REPORT

Malloc

Hyun Ho Kim 1001967176
CSE 3320 003 OPERATING SYSTEMS

- Executive summary
  - This program allows users to allocate memory blocks according to their certain requests. It offers four different allocation methods, including first fit, best fit, worst fit, and next fit. Users can free any blocks they have allocated, and can also re-allocate a block using the 'realloc' function, or allocate multiple blocks using the 'calloc' function.

- Description of the algorithms implemented.
  1) Frist fit
     - When memory is requested, the first fit algorithm finds a free block that fits the requested size from the beginning of the heap list until the end of it. If it finds a suitable block, it returns its location. Otherwise, it grows the heap list.
  2) Best fit
     - When memory is requested, the best fit algorithm finds a free block that is the best fit for the requested size. In this code, it saves and tracks the location where the difference between a block's size and the requested size is the smallest positive value while traveling through the heap. If it finds a suitable block, it returns its location. Otherwise, it grows the heap list.
  3) Worst fit
     - When memory is requested, the worst fit algorithm finds the largest free block available. In this code, it saves and tracks the location where the difference between a block's size and the requested size is the largest positive value while traveling through the heap. If it finds a suitable block, it returns its location. Otherwise, it grows the heap list.
  4) Next fit
     - When memory is requested, the next fit algorithm finds a free block that fits the requested size starting from the last added block until a tracker returns to the starting point. If the tracker reaches the end of the heap list, it starts again from the beginning to find a block. If it finds a suitable block, it returns its location. Otherwise, it grows the heap list.

- Test implementation
  1) Test 5 is designed to test reusability
     - It grows a heap with a large size, and then allocates smaller blocks of memory.
  2) Test 6 is designed to test whether the program can reuse memory allocated by the 'calloc' function
     - It grows a heap using 'calloc', frees it, and then allocates several blocks of memory.
  3) Test 7 is designed to test whether the program can coalesce memory blocks allocated by 'calloc'

- It grows a heap using two 'calloc' function calls, frees them, and then checks if they have been coalesced.

4) Test 8 is designed to test the functionality of the 'realloc' and 'malloc' functions
   - It grows a heap using 'malloc' first, and then calls the 'realloc' function.
5) Test 9 is designed to test the program's handling of zero-size allocations
   - It grows a heap with zero size and then frees it.

- Test results for all five candidates (malloc and your four algorithm implementations)

Test #	Allocation method	My function	Built-in function
5	First fit	real 0m0.007s user 0m0.005s sys 0m0.002s	real 0m0.003s user 0m0.003s sys 0m0.000s
5	Next fit	real 0m0.006s user 0m0.003s sys 0m0.002s	
5	Worst fit	real 0m0.007s user 0m0.007s sys 0m0.000s	
5	Best fit	real 0m0.007s user 0m0.002s sys 0m0.005s	
6	First fit	real 0m0.008s user 0m0.004s sys 0m0.000s	real 0m0.003s user 0m0.003s sys 0m0.000s
6	Next fit	real 0m0.004s user 0m0.003s sys 0m0.001s	
6	Worst fit	real 0m0.004s user 0m0.004s sys 0m0.000s	
6	Best fit	real 0m0.009s user 0m0.005s sys 0m0.000s	
7	First fit	real 0m0.008s user 0m0.004s sys 0m0.000s	real 0m0.003s user 0m0.001s sys 0m0.002s
7	Next fit	real 0m0.004s user 0m0.003s sys 0m0.002s	
7	Worst fit	real 0m0.007s user 0m0.002s sys 0m0.002s	
7	Best fit	real 0m0.006s user 0m0.004s sys 0m0.000s	
8	First fit	real 0m0.004s	real 0m0.003s

		user 0m0.002s sys 0m0.002s	user 0m0.003s sys 0m0.000s
8	Next fit	real 0m0.006s user 0m0.002s sys 0m0.002s	
8	Worst fit	real 0m0.004s user 0m0.004s sys 0m0.000s	
8	Best fit	real 0m0.004s user 0m0.004s sys 0m0.000s	
9	First fit	real 0m0.006s user 0m0.004s sys 0m0.000s	real 0m0.003s user 0m0.002s sys 0m0.001s
9	Next fit	real 0m0.006s user 0m0.004s sys 0m0.000s	
9	Worst fit	real 0m0.007s user 0m0.002s sys 0m0.002s	
9	Best fit	real 0m0.004s user 0m0.003s sys 0m0.001s	

- Explanation and interpretation of the results including any anomalies in the test results.
  The built-in function executed at a faster speed than the algorithms I implemented. Perhaps the built-in function has a better algorithm. The difference in execution speed depending on the allocation method varies depending on the test, but generally, Next fit showed a faster execution rate. The reason for this result is that Next fit remembers the location of the last added block, so there was no need to travel from the beginning of the heap list.

- Conclusion.
  - Through this project, we executed the algorithms for four allocation methods and tested their execution speeds. We found free blocks using first fit, next fit, worst fit, and best fit methods. Additionally, we gained insight into how our heap divides memory and how freed blocks are combined during program execution. We also compared the built-in functions for malloc, realloc, calloc, and free with our algorithms and measured their speeds. In conclusion, the built-in functions were significantly faster than our algorithms. And in my experiments, next fit showed a faster execution speed compared to other algorithms.