

INFO I320 Distributed Systems and Collaborative Computing

Indiana University East

Sample Syllabus for New Course

Instructor

Gregory D. Weber. See <http://mypage.iu.edu/~gdweber/> for office hours and other contact information.

Overview

Increasingly, business applications based on distributed computing are deployed through the Internet or intranets, including the World-Wide Web. This course covers the design and development of such systems. Design of a distributed system includes an architecture (e.g., client-server, peer-to-peer, or three-tier) and a protocol (a set of rules for exchanging data). Developing a distributed system involves designing, coding, and testing a set of programs that implement the protocol.

The course covers a variety of protocols that may be used as tools by the distributed applications programmer. Networks are complex, multi-layered systems. At the transport layer, TCP and UDP are the basic Internet protocols designed to get packets of data from one host to another. Higher up, protocols at the application layer are built on top of TCP or UDP. Application protocols include those used for file transfer (FTP), email (SMTP, POP3, IMAP), hypertext (HTTP), and custom applications. Many distributed applications are deployed through the World-Wide Web, using either client-side programming (applets, Javascript), server-side programming (CGI, servlets), or both. Business applications must frequently talk to a database, using protocols such as ODBC and JDBC. Finally, distributed object systems (e.g., using RMI or CORBA) provide a higher level of abstraction which makes distributed programming easier.

The course uses Java as the implementation programming language. Java provides sophisticated and relatively easy-to-use network programming capabilities. Most of the tools and techniques covered, however, can also be applied in C, C++, and other programming languages.

Prerequisites

Junior or senior standing; knowledge of the Java programming language, at a level equivalent to CSCI C243 or CSCI C307; knowledge of relational databases and SQL, equivalent to I340. Prospective students who are acquainted with C or C++ but not Java should be able to manage with some extra study, using resources such as the [Java Tutorial](#).

Textbook and Other Learning Resources

(Required) M. L. Liu, *Distributed Computing: Principles and Applications*. Pearson Addison-Wesley: 2004. ISBN 0-201-79644-9.

Additional readings on the World-Wide Web.

The I320 class web site (<http://mypage.iu.edu/~gdweber/courses/i320/>) will have lecture notes, programming examples, and links to additional readings and software.

Course Outline

Weeks 1-2

0. Java review or remedial study, as needed. Java Tutorial (online).
1. Introduction. Liu, chapter 1.
2. Interprocess communication. Liu, ch. 2.
3. Distributed computing paradigms. Liu, ch. 3.

Weeks 3-6

4. Sockets. Liu, ch. 4; Java Tutorial: Custom Networking Trail.
Project 1: Using sockets.
5. The client-server paradigm. Liu, ch. 5.
6. Group communication. Liu, ch. 6.

Weeks 7-8

- Midterm exam**, covering chapters 1-6.
7. Distributed Objects I: RMI. Liu, ch. 7 (omit Liu, ch. 8, Advanced RMI); Java Tutorial: RMI Trail.
Project 2: Using RMI.

Weeks 9-11

8. Internet Applications I: HTTP, HTML, XML, CGI. Liu, ch. 9; Raggett, HTML tutorial (online).
9. Internet Applications II: Applets and Servlets. Liu, ch. 11 (omit SOAP); J2EE Servlet Tutorial (online)
Project 3: Servlet.

Week 12

10. Database Connectivity (JDBC). Java Tutorial: JDBC Trail.
Project 4: Servlet with database.

Weeks 13-14

11. Distributed Objects II: CORBA. Liu, ch. 10; Tutorial: Getting Started with Java IDL.
12. Advanced distributed computing paradigms. Liu, ch. 12.

Final exam. Exam will be comprehensive but emphasize second half of course.

Projects

Students working in two-person teams will develop:

1. A client-server application using TCP sockets
2. A CGI and/or a servlet application
3. A distributed object application using RMI
4. A distributed application using a web browser as the front end, a CGI or servlet intermediary, and a backend database

Learning Objectives

The successful student will be able to:

- Distinguish between protocols (e.g., UDP vs. TCP, RMI vs. CORBA) and evaluate their strengths and weaknesses for distributed applications.
- Distinguish between distributed application paradigms (e.g., client-server, peer-to-peer, three-tier) and evaluate their strengths and weaknesses for distributed applications.
- Develop distributed Java programs using TCP sockets, CGI and/or servlets, RMI, and JDBC.