

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**

**«Функции malloc/free»**

**Выполнила:**

Студентка группы N32511

Синюта А.А. 

**Проверил:**

Ханов А.Р. \_\_\_\_\_

Санкт-Петербург

2023г.

## Задание:

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

Либо винда, либо линукс

Сложный (или)

1. Сравнить с другими малоками
2. Тестировать на живом процессе

## Ход работы

Код на C, выполняющий выполняет цикл с выделением памяти разных размеров и сохранением результатов времени выделения в файл "allocation\_times.txt".

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define FILENAME "allocation_times.txt"

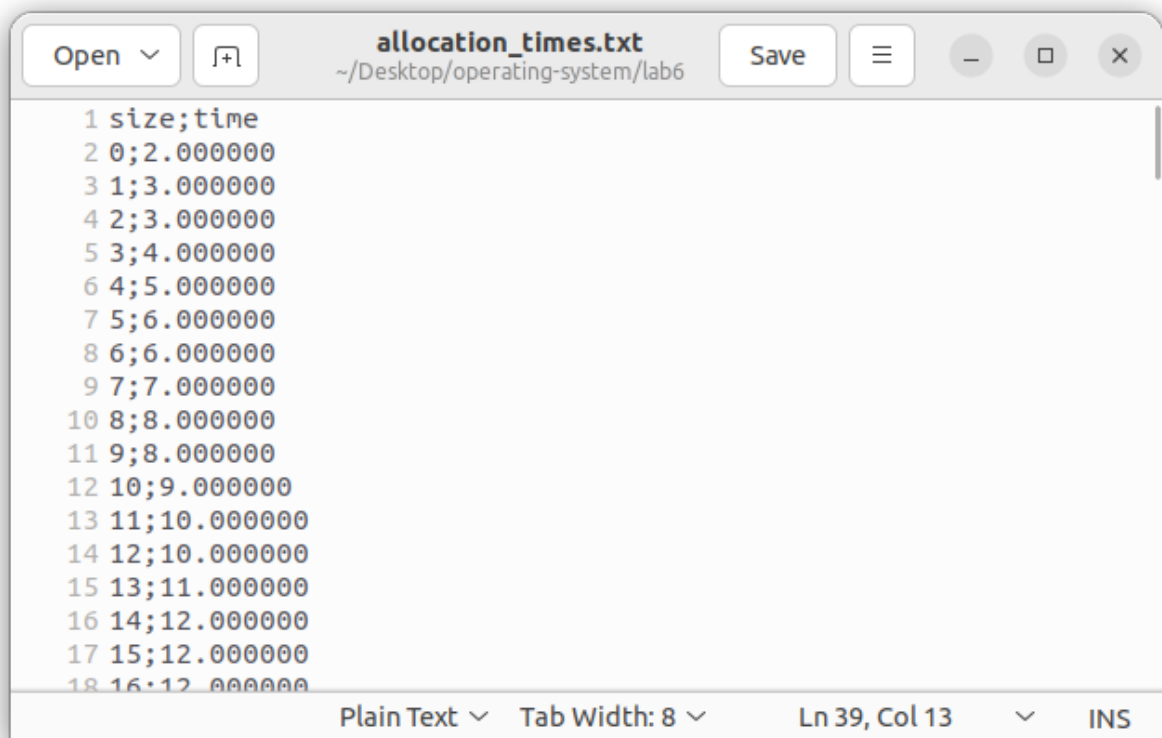
int main() {
    double time_spent = 0.0;
    FILE* file = fopen(FILENAME, "w");

    if (file == NULL) {
        printf("Failed to open file for writing.\n");
        return 1;
    }

    fprintf(file, "size;time\n");

    for (int i = 0; i < 10000; i++) {
        clock_t begin = clock();
        void* memory = malloc(i);
        free(memory);
        clock_t end = clock();
        time_spent += (double)(end - begin) / CLOCKS_PER_SEC;
        fprintf(file, "%d;%f\n", i, time_spent);
    }
    fclose(file);
    printf("Results saved to file %s\n", FILENAME);
    return 0;
}
```

В результате получаем такой файл с 10000 записями:



На основе этих данных построим график с помощью matplotlib Python3.

```
import matplotlib.pyplot as plt
```

```
filename = 'allocation_times.txt'
```

```
sizes = []
```

```
times = []
```

```
with open(filename, 'r') as file:
```

```
    next(file) # Пропускаем заголовок файла
```

```
    for line in file:
```

```
        size, time = line.strip().split(';')
```

```
        sizes.append(int(size))
```

```
        times.append(float(time))
```

```
plt.plot(sizes, times)
```

```
plt.xlabel('Размер памяти (байты)')
```

```
plt.ylabel('Время выделения (секунды)')
```

```
plt.title('Зависимость времени выделения от размера памяти')
```

```
plt.grid(True)
```

```
plt.savefig('chart.png')
```

```
plt.show()
```

Получаем такой график:

