

Leonardo Vecchi Meirelles - 12011ECP002

Cap. 20 - Questão 1)

A medida que o tamanho do espaço de endereçamento cresce, o tamanho da tabela de páginas também cresce resultando em mais sobrecarga de memória e tempos de acesso mais longos à tabela de páginas.

Esse problema torna-se significativo quando o tamanho da tabela excede o tamanho do cache de memória ou quando a tabela de páginas é muito grande para ser mantida na memória principal. Tal problema é conhecido como "page table size problem".

Cap 20 - Questão 2)

A tabela de páginas com segmentação funciona dividindo o espaço de endereço lógico de um processo em vários segmentos e, em seguida, dividindo cada segmento em páginas. Cada entrada da tabela de página (PTE) armazena o endereço base do page frame correspondente na memória física e alguns bits de controle.

Para traduzir um endereço lógico, o sistema primeiro usa o segment descriptor para localizar o endereço base e o comprimento do segmento e, em seguida, usa o número da página no endereço lógico para indexar na tabela de páginas desse segmento para obter o PTE correspondente. O endereço físico é então formado pela combinação do endereço base do page frame do PTE com o offset na página do endereço lógico.

A segmentação reduz o tamanho das tabelas de página dividindo o espaço de endereço lógico em blocos menores que podem ser gerenciados com mais eficiência.

Leonardo Vecchi Meirelles - 12011ECT002

Cap 20 - Questão 3)

Nas tabelas de páginas multiníveis, a tabela de páginas é dividida em tabelas menores. O endereço virtual é dividido em uma série de índices que correspondem aos níveis da tabela de página e cada índice é usado para percorrer a hierarquia da tabela até que a entrada final seja alcançada. Isso permite um uso mais eficiente da memória porque apenas as partes necessárias das tabelas de páginas precisam ser carregadas na memória.

As tabelas de páginas multinível também permitem o uso de espaços de endereços esparsos, onde grandes porções do espaço de endereçamento não são utilizados ou não são alocados.

Nesses casos, apenas as partes da tabela de páginas que correspondem às páginas alocadas precisam ser preenchidas, o que economiza espaço de memória.

Leonardo Vecchi Meirelles - 12011ECPO02

Cap 20 - Questão 4)

Com a introdução de tabelas de página multinível, o algoritmo de controle muda ligeiramente. Quando um processo gera um endereço virtual, os bits de ordem superior são usados para indexar na tabela de nível superior. A entrada encontrada na tabela de segundo nível aponta para uma tabela de terceiro nível, onde os bits de ordem inferior do endereço virtual são usados para indexar nesta tabela para localizar a entrada da tabela de páginas. Assim que a entrada da tabela de páginas for encontrada, o page frame físico correspondente poderá ser acessado. Se alguma das tabelas intermediárias não estiver presente na memória, ocorrerá uma falha de página (page fault) e a tabela ausente será trazida para a memória.

Leonardo Vecchi Meirelles - 12011ECT002

Cop20 - Questão 5)

Tabelas de páginas invertidas são uma forma ~~efetiva~~ alternativa de implementar tabelas de páginas, especialmente para sistemas com grande quantidade de memória física.

Em vez de cada processo ter sua própria tabela de páginas, há uma única tabela de páginas global que contém entradas para todas as páginas da memória física. Cada entrada na tabela contém informações sobre o intervalo de endereço virtual ao qual a página física corresponde, bem como informações sobre o processo que está usando essa página no momento.

Isso significa que, para cada endereço virtual, o sistema só precisa procurar uma única entrada na tabela global, em vez de pesquisar em tabelas específicas de processos potencialmente grandes. Isso pode reduzir bastante a quantidade de memória necessária para armazenar tabelas de páginas, bem como melhorar o desempenho ao reduzir o número de acessos à memória necessários para converter endereços virtuais em endereços físicos.

No entanto, gerenciar a tabela de páginas global e garantir que cada processo possa acessar apenas as páginas apropriadas pode ser mais complexo em comparação com as tabelas tradicionais. As tabelas de páginas invertidas também são menos flexíveis em termos de suporte a certas otimizações de tabelas de páginas, como compartilhamento de páginas entre processos.

Leonardo Vecchi Meirelles - 12031EC7002

Cap 21 - Questão 1)

O espaço de swap é uma área designada de um disco que pode ser usada pelo sistema operacional para armazenar temporariamente páginas que não estão sendo usadas na memória física. Quando a memória física disponível está baixa, o sistema operacional pode mover algumas das páginas para o espaço de swap para liberar espaço na memória física para outros processos.

O processo de mover páginas de e para o espaço de swap é conhecido como swapping. Quando um processo precisa de uma página que está atualmente no espaço de swap, o sistema operacional troca a página de volta para a memória física. Isso pode causar um atraso na execução do processo, pois acessar os dados do disco é mais lento do que acessá-los da memória física.

Leonardo Vecchi Meirelles - 12011ECPO02

Cap 21 - Questão 2)

Em um sistema de memória virtual, uma page fault ocorre quando uma página solicitada não está presente na memória física.

Quando ~~se~~ ocorre uma page fault, o sistema operacional precisa tratá-la trazendo-a do disco para a memória.

O fluxo do algoritmo de controle para lidar com uma page fault é o seguinte:

- 1- O processador gera uma exceção, indicando que ocorreu uma page fault.
- 2- O SO salva o estado atual do processo e examina o endereço virtual que causou a page fault para determinar a página que deveria estar presente na memória.
- 3- O SO verifica se a página está presente na memória observando a tabela de páginas. Se a página não estiver presente, ela precisará ser trazida do disco.
- 4- O SO seleciona uma página a ser removida da memória física para liberar espaço para a nova página. Esse processo é conhecido como page replacement.
- 5- A página selecionada é gravada no disco, se necessário, e sua entrada na tabela de páginas é atualizada para indicar que ela não está mais presente na memória física.
- 6- A nova página é lida do disco e sua entrada na tabela é atualizada para indicar que está presente na memória física.
- 7- O processador é então capaz de retomar a execução do processo.

Leonardo Vecchi Meirelles - 12011ECPOO2

Cap 22 - Questão 1)

A optimal page replacement policy funciona selecionando a página ~~ideal~~ que não será usada por mais tempo no futuro. Essa política requer o conhecimento de futuros acessos à memória, o que não é possível na prática. Assim, a política ótima é frequentemente usada como linha de base teórica para compor outras políticas.

Leonardo Vecchi Meirelles - 12011ECPOO2

Cap 22 - Questão 2)

A política FIFO mantém uma fila de todas as páginas na memória e despeja a página que foi trazida primeiro para a memória, ou seja, a página mais antiga. Quando ocorre uma page fault, o sistema operacional seleciona a página na frente da fila e a remove da memória. Em seguida, ele aloca a página solicitada em um page frame livre, e adiciona-a ao final da fila e atualiza a tabela de páginas de acordo. Essa política é simples de implementar e requer apenas uma pequena quantidade de memória para manter a fila, mas não considera os padrões de acesso passados ou futuros das páginas. Como resultado, ele pode tomar decisões ruins sobre quais páginas remover em determinados cenários.

Leonardo Vecchi Meirelles - 12011ECPO02

Cap 22 - Questão 3)

A política randômica seleciona aleatoriamente uma página para remover da memória quando ocorre uma page fault. Esta política não considera nenhuma informação sobre quão recente ou frequentemente uma página foi acessada. Essa pode ser uma política simples e eficaz em algumas situações, especialmente quando a carga de trabalho não exhibe nenhum padrão de acesso específico ou quando a pressão à memória é baixa. No entanto, também pode resultar em baixo desempenho quando páginas importantes ou acessadas com frequência são removidas por acaso.

Leonardo Vecchi Meirelles - 12011ECPO02

Cap 22 - Questão 4)

A política LRU é baseada na ideia de que a página que não é acessada há mais tempo tem menos probabilidade de ser usada novamente num futuro próximo.

Para implementar a política LRU, cada página possui um Timestamp que é atualizado cada vez que é acessado. Quando ocorre uma page fault, o sistema operacional procura a página com o timestamp mais antigo e a substitui pela nova página.

Leonardo Vecchi Meirelles - 12031ECT002

Cap 22 - Questão 5)

- Política Ótima: escolhe a página que não será usada por mais tempo no futuro. Essa política não é prática de implementar pois requer conhecimento futuro do uso da página, mas fornece taxa de page fault mais baixa possível.
- Política FIFO: substitui a página que está na memória há mais tempo. Essa política funciona bem para algumas cargas de trabalho, mas pode ter um desempenho ruim para outras. Ele tende a ter um bom desempenho para cargas de trabalho com localidade temporal, onde páginas acessadas recentemente provavelmente serão acessadas novamente em breve.
- Política aleatória: escolhe uma página aleatoriamente para substituir. Essa política é simples e rápida de implementar, mas pode ter um desempenho ruim para algumas cargas de trabalho. Ele tende a funcionar bem para cargas de trabalho sem localidade, onde os acessos à página são distribuídos aleatoriamente.
- Política LRU: substitui a página que não é utilizada há mais tempo. Essa política funciona bem para muitas cargas de trabalho, pois se aproxima da política ótima descartando a página menos utilizada recentemente. No entanto pode ser caro de implementar, pois requer o acompanhamento do tempo de uso de cada página.

Leonardo Vecchi Meirelles - 12031 ECP002

Cap 22 - Questão 6)

O problema com o LRU é que ele exige o acompanhamento da ordem exata em que as páginas foram acessadas, o que pode ser caro em termos de memória e computação. O LRU aproximado, também conhecido como algoritmo "Clock" ou "Second-Chance", resolve esse problema usando um buffer circular para acompanhar as páginas usadas recentemente.

Quando ocorre uma page fault, o algoritmo inspeciona a primeira página no buffer e verifica se ela foi usada recentemente (o bit de referência é definido). Nesse caso, o bit de referência é limpo e o algoritmo segue para a próxima página. Caso contrário, a página é selecionada para substituição. O algoritmo repete esse processo até que uma página adequada para substituição seja encontrada.

A ideia é que as páginas com o conjunto de bits de referência têm maior probabilidade de serem usadas novamente em um futuro próximo e recebem uma "segunda chance" de permanecer na memória. Isso fornece uma boa aproximação de LRU enquanto requer muito menos sobrecarga de memória.