1 Course title and credit points
The course is titled Multiprocessor Systems/Multiprocessorsystem and awards 7,5 ECTS credits. One credit point (högskolepoäng) corresponds to one credit point in the European Credit Transfer System (ECTS).

2 Decision and approval
This course is established by Head of Department of Computer Science and Engineering 2014-02-12. The course syllabus was revised by Head of Department of Computer Science and Engineering and applies from 2014-02-12.

3 Objectives
Since many years has parallelism been an approach to obtain high performance in computer systems. However, developing parallel programs is difficult and time consuming, and has therefore only been used in large-scale server systems. Today, that picture has changed with the introduction of multicore processors. Now is almost every computer equipped a small multiprocessor. In order to utilize the performance potential, the programs also need to be parallel. In other words, multiprocessor systems and parallel programming will be fundamental building blocks for contemporary as well as future computer systems. In order to be able to develop high-quality programs for future computers it is essential that the student develop a thorough understanding of different design principles for multiprocessor systems, as well as a thorough understanding of different methods and techniques for developing parallel computer programs.

4 Content
The course covers the following areas
• introduction to multiprocessor systems and parallel programming
• design principles for multiprocessor and parallel computer systems
• design principles for parallel programs
• programming models for parallel programs
• practical training in development of parallel programs

5 Aims and learning outcomes
Knowledge and comprehension
After completion of the course, the student will be able to:
• thoroughly describe the design and working conditions of different types of parallel computer system
• thoroughly describe different programming models for parallel computer systems
• describe the possibilities and problems that are present when developing parallel programs
• thoroughly describe different principles to design parallel programs to achieve high performance
• analyze the performance of parallel programs and parallel computer systems

Skills and abilities
After completion of the course, the student will be able to:
• practically apply different principles and techniques for developing parallel programs
• develop programs for different programming models, i.e., for both shared memory and distributed non-shared memory
• perform simple performance measurements and performance analysis of parallel programs
• perform suitable optimizations to improve the performance of a parallel program

Approach and ability to evaluate
After completion of the course, the student will be able to:
• in writing and orally present and motivate their solutions to project assignments
• independently and critically evaluate and analyze their solutions
6 Generic skills

7 Learning and teaching
The theoretical part of the course is covered and presented on lectures and/or classes. In addition, the students are expected to independently acquire the theoretical knowledge by self studies of relevant literature.

The theoretical knowledge is then applied practically in supervised laboratories and in project assignments that are solved independently or in groups within given time limits.

The teaching language is English.

8 Assessment and grading

<table>
<thead>
<tr>
<th>Code</th>
<th>Module</th>
<th>Credit</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1405</td>
<td>Written examination</td>
<td>3 ECTS</td>
<td>A–F</td>
</tr>
<tr>
<td>1415</td>
<td>Project: Multithreading</td>
<td>1.5 ECTS</td>
<td>G–U</td>
</tr>
<tr>
<td>1435</td>
<td>Project: Message-passing</td>
<td>1.5 ECTS</td>
<td>G–U</td>
</tr>
</tbody>
</table>

Determines the final grade for the course, which will only be issued when all components have been approved.

The course will be graded A Excellent, B Very good, C Good, D Satisfactory, E Sufficient, FX Insufficient, supplementation required, F Fail.

9 Course evaluation

The course coordinator is responsible for systematically gathering feedback from the students in course evaluations and making sure that the results of these feedback back into the development of the course.

10 Prerequisites

The student must have successfully completed a total of 90 ECTS in the subject area Computer Science of Software Engineering, including passed courses in Programming, 15 ECTS, Algorithms and data structures, 6 ECTS, Computer organization, 6 ECTS, and Operating/Real-time systems, 6 ECTS, or similar.

11 Field of education and subject area

The course is part of the field of education and is included in the subject area Computer Science and the subject area Software Engineering. The course can also be included in the subject area Software Engineering.

12 Restrictions regarding degree

The course cannot form part of a degree with another course, the content of which completely or partly corresponds with the contents of this course.

13 Additional information

Replaces DV2415 and DV2528.

14 Course literature and other teaching material

Main literature


Reference literature


