

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Кафедра вычислительной математики и программирования

спецкурс «Параллельные и распределенные вычисления»

ОТЧЕТ

Лабораторная работа № 1
«Знакомство с технологией CUDA»

Выполнил: Дмитроченко Б.А.
Группа: М8О-114М-22, **вариант 6**
Преподаватель: Семенов С. А.

Москва, 2022

Оглавление

1. Постановка задачи.....	3
2. Описание решения	3
3. Основные моменты кода	3
4. Результат работы программы.....	4

1. Постановка задачи

Вычислить значения функции синуса

2. Описание решения

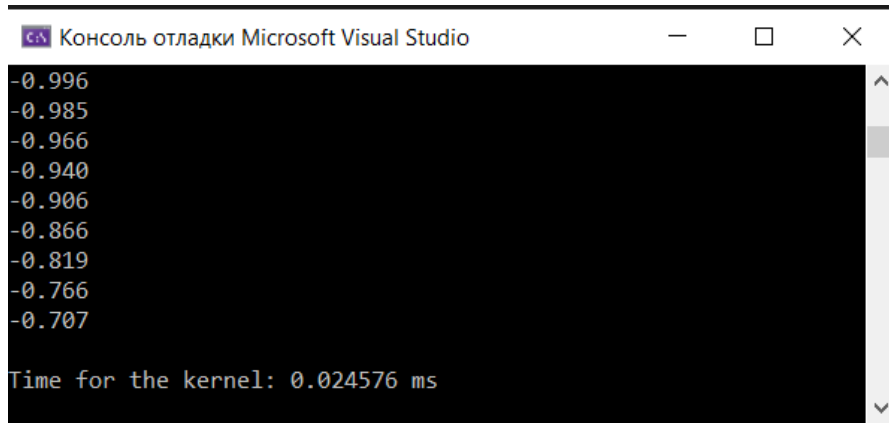
Создать массив из N значений (N задается в теле программы) и для каждого вычислить значение по формуле: $\sin(\frac{\pi N}{36})$

3. Основные моменты кода

Ниже представлена программная реализация задачи

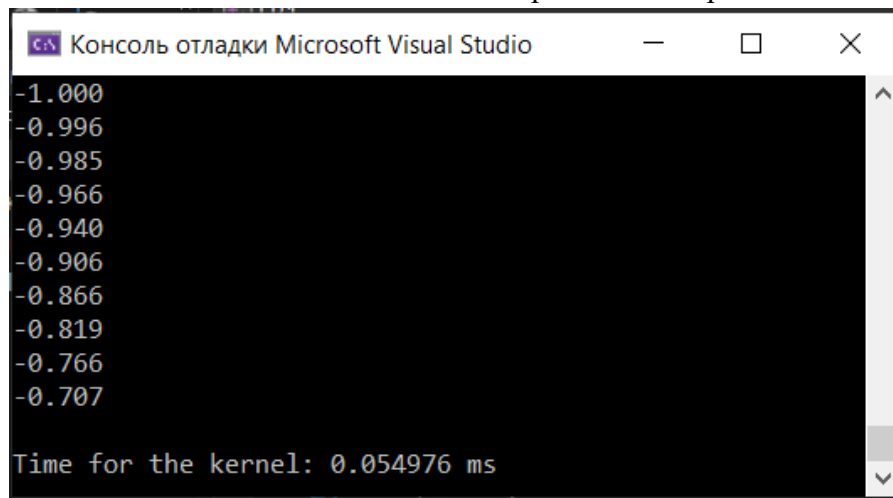
```
1  #include "cuda_runtime.h"
2  #include "device_launch_parameters.h"
3
4  #include <stdio.h>
5  #include <stdlib.h>
6  #include <math.h>
7  #include <iostream>
8
9  using namespace std;
10
11 #define PI 3.14159265
12
13 #define CSC(call) \
14 do { \
15     cudaError_t state = call; \
16     if (state != cudaSuccess) { \
17         fprintf(stderr, "ERROR: %s:%d. Message: %s\n", __FILE__, __LINE__, cudaGetErrorString(state)); \
18         exit(0); \
19     } \
20 } while (0); \
21
22 __global__ void kernel(double* v1, long long n) {
23     int i, idx = blockDim.x * blockIdx.x + threadIdx.x;
24     long long offset = blockDim.x * gridDim.x;
25     for (i = idx; i < n; i += offset) {
26         v1[i] = sin(PI*((float)i/36));
27     }
28 }
29
30 int main()
31 {
32     long long n = 1000000;
33     double* sin = (double*)malloc(n * sizeof(double));
34     double* sin_dev = (double*)malloc(n * sizeof(double));
35
36     cudaEvent_t start, stop;
37     float time;
38     cudaEventCreate(&start);
39     cudaEventCreate(&stop);
40
41     CSC(cudaMalloc(&sin_dev, sizeof(double) * n));
42     CSC(cudaMemcpy(sin_dev, sin, sizeof(double) * n, cudaMemcpyHostToDevice));
43
44     cudaEventRecord(start, 0);
45
46     kernel << <256, 256 >> > (sin_dev, n);
47
48     cudaEventRecord(stop, 0);
49     cudaEventSynchronize(stop);
50
51     CSC(cudaMemcpy(sin, sin_dev, sizeof(double) * n, cudaMemcpyDeviceToHost));
52     CSC(cudaFree(sin_dev));
53
54     for (long long i = 0; i < n; i++) {
55         printf("%.3f\n", sin[i]);
56     }
57
58     printf("\n");
59     free(sin);
60     cudaEventElapsedTime(&time, start, stop);
61     printf("Time for the kernel: %f ms\n", time);
62     return 0;
63 }
```

4. Результат работы программы



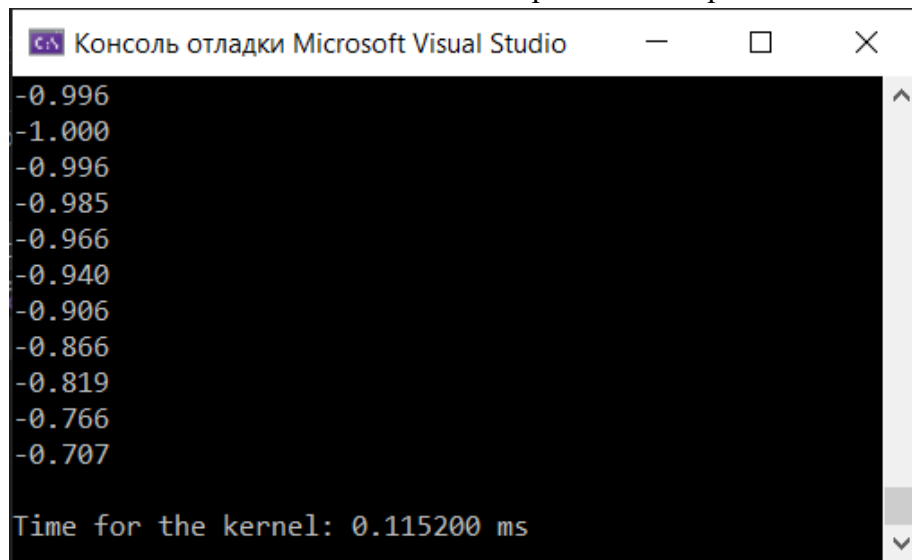
```
Консоль отладки Microsoft Visual Studio
-0.996
-0.985
-0.966
-0.940
-0.906
-0.866
-0.819
-0.766
-0.707
Time for the kernel: 0.024576 ms
```

Рис. 1. Окно вывода консольного приложения при $N = 1000$



```
Консоль отладки Microsoft Visual Studio
-1.000
-0.996
-0.985
-0.966
-0.940
-0.906
-0.866
-0.819
-0.766
-0.707
Time for the kernel: 0.054976 ms
```

Рис. 2. Окно вывода консольного приложения при $N = 10000$



```
Консоль отладки Microsoft Visual Studio
-0.996
-1.000
-0.996
-0.985
-0.966
-0.940
-0.906
-0.866
-0.819
-0.766
-0.707
Time for the kernel: 0.115200 ms
```

Рис. 3. Окно вывода консольного приложения при $N = 100000$

Ссылка на репозиторий:
<https://github.com/laggerlich/PDC/blob/master/kernel.cu>