Practical Information

Lecture 0 of TDA384/DIT391

Principles of Concurrent Programming

Gerardo Schneider

Chalmers University of Technology | University of Gothenburg

LP3 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/12262
GU login	https://canvas.gu.se/courses/12523~12262
Website	Lectures, labs, exams, … http://www.cse.chalmers.se/edu/course/TDA384_LP3/

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/12262/discussion_topics

https://canvas.gu.se/courses/12523~12262/announcements

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/68276113749?pwd=Zkh6MkVVZjAyZ2haWWk2QUVDU2Ztdz09
  - Meeting ID: 682 7611 3749; Passcode: 357491
  - Well done! You are here!
  - Lectures are recorded:
    - Some lectures might be pre-recorded (TBD)
    - Student video is not captured
    - If you ask a question your voice will be captured
      - Please do not unmute your mic; write in the chat if needed be
    - Videos will be shared through closed Canvas access
    - After the course: all student information will be deteled (chat conversations will be deleted or anonymized)
    - If you have any privacy concern, please contact me via email

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/12262/discussion_topics

    GU login https://canvas.gu.se/courses/12523~12262/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 (unless you prefer not to for privacy reasons)
- Mute your microphone!
- If you can’t hear me, write on the chat
  - **Unmute only as last resort** (remember your voice will be recorded)
- If you have a question, raise your hand (virtually!)
  - Ask questions on the chat
  - I might not answer the question immediately but take a look during the break or at the end of the lecture (to avoid interrupting the flow and recording)
- It is hard to present and manage the meeting at the same time!

The teaching team

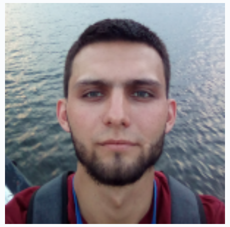
Lecturer/Examiner

- Gerardo Schneider

Teaching assistants (TAs)

- William Hughes

- Ivan Oleynikov



- Abhiroop Sarkar

- Carlos Tomé Cortiñas

If you have questions

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

Options 1 & 2 are quicker than options 3 & 4

Student Representatives

Chalmers

- Rakel Hellberg: rakelh@student.chalmers.se (TKTEM)
- Albin Karlsson: riddle.agk@gmail.com (TKDAT)
- Filip Lindset: filiplindset@hotmail.com (TKDAT)
- Carl Ridderstolpe: carl.ridderstolpe@hotmail.com (TKDAT)
- Adrian Eliasson: adreli@student.chalmers.se

GU

- Ryan Janson: gusjanry@student.gu.se

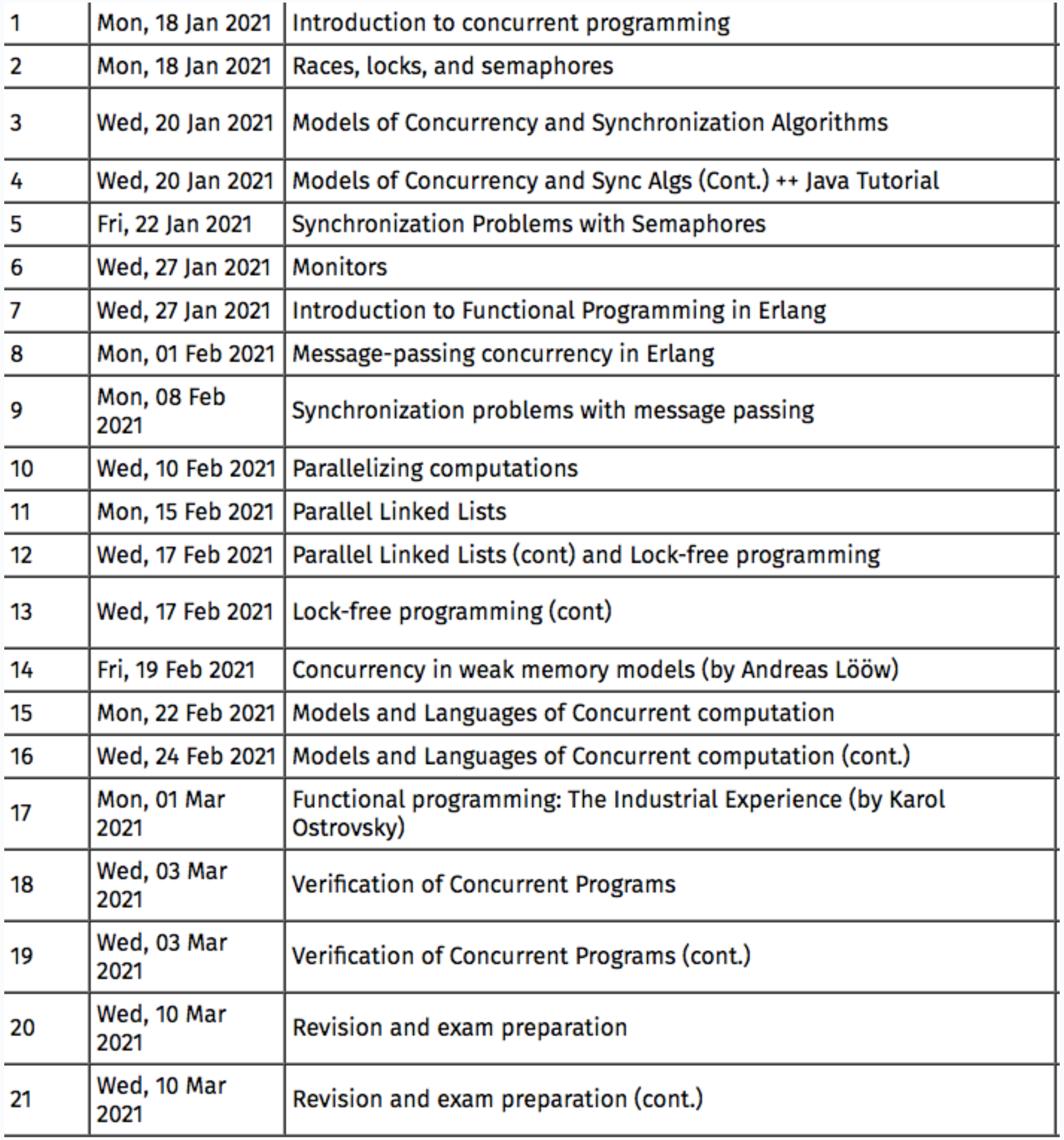
Main Learning Outcomes

- 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	Mon, 18 Jan 2021	Introduction to concurrent programming
2	Mon, 18 Jan 2021	Races, locks, and semaphores
3	Wed, 20 Jan 2021	Models of Concurrency and Synchronization Algorithms
4	Wed, 20 Jan 2021	Models of Concurrency and Sync Algs (Cont.) ++ Java Tutorial
5	Fri, 22 Jan 2021	Synchronization Problems with Semaphores
6	Wed, 27 Jan 2021	Monitors
7	Wed, 27 Jan 2021	Introduction to Functional Programming in Erlang
8	Mon, 01 Feb 2021	Message-passing concurrency in Erlang
9	Mon, 08 Feb 2021	Synchronization problems with message passing
10	Wed, 10 Feb 2021	Parallelizing computations
11	Mon, 15 Feb 2021	Parallel Linked Lists
12	Wed, 17 Feb 2021	Parallel Linked Lists (cont) and Lock-free programming
13	Wed, 17 Feb 2021	Lock-free programming (cont)
14	Fri, 19 Feb 2021	Concurrency in weak memory models (by Andreas Lööw)
15	Mon, 22 Feb 2021	Models and Languages of Concurrent computation
16	Wed, 24 Feb 2021	Models and Languages of Concurrent computation (cont.)
17	Mon, 01 Mar 2021	Functional programming: The Industrial Experience (by Karol Ostrovsky)
18	Wed, 03 Mar 2021	Verification of Concurrent Programs
19	Wed, 03 Mar 2021	Verification of Concurrent Programs (cont.)
20	Wed, 10 Mar 2021	Revision and exam preparation
21	Wed, 10 Mar 2021	Revision and exam preparation (cont.)

- 13 lectures + 2 guest lectures + 1-2 revision lectures
  - GL1: Andreas Lööw (Chalmers) on Weak Memory Models
  - GL2: Karol Ostrovsky (Ericsson) on Industrial Experience in using Func. Prog.
- Subject to change depending on the interaction with students, etc
- Some lectures might take less/ more time -> the schema might then be rescheduled

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

Labs

There will be one preparation lab and three “real” 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

- **Lab 0**: Set up and register your group (2 students) in Fire
- Make sure to check the lab/room schedule on the website.

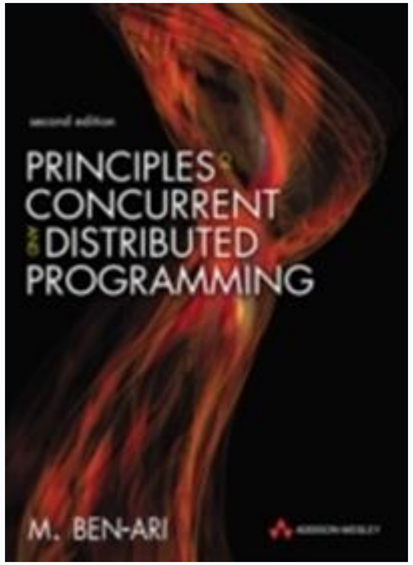
Do not share solutions to labs on Canvas (or anywhere else) !!!

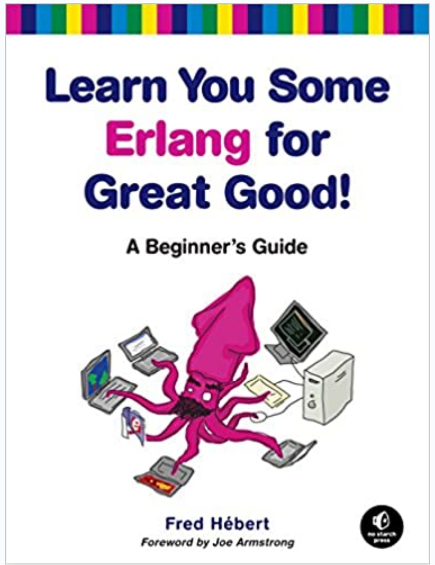
Slides and Reading Material

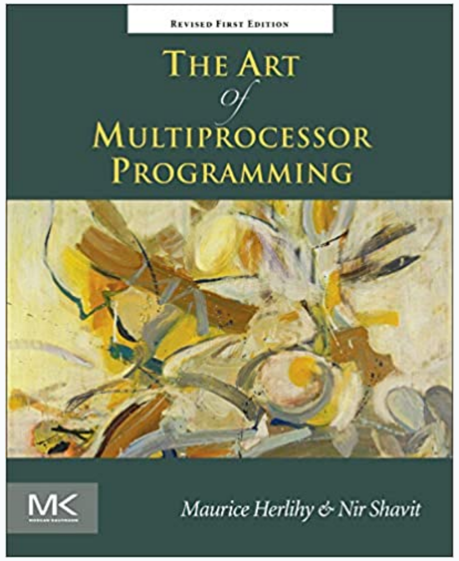
Lecture slides: will be on the website

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




Learn You Some Erlang for Great Good!
A Beginner's Guide
Fred Hébert
Foreword by Joe Armstrong
no starch press


Revised First Edition
The Art of Multiprocessor Programming
MK
Maurice Herlihy & Nir Shavit

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:
  - 16 March 2021
  - 9 June 2021 (re-exam)
- Check the website for updates!
- Exam grading: see the course website

Computing Resources

- Install Java and Erlang/OTP on your computers (**Lab 0**)
- 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 are two lab sessions this week (Wed and Fri) – What’s the point of that?

- LAB 0: Setup the system, create your group, etc.

- Deadline to have the setup and the small examples is **Jan 25**

- See description and more details in the webpage.

Erlang, Erlang, Erlang, …

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