

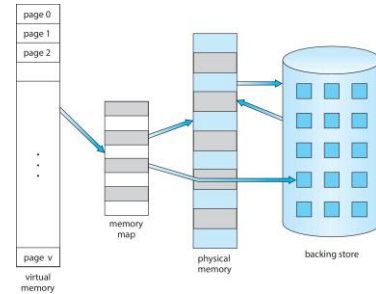


Universidade do Vale do Itajaí
Escola Politécnica

Memória Virtual

1

Memória virtual maior que a memória física



2

Paginação por Demanda



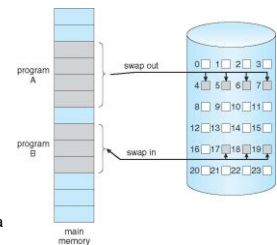
- Poderia trazer todo o processo para a memória no tempo de carga
- Ou trazer uma página para a memória apenas quando for necessário
 - Menos I/O necessário, sem I/O desnecessário
 - Menos memória necessária
 - Resposta mais rápida
 - Mais usuários
- Semelhante ao sistema de paginação com troca (diagrama à direita)
- A página é necessária ⇒ referência a ele
 - referência inválida ⇒ abortar
 - não-na memória ⇒ trazer para a memória
- **Trocador preguiçoso** – nunca faz o swap de uma página na memória, a menos que a página seja necessária
 - Swapper que lida com páginas é um paginador

3

Paginação por Demanda



- Poderia trazer todo o processo para a memória no tempo de carregamento
- Ou trazer uma página para a memória apenas quando for necessário
 - Menos I/O necessário, sem I/O desnecessário
 - Menos memória necessária
 - Resposta mais rápida
 - Mais usuários
- Semelhante ao sistema de paginação com troca (diagrama à direita)



4

Conceitos Básicos



- Com a troca, paginador adivinha quais páginas serão usadas antes de trocar novamente
- Em vez disso, o paginador traz apenas essas páginas para a memória
- Como determinar esse conjunto de páginas?
 - Precisa de uma nova funcionalidade MMU para implementar a paginação por demanda
- Se as páginas necessárias já são residentes de memória
 - Não há diferença em não-paginação por demanda
- Se a página for necessária e não residente de memória
 - Necessidade de detectar e carregar a página na memória a partir do armazenamento
 - ↳ Sem mudar o comportamento do programa
 - ↳ Sem programador precisando mudar código

5

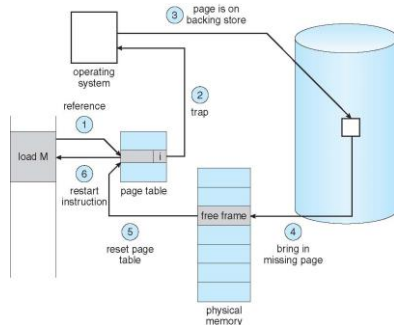
Passos para lidar com falta de página



1. Se houver uma referência a uma página, a primeira referência a essa página será um trap (interrupção de erro) para o sistema operacional
 - Falha na página
2. O sistema operacional olha para outra tabela para decidir:
 - Referência inválida \Rightarrow abortar
 - Só não está na memória
3. Encontre frame livre
4. Faz o swapping de página no frame através da operação de disco programada
5. Redefinir tabelas para indicar que a página agora está na memória
 - Definir bit de validação = **V**
6. Reinicie a instrução que causou a falta de página

6

Passos para lidar com falta de página (Cont.)



7

Lista de Frames livres



- Quando ocorre uma falha de página, o sistema operacional deve trazer a página desejada do armazenamento secundário para a memória principal
- A maioria dos sistemas operacionais mantém uma lista de frames livres - um pool de frames livres para satisfazer tais solicitações

head \rightarrow 7 \rightarrow 97 \rightarrow 15 \rightarrow 126 ... \rightarrow 75
- Sistema operacional normalmente aloca frames livres usando uma técnica conhecida como **zero-fill-on-demand** - o conteúdo dos frames é zerado antes de ser alocado
- Quando um sistema é iniciado, toda a memória disponível é colocada na lista de frames livres

8

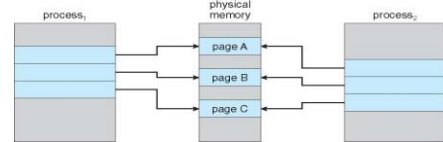
Copy-on-Write



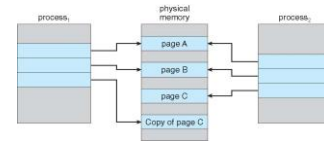
- **Copy-on-Write** (COW) permite que os processos de pais e filhos compartilhem inicialmente as mesmas páginas na memória
 - Se qualquer processo modificar uma página compartilhada, só então a página é copiada
- COW permite a criação de processos mais eficientes, pois apenas páginas modificadas são copiadas
- Em geral, páginas livres são alocadas a partir de um **pool** de páginas **zero-fill-on-demand**
 - Pool deve sempre ter frames livres para execução rápida de paginação por demanda
 - ▶ Não quero ter que liberar um frame, bem como outros processamentos na falta da página
 - Por que zerar uma página antes de alocá-la?

9

Antes do processo 1 modificar a página C



Após o processo 1 modificar a página C



10

O que acontece se não houver frame livre?



- Usado por páginas de processo
- Também em demanda do kernel, buffers de I/O, etc.
- Quanto (quantidade) alocar para cada um?
- Substituição de página – encontrar alguma página na memória, mas não realmente em uso, e trocar
 - Algoritmo – terminar? swap out? substituir a página?
 - Desempenho – queremos um algoritmo que resultará em número mínimo de faltas de página
- A mesma página pode ser trazida à memória várias vezes

11

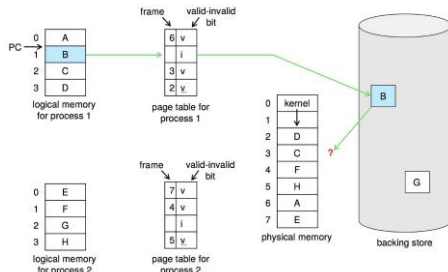
Substituição de página



- Evite a **alocação excessiva** da memória modificando a rotina de serviço de falta de página para incluir a substituição da página
- Use o **bit modificado (sujo)** para reduzir a sobrecarga das transferências de páginas – apenas páginas modificadas são escritas em disco
- A substituição da página completa a separação entre memória lógica e memória física – grande memória virtual pode ser fornecida em uma memória física menor

12

Necessidade de substituição de página



13

Substituição Básica de Página

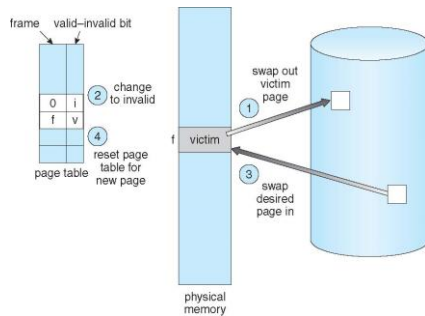


1. Encontre a localização da página desejada no disco
2. Encontre um frame livre:
 - Se houver um frame livre, use-o
 - Se não houver um frame livre, use um algoritmo de substituição de página para selecionar um **frame vítima**
 - Escreva o frame vítima para disco se estiver "sujo"
3. Trazer a página desejada para o (recém) frame livre; atualizar as tabelas de página e frames
4. Continue o processo reiniciando a instrução que causou o miss

Observe agora, potencialmente, 2 transferências de página por falta na página – aumenta EAT

14

Substituição de Página



15

Algoritmos de substituição de páginas e frames

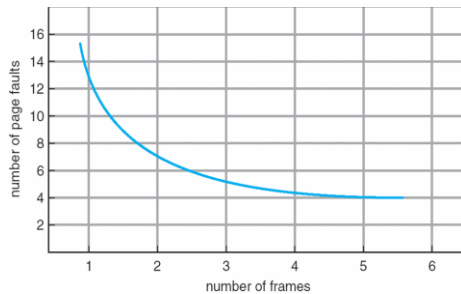


- **Algoritmo de alocação de frames determina**
 - Quantos frames para dar a cada processo
 - Quais frames para substituir
- **Algoritmo de substituição de página**
 - Quer a menor taxa de faltas de página tanto no primeiro acesso quanto no re-acesso
- Avalie o algoritmo executando-o em uma sequência específica de referências de memória (sequência de referência) e computando o número de faltas de página nessa sequência
 - String é apenas números de página, não endereços completos
 - O acesso repetido à mesma página não causa uma falta na página
 - Os resultados dependem do número de quadros disponíveis
- Em todos os nossos exemplos, a sequência de referência de números de página referenciados é

7,0,1,2,0,3,0,4,2,3,0,3,0,3,2,1,2,0,1,7,0,1

16

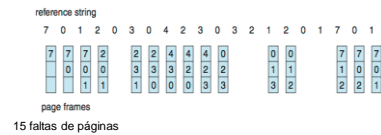
Gráfico de faltas de página vs. o número de frames



17

Algoritmo First-In-First-Out (FIFO)

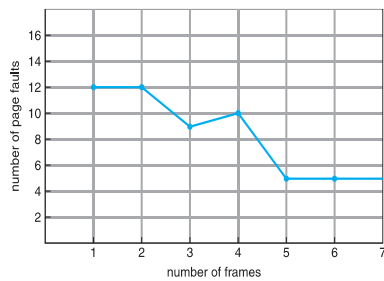
- String de referência: 7,0,1,2,0,3,0,4,2,3,0,3,0,3,2,1,2,0,1,7,0,1
- 3 quadros (3 páginas podem estar na memória por tempo por processo)



- Pode variar de acordo com a sequência de referência: considere 1,2,3,4,1,2,5,1,2,3,4,5
 - Adicionar mais frames pode causar mais falhas na página!
- Como rastrear "idades" de páginas?
 - Basta usar uma fila FIFO

18

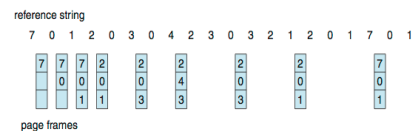
FIFO - Ilustrando a Anomalia de Belady



19

Algoritmo Ótimo

- Substitua a página que não será usada por mais tempo
 - 9 é ideal para o exemplo
- Como você sabe disso?
 - Não consigo ler o futuro
- Usado para medir o desempenho do seu algoritmo

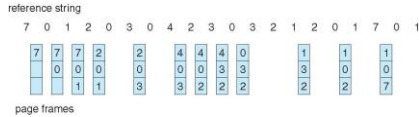


20

Algoritmo Least Recently Used (LRU)



- Use conhecimentos passados em vez de futuro
- Substitua a página que não foi usada na maior quantidade de tempo
- Associado tempo de último uso com cada página



- 12 falhas – melhor que a FIFO, mas pior que o ótimo
- Algoritmo geralmente bom e frequentemente usado
- Mas como implementar?

21

Algoritmo LRU (Cont.)



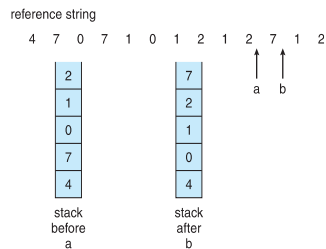
- Implementação de Contador
 - Cada entrada de página tem um contador; cada página é referenciada por meio dessa entrada, copie o relógio (clock) no contador
 - Quando uma página precisa ser alterada, olhe para os contadores para encontrar o menor valor
 - Pesquise através da tabela necessária
- Implementação de pilha
 - Mantenha uma pilha de números de página em um formulário de link duplo
 - Página referenciada:
 - movê-lo para o topo
 - requer 6 ponteiros a serem trocados
 - Mas cada atualização mais cara
 - Nenhuma busca por substituição

22

Algoritmo LRU (Cont.)



- LRU e ótimo são casos de algoritmos de pilha que não têm anomalia da Belady
- Uso de uma pilha para gravar referências de página mais recentes



23

Algoritmos de Aproximação para LRU



- LRU precisa de hardware especial e ainda lento
- Bit de referência**
 - A cada página associar um bit, inicialmente = 0
 - Quando a página é referenciada = bit definido para 1
 - Substitua qualquer um com bit de referência = 0 (se existir)
 - Nós não sabemos a ordem, no entanto

24

Algoritmos de Aproximação para LRU (cont.)

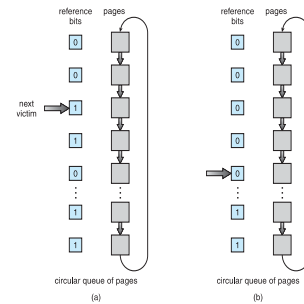


Algoritmo de segunda chance

- Geralmente FIFO, mais bit de referência fornecido por hardware
- **Substituição do relógio**
- Se a página a ser substituída tiver
 - ▶ Bit de referência = 0 -> substituí-lo
 - ▶ Bit referência = 1 então:
 - Atribui 0 ao bit referência e deixa a página na memória
 - substituir a próxima página, sujeito às mesmas regras

25

Algoritmo de Segunda Chance



26

Algoritmo aprimorado de segunda chance



- Melhora o algoritmo usando bit de referência e bit de modificado (se disponível) em conjunto
- Pegue o par ordenado (referência, modifique):
 - (0, 0) nem recentemente usado e não modificado – melhor página para substituir
 - (0, 1) não recentemente usado, mas modificado – não tão bom, deve escrever (salvar) antes da substituição
 - (1, 0) recentemente usado, mas "não sujo" – provavelmente será usado novamente em breve
 - (1, 1) recentemente usado e modificado – provavelmente será usado novamente em breve e precisa escrever antes da substituição
- Quando a substituição da página for solicitada, use o esquema de relógio (clock)
 - Talvez precise pesquisar fila circular várias vezes

27

Alocação de Frames



- Cada processo precisa de um número mínimo de frames
- Exemplo: IBM 370 – 6 páginas para lidar com instruções SS MOVE:
 - instrução é 6 bytes, pode abranger 2 páginas
 - 2 páginas para swap
 - 2 páginas do swap
- **O máximo, é claro, é o total de quadros no sistema**
- Dois grandes esquemas de alocação
 - alocação fixa
 - alocação prioritária
- Muitas variações

28

Alocação Fixa



- Alocação igualitária – Por exemplo, se houver 100 quadros (após a alocação de quadros para o SISTEMA OPERACIONAL) e 5 processos, dê a cada processo 20 quadros
 - Mantenha alguns como pool de buffer de frame livre
- Alocação proporcional – Alocar de acordo com o tamanho do processo
 - Dinâmico como grau de multiprogramação, tamanhos de processos mudam
 - s_i = size of process p_i

$$S = \sum s_i$$

$$m = \text{total number of frames}$$

$$a_i = \frac{s_i}{S} \times m$$

$$m = 64$$

$$s_1 = 10$$

$$s_2 = 127$$

$$a_1 = \frac{10}{137} \cdot 62 \approx 4$$

$$a_2 = \frac{127}{137} \cdot 62 \approx 57$$

29

Alocação Global vs. Local



- **Substituição global** – processo seleciona um frame de substituição do conjunto de todos os frames; um processo pode usar o frame de outro
 - Mas então o tempo de execução do processo pode variar muito
 - Mas maior rendimento, então mais comum
- **Substituição local** – cada processo seleciona apenas de seu próprio conjunto de frames alocados
 - Desempenho por processo mais consistente
 - Possivelmente mais memória subutilizada

Qual o Motivo????

30

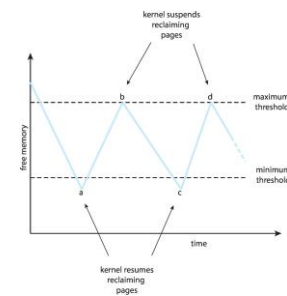
Páginas de Recuperação



- Uma estratégia para implementar a política global de substituição de páginas
- Todas as solicitações de memória estão satisfeitas da lista de frames livres, em vez de esperar que a lista caia para zero antes de começarmos a selecionar páginas para substituição,
- A substituição da página é acionada quando a lista fica abaixo de um determinado limite
- Esta estratégia tenta garantir que haja sempre memória livre suficiente para satisfazer novas solicitações

31

Exemplo de páginas de recuperação

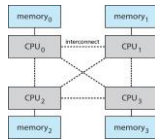


32

Acesso à memória não uniforme



- Até agora, assumimos que toda a memória acessada igualmente
- Muitos sistemas são NUMA – velocidade de acesso à memória varia
 - Considere placas de sistema contendo CPUs e memória, interconectadas em um barramento do sistema
- O desempenho ideal vem da alocação de memória "perto" da CPU em que o segmento está programado
- Modifica o escalonador para alocar o processo na mesma placa (ou núcleo) do sistema quando possível



33

Thrashing



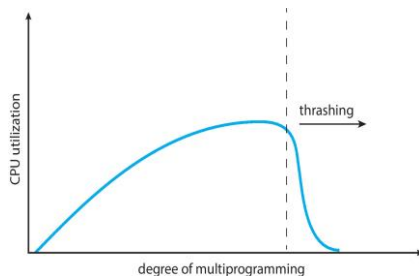
- Se um processo não tiver páginas "suficientes", a taxa de faltas de página é muito alta
 - Falta na página para obter página
 - Substituir frame existente
 - But quickly need replaced frame back
- Isso leva a:
 - Baixa utilização da CPU
 - Sistema operacional pensando que precisa aumentar o grau de multiprogramação
 - Outro processo adicionado ao sistema

34

Thrashing (Cont.)



- Thrashing** - Um processo está ocupado trocando páginas



35

Demanda Paginação e Thrashing



- Por que exige trabalho de paginação?
 - Modelo de localidade**
 - Processo migra de uma localidade para outra
 - Localidades podem se sobrepor
- Por que ocorre o thrashing?
 - tamanho da localidade > tamanho total da memória
- Limitar os efeitos usando a substituição local ou prioritária da página

36