Advanced Unix System Administration
Spring 2007

Course vitals

Computer Science 98/198
WTh 6-7 PM, 373 Soda Hall
2 units, CCN 26389 (lower division) and 26608 (upper division)
Course web site: http://www.ocf.berkeley.edu/sysadmin-class/2007-spring/advanced/

Facilitator

Steven Luo <sluo+decal@ocf.berkeley.edu>
Office hours: W 10-11, Th 11-12, Th 5-6 in the OCF Lab (Heller Lounge, MLK)

Prerequisites

Familiarity with the Unix command line. Some prior system administration experience is recommended, though not required; if you have no prior experience, come talk to me for some things you can do to become familiar with basic sysadmin tasks. See me or email me if you are not sure whether you meet these requirements.

Course philosophy, conduct and grading

The goal of this course is to give you some idea of what's going on "behind the scenes" of your system. This should allow you to set things up without excessive reference to step-by-step howtos and debug problems when they arise without resorting to voodoo debugging. (Not that voodoo debugging isn't occasionally useful, it's just not usually the best place to start.) As such, lectures will be mostly theoretical (though peppered with plenty of examples), while the homework assignments and projects will be designed to allow you to apply what you've learned to practical situations.

Short homework assignments will be handed out roughly once a week; these are intended to reinforce the ideas from lecture and give you a chance to explore them a bit in "real" systems. The assignments will be graded on a P/NP basis, and will together make up 20% of your grade.

There will be two end-of-chapter projects, one longer than the other, in which you will be given some tasks and asked to set up (and perhaps maintain for a bit) a system that can perform those tasks. Depending on class enrollment and the computing resources available to us, these may or may not be group projects. Together, these will make up 40% of your grade.

The last few weeks of the class will be taken up mostly by the final project. For this, you will be expected to design, set up, and maintain a fairly large-scale networked system

that does something useful. What exactly this system is supposed to accomplish is up to you, though I will need to approve your proposal. (Some ideas include a computing service offering shell accounts, mail, and web hosting; and an e-commerce system with separation between the front end web server and the back end database and order processing systems.) You will then be asked to work with someone else's system and try to expose flaws in the design and implementation. This project is 40% of your grade.

While the homework and the projects will be related to the material we discuss in lecture, you will frequently be asked to do things that we haven't gone over how to do in class. You are expected to be able to read documentation and figure out how to do these things yourself. Learning where to find relevant documentation and how to read it (whether well-written or unclear/incomplete/out-of-date) is an important skill (art?) for sysadmins. Of course, you're welcome to ask for help if you end up stuck or confused, and I will provide hints where I think the existing documentation is unhelpful, but you are expected to make an effort to figure out how to do it yourself first.

Course Outline

I. Operating systems from a practical perspective (3 weeks)

   A. Overview of the kernel, kernel space vs. user space
   B. Memory management
   C. I/O facilities
   D. Processes
   E. Shared libraries
   F. Startup and shutdown on Unix systems
   G. Instrumentation and debugging facilities
   H. Performance tuning considerations

   Those who have had CS 162 (Operating Systems) should find all of this review.

II. Networks and network applications (2.5 weeks)

   A. The OSI model
   B. How Ethernet networks work
   C. Encapsulating network protocols in Ethernet
   D. ICMP messages
   E. IP networking
   F. TCP and UDP links
   G. Packet filtering, NAT, and content inspection
   H. Layering application protocols on TCP or UDP and IP

I. Structure and operation of DNS

J. Common application protocols: Telnet, HTTP, FTP, SSH, SSL, etc.

III. Security (2.5 weeks)

A. Basic principles of security

B. Common vulnerabilities and methods of exploit

C. Securing local systems against attack

D. Securing networks against attack

E. Forensics and post-attack analysis

IV. Final project; additional topics depending on time and class interest

There is a bit of room for the schedule to slip, if needed; otherwise, I will talk briefly about other topics that the class wants to hear about. I'll save about 20 minutes a class for questions and discussion on the final project.