# ЭВМ и периферийные устройства

## Отчёт по лабораторной работе №6

## «Измерение степени ассоциативности КЭШ-памяти»

## Выполнил: Скопинцев Н.А. ФИТ НГУ 2 курс

## Преподаватель:

## Цель работы:

1. Экспериментальное определение степени ассоциативности кэш-памяти.

## Листинг программы

#include <iostream>

#include <limits>

#include <climits>

#include <random>

#include <vector>

#include <algorithm>

#define K 10

**double** traversal(**volatile** size\_t\* array, size\_t size) {

**union** ticks {

**unsigned** **long** **long** t64;

**struct** s32 {

**long** th, tl;

} t32;

} start, end;

//double cpu\_Hz = 2400000000ULL; // for 2.4 GHz CPU

**volatile** size\_t k = 0;

**volatile** size\_t i = 0;

**double** res = std::numeric\_limits<**double**>::max();

**for** (size\_t j = 0; j < K; j++) {

asm ("rdtsc\n":"=a"(start.t32.th),"=d"(start.t32.tl));

**for** (i = 0, k = 0; i < size; i++) {

k = array[k];

}

asm ("rdtsc\n":"=a"(end.t32.th),"=d"(end.t32.tl));

res = std::min(res, (**double**)(end.t64-start.t64));

}

**return** res / size;

}

**void** count(size\_t n) {

**volatile** size\_t \*S;

size\_t BlockSize = 32 \* 1024 + 256 \* 1024 + 3 \* 1024 \* 1024 / **sizeof**(size\_t);

size\_t Offset = 16 \* 1024 \* 1024 / **sizeof**(size\_t);

S = **new** size\_t[Offset \* n]; // Прямой обход

//// �нициализация массивов///

**for** (size\_t i = 0; i < BlockSize / n; i++) {

**for** (size\_t j = 0; j < n - 1; j++) {

S[j\*(Offset) + i] = (j+1)\*(Offset) + i;

}

S[(n - 1) \* Offset + i] = (i + 1) % (BlockSize / n);

}

///////////////////////////////////////

std::cout << traversal(S, Offset \* n) << std::endl;

**delete** [] S;

}

**int** **main**() {

std::cout << std::fixed;

std::cout.precision(0);

std::cout << "Число фрагментов\tЧисло тактов" << std::endl;

**for** (size\_t i = 1; i <= 32; i++) {

std::cout << i << "\t\t\t";

count(i);

}

**return** 0;

}

**Ключи компиляции:**

g++ main.cpp

./a.out

График

## Вывод

В процессе выполнения работы была исследована зависимость времени доступа к данным в памяти от их объёма и порядка их обхода.

Был доказан очевидный факт, что последовательный обход данных наиболее аппаратно предсказуем, и потому время доступа при таком обходе наименьшее.

Было подтверждено, что при малых размерах массива время доступа при использовании различных способов обхода приблизительно равно, поскольку весь массив помещается в кэш-память.

Нельзя не отметить факт роста среднего времени доступа к элементам при их случайном обходе.