Santa Clara University
Computer Science and Engineering
Advanced Operating System(COEN 383)
Project 4 Report

Group 4

Sudarshan Mehta
Piyush Joshi
Manasa Madiraju
Patrick Lee

Swapping and Paging Simulation

Overview

The objective of the Swapping and Paging Simulation project is to simulate the concurrent execution of **multiple processes**, incorporating various **page replacement algorithms.** This C programming language-based simulation encompasses **FCFS, LRU, LFU, MFU, and Random algorithms,** exploring process management and memory paging in an operating system.

Locality of Reference

Following a reference to a page i, there is a 70% probability that the next reference will be to page i-1,i or i+1. i wraps around from 10 to 0. In simpler terms, there is a 70% probability that, for a given i, Δi will be -1, 0, or +1. Otherwise, $|\Delta i| > 1$.

Code Overview

The code structure is outlined as follows:

- Definition of constants and data structures, including pages, processes, and a page list.
- Initialization of various parameters such as paging options, hit and miss counters, and a function pointer for the chosen page replacement algorithm.
- A loop simulating multiple system runs.
- Within each run, initialization of list of pages and a process queue.
- Creation of processes with random attributes like process ID, page count, arrival time, duration, and current page.
- Sorting of processes based on arrival time.
- The simulation then advances in time, loading processes into memory and simulating page references.
- Invocation of page replacement algorithms (e.g., FCFS, LRU, LFU, MFU, or Random) when required.
- Tracking of hit and miss statistics for each run.
- Calculation of the average number of successfully swapped-in processes at the end of each run.
- Computation and display of the final hit-miss ratio.

Explanation of Code

The code comprises various functions, each assigned to a specific task. Here's a brief description of each of the page replacement algorithms:

- **FCFS_FUNCTION:** Implements the First-Come-First-Served (FCFS) page replacement algorithm, replacing the page with the earliest arrival time.
- **LRU_FUNCTION:** Implements the Least Recently Used (LRU) page replacement algorithm, replacing the page that has not been used for the longest time.
- **LFU_FUNCTION:** Implements the Least Frequently Used (LFU) page replacement algorithm, replacing the page with the fewest references.
- **MFU_FUNCTION:** Implements the Most Frequently Used (MFU) page replacement algorithm, replacing the page with the most references.
- **R_FUNCTION:** Implements a random page replacement algorithm, selecting a page to replace randomly.

Conclusion

The Swapping and Paging Simulation project offers valuable insights into how different page replacement algorithms **impact memory management** and **page swapping** in an operating system. Through multiple simulations with various algorithms, the code facilitates the analysis and comparison of their performance.

This project establishes a robust foundation for delving into memory management and understanding the trade-offs associated with different page replacement strategies.

Algorithm	Hit Ratio	Pages Swapped
FCFS	0.694816	1492
LFU	0.682709	1607
LRU	0.696135	1486
MFU	0.682210	1564
Random	0.688814	1529

According to our findings, the **Least Recently Used Algorithm** demonstrated the **best performance** in terms of Hit Ratio.