Comp 533: Course Info

Instructor: Prasun Dewan (FB 150, dewan@unc.edu)
**Course Home Page**

Comp 533: Distributed Systems

**Course Overview**

This course will provide a practical overview of 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, extensibility, and routing and consistency of broadcast/multicast messages.

We will overview but not focus on any of the specific applications of distributed abstractions such as the Web, distributed data mining, distributed databases, 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 platform, called GOPC (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 exercised 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.

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 and how to run experiments to measure performance. 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.

**Pre-requisites:** The pre-requisites are knowledge of object-oriented programming, data structures, threads, operating systems, and networking. UNC Comp 401, 410, 530 and 431 cover these topics, respectively.

**Textbook:** There is no textbook covering the topics of this course. Notes, PPT slides, and videos on the covered material will be accessible from the course home page.

**Grades:** A grade will be assigned based on performance on homework programming assignments, quizzes, class participation, and exams. Exams will constitute 45% of the grade (midterm 20%, final 25%), homework assignments 40%, and class work (quizzes, class participation) 15%. There will be a midterm and a final. I reserve the right to apply a 5% fudge factor to give consideration to things such as good class participation and stellar programs. You are encouraged to discuss the assignments with fellow students but required to write/compile the solutions/programs individually. Also you cannot use solutions from previous offerings of the course. Not following these rules is a violation of the honor code policy.

**Syllabus Handout in UNC Format**

# First Assignment

First assignment is ready!

Tests if you are ready for the course

| Byte Communication | PPT PDF | YouTube-1 | YouTube-2 | YouTube-3 | Mix 2 | Mix-3 | http://rox-xmlrpc.sourceforge.net/niotut/ | Distributed Non-Blocking Halloween Simulation |
ASSIGNMENTS NATURE: HALLOWEEN SIMULATION

Make Beau Anderson’s 401 Halloween implementation distributed in multiple ways
**USE NON BLOCKING I/O**

<table>
<thead>
<tr>
<th>Distributed Non-Blocking Simulation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing non distributed simulation</td>
<td>Java NIO</td>
</tr>
<tr>
<td>OS + Network Concepts</td>
<td></td>
</tr>
</tbody>
</table>
Use Synchronous RMI

<table>
<thead>
<tr>
<th>Distributed RMI-based Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing non distributed simulation</td>
</tr>
<tr>
<td>Java Object Serialization</td>
</tr>
<tr>
<td>Sockets</td>
</tr>
</tbody>
</table>
GIPC Asynchronous RPC

- Like a PL course or Programming Course
- Focusing on Distributed Programs
- Compiler Course?

Distributed GIPC RPC-based Simulation

- Existing non distributed simulation
- GIPC RPC

- Java Object Serialization
- Java NIO
# Custom Serialization

<table>
<thead>
<tr>
<th>Distributed GIPC RPC-based Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing non distributed simulation</td>
</tr>
<tr>
<td>Custom Serialization (a la Web Services)</td>
</tr>
</tbody>
</table>
CUSTOM RPC COMPONENT

- Distributed GIPC RPC-based Simulation
  - Sync RPC
  - GIPC RPC
  - Custom Serialization (a la Web Services)
  - Java NIO
- Existing non-distributed simulation
GROUP RPC

- Distributed GIPC Group RPC-based Simulation
- Existing non-distributed simulation
- GIPC Group RPC
- GIPC RPC
- Custom Serialization (a la Web Services)
- Java NIO
Assignment Submission

Upload software to Sakai

Create video demonstration shareable with dewan@cs.unc.edu and yicheng@live.unc.edu

Post link to video as private post to instructors or as a public post
## SOFTWARE

### Downloads

<table>
<thead>
<tr>
<th>Software Description</th>
<th>Download File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beau Halloween Simulation (Eclipse Project, uncompress and link to oeall17.jar or oeall22.jar)</td>
<td><code>beau_project.zip</code></td>
</tr>
<tr>
<td>Coupled Halloween Simulations (Eclipse project, un-compress and link to Beaj project and objecteditor)</td>
<td><code>CoupledTrickOrTreat.zip</code></td>
</tr>
<tr>
<td>ObjectEditor</td>
<td><code>oeall17.jar</code></td>
</tr>
<tr>
<td>Latest ObjectEditor (jar)</td>
<td><code>oeall22</code></td>
</tr>
<tr>
<td>GIPC (Github library project)</td>
<td><code>GIPC</code></td>
</tr>
</tbody>
</table>
## Grade Distribution

<table>
<thead>
<tr>
<th>Category</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exams (Midterm, Final)</td>
<td>45% (20, 25%)</td>
</tr>
<tr>
<td>Programming Assignments (Home work)</td>
<td>40%</td>
</tr>
<tr>
<td>Class work (Quizzes, Class Participation)</td>
<td>15%</td>
</tr>
<tr>
<td>Fudge Factor (Class participation, other factors)</td>
<td>5%</td>
</tr>
</tbody>
</table>
Not all episodes in which you are involved in shared Google Doc (URL in Piazza announcement)

Identify each class participation episode in depth, giving the date of each episode and the people involved.

E.g.,
1/12, Thu:
Q: What is your favorite color?
Arthur: Blue
Q: What is your quest?
Arthur: To find the holy grail?
1/19, Thu:
Q: What is your favorite color?
Robin: Green, no, Blue

Will serve as distributed collaborative notes
Class Participation: Sakai Quizzes

Begin Assessment
"Class Discussion - 1" for COMP533.001.SP17

This assessment is due Sunday, 2017-Feb-12 12:00 AM.
There is no time limit for this assessment.
You can submit this assessment an unlimited number of times. Answers from previous attempts will not be available within the assessment during subsequent attempts. Your highest score will be recorded.

Copy text from google doc to Sakai answer before due date (resubmission will probably delete previous text)

Monthly submission and grading
**Why Class Participation**

- Instructor needs feedback
- Learn to speak in public (I will ask questions every one can answer such as what was interesting in the material you studied)
- Oral “exam” over whole semester
- Retain material better
LEARNING RESOURCES

No textbook!

Alternatives?
CLASSWORK RECORD

Shared discussion doc and quiz answers will help for exams
POWERPOINT OF SLIDES

ISSUES IN MESSAGE PASSING

Process/Thread

Message Queue

message³

Port (Mailbox)

message²

message¹

Process/Thread

Reliable message delivery?
In-order message delivery?
Port access (and message routing)?
Operations?
Synchronous vs asynchronous?
Blocking vs non blocking?
Buffering of messages at queue?
Location of communicating threads?
Can escape out into unsynchronized or no audio mode (WPS Office on Android will play synchronized audio)
ISSUES IN MESSAGE PASSING

- Reliable message delivery?
- In-order message delivery?
- Port access (and message routing)?
- Operations?
- Synchronous vs asynchronous?
- Blocking vs non-blocking?
- Buffering of messages at queue?
- Location of communicating threads?
Recorded YouTube Videos

ISSUES IN MESSAGE PASSING

Play 2X, rewind, pause, fast-forward to match understanding pace

Youtube video generated from PPT Recordings, does not allow slide-based browsing

PPT modes allow slide-based browsing but requires downloading PPT
Office Mix

Issues in Message Passing

- Reliable message delivery?
- In-order message delivery?
- Port access (and message routing)?
- Operations?
- Synchronous vs asynchronous?
- Blocking vs non-blocking?
- Buffering of messages at queue?
- Location of communicating threads?

Slide markers
SLIDE-BASED BROWSING
Long pauses, you may know the answer

Cannot hear student answer

Audio is not the fastest way to get information, specially when studying for an exam

Recordings of live lectures with q/a rather than 15 minute lessons

Can fast forward

You can get a clue from my answer
DESIGN SPACE OF STUDY MODES!

- PDF without audio and animations
- Slideshow with synchronized recording
- Slides with unsynchronized audio
- You Tube video
- Office Mix
- Shared Google Docs
- Quizzes with Answers
- Assignment Solutions
PACING YOURSELF

A variety of sources with different amounts of information

Each source can be browsed at your own pace
WHAT DO WE DO IN CLASS?

Live Lecture?

<table>
<thead>
<tr>
<th>Value</th>
<th>Without resources</th>
<th>With resources</th>
</tr>
</thead>
</table>

Without resources

With resources
WHAT DO WE DO IN CLASS?

- Homework
- Deep thinking done solo?
- Limited discussion with classmates?
- Will do debugging and some trouble shooting
What do we do in class?

- **Quizzes:** Test that you understood support material

- **One hour testing, puts pressure**

- **May not have discipline to access material**

- **No time pressure, open book, collaborative**
WHAT DO WE DO IN CLASS?

Discussion: of quizzes, recorded material

Experience matters less in answering questions if homework done

Approach assumes recorded material exists
Comp 734: Distributed Systems

Course Overview
This course will provide an implementation-oriented study of distributed systems. Some of the topics covered will include inter-process communication, group communication, synchronization, remote procedure call, peer to peer and centralized sessions, fire-walls, causal broadcast, atomic broadcast, scalability, fault tolerance, replication, and transactions/concurrency control. These are foundational concepts, which are becoming particularly relevant with the emerging areas of cloud computing and distributed games. These concepts will be introduced as layers in a general distributed infrastructure. Your projects will implement new layers and provide alternative implementations of some of the existing layers. When implementing a layer, you will act both as an application programmer, using abstractions of the layers below, and a systems programmer, defining abstractions for the layers above. The number of lines of code required by each layer will be relatively small; however the compositions of these layers will be complex.

The main difference between this course and a distributed programming/theory course is that it will address the design and implementation

First-year grads and several undergrads took this course
Class discussion will further clarify material
**Getting Help**

- Can discuss solutions with each other at a high level
- Not at the code level
- Sharing of code is honor code violation
- Can help each other with debugging as long as it does not lead to code sharing
- Assignments may contain solution in English (read only if stuck)
### Getting Help and Class Discussion

We will be using Piazza for class discussion and getting help. The system is highly catered to getting you help fast and efficiently from classmates and instructors. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. If you do not get a response within a day or two on Piazza, please send email.

Before posing a question, please check if this question has been asked before. This will reduce post clutter and reduce our burden. Repeat questions will be ignored by the instructors.

Piazza allows anyone to respond. So if you see a question that you think you can respond to, please do so, as that will reduce my burden and help you "teach" your fellow students.

This will be a form of class participation that will be noted when I allocate my fudge points!

Hope it works well

If you have any problems or feedback for the developers, email team@piazza.com.