



DD2360 Applied GPU Programming 7.5 credits

Tillämpad GPU-programmering

This is a translation of the Swedish, legally binding, course syllabus.

If the course is discontinued, students may request to be examined during the following two academic years

Establishment

Course syllabus for DD2360 valid from Spring 2019

Grading scale

A, B, C, D, E, FX, F

Education cycle

Second cycle

Main field of study

Computer Science and Engineering

Specific prerequisites

Language of instruction

The language of instruction is specified in the course offering information in the course catalogue.

Intended learning outcomes

On completion of the course, students should be able to:

- analyse GPU architecture, assess their advantages and identify potential software optimisations based on knowledge of the GPU architecture,
- design and implement a programme for a GPU for applications in scientific computing, machine learning, image and video processing, computer graphics or for a mobile phone,
- use experimental highly productive methods for GPU programming, such as GPU libraries and algorithmic packages, to speed up the development of large GPU applications,
- use efficient development tools for GPU programming such as debuggers and tools for measuring performance,
- prepare a written report about design development and the implementation of code for a GPU (with application to scientific computing, machine learning, image and video processing, computer graphics or for a mobile phone) and present the report at a seminar orally.

Course contents

The course focuses on three main questions:

- 1. GPU architecture. Computation and memory organisation of different commercial graphics processors will be introduced. A comparison with conventional CPUs and a presentation of new future GPUs will be given.
- 2. GPU programming with CUDA. CUDA's concepts and how to use them to develop applications for GPUs will be introduced through examples from different areas, such as image processing and scientific computing. Also development tools, such as debuggers and tools to measure performance will be presented.
- 3. GPU programming with GPU libraries and algorithmic packages. Packages for high productivity, for example the Thrust library, OpenACC and cuDNN will be presented. Different packages will be explained through examples from different areas of computer science.

The students will obtain access to the GPU cluster Tegner at PDC if they do not have access to a computer with GPU already.

Course literature

A series of articles that present design and implementation of applications for GPU will be published on the course web page. A book that partly covers the course content is "CUDA left Engineers" by D. Storti and M. Yurtoglu.

Examination

- LAB1 - Laboratory, 1.0 credits, grading scale: P, F
- LAB2 - Laboratory, 1.0 credits, grading scale: P, F

- LAB3 - Laboratory, 1.0 credits, grading scale: P, F
- PRO1 - Project, 4.5 credits, grading scale: A, B, C, D, E, FX, F

Based on recommendation from KTH's coordinator for disabilities, the examiner will decide how to adapt an examination for students with documented disability.

The examiner may apply another examination format when re-examining individual students.

Ethical approach

- All members of a group are responsible for the group's work.
- In any assessment, every student shall honestly disclose any help received and sources used.
- In an oral assessment, every student shall be able to present and answer questions about the entire assignment and solution.