Course Outline

Contents

- Course Details
- Course Summary
- Course Timetable
- Course Aims
- Student Learning Outcomes
- Assumed Knowledge
- Teaching Rationale
- Teaching Strategies
- Assessment
- Academic Honesty and Plagiarism
- Course Schedule
- Resources for Students
- Course Evaluation and Development

Course Details

Course Code	COMP9243
Course Title	Distributed Systems
Convener	Ihor Kuz (https://webcms3.cse.unsw.edu.au/users/z3139723)
Lectures	Tue 18:00-21:00, Ainsworth (J17) G02, weeks 1-7,9-13
Consults	Mon 15:00-16:00, K17, level 3, office 301H
Units of Credit	6
Course Website	http://cse.unsw.edu.au/~cs9243 (http://cse.unsw.edu.au/~cs9243)
Handbook Entry	http://www.handbook.unsw.edu.au/postgraduate/courses/current/COMP9243.html (http://www.handbook.unsw.edu.au/postgraduate/courses/current/COMP9243.html)

Course Summary

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.

Course Timetable

The course timetable is available here (/COMP9243/17s1/timetable) .

Course Aims

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 fulfils 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.

This course contributes to the development of the following graduate capabilities:

Graduate Capability	**Acquired in**
scholarship: understanding of their discipline in its interdisciplinary context	lectures, assignments
scholarship: capable of independent and collaborative enquiry	lectures, assignments, exam
scholarship: able to apply their knowledge and skills to solving problems	lectures, assignments, exams
scholarship: capable of effective communication	assignments, exams
leadership: enterprising, innovative and creative	assignments, exams
leadership: collaborative team workers	assignments
professionalism: capable of independent, self-directed practice	lectures, assignments

Assumed Knowledge

Before commencing this course, students should:

- be highly competent in the use of a systems programming language—preferably C
- be confident in the use of standard POSIX system libraries (including file handling, process management, signal handling, and socket programming).
- be able to learn the concurrent programming language Erlang on their own (a brief introduction is given in the first lecture).

Furthermore 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 (http://www.cse.unsw.edu.au/~cs3231)
2. COMP3331 or COMP9331 Computer Networks and Applications (http://www.cse.unsw.edu.au/~cs3331)

Teaching Strategies

- Lectures: introduce concepts and principles, show examples of how these can be applied in practice
- Assignments: allow students to experience the challenges of building distributed systems and to put into practice the principles learned in the lectures.
- Guest lectures and case studies: provide concrete examples of how the principles and paradigms learned in class apply to the real world.

Teaching Rationale

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.

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 5	Distributed Shared Memory (group assignment)	40
2	Week 7	Break	Router Network (individual assignment)	40
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 **two 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* on the Essential Advice (https://www.engineering.unsw.edu.au/computer-science-engineering/about-us/organisational-structure/student-services/policies/essential-advice-for-cse-students) page 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.

Academic Honesty and Plagiarism

UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All UNSW staff and students have a responsibility to adhere to this principle of academic integrity. Plagiarism undermines academic integrity and is not tolerated at UNSW. Plagiarism at UNSW is defined as using the words or ideas of others and passing them off as your own.

If you haven't done so yet, please take the time to read the full text of

- UNSW's policy regarding academic honesty and plagiarism (https://student.unsw.edu.au/plagiarism)

The pages below describe the policies and procedures in more detail:

- Student Code Policy (https://www.gs.unsw.edu.au/policy/documents/studentcodepolicy.pdf)
- Plagiarism Policy Statement (https://www.gs.unsw.edu.au/policy/documents/plagiarismpolicy.pdf)
- Plagiarism Procedure (https://www.gs.unsw.edu.au/policy/documents/plagiarismprocedure.pdf)
- Student Misconduct Procedure (https://www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf)

Course Schedule

Lectures : Tuesday 6-9 pm; Room: G02 Ainsworth Building (J17) (https://studentvip.com.au/unsw/kensington/maps/60859)

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

Week	Date	Topic
1	28 Feb	Introduction to Distributed Systems, Introduction to Erlang
2	7 Mar	System Architecture, Communication
3	14 Mar	Replication & Consistency, Distributed Shared Memory
4	21 Mar	Synchronisation & Coordination 1
5	28 Mar	Synchronisation & Coordination 2
6	4 Apr	<Ihor away - no lecture>
7	11 Apr	Fault Tolerance
	18 Apr	Break
8	25 Apr	<Anzac day>
9	2 May	<Ihor away - guest lecture> Cloud, Web Services, DevOps
10	9 May	Middleware, Naming, Distributed File Systems
11	16 May	Security
12	23 May	Distributed Systems in Practice
13	30 May	Review

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:

- Andrew S. Tanenbaum & Maarten van Steen: Distributed Systems: Principles and Paradigms, 3rd ed, 2017, approx $40, free PDF available on distributed-systems.net (https://www.distributed-systems.net/index.php/books/distributed-systems-3rd-edition-2017/) , hardcopy on Amazon.
  Good breadth of coverage, well organised. 2nd edition is also good.
- George Coulouris, Jean Dollimore & Tim Kindberg: Distributed Systems: Concepts and Design, 5 th ed, 2011, Addison-Wesley, approx $130. Available at UNSW bookshop and library.
  Up to date, recent edition. Broad coverage, but too verbose in many places. 4 th 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/ (http://www.cse.unsw.edu.au/~cs9243/) . It is recommended that you refer to this page regularly.

Course Evaluation and Development

This course is evaluated each session using the CATEI system as well as a more detailed course-specific survey.

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.

Resource created 5 years ago (Monday 13 February 2017, 04:32:46 PM), last modified 5 years ago (Tuesday 28 February 2017, 05:42:35 PM).

Comments (/COMP9243/17s1/forums/search?forum_choice=resource/6313) (/COMP9243/17s1/forums/resource/6313)

There are no comments yet.