

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Кафедра вычислительной математики и программирования

**спецкурс «Параллельные и распределенные вычисления»**

**ОТЧЕТ**

**Лабораторная работа № 1**  
**«Освоение программного обеспечения среды программирования**  
**NVIDIA»**

Выполнил: Патрикеева Л.В.  
Группа: М8О-114М-22, вариант 15  
Преподаватель: Семенов С. А.

Москва, 2022

## **Содержание**

1. Постановка задачи.....	2
2. Описание решения .....	2
3. Аппаратное обеспечение и ПО .....	2
4. Основные моменты кода .....	3
5. Результат работы программы .....	4

### **1. Постановка задачи**

Написать программу «Hello, World!» на CUDA.

### **2. Описание решения**

Запускаем ядро, где задаём размер сетки и размер блока, чтобы программа выполнялась параллельно. Возьмём сетку с 2 блоками для ядра, и каждый блок имеет 4 потока: Вывод будет производиться через CUDA. Так же будем выводить какая строка из какого потока. Т.к. все стандартные функции возвращают код исполнения, они обернуты в макрос, который проверяет исполнение на ошибки.

### **3. Аппаратное обеспечение и ПО**

Название: GeForce 840M

Размер глобальной памяти: 3150381056

Размер константной памяти: 65536

Размер разделяемой памяти: 49152

Регистров на блок: 32768

Максимум потоков на блок: 1024

Количество мультипроцессоров: 3

RAM: 16 GB

SSD: 500 GB

OS: Windows 10

IDE: Visual Studio 2017

## 4. Основные моменты кода

Ссылка на код в github:

<https://github.com/patrikeyeva/CUDA-Labs/tree/main/Lab1>

```
#include <fstream>
#include "stdlib.h"
#include <stdio.h>

#define CSC(call)

do {

    cudaError_t res = call;

    if (res != cudaSuccess) {
        fprintf(stderr, "ERROR in %s:%d. Message: %s\n",
            __FILE__, __LINE__, cudaGetErrorString(res));
        exit(0);
    }

} while (0)

__global__ void kernel(char* a, int n) {
    int idx = blockDim.x * blockIdx.x + threadIdx.x; //номер текущей нити
    printf("%s (current thread number %d)\n", a, idx);
}

int main() {
    const int n = 14;
    char chars[n] = "Hello, World!";

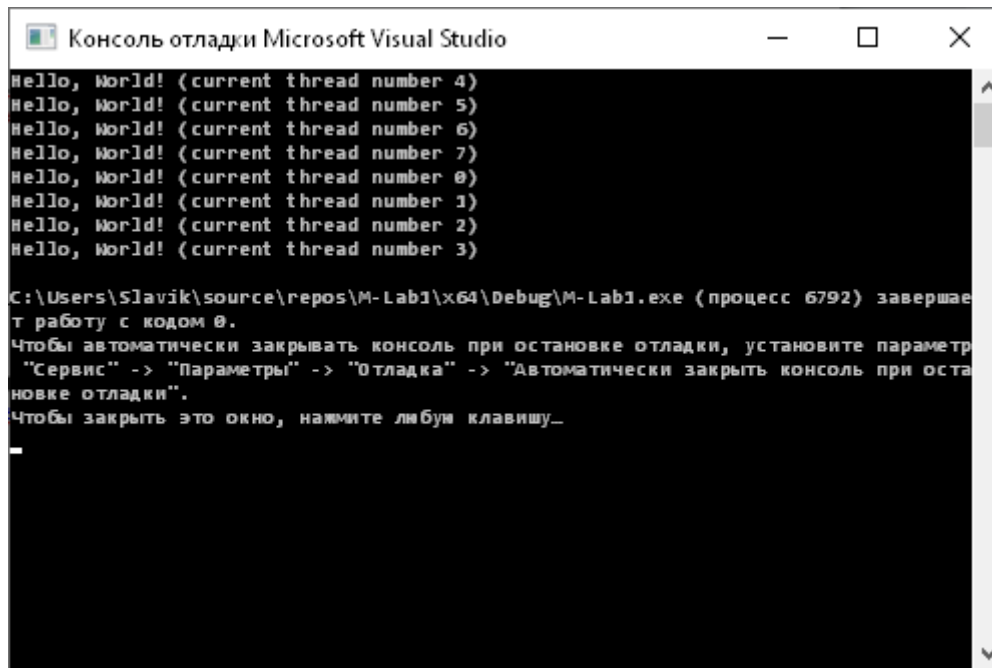
    char *arr_dev;
    CSC(cudaMalloc(&arr_dev, sizeof(char) * n));
    CSC(cudaMemcpy(arr_dev, chars, sizeof(char) * n, cudaMemcpyHostToDevice));
    CSC(cudaGetLastError());

    kernel << < 2, 4 >> > (arr_dev, n);
    cudaDeviceSynchronize();
    CSC(cudaGetLastError());

    CSC(cudaFree(arr_dev));

    return 0;
}
```

## 5. Результат работы программы



```
Консоль отладки Microsoft Visual Studio

Hello, World! (current thread number 4)
Hello, World! (current thread number 5)
Hello, World! (current thread number 6)
Hello, World! (current thread number 7)
Hello, World! (current thread number 0)
Hello, World! (current thread number 1)
Hello, World! (current thread number 2)
Hello, World! (current thread number 3)

C:\Users\Slavik\source\repos\M-Lab1\x64\Debug\M-Lab1.exe (процесс 6792) завершае
т работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, установите параметр
"Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть консоль при оста
новке отладки".
Чтобы закрыть это окно, нажмите любую клавишу_
_
```