

In this lab we were able to play around with different page table eviction schemes. The three different techniques were: First-In-First-Out (FIFO), Least Recently Used (LRU), and Optimal (OPT). FIFO involved looking for the page that had the least arrival time out of the current page tables and evicting it. LRU was a similar process but instead of arrival time we looked for the minimal access time. Lastly, OPT involved looking for the page that will be used furthest in the future. The OPT scheme can't be implemented in the real world due to the fact that it requires knowledge of the future. It has use though as a baseline for comparing schemes against as it is the fastest scheme. We tested our three implementations and reported the number of page faults we received when paired with different access schemes. Here was our results:

```
bash-4.4$ ./lab6run
The average number of page faults for FIFO with Random Access is 8760.
The average number of page faults for LRU with Random Access is 8759.
The average number of page faults for OPT with Random Access is 5925.
The average number of page faults for FIFO with Sequential Access is 10000.
The average number of page faults for LRU with Sequential Access is 10000.
The average number of page faults for OPT with Sequential Access is 8830.
The average number of page faults for FIFO with LR Workload Access is 883.
The average number of page faults for LRU with LR Workload Access is 874.
The average number of page faults for OPT with LR Workload Access is 611.
```

Overall, this lab was good for implementing in class concepts and to see how theory translates to practice. I got more experience with C, page tables, and OS concepts through this lab.