ITP 30002-XX Operating System, Spring 2020

Homework 5

Your name : Lee EunJong, Student ID : 21700556, Email address : 21700556@handong.edu

1. Introduction

HW5's main topic is that heap memory space management which is using the doubly linked list. smalloc2.0 offer dynamic memory allocation which is using best-fit algorithm(smalloc), merge unused memory as much as possible(sfree), print memory using status, change allocation size, reduce break point.

2. Approach

Task1 (smalloc using best-fit algorithm)

To implement smalloc that using best-fit algorithm, I searched all nodes in doubly linked list, and find minimum available data size. Then, allocate memory to that space.

Task2 (sfree)

First, change target node's status to unused, and find all continuously unused status node, and update data size of start node of continuous unused node and change links

Task3 (print_mem_uses)

I understand that the amount of memory retained by smalloc so far is that pagesize * retained page count. Also, I understand allocated memory at this moment is total using space, and not currently allocate space is total available space. I printed this information by using fprintf funcion and stderr.

Task4 (srealloc)

When srealloc function is called, I think there can exist 2 cases. One case is new size < original size. In this case just split original space. Another case is new size > original size. If available space exists at the next node, I used the space, and split unused space. If available space don't exist at the next node, I call smalloc function to allocate memory that size is new size, and I used memcpy function to migrate original data to the new space. then change links.

Task5 (sshrink)

I found continuous unused space from break point. I found start point of the continuous unused space, and I changed breakpoint by using brk function that parameter is the start point of the continuous unused space. then change links.

3. Evaluation

Test1, test2, test3 works well, so I think smalloc.ver2 is updated successfully.

Test4.c test

I think retained but not used memory is bigger, the efficiency is lower.

I found that when the blank space in front is greater than the empty space in the back, and allocation request of small space and allocation request of large space requested sequentially, first-fit algorithm works poorly. For example, suppose the memory situation is

Unused : 2000

Busy : 800

Unused : 1200

when smalloc(1000) and smalloc(1800) is executed, if we use ver1.0(first-fit), retained but not used memory size is 4400. However, if we use ver2.0(best-fit), retained but not used memory size is 336. We can show that ver2.0 algorithm works more efficiently

```
smalloc(1800):0x20e3020
==================== sm_containers ====================
  0:0x20e2020:  Busy:    1000:00 00 00 00 00 00 00 00
  1:0x20e2428:Unused:     968:00 00 00 00 00 00 00 00
  2:0x20e2810:  Busy:     800:00 00 00 00 00 00 00 00
  3:0x20e2b50:Unused:    1200:00 00 00 00 00 00 00 00
  4:0x20e3020:  Busy:    1800:00 00 00 00 00 00 00 00
  5:0x20e3748:Unused:    2232:00 00 00 00 00 00 00 00
```

```
the amount of memory retained smalloc so far : 4096
the amount of memory allocated by smalloc at this moment : 3600
the amount of memory retained by smalloc but not currently llocated : 336
```

4. Discussion

In my implement, I search all nodes in the linked list (time complexity : O(n)). however, if the nodes sorted by data size, time complexity to search is O(logn)(we can use binary search). Also, insertion time will be O(logn), because we can find location by using binary search, and just put data and change links. So, I think if we use best-fit algorithm that uses sorting, it works more efficiently.

I think if initialize and allocate function (scalloc) is added, it will be better.

I didn't know how malloc function works although I used the function many times. However, during I do this homework, I learned how malloc function works, and I understand memory structure more clearly

5. Conclusion

I upgrade smalloc ver1.0 to ver2.0 which has more functionality. By using linked list and change its structure, the above functions could be implemented. I understand more about dynamic memory management.