OS Lab8 Memory Management Guide

1. Basic requirements of code (50+10 points)

   Understand the given code and complete it. So that it can achieve basic memory management goals:

   1) allocate a given memory space to a process (15 points)
   2) kill a process and release the memory space the process kept(15 points)
   3) show the memory usage (5 points)
   4) realize three kinds of algorithms--First fit、Best fit、worst fit (15 points)
   5) realize Buddy system (bonus, 10 points)

2. Report (50 points)

3. Things help you to do this lab:

   a) How to make your vim powerful?

   i. Auto complement: https://www.linuxidc.com/Linux/2017-02/141088.htm
   ii. Supporting Chinese character: https://www.zhihu.com/question/22363620
   iii. Write more code

   b) More explanation about code:

   i. My idea: Keep a list of free-block, which stores memory blocks we can use. And also keep a list of allocated-block, which stores memory blocks we have allocated. When we allocated a memory block to a process, we also need to store the pid of this process. So when we want to kill process, we can find the correct block.

   ii. Two import functions you need to complete:

      1. int allocate_mem(allocated_block *ab) which give a block memory space

      2. int free_mem(allocated_block *ab) which release the memory space for this block

      For allocate, you should use your own algorithm to allocate memory space. That means you need to do something on the free-block list.

      For free, when we free the memory space, we need to add the block to the free-block list. At this time, you also need to change the list by your own algorithm.

iii. Some other things:

1. typedef pair<int, string> My_algo; This line define my own type called My_algo. You can declare a variable of this type. The first component is an integer, the second component is a string. That means:

```
My_algo algo;

algo = make_pair(0, “FirstFit”);

printf(“%d %s\n”, algo.first, algo.second);
```

After running this code, you will get result: 0 FirstFit

make_pair is a function, which you don’t need to write by your own.

2. NULL. When you deal with pointers, please pay attention to NULL. That is, only when a pointer is not NULL, it can has next element.

4. Some Test cases:

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

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

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