Linnæus University

Dnr: 2022/2719-3.1.2.2

Course syllabus

Faculty of Technology

Department of Computer Science and Media Technology

1DT909 Parallellprogrammering, 5 högskolepoäng
1DT909 Parallel programming, 5 credits

Main field of study
Computer Engineering

Subject Group
Computer Science

Level of classification
First Level

Progression
G1F

Date of Ratification
Approved by Faculty of Technology 2022-06-27
The course syllabus is valid from spring semester 2023

Prerequisites
Databases and data modeling (1DT903), 5 credits and Object-oriented programming (1DT904, 5 credits or 1DT905, 7,5 credits) or equivalent.

Objectives

After completing the course the student shall be able to:
Knowledge and understanding

- A.1 Explain problems that can occur when using shared resources such as locks, starvation and race conditions,
- A.2 explain some common methods for handling locks (e.g., semaphores), including their properties and limitations,
- A.3 explain different consistency models,
- A.4 reason about various properties of concurrent programs, such as accuracy and termination,
- A.5 explain the difference between shared memory and message passing, and
- A.6 explain the difference between parallelism, concurrency, and asynchronous execution.

Competence and skills

- B.1 Implement concurrent applications in Java and Python for a computer with shared memory, using threads and locking or asynchronous execution,
- B.2 implement lock-free data structures in Java,
- B.3 prove that (simple) concurrent programs are correct, and
- B.4 analyze a problem and implement a suitable concurrent solution that is correctly synchronized.

Judgement and approach

- C.1 Choosing and reasoning about whether shared memory, message passing, or asynchronous execution is appropriate for a given problem.

Content

The course introduces concurrent programming and the problems this entails, e.g., locks and race conditions. Different ways to deal with these problems, e.g., locking algorithms and message handling, are discussed and their limitations. The content and algorithms are exemplified using threads in Java and Python and with asynchronous execution in Python.
The following topics are covered:

- Processes and synchronization
- Scheduling
- Shared memory and messages
- Concurrent programming with threads and shared variables
- Critical sections
- Locks, barriers, semaphores, and monitors
- Asynchronous programming
- Distributed / concurrent algorithms
- Consistency models
- Continuous and lock-free data structures

Type of Instruction

The instruction consists of lectures and teacher-led laboratory work. The programming labs are carried out in pairs.

Examination

The examination of the course is divided as follows:

Code	Designation	Grade	Credits
2301	Written exam	AF	3,00
2302	Programming assignment	AF	2,00

The course is assessed with the grades A, B, C, D, E, Fx or F.

The grade A constitutes the highest grade level, the remaining grades follow in descending order where the grade E constitutes the lowest grade level for passing. The grade F means that the student's performance has been assessed as failed.
Assessment of student performance is made through programming assignments and a written exam.Renewed examination is given in accordance with Local rules for course and examination at undergraduate and advanced level at Linnaeus University.
To pass the course, grade E or higher is required for all parts. The final grade is decided

from: written exam (60%) and programming assignments (40%).
If the university decides that a student is entitled to special educational support due to a disability, the examiner has the right to give an adapted test or that the student completes the test in an alternative way.

Objectives achievement

The examination elements are linked to the course objectives in the following ways:

Goal	2301	2302
A.1	☑	
A.2	☑	☑
A.3	☑	
A.4	☑	☑
A.5	☑	
A.6	☑	
B.1		☑
B.2		☑
B.3	☑	
B.4	☑	☑
C.1		☑

Course Evaluation

During the implementation of the course or in close conjunction with the course, a course evaluation is to be carried out. Results and analysis of the course evaluation are to be promptly presented as feedback to the students who have completed the course. Students who participate during the next course instance receive feedback at the start of the course. The course evaluation is to be carried out anonymously.

Credit Overlap

The course cannot be included in a degree along with the following course/courses of which the content fully, or partly, corresponds to the content of this course: 1DT906 Parallel programming

Other

Grade criteria for the AF scale are communicated to the student through a special document. The student is to be informed about the grade criteria for the course by the start of the course at the latest. The course is conducted in such a way that the course participants' experiences and knowledge are made visible and developed. This means, for example, that we have an inclusive approach and strive for no one to feel excluded. This can be expressed in different ways in a course, for example by using the gender-neutral example.

Required Reading and Additional Study Material

Required reading:

- Herlihy, Maurice. och Shavit, Nir., *The Art of Multiprocessor Programming*, Morgan Kaufmann, latest edition. Pages: 400 of 536.