

# ***SkepsisMem: Um Simulador Didático e Interativo de Hierarquia de Memória***

Eros M. de Carvalho<sup>1</sup>, Sérgio M. Dias<sup>2</sup>, Jefferson W. da Cunha<sup>3</sup>, Frederico A. F. S. Araújo<sup>4</sup>,  
Carlos A. P. S. Martins<sup>5</sup>.

*Pontifícia Universidade Católica de Minas Gerais  
Departamento de Ciência da Computação - DCC*

*Belo Horizonte - Minas Gerais, Avenida Dom José Gaspar, 500 - Brasil*

<sup>1</sup>erosmc@yahoo.com, <sup>2</sup>sergiomariano@gmail.com, <sup>3</sup>jwcunha@gmail.com, <sup>4</sup>fredericofonsec@gmail.com,  
<sup>5</sup>capsm@pucminas.br

## **Resumo**

*É fato que o processo de aprendizagem de hierarquia de memória tende a ser um tanto quanto árduo em função do alto grau de abstração exigido por parte daqueles que iniciam o estudo nesta área. Buscou-se, neste trabalho, propor e desenvolver um simulador didático e interativo de hierarquia de memória, denominado SkepsisMem, a fim de tornar o processo de aprendizagem mais fácil e interessante. Como resultado, alcançou-se uma ferramenta capaz de simular um conjunto de acessos interativamente nos três níveis tradicionais da hierarquia de memória e exibir de forma gráfica as taxas de acerto e erro nos níveis simulados. Com o objetivo de propor um ambiente didático e interativo de hierarquia de memória, buscou-se reunir as principais características encontradas nos simuladores de hierarquia de memória de natureza didática ou comercial, além de considerar alguns aspectos relevantes com relação ao processo de aprendizagem como um todo.*

## **1. Introdução**

Para uma melhor compreensão de um sistema de hierarquia de memória [1] e dos aspectos relevantes com relação à memória cache, é preciso uma melhor compreensão sobre tais conceitos. Entretanto, não basta um aprofundamento teórico desses conceitos, pois o aprendizado efetivo de um processo é uma tarefa mais complexa que um simples estudo teórico a respeito de um conceito. Para um efetivo processo de aprendizagem, é necessário também um efetivo processo de percepção que permita uma correta reflexão por parte do elemento humano envolvido na

atividade de aprendizagem; mas essa reflexão no estudo de hierarquia de memória é muito difícil, uma vez que, para tal, o aprendiz necessita de uma grande capacidade de abstração. Um outro agravante, segundo Martins [2], é o docente não conseguir exibir vários níveis de hierarquia de memória de modo claro usando os métodos convencionais, como quadros e slides.

Além disso, segundo Brunner [3], é necessário que, no processo de aprendizagem, haja uma preocupação com o contexto e as experiências do aprendiz para que o mesmo tenha vontade e seja capaz de aprender. Mas como levar em consideração a experiência do aprendiz, se muitas vezes a única fonte de aprendizado que o mesmo possui são modelos teóricos encontrados em livros, artigos e aulas expositivas apresentadas pelos docentes? Devido a essas dificuldades no aprendizado da hierarquia de memória, faz-se necessária a utilização de simuladores que auxiliem na aprendizagem.

Partindo da necessidade de simuladores didáticos de hierarquia de memória que auxiliem no processo de aprendizagem, e da análise dos disponíveis na literatura, é proposto e desenvolvido neste trabalho um simulador didático e interativo de hierarquia de memória denominado *SkepsisMem*.

O *SkepsisMem* foi desenvolvido para ser interativo de tal forma que facilite a compreensão dos princípios da hierarquia de memória e o fluxo dos acessos em seus três principais níveis: memória *cache*, memória principal e memória virtual. Com o objetivo de facilitar ainda mais o processo de aprendizagem, o usuário tem a opção de especificar quais níveis da hierarquia deseja simular. Assim, esse processo não precisará ser, em um primeiro momento, complexo ao ponto de simular os três níveis da hierarquia, nem tão

simples após já se ter um maior conhecimento da mesma e do funcionamento do simulador.

Este trabalho tem a seguinte estrutura: na segunda seção são apresentados alguns dos principais simuladores de hierarquia de memória existentes e suas principais características. Na terceira seção é apresentado o simulador desenvolvido e os resultados da verificação do simulador. Já na quarta seção é realizada uma simulação com a ferramenta proposta e, finalmente, na quinta seção, as conclusões deste trabalho são apresentadas.

## 2. Trabalhos Relacionados

Existem muitos trabalhos relacionados à simulação de hierarquia de memória. Dentre eles, podemos destacar os simuladores: DCMSim [4], Dinero [5], VMSim [6], MipSit [7] e MSCSim [8].

Os simuladores DCMSim, Dinero e MipSit simulam memória *cache* e principal. A diferença entre os três é que no simulador Dinero pode ser utilizado até 5 níveis de *cache*. No DCMSim pode ser configurada a arquitetura da memória *cache* em seus três tipos principais: mapeamento direto, associativa por conjunto e completamente associativa, enquanto o MipSit permite a simulação da memória *cache* e *pipeline*.

O VMSim é um simulador de memória virtual e principal; nele, os dados são escolhidos automaticamente, ou seja, não existe um modo de configuração. Já o simulador de hierarquia de memória MSCSim é um simulador de memória *cache* com split que permite simular múltiplos níveis de memória *cache*.

É importante que um simulador de hierarquia de memória ofereça a possibilidade de configuração de um ambiente, onde possam ser definidas as características do mesmo, por exemplo, quais níveis se deseja simular, *cache* e principal [4, 5, 6, 7], principal e virtual [6, 8], *cache*, principal e virtual [8]. Além disso, é interessante que se possa configurar a arquitetura da *cache* [4, 8], as políticas de substituição (FIFO, *First In First Out*, LRU, *Last Recently Used*, Random) [4, 8], as políticas de escrita (*Write Through*, *WriteBack*) [4, 8], as penalidades por falta [4, 5, 8] e os múltiplos níveis de *cache* [5, 8].

Além das características do ambiente é importante que haja uma interatividade, de tal forma que o usuário possa avançar e/ou retroceder um ou mais passos da execução da simulação do *memory trace*. De todos os simuladores mencionados nenhum possui todas as características possíveis de se explorar em um

sistema de hierarquia de memória, todavia, o simulador MSCSim [8] foi o que apresentou o maior conjunto de recursos disponíveis.

## 3. SkepsisMem

Baseado na análise do estado da arte foi proposto e desenvolvido o *SkepsisMem – Um Simulador Didático e interativo de hierarquia de memória* [9], o qual busca agregar as principais características encontradas nos simuladores disponíveis na literatura e incorporar a interatividade e análise dos resultados de forma estatística. O simulador *SkepsisMem* foi desenvolvido segundo o paradigma da orientação por objetos, e com o intuito de ser uma ferramenta de suporte ao entendimento da hierarquia de memória.

Um dos principais diferenciais, mas não o único, em relação aos simuladores encontrados na literatura é o fato de ser interativo (avançar e/ou retroceder um ou mais acessos do *memory trace*). Pode-se, também, ao final de todo processo, obter um resultado por meio de gráficos e tabelas. Eles exibem informações sobre as taxas de acerto da *TLB*, da *tabela de página* e da *cache* e os tempos de acesso dos níveis simulados da hierarquia de memória. Ao longo da simulação, pode-se observar uma animação que exhibe o nível hierárquico no qual um dado ou instrução está sendo buscado em um determinado momento. Esta animação auxilia o usuário a visualizar a ocorrência de *cache miss*, *TLB miss* e *page fault*.

### 3.1. O processo de simulação

Em uma simulação, o usuário é obrigado primeiramente a carregar um *memory trace*, onde cada linha contém as seguintes informações: o endereço acessado, o tipo de conteúdo, se dado ou instrução, o tipo de operação, se leitura ou escrita, e o valor. Em seguida, ele deverá definir uma configuração para a hierarquia de memória. Neste momento, alguns conhecimentos são avaliados a partir das validações que existem no próprio simulador, como, por exemplo, verificar se a quantidade de posições em uma memória *cache* associativa por conjunto tem um número de entradas que seja uma potência inteira de dois. A simulação não é iniciada se o projeto estiver configurado erradamente e o usuário é alertado sobre o seu erro.

Como o *memory trace* e o projeto são independentes, é possível alterar qualquer um dos dois, permitindo alguns usos interessantes do simulador. Mantendo-se um mesmo *memory trace*, o

usuário poderá, por exemplo, fazer várias simulações com projetos diferentes. Com estas alterações, será possível concluir qual configuração de hierarquia de memória apresenta um desempenho melhor para aquele *memory trace*. Por outro lado, mantendo um mesmo projeto, o usuário poderá observar como uma determinada configuração de hierarquia de memória se comporta diante da variação do *memory trace*. Com este procedimento, o usuário poderá averiguar se uma determinada configuração de hierarquia de memória mantém taxas de acerto mais ou menos homogêneas. A separação entre projeto e *memory trace* é essencial, pois auxilia o aluno na reflexão da arquitetura configurada no simulador.

É importante salientar também a riqueza de informações exibidas durante a simulação. Uma janela com duas abas apresenta informações sobre o estado da *cache*, quais *slots* estão ocupados, a *tag* armazenada e até mesmo os dados presentes na *cache*. Uma segunda janela apresenta, em abas distintas, o estado presente da *TLB* e da *Tabela de Página*. Por fim, em uma terceira janela, exibe-se o *memory trace*, indicando a tradução, quando for o caso, do endereço virtual para o real, o tipo de dado e o tipo de operação realizada. Em uma outra aba desta janela são mostrados os acertos e faltas causados pelo acesso de cada endereço do *memory trace*. A exibição de todas essas informações auxilia o usuário a perceber as mudanças nos estados dos três níveis da hierarquia de memória durante a simulação.

#### 4. Resultados da Verificação

Para uma simulação no *SkepsisMem*, primeiramente, é preciso realizar a configuração da hierarquia de memória, conforme citado na Seção 3.1. Na figura 1 é apresentada a configuração de uma hierarquia de memória.

Nesta verificação foi realizada uma simulação dos três níveis de memória com o tamanho das memórias principal e virtual sendo, respectivamente, 2048 e 4096 unidades, *cache* tipo mapeamento direto, tamanho da memória *cache* sendo 8 unidades, grau de associatividade sendo 1, tamanho do bloco sendo 1, política de escrita nas memórias *cache*, principal e virtual sendo *write through*, política de substituição nas memórias *cache*, principal e virtual sendo FIFO, tamanho da página virtual sendo 64 unidades, tamanho da *TLB* sendo 8 unidades, e latência das memórias *cache*, principal e virtual sendo, respectivamente, 1, 5 e 20 unidades de tempo. O

*memory trace* utilizando na simulação, foi gerado pelo simulador, composto por 10 instruções.

Na Figura 2 é apresentada a simulação para a configuração proposta. Nela pode se ver a memória *cache*, a *TLB*, o *memory trace*, uma animação indicando qual nível da hierarquia está sendo acessado e uma tabela com estatísticas de *TLB miss*, *page fault*, *cache miss*, tempos de acesso e outras informações.

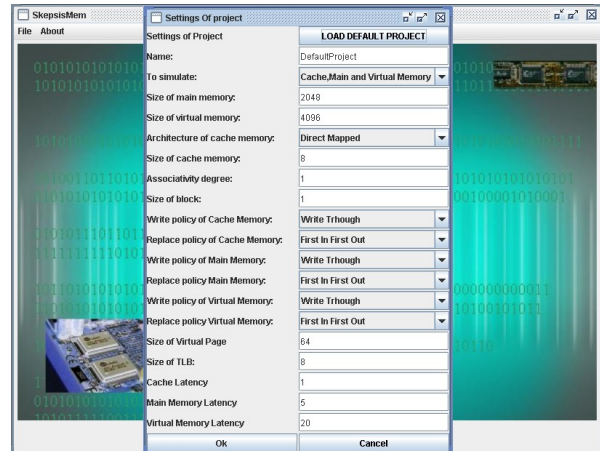


Figura 1. Configuração inicial.

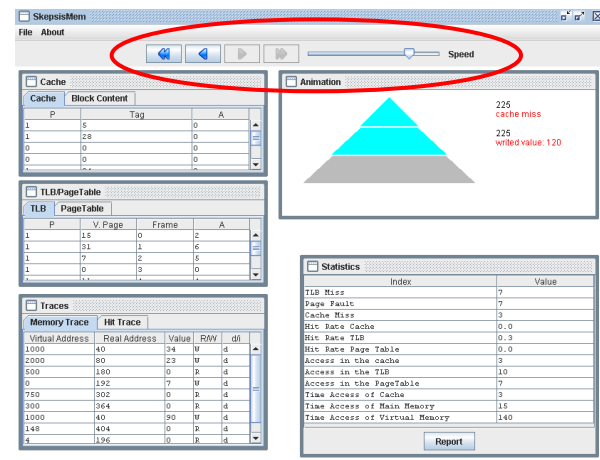


Figura 2. Simulação sendo realizada.

Como citado na Seção 3, em um simulador é importante que haja uma interatividade de tal modo que o usuário possa avançar e/ou retroceder um ou mais passos da execução do *memory trace*. Na Figura 2, mais exatamente na parte circulada, mostra como essa interatividade é realizada no simulador *SkepsisMem*. Com esse recurso o usuário pode interagir com o simulador adequando a velocidade da simulação ao seu ritmo de aprendizado e retrocedendo,

desfazendo cada uma das operações da *memory trace* já realizadas.

Ao final da simulação o simulador apresenta um relatório com informações da hierarquia de memória, informações estatísticas e um gráfico indicando o tempo de acesso em cada nível. Na Figura 3, o relatório para a simulação proposta é apresentado.

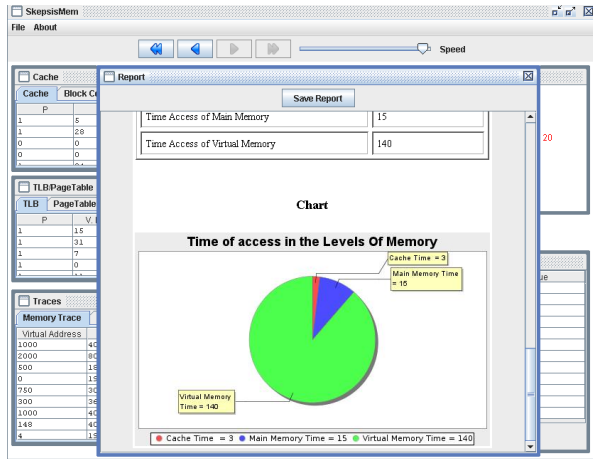


Figura 3. Resultado da simulação.

#### 4.1. Análise dos Resultados da Simulação

Para a simulação proposta, dentre os resultados obtidos, destaca-se o gráfico apresentado na Figura 3, onde é apresentado o tempo médio de acesso da memória *cache*, memória principal e memória virtual que foram de, respectivamente, 3, 15 e 140 unidades de tempo.

Com esse conjunto de recursos apresentados no simulador *SkepsisMem*, o entendimento da hierarquia de memória de um computador tende a ficar mais fácil e o processo de aprendizado menos árduo.

#### 5. Conclusões

Este trabalho teve como objetivo propor e desenvolver um simulador didático de hierarquia de memória, que reunisse as principais funcionalidades encontradas nos simuladores de mesmo propósito disponíveis na literatura. Acrescentar o aspecto da interatividade, recurso esse não disponível em outros simuladores, e possibilitar a visualização estatística dos resultados obtidos.

Na Tabela 1, as principais características dos simuladores encontrados na literatura e do simulador *SkepsisMem* são apresentadas. Pode-se notar que o objetivo desse trabalho foi parcialmente alcançado, visto que o *SkepsisMem* não é capaz de simular

múltiplos níveis de memória *cache*, recurso este que pretende-se adicionar em versões futuras.

Outra proposta de trabalho futuro é a expansão do simulador de tal forma que o mesmo torne-se um ambiente de aprendizagem não só de hierarquia de memória, mas de uma hierarquia computacional completa, ou seja, um ambiente que mostre o funcionamento de um dispositivo de processamento, de entrada, saída e da interação entre um sistema operacional e este.

Tabela 1. Simuladores e suas principais funcionalidades.

Simulador / Característica	SK [9]	MS [8]	DC [4]	DIN [5]	MIP [7]	VM [6]
<b>Configuração da hierarquia de memória</b>	X	X	X	X	X	
<b>Interatividade</b>	X		X			
<b>Animação</b>	X	X			X	X
<b>Relatório</b>	X	X	X	X	X	
<b>Documentação</b>	X	X				
<b>Níveis de Cache</b>		X		X		
<b>Cache+Principal</b>	X	X	X	X	X	
<b>Principal+Virtual</b>	X	X				X
<b>Cache+Principal+Virtual</b>	X	X				
<b>Configurar Políticas de Substituição</b>	X	X	X	X	X	

Além disso, pretende-se adicionar as funcionalidades da escolha do nível de detalhamento das simulações e que o próprio simulador seja capaz de propor modificações na configuração da hierarquia a fim de se obter um melhor desempenho para o *memory trace* em questão.

#### Referências

- [1] John L. Hennessy and David A. Patterson, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann, 3rd edition, 2003.
- [2] Martins, Carlos A. P. et al, "A New Learning Method of Microprocessor Architecture". In *32nd ASEE/IEEE Frontiers in Education Conference – FIE*, Boston, 2002, ppe. S1F16-S1F21.
- [3] Brunner, J., *Acts of Mining*. Cambridge, MA: Harvard University Press. 1990
- [4] Cordeiro, E. S., et al, "DCMSIM: Didactic Cache Memory Simulator", In *Proceedings of the Frontiers in Education Conference – FIE*, Boulder - Colorado, 2003, F1C14-F1C19
- [5] Rotithor, H.G., "On the effective use of a cache memory simulator in a computer architecture course", In *IEEE education*, Nov 1995, vol.38, no 4 pp.357-360
- [6] Jianfeng An, et al, "VMSIM: Virtual Machine Based a Full System Simulation Platform for Microprocessors' Functional Verification", In *Third International Conference on Information Technology: New Generations (ITNG'06)*, Vegas, Nevada, USA, 2006, pp. 245-249
- [7] M. Brorsson. "MipsIt-a simulation and development environment using animation for computer architecture education". In *Proceedings of 2002 Workshop on Computer Architecture Education*, Anchorage, Alaska, May 2002, pp. 65-72.
- [8] Coutinho, L. M. N.; Mendes, J. L. D.; Martins, C. A. P. S. "MSCSim – Multilevel and Split Cache Simulator", VI Workshop em *Sistemas Computacionais de Alto Desempenho (WSCAD)*, 2005, pp. 193-196
- [9] Skepsismem Home Page – <https://skepsismem.dev.java.net>