ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»**

**Факультет безопасности информационных технологий**

**Дисциплина:**

«Операционные системы»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6**

**«malloc/free»**

**Выполнила:**

Студент группа N32511

Бехит Мохаммед М.

Студент группа N32532

Эльгандур Йахиа А

**Проверил:**

Ханов А.Р.

Санкт-Петербург

2023 г.

**Задание :**

Протестировать функцию malloc/free и построить график зависимости времени выделения от размера запрашиваемой памяти.

**Теоретическая справка:**

**malloc()/free()** - Функция malloc() возвращает указатель void на выделенное пространство или значение NULL, если памяти недостаточно. - Функция malloc выделяет блок памяти размером не менее size байтов. Размер блока может превышать size байтов из-за дополнительных затрат места на хранение информации о выравнивании и обслуживании.

**new/delete** - New - пытается выделить и инициализировать объект или массив объектов указанного типа или заполнителя и возвращает подходящий типизированный ненулевой указатель на объект (или на исходный объект массива).

**sbrk()** - Функция sbrk() используется для изменения объема памяти, выделенной для вызывающего процесса. Изменение вносится путем сброса значения прерывания процесса и выделения соответствующего объема памяти. Объем выделенного пространства увеличивается по мере увеличения значения разрыва.

**Код программа:**

#include <iostream>

#include <vector>

#include <cstdlib>

#include <ctime>

#include <chrono>

using namespace std;

int main() {

int size = 1;

while (size <= 9000000) {

auto start = chrono::high\_resolution\_clock::now();

double\* ptr = static\_cast<double\*>(malloc(size)); // Cast the pointer to the correct type.

auto end = chrono::high\_resolution\_clock::now();

double duration = chrono::duration\_cast<chrono::milliseconds>(end - start).count(); // Specify the duration unit and add .count() to get the duration value.

if (ptr == nullptr) {

cout << "Error " << size << endl;

} else {

cout << size << " " << duration << " ms" << endl; // Print the duration unit.

free(ptr);

}

size += 100000;

}

return 0;

}

