

Trabalho Prático do 2º Bimestre

Simulador de memória cache

OBJETIVO

Implementar um programa que simule uma memória cache, com função de mapeamento associativo (aleatório), 2 tipos de algoritmos de substituição e política de escrita *write-back*.

DESCRIÇÃO

O trabalho consiste em realizar uma simulação de uma memória cache. O programa deve conter dois espaços de endereçamento, um para simular a memória principal e outro para simular a memória cache, e gerar acessos aos dados contidos na memória. O espaço da memória principal deve ter mais de mil posições, e o espaço da memória cache pode ter qualquer tamanho.

Memória principal
4565
3434
24121
232
5028
81092
91639
54432

Memória cache

O programa no seu início deve preencher o espaço da memória principal com valores numéricos aleatórios. Em seguida, o programa deve gerar aleatoriamente acessos aos valores, estes acessos podem ser somente leitura, e leitura e modificação, sendo que a modificação de um valor lido será a troca por outro valor aleatório.

Exemplo: como primeiro acesso à memória, gera-se o acesso ao valor “232” da memória principal. O programa deve verificar se o dado está na memória cache, como é a primeira simulação, o dado não estará, logo deve-se buscar um bloco da memória principal e alocar em uma linha da cache.

4565
3434
24121
8972
5028
81092
232
54432

Memória cache
232 54432

Em seguida o programa deve usar a função de mapeamento associativo (**um bloco da memória principal pode ser mapeado em qualquer linha da memória cache**) para armazenar o dado na memória cache.

Em algum momento do programa, a memória cache estará cheia de dados, e poderá ser solicitado uma leitura de um dado que não está na memória. Neste momento o seu trabalho deve usar um algoritmo de substituição para liberar uma das linhas da memória cache para que a linha seja preenchida com outro dado. Para isso deve ser utilizado algum algoritmo de substituição (LRU, LFU, FIFO, Aleatório). Deve ser implementado dois destes algoritmos de substituição, e deve ser permitido alterar entre um algoritmo de substituição e outro durante a execução da simulação.

O programa deve ainda utilizar a política de escrita write-back, ou seja, ao gerar um acesso de leitura e modificação à um dado, este deve ser modificado apenas na memória cache, e quando a linha da cache onde este dado está armazenado for selecionada pelo algoritmo de substituição para ser descartado, somente neste momento deve ser atualizado o valor da memória principal.

OBSERVAÇÕES

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

ENTREGA

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

O relatório deve conter:

- Introdução
- Objetivos/Justificativa
- Funcionamento da memória cache
- Funções de mapeamento
- Algoritmos de substituição
- Política de escrita write-back
- Decisões de projeto para a implementação
 - Estruturas de dados utilizadas
 - Tamanho da memória cache
 - Escolha dos algoritmos de substituição
 - Dentre outras 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 7)
- Relatório (0 a 3)

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 e-mail até as **23:59** do dia **11/09/2016** para maurilio.campanojr@gmail.com

O assunto do e-mail deve **obrigatoriamente** ser “**AC/UNESPAR – TRABALHO**”

REFERÊNCIAS

Stallings, William; Arquitetura e organização de computadores; 8ª edição, São Paulo; Pearson Pratic 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.