
CO3097 Programming Secure and Distributed Systems

Credits: 20 Convenor: Dr. S. Yang Semester: 1st

Prerequisites:	<i>Essential: CO1003,CO1004,CO1011</i>	<i>Desirable: CO2006</i>
Assessment:	<i>Coursework: 40%</i>	<i>Three hour exam in June: 60%</i>
Lectures:	<i>30 hours</i>	
Surgeries:	<i>10 hours</i>	Private Study: <i>100 hours</i>
Laboratories:	<i>10 hours</i>	

Subject Knowledge

Aims This course will equip students with the knowledge required to build secure and distributed applications in Java. The course covers both the fundamental problems facing distributed applications such as concurrency and security and more practical material describing how these problems can be tackled and implemented in Java.

Learning Outcomes Students will be able to: build simple distributed applications using Java's networking capabilities; build concurrent distributed applications using multiple threads; build distributed applications with security enhancements using Java's security and cryptographic extensions. They will be able to test such systems.

Methods Class sessions together with lecture slides, recommended textbook, worksheets, printed solutions, and some additional hand-outs and web support.

Assessment Marked coursework, traditional written examination.

Skills

Aims To teach students how to methodically solve problems given the techniques available to them.

Learning Outcomes Students will be able to: breakdown a problem to identify essential elements; apply prior knowledge of subject to analyse problems; design a plan to solve a problem; implement a planned solution and evaluate the implementation.

Methods Class sessions together with worksheets.

Assessment Marked coursework, traditional written examination.

Explanation of Prerequisites A significant aspect of the module will be the reinforcement of material delivered in lectures with practicals involving students implementing distributed systems in Java. Hence a basic knowledge of Java, as provided in CO1003 and CO1004, will be essential. A basic grounding in discrete mathematics will assumed during the lectures on security.

Although not essential, software design as found in CO2006 will aid the design of distributed systems.

Course Description The internet has caused a revolution in the way we use and think about computers. One of the key technologies underlying the internet is *distributed computing* which allows individual computing agents to be located, or *distributed*, on a network of computers but nevertheless work together on cooperative tasks. There are many motivations for distributing applications including:

- **Inherent Parallelism:** In distributed systems, many computers can work together at the solution of a problem simultaneously. Although this can be done using parallel computers, it is often cheaper to distribute applications over cheaper networks of PCs.
- **Inherent Distribution:** In many applications, distribution arises naturally. For example, in a traffic control system, data may be gathered at different measuring facilities, analysed at a central location and, depending upon the results of this analysis, commands sent back to improve the flow of traffic.

- **Scalability:** Distributed systems are easier to scale than centralised systems. For example, increasing computing power of a centralised system to cope with increasing demand is usually much more expensive than increasing the number of computer nodes in a distributed system.
- **Data Access:** It is often difficult and expensive to move large data sets between different locations. Instead, it can be more efficient to store data in locally and allow other users to remote access using a data server.
- **Resource Sharing:** In distributed systems, expensive resources such as printers can be used by devices distributed across a network. Again, this sharing can produce significant cost savings.

The development and programming of distributed systems is considerably more complex than that of local applications. Distributed applications usually run on spatially separated computers, consist of different components which need to communicate with each other and must ensure a consistent management of shared resources. In addition, of increasing importance is the issue of security in distributed systems, eg how can a component of a system communicate with another component in such away that malicious eavesdroppers cannot interfere. This course will introduce these and other fundamental problems in distributed computing, explain some of the solutions available and cover their implementation in the Java programming language. Students will thus gain insight into distributed computing and security as well as practical skills of immediate use.

Detailed Syllabus Distributed Programming: Networking in Java using sockets and streams, multi-threaded computation, liveness, safety, deadlock, livelock, mutual exclusion, communication protocols, semaphores and monitors, synchronous and asynchronous message passing in Java, client-server architectures.

Security: Security issues and concerns. Key management including generation, translation, agreement protocols and management paradigms. Authentication including message digests, MAC's, signatures, certificates. Symmetric and antisymmetric ciphers, eg DES, IDEA, RSA.

Java Support: Java security, Java Cryptography Architecture (JCA), Java Cryptography Extension (JCE).

Reading List

- [A] J. Farley, *Java Distributed Computing*; ISBN: 1565922069, O'Reilly. 1998..
- [A] M. Boger, *Java in Distributed Systems*; ISBN: 0471498386, Wiley. 2001..
- [A] J. Knudsen, *Java Cryptography*; ISBN: 1565924029, O'Reilly. 1998..
- [B] S. Oaks, *Java Security*; ISBN: 0596001576, O'Reilly. 1999..
- [B] J. Farley, *Java Distributed Computing*; ISBN: 1565922069, O'Reilly. 1998..
- [B] S. Hartley, *Concurrent Programming, The Java Programming Language*; ISBN: 0195113152, OUP. 1998..
- [B] D. Lea, *Concurrent Programming in Java: Design Principles and Patterns*; ISBN: 0201310090, Addison-Wesley. 1997..

Resources Lecture slides, web page, study guide, worksheets, handouts, lecture rooms with two OHPs, past examination papers.

Module Evaluation Course questionnaires, course review.