

## Project 3 Summary - CS4348.001

Name: Kaushik Nadimpalli

The purpose of this project was to simulate disk allocation methods. The disk will consist of 256 blocks of 512 bytes each. The first block is for the file allocation table and the second block is a bitmap for free space management. The remaining blocks hold data for the files. An array was used to store the disk for the program. Unlike previous projects, I had a new experience with this one since I chose to use C++ instead of Java. Given the amount of time we had, I wanted to try to implement it in another language, and the experience was worthwhile. Some challenges I ran into were understanding the logic behind implementing the code itself. While conceptually, I had an idea on how the three allocation methods worked, I struggled more than I anticipated when it came to the implementation of these methods in C++. I utilized header files in my program to keep track of all the functions I used in my source files. Moreover, I divided it into four different source files. As started in the instructions, I made sure that the only purpose of the disk was to read and write blocks by block number. I created some public methods and used them in that class, so that the disk itself served as its own object. Then, I created another file for the user interface. I also made sure in this file that input validation was taken care. Specifically, I made sure that regardless of how the user enters the allocation method name, it converts it to lowercase and then that will be linked to the enumeration and used I created for the bitmap, alloc, and the three methods. The driver file was the most important file for my program since it encompasses all the necessary functions to simulate the disk using one of the three methods. I had a multitude on public and private functions, depending on my own judgement, and used them to create the functionality behind the simulation. Last but not least, the main program is the project3 class and it contains the main method. The header file for this source file is included in all other files since it has important libraries that I used and this is also where I created my enum.

Some challenges I ran into while implementing this program had to do with my expertise in C++. It has been a while since I used this language so it took me more time than expected to learn some things I had forgotten from Unix class. Furthermore, I also had trouble implementing the specific methods. Instead of doing them as separate functions, I instead implemented them in more than one place in the Driver file - such as for removal or getting

information from an input file etc. For the contiguous allocation, each file occupies a contiguous set of blocks on the disk and it was hard to do that in a single function. However, the concept behind was once the starting block address and the length of the file, we can get the blocks occupied by the file. I had most trouble with index allocation since it took me a while to work with pointers again. This method contains the pointers to all the blocks occupied by a file and each file has its own index block. For linked/chain allocation, each file is a linked list of disk blocks and each directory entry has a pointer to the first disk block of the file. Overall, it was very consuming to implement them while also making sure that the functionality was working fine.

I had a good experience from this project. We did get enough time but I felt like we could've gotten example code like last time. The slides were only helpful conceptually and even the textbook was not too detailed and helpful for the implementation itself.