COMS W4118 — Operating Systems I

Prof. Steven M. Bellovin

smb+4118@cs. . .

`http://www.cs.columbia.edu/~smb`

What’s an Operating System?

- Traditional definition: “traffic cop”
- Manage multiple streams; prevent interference
- Protect resources from unauthorized access or abuse

Resources

- CPU time
- Main memory
- Disk space
- I/O bandwidth
- Network bandwidth

Multiple Users

- Individual users of the system
- Specialized applications (web servers, print spoolers, etc)
- System components

Fairness

- Give each “user” a fair share of resources
- What’s a user?
- What’s fair?

Security

- Prevent users from gaining more than their fair share
- Prevent users from accessing resources they shouldn't be able to
  - Other users' files
  - Other user's network traffic
  - The operating system's files and traffic
- But — security must be flexible. The OS must provide mechanisms that can implement many different security policies

Structure of the Course

- Lectures
- Approximately five homework assignments, all with programming and non-programming components
- Midterm, final

Prerequisites

- CSEE W3827 — Fundamentals of Computer Systems
  - Understanding of computer system architecture
  - Registers, cache, virtual memory, I/O, DMA, disks, etc
  - Interrupts
  - Basic knowledge of what assembler language is
- COMS W3157 or W3101
  - Understand how to use “make”, the C compiler, etc.
- The C programming language

Grading

Midterm	20%
Final	30%
Homeworks	50%

Yes, I curve.

Textbooks

- Tanenbaum, *Modern Operating Systems, 2nd Edition*, Prentice-Hall, 2001, ISBN 0-13-031358-0. (Required)
- Bovet and Desati, *Understanding the Linux Kernel, 3rd Edition*, O'Reilly, 2006, ISBN 0-596-00565-2. Note: get only the 3rd Edition, since it describes the Linux 2.6 kernel. (Required)
- Ritchie and Thompson, "The UNIX Time-Sharing System", *Communications of the ACM* 17:7, July 1974. `http://doi.acm.org/10.1145/361011.361061`
- Optional reading: Organick, *The Multics System; An Examination of its Structure*, MIT Press, 1972. On reserve in the Engineering library. Also see `http://www.multicians.org`
-

Logistics

- For grading issues, approach the TA within two weeks; if you don't receive a satisfactory answer, contact me.
- For issues relating to *this class*, email smb+4118@cs. . .
- That lets me auto-sort class-related mail and keep better track of things
- My office hours are posted; I try to note (too frequent) changes because of my travel schedule

TAs

- Yingbo Song <ys2242@. . .>
- Ye Ilho <iy2110@. . .>
- Japinder Chawla <jsc2127@. . .>

Lectures

- I prepare slides for each class, and upload them shortly before class time
- Well, occasionally they're uploaded shortly after class. . .
- Because the class is being recorded for CVN, you'll be able to watch any lectures you've missed.
- General access to the videos starts after the add/drop period ends

Homeworks

- A lot of it. . .
- As noted, approximately five homework assignments
- Homeworks are designed for practice, teaching, and evaluation
- Homeworks must be submitted electronically by the start of class
- Homeworks received later that day lose 5%, the next day 10%, two days late 20%, three days late 30%; after that, zero credit
- Exceptions granted only for *unforeseeable* events. Workload, day job, etc., are quite foreseeable.

Programming Assignments

- All programming assignments *must* be done in C.
- C++ and Java are *not* acceptable
- If you don't know C, you probably should not be in this class
- This is a necessary consequence of the course design, which involves working with the Linux kernel
- Most of the programming assignments will be done in groups

Groups

- You're encouraged to form your own groups of three students within the next week or so
- I'll fill in the blanks; in general, I'll try to avoid mixing CVN students, undergraduates, and graduate students
- I'll probably rearrange the groups during the semester, putting together students who do not contribute appropriately.

Co-operation versus Dishonesty

- Discussing homework with others is encouraged
- All programs and written material *must* be individual work unless otherwise instructed
- Please use appropriate file permission mechanisms to protect your homework. (Looking at other people's work is not allowed.)
- Zero tolerance for cheating or "outsourced homework"
- See the department's policy on academic honesty: `http://www.cs.columbia.edu/education/honesty`. I will assume that you have all read it; you are in any event responsible for its terms and provisions.

Responsibility

- You’re all adults
- You’re all responsible for your own actions
- If there’s something missing, you have to tell me

Practical Focus

- This is not a pure academic-style OS course
- We’ll be working with the Linux kernel
- Most of the homeworks will involve modifying the kernel

Working with Kernels

- *Very* different from ordinary applications programming
- Must fit into the existing structure
- You spend more time reading programs than writing them

Debugging Kernels

- Debugging kernels is hard
- Long build/boot/test cycle
- Hard to get debugging output
- Many “Heisenbugs”

Virtual Machines

- We’ll test our changes using *virtual machines*
- Virtual machines — an entire machine is simulated as a single application
- Much more detail in a few days

The CLIC Lab

- All programs and kernels *must* run on the CLIC machines
- Programs that don’t compile *on those machines* or kernels that don’t boot receive zero credit
- You need a CS account to use CLIC; see `https://www.cs.columbia.edu/˜crf/accounts/`
- You will be assigned particular CLIC machines for virtual machine use
- Some of the CLIC machines are for in-person use; others can only be accessed remotely
- New policy: no food or drink in the CLIC lab

Operating System Components

- What's in an OS?
- Is there a neat boundary?
- What clearly isn't part of the OS?

The Kernel

- Fundamental piece of the OS
- At a minimum, uses hardware protection mechanisms to isolate other pieces
- Provides communication paths linking different pieces of the system
- Provides interface to user application programs

Scheduler

- Allocates CPU time
- Handles scheduling priorities
- Ensures fairness

Memory Management

- Allocates memory to user applications
- Utilizes virtual memory to extend real memory space
- Ensures separation of memory between different applications

File Systems

- Layer files, directories, etc., on top of raw disk
- Enforce permissions

Device Drivers

- Hardware-dependent code to talk to I/O devices
- Provide type-specific common interface to higher layers
- Isolates hardware-specific complexity
- Part of the OS to prevent improper access to devices (other reasons as well)

More?

- Is the **`login`** command part of the OS?
- What about the **`mount`** command?
- The shell?
- The C compiler?
- A text-formatting package?
- Different people draw the line in different places
- One useful metric: Who can replace the component with a different one?

This Course

- We’ll cover the different pieces of an operating system
- We’ll see how they’re implemented in Linux
- We’ll learn how to change them

Reading Assignment

- Read Chapter 1 of Tanenbaum
- It’s the subject matter for the next few classes
- Reading and other homework assignments will be on the class Web site and on Courseworks