UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Simulador de Paginação de Memória

Sistemas Operacionais

GRUPO:

ANDREY BEZERRA VIRGINIO DOS SANTOS
RA: 10420696
IGOR SILVA ARAUJO
RA: 10428505
JULIA VITORIA BOMFIM DO NASCIMENTO
RA: 10425604
WILLIAM SARAN DOS SANTOS JUNIOR
RA: 10420128

Sumário:

1. Introdução..... 3
2. Descrição da Implementação..... 3
3. Análise Comparativa dos algoritmos..... 3
4. Conclusões..... 4

1. Introdução

Imagine um hotel com muitos quartos, mas com espaço limitado: quando todos querem entrar, alguém precisa sair para dar lugar. Bem-vindo ao mundo da paginação de memória, onde processos são como hóspedes e a memória física é um prédio com número limitado de quartos!

A paginação é uma técnica essencial dos sistemas operacionais modernos, permitindo alocar memória de forma inteligente, mesmo quando o espaço é apertado. Ela evita a tão temida fragmentação externa, organiza o caos e ainda promove o compartilhamento de recursos entre diferentes processos.

Neste projeto, tivemos a oportunidade de se tornar o "gerente" dessa memória, implementando um simulador de paginação. Os algoritmos utilizados para essa experiência foram o LRU e o FIFO.

2. Descrição da Implementação

Implementamos as estruturas de cada elemento da seguinte forma:

Página

- Valor booleano representa se está ou não na memória
- Valor booleano que representa se está sendo referenciada ou não
- Momento do último acesso
- Número do frame que está alocada, caso esteja na memória
- Valor que representam o processo que a página pertence
- Número da página

Processo

- PID (Process IDentifier)
- Tabela de páginas
- Número de páginas

Memória

- Array de frames para alocar a memória
- Número total de frames
- Número de frames ocupados
- Número de endereços por frame

Ao executar o programa, a memória é inicializada sem nenhuma página alocada. Após inicializar o contexto de memória, um processo é selecionado aleatoriamente para realizar a solicitação de um endereço virtual. O endereço então é convertido, e, se a página não estiver presente na memória, é feita a troca seguindo a política do algoritmo que está sendo utilizado no momento.

3. Análise Comparativa dos algoritmos

Na implementação do simulador de paginação, foram utilizados dois algoritmos de substituição de páginas: LRU (Least Recently Used) e FIFO (First-In, First-Out).

O algoritmo LRU tem como princípio remover da memória a página que ficou há mais tempo sem ser acessada. A principal desvantagem do LRU é o controle e atualização constantes das informações sobre o último acesso de cada página, o que implica maior uso dos recursos.

Na simulação realizada, o LRU demonstrou um desempenho maior, apresentando uma taxa de page faults menor em comparação com o FIFO. Isso acontece por conta da sua capacidade de manter em memória as páginas que estão sendo mais utilizadas pelos processos, reduzindo a necessidade de substituições desnecessárias.

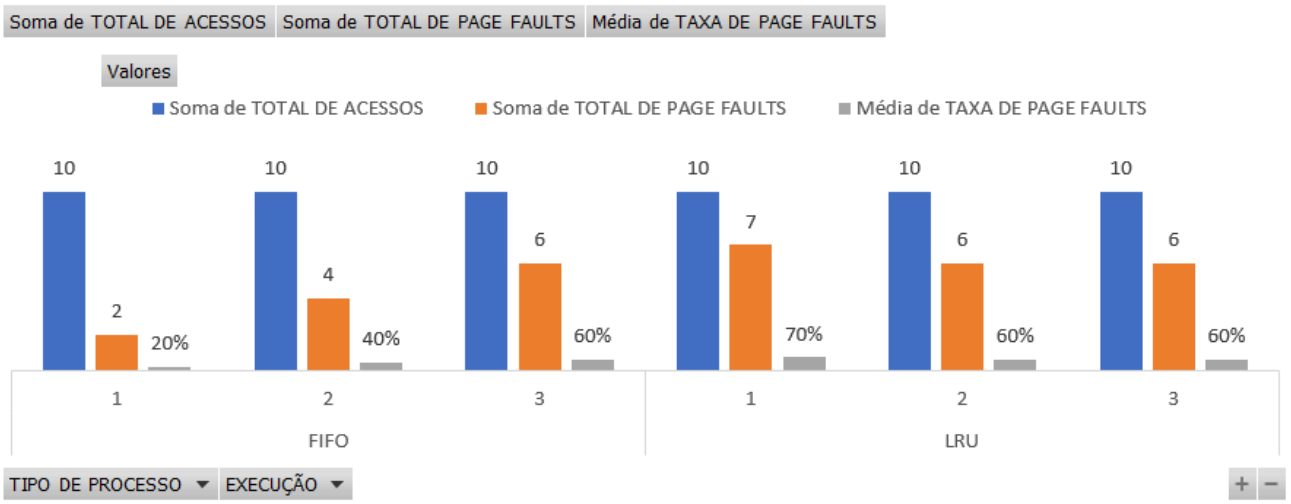
Por outro lado, o algoritmo FIFO opta por remover sempre a página que foi alocada há mais tempo, sem considerar se ela ainda está sendo utilizada pelos processos. Sua principal vantagem está na simplicidade de implementação. Contudo, essa simplicidade é a sua maior limitação: o FIFO, pode remover páginas que ainda são frequentemente acessadas, aumentando assim a taxa de page faults.

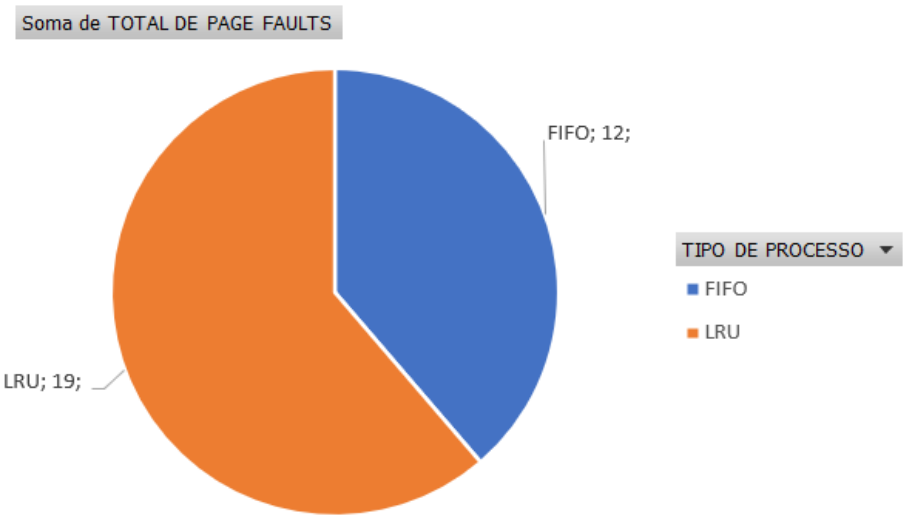
Durante a simulação, o FIFO apresentou uma taxa de page faults mais elevada do que o LRU, principalmente pela sua incapacidade de identificar e preservar as páginas mais utilizadas em memória.

Em termos gerais, a comparação entre os dois algoritmos evidenciou que o LRU, apesar de mais complexo, oferece melhores resultados em termos de desempenho, ao passo que o FIFO, embora mais simples, compromete a eficiência do sistema sob certas condições de carga.

Análise visual dos Algoritmos FIFO e LRU:

Rótulos de Linha	Soma de TOTAL DE ACESSOS	Soma de TOTAL DE PAGE FAULTS	Média de TAXA DE PAGE FAULTS
FIFO	**30**	**12**	**40%**
1	10	2	20%
2	10	4	40%
3	10	6	60%
LRU	**30**	**19**	**63%**
1	10	7	70%
2	10	6	60%
3	10	6	60%
Total Geral	**60**	**31**	**52%**





4. Conclusões

Este projeto implementou com sucesso um simulador de paginação, demonstrando na prática como os algoritmos LRU e FIFO gerenciam a memória. O LRU se mostrou mais eficiente, evitando mais page faults, mas com complexidade maior. Já o FIFO foi mais simples de implementar, porém menos inteligente em alguns casos. As principais dificuldades foram o gerenciamento preciso da memória e a simulação realista dos acessos. Para melhorias, poderiam ser testados outros algoritmos como Clock ou versões otimizadas do LRU. No geral, o trabalho comprovou que a escolha do algoritmo de substituição impacta diretamente no desempenho, sendo importante avaliar o trade-off entre eficiência e complexidade em cada cenário.