



Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Disciplina: Sistemas Operacionais I

Aula 18: Gerenciamento de Memória P2

Prof. Diogo Branquinho Ramos

diogo.branquinho@fatec.sp.gov.br

São José dos Campos - SP

Roteiro

- Paginação
 - Tradução de endereço
 - Estrutura da tabela de página
 - Cache da tabela de página
 - Proteção de memória
 - Página compartilhada
- Outras estruturas de tabela de página
- Segmentação

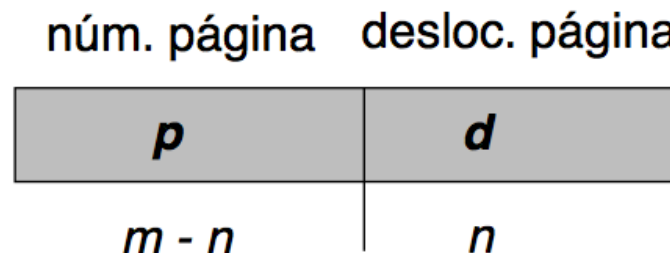
Método da paginação

- **Divisão das memórias em blocos fixos**
 - Memória física dividida em blocos de tamanho fixo, denominados **quadros** ou **frames** (tamanho é potência de 2, entre 512 bytes e 16 MB).
 - Memória lógica dividida em blocos do mesmo tamanho, denominados **páginas**.
 - Armazenamento de apoio dividido em blocos de tamanho fixo igual ao tamanho dos quadros.
- **Para executar um programa com tamanho de n páginas é preciso encontrar n quadros livres.**
- **Tabela de página**
 - Traduz endereços lógicos para físicos.

Esquema de tradução de endereço

- **Endereço lógico**

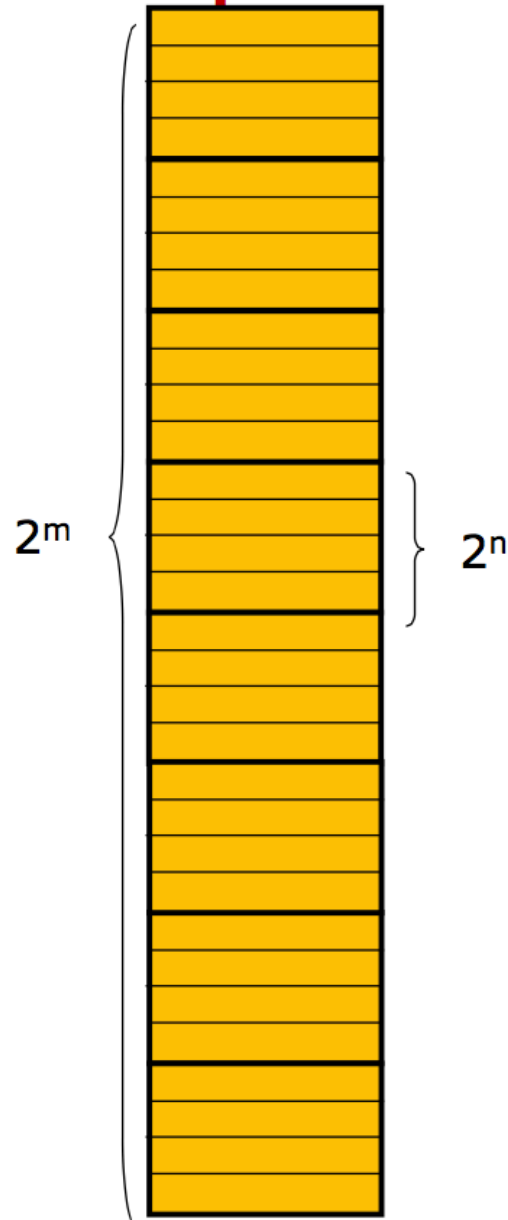
- Número de página (p) – usado como um índice para uma **tabela de página** que contém endereço de base de cada página na memória física.
- Deslocamento de página (d) – combinado com endereço de base para definir o endereço físico de memória que é enviado à unidade de memória.



$m = n^\circ$ de bits para o endereçamento

- Para um espaço de 2^m endereços lógicos com tamanho de página de 2^n endereços, temos 2^{m-n} páginas.

Esquema de tradução de endereço

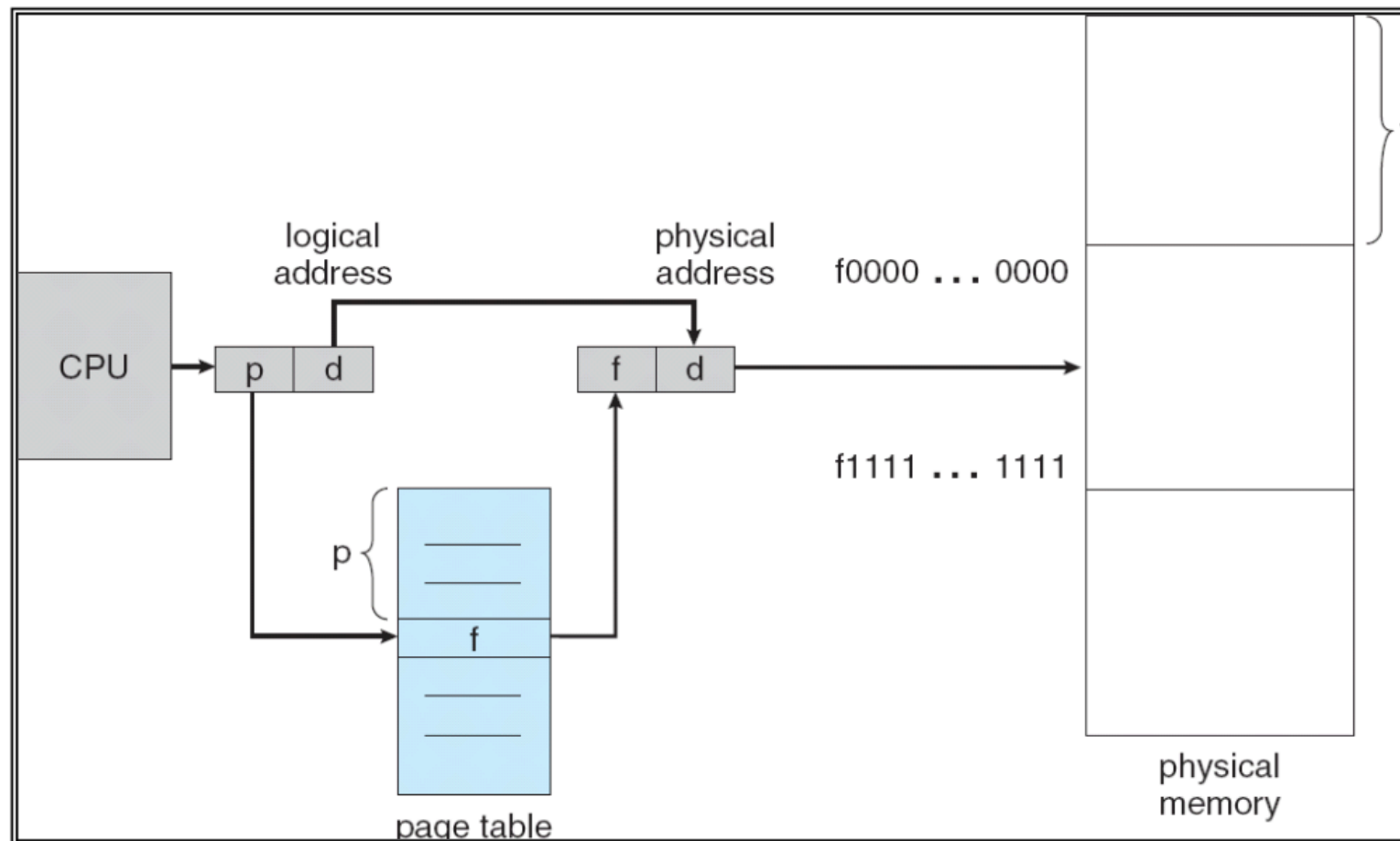


núm. página	desloc. página
<i>p</i>	<i>d</i>
$m - n$	n

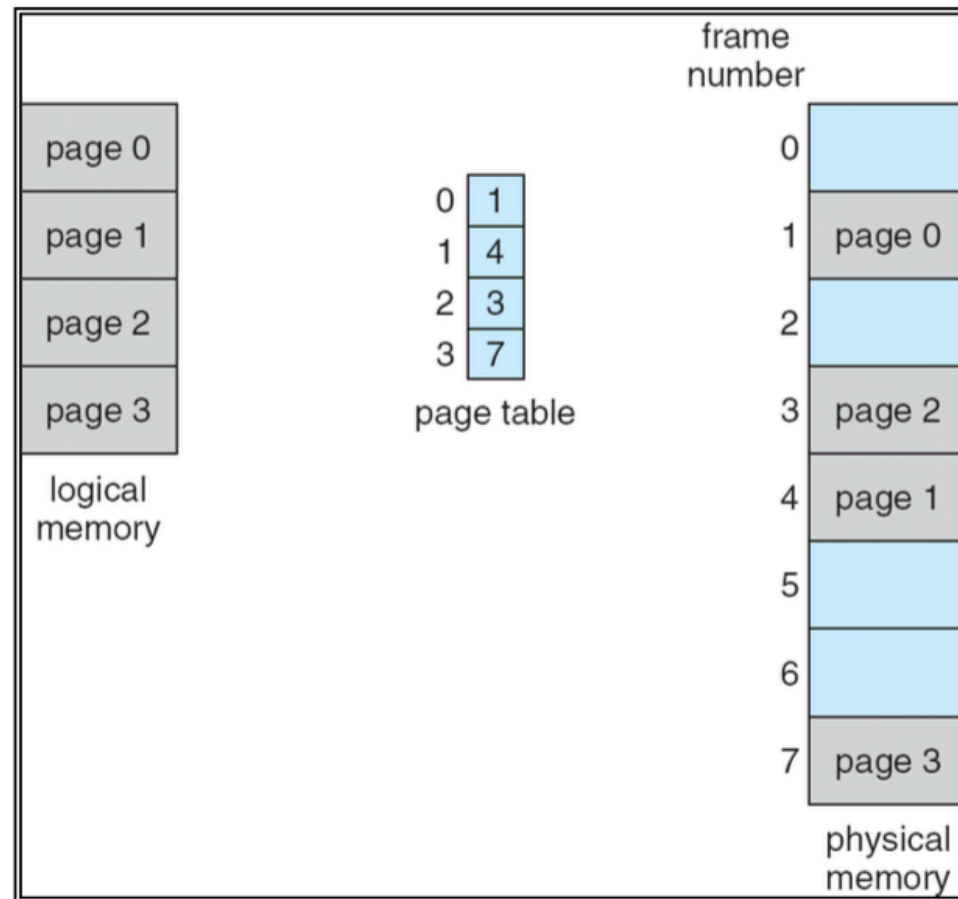
Bits!

$2^m = 32 \rightarrow m=5$
 $2^n = 4 \rightarrow n=2$
 $m - n = 3$ bits p/ representar **p**

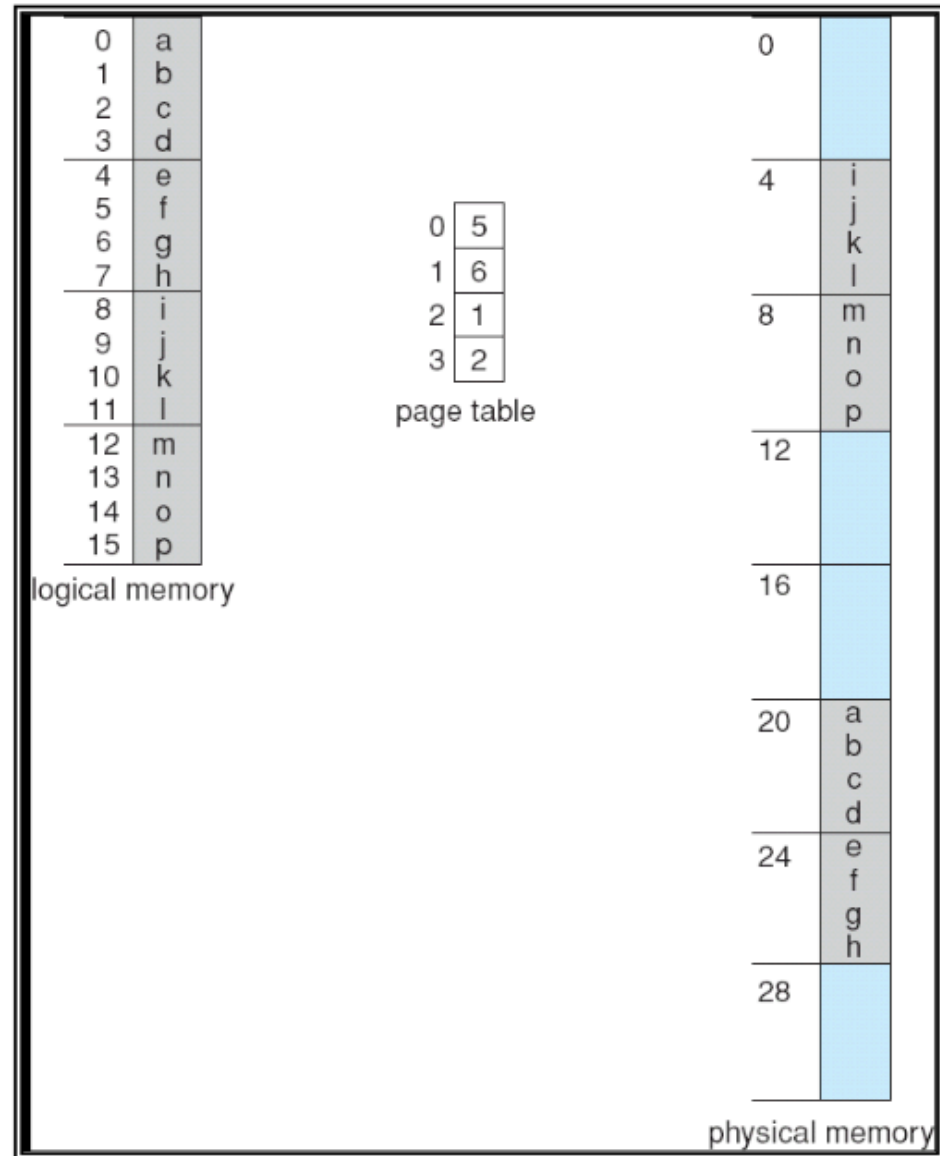
Hardware de paginação



Modelo de paginação da memória



Exemplo de paginação



Memória de 32 bytes e páginas de 4 bytes

Sistemas Operacionais I – Prof. Diogo 2015-1

Paginação

- **A paginação é uma forma de relocação dinâmica.**
 - Cada endereço lógico é associado pelo hardware de paginação a algum endereço físico.
 - O uso da página é semelhante ao uso de uma tabela de registradores de base, um para cada quadro de memória.
- **Esquema de paginação não oferece fragmentação externa.**
 - Qualquer quadro livre pode ser alocado a um processo que precisar dele.
- **Porém, pode gerar fragmentação interna.**
 - Os quadros são as unidades de espaço.

Paginação

- **Fragmentação interna**

- Ex: tamanho de página: 2048 bytes; processo: 72766 bytes. Ele precisaria de 35 páginas mais 1086 bytes. Receberia 36 quadros.
- Neste caso, qual o valor da fragmentação interna?
 - $2048 - 1086 = 962$ bytes.
- Qual o pior caso?
 - O processo precisar de n quadros + 1 byte: ele recebe $n + 1$ quadros. Fragmentação de quase um quadro inteiro.

- **Tamanho dos quadros**

- Páginas menores: fragmentação menor; maior tabela.
- Páginas maiores: maior eficiência de E/S; maior fragmentação.

Paginação

- **Exercício:**

- Quantos bytes podem ser endereçados pela memória física se o tamanho do quadro for 4 KB e sua tabela de páginas tiver 4 bytes (32 bits) para o endereço físico?

- $2^{32} \times 2^{12} = 2^{44} = 16 \text{ TB}.$

- **Aspecto de segurança importante**

- Separação clara entre a visão da memória pelo usuário e a memória física real.
 - O programa do usuário vê a memória como um recurso só dele.
- O processo é incapaz de acessar a memória que ele não possui: não tem como endereçar a memória fora de sua tabela de página – ela só inclui suas páginas!

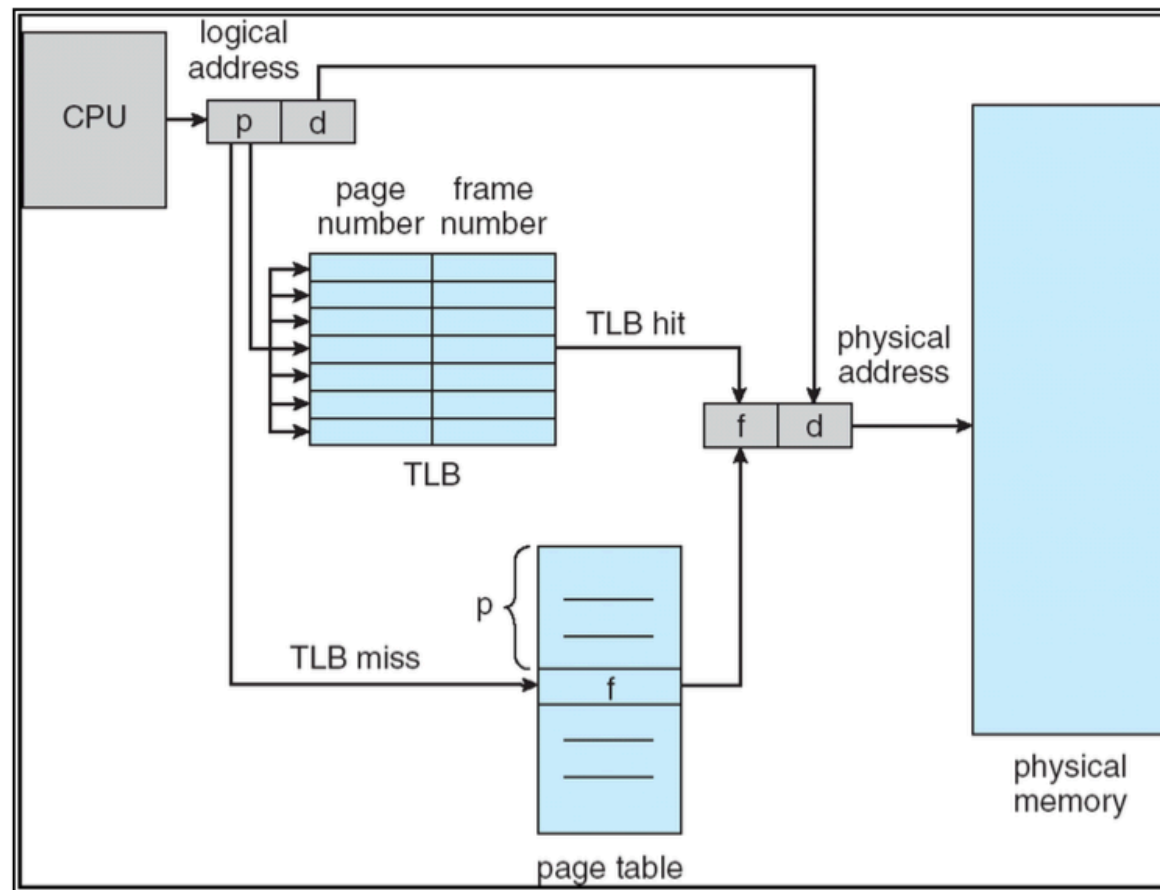
Implementação da Tabela de Página

- **Máquinas mais antigas**
 - Tabela mapeada em registradores dedicados.
 - Viável para tabelas pequenas (256 entradas).
- **Atualmente**
 - Tabelas maiores: mantida na memória principal.
 - 1 milhão de entradas.
 - PTBR: registrador de base da tabela de página.
 - Ponteiro para o início da tabela.
 - Nesse esquema, cada acesso de dado/instrução exige dois acessos à memória: um para a tabela de página e um para o dado/instrução.

Implementação da Tabela de Página

- **Solução**
 - O problema dos dois acessos à memória pode ser amenizado pelo uso de um cache de hardware especial para pesquisa rápida.
 - Chamada de **memória associativa** ou **translation look-aside buffers** (TLBs).
 - Cache da Tabela de Páginas. Guarda uma chave (página) e um valor (frame).
- **TLB hit: número da página está na TLB.**
- **TLB miss (falha de TLB): número de página não está da TLB.**
- **TLB cheia**
 - Políticas de substituição: LRU (usado menos recentemente) e aleatório.
 - Algumas entradas podem ser fixadas (kernel).

Hardware de paginação com TLB



Implementação da Tabela de Página

- **Novo problema!**
 - Toda troca de contexto exige realimentação da TLB.
 - Sem isso, toda vez que uma tabela de páginas for selecionada, a TLB precisa ser esvaziada para garantir que o próximo processo não use informações erradas.
- **Solução**
 - Incluir uma coluna na cache para identificar o processo.
 - Identifica exclusivamente cada processo para fornecer proteção do espaço de endereço para esse processo.
 - Permite que a TLB contenha entradas para vários processos diferentes ao mesmo tempo.

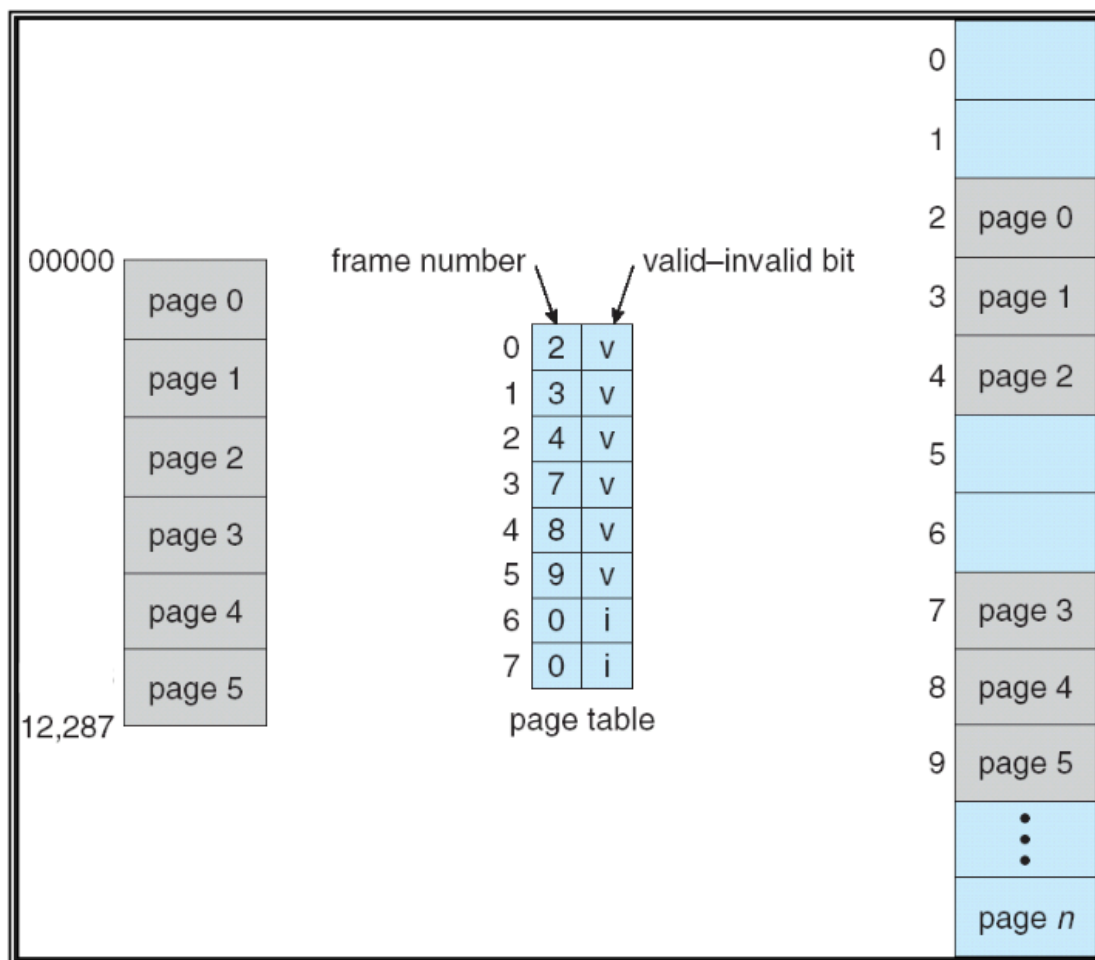
Tempo de Acesso Efetivo (T_{AE})

- Pesquisa na TLB = ε unidade de tempo
- Tempo de ciclo da memória = t_M .
- Taxa de acerto (α)
 - Porcentagem de vezes que um número de página é encontrado nos registradores associativos.
 - Razão relacionada ao número de registradores associativos.
- Tempo para acesso à memória mapeada: $t_M + \varepsilon$.
- Se não encontramos a página na TLB: $2.t_M + \varepsilon$.
- $T_{AE} = (t_M + \varepsilon). \alpha + (2.t_M + \varepsilon).(1 - \alpha)$
 - Pesa os casos por sua probabilidade.
 - Ex.: $\alpha=0,8$; $\varepsilon = 20$ ns; $t_M = 100$ ns.

Proteção de memória

- **Proteção de memória implementada associando-se um bit de proteção a cada quadro.**
 - É possível ter mais bits para definir uma página como leitura/escrita ou somente leitura.
- **Outro bit: válido-inválido anexado a cada entrada na tabela de página:**
 - **Válido:** indica que a página associada está no espaço de endereço lógico do processo, e por isso é uma página válida.
 - **Inválido:** indica que a página não está no espaço de endereço lógico do processo.

Bit de válido-inválido



Espaço de endereços: 14 bits
Tamanho da página: 2KB = 2^{11} bytes.

Número de páginas: $2^3 = 8$

→ $14 - 11 = 3$ bits

→ Tabela de página: 8 linhas

Programa: 12288 bytes

Páginas necessárias: 6

→ $12288 / 2048 = 6$

Páginas de 0 a 5 são **válidas**.

Páginas 6 e 7 são **inválidas**.

Qualquer tentativa de gerar um endereço nas páginas 6 ou 7 gerará um erro para o SO.

Páginas compartilhadas

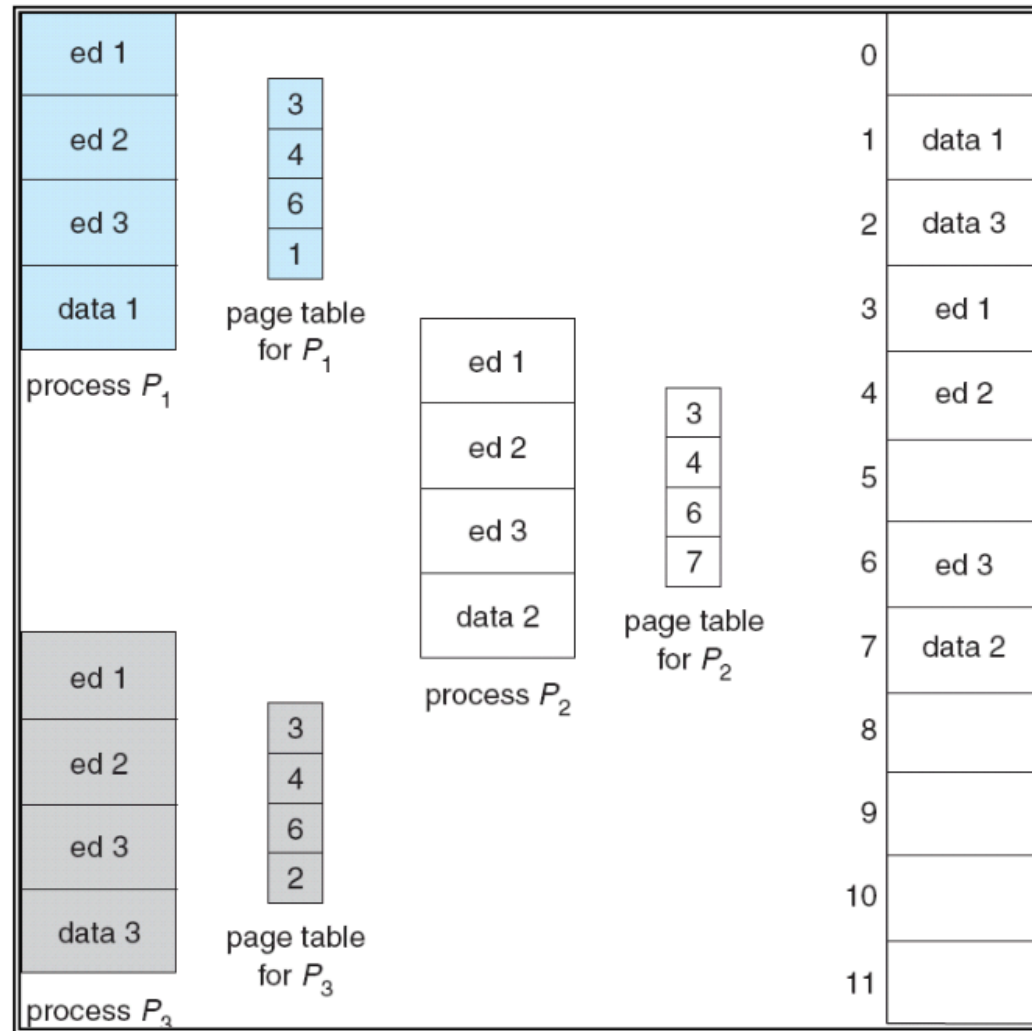
- **Código compartilhado**

- Uma cópia de código somente de leitura pode ser compartilhado entre processos (por exemplo, editores de texto, compiladores, sistemas de janela).
- Código compartilhado deve aparecer no mesmo local no espaço de endereço lógico de todos os processos.

- **Código e dados privados**

- Cada processo mantém uma cópia separada do código e dados privados.
- As páginas para o código e dados privados podem aparecer em qualquer lugar no espaço de endereço lógico.

Exemplo de páginas compartilhadas



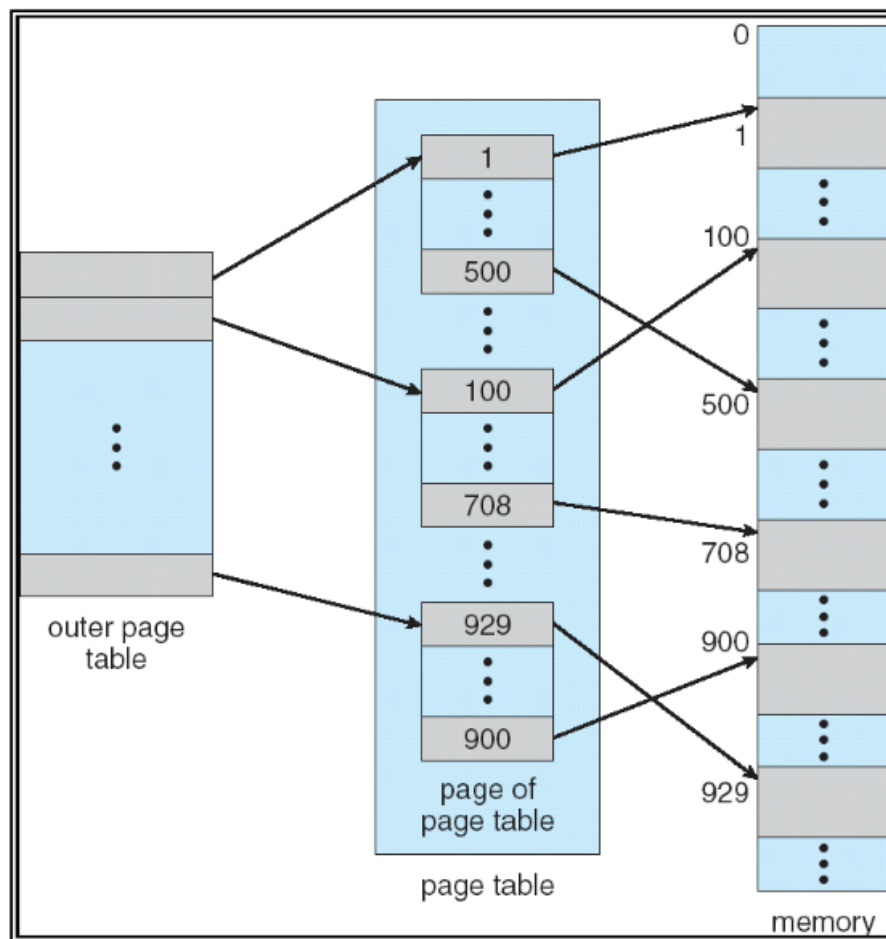
Outras estruturas para tabela de página

- **Paginação hierárquica**
- **Tabela de página em hash**
- **Tabela de página invertida**

Paginação hierárquica

- **A tabela de página pode ser bem grande: ~ 4 MB.**
 - Nem sempre será possível aloca-la de forma contígua na RAM!
- **Solução:**
 - Dividir a tabela em pedaços menores, quebrando o espaço de endereço lógico em múltiplas tabelas de página.
 - Uma técnica simples é uma tabela de página em dois níveis usada em máquinas de 32 bits;
 - Algumas máquinas de 64 bits usam tabela de página em três níveis.

Esquema de tabela de página em dois níveis



Exemplo de paginação em dois níveis

- **Máquinas de 32 bits com tamanho de página de 4 KB**
 - Um endereço lógico é dividido em:
 - Um número de página contendo 20 bits.
 - Um deslocamento de página contendo 12 bits.
 - Se a tabela de página for paginada, o número de página é dividido ainda em:
 - Um número de página de 10 bits.
 - Um deslocamento de página de 10 bits.

Exemplo de paginação em dois níveis

- Assim, um endereço lógico é o seguinte:

núm. página		desloc. página
p_1	p_2	d
10	10	12

- p_1 é um índice para a tabela de página mais externa;
- p_2 é o deslocamento da página dentro da tabela de página mais externa.

Esquema de tradução de endereço

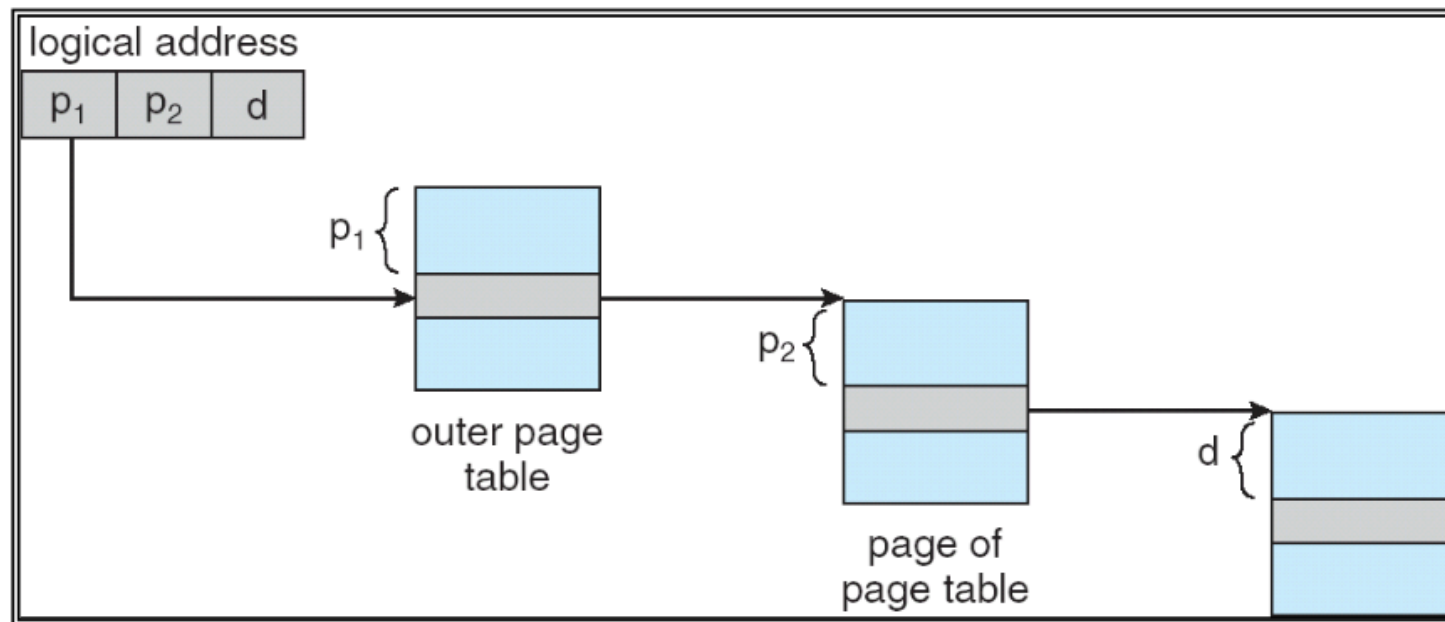


Tabela de página em hash

- **Definição**

- É uma estrutura de dados composta de um vetor de vetores que permite reorganizar os dados e diminuir a busca. Tem uma função que indica a linha no hash.
- Ex.: Função: linha = $V \% 10$

Valor	Resto
13	3
27	7
32	2
7	7
...	...

0		
1		
2	32	
3	13	
4		
5		
6		
7	27	7

Importância da Função:
Quanto mais os dados
tiverem distribuídos,
melhor!

Tabela de página em hash

- **Comuns em espaços de endereço > 32 bits**
- **O número de página virtual é dividido em uma tabela de página.**
 - Essa tabela de página consiste em uma cadeia de elementos traduzidos para o mesmo índice.
- **Números de página virtual são comparados nessa cadeia buscando uma combinação.**
 - Se uma combinação for achada, o quadro físico correspondente é extraído.
- **Cada entrada na tabela hash corresponde a várias páginas.**
 - A busca é um fator negativo.

Tabela de página em hash

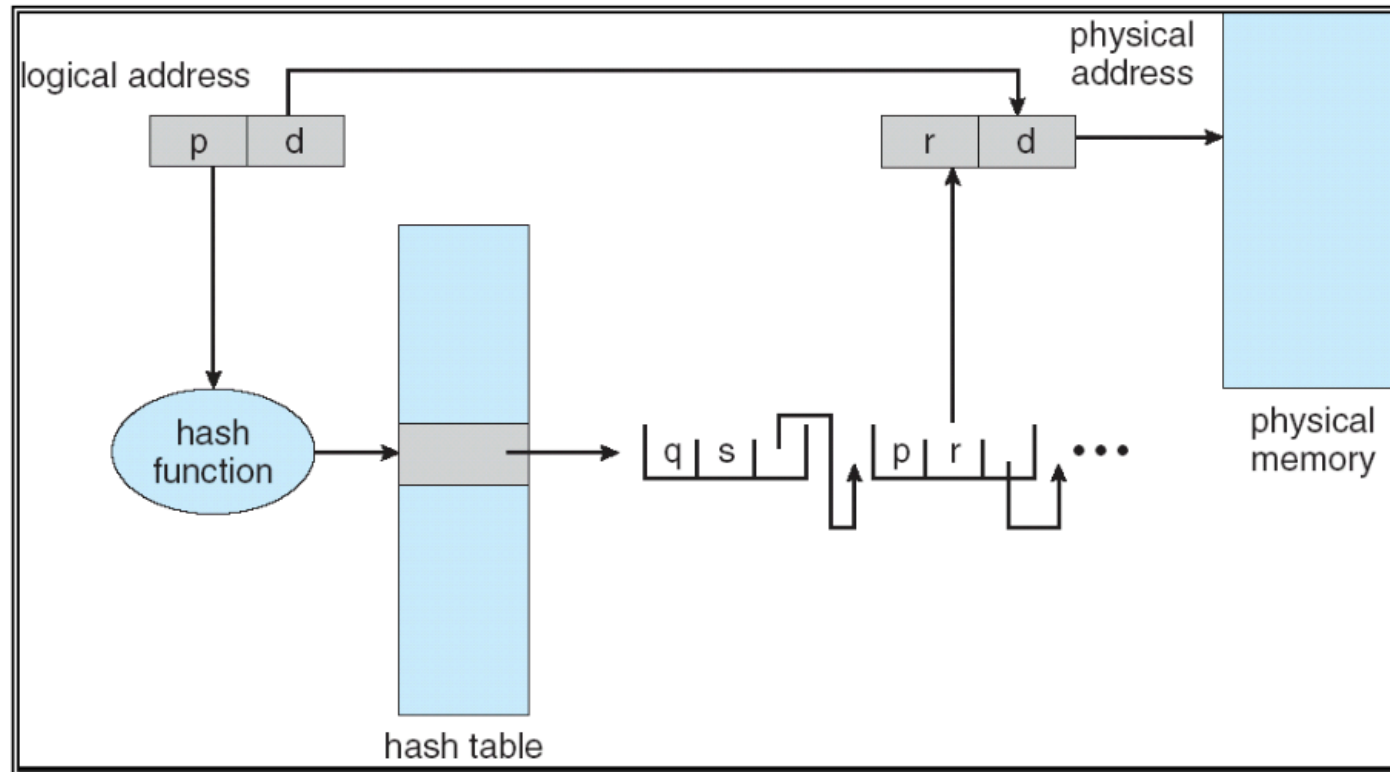


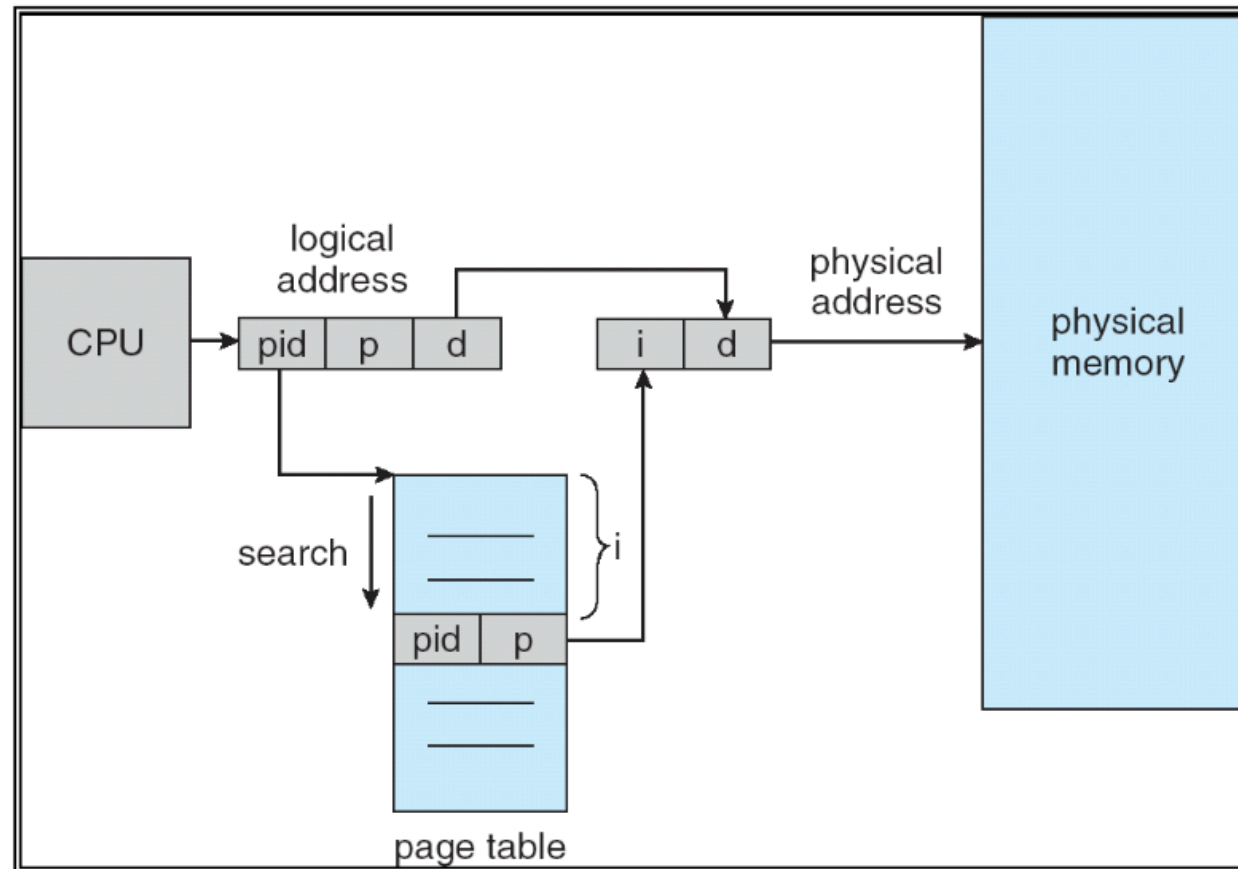
Tabela de página invertida

- **Até agora: tabela de página normal!**
 - Cada processo possui uma tabela de página associada.
 - A tabela de página possui uma entrada para cada página que o processo esteja usando (ou para cada endereço virtual, independente da validade deste).
 - Os processos referenciam páginas por meio dos endereços virtuais das páginas.
 - O SO traduz, então, essa referência para um endereço da memória física.
 - Desvantagem: cada tabela de página pode consistir em milhões de entradas consumindo grande quantidade de memória física para registrar como a outra memória física está sendo usada.

Tabela de página invertida

- **Possui uma entrada para cada frame de memória.**
- **Entrada consiste em:**
 - Página armazenada nesse local da memória real; e
 - PID do processo que possui essa página.
- **Somente uma tabela de página está no sistema.**
 - Diminui a memória necessária para armazenar cada tabela de página, mas aumenta o tempo necessário para pesquisar a tabela quando ocorre uma referência de página.
- **Pode usar tabela hash para limitar a busca nas entradas de tabela de página.**

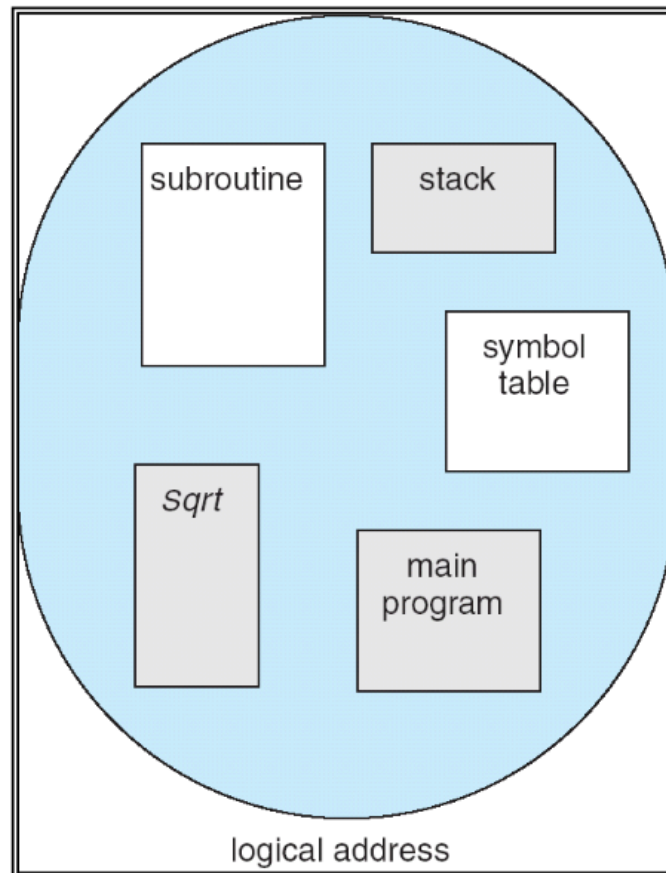
Arquitetura de tabela de página invertida



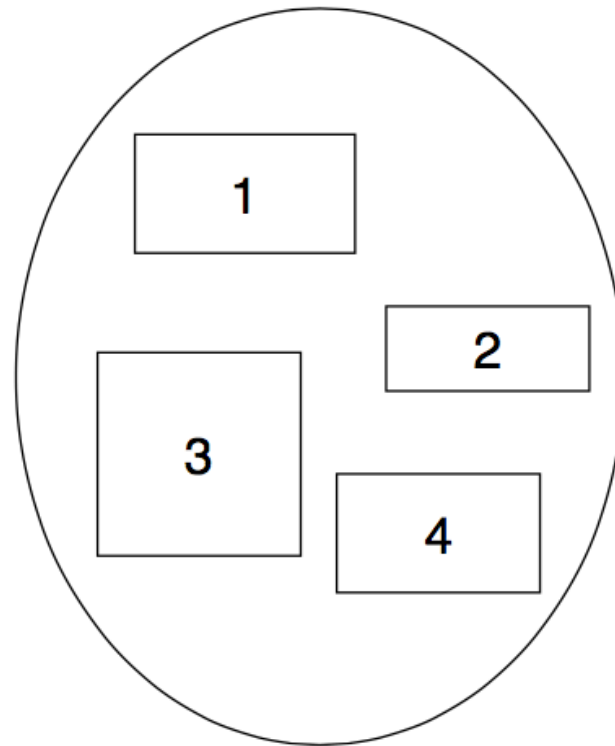
Segmentação

- **Esquema de gerenciamento de memória que admite visão da memória pelo usuário.**
- **Um programa é uma coleção de segmentos. Um segmento é uma unidade lógica como:**
 - programa principal
 - procedimento
 - função
 - método
 - objeto
 - variáveis locais, variáveis globais
 - bloco comum
 - pilha
 - tabela de símbolos
 - arrays

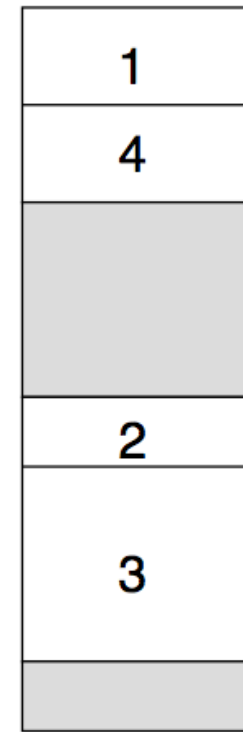
Visão de um programa pelo usuário



Visão lógica da segmentação



espaço do usuário



espaço da memória física

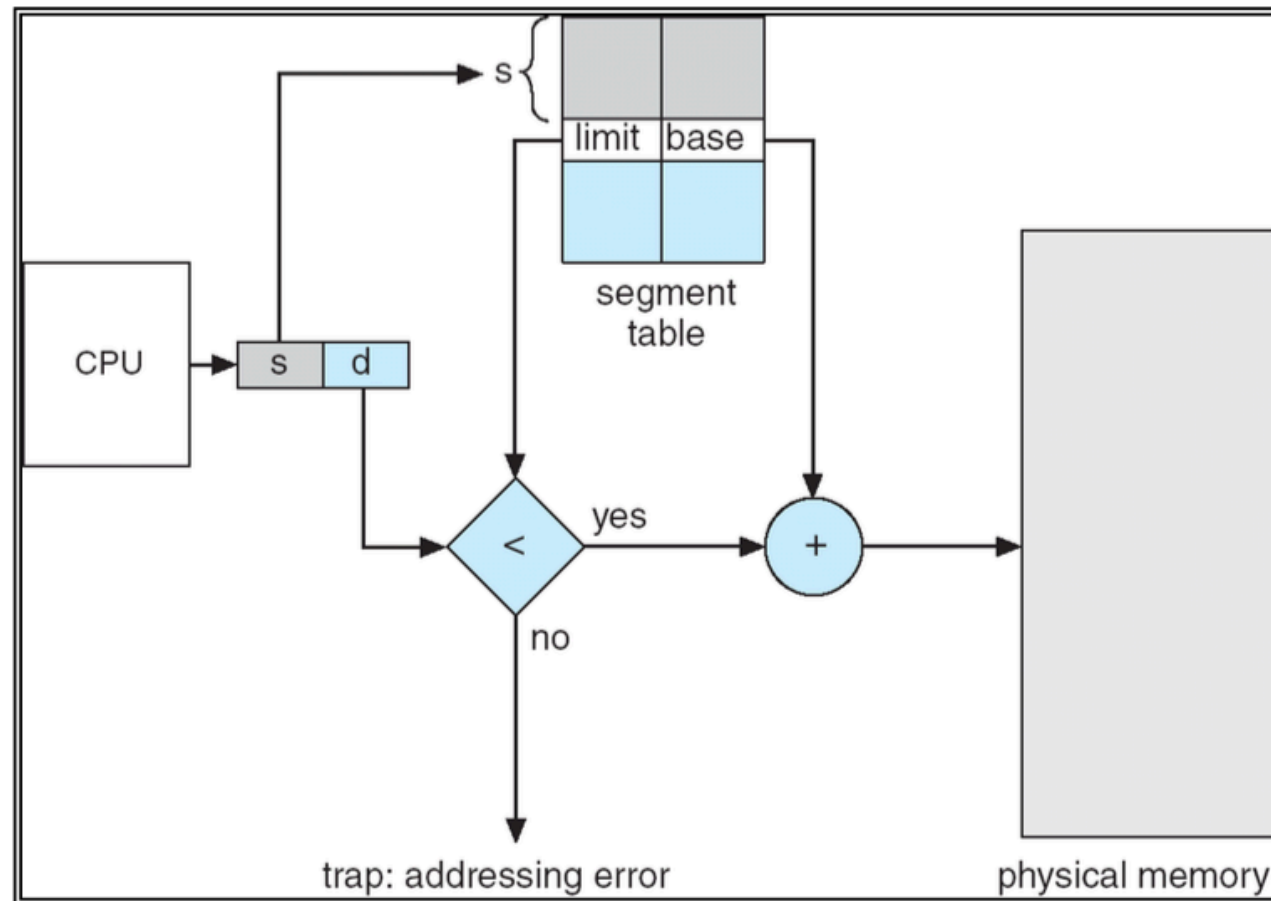
Arquitetura da segmentação

- **Endereço lógico consiste em uma tupla de 2 elementos:**
 - **s e d**
- **Tabela de segmento:**
 - Mapeia endereços físicos bidimensionais.
- **Cada entrada de tabela tem:**
 - base: contém o endereço físico inicial onde os segmentos residem na memória.
 - limite: especifica o tamanho do segmento.
- **Registrador de base da tabela de segmento (STBR) aponta para o local da tabela de segmento na memória**
- **Registrador de tamanho da tabela de segmento (STLR) indica o número de segmentos usados por um programa;**
 - número de segmento **s** é válido se **$s < \text{STLR}$**

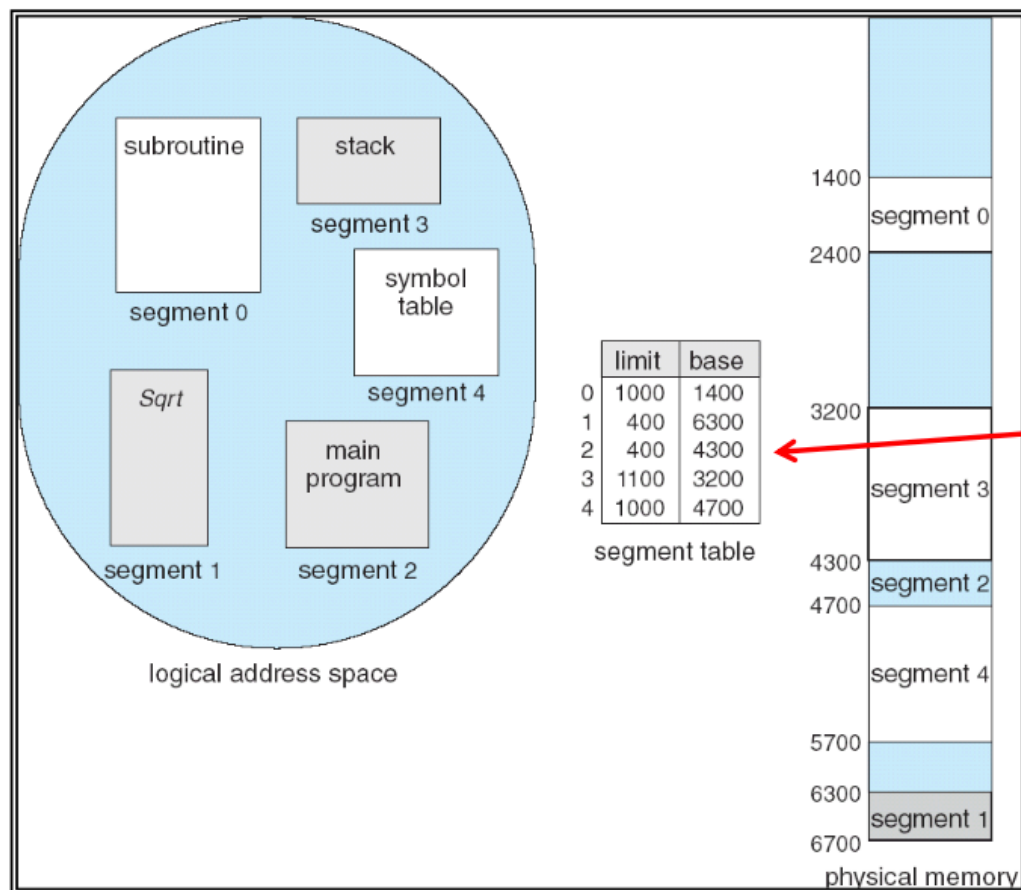
Arquitetura da segmentação

- **Proteção**
 - A cada entrada na tabela de segmento, pode ser associado:
 - bit de validação = 0 → segmento ilegal
 - privilégios read/write/execute
- **Bits de proteção associados a segmentos**
 - O compartilhamento de código ocorre no nível de segmento.
- **Como os segmentos variam em tamanho, a alocação de memória é um problema de alocação dinâmica de armazenamento.**
 - Paginação é alocação estática.

Hardware de segmentação



Exemplo de segmentação



A Tabela de Segmento possui não só referência de onde começa o frame, mas também do seu tamanho.