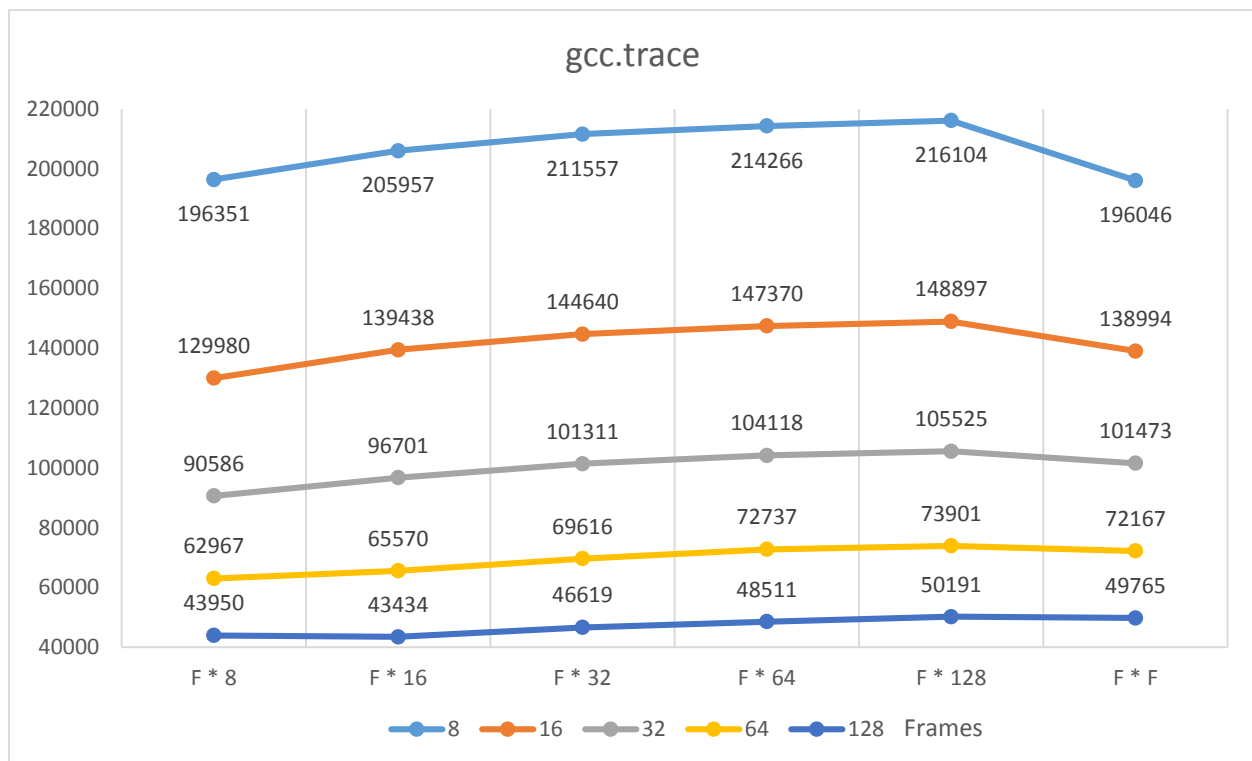
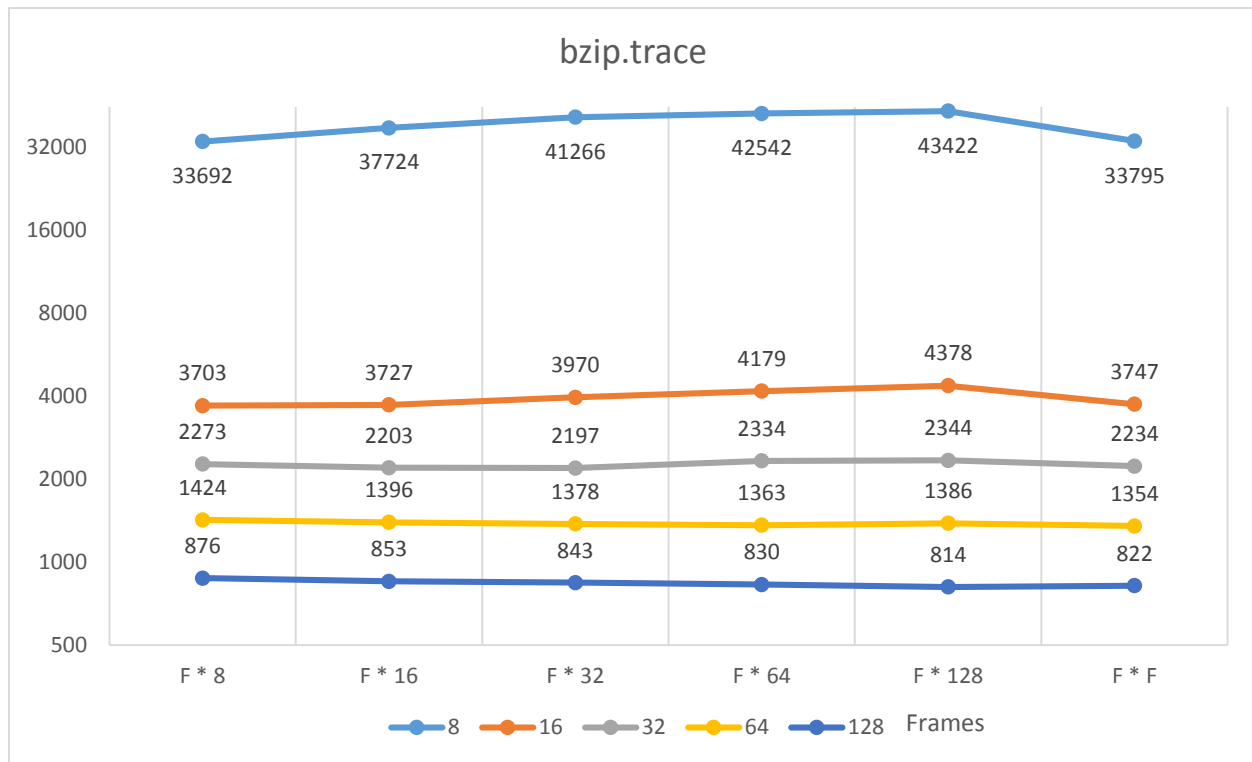


# Project 3

Mark Shanoudy

## Not Recently Used (NRU) Page Replacement Algorithm



The above graphs show the results from sampling various refresh rates to use in conjunction with the NRU page replacement algorithm. On the x-axis are the refresh rates that were sampled. The letter  $F$  represents the number of frames during that run. On the y-axis are the number of page faults. I decided to choose  $F * F$  as the refresh rate because it offered the average best page fault performance across all frame sizes and between trace files.

## Algorithm Comparison





## Conclusion

The below graphs show the page fault percentage difference of each page replacement algorithm vs. the optimal page replacement algorithm. When tested with the bzip trace file, the random page replacement algorithm showed an average page fault increase of 34.9%, the NRU algorithm showed an average page fault increase of 12.0%, and the clock algorithm showed an average page fault increase of 7.6%. When tested with the gcc trace file, the random page replacement algorithm showed an average page fault increase of 39.5%, the NRU algorithm showed an average page fault increase of 31.9%, and the clock algorithm showed an average page fault increase of 8.8%.

The clock algorithm vastly outperformed both the NRU and random page replacement algorithms. The data therefore suggests that the clock page replacement algorithm is most suited for use in an actual operating system.

