Lab 1

CMPUT 398

Objective

The purpose of this lab is to introduce the student to the CUDA hardware resources and the capabilities of CUDA.

All parts of this lab will be submitted as one zipped file through eclass. Details for submission are at the end of the lab.

Part 1 - C and Memory Allocation Questions

This class will make heavy use of low-level C constructs and concepts, especially pointers and memory management.

As a "warm-up", here are a few quick samples of code and their intended specifications. Each such piece of code is incorrect. Identify what is wrong with the code, and how it should be fixed. Answers to the questions of this part should be written to the file questions.txt (described below)

(Many of these problems allude to common errors encountered while writing both GPU and CPU code.)

Question 1: Creates an integer pointer, sets the value to which it points to 3, adds 2 to this value, and prints said value.

```
void test1(){

    int *a = 3;
    *a = *a + 2;
    printf("%d",*a);
}
```

Question 2: Creates two integer pointers and sets the values to which they point to 2 and 3, respectively.

```
void test2(){
   int* a,b;
   a = (int*) malloc(sizeof(int));
   b = (int*) malloc(sizeof(int));

   if (!(a && b)){
      printf("Out of memory");
      exit(-1);
   }

   *a = 2;
   *b = 3;
}
```

Question 3: Allocates an array of 1000 integers, and for i = 0,...,999, sets the i-th element to i.

```
void test3(){
   int i, *a = (int*) malloc(1000);

   if (!a){
      printf("Out of memory");
      exit(-1);
   }

   for (i = 0; i < 1000; i++)
      *(i+a)=i;
}
```

Question 4: Creates a two-dimensional array of size 3x100, and sets element (1,1) (counting from 0) to 5.

```
void test4(){
   int **a = (int**) malloc(3*sizeof(int*));
   a[1][1] = 5;
}
```

Question 5: Sets the value pointed to by a to an input, checks if the value pointed to by a is 0, and prints a message if it is.

```
void test5(){
   int *a = (int*) malloc(sizeof(int));
   scanf("%d",a);
   if (!a)
      printf("Value is 0\n");
}
```

Part 2 - Parallel Code Question

Given an input array x[n], suppose we have two output arrays y_1[n] and y_2[n], given by the difference equations:

y_1[n] = x[n-1] + x[n] + x[n+1]
y_2[n] = y_2[n-2] + y_2[n-1] + x[n]

Which calculation do you expect will have an easier and faster implementation on the GPU, and why?
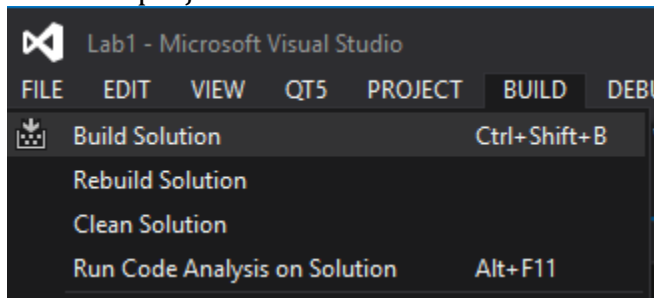
Part 3 - First CUDA Code

Here you will compile and submit your first CUDA project. This part of the assignment does not require you to do any coding, but you will become familiar with Visual Studios and the submission process. You are not expected to understand the details of the code, but should understand the process of compiling and running code that will be used in subsequent modules.
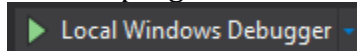
Steps:

1) Download "Lab1.zip" under **Lab 1 files** on e-class.

2) Unzip the file.

3) Open the Visual Studios Solution in Visual Studios 2013.

Lab1	9/2/2016 5:15 PM	File folder	
libwb	9/2/2016 4:03 PM	File folder	
Lab1.sln	9/2/2016 4:06 PM	Microsoft Visual S...	

4) Build the project.

Lab1 - Microsoft Visual Studio						
FILE	EDIT	VIEW	QT5	PROJECT	BUILD	DEBU
Build Solution			Ctrl+Shift+B			
Rebuild Solution						
Clean Solution						
Run Code Analysis on Solution			Alt+F11			

5) Run the program to make sure it works and answer the questions below.

> Local Windows Debugger

Part 4 - First CUDA Code – Questions

Important: Make sure you answer these questions by running the code on the lab machines. If you don't then the answers will be different.

Question 1: What is the device name (GPU)?

Question 2: Suppose you are launching a one-dimensional grid and block. If the hardware's maximum grid dimension is 65535 and the maximum block dimension is 512, what is the maximum number threads can be launched on the GPU?

Question 3: Under what conditions might a programmer choose not want to launch the maximum number of threads?

Question 4: What can limit a program from launching the maximum number of threads on a GPU?

Question 5: What is the maximum global memory size?

Question 6: What is global memory?

Question 7: What is the maximum constant memory size?

Question 8: What is constant memory?

Question 9: What is the maximum shared memory size per block?

Question 10: What is shared memory?

Question 11: What is the maximum block dimensions?

Question 12: What is the maximum grid dimensions?

Question 13: What is the warp size?

Question 14: What does warp size signify on a GPU?
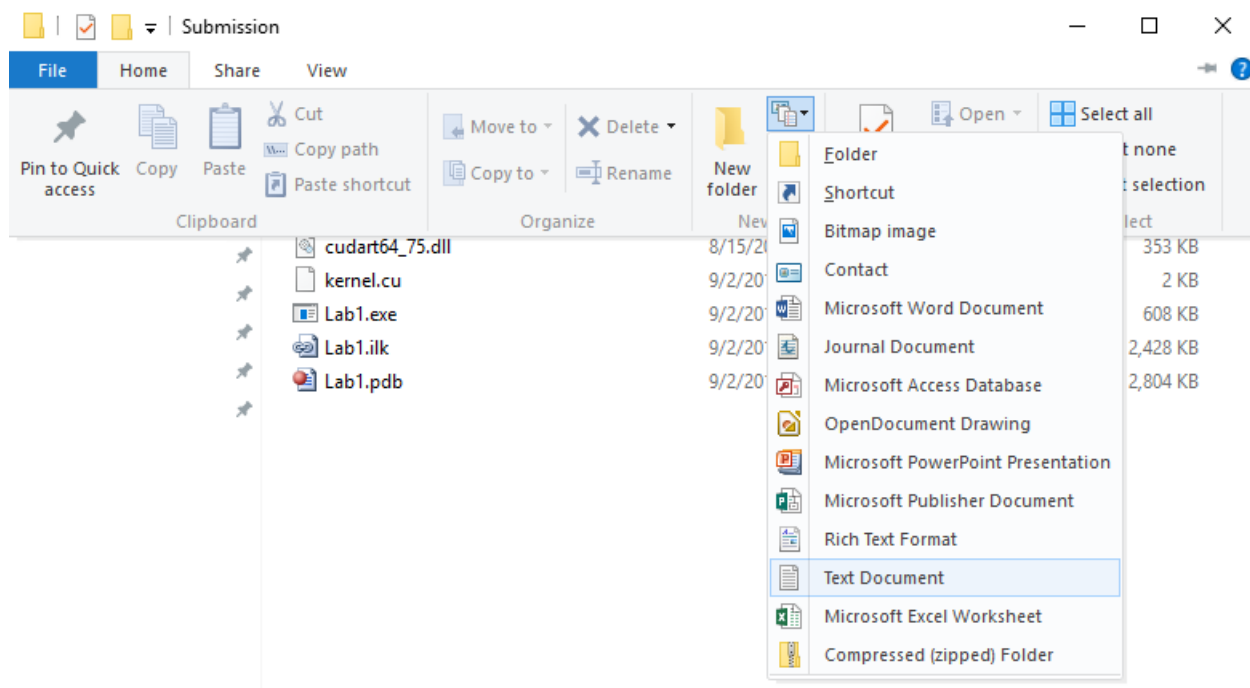
Submission

After you build your "First CUDA Code" you will see a new folder called "Submission" in the project directory.

Lab1	9/2/2016 5:26 PM	File folder	
libwb	9/2/2016 4:03 PM	File folder	
Submission	9/2/2016 5:26 PM	File folder	
Lab1.sdf	9/2/2016 5:27 PM	SQL Server Comp...	
Lab1.sln	9/2/2016 4:06 PM	Microsoft Visual S...	

If you open the folder you will see the following:

cudart64_75.dll	8/15/2015 6:21 PM	Application extens...	353 KB
kernel.cu	9/2/2016 5:28 PM	CU File	2 KB
Lab1.exe	9/2/2016 5:28 PM	Application	608 KB
Lab1.ilk	9/2/2016 5:28 PM	Incremental Linke...	2,428 KB
Lab1.pdb	9/2/2016 5:28 PM	Program Debug D...	2,804 KB

Create a new text file and name it "questions.txt":

Put your answers to the above questions (Parts 1, 2, and 4) in this newly created file. If you already answered your questions some other way, then you can submit the questions as a PDF. Make sure the file structure is as follows:

Submission

- cudart64_75.dll

- kernel.cu

- Lab1.exe

- Lab1.ilk

- Lab1.pdb

- questions.txt (or questions.pdf)

Finally zip the folder "Submission" and upload the zipped folder to eclass under **Lab 1 Submit**.

Mark Breakdown

Note that correct submission will also contribute to the marks given for Part 3.

Part	Breakdown (%)
Part 1	30 %
Part 2	10 %
Part 3/Submission	40 %
Part 4	20 %