

Curso de Ciência da Computação
Pontifícia Universidade Católica de Minas Gerais

Sistemas Operacionais

Capítulo IX – Gerência de Memória

Fundamentos

- ◆ Programas devem ser trazidos à memória e alocados a um processador para serem executados.
- ◆ *Fila de Entrada* – coleção de processos no disco esperando para serem trazidos à memória e executados.

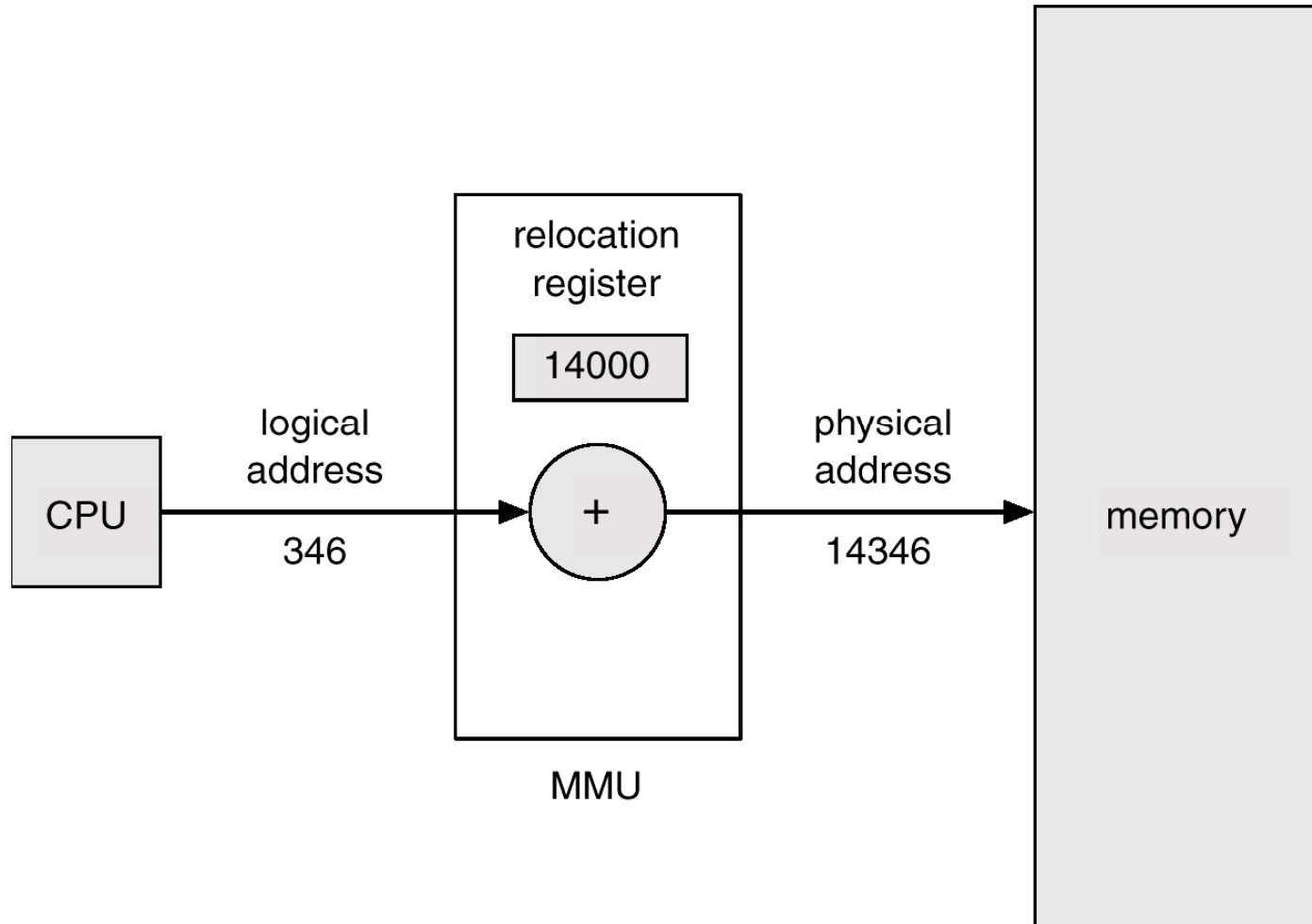
Espaço de endereçamento lógico e físico

- ◆ O conceito de espaço de endereçamento lógico limitado a um espaço de endereçamento físico é central para o gerenciamento de memória.
 - *Endereço lógico* – gerado pela CPU; também denominado *endereço virtual*.
 - *Endereço físico* – endereço visto pela unidade de memória.
- ◆ Endereços lógicos e físicos são os mesmos em tempo de compilação e carga. Eles se diferem apenas no tempo de execução.

Unidade de Gerenciamento de Memória (MMU)

- ◆ Dispositivo de hardware que mapeia o endereço virtual para o físico.
- ◆ No esquema da MMU, o valor no registro de relocação é adicionado a todo endereço produzido pelo processo no momento que é enviado à memória
- ◆ O programa do usuário lida com endereços lógicos, sem nunca saber a quais endereços físicos eles se referem.

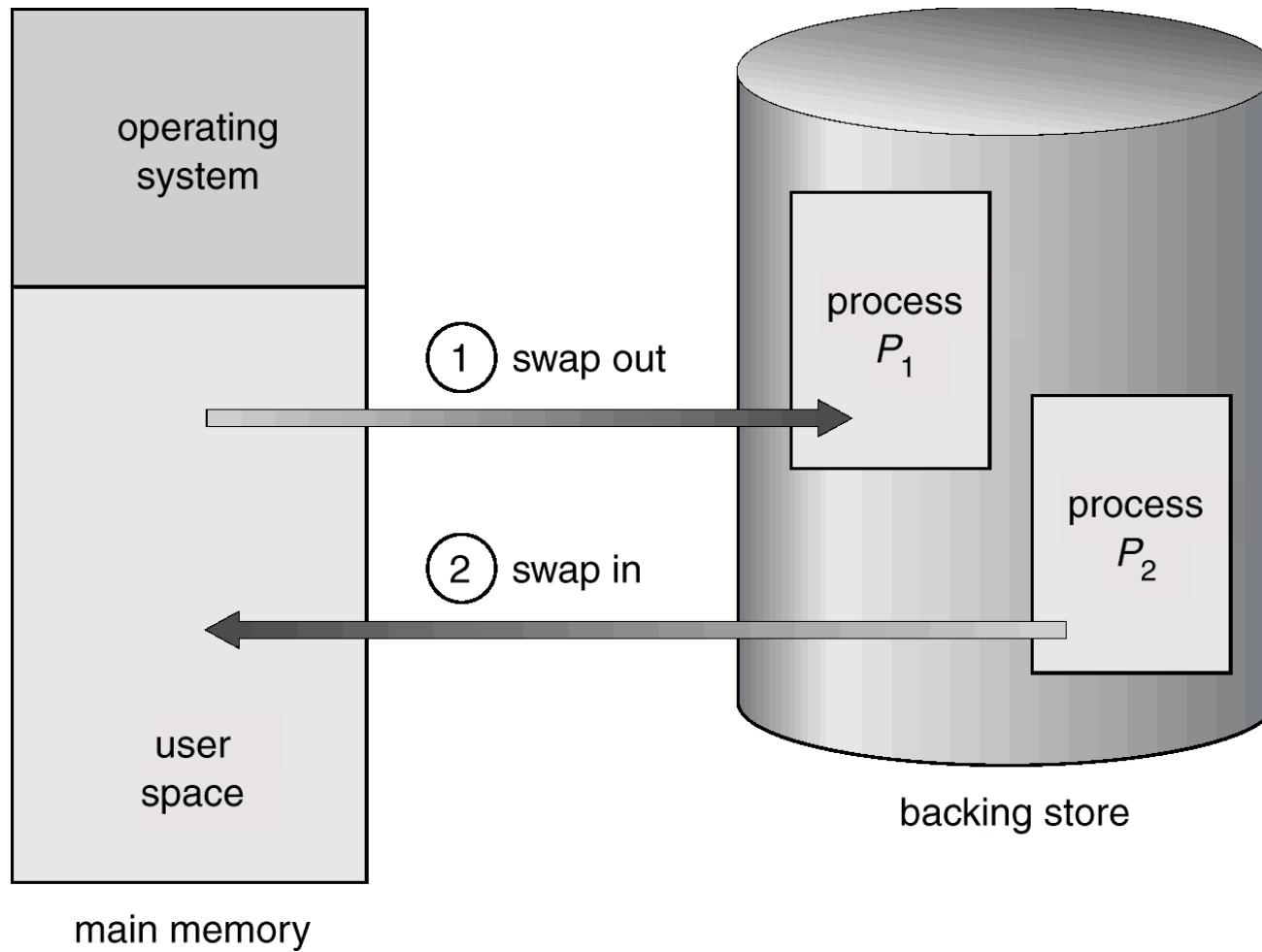
Espaço de endereçamento lógico e físico



Swapping

- ◆ Um processo pode ser removido temporariamente da memória para um dispositivo de armazenamento auxiliar, e trazido de volta à memória quando for retomar sua execução.
- ◆ Armazenamento auxiliar – disco rápido e grande o suficiente para guardar cópias das imagens de memória de todos os usuários. Deve prover acesso direto a estas imagens.
- ◆ *Roll out, roll in* – variante do swapping usado para algoritmos de escalonamento baseados em prioridades; processos de baixa prioridade são removidos para dar lugar a processos de maior prioridade.
- ◆ A maior parte do tempo de swap é de transferência; o tempo total é diretamente proporcional à quantidade de memória transferida.

Visão esquemática do Swapping

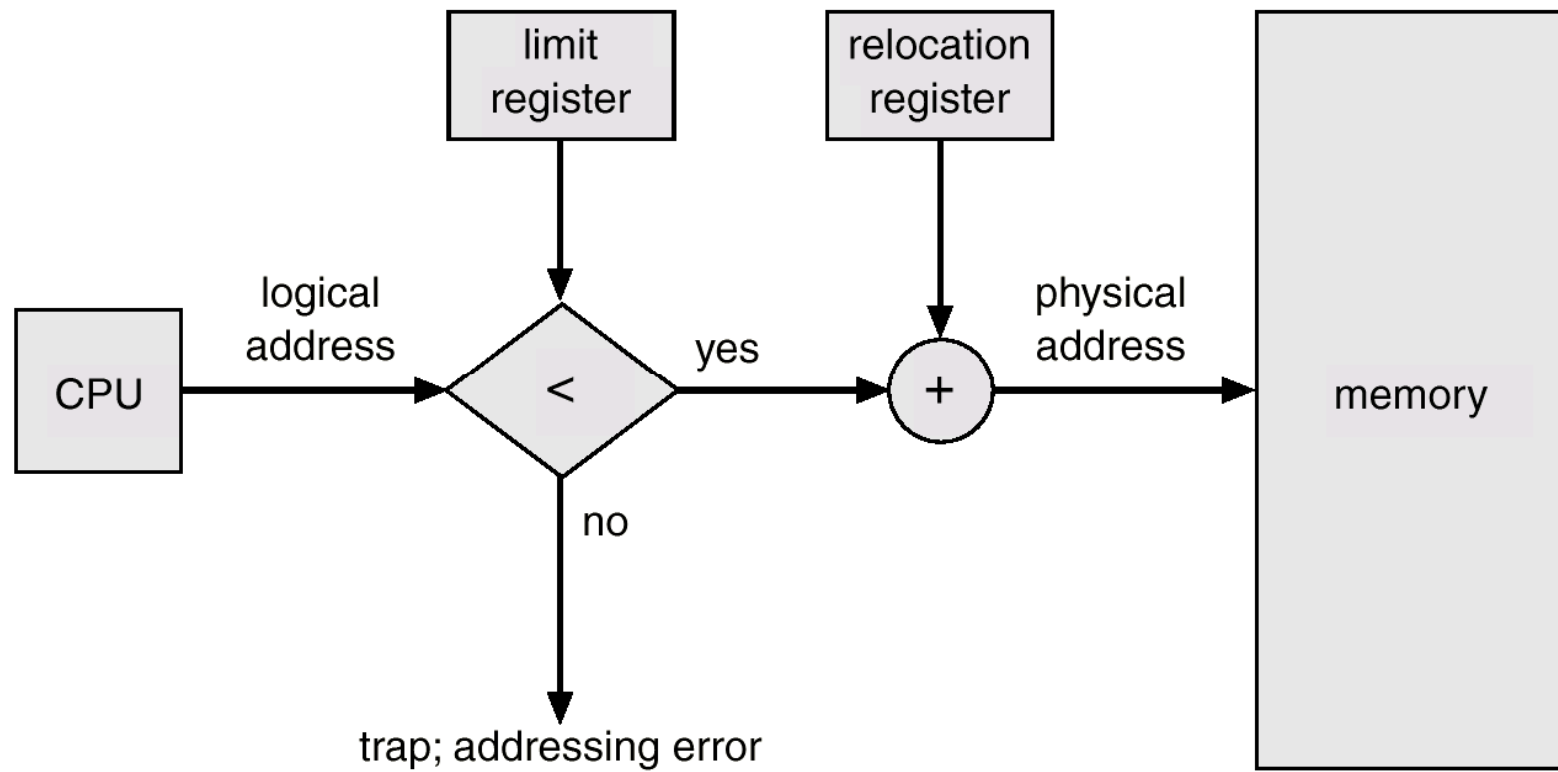


Alocação Contígua

- ◆ A memória principal tem normalmente 2 partições:
 - S.O. residente, em geral na parte baixa da memória, junto com o vetor de interrupções.
 - Processos do usuário mantidos na parte alta da memória.

- ◆ Partições simples
 - Registrador de relocação usado para proteger o S.O. dos processos de usuários e os processos dos usuários dos demais.
 - Registrador de relocação contém o valor do menor endereço físico. O registrador de limite contém a faixa de endereços lógicos.

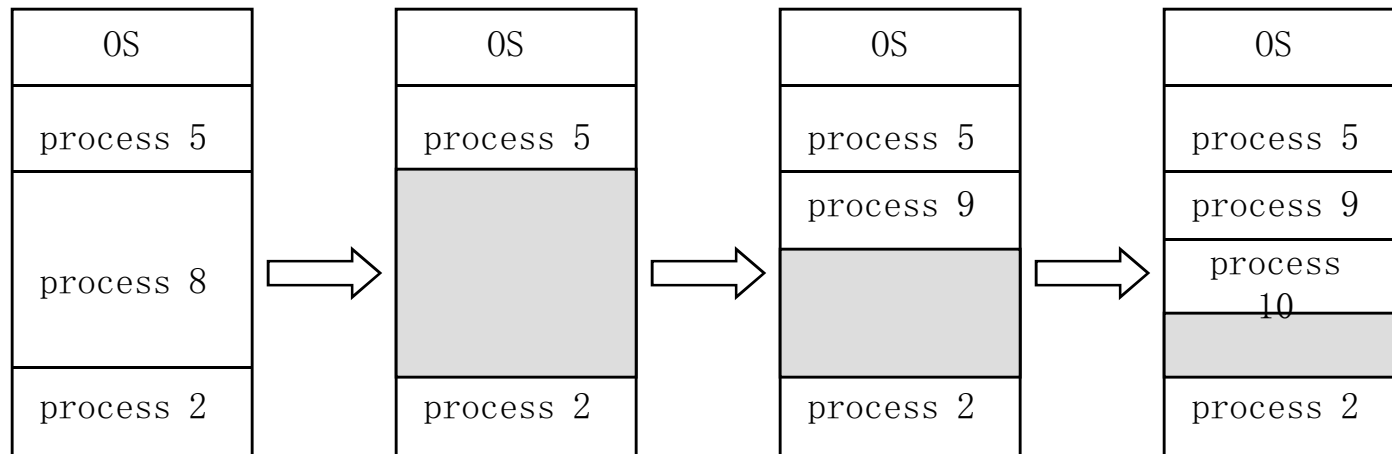
Hardware para relocação



Alocação Contígua

◆ Alocação de partições múltiplas

- *Buraco* – bloco de memória disponível; buracos de vários tamanhos se espalham pela memória.
- Quando um processo chega, é alocado em um buraco grande o suficiente para armazená-lo.
- O S.O. mantém informação sobre:
 - a) partições alocadas b) espaços livres (buracos)



Problema da Alocação de Memória Dinâmica

Como responder a um pedido de tamanho n de uma lista de buracos.

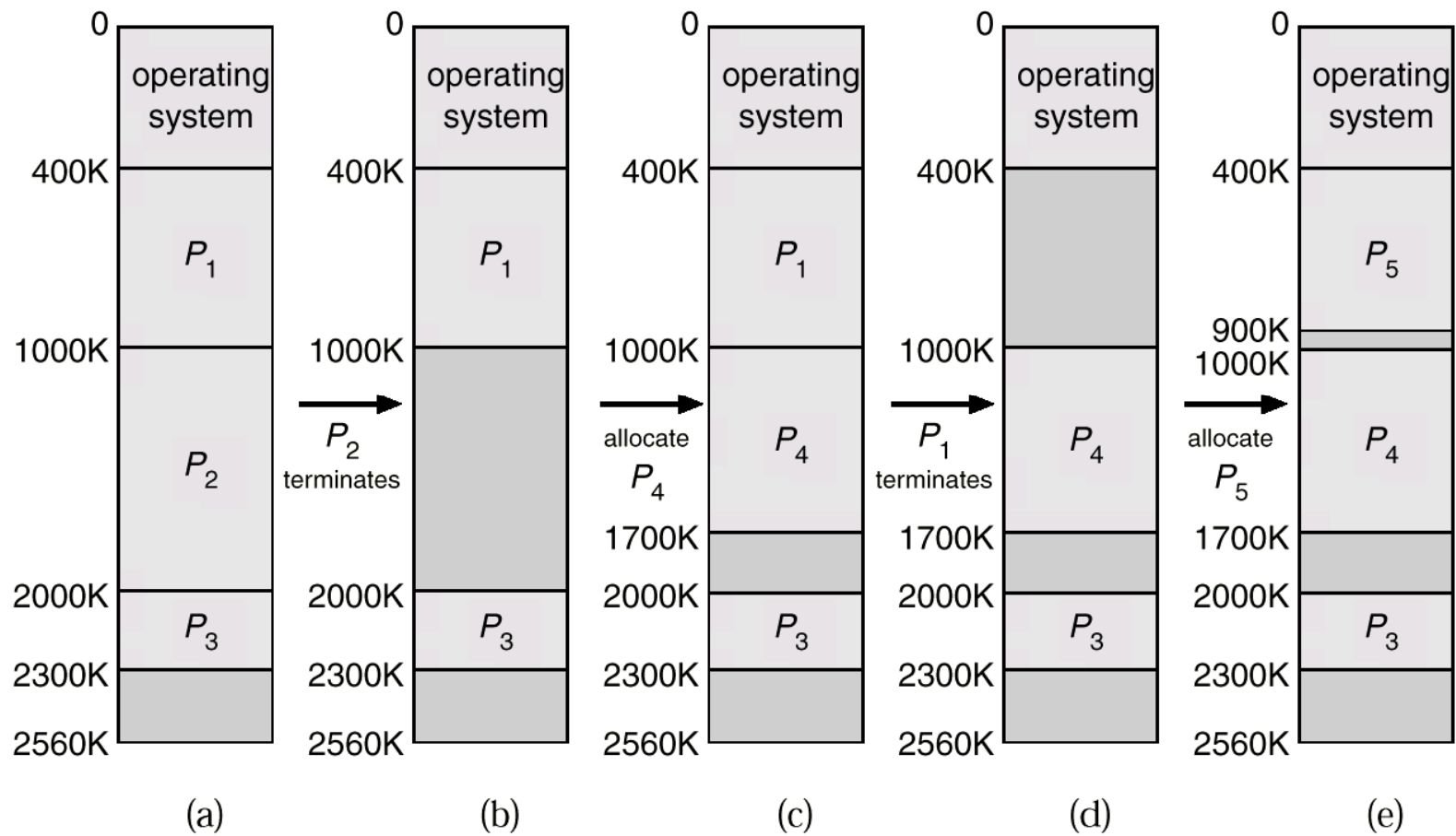
- ◆ **First-fit:** Aloca o primeiro buraco grande o suficiente.
- ◆ **Best-fit:** Aloca o menor buraco grande o suficiente; deve pesquisar toda a lista, a não ser que esteja ordenada.
- ◆ **Worst-fit:** Aloca o maior buraco.

First-fit e best-fit são melhores que o worst-fit em termos de velocidade e utilização de memória.

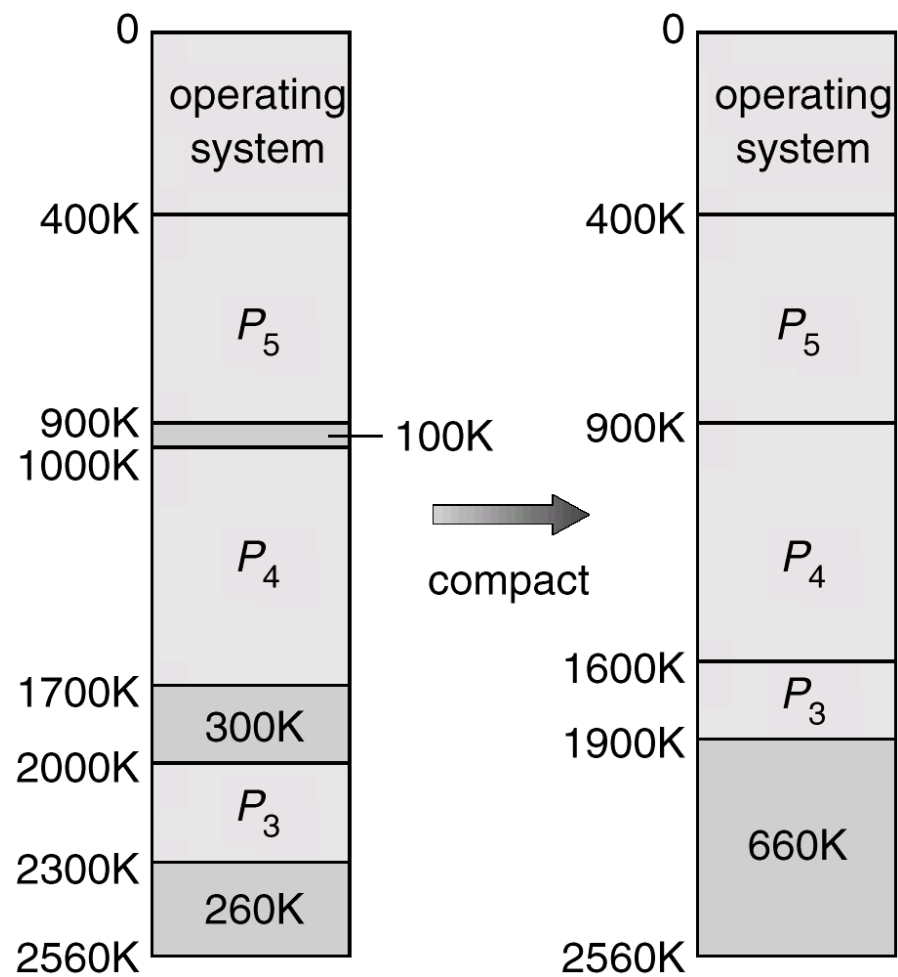
Fragmentação

- ◆ Fragmentação externa – Existe memória para satisfazer a requisição, mas ela não é contígua.
- ◆ Fragmentação interna – A memória alocada é maior que a necessária; esta diferença de memória fica ociosa dentro da partição.
- ◆ Reduzir fragmentação externa por compactação
 - Deslocar o conteúdo da memória para que toda a memória livre fique contígua.
 - Compactação é possível apenas quando a relocação é dinâmica e feita em tempo de execução.

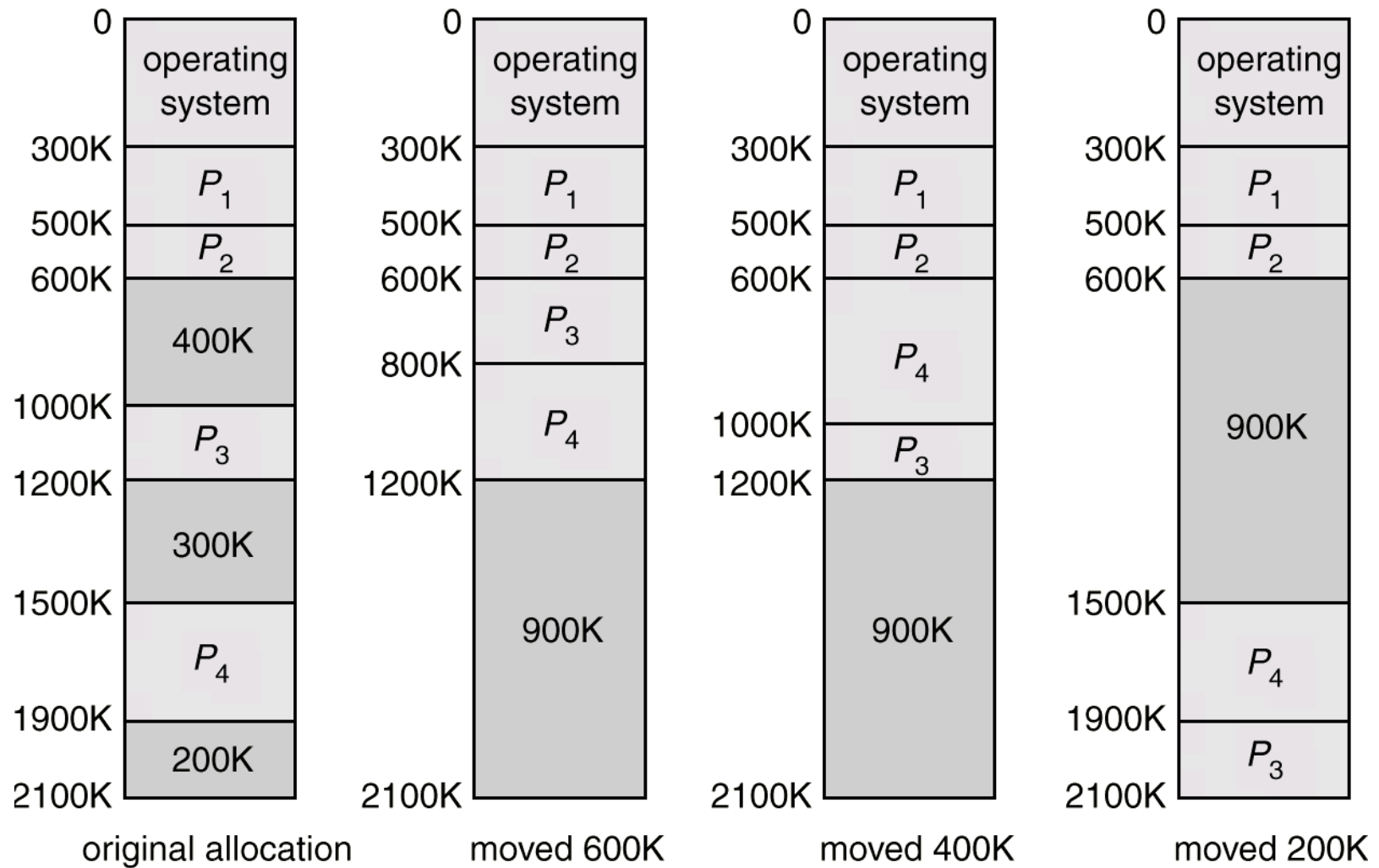
Fragmentação



Compactação



Compactação



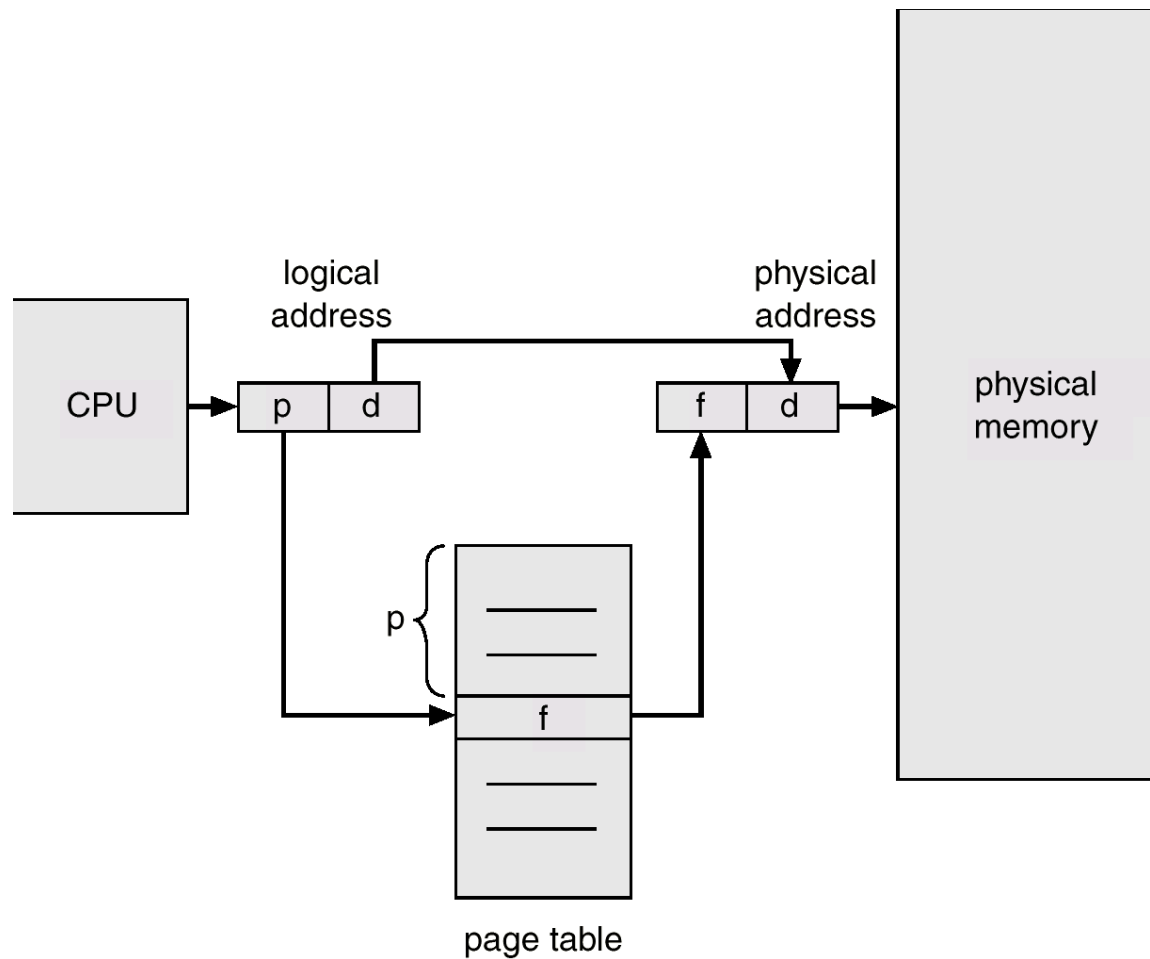
Paginação

- ◆ O espaço de endereçamento lógico de um processo pode ser não contíguo.
- ◆ Dividir a memória física em blocos de tamanho fixo chamados quadros (frames); tamanho é potência de 2, entre 512 bytes e 8192 bytes.
- ◆ Dividir memória lógica em blocos de mesmo tamanho chamados páginas.
- ◆ Manter registro de todos os quadros livres.
- ◆ Para executar um programa com tamanho de n páginas, é necessário encontrar n quadros livres e carregar o programa.
- ◆ Manter uma tabela de páginas para traduzir o endereço lógico em físico.
- ◆ Problema de fragmentação interna.

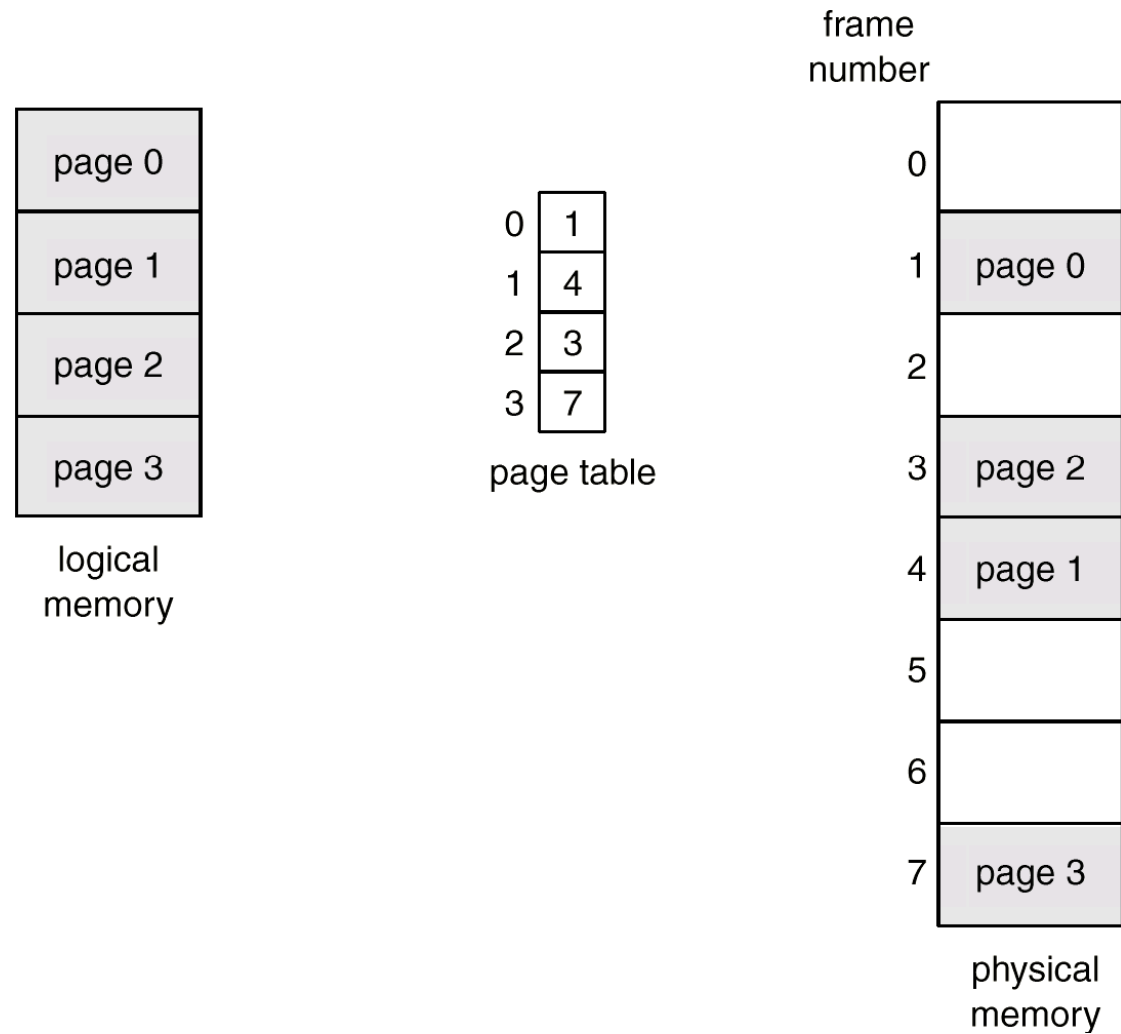
Esquema de Tradução de Endereços

- ◆ O endereço gerado pela CPU é dividido em:
 - *Número de página (p)* – usado como índice para a tabela de páginas que contém o endereço base de cada página na memória física.
 - *Deslocamento de página (d)* – combinado com o endereço base para definir o endereço de memória física que é enviado para a unidade de memória.

Arquitetura de Tradução de Endereço



Exemplo de paginação



Exemplo de paginação

0	a
1	b
2	c
3	d
4	e
5	f
6	g
7	h
8	i
9	j
10	k
11	l
12	m
13	n
14	o
15	p

logical
memory

0	5
1	6
2	1
3	2

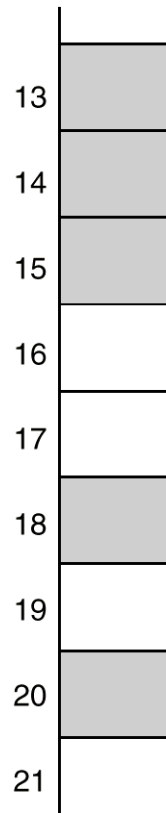
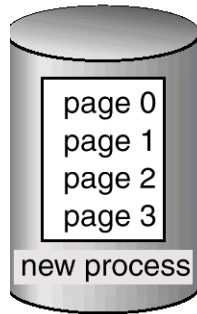
page table

0	
4	i j k l
8	m n o p
12	
16	
20	a b c d
24	e f g h
28	

physical memory

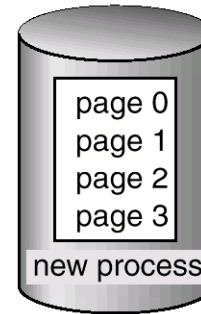
Exemplo de paginação – quadros livres

free-frame list
14
13
18
20
15



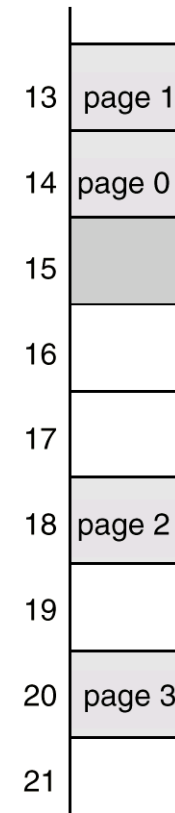
(a)

free-frame list
15



new-process page table

0	14
1	13
2	18
3	20

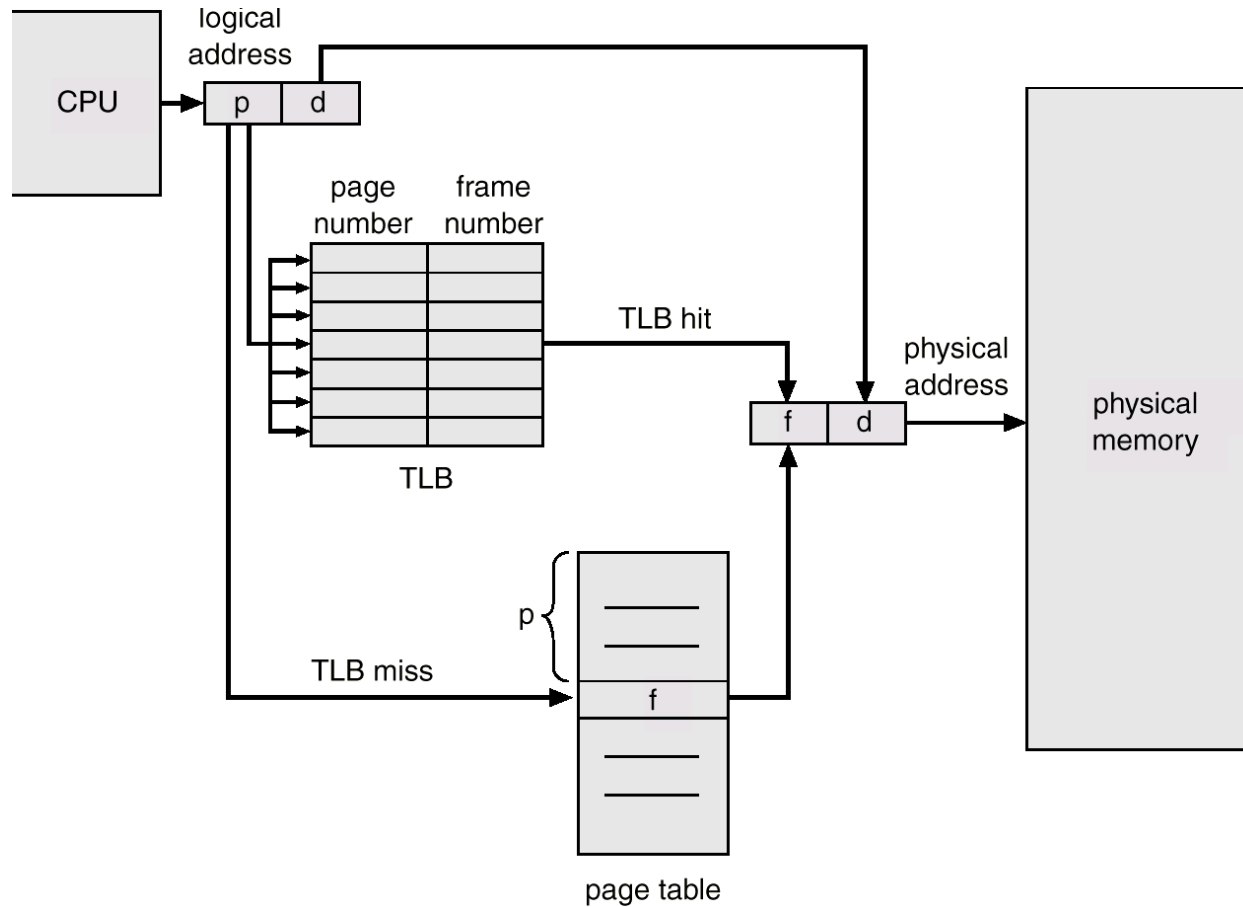


(b)

Implementação da Tabela de Páginas

- ◆ A tabela de páginas é mantida em memória primária.
- ◆ *O registrador base da tabela de páginas (PTBR)* aponta para a tabela de páginas.
- ◆ *O registrador de tamanho da tabela de páginas (PTLR)* indica o seu tamanho.
- ◆ Neste esquema, cada acesso a dados ou instruções requer 2 acessos à memória, um para a tabela de páginas e outro para o dado/instrução.
- ◆ O problema dos 2 acessos pode ser resolvido com o uso de uma cache especial de referência rápida chamado registradores associativos ou *translation look-aside buffers (TLBs)*

Hardware de paginação com TLB



Registradores Associativos

- ◆ Registradores associativos – busca paralela

# página	# quadro

Tradução de endereço (A' , A'')

- Se A' é um registrador associativo, busque o # do quadro.
- Senão busque o # do quadro da tabela de páginas na memória.

Tempo Efetivo de Acesso

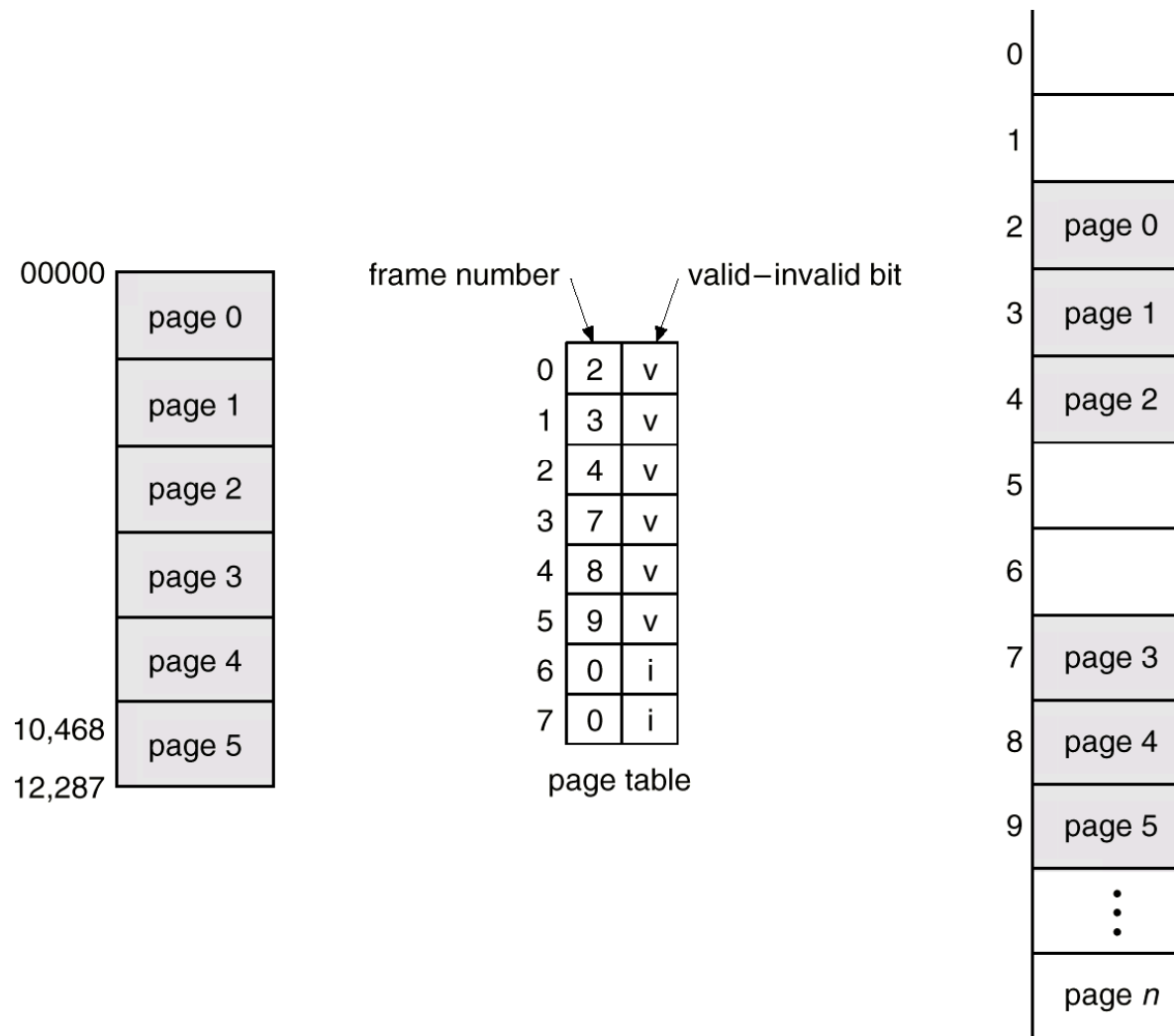
- ◆ Referência associativa = ϵ unidade de tempo
 - analisar registradores associativos
- ◆ Assumindo tempo de ciclo de memória de 1 microssegundo
- ◆ Taxa de acerto – percentagem de vezes que o número de página é encontrado nos registradores associativos; taxa relacionada com o número de registradores.
- ◆ Taxa de acerto = α
- ◆ Tempo efetivo de acesso (EAT)

$$\begin{aligned} \text{EAT} &= (1 + \epsilon) \alpha + (2 + \epsilon) (1 - \alpha) \\ &= 2 + \epsilon - \alpha \end{aligned}$$

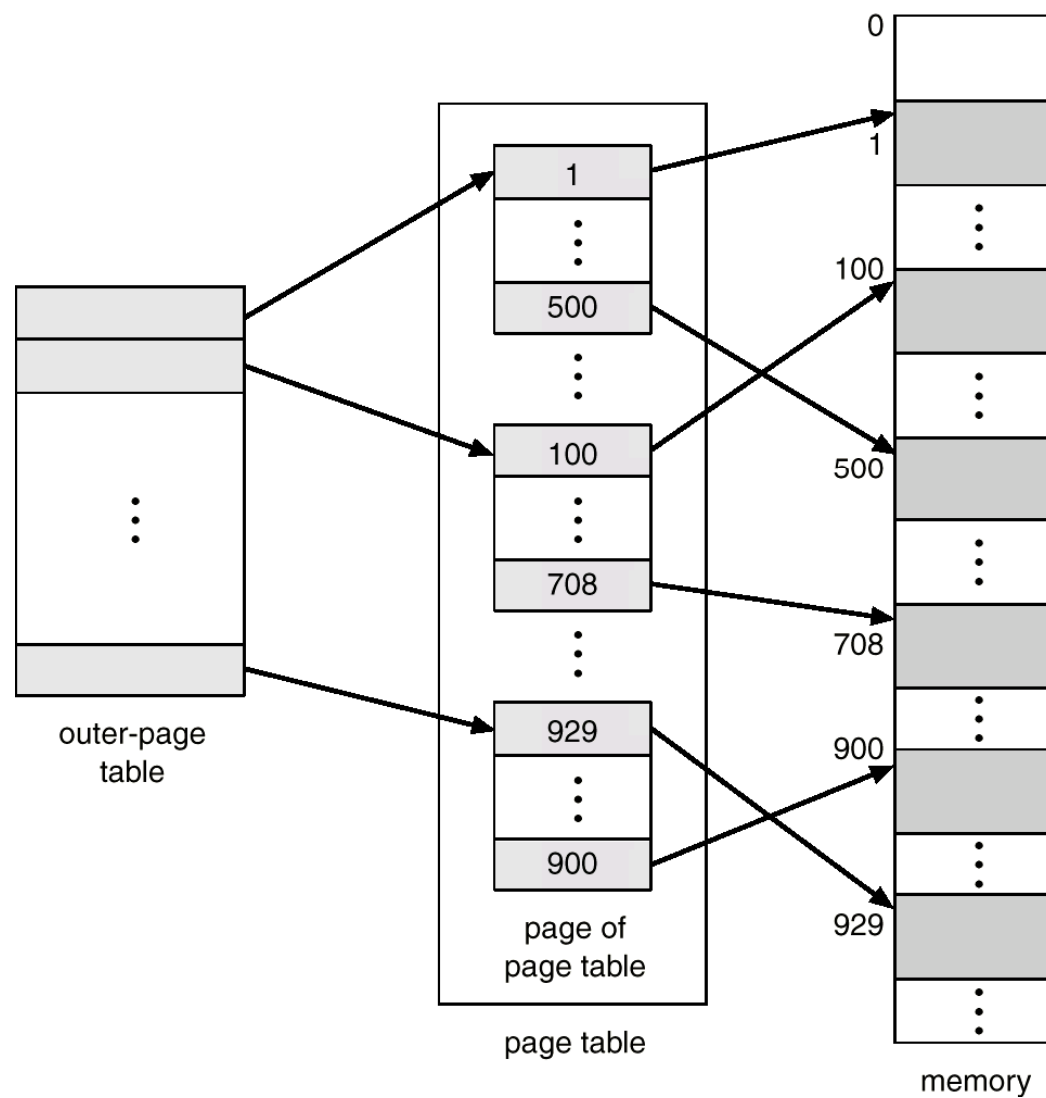
Proteção de Memória

- ◆ Proteção de memória implementada através da associação de um bit de proteção com cada quadro.
- ◆ *0 Bit Válido-inválido* é anexado a cada entrada na tabela de páginas:
 - “válido” indica que a página associada está no espaço de endereçamento lógico do processo e é uma página legal.
 - “inválido” indica que a página associada não está no espaço de endereçamento lógico do processo.

Proteção de Memória



Esquema de Paginação de 2 Níveis



Exemplo: Paginação de 2 Níveis (VAX)

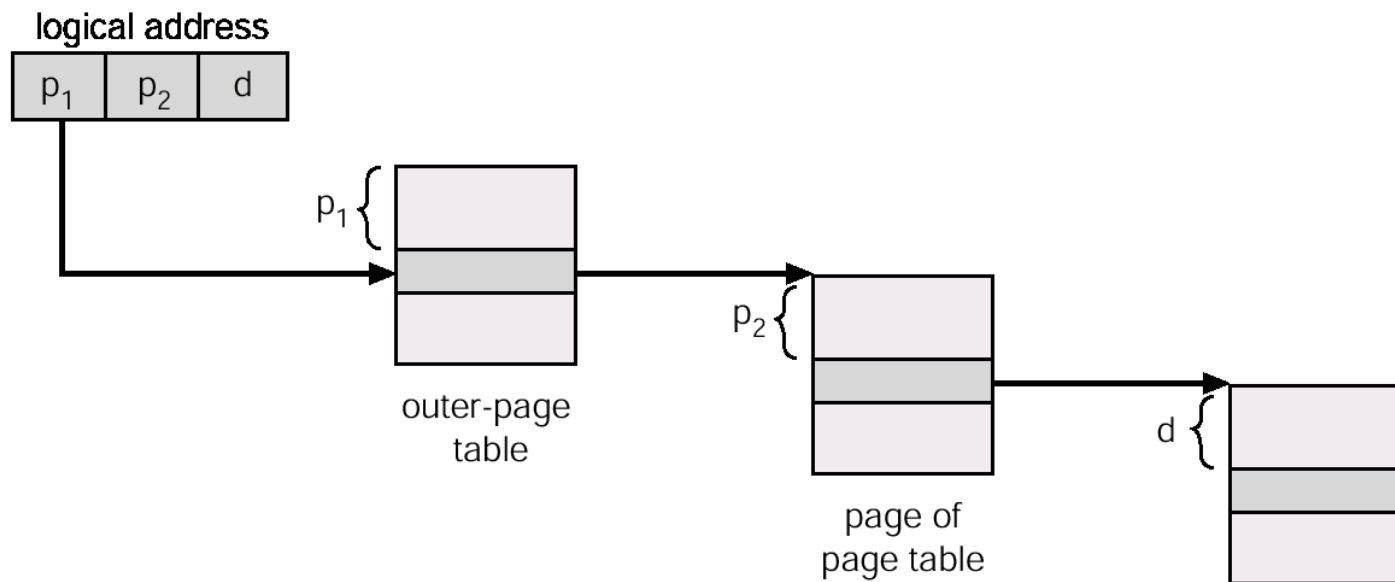
- ♦ Um endereço lógico (em máquina de 32 bits com tamanho de página de 4K) é dividido em:
 - um número de página consistindo de 20 bits.
 - Um deslocamento de página consistindo de 12 bits.
- ♦ Uma vez que a tabela de páginas é paginada, o número de página é dividido em:
 - um número de página de 10 bits.
 - Um deslocamento de página de 10 bits.
- ♦ Um endereço lógico é da forma:

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

onde p_i é um índice para a tabela de páginas externa, e p_2 é o deslocamento dentro da página da tabela de páginas externas.

Exemplo: Tradução de Endereço no VAX

- ◆ Esquema de tradução de endereço para uma arquitetura de paginação de dois níveis de 32-bits



Exemplo: Paginação de 4 níveis (Motorola 68030)

- ♦ Uma vez que cada nível é armazenado como uma tabela separada na memória, converter um endereço lógico em físico pode levar 4 acessos de memória.
- ♦ Mesmo que o tempo necessário para um acesso de memória for quintuplicado, a cache mantém a performance em nível razoável.
- ♦ Se o tempo de acesso à TBL é de 20ns e à memória é de 100ns, uma taxa de acerto de cache de 98% produz:

$$\begin{aligned}\text{tempo de acesso efetivo} &= 0.98 \times 120 + 0.02 \times 520 \\ &= 128 \text{ nanossegundos,}\end{aligned}$$

o qual é apenas 28% mais lento com relação ao tempo de acesso à memória.

Páginas compartilhadas

◆ Código compartilhado

- Uma cópia de código (leitura apenas) compartilhado entre processos (editores de texto, compiladores, etc.).

◆ Código e dados privados

- Cada processo mantém uma cópia separada de código e dados.
- As páginas para código e dados privados podem aparecer em qualquer posição no espaço de endereçamento lógico.

Páginas compartilhadas – exemplo

