###### 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 2 курса, 23208 группы

**Веретенникова Андрея Александровича**

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

Д.С. Иванишкин

Новосибирск 2024

**СОДЕРЖАНИЕ**

[ЦЕЛЬ 3](#_Toc18443921)

[ЗАДАНИЕ 3](#_Toc18443922)

[ОПИСАНИЕ РАБОТЫ 4](#_Toc18443923)

[ЗАКЛЮЧЕНИЕ 5](#_Toc18443924)

[Приложение 1 (2, …). *Наименование приложения* 6](#_Toc18443925)

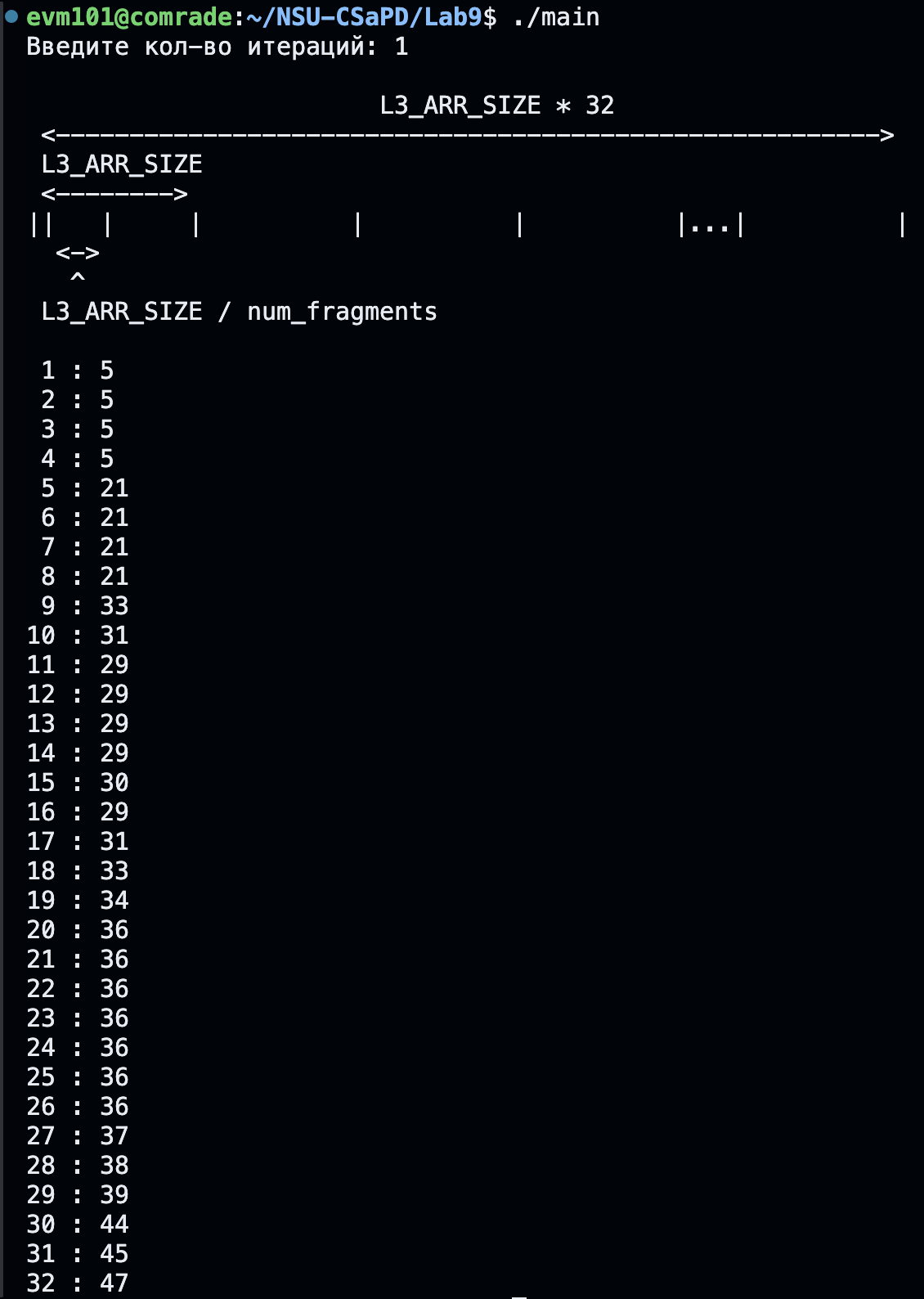
# ЦЕЛЬ

*Экспериментальное определение степени ассоциативности кэш-памяти.*

# ОПИСАНИЕ РАБОТЫ

*Описание работы:*

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.*

*Результаты:*

# *График зависимости:*

# ЗАКЛЮЧЕНИЕ

*В ходе лабораторной работы, я экспериментальное определил степень ассоциативности кэш-памяти.*

# Приложение 1. *Листинг*

#include <iostream>

#include <x86intrin.h>

#include <limits.h>

#include <iomanip>

const size\_t L3\_SIZE = 24 \* 1024 \* 1024;

const size\_t L3\_ARR\_SIZE = L3\_SIZE / sizeof(int);

using namespace std;

void initialize\_array(int\* array, int num\_fragments) {

int fragment\_size = L3\_ARR\_SIZE / num\_fragments;

int index = 0;

for (int j = 0; j < (num\_fragments - 1); j++) {

index = j \* L3\_ARR\_SIZE;

for (int i = 0; i < fragment\_size; i++) {

array[index + i] = index + i + L3\_ARR\_SIZE;

}

}

index = L3\_ARR\_SIZE \* (num\_fragments - 1);

for (int i = 0; i < fragment\_size; i++) {

array[index + i] = i + 1;

}

array[index + fragment\_size - 1] = 0;

}

size\_t measure\_access\_time(int\* array, int array\_size, int num\_fragments, int iterations) {

size\_t min\_duration = INT\_MAX;

int fragment\_size = L3\_ARR\_SIZE;

int k = 0;

for (int j = 0; j < iterations; j++) {

for (int iter = 0; iter < iterations; iter++) {

for (int i = 0; i < array\_size; i++) {

volatile int temp = array[i];

}

}

size\_t start = \_\_rdtsc();

for (int iter = 0; iter < iterations; iter++) {

do {

k = array[k];

} while (k != 0);

}

size\_t end = \_\_rdtsc();

size\_t duration = (end - start) / (fragment\_size \* iterations);

if (duration < min\_duration) {

min\_duration = duration;

}

}

return min\_duration;

}

void print\_logic() {

cout << endl;

cout << " L3\_ARR\_SIZE \* 32 " << endl;

cout << " <--------------------------------------------------------> " << endl;

cout << " L3\_ARR\_SIZE" << endl;

cout << " <-------->" << endl;

cout << "|| | | | | |...| |" << endl;

cout << " <->" << endl;

cout << " ^" << endl;

cout << " L3\_ARR\_SIZE / num\_fragments" << endl;

cout << endl;

}

int main() {

int iterations;

cout << "Введите кол-во итераций: ";

cin >> iterations;

print\_logic();

size\_t array\_size = L3\_ARR\_SIZE \* 32;

int\* array = new int[array\_size];

for (int num\_fragments = 1; num\_fragments <= 32; num\_fragments++) {

cout << setw(2) << num\_fragments << " : ";

initialize\_array(array, num\_fragments);

cout << measure\_access\_time(array, array\_size, num\_fragments, iterations) << endl;

}

delete[] array;

return 0;

}