



Universidade Federal da Bahia



Sistemas Operacionais

MATA58

Prof. Maycon Leone M. Peixoto

mayconleone@dcc.ufba.br

Programa

- Introdução aos Sistemas Operacionais
- Processos
- **Gerência de Memória**
 - Memória virtual
 - Paginação
 - Segmentação
- Sistemas de Arquivos
- Entrada/saída
- Segurança
- Exemplos de Sistemas Operacionais

Gerenciamento de Memória

Troca de Páginas - Paginação

□ Algoritmos vistos:

- Ótimo;
- NRU;
- FIFO;
- Segunda Chance;
- Relógio;

□ Hoje:

- LRU;
- *Working set*;
- *WSClock*;

Gerenciamento de Memória Troca de Páginas - Paginação

- Algoritmo *Least Recently Used Page Replacement* (LRU) ou *Menos Recentemente Usada* (MRU)
 - Troca a página menos referenciada/modificada recentemente;
 - Alto custo
 - Lista encadeada com as páginas que estão na memória, com as mais recentemente utilizadas no início e as menos utilizadas no final;
 - A lista deve ser atualizada a cada referência da memória;

Gerenciamento de Memória Troca de Páginas - Paginação

- Algoritmo *Least Recently Used Page Replacement* (LRU)
 - Pode ser implementado tanto por hardware quanto por software:
 - Hardware: MMU deve suportar a implementação LRU;
 - 1a. opção
 - Contador em hardware (64 *bits*) – conta instruções executadas;
 - Após cada referência à memória, o valor do contador é armazenado na entrada da tabela de páginas referente à página acessada;
 - Quando ocorre falta de página, o SO examina todos os contadores e escolhe a página que tem o menor valor

Gerenciamento de Memória Troca de Páginas - Paginação

- LRU – Hardware - 2a. Opção
 - Matriz nxn bits
 - Referencia a página k -> linha k=1 e coluna k=0

0,1,2,3,2,1,0,3,2,3

	0	1	2	3
0	0	1	1	1
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0

(a)

	0	1	2	3
0	0	0	1	1
1	1	0	1	1
2	0	0	0	0
3	0	0	0	0

(b)

	0	1	2	3
0	0	0	0	1
1	1	0	0	1
2	1	1	0	1
3	0	0	0	0

(c)

	0	1	2	3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0

(d)

	0	1	2	3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	1
3	1	1	0	0

(e)

	0	1	2	3
0	0	0	0	0
1	1	0	1	1
2	1	0	0	1
3	1	0	0	0

(f)

	0	1	2	3
0	0	1	1	1
1	0	0	1	1
2	0	0	0	1
3	0	0	0	0

(g)

	0	1	2	3
0	0	1	1	0
1	0	0	1	0
2	0	0	0	0
3	1	1	1	0

(h)

	0	1	2	3
0	0	1	0	0
1	0	0	0	0
2	1	1	0	1
3	1	1	0	0

(i)

	0	1	2	3
0	0	1	0	0
1	0	0	0	0
2	1	1	0	0
3	1	1	1	0

(j)

Gerenciamento de Memória Troca de Páginas - Paginação

- Algoritmo *Least Recently Used Page Replacement* (LRU)
 - Pode ser implementado tanto por hardware quanto por software:
 - Software: duas maneiras
 - NFU (*Not frequently used*) ou LFU (*least frequently used*);
 - *Aging* (Envelhecimento);

Gerenciamento de Memória

Troca de Páginas - Paginação

- Software: NFU ou LFU (*least*)
 - Para cada página existe um contador □ iniciado com zero e incrementado a cada referência à página;
 - Página com menor valor do contador é candidata a troca;
 - Esse algoritmo não se esquece de nada
 - Problema: pode retirar páginas que estão sendo referenciadas com frequência;
 - Compilador com vários passos: passo 1 tem mais tempo de execução que os outros passos □ páginas do passo 1 terão mais referências armazenadas;

Gerenciamento de Memória

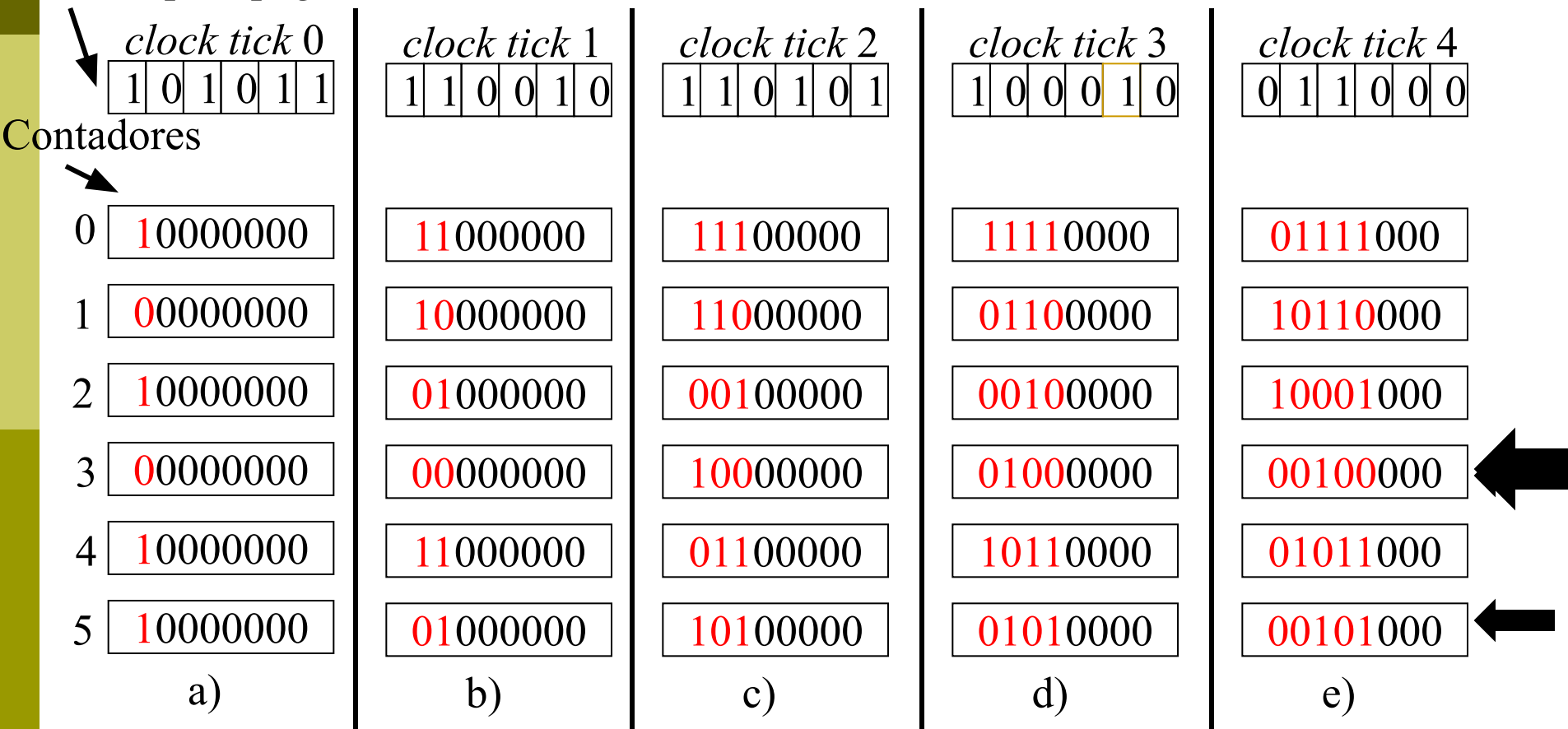
Troca de Páginas - Paginação

- Software: Algoritmo *aging* (envelhecimento)
 - Modificação do NFU, resolvendo o problema descrito anteriormente;
 - Além de saber **quantas vezes** a página foi referenciada, também controla **quando** ela foi referenciada;

Gerenciamento de Memória Troca de Páginas - Paginação

□ Algoritmo *aging*

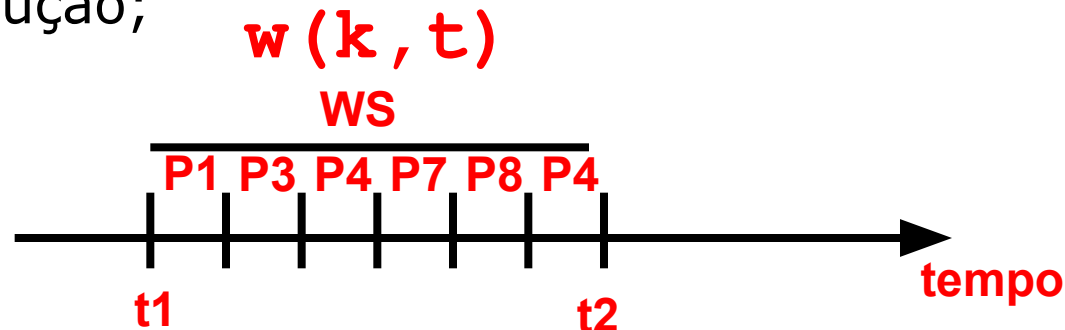
Bits R para páginas 0-5



Gerenciamento de Memória Troca de Páginas - Paginação

□ Algoritmo *Working Set* (WS):

- Paginação por demanda □ páginas são carregadas na memória somente quando são necessárias;
- Pré-paginação □ *Working set*
 - Carregar um conjunto de páginas que um processo está efetivamente utilizando (referenciando) em um determinado tempo t antes de ele ser posto em execução;



Gerenciamento de Memória Troca de Páginas - Paginação

- Algoritmo *Working Set* (WS):
 - Objetivo principal: reduzir a falta de páginas
 - Um processo só é executado quando todas as páginas necessárias no tempo t estão carregadas na memória;
 - SO gerencia quais páginas estão no *Working Set*;
 - Para simplificar □ o *working set* pode ser visto como o conjunto de páginas que o processo referenciou durante os últimos t segundos de tempo;
 - Utiliza *bit* R e o tempo de relógio (tempo virtual) da última vez que a página foi referenciada;

Gerenciamento de Memória Troca de Páginas - Paginação

■ Algoritmo *Working Set*:

* Se todas as páginas estiverem com $R=1$, uma página é escolhida aleatoriamente;

** Se todas as páginas estiverem no WS, a página mais velha com $R=0$ é escolhida;

Tempo do último Uso (TLU)	Bit R	
→ 2084	1	↓
	2003	1
	1980	1
	1213	0
	2014	1
	2020	1
	2032	1
	1620	0

Tabela de Páginas

Tempo virtual atual (CVT): 2204
 $age = CVT - TLU$
(Ex.: $2204 - 2084 = 120$)
 $\tau = \text{múltiplos } clock \text{ ticks}$

Percorrer as páginas examinando bit R;

Se $(R==1)^*$
página foi referenciada;
faz TLU da página igual ao CVT;

Se $(R==0 \text{ e } age > \tau)$
página não está no *working set*;
remove a página;

Se $(R==0 \text{ e } age \leq \tau)^{**}$
página está no *working set*;
remove página com maior *age* dentro do intervalo τ ;

Gerenciamento de Memória Troca de Páginas - Paginação

- Algoritmo *WSClock*:
 - *Clock + Working Set*;
 - Lista circular de molduras de páginas formando um anel a cada página carregada na memória;
 - Utiliza *bit R* e o tempo da última vez que a página foi referenciada;
 - *Bit M* utilizado para agendar escrita em disco;

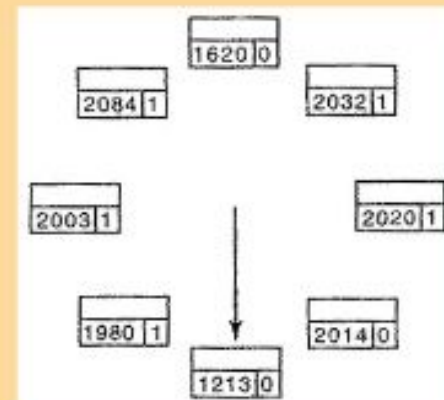
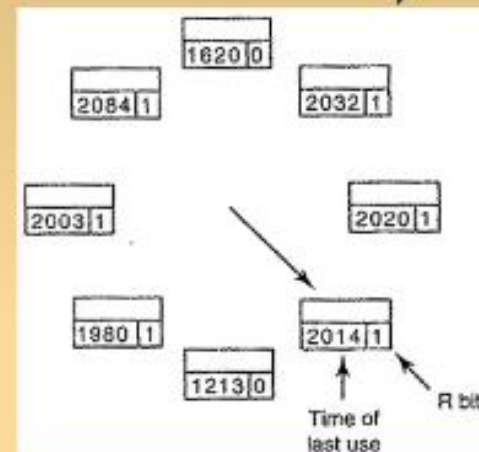
Algoritmo WSClock

- Clock + Working Set
- Amplamente usado, devido à sua simplicidade e performance
- Utiliza lista circular de páginas
 - Inicialmente vazia
 - À medida que mais páginas são carregadas, entram na lista, formando um anel
 - Cada entrada contém o tempo de último uso, além dos bits R e M

Algoritmo WSClock

- Algoritmo WSClock:
 - Funcionamento:
 - A cada page fault, a página da cabeça é examinada primeiro
 - Se $R=1$
 - A página foi usada durante o ciclo de clock corrente → não é candidata a remoção
 - Faz $R = 0$ e avança a cabeça à próxima página, repetindo o algoritmo para esta página

M não é mostrado na figura



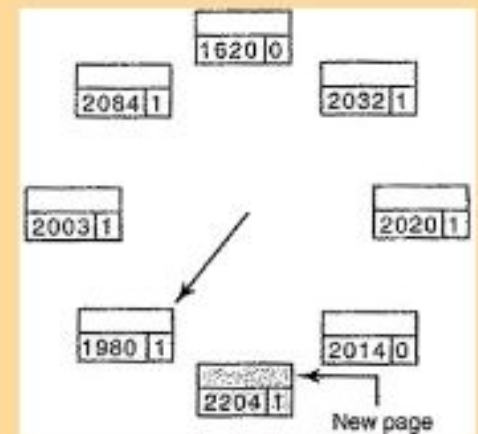
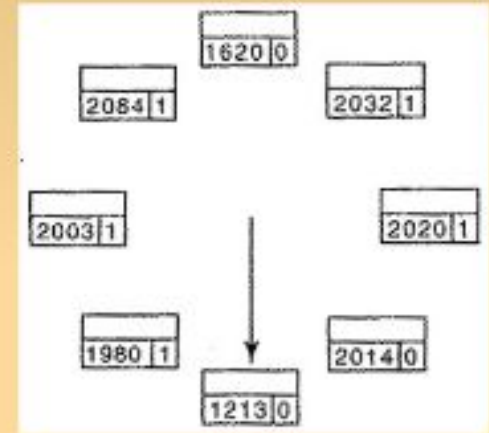
Algoritmo WSClock

- Funcionamento:

- Se $R=0$

- Se a idade for maior que o tamanho do working set t e a página estiver limpa ($M=0$) → não está no working set e uma cópia válida existe no disco

- A página é substituída
 - A cabeça da lista avança

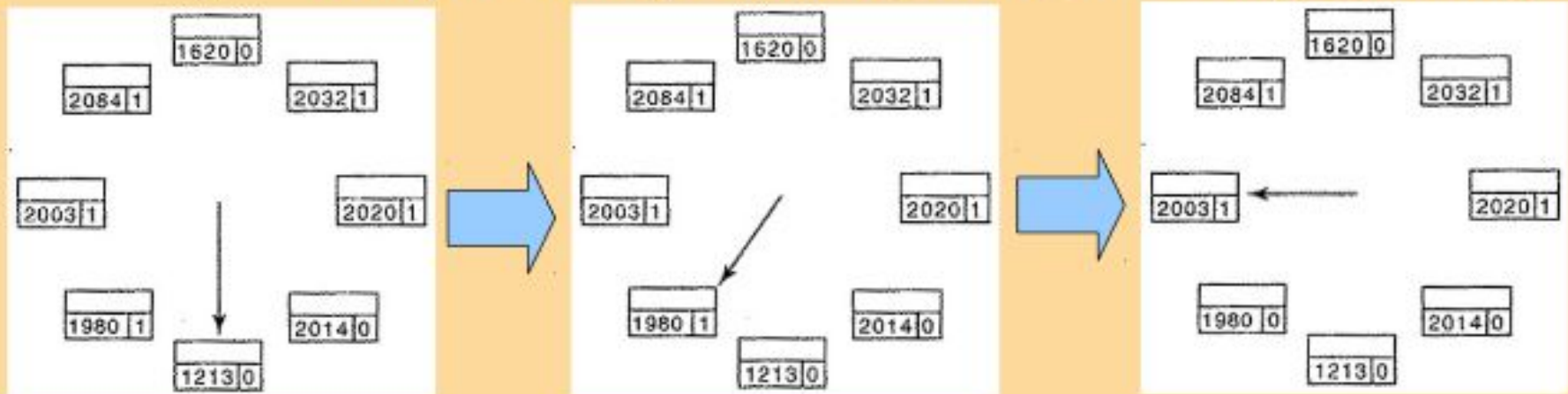


Algoritmo WSClock

- Funcionamento:

- Se $R=0$

- Se, contudo, a página estiver suja \rightarrow não possui cópia válida no disco
- Agenda uma escrita ao disco, evitando troca de processo
 - Avança a cabeça da lista, prosseguindo da página seguinte



Algoritmo WSClock

- Se a cabeça der uma volta completa na lista sem substituir:
 - E pelo menos uma escrita no disco foi agendada
 - A cabeça continua se movendo, em busca de uma página limpa
 - Em algum momento a escrita agendada será executada, marcando a página como limpa
 - E nenhuma escrita foi agendada
 - Todas as páginas estão no working set
 - Na falta de informação adicional, substitua qualquer página limpa
 - Se nenhuma página limpa existir, escolha qualquer outra e a escreva no disco

Algoritmos de Troca de Páginas

- Algoritmos de substituição local:
 - Working Set;
 - WSClock;
 - O conceito de working set se aplica somente a um único processo
→ não há working set para a máquina como um todo
- Algoritmos de substituição local/global:
 - Ótimo;
 - NRU;
 - FIFO;
 - Segunda Chance;
 - LRU;
 - Relógio;

Gerenciamento de Memória

Troca de Páginas

- 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;

Gerenciamento de Memória

Troca de Páginas

Falta de Página no Processo A

A0	10
A1	7
A2	5
A3	4
A4	6
A5	3
B0	9
B1	4
B2	6
B3	2
B4	5
B5	6
B6	12
C1	3
C2	5
C3	6

(a)

A0
A1
A2
A3
A4
A6
B0
B1
B2
B3
B4
B5
B6
C1
C2
C3

(b)

A0
A1
A2
A3
A4
A5
B0
B1
B2
A6
B4
B5
B6
C1
C2
C3

(c)

Configuração
inicial

Alocação local

Alocação
global

Gerenciamento de Memória

Troca de Páginas

- 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

Gerenciamento de Memória

Troca de Páginas - Paginação

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

Programa

- Introdução aos Sistemas Operacionais
- Processos
- **Gerência de Memória**
 - Memória virtual
 - Paginação
 - Implementação
 - Tabela de Páginas Invertida
 - Segmentação
- Sistemas de Arquivos
- Entrada/saída
- Segurança
- Exemplos de Sistemas Operacionais

Gerenciamento de Memória

Implementação da Paginação

- 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;

Gerenciamento de Memória

Implementação da Paginação

□ 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;

Gerenciamento de Memória

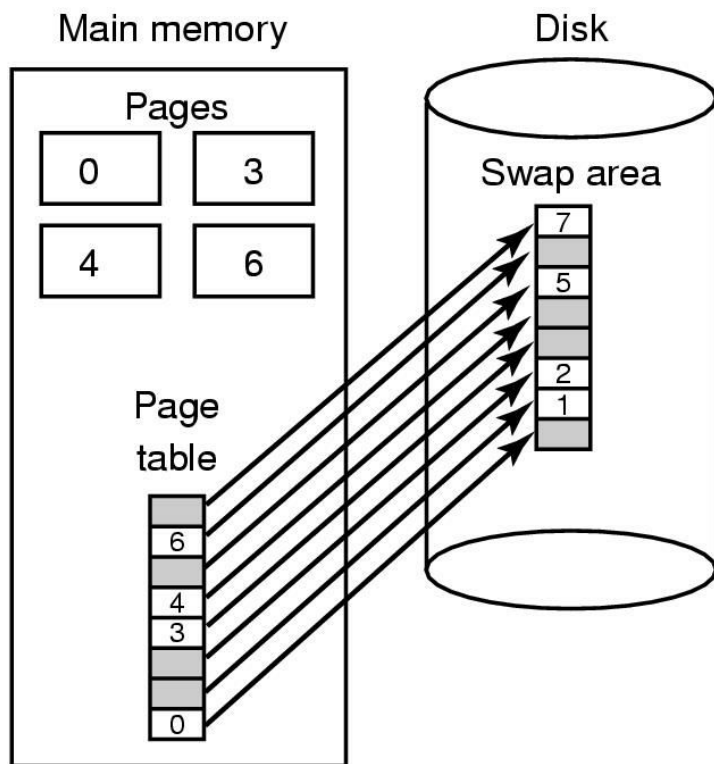
Implementação da Paginação

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

Gerenciamento de Memória

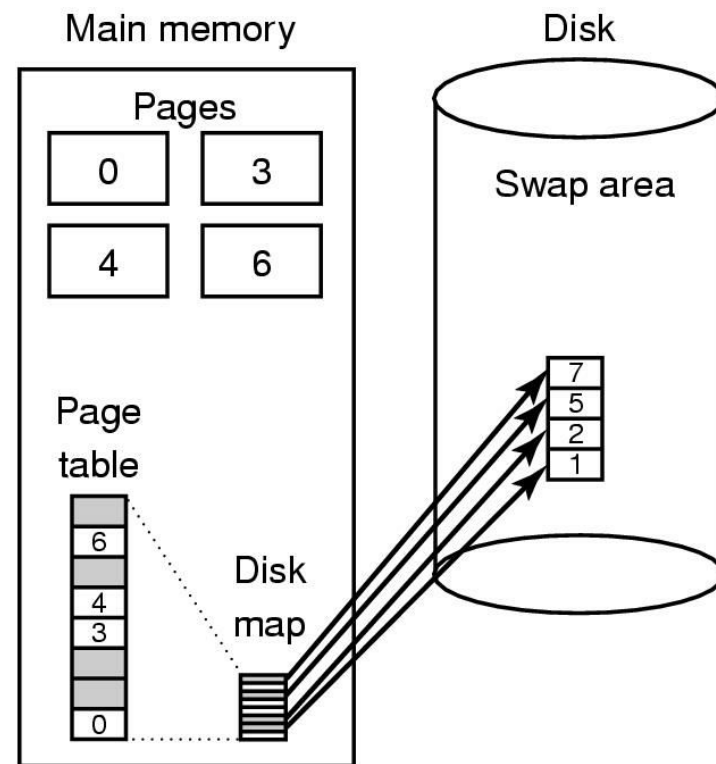
Implementação da Paginação

Como fica o disco – memória secundária



(a)

Área de troca estática



(b)

Área de troca dinâmica

Gerenciamento de Memória

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;

Gerenciamento de Memória

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>;

Gerenciamento de Memória

Tabela de Páginas Invertida

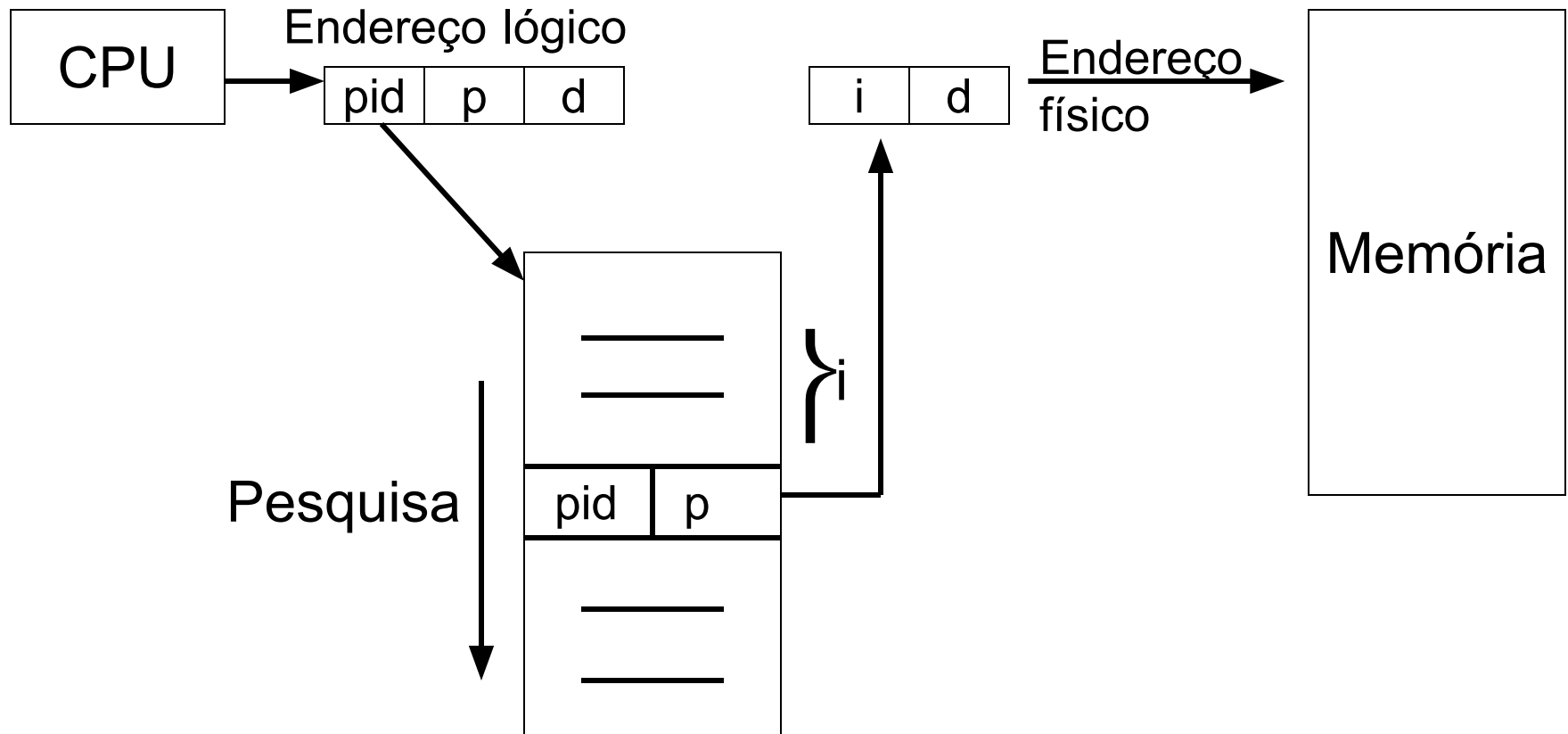


Tabela de páginas invertida

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

Gerenciamento de Memória

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;

Gerenciamento de Memória

Memória Virtual - 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 \square base + deslocamento;
 - Alocação de segmentos segue os algoritmos já estudados:
 - *FIRST-FIT*;
 - *BEST-FIT*;
 - *NEXT-FIT*;
 - *WORST-FIT*;
 - *QUICK-FIT*;

Gerenciamento de Memória

Memória Virtual - 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

Memória Virtual - Segmentação

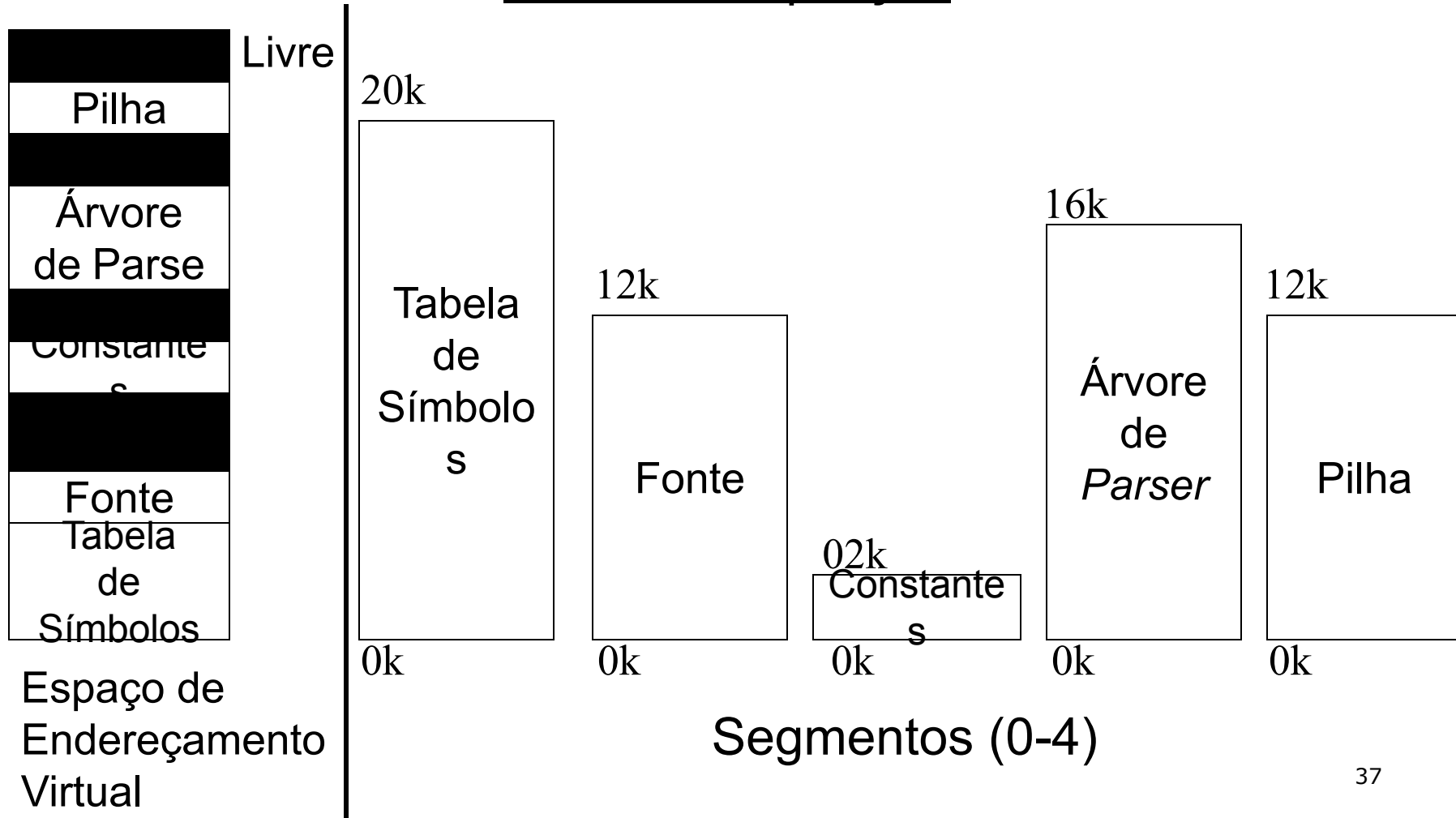
□ Segmentação:

- 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

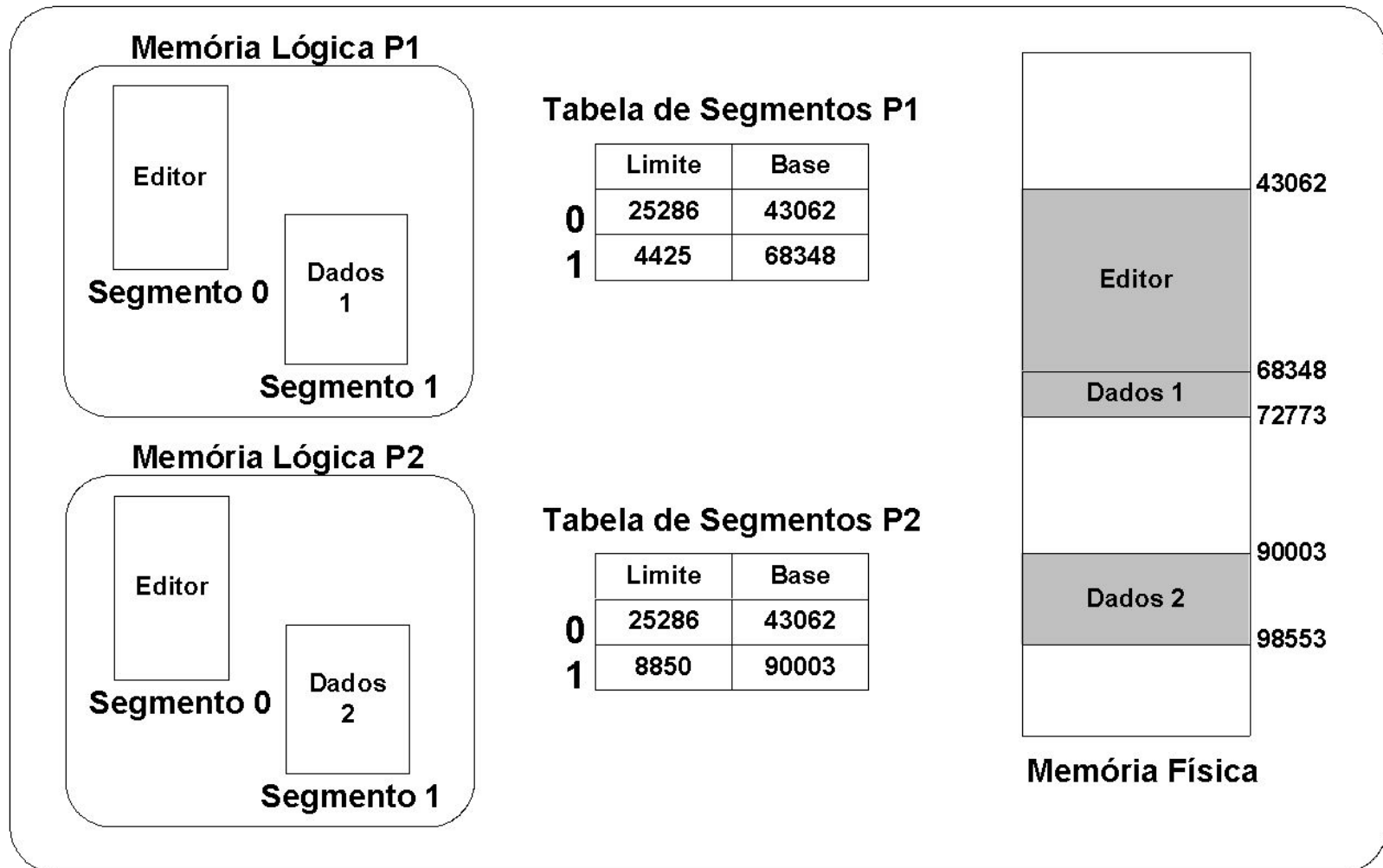
Memória Virtual - Segmentação

Tarefa: Compilação



Gerenciamento de Memória

Memória Virtual - Segmentação



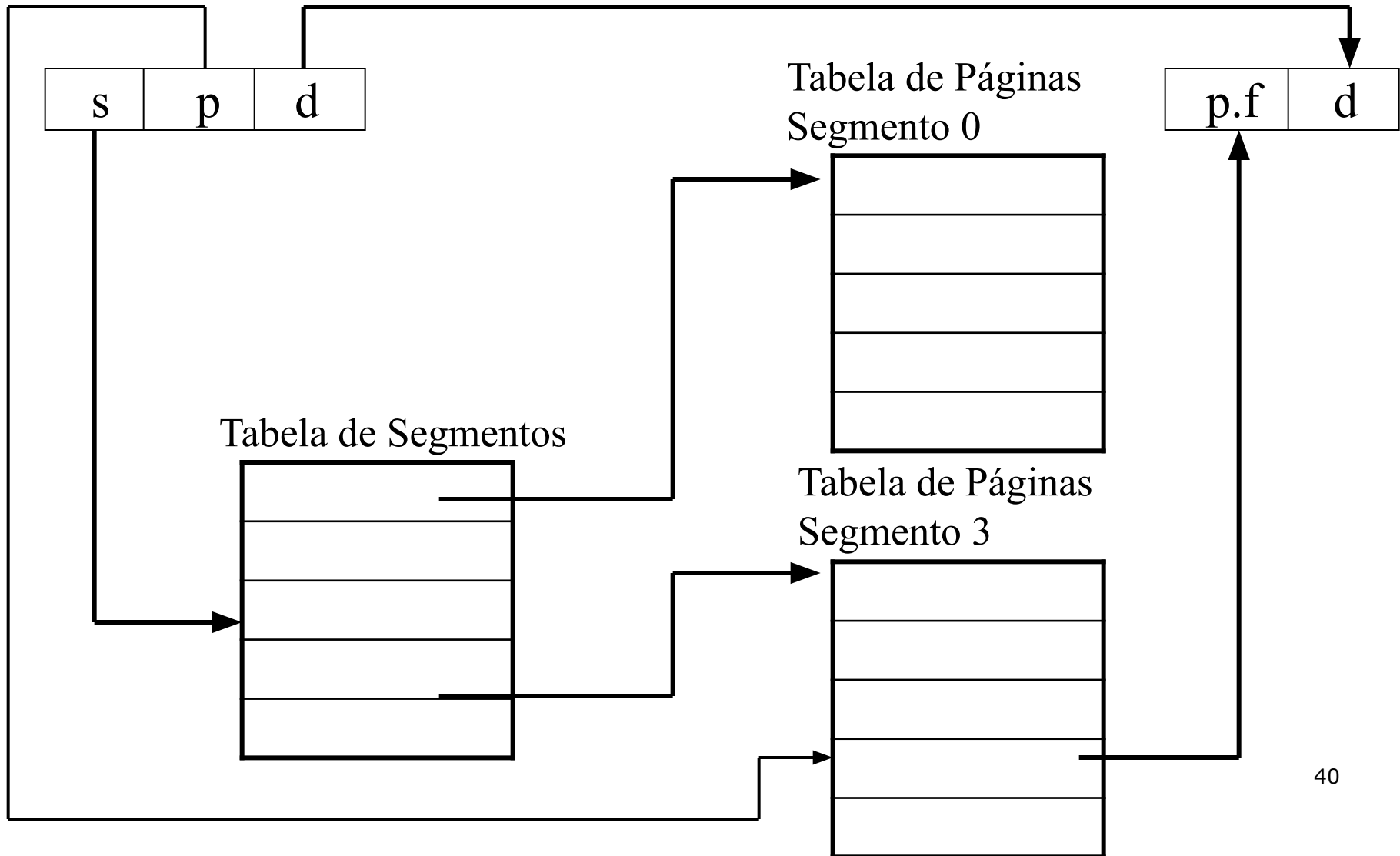
Gerenciamento de Memória

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

Memória Virtual

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

Memória Virtual

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

Programa

□ Gerência de Memória

- Gerenciamento de espaços
- Alocação de informação na memória
- Memória virtual
 - Paginação
 - Implementação
 - Tabela de Páginas Invertida
 - Segmentação

Programa

- Introdução aos Sistemas Operacionais
- Processos
- Gerência de Memória
- Sistemas de Arquivos
- Entrada/saída
- Segurança
- Exemplos de Sistemas Operacionais