

## **Задание**

Разработать модель простого аллокатора памяти.

Имеется статический (или динамический, распределяемый 1 раз) массив (буфер), размером N байтов. Требуется разработать следующие функции по выделению и освобождению памяти внутри этого массива:

```
void* my_alloc(size_t size);  
void my_delete(void *ptr);
```

Первая функция выделяет в массиве size байтов и возвращает пользователю указатель на выделенную память или NULL, если свободного запрошенного блока памяти выделить не удалось.

Вторая функция освобождает ранее выделенный блок памяти по указателю ptr или ничего не делает (возможно, кроме вывода информации об ошибке в определенном формате), если указатель не указывает на ранее выделенный блок.

Динамической памятью пользоваться запрещено.

Для управления блоками памяти необходимо определить и использовать собственную структуру MCB.

Разрешается использовать alloc 1 раз для выделения памяти, используемой для распределения пользователю через my\_alloc.

Для реализации используется язык C/C++, программа должна собираться с помощью команды:

```
$make
```

## **Вход и Выход**

Для тестирования должна быть разработана тестовая программа, на входе принимающая из стандартного потока параметры и выводящая карту распределения памяти в стандартный поток и результат выполнения операций в стандартный поток вывода.

## **Формат входного потока**

Первая строка содержит число, равное количеству байт  $100 < N < 10000$ . Следующие строки

содержат одну из команд:

## ALLOC S

Выделить в буфере S байт.

FREE P

Освободить ранее выделенный регион памяти, начинающийся с байта с номером Р (байты нумеруются с 0).

## INFO

Вывести информацию об использовании памяти.

MAP

Вывести карту памяти.

## Формат выходного потока

Каждая команда из входного потока порождает ровно одну строку в выходном потоке. Далее приведены примеры вывода каждой команды.

==== ALLOC ====

+ P

успешное выделение запрошенного размера памяти, начиная с байта с номером  $P$ .

—

неуспешное выделение (знак минус).

=== FREE ===

 $+$ 

—

=== INFO ===

N1 N2 N3 N4

- \* N1 - количество выделенных блоков памяти
- \* N2 - размер памяти выделенной пользователю
- \* N3 - размер свободной памяти
- \* N4 - размер максимального фрагмента свободной памяти, которая может быть выделена (без учета MBR)

=== MAP ===

```
m m m m m u u u u u u f f f f f u u u u
```

строка длиной N на алфавите {m,u,f}, где каждый символ соответствует использованию байта памяти:

- \* m - используется управляющими структурами
- \* u - отдан пользователю
- \* f - свободен

## Порядок отправки решений

Выполненные задания (только исходные тексты программ и скрипты сборки) архивируются в zip-архив с именем, сформированным как lastname.firstname-01.zip

где:

lastname, firstname— фамилия автора и имя записанные латиницей

Пример: petrov.vasily-01.zip

Архив высылается в отдельном письме на адрес [csc-krinkin@osll.ru](mailto:csc-krinkin@osll.ru). Суммарный размер письма не должен превышать 1Мб. Тема письма совпадает с именем файла архива без расширения (например, petrov.vasily-03).

Срок выполнения задания 02/11/2013 23:55

Задачи, высланные после срока или оформленные не по правилам не рассматриваются.

NB: вопросы по заданию и консультации можно получить, написав [csc-krinkin@osll.ru](mailto:csc-krinkin@osll.ru)