CS 342 - OPERATING SYSTEMS

PROJECT 3

ABDULLAH CAN ALPAY

21702686 - SECTION 3

22.04.2021

Project Description

In the below, I explained how I wrote the functions:

sbmem_init(int segsize)

Firstly, I unlink previously opened memory with the name SHARED_NAME and semaphore called SEM_MUTEX. Then I decide my maintenance size where I put the necessary information. These are shared memory size, maintenance memory size, the current number of a struct (also can be referred to as segments in the memory size), and the current number of processes using the library. The project description states that the maximum memory size is 256 KB, 2^{18}, and the minimum size request will be 128 bytes. Thus, in the worst case, we can have a 4096 linked list structure in the maintenance memory. After that, we allocate the necessary space for the shared memory with shm_open(SHARED_NAME, O_RDWR | O_CREAT, 0666) and ftruncate(fd, segmentsize + struct_size). When we allocate the space, we need to insert the previously mentioned information and our first data segment (memory size = segmentsize).

sbmem_remove()

I unlink the shared memory and semaphore.

sbmem_open()

Firstly, I open the semaphore, sem_open(SEM_MUTEX, O_RDWR). After that, the function waits for mutex2 since the function will access and adjust the data from the shared memory. It will check whether the library has allowed another process or not, and if the process_count less than or equal to 0, the function will return -1. Otherwise, it will decrement the available process count.

sbmem_alloc(int size)

The function will find the smallest power of 2, which is greater or equal to size (called pow2). Then, the function will traverse along the linked list nodes to find the perfect suit for the pow2 in which the node's pid is -1. If the program could not find this node, it will find the smallest node's length where the node's length is greater than pow2, and the node's pid is -1. It will divide the memory into two, and it will add a new node in the list pointing new memory address. Hence new list node is added to the maintenance, struct number will be incremented in the maintenance. While the function dividing the memory, if it finds the memory size equal to pow2, it will terminate the while and return the memory's address.

Important Point:
In inner_list struct, I stored id, which helps me to traverse the memory division. For list ids, I used a complete binary tree structure. In a complete binary tree, the parent node with id i will have children with id 2*i and 2*i + 1. When the function divides the memory, it will double the current node's id and give id + 1 to the newly created node.

sbmem_free(void* ptr)

The function first finds the correct node, which stores the information for the ptr's pointed memory. Then the function will check whether any neighbor is free (pid = -1) and their id difference is one. If the difference is indeed one, the function will combine these two nodes and make the new node's id half the previous id. To exemplify, let's assume we have two memory segments with id 4 (we free this memory segment) and 5 (which is already free). The function will combine these two memory segments since 5 - 4 = 1 and assign 2 to the new node. Similarly, since we delete one node from the structure, the function will decrement the struct number.

sbmem_close()

This function will simply increment the process count in the maintenance.

Experiment

First Experiment (first_experiment.c)

One process initializes memory with 256bytes. Then tries to allocate 31, 120, 6 respectively. Then, the process frees the 31 and 120 and tries allocating 65.

Initial Fragmentation: (32 - 31) + (8 - 6) + (128 - 120) = 11
Final Fragmentation: (8 - 6) + (128 - 65) = 65
Additional to experiment 1, I also tried using this library with 18 processes. As it is expected 8 processes are not allowed to use the library.

The output is in the Appendix, Experiment 1 Output.

Second Experiment (second_experiment.c)

In the second experiment, I used 2 processes and the initially shared memory size is 32768. The first process allocates 3800 and 124, and the second process allocates 120, 430, 230. After allocation, processes free the memories in an allocation order.

From that, we can compute the internal and external fragmentations.
For Process 1, Fragmentation: (4096 - 3800) + (128 - 124) = 300
For Process 2, Fragmentation: (128 - 120) + (512 - 430) + (256 - 230) = 116
Total Fragmentation = 300 + 116 = 416

The output is in the Appendix, Experiment 2 Output.

Before deriving any conclusion, let’s look at the third experiment.

Third Experiment (third_experiment.c)

The experiment is conducted with two processes and 32768 shared memory space.
The first process requests 250 bytes from the shared memory 5 times.
the second process requests 512 bytes from the shared memory 3 times.

For Process 1, Fragmentation: 5 * (256 - 250) = 30
For Process 2, Fragmentation: 3 * (512 - 512) = 0

The output is in Appendix, Experiment 3.

From the first 3 experiments we understand that the Buddy Algorithm always assigns the smallest power of 2 that is greater or equal than the requested size. Then we can say that fragmentation depends on the process requests. For the first three experiments, the fragmentation/process plot is given below.

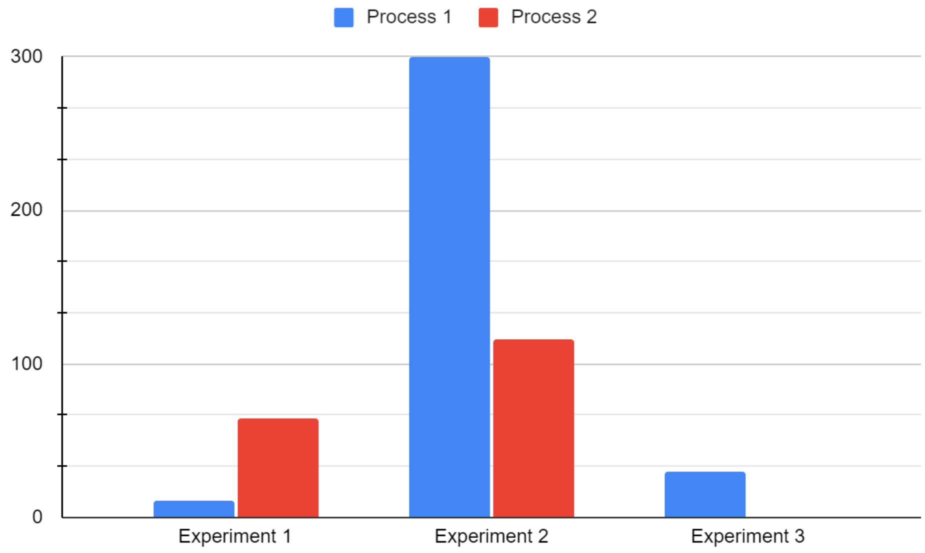

Figure - 1

Note that in experiment 1 blue column refers to the fragmentation before freeing the memory, and red column refers to when new allocation is made after freeing.

Fourth Experiment (fourth_experiment.c)

For the fourth experiment I use 3 processes, with the shared memory size 16384.
The first processor requests 240, 400, 500, 470 and 890 bytes.
The second processor requests 840, 970, 3800, 1024 and 500 bytes.
The third processor requests 512, 270, 1000, 250, 1025 bytes.

For Process 1 Fragmentation: (256 - 240) + (512 - 400) + (512 - 500) + (512 - 470) + (1024 - 890) = 316
For Process 2 Fragmentation: (1024 - 840) + (1024- 970) + (4096- 3800) + (1024- 1024) + (512- 500) = 546
For Process 3 Fragmentation: (512 - 512) + (512 - 270) + (1024 - 1000) + (256 - 250) + (2048 - 1025) = 1295

The output is in Appendix, Experiment 4.
The fragmentation/process graph is given below.

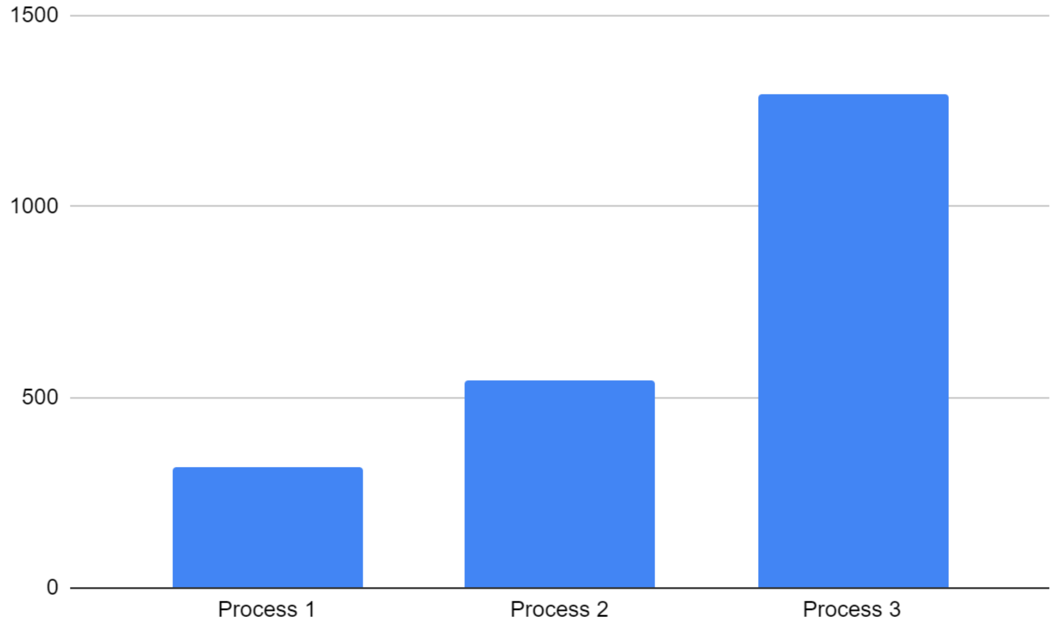

Figure - 2

Appendix

Experiment 1 Output:

```
Allocation for 21203, size = 31 ==> 32
Allocation for 21203, size = 120 ==> 128
Allocation for 21203, size = 6 ==> 8

Before freeing the allocated memory, the memory is (-1 means empty):
21203 pid, 32 memory size
21203 pid, 8 memory size
-1 pid, 8 memory size
-1 pid, 16 memory size
-1 pid, 64 memory size
21203 pid, 128 memory size
After freeing the memory, new memory:
-1 pid, 32 memory
21203 pid, 8 memory
-1 pid, 8 memory
-1 pid, 16 memory
-1 pid, 64 memory
21203 pid, 128 memory
Before freeing the allocated memory, the memory is (-1 means empty):
-1 pid, 32 memory size
21203 pid, 8 memory size
-1 pid, 8 memory size
-1 pid, 16 memory size
-1 pid, 64 memory size
21203 pid, 128 memory size
After freeing the memory, new memory:
-1 pid, 32 memory
21203 pid, 8 memory
-1 pid, 8 memory
-1 pid, 16 memory
-1 pid, 64 memory
-1 pid, 128 memory
Before freeing the allocated memory, the memory is (-1 means empty):
-1 pid, 32 memory size
21203 pid, 8 memory size
-1 pid, 8 memory size
-1 pid, 16 memory size
-1 pid, 64 memory size
-1 pid, 128 memory size
```

```
After freeing the memory, new memory:
-1 pid, 256 memory
Allocation for 21203, size = 65 ==> 128
Before freeing the allocated memory, the memory is (-1 means empty):
21203 pid, 128 memory size
-1 pid, 128 memory size
After freeing the memory, new memory:
-1 pid, 256 memory
ACCESS REQUEST ==> 21518
ACCESS GRANTED
ACCESS REQUEST ==> 21519
ACCESS GRANTED
ACCESS REQUEST ==> 21520
ACCESS GRANTED
ACCESS REQUEST ==> 21521
ACCESS GRANTED
ACCESS REQUEST ==> 21522
ACCESS GRANTED
ACCESS REQUEST ==> 21523
ACCESS GRANTED
ACCESS REQUEST ==> 21524
ACCESS GRANTED
ACCESS REQUEST ==> 21525
ACCESS GRANTED
ACCESS REQUEST ==> 21526
ACCESS GRANTED
ACCESS REQUEST ==> 21527
ACCESS GRANTED
ACCESS REQUEST ==> 21528
Process 21528 is not allowed to use the library
ACCESS REQUEST ==> 21529
Process 21529 is not allowed to use the library
ACCESS REQUEST ==> 21530
Process 21530 is not allowed to use the library
ACCESS REQUEST ==> 21531
Process 21531 is not allowed to use the library
ACCESS REQUEST ==> 21532
Process 21532 is not allowed to use the library
ACCESS REQUEST ==> 21533
Process 21533 is not allowed to use the library
ACCESS REQUEST ==> 21534
Process 21534 is not allowed to use the library
ACCESS REQUEST ==> 21535
Process 21535 is not allowed to use the library
```

Experiment 2 Output:

```
ACCESS REQUEST ==> 22909
ACCESS GRANTED
Allocation for 22909, size = 3800 ==> 4096
Allocation for 22909, size = 124 ==> 128
Before freeing the allocated memory, the memory is (-1 means empty):
22909 pid, 4096 memory size
22909 pid, 128 memory size
-1 pid, 128 memory size
-1 pid, 256 memory size
-1 pid, 512 memory size
-1 pid, 1024 memory size
-1 pid, 2048 memory size
-1 pid, 8192 memory size
-1 pid, 16384 memory size
After freeing the memory, new memory:
-1 pid, 4096 memory
22909 pid, 128 memory
-1 pid, 128 memory
-1 pid, 256 memory
-1 pid, 512 memory
-1 pid, 1024 memory
-1 pid, 2048 memory
-1 pid, 8192 memory
-1 pid, 16384 memory
Before freeing the allocated memory, the memory is (-1 means empty):
-1 pid, 4096 memory size
22909 pid, 128 memory size
-1 pid, 128 memory size
ACCESS REQUEST ==> 22910
-1 pid, 256 memory size
-1 pid, 512 memory size
ACCESS GRANTED
-1 pid, 1024 memory size
-1 pid, 2048 memory size
-1 pid, 8192 memory size
-1 pid, 16384 memory size
After freeing the memory, new memory:
-1 pid, 32768 memory
Allocation for 22910, size = 120 ==> 128
Allocation for 22910, size = 430 ==> 512
Allocation for 22910, size = 230 ==> 256
Before freeing the allocated memory, the memory is (-1 means empty):
22910 pid, 128 memory size
-1 pid, 128 memory size
22910 pid, 256 memory size
22910 pid, 512 memory size
-1 pid, 1024 memory size
-1 pid, 2048 memory size
-1 pid, 4096 memory size
-1 pid, 8192 memory size
-1 pid, 16384 memory size
After freeing the memory, new memory:
-1 pid, 256 memory
22910 pid, 256 memory
22910 pid, 512 memory
-1 pid, 1024 memory
-1 pid, 2048 memory
-1 pid, 4096 memory
-1 pid, 8192 memory
-1 pid, 16384 memory
Before freeing the allocated memory, the memory is (-1 means empty):
-1 pid, 256 memory size
22910 pid, 256 memory size
22910 pid, 512 memory size
-1 pid, 1024 memory size
-1 pid, 2048 memory size
-1 pid, 4096 memory size
-1 pid, 8192 memory size
-1 pid, 16384 memory size
After freeing the memory, new memory:
-1 pid, 256 memory
22910 pid, 256 memory
-1 pid, 512 memory
-1 pid, 1024 memory
-1 pid, 2048 memory
-1 pid, 4096 memory
-1 pid, 8192 memory
-1 pid, 16384 memory
Before freeing the allocated memory, the memory is (-1 means empty):
-1 pid, 256 memory size
22910 pid, 256 memory size
-1 pid, 512 memory size
-1 pid, 1024 memory size
-1 pid, 2048 memory size
```

```
-1 pid, 4096 memory size
-1 pid, 8192 memory size
-1 pid, 16384 memory size
After freeing the memory, new memory:
-1 pid, 32768 memory
```

Experiment 3 Output:

```
ACCESS REQUEST ==> 32735
ACCESS GRANTED
Allocation for 32735, size = 512 ==> 512
Allocation for 32735, size = 512 ==> 512
Allocation for 32735, size = 512 ==> 512
ACCESS REQUEST ==> 32736
ACCESS GRANTED
Allocation for 32736, size = 250 ==> 256
Allocation for 32736, size = 250 ==> 256
Allocation for 32736, size = 250 ==> 256
Allocation for 32736, size = 250 ==> 256
Allocation for 32736, size = 250 ==> 256
Before freeing the allocated memory, the memory is (-1 means empty):
32735 pid, 512 memory size
32735 pid, 512 memory size
32735 pid, 512 memory size
32736 pid, 256 memory size
32736 pid, 256 memory size
32736 pid, 256 memory size
32736 pid, 256 memory size
32736 pid, 256 memory size
-1 pid, 256 memory size
-1 pid, 1024 memory size
-1 pid, 4096 memory size
-1 pid, 8192 memory size
After freeing the memory, new memory:
32735 pid, 512 memory
32735 pid, 512 memory
32735 pid, 512 memory
-1 pid, 256 memory
32736 pid, 256 memory
32736 pid, 256 memory
32736 pid, 256 memory
32736 pid, 256 memory
-1 pid, 256 memory
-1 pid, 1024 memory
```

```
-1 pid, 4096 memory
-1 pid, 8192 memory
Before freeing the allocated memory, the memory is (-1 means empty):
32735 pid, 512 memory size
32735 pid, 512 memory size
32735 pid, 512 memory size
-1 pid, 256 memory size
32736 pid, 256 memory size
32736 pid, 256 memory size
32736 pid, 256 memory size
32736 pid, 256 memory size
-1 pid, 256 memory size
-1 pid, 1024 memory size
-1 pid, 4096 memory size
-1 pid, 8192 memory size
After freeing the memory, new memory:
32735 pid, 512 memory
32735 pid, 512 memory
32735 pid, 512 memory
-1 pid, 512 memory
32736 pid, 256 memory
32736 pid, 256 memory
32736 pid, 256 memory
-1 pid, 256 memory
-1 pid, 1024 memory
-1 pid, 4096 memory
-1 pid, 8192 memory
Before freeing the allocated memory, the memory is (-1 means empty):
32735 pid, 512 memory size
32735 pid, 512 memory size
32735 pid, 512 memory size
-1 pid, 512 memory size
32736 pid, 256 memory size
32736 pid, 256 memory size
32736 pid, 256 memory size
-1 pid, 256 memory size
-1 pid, 1024 memory size
-1 pid, 4096 memory size
-1 pid, 8192 memory size
After freeing the memory, new memory:
32735 pid, 512 memory
32735 pid, 512 memory
32735 pid, 512 memory
-1 pid, 512 memory
```

```
-1 pid, 256 memory
32736 pid, 256 memory
32736 pid, 256 memory
-1 pid, 256 memory
-1 pid, 1024 memory
-1 pid, 4096 memory
-1 pid, 8192 memory
Before freeing the allocated memory, the memory is (-1 means empty):
32735 pid, 512 memory size
32735 pid, 512 memory size
32735 pid, 512 memory size
-1 pid, 512 memory size
-1 pid, 256 memory size
32736 pid, 256 memory size
32736 pid, 256 memory size
-1 pid, 256 memory size
-1 pid, 1024 memory size
-1 pid, 4096 memory size
-1 pid, 8192 memory size
After freeing the memory, new memory:
32735 pid, 512 memory
32735 pid, 512 memory
32735 pid, 512 memory
-1 pid, 512 memory
-1 pid, 512 memory
32736 pid, 256 memory
-1 pid, 256 memory
-1 pid, 1024 memory
-1 pid, 4096 memory
-1 pid, 8192 memory
Before freeing the allocated memory, the memory is (-1 means empty):
32735 pid, 512 memory size
32735 pid, 512 memory size
32735 pid, 512 memory size
-1 pid, 512 memory size
-1 pid, 512 memory size
32736 pid, 256 memory size
-1 pid, 256 memory size
-1 pid, 1024 memory size
-1 pid, 4096 memory size
-1 pid, 8192 memory size
After freeing the memory, new memory:
32735 pid, 512 memory
32735 pid, 512 memory
```

```
32735 pid, 512 memory
-1 pid, 512 memory
-1 pid, 2048 memory
-1 pid, 4096 memory
-1 pid, 8192 memory
Before freeing the allocated memory, the memory is (-1 means empty):
32735 pid, 512 memory size
32735 pid, 512 memory size
32735 pid, 512 memory size
-1 pid, 512 memory size
-1 pid, 2048 memory size
-1 pid, 4096 memory size
-1 pid, 8192 memory size
After freeing the memory, new memory:
-1 pid, 512 memory
32735 pid, 512 memory
32735 pid, 512 memory
-1 pid, 512 memory
-1 pid, 2048 memory
-1 pid, 4096 memory
-1 pid, 8192 memory
Before freeing the allocated memory, the memory is (-1 means empty):
-1 pid, 512 memory size
32735 pid, 512 memory size
32735 pid, 512 memory size
-1 pid, 512 memory size
-1 pid, 2048 memory size
-1 pid, 4096 memory size
-1 pid, 8192 memory size
After freeing the memory, new memory:
-1 pid, 1024 memory
32735 pid, 512 memory
-1 pid, 512 memory
-1 pid, 2048 memory
-1 pid, 4096 memory
-1 pid, 8192 memory
Before freeing the allocated memory, the memory is (-1 means empty):
-1 pid, 1024 memory size
32735 pid, 512 memory size
-1 pid, 512 memory size
-1 pid, 2048 memory size
-1 pid, 4096 memory size
-1 pid, 8192 memory size
After freeing the memory, new memory:
```

```
-1 pid, 16384 memory
```

Experiment 4 Output:

```
ACCESS REQUEST ==> 32584
ACCESS GRANTED
Allocation for 32584, size = 200 ==> 256
Allocation for 32584, size = 300 ==> 512
Allocation for 32584, size = 400 ==> 512
Allocation for 32584, size = 500 ==> 512
Allocation for 32584, size = 600 ==> 1024
ACCESS REQUEST ==> 32585
ACCESS GRANTED
Allocation for 32585, size = 610 ==> 1024
Allocation for 32585, size = 710 ==> 1024
Allocation for 32585, size = 4096 ==> 4096
Allocation for 32585, size = 1024 ==> 1024
Allocation for 32585, size = 400 ==> 512
ACCESS REQUEST ==> 32586
ACCESS GRANTED
Allocation for 32586, size = 340 ==> 512
Allocation for 32586, size = 340 ==> 512
Allocation for 32586, size = 540 ==> 1024
Allocation for 32586, size = 140 ==> 256
Allocation for 32586, size = 1040 ==> 2048
Before freeing the allocated memory, the memory is (-1 means empty):
32584 pid, 256 memory size
32586 pid, 256 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 512 memory size
32586 pid, 512 memory size
32586 pid, 512 memory size
-1 pid, 512 memory size
32585 pid, 4096 memory size
32586 pid, 1024 memory size
-1 pid, 1024 memory size
32586 pid, 2048 memory size
After freeing the memory, new memory:
```

```
32584 pid, 256 memory
32586 pid, 256 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 1024 memory
32585 pid, 1024 memory
32585 pid, 1024 memory
32585 pid, 1024 memory
32585 pid, 512 memory
-1 pid, 512 memory
32586 pid, 512 memory
-1 pid, 512 memory
32585 pid, 4096 memory
32586 pid, 1024 memory
-1 pid, 1024 memory
32586 pid, 2048 memory
Before freeing the allocated memory, the memory is (-1 means empty):
32584 pid, 256 memory size
32586 pid, 256 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 512 memory size
-1 pid, 512 memory size
32586 pid, 512 memory size
-1 pid, 512 memory size
32585 pid, 4096 memory size
32586 pid, 1024 memory size
-1 pid, 1024 memory size
32586 pid, 2048 memory size
After freeing the memory, new memory:
32584 pid, 256 memory
32586 pid, 256 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 1024 memory
32585 pid, 1024 memory
32585 pid, 1024 memory
```

```
32585 pid, 1024 memory
32585 pid, 512 memory
-1 pid, 512 memory
-1 pid, 1024 memory
32585 pid, 4096 memory
32586 pid, 1024 memory
-1 pid, 1024 memory
32586 pid, 2048 memory
Before freeing the allocated memory, the memory is (-1 means empty):
32584 pid, 256 memory size
32586 pid, 256 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 512 memory size
-1 pid, 512 memory size
-1 pid, 1024 memory size
32585 pid, 4096 memory size
32586 pid, 1024 memory size
-1 pid, 1024 memory size
32586 pid, 2048 memory size
After freeing the memory, new memory:
32584 pid, 256 memory
32586 pid, 256 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 1024 memory
32585 pid, 1024 memory
32585 pid, 1024 memory
32585 pid, 1024 memory
32585 pid, 512 memory
-1 pid, 512 memory
-1 pid, 1024 memory
32585 pid, 4096 memory
-1 pid, 2048 memory
32586 pid, 2048 memory
Before freeing the allocated memory, the memory is (-1 means empty):
32584 pid, 256 memory size
32586 pid, 256 memory size
```

```
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 512 memory size
-1 pid, 512 memory size
-1 pid, 1024 memory size
32585 pid, 4096 memory size
-1 pid, 2048 memory size
32586 pid, 2048 memory size
After freeing the memory, new memory:
32584 pid, 256 memory
-1 pid, 256 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 1024 memory
32585 pid, 1024 memory
32585 pid, 1024 memory
32585 pid, 1024 memory
32585 pid, 512 memory
-1 pid, 512 memory
-1 pid, 1024 memory
32585 pid, 4096 memory
-1 pid, 2048 memory
32586 pid, 2048 memory
Before freeing the allocated memory, the memory is (-1 means empty):
32584 pid, 256 memory size
-1 pid, 256 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 512 memory size
-1 pid, 512 memory size
-1 pid, 1024 memory size
32585 pid, 4096 memory size
-1 pid, 2048 memory size
```

```
32586 pid, 2048 memory size
After freeing the memory, new memory:
32584 pid, 256 memory
-1 pid, 256 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 1024 memory
32585 pid, 1024 memory
32585 pid, 1024 memory
32585 pid, 1024 memory
32585 pid, 512 memory
-1 pid, 512 memory
-1 pid, 1024 memory
32585 pid, 4096 memory
-1 pid, 4096 memory
Before freeing the allocated memory, the memory is (-1 means empty):
32584 pid, 256 memory size
-1 pid, 256 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 512 memory size
-1 pid, 512 memory size
-1 pid, 1024 memory size
32585 pid, 4096 memory size
-1 pid, 4096 memory size
After freeing the memory, new memory:
32584 pid, 256 memory
-1 pid, 256 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 1024 memory
-1 pid, 1024 memory
32585 pid, 1024 memory
32585 pid, 1024 memory
32585 pid, 512 memory
-1 pid, 512 memory
-1 pid, 1024 memory
```

```
32585 pid, 4096 memory
-1 pid, 4096 memory
Before freeing the allocated memory, the memory is (-1 means empty):
32584 pid, 256 memory size
-1 pid, 256 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 1024 memory size
-1 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 512 memory size
-1 pid, 512 memory size
-1 pid, 1024 memory size
32585 pid, 4096 memory size
-1 pid, 4096 memory size
After freeing the memory, new memory:
32584 pid, 256 memory
-1 pid, 256 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 1024 memory
-1 pid, 1024 memory
-1 pid, 1024 memory
32585 pid, 1024 memory
32585 pid, 512 memory
-1 pid, 512 memory
-1 pid, 1024 memory
32585 pid, 4096 memory
-1 pid, 4096 memory
Before freeing the allocated memory, the memory is (-1 means empty):
32584 pid, 256 memory size
-1 pid, 256 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 1024 memory size
-1 pid, 1024 memory size
-1 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 512 memory size
-1 pid, 512 memory size
```

```
-1 pid, 1024 memory size
32585 pid, 4096 memory size
-1 pid, 4096 memory size
After freeing the memory, new memory:
32584 pid, 256 memory
-1 pid, 256 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 1024 memory
-1 pid, 1024 memory
-1 pid, 1024 memory
32585 pid, 1024 memory
32585 pid, 512 memory
-1 pid, 512 memory
-1 pid, 1024 memory
-1 pid, 8192 memory
Before freeing the allocated memory, the memory is (-1 means empty):
32584 pid, 256 memory size
-1 pid, 256 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 1024 memory size
-1 pid, 1024 memory size
-1 pid, 1024 memory size
32585 pid, 1024 memory size
32585 pid, 512 memory size
-1 pid, 512 memory size
-1 pid, 1024 memory size
-1 pid, 8192 memory size
After freeing the memory, new memory:
32584 pid, 256 memory
-1 pid, 256 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 1024 memory
-1 pid, 1024 memory
-1 pid, 2048 memory
32585 pid, 512 memory
-1 pid, 512 memory
-1 pid, 1024 memory
-1 pid, 8192 memory
```

```
Before freeing the allocated memory, the memory is (-1 means empty):
32584 pid, 256 memory size
-1 pid, 256 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 1024 memory size
-1 pid, 1024 memory size
-1 pid, 2048 memory size
32585 pid, 512 memory size
-1 pid, 512 memory size
-1 pid, 1024 memory size
-1 pid, 8192 memory size
After freeing the memory, new memory:
32584 pid, 256 memory
-1 pid, 256 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 1024 memory
-1 pid, 1024 memory
-1 pid, 4096 memory
-1 pid, 8192 memory
Before freeing the allocated memory, the memory is (-1 means empty):
32584 pid, 256 memory size
-1 pid, 256 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 1024 memory size
-1 pid, 1024 memory size
-1 pid, 4096 memory size
-1 pid, 8192 memory size
After freeing the memory, new memory:
-1 pid, 512 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 1024 memory
-1 pid, 1024 memory
-1 pid, 4096 memory
-1 pid, 8192 memory
Before freeing the allocated memory, the memory is (-1 means empty):
-1 pid, 512 memory size
```

```
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 1024 memory size
-1 pid, 1024 memory size
-1 pid, 4096 memory size
-1 pid, 8192 memory size
After freeing the memory, new memory:
-1 pid, 1024 memory
32584 pid, 512 memory
32584 pid, 512 memory
32584 pid, 1024 memory
-1 pid, 1024 memory
-1 pid, 4096 memory
-1 pid, 8192 memory
Before freeing the allocated memory, the memory is (-1 means empty):
-1 pid, 1024 memory size
32584 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 1024 memory size
-1 pid, 1024 memory size
-1 pid, 4096 memory size
-1 pid, 8192 memory size
After freeing the memory, new memory:
-1 pid, 1024 memory
-1 pid, 512 memory
32584 pid, 512 memory
32584 pid, 1024 memory
-1 pid, 1024 memory
-1 pid, 4096 memory
-1 pid, 8192 memory
Before freeing the allocated memory, the memory is (-1 means empty):
-1 pid, 1024 memory size
-1 pid, 512 memory size
32584 pid, 512 memory size
32584 pid, 1024 memory size
-1 pid, 1024 memory size
-1 pid, 4096 memory size
-1 pid, 8192 memory size
After freeing the memory, new memory:
-1 pid, 2048 memory
32584 pid, 1024 memory
-1 pid, 1024 memory
-1 pid, 4096 memory
```

```
-1 pid, 8192 memory
Before freeing the allocated memory, the memory is (-1 means empty):
-1 pid, 2048 memory size
32584 pid, 1024 memory size
-1 pid, 1024 memory size
-1 pid, 4096 memory size
-1 pid, 8192 memory size
After freeing the memory, new memory:
-1 pid, 16384 memory
```