

# Основы OpenCL

## Лекция 3: Язык OpenCL C

Авторы курса:

- Геллер Олег
- Казёнов Андрей



# Основы OpenCL



## Структура приложения OpenCL C

Приложение OpenCL состоит из двух частей:

➤ Код, выполняемый на хосте, содержит вызовы функций библиотеки OpenCL для управления расчетами и может быть написан на любом языке программирования.

➤ Код, выполняемый на вычислительных устройствах. Компилируется драйвером каждого устройства. Этот код должен быть написан на специальном языке -- OpenCL C.



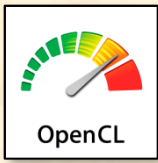
# Основы OpenCL



## Язык OpenCL C

**OpenCL C** основан на стандарте **ISO/IEC 9899:1999** языка **C**, однако содержит некоторые расширения и ограничения:

- Содержит специальные механизмы для написания **ядер** – массивно параллельных процедур
- Содержит векторные типы данных, отсутствующие в **C**
- Имеются специальные ключевые слова
- Отсутствуют стандартные библиотеки
- Имеются встроенные функции, реализующие широкую функциональность
- Предусмотрен механизм **extensions** (расширений), для функциональности не вошедшей в стандарт, но поддерживаемой некоторыми устройствами и платформами.
- Некоторые **extensions** (расширения) входят в следующие стандарты языка OpenCL C



# Основы OpenCL



## Типы данных OpenCL C

Поддерживаются стандартные скалярные типы данных, имеется ряд синонимов:

<b>void</b>	<b>int</b>	<b>unsigned int = uint</b>
<b>bool</b>	<b>long</b>	<b>unsigned long = ulong</b>
<b>char</b>	<b>unsigned char = uchar</b>	<b>float</b>
<b>short</b>	<b>unsigned short = ushort</b>	<b>half</b>

Поддерживаются векторные типы данных:

<b>char<math>N</math></b>	<b>long<math>N</math></b>	<b>uint<math>N</math></b>
<b>short<math>N</math></b>	<b>uchar<math>N</math></b>	<b>ulong<math>N</math></b>
<b>int<math>N</math></b>	<b>ushort<math>N</math></b>	<b>float<math>N</math></b>

$N = 2, 4, 8, 16$

Специальные типы данных:

- image2d\_t**
- image3d\_t**
- sampler\_t**
- event\_t**



# Основы OpenCL

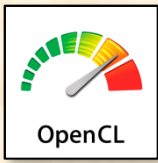


## Операторы OpenCL C

Поддерживаются стандартные операторы:

`+, -, *, /, [], ++, --, <, >, <=, >=, ==, !=, &, |, ~, ^, &&, ||, !`

**Для векторных типов операторы работают покомпонентно**



# Основы OpenCL



## Спецификаторы функций OpenCL C

### **\_\_kernel (kernel)**

Показывает, что функция является ядром

- Ядро не может возвращать значения
- Ядро может вызываться с хоста
- Ядро может вызываться из другой функции
- Аргументы ядер, являющиеся указателями, должны принадлежать к адресным пространствам **\_\_constant**, **\_\_local** или **\_\_global**
- Аргументы ядер не могут быть указателями на указатели
- Аргументы ядер не могут иметь типы **bool**, **half** и **size\_t**

Пример:

```
__kernel void example_kern(__global float *in, __global float *out, int n)
```



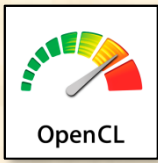
# Основы OpenCL



## Спецификаторы адресных пространств OpenCL C

- **\_\_global (global)**  
Спецификатор глобальной памяти
- **\_\_local (local)**  
Спецификатор локальной памяти
- **\_\_constant (constant)**  
Спецификатор константной памяти
- **\_\_private (private)**  
Спецификатор приватной памяти

Спецификатором по умолчанию является **\_\_private**



# Основы OpenCL

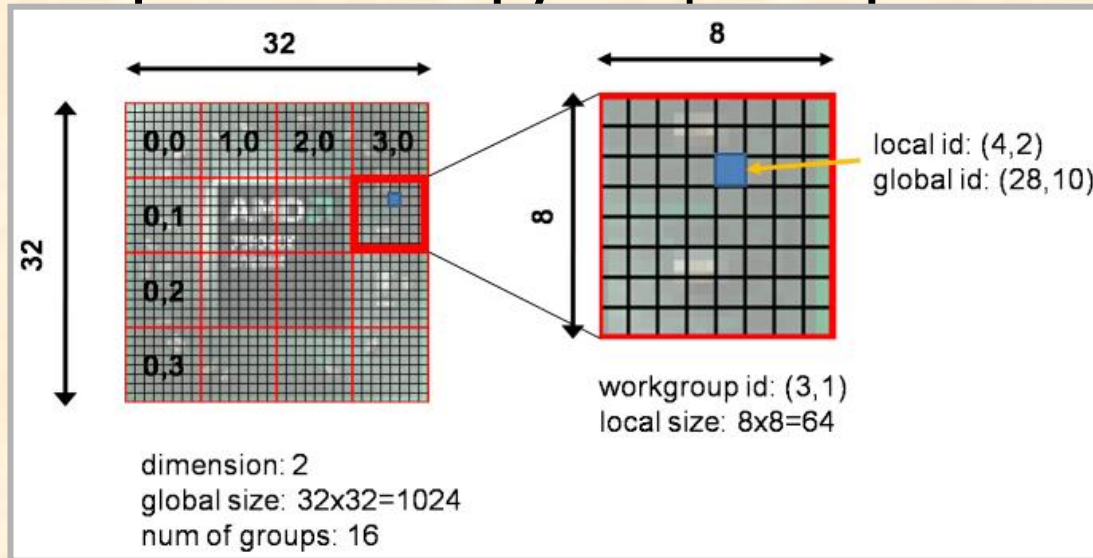


## Ограничения

- Указатели, декларированные со спецификатором адресного пространства могут указывать только на память из соответствующего пространства
  - Указатели на функции не поддерживаются
  - Побитовое обращение к переменным не поддерживается
  - Заголовки стандартных библиотек не поддерживаются:  
`assert.h, ctype.h, complex.h, errno.h, fenv.h, float.h, inttypes.h, limits.h, locale.h, setjmp.h, signal.h, stdarg.h, stdio.h, stdlib.h, string.h, tgmath.h, time.h, wchar.h and wctype.h`
  - Рекурсия не поддерживается
- Запись по указателю в переменные, типов размером меньших 32 бит (**char, uchar, char2, uchar2, short, ushort, half**) не поддерживается
- Поля структуры и объединения (**struct, union**) должны принадлежать к одному адресному пространству



## Встроенные функции OpenCL C



### Функции Work-Item

`uint get_work_dim ()`

`size_t get_global_size (uint dimindx)`

`size_t get_global_id (uint dimindx)`

`size_t get_local_size (uint dimindx)`

`size_t get_local_id (uint dimindx)`

`size_t get_num_groups (uint dimindx)`

`size_t get_group_id (uint dimindx)`



# Основы OpenCL



## Встроенные функции OpenCL C

### Функции Work-Item

`uint get_work_dim ()`

`size_t get_global_size (uint dimindx)`

`size_t get_global_id (uint dimindx)`

`size_t get_local_size (uint dimindx)`

`size_t get_local_id (uint dimindx)`

`size_t get_num_groups (uint dimindx)`

`size_t get_group_id (uint dimindx)`