CS 425 (ECE 428) - Distributed Systems

Instructors:

This course has been taught by a variety of CS and ECE professors. Recently, it has been taught by Professors Vaidya, Borisov, and Gupta.

Prerequisites:

Officially, the prerequisite is one of ECE 391 or CS 241, especially the networking portions.

When to Take It:

This class requires strong programming skills and a good understanding of asynchrony in programs; exceptional students may find it possible to take it at the same time as ECE 391 or CS 241. This class is offered every semester.

Class Content:

CS 425 is an overview of Distributed algorithms and Distributed Systems concepts. The class starts off with a discussion of different system models which will be used throughout the semester - namely, synchronous and asynchronous networks. After this, the class covers failure detection, timing and ordering issues (in the context of multicast), and continues to discuss the impossibility of consensus in an asynchronous system if even 1 process might crash. From here, many different topics are covered, such as distributed hash tables, distributed mutual exclusion algorithms, leader election, concurrency control (transactions), byzantine fault tolerance, replication, distributed shared memory, and various approaches to consistency, partition resilience, and availability, to name several; some major themes throughout the course are Fault Tolerance and scaleability (peer-to-peer designs). If time remains, other topics may be covered.

Work:

The lectures tend to move pretty fast and cover a lot of material. The class has two midterms and a final. The class also has several homeworks and 2 or 3 MPs. The homeworks are not given very often, and often will necessitate a bit of review, but otherwise are fairly simple. The MPs are only required for students who choose to select the four credit hr version. This is highly recommended because this is where a good portion of the learning (and work) for the class comes in. MPs are done in groups of 2, so finding a good partner is a great asset. The MPs usually take about a month, and are on topics such as reliable, causally-ordered multicast and distributed hash tables. The MPs require socket programming and some understanding of the basic OS utilities such as how processes work and how to use system calls. The overall workload for the class varies based on whether there is currently an MP - the workload is pretty light when there is no MP going on, but MPs take a serious amount of time. Unlike earlier classes such as CS 225 and ECE 190, MPs often require careful planning and cannot be implemented in a few nights.

The MPs had a bit of an overhaul in Fall 2012 when Professor Gupta taught the class again. The professors all borrow MPs from one another so the same set of MPs may be reused. Rather than having separate MPs and tight specifications, the MPs built on each other to create a distributed processing framework similar to Hadoop. Any languages were allowed, so if you and a partner can agree on a high-level language you can avoid dealing with very low-level details (there were groups using C/C++, Java, Python and Go in Spring 2016). However, falling behind early on would make later MPs more difficult; while code was released for an MP when it was over, it's much easier to understand your own system than someone else's. This organization makes the MPs more challenging but also very rewarding, and gives a good sense of the challenges associated with working in distributed systems.

Life After:

Students who enjoy this class may want to consider CS 438 (Communication Networks). This course prepares students to design distributed systems such as peer-to-peer networks or cloud-style services, so this class is great preparation for working on internet or cloud applications.