Global Learning Semesters
Course Syllabus
Course: COMP-612 Distributed Systems
Department: Computer Science
Host Institution: University of Nicosia, Nicosia, Cyprus

Course Summary

Course Code	Course Title	Recommended Credit Hours
COMP-612	Distributed Systems	4
Semester Offered	**Contact Hours**	**Prerequisites**
Please contact us	42-45	Students are expected to have undergraduate-level knowledge of programming and data structures, a good foundation on computer architecture and/or assembly language, operating systems (especially the understanding of the notion of processes and threads, inter-process communication and basic knowledge of networking). Fundamental mathematical skills will be an advantage. Referring to the Computer Science undergraduate-level courses offered at Intercollege, the students are expected to have covered the syllabus of the following: COMP-255 (C++ Language Programming) or COMP-511 (Initial Programming), COMP-254 (Assembly Language) and/or COMP-335 (Computer Organization and Architecture) or COMP-515 (Computer Architecture and Assembly Language), COMP-301 (Data Structures) or COMP-516 (Data Structures and Algorithms), COMP-354 (Operating Systems) or COMP-516 (Computer Networks and Operating Systems) and/or COMP-350 (Systems Software Programming) or COMP-517 (Object-Oriented Software Design and Development).
Department	**Level of Course**	**Language of Instruction**
Computer Science	Upper Division	English

Course Description

This course covers the principles of design, construction and programming of distributed systems. The presentation of concepts will be complemented with the examination of applications and cases, together with practical work, which includes programming assignments that require the use of distributed systems. The course will be delivered as a mixture of lectures, laboratory work, projects and software development.

Prerequisites

Students are expected to have undergraduate-level knowledge of programming and data structures, a good foundation on computer architecture and/or assembly language, operating systems (especially the understanding of the notion of processes and threads, inter-process communication and basic knowledge of networking). Fundamental mathematical skills will be an advantage. Referring to the Computer Science undergraduate-level courses offered at Intercollege, the students are expected to have covered the syllabus of the following: COMP-255 (C++ Language Programming) or COMP-511 (Initial Programming), COMP-254 (Assembly Language) and/or COMP-335 (Computer Organization and Architecture) or COMP-515 (Computer Architecture and Assembly Language), COMP-301 (Data Structures) or COMP-516 (Data Structures and Algorithms), COMP-354 (Operating

Systems) or COMP-516 (Computer Networks and Operating Systems) and/or COMP-350 (Systems Software Programming) or COMP-517 (Object-Oriented Software Design and Development).

Topic Areas

1. Definition of a distributed system.
2. Hardware and software concepts and architectural models.
3. Communication. Networking concepts. Remote Procedure Call. Remote Object Invocation. Message-oriented and stream-oriented communication.
4. Operating system architecture. Processes and threads. Clients and servers. Software agents.
5. Naming issues. Name services, locating, invoking and removing distributed entities.
6. Synchronization. Clocks, events and mutual exclusion.
7. Replication and consistency.
8. Fault tolerance, reliability and consistency. Distributed transactions.
9. Security. Principles, techniques and tools.
10. Distributed Shared memory.
11. Distributed file systems.
12. Distributed Object-based systems. Comparative study of systems (e.g. CORBA).

Course Assessment

Students will be assessed through a series of weekly assignments, individual/group programming/research projects, a midterm and a final exam. The individual programming projects will require the use of the Distributed Systems Lab at the student's own time. The Lab will also be used for guided Lab exercises, some of which will be part of the weekly assignments. The percentages contributing to the final grade are as follows:

Weekly Assignments:	5%
Projects (5):	25%
Midterm Examination:	20%
Final Examination:	50%

Description of course assessment:

Weekly assignments: These will consist of brief questions/exercises based on the chapter covered during that week.

Project work: These will include around five small projects each having duration of 2-3 weeks. The majority of these will be individual programming projects. Assessment will also include one group programming project and a research project including a presentation. The programming projects will require the use of the C/C++ programming language and the programming interfaces provided by the UNIX operating system.

Example projects

1) Develop a system to compute the solution(s) to a non-linear equation using an iterative algorithm. This must be implemented as a parent process with several child processes. The computation is performed by a child process given the initial condition and the result is returned to the parent. Non-convergence and duplicate solutions must be handled by the system. Alternatively realize it using threads.
2) Develop a client-server system to act as a directory service as follows: the server will have a file containing names, city and telephone number. The client may perform a query to get the number for a given name and town and must expect a response from the server containing a list of all matching records. Modify the system to permit the client to perform insertions and deletions, handling issues such as the concurrency of the server, which must be realized in Java as a multi-threaded application.
3) Modify the above system so that different machines hold directories for different cities, each running a server process. The client must be such that when it contacts any server with a query, the server will analyze the query and, if necessary, broadcast it to the other servers, collect the responses and forward them back to the client. MPI must be used to realize this system.

4) Group project (2-3 students in each group) to implement a distributed image processing system. The project will require the assessment of a given processing method to determine how it should be implemented efficiently. The design and implementation of the solution must demonstrate how issues such as fault tolerance, consistency and synchronization were taken into consideration.
5) Term paper on distributed/networked file systems. The students must perform a thorough literature survey on standards, protocols and implementations, and perform a comparative study of such file systems, clearly commenting on issues such as security and efficiency.

Midterm examination: It will contain the material covered up to the time of the midterm.

Final Examination: This examination will be comprehensive and it will include all the material covered throughout the semester.

Readings and Resources

Required Textbooks

1. Andrew S. Tanenbaum and Maarten van Steen, Distributed Systems: Principles and Paradigms, 2002, Prentice Hall (ISBN: 0130888931).

2. George Coulouris, Jean Dollimore and Tim Kindberg, Distributed Systems: Concepts and Design (Third Edition), 2001, Addison-Wesley (ISBN 0-201-619-180).

3. W. Richard Stevens, Unix Network Programming: Networking APIs, Vol. 1, 1998, Prentice Hall, (ISBN: 0-13-490012-X).

4. W. Richard Stevens, Unix Network Programming Vol. 2: Inter-process Communication, 2nd Ed., 1999, Prentice Hall (ISBN: 0-13-081081-9).

5. Andrew S. Tanenbaum, Modern Operating Systems (Second Edition), Prentice Hall (ISBN: 0130313580).

6. Avi Silberschatz, Abraham Silberschatz, Greg Gagne, Applied Operating System Concepts, John Wiley & Sons (ISBN: 0471365084).

7. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Concepts (6th Edition), John Wiley & Sons (ISBN: 0471417432).