Practical Information

Lecture 0 of TDA384/DIT391

Principles of Concurrent Programming

Nir Piterman

Chalmers University of Technology | University of Gothenburg

SP1 2020/2021

UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Canvas Room and Course Website

Make sure to regularly check the Canvas Room and Course Website:

Canvas	Announcements, discussion forum, videos
CTH login	https://canvas.chalmers.se/courses/10300
GU login	https://canvas.gu.se/courses/12523~10300
Website	Lectures, labs, exams, … http://www.cse.chalmers.se/edu/course/TDA384_LP1

These should be your 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://canvas.chalmers.se/courses/10300/discussion_topics

https://canvas.gu.se/courses/12523~10300/discussion_topics

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

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

Covid 19 – The course is running virtually!

- Lectures:
  - Lectures are running on zoom:
  - https://gu-se.zoom.us/j/69697080492?pwd=LzRIK09xSU5oQ0N0Z2liRW14bmVPQT09
  - Meeting ID: 69697080492 Password: 213769
  - Well done! You are here!
  - Lectures are recorded:
    - Student video is not captured.
    - If you ask a question your voice will be captured.
    - Videos will be shared through closed Canvas access.
    - After course: either delete or remove all student information.

Covid 19 – The course is running virtually!

- Labs:
  - There are two forums at the top of the forum page:

    CTH login https://canvas.chalmers.se/courses/10300/discussion_topics

    GU login https://canvas.gu.se/courses/12523~10300/discussion_topics

    - Lab assistance requests – during a lab, post a link to your zoom meeting
    - Demo requests – post a request for a demo (including a zoom meeting link), TA will allocate time

Zoom Etiquette

- Login with your Chalmers/GU account (use SSO login).
- Set your profile's photo.
- Mute your microphone!
- If you can't hear me, write on the chat. Unmute only as last resort.
- If you have a question, raise your hand (virtually!). Ask questions on the chat.
- It is hard to present and manage the meeting at the same time!

The teaching team

Lecturer/Examiner

- Nir Piterman

Teaching assistants (TAs)

- Matthías Páll Gissurarson
- Max Algehed
- Agustin Mista

If you have questions

- Ask them during lectures and lab sessions.
- Post them on the discussion forum.
- Send an email to pcp-teachers@lists.chalmers.se
- Book an appointment with the teacher or TAs (by email).

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

Student Representatives

Chalmers

- Mojtaba Ataie mojtaba.ataie99@gmail.com
- Loke Gustafsson esnuid+ladok@protonmail.com
- Sebastian Holm SebastianHoolm@hotmail.com
- Samuel Ivarsson samuelvalfridivarsson@gmail.com
- You?

GU

- You?

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

Class #	Date	Topic
1	Mon, 31 August 2020	Introduction to concurrent programming
2	Mon, 31 Aug 2020	Races, locks, and semaphores
3	Fri, 4 Sep 2020	Models of Concurrency and Synchronization Algorithms
4	Mon, 7 Sep 2020	Synchronization Problems with Semaphores
5	Thu, 10 Sep 2020	Monitors
6	Fri, 11 Sep 2020	Introduction to Functional Programming in Erlang
7	Mon, 14 Sep 2020	Message-passing concurrency in Erlang
8	Fri, 18 Sep 2020	Concurrency in weak memory models (by Andreas Lööw)
9	Mon, 21 Sep 2020	Synchronization problems with message passing
10	Fri, 25 Sep 2020	Parallelizing computations
11	Mon, 28 Sep 2020 or Fri, 2 Oct 2020	Functional Programming^{XP}: The Industrial Experience (by Karol Ostrovsky)
12	Mon, 28 Spe 2020 or Fri, 2 Oct 2020	Parallel Linked Lists
13	Mon, 5 Oct 2020	Lock-free programming
14	Mon, 12 Oct 2020	Models and Languages of Concurrent computation
15	Fri, 16 Oct 2020	Verification of Concurrent Programs
16	Mon, 19 Oct 2020	Formal reasoning, Examples, and Proofs, Exam prep
17	Fri, 23 Oct 2020	Erlang and Java Tutorials (to be moved earlier!)

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

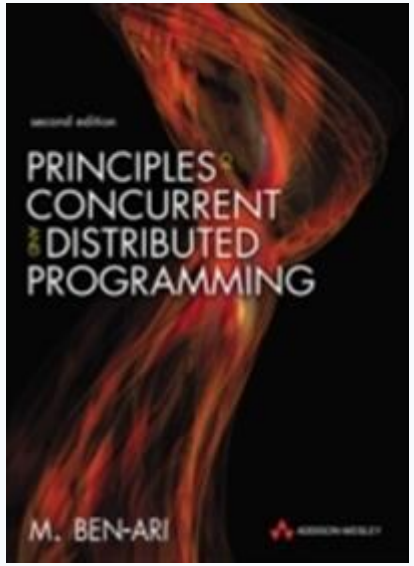
Do not share solutions to labs on Canvas (or anywhere else) !!!

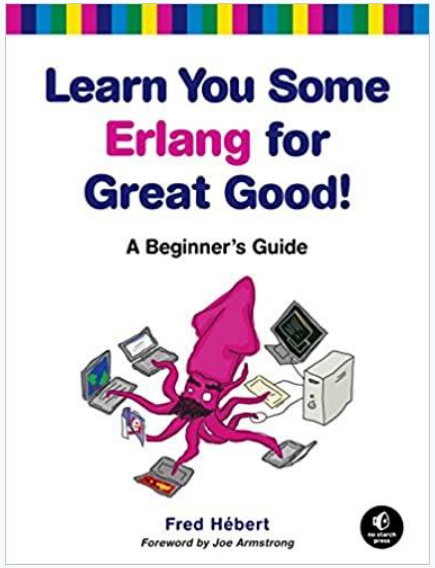
Slides and Reading Material

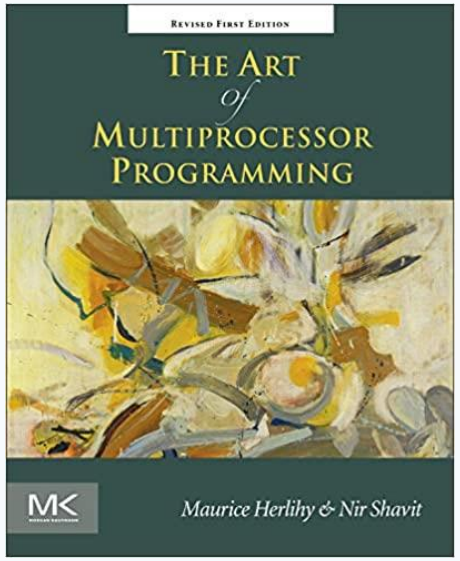
Lecture slides: will be on the website.

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

It is currently not known whether exams will be held online / on campus.

- 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:
  - 24 October 2020
  - 16 March 2021 (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.

There's a lab on Thursday – What's the point of that?

- Setup the train system!
- Start playing with it with sequential programs:
  - Have only one train.
  - Start and stop.
  - Check distances and speeds.
  - When is a train on a switch?
  - Make plans.

Erlang, Erlang, Erlang, …

- Start early!
- Install the Erlang environment.
- Start the online tutorial.
- Especially if never done functional programming before.