COSC1112/1114 Operating Systems Principles Assignment 1 REPORT

Due Date: 9PM 6th of September 2019

Student Number: s3701799

Student Name: Syed Hariz bin Syed Azmi

Table of Contents

Introduction

Modern computer systems must adapt to important requirements such as efficient memory management, resource management, implementation process, great inter process management etc. The main focus above all, is the efficiency of memory management. Many research are being carried out to improve the way memory is being allocated for an application. The main problem faced by a memory allocation algorithm is to efficiently allocate requested memory blocks with minimum response time along with minimum memory loss. A traditional memory loss problem called fragmentation of memory which keeping the reference to those blocks that has been allocated and to the blocks which are free to be allocated during the next demand (1). Providing a memory block for an application is not enough, the efficiency of real time systems rely on the availability of this memory blocks with minimum fragmentation in timely manner.

In the presence of different memory management techniques, this experiment focuses on the algorithms of Dynamic Memory Allocation (DMA). Dynamic Memory Allocation is sometimes called as Manual Memory Management. In Dynamic Memory Allocation, the memory is located at run time. It is allocated whenever a program or application demands with requires amount of bytes (2). However, this experiment focuses on 3 different memory allocation techniques from DMA; First fit, Best fit and Worst fit.

The purpose of implementing this experiment is to conduct findings between those 3 described allocation strategies based on its time performance and allocation of data. The manual memory allocation program is being presented using C++ language. Reason of this choice is to show allocation and de-allocation of a memory by developing functions called alloc() and dealloc() as required. 3 mentioned algorithms were created according to its own requisite functionalities;

First fit

The first chunk that is big enough, split and return to the user a chunk that is big enough to meet their needs. Add the rest of the chunk back onto the free list.

Best fit

The chunk whose size is the closest match to the allocation need. Often that will be exactly the size required by the client no split is needed.

Worst fit

Find the largest chunk and split off the part needed and store that in the allocated list. Store the rest of the chunk back on the free list.

Further findings are being elaborated through-out this report to extend further understanding and differentiate between those 3 constructed memory allocation strategies. 3 of those algorithms are being tested with 2 different kinds of data which varies between type of data in the file and size of the file to collect meaningful statistics between them. Two linked-list are being used to manage memory; 1st linked-list is for memory allocation and 2nd linked-list for chunks that have been freed.

Experiment description & results

Experiments that are being carried out uses a separate source file from the program. Therefore, the program reads a file containing a large chunk of data and store them in a list before allocating them in the memory. When the program starts user can choose running the First fit, Best fit or Worst fit algorithm to allocate the data. The program then runs the chosen algorithm and displays allocated bytes and freed bytes together with old program edge and new program edge to indicate that data are being allocated in the memory.

Experiment 1 – Random names

Name	**experiment1**
Type	.txt
Description	A file that consists a large chunk of random names. However, only first names are being presented in the file.
File size	175 KB
Data volume	21985
Results First fit	Old Program Edge: 0x7fffc3320000 New Program Edge: 0x7fffc3345fd0 Allocated bytes: 6650 Freed Bytes: 400 *Refer to Appendices: Appendix A under experiment1.txt section for application view*
Results Best fit	Old Program Edge: 0x7fffea899000 New Program Edge: 0x7fffea8c1de1 Allocated bytes: 7035 Freed Bytes: 1995 *Refer to Appendices: Appendix A under experiment1.txt section for application view*
Results Worst fit	Old Program Edge: 0x7fffbfd43000 New Program Edge: 0x7fffbfd6c300 Allocated bytes: 7316 Freed Bytes: 3213 *Refer to Appendices: Appendix A under experiment1.txt section for application view*

Name	**experiment1(half)**
Type	.txt
Description	A file that consists a large chunk of random names. However, only first names are being presented in the file. In this text file, the total volume of the original volume (experiment1.txt) is split into half.
File size	87.7 KB
Data volume	10993
Results First fit	Old Program Edge: 0x7ffff099a000 New Program Edge: 0x7ffff09bfddc Allocated bytes: 6637 Freed Bytes: 415 *Refer to Appendices: Appendix A under experiment1(half).txt section for application view*
Results Best fit	Old Program Edge: 0x7fffbd9e6000 New Program Edge: 0x7fffbda0ede3 Allocated bytes: 7136 Freed Bytes: 1925 *Refer to Appendices: Appendix A under experiment1(half).txt section for application view*
Results Worst fit	Old Program Edge: 0x7fffd0419000 New Program Edge: 0x7fffd04423d2 Allocated bytes: 7446 Freed Bytes: 3292 *Refer to Appendices: Appendix A under experiment1(half).txt section for application view*

Experiment 2 – Random numbers

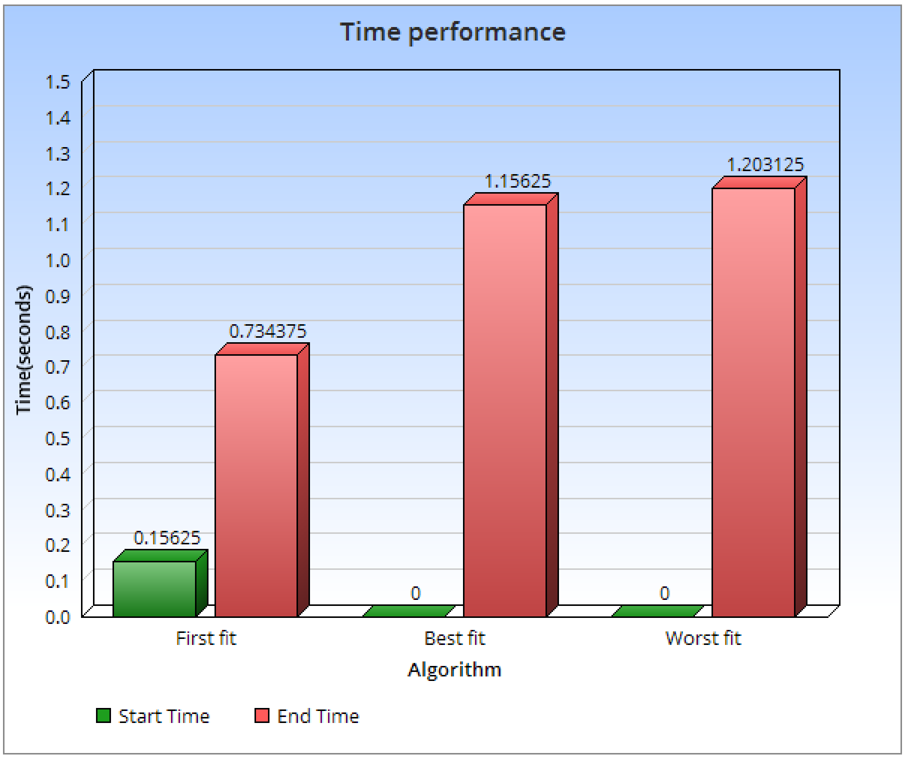
Name	**experiment2**
Type	.txt
Description	A file that consists a large chunk of random numbers. Odds and even numbers are included in the text file. Numbers are from 1 to 10000.
File size	57.5 KB
Data volume	10000
Results First fit	Old Program Edge: 0x7fffe4d87000 New Program Edge: 0x7fffc51f820c Allocated bytes: 4742 Freed Bytes: 76 *Refer to Appendices: Appendix B under experiment2.txt for application view*
Results Best fit	Old Program Edge: 0x7fffe4d0b000 New Program Edge: 0x7fffe4d346bf Allocated bytes: 4794 Freed Bytes: 4456 *Refer to Appendices: Appendix B under experiment2.txt for application view*
Results Worst fit	Old Program Edge: 0x7fffe2501000 New Program Edge: 0x7fffe252a6a0 Allocated bytes: 4800 Freed Bytes: 4436 *Refer to Appendices: Appendix B under experiment2.txt for application view*

Analysis

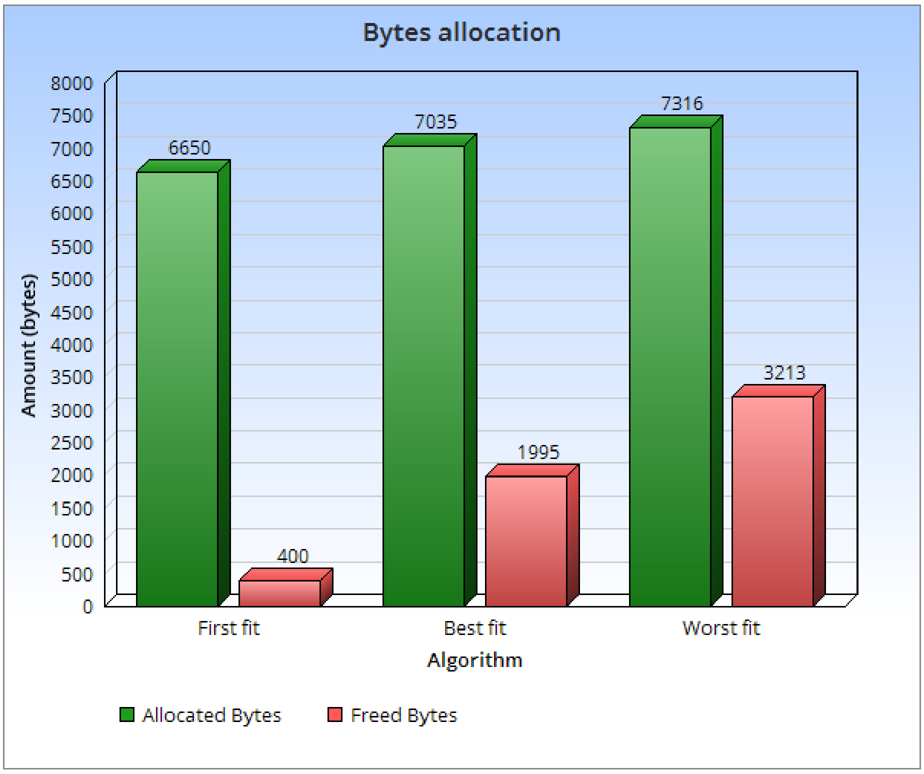
In this section of the report, graphs are being represented to illustrate the time taken for all 3 allocation strategies to process its algorithms according to the experimental file that was used. In the program, a simple getrusage function was added to examine the resource usage of a process and measure the time that was needed for each algorithm to complete its allocation task. The time are also being displayed in the application which can be viewed under Appendices section for reference.

The graphs are being utilized to dictate performance of 3 described algorithms where comparisons are being made based on results that were prompted in the application. Further discussion will be clarified to oversee detailed comparison between them.

experiment1.txt

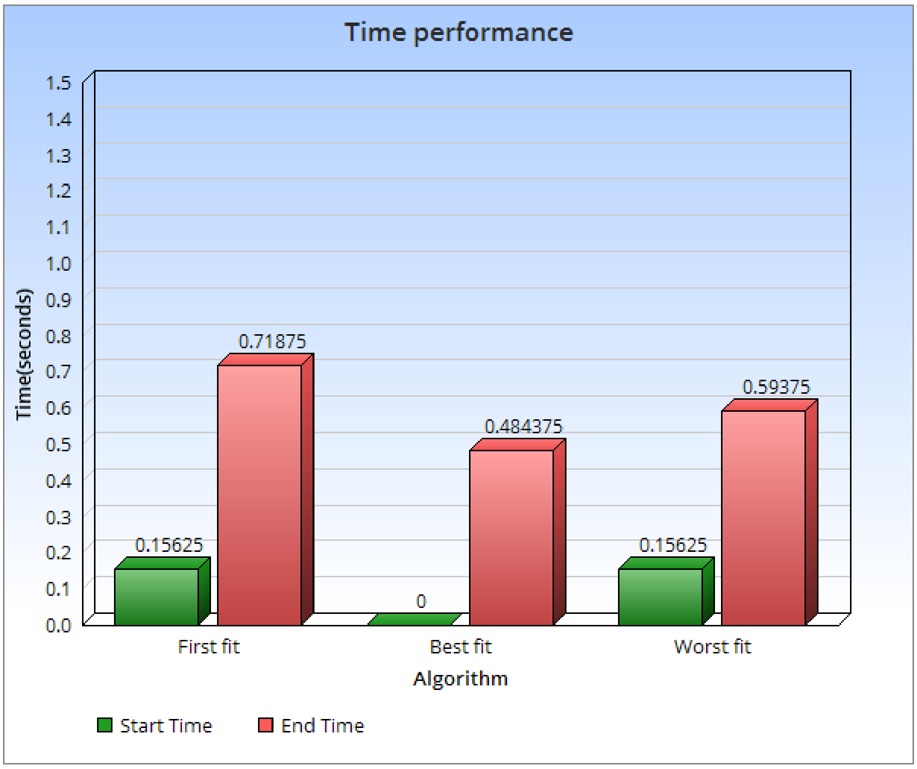


Graph 1: Time performance between 3 algorithms

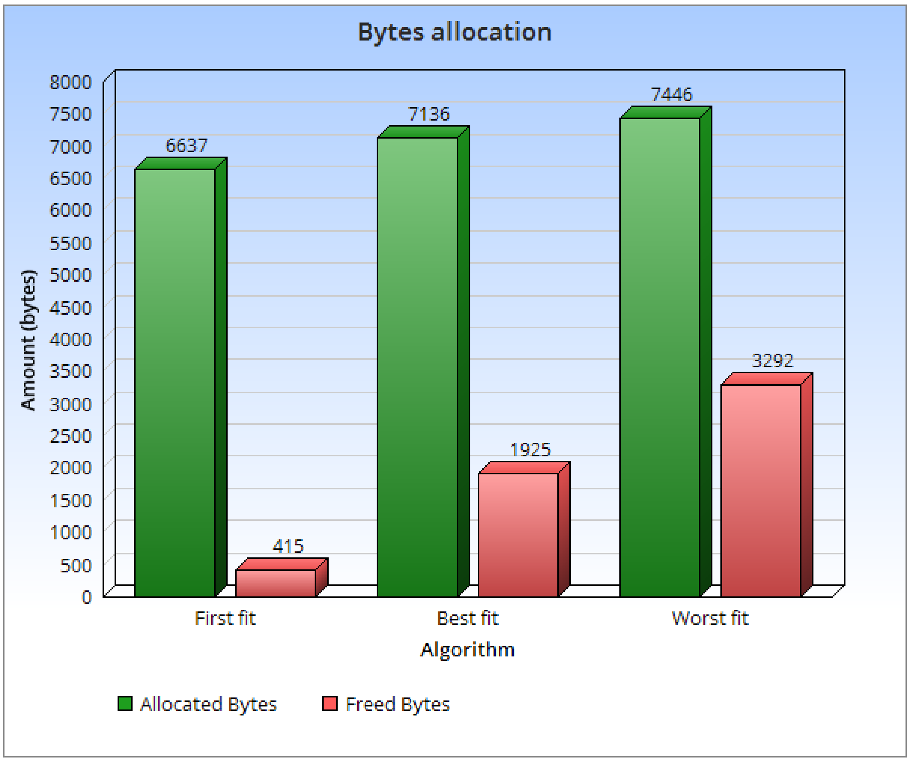


Graph 1.1: Bytes allocation between 3 algorithms

experiment1(half).txt

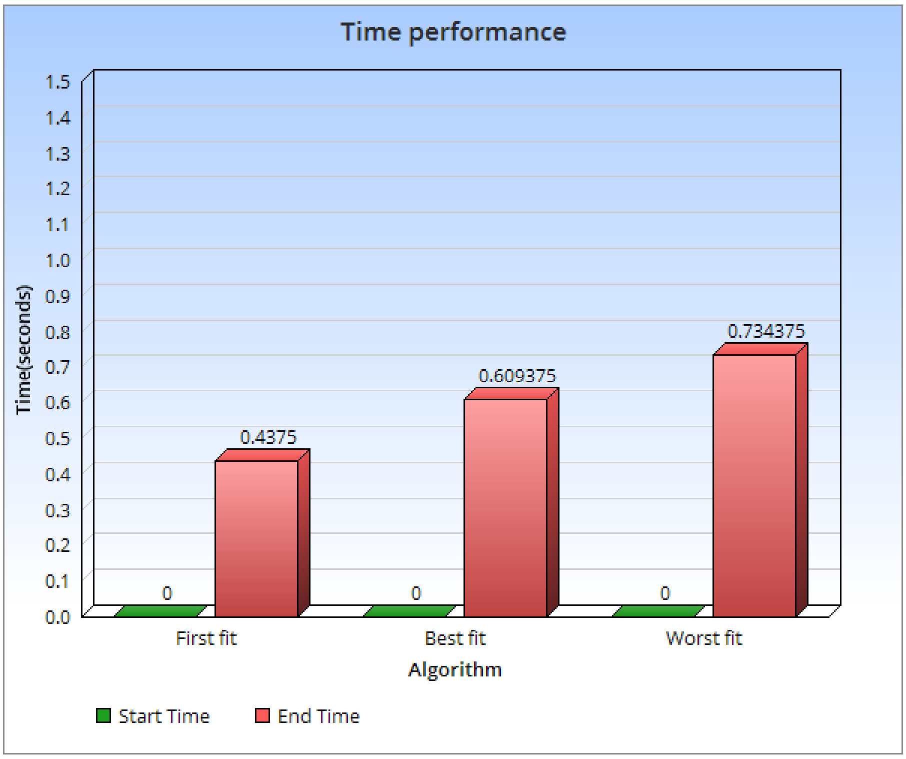


Graph 2: Time performance between 3 algorithms

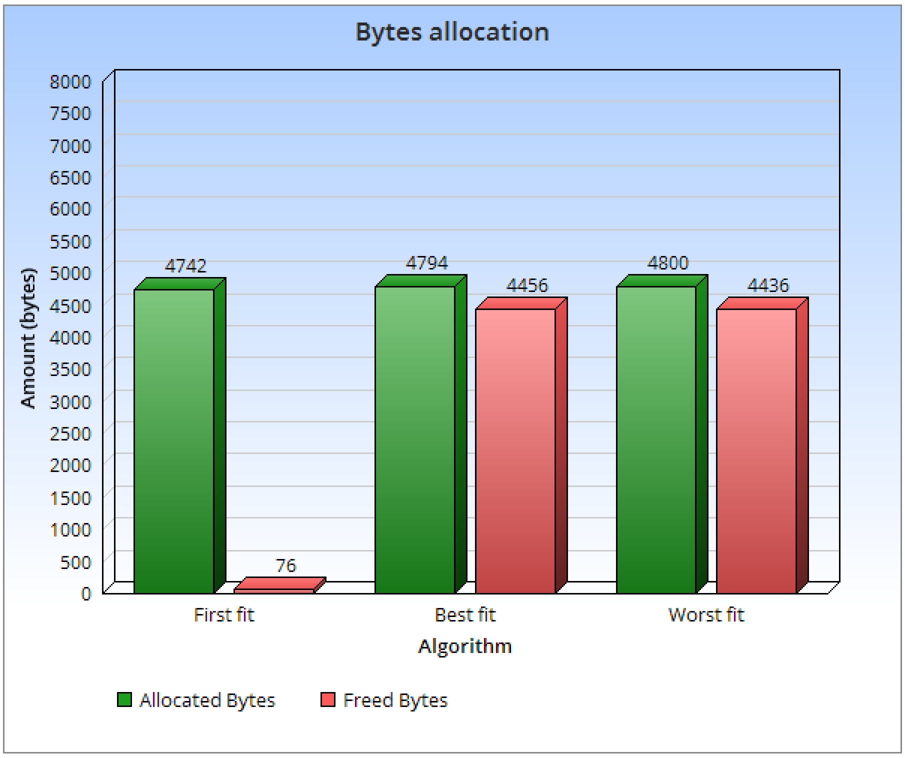


Graph 2.1: Bytes allocation between 3 algorithms

experiment2.txt



Graph 3: Time performance between 3 algorithms



Graph 3.1: Bytes allocation between 3 algorithms

Discussion

Based on information and statistics that were collected, we can easily compare between the 3 algorithms to present the best algorithm for memory allocation. Focusing on Time Performance between the 3 algorithms with 3 different experiment files and contrary data volume, we can conclude that the First Fit algorithm has the most efficient performance. Referring to Graph 1 and Graph 3, the First Fit ends its process much faster than the other 2 algorithms. Particularly in Graph 1, it took 22% less time of Worst Fit algorithm for First Fit algorithm to finish its process and 25% less than Best Fit algorithm. Although, when testing the algorithm with half the size of experiment1.txt, First Fit algorithm falls down to 2nd place behind Worst Fit algorithm. This shows that the First Fit algorithm is inconsistent with its process performance. Another downfall with First Fit algorithm detected is, it takes an extra amount of time for its process to get started as shown in its Start Time in Graph 1 and Graph 2. When testing with experiment2.txt with a data volume of 10000, no Start Time delay is needed for First Fit algorithm. We can deduce that this algorithm needs an extra time to start when processing large amount of data volume.

The second comparison is how many bytes are being allocated and freed by 3 of the described algorithms. Based on the graphs that were plotted, it is shown that the Worst Fit algorithm needs more bytes than the other 2 algorithm when allocating data. First fit algorithms freed less bytes between Best Fit and Worst Fit algorithm. First fit algorithm also uses less amount of memory when allocating data as shown on Graph 1.1,2.2 and 3.3. In memory management, it is important for an algorithm to use less amount of memory space and simultaneously process memory allocation fast.

Recommendation

As mentioned, the importance of memory management efficiency has been broached many times. The main problem faced by a memory allocation algorithm is to efficiently allocate requested memory blocks with minimum response time. There are some disadvantages that can be expressed with Dynamic Memory Allocation method. One of the problem is, memory leak. Memory leak is a condition in which some application continuously allocate memory without giving it up and finally meet a dead end, where no memories are available to give out to (2). Another disadvantage can be discussed is time consuming. Comparing to static memory management, dynamic memory allocation consumes more time (2). For personal recommendation, there are many more techniques that can be used for memory allocation such as Hoard or two level sequential fit. With comparative analysis, it is found that Hoard and two level sequential fit has faster response time and minimum amount of fragmented memory (1) which is ideal for real time applications.

Conclusion

This report has represented the role of Dynamic Memory Allocation in memory management. Precisely focused on experimenting 3 algorithms from DMA which are First fit, Best fit and Worst fit. A discussion has been raised to which algorithm is the best memory allocation strategy amongst them, from making statistic comparisons between them and listing down the pros and cons of the best algorithm. A recommendation has been made on how to improve memory management with the help of research papers that were published. Finally, we can conclude that dynamic memory allocation is an art of handling computer memory with some complexity cost.

Appendices

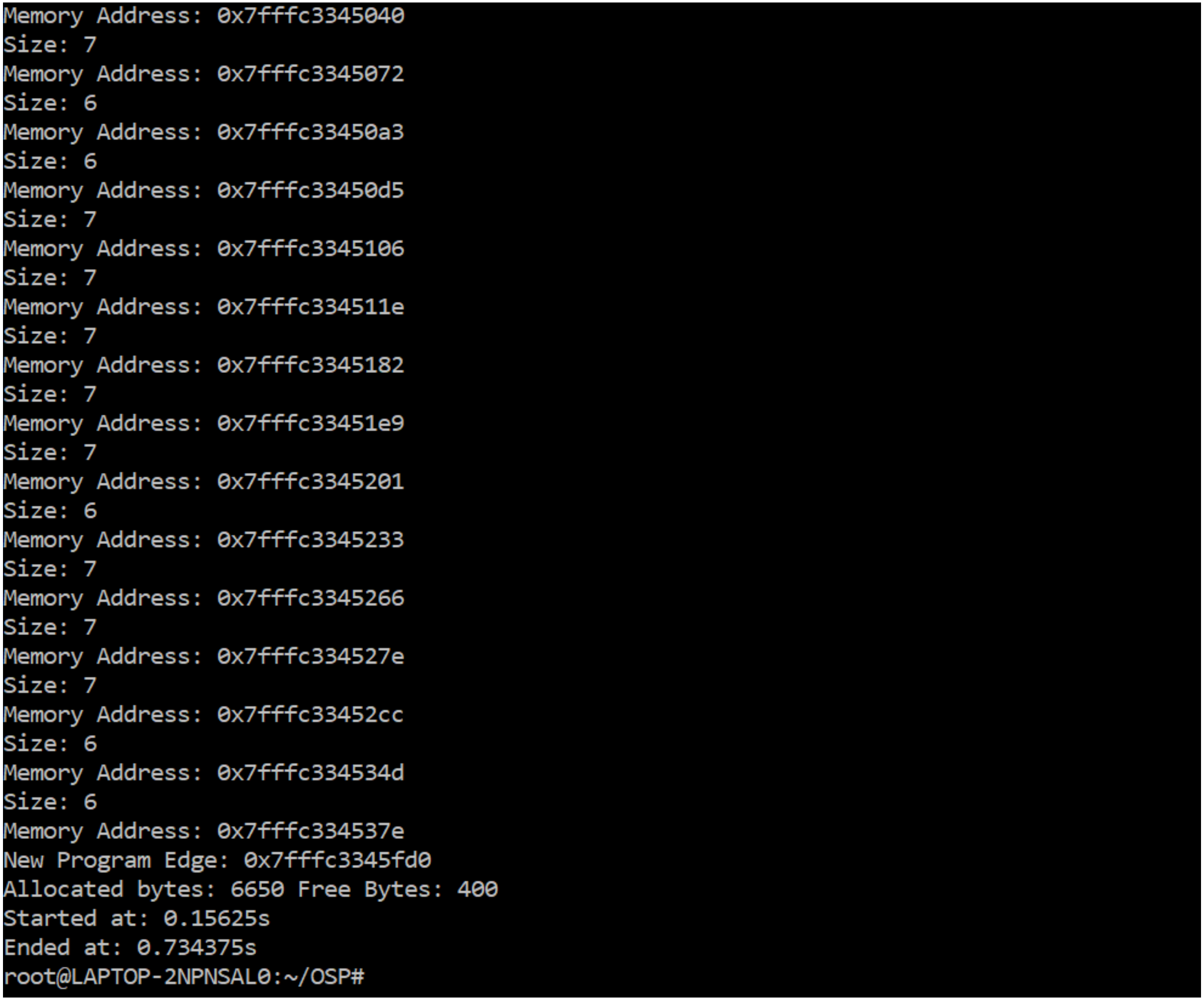
Application views of all 3 memory allocation strategies according to experiment files.

Google Drive access to experiment text files:
https://drive.google.com/drive/folders/1g29I0olMcpPVN-W3pVq3Xi4Yl1Vv-D9z?usp=sharing

Appendix A

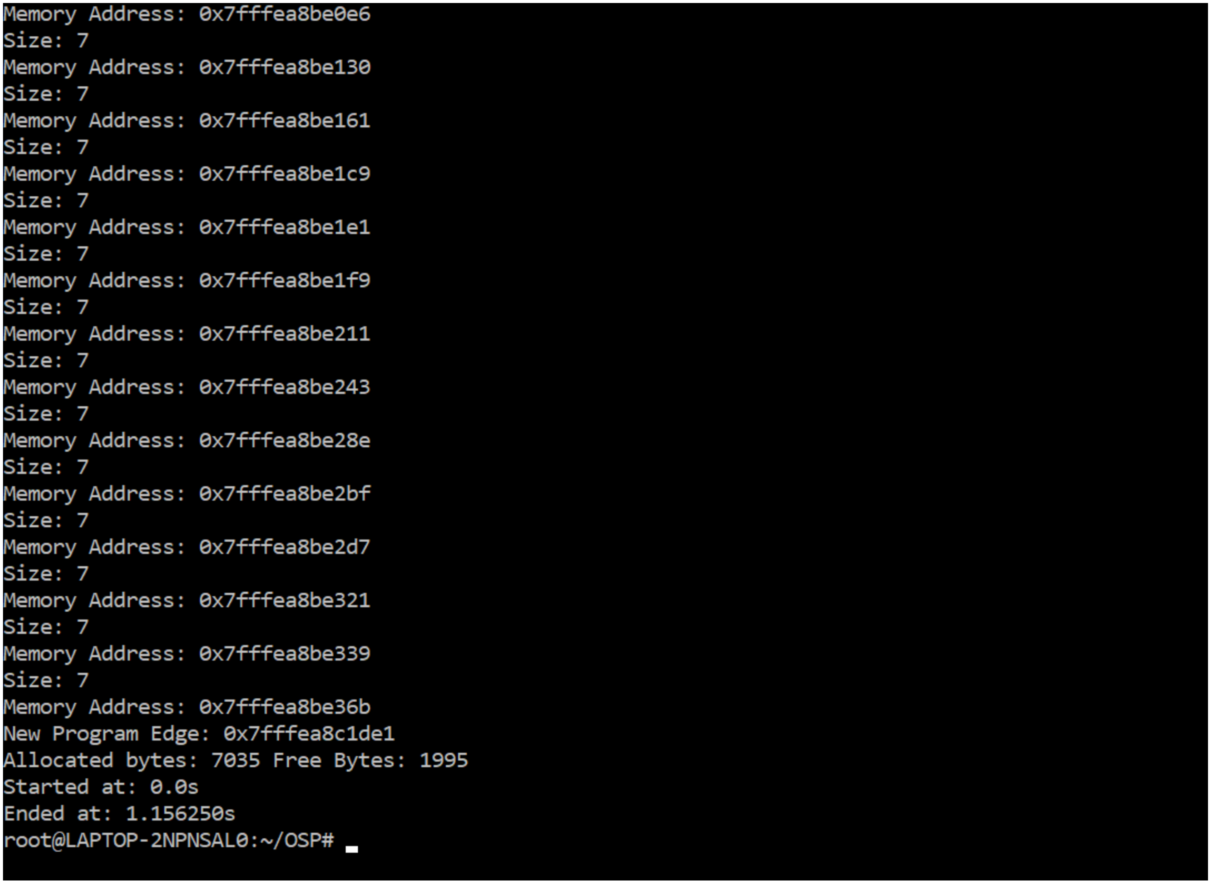
experiment1.txt

First fit



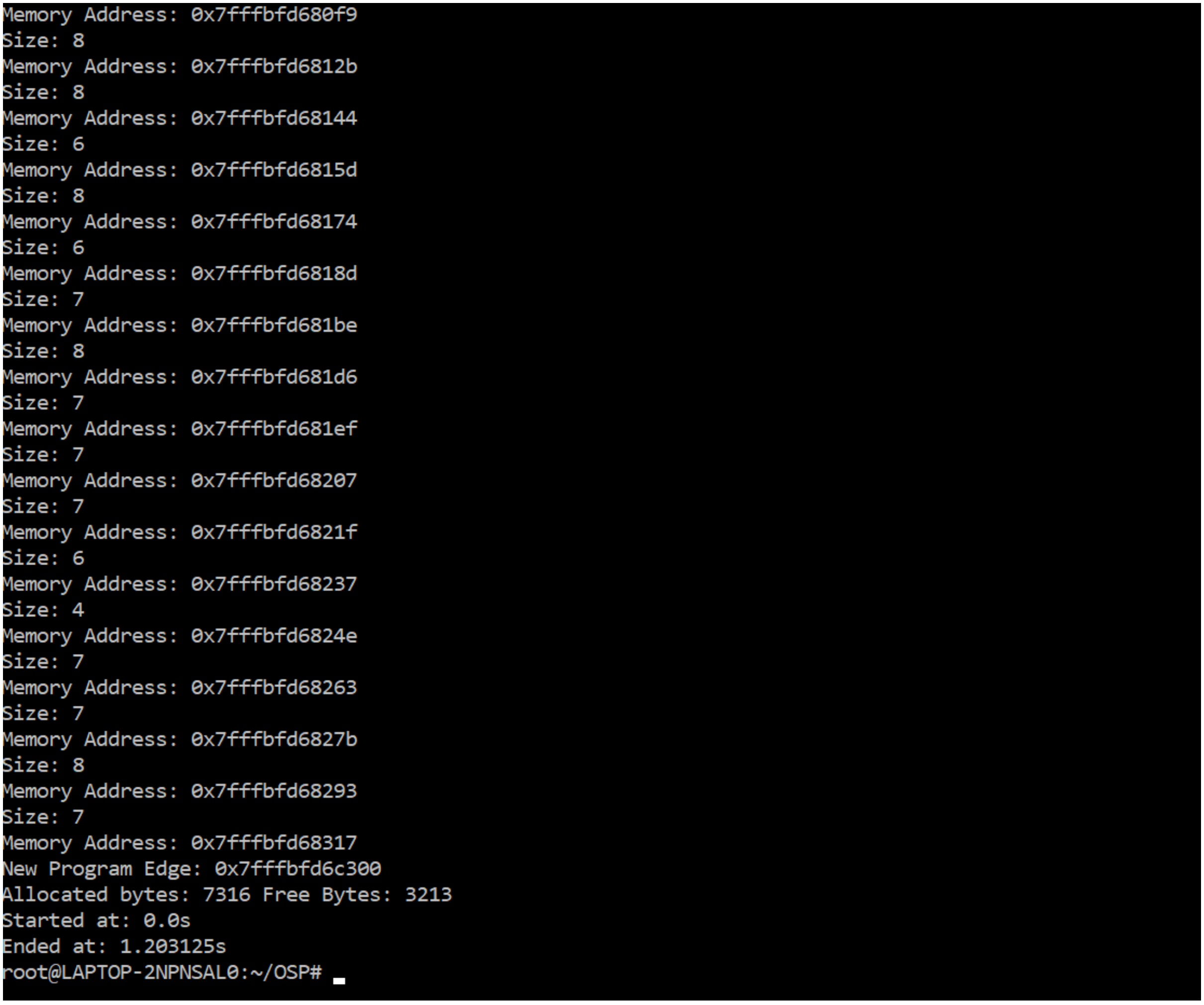
```
Memory Address: 0x7fffc3345040
Size: 7
Memory Address: 0x7fffc3345072
Size: 6
Memory Address: 0x7fffc33450a3
Size: 6
Memory Address: 0x7fffc33450d5
Size: 7
Memory Address: 0x7fffc3345106
Size: 7
Memory Address: 0x7fffc334511e
Size: 7
Memory Address: 0x7fffc3345182
Size: 7
Memory Address: 0x7fffc33451e9
Size: 7
Memory Address: 0x7fffc3345201
Size: 6
Memory Address: 0x7fffc3345233
Size: 7
Memory Address: 0x7fffc3345266
Size: 7
Memory Address: 0x7fffc334527e
Size: 7
Memory Address: 0x7fffc33452cc
Size: 6
Memory Address: 0x7fffc334534d
Size: 6
Memory Address: 0x7fffc334537e
New Program Edge: 0x7fffc3345fd0
Allocated bytes: 6650 Free Bytes: 400
Started at: 0.15625s
Ended at: 0.734375s
root@LAPTOP-2NPNSAL0:~/OSP#
```

Best fit



```
Memory Address: 0x7fffea8be0e6
Size: 7
Memory Address: 0x7fffea8be130
Size: 7
Memory Address: 0x7fffea8be161
Size: 7
Memory Address: 0x7fffea8be1c9
Size: 7
Memory Address: 0x7fffea8be1e1
Size: 7
Memory Address: 0x7fffea8be1f9
Size: 7
Memory Address: 0x7fffea8be211
Size: 7
Memory Address: 0x7fffea8be243
Size: 7
Memory Address: 0x7fffea8be28e
Size: 7
Memory Address: 0x7fffea8be2bf
Size: 7
Memory Address: 0x7fffea8be2d7
Size: 7
Memory Address: 0x7fffea8be321
Size: 7
Memory Address: 0x7fffea8be339
Size: 7
Memory Address: 0x7fffea8be36b
New Program Edge: 0x7fffea8c1de1
Allocated bytes: 7035 Free Bytes: 1995
Started at: 0.0s
Ended at: 1.156250s
root@LAPTOP-2NPNSAL0:~/OSP#
```

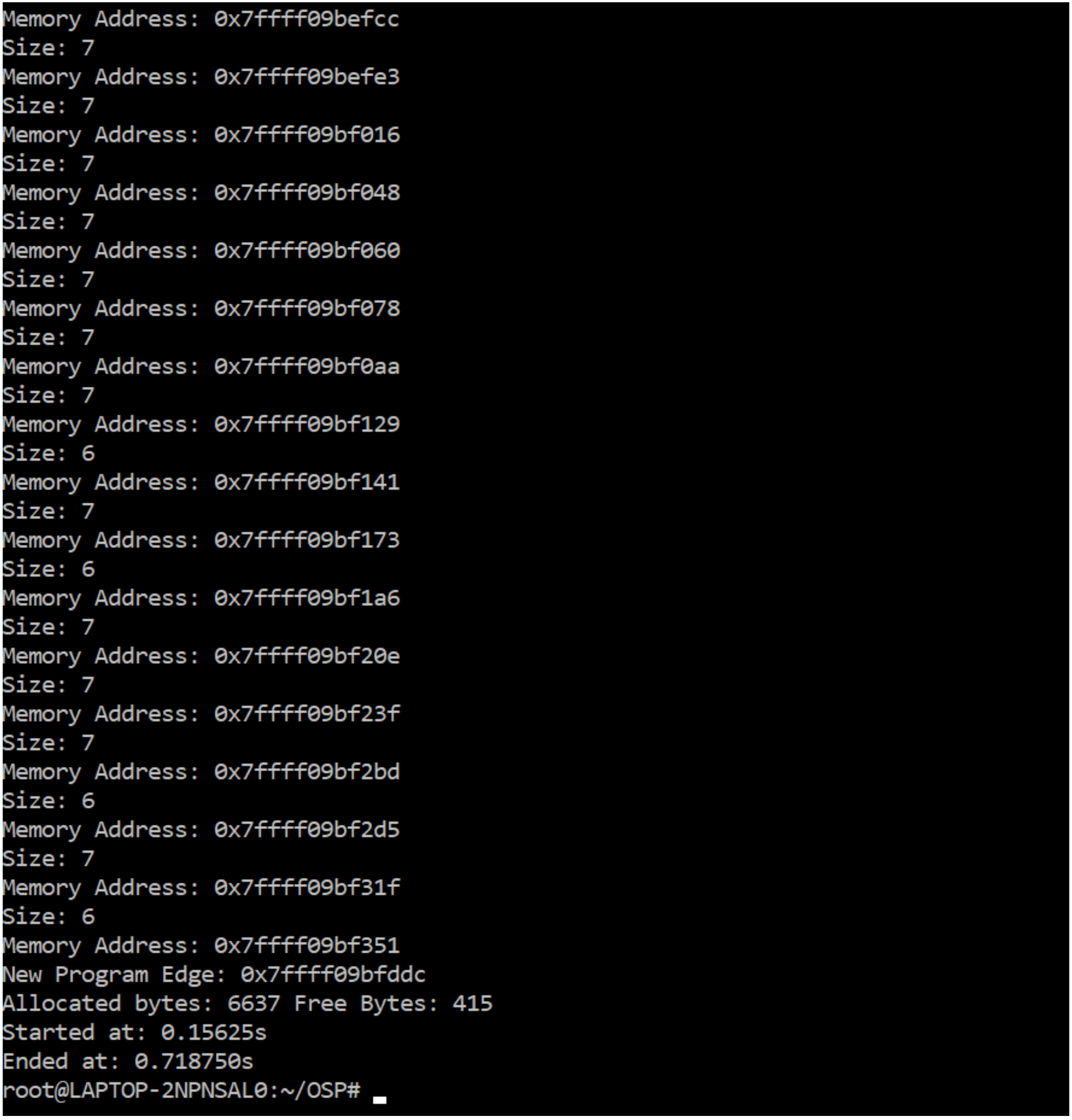
Worst fit



```
Memory Address: 0x7fffbfd680f9
Size: 8
Memory Address: 0x7fffbfd6812b
Size: 8
Memory Address: 0x7fffbfd68144
Size: 6
Memory Address: 0x7fffbfd6815d
Size: 8
Memory Address: 0x7fffbfd68174
Size: 6
Memory Address: 0x7fffbfd6818d
Size: 7
Memory Address: 0x7fffbfd681be
Size: 8
Memory Address: 0x7fffbfd681d6
Size: 7
Memory Address: 0x7fffbfd681ef
Size: 7
Memory Address: 0x7fffbfd68207
Size: 7
Memory Address: 0x7fffbfd6821f
Size: 6
Memory Address: 0x7fffbfd68237
Size: 4
Memory Address: 0x7fffbfd6824e
Size: 7
Memory Address: 0x7fffbfd68263
Size: 7
Memory Address: 0x7fffbfd6827b
Size: 8
Memory Address: 0x7fffbfd68293
Size: 7
Memory Address: 0x7fffbfd68317
New Program Edge: 0x7fffbfd6c300
Allocated bytes: 7316 Free Bytes: 3213
Started at: 0.0s
Ended at: 1.203125s
root@LAPTOP-2NPNSAL0:~/OSP#
```

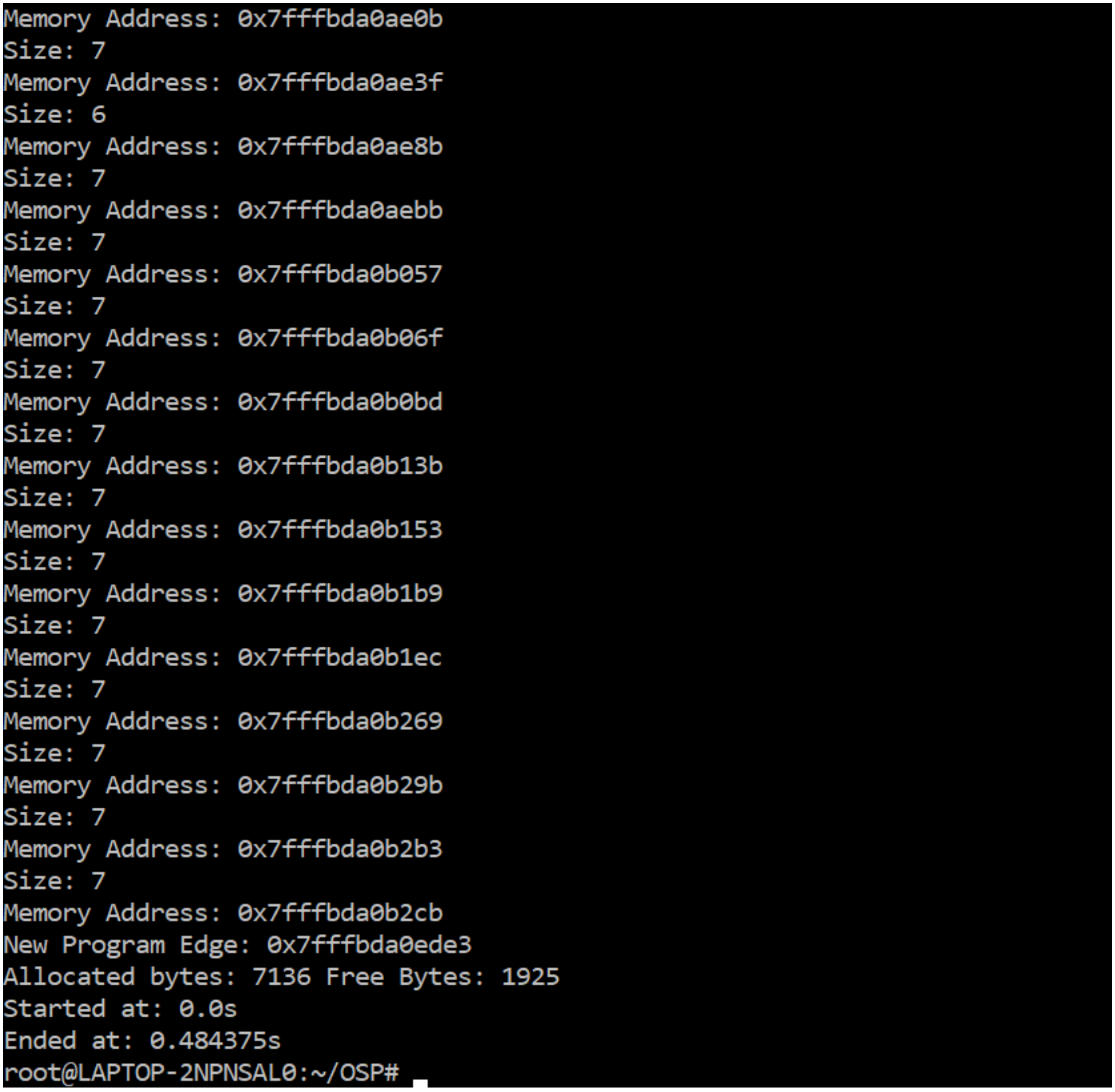
experiment1(half).txt

First fit



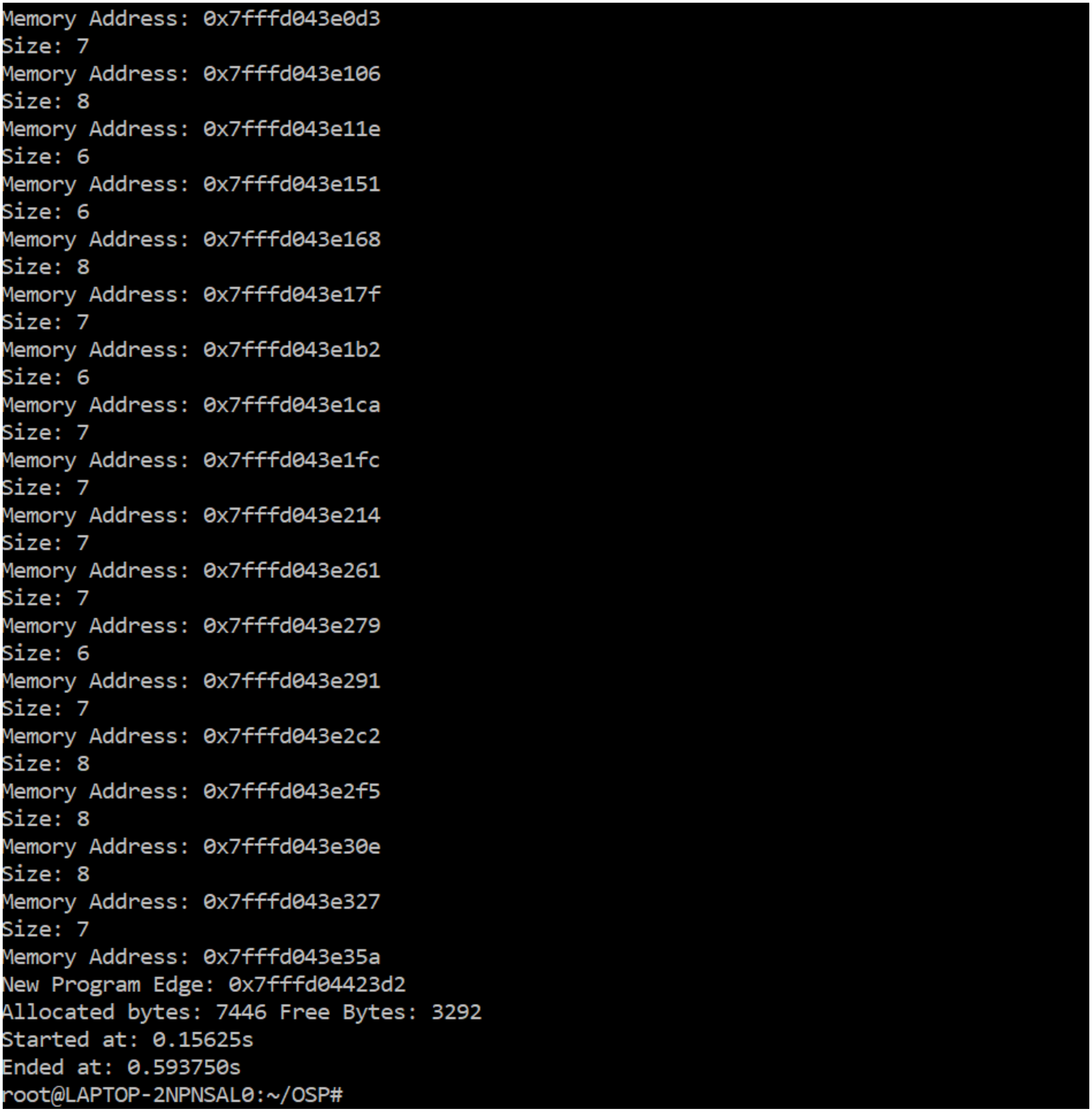
```
Memory Address: 0x7ffff09befcc
Size: 7
Memory Address: 0x7ffff09befe3
Size: 7
Memory Address: 0x7ffff09bf016
Size: 7
Memory Address: 0x7ffff09bf048
Size: 7
Memory Address: 0x7ffff09bf060
Size: 7
Memory Address: 0x7ffff09bf078
Size: 7
Memory Address: 0x7ffff09bf0aa
Size: 7
Memory Address: 0x7ffff09bf129
Size: 6
Memory Address: 0x7ffff09bf141
Size: 7
Memory Address: 0x7ffff09bf173
Size: 6
Memory Address: 0x7ffff09bf1a6
Size: 7
Memory Address: 0x7ffff09bf20e
Size: 7
Memory Address: 0x7ffff09bf23f
Size: 7
Memory Address: 0x7ffff09bf2bd
Size: 6
Memory Address: 0x7ffff09bf2d5
Size: 7
Memory Address: 0x7ffff09bf31f
Size: 6
Memory Address: 0x7ffff09bf351
New Program Edge: 0x7ffff09bfddc
Allocated bytes: 6637 Free Bytes: 415
Started at: 0.15625s
Ended at: 0.718750s
root@LAPTOP-2NPNSAL0:~/OSP#
```

Best fit



```
Memory Address: 0x7fffbda0ae0b
Size: 7
Memory Address: 0x7fffbda0ae3f
Size: 6
Memory Address: 0x7fffbda0ae8b
Size: 7
Memory Address: 0x7fffbda0aebb
Size: 7
Memory Address: 0x7fffbda0b057
Size: 7
Memory Address: 0x7fffbda0b06f
Size: 7
Memory Address: 0x7fffbda0b0bd
Size: 7
Memory Address: 0x7fffbda0b13b
Size: 7
Memory Address: 0x7fffbda0b153
Size: 7
Memory Address: 0x7fffbda0b1b9
Size: 7
Memory Address: 0x7fffbda0b1ec
Size: 7
Memory Address: 0x7fffbda0b269
Size: 7
Memory Address: 0x7fffbda0b29b
Size: 7
Memory Address: 0x7fffbda0b2b3
Size: 7
Memory Address: 0x7fffbda0b2cb
New Program Edge: 0x7fffbda0ede3
Allocated bytes: 7136 Free Bytes: 1925
Started at: 0.0s
Ended at: 0.484375s
root@LAPTOP-2NPNSAL0:~/OSP#
```

Worst fit

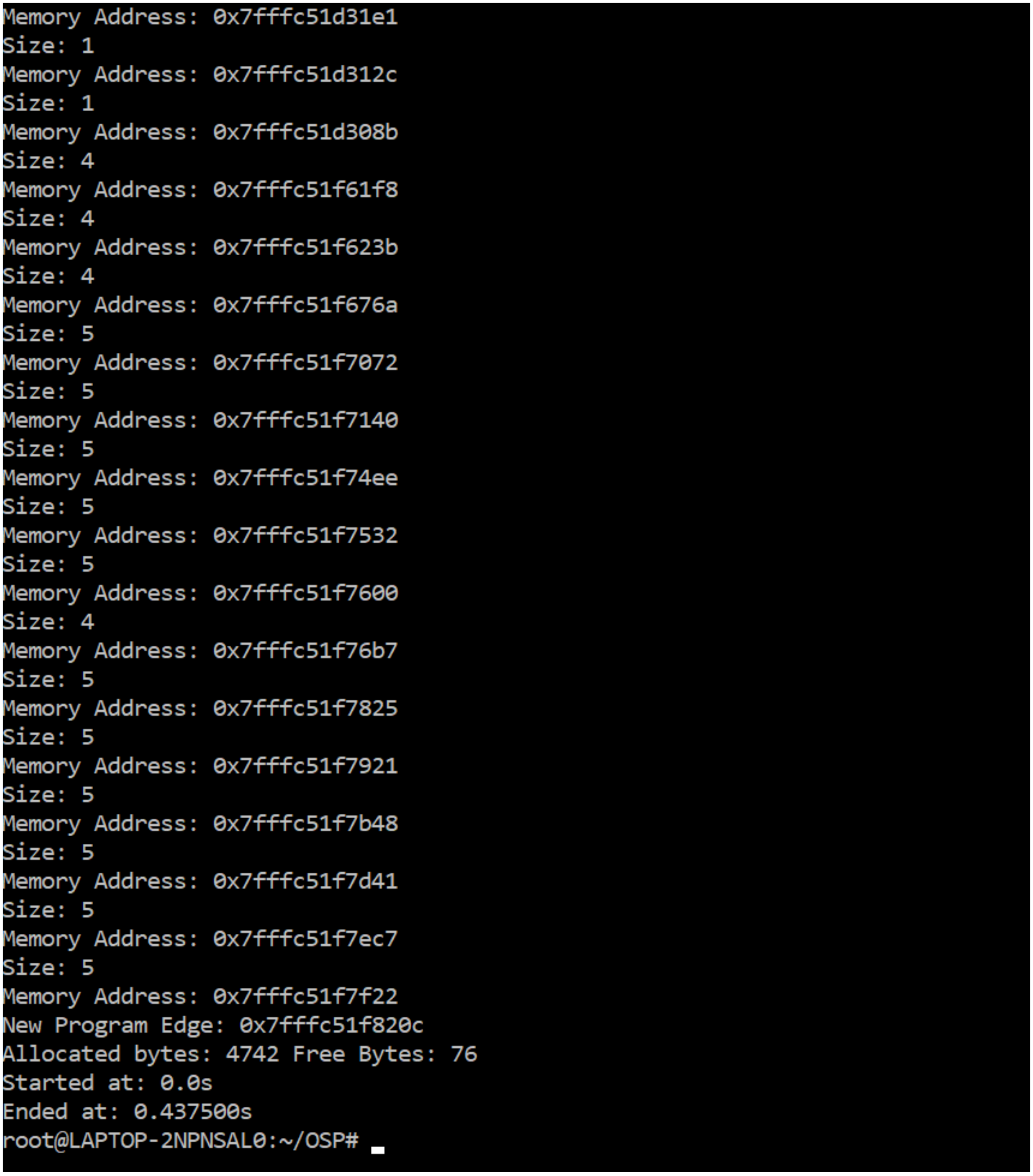


```
Memory Address: 0x7fffd043e0d3
Size: 7
Memory Address: 0x7fffd043e106
Size: 8
Memory Address: 0x7fffd043e11e
Size: 6
Memory Address: 0x7fffd043e151
Size: 6
Memory Address: 0x7fffd043e168
Size: 8
Memory Address: 0x7fffd043e17f
Size: 7
Memory Address: 0x7fffd043e1b2
Size: 6
Memory Address: 0x7fffd043e1ca
Size: 7
Memory Address: 0x7fffd043e1fc
Size: 7
Memory Address: 0x7fffd043e214
Size: 7
Memory Address: 0x7fffd043e261
Size: 7
Memory Address: 0x7fffd043e279
Size: 6
Memory Address: 0x7fffd043e291
Size: 7
Memory Address: 0x7fffd043e2c2
Size: 8
Memory Address: 0x7fffd043e2f5
Size: 8
Memory Address: 0x7fffd043e30e
Size: 8
Memory Address: 0x7fffd043e327
Size: 7
Memory Address: 0x7fffd043e35a
New Program Edge: 0x7fffd04423d2
Allocated bytes: 7446 Free Bytes: 3292
Started at: 0.15625s
Ended at: 0.593750s
root@LAPTOP-2NPNSAL0:~/OSP#
```

Appendix B

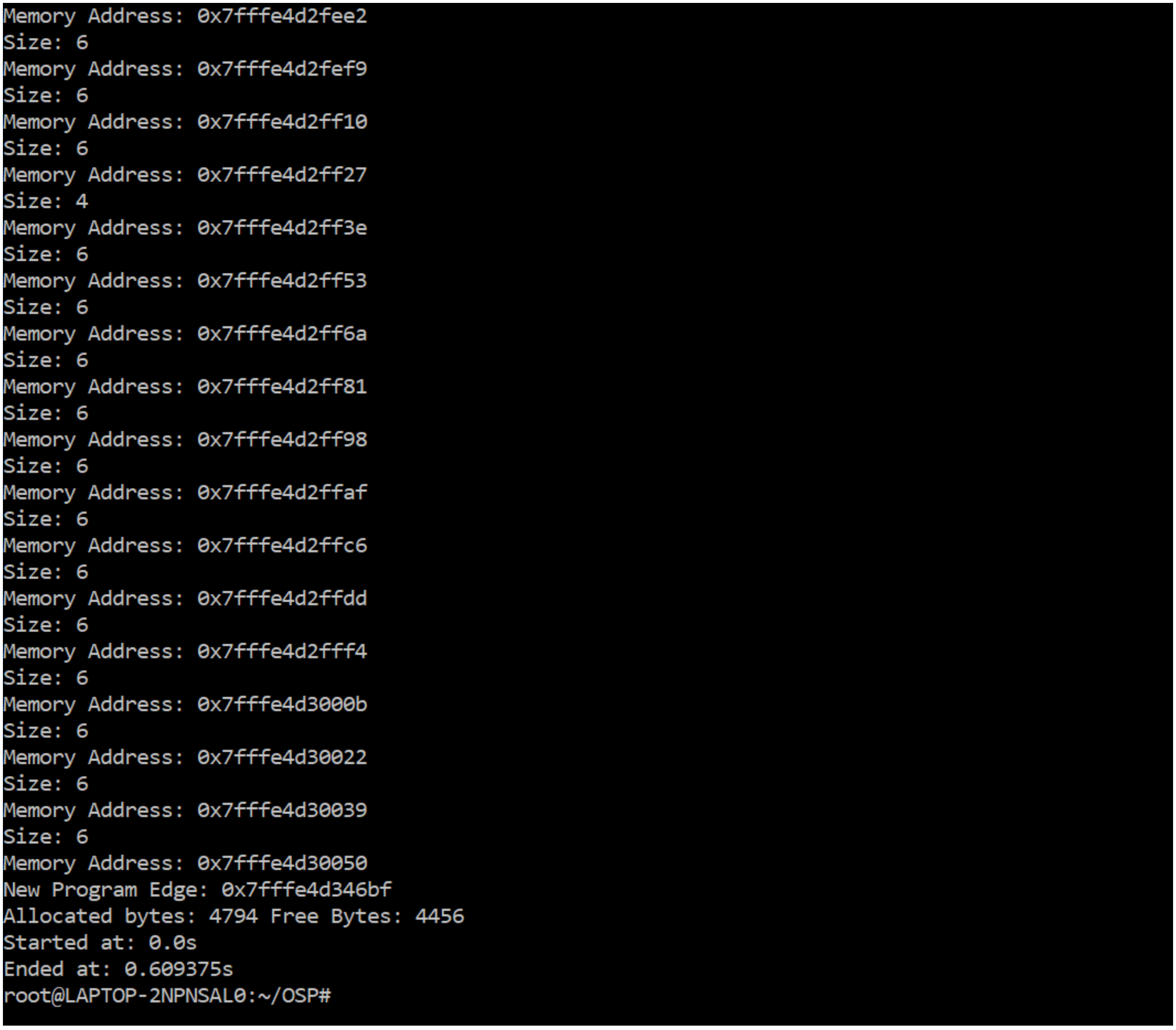
experiment2.txt

First fit



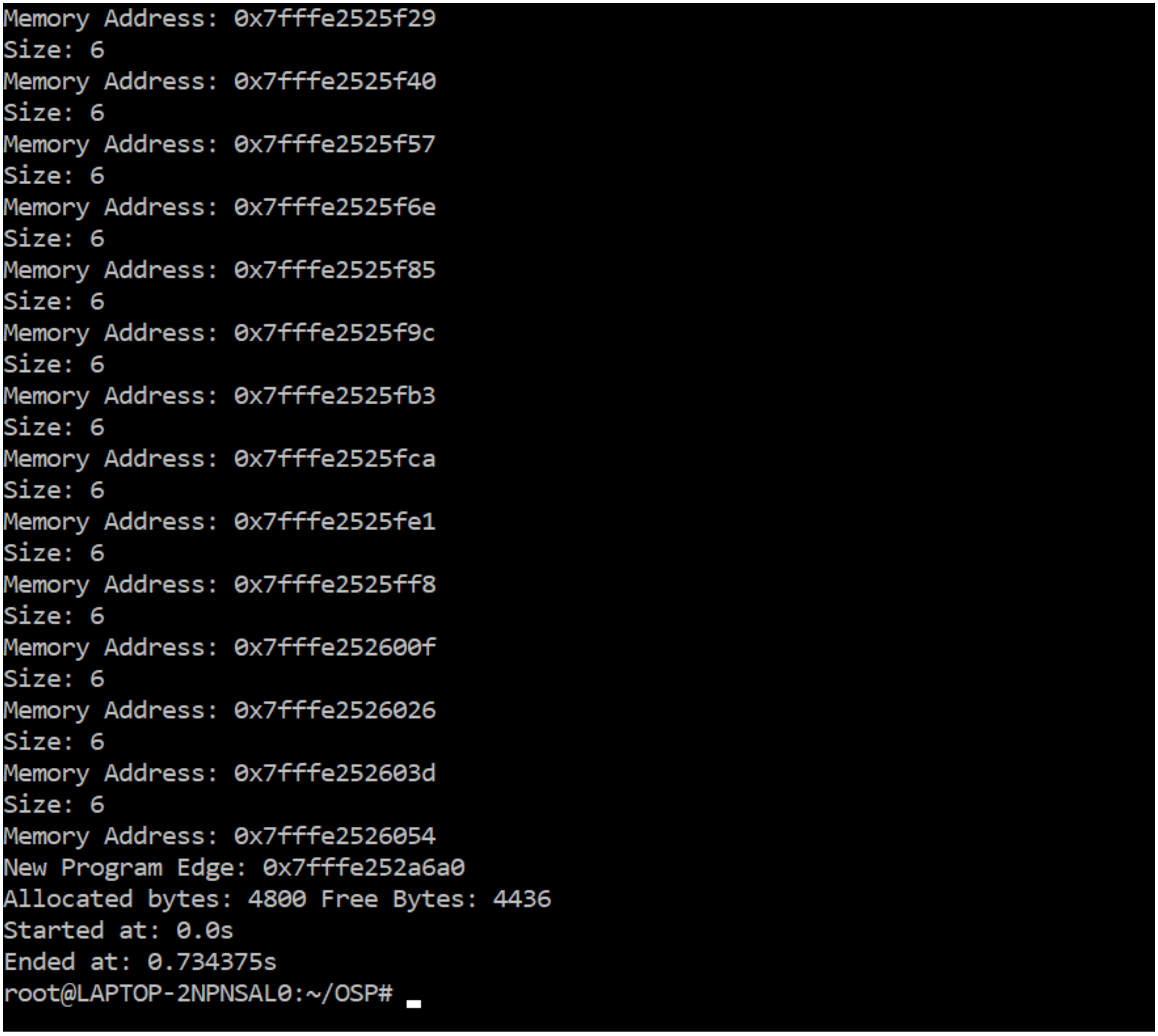
```
Memory Address: 0x7fffc51d31e1
Size: 1
Memory Address: 0x7fffc51d312c
Size: 1
Memory Address: 0x7fffc51d308b
Size: 4
Memory Address: 0x7fffc51f61f8
Size: 4
Memory Address: 0x7fffc51f623b
Size: 4
Memory Address: 0x7fffc51f676a
Size: 5
Memory Address: 0x7fffc51f7072
Size: 5
Memory Address: 0x7fffc51f7140
Size: 5
Memory Address: 0x7fffc51f74ee
Size: 5
Memory Address: 0x7fffc51f7532
Size: 5
Memory Address: 0x7fffc51f7600
Size: 4
Memory Address: 0x7fffc51f76b7
Size: 5
Memory Address: 0x7fffc51f7825
Size: 5
Memory Address: 0x7fffc51f7921
Size: 5
Memory Address: 0x7fffc51f7b48
Size: 5
Memory Address: 0x7fffc51f7d41
Size: 5
Memory Address: 0x7fffc51f7ec7
Size: 5
Memory Address: 0x7fffc51f7f22
New Program Edge: 0x7fffc51f820c
Allocated bytes: 4742 Free Bytes: 76
Started at: 0.0s
Ended at: 0.437500s
root@LAPTOP-2NPNSAL0:~/OSP#
```

Best fit



```
Memory Address: 0x7fffe4d2fee2
Size: 6
Memory Address: 0x7fffe4d2fef9
Size: 6
Memory Address: 0x7fffe4d2ff10
Size: 6
Memory Address: 0x7fffe4d2ff27
Size: 4
Memory Address: 0x7fffe4d2ff3e
Size: 6
Memory Address: 0x7fffe4d2ff53
Size: 6
Memory Address: 0x7fffe4d2ff6a
Size: 6
Memory Address: 0x7fffe4d2ff81
Size: 6
Memory Address: 0x7fffe4d2ff98
Size: 6
Memory Address: 0x7fffe4d2ffaf
Size: 6
Memory Address: 0x7fffe4d2ffc6
Size: 6
Memory Address: 0x7fffe4d2ffdd
Size: 6
Memory Address: 0x7fffe4d2fff4
Size: 6
Memory Address: 0x7fffe4d3000b
Size: 6
Memory Address: 0x7fffe4d30022
Size: 6
Memory Address: 0x7fffe4d30039
Size: 6
Memory Address: 0x7fffe4d30050
New Program Edge: 0x7fffe4d346bf
Allocated bytes: 4794 Free Bytes: 4456
Started at: 0.0s
Ended at: 0.609375s
root@LAPTOP-2NPNSAL0:~/OSP#
```

Worst fit



```
Memory Address: 0x7fffe2525f29
Size: 6
Memory Address: 0x7fffe2525f40
Size: 6
Memory Address: 0x7fffe2525f57
Size: 6
Memory Address: 0x7fffe2525f6e
Size: 6
Memory Address: 0x7fffe2525f85
Size: 6
Memory Address: 0x7fffe2525f9c
Size: 6
Memory Address: 0x7fffe2525fb3
Size: 6
Memory Address: 0x7fffe2525fca
Size: 6
Memory Address: 0x7fffe2525fe1
Size: 6
Memory Address: 0x7fffe2525ff8
Size: 6
Memory Address: 0x7fffe252600f
Size: 6
Memory Address: 0x7fffe2526026
Size: 6
Memory Address: 0x7fffe252603d
Size: 6
Memory Address: 0x7fffe2526054
New Program Edge: 0x7fffe252a6a0
Allocated bytes: 4800 Free Bytes: 4436
Started at: 0.0s
Ended at: 0.734375s
root@LAPTOP-2NPNSAL0:~/OSP#
```

Bibliography

1. Awais MA. Memory Management: Challenges and Techniques. ACSIJ Advances in Computer Science: an International Journal. 2016 March; 5(2).

2. Patil N. Dynamic Memory Allocation: Role in Memory Management. International Journal of Current Engineering and Technology. 2014 April; 4(2).