CS 4210: Advanced Operating Systems
Summer 2004 Syllabus

Meeting Times: M-W-F, 2:40-3:50, CCB 101

Instructor: Ada Gavrilovska, CCB 226C, 404 385 2029
ada@cc.gatech.edu
Office Hours: M, W 1:00-2:00 (tentative—make suggestions!) or by appointment

Teaching Assistant: Abhishek Singh (abhi@cc)

Course Outline:

We will discuss operating system abstractions and their implementations. The core of the course contains concurrent programming (threads and synchronization), inter-address communication, and an introduction to distributed operating systems. Other topics may be added, especially in conjunction with related programming projects. Such topics include memory management (especially virtual memory subsystems), dynamic libraries, "avant-garde" kernel architectures (microkernels, exokernels), and file systems (e.g., log-structured file systems).

Textbooks: This is a seminar course—no textbook is strictly required, as I will be passing out copies of research papers. Here are some suggested background books:

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.
(I assume you already have it from previous courses!)

Exams: All exams are closed-books, closed-notes. Current schedule:
Midterm: Monday June 14

Final Exam (current tentative date: Wed Jul 28, 11:30-2:20)

Grading: 55% exams: 25%, 30%
45% programming assignments and homeworks (roughly, 35% programming assignments, 10% homeworks)

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 *both to me and to the TA*) 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: The Institute Student Conduct Code is printed on pages 336-339 of the GT General Catalog. You and I are expected to abide by it. I do not deal with cases of academic misconduct in person (the Office of the Senior Associate Dean of Students will be notified).

Other Resources:

Newsgroup: git.cc.class.cs4210 on the CoC news server (news.cc.gatech.edu).
Web page:
`http://www.cc.gatech.edu/classes/AY2004/cs4210_summer/`

Caveats: I have had a lot of problems over the years and most of these take a lot of my time to fix. Here are some general warnings:

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 *well 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 are an exception, of course, but "I was stuck in the dentist's office—he said he'd see me at 1 but didn't until 3" is not an emergency!

Collaboration: No collaboration is allowed in projects and homeworks. Specifically, showing someone your source code is not allowed, unless you only show small fragments (say, 10 reasonably formatted lines) and these only concern low-level technical issues. Collaboration on how to solve specific technical problems is allowed (and encouraged) through the newsgroup. The rule-of-thumb for collaboration is: if you feel comfortable posting the info to the newsgroup (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.