Course Website

http://www.cs.pitt.edu/~jacklange/teaching/cs2510-f15

Course Description

The course provides a broad and in-depth study of to main fundamental principles and abstractions for the design and engineering of distributed systems. A distributed system consists of a collection of autonomous computers linked by a communication network and equipped with distributed system software. The computers interact with each other in order to achieve a common goal. The software enables these computers to coordinate their activities and to share the system hardware, software, and data. In contrast to a network, where users are aware of the location and main attributes of the computing, storage and communications components of the
system, a distributed system appears to its users as a single, fully integrated and coherent computing facility.

The first part of the course focuses on the architectural foundations and design principles of distributed systems, namely modularity, communications, naming, synchronization, concurrency, atomicity, consistency and replication, fault-tolerance and security. The second part of the course covers key systems that use principles and paradigms of distributed systems, including distributed files systems, distributed object-based systems, distributed coordination-based systems and distributed web-based systems. The last part of the course explores the emerging areas of data-intensive computing, with a focus on distributed system architecture that support cloud computing and Internet-scale services and applications.

**Learning Objectives**

Upon successful completion of the course, students are expected to:

- Gain an understanding of the fundamental principles, building blocks and techniques behind the design of distributed systems.
- Develop the ability to reason about distributed algorithms, including their correctness and performance.
- Gain understanding of the nature of failures and the security challenges faced by distributed systems programs and applications and explore appropriate solutions to tolerate failure and achieve security.
- Gain understanding of common distributed systems paradigms and clearly elucidate their benefits and limitations.
- Gain practical experience and insight into the design and implementation of distributed systems and applications through project design and implementation, using standards communications models primitives, programming and resource management paradigms and data management primitives.

**Main Topics**

The main topics covered in the course include:

- **Part I - Fundamental Concepts, Abstractions and Design Principles**
  - Introduction and Background
    - Definition, Goals and Basic Concept
  - Communication Architecture and Models
    - Layering, Interfaces and Protocols
    - Remote Procedure Call
    - Remote Object Invocation
    - Message and Stream-Oriented Communication
Processes and Computing Models

- Threads
- Client-Server Computing Model
- Code Migration

Naming and Addressing

- Names, Identifiers and Addresses
- Name Resolution Design and Implementation
  - Domain Name System and X.500
- Locating Mobile Entities
- Referencing and Unreferencing Objects

Synchronization

- Clock Synchronization and Logical Clocks
- Global State
- Election and Mutual Exclusion Algorithms
- Distributed Transactions

Consistency and Replication

- Data-Centric and Client-Centric Replication
- Distributed Protocols
- Consistency Protocols

Fault Tolerance

- Basic Concepts and Design Issues
- Reliable Client-Server Communications
- Reliable Group Communications
- Distributed Commit Protocols
- Failure Recovery - Checkpointing and Message Logging

Security

- Basic Concepts and Design Issues
- Cryptography and Secure Channels
- Access Control, Firewalls and Secure Mobile Code

Part II - Distributed Systems Paradigms

- Distributed Object-Based Systems
- Distributed File Systems
- Distributed Web-Based Systems
Part III - Emerging Trends in Distributed Systems

- Cloud Computing and Data-Intensive Processing
  - Computing as a Utility
  - Data Centers Architecture and Deployment
- Virtualization in the Cloud
- Data Processing on Large Clusters
  - Higher-Level Languages for Large-Scale Data Processing
    - MapReduce
  - Large Tables and Data Placement in Cloud Computing
    - Hadoop, PigLatin, Hbase and Big Table

Course Prerequisites

The course assumes basic knowledge of operating systems and systems programming:

- At a minimum, the course requires a successful completion of an undergraduate operating systems course, and knowledge of UNIX and C.
- Ideally, completion of an advanced undergraduate operating systems course.

Course References and Reading Material


**Reading Assignments:** Selected readings will be assigned throughout the semester and posted regularly on the course website.

**Introductory OS References:** Operating systems background material is well-covered in several text books:


**Unix Programming Reference:**
Office Hours

- Tuesday: 2:00 p.m. – 3:00 p.m.
- Thursday: 2:00 p.m. - 3:00 p.m.
- Additional hours by appointment

Email to the Instructor & TA: To ensure quick and timely response to email, the subject line of any correspondence must begin with ‘CS2510: Mail Subject’.

Important Dates

- Project Due Dates: As stated in the project description document, for each project.

Course Evaluation

Students are encouraged to attend class regularly, read the assigned reading material and participate in class discussions. The final grade will be based upon the following weights of midterm and final exams, project assignments, homework problems and reading assignments. The weights of the individual projects will vary slightly by the difficulty of the project:

<table>
<thead>
<tr>
<th></th>
<th>Midterm Exam</th>
<th>Final Exam</th>
<th>Project Assignments</th>
<th>Homework Problems, Reading and Programming Assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight (%)</td>
<td>20%</td>
<td>30%</td>
<td>30%</td>
<td>20%</td>
</tr>
</tbody>
</table>

The midterm and final exam will be in-class, closed-book exam. The midterm will cover all material up to the exam date. The final exam will cover will emphasize material covered since the midterm.

The homework assignments will combine textbook-like questions as well as hands-on experimental exercises and small programming assignments. There will be no more than five homework assignments.

Two projects will be assigned in this course. Successful completion of each project requires the submission of a design and a final report, in addition a tar file containing the developed software, if applicable, as specified by the project. The final report carries the most weight, and will be graded based on the correctness and completeness of the implementation, analysis and clarity of the presentation. More information about the project component of the course will be made available of the course web site.
Students are expected and strongly encouraged to actively participate in class discussions.

**Exam and Makeup Policy**

- Students are expected to take their exams on time and as scheduled by the instructor. Students who are unable to take an exam due to extenuating circumstances should contact the instructor immediately.
- Failure to notify the instructor prior to missing an exam will automatically result in a zero for the exam.
- Students with legitimate reasons for missing a scheduled exam are required to schedule a makeup exam at the earliest convenient time.

**Homework and Project Policies**

- Students are expected to check the course webpage regularly for announcements, class schedules, lecture notes, homework assignments, reading assignments, and other related course material.
- Homework and Project assignments must be turned in at the start of the class period on the date that they are due. Typically, homework is due one week after it is assigned unless otherwise mentioned.
- Students who are unable to attend the class may drop their homework, prior to the scheduled class time, in the instructor's mailbox in Sennott Square or at the front desk of the Computer Science Department.
- Unless a valid reason is provided and permission is granted by the instructor, prior to the due date, late submission will NOT be accepted.
- No homework will be accepted after the solution is posted on the website.
- For a regrading consideration, students must return their assignment with a note explaining their concern to the teaching assistance, no later than two weeks after the day the assignment was returned. If the issue is still not resolved, students must bring the matter to the attention of the instructor soon thereafter.

**Academic Integrity**

Students are expected to comply with the University of Pittsburgh’s Policy on Academic Integrity.

- Unless explicitly stated otherwise, assigned homework and projects in this course are to be carried out independently by each student. Students are encouraged to discuss homework and project assignments, as long as the discussions are limited to gaining understanding of the problem statement, acquiring related background or exploring general approaches to the solutions. Any other form of collaboration, including sharing homework and project solutions with other students is considered cheating.
All acts of cheating will be reported to the appropriate University authority, and **all involved parties will fail the course.**

Students are strongly encouraged to carefully read, understand and abide by the Academic Integrity Code for the School of Arts and Sciences.

**Disability Resources and Services**

If you have a disability for which you are or may be requesting an accommodation, you are encouraged to contact the instructor and Disability Resources and Services, 216 William Pitt Union (412-648-7890 or 412-383-7355(TTY)), as early as possible in the term.

**Religious Observances**

In order to accommodate the observance of religious holidays, students should inform the instructor by email, within the first two weeks of the term, of any such days which conflict with scheduled class activities.