ITP 30002-01 Operating System, Spring 2020
Homework 4
Name : Min Jae Yi (이민재) **Student number** : 21800511 **Email Address** : 21800511@handong.edu

1. Introduction & Approach

The main function that should be implemented is as below

<u>*Smalloc2.0.c*</u>
a. implementing best-fit algorithm
b. merge unused container while free() happens
c. print memory size information
d. implement memory resizing
e. implement shrinking program break point

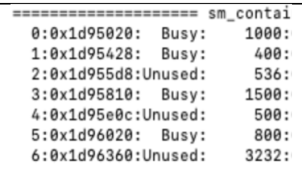
Table 1 : problem define

Brief approach for each problem. (matches with Table 1 For example Table 2. a refers the Table 1. a solutions)

a. Look on every container and find the minimum size of container that can be fit. (Same as finding min value)
b. Erase other containers which is connected to the target container and merge the size. (Use doubly linked list. First look on left side and merge. Next look on right side and merge)
c. Look on every container and count.
d. Look on the possible maximum size and if the size is available extend, and if not free the current container and malloc with the memory copy. (When memory needs reallocation because of the exceed limit use smalloc() and memcpy(). If the memory does not exceed merge the unused container and split with sm_container_split())
e. Look on the latest used container and give the breakpoint where is the end of latest used container. (Use brk() to set the breakpoint)

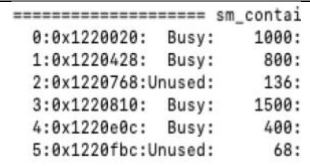
Table 2 : problem approach

3. Evaluation



```
==================== sm_contai
  0:0x1d95020:  Busy:    1000:
  1:0x1d95428:  Busy:     400:
  2:0x1d955d8:Unused:     536:
  3:0x1d95810:  Busy:    1500:
  4:0x1d95e0c:Unused:     500:
  5:0x1d96020:  Busy:     800:
  6:0x1d96360:Unused:    3232:
```

First-fit



```
==================== sm_contai
  0:0x1220020:  Busy:    1000:
  1:0x1220428:  Busy:     800:
  2:0x1220768:Unused:     136:
  3:0x1220810:  Busy:    1500:
  4:0x1220e0c:  Busy:     400:
  5:0x1220fbc:Unused:      68:
```

Best-fit

Table 3 : function 1

Table 3 shows the difference between first-fit algorithm and best-fit algorithm. Left image shows three Unused containers and right image shows two Unused containers. This shows that Best-fit can manage memory more optimized however it took more time since it should explore every container.

To check the performance I have compare the smalloc and smalloc2 with **Test3.c** by using the "time" command.

real 0m0.002s	real 0m0.001s
user 0m0.002s	user 0m0.001s
sys 0m0.000s	sys 0m0.000s
Smalloc v2	**Smalloc v1**

Table 4 : Prefornmance of smalloc

This shows that smalloc2 takes much more time than smalloc. However, smalloc2 has high performance optimizing the memory space. So execution time and managing optimizing has trade-off.

4. Discussion

I had some trouble while implementing the merging unused container. My first approach was finding the closest left and right used container. However, since there were some complicate logics there were many bugs. I found that I can implement merging just by comparing one by one and it got solved.