

Sistemas Operacionais I

Profa. Kalinka Regina Lucas Jaquie Castelo Branco
kalinka@icmc.usp.br

Universidade de São Paulo

Novembro de 2020

- Algumas questões que surgem com relação à Paginação:
 - Onde armazenar a tabela de páginas?
 - **Qual a estrutura de uma entrada na tabela de páginas?**

Estrutura de uma tabela de páginas: 32 bits (mais comum)

- Algumas questões que surgem com relação à Paginação:
 - Onde armazenar a tabela de páginas?
 - Qual a estrutura de uma entrada na tabela de páginas?
 - **Quantas páginas reais serão alocadas a um processo?**

Alocação de Páginas

- Quantas páginas reais serão alocadas a um processo?
 - Duas estratégias:
 - **Alocação Fixa ou Estática:** cada processos tem um número máximo de páginas reais, definido quando o processo é criado.
 - O limite pode ser igual para todos os processos;
 - **Vantagem:** Simplicidade;
 - **Desvantagem:** (i) número muito pequeno de páginas reais pode causar muita paginação; (ii) número muito grande de páginas reais causa desperdício de memória principal.

- Algumas questões que surgem com relação à Paginação:
 - Onde armazenar a tabela de páginas?
 - Qual a estrutura de uma entrada na tabela de páginas?
 - Quantas páginas reais serão alocadas a um processo?
 - **Quando uma página deve ser carregada para a memória?**

Busca de Página

- Políticas de busca determinam quando uma página deve ser carregada para a memória.
 - **Paginação Simples:**
 - Todas as páginas virtuais do processo são carregadas para a memória principal;
 - Assim, sempre todas as páginas são válidas.
 - **Paginação por demanda - Demand Paging:**
 - Apenas as páginas efetivamente acessadas pelo processo são carregadas na memória principal;
 - Quais páginas virtuais foram carregadas - Bit de controle (bit de residência);
 - Página inválida.
 - **Paginação Antecipada - Anticipatory Paging:**
 - Carrega para a memória principal, além da página referenciada, outras páginas que podem ou não ser necessárias para o processo.

- Algumas questões que surgem com relação à Paginação:
 - Onde armazenar a tabela de páginas?
 - Qual a estrutura de uma entrada na tabela de páginas?
 - Quantas páginas reais serão alocadas a um processo?
 - Quando uma página deve ser carregada para a memória?
 - **Como trazer uma página para a memória?**

Busca de Página

- Página inválida: **MMU** gera uma interrupção de proteção e aciona o sistema operacional;
 - Se a página está fora do espaço de endereçamento do processo, o processo é abortado;
 - Se a página ainda não foi carregada na memória principal, ocorre uma **falta de página** (*page fault*).

Busca de Página

- **Falta de Página**

- Processo é suspenso e seu descritor é inserido em uma **fila especial** – fila dos processos esperando uma página virtual;
- Uma página real livre deve ser alocada;
- A página virtual acessada deve ser localizada no disco;
- Operação de leitura de disco, indicando o endereço da página virtual no disco e o endereço da página real alocada.

Busca de Página

- Após a leitura do disco:
 - Tabela de páginas do processo é corrigida para indicar que a página virtual agora está válida e está na página real alocada;
 - *Pager*: carrega páginas específicas de um processo do disco para a memória principal.
 - O descritor do processo é retirado da **fila especial** e colocado na fila do processador.

- Algumas questões que surgem com relação à Paginação:
 - Onde armazenar a tabela de páginas?
 - Qual a estrutura de uma entrada na tabela de páginas?
 - Quantas páginas reais serão alocadas a um processo?
 - Quando uma página deve ser carregada para a memória?
 - Como trazer uma página para a memória?
 - **Como liberar espaço na memória?**

Troca de Páginas

- A liberação é feita por meio de **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**.

Troca de Páginas

	Age
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)

- a) Configuração inicial.
- b) Alocação local.
- c) Alocação global.

Troca de Páginas

- **Algoritmos de substituição local** alocam uma fração fixa de memória para cada processo; enquanto que **algoritmos de substituição global** alocam molduras de páginas entre os processos em execução, variando o número de páginas no tempo.

Troca de Páginas

Memória Virtual

0	A
1	B
2	C
3	D
4	E
5	F
6	G
7	H

Tabela de Páginas Simplificada

0		i
1		i
2	10	v
3	3	v
4		i
5		i
6	4	v
7		i

Página Virtual

Página Real

Memória Principal

0	
1	
2	
3	D
4	G
5	
6	
7	
8	
9	
10	C
11	
12	
13	
14	
15	

Troca de Páginas

- Se todas as páginas estiverem ocupadas, uma página deve ser retirada: página vítima;
- Exemplo:
 - Dois processos P1 e P2, cada um com 4 páginas virtuais;
 - Memória principal com 6 páginas.

Troca de Páginas

Memória Virtual P1

0	A
1	B
2	C
3	D

Tabela de Páginas P1 Simplificada

0	1	v
1	5	v
2		i
3	0	v

Memória Principal

0	D
1	A
2	F
3	E
4	G
5	B

3 páginas de cada processo

Memória Virtual P2 Tabela de Páginas P2 Simplificada

0	E
1	F
2	G
3	H

0	3	v
1	2	v
2	4	v
3		i

→ P2 tenta acessar página 3! Falta de Página!

Troca de Páginas

Memória Virtual P1

0	A
1	B
2	C
3	D

Tabela de Páginas P1 Simplificada

0	1	v
1	5	v
2		I
3	0	v

Memória Virtual P2

0	E
1	F
2	G
3	H

Tabela de Páginas P2 Simplificada

0	3	v
1	2	v
2		I
3	4	v

Memória Principal

0	D
1	A
2	F
3	E
4	H
5	B

3 páginas de cada processo

→ Página 2 (virtual) é escolhida como vítima!

Troca de Páginas

- Algoritmos:
 - Ótimo;
 - NRU;
 - FIFO;
 - Segunda Chance;
 - Relógio;
 - LRU;
 - Working set;
 - WSClock.

Troca de Páginas

- Algoritmo Ótimo: Retira da memória a página que tem menos chance de ser referenciada.
 - Praticamente impossível de se saber;
 - Impraticável;
 - Usado em simulações para comparação com outros algoritmos.

Troca de Páginas

- Algoritmo *Not Recently Used Page Replacement* (NRU) - troca as páginas não utilizadas recentemente.
 - 02 bits associados a cada página - R e M
 - Classe 0 - não referenciada, não modificada;
 - Classe 1 - não referenciada, modificada;
 - Classe 2 - referenciada, não modificada;
 - Classe 3 - referenciada, modificada
 - R e M são atualizados a cada referência à memória.

Uma vez que a página foi lida, o valor do bit R será sempre igual a 1 até que o SO reinicialize-o.

Troca de Páginas

- Algoritmo (NRU)
 - Periodicamente, o bit R é limpo para diferenciar as páginas que não foram referenciadas recentemente;
 - A cada *tick* do relógio ou interrupção de relógio.
 - Classe 3 - Classe 1.
 - Vantagens: fácil de entender, eficiente para implementar e fornece bom desempenho.

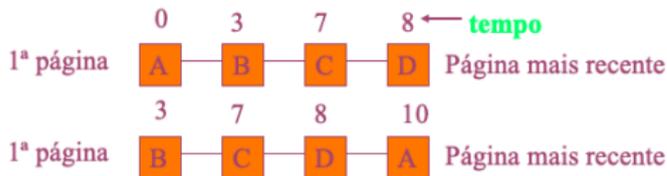
Troca de Páginas

- Algoritmo *First-in First-out Page Replacement* (FIFO)
 - SO mantém uma lista das páginas correntes na memória;
 - A página no início da lista é a mais antiga e a página no final da lista é a mais nova.
 - Simples, mas pode ser ineficiente, pois uma página que está em uso constante pode ser retirada;
 - Pouco utilizado.

Troca de Páginas

- Algoritmo da Segunda Chance.
 - FIFO + bit R;
 - Página mais velha é candidata em potencial.

Se o bit $R=0$, então página é retirada da memória, senão, $R=0$ e se dá uma nova chance à página colocando-a no final da lista.



Se página A com $R=1$; e falta de página em tempo 10; Então $R=0$ e página A vai para final da lista;

Troca de Páginas

- Algoritmo do Relógio.
 - Lista circular com ponteiro apontando para a página mais antiga;
 - Algoritmo se repete até encontrar $R=0$.

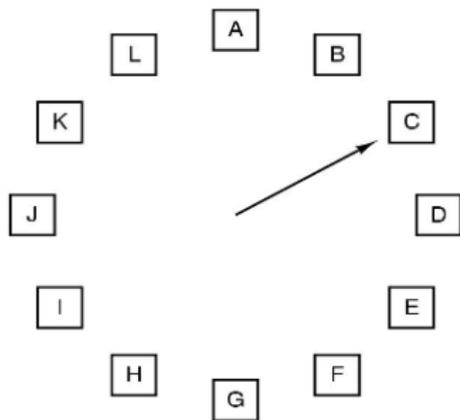
Se $R=0$

- troca de página
- desloca o ponteiro

Se $R=1$

- $R = 0$
- desloca o ponteiro
- continua busca

Troca de Páginas



When a page fault occurs, the page the hand is pointing to is inspected. The action taken depends on the R bit:

R = 0: Evict the page

R = 1: Clear R and advance hand

Troca de Páginas

- Algoritmo *Least Recently Used Page Replacement* (LRU).
 - 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.

Troca de Páginas

Algoritmo *Least Recently Used Page Replacement* (LRU).

- Pode ser implementado tanto por hardware quanto por software:
 - Hardware: MMU deve suportar a implementação LRU;
 - 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 menor valor;
 - Tabela de páginas armazena o valor desse contador para saber quantas vezes a página foi usada.

Software: duas maneiras

- NFU (*Not frequently used*);
- Aging (Envelhecimento);

Troca de Páginas

Software *Not Frequently Used*.

- Para cada página existe um contador - iniciado com zero e incrementado a cada referência à pagina
 - Página com menor valor do contador é candidata a troca.
 - Como 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.

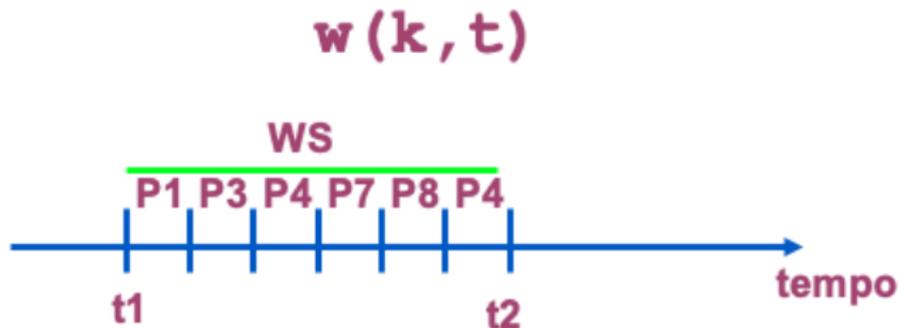
Troca de Páginas

- Software *Aging*.
 - Modificação do NFU, resolvendo o problema descrito anteriormente.
 - Além de saber **quantas vezes** a página foi referenciada, também controla **quando** ela foi referenciada.
 - Geralmente, 8 bits são suficientes para o controle se as interrupções de relógio (*clock ticks*) ocorrem a cada 20ms (10^3).

Troca de Páginas - Aging

Troca de Páginas

- Algoritmo *Working Set* (WS).
 - Paginação por demanda - páginas são carregadas na memória somente quando são necessárias;
 - Pré-paginação - *Working set*
 - Conjunto de páginas que um processo está efetivamente utilizando (referenciando) em um determinado tempo t .



Troca de Páginas

- 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;

Troca de Páginas - Working Set

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

- * Se todas as páginas estiverem com $R=1$, uma página é escolhida Randomicamente;
- ** Se todas as páginas estiverem no WS, a página mais velha com $R=0$ é escolhida;

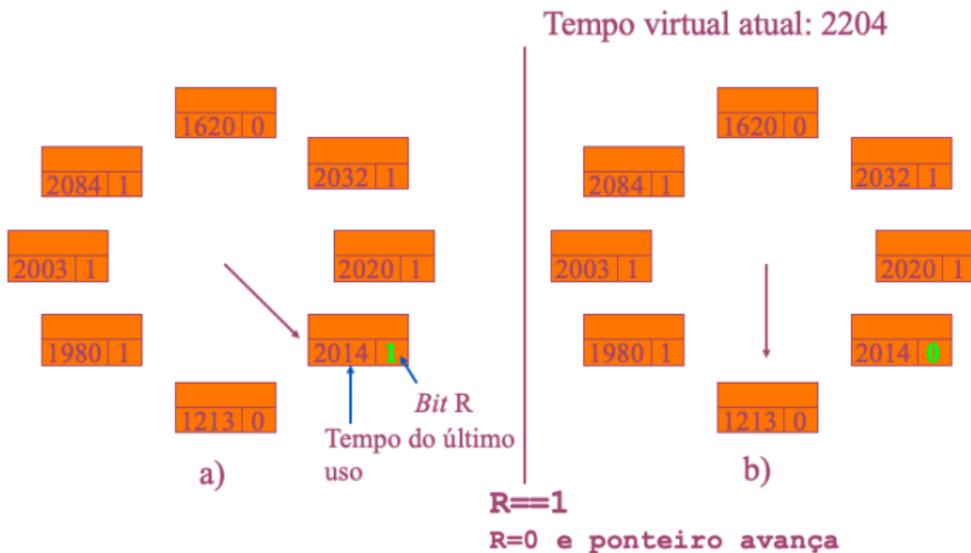
Tempo virtual atual (CVT): 2204
 $age = CVT - TLU$
 (Ex.: $2204 - 2084 = 120$)
 $\tau = \text{múltiplos } clock \text{ ticks}$

- 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*; guarda página com maior *age*;

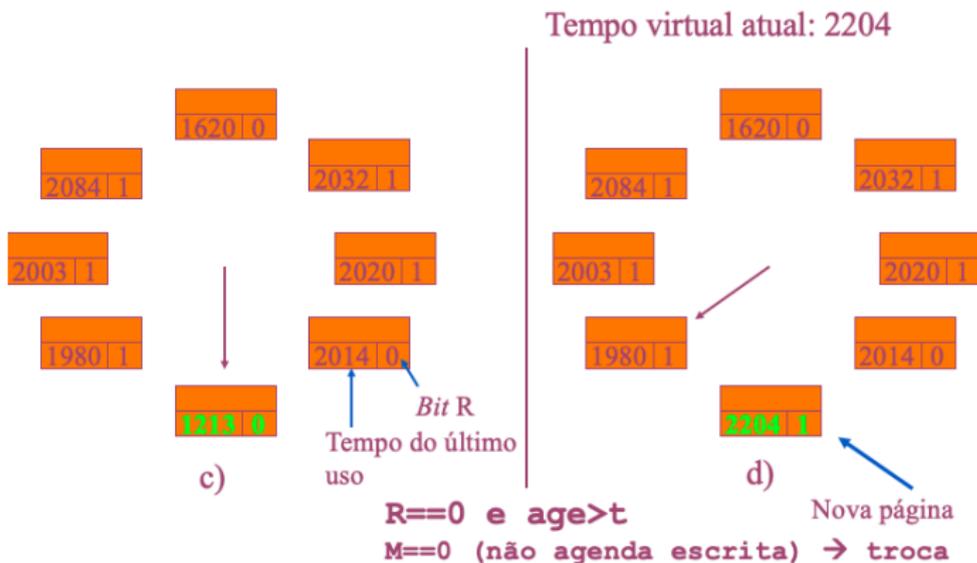
Troca de Páginas

- Algoritmo *WSClock*.
 - *Clock + Working Set*.
 - Lista circular 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.

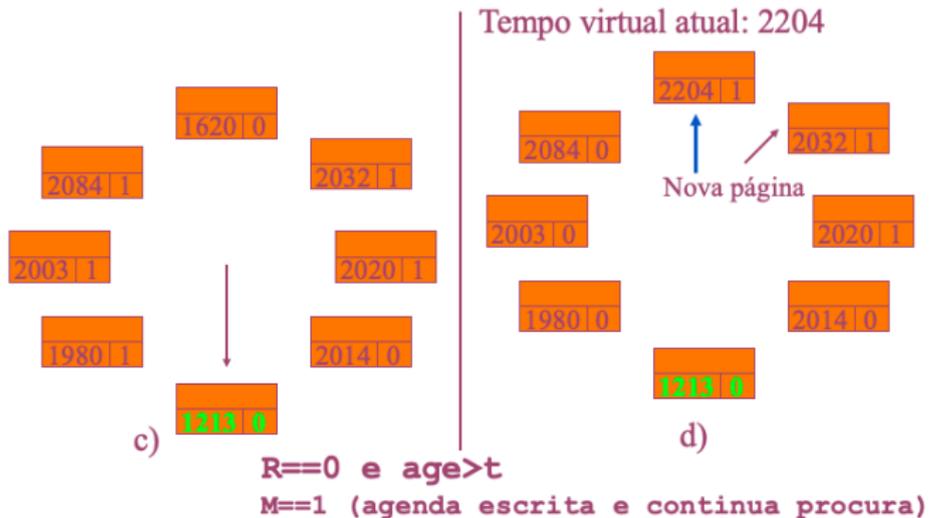
Troca de Páginas - *WSClock*



Troca de Páginas - *WSClock*



Troca de Páginas - *WSClock*



Troca de Páginas

- Algoritmo *WSClock*.
 - Se todas estiverem com $M=1$; então escreve página atual no disco, e troca a página;
 - Melhor desempenho - menos acessos ao disco.

- Algoritmo de substituição local:
 - *Working Set*.
 - *WSClock*.
- Algoritmos de Substituição Local/Global
 - Ótimo;
 - NRU;
 - FIFO;
 - Segunda Chance;
 - LRU;
 - Relógio.

Troca de Páginas

Resumo dos Algoritmos de Substituição de Páginas

Algoritmo	Comentário
Ótimo	Não implementável, mas útil como um padrão de desempenho
NRU	Muito rudimentar
FIFO	Pode descartar páginas importantes
Segunda Chance	Algoritmo FIFO bastante melhorado
Relógio	Realista
MRU	Excelente algoritmo porém difícil de ser implementado de maneira exata
NFU	Aproximação bastante rudimentar do MRU
Envelhecimento	Algoritmo bastante eficiente que se aproxima bem do MRU
Conjunto de trabalho	Implementação cara
WSClock	Algoritmo bom e eficiente

Implementação de Paginação

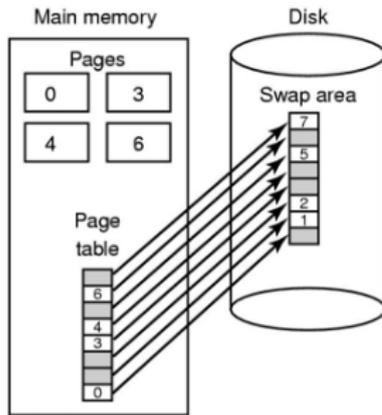
- Até agora – somente como uma página é selecionada para remoção. Mas onde essa página descartada da memória é colocada?
- **Memória Secundária – Disco**
 - A área de troca (*swap area*) é gerenciada como uma lista de espaços disponíveis;
 - O endereço da área de troca de cada processo é mantido na tabela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

Implementação de Paginação

- Memória Secundária – Disco.
 - 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.

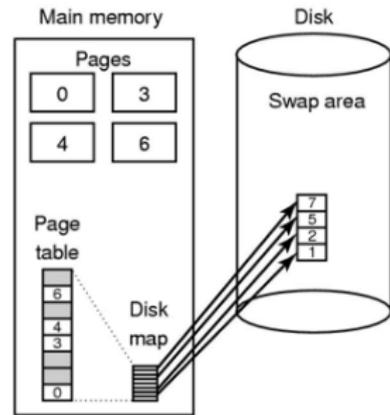
Implementação da Paginação

Como fica o disco – memória secundária



(a)

Área de troca estática



(b)

Área de troca dinâmica

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 - base + deslocamento;
 - Alocação de segmentos segue os algoritmos já estudados:
 - FIRST-FIT;
 - BEST-FIT;
 - NEXT-FIT;
 - WORST-FIT;
 - QUICK- FIT.

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.

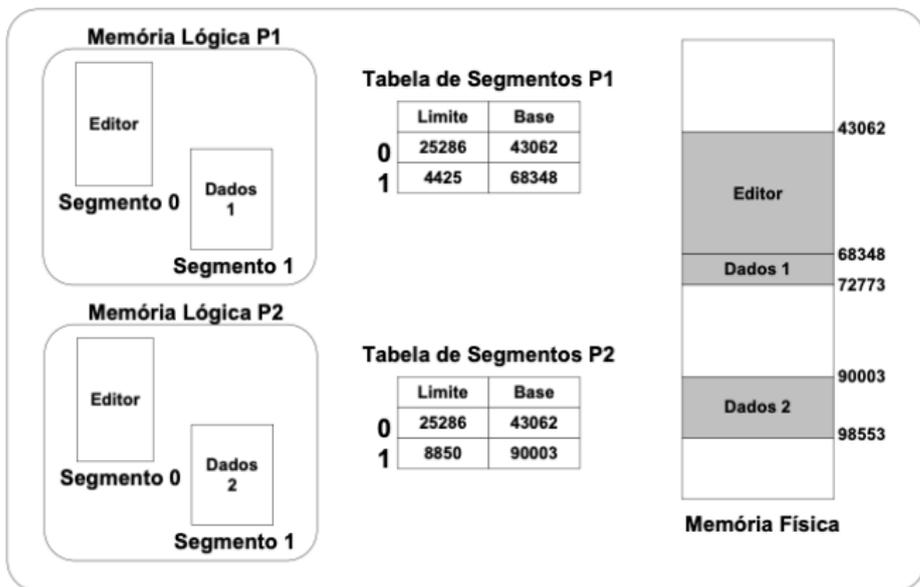
Memória Virtual - Segmentação

- Segmentação
 - Problemas encontrados - embora haja espaço na memória, não há espaço contínuo:
 - Política de realocação: um ou mais segmentos são realocados 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.

Implementação da Paginação



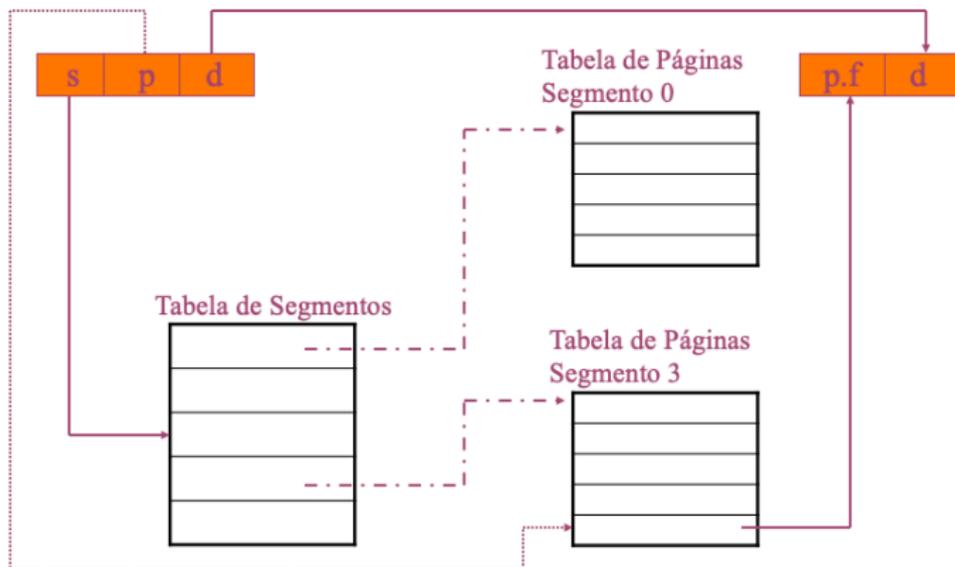
Implementação da Paginação



Memória Virtual - 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.

Implementação da Paginação



Memória Virtual - *Thrashing*

- *Thrashing* (paginação excessiva).
 - Associado com o problema de definição do número de páginas/segmentos - troca de páginas/segmentos é uma tarefa cara e lenta;
 - Se o processo tiver um número de páginas muito reduzido, ele pode ficar muito tempo esperando pelo atendimento de uma falta de página - muitos processos bloqueados.

Memória Virtual - *Thrashing*

- Evitar o problema (paginação):
 - Taxa máxima aceitável de troca de páginas;
 - Suspende alguns processos, liberando páginas físicas (*swapping*);
 - Risco de aumentar o tempo de resposta dos processos.
 - Determinar periodicamente o número de processos em execução e alocar para cada um mesmo número de páginas.
 - Problema: processos grandes teriam o mesmo número de páginas de processos pequenos, causando paginação excessiva.

Memória Virtual - *Thrashing*

- Possível solução: Número de páginas proporcional ao tamanho do processo - alocação dinâmica durante a execução dos processos;
 - PFF (*Page Fault Frequency*): algoritmo informa quando aumentar ou diminuir a alocação de páginas de um processo, controlando o tamanho do conjunto de alocação.

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

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

Continuemos com **DISPOSITIVOS DE ENTRADA E SAÍDA ...**