Lab5:MallocLab

Junhao Yu

June 3, 2022

1 Problems I Met and How I Solved Them

To be honest, this lab really caused me trouble. For instance, the macros are really annoying. Only after I had written a whole lot of codes then I realized that some of the operations could be packed in a macro. Also, I found it really difficult to debug since the problems could be anywhere. Macro is possibly wrong, pointers are possibly wrong and code itself could be possibly wrong.

To be specific, the main problems I've met are as followed.

1. How to link the segregated list. Since the handout instructed that no structures should be defined, I was really obfuscated with this problem. Then after I searched the Internet, I found out that a careful use of char and its pointer can solve this problem.(Details I explained in the mm.c).

2. How to increase the throughput. I first thought of explicit free list and best fit, but throughput didn't increase very much contrast to implicit list or next fit. Then I realized that it was all because of the search cost. Finally, I uesd the segregated list since it could limit the search cost to O(logn + r) where n is the block size and r is the number of blocks in a free list.

3. How to better realloc. At the very first, I only used the cases listed in the handout to realloc. But my perfomance was only 86, no matter how I changed the other part. Then, I suddenly realized that if I always split the block and used the first half of block, there would be external fragemnts. Then I tried a few threshold to determine whether I uesd the first half or the second half. Then my performance came to 90. Finally, after one day of modifying other parts of my code in vein, I realized that there must have been better way of realloc. After carafully debugging, I realized that sometimes I could simply extend the target block rather than copy it. In the end, my performance was 96.

2 What I Have Learned

1. Get a better knowledge of how the malloc, realloc and free works.

2. Get a better knowledge of how to use seglist to improve the performance and how to better use the memory.