

Gerência de Memória Swapping, Memória Virtual, Segmentação e Paginação

Sistemas Operacionais
Prof. Robson Augusto Siscoutto

Plano de Aulas

- **Gerência de Memória**
 - Introdução;
 - Partições Fixas e Variáveis
 - Alocação Contígua Simples;
 - Alocação Particionada:
 - Estática e Dinâmica;
 - Formas de gerenciamento;
 - Mapa de Bits
 - Lista Encadeada
 - Sistema Buddy
 - Swapping;
 - Memória Virtual
 - Memória Virtual, Paginação e Segmentação
 - Algoritmos de Substituição de Páginas

Contextualizando...

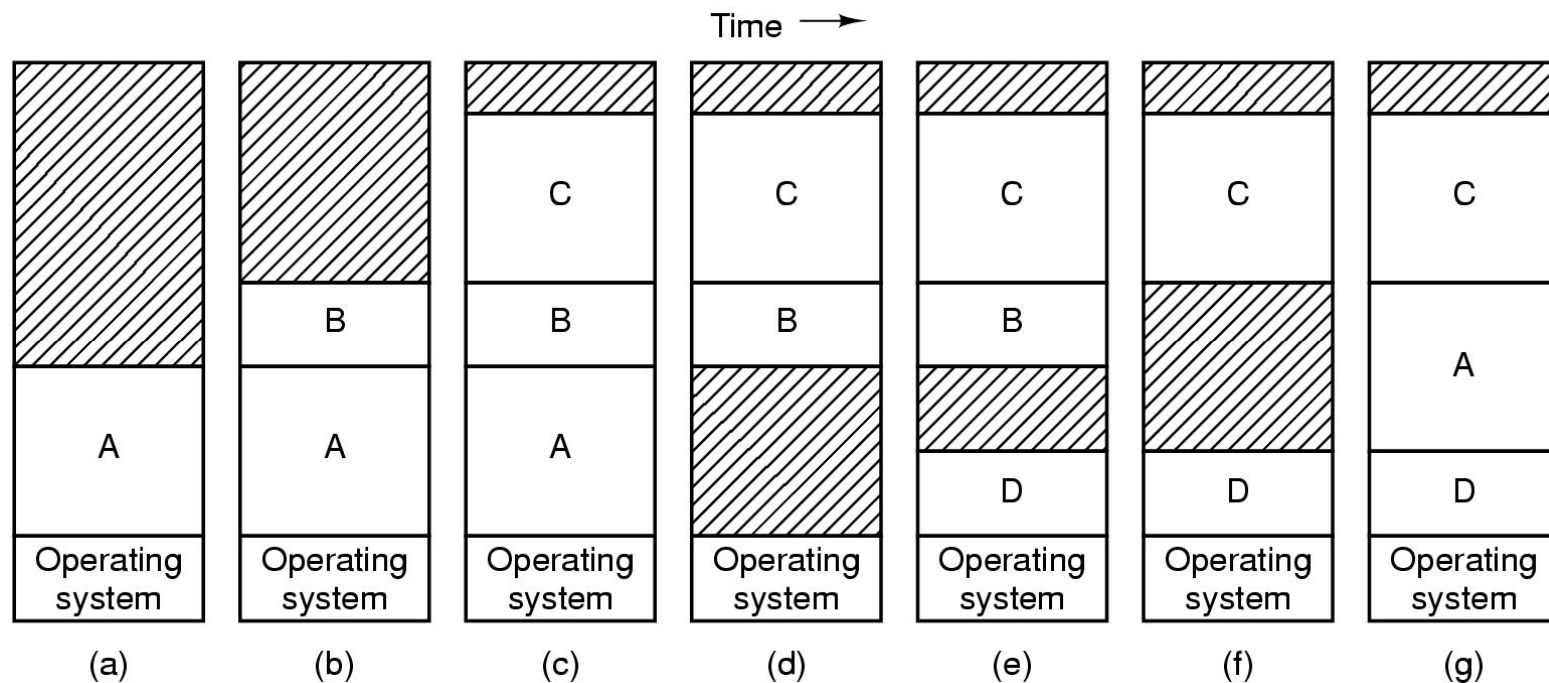
- Evolução de Organizações de Memória:

Real	Real		Virtual		
Sistemas monousuário dedicados	Sistemas de multiprogramação de memória real		Sistemas de multiprogramação de memória virtual		
	Multiprogramação de partição fixa	Multiprogramação de partição variável	Paginação pura	Segmentação pura	Paginação e segmentação combinadas
	Absoluta	Realocável			

Gerenciamento de Memória

Swapping

- Utilizada para tentar resolver o problema da falta de memória;
- Conceito:
 - o sistema escolhe um programa residente que é levado da memória para o disco (Swapped Out), retornando posteriormente para a memória principal (Swapped In), como se nada tivesse ocorrido;



Gerenciamento de Memória

Swapping

- Objetivo do Swapping:
 - permitir um maior compartilhamento da memória;
 - Aumentar o Throughput do sistema:
 - numero de processos processados;
- Desvantagens:
 - normalmente é mais eficiente em sistemas de poucos usuários e com aplicações pequenas;
 - possui um elevado custo nas operações de Entrada/Saída;
- Técnicas utilizadas:
 - Mapa de bits, lista encadeadas e buddy;

Gerenciamento de Memória

Memória Virtual

- **Conceito:**
 - Utilizar memória principal e secundária combinadas, dando a ilusão de existir uma memória muito maior que a principal;
- **Espaço de Endereçamento Virtual:**
 - Um processo, em um ambiente de memória virtual, não realiza referências a endereços físicos reais de memória, mas apenas a endereços virtuais;
 - No momento da execução da instrução, o endereço virtual é traduzido para o endereço físico:
 - o processador acessa apenas posições reais de memória;
 - A tradução é conhecida como MAPEAMENTO.

Memória Virtual

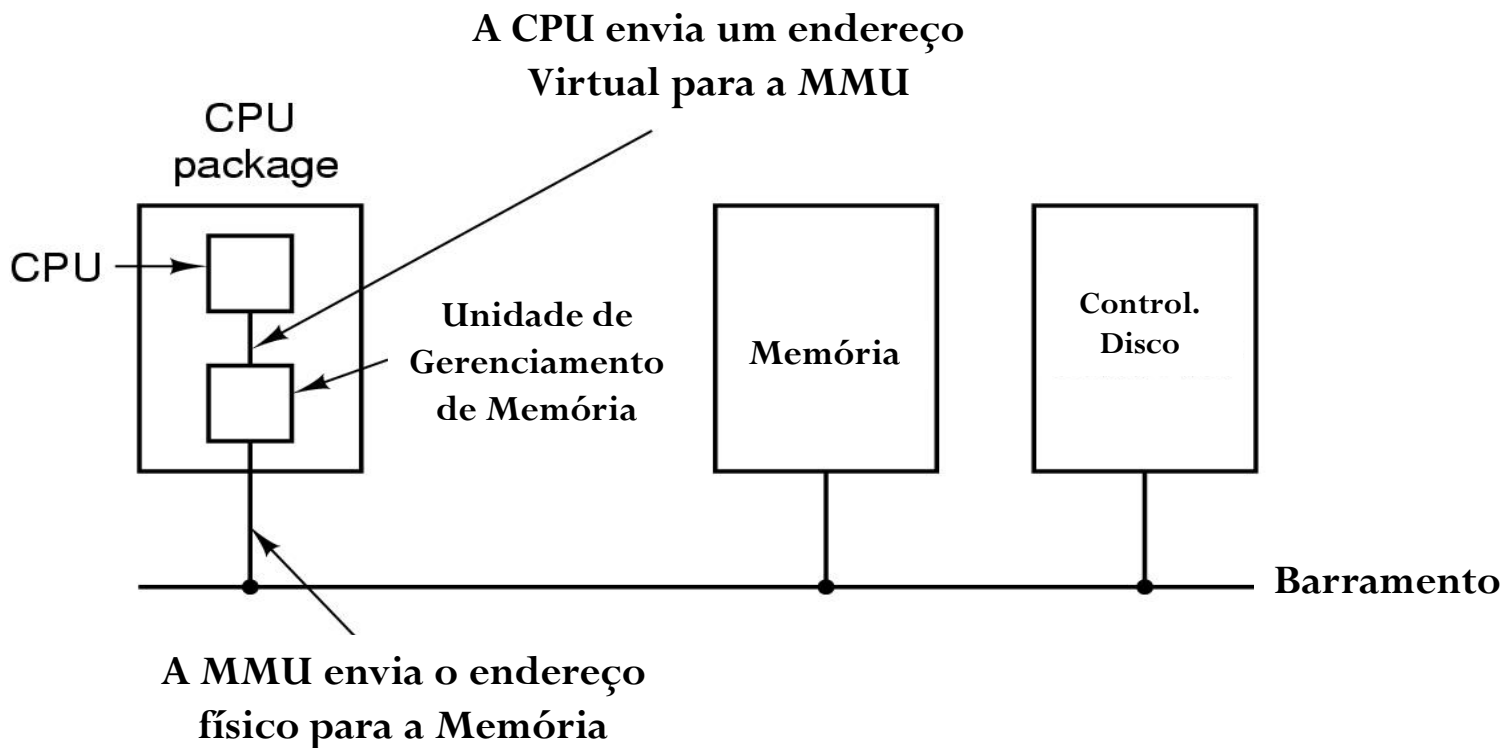
Conceitos Básicos

- Utilizar Memória Principal e Secundárias Combinadas, Criando a ilusão de que existe mais memória do que a disponível no sistema.
- Existem dois tipos de endereço nos sistemas de memória virtual:
 - **Endereços Virtuais** - EV: Referenciados por processos ;
 - **Endereços Físicos** - ER: Localizações físicas na memória principal.

Memória Virtual

Conceitos Básicos

- Tradução: endereços virtuais para endereços físicos:
 - Unidade de gerenciamento de memória (MMU)



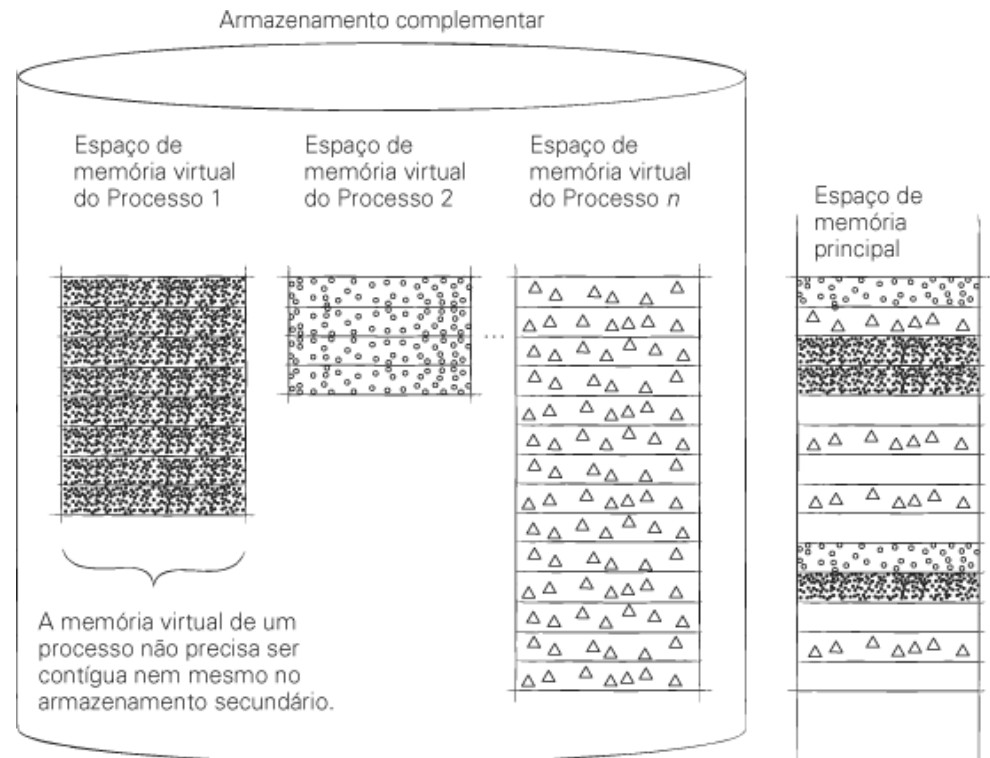
Memória Virtual

Conceitos Básicos

- $|EV|$ em geral é bem maior que $|ER|$.
 - O sistema operacional armazena partes de EV de cada processo (instruções e dados para execução) na Memória Principal e deixa a maior parte na memória secundária (programas e dados).

- Armazenamento de dois níveis:

- O sistema operacional move porções de EV entre a memória principal (e os caches) e o armazenamento secundário.



Memória Virtual

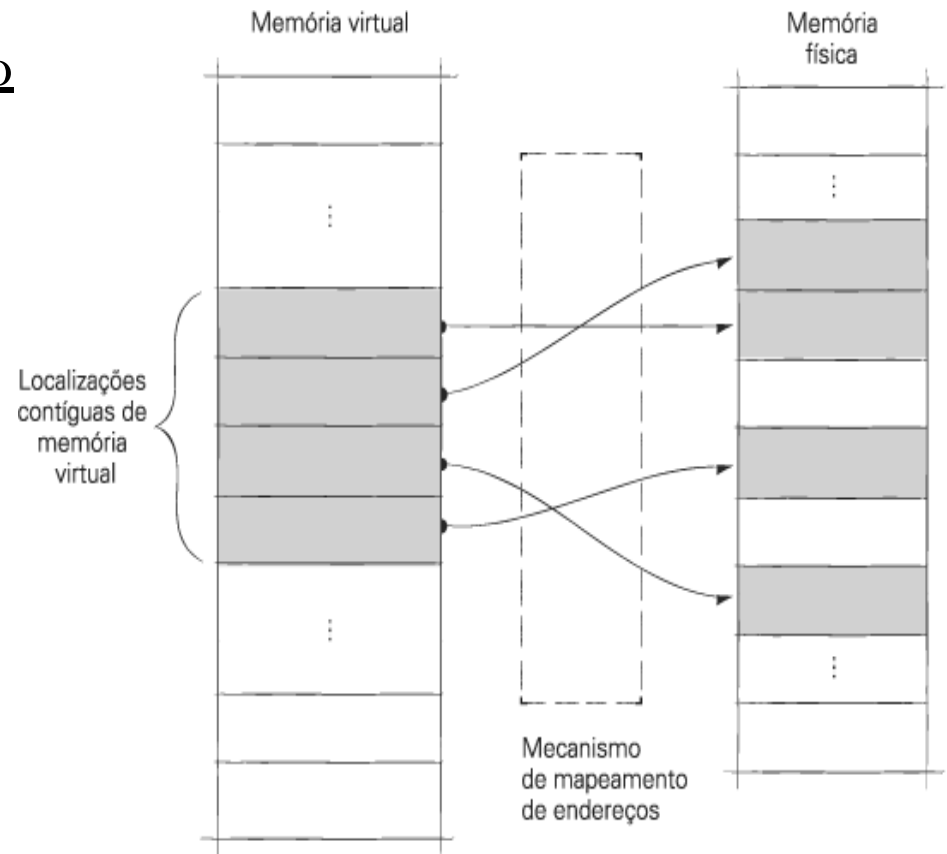
Mapeamento

- **Mapeamentos de Endereço:**

- Indicam quais regiões do espaço de endereço virtual de um processo, estão na memória principal no momento e onde estão localizadas.
- Utiliza-se Tabelas de Mapeamento por Processo;

- **Contigüidade Artificial**

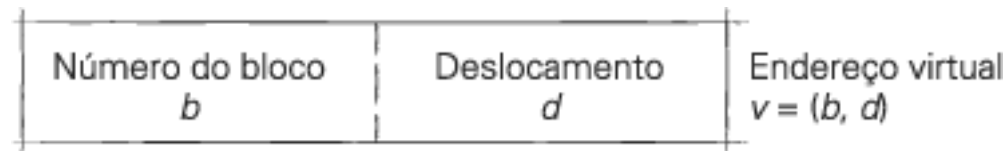
- Os endereços virtuais contíguos podem **não** corresponder aos endereços de memória real contíguos.



Memória Virtual

Mapeamento

- As Tabelas de Mapeamento mapeiam Blocos:
 - O sistema representa endereços como pares ordenados:

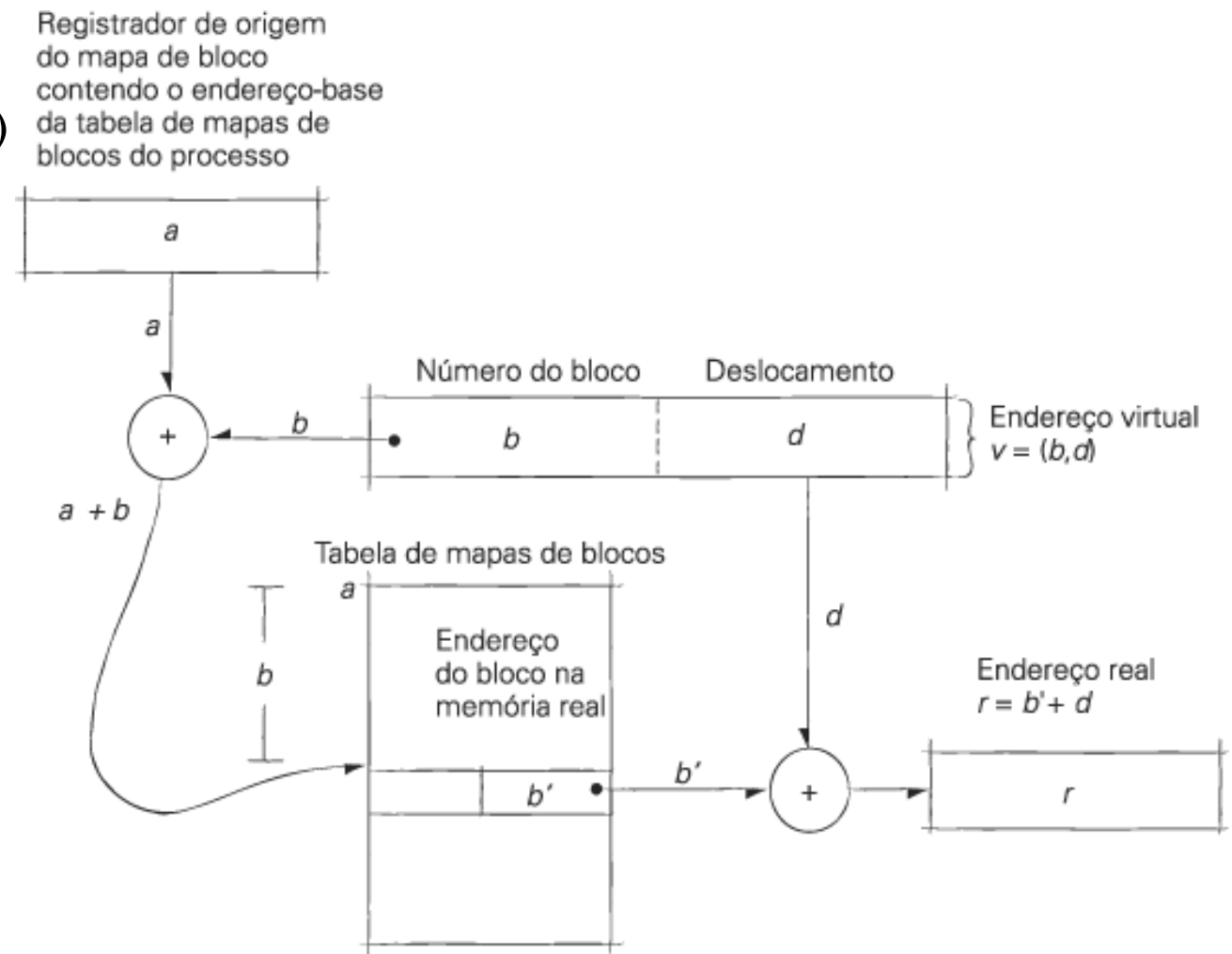


- Blocos de Tamanho Fixo = Página
 - Essa técnica denomina-se paginação.
- Blocos de Tamanhos Diferentes = Segmentos
 - Essa técnica denomina-se segmentação.

Memória Virtual

Mapeamento

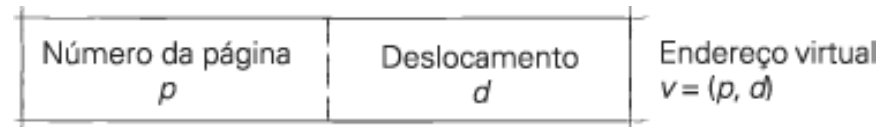
- Tradução de Endereço Virtual com Mapeamento de Bloco:
 - Dado um endereço $v = (b, d)$



Memória Virtual

Paginação

- A paginação usa o mapeamento de blocos de tamanho fixo.
- O endereço virtual no sistema de paginação é um par ordenado $v = (p, d)$.
 - p é o **número da página** na memória virtual na qual o item referenciado reside.
 - d é o **deslocamento** do início da página p na qual o item referenciado está localizado.



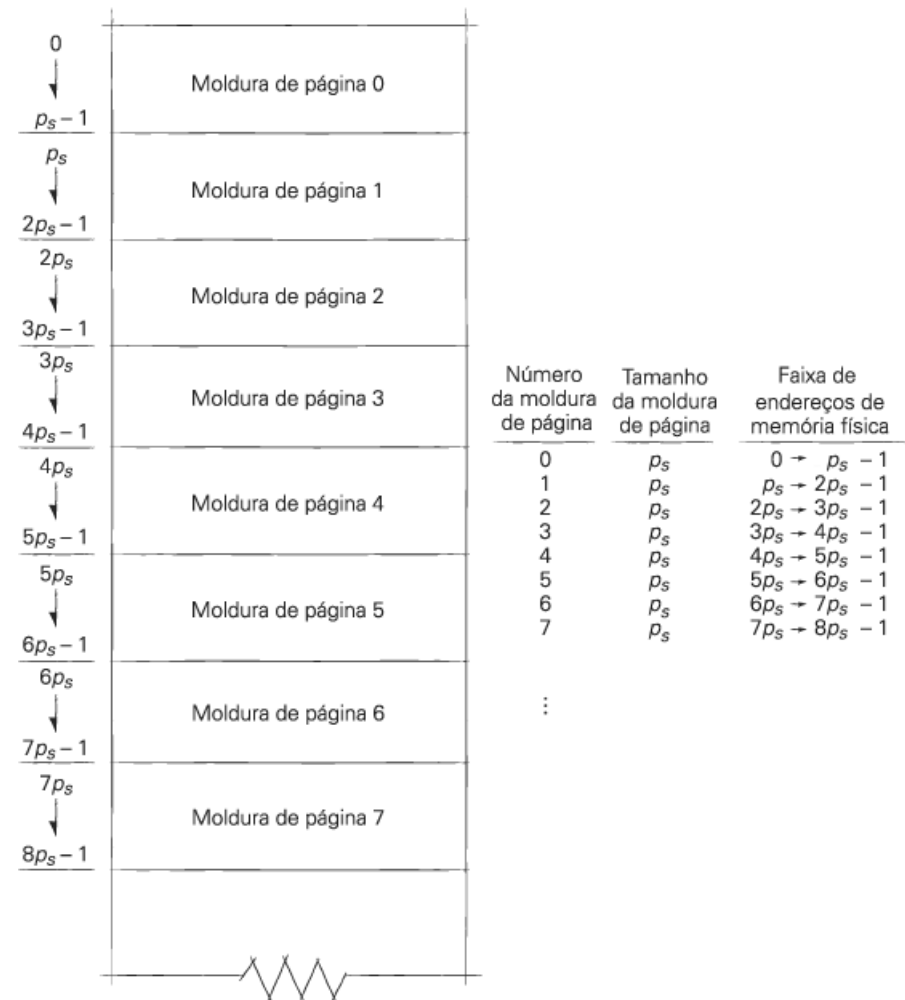
- Espaço de Endereçamento Virtual é dividido em blocos fixos denominadas **páginas**;
- Espaço de Endereçamento Real é dividido em blocos fixos denominados **molduras de páginas**;

Memória Virtual

Paginação

- Memória principal dividida em **Molduras de Páginas**

- Começa em um endereço da memória principal que é um múltiplo inteiro do tamanho fixo de página (p_s).
- Considera páginas do mesmo tamanho;



Memória Virtual

Paginação

- Correspondência entre endereços de memória virtual e endereços de memória física em um sistema de paginação pura.

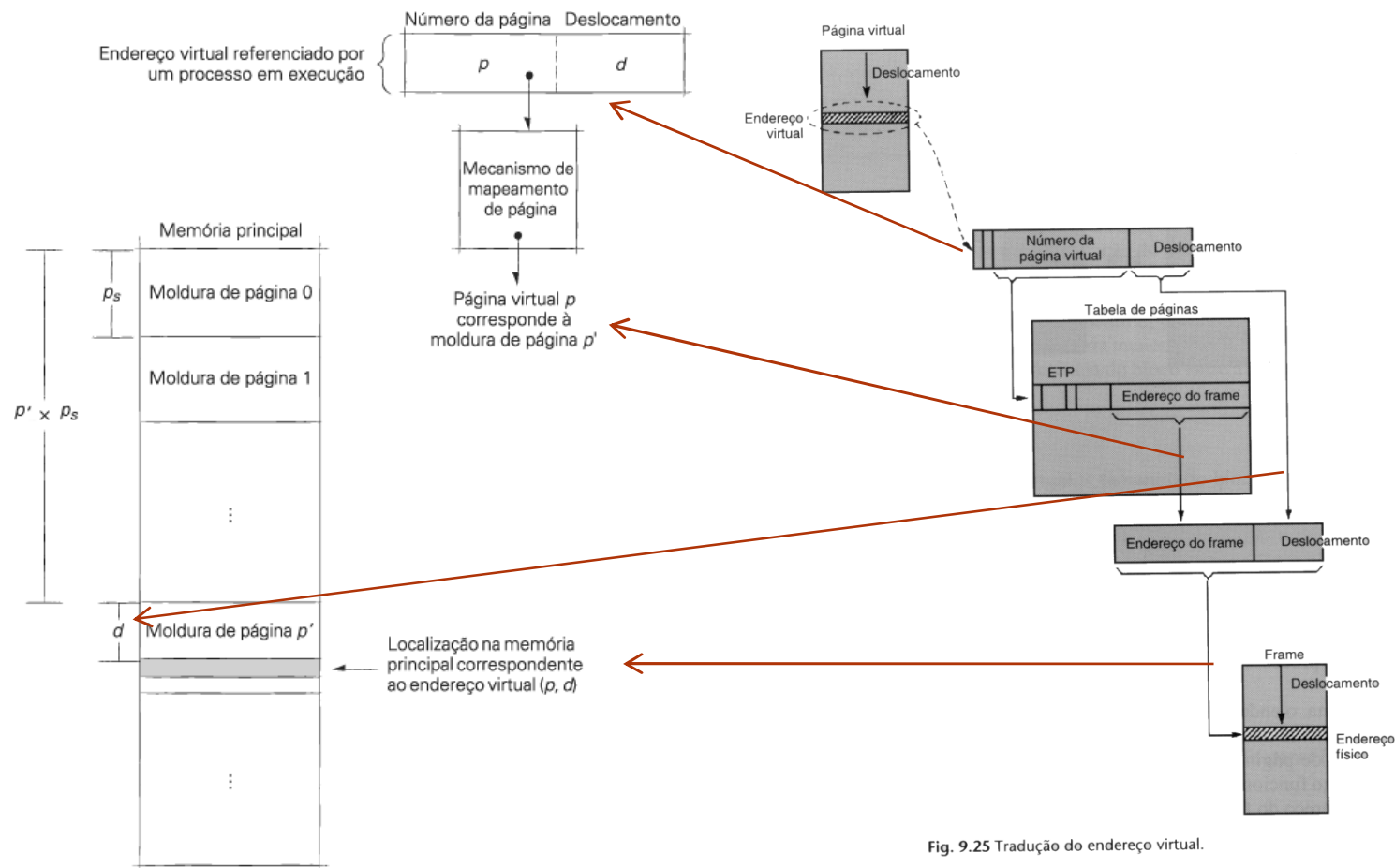


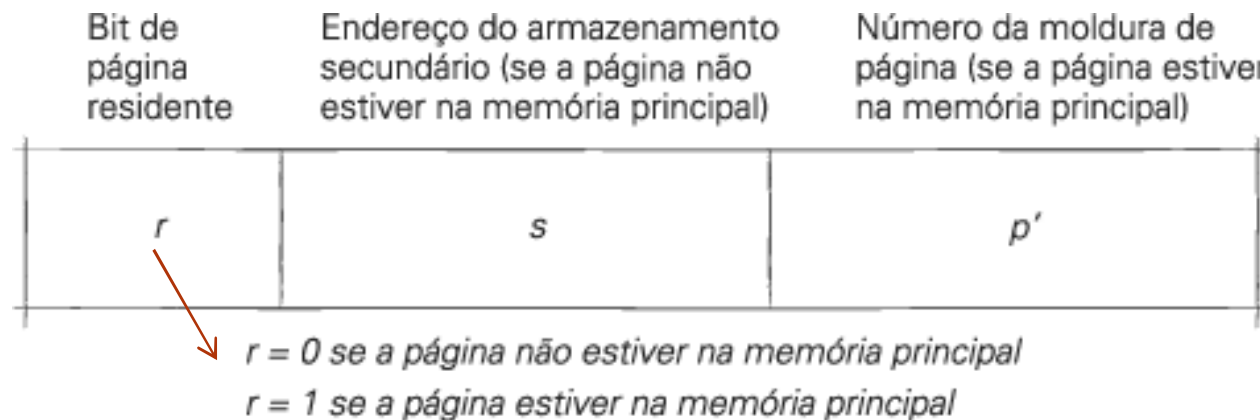
Fig. 9.25 Tradução do endereço virtual.

Memória Virtual

Paginação

- Entrada de Tabela de Páginas

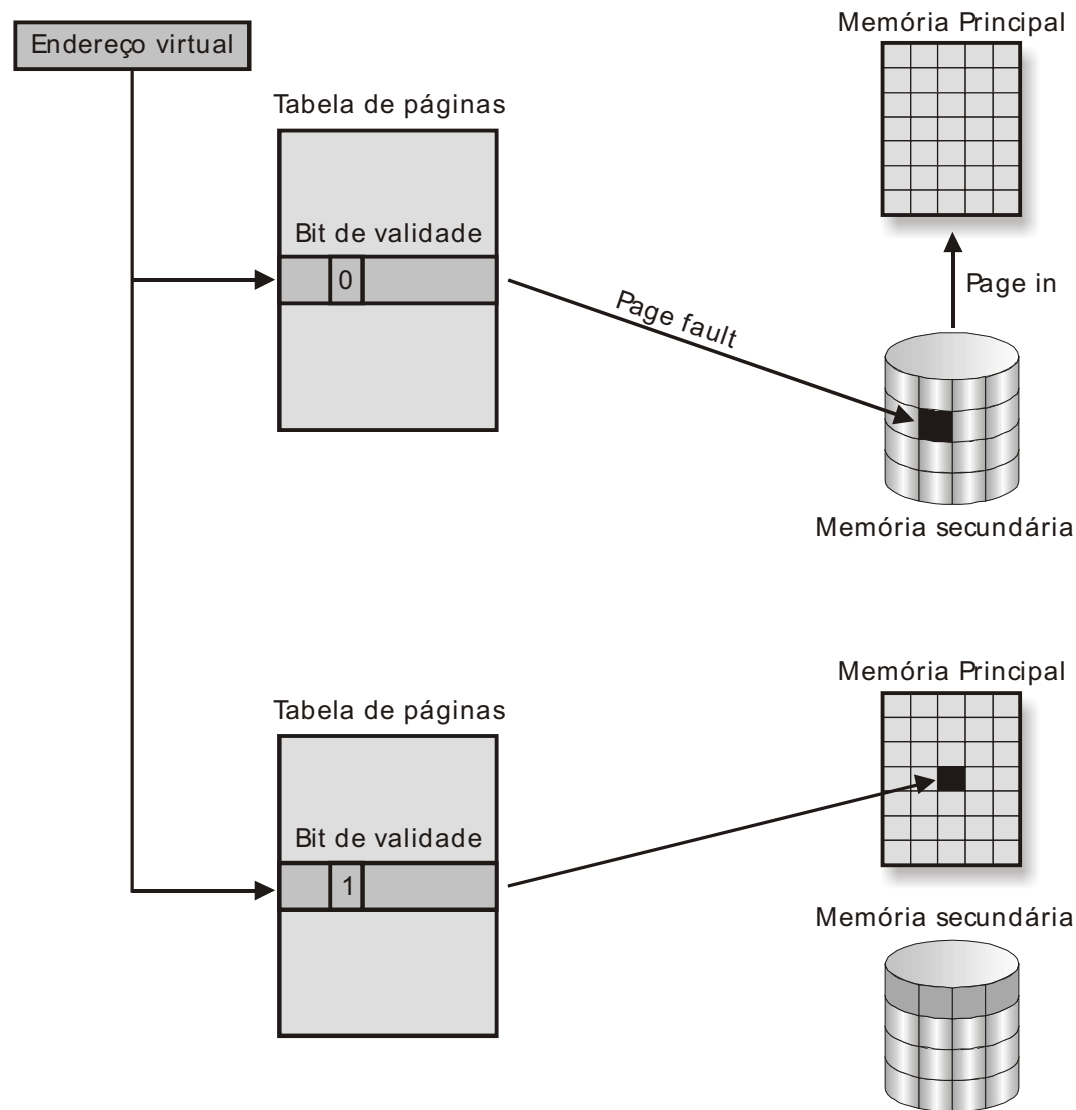
- Contém um bit residente para indicar se a página está na memória.
 - Se estiver, o p' armazena o número da moldura da página.
 - Do contrário, o s armazena a localização da página no armazenamento secundário.



- Quanto a busca da página “faltante” no disco, existem duas técnicas:
 - Paginação por Demanda: quando a página for referenciada;
 - Paginação Antecipada: tenta prevê a utilização da página e faz uma busca antecipada;

Memória Virtual

Paginação

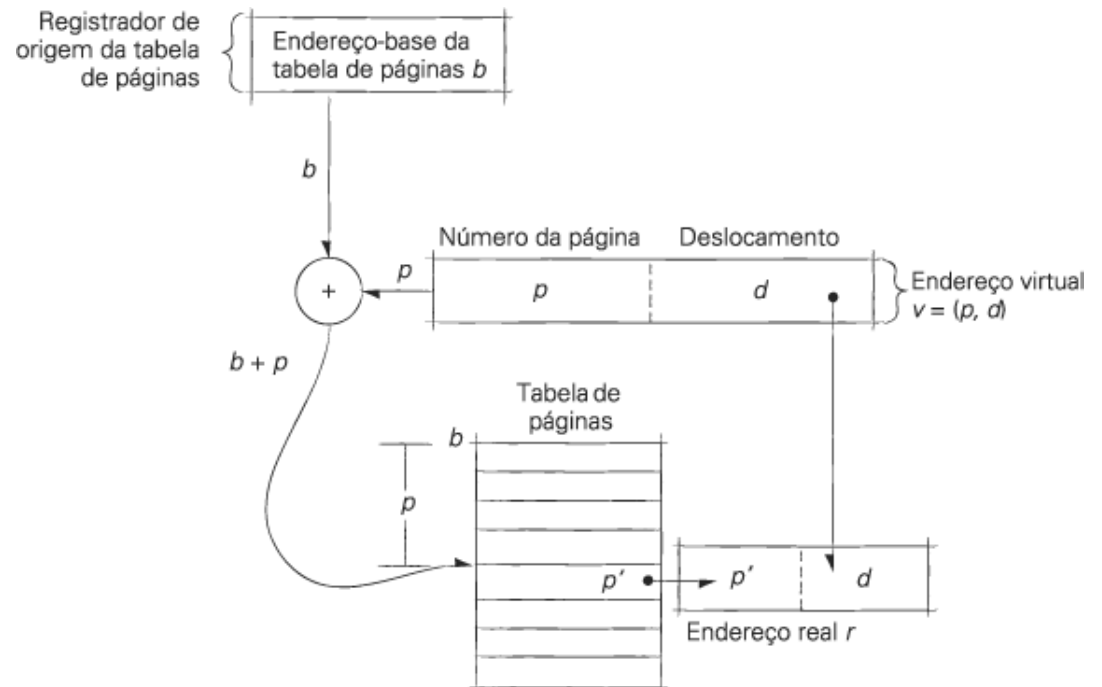


Memória Virtual

Paginação

- Tradução por **Mapeamento Direto**:

- O processo referencia o endereço virtual $v = (p, d)$.
 - A MMU adiciona o endereço-base da tabela de páginas do processo, b , ao número da página referenciada, p .
 - $b + p$ forma o endereço de memória principal da tabela de páginas para a página p .
 - O sistema concatena p' com o deslocamento, d , para formar o endereço real, r .

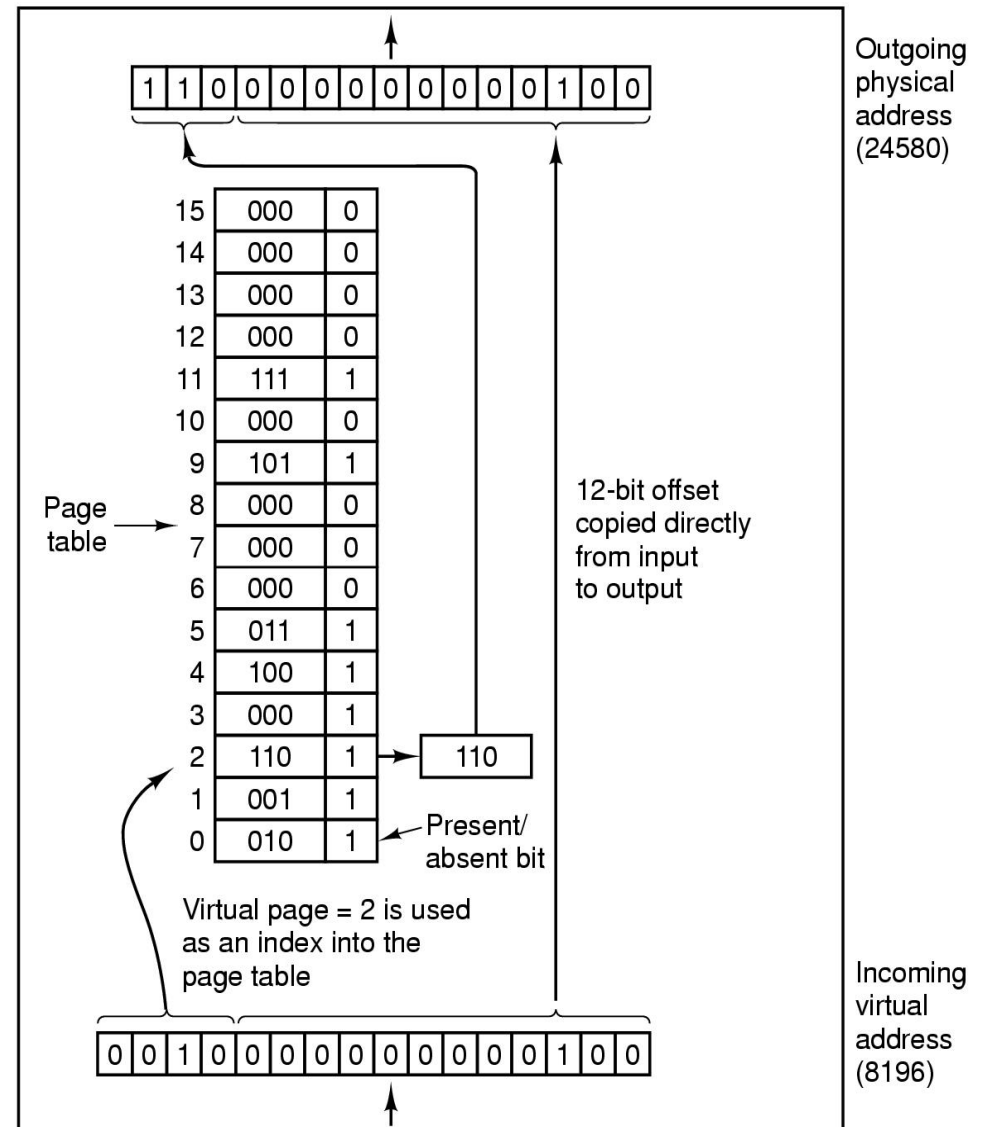


Memória Virtual

Paginação

- Tradução por Mapeamento Direto - Exemplo:

- Endereço Virtual 8192 de 16 bits (0010 000000000100) que chega na MMU;
- Temos um número de páginas de 4 bits com deslocamento de 12 bits
 - Com 4 bits conseguimos 16 páginas virtuais (2^4)
 - Com 12 bits conseguimos endereçar 4096 bytes dessa página.



Memória Virtual

Paginação

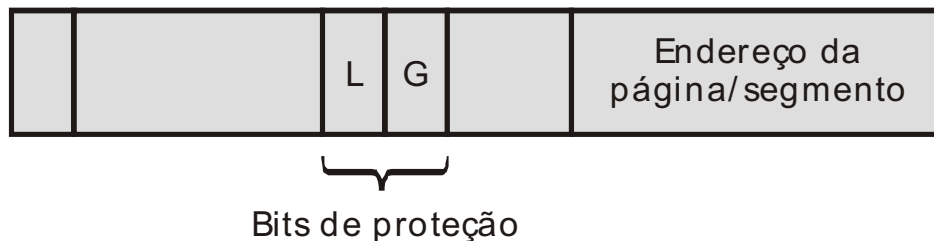
- Tradução por Mapeamento Associativo ou TLB – Tabela de Tradução de Endereços (Translation Lookaside Buffer):
 - Uma forma para aumentar o desempenho na busca pelo número da moldura de página é manter na MMU uma tabela com um pequeno número de entradas:
 - Mapear os endereços virtuais para reais;
 - **Funcionamento:**
 - endereço virtual apresentado a MMU para tradução;
 - O HW verifica se o número da página está na TLB, comparando em paralelo, com todos as entradas;
 - Encontrada e acesso permitido, acessa moldura de página real;
 - Se não estiver, busca na tabela de páginas e substitui a entrada na TLB

Memória Virtual

Paginação

- Campos da TLB

Campo	Descrição
Tag	Endereço virtual sem o deslocamento.
Modificação	Bit que indica se a página foi alterada.
Referência	Bit que indica se a página foi recentemente referenciada, sendo utilizada para a realocação de entradas na TLB.
Proteção	Define a permissão de acesso à página.
Endereço físico	Posição do frame na memória principal.

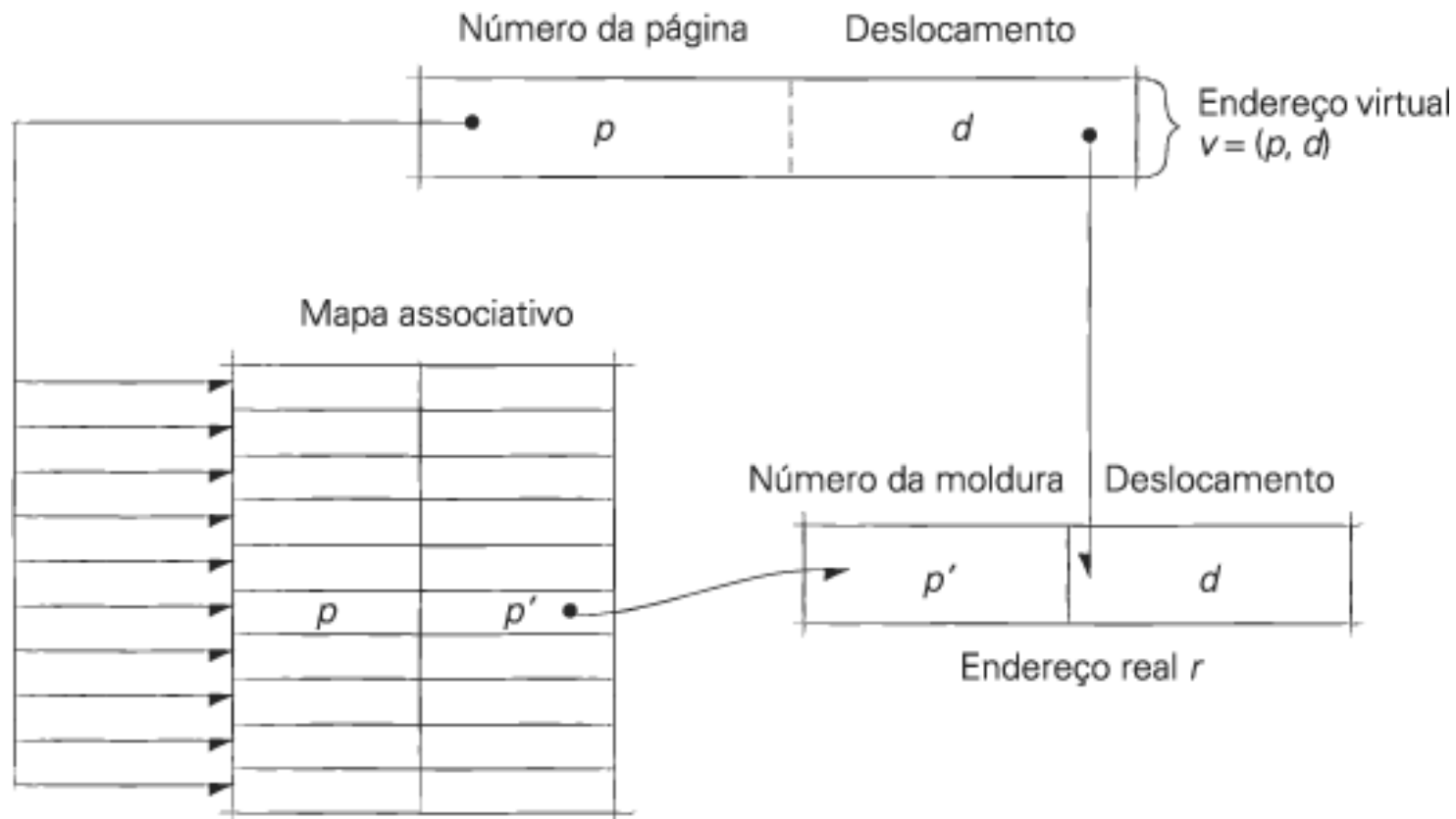


LG	Descrição
00	Sem acesso
10	Acesso de leitura
11	Acesso para leitura/gravação

Memória Virtual

Paginação

- Tradução por Mapeamento Associativo ou TLB



Memória Virtual

Paginação

- Tradução por Mapeamento Associativo

Entrada na TLB

Valid	Virtual page	Modified	Protection	Page frame
1	140	1	RW	31
1	20	0	R X	38
1	130	1	RW	29
1	129	1	RW	62
1	19	0	R X	50
1	21	0	R X	45
1	860	1	RW	14
1	861	1	RW	75

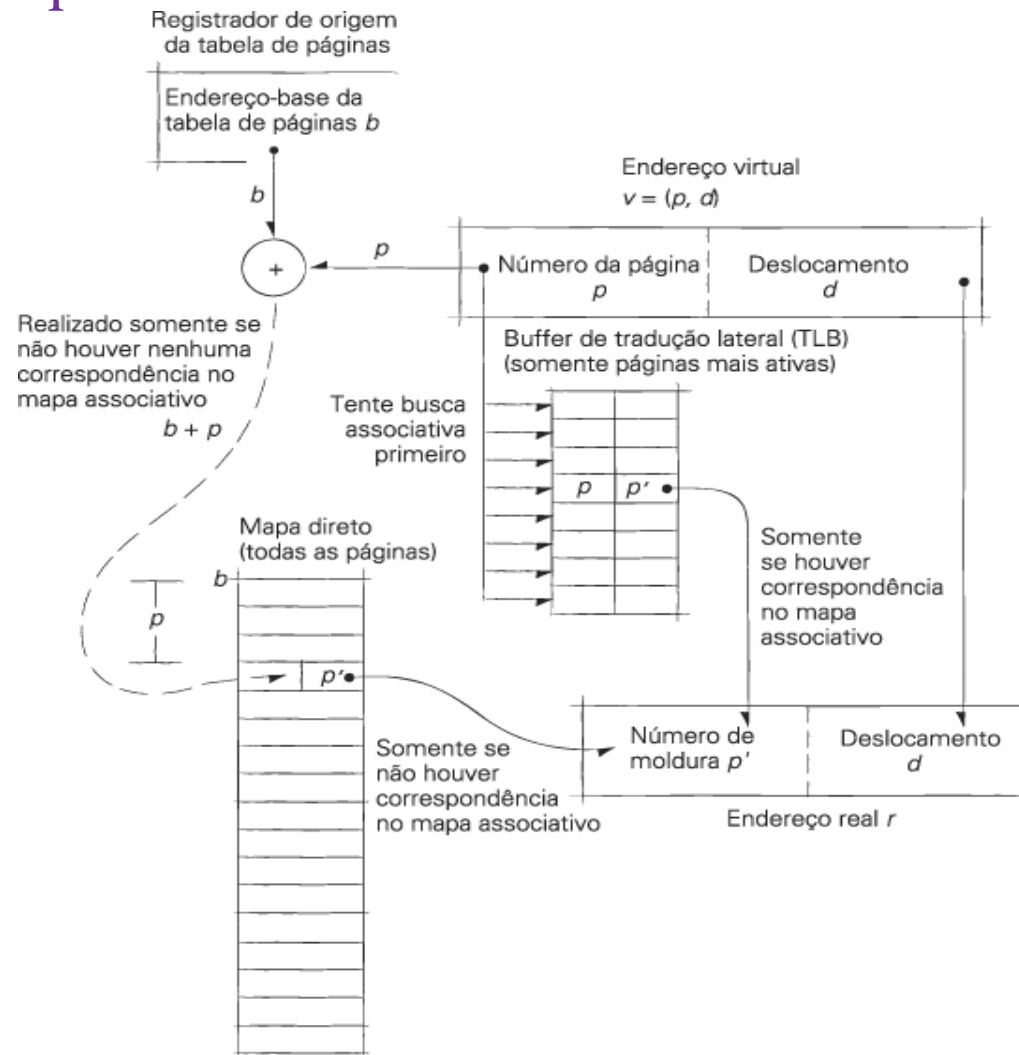
Exemplo:

- processo em um laço que referencia constantemente as páginas virtuais 19, 20 e 21 – proteção RX;
- Os dados referenciados estão localizados nas páginas 129 e 130;
- A página 140 contém índices de um vetor;
- A pilha localiza-se nas páginas 860 e 861;

Memória Virtual

Paginação

- Tradução por Mapeamento Direto/Associativo



Memória Virtual

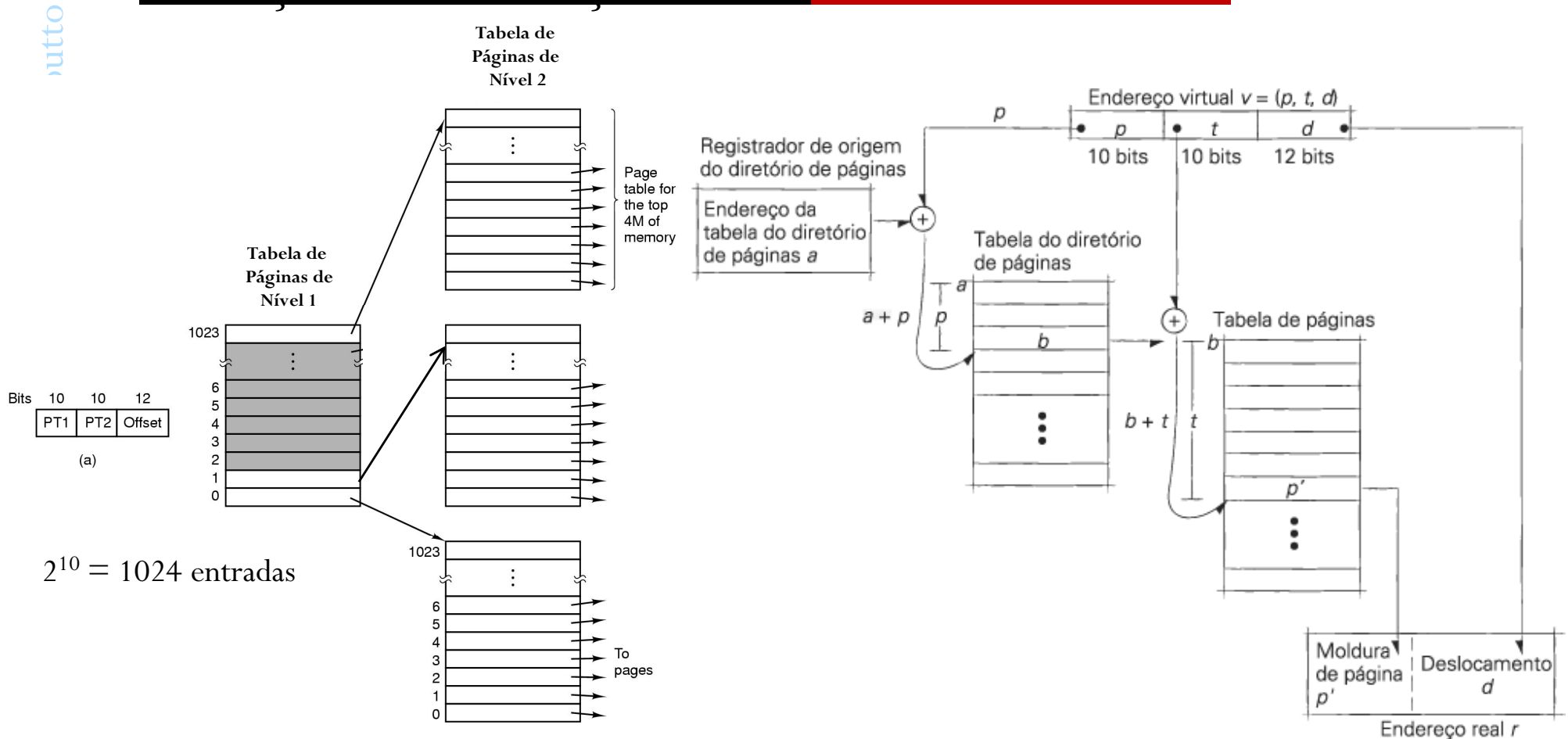
Paginação

- Tradução do Endereço usando Tabelas Multiníveis:
 - Minimizar o problema de armazenar tabelas de páginas muito grande na memória;
 - Evitar que todas as tabelas de páginas estejam na memória ao mesmo tempo;
 - Tabelas Multiníveis permitem armazenarem porções da tabela de páginas de um processo em localizações não contíguas da memória principal;
- Hierarquia das Tabelas de Páginas
 - Cada nível contém uma tabela que armazena ponteiros para tabelas do nível que está abaixo.
 - O nível mais inferior compreende as tabelas que contêm traduções de endereço.

Memória Virtual

Paginação

- Tradução do Endereço usando Tabelas Multiníveis:



Memória Virtual

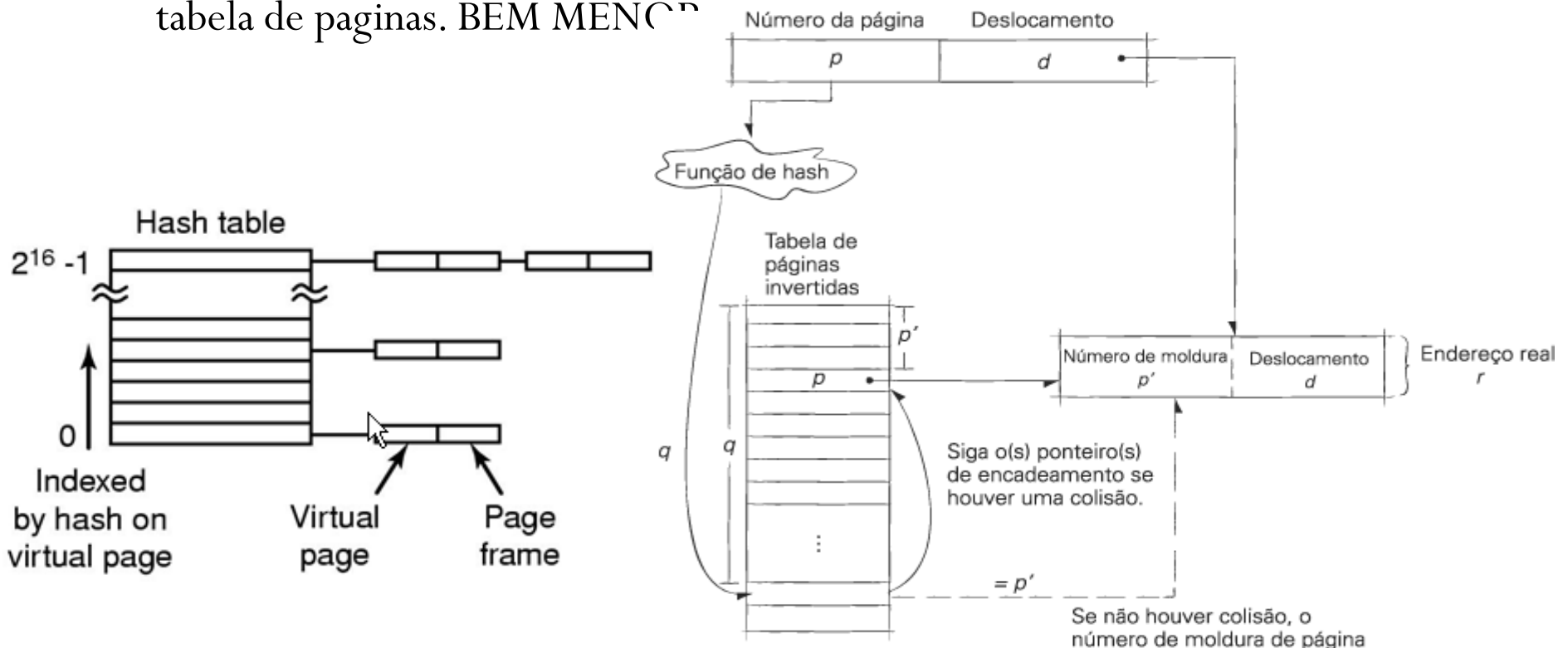
Paginação

- Tradução do Endereço usando Tabelas de Páginas Invertidas:
 - Tabela de páginas = uma entrada por página;
 - Dependendo do tamanho do espaço de endereço virtual, o tamanho da tabela de página é inviável;
 - Com o surgimento de máquinas com 64 bits, ai sim que a tabela se torna inviável;
 - Ex: espaço de endereçamento virtual de 64 bits e paginas de 4 KB (2^{12})
 - $2^{64} - 2^{12} = 2^{52}$ entradas, se cada entrada tiver 8 bytes $\cong 30\ 000\ 000$ de Gigabytes = Inviável
 - **Solução: Tabelas de Páginas Invertidas**
 - Existe apenas uma entrada por moldura de página da memória real, em vez de uma entrada por página do espaço de endereçamento virtual;

Memória Virtual

Paginação

- Tradução do Endereço usando Tabelas de Páginas Invertidas:
 - Ex: Usando Tabela Invertida e com endereços virtuais de 64 bits, uma página de 4KB e 256 MB de RAM, a tabela de páginas invertida teria:
 - $256 \text{ MB} / 4 \text{ KB} = 62500$ entradas reais, logo $2^{16} = 65536$ entradas necessárias na tabela de paginas. BEM MENOS



Memória Virtual

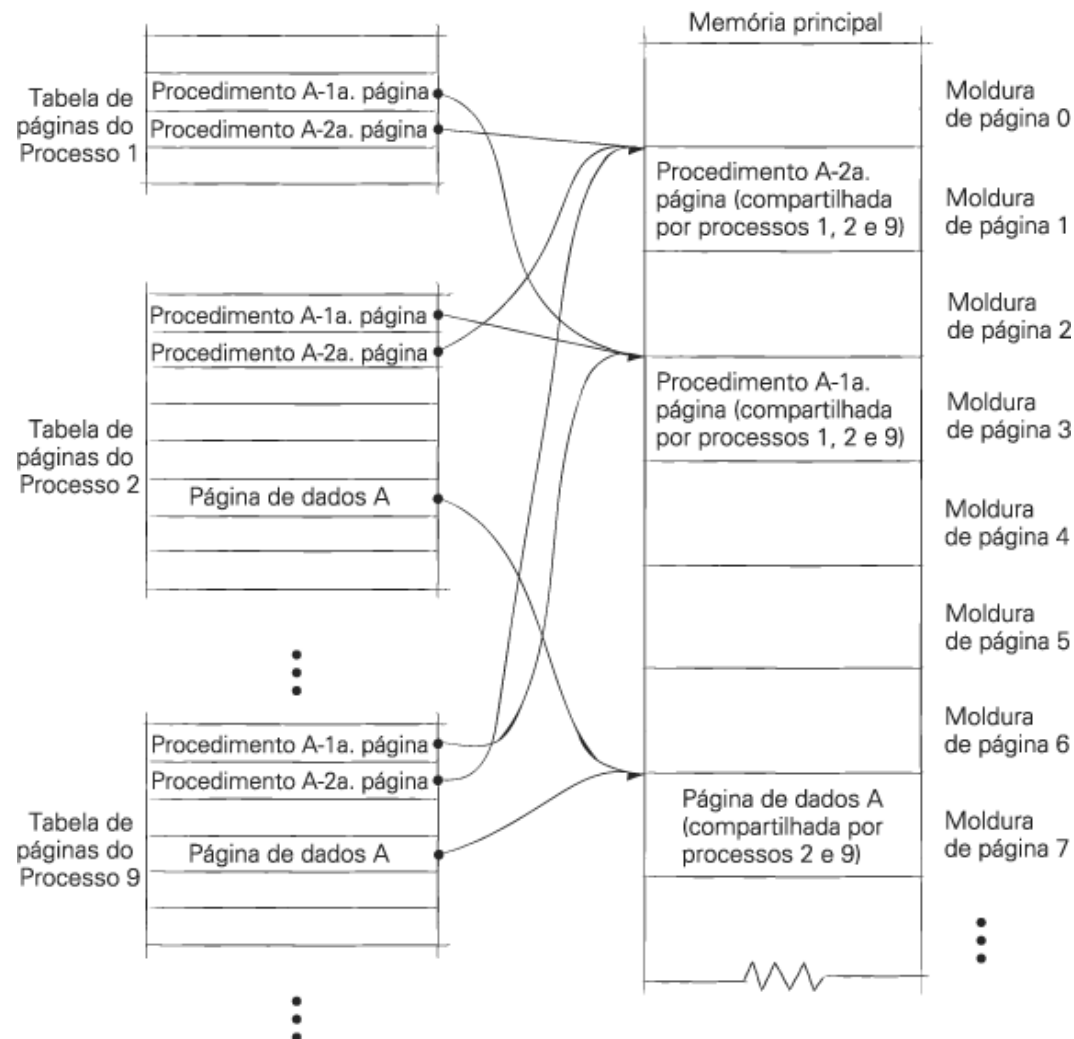
Paginação

- Compartilhamento em um Sistema de Paginação
 - Diminui o consumo de memória por programas que usam dados e/ou instruções comuns.
 - Exige que o sistema identifique cada página como compartilhável ou não compartilhável.

Memória Virtual

Paginação

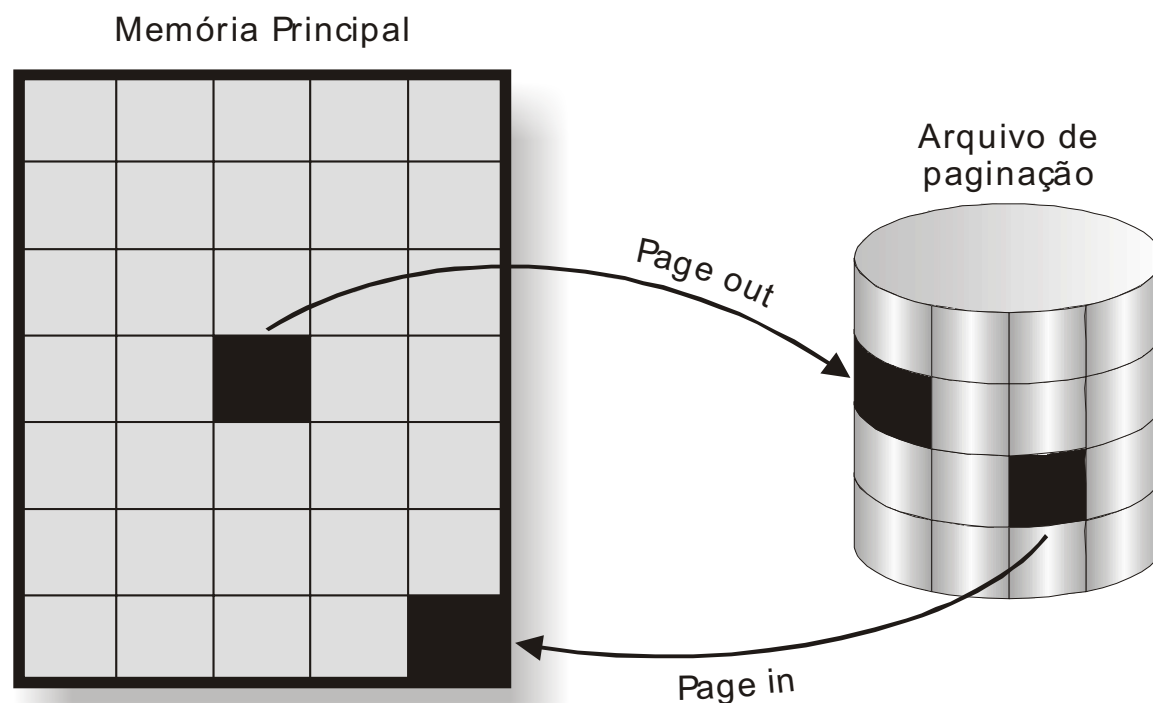
- Compartilhamento em um Sistema de Paginação



Memória Virtual

Algoritmos de Substituição de Páginas

- O maior problema na gerência de memória virtual por paginação não é decidir que página carregar para a memória, mas quais páginas remover;
- Para resolver tal problema foram propostos vários algoritmos de substituição de páginas:



Memória Virtual

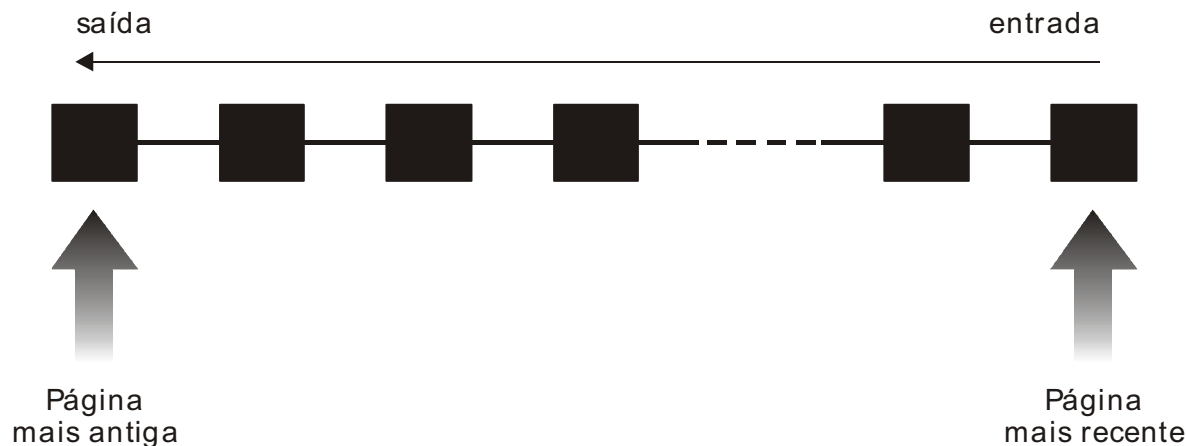
Algoritmos de Substituição de Páginas

- **Algoritmos Ótimo:**
 - escolher uma página que não fosse referenciada em um futuro próximo;
 - O sistema Operacional não tem como saber se uma página vai ou não ser referenciada novamente;

Memória Virtual

Algoritmos de Substituição de Páginas

- **Algoritmo Aleatório;**
 - Escolhe uma página aleatória;
 - Raramente utilizada;
- **Algoritmo FIFO;**
 - escolhe a página que primeiro foi utilizada;
 - Também o SO não tem como saber se será referenciada;



Memória Virtual

Algoritmos de Substituição de Páginas

- **Algoritmo LRU – Least Recently Used;**
 - Escolhe a página menos utilizada recentemente;
 - Problema de overhead durante a atualização de cada página referenciada;
- **Algoritmo NRU – Not Recently Used;**
 - escolhe a página que não foi recentemente utilizada;
 - utiliza um flag para saber qual página foi ou não referenciada;

Memória Virtual

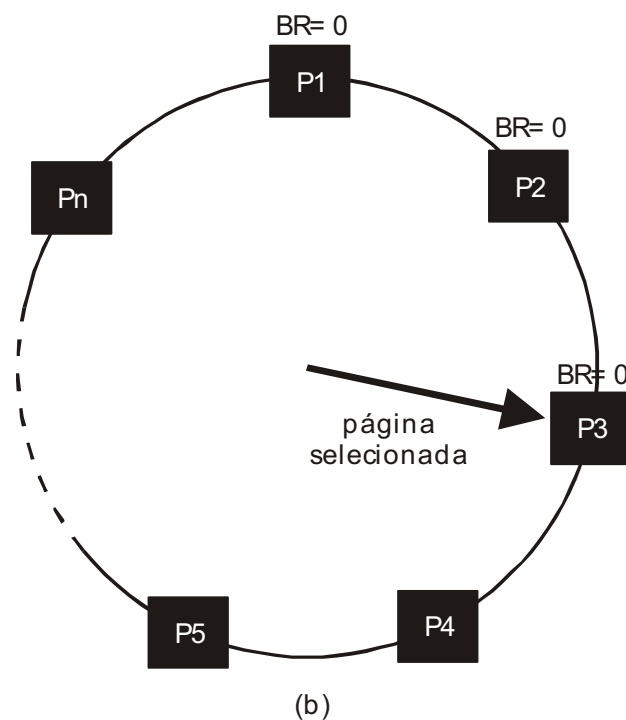
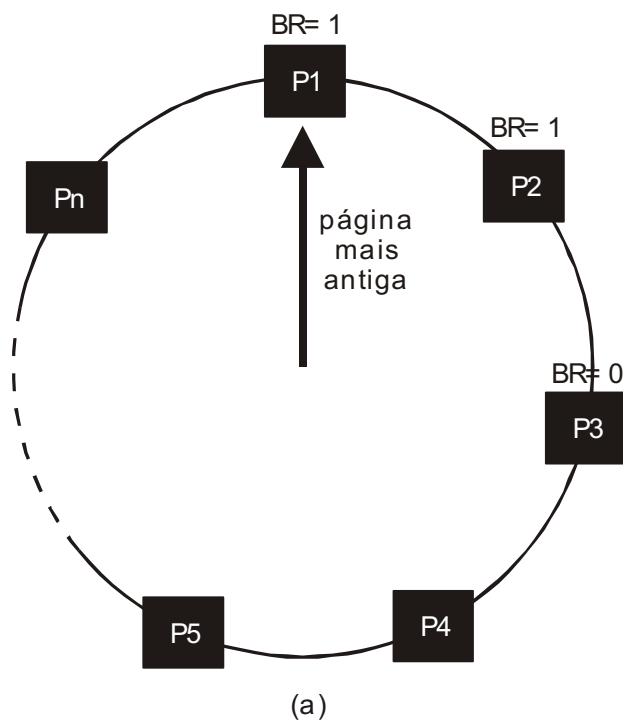
Algoritmos de Substituição de Páginas

- **Algoritmo LFU – Least Frequently Used;**
 - escolhe a página menos referenciada, ou seja, a menos freqüentemente utilizada;
 - utiliza um contador do numero de referencias feitas às páginas;
 - a página que tiver o contador menor será removida;
 - problema: páginas recentes chegadas são sempre escolhidas primeiro;

Memória Virtual

Algoritmos de Substituição de Páginas

- FIFO Circular (Clock)



Memória Virtual

Segmentação

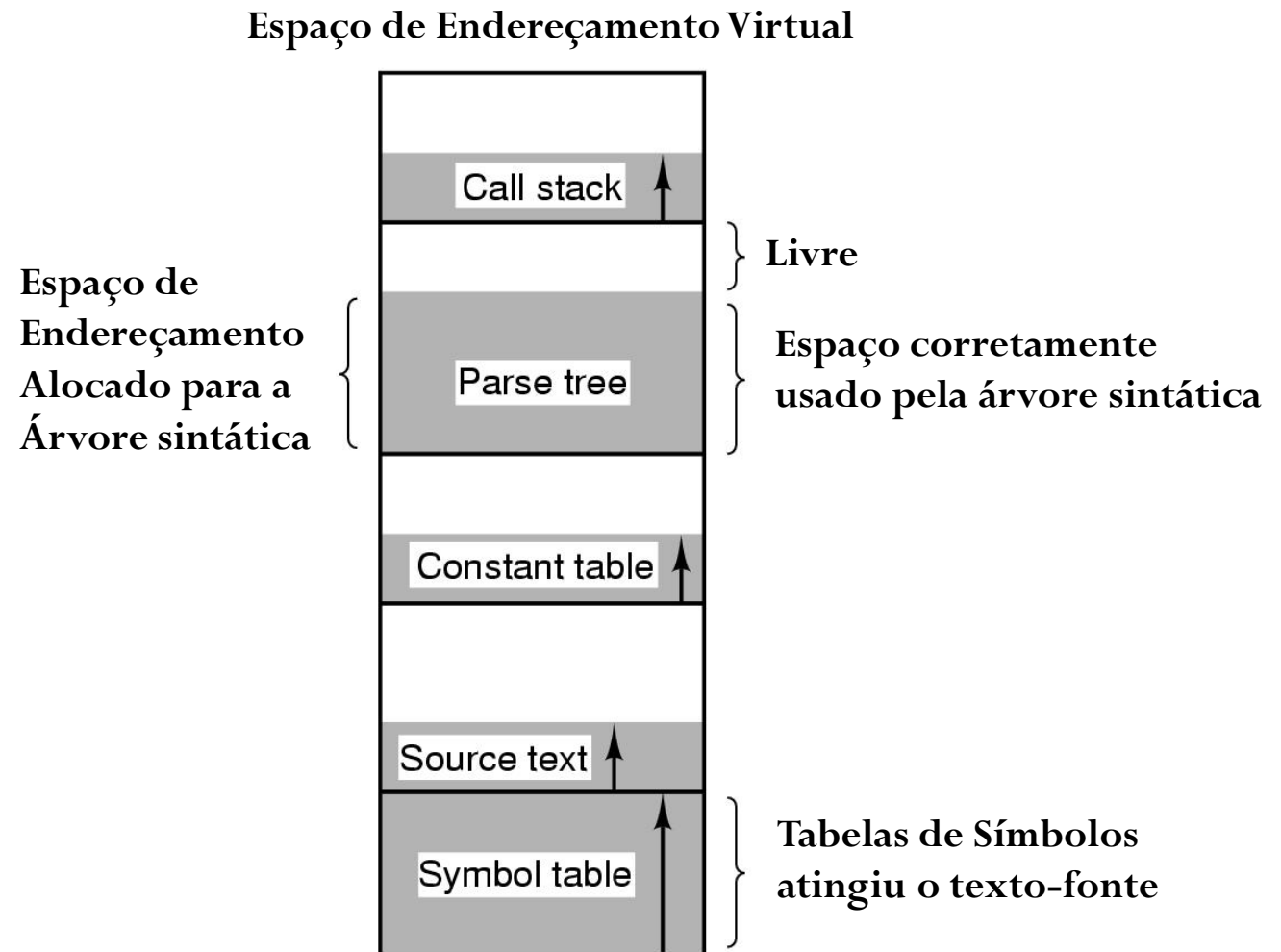
- **Contextualizando...**

- Memória virtual paginada possui endereçamento unidimensional que vai de 0 a algum máximo, um após ao outro;
- Devido a sua forma de alocação física, os programas devem ser alocados em regiões contiguas de endereçamento virtual.
- Caso parte desse programa cresça dinamicamente (p.ex. tabelas), estas poderá atingir outra parte do programa;

Memória Virtual

Segmentação

- Contextualizando...



Memória Virtual

Segmentação

- **Contextualizando...**

- É necessário livrar o programador da obrigatoriedade de gerenciar a expansão e a contração de tabelas, p.ex. do mesmo modo que a memória virtual elimina a preocupação de organizar o programa em overlays.

- **Solução:** prover a máquina com muitos espaços de endereçamento completamente independentes, chamados **segmentos**.

Memória Virtual

Segmentação

- Segmentos:
 - Bloco de dados ou instruções de um programa.
 - Contém uma parte lógica significativa do programa (por exemplo, procedimento, conjunto, pilha).
 - Cada segmento é constituído de uma seqüência linear de endereços que pode ser qualquer um, de 0 a algum máximo permitido.
 - Os segmentos não têm de ser do mesmo tamanho ou adjacentes entre si na memória principal.
- É possível executar um processo se suas instruções atuais e dados referenciados estiverem em segmentos na memória principal.

Memória Virtual

Segmentação

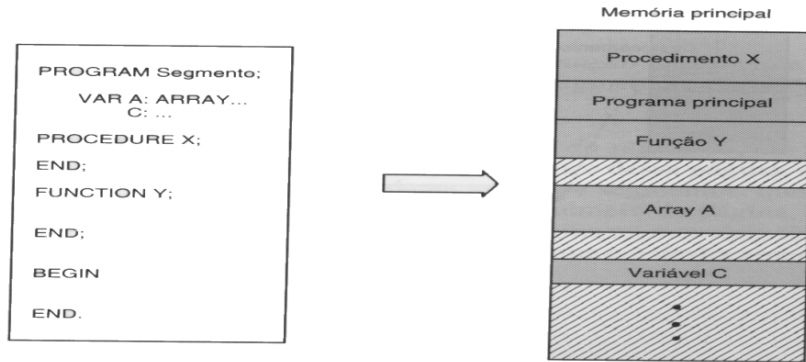
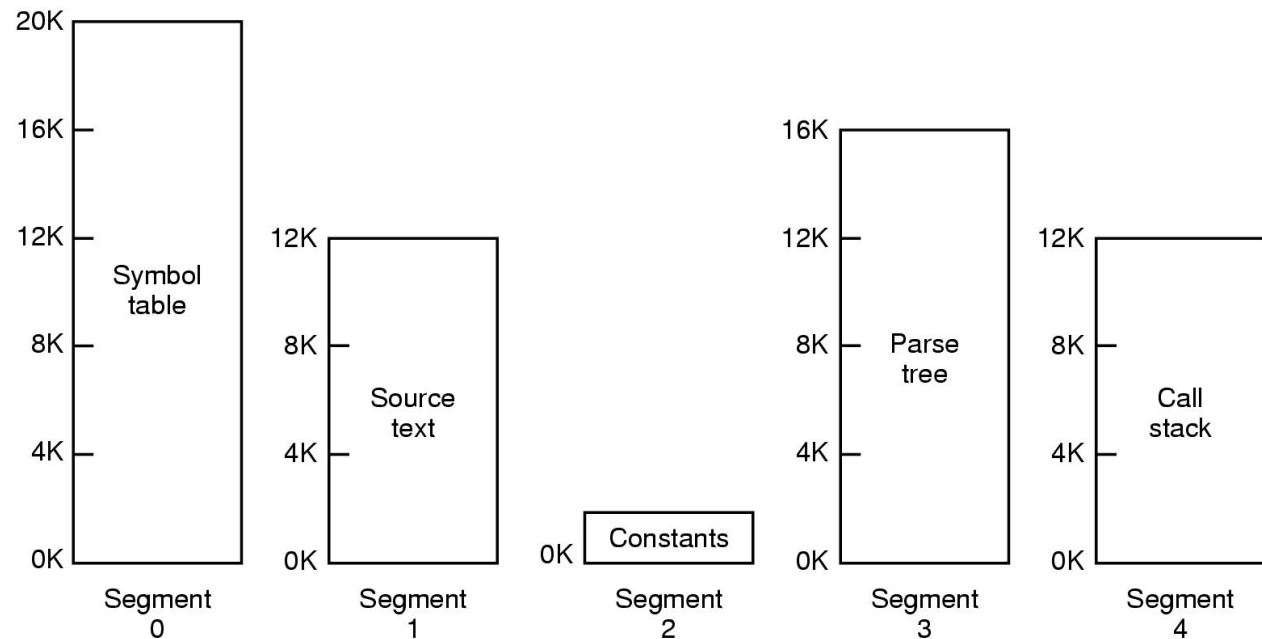


Fig. 9.31 Segmentação.



Memória Virtual

Segmentação

- O processo referencia um endereço de memória virtual $v = (s, d)$.
 - s é o número do segmento na memória virtual.
 - d é o deslocamento dentro do segmento s no qual o item referenciado está localizado.

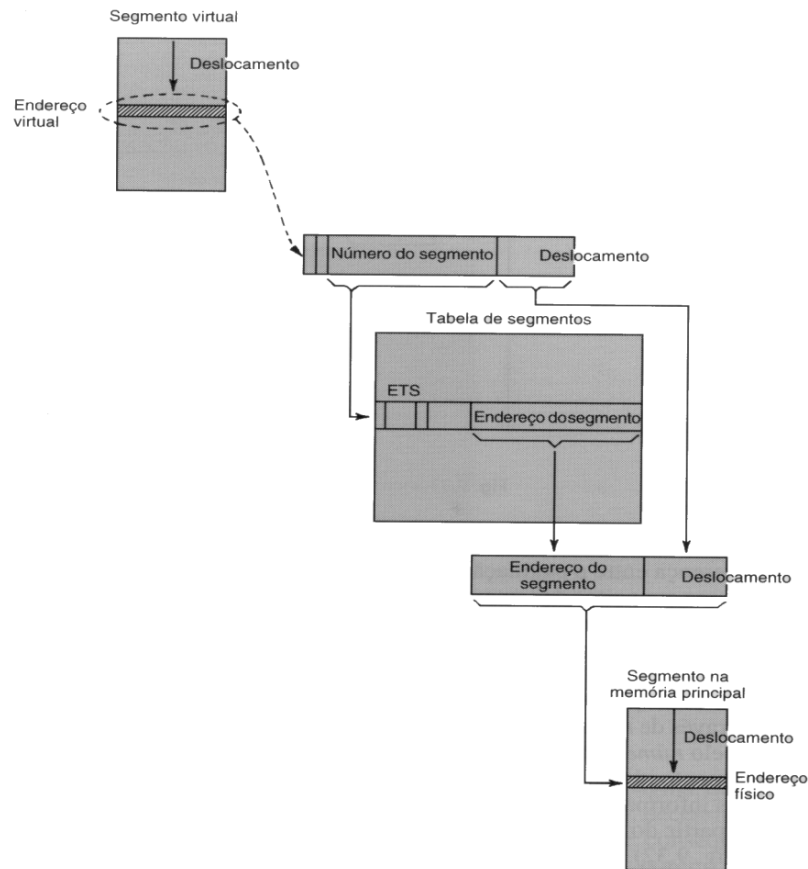


Memória Virtual

Segmentação

- Tradução de Endereço de Segmentação por **Mapeamento Direto**

scoutto



Registrador de origem da
tabela de mapas de
segmentos

Endereço-base, b , da
tabela de segmentos

b

+

$b + s$

s

Tabela de mapa
de segmentos

s'

+

$s' + d$

Endereço virtual
 $v = (s, d)$

Número do
segmento s

Deslocamento
 d

b

d

s'

+

$s' + d$

Endereço real
 $r = s' + d$

r

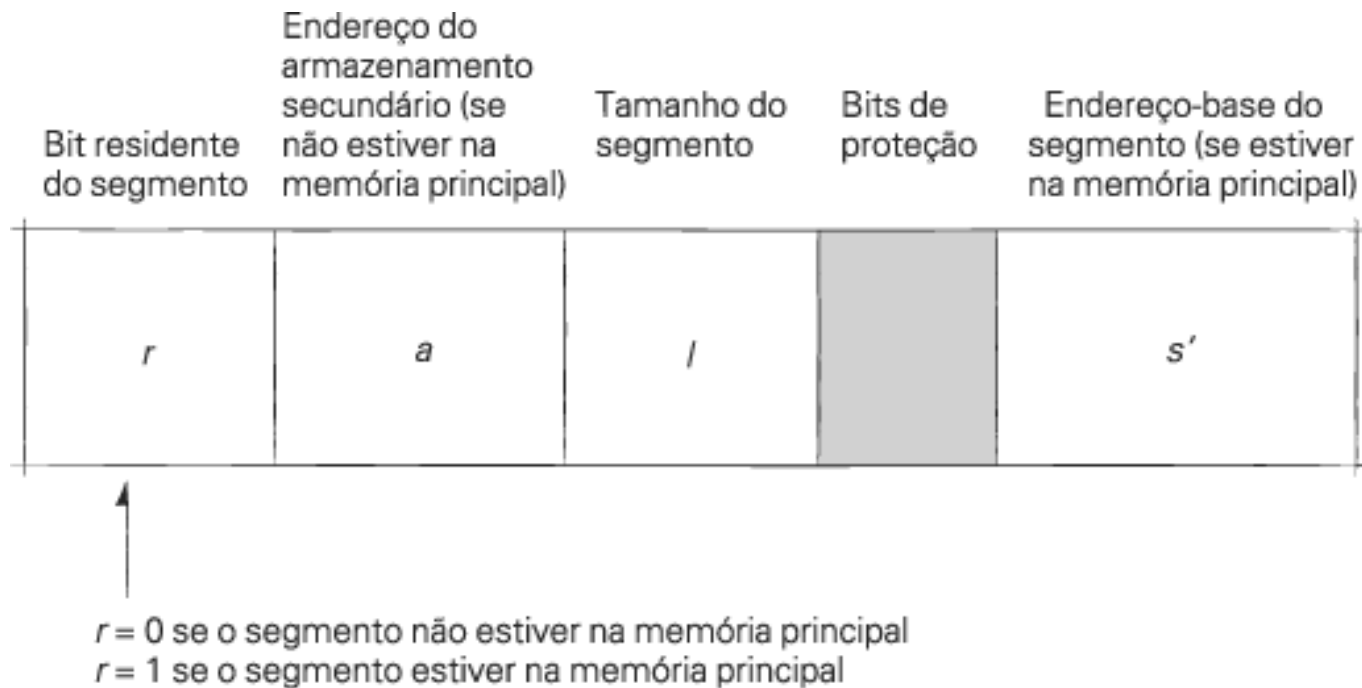
Obs: Não se pode apenas concatenar d com s' , como se faz em sistema de paginação pura, porque os segmentos têm tamanhos diferentes = adicionar.

Fig. 9.32 Tradução do endereço virtual.

Memória Virtual

Segmentação

- Entrada da Tabela de Mapeamento de Segmentos
 - Indica que o segmento s inicia-se no endereço da memória real s' .



Memória Virtual

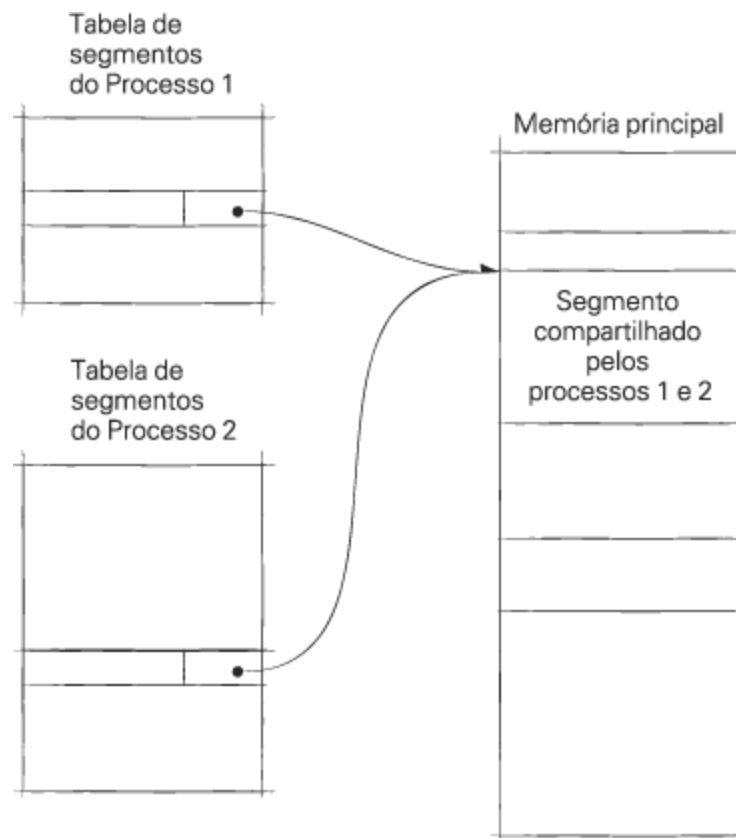
Segmentação

- Compartilhamento em um Sistema de Segmentação
 - A segmentação reduz a sobrecarga de compartilhamento em comparação com o compartilhamento sob paginação pura:
 - Segmentação: Permite um bloco inteiro de memória compartilhada caiba dentro de um só segmento e mantém informações de compartilhamento para um segmento;
 - Já em paginação, poderia consumir diversas páginas, o SO teria que manter informações de compartilhamento para cada página.
 - Além disso, as estruturas de dados podem crescer ou diminuir sem mudar as informações compartilhadas associadas com o segmento da estrutura.

Memória Virtual

Segmentação

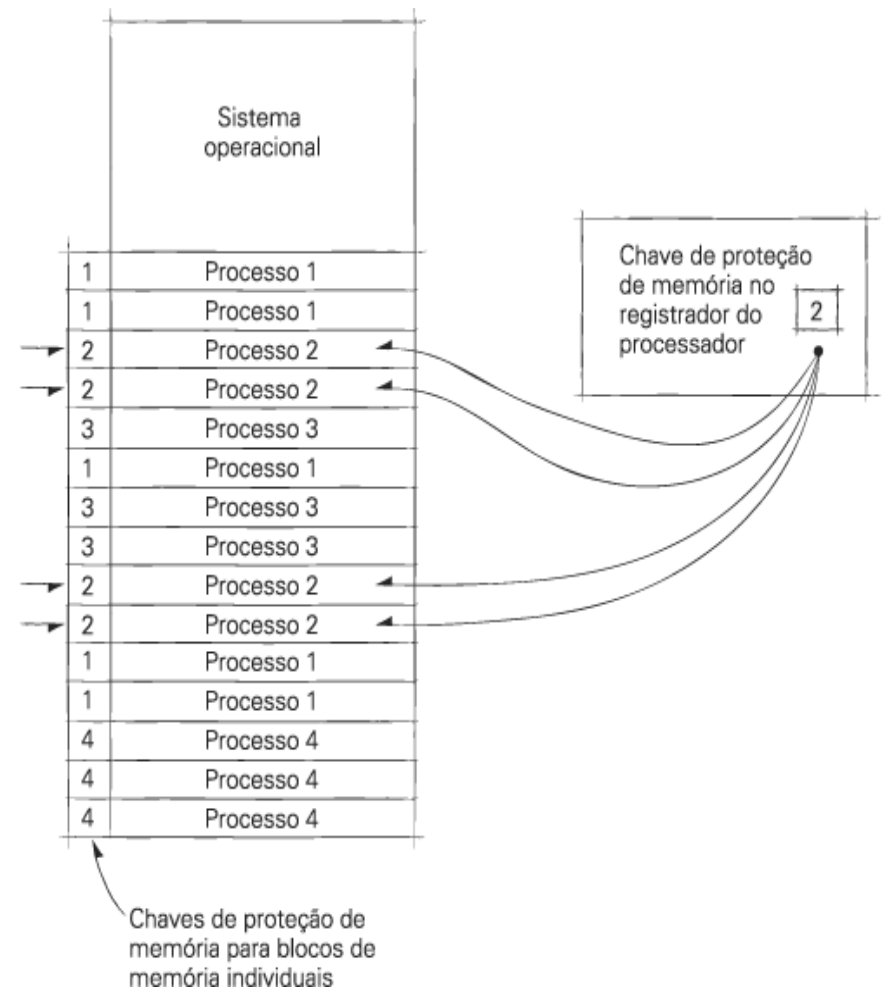
- Compartilhamento em um Sistema de Segmentação Pura.



Memória Virtual

Segmentação

- Proteção e Controle de Acesso em Sistemas de Segmentação
 - Um esquema de implementação de proteção de memória nos sistemas de segmentação são as chaves de proteção de memória.
 - Chave de proteção
 - Está associada ao processo.
 - Se a chave de proteção do processador e do bloco solicitado for a mesma, o processo pode acessar o segmento.

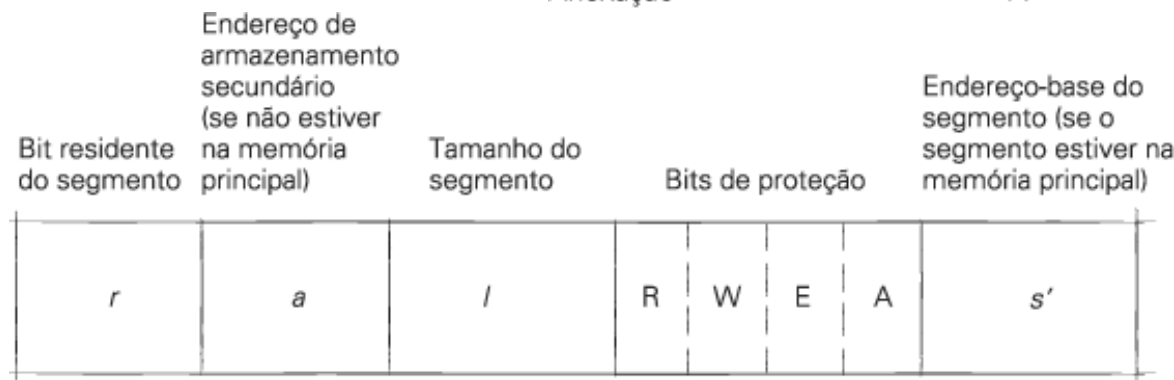


Memória Virtual

Segmentação

- Um esquema mais comum é usar bits de proteção que especificam se um processo pode ler, gravar, executar código ou ser anexado a um segmento.

<i>Tipo de acesso</i>	<i>Abreviatura</i>	<i>Descrição</i>
Leitura	R	Este segmento pode ser lido.
Escrita	W	Este segmento pode ser modificado.
Execução	E	Este segmento pode ser executado.
Anexação	A	Este segmento permite anexação de informações ao seu final.



Bits de proteção: (1=sim, 0=não)

R - Acesso de leitura
W - Acesso de escrita
E - Acesso de execução
A - Acesso de anexação

Memória Virtual

Segmentação

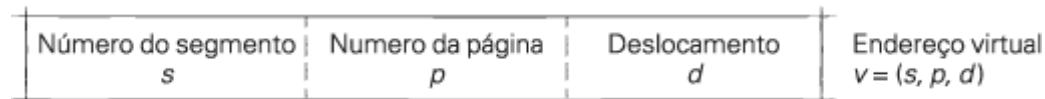
- Combinação de acessos de leitura, escrita e execução que resultam em modos úteis de controle de acesso.

<i>Modo</i>	<i>Leitura</i>	<i>Escrita</i>	<i>Execução</i>	<i>Descrição</i>	<i>Aplicação</i>
Modo 0	Não	Não	Não	Nenhum acesso permitido	Segurança.
Modo 1	Não	Não	Sim	Somente execução	Um segmento é disponibilizado para processos que não podem modificá-lo nem copiá-lo, mas podem executá-lo.
Modo 2	Não	Sim	Não	Somente escrita	Essas possibilidades não são úteis, porque conceder acesso de escrita sem acesso de leitura não é prático.
Modo 3	Não	Sim	Sim	Escrita/execução, mas não pode ser lido	
Modo 4	Sim	Não	Não	Somente leitura	Recuperação de informações.
Modo 5	Sim	Não	Sim	Leitura/execução	Um programa pode ser copiado ou executado, mas não pode ser modificado.
Modo 6	Sim	Sim	Não	Leitura/escrita, mas sem execução	Protege dados de uma tentativa errônea de executá-los.
Modo 7	Sim	Sim	Sim	Acesso irrestrito	Esse acesso é concedido a usuários de confiança.

Memória Virtual

Segmentação com Paginação

- Os segmentos ocupam uma ou mais páginas.
- Nem todas as páginas do segmento precisam estar na memória principal ao mesmo tempo.
- As páginas contíguas na memória virtual não precisam ser contíguas na memória principal.
- O endereço da memória virtual é implementado como uma tripla ordenada $v = (s, p, d)$.
 - **s** é o número do segmento.
 - **p** é o número da página dentro do segmento.
 - **d** é o deslocamento dentro da página na qual o item desejado está localizado.



Memória Virtual

Segmentação com Paginação

- Tradução de endereço virtual com mapeamento combinado associativo/direto em um sistema de segmentação/paginação.

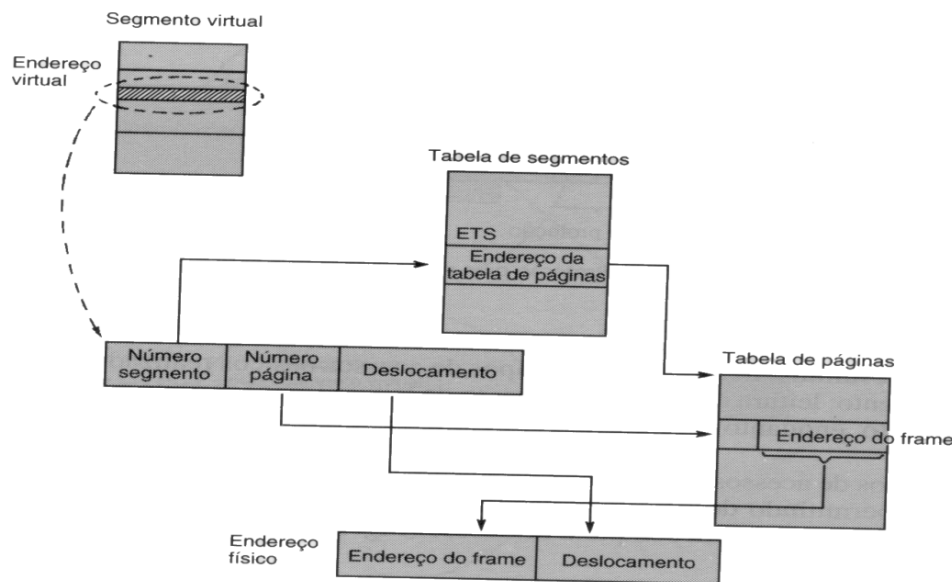
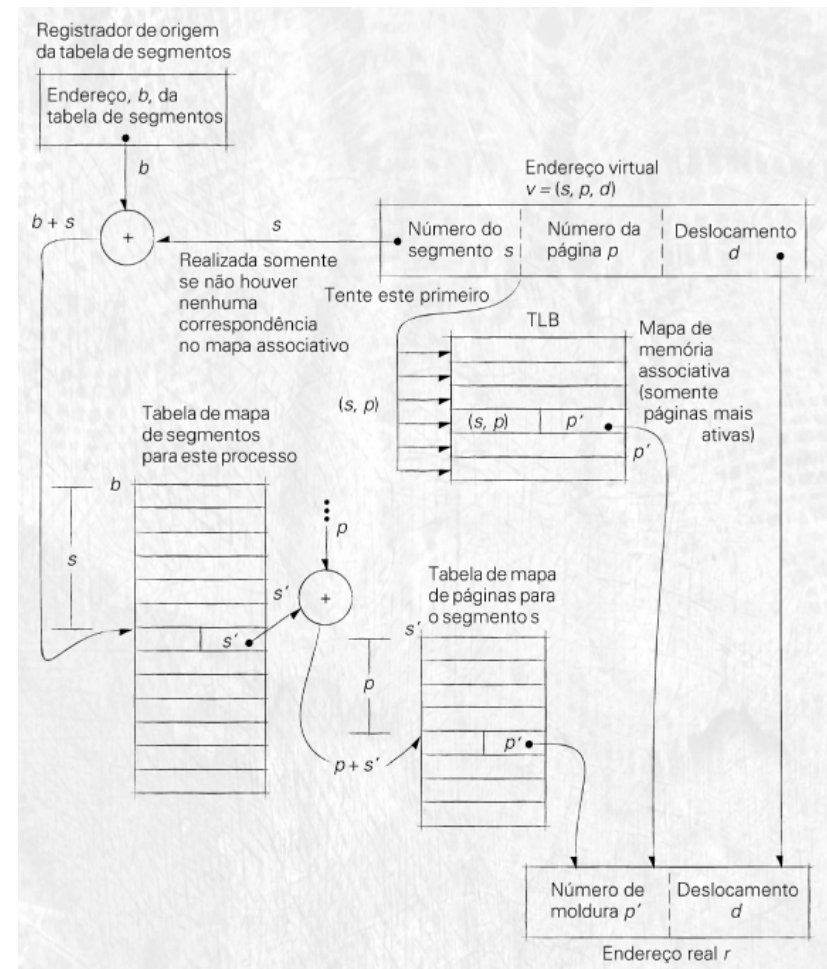


Fig. 9.33 Segmentação com paginação.



Memória Virtual

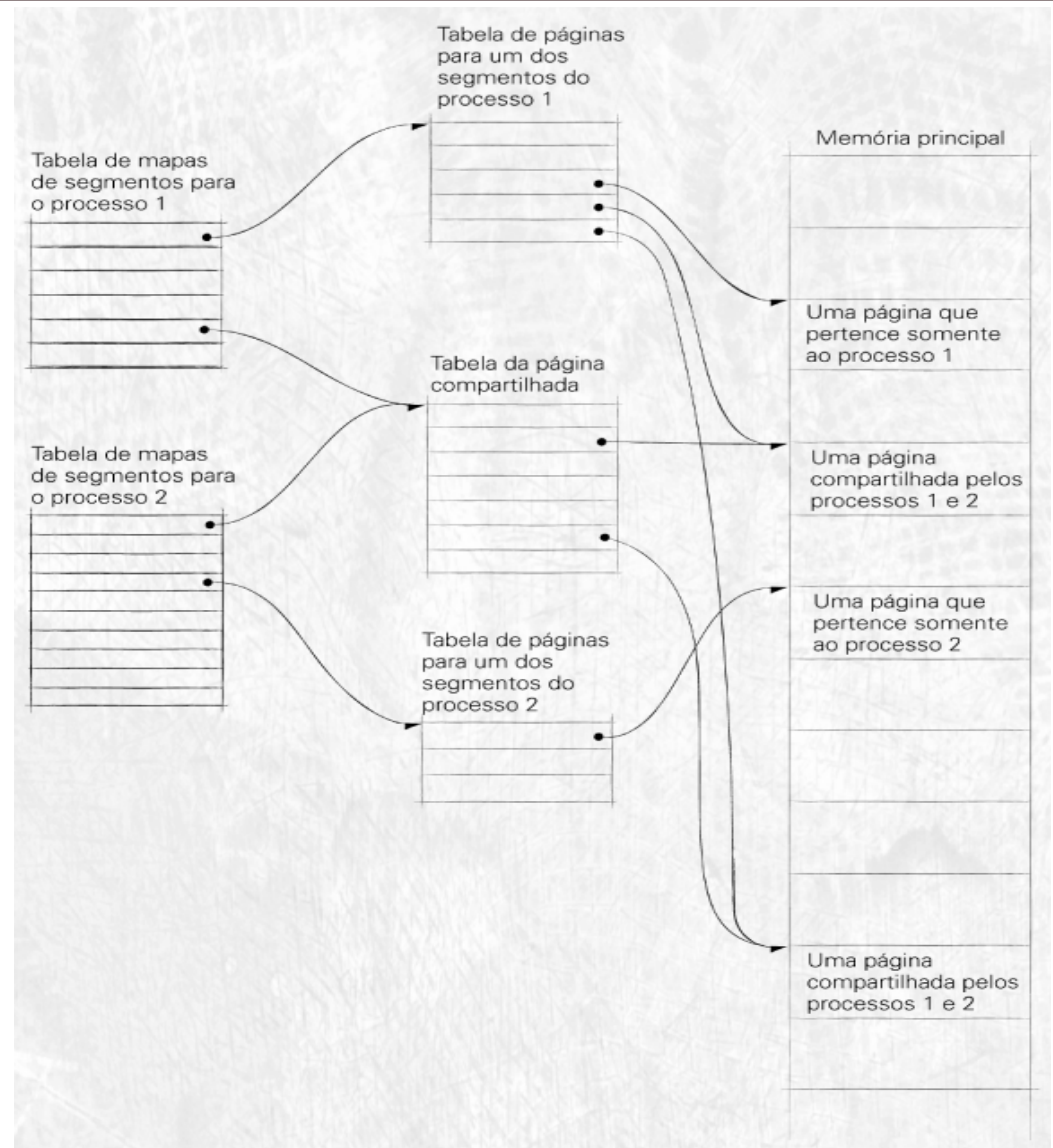
Segmentação com Paginação

- Compartilhamento e proteção em um sistema de segmentação / paginação
 - Dois ou mais processos compartilham memória quando cada processo dispõe de uma entrada de tabela de mapeamento de segmentos que indica a mesma tabela de páginas.
 - O compartilhamento requer um gerenciamento cuidadoso por parte do sistema operacional.
 - Ex: o que aconteceria se uma página que estivesse chegando substituísse uma página compartilhada por muitos processos?
 - O SO teria que atualizar o bit residente das entradas de tabelas de páginas correspondente para cada processo que estivesse compartilhando a página.
 - O SO iria incorrer em substancial sobrecarga para determinar os processos e suas tabelas de páginas para atualização.

Memória Virtual

Segmentação com Paginação

- Dois processos que compartilham um segmento em um sistema de segmentação/paginação.



Memória Virtual

Paginação x Segmentação

Consideração	Paginação	Segmentação
O programador precisa estar ciente de que esta técnica está sendo usada?	Não	Sim
Quantos espaços de endereçamentos lineares existem?	Um	Muitos
O espaço de endereçamento total pode exceder o tamanho da memória física?	Sim	Sim
Os procedimentos e os dados podem ser diferenciados e protegidos separadamente?	Não	Sim
As tabelas com tamanhos variáveis podem ser acomodadas facilmente?	Não	Sim
O compartilhamento de procedimentos entre usuários é facilitado?	Não	Sim
Por que essa técnica foi inventada?	Para fornecer um grande espaço de endereçamento linear sem a necessidade de comprar mais memória física	Para permitir que programas e dados sejam quebrados em espaços de endereçamento logicamente independentes e para auxiliar o compartilhamento e a proteção

Bibliografia

- FLYNN, Ida M.; MCHOES, A. M. **Introdução aos sistemas operacionais**. São Paulo: Pioneira Thomson, 2002, 434 p.
- OLIVEIRA, R. S.; CARISSIMI, A. S.; TOSCANI, S. S. **Sistemas operacionais**. 3 ed. Porto Alegre: Bookman, , 2008. 259p.
- SILBERSCHATZ, A.; GAGNE, G.; GALVIN, P. B. **Fundamentos de sistemas operacionais**, 6 ed. Rio de janeiro: LTC Editora, 2004. 600 p.
- TANENBAUM, A. S.; WOODHULL, A. S. **Sistemas operacionais: projeto e implementação**. 3 ed. Porto Alegre: Bookman, 2008. 992 p.
- TANENBAUM, A. S. **Sistemas operacionais modernos**, 2 ed. São Paulo: Prentice-Hall, 2003. 707 p.
- TOSCANI, S. S.; OLIVEIRA, R.S.; CARISSIMI, A. S. **Sistemas operacionais e programação concorrente**. Porto Alegre: Sagra-Luzzatto, 2003. 247 p.