DECS Assignment: 3
Memory Allocation and Management

Mitul Tyagi, Roll no. : 193079017
Tarun Saurabh, Roll no. : 203050009

18th October 2020

1 Introduction

This assignment implements a memory allocator using various dynamic memory allocation techniques. There are two C implementations namely, mm1.c and mm2.c. Each of these files contains four main functions: mm_init(), mm_malloc(), mm_realloc() and mm_free().

- mm_init() :- It performs the initialization, like resetting the heap, setting all the data structures and variables to the staring values.
- mm_malloc() :- It returns a pointer to an allocated block, with the payload of at-least passed size bytes.
- mm_realloc() :- It accepts a pointer to an already allocated chunk and the new size to be allocated while maintaining the user data.
- mm_free() :- It frees the block pointed by ptr.

1.1 mm1.c

This implementation uses an **Implicit Free List** which links all the chunks, whether allocated or free. Each chunk has two metadata blocks on either side,*header and footer*, for both ways *coalescing*. The algorithms implemented, are *Best-Fit*, *Worst-Fit* and *First-Fit* to obtain the free chunk during the malloc call. The data structure used to implement the free list is *Linked-List*. The *mm_realloc()* function is also optimized in which if the next block is free and big enough then it is used.

Some of the important functions are: *get_free_block()* to obtain the best-fit free block, *alloc_space()* to allocate a new chunk from the heap and *coalesce()* to merge the free blocks.

1.2 mm2.c

This implementation uses a **Modified AVL Tree** which maintains all the free blocks in a *Balanced Binary Search Tree*. Each chunk has two metadata blocks on either side, *header and footer*, for both ways *coalescing*. *Best-Fit* allocation algorithm is implemented to obtain the free block from the tree during the malloc call, and then the corresponding node is deleted from the tree.

Some of the important functions are: *insert()* to insert the free block in the tree, *alloc_space()* to allocate a new chunk from the heap, *coalesce()* to merge the free blocks and, *findNode_InOrder()* to find the best-fit chunk in the tree and *deleteNode* to remove the chunk from the tree. The *mm_realloc()* function is also optimized in which if the next block is free and big enough then it is used.

During the *malloc call* the free-tree is searched and if the free chunk size is greater than or equal to the required size to be allocated, then that chunk is allocated and the corresponding node is deleted from the tree. Otherwise the heap size is increased and the new chunk is allocated.

During the *free call*, the freed chunk is inserted into the tree. Two way coalescing is performed for the inserted free chunk.

2 Results and Outcomes

Both the implementations were tested on various trace files provided. The results are show below.

2.1 Results for mm1.c with Best-fit

```
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$ ./mdriver1 -v -f traces/short1-bal.rep
Team Name:MiSau
Member 1 :Mitul Tyagi:mitulty@ee.iitb.ac.in
Member 2 :Tarun Saurabh:tarunsaurabh@cse.iitb.ac.in
Measuring performance with gettimeofday().

Results for mm malloc:
trace  valid  util     ops      secs      Kops
 0       yes   99%      12  0.000002      5000
Total          99%      12  0.000002      5000

Perf index = 60 (util) + 40 (thru) = 100/100
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$
```

Figure 1: short1-bal.rep-mm1.c

```
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$ ./mdriver1 -v -f traces/short2-bal.rep
Team Name:MiSau
Member 1 :Mitul Tyagi:mitulty@ee.iitb.ac.in
Member 2 :Tarun Saurabh:tarunsaurabh@cse.iitb.ac.in
Measuring performance with gettimeofday().

Results for mm malloc:
trace  valid  util     ops      secs      Kops
 0       yes   99%      12  0.000002      7500
Total          99%      12  0.000002      7500

Perf index = 60 (util) + 40 (thru) = 100/100
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$
```

Figure 2: short2-bal.rep-mm1.c

```
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$ ./mdriver1 -v -f traces/short3-bal.rep
Team Name:MiSau
Member 1 :Mitul Tyagi:mitulty@ee.iitb.ac.in
Member 2 :Tarun Saurabh:tarunsaurabh@cse.iitb.ac.in
Measuring performance with gettimeofday().

Results for mm malloc:
trace  valid  util     ops      secs      Kops
 0       yes   97%      10  0.000001      8333
Total          97%      10  0.000001      8333

Perf index = 58 (util) + 40 (thru) = 98/100
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$
```

Figure 3: short3-bal.rep-mm1.c

```
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$ ./mdriver1 -v -f traces/cp-decl-bal.rep
Team Name:MiSau
Member 1 :Mitul Tyagi:mitulty@ee.iitb.ac.in
Member 2 :Tarun Saurabh:tarunsaurabh@cse.iitb.ac.in
Measuring performance with gettimeofday().

Results for mm malloc:
trace  valid  util     ops      secs      Kops
 0       yes   39%    6648  0.020377       326
Total          39%    6648  0.020377       326

Perf index = 23 (util) + 22 (thru) = 45/100
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$
```

Figure 4: cp-decl-bal.rep-mm1.c

```
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$ ./mdriver1 -v -f traces/random-bal.rep
Team Name:MiSau
Member 1 :Mitul Tyagi:mitulty@ee.iitb.ac.in
Member 2 :Tarun Saurabh:tarunsaurabh@cse.iitb.ac.in
Measuring performance with gettimeofday().

Results for mm malloc:
trace  valid  util     ops      secs      Kops
 0       yes   85%    4800  0.015953       301
Total          85%    4800  0.015953       301

Perf index = 51 (util) + 20 (thru) = 71/100
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$
```

Figure 5: random-bal.rep-mm1.c

```
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$ ./mdriver1 -v -f traces/binary-bal.rep
Team Name:MiSau
Member 1 :Mitul Tyagi:mitulty@ee.iitb.ac.in
Member 2 :Tarun Saurabh:tarunsaurabh@cse.iitb.ac.in
Measuring performance with gettimeofday().

Results for mm malloc:
trace  valid  util     ops      secs      Kops
 0       yes   54%   12000  0.175264        68
Total          54%   12000  0.175264        68

Perf index = 32 (util) + 5 (thru) = 37/100
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$
```

Figure 6: binary-bal.rep-mm1.c

```
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$ ./mdriver1 -v -f traces/realloc-bal.rep
Team Name:MiSau
Member 1 :Mitul Tyagi:mitulty@ee.iitb.ac.in
Member 2 :Tarun Saurabh:tarunsaurabh@cse.iitb.ac.in
Measuring performance with gettimeofday().
^[[24;2~
Results for mm malloc:
trace  valid  util     ops      secs      Kops
 0       yes   69%   14401  0.000160     90288
Total          69%   14401  0.000160     90288

Perf index = 42 (util) + 40 (thru) = 82/100
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$
```

Figure 7: realloc-bal.rep-mm1.c

2.2 Results for mm2.c with Best-Fit

```
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$ ./mdriver2 -v -f traces/short1-bal.rep
Team Name:MiSau
Member 1 :Mitul Tyagi:mitulty@ee.iitb.ac.in
Member 2 :Tarun Saurabh:tarunsaurabh@cse.iitb.ac.in
Measuring performance with gettimeofday().

Results for mm malloc:
trace  valid  util     ops      secs      Kops
 0       yes   99%      12  0.000001     10909
Total          99%      12  0.000001     10909

Perf index = 60 (util) + 40 (thru) = 100/100
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$
```

Figure 8: short1-bal.rep-mm2.c

```
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$ ./mdriver2 -v -f traces/short2-bal.rep
Team Name:MiSau
Member 1 :Mitul Tyagi:mitulty@ee.iitb.ac.in
Member 2 :Tarun Saurabh:tarunsaurabh@cse.iitb.ac.in
Measuring performance with gettimeofday().

Results for mm malloc:
trace  valid  util     ops      secs      Kops
 0       yes   99%      12  0.000001      9231
Total          99%      12  0.000001      9231

Perf index = 60 (util) + 40 (thru) = 100/100
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$
```

Figure 9: short2-bal.rep-mm2.c

```
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$ ./mdriver2 -v -f traces/short3-bal.rep
Team Name:MiSau
Member 1 :Mitul Tyagi:mitulty@ee.iitb.ac.in
Member 2 :Tarun Saurabh:tarunsaurabh@cse.iitb.ac.in
Measuring performance with gettimeofday().

Results for mm malloc:
trace  valid  util     ops      secs      Kops
 0       yes   97%      10  0.000001     10000
Total          97%      10  0.000001     10000

Perf index = 58 (util) + 40 (thru) = 98/100
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$
```

Figure 10: short3-bal.rep-mm2.c

```
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$ ./mdriver2 -v -f traces/cp-decl-bal.rep
Team Name:MiSau
Member 1 :Mitul Tyagi:mitulty@ee.iitb.ac.in
Member 2 :Tarun Saurabh:tarunsaurabh@cse.iitb.ac.in
Measuring performance with gettimeofday().

Results for mm malloc:
trace  valid  util     ops      secs      Kops
 0       yes   37%    6648  0.016032       415
Total          37%    6648  0.016032       415

Perf index = 22 (util) + 28 (thru) = 50/100
```

Figure 11: cp-decl-bal.rep-mm2.c

```
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$ ./mdriver2 -v -f traces/random-bal.rep
Team Name:MiSau
Member 1 :Mitul Tyagi:mitulty@ee.iitb.ac.in
Member 2 :Tarun Saurabh:tarunsaurabh@cse.iitb.ac.in
Measuring performance with gettimeofday().

Results for mm malloc:
trace  valid  util     ops      secs      Kops
 0       yes   76%    4800  0.006903       695
Total          76%    4800  0.006903       695

Perf index = 46 (util) + 40 (thru) = 86/100
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$
```

Figure 12: random-bal.rep-mm2.c

```
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$ ./mdriver2 -v -f traces/binary-bal.rep
Team Name:MiSau
Member 1 :Mitul Tyagi:mitulty@ee.iitb.ac.in
Member 2 :Tarun Saurabh:tarunsaurabh@cse.iitb.ac.in
Measuring performance with gettimeofday().

Results for mm malloc:
trace  valid  util     ops      secs      Kops
 0       yes   54%   12000  0.219557        55
Total          54%   12000  0.219557        55

Perf index = 32 (util) + 4 (thru) = 36/100
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$
```

Figure 13: binary-bal.rep-mm2.c

```
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$ ./mdriver2 -v -f traces/realloc-bal.rep
Team Name:MiSau
Member 1 :Mitul Tyagi:mitulty@ee.iitb.ac.in
Member 2 :Tarun Saurabh:tarunsaurabh@cse.iitb.ac.in
Measuring performance with gettimeofday().

Results for mm malloc:
trace  valid  util     ops      secs      Kops
 0       yes   46%   14401  0.000214     67169
Total          46%   14401  0.000214     67169

Perf index = 27 (util) + 40 (thru) = 67/100
morack@morack-Lenovo-IdeaPad-S145-15IWL:~/PA_3_handout$
```

Figure 14: realloc-bal.rep-mm2.c

3 Pseudo-Algorithms for mm2.c

```
def malloc(size):
    Search in tree for best fit size
    if found the then:
        delete from tree (perform balancing if required)
    else :
        increase heap size and create new chunk

def free(ptr):
    Go to the location and mark it free
    Insert node into tree (perform balancing if required)

def realloc(ptr,size):
    if next_chunk free and big enough:
        increase the size
        return
    else:
        allocate a new chunk and copy the data
        return
```

4 Conclusion

Time complexity for mm1.c is $\mathcal{O}(2n^2)$ and for mm2.c is $\mathcal{O}(n^2 + n \log n)$

5 Bibliography

References

[1] Rezaei,Mehran and Kavi,Krishna(2000).A New Implementation Technique for Memory Management. Proceedings of the 2000 SoutheastCon, Nashville, TN,April, 2000.