

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Факультет информационных технологий

Кафедра параллельных вычислений

ОТЧЕТ

О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

«Измерение степени ассоциативности кэш-памяти»

студентки 1 курса, 24202 группы

Корсун Дарья Андреевна

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:
Кандидат технических наук
Владислав Александрович
Перепёлкин

Новосибирск 2025

СОДЕРЖАНИЕ

ЦЕЛЬ	4
ЗАДАНИЕ	4
ОПИСАНИЕ РАБОТЫ.....	5
ЗАКЛЮЧЕНИЕ	8

ЦЕЛЬ

Экспериментальное определение степени ассоциативности кэш-памяти.

ЗАДАНИЕ

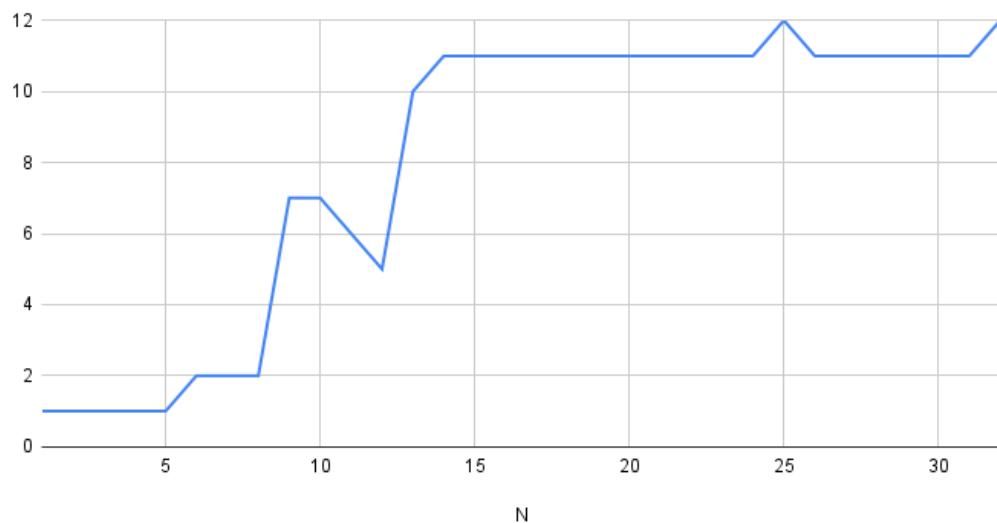
Написать программу, выполняющую обход памяти в соответствии с заданием.

2. Измерить среднее время доступа к одному элементу массива (в тактах процессора) для разного числа фрагментов: от 1 до 32. Построить график зависимости времени от числа фрагментов.
3. По полученному графику определить степень ассоциативности кэш-памяти, сравнить с реальными характеристиками исследуемого процессора.
4. Составить отчет по лабораторной работе

ОПИСАНИЕ РАБОТЫ

Для проведения эксперимента выделили массив достаточно большого размера, чтобы вместить все необходимые фрагменты с требуемым смещением. Элементы массива образуют связанный список, где значение каждого элемента является индексом следующего. Обход фрагментов данных организуем таким образом, чтобы подряд происходили обращения к элементам разных фрагментов, отстоящим на заданное смещение.

Графики зависимости среднего времени доступа к элементу массива от числа фрагментов:



На основе графика:

1. Степень ассоциативности L1 и L2 уровня кэш-памяти: 8

```

#include <chrono>
#include <climits>
#include <cstdint>
#include <fstream>
#include <iostream>

using namespace std;
using namespace std::chrono;

long long getCpuTicks() {
    auto now = high_resolution_clock::now();
    auto duration = now.time_since_epoch();
    return duration_cast<nanoseconds>(duration).count();
}

int* getArray(int n, int size, int offset) {
    int arraySize = offset * n;
    int* arr = new int[arraySize];

    for (int i = 0; i < arraySize; ++i) {
        arr[i] = 0;
    }

    for (int i = 0; i < size / n; ++i) {
        for (int j = 0; j < n - 1; ++j) {
            int current_pos = i + j * offset;
            int next_pos = i + (j + 1) * offset;
            arr[current_pos] = next_pos;
        }

        int last_pos = i + (n - 1) * offset;
        arr[last_pos] = (i + 1) % (size / n);
    }
}

return arr;
}

long long measure(int* array, int size) {
    volatile int k = 0;
    uint64_t start = getCpuTicks();

    for (int i = 0; i < size; i++) {
        k = array[k];
    }

    uint64_t end = getCpuTicks();
    return (end - start) / size;
}

```

```
int main() {
    const int size = (32 * 1024) / sizeof(int);
    const int offset = (16 * 1024 * 1024) / sizeof(int);
    const int N = 32;
    const int iterations = 1000;

    for (int n = 1; n < N + 1; ++n) {
        int* arr = getArray(n, size, offset);
        uint64_t minTicks = UINT64_MAX;

        for (int i = 0; i < iterations; ++i) {
            uint64_t ticks = measure(arr, size);
            if (ticks < minTicks) {
                minTicks = ticks;
            }
        }

        std::cout << minTicks << std::endl;
        delete[] arr;
    }

    return 0;
}
```

ЗАКЛЮЧЕНИЕ

Было проведено экспериментальное определение степени ассоциативности кэш-памяти. Были получены следующие результаты:
1. Степень ассоциативности L1 и L2 уровня кэш-памяти: 8