Tim Clerico

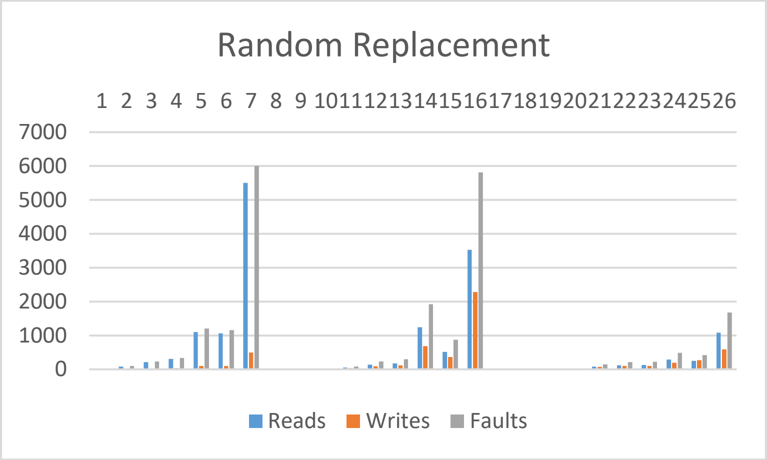
3/30/18

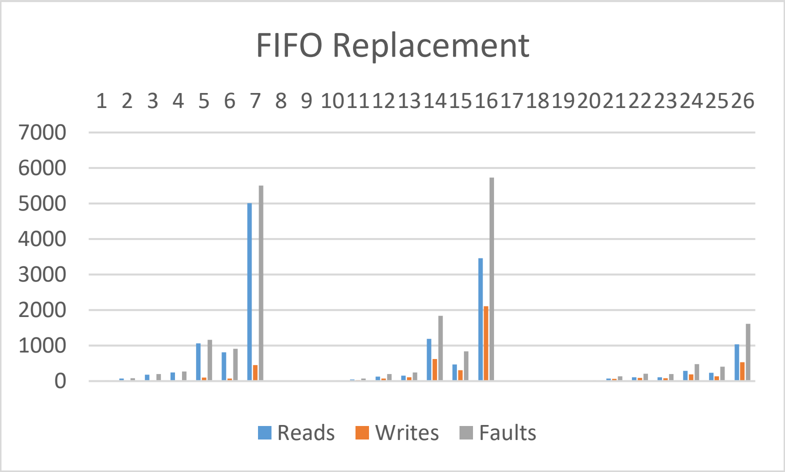
Operating Systems

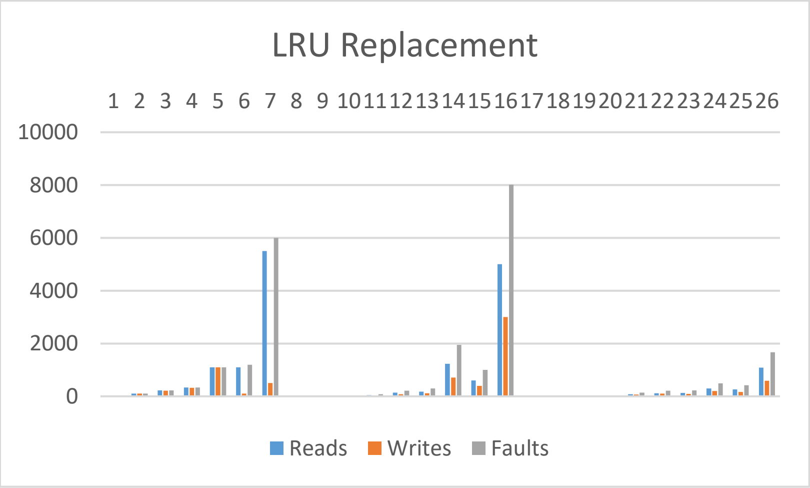
Scheduling Report

For this assignment we were tasked with implementing algorithms for process scheduling using page and frame tables. Although it is only a simulation and we are not dealing with true page faults and scheduling, the algorithms still function the same. The functions we were tasked with implementing were Random Replacement, FIFO, and LRU. For the Random Replacement, a random frame is selected using a random number generator, and the page in that frame is replaced with the new one. First In First Out (FIFO) is a fairly self-explanatory algorithm. I decided to use an array to keep track of the frames in use, replacing the 'oldest' frame when a page fault occurred. Lastly is the Least Recently Used algorithm. I decided the best way to implement this was by using something similar to a stack. By using a doubly linked list with pointers to the front and end I emulated a stack. In theory, when a page that exists in the frame table is used it would move up to the top of the 'stack', leaving the least used page at the bottom. Unfortunately, in our simulation environment we couldn't truly implement LRU due to not being able to trap the reads or writes in order to update the 'stack'.

After implementing each algorithm, I went to work testing each one. I started by having the same number of frames as pages, but slowly increased the number of pages. Below are the graphs for each algorithm that includes all three of the provided programs for testing (scan, sort, focus).


Random Replacement
Reads
Writes
Faults


FIFO Replacement
Reads
Writes
Faults


LRU Replacement
Reads
Writes
Faults