Comp 533: Distributed Systems

Bulletin Description

Prerequisite, COMP 431/530. Permission of the instructor for students lacking the prerequisite. Applications, design, implementation, and performance of distributed abstractions – software abstractions that that are layered above the network and are supported by the operating system, programming language, and middleware.

General Course Info

Term:	Spring 2018
Department:	COMP
Course Number:	533
Section Number:	001
Time:	TR 2 – 3:15pm
Location:	Room FB007
Website:	http://www.cs.unc.edu/~dewan/533/current/index.html

Instructor Info

Name:	Prasun Dewan
Office:	FB150
Email:	dewan@cs.unc.edu
Phone:	5906123
Web:	http://www.cs.unc.edu/~dewan
Office Hours:	TR 16:15 – 17:15

Teaching Assistants: ???

Textbooks and Resources

There is no text book covering the topics of this course. Notes, PPT slides, and videos on the covered material will be accessible from the course home page. They will not be posted on Sakai, which will be used however for submitting programs and quizzes.

Course Description

This course will provide a practical overview distributed systems. It will be driven by a series of implementation-based projects

We will cover the design, implementation, performance, and applications of abstractions for sharing information among distributed processes. These sit

between the network layer and application and include: distributed shared memory, byte and object communication, remote procedure call, and broadcast/multicast sessions. Some of the general issues we will address are naming; synchronization; extensibilty; and routing and consistency of broadcast/multicast messages in the face of failures.

We will overview but not focus on any of the specific applications of distributed abstractions such as the Web, distributed data mining, distributed data bases, file systems and distributed collaborative systems, which are covered in other courses. Our focus will be on foundational concepts applicable to all of these application areas. These concepts will be introduced as layers that sit above the OS and networks.

The assignments and lectures will have two related goals. The first is to expose you to standard implementation of the covered abstractions in Java environment and other distributed systems and discuss how these abstractions can be used in a realistic programs. The assignments addressing this goal will involve the use of only Java abstractions. The second is to expose you to the implementation of these abstractions. I have developed a Java-based teaching/research testbed, called GIPC (for Generalized Interprocess communication), that provides implementation of these abstractions. In the assignments addressing this goal, you will replace parts of this extendible system to understand implementation concepts. The result of the replacement exercises will be a system that is more sophisticated in many ways than the state of the art, and you are likely to use it for programming future distributed programs.

Target Audience

The target audience is students wishing to learn in depth the nature of practical distributed systems.

Prerequisites

The pre-requisites are knowledge of object-oriented programming, data structures, and threads. UNC Comp 401, 410, 530 and 431 cover these topics.

Goals and Key Learning Objectives

At the end of the course, you will have a basic understanding of how distributed software works, the potential uses of this software, the design and implementation space of distributed abstractions; the performance of alternative distributed abstractions; how to run experiments (on destktop and virtual Cloud machines) to measure performance, and implementation of

key fault-tolerant distributed algorithms for achieving consensus As a distributed program is also a parallel program, you will also sharpen your understanding of threads and thread synchronization. Because of the emphasis on assignments, you will gain practice with the use and implementation of advanced software engineering concepts such as layers, generic types, factories, and abstract factories.

Course Requirements

You must attend lectures and participate in class discussions, take quizzes, implement assignments, and take a midterm and a final exam.

You must submit the source code of your program (with pledge signed) and videos demoing executions of the program on test data.

Examinations are closed book, notes and program listings; computers and collaboration are not allowed either.

Key Dates

Midterm: Class time, Tuesday Mar 6th (in class)
Final: 12pm on Monday, May 7 (in class)

Class Lecture Structure

We will use a combination of flipped and regular lectures. A class may (partly or completely):

- *Live-classes*: involve the lecturer presenting new material for which there is no preexisting recordings.
- *Class-work of choice:* give students freedom to listen to recorded lectures (using headphones), implement assignments, take open-book quizzes, working with classmates of their choosing, based on recorded and live quizzes.
- *Discussions:* involve discussions (based on the live and recorded lectures and the quizzes) that allow students to raise issues and answer questions.

Diaries

Students will be required to edit on Piazza a running diary of *at least one activity* in that class period that is visible to instructors. A reported activity can include:

- Some specific progress you made on an assignment or quiz,
- An issue you faced that prevented you from making some intended progress.
- Distillation of one specific aspect of something you learned through class discussion, class lecture, or recorded lecture.

The intention here is to (a) help you crystallize at least one concept in each class, (b) give those too shy to participate in oral class discussions a way to

get some credit for class activities, (c) give me basis to evaluate your written skills, which I am often asked to do when writing recommendations.

A diary for a particular lecture should also indicate (specific) questions you have answered in class that day along with the (specific) answers you gave. The rationale for this is to hone your skills in taking oral exams. You can report incorrect and partially correct answers.

Here is an example of diary showing the desired format:

Date: 1/11

Activity: Learned that a selector can wait for a specific set of NIO operations to complete.

QA:
Question: Why allow multiple pending operations?
My answer: To increase concurrency without requiring thread overhead.

Please follow this exact format as we may automatically process some aspects of the diaries.

You can report diary entries only for the days you attend class or have been excused by the instructor. If you have been excused, just write "excused absence" for the activity.

Date: 1/11

Activity: Excused absence

Grading Criteria

A grade will be assigned based on performance on homework programming assignments, quizzes, class participation, and exams. Here is the breakdown:

Midterm	25%
Final	15%
5 Individual Regular Assignments	40%
Regular Collaborative Open-Book Quizzes	10%
Class Activities based on Diary	5%
Q/A based on diary	5%
Extra Credit Serialization Individual Assignment	12%
Extra Credit Serialization Open-Book Collaborative Quizzes	3%

The extra credit assignment and quiz gives shy students a chance to compensate for the points these lose for class discussion.

I reserve the right to apply a 5% fudge factor to give consideration to other factors such as extraordinary class participation.

Course Policies

Students are requires to attend each class unless there are extenuating circumstances. If such circumstances occur, you should access the class material posted for missed classes, and contact classmates to become aware of the announcements that were made.

Assignments are due at 11:55pm on each specified due date. Homework assignments will be penalized 10% for each class session late and will not be accepted more than ten days late. Quiz submissions will not be accepted after the due date.

Honor Code

You are encouraged to discuss the assignments with fellow students but required to write/code the solutions/programs individually. Also you cannot use solutions from previous offerings of the course. You cannot make incorrect diary reports. Not following these rules is a violation of the honor code policy

Course Schedule

If possible, a schedule of topics covered by the course organized by course date or week number.

1. Course Information and Introduction to Distributed Systems
2. Byte Communication
3. NIO
4. Threads
5. RMI Abstraction
6. Thread Coordination
7. IPC Design Space
8. NIO Producer Consumer Interaction
9. RPC Implementation
10. Object Serialization
11. Fault Tolerant Distributed Consensus
12. Semi Synchronous Replication
13. Centralized Consensus
14. Paxos: Distributed Consensus

Disclaimer

The professor reserves to right to make changes to the syllabus, including project due dates and test dates. These changes will be announced as early as possible.