Sharon Shin

October 1st, 2022

Assignment 5

ICS-462

This assignment has taught me three different approaches to handle page replacement through FIFO, or removing the first in page in the frames, LRU, or removing the last recently used page in frames, and Optimal, or removing the last occurring or non-existing page in frames in the case of a page fault. After graphing out the performance of the three different algorithms, it seems like there are many factors in play when deciding which algorithm is the right choice to implement. First, it depends on how many frames you have. The lower the frames, it may not matter as much which algorithm you choose because they all tend to perform the same. Once you 3 frames, it matters more. Also, it depends on the page. As you as see below, the algorithm’s performance is heavily dependent on the page itself.

Chart, line chart

Description automatically generated

For this randomized page, it appears that 1 or 2 frame(s) performs the same for all three algorithms. Even for 7 frames, they all perform relatively the same. It is between 3-6 frames where we see the largest difference in performance.

Chart, line chart

Description automatically generated

For this given page, again, 1-2 frame(s) perform the same. From 3 frames and on, it is clear that Optimal is the best approach for the best performance.

Chart, line chart

Description automatically generated

For the second given page, LRU and Optimal perform similarly, with FIFO performing a tad worse. Optimal only performs better than the other two in the case of three frames. This shows how the page can be a factor in performance.

Chart, line chart

Description automatically generated

Of course, an average of the three algorithms’ performance for the three pages would show a better trend for what algorithm tends to perform the best. Optimal performs the best after 3 frames and will never perform worse than FIFO and LRU. Optimal is the clear winner, however the implementation may be harder than the other two. I found this assignment to be very fun and a good refresher on data structure.

This assignment took me about a week, or about a total of 20 hours to implement, document, and test, including the time to create these graphs and check my work by hand.