UFMG - Universidade Federal de Minas Gerais
COLTEC- Colégio Técnico
PROF(A): Virgínia Fernandes Mota
Disciplina: OCS Teoria
Aluno: Ítalo Dell Areti

Exercício 1

A memória virtual é a Ilusão de uma memória maior do que a que realmente se tem (sem os altos custos da DRAM). A memória física (DRAM) age como uma cache para a memória virtual (Virtual memory - hard disk), somente um sub-conjunto da memória virtual está na memória física, o processador gera endereços virtuais porém faz acesso à memória física.Esses endereços virtuais têm que serem traduzidos em endereços físicos.

A memória virtual consiste em uma grande tabela, que contém os endereços físicos de memória, que têm os dados necessários para a execução de um processo, mas a memória virtual não contém os dados em si, apenas seus endereços, que podem estar tanto na memória primária, quanto no disco.

Exercício 2

O esquema de write-through mantém a consistência dos dados tanto na memória primária, quanto no disco, escrevendo-os ao mesmo tempo. Como o tempo de acesso ao disco é muito maior, é inviável, esperar sua escrita para o prosseguimento da execução do programa. Aplica-se então outras técnicas, como o write-back, que apenas escreve no disco, quando os dados forem requisitados futuramente, os deixando em um buffer de escrita, para que a execução do programa não sofra stall.

Exercício 3

Como o processo de write-through é inviável, utiliza-se o esquema de write-back, para manter a consistência dos dados na memória primária e no disco. Quando uma página precisa ser sobre-escrita, verifica-se através de um bit(dirty bit) se a página precisa ser escrita no disco antes de ser sobre-escrita. Este bit indicará se a página já foi lida anteriormente a marcando então com o dirty bit em 1.

Exercício 4

a) A memória cache é o nível mais alto da hierarquia de memórias do computador, e ela é subdividida em blocos de memória, que armazenarão conjuntos de palavras. Para a identificação de cada bloco, temos dois componentes, conhecidos como tag, e índice. O índice é uma marcação definida na memória cache, e corresponderá aos bits seguidos dos mais significativos de um endereço de memória.

A tag corresponderá aos bits mais significativos do endereço buscado, e será o que diferenciará por exemplo, os dados em um mapeamento direto de cache. O offset de bloco, será o que escolherá qual palavra do bloco está sendo buscada, e o offset de byte, qual byte da palavra está sendo buscado, já que as palavras são divididas em bytes.

b) 512 bits = 16 palavras = 2e9
2e9 * 32 bits = 1 palavra = 16kbits

c) Os misses e os hits são detectados ao comparar dois dos campos de cada bloco da cache: a tag e o bit de validade.
Compara-se a tag, verificando se o endereço requisitado corresponde com o que está presente na cache, e verifica-se o bit de validade é igual a 1, o que corresponde que o dado é valido, caso contrário, o dado não deve ser usado, e deve ser buscado em um nível inferior de memória. Um hit na cache ocorre quando a tag é igual aos bits mais significativos do endereço buscado, e quando o bit de validade é igual a 1.

Exercício 5

Alguns conceitos…

- Page Size: Quantidade de memória transferida do disco para a DRAM (bloco)
- Page Table : Lookup table usada para traduzir o endereço virtual em endereço físico
- Virtual page number :
- Page offset : Deslocamento na pagina.
- Physical page number : Localização da página virtual

a) O endereço inicial da tabela de páginas é dados pelo Page offset que é 12.

2^n Bits 2^{12} É o tamanho da página é 4 KB ou 12 bits.

B)
Tamanho da Page Table (p/ end 32 bits, pág de 4KB, 4B / linha da Page Table)

$2^{32} / 2^{12} = 2^{20}$

Ou 1.048.576 de entradas.

C) O endereço virtual é $_2$32 de bits(4GB) e o real é $_2$30 bits(1GB)

Exercício 6

São dois os principais princípios de localidade:

Localidade temporal: É o princípio que diz que um item referenciado uma vez, tem a tendência de ser referenciado novamente. Um exemplo deste funcionamento e de seu aproveitamento é em loops, que repetem instruções. Quando instruções repetidas são executadas, se elas já estão numa posição mais alta de hierarquia de memória, permitem com que a esta execução não resulte em uma falha, não sendo necessário procurar em outro nível a mesma instrução, pois a mesma já foi chamada alguma vez.

Localidade espacial: Nos diz que quando um item é referenciado, os itens que estão próximos a ele, têm a tendência de serem referenciados em breve. Este princípio é bastante usado já que a maioria dos programas, executam suas instruções sequencialmente, e consequentemente, utilizam instruções que estão dispostas de forma ordenada na memória. Este recurso

ser aproveitado, para um melhor desempenho do processador, que encontrará o item na posição da hierarquia de memória mais alta, evitando uma possível falha, e assim executando estas tarefas em menos tempo.

Exercício 7

A hierarquia de memória é existente nos computadores, já que atualmente dispomos de tecnologias diferentes, com recursos e preços diferentes, levando então a necessidade de organizar tipos de memórias diferentes para um bom desempenho computacional. Na maioria dos computadores, a hierarquia de memória é disposta de tal maneira (Cima para baixo):

SRAM ou cache: É a memória mais rápida, mas também a mais cara. Ela é colocada mais próxima ao processador, e permite com que quando os dados sejam requisitados do processador, eles sejam entregues a ele o mais rápido possível, caso estejam presentes na mesma.

Memória primária ou RAM: É uma memória com menos custo que a cache, rápida(menos que a cache), mas que ainda não permite quantidades grandes de armazenamento como outras tecnologias(disco). Ela é organizada de forma com que vários processos diferentes possam a utilizar, com devida proteção e velocidade.

Disco: É a memória com maior quantidade de armazenamento quando se trata de preço e densidade de memória. Apesar de ter altas capacidades, é a memória mais lenta, com um custo de acesso de milhões de ciclos no processador.

Entre a cache e a memória primária, podem existir outros níveis, como níveis secundários de cache, que irão tornar o processo de acesso ainda mais rápido.

Os níveis mostrados, estão organizados de forma hierárquica, com a velocidade disposta de baixo para cima, e a quantidade de armazenamento de cima para baixo. O nível hierárquico mais alto faz a comunicação direta com o processador, e deve ser ressaltado que cada nível se comunica apenas com os seus adjacentes, nunca pulando um nível na hierarquia.

Quando um dado é requisitado por um nível, ele devem ir ao adjacente para encontrá-lo, caso neste não esteja presente.
Os dados que estão dispostos em qualquer nível, devem estar coerentes com os outros, não mantendo informações diferentes em níveis diferentes. Tal coerência é feita por métodos como write-through ou write-back, políticas de escritas comuns nas hierarquias de memória.

Exercício 8

Definições:

Disneyland: write-back.
PortAventura: write-through.

Existem duas principais políticas de escrita na memória. Elas são necessárias, para manter a consistência dos dados em todos os níveis na hierarquia de memória. Por exemplo, quando algum item é modificado na cache, ele deve ser também alterado na memória primária, para que haja tal consistência dos dados, e que a organização das informações permaneçam em ordem.

O método write-through é o mais simples, que consiste em sempre escrever os novos dados na cache e na memória primária, criando um stall no processador até que ambas estejam de acordo.
Apesar de sua simplicidade, ela não apresenta um bom desempenho, já que uma vez que uma informação é alterada em um nível alto, deve-se esperar até que o dado seja alterado no nível mais baixo também, o que reduz drasticamente o desempenho do processador.

A implementação deste método é inviável por exemplo com a coerência entre a memória primária e o disco, já que o tempo de acesso do segundo é enorme, e acarretaria num desempenho péssimo.

Para solucionar os problemas do método write-through, podemos implementar o write-back, que dispõe de um buffer de escrita, que conterá os dados que são alterados na cache, para aguardarem para serem escritos na memória primária. Quando um item é alterado na cache, ele é enviado ao buffer de escrita, e aguarda até que este item escrito seja substituído na cache, indicando que ele precisa ser armazenado.

O método de write-back, apesar de ser mais caro, por precisar de um hardware específico para o controle desta tarefa, que é complexo e mais trabalhoso de ser implementado, ele é um método que permite um melhor aproveitamento do desempenho do processador, evitando stalls desnecessários, e mantendo a coerência necessária em ambos os níveis de memória. O método de write-through é bom para casos simples, e que não demandam grande processamento, mesmo assim o write-back é mais vantajoso, funcionando com melhor desempenho, mesmo que seja mais caro.

#TeamDisneyland!