Advanced Unix System Administration

Lecture 1
September 15, 2008

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

Administrative Stuff

- CCN: 26205 (lower div), 26204 (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: for now, W 4-5:30 in 251 LeConte, by appointment

Administrative Stuff

- Grading: 20% HW (P/NP), 30% midterm project, 50% 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
- Please don't just disappear from class!

Course Outline

- Tentative only – probably need to trim
- OS stuff – 5 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
  - 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

Standards and “Standards”

“The nice thing about standards is that you have so many to choose from.”
- Andrew Tanenbaum

Standards and “Standards”

- Unix has a complicated and very fragmented history
  - The really fast summary:
    - Created in the early 70s at Bell Labs by Thompson and Ritchie
    - Berkeley added various customizations to form BSD
    - Commercial vendors created a massive number of subtly incompatible versions forked from AT&T releases (System III, System V) or BSD in the 1980s

Standards and “Standards”

- Fragmentation = nightmare for ISVs
- Backlash against fragmentation caused vendors to try to standardize
  - But they couldn't agree on what to standardize on, and so formed competing standards – SVR4, Unix International, OSF/1
  - Meanwhile, other OSes gained ground
  - Eventually, the standards groups agreed to merge

Standards and “Standards”

- The Single Unix Specification (POSIX, ...)
  - Specifies behavior seen by applications, not by administrators
    - Possible to have a UNIX [tm] without having any Unix-derived code
    - Edicts have significant influence on OS behavior
  - Tries to include everything from BSD and System V and the other standards
    - Makes the standard a big mess!
  - Occasionally implements new interfaces where none of the existing interfaces are satisfactory

Standards and “Standards”

- Pseudo-standards
  - BSD – behavior was never written down in a standards document, but code base influential
  - SVR4 – the merger of AT&T and Sun's Unix; somewhat standardized by SVID; code base influential
  - Linux, *BSD, Solaris, ... – newer systems, especially the openly developed ones, are influential in establishing new interfaces