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»**

Выполнила: Сучкова Наталья

Группа: М80-114М-22, вариант 20

Преподаватель: Семенов С. А.

Москва, 2022

Содержание

[1. Постановка задачи 2](#_Toc52570380)

[2. Описание решения 2](#_Toc52570381)

[3. Аппаратное обеспечение и ПО 2](#_Toc52570382)

[4. Основные моменты кода 2](#_Toc52570383)

[5. Результат работы программы 3](#_Toc52570384)

[6. Сравнение скорости выполнения на CPU и GPU 3](#_Toc52570385)

[7. Выводы 4](#_Toc52570386)

[8. Приложения 5](#_Toc52570387)

# 1. Постановка задачи

Задача заключается в том, чтобы задать 1024 числа и выдать чётное / нечётное.

# 2. Описание решения

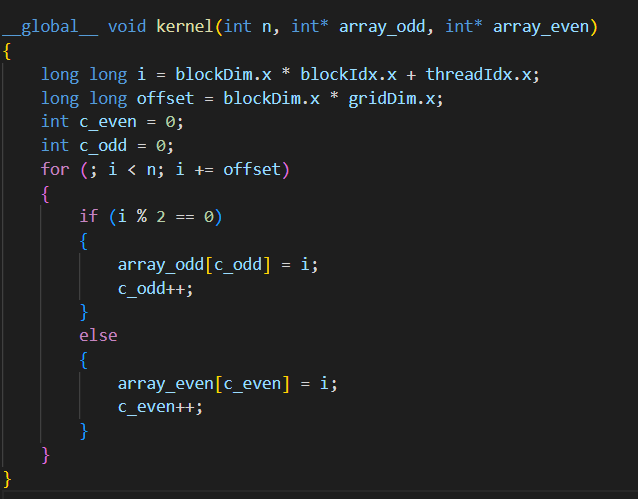
В данном решении, мы задаем множество из первых 1024х целых неотрицательных чисел без повторений, затем разделяем исходный массив на два других массива. В каждом из которых содержатся только четные или нечетные числа. Исходя из задачи, видно, что имеет смысл выполнить параллельное заполнение массивов с четными и нечетными числами. Данная часть выполняется с помощью параллельного программирования на CUDA, затем выполняется последовательный вывод результата разделения исходного массива.

# 3. Аппаратное обеспечение и ПО

Видеокарта NVIDIA GeForce GTX 1650, Windows 10

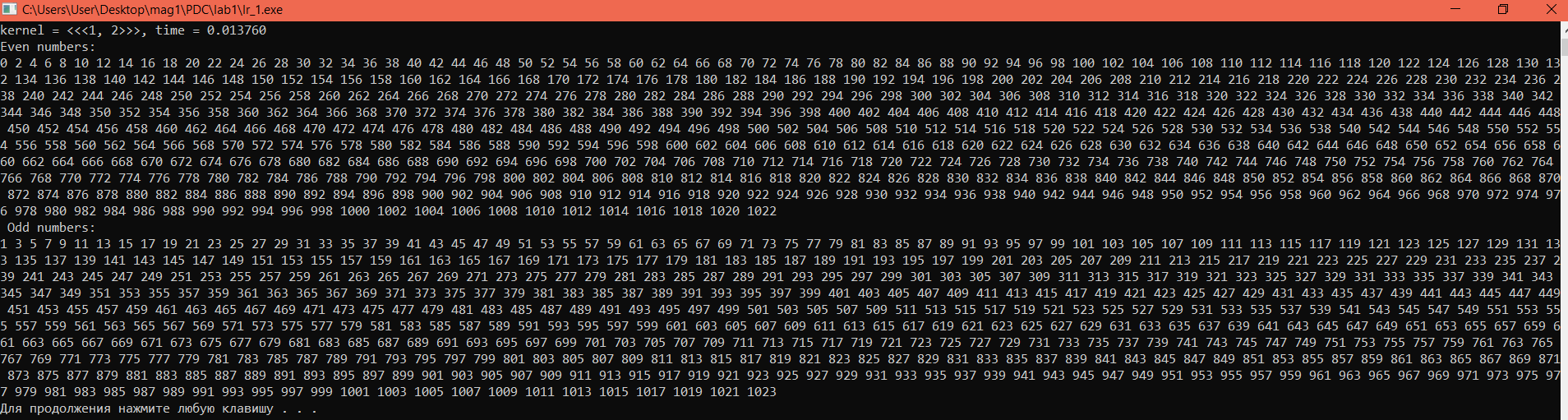
# 4. Основные моменты кода

На рисунке 1 приведен программный код части с использованием CUDA для параллельного программирования.



**Рис 1.** Программный код CUDA параллельного разделения массива.

# 5. Результат работы программы



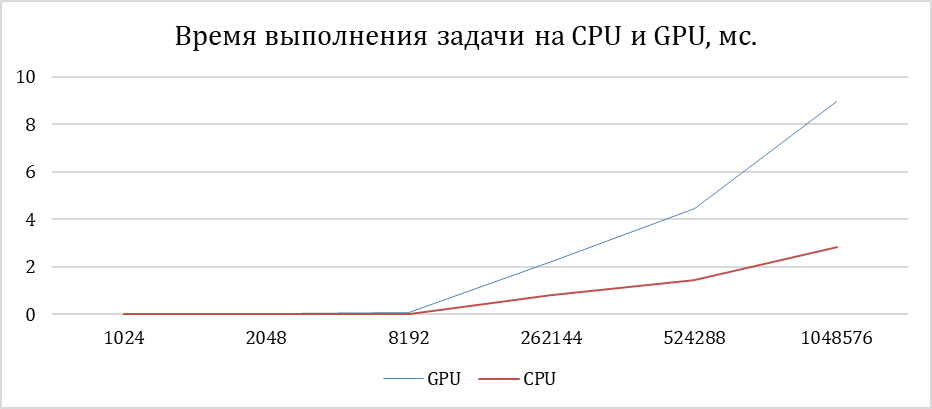
**Рис 2.** Вывод результата разделения.

# 6. Сравнение скорости выполнения на CPU и GPU

Для сравнения динамики скорости выполнения задачи на CPU и GPU посмотрим время выполнения разделения для большего количества чисел N в исходном массиве.

**Таблица 1.** Время выполнения программы при различных значениях *N*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N** | | **GPU** | **CPU** | **tCPU/tGPU** |
| **время выполнения, мс** | **время выполнения, мс** |
| N | 1024 | 0.01392 | 0.0022 | 0.15805 |
| N\*2 | 2048 | 0.022336 | 0.0041 | 0.18356 |
| N\*8 | 8192 | 0.07536 | 0.0214 | 0.28397 |
| N\*256 | 262144 | 2.24256 | 0.8259 | 0.36828 |
| N\*512 | 524288 | 4.475072 | 1.4436 | 0.32259 |
| N\*1024 | 1048576 | 8.955872 | 2.8247 | 0.31540 |



**Рис 3.** График зависимости времени выполнения программы от порядка матрицы *N*.

При запуске программы с различными значениями *N* видно, что вычисления на видеокарте производились медленнее, чем на основном процессоре компьютера.

# 7. Выводы

В лабораторной работе №1 проведено сравнение скорости выполнения алгоритма разделения одномерного массива конечной длины по условию на CPU и GPU. Для такой простой операции, как проверка условия и последующая вставка элемента в массив, по скорости выполнения CPU показывает себя как более быстрый. Это объясняется дополнительным расходом времени на разбиение задачи, передачу данных между процессорами. Для более сложных операций выполнение на графическим процессоре должно быть более эффективным.

# 8. Приложения

GitHub ссылка на программный код ЛР1:

<https://github.com/natalia-sd/PDC/tree/main/lab_1>