6/26HW malloc.c

1. best_fitの実装

＜実装の流れ＞

①**best_fit**検索...my_mallocのループを更新し要求されたサイズを収容できる最小のブロックを見つけるようにする

→断片化を最小限に抑える

②割り当てと解放...変更なし。メモリを正しく割り当て空きリストを管理する

③edge case …適切なブロックが見つからない場合mmap_from_systemで新しいブロックの要求＋割り当てを試みる

→新しい割り当てのための場所を確保

```
./malloc_challenge.bin
Welcome to the malloc challenge!
size_of(uint8_t *) = 8
size_of(size_t) = 8
Running tests...
Finished!

====================================================
Challenge #1    |   simple_malloc =>       my_malloc
--------------- + --------------- => ---------------
       Time [ms]|              13 =>            1015
Utilization [%] |              70 =>              70
====================================================
Challenge #2    |   simple_malloc =>       my_malloc
--------------- + --------------- => ---------------
       Time [ms]|               4 =>             644
Utilization [%] |              40 =>              40
====================================================
Challenge #3    |   simple_malloc =>       my_malloc
--------------- + --------------- => ---------------
       Time [ms]|              75 =>             783
Utilization [%] |               9 =>              51
====================================================
Challenge #4    |   simple_malloc =>       my_malloc
--------------- + --------------- => ---------------
       Time [ms]|           19886 =>            7280
Utilization [%] |              15 =>              72
====================================================
Challenge #5    |   simple_malloc =>       my_malloc
--------------- + --------------- => ---------------
       Time [ms]|           14346 =>            4045
Utilization [%] |              15 =>              75

Challenge done!
Please copy & paste the following data in the score sheet!
1015,70,644,40,783,51,7280,72,4045,75,
```

2. free_list_bin実装の追加

free_list_bin : 特定サイズ範囲に対応する複数の空きリストを使ってメモリ割り当てと解放の効率をあげる考え。

＜実装の流れ＞

各binに対応する空きリストを定義する

my_initialize関数で各binの初期化

my_malloc関数で適切なbinを探索

my_free関数で適切なbinに空きメモリブロックを追加

```
./malloc_challenge.bin
Welcome to the malloc challenge!
size_of(uint8_t *) = 8
size_of(size_t) = 8
Running tests...
Finished!

==================================================
Challenge #1    |   simple_malloc =>       my_malloc
--------------- + --------------- => ---------------
       Time [ms]|              13 =>              12
Utilization [%] |              70 =>              70
==================================================
Challenge #2    |   simple_malloc =>       my_malloc
--------------- + --------------- => ---------------
       Time [ms]|               7 =>               7
Utilization [%] |              40 =>              40
==================================================
Challenge #3    |   simple_malloc =>       my_malloc
--------------- + --------------- => ---------------
       Time [ms]|              82 =>               8
Utilization [%] |               9 =>              51
==================================================
Challenge #4    |   simple_malloc =>       my_malloc
--------------- + --------------- => ---------------
       Time [ms]|           16458 =>             287
Utilization [%] |              15 =>              72
==================================================
Challenge #5    |   simple_malloc =>       my_malloc
--------------- + --------------- => ---------------
       Time [ms]|           11408 =>             186
Utilization [%] |              15 =>              75

Challenge done!
Please copy & paste the following data in the score sheet!
12,70,7,40,8,51,287,72,186,75,
```