Project 3 Summary

Originally, I approached the project in C/C++ using their bit manipulation class, but then I realized that some of the things I wanted to write such as custom functions and other quality of life classes would actually just take an exorbitant amount of time. So, after spending a few hours in C/C++ I decided to move to Java and I had to copy over the logic of what I had done and convert it into Java code. I approached the project using Java’s Bitset class (which is used as a block). After a while I realized that bit manipulation became rather tedious to do, so before even beginning the implementation of a file allocation system I wrote helper functions to handle the bit manipulation. I extended the bitset class to add a pointer to the last location inside the bitset which allowed me to write an append function, a function that takes any number string inputs and appends them to the bitset. The disk has an ArrayList (for easy access and manipulation) The project was not very difficult overall, but I felt like it was unnecessarily tedious at times. The most difficult part of this project was writing the helper functions that I used, and the actual implementation of concurrent, indexed, and chained file allocation was very straightforward. It was good to see, or at least experience how a real file system would apply such allocation methods, and I learned a good bit about the bitset class and the bit manipulation needed to store things in the file system. The end project is very straightforward like the details define. It’s a text-based user interface with multiple options for the user to choose from. Any data stored in the program is lost on restart. The user can select to display a file, display the file allocation table, display the free space table, display a specific block, save a file from the file system, save a file to the file system, and exiting the program. The program should be able to read and processes text documents.