Aly Ghallab

CMPSC 473

P3 Report

The purpose of this project was to implement a paging system that implements First-In-First-Out and Third chance policies. The system relies on the program throwing a segmentation fault due to access of memory without proper permissions and I had to implement the signal handler, which essentially served as a catch for when the segmentation fault happened. Getting the page meta data (such as the address, page number, etc) from the segmentation fault was the biggest challenge to starting the project which was not too bad. I personally struggled with getting the physical address to be accurate until I realized that I was not tracking the physical frame index that I needed to use to calculate the physical address. For both FIFO and Third chance I used a circular linked list as my main data structure. Once I was able to track the physical frame index of a page by inserting them into my node structure FIFO was pretty easy.

The third chance has presented issues. It is essentially a modified FIFO but when a function is reread or written into a frame, I used a flag to keep track of those which essentially helped me determine if a particular page qualified for a second or third chance. However I had to modify my eviction helper function for my linked list to be able to implement third chance. I ran into a lot of hurdles and bugs but I do believe I was able to get it working, or at the very least mostly working.

In interface.c my main functions and majority of the code are located. Mm_innit() mallocs and initializes my data structures (i.e elements of my circular linked list). I also use it to get and store the parameters from main.c such as the number of frames, page size, etc. My second function is obviously the signal handler, it only gets called when the program throws a segmentation fault and I essentially had to give the appropriate access using mprotect. A challenging part of this project was figuring out

when to set a page to have no permissions to trigger a segmentation fault, this was necessary to figure out for type 3 “Track a "read" refers to the page that has Read and/or Write permissions on.” Fault and type 4 “Track a "write" refers to the page that has Read-Write permissions on.” Ultimately it’s this part of the project that brings me my biggest challenges and concerns when grading, as it is the part I have the shallowest understanding of. My initial idea was to set mprotect to None after I was done reading/writing every page but I figured that would not be optimal and sure enough, it says not to do so in the assignment description.

All in all, I thought the project was of the appropriate difficult where it was challenging and made sure I understood the concepts yet not too overwhelming where it was not feasible to code.