Advanced Unix System Administration

Lecture 1
January 31, 2007

Steven Luo
<sluo+decal@OCF.Berkeley.EDU>

Administrative Stuff

- CCN: 26389 (lower div), 26608 (upper div)
  - Make sure you're in the right section, for the right number of units!
  - If waitlisted for CS 98, and are a junior/senior, take CS 198 instead
- Office hours: W 10-11, Th 11-12, Th 5-6, by appointment
- Make-up section: F 5-6 (see me if interested)

Administrative Stuff

- Grading: 20% HW (P/NP), 40% chapter projects, 40% final project
  - Final project a chance to “get creative” and build something that you're interested in
- Prerequisites
  - “Prior system administration” doesn't need to mean more than having set up and played with your own Linux/BSD box for a bit
  - You do need to be able to read documentation

Course Outline

- OS stuff – 3 weeks
  - Aiming for a practical perspective
- Networking – 2.5 weeks
  - A tour of a TCP/IP stack from bottom to top
- Security – 2.5 weeks
- Final project and additional topics
  - Room for the schedule to slip if needed
  - What do you want to hear?

Kernel Space, User Space

- Kernel
  - The component at the core of the OS
  - First part of the OS to load
  - Provides central services – process management, memory management, etc.
  - Runs privileged on the CPU
  - Usually also provides device drivers, network stack, and other hardware-related or performance-critical functions

Kernel Space, User Space

- User space
  - Most applications and services run in user space
  - Some core parts of the OS (init, hardware detection, etc.) do run in user space, usually with kernel cooperation
  - Runs unprivileged on the hardware to provide better isolation and fault tolerance

Kernel Space, User Space

- Communication between kernel and user space
  - Kernel exposes functions to userspace via syscalls
  - Invoked via an interrupt or via special processor support
  - Requires a context switch, which is slow
  - Other mechanisms such as shared memory
  - What about /proc? read() is a syscall too!

Kernel Space, User Space

- Microkernels
  - Not that much stuff absolutely has to run in kernel space
  - Advantages to keeping code out of kernel: easier development, more flexibility, security
  - Disadvantages: more overhead and more abstraction = slower code
  - Distinction between microkernels and traditional “monolithic” kernels is blurring