

Aula 15

Sistemas Operacionais I

Gerenciamento de Memória - Parte 4

Prof. Julio Cezar Estrella

jcezar@icmc.usp.br

Material adaptado de

Sarita Mazzini Bruschi

baseados no livro Sistemas Operacionais Modernos de A. Tanenbaum

Memória Virtual

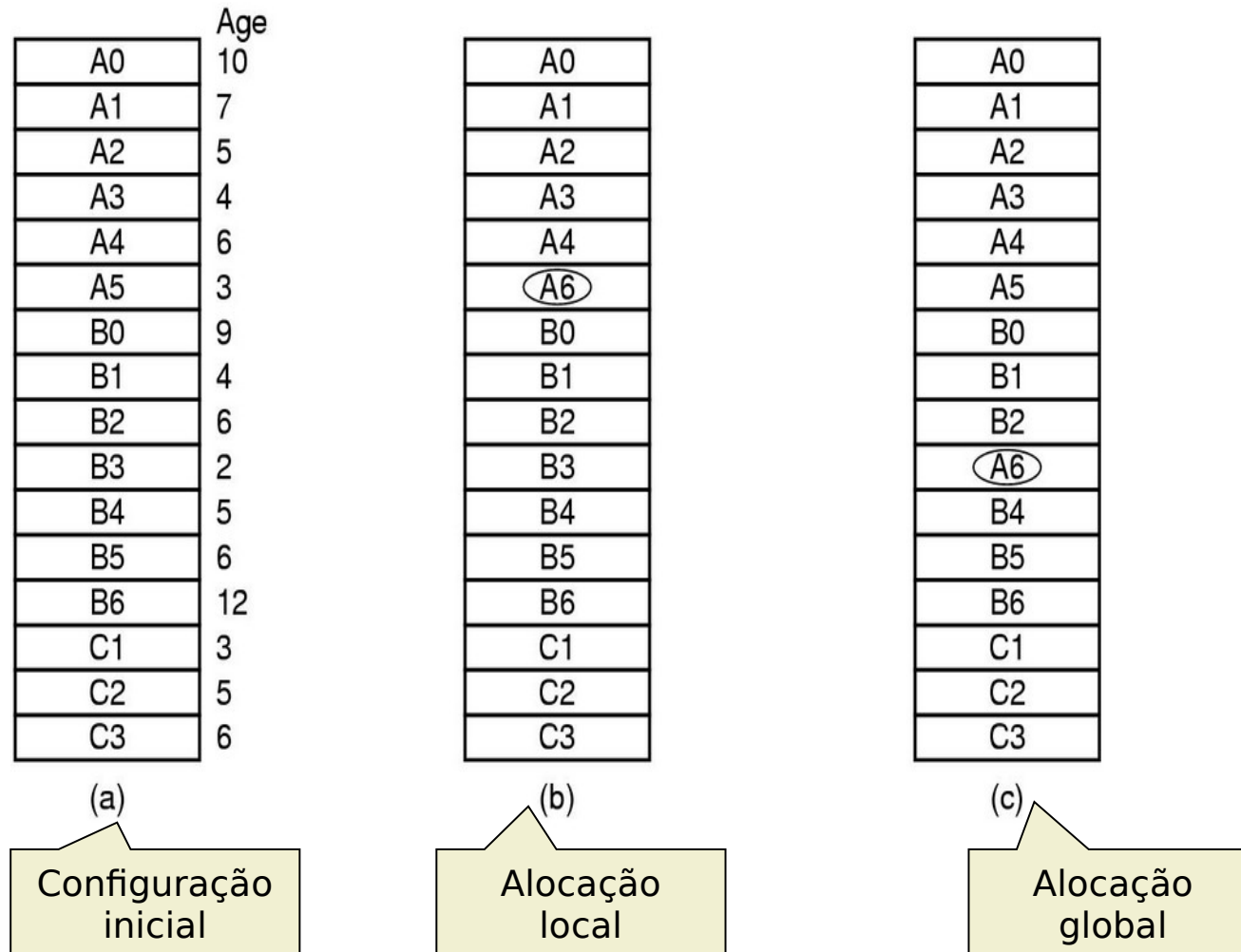
Paginação – Troca de Página

- Política de Substituição Local: páginas dos próprios processos são utilizadas na troca;
 - Dificuldade: definir quantas páginas cada processo pode utilizar;
- Política de Substituição Global: páginas de todos os processos são utilizadas na troca;
 - Problema: processos com menor prioridade podem ter um número muito reduzido de páginas, e com isso, acontecem muitas faltas de páginas;

Memória Virtual

Paginação – Troca de Página

Falta de Página no Processo A



Memória Virtual

Paginação – Troca de Página

- Política de alocação local (número fixo de páginas/processo) permite somente política de substituição local de páginas
- Política de alocação global (número variável de páginas/processo) permite tanto a política de substituição de páginas local quanto global

Memória Virtual

Paginação – Troca de Página

- Algoritmos de substituição local:
 - *Working Set*;
 - *WSClock*;
- Algoritmos de substituição local/global:
 - Ótimo;
 - NRU;
 - FIFO;
 - Segunda Chance;
 - LRU;
 - Relógio;

Memória Virtual

Paginação – Implementação

- Até agora, vimos somente como uma página é selecionada para remoção. Mas onde a página descartada da memória é colocada?
- **Memória Secundária - Disco**
 - A área de troca (*swap area*) é gerenciada como uma lista de espaços disponíveis;
 - O endereço da área de troca de cada processo é mantido na tabela de processos;
 - Cálculo do endereço: MMU;

Memória Virtual

Paginação – Implementação

- Memória Secundária – Disco
 - Possibilidade A - Assim que o processo é criado, ele é copiado todo para sua área de troca no disco, sendo carregado para memória quando necessário;
 - Área de troca diferente para dados, pilha e programa, pois área de dados pode crescer e a área de pilha crescerá certamente;

Memória Virtual

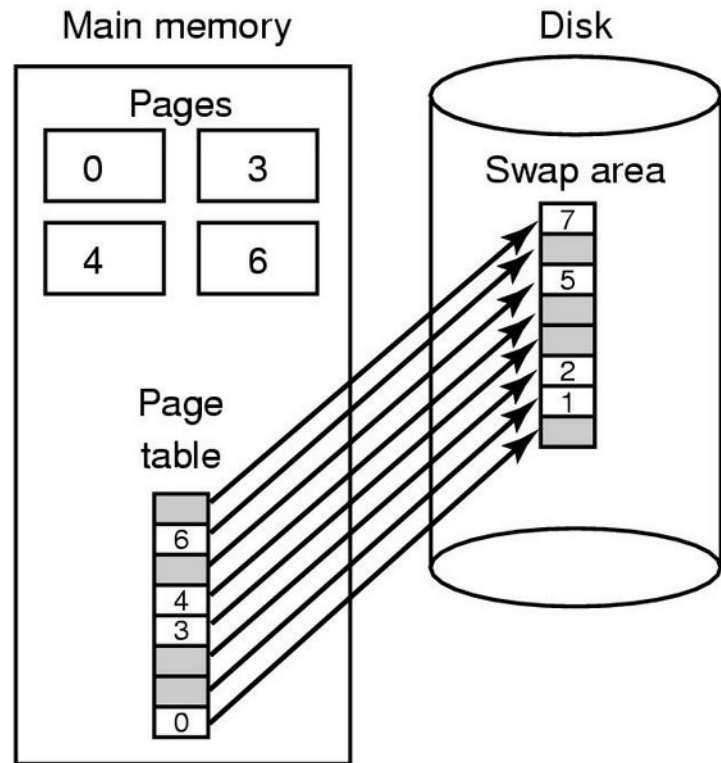
Paginação – Implementação

- Memória Secundária – Disco (cont.)
 - Possibilidade B - Nada é alocado antecipadamente, espaço é alocado em disco quando a página for enviada para lá. Assim, o processo na memória RAM não fica “amarrado” a uma área específica;

Memória Virtual

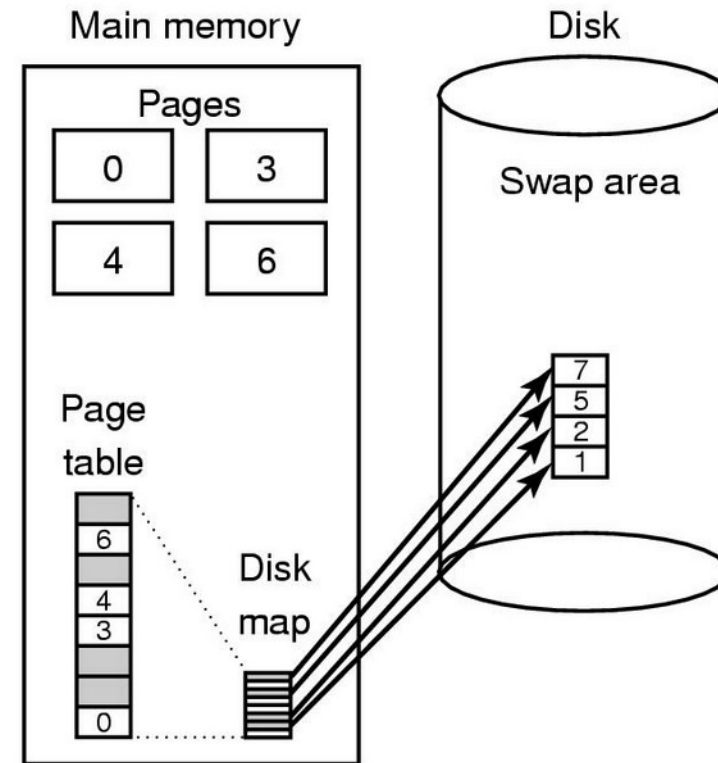
Paginação - Implementação

- **Como fica o disco - memória secundária**



(a)

Área de troca estática



(b)

Área de

Memória Virtual

Paginação – Tabela de Páginas Invertida

- Geralmente, cada processo tem uma tabela de páginas associada a ele → classificação feita pelo endereço virtual;
 - Pode consumir grande quantidade de memória;
- Alternativa: tabela de páginas invertida
 - SO mantém uma única tabela para as molduras de páginas da memória;
 - Cada entrada consiste no endereço virtual da página armazenada naquela página real, com informações sobre o processo dono da página virtual;
 - Exemplos de sistemas: IBM System/38, IBM RISC System 6000, IBM RT e estações HP Spectrum;

Memória Virtual

Paginação – Tabela de Páginas Invertida

- Quando uma referência de memória é realizada (página virtual), a tabela de páginas invertida é pesquisada para encontrar a moldura de página correspondente;
 - Se encontra, o endereço físico é gerado →
<i, deslocamento>;

Memória Virtual

Paginação – Tabela de Páginas Invertida

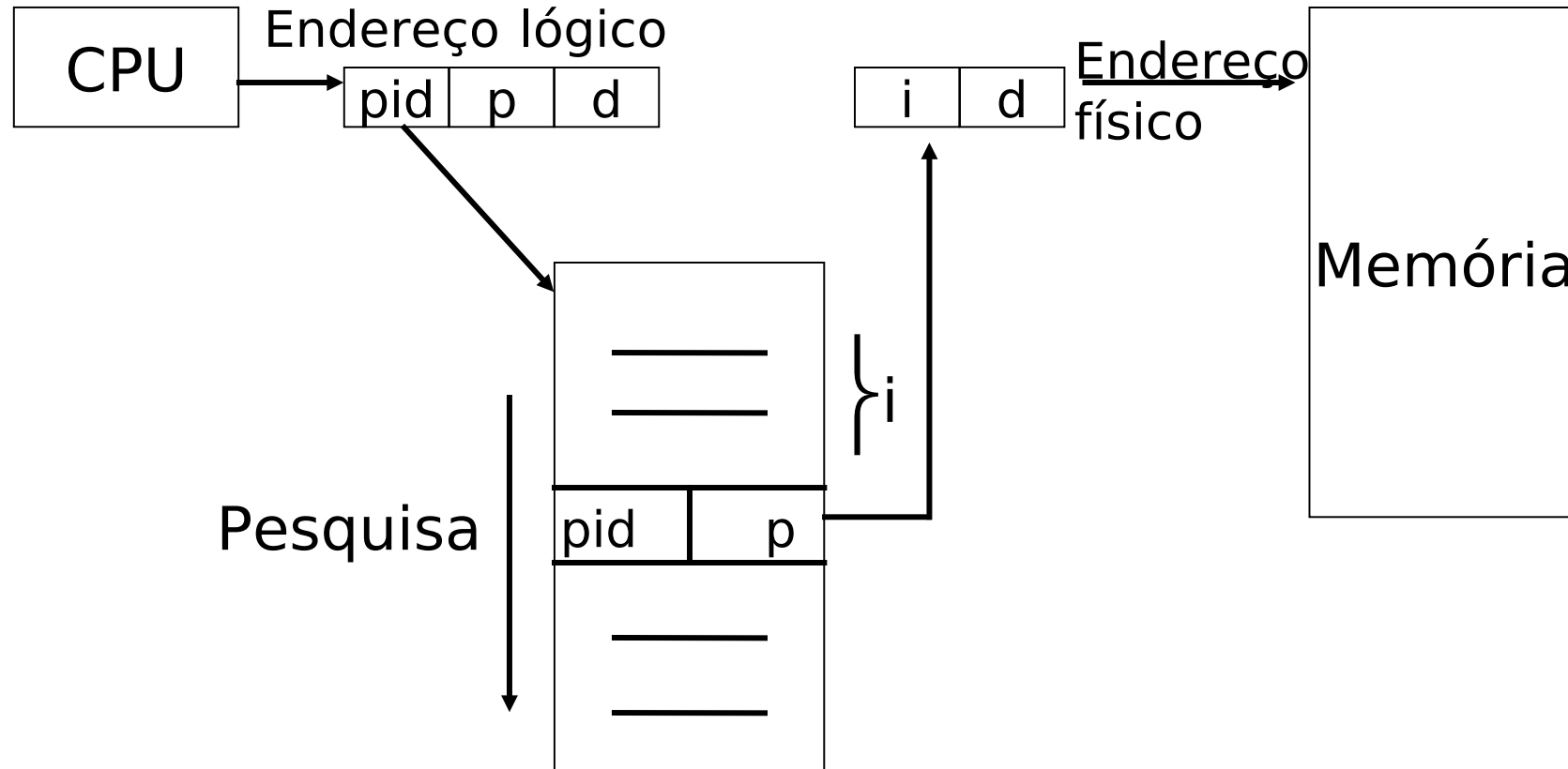


Tabela de páginas invertida

Endereço lógico: <id processo (pid), número página (p), deslocamento (d)>

Memória Virtual

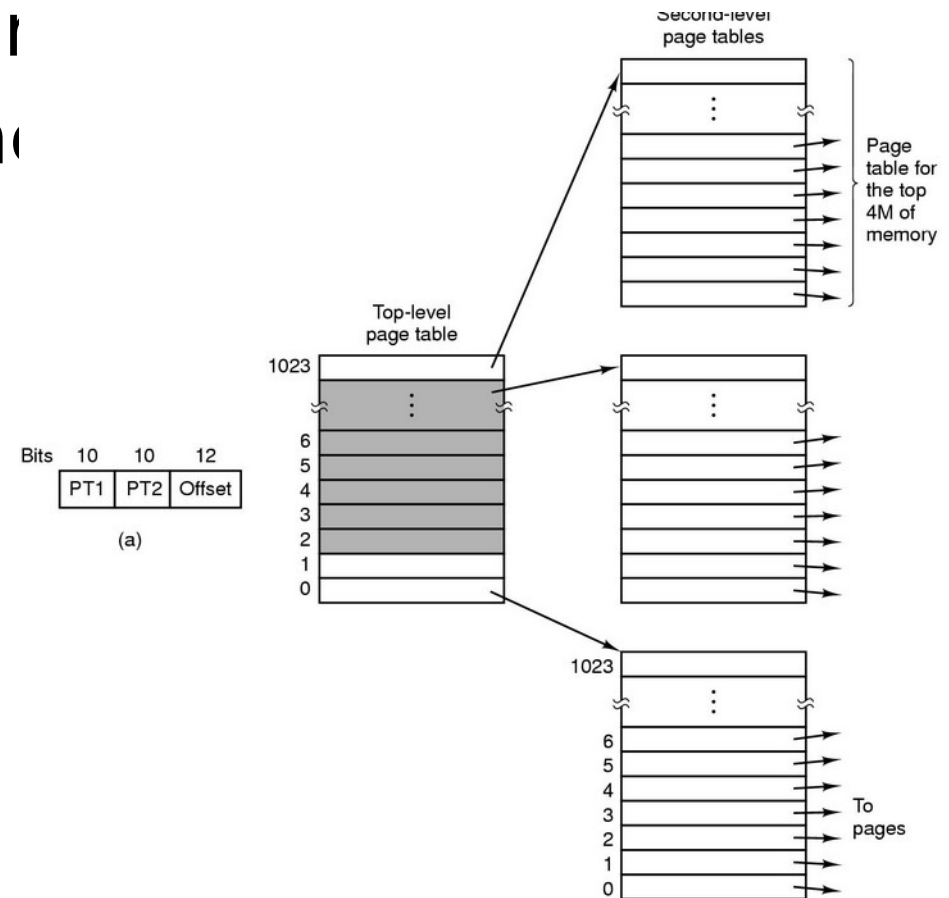
Paginação – Tabela de Páginas Invertida

- Vantagens:
 - Ocupa menos espaço;
 - É mais fácil de gerenciar apenas uma tabela;
- Desvantagens:
 - Aumenta tempo de pesquisa na tabela, pois, apesar de ser **classificada por endereços físicos**, é **pesquisada por endereços lógicos**;
 - Aliviar o problema: tabela *hashing*;
 - Uso da TLB (memória associativa) para manter entradas recentemente utilizadas;

Memória Virtual

Tabela de Páginas Multinível

- Objetivo: não precisar manter toda a tabela de páginas na memória



Gerenciamento de Memória

Segmentação

- Segmentação: Visão do programador/compilador
 - Tabelas de segmentos com **n** linhas, cada qual apontando para um segmento de memória;
 - Vários espaços de endereçamento;
 - Endereço real \rightarrow base + deslocamento;
 - Alocação de segmentos segue os algoritmos já estudados:
 - *FIRST-FIT*;
 - *BEST-FIT*;
 - *NEXT-FIT*;
 - *WORST-FIT*;
 - *QUICK-FIT*;

Gerenciamento de Memória

Segmentação

- Segmentação:
 - Facilita proteção dos dados;
 - Facilita compartilhamento de procedimentos e dados entre processos;
 - MMU também é utilizada para mapeamento entre os endereços lógicos e físicos;
 - Tabela de segmentos informa qual o endereço da memória física do segmento e seu tamanho;

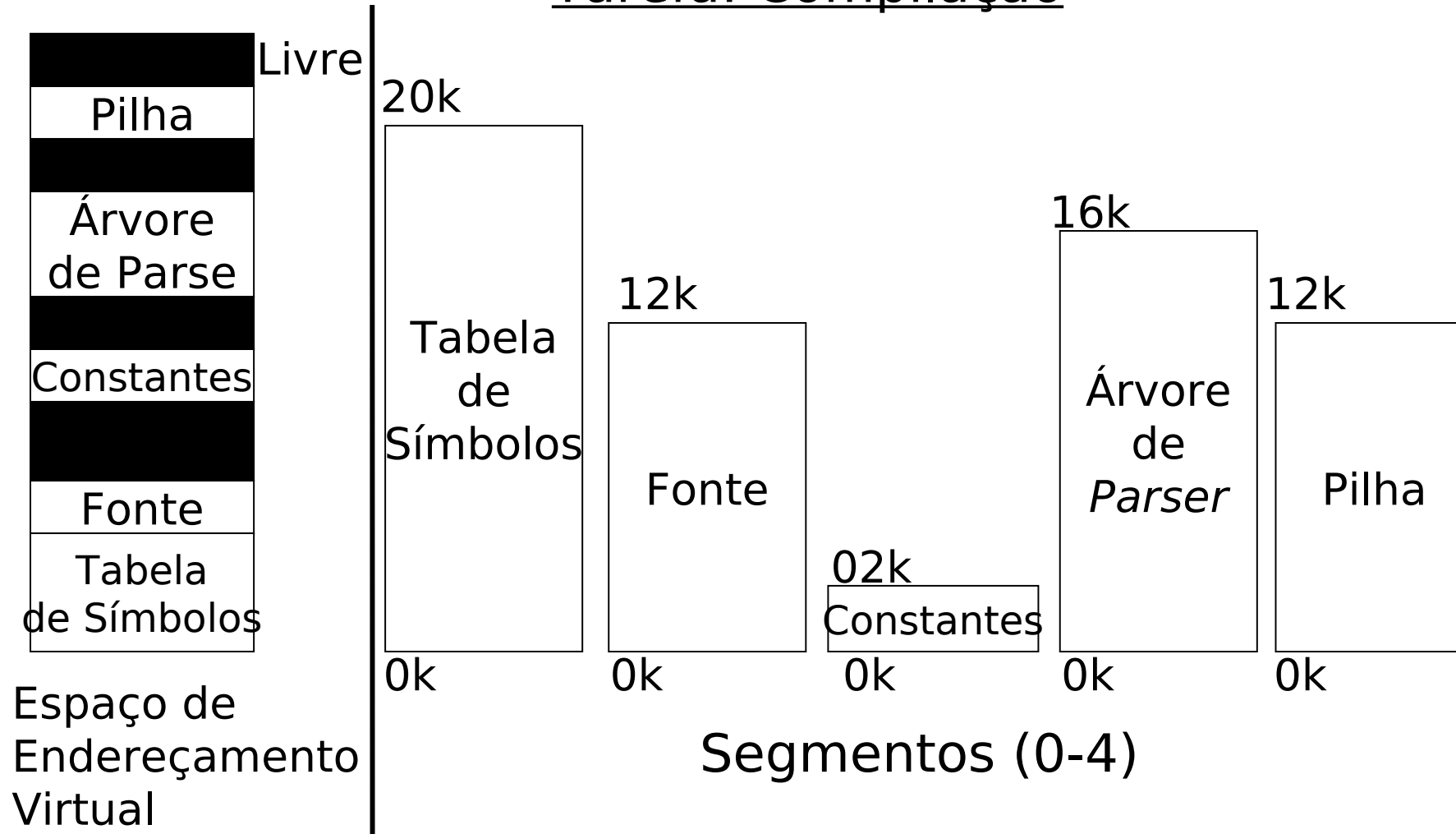
Gerenciamento de Memória

Segmentação

- Segmentação:
 - Problemas encontrados → embora haja espaço na memória, não há espaço contínuo:
 - Política de relocação: um ou mais segmentos são relocados para abrir espaço contínuo;
 - Política de compactação: todos os espaços são compactados;
 - Política de bloqueio: fila de espera;
 - Política de troca: substituição de segmentos;
 - Sem fragmentação interna, com fragmentação externa;

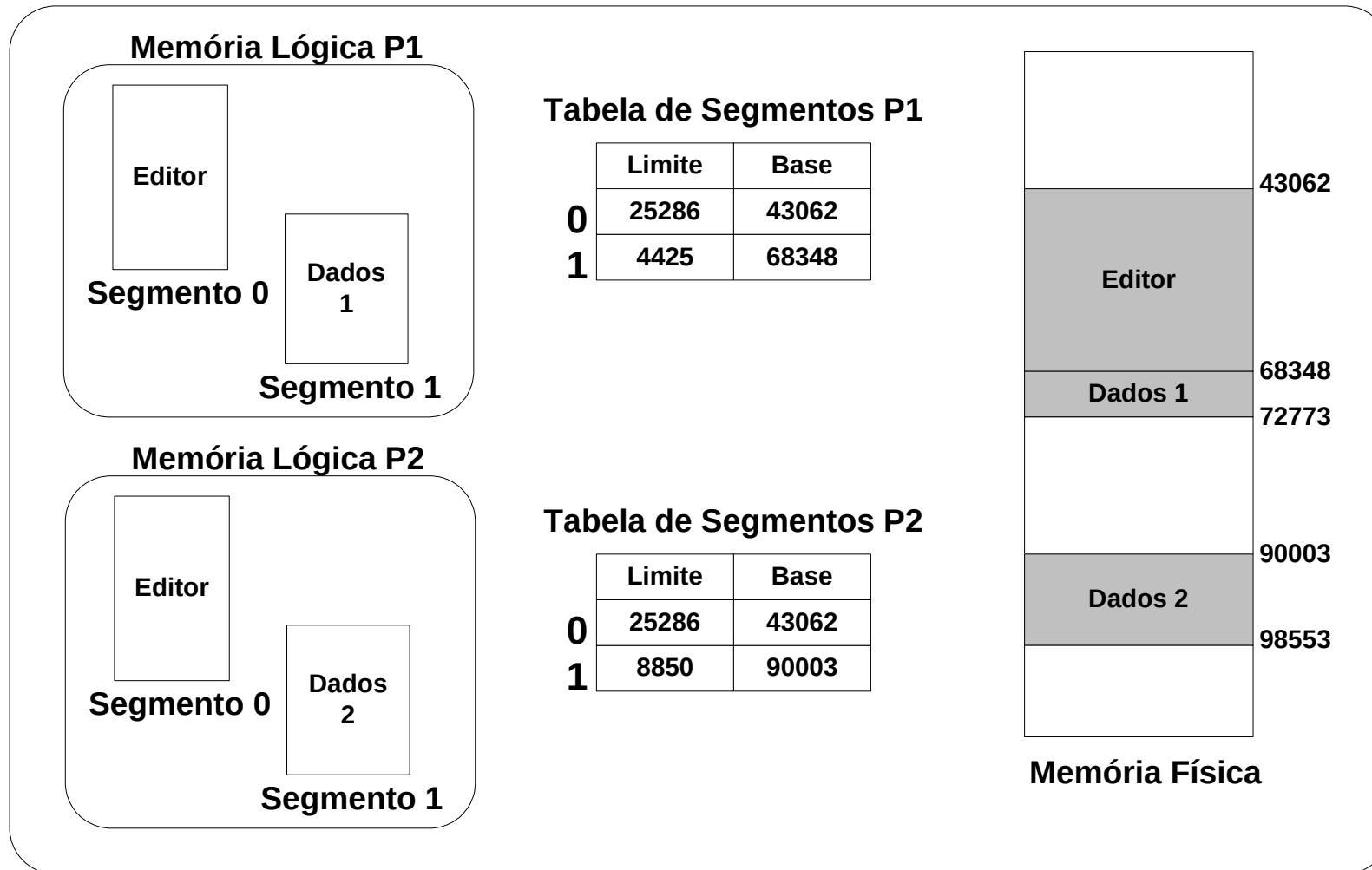
Gerenciamento de Memória Segmentação

Tarefa: Compilação



Gerenciamento de Memória

Segmentação



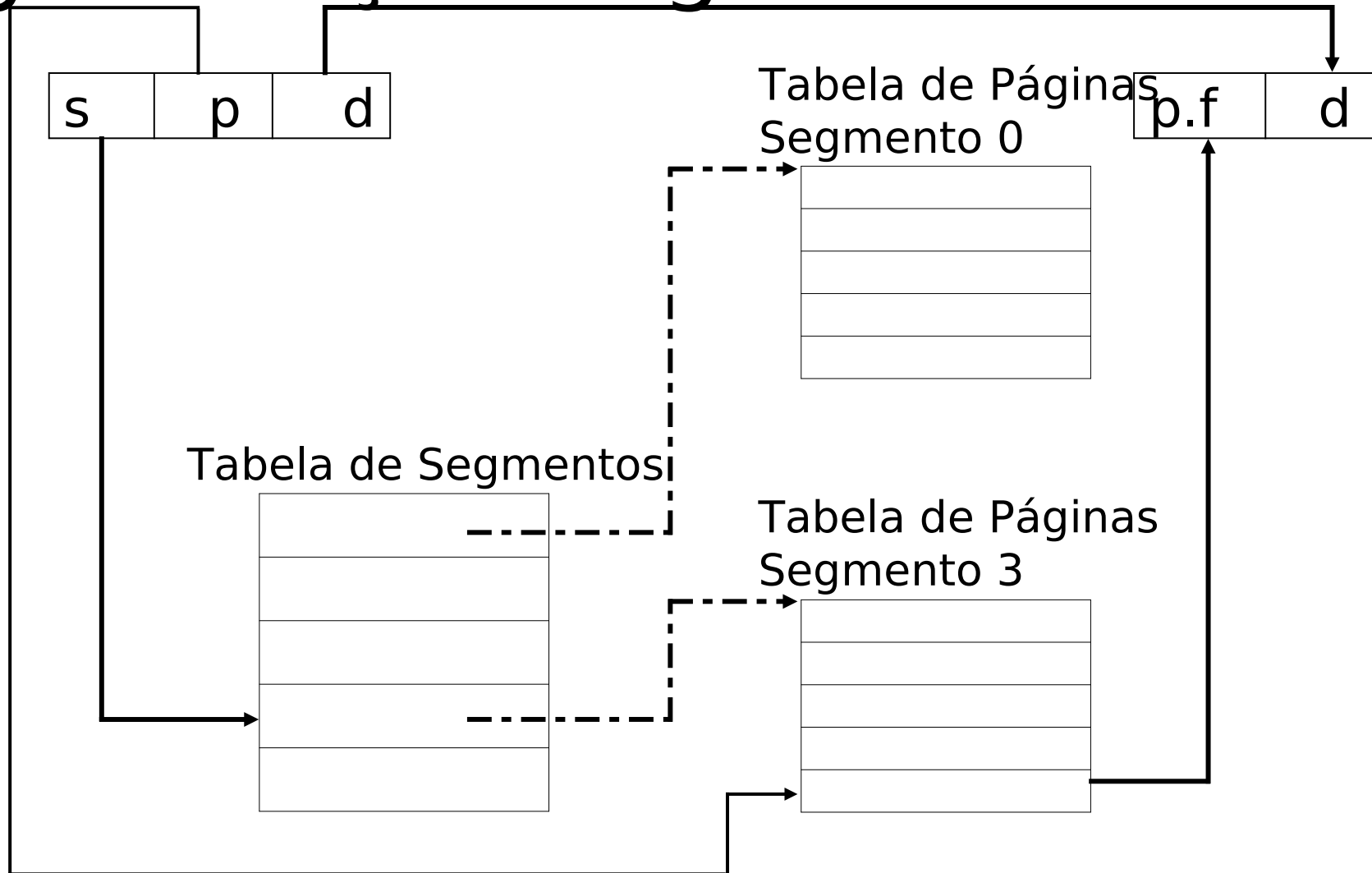
Gerenciamento de Memória

Segmentação-Paginada

- Espaço lógico é formado por segmentos
 - Cada segmento é dividido em páginas lógicas;
 - Cada segmento possui uma tabela de páginas → mapear o endereço de página lógica do segmento em endereço de página física;
 - No endereçamento, a tabela de segmentos indica, para cada segmento, onde sua respectiva tabela de páginas está;
 - Multics, Pentium

Gerenciamento de Memória

Segmentação-Paginada



Gerenciamento de Memória

Consideração

Paginação

Segmentação

| | | |
|--|-----|--------|
| Programador deve saber da técnica? | Não | Sim |
| Espaços de endereçamento existentes | 1 | Vários |
| Espaço total de endereço pode exceder memória física? | Sim | Sim |
| É possível distinguir procedimento de dados e protegê-los? | Não | Sim |

Gerenciamento de Memória

| Consideração | Paginação | Segmentação |
|---|--|---|
| Tabelas de tamanho variável podem ser acomodadas sem problemas? | Não | Sim |
| Compartilhamento de procedimentos entre usuário é facilitado? | Não | Sim |
| Por que? | Para obter espaço de endereçamento maior sem aumentar memória física | Para permitir que programas e dados possam ser divididos em espaços de endereçamento logicamente independentes; compartilhamento e proteção |