Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное

образовательное учреждение высшего образования

«Южно-Уральский государственный университет

(национальный исследовательский университет)»

Высшая школы электроники и компьютерных наук

Кафедра системного программирования

ОТЧЕТ  
о лабораторной работе №8  
по дисциплине «Технологии параллельного программирования»

Выполнил:   
студент группы КЭ-220   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Голенищев А. Б.   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.   
   
Отчет принял:   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Жулев А. Э.  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

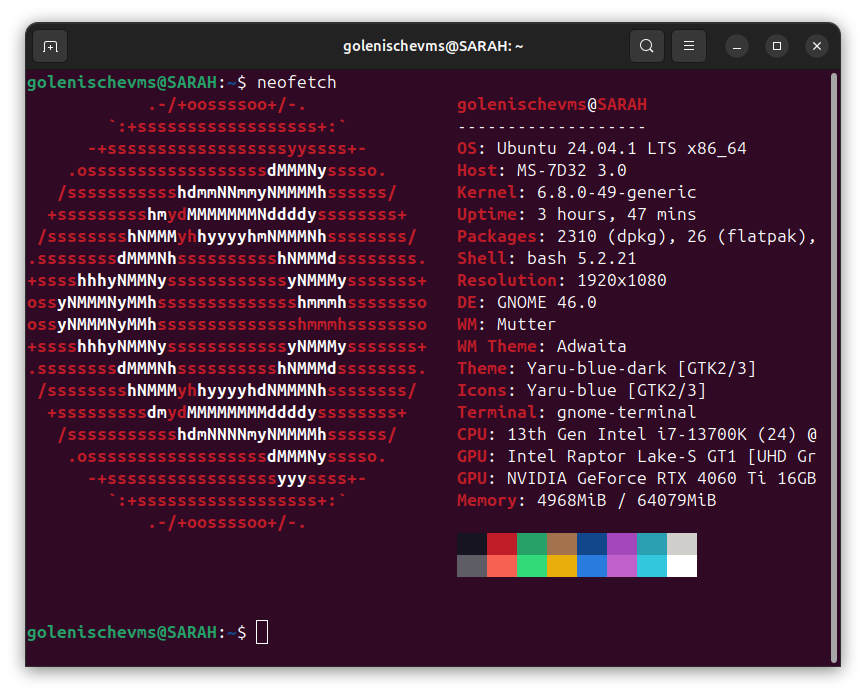
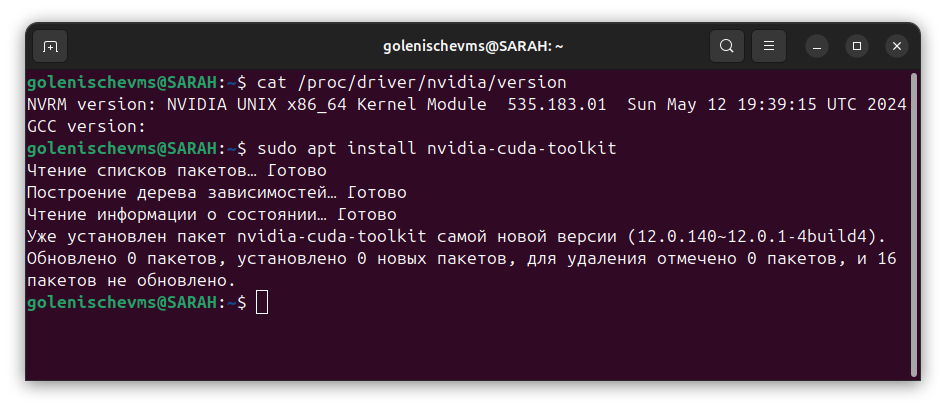
Рассмотрим конфигурацию компьютера с видеокартой, рисунок 1. Выполним проверку версии драйвера видеокарты и установим cuda, рисунок 2.

Рисунок 1. Конфигурация оборудования используемого компьютера

Рисунок 2. Проверка версии драйвера Nvidia и установка Cuda в Ubuntu

Напишем программу «Привет мир!» для запуска параллельного кода с использованием графического ускорителя, листнинг 1. Проверим ее работу,   
рисунок 3.

Листнинг 1. Код первой параллельной программы

#include <stdio.h>

#include <cuda\_runtime.h>

// Golenishchev Artem, KE-220 Lab 8

\_\_global\_\_ void HelloKernel()

{

printf("Hello world!\n");

}

int main()

{

HelloKernel<<<1, 1>>>();

cudaDeviceSynchronize(); // Синхронизация с устройством

return 0;

}

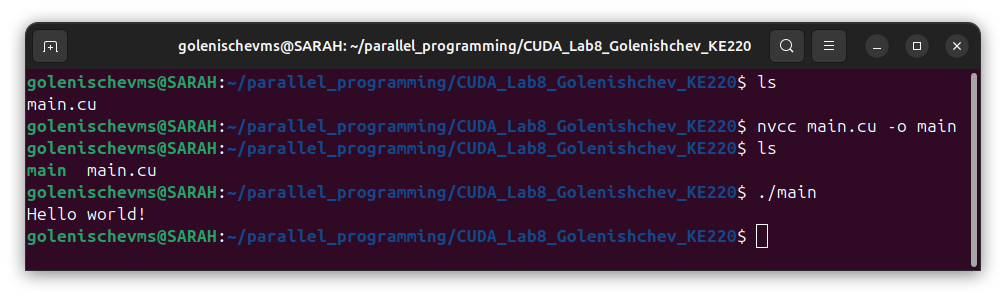


Рисунок 3. Результат выполнения поограммы «Привет, мир!»   
с использованием графического ускорителя NVIDIA

***Выводы:***

Изучили основы работы с CUDA, включая запуск простых ядер на GPU, использование асинхронного выполнения операций и необходимость синхронизации хоста и устройства для корректного вывода данных. Узнали, что функция cudaDeviceSynchronize() используется для ожидания завершения работы GPU, а также изучили особенности вывода данных с помощью printf внутри ядер CUDA.