Disciplina: Sistemas Operacionais

Atividade Prática: Paginação

Objetivo

O objetivo deste projeto é escrever um programa para simular o comportamento dos algoritmos clássicos de substituição de páginas na gerência de memória de um sistema operacional. O programa deverá permitir a execução e comparação dos seguintes algoritmos de substituição:

- FIFO (First-In, First-Out)
- LRU (Least Recently Used)
- Ótimo
- Segunda Chance
- Clock
- NRU (Not Recently Used)
- LFU (Least Frequently Used)
- MFU (Most Frequently Used)

Ao final desta atividade, o aluno deverá ser capaz de:

- Compreender o funcionamento dos algoritmos de substituição de páginas;
- Entender o impacto do número de molduras de página (frames) no desempenho da memória virtual;
- Calcular métricas como número de faltas de página, taxa de faltas, evicções e escritas em disco;
- Analisar experimentalmente a anomalia de Belady e o efeito da localidade de referência.

Diretrizes para Implementação

O simulador deverá ler uma cadeia de referências (arquivo .trace) e simular a execução de cada algoritmo conforme o número de molduras informado. O programa pode ser desenvolvido em C, C++ ou Python e deve seguir a seguinte interface de execução:

```
./pager --algo <ALGO> --frames <N> --trace <arquivo> [--verbose]
```

Exemplo de uso:

```
./pager --algo lru --frames 3 --trace silberschatz2001.trace
```

Cadeia de Referências Oficial

Será utilizada a seguinte sequência de acessos fictícia, obtida de Silberschatz et al. (2001), com 20 acessos a 6 páginas distintas:

```
7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1
```

O arquivo de entrada deve conter uma referência por linha, conforme o exemplo abaixo, e ser nomeado silberschatz2001.trace:

```
7
0
1
2
0
3
0
4
2
3
0
3
2
1
2
0
1
7
0
1
```

Saída Esperada do Simulador

```
Algoritmo: LRU
Frames: 3
Referências: 20
Faltas de página: 12
Taxa de faltas: 60.00%
Evicções: 12
Conjunto residente final:
frame_ids:   0   1   2
page_ids:    1   7   0
```

O simulador deverá ser executado para todos os algoritmos implementados com o traço silberschatz2001.trace, considerando dois tamanhos de memória física: 3 e 4 molduras.

Resultados esperados:

Molduras	FIFO	LRU	Ótimo
3	15	12	9
4	10	8	8

Métricas de Desempenho a Calcular

- Número total de faltas de página
- Taxa de faltas (%)
- Evicções (quantas páginas foram removidas)X

Pontos importantes:

- O simulador pode ser escrito em C, Java, Javascript ou Python
- Cada equipe deve fornecer o código devidamente comentado para análise, juntamente com um documento explicando as decisões de implementação utilizadas (classes, estruturas de dados utilizadas, padrões de projeto (se for o caso). Sugere-se que cada bloco possua alguma estrutura que mapeie informações para controle, como id e tamanho. Essa estrutura deve estar descrita no documento