LUNDS UNIVERSITET
Lunds Tekniska Högskola

Course syllabus

Multicoreprogrammering
Multicore Programming

EDAN26, 7,5 credits, A (Second Cycle)

Valid for: 2023/24
Faculty: Faculty of Engineering, LTH
Decided by: PLED C/D
Date of Decision: 2023-04-18

General Information

Elective for: C4-pv, D4-is, D4-pv, E4-pv, F4, F4-pv, Pi4-pv
Language of instruction: The course will be given in Swedish

Aim

The purpose of the course is that to learn how to implement an algorithm as efficiently as possible on a particular multicore computer with a focus on multiprocessors with shared memory. Another purpose of the course is to gain insights into advantages and disadvantages of different programming languages for multicore programming, such as Java, Scala, C, and OpenMP. The students will also learn about lock-free algorithms and data structures.

Learning outcomes

Knowledge and understanding
For a passing grade the student must

- understand how cache memories work on multicores
- understand the importance of reducing the number of cache misses
- understand the importance of load-balancing between different processors
- understand how a sequential C program can be parallelised while taking the above into consideration
- understand advantages and disadvantages of these two multicore architectures with respect to (1) performance and (2) how complicated the programming becomes
- understand advantages and disadvantages of different programming languages
- understand when lock-free data structures are suitable

Competences and skills
For a passing grade the student must

- be able to use Pthreads to create threads and synchronise these on a multiprocessor
- be able to use OpenMP to parallelise compute intensive sequential C programs
- be able to improve the performance of a given sequential program by parallelising it in a way which maximises its performance.
- be able to decide on programming language for a given problem

Judgement and approach
For a passing grade the student must

- be able to implement a suitable version of an algorithm while taking into account both the demands of the application and the multicore architecture.
- be able to judge which multicore architecture is suitable for a given algorithm.
- be able to judge which programming language is suitable for a given algorithm.

Contents

Multiprocessors, classification of cache misses, decomposition of the computation into tasks. Assignment of tasks to threads. Mapping of threads to processors. Programming models, synchronisation, and communication. Owner computes rule. Memory consistency models. Sequential consistency. Weak ordering. Release consistency, Pthreads, OpenMP, current research trends in multiprocessor computer architecture.

Examination details

Grading scale: TH - (U,3,4,5) - (Fail, Three, Four, Five)
Assessment: Oral exam. To pass the course, the exam, laboratory exercises, and a programming assignment must be passed. The final grade of the course is based om the result of the oral exam.

The examiner, in consultation with Disability Support Services, may deviate from the regular form of examination in order to provide a permanently disabled student with a form of examination equivalent to that of a student without a disability.

Parts
Code: 0122. **Name:** Compulsory Course Items.
Credits: 3,5. **Grading scale:** UG. **Assessment:** Approved compulsory itemns. **Contents:** Laboratory work and a project.
Code: 0222. **Name:** Examination.
Credits: 4. **Grading scale:** TH. **Assessment:** Approved examination. **Contents:** Oral examination.

Admission

Admission requirements:

- EDAA01 Programming - Second Course or EDAA30 Programming in Java - Second Course

The number of participants is limited to: No
The course overlaps following course/s: EDA116, EDAN25

Reading list

- Jonas Skeppstedt and Christian Söderberg: Writing Efficient C Code: A Thorough Introduction, 3rd edition. Skeppberg, 2019, ISBN: 9781723831157.

Contact and other information

Course coordinator: Jonas Skeppstedt, jonas.skeppstedt@cs.lth.se
Course homepage: http://cs.lth.se