

# ОСНОВЫ ТЕХНОЛОГИИ CUDA

Семинар 4: Работа с глобальной памятью

Автор курса:

➤ Казённых Андрей

## Типы памяти в CUDA

Тип памяти	Доступ	Уровень выделения	Скорость работы
Регистры	R/W	Per-thread	Высокая(on-chip)
Локальная	R/W	Per-thread	Низкая (DRAM)
Shared	R/W	Per-block	Высокая(on-chip)
<b>Глобальная</b>	<b>R/W</b>	<b>Per-grid</b>	<b>Низкая (DRAM)</b>
Constant	R/O	Per-grid	Высокая(L1 cache)
Texture	R/O	Per-grid	Высокая(L1 cache)

## Функции для работы с глобальной памятью

➤ `cudaError_t cudaMalloc ( void ** devPtr, size_t size );` *Выделение **size** памяти на устройстве и записывание адреса в **devPtr***

➤ `cudaError_t cudaMallocPitch ( void ** devPtr, size_t * pitch, size_t width, size_t height );`

*Выделение памяти под двумерный массив размером **width \* height**, возвращает указатель **devPtr** на память и смещение для каждой строки **pitch***

➤ `cudaError_t cudaFree ( void * devPtr );`

*Освобождение памяти по адресу **devPtr** на устройстве*

➤ `cudaError_t cudaMemcpy ( void* devPtr, int value, size_t count );`

*Заполнение памяти по адресу **devPtr** значениями **value** на размер **count***

## Функции для работы с глобальной памятью

➤ `cudaError_t cudaMemcpy ( void * dst, const void * src, size_t count, enum cudaMemcpyKind kind );`

*Копирование данных между устройством и хостом*

***dst** – указатель на память приемник*

***src** – указатель на память источник*

***count** – размер копируемой памяти в байтах*

***kind** – направление копирования может принимать значения:*

➤ ***cudaMemcpyHostToDevice** – с хоста на устройство*

➤ ***cudaMemcpyDeviceToHost** – с устройства на хост*

➤ ***cudaMemcpyDeviceToDevice** – с устройства на устройство*

➤ ***cudaMemcpyHostToHost** – с хоста на хост*

➤ `cudaError_t cudaMemcpyAsync ( void * dst, const void * src, size_t count, enum cudaMemcpyKind kind, cudaStream_t stream );`

*Аналогично **cudaMemcpy** только асинхронно в потоке **stream***



# ОСНОВЫ ТЕХНОЛОГИИ CUDA



## Стандартная схема работы с глобальной памятью

```
float * devPtr;           // pointer device memory
                          // allocate device memory
cudaMalloc ( (void **) &devPtr, 256*sizeof ( float ) );

                          // copy data from host to device memory
cudaMemcpy ( devPtr, hostPtr, 256*sizeof ( float ),
cudaMemcpyHostToDevice );

                          // process data

                          // copy results from device to host
cudaMemcpy ( hostPtr, devPtr, 256*sizeof( float ),
cudaMemcpyDeviceToHost );

                          // free device memory
cudaFree   ( devPtr );
```



# Основы технологии CUDA



Сборка CUDA программ

**`nvcc file.cu -o file.o`**

В нашем курсе все примеры имеют CmakeList.txt

**`cmake . & make`**



# Основы технологии CUDA



Доступ к вычислительной машине

```
ssh -p 15136 user@hpc.mipt.ru
```

Рабочие файлы примеров

```
/opt/hpcschool/cuda
```

**Все примеры копировать в свою  
рабочую папку!!!**

## Пример: Нахождение числа Пи

- Пример лежит в файле pi.cu
- Ядро занимается расчетом площади трапеций
- Суммирование полученных площадей производится на хосте
- => Необходимо создать массив для передачи данных с устройства на хост

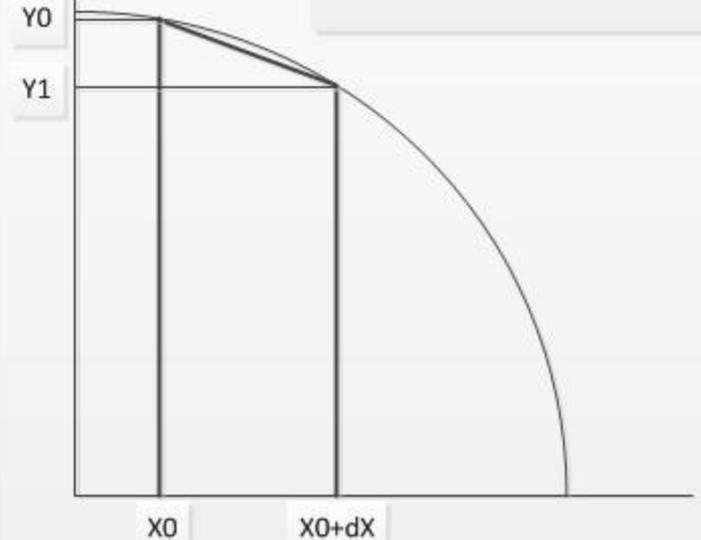
$$\pi = \int_0^1 \sqrt{1 - x^2} dx$$

$$Y_0 = \sqrt{1 - X_0^2}$$

$$R = 1$$

$$Y_1 = \sqrt{1 - (X_0 + dX)^2}$$

$$S = dX (Y_0 + Y_1) / 2$$





# ОСНОВЫ ТЕХНОЛОГИИ CUDA



## Пример: Нахождение числа Пи: Ядро

```
__global__ void pi_kern(CUDA_FLOAT *res)
{
    int n = threadIdx.x + blockIdx.x * BLOCK_SIZE;
    CUDA_FLOAT x0 = n * 1.f / (BLOCK_SIZE * GRID_SIZE); // Начало
отрезка интегрирования
    CUDA_FLOAT y0 = sqrtf(1 - x0 * x0);
    CUDA_FLOAT dx = 1.f / (1.f * BLOCK_SIZE * GRID_SIZE *
THREAD_SIZE); // Шаг интегрирования
    CUDA_FLOAT s = 0; // Значение интеграла по отрезку, данному
текущему треду
    CUDA_FLOAT x1, y1;
    for (int i=0; i < THREAD_SIZE; ++i)
    {
        x1 = x0 + dx;
        y1 = sqrtf(1 - x1 * x1);
        s += (y0 + y1) * dx / 2.f; // Площадь трапеции
        x0 = x1; y0 = y1;
    }
    res[n] = s; // Запись результата в глобальную память
}
```



# Основы технологии CUDA

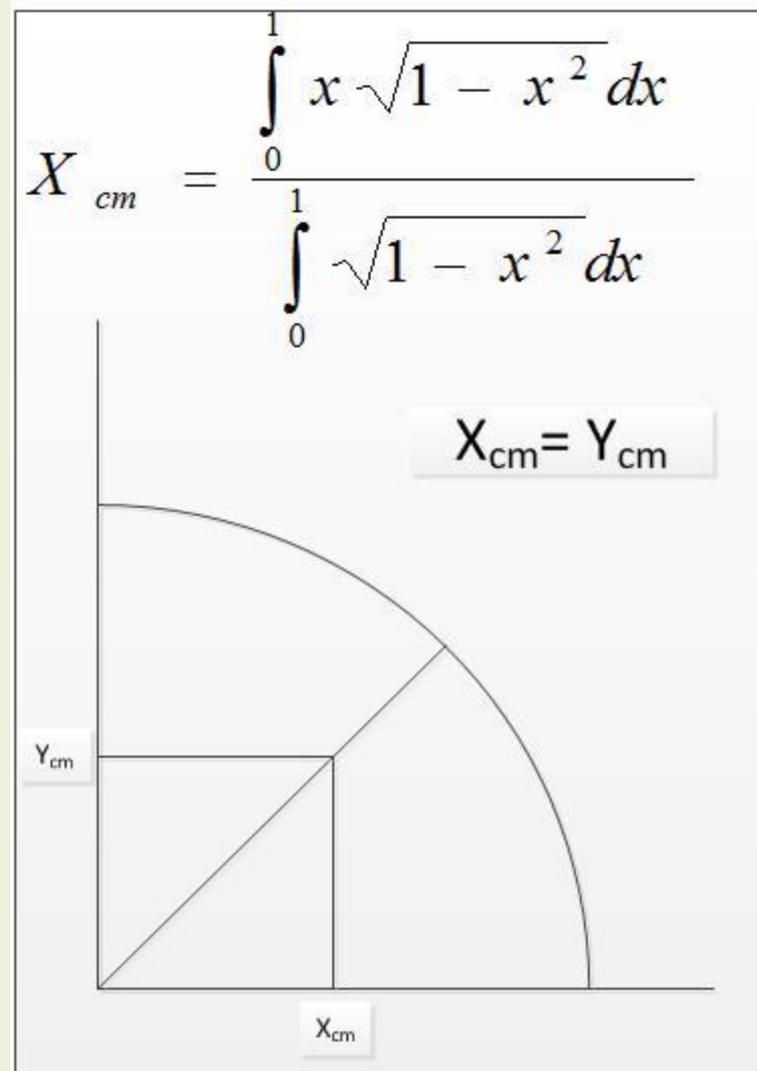


Пример: Нахождение числа Пи: Main

**Смотрите в примере на сервере**

## Задача №1: Нахождение центра масс четверти круга

- Можно добавить ещё один массив в параметры ядра и записать результат второго интеграла в него
- Можно написать самостоятельное второе ядро
- Найдите  $X_{cm}$ ?!





# Ресурсы курса

Сайт: [HPC.MIPT.RU](http://HPC.MIPT.RU)  
Раздел образование

Автор курса:

Казённов Андрей

➤ E-mail: [kazenov@gmail.com](mailto:kazenov@gmail.com)

➤ ICQ: 622774