University of Pennsylvania / CIS Department
Spring 2012
CIS-505 -- Software Systems

Time and Place

Time: Tuesdays and Thursdays, 1330-1500
Place: Skirkanich Auditorium

Course Staff

Lecture (CIS-505):

- **Matt Blaze**
  Email: blaze at-sign cis.upenn.edu.
  Office: Levine 611
  Office Hours: Tuesday 1630-1730

Lab / Teaching Assistants:

- TBA

Description

This course is an advanced introduction to the theory and practice of modern software systems. We will focus on fundamental concepts of distributed systems and the design principles for building large scale computational systems. The course will involve written assignments, two programming projects, a midterm, and a final examination.

Prerequisites

A good undergraduate operating systems course. Working knowledge of C/C++ programming in the Unix/Linux environment. Reading knowledge of Java.

Textbook, Readings, Web Site

The required text for the course (available from the Penn Bookstore either now or soon) is:

- Andrew S. Tanenbaum and Maarten van Steen, *Distributed Systems: Principles and Paradigms (2/e)*. Prentice Hall, 2007.

Most of the readings will be from the textbook, but we will also refer to other sources from time to time. In particular, we will study technical manuals, system source code, and research papers. The additional readings will generally be available in online form; check the course web page for URLs or pointers.

An additional book may be helpful in completing the projects:

- W. Richard Stevens and Stephen A. Rago. *Advanced Programming in the UNIX Environment (2/e)*. Addison-Wesley Professional. 2005. (available online, but you probably want a printed

copy).

The course web site (with the current version of this text, updated office hours, schedule, etc) can be found online (right here!) at:

- http://www.crypto.com/courses/fall12/cis505/

Course-work, Grading, Policies

Your grade will be based on two in-class exams (a midterm, worth 15% of your grade, plus a cumulative final, worth 35% of your grade) two substantial team software projects (worth 15% and 25% of your grade), and several homeworks, due throughout the semester (collectively worth the remaining 10% of your grade).

Details on the programming projects will be distributed separately shortly. Note that they will require a working knowledge of the C programming language. If you don't have experience programming in C in the Unix/Linux environment you may feel lost and under-prepared. Note that the first homework will given out next week, and will be due about a week later.

There will be a course mailing list, to which you were (or will soon be) automatically subscribed, assuming you have properly registered. Announcements for this course will be sent to the mailing list, and reflected, where appropriate, on the course web page.

Late work will be accepted for 50% credit for up to 24 hours past the deadline. It will not be accepted *at all* after 24 hours. This is a firm policy applied bureaucratically and without mercy.

Homework / Software Projects (last updated 1/10/12)

A significant part of this course involves team software projects, due throughout the semester. Collectively, the projects (with milestones) will comprise 40% of your course grade. There will also be several shorter homework assignments (to be done individually), involving programming, paper reviews, and exercises, which in total will be worth 10% of your grade.

Information on the projects will be posted in this space.

Preliminary Schedule (based on previous years)

Note: The schedule and topic list is subject to change! See the course web site (this page) for the latest schedule.

Note that the assignments and readings are not complete beyond the first week of the course. The course is being revised, and will change from previous years.

Lecture notes will be posted and made available *after* each class meeting.

Date	Topics	Readings, Homework, etc. (Readings to be done prior to the class session with which they are listed).
1/12/12	Introduction; course logistics; What are software systems?	
1/17/12	Processes, threads, and synchronization Unix system calls and primitives	Tanenbaum, Chapter 1-3 Unix "man" pages for read(2), write(2), fork(2), execve(2), execl(2), wait(2), exit(2)