Parth Sharma
1001954528

Malloc Assignment

Executive Summary/ Description

The memory allocator is a memory allocation algorithm that allows a programmer to dynamically allocate memory. In this assignment though, I have implemented a custom malloc implementation with different test cases and fits, mainly Best, Worst, Next and First fit with different test cases which test the splitting and coalescing of free blocks.

The custom implementation of malloc manages memory using a data structure such as linked list to keep track of allocated and free blocks. The data structure typically contains information about the size of each block, whether it is free or allocated, and a pointer to the next block in the list.

First Fit - Searches the heaplist for the first free block that is large enough to accommodate the requested size. The block is allocated to the user, with any remaining space in the block being marked as used.
Best Fit - Searches the heaplist for the free block that is closest in size to the requested size. This strategy reduces/minimises fragmentation by using the smallest possible free block.
Worst Fit - Searches the heaplist for the free block that is farthest in size to the requested size. This strategy maximises fragmentation by using the largest possible free block.
Next Fit - Similar to the first fit algorithm, but instead of starting at the start of the heaplist, it starts at the next free block in the list after the most recent allocation. This reduces time and fragmentation.
In the assignment I also split blocks when needed. This involves breaking up larger free blocks of memory into smaller ones to better fit the allocation requests. This reduces memory wastage and fragmentation and is a more efficient way of allocating memory.

When a block of memory is freed, it may leave a gap between adjacent free blocks. Over time, these gaps get fragmented. The approach I went through is where I loop through the heap and coalesce two or more adjacent blocks to reduce fragmentation.
The calloc function is used to allocate memory for an array of elements, however it initialises the allocated memory to all zeroes, while realloc is used to change the size of a previously allocated block of memory.

Parth Sharma
1001954528

Test Implementation

In addition to the provided tests, to further stretch the possibilities of the custom malloc I wrote four new tests.

stressMallocTest.c

- I stress test my overall malloc implementation. I malloc for 10,000 times and then free the allocated memory.

Test6.c

- I test my coalescing, while mallocing and freeing multiple times with same values, which results in a significant amount of num_coalesces.

Test7.c

- In this test I try to get to the max heap size possible.

Test8.c

- In this test I test splitting blocks.

Test Results

	System Malloc	Custom Malloc	Best Fit				Worst Fit			
	Performance	Performance	Performance	split/heap	Fragmentation	Max Heap	Performance	split/heap	Fragmentation	Max Heap
test1	0.001	0.006	0.002	0	No	66560	0.001	0	no	66560
test2	0.001	0.003	0	0	Yes	1115136	0.004	0.000001	yes	1115136
test3	0.001	0.002	0.002	0.000292	Yes	3424	0.001	0.000292	yes	3424
test4	0	0	0.003	0.000326	yes	3072	0.001	0.000326	yes	3072
stressMallocTest	0.001	0.009	0.008	0.000012	No	80000	0	0.000012	no	80000
test6	0.002	0.001	0	0.003597	yes	1112	0.003	0.003597	yes	1112
test7	0.001	0.001	0.007	0.000002	no	502000	4	0.000002	no	502000
test8	0	0.001	0.003	0.000032	yes	31024	0.002	0.000032	yes	31024

Next Fit				First Fit			
Performance	split/heap	Fragmentation	Max Heap	Performance	split/heap	Fragmentation	Max Heap
0.002	0	no	66560	0.003	0	no	66560
0.001	0.000001	yes	1115136	0	0.000001	yes	1115136
0.003	0.000292	yes	3424	0.001	0.000292	yes	3424
0.003	0.000326	yes	3072	0.003	0.000326	yes	3072
0.001	0	no	81024	0.012	0	no	81024
0.001	0.003597	yes	1112	0	0.003597	yes	1112
0	0.000002	no	502000	3	0.000002	no	502000
0	0.000032	yes	31024	0.002	0.000032	yes	31024`

Discussion

These test results show that system malloc call is faster than the custom malloc implementation. Although I cannot get the fragmentation information of the system malloc, my coalescing and splitting the blocks prove that fragmentation does not exist in the custom implementation.

These test results also show that best and worst fit take the most time.

Conclusion

From my tests and writing the custom implementation of malloc, I think next fit is the most useful out of everything.