# Course Outline

# COMP9243
# Distributed Systems
## Semester 1 2014

## Contents


- Course Staff
- Course Details
- Course Aims
- Student Learning Outcomes
- Related Courses and Programs
- Course Philospohy and Teaching Strategies
- Course Schedule
- Assessment
- Resources for Students
- Academic Honesty and Plagiarism
- Course Evaluation and Development
- Further Information


## Course Staff

**Contact Details**

| Staff Name | Role | Email | Phone |
|---|---|---|---|
| Ihor Kuz | Course Convenor | ikuz at cse.unsw.edu.au | 0283060582 |

**Consultations**

| Staff Name | Day/Time | Location |
|---|---|---|
| Ihor Kuz | Monday 1500-1600 | L5 (NICTA) - Level 4 E423 |

## Course Details

| | |
|---|---|
| **Course Code:** | COMP9243 |
| **Course Title:** | Distributed Systems |
| **Units of Credit:** | 6 |
| **Course WebSite:** | http://www.cse.unsw.edu.au/~cs9243/ |
| **Handbook Entry:** | http://www.handbook.unsw.edu.au/postgraduate/courses/2014/COMP9243.html |

Definition of Units of Credit

Along with lectures and the exam, the assignments make up a significant part of the course load.

## Course Aims

A distributed system is a computer system consisting of several independent computers, connected by a network, that can work together to perform a task or provide a service. Typical examples include: the World Wide Web, cloud computing, networked file systems, DNS, and massive multiprocessor supercomputers.

In this course we aim to provide students with a deeper understanding of distributed systems. In particular we focus on the principles, techniques, and practices relevant to the design and implementation of such systems. The course takes a systems-oriented view of distributed systems, concentrating on infrastructure software and providing hands-on experience implementing distributed systems.

Concretely the course has the following objectives:

- Present the principles underlying the functioning of distributed systems;
- Create an awareness of the major technical challenges in distributed systems design and implementation;
- Expose students to modern and classic technology used in distributed systems and their software;
- Expose students to past and current research issues in the field of distributed systems;
- Provide experience in the implementation of typical algorithms used in distributed systems.

# Student Learning Outcomes

After completing this course you will be able to:

- Explain what a distributed system is, why you would design a system as a distributed system, and what the desired properties of such systems are;
- List the principles underlying the functioning of distributed systems, describe the problems and challenges associated with these principles, and evaluate the effectiveness and shortcomings of their solutions;
- Recognise how the principles are applied in contemporary distributed systems, explain how they affect the software design, and be able to identify features and design decisions that may cause problems;
- Design a distributed system that fulfills requirements with regards to key distributed systems properties (such as scalability, transparency, etc.), be able to recognise when this is not possible, and explain why;
- Build distributed system software using both basic OS mechanisms as well as higher-level middleware and languages.

These learning outcomes cover many of the UNSW graduate attributes including: an in-depth engagement with the relevant disciplinary knowledge in its inter-disciplinary context, the capacity for analytical and critical thinking and for creative problem solving, and the skills of effective communication.

# Related Courses and Programs

Students are expected to be highly competent in the use of a systems programming languageÂ—preferably C. Students not familiar with C (or any other language used for assignments) will be expected to learn it on their own. Furthermore, students are expected to be confident in the use of standard POSIX system libraries (including file handling, process management, signal handling, and socket programming). Moreover, students are required to become competent in the concurrent programming language Erlang on their own (a brief introduction is given in the first lecture).

### Course Prerequisites

Since networks make up a key part of distributed systems, and since many of the key challenges and solutions presented in the course extend those found in operating systems, a solid background in both networking and operating systems is essential. The course builds on many of the topics covered in the prerequisites.

1. COMP3231 or COMP9201 Operating Systems
2. COMP3331 or COMP9331 Computer Networks and Applications

### Parallel Teaching

Only COMP9243 students attend these classes. The students include postgraduates and senior undergraduates.

# Course Philospohy and Teaching Strategies

The learning focus in this course is primarily on lectures and assignments. Relevant research publications and other material that goes deeper into the subjects handled during lectures are also provided, however, these are usually optional. We also endeavour to provide examples of how the material covered in the course is put into practice through guest lectures from industry practitioners.

While the assignments contribute a portion of the final mark, one of their primary aims is to provide an opportunity to put into practice some of the material presented in the lectures and to get a feel for the complexity of designing and programming distributed systems. To this end, we also provide unmarked exercises that encourage students to further explore the topics presented in the course.

Finally, the lecturer is also always available to engage in further discussion related to the course material.

# Course Schedule

*Lectures*: Tuesday 6-9pm; Room: Law 388 (K-F8-388)

The following is a provisional list of topics covered in the lectures (subject to change):

| Week | Date | Topic |
|---|---|---|
| 1 | 4 MarÂ | Introduction to Distributed Systems, Introduction to ErlangÂ |
| 2 | 11 Mar | System Architecture, Communication |
| 3 | 18 Mar | Replication & Consistency, Distributed Shared Memory |
| 4 | 25 Mar | Synchronisation & Coordination 1 |
| 5 | 1 Apr | Synchronisation & Coordination 2 |
| 6 | 9 Apr | Fault Tolerance |
| 7 | 15 Apr | Naming, Distributed File Systems, Middleware |
|  | 22 Apr | Break |

| | | |
|---|---|---|
| 8 | 29 Apr | Cloud |
| 9 | 6 May | Security |
| 10 | 13 May | Web Services, DevOps |
| 11 | 20 May | Distributed Systems in Practice |
| 12 | 27 May | Review |

# Assessment

## Assignments

There will be **three** programming assignments, which contribute in specified parts to the total assignment mark for this course. *Tentative* details are as follows:

| # | Release - Beginning of | Due - End of | Topic | Marks |
|---|---|---|---|---|
| 1 | Week 3 | Week 6 | Distributed Shared Memory (group assignment) | 50 |
| 2 | Week 8 | Week 9 | Router Network (individual assignment) | 30 |
| 3 | Week 10 | Week 11 | Cloud (group assignment) | 20 |

Assignment submissions need to be coded in C, Erlang, Java, or Python (as determined by the assignment specification). Depending on the assignment, submission will be tested on the School's `vina` cluster running under Debian GNU/Linux, or on Amazon's AWS service.

The penalty for late submission of assignments will be 6% (of the worth of the assignment) subtracted from the raw mark per day of being late. In other words, earned marks will be lost. For example, assume an assignment worth 25 marks is marked as 20, but had been submitted two days late. The late penalty will be 3 marks ((0.06 * 25) * 2), resulting in a mark of 17 being awarded. No assignment will be accepted later than **one** week after the deadline.

## Final Exam

A **three hour open book** written final examination will be conducted.

### Supplementary exam

Supplementary exams will only be awarded in serious cases where the student has submitted a fully documented request for Special Consideration within 3 working days of the exam. In accordance with School policy, supplementary exams will **not** be awarded as a second chance for poorly performing students. In particular, it is unlikely that a supplementary will be awarded to students who have actually sat the proper exam. Make up your mind whether or not you are sick before attempting the exam! Moreover, a supplementary exam will only be awarded where a student's performance during the semester has been of satisfactory standard. Please refer to the sections under the heading *Special Consideration - Illness & Misadventure* and *CSE Supplementary Assessment Policy* in the Yellow Form for further details.

Supplementary exams will be **oral**. Supplementary final exams will be held either on the day before, or the day after, the written supplementary exams held for other courses.

## Final Mark

The final mark will be the **weighted average** of the exam mark and the total assignment mark, where the exam mark accounts for 65% and the assignment mark accounts for 35% of the final mark . No-one will be allowed a passing grade unless their exam mark was at least 50% of the maximum possible exam mark.

# Resources for Students

### Textbooks

There is no single textbook for this course. Lecture material will be taken from a variety of sources, including recent research publications.

### Reference Books

Books covering a significant part of the lecture material are the following:

- George Coulouris, Jean Dollimore & Tim Kindberg: Distributed Systems: Concepts and Design, 5th ed, 2011, Addison-Wesley, approx $130. Available at UNSW bookshop and library.
  *Up to date, recent edition. Only up-to-date distributed systems textbook. Broad coverage, but too verbose*

*in many places. 4th edition is also good*

- Andrew S. Tanenbaum & Maarten van Steen: Distributed Systems: Principles and Paradigms, 2nd ed, 2007, Pearson Prentice Hall, approx $120. Available at UNSW bookshop and library.
  *Getting a bit old. Good breadth of coverage, well organised. 1st edition is also good.*
- Pradeep K. Sinha: Distributed Operating Systems, 1997, IEEE Press, approx $140. Available at UNSW library.
  *Good coverage (breadth and depth). Getting a bit dated.*
- Doreen L. Galli: Distributed Operating Systems, 1999, Prentice Hall, approx $100. Available at UNSW library.
  *Extensive examples, but rather shallow in its coverage, very little theory. Leaves out too much detail.*
- Mukesh Singhal & Niranjan G. Shivaratri: Advanced Concepts in Operating Systems, 1994, McGraw-Hill, approx $???.
  *Somewhat dated, but that's not a problem as the book concentrates on the theoretical foundations. Case studies are out of date.*

None of these books covers all the course material.

**Other Material**

Reference to research papers will be provided throughout, and it is recommended that you read them. Lecture notes and other information can be found under the course's WWW home page at URL http://www.cse.unsw.edu.au/~cs9243/. It is recommended that you refer to this page regularly.

# Academic Honesty and Plagiarism

Assignments will be marked as individual or team assignments as stated in the specification. Team work is not permitted in individual assignments; cooperation between teams is not permitted in team assignments. Both will be regarded as cheating. Copying of assignment code leads to immediate failure of the course with a result of 0FL. Negative or reduced marks will be imposed in less severe cases (e.g., team work in individual assignments). Cases of repeated cheating (in this or other courses) will be treated as academic misconduct and will result in an application being made to have the offender(s) suspended from the University. Please carefully read the discussion under the heading *Originality of Assignment Submissions* on the Yellow Form to which you agree to be legally bound when enrolling for this course and/or signing up for the use of the computing facilities of the School.

This is no idle threat. We check for cheating, and we go after cheaters, to protect those who get their marks through honest work. In the past students have failed due to cheating.

## What is Plagiarism?

Plagiarism is the presentation of the thoughts or work of another as one's own.* Examples include:

- direct duplication of the thoughts or work of another, including by copying work, or knowingly permitting it to be copied. This includes copying material, ideas or concepts from a book, article, report or other written document (whether published or unpublished), composition, artwork, design, drawing, circuitry, computer program or software, web site, Internet, other electronic resource, or another person's assignment without appropriate acknowledgement;
- paraphrasing another person's work with very minor changes keeping the meaning, form and/or progression of ideas of the original;
- piecing together sections of the work of others into a new whole;
- presenting an assessment item as independent work when it has been produced in whole or part in collusion with other people, for example, another student or a tutor; and,
- claiming credit for a proportion a work contributed to a group assessment item that is greater than that actually contributed.**

Submitting an assessment item that has already been submitted for academic credit elsewhere may also be considered plagiarism.

The inclusion of the thoughts or work of another with attribution appropriate to the academic discipline does not amount to plagiarism.

Students are reminded of their Rights and Responsibilities in respect of plagiarism, as set out in the University Undergraduate and Postgraduate Handbooks, and are encouraged to seek advice from academic staff whenever necessary to ensure they avoid plagiarism in all its forms.

The Learning Centre website is the central University online resource for staff and student information on plagiarism and academic honesty. It can be located at: www.lc.unsw.edu.au/plagiarism.

The Learning Centre also provides substantial educational written materials, workshops, and tutorials to aid students, for example, in:

- correct referencing practices;
- paraphrasing, summarising, essay writing, and time management;
- appropriate use of, and attribution for, a range of materials including text, images, formulae and concepts.

Individual assistance is available on request from The Learning Centre.

Students are also reminded that careful time management is an important part of study and one of the identified causes of plagiarism is poor time management. Students should allow sufficient time for research, drafting, and the proper referencing of sources in preparing all assessment items.

* Based on that proposed to the University of Newcastle by the St James Ethics Centre. Used with kind permission from the University of Newcastle

** Adapted with kind permission from the University of Melbourne.

## Course Evaluation and Development

Student feedback on this course, and on the lecturing in this course, will be gathered via questionnaires (including the CATEI evaluation) held at or after the end of the course. Student feedback is taken seriously, and continual improvements are made to the course based in part on this feedback. While no serious issues have been raised in the CATEI and other evaluations, we have made changes to assignments and assignment specifications based on student comments and updated lecture material to ensure that it is up-to-date as well as to improve sections that have been unclear to students in the past. We will also aim to provide more tutorial style questions during the course and review solutions with the class. Since this is a fast moving field, we have endeavoured to include more relevant material on current trends in distributed computing, in particular cloud computing and the design of large scale distributed systems and distributed databases. We also include (subject to availability of speakers) relevant guest lectures from industry practitioners to provide examples of how the material covered in class is used in practice.

## Further Information

- Information on UNSW Health and Safety policies and expectations
- Equity and Disabilities issues