Smalloc

Simple Dynamic Memory Allocation Library

21700583 JiHyun Lee

1. Version1.1

A. Implement

print_sm_uses( )

Print_sm_uses() print out three functions;1. The amount of memory retained by smalloc so far, 2. The amount of memory allocated by smalloc 3. The amount of memory retained by smalloc but not currently allocated. To implement this, there is iterator that collect amount of total retained memory and allocated memory. Using status variable in struct, distinguishes allocated or not. Not allocated memory is just subtraction result of (total sum – allocated_momry_sum). After iterating, print out three information.

```
==================== sm_containers ====================
  0:0xe67020:  Busy:    1000:00 00 00 00 00 00 00 00
  1:0xe67428:Unused:     968:00 00 00 00 00 00 00 00
  2:0xe67810:  Busy:    1000:00 00 00 00 00 00 00 00
  3:0xe67c18:Unused:    1000:00 00 00 00 00 00 00 00
  4:0xe68020:  Busy:    2500:00 00 00 00 00 00 00 00
  5:0xe68a04:Unused:    1532:00 00 00 00 00 00 00 00
=========================================================
Total retained size for smalloc :     8000
Total allocated size             :     4500
Retaind but not allocated size   :     3500
```

Firstfit to Bestfit

Version 1.0 was implemented as Firstfit. In Version 1.1 it uses Bestfit algorithm. In smalloc( ), by doing iterate, if needed size is smaller than hole size and if that hole size is smaller than so far, than saved that hole as a bestfit hole. After iterating, allocate memory to founded Best fit hole.

Test

This is result of Firstfit and Bestfit each.

Firstfit

```
==================== sm_containers ====================
=======================================================
smalloc(5000)
==================== sm_containers ====================
  0:0x23a1020:  Busy:    5000:00 00 00 00 00 00 00 00
  1:0x23a23c8:Unused:    3128:00 00 00 00 00 00 00 00
=======================================================
smalloc(3500)
==================== sm_containers ====================
  0:0x23a1020:  Busy:    5000:00 00 00 00 00 00 00 00
  1:0x23a23c8:Unused:    3128:00 00 00 00 00 00 00 00
  2:0x23a3020:  Busy:    3500:00 00 00 00 00 00 00 00
  3:0x23a3dec:Unused:     532:00 00 00 00 00 00 00 00
=======================================================
smalloc(10)
==================== sm_containers ====================
  0:0x23a1020:  Busy:    5000:00 00 00 00 00 00 00 00
  1:0x23a23c8:  Busy:      10:00 00 00 00 00 00 00 00
  2:0x23a23f2:Unused:    3086:00 00 00 00 00 00 00 00
  3:0x23a3020:  Busy:    3500:00 00 00 00 00 00 00 00
  4:0x23a3dec:Unused:     532:00 00 00 00 00 00 00 00
=======================================================
```

When allocating 10bytes, first fit algorithm allocates memory to 3128 size hole.

Bestfit

```
smalloc(5000)
==================== sm_containers ====================
  0:0x221e020:  Busy:    5000:00 00 00 00 00 00 00 00
  1:0x221f3c8:Unused:    3128:00 00 00 00 00 00 00 00
=======================================================
smalloc(3500)
==================== sm_containers ====================
  0:0x221e020:  Busy:    5000:00 00 00 00 00 00 00 00
  1:0x221f3c8:Unused:    3128:00 00 00 00 00 00 00 00
  2:0x2220020:  Busy:    3500:00 00 00 00 00 00 00 00
  3:0x2220dec:Unused:     532:00 00 00 00 00 00 00 00
=======================================================
smalloc(10)
==================== sm_containers ====================
  0:0x221e020:  Busy:    5000:00 00 00 00 00 00 00 00
  1:0x221f3c8:Unused:    3128:00 00 00 00 00 00 00 00
  2:0x2220020:  Busy:    3500:00 00 00 00 00 00 00 00
  3:0x2220dec:  Busy:      10:00 00 00 00 00 00 00 00
  4:0x2220e16:Unused:     490:00 00 00 00 00 00 00 00
=======================================================
```

When allocating 10bytes, Best fit algorithm allocates memory to 532 size hole.

2. Version 1.2

A. Implement

sm_unused_container

sm_unused_container is linked list of unused memory. In sm_container_split function, there are part for managing unused linked list. In sm_container_split_function, divide various circumstance as 1. no initialed unused container(sm_unused_containers == NULL), 2. unused linked list is existing and splited hole is first part of retained memory 3. unused linked is existing list and splited hole is last part of retained memory 4. unused linked list is existing and splited hole is middle part of retained memory.
According to each circumstance, apply different managing way.

And if print out the linked list, it shows like below.

```
==================== sm_containers ====================
  0:0xf78028:  Busy:    1000:00 00 00 00 00 00 00 00
  1:0xf78438:Unused:    3500:00 00 00 00 00 00 00 00
  2:0xf79028:  Busy:    2500:00 00 00 00 00 00 00 00
  3:0xf79a14:  Busy:    1000:00 00 00 00 00 00 00 00
  4:0xf79e24:Unused:     476:00 00 00 00 00 00 00 00
=======================================================
unused linked list
    3500 ->     476 ->
```

Merge unused memory

Merging adjacent memory is effective to managing memory. To implement this, there is merge function, and it invoked after sfree() function. In merge function, finding adjacent unused memory using unused_container linked. After finding adjacent unused container, then frontal container merge next container.

Without merge function

```
smalloc(2500)
==================== sm_containers ====================
  0:0x19a0028:  Busy:    2000:00 00 00 00 00 00 00 00
  1:0x19a0820:Unused:    2016:00 00 00 00 00 00 00 00
  2:0x19a1028:  Busy:    2500:00 00 00 00 00 00 00 00
  3:0x19a1a14:Unused:    1516:00 00 00 00 00 00 00 00
=======================================================
sfree(0x19a0028)
==================== sm_containers ====================
  0:0x19a0028:Unused:    2000:00 00 00 00 00 00 00 00
  1:0x19a0820:Unused:    2016:00 00 00 00 00 00 00 00
  2:0x19a1028:  Busy:    2500:00 00 00 00 00 00 00 00
  3:0x19a1a14:Unused:    1516:00 00 00 00 00 00 00 00
=======================================================
smalloc(1000)
```

With merge function

```
smalloc(2500)
==================== sm_containers ====================
  0:0x2207028:  Busy:    2000:00 00 00 00 00 00 00 00
  1:0x2207820:Unused:    2016:00 00 00 00 00 00 00 00
  2:0x2208028:  Busy:    2500:00 00 00 00 00 00 00 00
  3:0x2208a14:Unused:    1516:00 00 00 00 00 00 00 00
=======================================================
sfree(0x2207028)
==================== sm_containers ====================
  0:0x2207028:Unused:    4540:00 00 00 00 00 00 00 00
  1:0x2208028:  Busy:    2500:00 00 00 00 00 00 00 00
  2:0x2208a14:Unused:    1516:00 00 00 00 00 00 00 00
=======================================================
```

After sfree, unused memory becomes 4540 bytes.

3. How to improve

Double linked list for unused memory

Unused linked list was implemented as one way linked list. This was inconvenience and less efficient when merging unused memory than a double linked list. I implement merge function by iterating all linked list. Just examine freed node would be efficient way. However, to doing so, I have to iterate the linked list to find frontal node of freed node. If it was implemented as Doubled linked list, then merging will be much easier and efficient.

Compaction

After implement all things, I thought compaction of unused memory can be implemented quiet easily. By using unused linked list, and retained linked list, and swapping nodes well, it will be implemented. If smalloc api has compaction function, this would be useful for managing memory.