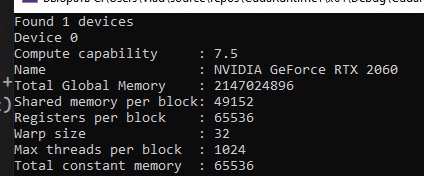
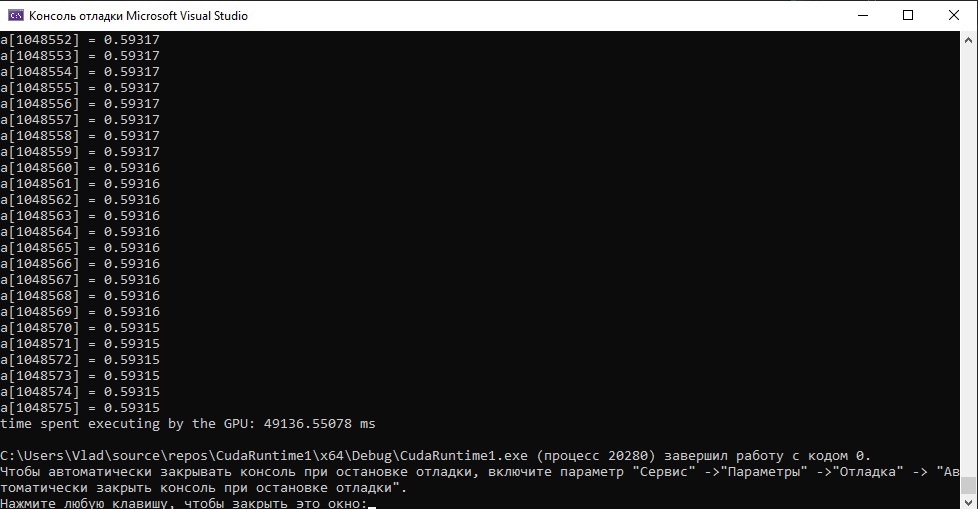
**Лабораторная работа №1**

**Основы работы с технологией CUDA. Гибридное программирование. Работа с глобальной памятью**

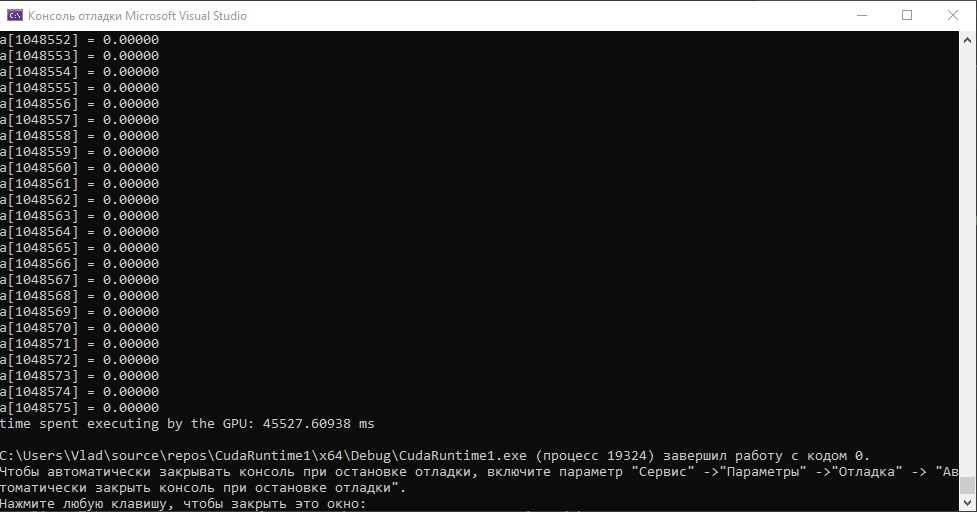
**Цель:** изучить модель программирования в CUDA, иерархию памяти в CUDA и основные особенности работы с глобальной памятью.

**Задание 1.** В MS Visual Studio создать проект CUDA VS Wizard. Ознакомиться и запустить программу «Hello world». Получить информацию об устройстве. Измерить время выполнения программы. **Результаты занести в отчёт.** Запустить программу«Hello world» на всех мультипроцессорах в GPU. Измерить время выполнения программы. **Результаты занести в отчёт.**

****

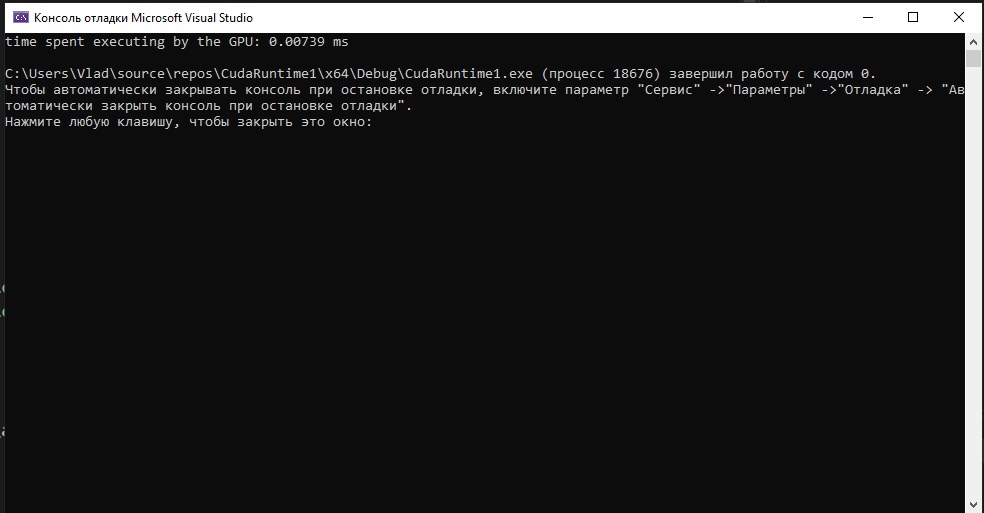


На всех мультипроцессорах в GPU



**Задание 2.** Написать программу на Cи с использованием CUDA runtime API в соответствии с вариантом задания. Измерить время работы программы для различных значений параметров. **Результаты занести в отчёт.** Написать программу для верификации результатов.

|  |  |
| --- | --- |
| **Вариант** | **Задание** |
| **1** | Даны два вектора А и В из N натуральных (ненулевых) элементов (задаются случайно). Вектора расположены в глобальной памяти.  Написать программу, выполняющую перемножение двух векторов на GPU. |



**Контрольные вопросы**

1. Что такое гибридное программирование?

Гибридное программирование - это написание программы для гетерогенной аппаратной вычислительной структуры, например, для системы, состоящей из центрального процессора CPU и графического ускорителя GPU.

1. Что такое CUDA?

CUDA (Compute Unified Device Architecture) –

- технология (библиотеки и расширенный Cи), предназначенная для разработки приложений для массивно-параллельных вычислительных устройств, заметно облегчает написание GPGPU (General Purposed Graphical Processing Unit)-приложений;

- программно-аппаратная архитектура.

1. Основные положения программной модели CUDA?

GPU (Graphical Processing Unit, device) – это вычислительное устройство, которое:

- состоит из массива потоковых мультипроцессоров (streaming multiprocessor, SM);

- является сопроцессором к центральному процессору CPU (host);

- имеет собственную память (DRAM);

- выполняет одновременно большое количество нитей.

1. Из чего состоит программный стек CUDA?

Программный код состоит из последовательных и параллельных частей, выполняющихся на CPU и GPU соответственно. Программа, использующая GPU, состоит из следующих частей:

- программного кода для GPU, описывающего необходимые вычисления и работу с памятью;

- программного кода для CPU, в котором осуществляется управление памятью GPU (выделение/освобождение), обмен данными между GPU/CPU, запуск кода для GPU, обработка результатов и прочий последовательный код.

1. Что такое ядро в CUDA?

Ядро (kernel) – это функция, которая работает на GPU и которая может быть вызвана только с CPU. Ядро выполняется на сетке из блоков.

1. Какие расширения языка Си вводятся в CUDA?

Вводимые в CUDA расширения языка Си состоят из:

- спецификаторов функций, показывающих, где будет выполняться функция и откуда она может быть вызвана;

- спецификаторов переменных, задающих тип памяти, используемый для данных переменных;

- директивы для запуска ядра из кода;

- встроенные переменные, содержащие информацию о текущей нити;

- дополнительные типы данных.

1. Какие встроенные переменные поддерживаются в CUDA и для чего они нужны?

В CUDA поддерживаются следующие встроенные переменные, содержащие информацию о текущей нити (рисунок 3):

- dim3 gridDim; // размер сетки

- uint3 blockIdx; // индекс текущего блока в сетке

- dim3 blockDim; // размер блока

- uint3 threadIdx; // индекс текущей нити в блоке

- int warpSize; // размер warp’а

1. Какие ограничения вводятся на функции, выполняемые на GPU?

Ограничения на функции, выполняемые на GPU:

* нельзя брать адрес функции (за исключением \_\_global\_\_);
* не поддерживается рекурсия;
* не поддерживаются static-переменные внутри функции;
* не поддерживается переменное число входных аргументов.