UNIVERSIDADE DO VALE DO ITAJAÍ
CURSO DE CIÊNCIAS DA COMPUTAÇÃO

Políticas e Medidas de Segurança em Bancos de Dados Relacionais: Protegendo Informações Sensíveis em Ambientes Corporativos

Autores
JALLISON ALFREDO JIMENEZ
Universidade do Vale de Itajaí
jallisonalfredo@edu.univali.br

MATHEUS VOLTOLINI
Universidade do Vale de Itajaí
Matheus_voltolini@edu.univali.br

ITAJAÍ 2025

Sumário

Resumo

Este trabalho apresenta o desenvolvimento de uma simulação em C++ para a tradução de endereços de memória virtual em sistemas operacionais. A implementação abrange conceitos fundamentais como a Tabela de Páginas (Page Table), o Buffer de Tradução de Endereços (Translation Lookaside Buffer - TLB) com política de substituição LRU (Least Recently Used), tratamento de faltas de página (page faults) e escrita de páginas sujas (dirty pages) de volta ao armazenamento secundário (backing store). A simulação permite a análise detalhada do comportamento do sistema de memória virtual, fornecendo métricas como acertos e faltas na TLB, acertos na tabela de páginas, número de faltas de página e escritas de páginas sujas. Os resultados obtidos demonstram a eficácia da implementação na compreensão dos mecanismos de gerenciamento de memória em sistemas operacionais.

Palavras-chave: Memória Virtual, Tradução de Endereços, Tabela de Páginas, TLB, LRU, Page Fault, Backing Store, C++.

1. Introdução

A memória virtual é uma técnica essencial em sistemas operacionais modernos, permitindo que programas utilizem mais memória do que a fisicamente disponível, através do mapeamento de endereços virtuais para endereços físicos. Este processo é gerenciado por estruturas como a Tabela de Páginas e o TLB, que auxiliam na tradução eficiente dos endereços e na manutenção da integridade e segurança do sistema.

Este trabalho tem como objetivo simular o processo de tradução de endereços de memória virtual, implementando os principais componentes envolvidos, como a Tabela de Páginas, o TLB com política LRU, tratamento de faltas de página e gerenciamento de páginas sujas. A simulação é desenvolvida em C++, proporcionando uma compreensão prática e detalhada do funcionamento desses mecanismos.

2. Fundamentação Teórica

2.1 Memória Virtual

A memória virtual permite que programas utilizem um espaço de endereçamento maior do que a memória física disponível, através do mapeamento de endereços virtuais para físicos. Isso é realizado pela Unidade de Gerenciamento de Memória (MMU), que utiliza estruturas como a Tabela de Páginas para gerenciar esse mapeamento .

2.2 Tabela de Páginas

A Tabela de Páginas é uma estrutura de dados que mantém o mapeamento entre páginas virtuais e molduras de páginas físicas. Cada entrada na tabela contém informações como o número da moldura física correspondente, bits de validação, acesso e modificação .

2.3 Translation Lookaside Buffer (TLB)

O TLB é uma memória cache associativa que armazena recentemente traduções de endereços virtuais para físicos, visando acelerar o processo de tradução e reduzir o número de acessos à Tabela de Páginas .

2.4 Política de Substituição LRU

A política LRU (Least Recently Used) é utilizada para gerenciar a substituição de entradas no TLB e de páginas na memória física. Ela substitui a entrada ou página que não foi utilizada há mais tempo, assumindo que páginas recentemente acessadas têm maior probabilidade de serem acessadas novamente em breve .

2.5 Page Fault e Backing Store

Uma falta de página (page fault) ocorre quando uma página referenciada não está presente na memória física. Nesse caso, o sistema operacional deve buscar a página no armazenamento secundário (backing store) e carregá-la na memória física, atualizando a Tabela de Páginas e o TLB conforme necessário .

3. Desenvolvimento

3.1 Estrutura do Código

O código implementa uma simulação de tradução de endereços de memória virtual, com suporte a endereços de 16 e 32 bits. As principais estruturas e componentes implementados incluem:

- PageTableEntry: Estrutura que representa uma entrada na Tabela de Páginas, contendo informações sobre validade, acesso, modificação e número da moldura física.
- TLB: Classe que implementa o Buffer de Tradução de Endereços com política de substituição LRU, utilizando uma combinação de unordered_map e deque para gerenciamento eficiente das entradas.
- Memória Física: Representada por um vetor bidimensional que simula os quadros de página disponíveis na memória física.
- Backing Store: Arquivo que simula o armazenamento secundário, utilizado para carregar páginas não presentes na memória física durante uma falta de página.

3.2 Processamento de Endereços

O programa processa endereços fornecidos via linha de comando ou arquivo, determinando se o endereço é de 16 ou 32 bits. Para cada endereço, o processo de tradução envolve:

1. Consulta ao TLB: Verifica se a tradução do endereço virtual para físico está presente no TLB.
2. Consulta à Tabela de Páginas: Em caso de falta no TLB, verifica se a página correspondente está presente na Tabela de Páginas.
3. Tratamento de Page Fault: Se a página não estiver presente na Tabela de Páginas, ocorre uma falta de página, e a página é carregada do backing store para a memória física.
4. Atualização do TLB e Tabela de Páginas: Após carregar a página, o TLB e a Tabela de Páginas são atualizados com as novas informações.
5. Leitura do Valor na Memória Física: O valor correspondente ao endereço físico é lido e exibido.

3.3 Métricas Coletadas

Durante a execução, o programa coleta as seguintes métricas:

- Total de endereços processados.
- Número de acertos e faltas no TLB.
- Número de acertos na Tabela de Páginas.
- Número de faltas de página.
- Número de páginas sujas escritas de volta ao backing store.

Essas métricas são registradas em um arquivo de relatório ao final da execução.

4. Resultados

A simulação foi executada com diferentes conjuntos de endereços e tamanhos de página, permitindo a análise do comportamento do sistema de memória virtual. Os resultados demonstraram a eficácia do TLB em reduzir o número de acessos à Tabela de Páginas e a importância da política de substituição LRU na manutenção da eficiência do sistema. Além disso, observou-se o impacto das faltas de página no desempenho geral e a necessidade de gerenciamento adequado das páginas sujas para garantir a integridade dos dados.

5. Considerações Finais

A implementação da simulação de tradução de endereços de memória virtual em C++ proporcionou uma compreensão aprofundada dos mecanismos envolvidos no gerenciamento de memória em sistemas operacionais. A utilização de estruturas como a Tabela de Páginas, o TLB com política LRU e o tratamento de faltas de página permitiu a análise detalhada do desempenho e eficiência do sistema. Este trabalho serve como base para estudos futuros na área de sistemas operacionais e gerenciamento de memória.

Referências

1. Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Sistemas Operacionais: Conceitos e Design*. LTC.

2. Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems*. Pearson.

3. Stallings, W. (2018). *Operating Systems: Internals and Design Principles*. Pearson.

4. Wikipedia. (2025). *Memória virtual*. Disponível em: https://pt.wikipedia.org/wiki/Mem%C3%B3ria_virtual

5. Wikipedia. (2025). *Translation lookaside buffer*. Disponível em: https://pt.wikipedia.org/wiki/Translation_lookaside_buffer

6. Universidade Federal de Pernambuco. (n.d.). *Memória Virtual - Sistemas Operacionais*. Disponível em: https://www.cin.ufpe.br/~if674cc/aulas/AulaInfraHW-MemoriaVirtual.pdf

7. GeeksforGeeks. (2025). *Page Replacement Algorithms in Operating Systems*. Disponível em: https://www.geeksforgeeks.org/page-replacement-algorithms-in-operating-systems/