###### 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, группы 22208

Новикова Григория Андреевича

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

Антон Юрьевич Кудинов

Новосибирск 2023

1. **Цель**

Экспериментально определить степени ассоциативности кэш-памяти разных уровней.

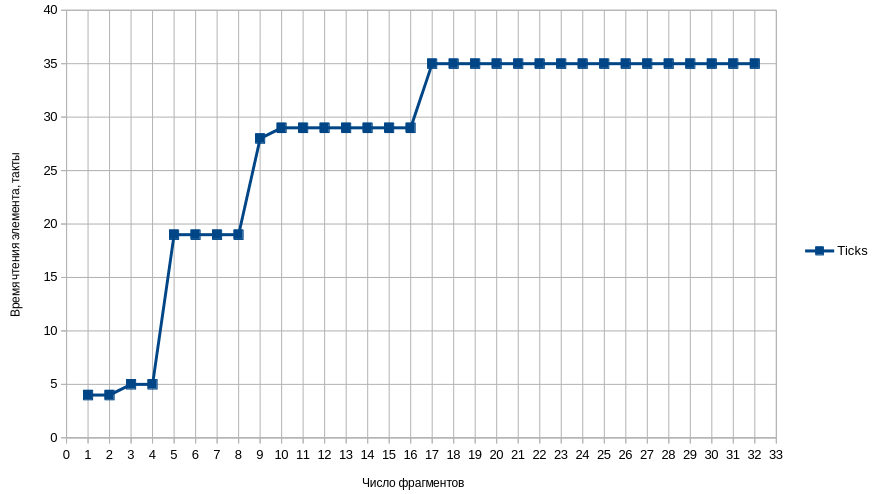
1. **Задание**

На основе обхода циклического массива, что вызывает кэш-буксование, сделать среднюю оценку количества тактов для обращения к одному элементу массива. На основе полученных данных выяснить степени ассоциативности кэш-памяти процессора.

1. **График роста тактов обращения**

В лабораторной работе создавался массив с фрагментами со смещением в 12МБ (размер L3 кэша).

Следующий график показывает зависимость времени обращения к одному элементу в тактах от количества фрагментов в массиве:



1. **Листинг программы**

#include <iostream>

#include <limits.h>

#include <x86intrin.h>

#include <fstream>

using namespace std;

using ull = unsigned long long;

using uint = unsigned int;

uint\* CreateArray(const uint size, const uint offset, const uint numberOfFragments) {

uint\* arr = new uint[size];

const uint cacheSize = offset;

const uint blockSize = cacheSize / numberOfFragments;

for (int i = 0; i < blockSize; ++i) {

int k = i;

for (int j = 1; j < numberOfFragments; ++j) {

arr[k] = k + offset;

k += offset;

if (k > cacheSize \* 32) {

throw std::exception();

}

}

arr[k] = (i + 1) % blockSize;

}

return arr;

}

double GetTraversTime(uint\* arr, uint& countElements) {

ull mintime = ULLONG\_MAX;

for (uint k = 0; k < 1000; ++k) {

uint next = 0;

ull start = \_\_rdtsc();

do {

++countElements;

next = arr[next];

} while (next != 0);

next = arr[next];

++countElements;

ull end = \_\_rdtsc();

if (next == -1) {

cout << "-1";

}

ull currTime = end - start;

mintime = mintime > currTime ? currTime : mintime;

}

countElements /= 1000;

return mintime;

}

int main() {

const uint offset = 12 \* 1024 \* 1024 / sizeof(uint); // 12 MB

const uint Size = offset \* 32;

cout << "Size (KB) " << Size / 1024 \* sizeof(uint) << endl;

ofstream out("графики.csv");

for (uint i = 1; i <= 32; ++i) {

uint\* arr = CreateArray(Size, offset, i);

cout << "==============================" << endl;

cout << "Number of fragments " << i << endl;

uint countElements = 0;

ull time = GetTraversTime(arr, countElements);

time /= countElements;

cout << "Average time " << time << endl;

cout << "==============================" << endl;

if (out.is\_open()) {

out << i << ',' << time << '\n';

}

delete [] arr;

}

return 0;

}

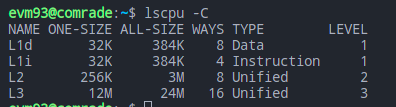
**Команда для компиляции:**

**g++ -O1 main.cpp**

1. **Оценка ассоциативности кэша**

На графике заметно три скачка. Первый скачок происходит после 4 фрагментов, что, скорее всего, говорит о степени ассоциативности TLB-кэша (буфера трансляции адресов). Следующий скачок роста времени обращения произошел на 8-ми фрагментах, что соответствует ассоциативности L1 и L2. Третий скачок произошёл на 16 фрагментах, что соответствует ассоциативности L3.

Информация об ассоциативности, полученная из конфигураций:



1. **Заключение**

В ходе данной работы было изучено строение множественно-ассоциативного кэша и эксперементально получены степени ассоциативности кэш памяти первого, второго и третьего уровней.