Virtual Memory Manager Simulator: Implementation and Evaluation

Choudari Harshitha Reddy (CS22BT015)
Mubarakpur Keerthi(CS22BT032)
N.Soumya (CS22BT036)

November 11, 2024

1 Introduction

This report details the development and assessment of a Virtual Memory Manager (VMM) simulator. The simulator emulates the functionality of virtual memory systems and employs various page replacement policies to manage memory allocation for multiple processes. The primary objectives of this assignment include:

- Simulating virtual memory management by implementing different page replacement strategies.
- Examining how page size and memory frame allocation influence the number of page faults.
- Evaluating the effectiveness of distinct allocation policies, specifically Global and Local.

The simulator is designed to accept command-line inputs and operates by analyzing memory access traces collected from real-world applications.

2 Simulator Design

2.1 Page Table Structure

The PageTable class was designed using a Hashed Page Table approach to store mappings between virtual and physical addresses efficiently. Four instances of PageTable were created, representing the four processes in the trace files and will be changed acc to requirement

2.2 Frame Status Structure

The FrameStatus class manages the status of memory frames, tracking allocated and free frames, and supports frame replacement based on the policy selected.

2.3 Page Replacement Policies

The following page replacement policies were implemented and tested:

- **Optimal**: Replaces the page that will not be used for the longest period.
- **FIFO**: Replaces pages in the order they were loaded.
- **LRU**: Replaces the least recently used page.
- **Random**: Randomly selects a page to replace.

2.4 Allocation Policies

The two allocation policies implemented are:

- **Global**: All processes share a common pool of memory frames.
- **Local**: Each process is allocated a fixed number of memory frames. In the case of Local, the "equal allocation" policy is used, where each process is given an equal share of available frames.

3 Results and Analysis

3.1 Results Table

The following table summarizes the number of page faults for various configurations of the simulator:

Configuration		**Page Faults**		**Memory Frames**
Replacement Policy	**Page Size**	**Global**	**Local**	
Optimal	4 KB	154789	218614	4
FIFO	4 KB	244462	218614	4
LRU	4 KB	236409	218614	4
Random	4 KB	246771	218614	4
Optimal	4 KB	35847	49361	16
FIFO	4 KB	84713	88163	16
LRU	4 KB	61595	69547	16
Random	4 KB	88869	85968	16
Optimal	4 KB	17019	21573	32
FIFO	4 KB	40227	40195	32
LRU	4 KB	29206	36410	32
Random	4 KB	43093	41731	32
Optimal	4 KB	8491	10553	64
FIFO	4 KB	19411	17444	64
LRU	4 KB	14204	14939	64
Random	4 KB	20051	18275	64

Table 1: Summary of Page Faults for Different Configurations (Global vs Local)

3.2 Graphical Representation of Page Faults

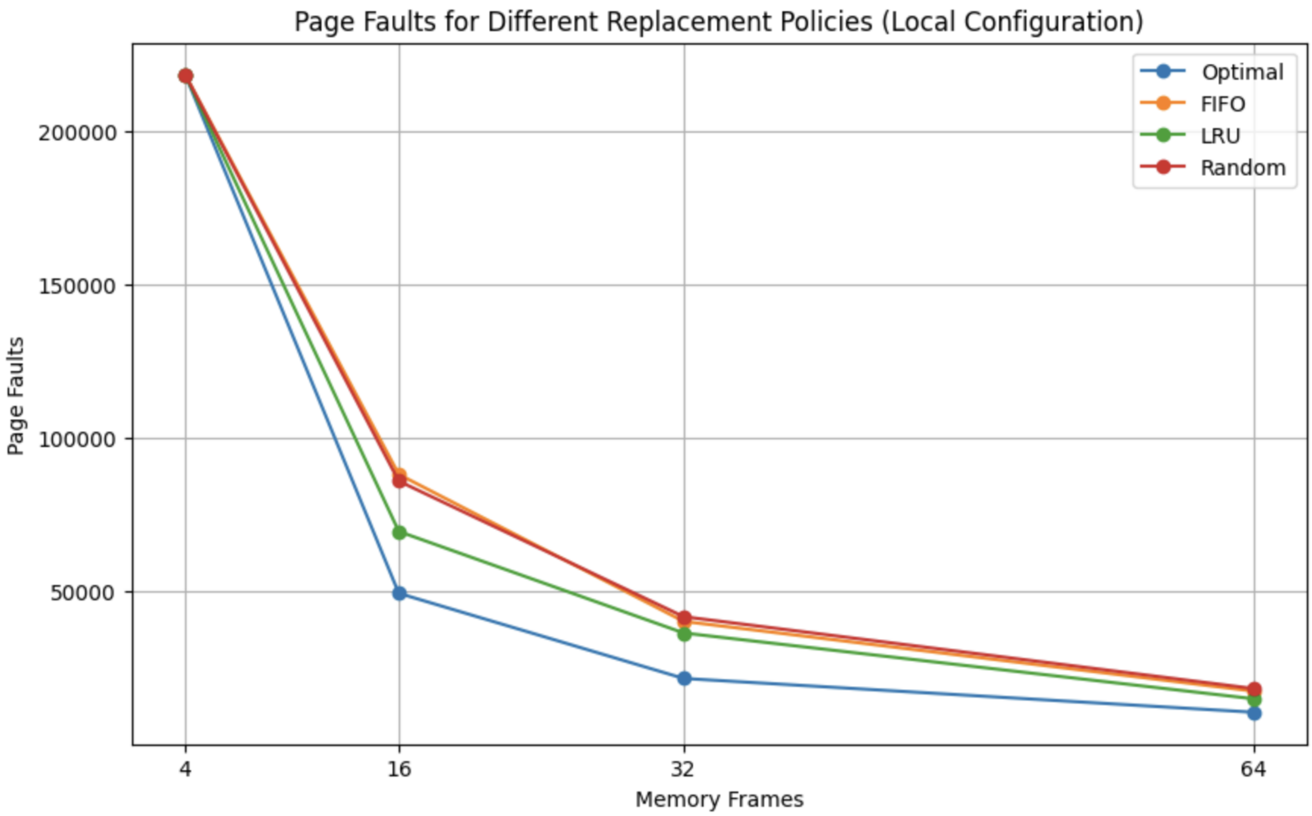

Figure 1: Page Faults for Different Replacement Policies [LOCAL]

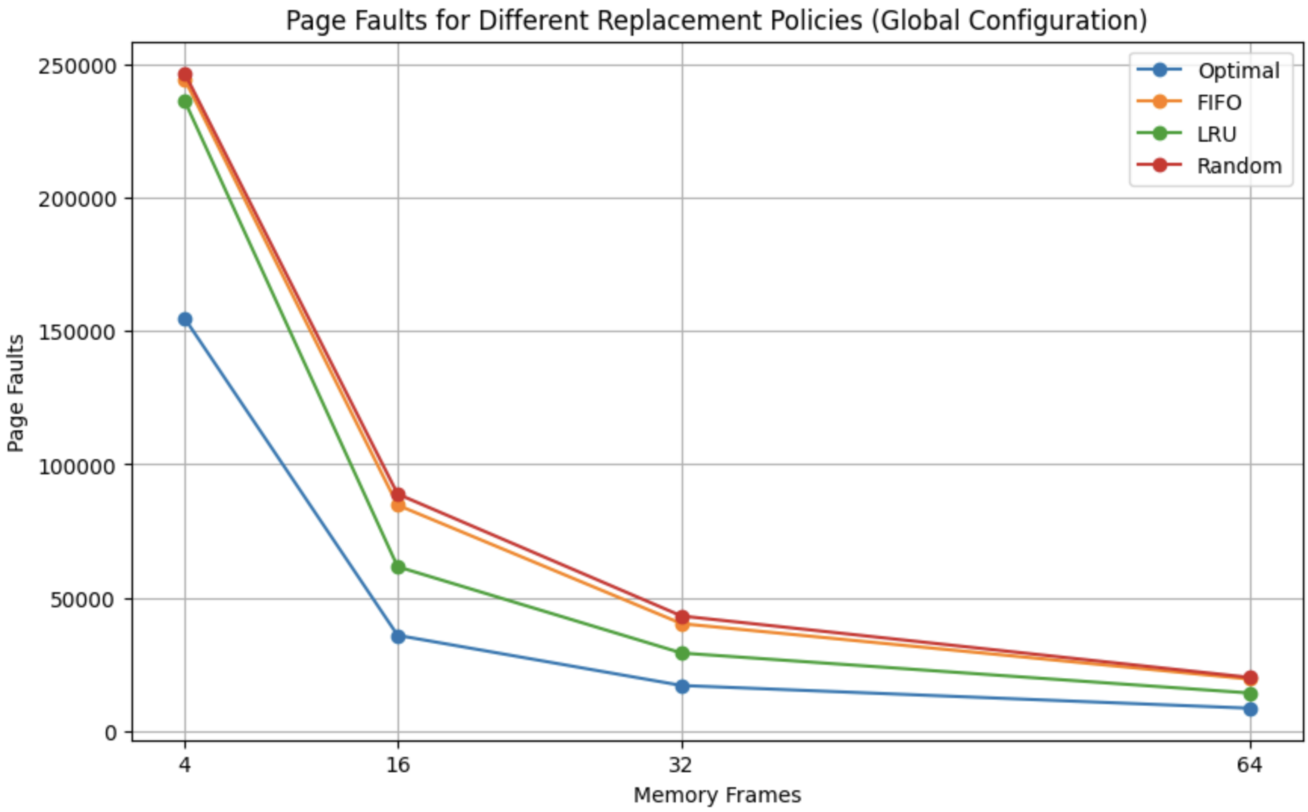

Figure 2: Page Faults for Different Replacement Policies [GLOBAL]

3.3 Analysis of Results

The results indicate significant differences in the number of page faults depending on the combination of page size, memory frame allocation, and replacement policies. The Optimal policy consistently shows the lowest page faults, while FIFO tends to have the most.

4 Conclusion

This project successfully demonstrates the implementation of a Virtual Memory Manager simulator. The simulation of various page replacement and allocation policies highlights the impact of memory management strategies on system performance. Future work could explore additional replacement policies and hybrid approaches.