

Санкт-Петербургский государственный электротехнический университет им.
В.И. Ульянова (Ленина)

Разработка отладчика для OpenCL

Выполнил:

Лосев М.Л., гр. 7383

Руководитель:

Беляев Сергей Алексеевич, к.т.н., доцент

Консультант:

Лукин Михаил Андреевич, к.т.н., доцент

Санкт-Петербург, 2021

Цель и задачи

Актуальность: отладка на устройствах AMD осуществляется с помощью CodeXL, который не поддерживает новые графические процессоры.

Цель: разработать устройствовнезависимый отладчик для OpenCL

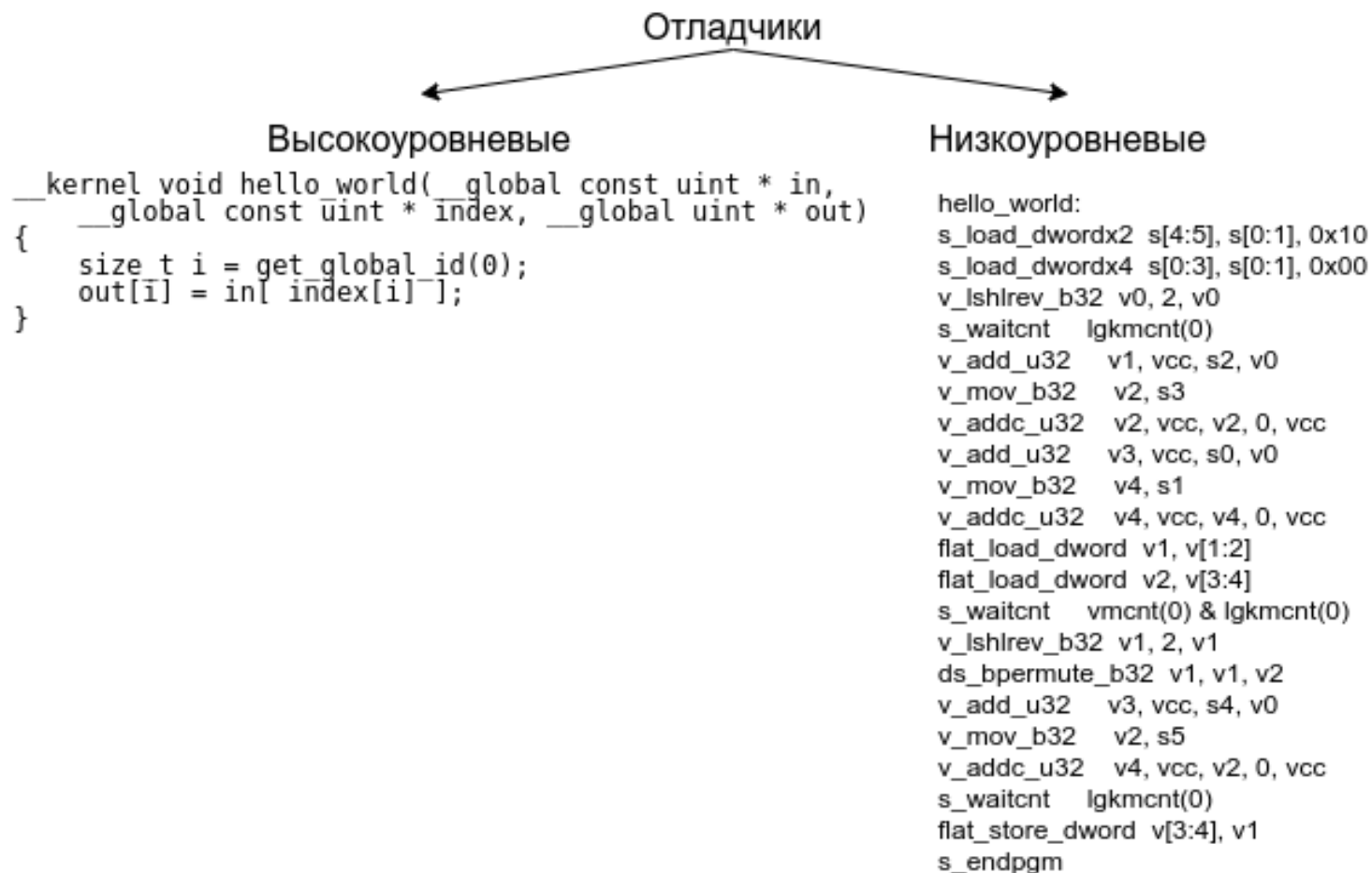
Задачи:

1. Обзор аналогов
2. выбор подхода и технологий разработки;
3. Реализация получения значений переменных;
4. Реализация интерпритации значений переменных;

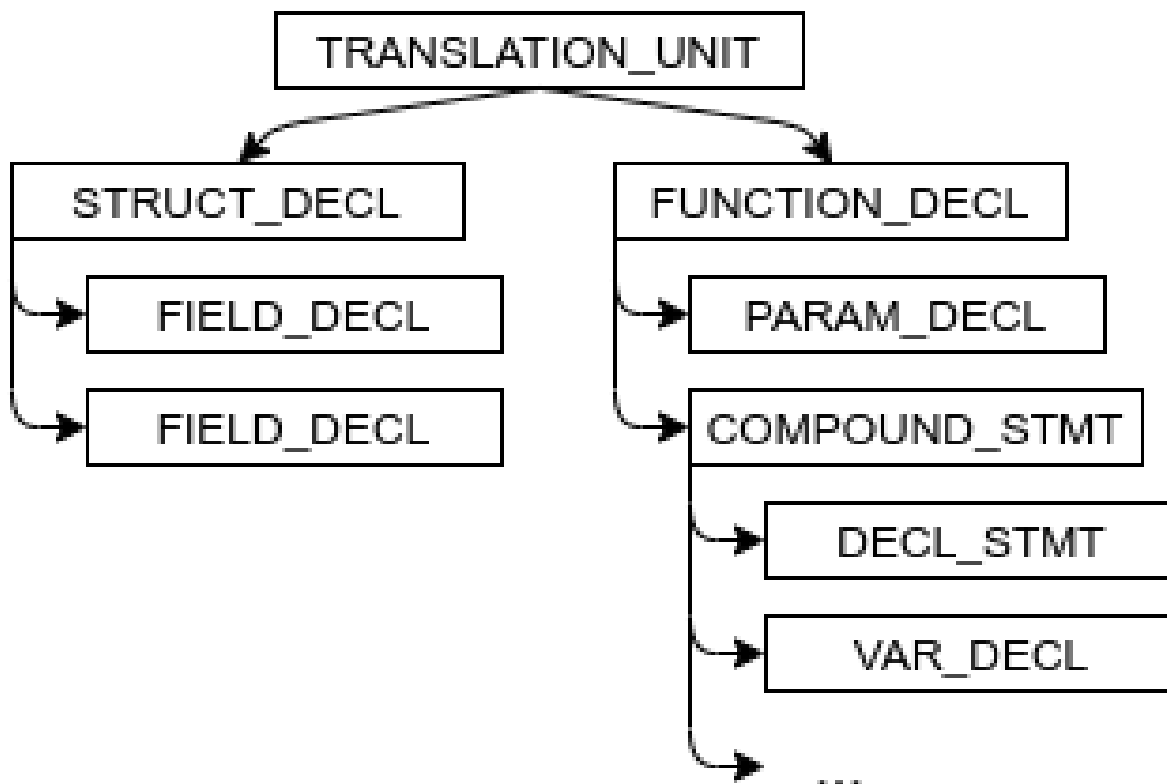
Сравнение аналогов

| | CodeXL | Oclgrind | OpenCL API Debugger | NVIDIA Nsight | gDEDebugger |
|------------------------------|--------|----------|---------------------------|------------------|-------------|
| Платформо- независимость | - | + | + | + | + |
| Устройство- независимость | - | + | - | - | + |
| Интеграция с IDE | + | - | + | + | - |

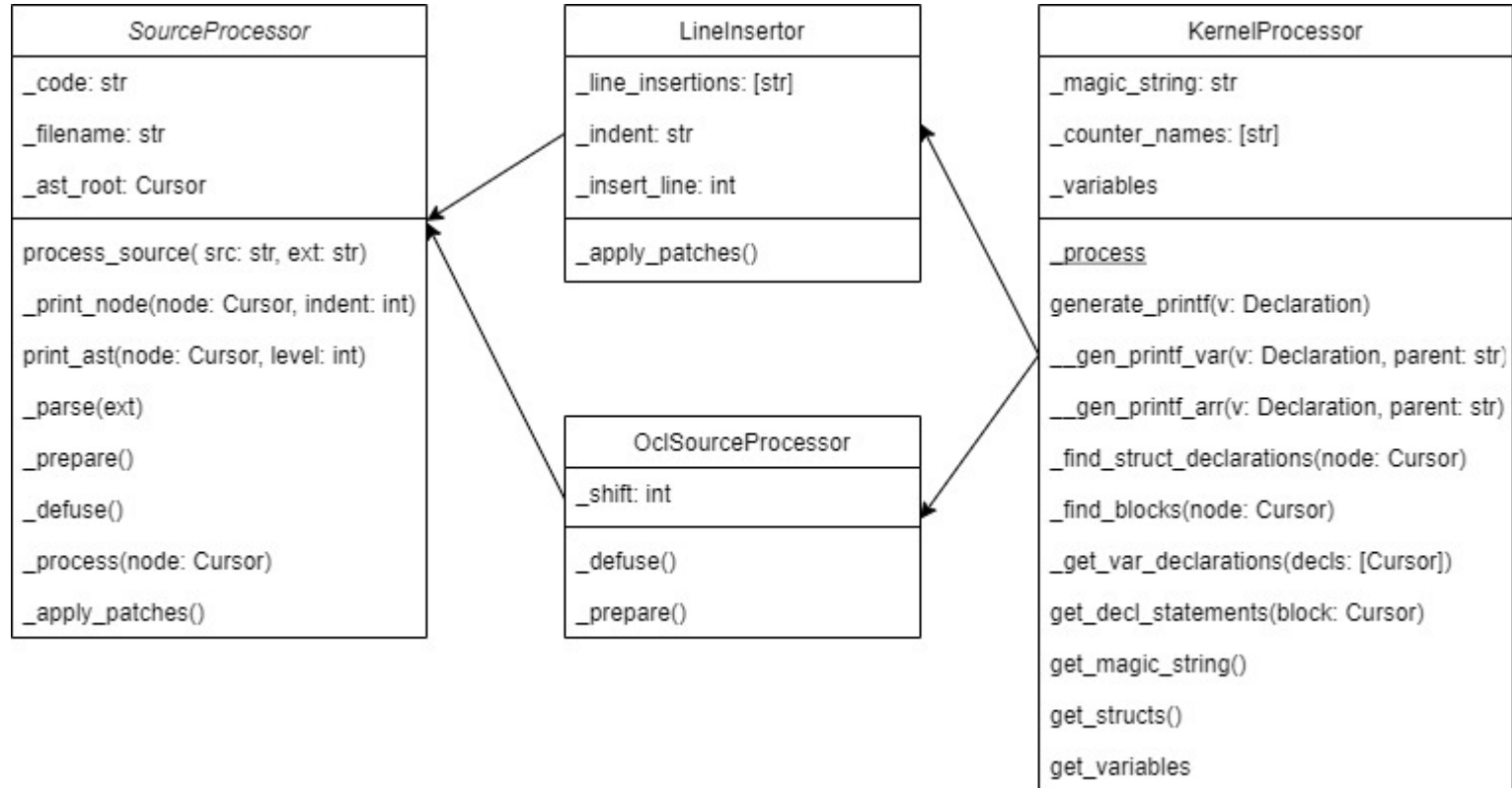
Подходы к разработке отладчиков



Получение значений переменных. Clang



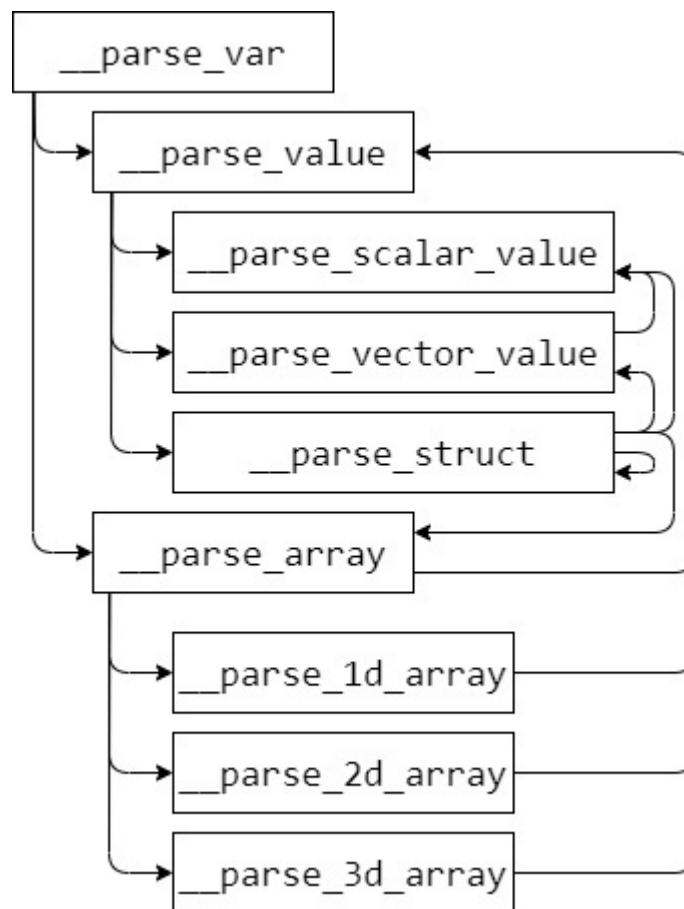
Получение значений переменных. Модификация кода ядра



Интерпритация значений переменных. Формат данных

| Объявление | Пример представления |
|-------------------------------------|-----------------------------------|
| <code>int a = 18;</code> | <code>"a 12"</code> |
| <code>int2 a = (int2)(1, 2);</code> | <code>"a 1,2"</code> |
| <code>int a[3] = {1, 2, 3};</code> | <code>"a 0x00004d08 1 2 3"</code> |

Интерпритация значений переменных. Реализация



Пример работы отладчика

```
1 struct my_struct {
2     uint count;
3     double2 v;
4 };
5
6 __kernel void test(__global int* message, __global char* debuggingBuffer)
7 {
8     struct my_struct s = {0, (double2)(0.1, 0.2)};
9 }
10
11
```

```
/usr/bin/python3.8 /home/mikhail/src/GRADUATE_WORK/d2/OclDebugger
{'name': 's', 'gid': 0, 'value': {'count': 0, 'v': [0.1, 0.2]}}
{'name': 's', 'gid': 3, 'value': {'count': 0, 'v': [0.1, 0.2]}}

Process finished with exit code 0
```

Заключение

- Прodelанный обзор аналогов показал необходимость разработки устройственезависимого OpenCL-отладчика с возможностью интеграции в IDE
- Выбран высокоуровневый подход и библиотека Clang для работы с кодом
- Спроектирован и опубликован отладчик для OpenCL

Дальнейшие направления разработки включают в себя создание адаптера для интеграции отладчика в IDE, добавление нового функционала в отладчик

Апробация работы

- Репозиторий проекта
<https://github.com/LosevMikhail/OclDebugger>.
- Отладчик внедрен в проект, реализуемый ООО “Судо”



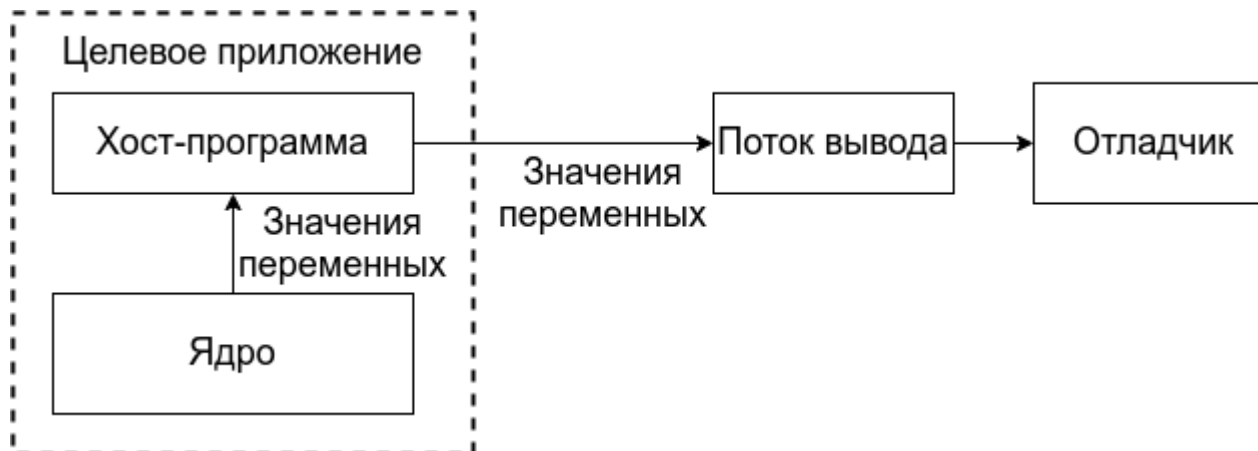
Акт о внедрении



Репозиторий

Запасные слайды

Получение значений переменных. Схема



Обеспечение качества разработки

•

