

## Modelo de Relatório Técnico – Sistemas Operacionais

Acadêmicos: Gabriel Tomazini Marani, Pedro Augusto Senger Piana

Data: 08/02/2025

## Sumário

1 Introdução	1
2 Materiais e Métodos	1
3 Resultados e Discussão	2
4 Referências Bibliográficas	3

## 1 Introdução

A gerência de memória é um aspecto essencial dos sistemas operacionais modernos, sendo responsável por mapear endereços lógicos para físicos e gerenciar a alocação de páginas na memória. Neste contexto, o uso de políticas de substituição de páginas influencia diretamente o desempenho do sistema. Este trabalho visa implementar um simulador de operações da Memory Management Unit (MMU), empregando as políticas de substituição de páginas LRU (Least Recently Used) e Segunda Chance, comparando sua eficácia com base na taxa de falhas de página e na eficiência do Translation Lookaside Buffer (TLB).

## 2 Materiais e Métodos

Para a implementação do simulador, utilizamos:

- **Linguagem de programação:** C
- **Arquitetura:** Paginação de um nível
- **Memória:** 64 frames físicos
- **Tamanho da página:** 4 KB
- **Formato do trace:** Arquivo contendo endereços lógicos em hexadecimal e tipo de acesso (leitura ou escrita)
- **Estruturas principais:**

- **Tabela de páginas** com entradas contendo número de página, número de frame, bit de validade e contador LRU/referência
- **TLB** com 16 entradas
- **Políticas de substituição:** LRU e Segunda Chance

O simulador segue um fluxograma definido para a resolução de endereços, considerando os acessos à TLB e a tabela de páginas, contabilizando falhas, hits e misses da TLB. As políticas de substituição de páginas são utilizadas quando ocorre uma falha de página, as devidas estruturas de dados foram implementadas para cumprir com a simulação.

O código pode ser acessado através do [github](#).

### 3 Resultados e Discussão

A execução do simulador resultou nos seguintes dados:

Arquivo:	swim.trace		
Método Utilizado	Page Faults	TLB Hits	TLB Misses
LRU	2543	783413	216587
Segunda Chance	2602	783366	216634

Arquivo:	sixpack.trace		
Método Utilizado	Page Faults	TLB Hits	TLB Misses
LRU	3953	857790	142210
Segunda Chance	3953	857790	142210

Arquivo:	gcc.trace		
Método Utilizado	Page Faults	TLB Hits	TLB Misses
LRU	2872	860139	139861
Segunda Chance	2885	860126	139874

Arquivo:	bzip.trace		
Método Utilizado	Page Faults	TLB Hits	TLB Misses

<b>LRU</b>	318	996107	3893
<b>Segunda Chance</b>	329	996100	3900

Os resultados indicam que a política LRU teve uma leve vantagem sobre a Segunda Chance, apresentando um menor número de falhas de página. Em relação à eficiência da TLB, os valores de hits e *misses* são muito semelhantes entre ambas as políticas, sugerindo que o impacto da política de substituição de páginas na TLB é mínimo.

A LRU se mostrou ligeiramente mais eficiente porque prioriza a remoção das páginas menos utilizadas recentemente, enquanto a Segunda Chance pode manter páginas antigas na memória se seu bit de referência for ativado. Entretanto, a Segunda Chance pode ser mais eficiente em cenários onde os acessos são mais distribuídos entre diversas páginas.