# CIS 630: Distributed Systems
Fall 2008

## Logistics

*Prerequisites*: CIS 415 (Operating Systems) or equiv.; 429/529 (Computer Architecture)

**Classes and Final time slot**

Class Day/Time: Tuesday/Thursday, 2:00pm - 3:20pm, Pacific 11
Final Meeting Day/Time: Monday, Dec. 8, 1:00pm - 3:00pm, Pacific 11

> **__Note__: The final meeting day will be for project presentations. We only have one term exam before Thanksgiving break.*

**Instructor**

Dr. Matthew J. Sottile
Office: 203 Deschutes Hall
Phone: 346-0375
E-mail: matt@cs.uoregon.edu
Office Hours: MW, 3:00 to 4:30pm, or by appointment.

## Overview of Course

A distributed system is a set of programs that executed on a set of computational resources that are connected by a network, communicating and coordinating their activities by passing messages. Many important modern technologies are distributed systems, taking advantage of distributed computational resources to reach levels unachievable in isolated systems in terms of scalability, capacity, availability and redundancy. This course will cover the theory and models of distributed systems design and implementation with relevant case studies. The focus of the course will be: interprocess communication, processes and threads, issues that arise in concurrent systems, distributed file systems, distributed time synchronization, and distributed shared memory. The general properties of distributed systems that we will study are:

- The distributed architecture of the system
- Networked interaction and logical interaction
- Shared state and coordinated operation
- Correctness issues arising from concurrent interacting processes

The content of this course is based on that taught in previous years with updates based on current trends and new technologies found in present distributed systems.

# Academic Integrity

### University

The University of Oregon's student conduct code explains the rules for cheating and plagiarism. It also discusses the penalties. Please read it. For more information on student academic integrity, see the brochure on academic integrity put out by the UO Dean of Students, and the Student Conduct web page.

The student conduct code allows an instructor to impose an appropriate sanction for a student found guilty of academic dishonesty, up to and including an N or an F. I will impose an N or an F for any such offenses in this course.

### Course

For this course, some assignments may be categorized as group work. For these assignments, you will work in a group that works together to produce one solution. Any assignment or exam not categorized as group work must be done individually. You are encouraged to generally discuss problems with other groups or students, but you may never use some other groups or student's solution or code directly in any way (ie, you shouldn't look at someone else's solution to the problem). The use of sources (ideas, quotations, paraphrases) must be properly acknowledged and documented in any written assignment.

# Grading Policy

- 5% Problem Sets (easy)
- 10% Programming Exercises
- 10% Research paper summaries
- 25% Term paper and presentation
- 25% Term exam
- 25% Term project and presentation

Problem sets are not graded, but recorded as turned in/not turned in. They are for your benefit, so you are encouraged to not skip them.

### Late policy

*On time*: `1.0 * Grade`
*Late*: `max(0.0, 1.0 - (days late * 0.25)) * Grade`

Interpretation: 25% off the graded score of the assignment for each day late, with no credit after the 3rd day. This does not apply to the term project or term paper. Those must be turned in on time and will receive a zero if late.

# Miscellaneous

**Test make-up**: We only have one exam, and it is scheduled for the Tuesday before Thanksgiving. I will not offer make-ups for anyone who chooses to take an extended Thanksgiving vacation that spans that Tuesday – it's an official school day, so you are expected to be here. Emergencies that prevent you from taking the exam are obviously an exception.

**Collaboration**: The project is a group project. Papers, exams, problem sets, and programming exercises are individual. It is obviously acceptable to ask questions of your peers ("Why does my code core dump?"), but sharing code/answers is unacceptable. You are encouraged to ask questions and talk amongst yourselves while doing the work as part of the learning process, but you should formulate the final answer and deliverable individually. You should be able to recreate your entire solution in isolation.

**Programming help**: While you are on your own to write your programs, if you reach a point of total agony and are completely stuck, I am willing to help you debug and fix the program. Use my office hours for this. I am not forgiving if you turn in a project or programming problem that is completely broken with the excuse that "I couldn't figure it out". If you didn't approach me prior to turning it in, I can only assume that you left it to the last minute and didn't have time to make it work. Just ask – I can probably help dislodge you and get you going in the right direction again.

**References**: When writing papers or reports, please rely on literature from textbooks, journal articles, and conference proceedings. While it is tempting to derive all of the content for your write-ups from sources such as Wikipedia, you should realize that Wikipedia derives its content from sources like books, journals and conferences. Often Wikipedia articles cite these. As a graduate student, you should get into the habit of diving deeper into topics and chasing down references like this to read the original sources and fully understand them instead of just working with the synopsis that the web provides. A good use of Wikipedia is as a starting point in your search, not the endpoint.