Lab 3 – LRU Buffer Pool

# Thursday

10-22-2020

## Objectives

* Read through lab contents to understand scope of lab
* Review buffer pools/buffers from textbook
* Plan implementation
* Start to implement

## Notes

* After reviewing the material for the lab, I think I have an understanding of what is expected.
* I need to first create the buffer block to manage the data. It will have a private variables for the data, its index, and its size
* Once the buffer block is implemented, I can implement the buffer pool, which needs to have access to the 5 buffers. I’m not sure what the best implementation is here. I’m thinking about an array of pointers, but that would require shifting all elements in the array pool when they are changed.
* I’m gonna do some research on buffer pool implementations.
* I need to remember to use smart pointers. Making a smart pointer for each buffer might be smart as I can make it auto delete when it goes out of scope.

## Results

* I built out the buffer block class
* I started to build out the buffer pool class
* I was very surprised and concerned that after building out both classes without testing, my code compiled first try. Not only that, but it partly worked as expected. God is good.
* As of now, the buffer pool appears to manage the blocks as expected with the LRU heuristic, but the output of the data contained in the blocks is not English. It is “ “. I’ll look into that tomorrow.

# Friday 10-23-2020

## Objectives

* Debug program

## Notes

* Debugging is always a humbling experience
  + I noticed that I never read in the new section of the file when getting a block not in the buffer
  + I noticed that I swapped the sz and pos variables in my functions, which threw several things off
  + It appears to me as though the data should be working correctly, but instead the output is jumbled nonsense.
  + I think this might have to do something with the way char arrays are implemented.
  + Nope. Here was the problem:
    - When I iterated through the block to get the value requested, I was assigning i (my iteration variable) to be the starting position that I needed from the block. I then was assigning data[i] to block[i]. The problem with this is that i is out of range in data. I ended up using pos as an offset.

## Results

* The lab works as expected