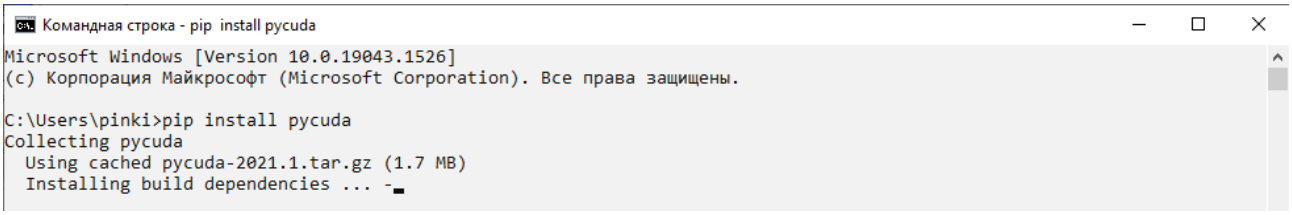
Основы работы с PyCUDA

САМОСТОЯТЕЛЬНАЯ РАБОТА

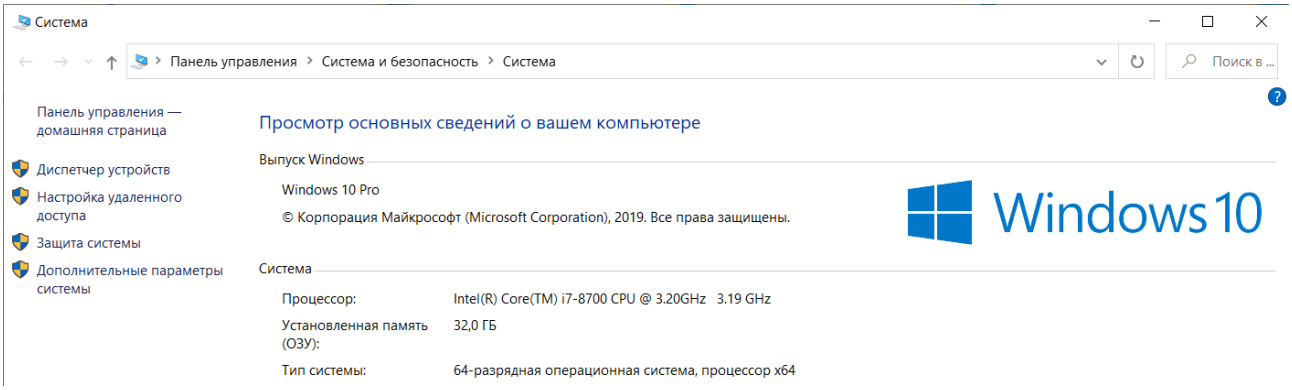
http://www.kimrt.ru
Онлайн-курс
Суперкомпьютерные технологии в задачах моделирования

Установка. Для установки *PyCuda* потребуется *Python*, *Numpy*, *Cuda toolkit*, *Visual Studio* [1] (C++ компилятор). Установка *Python* рассмотрена в самостоятельной работе «Создание оконных приложений с помощью Python и Tkinter». Установка *Numpy* рассмотрена в самостоятельной работе «Основы работы с библиотеками SciPy и NumPy». Аналогичные вопросы по *Cuda toolkit* и *Visual Studio* рассмотрены в самостоятельных работах «Установка CUDA toolkit» и «Создание OpenGL проекта в Visual Studio». Осталось установить библиотеку *PyCuda*. Для этого перейдем в *cmd* и выполним следующую команду.

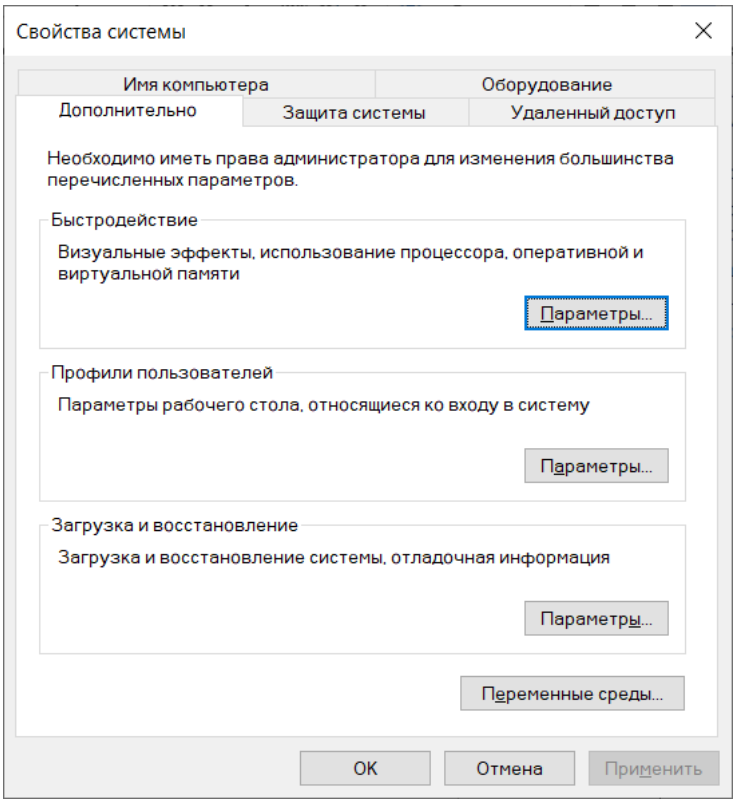
pip install pycuda



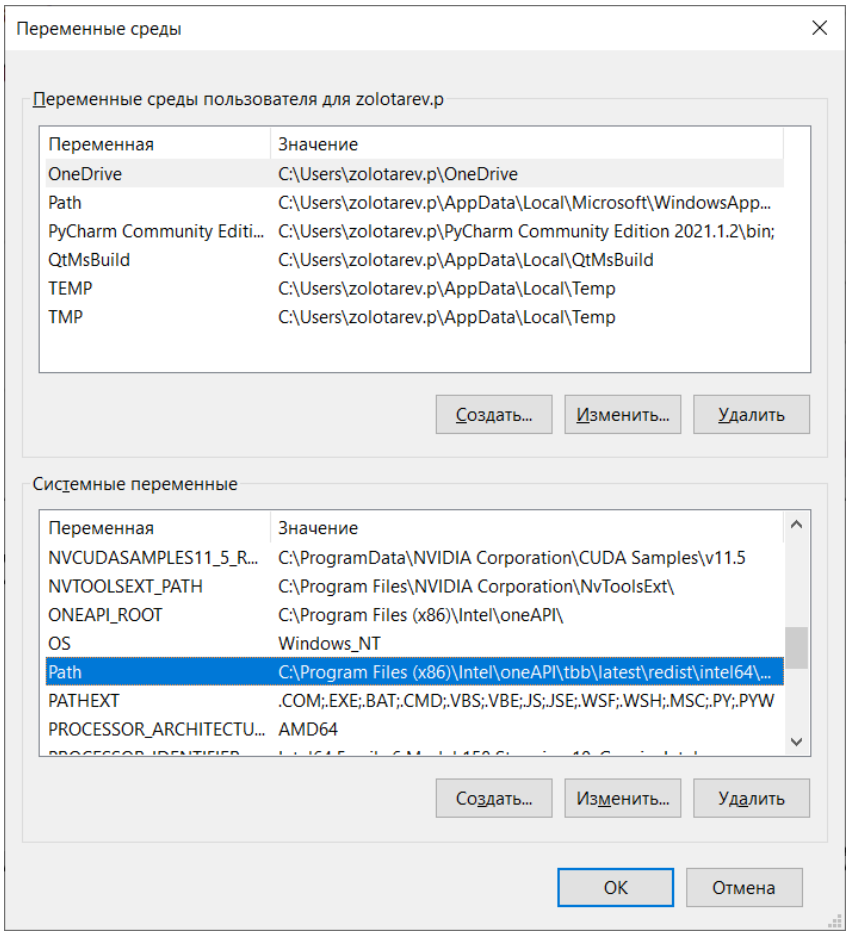
Необходимо добавить путь к *cl.exe* в переменную *PATH*. Программа *cl.exe* — это средство, управляющее компиляторами и компоновщиком Microsoft C++ (MSVC) C и C++ [2]. Для добавления пути в переменную *PATH* перейдите в *Панель управления → Система и безопасность → Система*. Это можно сделать несколькими способами, например, через папку *Этот компьютер*. Перейдите в папку и кликните правой кнопкой мыши на свободное место и выберите пункт *Свойства*. Откроется окно как на картинке ниже, если работа выполняется в операционной системе Windows 10.



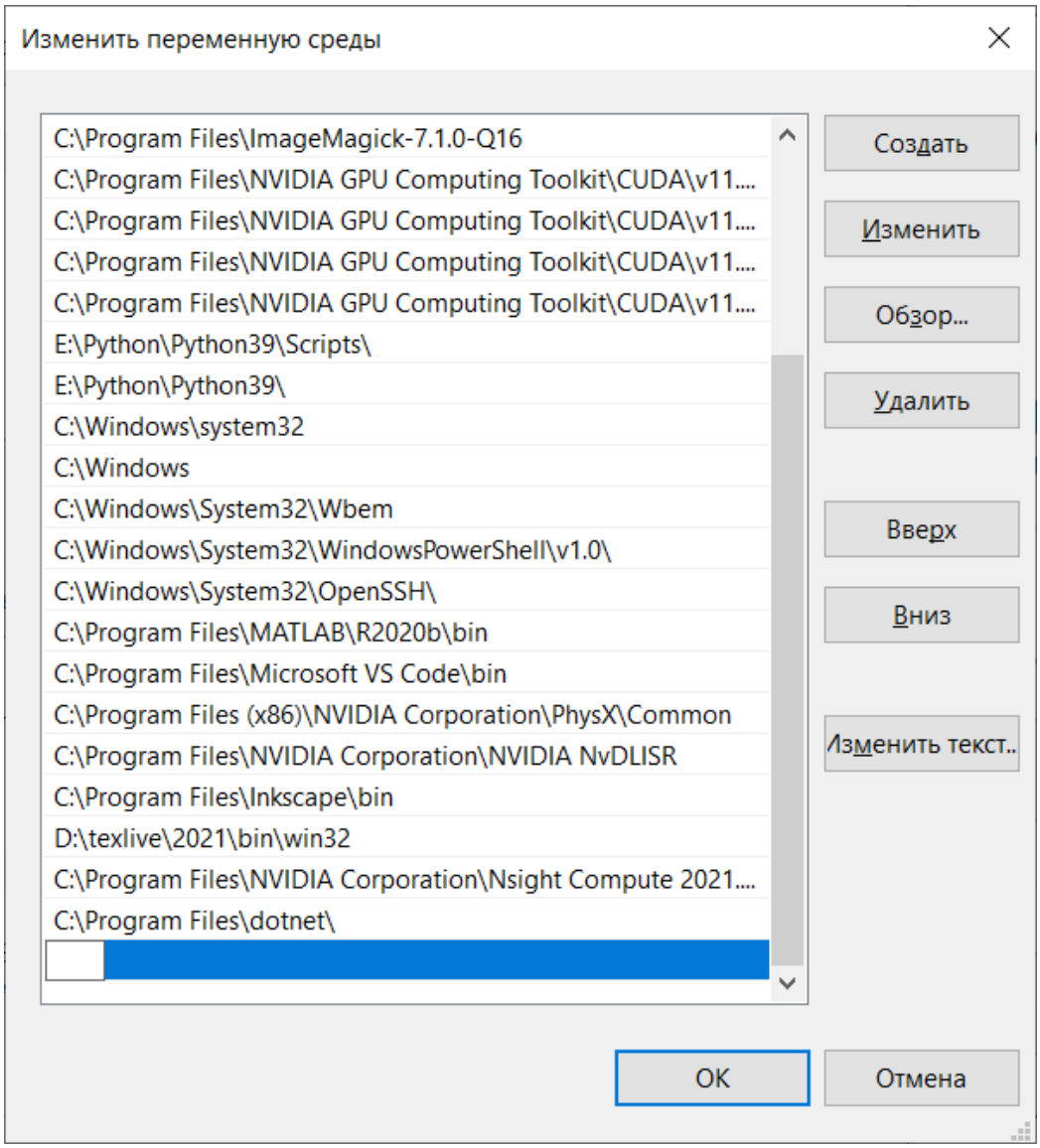
Нажмите на *Дополнительные параметры системы*.



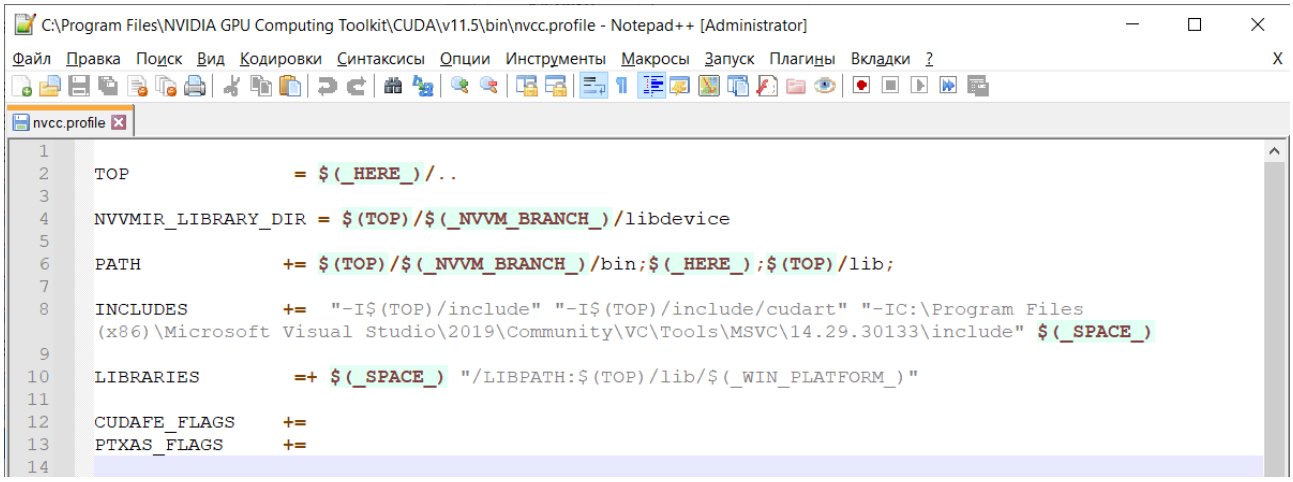
Затем нажмите на кнопку *Переменные среды…* и найдите переменную *Path* в списке *Системные переменные*.



Выделите *Path* и нажмите кнопку изменить. В следующем окне нажмите кнопку создать и в появившуюся пустую строку снизу вставьте расположение файла *cl.exe*. Например *C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\VC\Tools\MSVC\14.29.30133\bin\Hostx64\x64*. После чего нажмите кнопку *OK*.



Далее необходимо изменить файл *nvcc.profile*. Этот файл находится в папке Cuda toolkit. Например *C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.5\bin*. Открываем этот файл с помощью текстового редактора и изменяем строку начинающуюся с *INCLUDES*, на *INCLUDES += "-I$(TOP)/include" "-I$(TOP)/include/cudart" "-IC:\Program Files (x86)\Microsoft Visual Studio\2019\Community\VC\Tools\MSVC\14.29.30133\include" $(_SPACE_)*.



В третьих кавычках указывается путь до папки *include* и он может отличаться в зависимости от версии *Visual Studio* и компилятора.

Создание проекта. Открываем PyCharm и создаем новый проект. В проекте создаем файл с расширением «.py» и добавляем код из листинга 1 [3, 4].

Листинг 1. Обработка матрицы на видеокарте

```
# импорт pycuda
import pycuda.driver as cuda
import pycuda.autoinit
from pycuda.compiler import SourceModule
# импорт numpy
import numpy
# генерация массива
a = numpy.random.randn(4, 4)
# преобразование массива
a = a.astype(numpy.float32)
# выделение памяти
a_gpu = cuda.mem_alloc(a.size * a.dtype.itemsize)
# перенос данных в видеокарту
cuda.memcpy_htod(a_gpu, a)
# создание модуля из исходного CUDA кода, в котором значения в исходном массиве удваиваются
mod = SourceModule("""
    __global__ void doublify(float *a)
    {
```

```
        int idx = threadIdx.x + threadIdx.y*4;
        a[idx] *= 2;
    }
    """)
# выполнение функции удвоения
func = mod.get_function("doublify")
func(a_gpu, block=(4, 4, 1), grid=(1, 1), shared=0)
# создание пустого массива для удвоенных значений
a_doubled = numpy.empty_like(a)
# перенос данных из видеокарты
cuda.memcpy_dtoh(a_doubled, a_gpu)
# вывод
print("original array:")
print(a)
print("doubled with kernel:")
print(a_doubled)
```

Также можно ознакомиться с дополнительным материалом: инструкция по установке с использованием *anaconda* [5] и *CUDA Python* от *Nvidia* [6].

http://www.kimrt.ru/index/course_stm/0-24

Проект реализуется победителем Конкурса на предоставление грантов преподавателям магистратуры 2020/2021 благотворительной программы «Стипендиальная программа Владимира Потанина» Благотворительного фонда Владимира Потанина.

Источники

1. Installing PyCuda on Windows. URL: https://wiki.tiker.net/PyCuda/Installation/Windows/
2. Compiler Options. URL: https://docs.microsoft.com/ru-ru/cpp/build/reference/compiler-options?view=msvc-170
3. Examples. URL: https://github.com/inducer/pycuda/blob/main/examples/demo.py
4. Getting started. URL: https://documen.tician.de/pycuda/tutorial.html
5. PyCUDA installation on Windows 10. URL: https://www.youtube.com/watch?v=C8kaG2EDWcs
6. CUDA Python. URL: https://developer.nvidia.com/cuda-python