Course Catalog Description: Principles of distributed systems including system design, distributed algorithms, consistency and concurrency, and reliability and availability. The role of these foundational issues in distributed file systems, distributed computing, and data-driven systems.

Credits: 4 cr.
Prerequisite(s): C343 or H343

General Description:

Much of the exciting computing taking place today is distributed. Web services, cloud computing, virtualization, peer-to-peer systems and Internet computing all have a foundation in distributed systems concepts. In this course you will learn foundational concepts of distributed systems, including algorithmic, architectural, and experimental.

You will also get hands-on programming experience in large-scale, or sometimes called, “system-level programming”. The course is intended to give the student a balanced experience that will grow their foundational understanding of distributed systems, and at the same time provide valuable hands-on experience that puts the foundational understanding in a context that can be of immediate use.

Topics and Agenda

<table>
<thead>
<tr>
<th>Weeks 1 - 2</th>
<th>Overview and background</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weeks 3 - 4</td>
<td>Systems: modularity, abstraction, layering</td>
</tr>
<tr>
<td>Weeks 5 - 6</td>
<td>Concurrency and virtualization in distributed systems</td>
</tr>
<tr>
<td>Week 7</td>
<td>Midterm review and midterm</td>
</tr>
<tr>
<td>Week 8</td>
<td>Performance evaluation and benchmarks</td>
</tr>
<tr>
<td>Weeks 9 – 11</td>
<td>Distributed file systems and data management servers</td>
</tr>
</tbody>
</table>
Week 12 | Data consistency
---|---
Weeks 13 – 14 | Distributed algorithms: clocks, elections, mutual exclusion
Week 15 | Fault tolerance: resilience, reliable group communication, recovery
Week 16 | Final Exam

**Learning Objectives**

- Acquire a working knowledge of the foundational issues in distributed systems: consistency, concurrency, race conditions, modularity, abstraction, resilience and their role in distributed systems.
- Learn the methodology and importance behind performance evaluations
- Learn to write software that is part of a larger system, that relies on other software; gain understanding of importance of good interface design
- Acquire in-depth knowledge of technical solutions in existence that address the data storage and access, and have practice verbally articulating strengths and weaknesses of each

**Textbook and materials**

The course will use the following textbook. It is strongly suggested that you have your own copy of the book.


The course may use supplemental materials and access to these will be provided.

**Projects**

The course will include two projects that build upon one another in sequence. The projects require programming, and will grow the student’s skills as a systems programmer, better qualified to contribute to a large-scale systems development project. Distributed systems today are too large for any one person to write, so the systems programmer must be comfortable working with APIs, libraries, and code from other programmers and other organizations. The student will program in Java or C#. The programming projects are group projects. The project grade will be based on a demo, the quality of the code, and a written report.

**Grading**

The course grade is determined by the student's performance over several areas: projects (35%), homework (30%), and exams (35%).