Luke Kraus

CS 1550

3-21-2017

Project 3 Write-up

Refresh:

The refresh rate that I decided was best was a refresh rate of 40. As you can see in the graph the value of 40 is very close to the most optimal. A refresh rate of less than 40 leads to more page faults, and this remains true for each of the frame amounts except for 8. For 8 frames, at the value of about 30 there is an increase in the number of page faults. I chose 40 because I thought that it gave a value that is close to 50, which seems to be very optimal, but also was closer to 30 because of the issue with increasing for 8 frames.
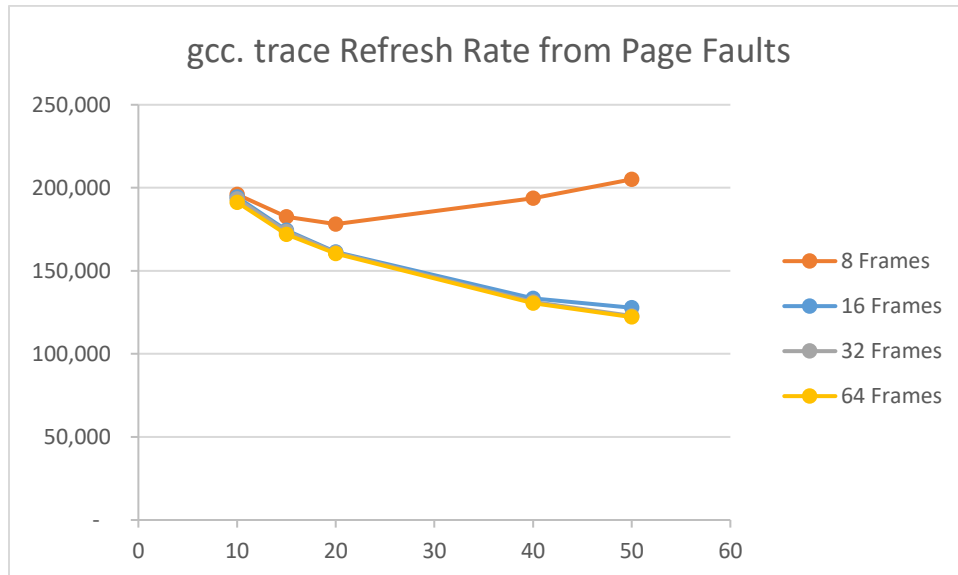
Algorithms:

The algorithms that were tested in the this project were Clock, NRU (Not Recently Used), Random, and OPT. Out of these 4 OPT was obviously the best. This was to be expected, but it is not possible to implement. So after looking at the statistics I came to the conclusion that Clock would be the best to implement. It gave the lowest number of page faults. The one drawback for this algorithm is that it does have slightly more disk writes then other algorithms, but this is not as important when compared to the other metrics. The next best algorithm was not as clear. NRU and Random both had their positives and negatives here. Obviously Random had a much higher number of disk writes because of the random nature. NRU was a better algorithm, page fault wise, when there were 20 or less frames, but when it crossed approximately 25 page faults,

Random became the better algorithm as it dropped in number of page faults while NRU flat lined. Overall, if I was going to implement an OS I would chose the Clock algorithm out of the 4 that we tested because of the lower number of page faults.

gcc.trace Page Faults