GÖTEBORGS UNIVERSITET

Faculty Board of IT University

DIT290, Distributed systems, advanced course, 7.5 higher education credits

Second Cycle

This syllabus in English is the binding document.

1. Confirmation

The Faculty Board/the Dean of IT Faculty established the course plan at 2006-11-17. It has been revised 2009-05-20 to be valid from spring term 2010.

Field of education: Science

Department: Computer Science and Engineering

2. Position in the educational system

The course is a part of the Computer Science Master's programme and an single subject course at the University of Gothenburg.

3. General prerequisites

The requirement for the course is to have successfully completed a first year studies within the subject Computer Science or equivalent. Specifically, the course DIT240 Distributed systems is required.

4. Course content

In this course the students first will get to know the points of inherent difference and strength of distributed systems compared with sequential or strongly-coupled systems; consequently, the aim of the course is to teach the students to study the issues and problems that have to be addressed and solved efficiently for these differences to be taken advantage of, so that the system retains its strength and high potential.

In particular, we will address and study the following issues in distributed systems:

Synchronization:

Continuation (after the course DIT240 Distributed systems) of the study of distributed clock-synchronization and its uses in mutual exclusion; study about resource allocation in general. Advance one step further, to see how to get solutions without the need for clock synchronisation (and hence of lower cost in practice).

Replication:

The advantages and costs of replicating data: the potential improvement in response times and reliability, and the extra communication costs involved in keeping data consistent.

Concurrency control:

Study of different types of locking, deadlock detection. Additionally, study of concurrency control methods without locking.

Recovery and Fault Tolerance:

Check-pointing, optimistic and pessimistic recovery control.

Fault-tolerant Agreement in Distributed Systems:

(a very special and significant problem, since it is a key issue in most synchronization and coordination problems in distributed systems) Study of the basic impossibility results and discuss their implications; proceed with solutions and protocols for systems with certain strengths and design structures.

Distributed Shared Memory:

The difference between shared-memory-based programming and message-passing-based programming; types of consistency; write-update and write-invalidate protocols.

5. Learning outcomes

After the course the student should be able to:

- understand and solve problems that distributed systems like P2P, collaborative environment systems or even parts of the core internet technology have to solve in order to be able to function efficiently and correctly.

The student will also be able to combine theoretical studies with their respective application oriented consequences.

6. Required reading

See separate literature list.

7. Assessment

There will be a final written examination, as well as a couple of assignments during the term, in combination with 2 practical projects.

A student who has failed a test twice has the right to change examiner, unless weighty argument can be adduced. The application shall be sent to the department and has to be in writing.

8. Grading scale

The course is graded with the following marks: Fail, Pass, Pass with Distinction.

9. Course evaluation

The course is evaluated through meetings both during and after the course between teachers and student representatives. Further, an anonymous questionnaire can be used to ensure written information. The outcome of the evaluations serves to improve the course by indicating which parts could be added, improved, changed or removed.

10. Additional information

The course is held in English