SUSTech CS302 OS Lab8 Report

Title: Memory allocation and segmentation paging mechanism

Name: ____唐千栋____, **Student ID:** __11612730______

Time: __2019___Year __4__Month __23__Day

Experimental Environment: linux, cpp

Objective: Master the process of memory allocation, understand the causes of fragmentation; learn how to reduce fragmentation; understand the mechanism of segmentation and paging.

Deadline: 11:59 AM, 2019-04-24

Summit by: Blackboard

Task：

Task 1.Understand the source codes

Task 2.Edit and modify the source codes

Experiments:

1. fundamental：

- **What is Uniprogramming:**

  Uniprogramming allows only one program sits in main memory at a time.

- **What is the shortcoming of Uniprogramming:**

  Uniprogramming is not efficient since it only allows one program at a time. It is not powerful.

- **What is Multiprogramming:**

  Multiprogramming allows many programs sit in main memory at a time.

- What is the shortcoming of Multiprogramming:

  Multiprogramming is more difficult because for multiprogramming, we need to consider fragmentation problem.

- What is the segmentation mechanism and its advantages & disadvantages:

  Segmentation is that each job is divided into several segments of different sizes, one for each module that contains pieces that perform related functions.

  Advantages: (1) No internal fragmentation (2) Segment table consumes less space in comparison to page table in paging.

  Disadvantages: (1) As processes are loaded and removed from the memory, the free memory space is broken into little spaces, caused external fragmentation.

- What is the paging mechanism and its advantages & disadvantages:

  Paging is a memory management scheme which a computer stores and retrieves data from secondary storage for use in main memory. In this scheme, the operating system retrieves data from secondary storage in same-size blocks called pages.

  Advantages: (1) Paging reduces external fragmentation (2) Paging is simple and efficient

  Disadvantages: (1) Paging can cause internal fragmentation (2) Paging requires extra memory space

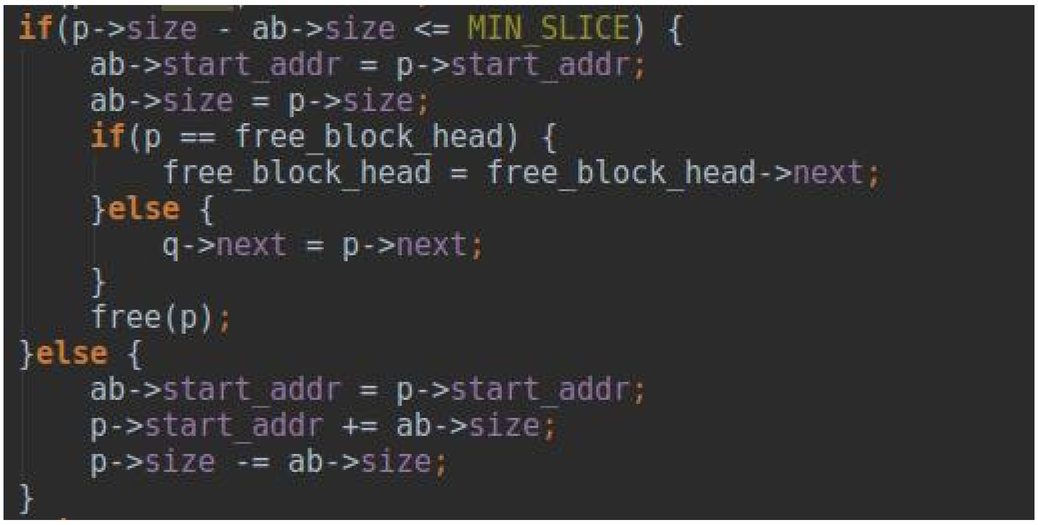
2. Memory management(based on the code):

- What kind of data structure is the memory block used for storage?

  Linked list

□ How to reduce the occurrence of internal fragmentation when allocating memory?

Briefly explain why this method works.



```
if(p->size - ab->size <= MIN_SLICE) {
    ab->start_addr = p->start_addr;
    ab->size = p->size;
    if(p == free_block_head) {
        free_block_head = free_block_head->next;
    }else {
        q->next = p->next;
    }
    free(p);
}else {
    ab->start_addr = p->start_addr;
    p->start_addr += ab->size;
    p->size -= ab->size;
}
```

When the difference between free memory block size and the process memory size is smaller than minimal internal slice, we would directly give the process the free memory block size. By giving the minimal internal slice, we can ensure there is no internal fragmentation whose size is greater than minimal threshold.

□ What kind of strategy does the original program take to allocate memory?

Briefly describe the benefits of doing so.

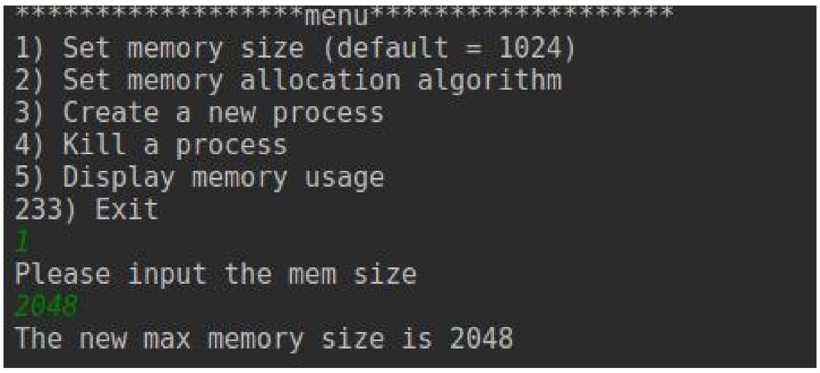
It take First Fit strategy.

It tends to use memory space in low memory address, and the memory in high memory address is rarely used. It is beneficial for the latter process which needs large memory size. It is fast because it searches only the first block which is enough to assign a process. And it is simple.

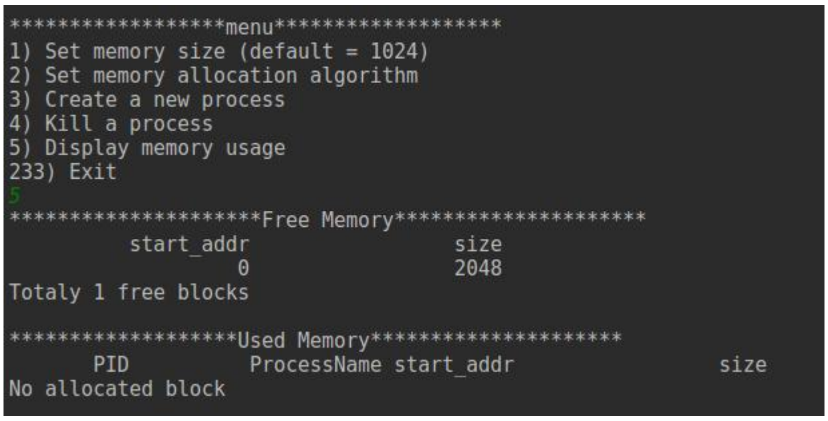
3. Code result, screenshot and your comments:

For First Fit:

A)(1, 2048), 5, (3, 1024), 3(1023), 5, (4, 1), 5, (4, 2), 233



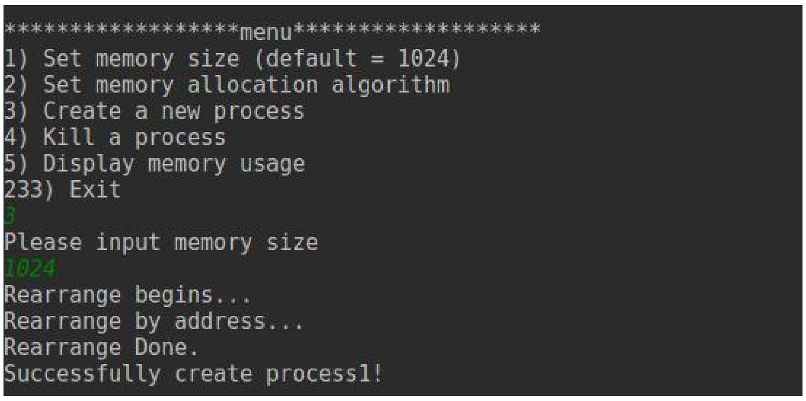
```
******************menu*******************
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
1
Please input the mem size
2048
The new max memory size is 2048
```



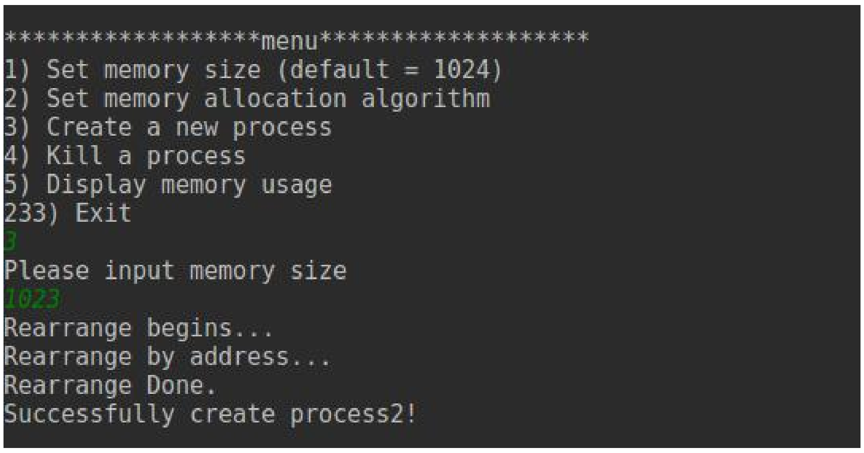
```
*****************menu******************
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
5
********************Free Memory********************
          start_addr                size
                   0                2048
Totaly 1 free blocks

******************Used Memory********************
       PID          ProcessName start_addr                 size
No allocated block
```

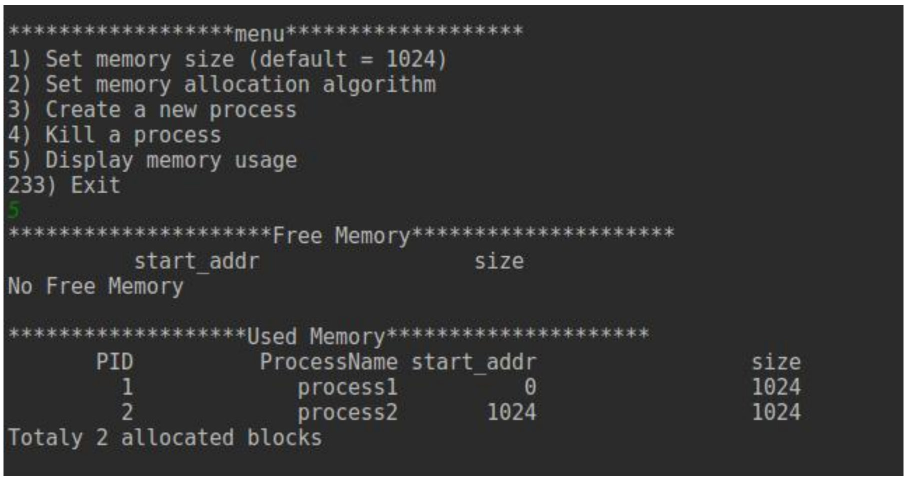
We can see that free memory size is 2048



```
*****************menu******************
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
3
Please input memory size
1024
Rearrange begins...
Rearrange by address...
Rearrange Done.
Successfully create process1!
```



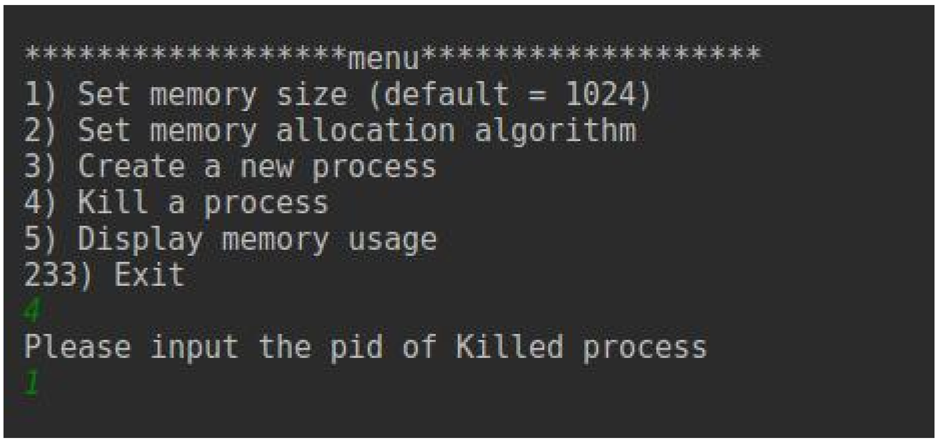
```
*****************menu******************
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
3
Please input memory size
1023
Rearrange begins...
Rearrange by address...
Rearrange Done.
Successfully create process2!
```



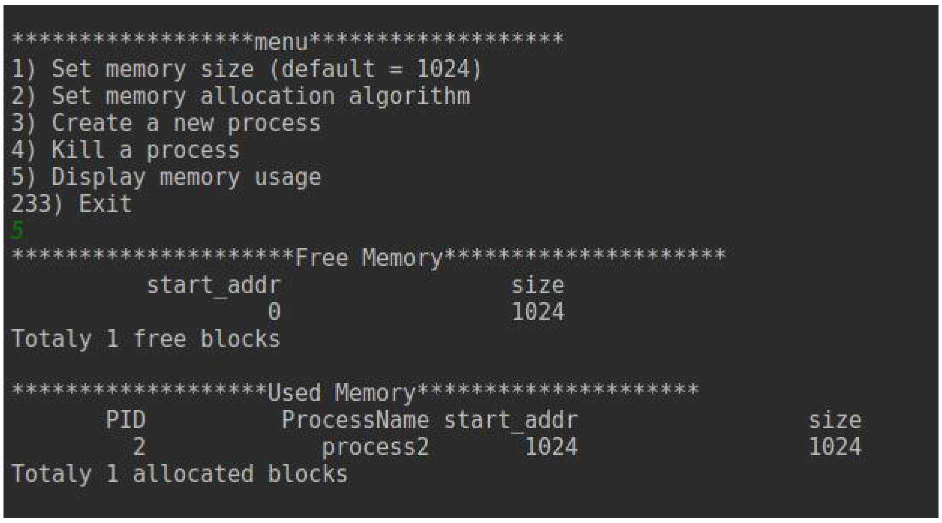
```
*****************menu******************
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
5
********************Free Memory********************
          start_addr                size
No Free Memory

******************Used Memory*******************
       PID          ProcessName start_addr                 size
         1             process1          0                 1024
         2             process2       1024                 1024
Totaly 2 allocated blocks
```

Process2’ size is 2014 cause 2014-2013<=MIN_SLICE



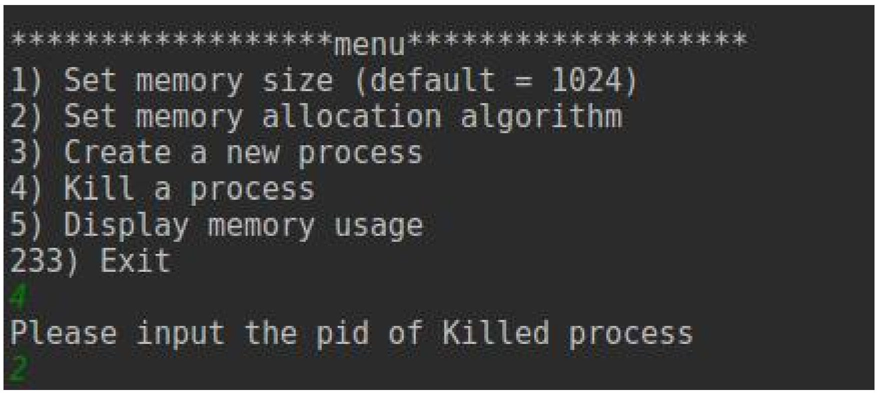
```
*****************menu******************
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
4
Please input the pid of Killed process
1
```



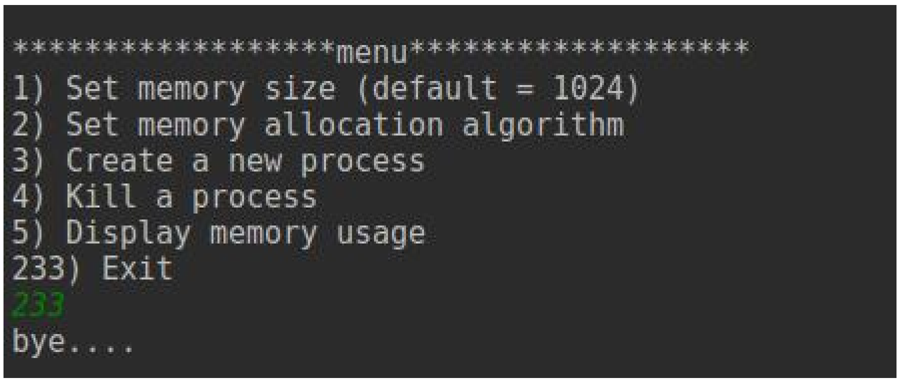
```
*****************menu******************
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
5
********************Free Memory********************
          start_addr                size
                   0                1024
Totaly 1 free blocks

******************Used Memory*******************
       PID          ProcessName start_addr                 size
         2             process2       1024                 1024
Totaly 1 allocated blocks
```

Process1 is killed

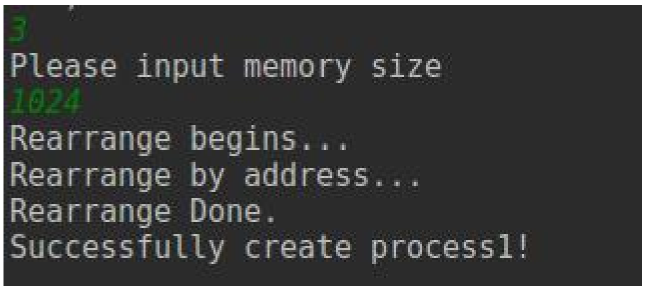


```
*****************menu******************
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
4
Please input the pid of Killed process
2
```

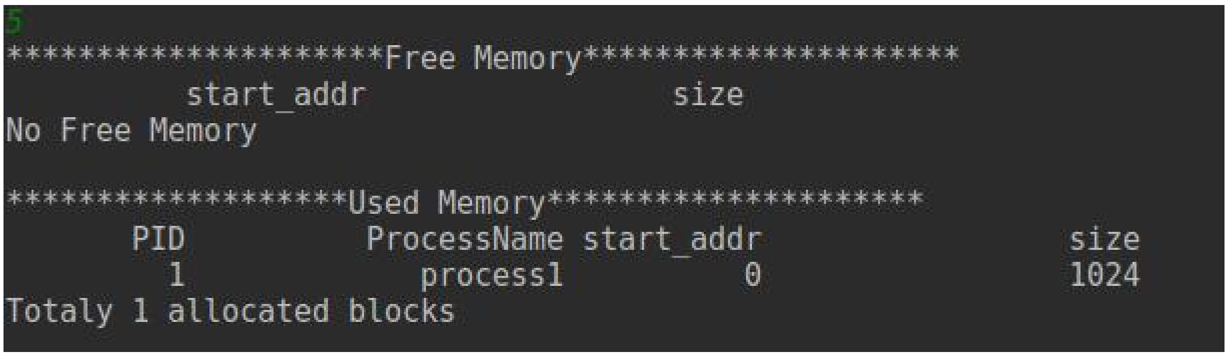


```
*****************menu******************
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
233
bye....
```

B) (3, 1024), 5, (3, 1), (4, 1), 5, (4, 2), 233

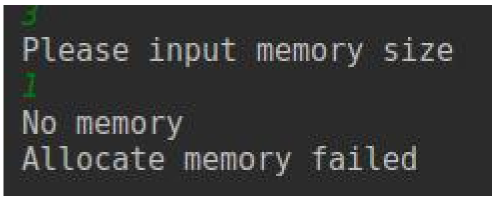


```
3
Please input memory size
1024
Rearrange begins...
Rearrange by address...
Rearrange Done.
Successfully create process1!
```



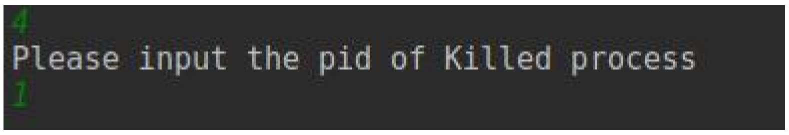
```
5
********************Free Memory*********************
         start_addr                size
No Free Memory

******************Used Memory********************
      PID         ProcessName start_addr                size
        1            process1        0                  1024
Totaly 1 allocated blocks
```

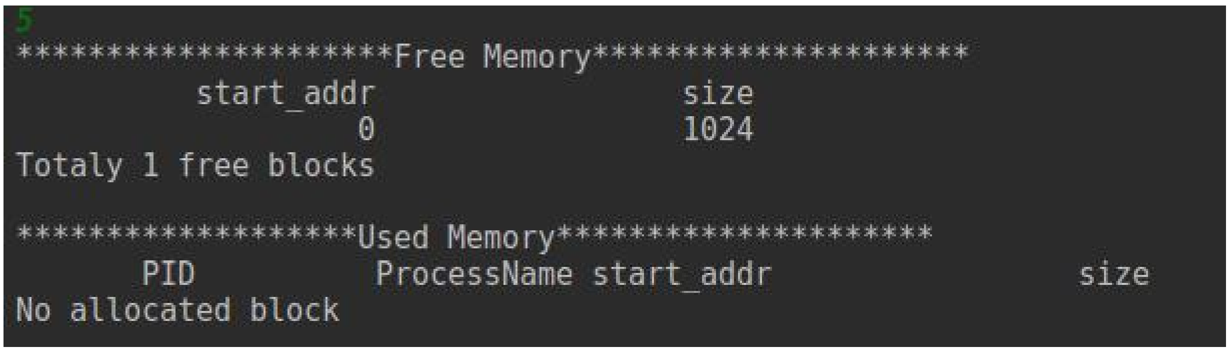


```
3
Please input memory size
1
No memory
Allocate memory failed
```

Process2 can not be allocated because no memory left.

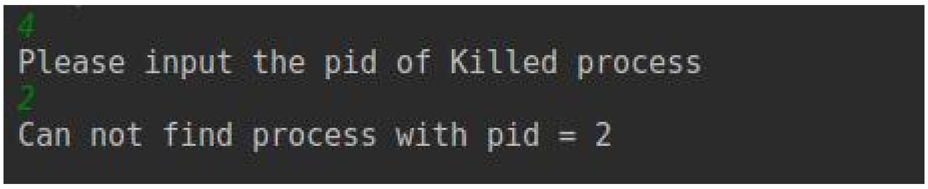


```
4
Please input the pid of Killed process
1
```



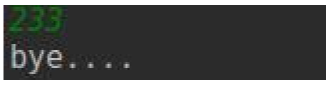
```
5
*********************Free Memory*********************
          start_addr                 size
                   0                 1024
Totaly 1 free blocks

*******************Used Memory*********************
       PID          ProcessName start_addr                size
No allocated block
```



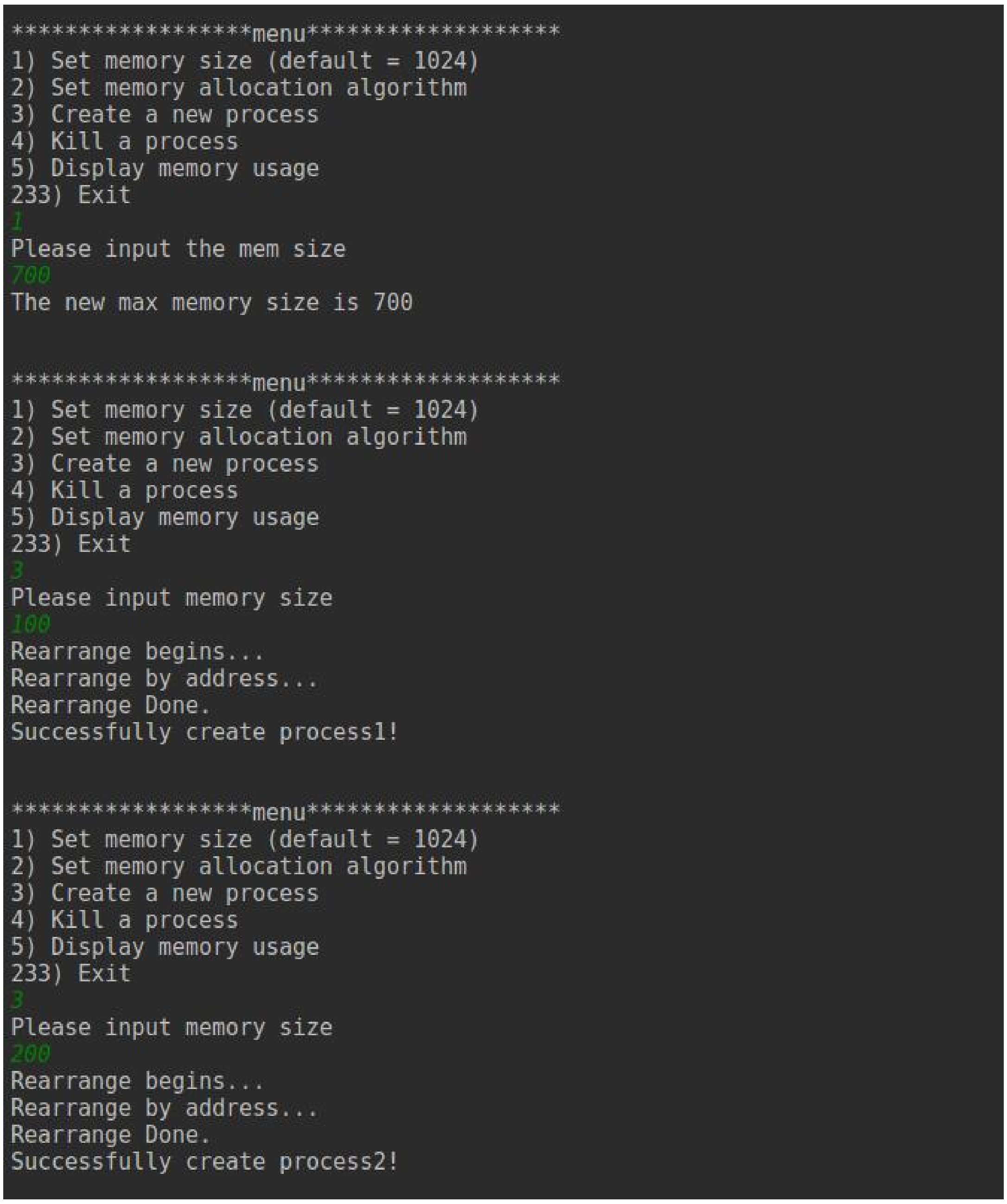
```
4
Please input the pid of Killed process
2
Can not find process with pid = 2
```

There is no process2 in allocated memory.



```
233
bye....
```

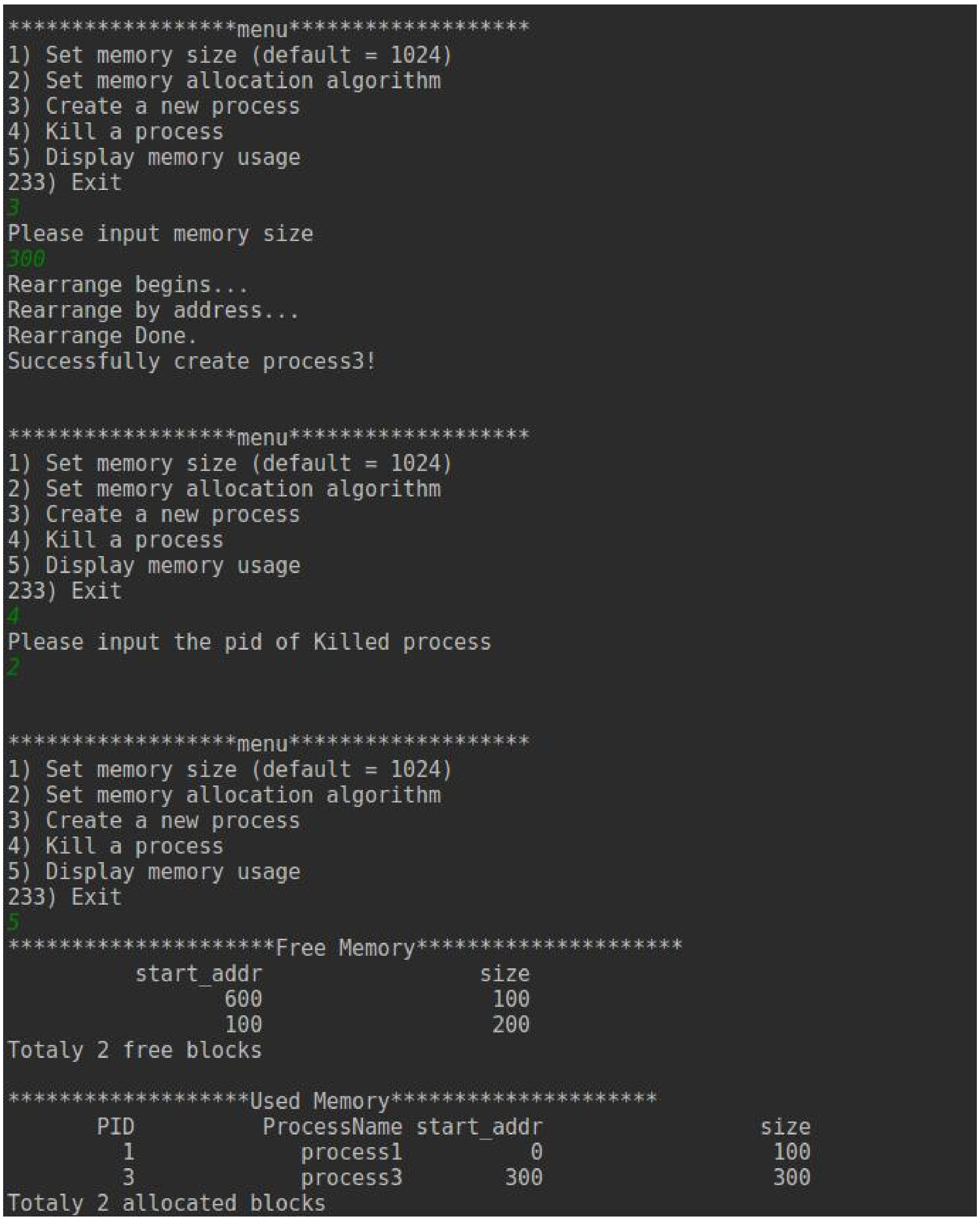
C) (1, 700), (3, 100), (3, 200), (3, 300), (4, 2), 5, (3, 300), 5, (4, 3), 5, 233



```
*****************menu******************
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
1
Please input the mem size
700
The new max memory size is 700


*****************menu******************
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
3
Please input memory size
100
Rearrange begins...
Rearrange by address...
Rearrange Done.
Successfully create process1!


*****************menu******************
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
3
Please input memory size
200
Rearrange begins...
Rearrange by address...
Rearrange Done.
Successfully create process2!
```



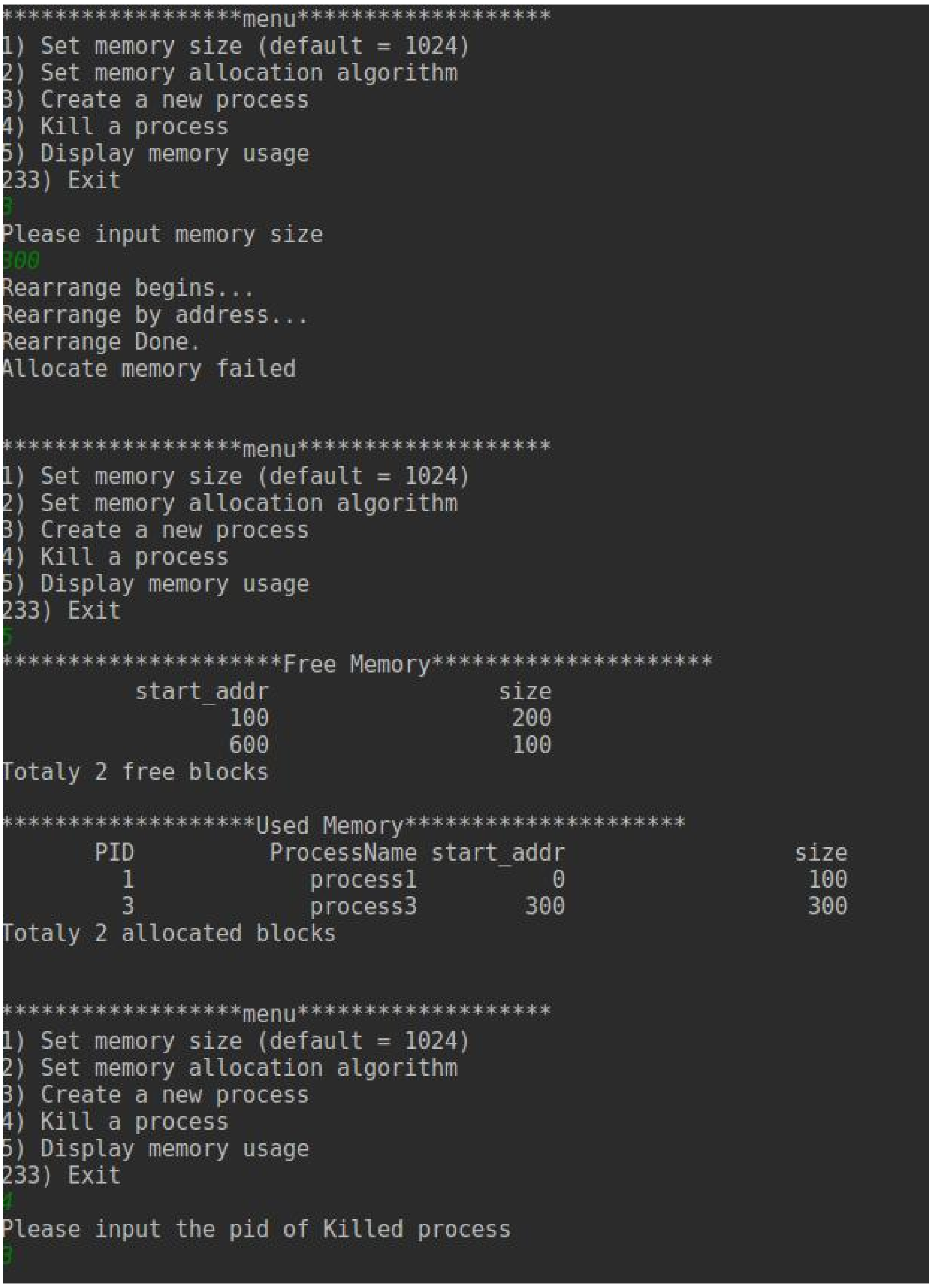
```
*****************menu******************
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
3
Please input memory size
300
Rearrange begins...
Rearrange by address...
Rearrange Done.
Successfully create process3!


*****************menu******************
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
4
Please input the pid of Killed process
2


*****************menu******************
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
5
********************Free Memory********************
          start_addr                 size
                 600                  100
                 100                  200
Totaly 2 free blocks

******************Used Memory********************
       PID          ProcessName start_addr                size
         1             process1          0                 100
         3             process3        300                 300
Totaly 2 allocated blocks
```

Process 2 is killed and there are two processes, process1 and process3



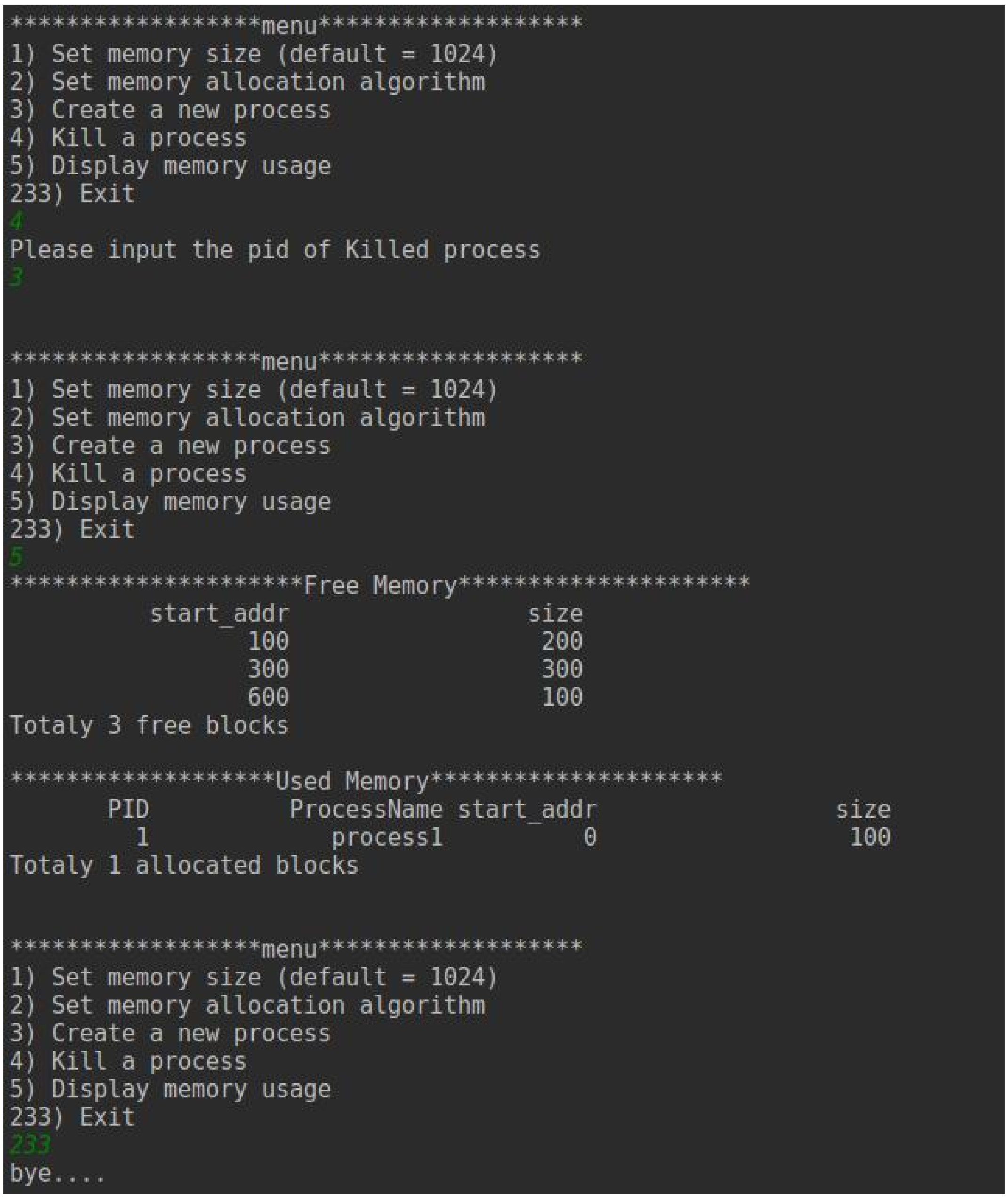
```
*****************menu******************
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
3
Please input memory size
300
Rearrange begins...
Rearrange by address...
Rearrange Done.
Allocate memory failed


*****************menu******************
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
5
********************Free Memory********************
          start_addr                 size
                 100                  200
                 600                  100
Totaly 2 free blocks

*******************Used Memory********************
       PID          ProcessName start_addr                 size
         1             process1          0                  100
         3             process3        300                  300
Totaly 2 allocated blocks


*****************menu******************
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
4
Please input the pid of Killed process
3
```

Process3 with memory 300 size can not be allocated cause there is no continuous free memory with 300 size.



```
*****************menu******************
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
4
Please input the pid of Killed process
3


*****************menu******************
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
5
*********************Free Memory*********************
          start_addr                 size
                 100                  200
                 300                  300
                 600                  100
Totaly 3 free blocks

******************Used Memory*********************
       PID          ProcessName start_addr                 size
         1             process1          0                  100
Totaly 1 allocated blocks


*****************menu******************
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
233
bye....
```

4. Problems you meet and your solutions

- Problem1:

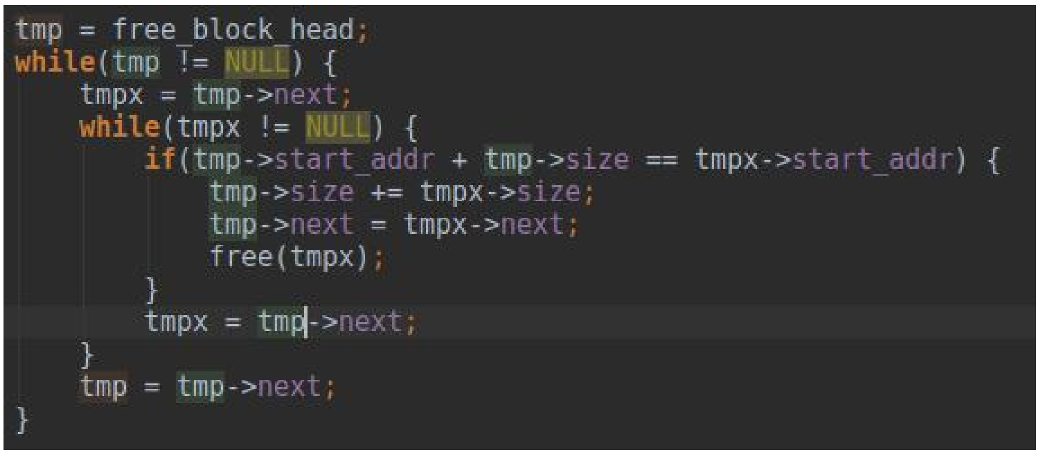
  Segmentation fault

- Solution1:

  read null pointer value.

- Problem2:

Endless loop



```
tmp = free_block_head;
while(tmp != NULL) {
    tmpx = tmp->next;
    while(tmpx != NULL) {
        if(tmp->start_addr + tmp->size == tmpx->start_addr) {
            tmp->size += tmpx->size;
            tmp->next = tmpx->next;
            free(tmpx);
        }
        tmpx = tmp->next;
    }
    tmp = tmp->next;
}
```

- **Solution2:**

Change tmpx=tmp->next to tmpx=tmpx->next

Conclusion:

I learn about First Fit, Best Fit and Worst Fit memory management algorithm.