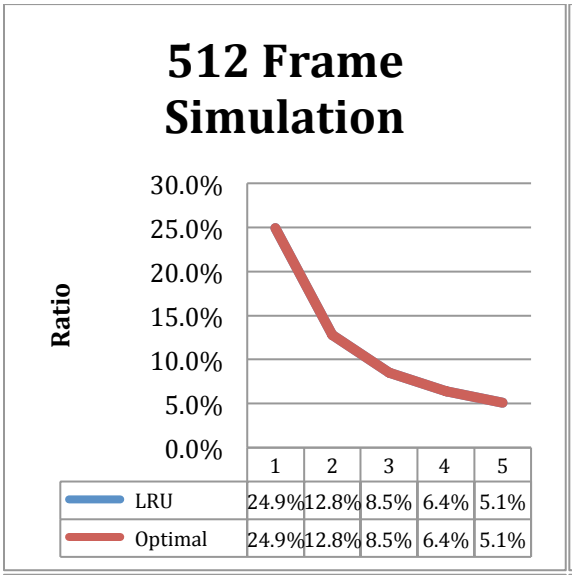
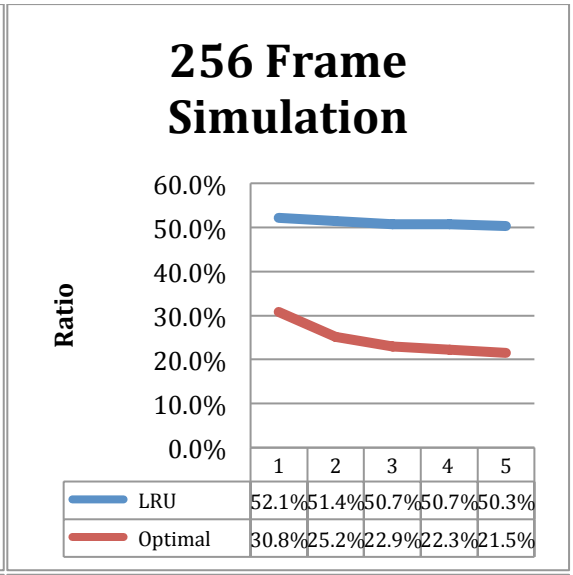
Rob Fitzgerald
CSCI 3453 - Operating Systems - Summer 2015
Lab 3 - Paging Replacement Algorithms

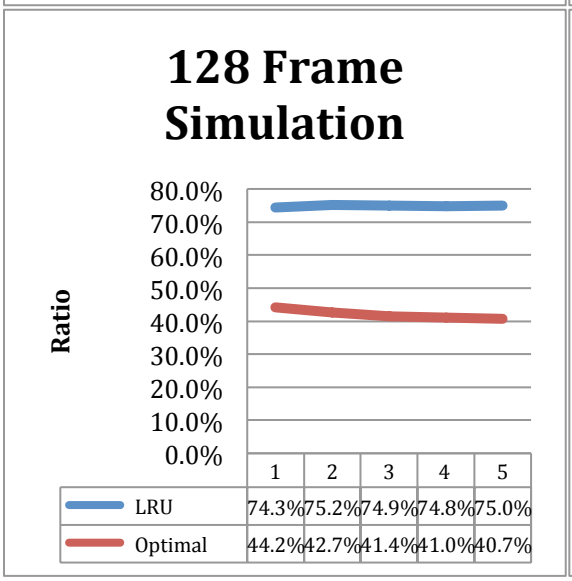
This lab presents two page replacement algorithms in comparison with a common set of simulated page replacements. Both algorithms are only concerned with the specific process of freeing memory to allocate a page which encountered a page fault. Least Recently Used (LRU) algorithm finds the page that is oldest, in that it is the least recently accessed by the process, and it removes it to free space. Optimal algorithm, on the other hand, revues upcoming page requests, and removes the memory resident page which will not be acccessed again for the longest period of time.

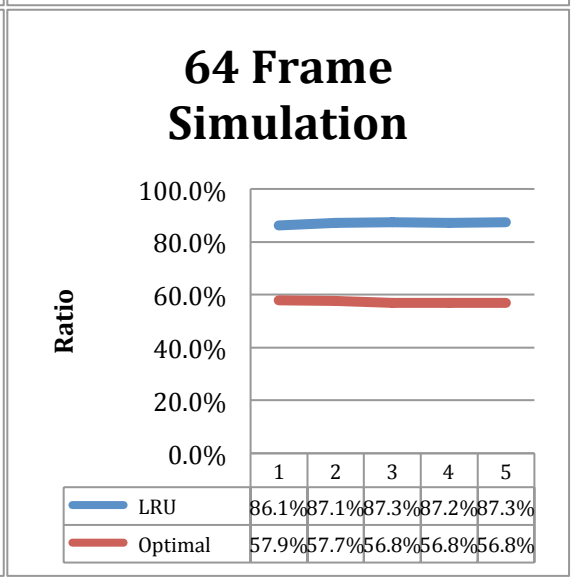
In our simulation, there are 512 different pages of which can be accessed. Our control group test is for a frame count of 512. Since there is a one-to-one ratio of pages to frames, there will only be one page fault in a page's lifetime. This test of 512 frames is our baseline for performance, since both algorithms will have the same behavior. For any frame size less than 512, we see that the optimal algorithm lives up to it's name, and consistently holds about 30% above our test of pure random number generation. In this case, regarding optimal, the performance seems to improve over time; as the number of accesses increases, the optimal solution seems to always decrease. Alternatively, LRU seems to get worse over time when smaller frame sizes are used. This is their relative behavior; in both cases, as frame count approaches pages count, both algorithms improve their overall page replacement percentage. Therefore, to decrease the page fault rate of either algorithm, the frame size should be increased.

Results from the simulation can be found below. The data table shows percentage of page faults for every 2,000 page accesses (numbered 1,2,3,4,5).









Since a running process has many different characteristics, it is likely useful to implement some degree of both algorithms in order to balance the needs of the processor with the needs of the process' working set.