

RELATÓRIO TÉCNICO - Projeto de Sistemas Operacionais

Grupo:

- Cheuk Ki Yu - 10419664
- Guilherme Rainho Geraldo - 10418251
- Hugo Rafael Weng - 10417866

Simulador de Paginação de Memória

1. Introdução

O gerenciamento de memória é uma função crítica em sistemas operacionais modernos, responsável por controlar como os processos acessam a memória física por meio da paginação. O objetivo deste projeto foi desenvolver um simulador de paginação que implementa dois algoritmos clássicos de substituição de páginas: FIFO (First-In-First-Out) e LRU (Least Recently Used). O simulador processa acessos a páginas provenientes de múltiplos processos, gerenciando a memória física limitada, e coleta estatísticas para análise comparativa do desempenho dos algoritmos.

2. Descrição da Implementação

2.1 Estruturas de dados utilizadas:

- Frame:** Representa um quadro na memória física, contendo informações sobre qual processo (pid), qual página está armazenada, endereço inicial e um indicador de validade.
- FIFOQueue:** Implementa uma fila simples para gerenciar a ordem dos frames alocados no algoritmo FIFO.
- LRUTracker:** Mantém um vetor com o contador de uso mais recente para cada frame, permitindo identificar o frame menos recentemente utilizado.
- Processo:** Representa um processo em execução, com sua tabela de páginas armazenando para cada página o índice do frame correspondente na memória física, ou -1 se não estiver alocada.

2.2 Algoritmos Implementados

- FIFO (First-In-First-Out):** O algoritmo mantém uma fila dos frames alocados e substitui sempre o frame que está há mais tempo na memória (o que entrou primeiro).

- ii. LRU (Least Recently Used): O algoritmo acompanha o último acesso de cada frame e escolhe para substituição o frame que foi acessado há mais tempo, baseado em contadores de uso atualizados a cada acesso.

2.3 Decisões de Projeto

- i. O tamanho da página foi fixado em 4 KB, com memória física de 16 KB total, resultando em 4 frames.
- ii. O simulador suporta até 10 processos e 100 páginas por processo, com tabelas de páginas simples armazenando mapeamento frame-página.
- iii. O arquivo de entrada é um CSV com linhas no formato pid,page,address representando acessos virtuais a endereços.
- iv. Para visualização, o estado da memória física é impresso após cada acesso.
- v. Para facilitar a simulação temporal, foi implementada uma pausa de 0,3 segundos entre cada acesso.
- vi. O código inclui verificação de erros, como limite máximo de processos e formatação incorreta das linhas de entrada.

2.4 Limitações da Implementação

- i. O número de frames é muito pequeno (4 frames), o que pode não refletir cenários reais.
- ii. A simulação não considera troca para disco nem gerenciamento de escrita/leitura diferenciados.
- iii. A tabela de páginas é fixa e limitada, e não há implementação de segmentação.
- iv. O simulador depende do arquivo arquivos.csv para os acessos, não possui interface para entradas dinâmicas.
- v. Não há tratamento avançado de processos concorrentes ou tempos reais de execução.

3. Análise Comparativa dos Algoritmos

3.1 Comparação entre os algoritmos implementados

- i. FIFO: Fácil de implementar e com baixo custo computacional, mas pode apresentar baixa eficiência, pois não considera a frequência ou recência de uso das páginas.
- ii. LRU: Geralmente oferece melhor desempenho porque evita substituir páginas que foram usadas recentemente, mas requer manutenção de estruturas adicionais para acompanhar o uso.

3.2 Análise dos resultados obtidos

- i. A simulação exibe o total de acessos, falhas de página (page faults), acertos (hits) e as respectivas taxas percentuais para cada algoritmo.
- ii. Espera-se que o LRU apresente menor taxa de page faults, refletindo maior eficiência na utilização da memória.
- iii. A escolha do algoritmo impacta diretamente a quantidade de trocas de página, influenciando o desempenho do sistema simulado.

4. Conclusões

O projeto permitiu a implementação e o entendimento prático dos conceitos de paginação e substituição de páginas em sistemas operacionais. A experiência mostrou as diferenças claras entre algoritmos simples como FIFO e mais sofisticados como LRU, evidenciando os trade-offs entre simplicidade e eficiência. Além disso, o desenvolvimento do simulador reforçou conhecimentos em estruturas de dados, manipulação de arquivos, e lógica de controle para simulação temporal. A limitação de recursos e a necessidade de manutenção das tabelas de páginas e contadores ressaltam desafios reais no gerenciamento de memória. Para trabalhos futuros, seria interessante incluir algoritmos adicionais, simular mais processos, implementar escrita para disco, e criar visualizações gráficas para facilitar a análise comparativa.