

Relatório do Trabalho 1 de Sistemas Operacionais I
Maurício Konrath e Nicolas Elias

Simulação de Algoritmos de Substituição de Páginas

Florianópolis

2023-2

Resumo

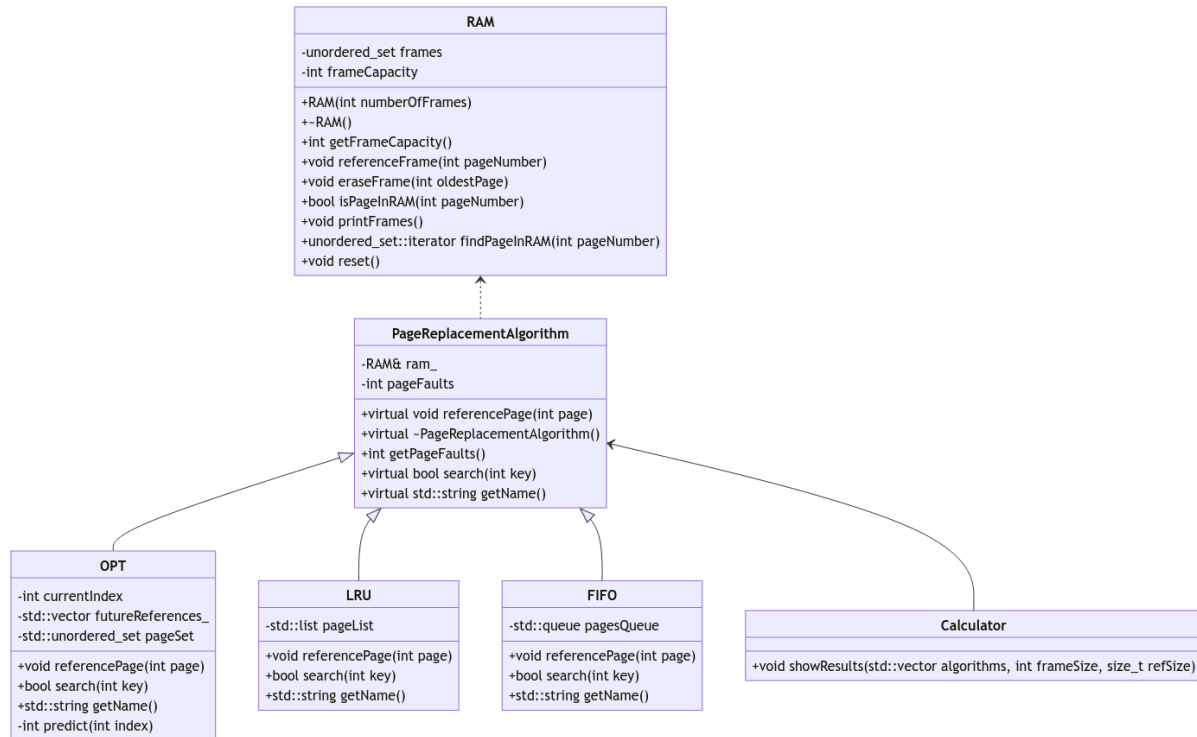
Este relatório documenta o desenvolvimento e os resultados do Trabalho 2 de Sistemas Operacionais 1. O objetivo deste trabalho era desenvolver os algoritmos de substituição de páginas vistos em aula em C++ e realizar uma análise de desempenho com base em métricas como o número de quadros e as referências usadas.

Introdução

O gerenciamento de memória é crucial em sistemas de computação, especialmente quando abordamos a substituição de páginas na RAM. Este relatório apresenta uma solução para simular três algoritmos amplamente utilizados para essa substituição: FIFO (First In, First Out), LRU (Least Recently Used) e OPT (Optimal).

Estrutura

Na imagem 1 pode-se observar o diagrama UML, produzido para o trabalho a fim de se ter mais clareza em relação ao código produzido



[\(clique aqui para abrir\)](#)

Imagem 1 - Diagrama UML

Arquitetura da Solução

Abstração da Memória RAM:

A primeira etapa para a solução foi criar uma abstração da memória RAM. Esta abstração é representada por uma classe que possui:

Frames: Estrutura hash onde as páginas são alocadas, proporcionando acesso rápido e eficiente às páginas e simulando o conceito de gerência de memória.

FrameCapacity: Define a quantidade total de molduras que podem ser alocadas na RAM.

Algoritmo de Substituição de Páginas:

Para possibilitar a simulação dos diferentes algoritmos de substituição, foi projetada uma classe abstrata chamada **PageReplacementAlgorithm**. Esta classe incorpora:

Método `referencePage`: Usado para tentar referenciar uma página na RAM.

Método `search`: Procura se uma página específica já está presente na RAM.

Qualquer algoritmo a ser implementado deve herdar dessa classe abstrata e moldar seus métodos conforme as especificidades de cada técnica.

Implementação dos Algoritmos:

FIFO: Como o nome indica, adota-se uma abordagem de fila. A página que entra primeiro é a primeira a sair. Este é o algoritmo mais simples dentre os três.

LRU: Com o uso da biblioteca `list` do C++, que já oferece uma lista encadeada, o LRU foi implementado de forma eficaz. Sempre que uma página é referenciada, ela é adicionada ao início da lista. Quando é necessário remover uma página, a última da lista (a menos recentemente usada) é selecionada.

OPT: Este algoritmo emprega duas estruturas principais. A primeira é um vector, que, por ser alocado de forma contígua na memória, oferece um percurso eficiente, especialmente útil no método `predict`. A segunda é a estrutura `hash` (similar aos `Frames`) que armazena referências, permitindo a remoção rápida de índices específicos.

Classe Calculator:

Há também uma classe denominada `calculator`, que tem a função de receber os objetos após a execução e formatar a saída conforme especificado no enunciado.

Execução

Foram elaborados testes levando em conta uma amostra de seis quadros (2, 3, 6, 10, 100 e 1000), todos foram executados utilizando os três arquivos de referência, disponibilizado pelo professor..

Cenário com 2 quadro e 1000000 referências:

Resultados:

FIFO: 370856 PFs

LRU: 308261 PFs

OPT: 264410 PFs

Tempo de execução:

FIFO = 44636.9 ms

LRU = 42214.7 ms

OPT = 42882.4 ms

Cenário com 3 quadro e 1000000 referências:

Resultados:

FIFO: 275457 PFs

LRU: 218498 PFs

OPT: 173436 PFs

Tempo de execução:

FIFO = 43123.2 ms

LRU = 42173.3 ms

OPT = 43854.6 ms

Cenário com 6 quadro e 1000000 referências:

Resultados:

FIFO: 177535 PFs

LRU: 143875 PFs

OPT: 100187 PFs

Tempo de execução:

FIFO = 38597.3 ms

LRU = 40096.3 ms

OPT = 39834.7 ms

Cenário com 10 quadro e 1000000 referências:

Resultados:

FIFO: 126890 PFs

LRU: 103959 PFs

OPT: 69746 PFs

Tempo de execução:

FIFO = 55899.9 ms

LRU = 46852.8 ms

OPT = 40612.1 ms

Cenário com 100 quadro e 1000000 referências:

Resultados:

FIFO: 29040 PFs

LRU: 22140 PFs

OPT: 12116 PFs

Tempo de execução:

FIFO = 43451 ms

LRU = 44897.9 ms

OPT = 52204.6 ms

Cenário com 1000 quadro e 1000000 referências:

Resultados:

FIFO: 1485 PFs

LRU: 1276 PFs

OPT: 1260 PFs
Tempo de execução:
FIFO = 46902.1 ms

Cenário com 2 quadro e 24 referências:

Resultados:
FIFO: 17 PFs
LRU: 17 PFs
OPT: 15 PFs
Tempo de execução:
FIFO = 0.002915 ms
LRU = 0.002104 ms
OPT = 0.002595 ms

Cenário com 3 quadro e 24 referências:

Resultados:
FIFO: 15 PFs
LRU: 14 PFs
OPT: 11 PFs
Tempo de execução:
FIFO = 0.002304 ms
LRU = 0.001724 ms
OPT = 0.002114 ms

Cenário com 6 quadro e 24 referências:

Resultados:
FIFO: 8 PFs
LRU: 8 PFs
OPT: 8 PFs
Tempo de execução:
FIFO = 0.002214 ms
LRU = 0.001593 ms
OPT = 0.001843 ms

LRU = 44006.1 ms
OPT = 43154.7 ms

Cenário com 10 quadro e 24 referências:

Resultados:
FIFO: 8 PFs
LRU: 8 PFs
OPT: 8 PFs
Tempo de execução:
FIFO = 0.002695 ms
LRU = 0.002064 ms
OPT = 0.001483 ms

Cenário com 100 quadro e 24 referências:

Resultados:
FIFO: 8 PFs
LRU: 8 PFs
OPT: 8 PFs
Tempo de execução:
FIFO = 0.002044 ms
LRU = 0.001573 ms
OPT = 0.001082 ms

Cenário com 1000 quadro e 24 referências:

Resultados:
FIFO: 8 PFs
LRU: 8 PFs
OPT: 8 PFs
Tempo de execução:
FIFO = 0.002124 ms
LRU = 0.001623 ms
OPT = 0.001102 ms

Cenário com 2 quadro e 30 referências:

Resultados:

FIFO: 30 PFs

LRU: 30 PFs

OPT: 24 PFs

Tempo de execução:

FIFO = 0.002615 ms

LRU = 0.001774 ms

OPT = 0.002905 ms

Cenário com 3 quadro e 30 referências:

Resultados:

FIFO: 30 PFs

LRU: 30 PFs

OPT: 19 PFs

Tempo de execução:

FIFO = 0.002816 ms

LRU = 0.002124 ms

OPT = 0.003146 ms

Cenário com 6 quadro e 30 referências:

Resultados:

FIFO: 6 PFs

LRU: 6 PFs

OPT: 6 PFs

Tempo de execução:

FIFO = 0.002295 ms

LRU = 0.001793 ms

OPT = 0.001062 ms

Cenário com 10 quadro e 30 referências:

Resultados:

FIFO: 6 PFs

LRU: 6 PFs

OPT: 6 PFs

Tempo de execução:

FIFO = 0.002034 ms

LRU = 0.001763 ms

OPT = 0.001062 ms

Cenário com 100 quadro e 30 referências:

Resultados:

FIFO: 6 PFs

LRU: 6 PFs

OPT: 6 PFs

Tempo de execução:

FIFO = 0.001844 ms

LRU = 0.001463 ms

OPT = 0.001312 ms

Cenário com 1000 quadro e 30 referências:

Resultados:

FIFO: 6 PFs

LRU: 6 PFs

OPT: 6 PFs

Tempo de execução:

FIFO = 0.002445 ms

LRU = 0.001733 ms

OPT = 0.001422 ms

Interpretação dos Resultados Obtidos

Eficiência dos Algoritmos (medida em Page Faults - PFs):

Cenários com 1.000.000 de referências:

Com um baixo número de quadros (2 e 3), o OPT supera os outros dois algoritmos em termos de PFs.

À medida que o número de quadros aumenta (6, 10, 100, 1000), a diferença em PFs entre os algoritmos se torna menos pronunciada, mas o OPT ainda se mantém como o mais eficiente, seguido pelo LRU e, finalmente, pelo FIFO.

Cenários com 24 referências:

Todos os algoritmos apresentam um desempenho muito semelhante quando há um alto número de quadros (6, 10, 100, 1000).

Com um baixo número de quadros (2 e 3), o OPT ainda se mostra superior.

Cenários com 30 referências:

Em cenários com 6, 10, 100 e 1000 quadros, todos os algoritmos têm o mesmo número de PFs.

Com apenas 2 e 3 quadros, o OPT demonstra claramente sua superioridade, tendo menos PFs que os outros dois algoritmos.

Tempo de Execução:

Em geral, os tempos de execução não variam drasticamente entre os algoritmos.

Para 1.000.000 de referências, o FIFO e o LRU tendem a ser mais rápidos no início, mas o OPT se torna mais rápido com mais quadros.

Em cenários com 24 e 30 referências, os tempos de execução são muito próximos, com variações mínimas.

Conclusões Gerais:

OPT: Consistentemente, o algoritmo OPT apresenta o menor número de PFs em quase todos os cenários, tornando-o o algoritmo mais eficiente em termos de substituição de páginas. No entanto, seu tempo de execução varia dependendo do número de quadros e referências.

LRU: O LRU geralmente fica no meio em termos de PFs, sendo mais eficiente que o FIFO mas menos do que o OPT. Seu tempo de execução é bastante competitivo em comparação com os outros dois algoritmos.

FIFO: Este é o algoritmo menos eficiente em termos de PFs em quase todos os cenários. Em termos de tempo de execução, ele tende a ser mais rápido com um menor número de quadros, mas essa vantagem diminui à medida que o número de quadros aumenta.

Implicações Práticas:

Se a métrica principal for minimizar os PFs, o OPT é claramente o melhor algoritmo a ser usado. No entanto, o OPT exige um conhecimento prévio das referências de página, o que pode não ser prático em muitos cenários do mundo real. O LRU, por outro lado, oferece um bom equilíbrio entre eficiência e tempo de execução sem a necessidade de conhecimento prévio, tornando-o uma escolha preferida em muitos sistemas operacionais. O FIFO, sendo o mais simples, pode ser útil em situações onde a sobrecarga de implementação precisa ser minimizada, mas sua eficiência é geralmente inferior aos outros dois algoritmos.

Conclusão

Através da implementação e simulação desses algoritmos, é possível obter insights valiosos sobre seu desempenho e eficácia em diferentes cenários. A modularização e abstração adotadas garantem uma solução flexível e expansível para futuras melhorias ou adições de novos algoritmos. Foi notado também que ao utilizarmos quadros superiores à quantidade de referências, o valor do desempenho se mantém constante.

Referências

Material disponibilizado pelos professores.