COEN 383: Advance Operating Systems

Project 4 **Page Replacement Algorithm Simulator**
Fall 2023 Mon-Wed 7:10 am Group 8

Team members:

Akaash Meghraj Trivedi (W1650955)
Anagha Viswanath (W1651360)
Michael Zarrabi (W1479183)
Nakul Pravin Thombare (W1650130)
Sai Sanjana Reddy Vatte (W1650167)

Objective:

The primary objective of this project is to delve into the practical implementation of page replacement algorithms. By designing and coding a program, we aim to create a simulation that mirrors the swapping and paging of page replacement algorithms.

Workload Generation Summary:

We crafted a workload of 150 jobs, each defined by <process Name, Process size in pages, Arrival time, Service Duration>. The jobs were randomly generated and sorted based on arrival time, organized as a linked list. Process sizes vary between 5, 11, 17, and 31 MB, with service durations evenly distributed over 1 to 5 seconds. This diverse and organized workload provides a realistic simulation environment, capturing the variability inherent in real-world computing scenarios.

Simulation Overview:

- The workload was generated and organized into a sorted queue based on arrival time.
- A free page list was created and initialized with 100 pages, each of 1 MB.
- Jobs were sequentially picked from the queue. If there were at least 4 free pages, the process commenced; otherwise, it waited until sufficient pages were available. Each process was represented by a header and a linked list of its memory-resident pages.
- Records were generated when a job started or completed, including timestamp, process name, entry/exit, size in pages, service duration, and memory map.
- During execution, each job generated a memory reference every 100 milliseconds to a random page within its virtual address space. Corresponding records were created, indicating the timestamp, process name, page referenced, whether the page was in memory, and the process/page number to be evicted if needed.

- In cases where memory was fully utilized and a referenced page was not in memory, a chosen page replacement algorithm was applied to select a victim page for eviction and make room for the required page.
- The simulator was run five times for 1-minute intervals, each time using a different page replacement algorithm (FIFO, LRU, LFU, MFU, and random pick).
- The simulation continued until the completion of the 1-minute interval, with statistics collected and saved before exiting.
- For an additional five runs, the hit/miss ratio of pages referenced by running jobs was computed for each run, and the average over the five runs was determined.
- A separate simulation was conducted for 100 page references, with records printed for each reference, including timestamp, process name, page referenced, whether the page was in memory, and the process/page number to be evicted if needed.
- The average number of processes successfully swapped in was calculated for each replacement algorithm over the five runs.

Conclusion:

This assignment provided a comprehensive exploration of paging and swapping for page replacement algorithms in a practical context. The simulation of page replacement dynamics not only reinforced theoretical concepts but also required thoughtful design decisions to create a coherent and realistic program.