CHALMERS

UNIVERSITY OF
GOTHENBURG

Practical information

Lecture 0 of TDA384/DIT391
Principles of Concurrent Programming

Sandro Stucki

Chalmers University of Technology | University of Gothenburg

SP3 2018/2019

Based on course slides by Carlo A. Furia

Canvas room and course website

Make sure to regularly check the Canvas room and course website:

Canvas	announcements, discussion forum.
CTH login	`https://chalmers.instructure.com/courses/3777/`
GU login	`https://gu.instructure.com/courses/12523~3777/`
Website	lectures, labs, exams, . . . `http://www.cse.chalmers.se/edu/course/TDA384_LP3/`

These are the primary sources of information about the course.

Discussion forum

Use the Canvas discussion forum for questions and discussions of general interest to the course:

`https://chalmers.instructure.com/courses/3777/discussion_topics`

The forum URL is of course linked from the course website.

Discussion forum

Use the Canvas discussion forum for questions and discussions of general interest to the course:

`https://chalmers.instructure.com/courses/3777/discussion_topics`

The forum URL is of course linked from the course website.

!!! Do not share solutions to labs on Canvas (or anywhere else) !!!

The teaching team

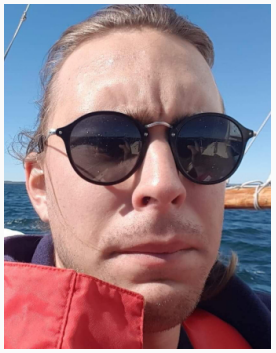

Course responsibles

Lecturer Sandro Stucki

Examiner K.V.S. Prasad


Teaching assistants (TAs)

- Alexander Sjösten
- Andreas Lööw
- Iulia Bastys
- Maximilian Algehed

If you have questions...

1. ask them during the lectures and lab sessions,
2. post them in the discussion forum on Canvas,
3. send an email to the teacher or TAs,
4. book an appointment with the teacher or TAs (by email).

Protip: options 1 & 2 are quicker than options 3 & 4.

Student representatives

Chalmers student representatives

- Alexander Arvidsson, alearv@student.chalmers.se
- Anton Berneving, antbern@student.chalmers.se
- Pontus Lindblom, ponlind@student.chalmers.se
- Adam Oliv, adamoli@student.chalmers.se
- Jacob Thorselius Pedersen, jacped@student.chalmers.se

GU student representatives

- Jan Jürgen Eisenmenger, guseisja@student.gu.se
- Erik Klingberg, gusklier@student.gu.se
- Jens Madsen, gusmadje@student.gu.se
- Gustav Pihlquist, guspihgu@student.gu.se
- Nuha Batool Taqi, gustaqnu@student.gu.se

Main learning goals

By the end of the course you should be able to

- understand the problems common to concurrent and parallel systems,
- demonstrate techniques and patterns to reason about and write correct and efficient concurrent programs,
- apply those techniques and patterns in modern programming languages.

Overview of the course

- Introduction to concurrency.
- Part 1. Classic, shared-memory concurrency in Java:
  - java threads,
  - locks, semaphores, and monitors.
- Part 2. Message-passing concurrency:
  - Erlang and the actor model.
- Part 3. Parallelizing computations:
  - fork/join parallelism,
  - lock-free programming.

Lectures

1	Introduction to concurrent programming	21 January 2019
2	Races, locks, and semaphores	21 January 2019
3	Models of concurrency & synchronization algorithms	23 January 2019
4	Synchronization problems with semaphores	28 January 2019
5	Monitors	30 January 2019
6	Introduction to functional programming in Erlang	4 February 2019
	Erlang Tutorial	4 February 2019
7	Message-passing concurrency in Erlang	11 February 2019
8	Synchronization problems with message passing	13 February 2019
9	Parallelizing computations	18 February 2019
10	Parallel linked lists	20 February 2019
11	Lock free programming	25 February 2019
	Guest lecture: Niklas Gustavsson, Spotify	27 February 2019
12	Models and languages of concurrent computation	4 March 2019
13	Verification of concurrent programs & recap	6 March 2019
14	Further examples and proofs	11 March 2019

Make sure to check the up-to-date schedule on the website.

Labs

There will be three labs – one for each part of the course.

1. Trainspotting (Java)
2. CCHAT (Erlang)
3. A-mazed (Java)

Descriptions of the labs, deadlines, and rules are on the website.

- Register your group (2 students) in Fire.
- Make sure to check the lab/room schedule on the website.

Labs

There will be three labs – one for each part of the course.

1. Trainspotting (Java)
2. CCHAT (Erlang)
3. A-mazed (Java)

Descriptions of the labs, deadlines, and rules are on the website.

- Register your group (2 students) in Fire.
- Make sure to check the lab/room schedule on the website.

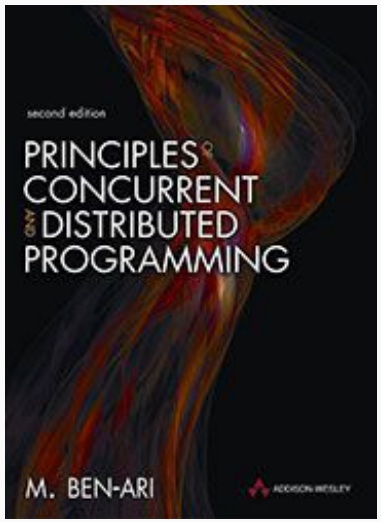
!!! Do not share solutions to labs on Canvas (or anywhere else) !!!

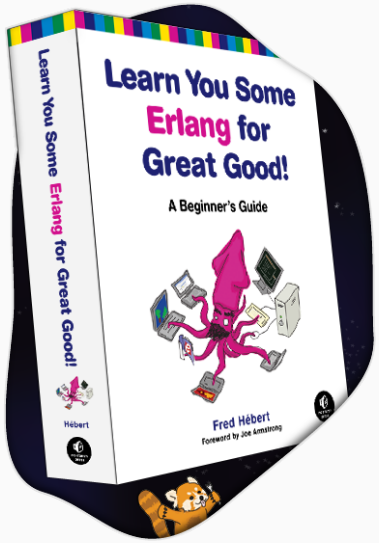
Slides and reading material

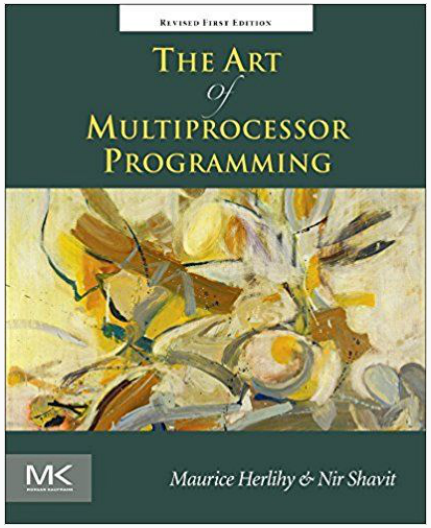
Lecture slides: on the website.

Books:

- Ben-Ari: *Principles of concurrent and distributed programming*, 2nd edition.
- Hébert: *Learn you some Erlang for great good* (free online),
- Herlihy & Shavit: *The art of multiprocessor programming*







Exam

- Open-book exam:
  - max. 2 textbooks,
  - max. 4 two-sided A4 sheets of notes (printed or handwritten),
  - an English dictionary.
- All topics in the lectures can be examined (except the guest lecture).
- See exams of previous years for examples (on the website).
- Exam dates:
  - Tuesday, 19 March, 8:30–12:30,
  - Wednesday, 21 August, 14:00–18:00 (re-exam),
  - check the website for updates!
- Exam grading: see the course website.

Computing resources

- Install Java and Erlang/OTP on your computers.
- Try out the examples presented in class; the complete examples will be available on the website for each lecture.
- Lab 1 (Trainspotting) requires a simulator, which runs on the lab computers (Unix/Linux workstations).
- See the course website for instructions on how to
  - use the lab computers, and
  - set up Java & Erlang/OTP on your own computers.

These slides' license