what is offered here?

Fundamentals & Overview
as well as perspectives, paths, methods, implementations, and open questions

Concurrent & Distributed Systems

Organization & Contents

who could be interested in this?
anybody who ...
... wants to work with real-world scale computer systems
... would like to learn how to analyse and design operational and robust systems
... would like to understand more about the existing trade-off between theory, the real-world, traditions, and pragmatism in computer science
... would like to understand why concurrent systems are an essential basis for most contemporary devices and systems
who are these people? – introductions

This course will be given by

Uwe R. Zimmer & Alistair Rendell

Your individual tutors are

Abigail Thomas, Alex Smith, Ian Mallett,
Michael Bennett, Robin Monro, Yaya Lu, Zara Kay

Text book for the course

[Ben-Ari06]
M. Ben-Ari
Principles of Concurrent and Distributed Programming

Many algorithms and concepts for the course are in there – but not all!

References for specific aspects of the course are provided during the course and are found on our web-site.

how will this all be done?

Lectures:
- 2 x 1.5 hours lectures per week ... all the nice stuff
  Tuesday & Thursday 16:30 (both in R.N. Robertson - which is: here)

Laboratories:
- 2 hours per week ... all the rough and action stuff
  time slots: on our web-site
  enrolment: https://cs.anu.edu.au/streams/ (open since last Friday)

Resources:
- Introduced in the lectures and collected on the course page:
  https://cs.anu.edu.au/courses/comp2310/ ... as well as schedules, slides,
  sources, links to forums, etc. pp. ... keep an eye on this page!

Assessment (for discussion):
- Exam at the end of the course (50%)
  plus one hurdle lab in week 4 (5%)
  plus two assignments (15% + 15%)
  plus one mid-semester exam (15%)

Topics

Language refresher [3]
1. Concurrency [3]
5. Data Parallelism [1]
7. Safety and liveness [2]
8. Distributed systems [4]
9. Architectures [1]
Organization & Contents

Topics

1. Concurrency [3]
   1.1. Forms of concurrency [1]
   - Coupled dynamical systems
   1.2. Models and terminology [1]
   - Abstractions
   - Interleaving
   - Atomicity
   - Proofs in concurrent and distributed systems
   1.3. Processes & threads [1]
   - Basic definitions
   - Process states
   - Implementations


5. Scheduling [2]


7. Architectures for CDS [1]

8. Distributed systems [7]

 Topics

1. Concurrency [3]


5. Scheduling [2]


7. Architectures for CDS [1]

8. Distributed systems [7]

 Topics

1. Concurrency [3]


5. Scheduling [2]


7. Architectures for CDS [1]

8. Distributed systems [7]

 Topics

1. Concurrency [3]


5. Scheduling [2]


7. Architectures for CDS [1]

8. Distributed systems [7]
### Topics

1. Concurrency [3]
5. Scheduling [2]
7. Architectures for CDS [1]
8. Distributed systems [7]

#### 7. Architectures for CDS [1]

- Essential time-independent safety properties
- Forms of livelocks
- Classification of fairness
- Detection
- Avoidance
- Prevention (& recovery)
- Definitions

#### 8. Distributed systems [7]

- Networks [1]
  - OSI model
  - Network implementations
- Global times [1]
  - Synchronized clocks
  - Logical clocks
- Distributed states [1]
  - Consistency
  - Snapshots
  - Termination
- Forms of distribution/redundancy [1]
  - Computation
  - Memory
  - Operations
- Transactions [2]
24 Lectures

1. Concurrency (3)
   1.1. Forms of concurrency (1)
   1.2. Models and terminology (1)
   1.3. Processes & threads (1)

2. Mutual exclusion (2)
   2.1. by shared variables (1)
   2.2. by test-and-set hardware support (0.5)
   2.3. by semaphores (0.5)

3. Condition synchronization (4)
   3.1. Shared memory synchronization (2)
   3.2. Message passing (2)

4. Non-determinism in concurrent systems (2)
   4.1. Correctness under non-determinism (1)
   4.2. Select statements (1)

5. Scheduling (2)
   5.1. Problem statement and design space (1)
   5.2. Basic scheduling methods (1)

6. Safety and liveness (3)

7. Architectures for CDS (1)

8. Distributed systems (7)

Laboratories & Assignments

Laboratories

1. Concurrency language support basics (in Ada) (3)
   1.1. Structured, strongly typed programming (3)
   1.2. Generic, re-usable programming (3)
   1.3. Concurrent processes (3)

Assignments

1. Concurrent programming (15%)
   1. Concurrent programming (15%)
   2. Concurrent programming in multi-core systems (15%)

Examinations

1. Mid-term check (10%)
   2. Final exam (55%)

Marking

The final mark is determined by
1. Assignments 
2. Examinations 
3. Mid-term check 

Laboratories

1. Concurrency language support basics (in Ada) (3)
   1. Forms of concurrency (3)
   2. Models and terminology (3)
   3. Processes & threads (3)

Assignments

1. Concurrency programming (15%)
   1. Mutual exclusion (15%)
   2. Mutual exclusion (15%)
   3. Concurrent programming in multi-core systems (15%)

Examinations

1. Mid-term check (10%)
   2. Final exam (55%)

Marking

The final mark is determined by
1. Assignments 
2. Examinations 
3. Mid-term check