

Новосибирский Государственный Университет

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ**

Курс “ЭВМ и периферийные устройства”

Лабораторная работа №8

«ВЛИЯНИЕ КЭШ-ПАМЯТИ НА ВРЕМЯ ОБРАБОТКИ МАССИВОВ»

Выполнил:

Пятаев Егор, гр. 15206

Преподаватель:

Городничев Максим Александрович

Новосибирск 2016

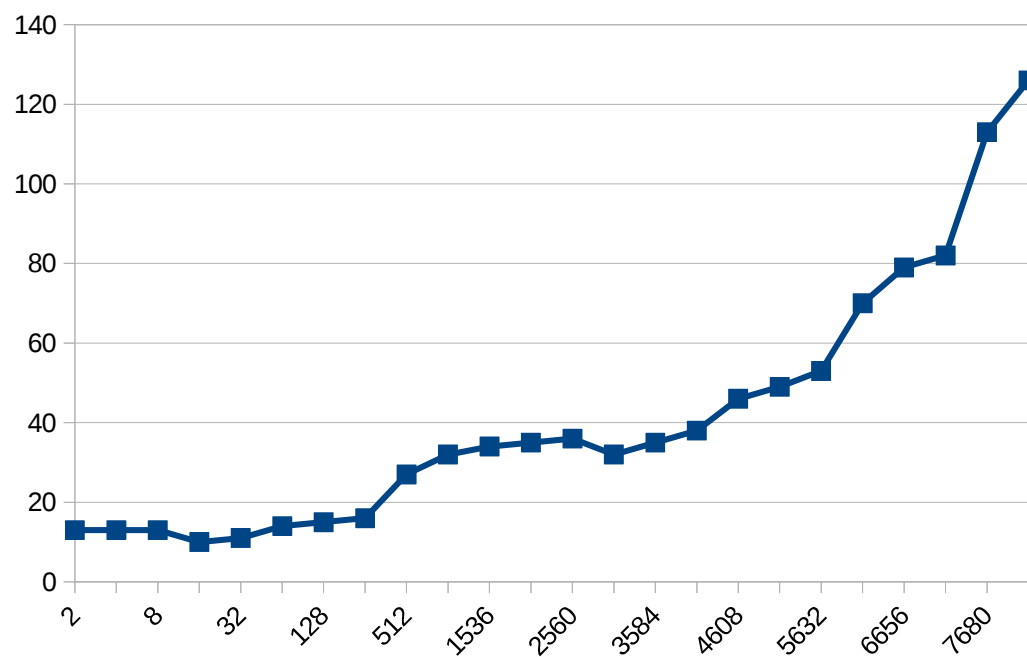
## Цели работы

1. Исследование зависимости времени доступа к данным в памяти от их объема.
2. Исследование зависимости времени доступа к данным в памяти от порядка их обхода.

## Результаты замеров времени

Обход элементов в случайном порядке:

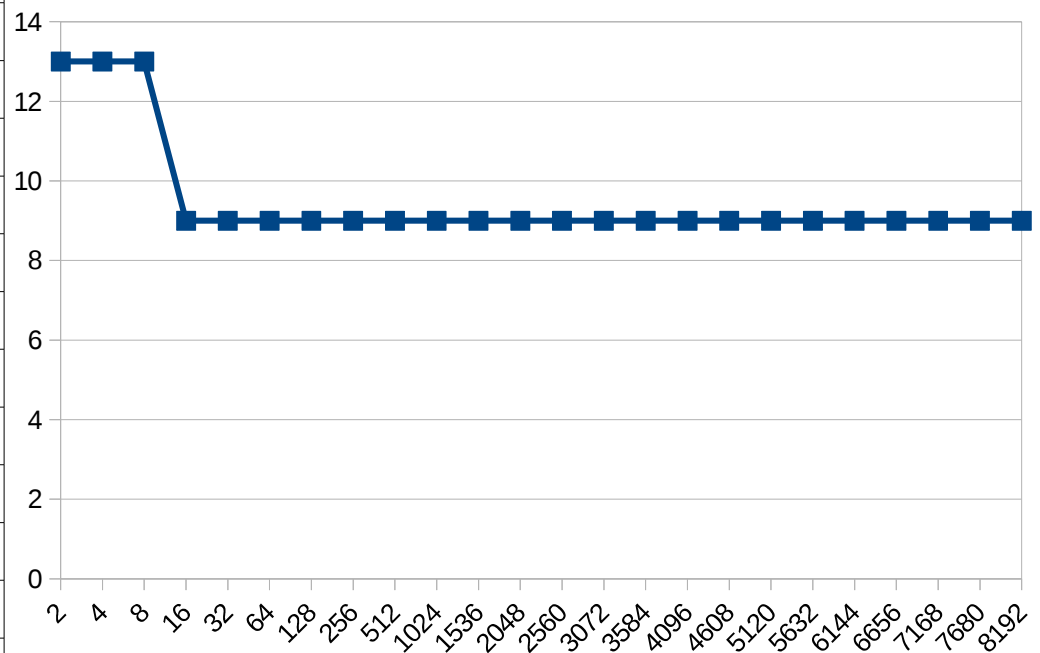
1	13
2	13
4	13
8	13
16	10
32	11
64	14
128	15
256	16
512	27
1024	32
1536	34
2048	35
2560	36
3072	32
3584	35
4096	38
4608	46
5120	49
5632	53
6144	70
6656	79
7168	82
7680	113
8192	126



## Результаты замеров времени

Обход элементов в прямом порядке:

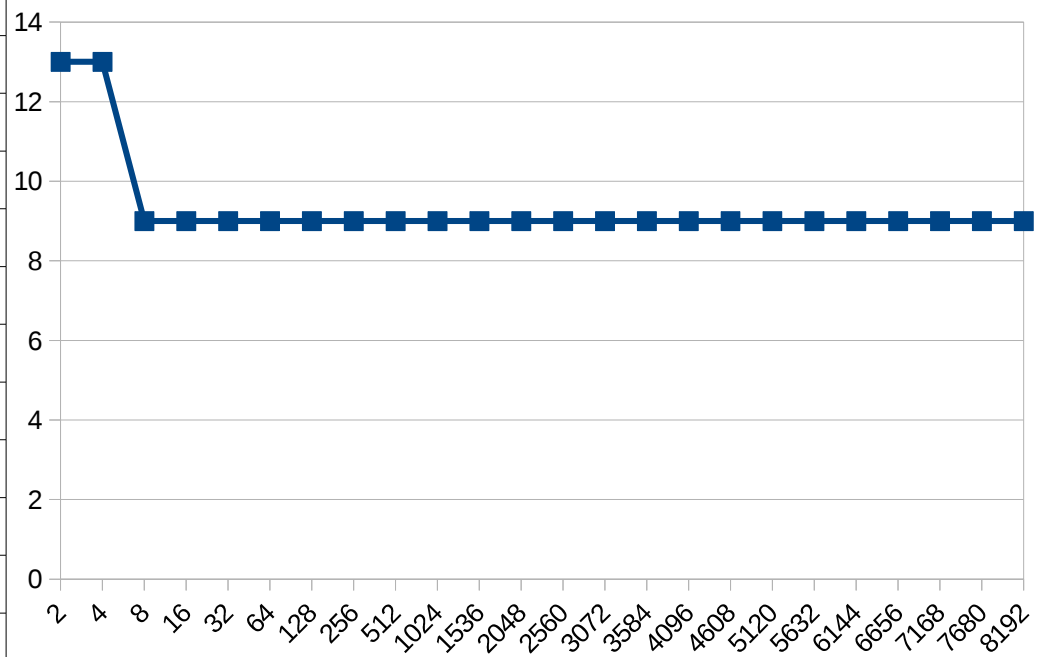
1	13
2	13
4	13
8	13
16	9
32	9
64	9
128	9
256	9
512	9
1024	9
1536	9
2048	9
2560	9
3072	9
3584	9
4096	9
4608	9
5120	9
5632	9
6144	9
6656	9
7168	9
7680	9
8192	9



## Результаты замеров времени

Обход элементов в обратном порядке:

1	13
2	13
4	13
8	9
16	9
32	9
64	9
128	9
256	9
512	9
1024	9
1536	9
2048	9
2560	9
3072	9
3584	9
4096	9
4608	9
5120	9
5632	9
6144	9
6656	9
7168	9
7680	9
8192	9



## Листинг реализованной программы

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <time.h>

void swap(size_t *a, size_t *b){
    int c;
    c = *a;
    *a = *b;
    *b = c;
}

int main(int argc, char* argv[]){
    int N;
    size_t *a = NULL;
    union ticks {
        unsigned long long t64;
        struct s32 {
            long th, tl;
        } t32;
    } start, end;
    srand(time(0));
    int K = 10;
    unsigned long long T, t;

    for(N = 128; N <= 8192*128; ){
        a = (size_t*)malloc(N*sizeof(size_t));
        if(!a){
            printf("NO MEMORY!\n");
            exit(1);
        }
        /* _____ */
        // straight
        for(int o = 0; o < N; o++){
            a[o] = o + 1;
        }
        a[N - 1] = 0;
        T = -1;
        for(int q = 0; q < 5; ++q) {
            asm("rdtsc\n":"=a"(start.t32.th), "=d"(start.t32.tl));
            for (int k=0, i=0; i<N*K; i++){
                k = a[k];
            }
            asm("rdtsc\n":"=a"(end.t32.th), "=d"(end.t32.tl));
            t = (end.t64 - start.t64) / (N * K);
            if(t < T) {
                T = t;
            }
        }
        // printf("%d K6: Ticks taken: %llu\n", N/128, T);
    }
```

```

/*_____*/
// back
for(int o = N - 1; o >= 0; o--){
    a[o] = o - 1;
}
a[0] = N - 1;
T = -1;
for(int q = 0; q < 5; ++q) {
    asm("rdtsc\n":"=a"(start.t32.th), "=d"(start.t32.tl));
    for (int k=0, i=0; i<N*K; i++){
        k = a[k];
    }
    asm("rdtsc\n":"=a"(end.t32.th), "=d"(end.t32.tl));
    t = (end.t64 - start.t64) / (N * K);
    if(t < T) {
        T = t;
    }
}
printf("%d K6: Ticks taken: %llu\n", N/128, T);

/*_____*/
// random
for(int o = 0; o < N; o++){
    a[o] = o + 1;
}
a[N - 1] = 0;
for(int o = 0; o < N; o++){
    int u = rand() % N;
    swap(&a[u], &a[o]);
}
T = -1;
for(int q = 0; q < 5; ++q) {
    asm("rdtsc\n":"=a"(start.t32.th), "=d"(start.t32.tl));
    for (int k=0, i=0; i<N*K; i++){
        k = a[k];
    }
    asm("rdtsc\n":"=a"(end.t32.th), "=d"(end.t32.tl));
    t = (end.t64 - start.t64) / (N * K);
    if(t < T) {
        T = t;
    }
}
// printf("%d K6: Ticks taken: %llu\n", N/128, T);
/*_____*/
free(a);
if(N < 512*128) N*=2;
else N+=512*128;
}
return 0;
}

```

## Выводы

Для достижения поставленных целей была написана программа для определения размера кэша процессора. Был произведен обход массива в прямой, обратный и случайный, замерено время во всех случаях. Из замеров видно, что при прямой и обратный обходы происходят за линейное время (аппаратная предвыборка), а при случайном видны “скачки”, которые происходят при размере массивов в 32, 256, 6144 Кб, что соответствует 3-ем уровням кэш памяти, это совпадает с данными полученные утилитой *lscpu* (linux).