###### 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, 22202 группы

**Бальчинова А.С.**

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

В.А. Перепёлкин

Новосибирск 2023

**СОДЕРЖАНИЕ**

[ЦЕЛЬ 3](#_Toc18443921)

[ЗАДАНИЕ 3](#_Toc18443922)

[ОПИСАНИЕ РАБОТЫ](#_Toc18443923) 3-4

[ЗАКЛЮЧЕНИЕ 5](#_Toc18443924)

# ЦЕЛИ

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.

# ОПИСАНИЕ РАБОТЫ

1. Реализация программы на языке программирования C++.

#include <iostream>  
#include <fstream>  
#include <intrin.h>  
using namespace std;  
  
//Заполнение массива.  
void fillArray(int\* array, int numbOfBanks, int fragmentSize, int offset) {  
 for (int i = 0; i < numbOfBanks-1; i++) {  
 for (int j = 0; j < fragmentSize; j++) {  
 array[offset \* i + j] = offset \* (i+1) + j;  
 }  
 }  
 for (int j = 0; j < fragmentSize-1; j++) {  
 array[offset \* (numbOfBanks-1) + j] = j+1;  
 }  
 array[offset \* (numbOfBanks-1) + fragmentSize-1] = 0;  
}  
//Обход массива.  
void arrayTraversal(const int \*array, const int len, const int amount) {  
 for (int k = 0, i = 0; i < len \* amount; i++) {  
 k = array[k];  
 }  
}  
  
//Size – объем кэшпамяти, N – число фрагментов, Offset – смещение между началами соседних фрагментов.  
int main() {  
 int arraySize = 256 \* 1024 \* 16 \* 32;  
 int size = 256 \* 64; //64 кб: L1   
 int offset = 256 \* 1024 \* 8;  
 int\* array = new int[arraySize];  
 ofstream file("data.txt");  
 for (int n = 1; n <= 32; n++) {  
 fillArray(array, n, size/n, offset);  
 unsigned long long t = LONG\_LONG\_MAX;  
 for (int i = 0; i < 30; i++) {  
 arrayTraversal(array, size, 1);  
 unsigned long long start = \_\_rdtsc();  
 arrayTraversal(array, size, 10);  
 unsigned long long end = \_\_rdtsc();  
 t = min(end - start, t);  
 }  
 file << n << ", " << t / 10 / size << endl;  
 }  
 delete[] array;  
}

1. Запуск программы и результат.

На графике видно замедление после 8-ми фрагментов, что соответствует степени ассоциативности кэш-памяти 1-го и 2-го уровня.

# ЗАКЛЮЧЕНИЕ

Экспериментально определили степени ассоциативности кэш-памяти.