

MALLOC ASSIGNMENT REPORT

EXECUTIVE SUMMARY:

This assignment was basically an implementation of malloc and free. There were counters to track statistics like number of successful malloc and free calls, reuse of blocks, splitting and coalescing of blocks. There were four heap management strategies: first fit, best fit, worst fit and next fit. There was also implementation of calloc and realloc along with four of my own tests. Those four tests were to check basic implementation of malloc and free besides the four that were already provided.

DESCRIPTION:

Algorithms implemented:

First Fit: It iterates through the linked list of free blocks until it finds a block that is both free and large enough to satisfy the requested size

Best Fit: It iterates through the linked list of free blocks(heapList) and keeps track of the block with the smallest size that is still large enough to satisfy the requested size(best_fit). Once it reaches the end of the list, it returns the best_fit block

Worst Fit: It iterates through the linked list of free blocks(heapList) and keeps track of the block with the largest size that is still large enough to satisfy the requested size(worst_fit). Once it reaches the end of the list, it returns the worst_fit block.

Next Fit: It starts the search from the last_alloc block (which is the last block that was allocated) and iterates through the linked list of free blocks (heapList) until it finds a block that is both free and large enough to satisfy the requested size. If it reaches the end of the list without finding a suitable block, it wraps around to the beginning of the list and continues the search from there.

TEST IMPLEMENTATION:

I wrote four simple tests to check basic implementation of malloc and free.

Test 1:

This test allocated memory for an integer using malloc() and stored 25 in it. It should print out 25, which is the allocated memory. It then freed the memory using free(). This test was a way to check if allocation and deallocation were working correctly for a single integer

Test 2:

This test was very similar to test 1, as it allocated 10 to an integer and prints out the value of the allocated memory. The only difference here was that it tried to access the value of the allocated memory after deallocating it which should not be possible. It was a way of checking if accessing memory after deallocation results in undefined behaviour.

Test 3:

This test allocated memory for an integer inside a for loop that iterated 5 times. It stored values in it and then freed it inside a loop. The test was to check if multiple memory allocations and deallocations within a loop would work correctly instead of resulting in memory leaks, leading to segfaults.

Test 4:

This test was very similar to test 2 and again checks if accessing memory after deallocation results in undefined behaviour.

TEST RESULTS:

mytest1	mallocs	frees	reuses	grows	splits	coalesces	blocks	requested	max heap
First fit	4	1	2	2	0	0	0	1028	1028
Best fit	4	1	2	2	0	0	0	1028	1028
Worst fit	4	1	2	2	0	0	0	1028	1028
Next fit	4	1	2	2	0	0	0	1028	1028
mytest2									
First fit	4	1	2	2	0	0	0	1044	1044
Best fit	4	1	2	2	0	0	0	1044	1044
Worst fit	4	1	2	2	0	0	0	1044	1044
Next fit	4	1	2	2	0	0	0	1044	1044
mytest3									
First fit	12	5	6	2	0	0	0	1044	1028
Best fit	12	5	6	2	0	0	0	1044	1028
Worst fit	12	5	6	4	2	3	0	1044	1036
Next fit	12	5	6	4	1	3	0	1044	1036
mytest4									
First fit	4	1	2	2	0	0	0	1028	1028
Best fit	4	1	2	2	0	0	0	1028	1028
Worst fit	4	1	2	2	0	0	0	1028	1028
Next fit	4	1	2	2	0	0	0	1028	1028

Real time(s)	Test
0.002	First fit
0.004	Next fit
0.003	Best fit
0.004	Worst fit
0.005	Test1
0.003	Test2
0.003	Test3
0.004	Test4

EXPLANATION, INTERPRETATION AND ANOMALIES:

My tests were able to see different variations of malloc and free being tested based on very similar scenarios. The time difference between the performances differed barely by a milli second but first fit and best fit seemed to perform faster compared to others. There were mainly issues with making my own tests. I got an assertion error for test 1 due to my malloc.c code which prompted me to change my test. I got segmentation faults for test 3 but I was eventually able to fix them.

CONCLUSION:

In conclusion, this assignment helped me see the difference in performance of different algorithms for heap management while also understanding different aspects of memory management.