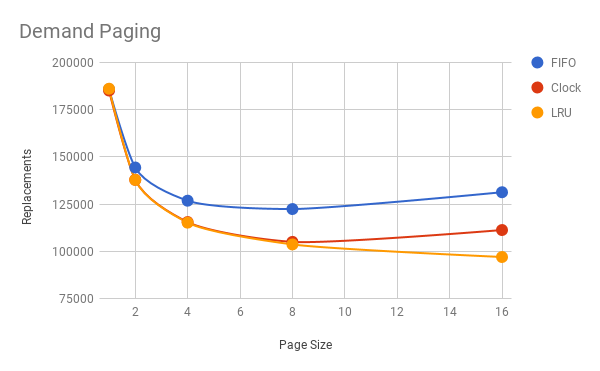
Michael Iacona and Max Melo

We expected the FIFO replacement method to be the worst, clock to be the next best, and LRU to be the best. We expected demand paging to initially have a higher number of replacements and then decrease as the page size increased. We expected pre paging to have a smaller number of replacements than demand paging at the start and then decrease as page size increased. We eventually saw that our expectation was correct, FIFO was the worst followed by clock and LRU. However, FIFO and Clock had an increase in replacements from page size 8 to 16. Pre-paging matched our expectations and had less replacements overall. However, FIFO and Clock had an increase in replacements towards the end for demand and pre-paging. We believe this is due to Belady’s Anomaly which is common with FIFO, so it is not surprising that Clock encountered this issue as well since it is an extension of FIFO. LRU has an increase from 8 to 16 as well for pre-paging. We believe this is a result of pages being kept in memory longer and having a higher chance of pages being dropped from memory before they are used again.

LRU was the simplest to 8implement because pages are updated with a timestamp when accessed. FIFO was the next simplest and Clock was the hardest because it required you to keep track of all the access times and the modified bit for each page. A random access memory trace would affect the results. All methods would likely have a higher number of page faults for all three swapping methods. Pre-paging would be affected much more than demand because there would be a much higher number of swaps for each fault. The memory would likely be filled with a lot of pages that are not going to be accessed soon due to the randomness.

