CS 5/7343 Fall 2021

Programming Homework 4

Custom Page Replacement Algorithm

The replacement algorithm I have implemented is random. The algorithm first checks for a page fault be iterating through the pages to search for the next page. If the next page is found, then -1 is returned to indicate a page hit. However, if the next page is not found, then a page is selected for replacement at random. This method incurs very little overhead and only requires storing the previous random number to generate the next.

The random number generator chosen is not the standard library rand(). Instead, the xoshiro256+ pseudorandom number generator is implemented for its speed (https://prng.di.unimi.it/xoshiro256plus.c). Benchmarking of this generator shows 510% performance relative to the base rand implementation (https://thompsonsed.co.uk/random-number-generators-for-c-performance-tested).

Personal testing indicates that this method usually produces fewer page faults than FIFO. The clearest benefit to using this method is there is no longer a need to track or manipulate page references. But an obvious negative is that the random choices may be bad and swap out a page right before it will be used.