

CS-302

Parallelism and concurrency in software

Cursus	Sem.	Type
Communication systems	BA6	Opt.
Computer science	BA6	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	6 weekly
Lecture	3 weekly
Exercises	3 weekly
Number of positions	

Remark

Pas donné en 2023-24

Summary

From sensors, to smart phones, to the world's largest datacenters and supercomputers, parallelism & concurrency is ubiquitous in modern computing. There are also many forms of parallel & concurrent execution in modern platforms with varying degrees of ease of programmability, performance & efficiency.

Content

The goal of this course is to provide a deep understanding of the fundamental principles and trade-offs involved in constructing efficient parallel or concurrent software.

- Performance vs. efficiency
- Forms of parallelism
- Communication models
- Memory models
- Functional parallelism
- Domain-specific languages
- Throughput parallelism
- Data parallelism
- Distributed data parallelism
- Forms of concurrency
- Asynchronous programming
- Coroutines and futures

Learning Prerequisites**Required courses**

- CS-200 Computer architecture
- CS-214 Software construction

Recommended courses

CS-202 Computer systems

Important concepts to start the course

- Programming in C/C++, Java or Scala
- Basic assembly language programming
- Basic use of tools to debug

Learning Outcomes

By the end of the course, the student must be able to:

- Construct parallel software
- Construct concurrent software
- Construct efficient software
- Design software for various platforms including CPUs, accelerators and clusters

Teaching methods

- Lectures
- Homework
- Projects

Expected student activities

- Standalone homeworks
- Projects in teams

Assessment methods

- 20% homework
- 30% projects
- 20% midterm
- 30% final

Supervision

Office hours	Yes
Assistants	Yes

Prerequisite for

- CS-471 Advanced multiprocessor architecture
- CS-453 Concurrent computing
- CS-451 Distributed algorithms