UNIVERSIDADE ESTADUAL DE MARINGÁ
Ciência da Computação
ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES II – PROFESSOR MAURÍLIO

Trabalho Prático
Simulador Protocolo MESI em uma aplicação

OBJETIVO

Implementar um programa que simule uma memória cache gerenciada utilizando o protocolo MESI (Modify, Exclusive, Shared, Invalid) implementado em uma aplicação real.

DESCRIÇÃO

O trabalho consiste em realizar uma simulação de pelo menos **três processadores**, cada um com um nível de memória cache dedicada e uma memória RAM compartilhada entre os processadores.
O programa deve conter **dois espaços de endereçamento**, um para simular a memória principal e outro para simular a memória cache alocada a cada um dos processadores.

O programa deve solicitar acessos aos dados contidos na memória. O espaço da memória principal deve ter pelo menos 50 posições, e o espaço da memória cache deve ter pelo menos 5 posições. O tamanho do bloco da memória RAM deve ser levado em conta, pois um bloco da memória RAM é alocado a uma linha da memória cache.

As solicitações podem ocorrer em cada um dos processadores, ou seja, o processador 1 pode solicitar um dado da posição X da memória RAM e o processador 2 também pode solicitar dados de qualquer posição da memória.

A representação da memória e das cache devem ser similares às representações abaixo

Memória principal	
Linha	**Dado**
0	3434
1	24121
2	232
3	5028
4	81092
5	91639
6	54432
...	...

Memória cache – P1	Tags – P1

Memória cache – P2	Tags – P2

O programa no seu início deve preencher o espaço da memória principal com valores aleatórios (numéricos, alfa numéricos, ou outras informações de acordo com a aplicação). **Em seguida, o programa deve possibilitar ao usuário selecionar um processador, e realizar a solicitação de um dado da memória principal, sendo essa solicitação com ou sem alteração de seu valor (leitura ou escrita)**.

A **função de mapeamento** da memória cache deve preencher as linhas em ordem aleatória, o algoritmo de substituição a ser utilizado deve ser o **FIFO** e a política de escrita deve ser o **write-back**.

Quando a solicitação é realizada, cada linha da memória cache pode estar associada aos valores de acordo com o **protocolo MESI** (modify – exclusive – shared – invalid – modificada, exclusiva, compartilhada e inválida). Esta informação deve estar visível ao visualizar a representação da memória cache.

Uma linha da cache é modificada, quando o seu valor foi alterado na cache e ainda permanece inalterado na RAM.
Uma linha da cache é exclusiva quando somente um processador está acessando a posição da memória RAM.
Uma linha da cache é compartilhada se mais de um processador está acessando a posição da memória RAM.
Uma linha da cache é inválida se outro processador alterou o valor da posição da memória RAM.

Os seguintes tipos de transações devem estar definidos:

- RH (read hit – leitura com acerto) – quando a leitura de um dado é solicitada e este já está na cache do processador solicitante – (o estado é mantido – se está modificado fica modificado, se está compartilhado fica compartilhado, e se está exclusivo fica exclusivo)
- RM (read miss – leitura com falha) – quando a leitura de um dado é solicitada e este não está na cache do processador solicitante – (se o dado está presente em outra cache a linha é marcada como compartilhada, e se o dado não está presente em outra cache é marcado como exclusiva)
- WM (write miss – escrita com falha) – quando a escrita de um dado é solicitada e este dado não está na cache do processador solicitante - (após a linha ser carregada é marcada como modificada, se a linha estiver presente em outra cache, é marcada como inválida na outra)

- WH (write hit – escrita com acerto) – quando a escrita de um dado é solicitada e este dado está na cache do processador solicitante – (se a linha é compartilhada muda-se para modificada e todas as outras para inválida, se a linha é exclusiva muda-se para modificada, e se a linha já está marcada como modificada o seu estado é mantido)

Ao executar qualquer solicitação da memória deve ser destacado se ocorreu um RH, RM, WM ou WH, destacando ainda o estado da linha (modificada, exclusiva, compartilhada ou inválida)

O campo TAGS da memória cache deve armazenar as informações relevantes sobre a linha da memória cache, como a posição correspondente da memória RAM (bloco) e as informações referentes ao protocolo MESI. Além disso estas informações NÃO DEVEM estar armazenadas na memória RAM, somente na memória cache.

O simulador deve estar inserido em uma aplicação real qualquer, tal como um jogo, um sistema de reserva/compra de passagens, de bilhetes de cinema etc. Cada dupla deve escolher a sua aplicação, e a mesma deve ser diferente das outras equipes. Assim, é necessário indicar no link dos temas a aplicação a ser desenvolvida.

Assim, as funções da aplicação devem definir leituras e escritas em dados compartilhados, tal como exemplo um jogo com um mundo compartilhado, que pode ser acessado por mais de um usuário. No jogo por exemplo, algumas destas informações compartilhadas podem ser lidos e/ou escritos e a gerência destas leituras e escritas é feita com base no protocolo MESI.

Na aplicação, o usuário deve ser capaz de selecionar "um processador", escolher se quer leitura ou escrita, e em qual informação deseja solicitar, sendo exibido então as informações associadas ao protocolo MESI.

OBSERVAÇÕES

Serão aceitos trabalhos nas seguintes linguagens de programação: C, C++, C#, Pascal, Delphi, Java, JavaScript, Python, ou qualquer uma linguagem de programação esotérica. Demais linguagens sob consulta com o professor.

ENTREGA

Cada **dupla** deve entregar três arquivos, um **relatório técnico,** o(s) **arquivos fonte do programa**, e um **arquivo executável**, bem como as **instruções** para **compilar** e **rodar** o programa.

O relatório deve conter:

- Introdução
- Objetivos/Justificativa
- Funcionamento da memória cache
- Funcionamento do protocolo MESI
- Funcionamento da aplicação (regras de negócio)
- **Decisões de projeto para a implementação**
    - Estruturas de dados utilizadas
    - Gerenciamento do protocolo MESI
    - A aplicação (questões relativas à implementação)
- Conclusão
- Referências

O código fonte deve estar **comentado** em suas partes principais, e **bem estruturado**.

AVALIAÇÃO

A avaliação do trabalho será a soma das seguintes notas:

- Código fonte (0 a 6) – 3,0
- Relatório (0 a 2) – 1,0
- Apresentação (0 a 2) – 1,0

A apresentação será somente para o professor, que solicitará a execução de comandos e acessos ao simulador.

Trabalhos copiados serão **zerados**.
Trabalhos que não atendam as especificações deste documento serão **zerados**.
O não cumprimento de **qualquer** um dos itens deste documento terá seu trabalho **zerado.**

DATA DE ENTREGA

Envio dos arquivos via formulário online até as **23:59** do dia **22/08/2024** (o formulário será disponibilizado no Classroom da disciplina)

Apresentação prevista para o dia 29/08/2024

REFERÊNCIAS

Stallings, William; Arquitetura e organização de computadores; 8ª edição, São Paulo; Pearson Pratice Hall, 2010

Tanenbaum A. S.; Organização Estruturada de Computadores 5ª edição Pearson 2007

Hennesy, J.; Patterson, D.; Organização e Projeto de Computadores 3ª Edição Ed. Campus 2005

Monteiro, Mario. Introdução à Organização de Computadores. Editora LTC, 2007.

Carter, Nicholas. Arquitetura de Computadores Coleção Schaum Ed. Bookman 2003

WEBER, Raul Fernando. Fundamentos de arquitetura de computadores. 3. ed. Porto Alegre: Bookman: Instituto de Informática da UFRGS, 2008.