Scott Dobberpuhl and Jackson Gudorf

For our last project the whole thing appears to be working. Jackson and I, I think, understand cache well and this project ran into some errors that were semi-easy to fix. One error that we were getting a lot during development was a segmentation fault where we were setting the block size incorrectly and it was what took the most time to fix while writing this simulator. Another error we were getting was we were somehow incorrectly allocation our memory for the block data and this also caused some segmentation faults while trying to solve our other error, that was stated previously. The last large error that caused some grief was we were accessing memory addresses incorrectly, but that one was solved quickly.

The way our simulator works is very similar to the single cycle, but with some odd tweaks in there. First, we do the usual of setting all our variables to their default values and initialize our cache variables. We then create our structs based on the configuration we received from the command line arguments. Then we start to our while loop that will run through every instruction till it hits a halt. We first do our instruction fetch and our first check for cache and do our hit and misses. If it's a miss, we read into cache and move on. If it is a hit, we just get it. Right after we update our LRU. We check for halt and if we hit it, we stop.