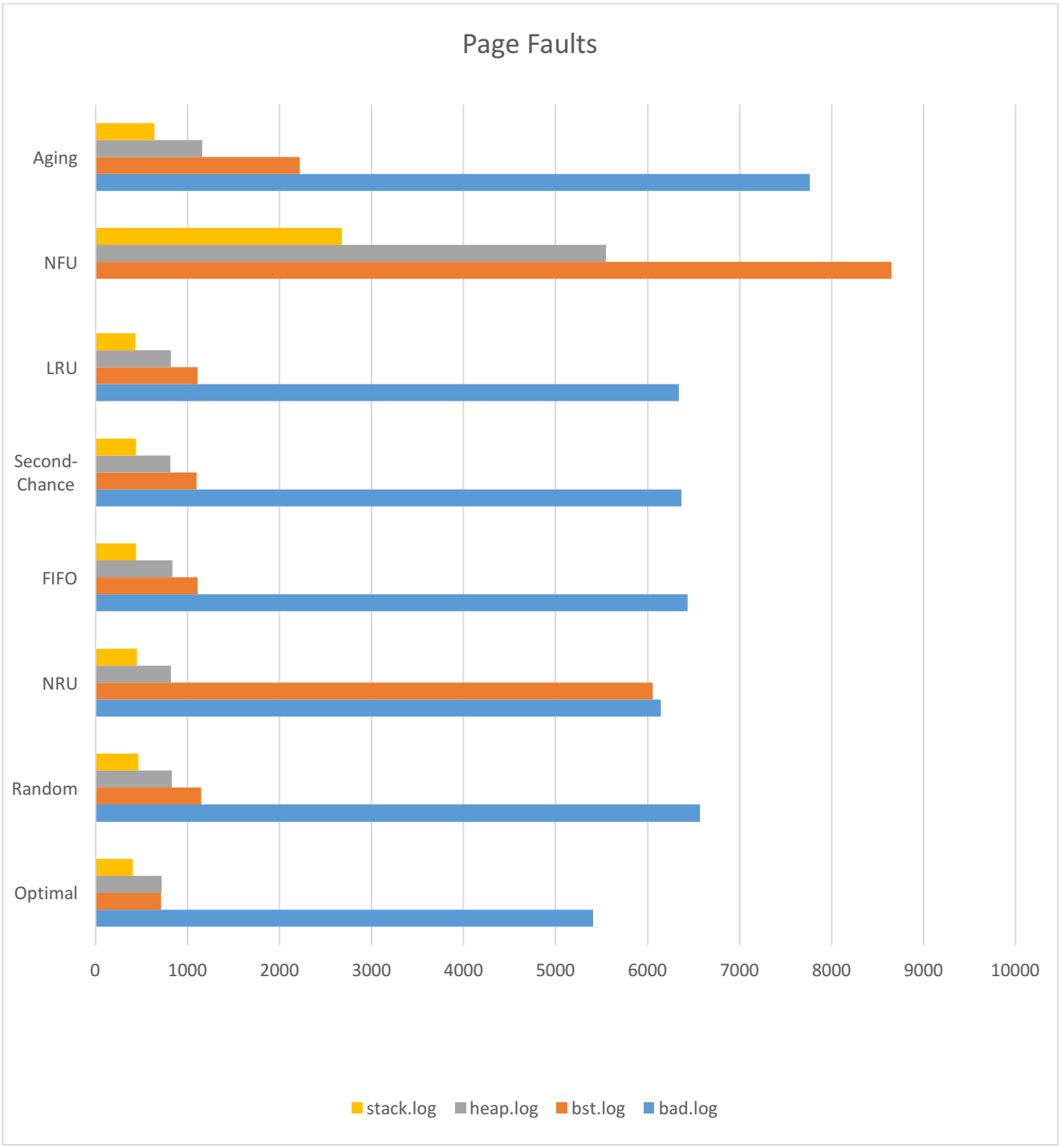
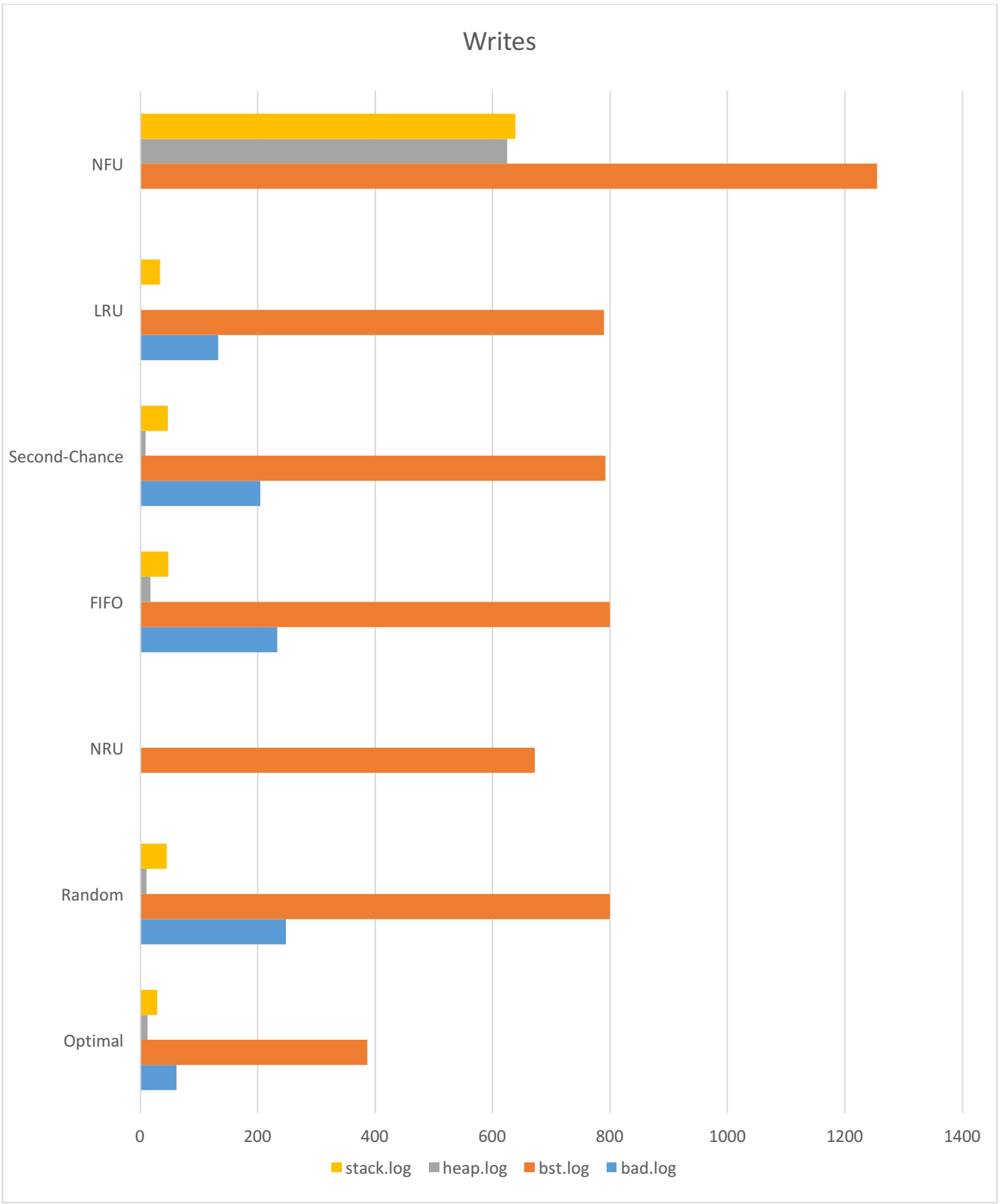
Robert Lowman
CS364: Operating Systems
Dr. Hogg
29 November, 2016

Project #3 Report



* bad.log for NFU not shown as it was way off the charts (58962)



* NFU bad.log not shown (3402)

	Page Faults	Writes
Usual	5406	62
Double Page	5245	34
Half Frame	5406	60
Double Page & Half Frame	5245	34

Looking at the different log files, it is apparent that the bad.log had considerably more page faults than other files, meaning that this log file contains more cases where each algorithm becomes more expensive. While the bst.log program had less page faults, there were many more writes than other log files for this statistic. Stack.log and heap.log were quite similar, however stack.log had more page faults and more writes.

Looking at all the different algorithms, it is apparent that they all produce different results. With this data, you can see that NFU is the absolute worst algorithm as it has considerably more page faults and writes, and the pre-written Optimal is the best. I believe that my aging algorithm has a bug, as the results are rather bad compared to how I imagined them to be as this should be our most efficient algorithm. Seeing the difference in results for random and most of our algorithms not being the great, it makes us wonder how efficient these algorithms can be as just randomly picking a page is not much difference in these metrics.

Most of these algorithms produced results that are very similar, which makes me wonder if my page size and frame count were of sizes that could really utilize the algorithms. However, it is apparent on which algorithms worked best for each log file. Most algorithms that were expensive in the bad.log were also quite expensive in the bst.log, and vice versa. This proves that

our algorithms are quite consistent in their results and work similarly no matter what the cases are.

When we doubled the page, it lowered the amount of page faults which means that there is a correlation between page size and page faults. When the number of frames decreased, so did the number of writes, which makes me think that with less frames there is less likely to be a write, as there is less possible choices for it to be written to.

My Aging algorithm results were quite surprising, as we mentioned that this algorithm should be one of the most efficient. The page fault count seemed quite high, and also the amount of writes. This makes me think that implementation of this algorithm could be wrong, such as the counter not being incremented in the correct manner.