

Virtual Memory Management Problem

This program simulates the following page replacement algorithms:

First In First Out (FIFO)

- When a page fault occurs, the page that has been in memory the longest is replaced

Least Recently Used (LRU)

- When a page fault occurs, the least recently used page in memory is replaced

Most Recently Used (MRU)

- When a page fault occurs, the most recently used page in memory is replaced

Optimal

- When a page fault occurs, it looks forward in time to find the page to replace

When reviewing the results from [vmemman_output_example.txt](#) you can see that there is a pattern for every page replacement algorithm, which is that as the number of frames increases, the percentage of page faults decreases. This means that it is having better performance because less page faults mean more hits resulting in better performance. This makes sense because as the number of pages increase, there are more chances for hits decreasing the page fault rate.

When comparing LRU with FIFO, LRU outperforms FIFO by a very small margin. The optimal algorithm had the best performance when taking a look at the overall % of page faults throughout all of the page sizes and number of pages. MRU comes in second place, followed by LRU and FIFO respectively.

Overall Average Page Fault

1. Optimal

- $(68.80 + 53.90 + 45.00 + 37.90 + 11.27 + 3.40 + 13.97 + 1.73 + 1.63) / 9 = 26.4\%$

2. Least Recently Used (LRU)

- $(80.00 + 60.10 + 42.07 + 60.47 + 22.80 + 3.57 + 26.03 + 1.90 + 1.83) / 9 = 33.19\%$

3. First In First Out (FIFO)

- $(80.37 + 61.00 + 42.97 + 61.40 + 23.60 + 3.57 + 26.67 + 1.90 + 1.83) / 9 = 33.70\%$

4. Most Recently Used (MRU)

- $(65.70 + 64.23 + 62.60 + 38.07 + 35.77 + 34.30 + 16.30 + 15.77 + 14.20) / 9 = 38.54\%$