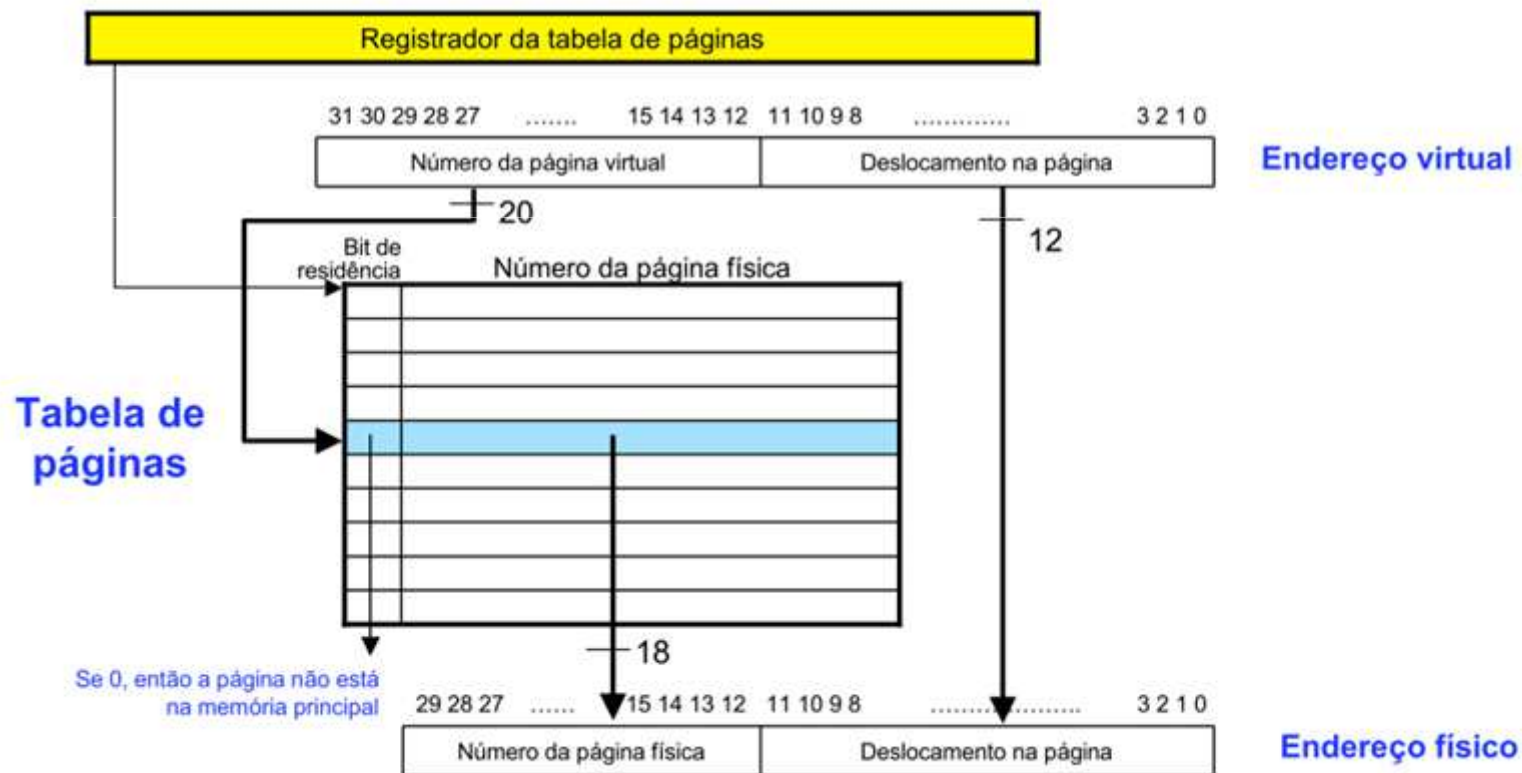
Mapeamento virtual para físico



Mapeamento virtual para físico



Mapeamento virtual para físico via tabela de páginas





Faltas de página

- Bit de residência = 0 indica falta de página;
- Sistema operacional assume o controle, por meio do **mecanismo de exceção**;
- O sistema operacional precisa:
 - Encontrar a página faltante no nível hierárquico inferior (geralmente, no HD);
 - Decidir em que lugar da memória principal deve ser colocada a página requisitada.
- O endereço virtual, por si só, não informa em que posição do HD está a página que gerou a falta de página.



Substituição de páginas na memória

- As páginas físicas podem ficar totalmente ocupadas a medida que páginas lógicas de processos são carregadas;
- O algoritmo de substituição de páginas é responsável pela escolha de uma página “vítima”.



Substituição de páginas na memória

- Bits auxiliares:
- ✓ Objetivo: auxiliar a implementação do mecanismo de substituição de páginas;
 - Bit de sujeira (*dirty bit*): indica se uma página foi alterada durante a execução de um processo.
 - Bit de referência (*reference bit*): indica se uma página foi acessada dentro de um intervalo de tempo.
 - Bit de tranca (*lock bit*): Evita que uma página seja selecionada como "vítima".



Substituição de páginas na memória

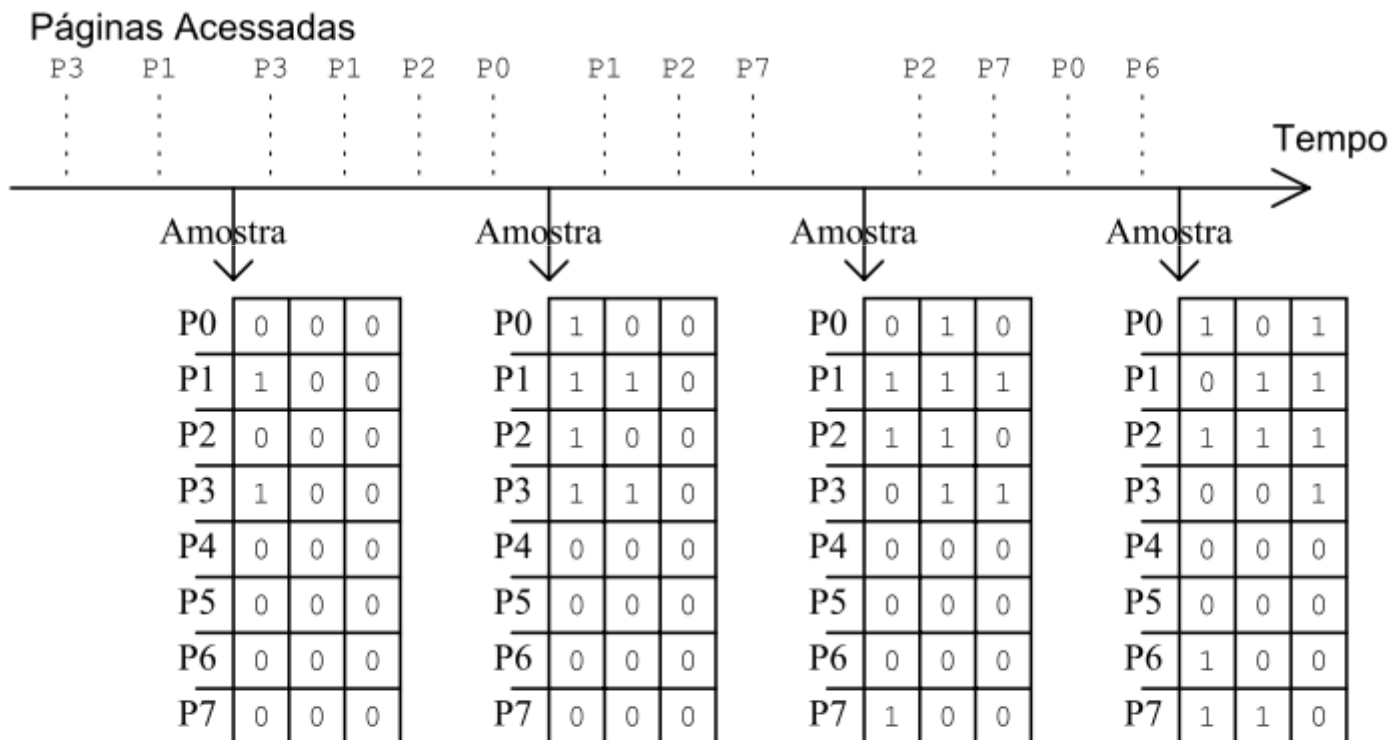
- Selecionar para substituição uma página que será referenciada dentro do maior intervalo de tempo (algoritmo ótimo).
- Algoritmos de substituição de páginas:
 - *First-In, First-Out* (FIFO);
 - *Last Recently Used* (LRU);
 - Baseado em contadores;
 - Algoritmo da segunda chance;

Exemplo: FIFO

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

Substituição de páginas na memória

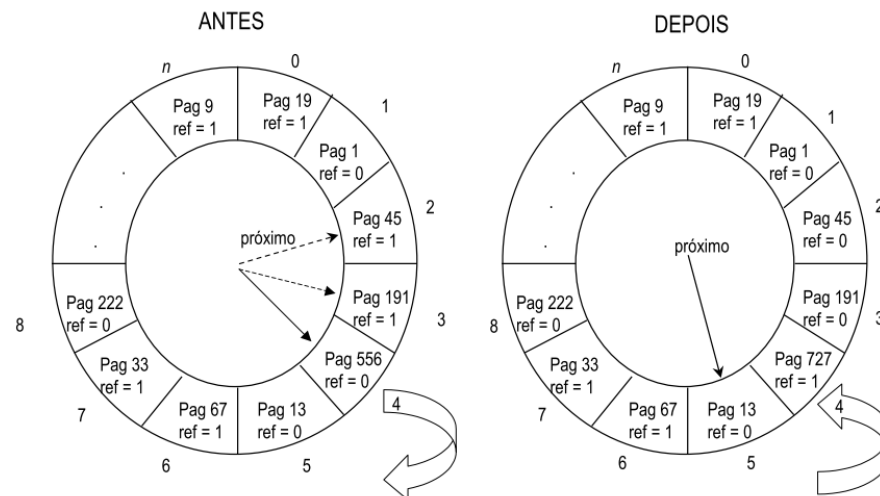
Aproximação para o LRU:



Histórico dos Bits de Referência após cada amostragem

Substituição de páginas na memória

- Algoritmo de segunda chance (aproximação para o FIFO)
 - Baseado no bit de referência
 - Considera que as páginas lógicas formam uma lista circular:
 - Apontador percorre a lista circular informando qual será a próxima “vítima”.
 - Se a página apontada (sentido horário) tem o bit de referência=1, então:
 - * Posiciona o bit de referência em zero e mantém página na memória;
 - * Substitui próxima página que tem bit de referência=0;





Substituição de páginas na memória

- Algoritmo de segunda chance melhorado

- Considera além do bit de referência um bit de modificação (bit de uso);
 - * bit uso = 0: quando a página é carregada na memória;
 - * bit uso = 1: quando a página é modificada.
- Durante pesquisa por página a ser substituída modifica bit de referência de 1 para 0 (algoritmo do relógio)
 - * Privilegia a substituição de frames com bit uso=0;
 - * 4 combinações:
 - ✓ 00: Não referenciada, não modificada;
 - ✓ 01: Não referenciada, porém modificada;
 - ✓ 10: Recentemente referenciada, não modificada;
 - ✓ 11: Recentemente referenciada e modificada.



Substituição de páginas na memória

- Algoritmo segunda chance para MMU sem bit de referência;
- Baseada em uma lista circular de páginas lógicas e um apontador (algoritmo do relógio) e uma lista de páginas físicas livres;
- Em caso de falta de página:
 - Remove uma página física da lista livres para a qual será lida a página faltosa;
 - Página “vítima” é marcada como inválida e incluída na lista de livres;
 - Se essa página “vítima” for acessada logo em seguida ela simplesmente é retirada da lista de livres e inserida na lista de páginas lógicas válidas.



Segmentação

- A segmentação é uma técnica de gerência de memória, onde os processos são divididos logicamente em sub-rotinas e estruturas de dados, e colocados em blocos de informações na memória.
- Os blocos tem tamanhos diferentes e são chamados de segmentos, cada um com seu próprio espaço de endereçamento.
- A grande diferença entre a paginação e a segmentação é que, enquanto a primeira divide a memória em partes de tamanho fixos, sem qualquer ligação com a estrutura do processo, a segmentação permite uma relação entre a lógica do processo e sua divisão na memória.



Segmentação

- Ex.: Um compilador tem muitas tabelas construídas em tempo de compilação, geralmente inclui:
 - ✓ O código fonte;
 - ✓ A tabela de símbolos;
 - ✓ A tabela com todas as constantes;
 - ✓ A árvore sintática;
 - ✓ A pilha usada pelas chamadas de rotina.



Paginação x Segmentação

| Consideração | Paginação | Segmentação |
|--|-------------------------------|--|
| O programador precisa estar ciente dela? | Não | Sim |
| Quantos espaços de endereços lineares há | 1 | Muitos |
| O espaço de endereço virtual pode ser maior do que o tamanho da memória? | Sim | Sim |
| Tabelas de tamanhos variáveis podem ser manipulados com facilidade? | Não | Sim |
| Por que a técnica foi inventada? | Para simular memórias grandes | Para fornecer vários espaços de endereço |



Bibliografia:

- Tanenbaum, A. S. Organização Estruturada de Computadores. 5 ed – Editora Pearson, 2007.
- Tanenbaum, A. S. Sistemas Operacionais. 3 ed – Editora Pearson, 2010.