
Memória Virtual

Prof. Dr. José Luís Zem

Prof. Dr. Renato Kraide Soffner

Prof. Ms. Rossano Pablo Pinto



Faculdade de Tecnologia de Americana

Centro Paula Souza

Tópicos

- Introdução
 - Espaço de Endereçamento Virtual
 - Mapeamento
 - Paginação
 - Segmentação
 - Segmentação com Paginação
 - Proteção
 - Compartilhamento de Memória
 - Swapping em Memória Virtual
 - Trashing
-

Introdução

- ❑ Memória Virtual (virtual memory) é uma técnica bastante poderosa de gerenciamento de memória, onde as memórias principal e secundária são combinadas , oferecendo ao usuário a impressão de existir uma memória muito maior que a memória principal.
 - ❑ O conceito está baseado em desvincular o endereçamento feito pelo processo dos endereços físicos da memória principal
 - ❑ Desta maneira, os processos e suas estruturas de dados deixam de estar limitados ao tamanho da memória física disponível.
-

Introdução

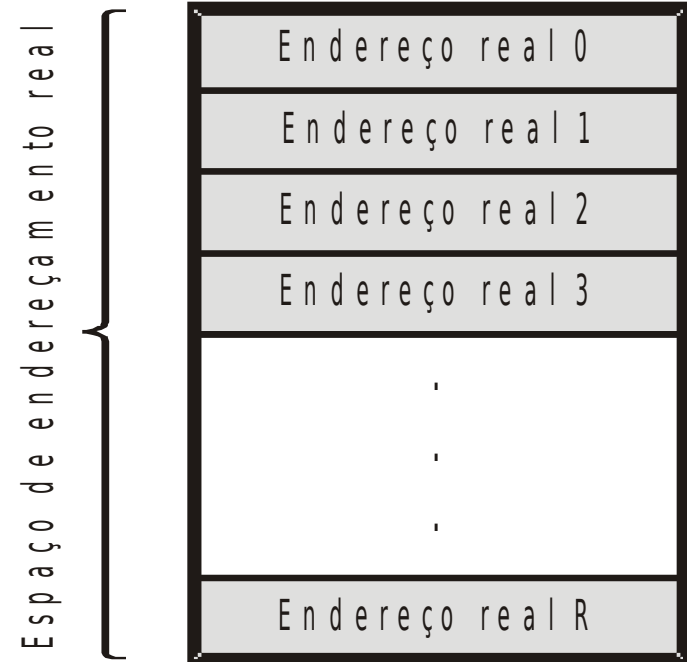
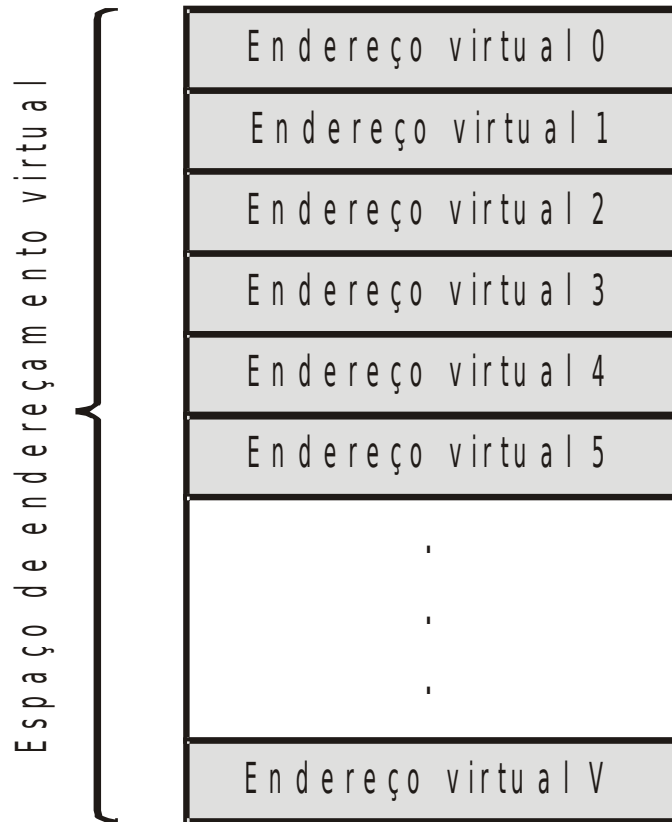
Endereço Físico

5 0 0		V E T [1]
5 0 1		V E T [2]
5 0 2		V E T [3]
5 0 3		V E T [4]
5 0 4		V E T [5]
.	.	.
.	.	.
.	.	.
5 9 9		V E T [1 0 0]

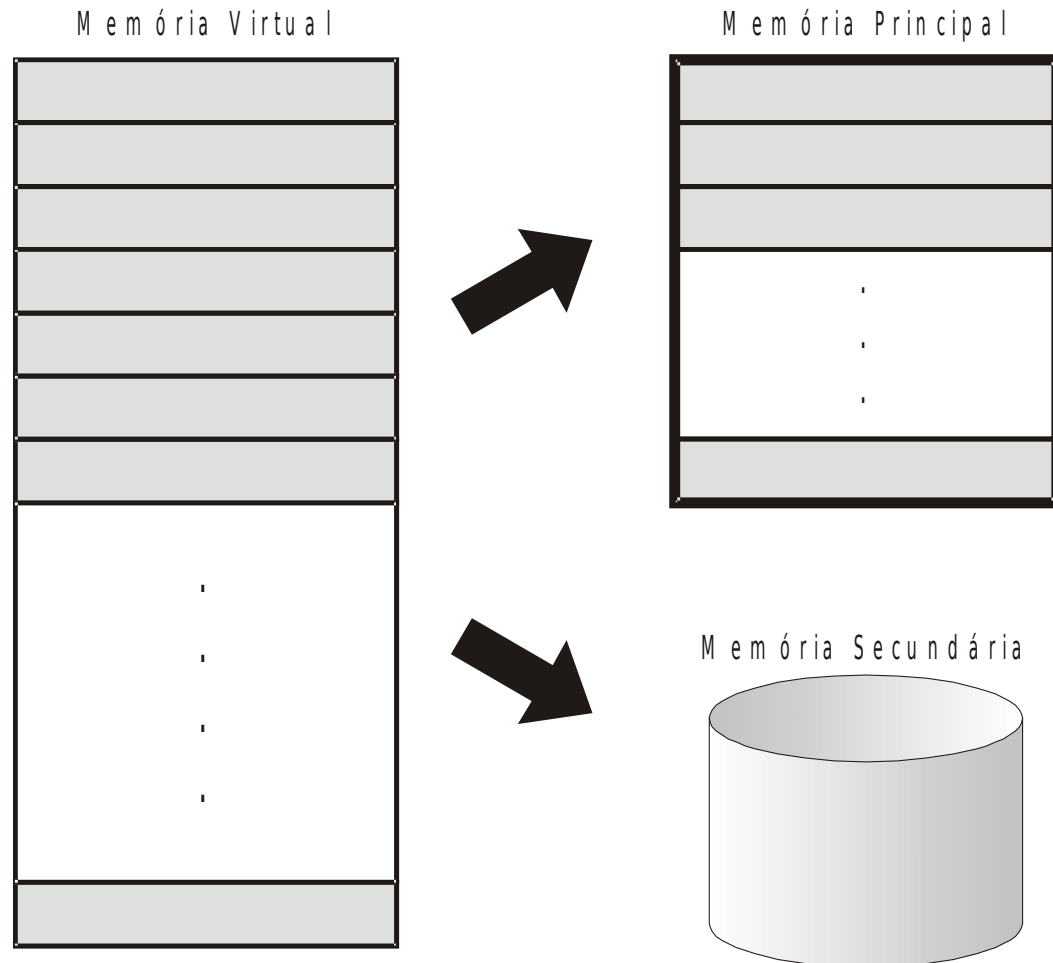
Espaço de Endereçamento Virtual

- ❑ Um programa no ambiente de memória virtual não faz referência a endereços físicos de memória (endereços reais), mas apenas a endereços virtuais.
 - ❑ No momento da execução de uma instrução, o endereço virtual é traduzido para um endereço físico para poder ser atendido pela CPU.
 - ❑ O mecanismo de tradução do endereço virtual para o endereço físico é denominado mapeamento.
 - ❑ O conjunto de endereços virtuais que os processos podem endereçar é chamado espaço de endereçamento virtual. Da mesma maneira, o conjunto de endereços reais é chamado de espaço de endereçamento real.
-

Espaço de Endereçamento Virtual



Espaço de Endereçamento Virtual

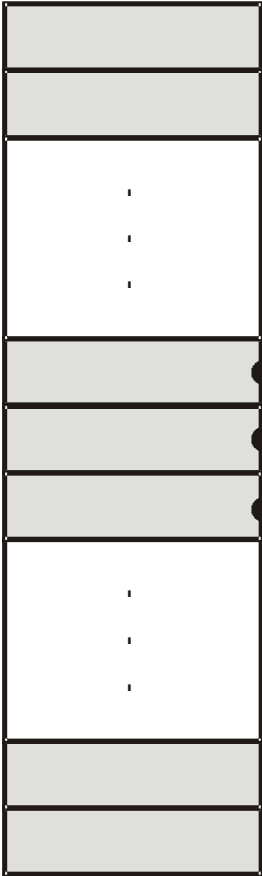


Mapeamento

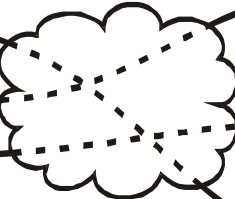
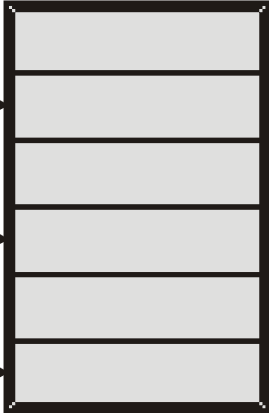
- Toda a estrutura do processo ou programa executado em seu contexto precisa estar no espaço de endereçamento real para poder ser referenciado pelo processador. Portanto, deve existir um mecanismo que transforme endereços virtuais em endereços reais.
 - Cada processo tem o mesmo espaço de endereçamento virtual, como se possuísse memória própria. O mecanismo de tradução se encarrega de manter as tabelas de mapeamento exclusivas para cada processo, relacionando os endereços virtuais do processo às suas posições na memória física.
-

Mapeamento

M e m ó r i a V i r t u a l

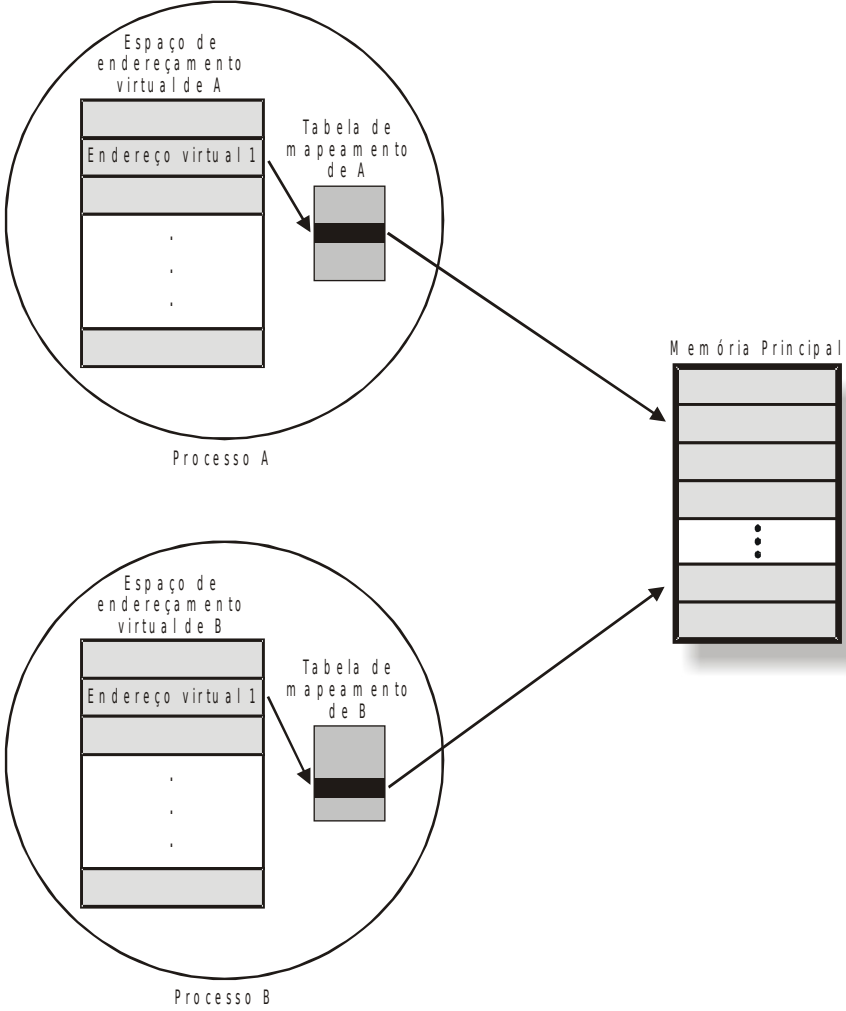


M e m ó r i a P r i n c i p a l



M a p e a m e n t o

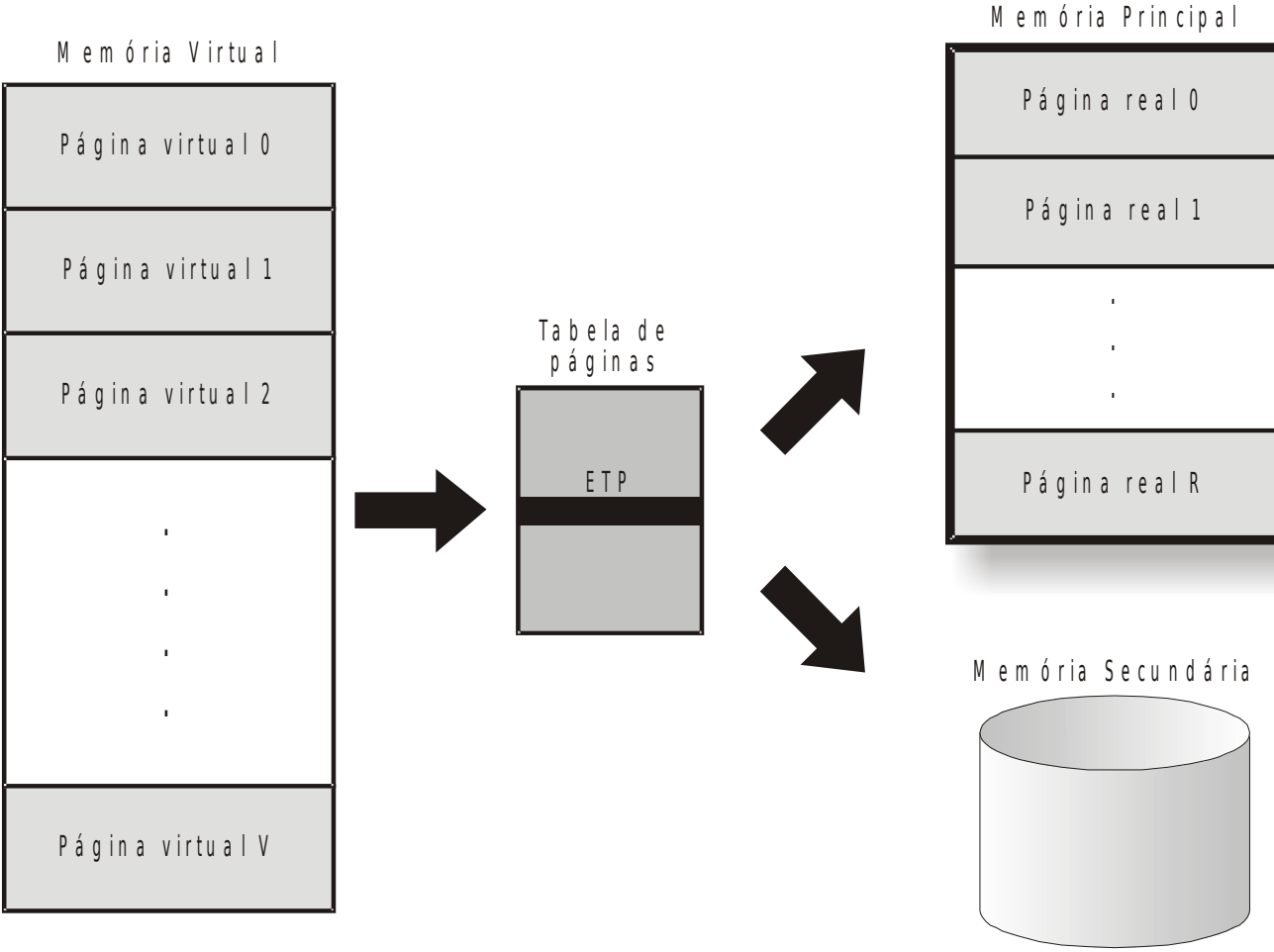
Mapeamento



Paginação

- É a técnica de gerência de memória onde o espaço de endereçamento virtual e o espaço de endereçamento real são divididos em blocos do mesmo tamanho, chamados páginas.
 - As páginas no espaço virtual são denominadas páginas virtuais, enquanto as páginas no espaço real são chamadas de páginas reais ou frames.
 - Todo o mapeamento é realizado em nível de página, através de tabelas de páginas.
 - Cada página virtual do processo possui uma entrada na tabela (Entrada na Tabela de Página - ETP), com informações sobre o mapeamento.
-

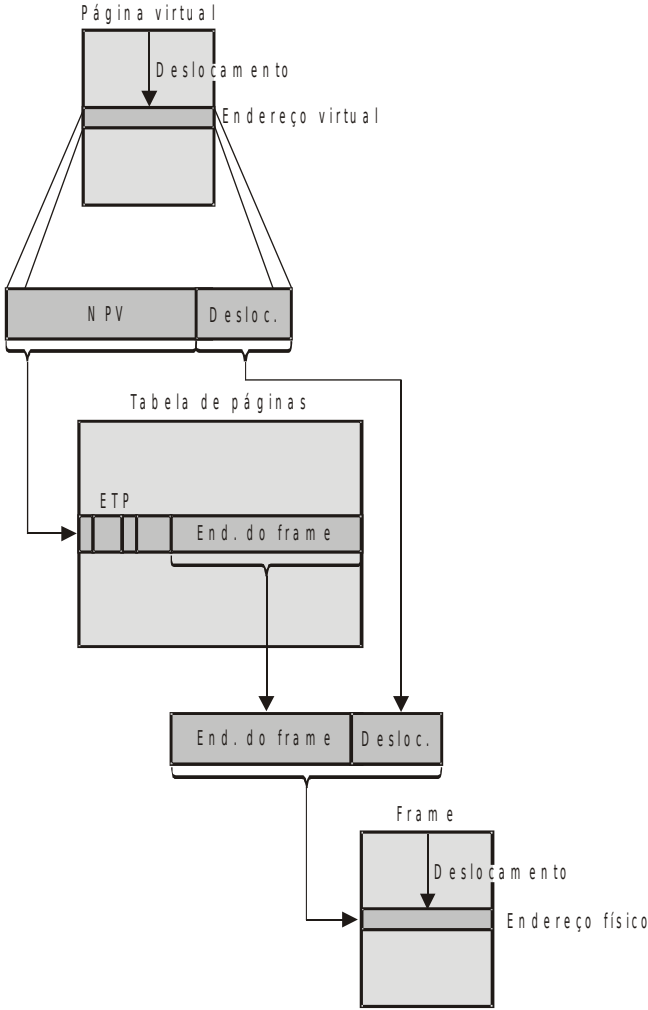
Paginação



Paginação

- Quando um programa é executado, as páginas virtuais são transferidas da memória secundária para a memória principal e colocadas em frames.
 - Sempre que o programa fizer referência a um endereço virtual, o mecanismo de mapeamento localiza, na ETP da tabela do processo, o endereço físico do frame.
 - O endereço virtual é formado pelo número da página virtual e um deslocamento dentro da página.
-

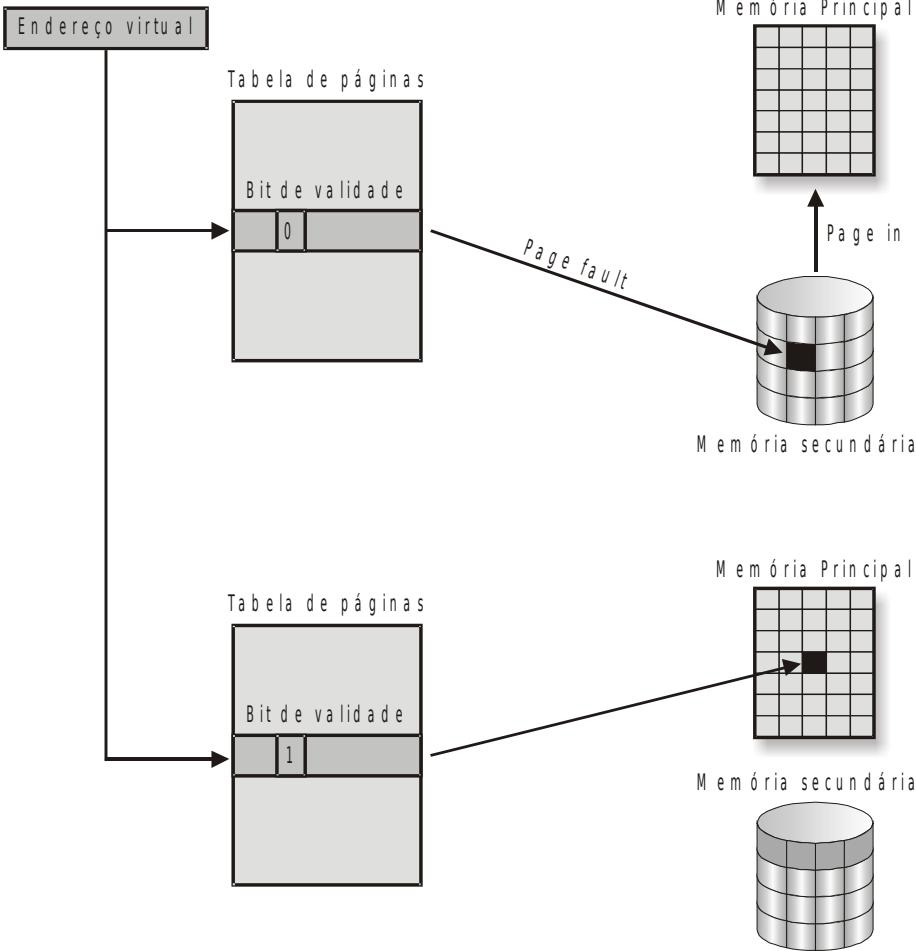
Paginação



Paginação

- Além da informação sobre a localização da página virtual, a ETP possui outras informações.
 - Bit de validade que indica se uma página está ou não na memória física (valid bit ou residence bit).
 - Bit de indicação se uma página sofreu alteração após ser carregada (modify bit ou dirty bit).
-

Paginação



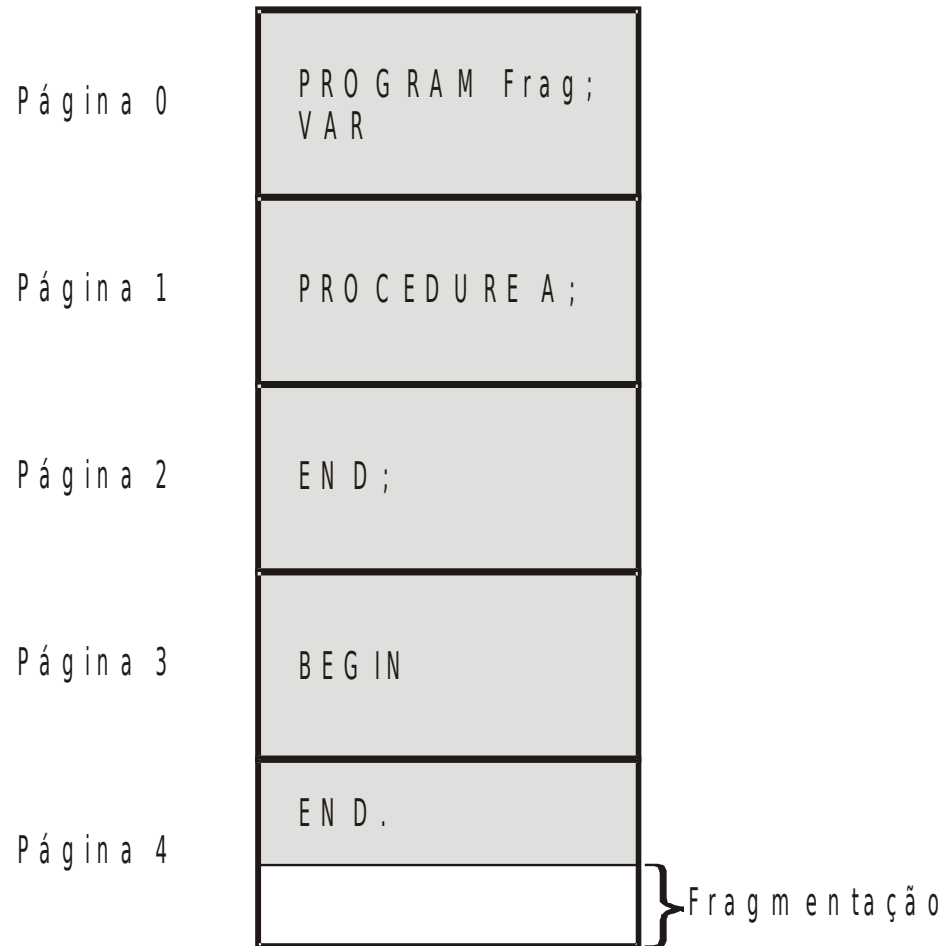
Paginação

- Paginação por Demand (Demand Paging)
 - As páginas dos processos são transferidas da memória secundária para a principal apenas quando são referenciadas.
 - Paginação por Antecipação (Anticipatory Paging)
 - As páginas próximas àquela referenciada também são carregadas, simulando uma previsão de quais páginas serão necessárias, trazendo-as antecipadamente para a memória.
-

Paginação

- ❑ O problema da fragmentação também existe em sistemas pagináveis.
 - ❑ A fragmentação só é encontrada, realmente, na última página, quando o código não a ocupa por completo.
 - ❑ A maior ou menor fragmentação é consequência do tamanho da página.
-

Paginação



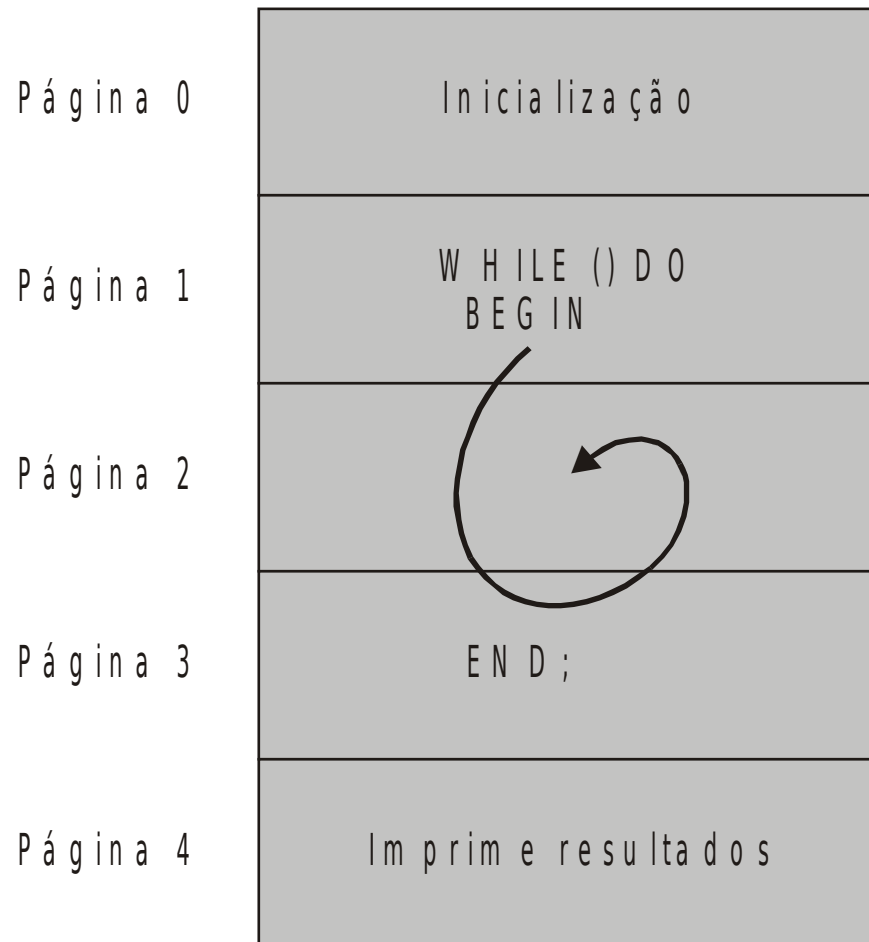
Paginação

- Páginas pequenas:
 - necessitam de tabelas de mapeamento maiores;
 - provocam maior taxa de paginação;
 - aumentam o número de acessos à memória secundária;
 - geram menor fragmentação.
 - O tamanho da página varia de sistema para sistema, mas normalmente está entre 512 bytes a 4 Kbytes.
-

Paginação - Working Set

- ❑ O conceito de Working Set surgiu a partir da análise da taxa de paginação dos processos.
 - ❑ Quando um programa iniciava sua execução, percebia-se uma elevada taxa de page faults, que se estabilizava com o decorrer de sua execução.
 - ❑ Localidade é a tendência que existe em um programa de fazer referências às posições de memória relativamente próximas.
-

Paginação - Working Set



Paginação - Working Set

- ❑ A localidade tem um forte relacionamento com a maneira com que a aplicação foi escrita.
 - ❑ Se um programa foi desenvolvido utilizando-se as técnicas estruturadas, o conceito de localidade estará presente com certeza.
 - ❑ Working Set é o conjunto de páginas que um processo referencia constantemente, e por isso deve permanecer na memória principal.
 - ❑ Caso contrário, aumenta-se a paginação, comprometendo o desempenho.
-

Paginação - Working Set

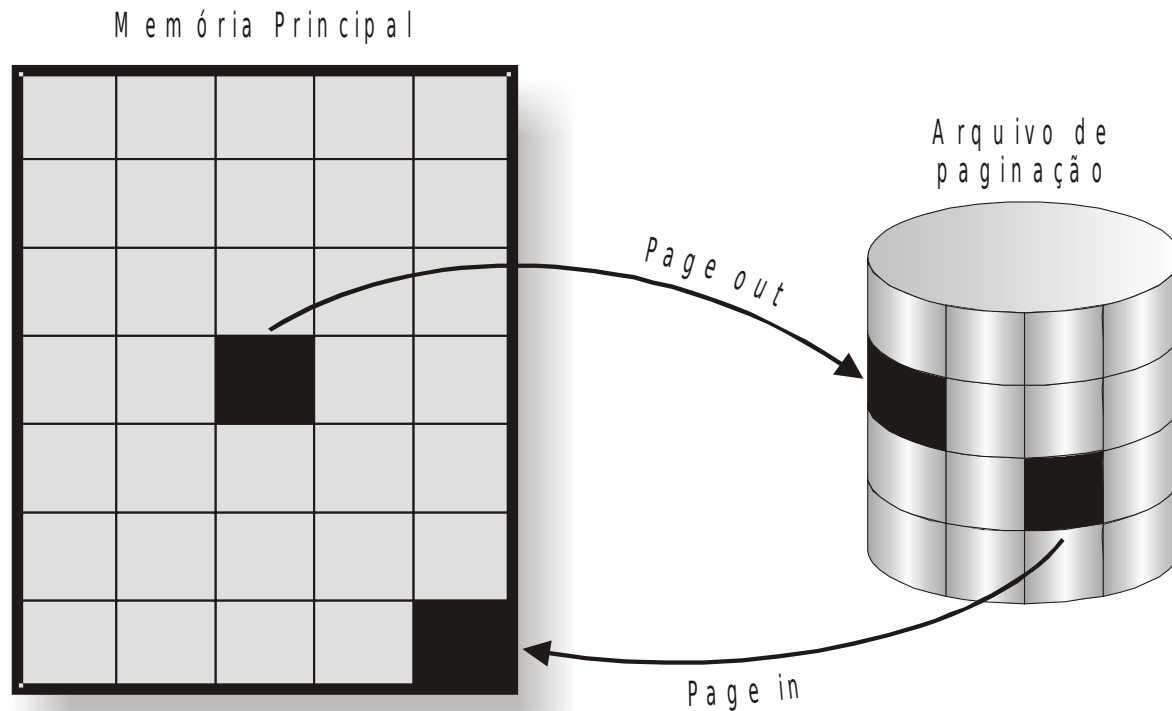
- ❑ O Working Set deve possuir um limite de páginas.
 - ❑ Quanto maior for o Working Set, menor será a probabilidade de ocorrerem page faults.
 - ❑ Quanto maior o Working Set, maior o consumo de memória e menor a quantidade de processos carregados na memória principal ao mesmo tempo, conseqüentemente, menor o grau de multiprogramação.
-

Paginação - Realocação de Páginas

- ❑ O maior problema na gerência de memória virtual por paginação não é decidir qual página carregar para a memória, mas sim, quais páginas remover.
 - ❑ Qualquer estratégia de realocação de página deve considerar se uma página foi ou não modificada, antes de liberá-la para outro processo.
 - ❑ Possíveis alterações feitas na página quando ocupada podem ser perdidas.
-

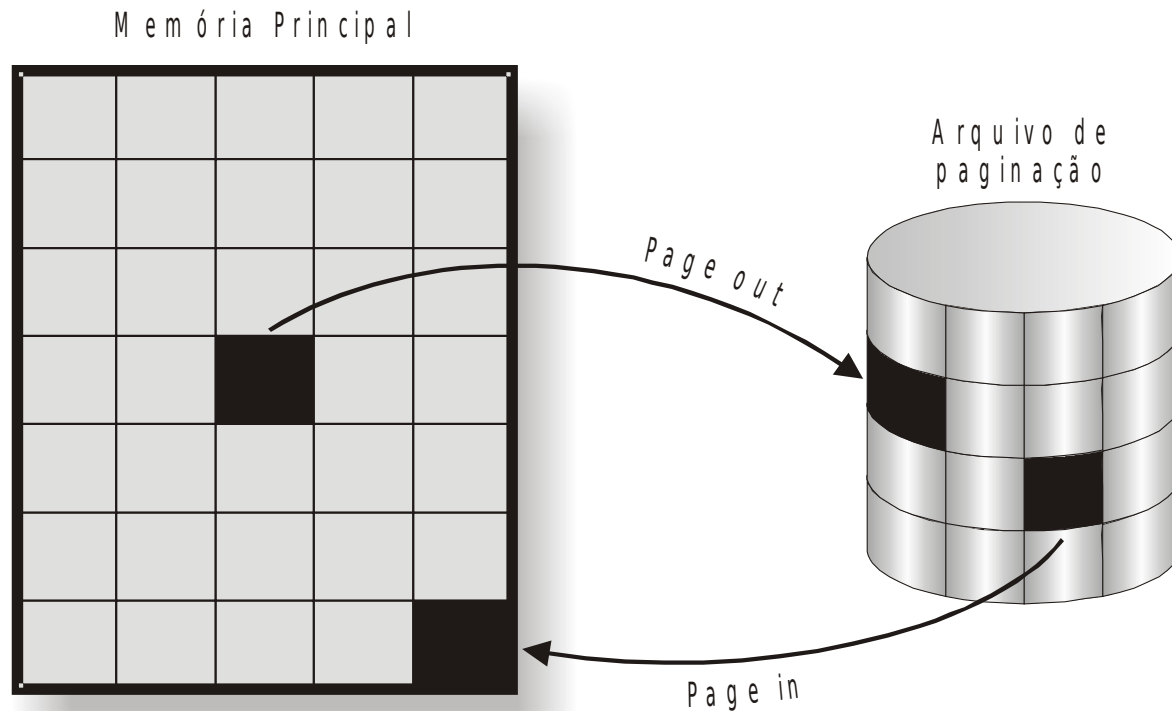
Paginação - Realocação de Páginas

- O sistema possui um arquivo de paginação (page file) onde as páginas modificadas ficam armazenadas.



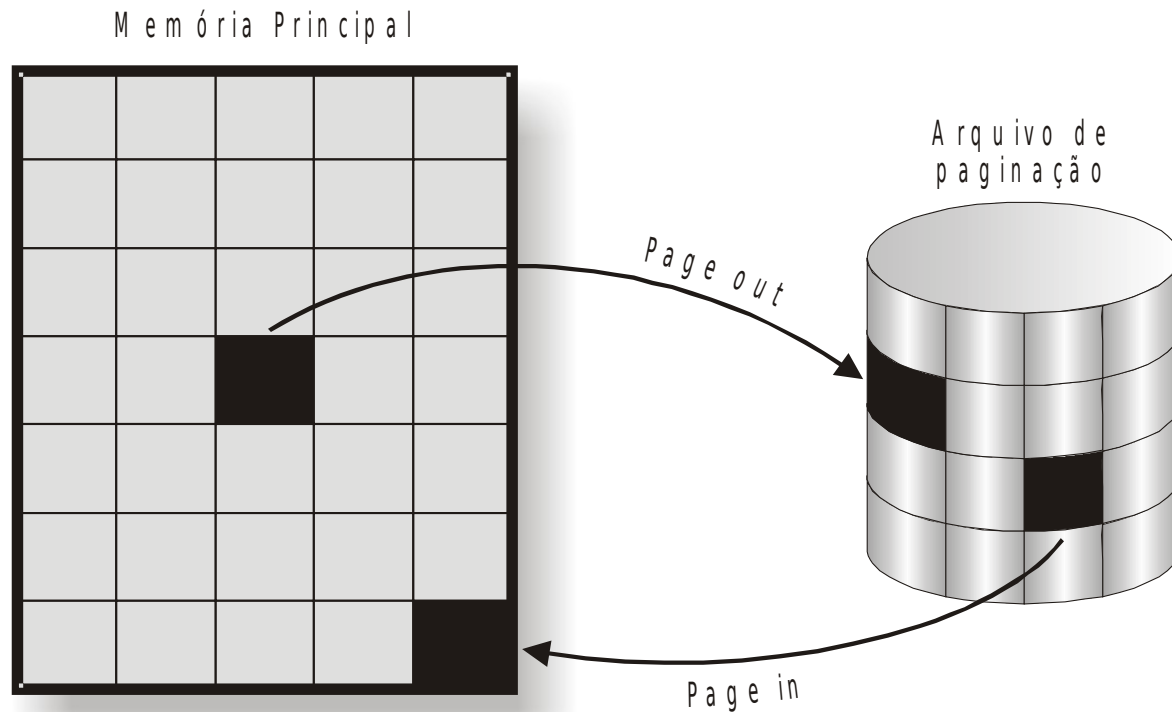
Paginação - Realocação de Páginas

- Sempre que o sistema liberar uma página, deverá gravá-la no arquivo de páginas (page out), preservando o seu conteúdo.



Paginação - Realocação de Páginas

- Quando uma destas páginas alteradas for referenciada novamente, ela será recuperada do arquivo de páginas e trazida para o Working Set do processo (page in).



Paginação - Realocação de Páginas

- ❑ O sistema sabe se uma página foi alterada ou não através do bit de modificação (modify bit ou dirty bit), que existe na ETP de cada tabela de página.
 - ❑ A melhor estratégia de realocação seria aquela onde a escolha caísse sobre uma página que não fosse referenciada num futuro próximo, porém o SO têm dificuldade em prever isso.
 - ❑ Assim, algumas estratégias são adotadas pelos SOs para a realocação de páginas.
-

Paginação - Realocação de Páginas

- Aleatória (RANDOM)
 - Esta escolha não utiliza critério algum para a seleção.
 - Todas as páginas do working set tem as mesmas chances de serem selecionadas, inclusive páginas que são freqüentemente referenciadas.
 - Apesar de ser uma estratégia que consome poucos recursos não é tão comum.
-

Paginação - Realocação de Páginas

- First In First Out (FIFO)
 - A página que primeiro foi utilizada será a primeira a ser escolhida.
 - Implementação bastante simples, sendo necessário apenas uma fila onde as páginas mais antigas estão no início da fila e as mais novas no final.
 - As páginas podem ter que voltar várias vezes à fila.
-

Paginação - Realocação de Páginas

- Least Recently Used (LRU)
 - Esta estratégia seleciona a página utilizada menos recentemente, ou seja, a página que está mais tempo sem ser referenciada.
 - Gera overhead causado pela atualização em cada página referenciada, do momento do último acesso, além do algoritmo de busca das páginas.
-

Paginação - Realocação de Páginas

- Not Used Recently (NRU)
 - Bastante semelhante a LRU.
 - Nesta estratégia existe um flag que permite ao sistema a implementação do algoritmo. O flag de referência indica quando a página foi referenciada ou não, e está associado a cada entrada na tabela de páginas.
 - Inicialmente, todas as páginas estão com flag indicando que não foram referenciadas (0).
 - A medida que as páginas são referenciadas, o flag é modificado pelo hardware (1), depois de certo tempo é possível saber quais páginas foram referenciadas ou não.
-

Paginação - Realocação de Páginas

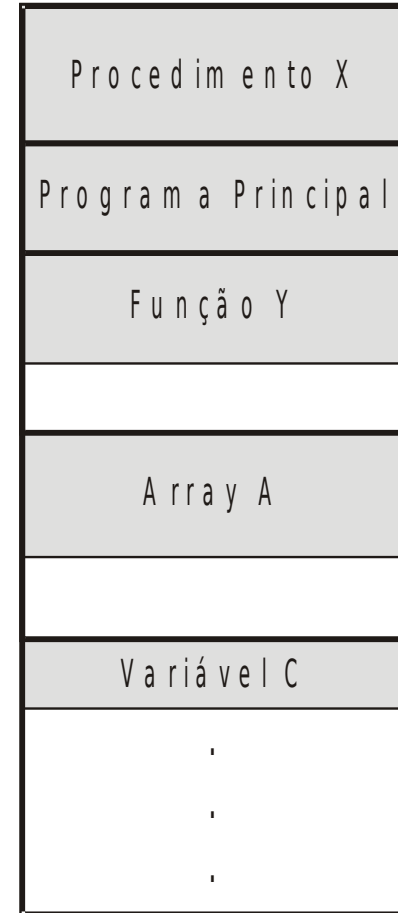
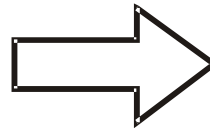
- Least Frequently Used (LFU)
 - Neste esquema, a página menos referenciada, ou seja, a menos freqüentemente utilizada será a página escolhida.
 - Para isso, é mantido um contador di número de referências feitas às páginas.
 - A página que contiver o contador com o menor número de referências será a página escolhida.
 - Este algoritmo privilegia páginas que são mais utilizadas.
 - Páginas que acabaram de entrar na memória podem ser escolhidas para deixar a mesma.
-

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

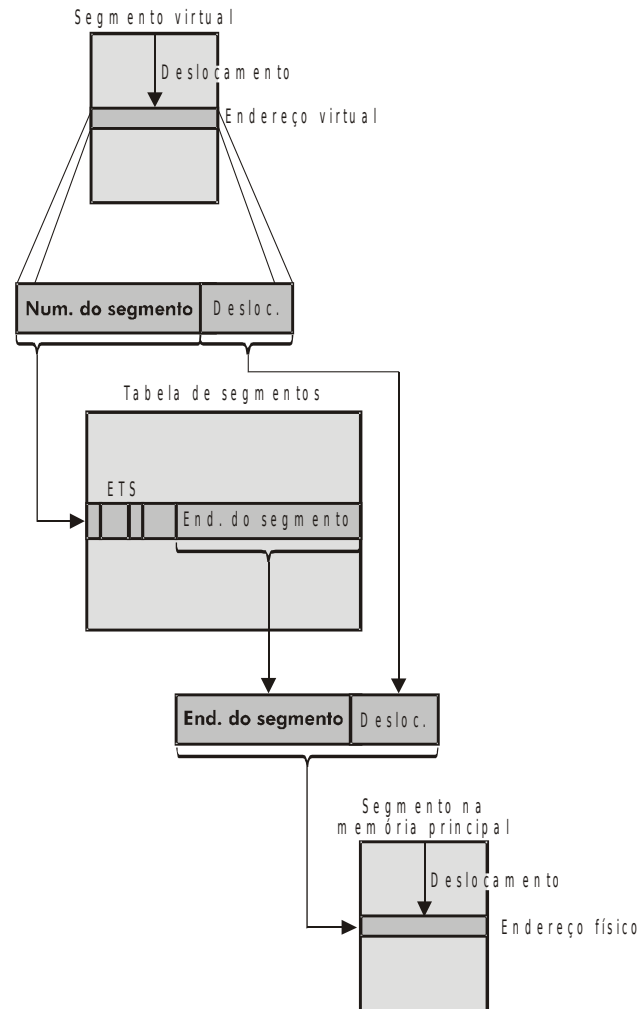
```
PROGRAM Segmento;  
  VAR A: ARRAY...  
      C: ...  
PROCEDURE X;  
END;  
FUNCTION Y;  
  
END;  
  
BEGIN  
  
END.
```



Segmentação

- ❑ Os segmentos são mapeados através de tabelas de mapeamento de segmentos (TMS).
 - ❑ Os endereços são compostos pelo número do segmento e um deslocamento dentro do segmento.
 - ❑ O número do segmento identifica unicamente uma entrada na tabela de segmentos (ETS), onde estão as informações sobre o segmento na memória real.
-

Segmentação



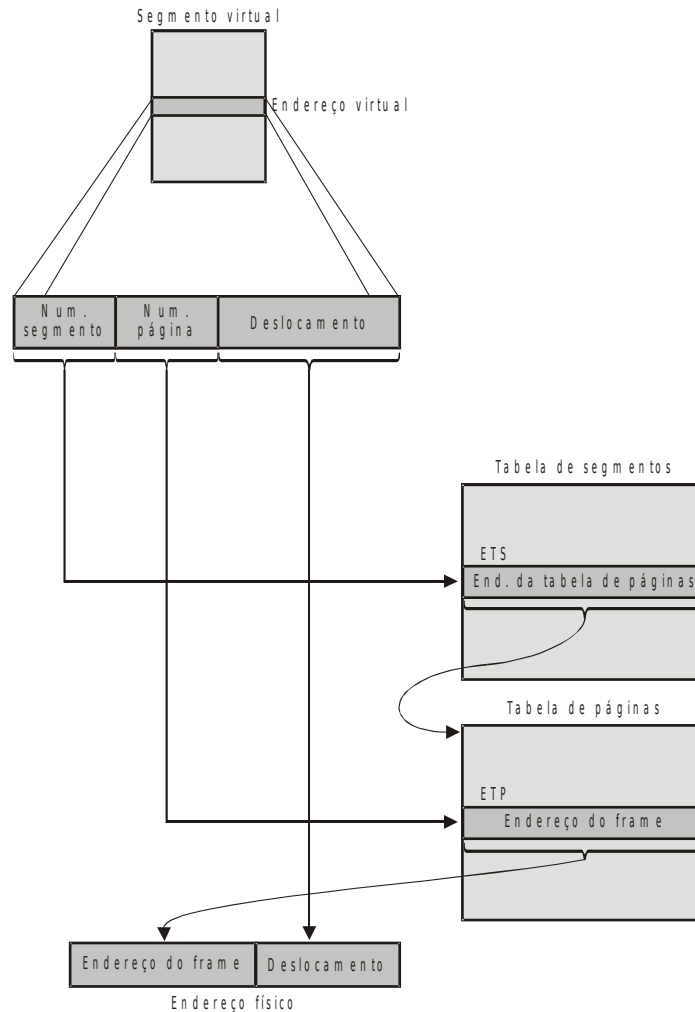
Segmentação

- ❑ O SO mantém uma tabela com áreas livres e ocupadas da memória.
 - ❑ Quando um novo processo é carregado para a memória, o sistema localiza um espaço livre que o acomode.
 - ❑ As estratégias para escolha da área livre podem ser as mesmas utilizadas na Alocação Particionada Dinâmica, ou seja, best fit, worst fit ou first fit.
 - ❑ O problema da fragmentação também ocorre nesse modelo.
 - ❑ Um outro problema é sua complexidade, pois o SO deve manter tabelas de segmentos que variam de acordo com o número de segmentos.
-

Segmentação com paginação

- Sistemas que implementam a segmentação com paginação permitem a divisão lógica dos processos em segmentos e, por sua vez, cada segmento é dividido, fisicamente, em páginas.
-

Segmentação com paginação



Proteção

- A proteção é necessária para impedir que um processo, ao obter acesso a uma página ou segmento do sistema, a modifique ou mesmo tenha acesso a ela.
- Este esquema é implementado utilizando-se as entradas das tabelas de mapeamento, onde alguns bits especificam os acessos permitidos a cada uma das páginas / segmentos.

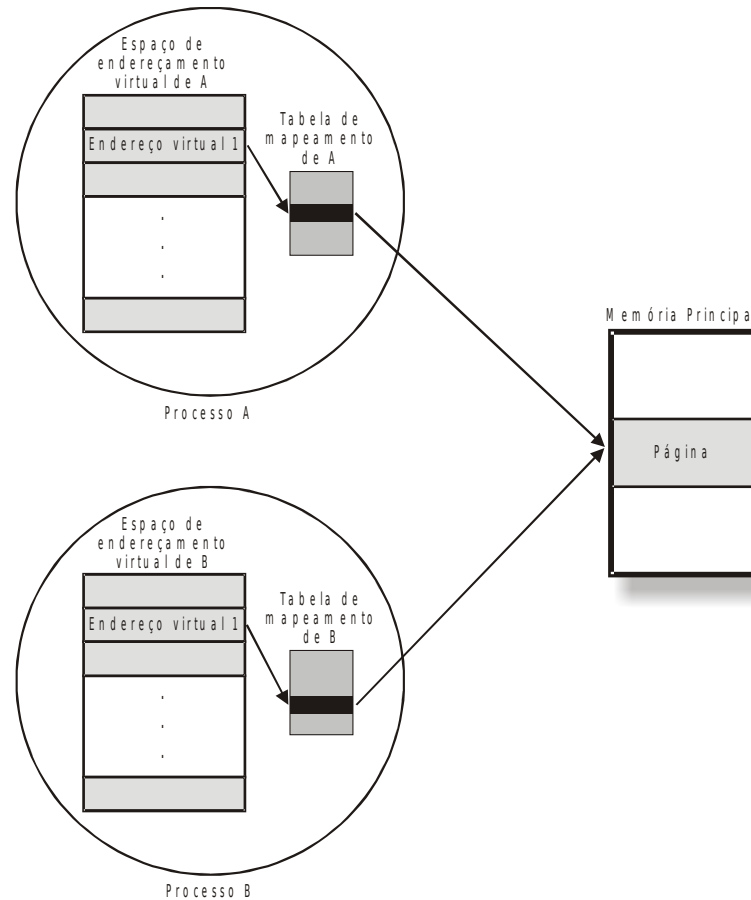


Proteção

LG	Descrição
00	Sem acesso
10	Acesso de leitura
11	Acesso para leitura/gravação

Compartilhamento de Memória

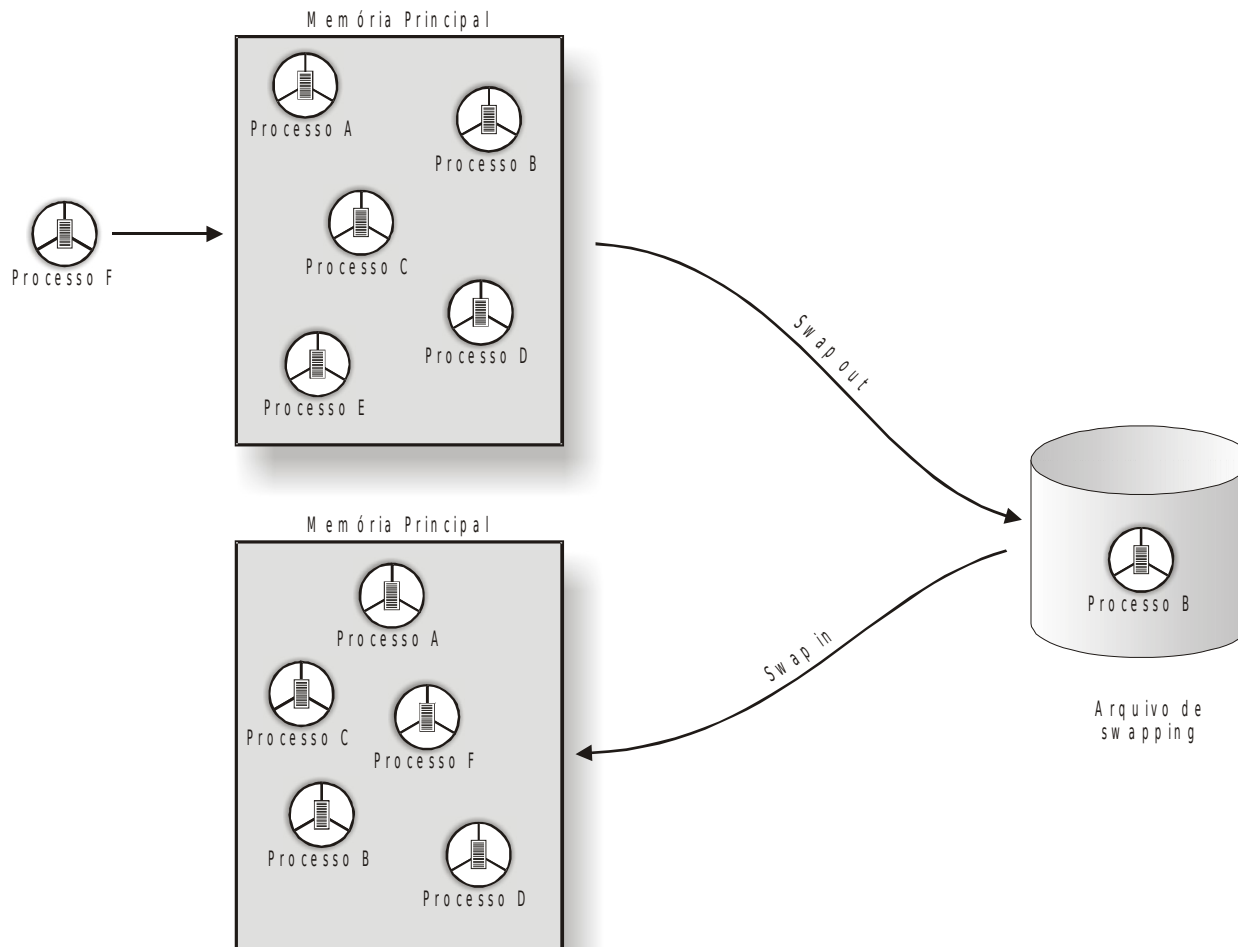
- Em ambientes multiprogramáveis, é comum a utilização de códigos reentrantes.



Swapping em Memória Virtual

- ❑ O conceito de swapping continua válido quando aplicado à memória virtual.
 - ❑ Em sistemas que utilizam esta técnica, quando existem novos processos que devem ser processados e não existe mais memória real suficiente, o SO seleciona um ou mais processos que deverão deixar a memória para ceder espaço aos novos processos.
 - ❑ Os critérios utilizados para escolha do processo são a prioridade e o estado do processo.
 - ❑ Na prioridade, escolhe-se, dentre os processos, os de menor prioridade e no critério estado do processo escolhe-se aqueles processos que estão aguardando algum evento (bloqueados).
-

Swapping em Memória Virtual



Trashing

- ❑ Trashing pode ser definido como sendo a excessiva transferência de páginas / segmentos entre a memória principal e a memória secundária.
 - ❑ Esse problema aplica-se tanto a sistemas que implementam paginação como para aqueles que implementam a segmentação.
 - ❑ O trashing ocorre quando existem mais processos competindo por memória do que espaço disponível.
-

Trashing

- ❑ O SO tenta administrar a memória de maneira que todos os processos sejam atendidos.
 - ❑ De qualquer maneira, se existir mais processos para serem executados do que memória disponível, a solução seria acrescentar mais memória principal.
-

Memória Virtual

Prof. Dr. José Luís Zem

Prof. Dr. Renato Kraide Soffner

Prof. Ms. Rossano Pablo Pinto



Faculdade de Tecnologia de Americana

Centro Paula Souza