

**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»

студента Бородина Артёма Максимовича 2 курса, 19205 группы
Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:
к.т.н, доцент
А.Ю. Власенко

Новосибирск 2020

СОДЕРЖАНИЕ

[ЦЕЛЬ](#)

[ЗАДАНИЕ](#)

[ОПИСАНИЕ РАБОТЫ](#)

[ЗАКЛЮЧЕНИЕ](#)

[Приложение 1.](#) *Код программы*

[Приложение 2.](#) *Замеры времени*

ЦЕЛЬ

1. Экспериментальное определение степени ассоциативности кэш-памяти.

ЗАДАНИЕ

1. 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

1. Была изучена структура процессора, степень ассоциативности и понятие «буксования» кэш-памяти.
2. Была написана программа ([Приложение 1](#)), совершающая обход нескольких фрагментов данных в памяти, отстоящих друг от друга на смещение, кратное размеру банка. Также для каждого размера массива было замерено среднее время доступа к элементу в тактах процессора.
3. Программа была скомпилирована с ключом -O1.
4. По полученным данным был сделан график ([Приложение 2](#)).
5. По графику можно увидеть, что время доступа к элементу возросло после увеличения количества фрагментов до 5, что соответствует степени ассоциативности кэша 1го уровня.

ЗАКЛЮЧЕНИЕ

Были получены знания о работе кэш-памяти, изучена взаимосвязь скорости доступа к элементу массива в зависимости от степени ассоциативности кэш-памяти.

Приложение 1. Код программы

```
#include <iostream>
#include <algorithm>
#include <random>
#include <fstream>
#include <intrin.h>

#define TIME_MEASURE_COUNT 10
#define MEASURE_AMOUNT 10

double MeasureTime(const int *array, int size) {
    unsigned long long minDuration = ULONG_LONG_MAX;
    int k;
    for (int i = 0; i < TIME_MEASURE_COUNT; ++i) {
        unsigned long long start = __rdtsc();
        k = 0;
        for (int j = 0; j < size; ++j) {
            k = array[k];
        }
        unsigned long long end = __rdtsc();
        unsigned long long tmpDuration = end - start;
        if (tmpDuration < minDuration) {
            minDuration = tmpDuration;
        }
    }
    if (k) {
        std::cerr << " ";
    }
    return minDuration / double(size);
}

int main() {
    int i, j;
    int cacheSize = 256 * 16;
    int offset = 256 * 1024 * 8;

    std::ofstream output("../file.csv");
    if (!output) {
        std::cerr << "Can not open output file";
        return 0;
    }
    output << "ticks" << std::endl;

    for (int n = 1; n <= 32; ++n) {
        int array[offset * n];
        std::cout << "memory: " << offset * n / 256 << "KB fragmentNum:" << n << std::endl;
        for (i = 0; i < n - 1; ++i) {
            for (j = 0; j < cacheSize / n; ++j) {
                array[i * offset + j] = (i + 1) * offset + j;
            }
        }
        for (j = 0; j < cacheSize / n - 1; ++j) {
            array[(n - 1) * offset + j] = j + 1;
        }
        array[(n - 1) * offset + j] = 0;

        double ticks = 0;
        for (i = 0; i < MEASURE_AMOUNT; ++i) {
            ticks += MeasureTime(array, cacheSize);
        }

        std::cout << ticks / MEASURE_AMOUNT << std::endl;
        output << n << " " << ticks / MEASURE_AMOUNT << std::endl;
    }
}
```

Приложение 2. Замеры среднего времени доступа к элементу

