**SD LAB 5: Memory Manager.**



Student: Johnson Domacasse

Date: 27 Nov 2023

Student#: 4471709

Teacher: Lake Lakeman

Contents

[Introduction: 3](#_Toc151998258)

[Doubly linked lists: 3](#_Toc151998259)

[Problems: 3](#_Toc151998260)

[References: 4](#_Toc151998261)

# Introduction:

*The purpose of this document is to show or justify the decisions that were made during this assignment. Below you will find the sections that talk about the functions that were used, of which were added extra and finally some problems I had making the assignment.*

The idea I had for the design and after I looked at the API that was provided to me initially was that the Mlist class would act as the placeholder for all of my functions. The memory manager class simply needs to have the claim memory and the free memory implemented based on the functions that are implemented in the Mlist class.

Something additional I did (which doesn’t have any effect on the grade) is get some understanding on the make file. I noticed that when I was making the executable and tested it with the input text files, it would create an out text file. Everytime I cleaned, that file stayed, so I changed the make file in a way that cleans everything, including that out text file.

# Doubly linked lists:

For this assignment, After some back and forth with researching I have decided to implement a doubly linked lists for my allocate and free lists. The reasons for this is because initially I thought I would need to traverse the list from both back and front. I realized in the implementation that I would not be doing this but rather I kept the doubly linked list implementation because it made for easier access to some nodes in the list itself.

In order to implement the doubly linked list, I had to make a small adjustment to the ITEM struct which was to add the “prev” variable to it. Next to this I implemented four functions to handle the insertion and deletion from the list. **[1][2]**

# Problems:

Throughout this assignment I had numerous problems. Here are some that I can remember (had a big impact) and how I solved them:

* **Merging the memory chunks:** When I finally had the implementation more or less working, I realised that some chunks of memory do not get merged like they should. For example if I had two chunks of memory, each with a size of 10 that were allocated one after the other, then when I free them, they should be merged into one memory chunk with a size of 10. They simply get put as two memory chunks with size 10 back into the free list. To solve this a function was made to merge the two memory chunks when applicable. **Note:** This functions was developed in the free list class because this was the only location that I would need to merge the memory chunks.
* **Sorting the memory chunks:** As I was implementing the free memory function, I noticed that although the memories do get freed, they do not go into their correct place. For example, if I allocate a block at address 1020 and another at 1030, and then free the one at 1030 then the one at 1020 they would not be in the correct order.So instead of it being the address at 1020 first then 1030, it would be 1030 then 1020.To solve this issue, the add in order function was later developed (initially it was 3 functions: add first, add after, remove). This way I could add the nodes in the correct order. **[3]**

# References:

*This section is to give all of the references that I have used in the making of this assignment. Some of which I solved my problems with. Others for simple operations like how to insert a node in a doubly linked list.*

**[1] –** GeeksforGeeks. (2023, September 26). *Introduction to doubly linked list*. GeeksforGeeks. <https://www.geeksforgeeks.org/data-structures/linked-list/doubly-linked-list/>

**[2] -** YouTube. (2020b, October 14). *Introduction to doubly linked list*. YouTube. <https://www.youtube.com/watch?v=e9NG_a6Z0mg>

**[3] -** GeeksforGeeks. (2023a, June 1). *Insertion in a doubly linked list*. GeeksforGeeks. <https://www.geeksforgeeks.org/introduction-and-insertion-in-a-doubly-linked-list/>

**[4] -** GeeksforGeeks. (2023a, January 10). *Delete a node in a doubly linked list*. GeeksforGeeks. <https://www.geeksforgeeks.org/delete-a-node-in-a-doubly-linked-list/>

**[5] -** YouTube. (2013, November 13). *Doubly linked list - implementation in C/C++*. YouTube. <https://www.youtube.com/watch?v=VOQNf1VxU3Q>