**Задание:**

Исследовать два аллокатора памяти: необходимо реализовать два алгоритма

аллокации памяти и сравнить их по следующим характеристикам:

– Фактор использования   
– Скорость выделения блоков   
– Скорость освобождения блоков   
– Простота использования аллокатора

Требуется создать две динамические библиотеки, реализующие два аллокатора, соответственно. Библиотеки загружаются в память с помощью интерфейса ОС (dlopen / LoadLibrary) для работы с динамическими библиотеками. Выбор библиотеки, реализующей аллокатор, осуществляется чтением первого аргумента при запуске программы (argv[1]). Этот аргумент должен содержать путь до динамической библиотеки (относительный или абсолютный).

Если аргумент не передан или по переданному пути библиотеки не оказалось, то указатели на функции, реализующие API аллокатора ниже, должны быть присвоены функциям, которые оборачивают системный аллокатор ОС (mmap / VirtualAlloc) в этот API. Эти аварийные оберточные функции должны быть реализованы внутри программы, которая загружает динамические библиотеки (см. пример на GitHub Gist).

Каждый аллокатор памяти должен иметь функции аналогичные стандартным

функциям malloc и free (realloc, опционально). Перед работой каждый

аллокатор инициализируется свободными страницами памяти, выделенными

стандартными средствами ядра (mmap / VirtualAlloc). Необходимо

самостоятельно разработать стратегию тестирования для определения ключевых характеристик аллокаторов памяти. При тестировании нужно свести к минимуму потери точности из-за накладных расходов при измерении ключевых характеристик, описанных выше.

Каждый аллокатор должен обладать следующим интерфейсом (могут быть отличия в зависимости от особенностей алгоритма):

* Allocator\* allocator\_create(void \*const memory, const size\_t

size) (инициализация аллокатора на памяти memory размера size);

* void allocator\_destroy(Allocator \*const allocator)

(деинициализация структуры аллокатора);

void\* allocator\_alloc(Allocator \*const allocator, const

size\_t size) (выделение памяти аллокатором памяти размера size);

* void allocator\_free(Allocator \*const allocator, void \*const

memory) (возвращает выделенную память аллокатору);

Вариант: Списки свободных блоков (первое подходящее) и алгоритм Мак-Кьюзи-Кэрелса.