UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

COURSE SYLLABUS

Distributed Systems

2526-2-E3101Q112

Aims

Knowledge and Understanding

By the end of the course, students will have acquired:

- Fundamental knowledge of the architectures, communication models, and key properties of distributed systems, with particular emphasis on client-server and peer-to-peer paradigms
- An understanding of both synchronous and asynchronous communication models, including message-oriented (persistent and transient) and push/pull mechanisms
- Knowledge of the core technologies enabling distributed communication, such as TCP/IP sockets, Remote Procedure Call (RPC), and Remote Method Invocation (RMI)
- Understanding of the functioning of distributed web applications, including the foundations of HTTP, REST, and architectural patterns such as MVC
- Knowledge of the fundamental concepts of concurrent programming and the use of state machines for modeling the behavior of distributed systems
- The ability to critically assess when and how distributed solutions are appropriate for specific computational problems

Applying Knowledge and Understanding

Upon completion of the course, students will be able to:

- Design and develop basic distributed applications on web architectures using Java, servlets/JSP, JavaScript, and AJAX techniques
- Employ TCP/IP sockets to implement client-server and peer-to-peer communication models
- Apply remote communication technologies such as RPC and RMI to support distributed component interaction
- Integrate web technologies (HTML5, CSS, JSON, REST APIs) to build interactive and dynamic distributed applications

- Design and implement concurrent applications in Java using threads, synchronization mechanisms, and monitors
- Experiment with the technologies and tools introduced in the course through hands-on lab sessions and project-based activities

Transversal Skills

The course also fosters the development of the following transversal skills:

Independent judgment: through project work and exercises requiring critical evaluation of technological choices in real-world scenarios.

Communication skills: by producing technical documentation and presenting implemented solutions during practical activities.

Learning skills: by providing theoretical, methodological, and practical tools that enable students to independently deepen and update their expertise in the field of distributed systems.

Contents

Course Content Summary

- **Distributed Architectures**: client-server, peer-to-peer, and tiered models
- **Inter-process Communication**: TCP/IP sockets, RPC, RMI, asynchronous messaging
- **Distributed Web Applications**: HTTP, REST, servlets/JSP, MVC pattern
- **Dynamic Web Application**: JavaScript, AJAX, introduction to Node.js
- **Data Representation**: HTML5, CSS, JSON, RDF
- **Concurrency**: multithreaded programming in Java, synchronization
- **Laboratory**: guided development of basic distributed applications

Detailed program

Certamente. Ecco la traduzione formattata con Markdown.

Detailed Syllabus

1. Fundamentals of Distributed Systems

- Definitions and characteristic properties of distributed systems.
- Reference architectural models: client-server, peer-to-peer, layered, and tiered architectures.
- Names, identifiers, and addresses: URI and DNS.

2. Communication Models

- Synchronous and asynchronous communication models.
- Message-oriented communication: persistent and transient models.
- Push and pull interactions.

3. Communication Technologies

- TCP/IP socket programming: client-server applications and application-level protocols.
- Procedure call-based communication: Remote Procedure Call (RPC) and Remote Object Invocation (Java RMI).

4. Distributed Web Applications

- Fundamentals of the Web: URI and the HTTP protocol.
- Request/response mechanisms.
- Designing web applications using servlets and JSP.
- The Model-View-Controller (MVC) architectural pattern.
- Introduction to REST web services and Web APIs.

5. Dynamic Web Application

- Introduction to JavaScript and AJAX techniques.
- Development of interactive client-side applications.
- Debugging and dynamic modification of web pages.
- Invoking remote services.
- A brief overview of Node.js.

6. Data Modeling and Representation

- HTML5 and CSS for data presentation.
- XML and JSON as formats for information transfer.
- Examples of API-based mashups.

7. Concurrent Programming

- Synchronization concepts and the use of monitors.
- Multi-threaded programming in Java with shared memory.
- A brief overview of behavior modeling using finite state machines.

8. Laboratory Activities

- Design and development of simple distributed applications.
- Guided experimentation with the technologies presented during the course, with support for the completion of an optional project.

Prerequisites

The prerequisites for this course include the following knowledge and skills:

- **Object-Oriented Programming in Java and JUnit tests**: This knowledge is expected from the "Programmazione 2" course.
- **Processes and inter-process communication**: This topic is covered in the "Reti e Sistemi Operativi" (Networks and Operating Systems) course.
- **TCP/IP Protocol**: Also from the "Reti e Sistemi Operativi" course.
- **Markup Languages (XML and HTML) and related manipulation tools**: This includes understanding and working with these languages.
- **Finite State Automata**: This knowledge comes from the "Linguaggi e Computabilità" (Languages and

Computability) course.

Teaching form

The course's teaching is organized into several components:

- **Lectures and Classroom Exercises**:
    - 32 hours of frontal lectures are planned.
    - 20 hours of classroom exercises will be conducted.
    - Up to a maximum of 20% of lectures and exercises may be delivered remotely via audio-video recordings.

- **Interactive Teaching (Laboratory and Online Support)**:
    - An additional 24 hours of laboratory activities are scheduled.
    - Further demonstrations or explanations will be offered through the e-learning website.
    - The e-learning site will also include support tools such as web forums and FAQs to foster interaction.
    - These activities aim to provide support from instructors and participating students, offering demonstrations or practical suggestions for solving problems and exercises.
    - Specific support will be provided for the optional project at the end of the laboratory activities.

- **Individual Study**:
    - Independent study will be supported by:
        - Recommended textbooks.
        - Teaching materials available on the e-learning website.
        - Interactive activities accessible on the e-learning platform.

- **Language of Instruction**:
    - The course will be delivered in Italian.

Textbook and teaching resource

On the e-learning site are available :

- slides of the lessons in pdf format.
- further material (articles to complete the reference texts, links to online resources, exercises to be carried out).
- equipment and solutions for exercises carried out in the laboratory.

Textbooks:

Distributed Systems: Principles and Paradigms - 2nd edition, Andrew S. Tanenbaum and Maarten van Steen,

Pearson - Prentice Hall, 2007.

Already adopted in the course of Networks and Operating Systems:

Reti di calcolatori e internet – Un approccio top-down 4a Edizione, James F. Kurose, Keith W. Ross Addison Wesley – 2008, ISBN 9788871924557
Chapter 1, 2

A. Silberschatz, P. Baer Galvin, G. Gagne, Sistemi operativi - Concetti ed esempi, 8/Ed. 2009, ISBN 9788871925691
Chapter 3, 4, 6, 7, 16

Semester

Second semester

Assessment method

The assessment of learning for the "Distributed Systems" course is structured into several components, each contributing to the final grade:

- **Final Exam**:
    - The final exam can contribute up to 30 points.
    - It consists of a written test and a possible oral examination.
    - **Written Test in the Laboratory**: The written test is conducted in the laboratory on a PC and is divided into two phases:
        - **Phase 1: Closed Questions**: This phase consists of closed questions on all course topics. To access Phase 2, it is necessary to pass this phase by achieving at least 50% correct answers.
        - **Phase 2: Mixed Questions**: This phase includes open or closed questions with comments, covering all topics discussed.
    - Passing the exam requires a score of 18/30 or higher.
    - **Types of Questions**: The written test includes:
        - Questions on the concepts presented.
        - Reasoning and deduction questions.
        - Problem-solving exercises requiring the development of a solution to an assigned problem.

- **Oral Exam (Optional)**:
    - The oral exam is at the discretion of the instructor.
    - It involves the review of the written assignments, accompanied by a discussion and any additional questions posed by the instructor.

- **Laboratory (Optional Project)**:
    - Laboratory evaluation is based on a final project.

    - This project is optional and can contribute up to 4 points to the final grade.

- **Mid-term Tests (Substitute for the Exam)**:
    - Mid-term tests are scheduled and substitute for the first exam session. Students who do not take them will need to sit for the second session.
    - These mid-term tests can also be taken by students enrolled in their third year or those who are off-course (fuori corso).
    - **First Test**: Consists of closed questions.
    - **Second Test**: Consists of closed questions and mixed questions.
    - Admission to the second test is granted to those who achieved a score of 18/30 or higher in the first test.
    - There are no make-up opportunities for the mid-term tests.

- **Final Grade**:
    - The final grade is calculated as the sum of the score obtained in the final exam and the optional project score.

Office hours

Prof. Ciavotta: Tuesday from 12:30 to 14:30 by appointment.

Prof. Savi: By appointment via email at marco.savi@unimib.it.

Questions and discussions on teaching topics can be posed using the forums in e-learning.

Sustainable Development Goals

INDUSTRY, INNOVATION AND INFRASTRUCTURE