

# Introduction to CUDA

## 1. Create and run the vecAdd.cu program.

### 1.1. Use text editor to create the file “vecAdd.cu” with the content below

```
1  #include <stdio.h>
2  #define N 256
3
4  __global__ void vecAdd(int *A, int *B, int *C){
5      int i = threadIdx.x;
6      C[i] = A[i] + B[i];
7  }
8
9  int main(int argc, char *argv[]){
10     int i;
11     int size = N * sizeof(int);
12     int a[N], b[N], c[N], *devA, *devB, *devC;
13
14     for(i=0; i<N; i++){
15         a[i]=1; b[i]=2;
16     }
17
18     cudaMalloc( (void**) &devA, size);
19     cudaMalloc( (void**) &devB, size);
20     cudaMalloc( (void**) &devC, size);
21
22     cudaMemcpy( devA, a, size, cudaMemcpyHostToDevice);
23     cudaMemcpy( devB, b, size, cudaMemcpyHostToDevice);
24
25     vecAdd<<<1, N>>>(devA, devB, devC);
26
27     cudaMemcpy( c, devC, size, cudaMemcpyDeviceToHost);
28     cudaFree(devA);
29     cudaFree(devB);
30     cudaFree(devC);
31
32     for(i=0; i<N; i++){
33         printf("%d ", c[i]);
34     }
35     printf("\n");
36 }
```

### 1.2. Upload the following program to Server at IP = 10.34.110.222 (only from inside Mahidol Campus) and use your username=u5xxxxxx. You may use WinSCP or Filezilla program to transfer your file to the server

### 1.3. Compile the program using the following command (using emulator):

```
nvcc -deviceemu -o vecAdd vecAdd.cu
```

### 1.4. Run the program.

```
./vecAdd
```

- Given an array A of 256 integers, write a CUDA program named vecInc.cu to increase the value of each element in the array A by one. For example, if  $A = \{1, 3, \dots, 509, 511\}$ , then after calling a CUDA kernel, array A becomes  $\{2, 4, \dots, 510, 512\}$ .
- Modify vecInc.cu from Question 2 into vecInc2.cu to work with an array A of any size (e.g. A[1000]) but using only 256 threads.
- Create a MS Word document (u5xxxxxx.docx), and put your source code from step 2. (vecInc.cu) and step 3. (vecInc2.cu), along with the screenshots of their result.