UNIVERSIDADE DE FORTALEZA
CENTRO DE CIÊNCIAS TECNOLÓGICAS
CURSO: CIÊNCIA DA COMPUTAÇÃO

Simulador de Algoritmos de Substituição de Páginas

Autores: Gisele Costa Siqueira
Marcio Paulo de Almeida Lima

Palavras-chave: memória virtual. substituição de páginas. algoritmos. FIFO. LRU.

Resumo

Este trabalho apresenta o desenvolvimento de um simulador em linguagem Java para análise comparativa de algoritmos de substituição de páginas utilizados no gerenciamento de memória virtual. Foram implementados os algoritmos FIFO, LRU, Relógio e Ótimo, com o objetivo de avaliar o número de faltas de página em diferentes cenários de acesso à memória. O simulador permite inserção de sequências de referência, definição do número de molduras e exibição de resultados detalhados via console e interface gráfica em Swing. Os resultados obtidos permitem visualizar o desempenho relativo de cada algoritmo.

Introdução

O gerenciamento eficiente da memória virtual é essencial para o desempenho dos sistemas operacionais modernos. Quando um processo requer mais memória do que a disponível fisicamente, o sistema precisa decidir quais páginas manter em memória e quais substituir. Essa decisão é feita por algoritmos de substituição de páginas, que afetam diretamente o tempo de resposta e a taxa de faltas de página.

Entre os algoritmos clássicos estudados estão o FIFO (First In, First Out), LRU (Least Recently Used), Relógio (Clock) e Ótimo (Optimal). Cada um apresenta vantagens e limitações, variando em complexidade e eficiência. O presente trabalho tem como objetivo implementar e comparar esses algoritmos em um simulador, proporcionando uma visão prática sobre seus comportamentos.

Metodologia

O simulador foi desenvolvido na linguagem **Java**, utilizando **estruturas de dados apropriadas** para cada algoritmo:

- **FIFO:** fila (Queue) para armazenar as páginas na ordem de chegada.
- **LRU:** LinkedHashMap em modo de acesso para manter a ordem de uso recente.
- **Relógio:** vetor circular com bits de uso (booleanos) representando a “segunda chance”.

- **Ótimo:** busca antecipada do uso futuro das páginas para decidir a substituição.

O programa recebe como entrada uma sequência de números inteiros representando referências de páginas e o número de molduras disponíveis. Cada método calcula o total de faltas de página e apresenta os resultados em formato textual e gráfico (GUI Swing).

Resultados e Discussão

Durante os testes, foi utilizada a sequência de referência `7,0,1,2,0,3,0,4,2,3,0,3,2` com **3 molduras**. Os resultados obtidos foram os seguintes:

Algoritmo	Faltas de Página
FIFO	10
LRU	9
Relógio	9
Ótimo	7

Observa-se que o **algoritmo Ótimo** apresentou o menor número de faltas de página, como esperado, uma vez que ele realiza a substituição ideal, ou seja, remove a página que só será necessária mais adiante na sequência.

O **LRU (Least Recently Used)** e o **Relógio (Clock)** apresentaram desempenhos muito próximos, ambos com **9 faltas de página**, o que evidencia que o Relógio é uma boa aproximação do LRU, porém com menor custo computacional.

Já o **FIFO (First In, First Out)** foi o menos eficiente, com **10 faltas de página**, reforçando a limitação desse método em cenários onde páginas antigas voltam a ser utilizadas logo após a substituição.

O gráfico comparativo gerado pela interface Swing permite visualizar claramente essas diferenças, evidenciando a superioridade do algoritmo Ótimo e o equilíbrio entre desempenho e simplicidade do Relógio.

Conclusão

A execução do simulador permitiu comparar o comportamento prático dos principais algoritmos de substituição de páginas. Com a sequência de teste utilizada, observou-se que o **algoritmo Ótimo** apresentou o melhor desempenho, com apenas **7 faltas de página**, confirmando seu caráter teórico ideal.

Os algoritmos **LRU** e **Relógio** obtiveram resultados semelhantes, ambos com **9 faltas de página**, demonstrando que o Relógio é uma boa aproximação do LRU, porém com menor custo de implementação. Já o **FIFO**, embora simples, foi o menos eficiente, registrando **10 faltas de página** e evidenciando a possibilidade de **anomalia de Belady**, em que o aumento do número de

molduras não necessariamente reduz as faltas.

O simulador desenvolvido mostrou-se uma ferramenta eficaz para **visualizar e compreender** o impacto das diferentes políticas de substituição de páginas no desempenho de um sistema de memória virtual.

Referências

- SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G. **Fundamentos de Sistemas Operacionais.** 9. ed. Rio de Janeiro: LTC, 2014.
- TANENBAUM, A. S.; BOS, H. **Modern Operating Systems.** 4. ed. Pearson, 2015. STALLINGS, W. **Operating Systems: Internals and Design Principles.** 9. ed. Pearson, 2018.
- DEV MEDIA. *Introdução à Interface GUI no Java.* Disponível em: https://www.devmedia.com.br/introducao-a-interface-gui-no-java/25646.
- PRIMEFACES. *Framework JSF para aplicações web.* Disponível em: https://www.primefaces.org/.