

## Практическая работа 6: Программирование на OpenCL для CPU и GPU

### Цель работы:

- Изучение основ программирования на OpenCL.
- Разработка кросс-платформенного приложения для выполнения параллельных вычислений на CPU и GPU.

### Описание:

В рамках этой практической работы нужно разработать приложение, выполняющее операцию элементного сложения двух массивов. Программа должна быть написана с использованием OpenCL и запущена как на CPU, так и на GPU для сравнения производительности.

### Задача №1.

#### 1. Подготовка окружения:

1. Установите необходимые драйверы для OpenCL на вашей системе.
  - a. Для CPU: установите драйверы от производителя процессора (Intel, AMD).
  - b. Для GPU: установите драйверы от производителя видеокарты (NVIDIA, AMD).
2. Убедитесь, что OpenCL-библиотеки доступны в вашей системе:
  - a. Linux: `/usr/lib/libOpenCL.so`
  - b. Windows: `OpenCL.dll`
3. Настройте среду разработки (например, VS Code, CLion или Visual Studio).
4. 2. Реализация задачи:

### Шаг 1. Создайте ядро OpenCL (файл `kernel.cl`):

Напишите ядро, выполняющее операцию сложения двух массивов:

```
__kernel void vector_add(__global const float* A,
                        __global const float* B,
                        __global float* C) {

    int id = get_global_id(0); // Определение глобального ID

    C[id] = A[id] + B[id];      // Выполнение операции сложения
}
```

## **Шаг 2.** Создайте программу на C/C++:

Реализуйте программу, которая:

1. Инициализирует платформу и устройство OpenCL.
2. Создает контекст и командную очередь.
3. Загружает и компилирует ядро.
4. Подготавливает данные (массивы A, B и C).
5. Выполняет ядро и считывает результаты.

## **Шаг 3.** Выполните программу:

1. Скомпилируйте код с использованием OpenCL-библиотеки.
2. Измерьте время выполнения программы на CPU и GPU (например, с использованием *clock()* или профилировщиков).

## **Шаг 4.** Сравните производительность:

Сравните производительность выполнения программы на CPU и GPU. Отчет должен содержать:

- Время выполнения на CPU.
- Время выполнения на GPU.
- График сравнения.

## **Задача №2.**

### **Описание задачи:**

Реализуйте программу для параллельного умножения двух матриц с использованием OpenCL. Матрицы A и B имеют размеры  $N \times M$  и  $M \times K$  соответственно. Программа должна вычислить результирующую матрицу C размером  $N \times K$ .

### **Шаг 1.** Создайте ядро OpenCL для матричного умножения:

### **Шаг 2.** Реализуйте программу:

1. Подготовьте матрицы A, B и C.
2. Передайте размеры матриц (N, M, K) в качестве аргументов ядра.
3. Настройте размеры рабочей группы, соответствующие размеру матрицы C (глобальная рабочая группа).

### **Шаг 3.** Проверьте корректность результатов:

Сравните результаты OpenCL с последовательной реализацией матричного умножения на CPU.

**Контрольные вопросы:**

1. Какие основные типы памяти используются в OpenCL?
2. Как настроить глобальную и локальную рабочую группу?
3. Чем отличается OpenCL от CUDA?
4. Какие преимущества дает использование OpenCL?

**Отчет:**

В отчете должно быть:

1. Код программы с комментариями.
2. Скриншоты или текстовые результаты выполнения программы.
3. Сравнение времени выполнения на CPU и GPU.
4. Ответы на контрольные вопросы.