POLÍTICAS DE SUBSTITUIÇÃO DE CACHE: FIFO E LRU SIMULADO EM JAVA

Paulo Roberto de Souza Carneiro Filho
Henrico Gramacho Oliveira
Lucca Dávila Bacelar
Vitor Reis dos Santos

1 INTRODUÇÃO

Com o aumento do volume de dados e a necessidade de um aumento em desempenho, habilidades e técnicas de gerenciamento de memória acabam por se tornar fundamentais. Considerando isso, destaca-se o uso de caches. Entretanto, considerando capacidades limitadas, acaba por se tornar necessário a aplicação e uso de políticas de substituição para apoiar na decisão de qual item será removido quando o cache é preenchido. Este trabalho mostra a implementação de duas políticas de substituição: FIFO (First In, First Out) e LRU (Least Recently Used), com base em um simulador desenvolvido em Java.

2 TEORIA

2.1 CACHE

Caches são espaços de armazenamento temporário rápido que guardam cópias de dados para que eles possam ser acessados mais rapidamente em solicitações. Quando atinge sua capacidade máxima, uma política de substituição determina qual elemento será removido para dar espaço a um novo.

2.2 POLÍTICA FIFO (FIRST IN, FIRST OUT)

A política FIFO remove do cache o item mais antigo, ou seja, o primeiro item adicionado. Essa abordagem é simples de implementar e exige apenas o controle de ordem de inserção. No entanto, ela pode apresentar problemas de desempenho, como o fenômeno de Belady, onde mais memória resulta em mais falhas de página.

2.3 POLÍTICA LRU (LEAST RECENTLY USED)

A política LRU é uma política de gerenciamento de cache que prioriza a remoção de dados menos recentemente utilizados quando o cache atinge sua capacidade máxima. Essa política se baseia no princípio de localidade temporal, assumindo que dados recentemente acessados provavelmente serão acessados novamente em breve. Embora mais eficiente que FIFO em muitos cenários, sua implementação é mais complexa, exigindo controle da ordem de acesso.

3 METODOLOGIA

O simulador foi desenvolvido em Java, utilizando Swing para a criação da interface gráfica. O programa faz com que o usuário tenha a possibilidade de selecionar entre as políticas FIFO e LRU, inserir inputs de números com a intenção de simular “acessos”, e observar o comportamento do cache. A política selecionada é alterável durante a execução, e o conteúdo do cache é exibido em tempo real, levando “wipe” sempre que a política for trocada e é possível finalizar o processo inserindo a palavra “sair”.

4 RESULTADOS

4.1 COMPARAÇÃO DE FUNCIONAMENTO

Tabela 1 – Comparação entre as políticas de substituição FIFO e LRU

Critério	FIFO	LRU
Critério de remoção	Mais antigo (inserido primeiro)	Menos recentemente utilizado
Eficiência (geral)	Menor	Maior, especialmente em acessos repetidos
Complexidade	Baixa	Moderada
Casos de mau desempenho	Belady	Pouco comum

4.2 EFICIÊNCIA COMPUTACIONAL

- FIFO: possui desempenho constante nas operações de inserção e remoção, mas não é eficiente no reaproveitamento de espaço.
- LRU: apesar de ter um custo mais alto, é reduzido significativamente o número de falhas em cenários reais.

5 CONCLUSÃO

Nosso trabalho apresentou a implementação de um simulador de políticas de substituição de cache, mostrando as diferenças práticas entre FIFO e LRU. Observou-se que, embora o FIFO seja mais simples, LRU tende a oferecer maior eficiência em ambientes com padrões repetitivos, justificando seu uso em sistemas mais modernos. A interface desenvolvida também serviu como boa ferramenta para experimentação e análise, além de facilitar o entendimento do programa.

REFERÊNCIAS

SILBERSCHATZ, Abraham; GALVIN, Peter Baer; GAGNE, Greg. Sistemas Operacionais: Conceitos. 9. ed. São Paulo: Pearson, 2013. (utilizado de guia e apoio para 2.1, 2.2, 2.3 e 4.3)

TANENBAUM, Andrew S. Organização Estruturada de Computadores. 6. ed. São Paulo: Pearson, 2014. (utilizado de guia e apoio para 2.1 a 2.3 e 4.1)

Java SE Documentation. Oracle. Disponível em: https://docs.oracle.com/javase/8/docs/. Acesso em: 01 jun. 2025. (utilizado de guia e apoio para 3)