

Sistemas Operacionais

Prof. Dr. Helder Oliveira

Plano de Aula

- Algoritmos de substituição de páginas
- Questões de projeto para sistemas de paginação
- Questões de implementação
- Segmentação

Falta de página

- Sistema operacional escolhe uma página para remover a fim de liberar a memória para a que esta chegando.
- Se a página a ser removida foi modificada enquanto estava na memória, ela precisa ser reescrita.
- Se, no entanto, ela não tiver sido modificada portanto não é preciso reescrevê-la.
 - A página a ser lida simplesmente sobrescreve a página que está sendo removida.
- **Algoritmos de Substituição de página**

Algoritmos de Substituição de página

- Utilizados quando acontece falta de páginas.
- Podemos simplesmente escolher aleatoriamente uma página e descartar.
- O desempenho do sistema pode ser influenciado pela escolha da página.
- Muitos trabalhos, têm sido feitos sobre o assunto dos algoritmos de substituição de páginas.
- Problema da “substituição de páginas” ocorre em outras áreas do projeto de computadores.

Algoritmos de Substituição de página

Questão:

- Quando uma página será removida da memória, ela deve ser uma das páginas do próprio processo que causou a falta ou pode ser uma pertencente a outro processo?

O algoritmo ótimo de substituição de página

- Fácil de descrever.
- Impossível de implementar.
- Funcionamento:
 - Quando ocorre uma falta de página e existem um conjunto de páginas na memória.
 - Uma página será referenciada na próxima instrução.
 - Muitas outras não serão referenciadas tão cedo.
 - Páginas são rotuladas com o numero de instruções que serão executadas antes de aquela pagina ser referenciada pela primeira vez.
 - O algoritmo ótimo diz que a página com o maior rótulo deve ser removida.

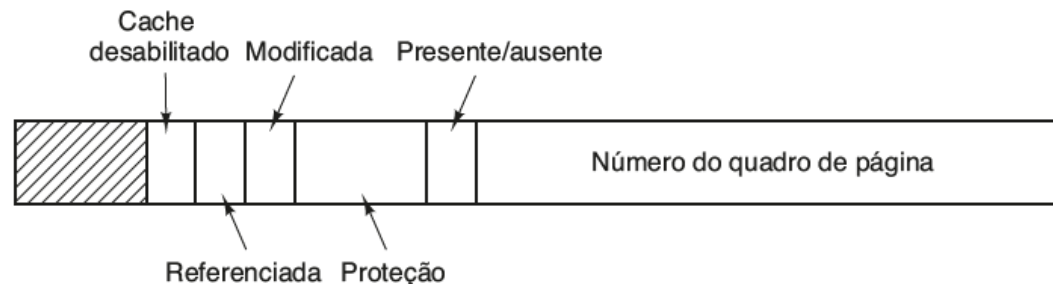
O algoritmo ótimo de substituição de página

- Funciona:
 - Cada página é marcada com o número de instruções que serão executadas antes que a página seja referenciada.
 - Retira da memória a página que tem menos chance de ser referenciada.
- Algoritmo irrealizável.
- Usado em simulações para comparação com outros algoritmos.

O algoritmo de substituição de páginas NRU

- O algoritmo de substituição de páginas não usadas recentemente.
- Utilizam dois bits **R**Referenciada e **M**Modificada que servem para o S.O. coletar estatísticas de uso de páginas úteis.
- R é colocado sempre que a página é referenciada (lida ou escrita).
- M é colocado quando a página é escrita (isto é, modificada).
- Bits precisam ser atualizados em cada referência de memória

FIGURA 3.11 Uma entrada típica de uma tabela de páginas.



O algoritmo de substituição de páginas NRU

- Os bits R (**lida ou escrita**) e M (**escrita**) podem ser usados para construir um algoritmo de paginação simples.
- Quando ocorre uma falta de página, o sistema operacional inspeciona todas as páginas e as divide em quatro categorias baseadas nos valores atuais de seus bits R e M:
 - Classe 0: não referenciada, não modificada.
 - Classe 1: não referenciada, modificada.
 - Classe 2: referenciada, não modificada.
 - Classe 3: referenciada, modificada.

O algoritmo de substituição de páginas NRU

- Classe 1: não referenciada, modificada
 - Parecem impossíveis.
 - Mas ocorrem quando uma página de classe 3 tem o seu bit R limpo por uma interrupção de relógio.
 - Interrupções de relógio não limpam o bit M porque essa informação é necessária para saber se a página precisa ser reescrita para o disco ou não.

O algoritmo de substituição de páginas NRU

- Remove uma página ao acaso de sua classe de ordem mais baixa que não esteja vazia.
- Considera melhor remover uma página modificada, mas não referenciada, a pelo menos um tique do relógio do que uma página não modificada que está sendo intensamente usada.
- Fácil de compreender.
- Moderadamente eficiente de implementar.
- Desempenho adequado.

O algoritmo de substituição de páginas FIFO

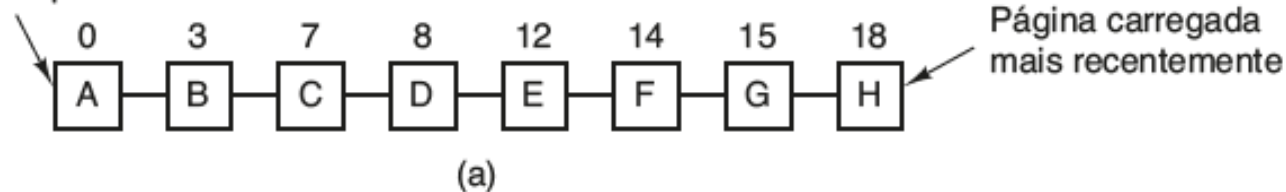
- O algoritmo de substituição de páginas primeiro a entrar, primeiro a sair.
- Mantém uma fila de páginas corrente na memória.
- **A página do início da fila é a mais antiga e a página no final da fila é a mais nova.**
- Quando ocorre uma falta de página, a página do início é removida.
- A nova é inserida no final da fila.
- Simples, mas pode ser ineficiente, pois uma página em uso constante pode ser retirada.
- Pouco utilizado.



O algoritmo de substituição de páginas FIFO

FIGURA 3.15 Operação de segunda chance. (a) Páginas na ordem FIFO. (b) Lista de páginas se uma falta de página ocorrer no tempo 20 e o bit R de A possuir o valor 1. Os números acima das páginas são seus tempos de carregamento.

Página carregada primeiro



O algoritmo de substituição de páginas 2ª chance

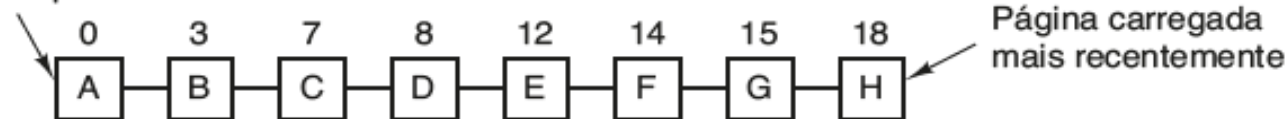
- Semelhante ao FIFO
- Inspecciona o bit R da página mais antiga.
- **Ideia:**
 - Se R for 0, a página é velha e pouco utilizada, portanto é substituída imediatamente.
 - Se o bit R for 1, o bit é limpo, e a página é colocada no fim da lista de páginas, e seu tempo de carregamento é atualizado como se ela tivesse recém-chegado na memória. Então a pesquisa continua.

O algoritmo de substituição de páginas 2ª chance

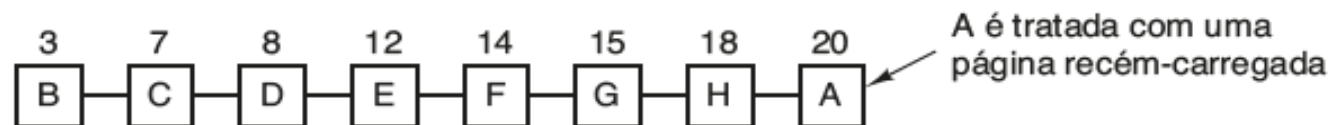
- Operação:

FIGURA 3.15 Operação de segunda chance. (a) Páginas na ordem FIFO. (b) Lista de páginas se uma falta de página ocorrer no tempo 20 e o bit *R* de *A* possuir o valor 1. Os números acima das páginas são seus tempos de carregamento.

Página carregada primeiro



(a)



(b)

- Resumo

- Em resumo o algoritmo de substituição de páginas 2ª chance procura por uma página antiga que não esteja referenciada no intervalo de relógio mais recente.

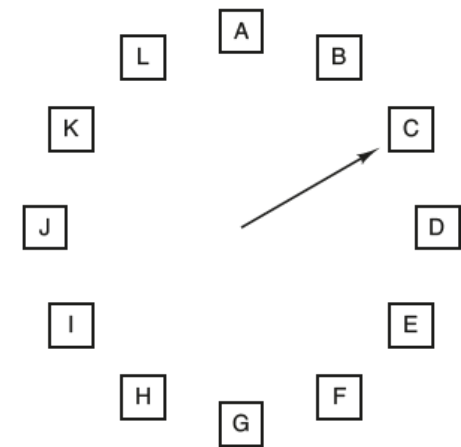
O algoritmo de substituição de páginas 2ª chance

- FIFO + bit R.
- Inspecciona o bit R da página mais velha.
- Se for 0 – é velha e não usada recentemente, então troca.
- Se for 1 – o bit é feito 0.
- A página é colocada no final da fila.
- Seu tempo de carga é modificado, fazendo parecer que chegou recentemente na memória (recebe uma segunda chance).
- A busca continua.

O algoritmo de substituição de páginas do relógio

- Melhoria para o Segunda Chance.
- Lista circular com ponteiro apontando para a página mais antiga, na forma de um relógio.
- A cabeça aponta para a página mais antiga
- Se $R_c = 0$, substitui a pagina
- Se não, avança para o próximo.

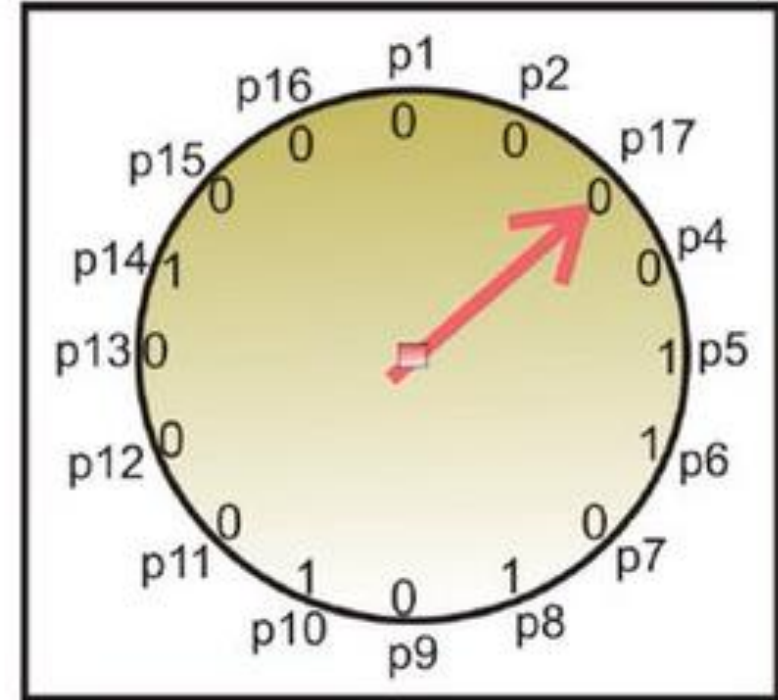
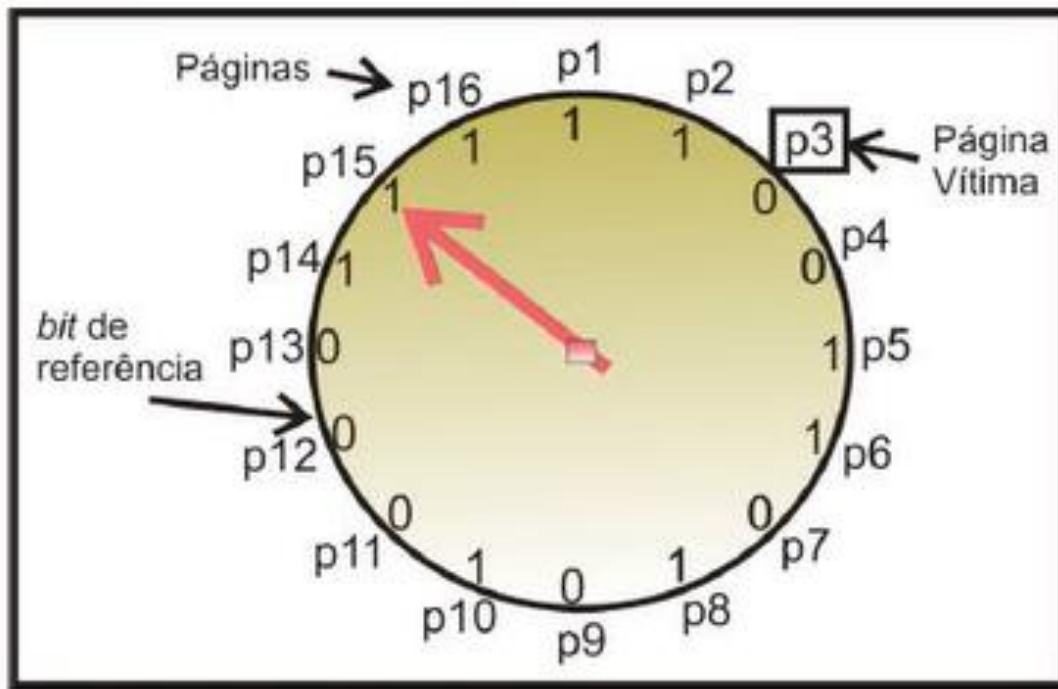
FIGURA 3.16 O algoritmo de substituição de páginas do relógio.



Quando ocorre uma falta de página, a página indicada pelo ponteiro é inspecionada. A ação executada depende do bit R:

R = 0: Remover a página
R = 1: Zerar R e avançar o ponteiro

O algoritmo de substituição de páginas do relógio



Algoritmo de substituição de páginas (LRU)

- Algoritmo de substituição de páginas usadas menos recentemente (LRU).
- Aproximação para o algoritmo ótimo.
- Baseada na observação de que as páginas que foram usadas intensamente nas últimas instruções provavelmente o serão em seguida de novo.
- Páginas que não foram usadas há eras provavelmente seguirão sem ser utilizadas por um longo tempo.
- Algoritmo realizável

Algoritmo de substituição de páginas (LRU)

- Teoricamente realizável.
 - Nem um pouco barato.
- Necessário que seja mantida uma lista encadeada de todas as páginas na memória.
 - Página mais recentemente usada na frente e a menos recentemente usada na parte de trás.
 - Lista precisa ser atualizada a cada referência de memória.
 - Encontrar uma página na lista, deletá-la e então movê-la para a frente é uma operação que demanda muito tempo, mesmo em hardware .
- Maneiras de implementar LRU
 - Software
 - Hardware

Algoritmo de substituição de páginas (LRU)

- **Nível de Hardware**

- MMU deve suportar a implementação de LRU.
- Contador em hardware (64 bits) incrementado automaticamente após cada acesso.
- Tabela de páginas armazena o valor desse contador – C – em cada entrada.
- Em uma falta de página, o SO examina todas as entradas na tabela, para encontrar a com menor C.

Algoritmo de substituição de páginas (LRU)

- **Nível de Software**

- **Algoritmo de páginas não usadas frequentemente**
- Para cada página existe um contador implementado em software. Iniciado com zero.
- Em uma falta de página, escolhe a página com o menor contador.
- Problema:
 - Esse algoritmo não se esquece de nada.
 - Páginas frequentemente acessadas muito no início não serão candidatas.

Algoritmo de substituição de páginas (LRU)

- **Algoritmo de envelhecimento**
- Modificação:
 - Primeiro, os contadores são deslocados um bit à direita antes que o bit R seja acrescentado.
 - Segundo, o bit R é adicionado ao bit mais à esquerda em vez do bit mais à direita.

Algoritmo de substituição de páginas (LRU)

FIGURA 3.17 O algoritmo de envelhecimento simula o LRU em software. São mostradas seis páginas para cinco interrupções de relógio. As cinco interrupções de relógio são representadas por (a) a (e).

	Bits <i>R</i> para as páginas 0–5, interrupção de relógio 0	Bits <i>R</i> para as páginas 0–5, interrupção de relógio 1	Bits <i>R</i> para as páginas 0–5, interrupção de relógio 2	Bits <i>R</i> para as páginas 0–5, interrupção de relógio 3	Bits <i>R</i> para as páginas 0–5, interrupção de relógio 4
	1 0 1 0 1 1	1 1 0 0 1 0	1 1 0 1 0 1	1 0 0 0 1 0	0 1 1 0 0 0
Página					
0	10000000	11000000	11100000	11110000	01111000
1	00000000	10000000	11000000	01100000	10110000
2	10000000	01000000	00100000	00010000	10001000
3	00000000	00000000	10000000	01000000	00100000
4	10000000	11000000	01100000	10110000	01011000
5	10000000	01000000	10100000	01010000	00101000
	(a)	(b)	(c)	(d)	(e)

Algoritmo de substituição de páginas (LRU)

- Algoritmo de substituição de páginas usadas menos recentemente (LRU).
- Ideia:
- Páginas muito usadas ultimamente provavelmente usadas novamente nas próximas.
- Troca a página que permaneceu em desuso pelo maior tempo .
- Alto Custo:
 - Deve-se manter lista encadeada com todas as páginas que estão na memória com as mais recentes.
- Implementada em Hardware ou Software.

Algoritmo de substituição de páginas do conjunto de trabalho

- Na forma mais pura de paginação, os processos são inicializados sem nenhuma página na memória.
 - CPU detecta falta de páginas.
 - SO traz primeira instrução.
 - **Paginação por demanda** - páginas são carregadas apenas sob demanda, não antecipadamente.
- Localidade de referencia
 - Durante qualquer fase de execução o processo referencia apenas uma fração relativamente pequena das suas páginas.
 - O conjunto de páginas que um processo está atualmente usando é o seu **conjunto de trabalho**.

Algoritmo de substituição de páginas do conjunto de trabalho

- Conjunto de trabalho.
 - Se todo o conjunto de trabalho está na memória, o processo será executado sem causar muitas faltas até passar para outra fase de execução.
 - Se a memória disponível é pequena demais para conter todo o conjunto de trabalho, o processo causará muitas faltas de páginas e será executado lentamente, já que executar uma instrução leva alguns nanossegundos e ler em uma página a partir do disco costuma levar 10 ms.
 - Um programa causando faltas de páginas a todo o momento está **ultrapaginando (thrashing)**.

Algoritmo de substituição de páginas do conjunto de trabalho

- **Multiprogramação**

- Processos muitas vezes são movidos para o disco.
- O que fazer quando um processo é trazido de volta outra vez?
 - Tecnicamente nada.
 - Processo simplesmente causará faltas de páginas até que seu conjunto de trabalho tenha sido carregado.
 - Algo lento, e também desperdiça um tempo considerável de CPU.
 - Solução: controlar o conjunto de trabalho de cada processo e certificar-se de que ele está na memória antes de deixar o processo ser executado (**modelo do conjunto de trabalho**).
 - Carregar as páginas antes de deixar um processo ser executado também é chamado de **pré--paginação**.

Algoritmo de substituição de páginas do conjunto de trabalho

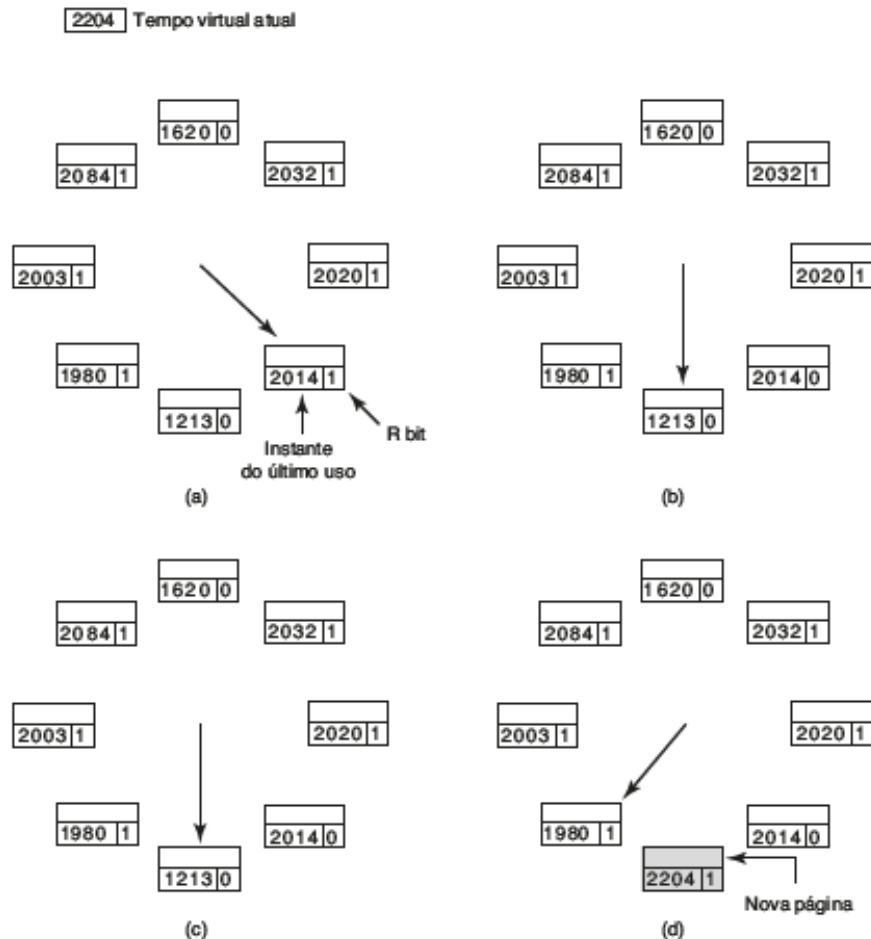
- Conjunto de páginas que um processo está efetivamente utilizando em determinado tempo t .
- Um processo só é executado quando todas páginas necessárias no tempo t estão na memória.
- A ideia é determinar o conjunto de trabalho de cada processo e tê-lo na memória antes de rodar o processo.
- Utiliza o bit **R**.

O algoritmo de substituição de página WSClock

- Um algoritmo melhorado, que é baseado no algoritmo de relógio mas também usa a informação do conjunto de trabalho.
- Devido a simplicidade de implementação e bom desempenho, ele é amplamente usado na prática.
- Utiliza lista circular de páginas com o tempo do último acesso.
 - Inicialmente vazia.
 - À medida que mais páginas são adicionadas, elas vão para a lista para formar um anel.
 - Cada entrada contém o campo do Instante do último uso do algoritmo do conjunto de trabalho básico, assim como o bit R (mostrado) e o bit M (não mostrado).
 - Como ocorre com o algoritmo do relógio, a cada falta de página, a que estiver sendo apontada é examinada primeiro.

O algoritmo de substituição de página WSClock

FIGURA 3.20 Operação do algoritmo WSClock. (a) e (b) exemplificam o que acontece quando $R = 1$. (c) e (d) exemplificam a situação $R = 0$.



O algoritmo de substituição de página WSClock

- Relógio + conjunto de trabalho.
- Amplamente usado, devido à sua simplicidade e performance.
- Utiliza lista circular de páginas com o tempo do último acesso.
- A medida que mais páginas são carregadas, entram na lista com as páginas do conjunto de trabalho.
- Cada entrada contém o tempo do ultimo uso, além dos bits **R** e **M**.

Algoritmos de substituição de páginas

FIGURA 3.21 Algoritmos de substituição de páginas discutidos no texto.

Algoritmo	Comentário
Ótimo	Não implementável, mas útil como um padrão de desempenho
NRU (não usado recentemente)	Aproximação muito rudimentar do LRU
FIFO (primeiro a entrar, primeiro a sair)	Pode descartar páginas importantes
Segunda chance	Algoritmo FIFO bastante melhorado
Relógio	Realista
LRU (usada menos recentemente)	Excelente algoritmo, porém difícil de ser implementado de maneira exata
NFU (não frequentemente usado)	Aproximação bastante rudimentar do LRU
Envelhecimento (<i>aging</i>)	Algoritmo eficiente que aproxima bem o LRU
Conjunto de trabalho	Implementação um tanto cara
WSClock	Algoritmo bom e eficiente

Questões de projeto para sistemas de paginação

- Políticas de alocação local versus global
- Controle de Carga
- Tamanho de página
- Espaços separados de instruções e dados
- Páginas compartilhadas
- Bibliotecas compartilhadas
- Arquivos mapeados
- Política de limpeza

Políticas de alocação local vs. global

- Algoritmos locais correspondem a alocar a todo processo uma fração fixa da memória.
- Algoritmos globais alocam dinamicamente quadros de páginas entre os processos executáveis. Desse modo, o número de quadros de páginas designadas a cada processo varia com o tempo.
- Em geral, algoritmos globais funcionam melhor, especialmente quando o tamanho do conjunto de trabalho puder variar muito através do tempo de vida de um processo.

Políticas de alocação local vs. global

- O algoritmo de substituição de página deve tentar encontrar a página usada menos recentemente considerando apenas as seis páginas atualmente alocadas para A , ou ele deve considerar todas as páginas na memória?

FIGURA 3.22 Substituição de página local *versus* global. (a) Configuração original. (b) Substituição de página local. (c) Substituição de página global.

	Idade		
A0	10	A0	
A1	7	A1	
A2	5	A2	
A3	4	A3	
A4	6	A4	
A5	3	A6	
B0	9	B0	
B1	4	B1	
B2	6	B2	
B3	2	B3	
B4	5	B4	
B5	6	B5	
B6	12	B6	
C1	3	C1	
C2	5	C2	
C3	6	C3	
(a)		(b)	(c)

Controle de Carga

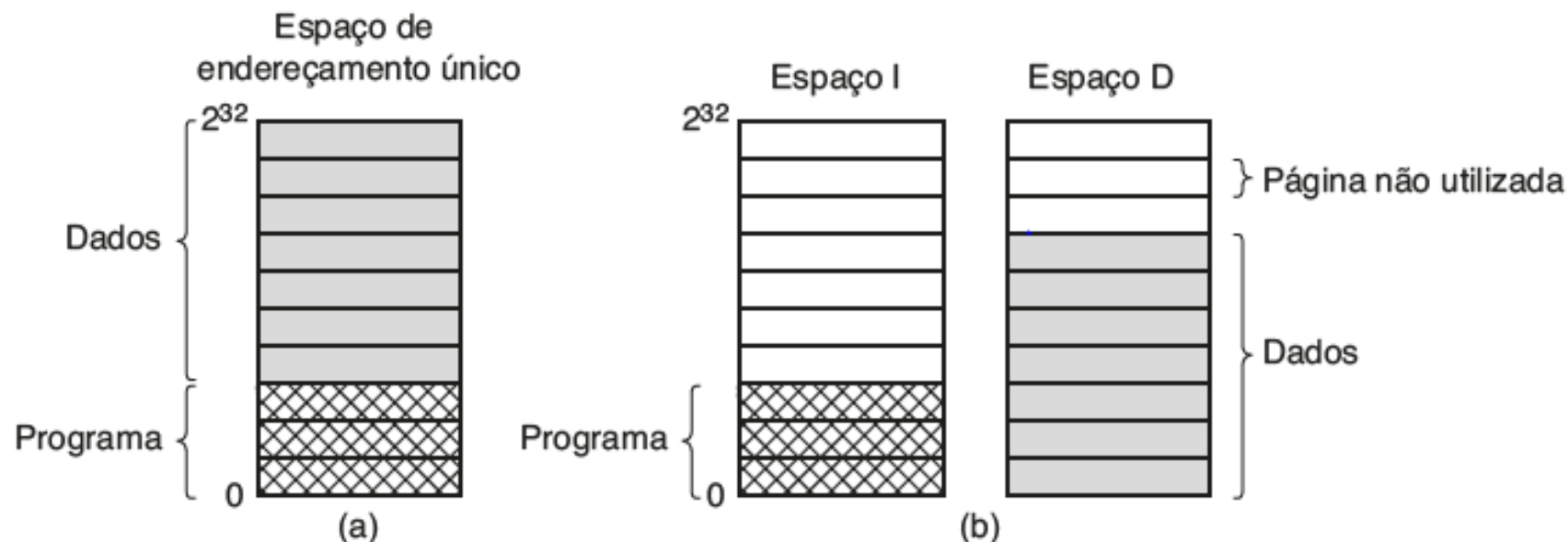
- Caso ocorra uma ultrapaginação, será necessário dar mais memória a alguns processos. Pelo fato de não haver um modo de dar mais memória àqueles processos que precisam dela sem prejudicar alguns outros, a única solução real é livrar-se temporariamente de alguns processos.
- Uma boa maneira de reduzir o número de processos competindo pela memória é levar alguns deles para o disco e liberar todas as páginas que eles estão segurando.
- É preciso considerar também o grau de multiprogramação, ou seja, o tamanho dos processos e frequência da paginação ao decidir qual processo deve ser trocado, bem como as características.

Tamanho de página

- É um parâmetro que pode ser escolhido pelo sistema operacional.
- Determinar o melhor tamanho de página exige equilibrar vários fatores, pois não há um tamanho ótimo geral.
- O espaço extra de uma página que é desperdiçado é denominado fragmentação interna.
- Um tamanho de página grande pode causar mais desperdício de espaço na memória.
- Páginas pequenas ocupam muito espaço no TLB.

Espaços separados de instruções e dados

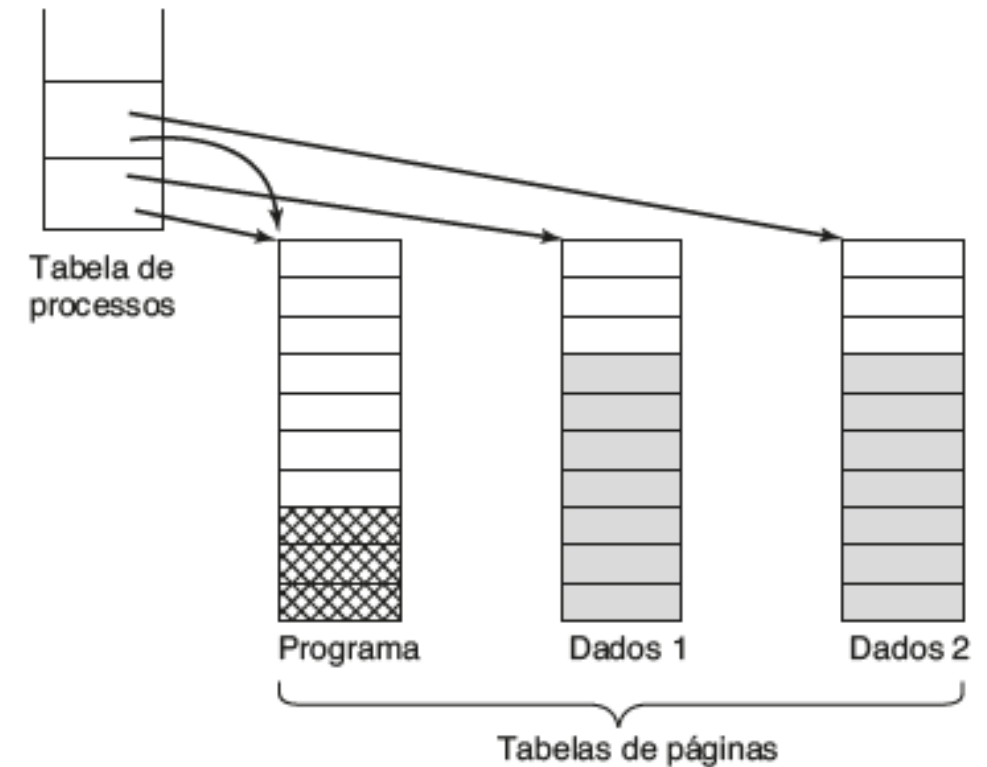
- Se o espaço de endereçamento for grande o suficiente, tudo funcionará bem. Se for pequeno demais, ele força os programadores a encontrar uma saída para fazer caber tudo nesse espaço.
- Uma solução é ter dois espaços de endereçamento diferentes para instruções (código do programa) e dados, chamados de espaço I e espaço D.



Páginas compartilhadas

- É eficiente compartilhar as páginas para evitar ter duas cópias da mesma página na memória ao mesmo tempo.
- Pesquisar todas as tabelas de páginas para ver se uma página está sendo compartilhada normalmente é muito caro, portanto estruturas de dados especiais são necessárias para controlar as páginas compartilhadas, em especial se a unidade de compartilhamento for a página individual (ou conjunto de páginas), em vez de uma tabela de páginas inteira.

FIGURA 3.25 Dois processos que compartilham o mesmo programa compartilhando sua tabela de páginas.

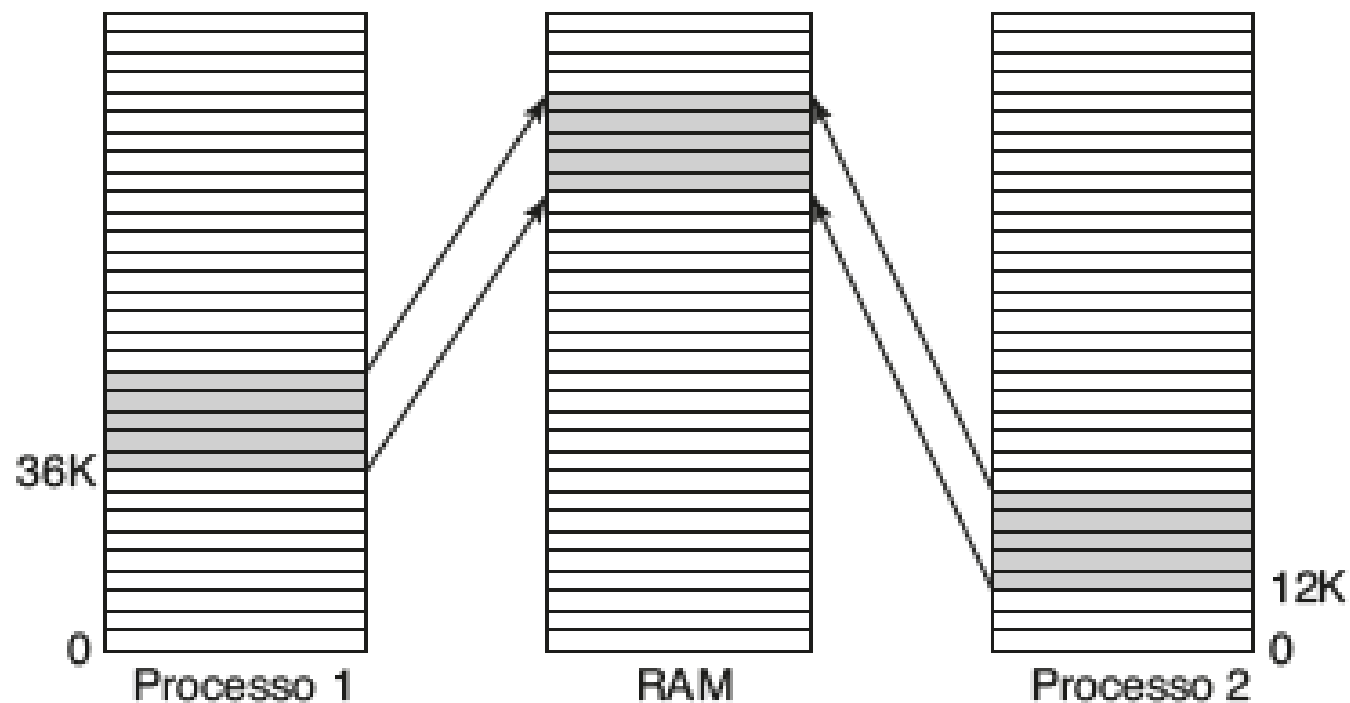


Bibliotecas compartilhadas

- São chamadas de DLLs ou Dynamic Link Libraries — Bibliotecas de Ligação Dinâmica — no Windows).
- Além de tornar arquivos executáveis menores e também salvar espaço na memória, têm outra vantagem importante: se uma função em uma biblioteca compartilhada for atualizada para remover um erro, não será necessário recompilar os programas que a chamam e os antigos arquivos binários continuam a funcionar.
- Desvantagem: a realocação durante a execução não funciona. É preciso usar o método copiar na escrita e criar novas páginas para cada processo compartilhando a biblioteca, realocando-as dinamicamente quando são criadas.

Bibliotecas compartilhadas

FIGURA 3.26 Uma biblioteca compartilhada sendo usada por dois processos.



Arquivos mapeados

- Trata-se de um processo que pode emitir uma chamada de sistema para mapear um arquivo em uma porção do seu espaço virtual.
- Em vez de fazer leituras e gravações, o arquivo pode ser acessado como um grande arranjo de caracteres na memória. Em algumas situações, os programadores consideram esse modelo mais conveniente.
- As bibliotecas podem usar esse mecanismo se arquivos mapeados em memória estiverem disponíveis.

Política de limpeza

- Os sistemas de paginação geralmente têm um processo de segundo plano, chamado de daemon de paginação que, periodicamente, inspeciona o estado da memória e garante que todos os quadros estejam limpos, assim eles não precisam ser escritos às pressas para o disco quando requisitados.

Questões de implementação

- Há quatro momentos em que o sistema operacional tem de se envolver com a paginação: na criação do processo, na execução do processo, em faltas de páginas e no término do processo.
- Backup de instrução: quando um programa referencia uma página que não está na memória, a instrução que causou a falta é parada no meio da sua execução e ocorre uma interrupção para o sistema operacional. Após o sistema operacional ter buscado a página necessária, ele deve reiniciar a instrução que causou a interrupção.
- Separação da política e do mecanismo: esse princípio pode ser aplicado ao gerenciamento da memória fazendo que a maioria dos gerenciadores de memória seja executada como processos no nível do usuário.

Questões de implementação

- Retenção de páginas na memória: se um dispositivo de E/S estiver no processo de realizar uma transferência via DMA para aquela página, removê-lo fará que parte dos dados seja escrita no buffer a que eles pertencem, e parte seja escrita sobre a página recém-carregada. Uma solução para esse problema é trancar as páginas engajadas em E/S na memória de maneira que elas não sejam removidas. Trancar uma página é muitas vezes chamado de fixação (pinning) na memória.
- Gerenciamento de disco: o algoritmo mais simples para alocação de espaço em disco consiste na criação de uma área de troca especial no disco ou, até melhor, em um disco separado do sistema de arquivos (para equilibrar a carga de E/S).

Segmentação

- Divisão do espaço de endereçamento em um número de partições com tamanhos distintos.
- Aproxima-se mais da visão do programador: um programa é uma coleção de segmentos de tamanho variável.
- Os compiladores e montadores criam automaticamente os segmentos que constituem o programa.
- Na carga do programa cada segmento recebe um número de segmentos específicos

Segmentação

- Ajuda a lidar com estruturas de dados que podem mudar de tamanho durante a execução e simplifica a ligação e o compartilhamento.
- Facilita proporcionar proteção para diferentes segmentos.
- Às vezes a segmentação e a paginação são combinadas para proporcionar uma memória virtual bidimensional.

Segmentação X Paginação

FIGURA 3.32 Comparação entre paginação e segmentação.

Consideração	Paginação	Segmentação
O programador precisa saber que essa técnica está sendo usada?	Não	Sim
Há quantos espaços de endereçamento linear?	1	Muitos
O espaço de endereçamento total pode superar o tamanho da memória física?	Sim	Sim
Rotinas e dados podem ser distinguidos e protegidos separadamente?	Não	Sim
As tabelas cujo tamanho flutua podem ser facilmente acomodadas?	Não	Sim
O compartilhamento de rotinas entre os usuários é facilitado?	Não	Sim
Por que essa técnica foi inventada?	Para obter um grande espaço de endereçamento linear sem a necessidade de comprar mais memória física	Para permitir que programas e dados sejam divididos em espaços de endereçamento logicamente independentes e para auxiliar o compartilhamento e a proteção

Segmentação com paginação

- Se os segmentos forem grandes, talvez não seja possível mantê-los na memória principal em sua totalidade. Isso leva à ideia de realizar a paginação dos segmentos, de maneira que apenas aquelas páginas de um segmento que são realmente necessárias tenham de estar na memória.
- O sistema MULTICS e o x86 de 32 bits da Intel dão suporte à segmentação e à paginação. Ainda assim, fica claro que poucos projetistas de sistemas operacionais se preocupam mesmo com a segmentação (pois são casados a um modelo de memória diferente). Em consequência, ela parece estar saindo rápido de moda. Hoje, mesmo a versão de 64 bits do x86 não dá mais suporte a uma segmentação de verdade.

Pesquisa em gerenciamento de memória

- Hoje, a pesquisa sobre a paginação se concentra em tipos novos de sistemas.
- Paginação em sistemas com múltiplos núcleos: esses tipos de sistemas tendem a possuir muita memória cache compartilhada de maneiras complexas.
- Paginação em sistemas NUMA: diversas partes da memória podem ter diferentes tempos de acesso.
- Smartphones e tablets.
- Sistemas em tempo real.

Segmentação

- A memória lógica é constituída por um conjunto de segmentos, cada um com nome e um tamanho (na prática, os segmentos são identificados por números e não por nomes).
- Páginas têm tamanho fixo, segmentação não.
- Uma posição de memória lógica é referida por um par (s, d)
 - d é o numero de segmentos.
 - d é o deslocamento (offset) dentro do segmento.

Leitura

- SISTEMAS OPERACIONAIS MODERNO 4ª edição
 - 3 - Gerenciamento de memória

Dúvidas?