Mason Walls
Section 7

Lab 6: Page Replacement Algorithms

Summary:

In this lab we were tasked to write three different page replacement algorithms. First in first out, Least recently used, and Optimal. First in first out and least recently used were pretty easy to implement. All that was necessary was to iterate through the page frames, and see which page was either put in first, or accesses least recently. These algorithms were able to be programmed to have a worse case run time if O(n), with n being the number of frames. Optimal required us to check if a frame was going to be used again in the future. If so, we had to find the most recent use, and replace that frame. If a frame wasn't going to be accessed ever again, we would remove that one, as it is taking up space we need. This required a set of nested for loops. One to check each frame, and one to see if that specific page id was going to be accessed again. This made the worst case run time of the optimal algorithm be O(n*m) were n is the number of pages and m is the number of accesses.

Overall, this lab taught me a lot about how a computer can access memory, and how advantages such as knowing when a page is going to be accessed, can speed up memory times.