CIS 630: Distributed Systems
Fall 2007 Syllabus

Meeting Times: T-Th, 12-1:20pm, Deschutes 200

Instructor: Yannis Smaragdakis, Deschutes 245, 541-3463473
yannis@cs.uoregon.edu
Office Hours: by appointment—available most late afternoons and early evenings

Course Outline:

We will discuss distributed system abstractions and their implementations. The core of the course contains concurrent programming (threads and synchronization), explicit inter-process communication (network programming, middleware), and a variety of other distributed systems topics (time coordination, distributed file systems, transactions, distributed shared memory, etc.).

Textbooks: I will be passing out copies of research papers and occasionally notes. Yet, if you need to refer to a second source for the same topics, the following are good background books:

Distributed Systems: Concepts and Design, Coulouris, Dollimore, Kindberg, Addison-Wesley, 2005.
(Not required, but strongly recommended.)

Distributed Systems, Tanenbaum and VanSteen. Prentice Hall.
(A good textbook for the subject. New version of the good but out-of-date “Distributed Operating Systems” by Tanenbaum.)

Multithreaded Programming with Pthreads, Lewis and Berg. Prentice Hall.
(Excellent book on multithreading and systems issues—not too Pthread-specific at all.)

Pthreads Programming, Nichols, Buttlar, Farrell. O’Reilly.
(Fairly good and inexpensive Pthreads manual.)

Operating Systems Concepts, Silberschatz and Galvin, 6th Ed. Addison-Wesley.
(Operating Systems background textbook.)

Exams: All exams are closed-books, closed-notes. Current schedule:
Midterm: Thursday October 25
Final Exam: in regular exam period

Grading: 40% exams (20%+20%)
10% paper presentation/review
10% (easy!) homeworks
30% projects (15%+15%)
10% extra credit/class participation

Homework and Assignment Due Times:

Homeworks are due by class time on the due date (either by email before class or as hard-copy in class). Programming assignments (by email) are due by midnight (11:59pm) of the due date.

Late Penalty Policy (only for programming assignments):

This late penalty policy applies **only to projects, not to homeworks**! 5% penalty for each day late, up to 5 days (25%), weekends together with Monday count as a single day, holidays do not count as late days.

You can think of the "late penalty" policy as an "early credit" policy: the real deadline is five days after the posted deadline, but if you submit before that, you get early credit.

Honor Code: Familiarize yourselves with the UO Student Conduct Code. I follow the standard process in all academic misconduct cases. This means that even if we both agree on a sanction, I have to report it to the Director of Student Conduct.

Other Resources:

Newsgroup: uo.classes.cis.cis630 on the CoC news server (news.cs.uoregon.edu). We may just use an email list, though.

Web page:
`http://www.cs.uoregon.edu/classes/07F/cis630/`

Caveats:

Test make-ups: You miss an exam, you lose the points! If you have a serious reason to miss a scheduled exam (e.g., sickness, or serious reason to travel) make sure you tell me *in advance* of the exam. After the exam is over, if you have not taken it and you have not told me why you didn't take it, you are out of luck! Emergencies that also prevent you from notifying are an exception, of course.

Collaboration: No collaboration is allowed in projects and homeworks. All work submitted in this course must be your own and produced exclusively for this course. Nevertheless, there is a difference between seeking technical help (e.g., why does my program crash here?) and giving/receiving large sections of source code. You should feel free to ask questions and give answers about technical matters. It is fine to share or re-use short code segments (say, 10 reasonably formatted lines) that can be considered "standard infrastructure" for a task (e.g., opening sockets, or creating threads). You are encouraged to do this through the newsgroup/forum. The rule-of-thumb for collaboration is: if you feel comfortable posting the info to the newsgroup/forum (for everyone, including me, to see), then it is ok. Otherwise, it is not.

Debugging with me: I will spend time helping you debug your programs, but this typically will happen only when you are totally lost. If you come to me with a problem, make sure you have spent a lot of time trying to resolve it. Also, make sure you have a setup where I can quickly reproduce the problem, and ideally even a much simplified example where the problem still exhibits itself.

Project descriptions: For *any* major departure from the project requirements (different deliverable, very different mechanism, etc.) consult with me first! Do not assume that the project is just about testing whether you can program! It is not! The project is about implementing what I ask for. If I say "use RPC" and you instead decide to use sockets, you will lose many points, even if you are an expert programmer, and you have a great deliverable, and you spent twice the time that everybody else did coding your project to perfection.