Assignment 1

Multiprocessor Programming 521288S

Miika Sikala 2520562, msikala18@student.oulu.fi

Task: Self study and platform Installation

Expected Result: A report on the working development environment with necessary installations for c/c++ programming, OpenCL done on your machine. A 1–2-page report (at max) on a brief description of the necessary tools you have installed for this course.

My system

OS:	Windows 10 Pro, 64-bit (x64-based processor)
CPU:	AMD FX-8520 8-core
GPU:	AMD Radeon RX580 8GB

Platform setup

I had most of the necessary tools installed on my machine already.

For compiling I already had *MinGW gcc* installed, but I will most likely be using the compiler chain that shipped with *Visual Studio*. For IDE, I had Visual Studio 2019 installed with the compiler chain as mentioned. All I had to do was to install and setup the OpenCL SDK.

I downloaded the installer from OCL-SDK repository[1] and ran it. It extracted the SDK headers and libs to the file system and set an environment variable `OCL_ROOT` pointing to that location.

Finally, I created a test project in Visual Studio and configured it in *Project Properties* to use the OpenCL SDK as follows:

- *C/C++* > *General* > *Additional Include Libraries*: `$(OCL_ROOT)\include`
  - Include location of the headers
- *Linker* > *General* > *Additional Library Directories*: `$(OCL_ROOT)\lib\x86`
  - Location of the static libraries for 32-bit system
- *Linker* > *General* > *Additional Library Directories*: `$(OCL_ROOT)\lib\x86_64`
  - Location of the static libraries for 64-bit system
- *Linker* > *Input*> *Additional Dependencies*: `opencl.lib`
  - The static library that was shipped with the SDK (either 32 or 64-bit)

To check if the display driver had OpenCL installed, I checked the installed display drivers for my GPU through Windows Device Manager. Among others, it had `C:\WINDOWS\system32\OpenCL.dll` (32-bit) and `C:\WINDOWS\SysWOW64\OpenCL.dll` (64-bit).

To test the installation, I ran a simple Hello World example that called a simple kernel code and printed the result. After a small version related modification, I was able to run the code.

[1] https://github.com/GPUOpen-LibrariesAndSDKs/OCL-SDK/releases/tag/1.0

Since the SDK is version 2.0, some of the functions of 1.2 are deprecated (e.g., `clCreateCommandQueue`). To use old APIs, we can define `CL_USE_DEPRECATED_OPENCL_2_0_APIS` before including the OpenCL headers.

To run OpenCL on CPU, I had to downgrade my display drivers to AMD Adrenalin 18.7.1, since the support for AMD CPUs was dropped by AMD [1].

I also tested the implementation using gcc as a compiler. We can include the OpenCL libraries using "-l" flag.

Summary

IDE:	Visual Studio 2019 / Sublime Text 4
C/C++:	Visual C++ / gcc
OpenCL SDK:	OCL SDK-Light (OpenCL 2.0)
OpenCL driver:	Latest AMD Radeon display drivers / AMD Adrenalin 18.7.1

References

[1] https://community.amd.com/t5/opencl/missing-opencl-cpu-support-under-windows/td-p/310400

Reporting

It took some time to study the state of AMD CPU drivers.

Task	**Hours**
System setup & testing	3,5h
Writing the report	1h
Total	**4,5h**