CUDA 프로그래밍

CUDA Programming

biztripcru@gmail.com

Error Check

에러 체크

내용 contents

- another simple CUDA kernel example
- CUDA error check methods
- cudaError_t
- cudaGetErrorName, cudaGetErrorString
- cudaGetLastError, cudaPeekAtLastError
- CUDA error check macros

add-kernel 예제

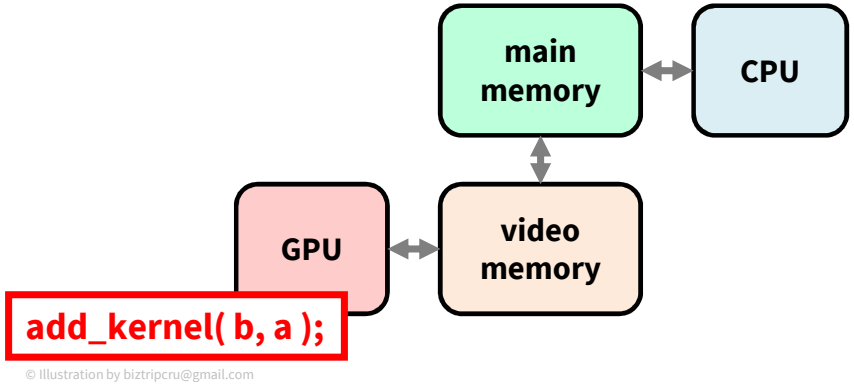
- **1D array (vector) 에 대한 더하기**
  - SIZE = 8
  - 1D array a[SIZE], b[SIZE] : SIZE 개의 숫자
  - 더하기 : b[i] = a[i] + 1.0

- **병렬 처리로 더하기 parallel addition**
  - CUDA kernel 로 처리
  - **kernel : 병렬처리가 가능한 함수 function**
  - kernel launch : kernel 을 병렬 처리로 call

Program: add-kernel.cu

```
#include <stdio.h>

// CUDA kernel function
__global__ void add_kernel( float* b, const float* a ) {
   int i = threadIdx.x;
   b[i] = a[i] + 1.0f;
}
```

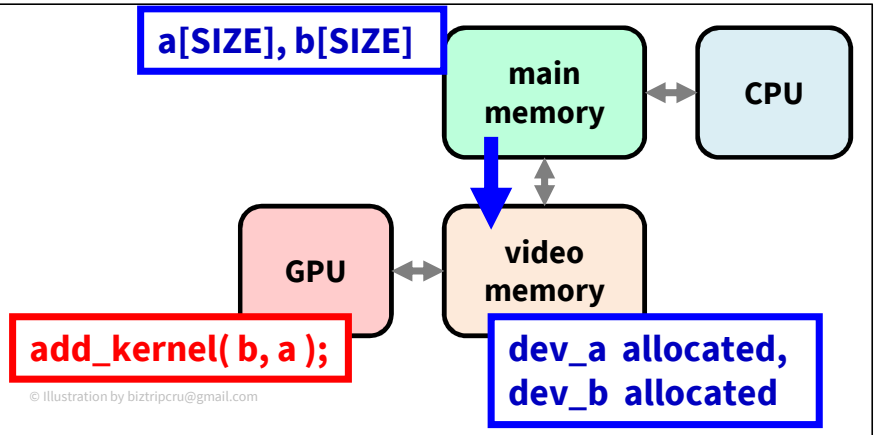


- **later in "main( )",**

```
const int SIZE = 8;
// CUDA kernel call
add_kernel<<<1,SIZE>>>( dev_b, dev_a );
```

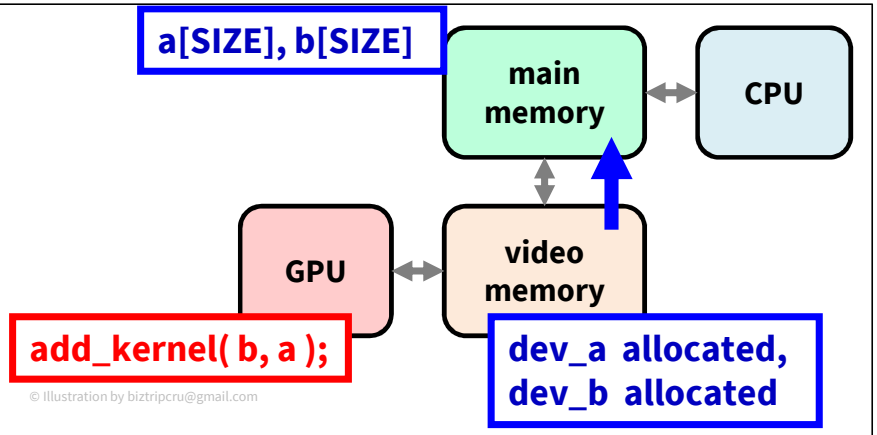
Program: add-kernel.cu 계속

```
int main(void) {
  // host-side data
  const int SIZE = 8;
  const float a[SIZE] = { 0., 1., 2., 3., 4., 5., 6., 7.  };
  float b[SIZE] = { 0., 0., 0., 0., 0., 0., 0., 0. };
  // print source
  printf("a = {%f,%f,%f,%f,%f,%f,%f,%f}\n",
        a[0], a[1], a[2], a[3], a[4], a[5], a[6], a[7]);
  // device-side data
  float* dev_a = nullptr;
  float* dev_b = nullptr;
  // allocate device memory
  cudaMalloc( (void**)&dev_a, SIZE * sizeof(float) );
  cudaMalloc( (void**)&dev_b, SIZE * sizeof(float) );
  cudaMemcpy( dev_a, a, SIZE * sizeof(float), cudaMemcpyHostToDevice); // dev_a = a;
```



Program: add-kernel.cu 계속

```
  // kernel
  add_kernel<<<1,SIZE>>>( dev_b, dev_a );
  cudaDeviceSynchronize();
  // print the result
  cudaMemcpy( b, dev_b, SIZE * sizeof(float),
              cudaMemcpyDeviceToHost); // b = dev_b;
  printf("b = {%f,%f,%f,%f,%f,%f,%f,%f}\n",
       b[0], b[1], b[2], b[3], b[4], b[5], b[6], b[7]);
  // free device memory
  cudaFree( dev_a );
  cudaFree( dev_b );
  // done
  return 0;
}
```



Program: add-kernel.cu 계속

- **실행 결과**

```
const float a[SIZE] = { 0., 1., 2., 3., 4., 5., 6., 7. };
float b[SIZE] = { 0. };
```

Windows PowerShell

```
C:\cuda-work> .\021-2-cuda-kernel.exe
a = {0.000000,1.000000,2.000000,3.000000,4.000000,5.000000,6.000000,7.000000}
b = {1.000000,2.000000,3.000000,4.000000,5.000000,6.000000,7.000000,8.000000}

C:\cuda-work>
```

CUDA function rules

- 모든 CUDA 함수는 "cuda"로 시작
- 대부분은 에러 코드error code를 리턴 (성공시는 cudaSuccess).

- **Example:**

```
if (cudaMalloc( &devPtr, SIZE ) != cudaSuccess) {
   exit(1);
}
```

cudaError_t : data type

- **typedef enum cudaError cudaError_t**
- **possible values:**

  cudaSuccess, cudaErrorMissingConfiguration, cudaErrorMemoryAllocation, cudaErrorInitializationError, cudaErrorLaunchFailure, cudaErrorLaunchTimeout, cudaErrorLaunchOutOfResources, cudaErrorInvalidDeviceFunction, cudaErrorInvalidConfiguration, cudaErrorInvalidDevice, cudaErrorInvalidValue, cudaErrorInvalidPitchValue, cudaErrorInvalidSymbol, cudaErrorUnmapBufferObjectFailed, cudaErrorInvalidHostPointer, cudaErrorInvalidDevicePointer, cudaErrorInvalidTexture, cudaErrorInvalidTextureBinding, cudaErrorInvalidChannelDescriptor, cudaErrorInvalidMemcpyDirection, cudaErrorInvalidFilterSetting, cudaErrorInvalidNormSetting, cudaErrorUnknown, cudaErrorNotYetImplemented, cudaErrorInvalidResourceHandle, cudaErrorInsufficientDriver, cudaErrorSetOnActiveProcess, cudaErrorStartupFailure, cudaErrorApiFailureBase

cudaGetErrorName(err)

- const char* **cudaGetErrorName**(cudaError_t *err*);
  - *err* : error code to convert to a string
  - returns:
    - const char* to a **NULL-terminated string**
    - NULL / nullptr if the error code is not valid

- **cout << cudaGetErrorName(cudaErrorMemoryAllocation) << endl;**
  - shows:
    cudaErrorMemoryAllocation

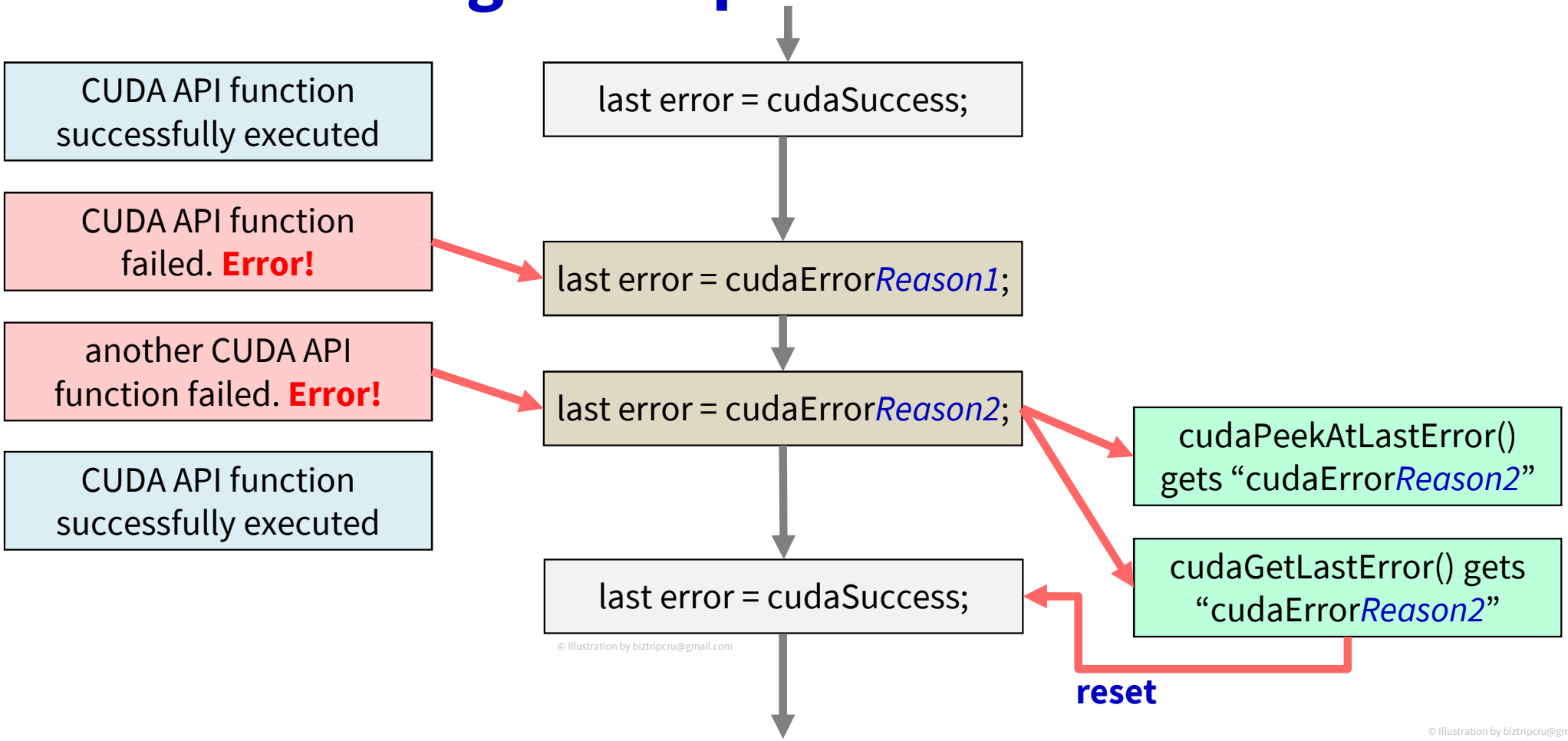
cudaGetErrorString(err)

- const char* **cudaGetErrorString**(cudaError_t err);
  - err : error code to convert to an explanation string
  - returns:
    - const char* to a **NULL-terminated string**
    - NULL / nullptr if the error code is not valid

- **cout << cudaGetErrorString(cudaErrorMemoryAllocation) << endl;**
  - shows:

    out of memory

CUDA Error Flag Concept



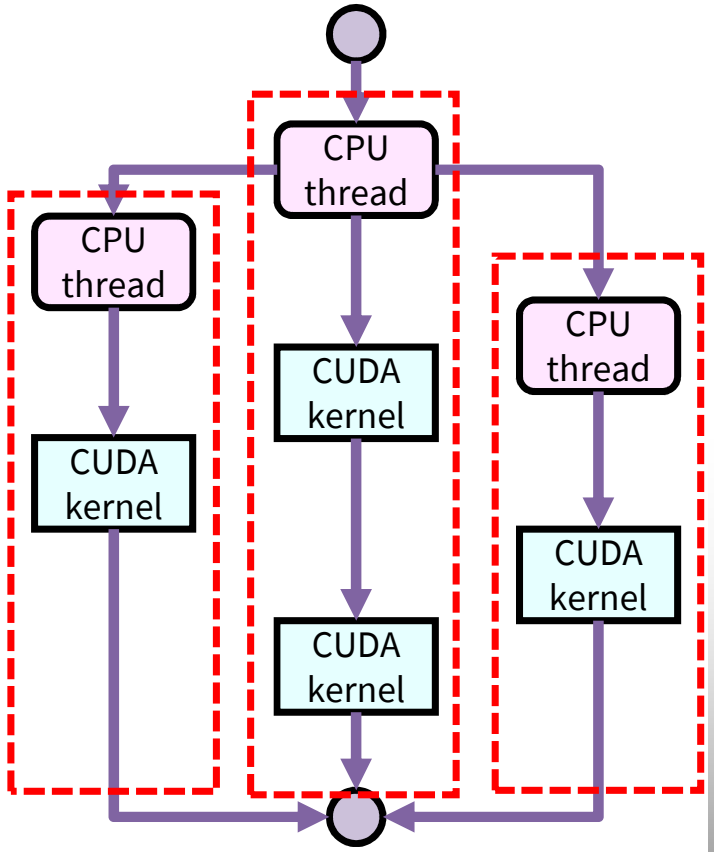
cudaGetLastError( )

- cudaError_t **cudaGetLastError**(void);
  - returns **the last error** due to CUDA runtime calls in the same host thread
  - and **resets** it to **cudaSuccess**
  - So, if no CUDA error since the last call, it returns **cudaSuccess**
  - For multiple errors, it contains **the last error only.**
- cudaError_t **cudaPeekAtLastError**(void);
  - returns the last error
  - Note that this call does **NOT** reset the error to **cudaSuccess**
  - So, the last error code is still available

cudaGetLastError( ) – thread 단위 처리

- cudaError_t **cudaGetLastError**(void);
  - returns **the last error** due to CUDA runtime calls in the same host thread

- **CUDA error 처리는 CPU thread 기준**
  - CPU thread 마다 별도의 error 상태 관리



3개의 CUDA error 관리

Error Check

```
  …
  // print the result
  printf("b = {%f,%f,%f,%f,%f,%f,%f,%f}\n", b[0], b[1], b[2], b[3], b[4], b[5], b[6], b[7]);
  // error check
  cudaError_t err = cudaGetLastError();
  if (cudaSuccess != err) {
    printf("CUDA: ERROR: cuda failure \"%s\"\n", cudaGetErrorString(err));
    exit(1);
  } else {
    printf("CUDA: success\n");
  }
  // done
  return 0;
}
```

Windows PowerShell

```
C:\cuda-work> .\021-3-error-check.exe
a = {0.000000,1.000000,2.000000,3.000000,4.000000,5.000000,6.000000,7.000000}
b = {1.000000,2.000000,3.000000,4.000000,5.000000,6.000000,7.000000,8.000000}
CUDA: success

C:\cuda-work>
```

CUDA error check macro

```
#define CUDA_CHECK_ERROR()   \
    cudaError_t e = cudaGetLastError(); \
    if (cudaSuccess != e) { \
      printf("cuda failure \"%s\" at %s:%d\n", \
          cudaGetErrorString(e), __FILE__, __LINE__); \
      exit(1); \
    }
```

- **semi-colon problem ...**
  - CUDA_CHECK_ERROR() → OK.
  - CUDA_CHECK_ERROR(); → syntax error !

CUDA error check macro 계속

```
#define CUDA_CHECK_ERROR()  do { \
    cudaError_t e = cudaGetLastError(); \
    if (cudaSuccess != e) { \
      printf("cuda failure \"%s\" at %s:%d\n", \
          cudaGetErrorString(e), __FILE__, __LINE__); \
      exit(1); \
    } \
  } while (0)
```

- **semi-colon problem …**
  - CUDA_CHECK_ERROR(); → OK.

Release mode, Debug mode

- **most C/C++ compilers define:**
  - _DEBUG for debug mode
  - NDEBUG for release mode

- **so...**

```
#if defined(NDEBUG)
        ... code for release mode ...
#else
        ... code for debug mode ...
#endif
```

Release mode, Debug mode 계속

```
#if defined(NDEBUG)   // release mode
#define  CUDA_CHECK_ERROR( )  0
#else     // debug mode
#define  CUDA_CHECK_ERROR( )      do { \
   cudaError_t  e = cudaGetLastError();  \
   if (cudaSuccess != e) { \
      printf("cuda failure \"%s\" at %s:%d\n", \
             cudaGetErrorString(e), __FILE__, __LINE__); \
      exit(1); \
   } \
} while (0)
#endif
```

- **debug mode 에서는**
  - CUDA Error를 check
  - 화면에 친절하게 print
- **__FILE__ : current file name**
- **__LINE__ : current line number**
- **release mode 에서는**
  - 0; → 아무런 일도 없음 (무시)

- **"./common.cpp" 에 추가**
  - 실제로는 if 문 제거,
  - 항상 error check 함 !

error-macro.cu

```
...
// print the result
printf("b = {%f,%f,%f,%f,%f,%f,%f,%f}\n", b[0], b[1], b[2], b[3], b[4], b[5], b[6], b[7]);
// error check
CUDA_CHECK_ERROR();
// done
return 0;
}
```

```
cudaError_t err = cudaGetLastError();
if (cudaSuccess != err) {
  printf("cuda failure \"%s\" at %s:%d\n", cudaGetErrorString(e), __FILE__, __LINE__);
  exit(1);
}
```

~/secure/

```
linux/cuda-work > ./09c-error-macro.exe
a = {0.000000,1.000000,2.000000,3.000000,4.000000,5.000000,6.000000,7.000000}
b = {1.000000,2.000000,3.000000,4.000000,5.000000,6.000000,7.000000,8.000000}
linux/cuda-work >
```

error-detected.cu

```
…
  // print the result
   cudaMemcpy( b, dev_b, SIZE * sizeof(float), cudaMemcpyDeviceToDevice); // ERROR !
   printf("b = {%f,%f,%f,%f,%f,%f,%f,%f}\n", b[0], b[1], b[2], b[3], b[4], b[5], b[6], b[7]);
   // error check
   CUDA_CHECK_ERROR();
   // done
   return 0;
}
```

~/secure/

```
linux/cuda-work > ./09d-error-detected.exe
a = {0.000000,1.000000,2.000000,3.000000,4.000000,5.000000,6.000000,7.000000}
b = {0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000}
cuda failure "invalid argument" at 09d-error-detected.cu:53
linux/cuda-work >
```

내용 contents

- another simple CUDA kernel example
- CUDA error check methods
- cudaError_t
- cudaGetErrorName, cudaGetErrorString
- cudaGetLastError, cudaPeekAtLastError
- CUDA error check macros → CUDA_CHECK_ERROR();

Error Check

에러 체크

폰트 끝단 일치 → 큰 교자 타고 혼례 치른 날

정참판 양반댁 규수 큰 교자 타고 혼례 치른 날

정참판 양반댁 규수 큰 교자 타고 혼례 치른 날

본고딕 Noto Sans KR

The quick brown fox jumps over the lazy dog

The quick brown fox jumps over the lazy dog

The quick brown fox jumps over the lazy dog

Source Sans Pro

Mathematical Notations $O(n \log n)$

Source Serif Pro