What is an Operating System

Andy Konwinski

CS61CL

Dec 2, 2009

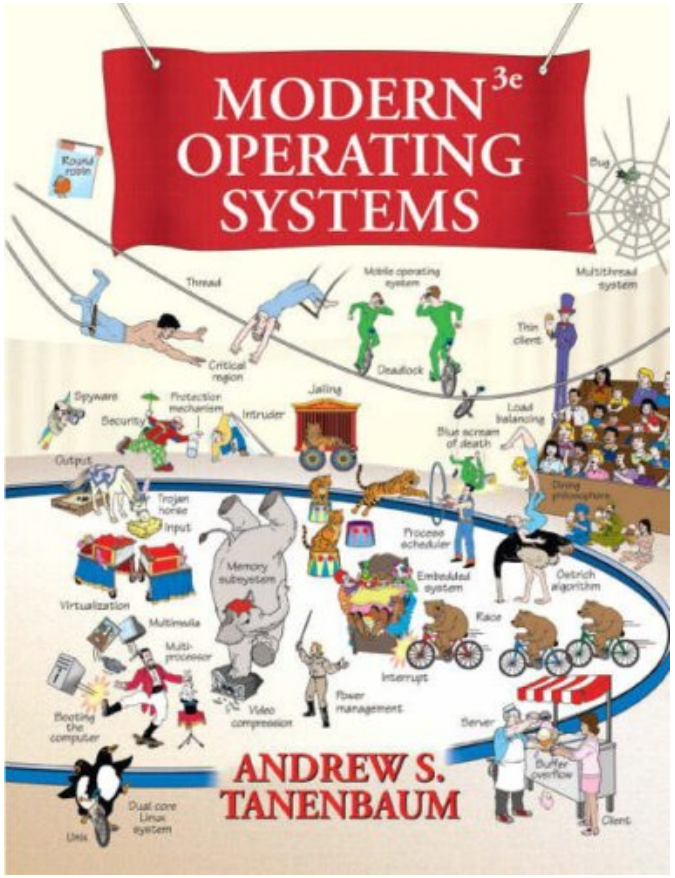
Lecture 13

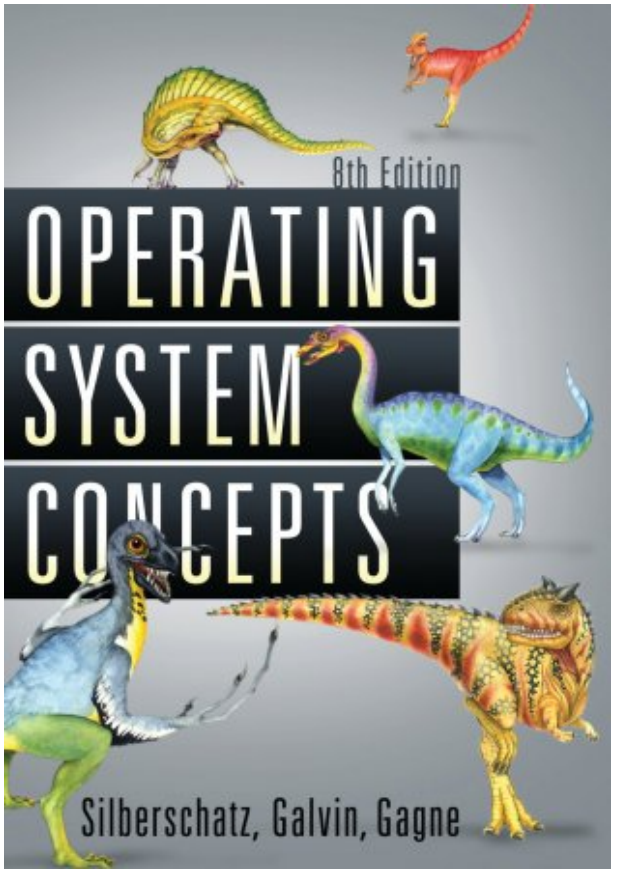
Today

- What is an operating system
- Dual mode operation: kernel vs. user mode
- Current trends and issues

What is an Operating System

- Resources
  - Textbooks I like:



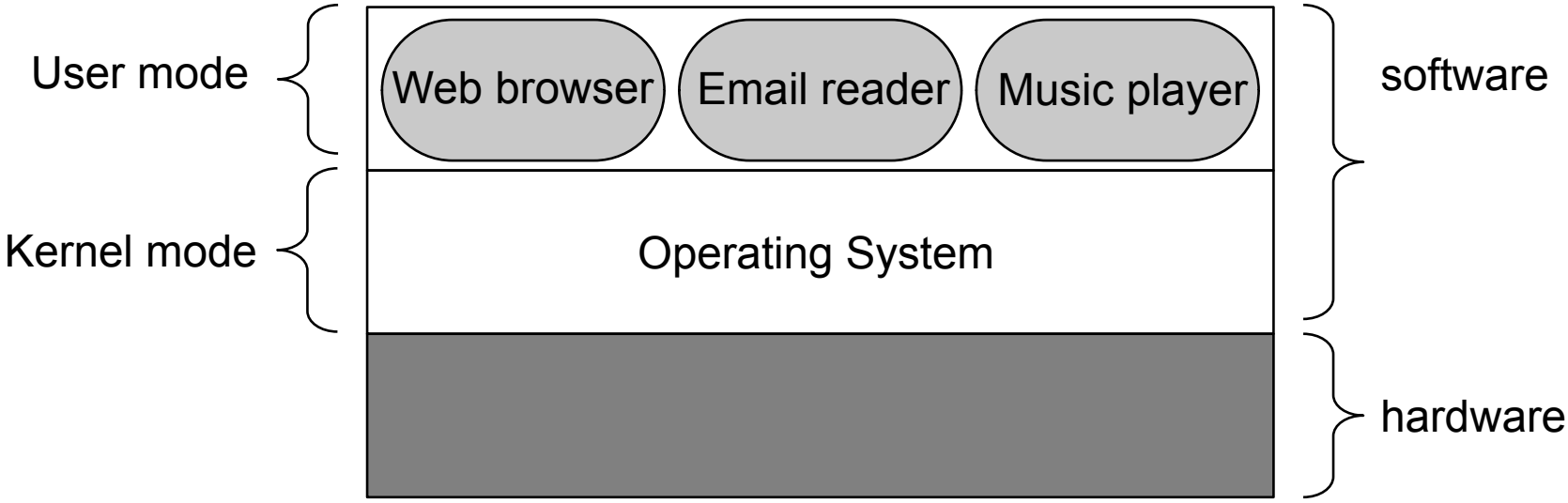


What is an Operating System

- No single all encompassing definition
- Used to be an actual person, an “operator”
  - You were operator for your CAL16 processor
- General definition:
  - A layer of software that provides user programs with a simpler, cleaner, model of the computer and handles the messy job of managing the resources.

What is an Operating System

- Where the OS fits in

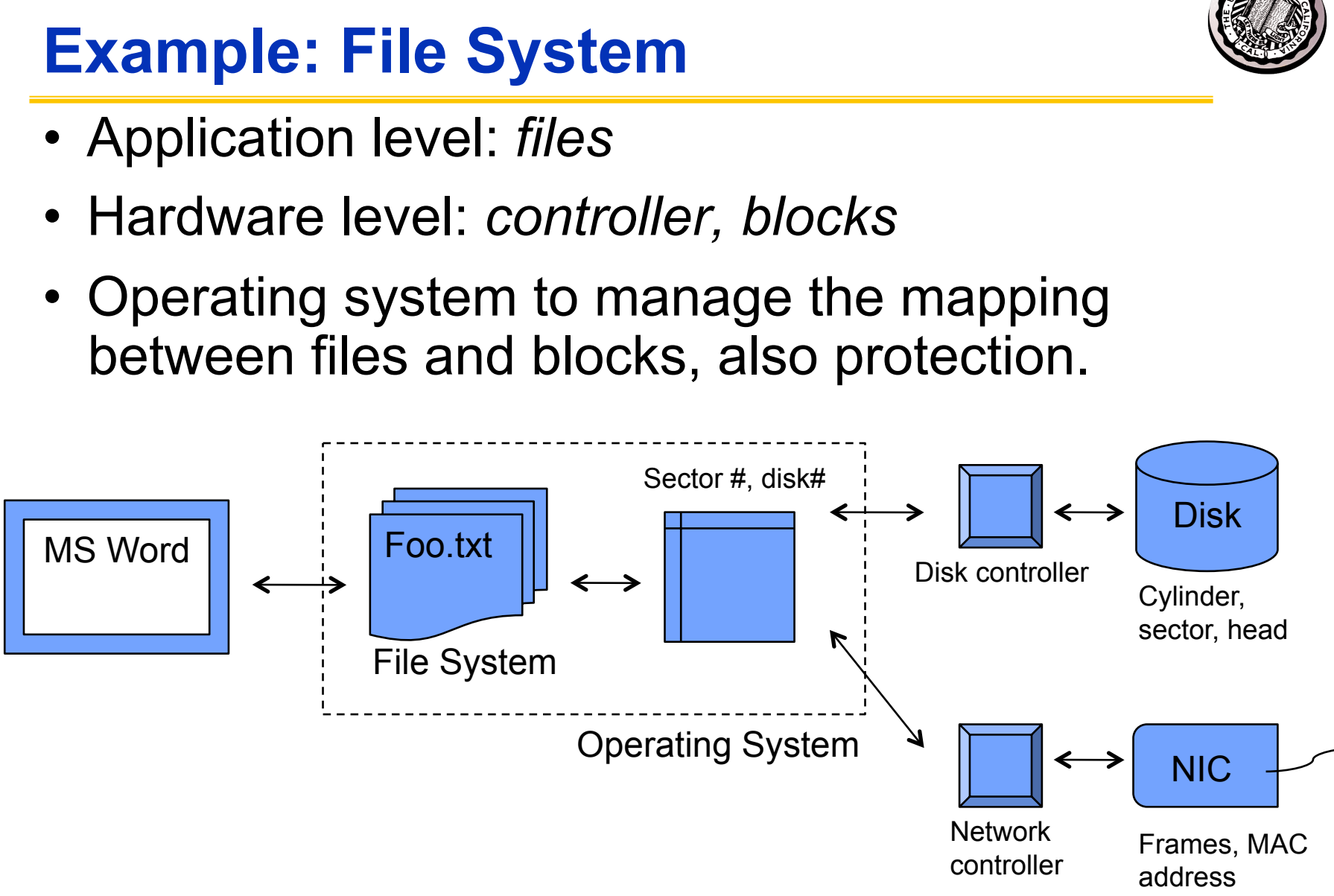


(modified version of fig 1-1, tanenbaum, pg2)

Part 1: Clean abstractions of HW

- Hardware is messy (e.g. assembly lang., interacting with a device)
- Application developers want useful high level abstractions

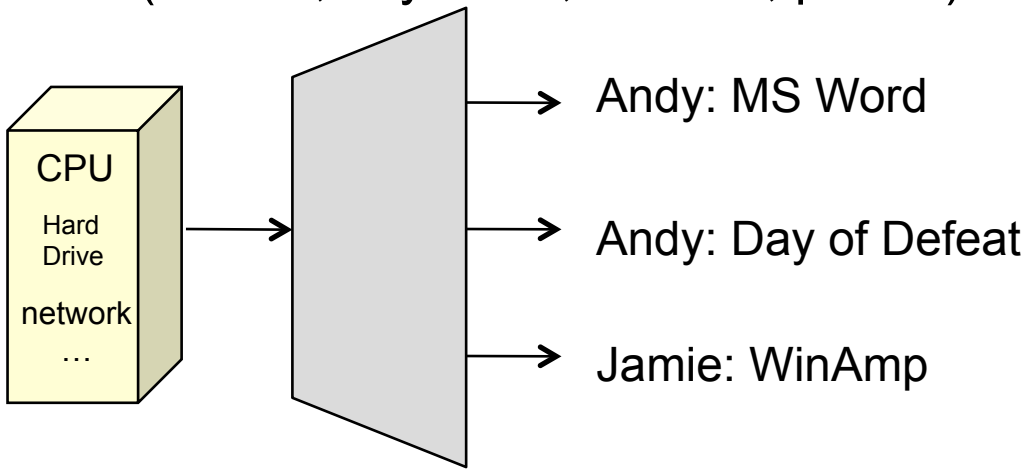
Example: File System



- Application level: *files*
- Hardware level: *controller, blocks*
- Operating system to manage the mapping between files and blocks, also protection.

Part 2: Resource manager

- Multiplex one set of hardware resources between apps/users
  - CPU/Memory/cache
  - I/O devices
    - » Communication (network cards, hard drive)
    - » Human I/O (mouse, keyboard, monitor, printer)



Part 2b: fault/performance isolation

- Resources should be shared **fairly**
- Isolation between users (processes)
  - Fault isolation: when one program crashes, it should not cause others to crash
  - Performance isolation: e.g. if spotlight runs, my Quicktime movie shouldn't skip.

Example: virtual memory

- Each application sees continuous, nearly infinite, mem. address namespace
- Hardware provides: finite memory, page table base register, TLB, disk
- Operating system: orchestrates.
  - manages page table entries (e.g. updating Valid bit when swapping), flushes the TLB when necessary, pages to disk, etc.

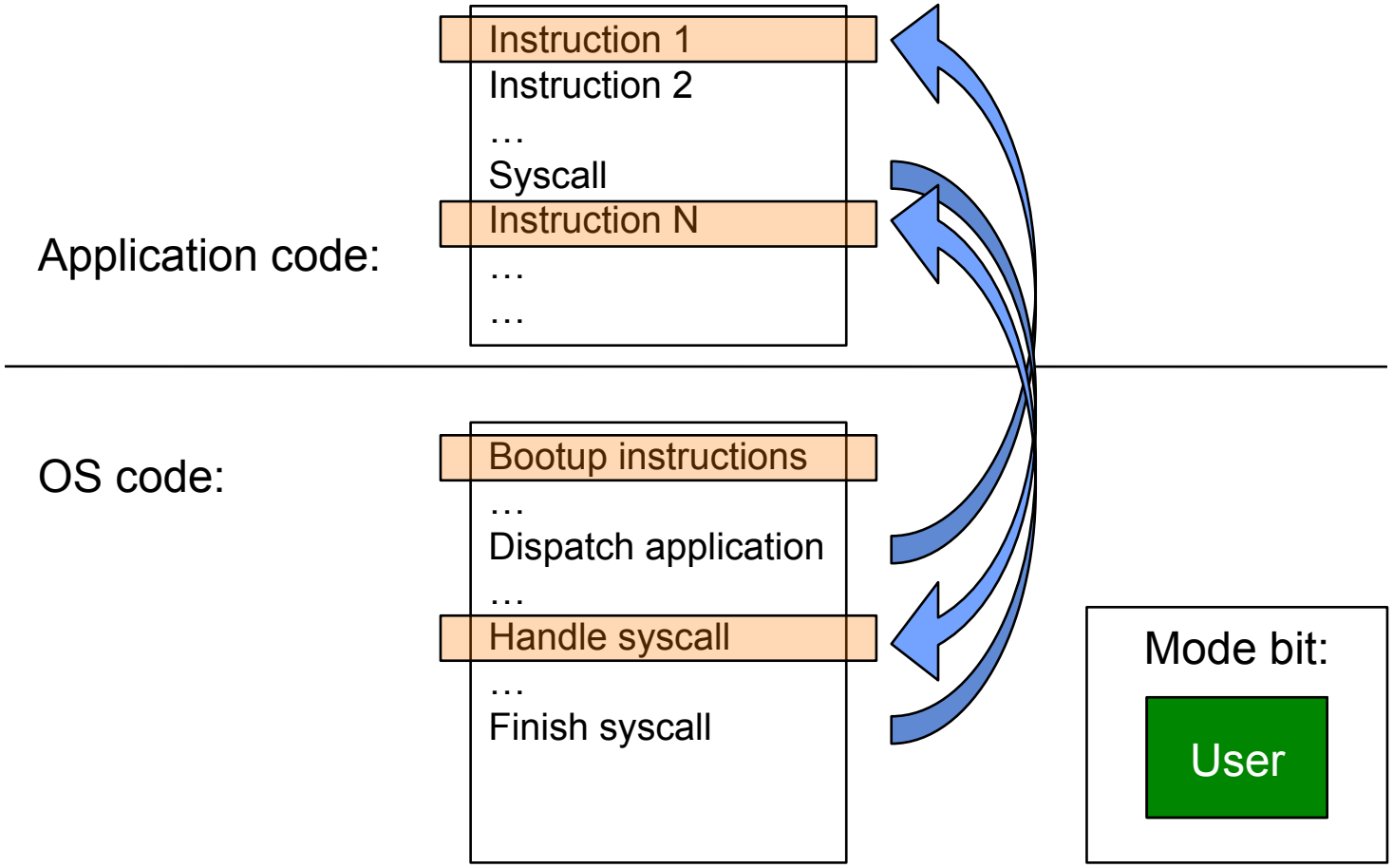
Administrative

- Midterm 2 back last week, regrades done
- This is final class
- Optional lab on threads
- Regrade requests for Midterm 2 due by end of day Friday
- Review lecture next week

Dual mode operation

- Hardware knows that an operating system will be used, and that it needs more privileges than application software.
- Hardware bit for user/kernel mode.
- Need kernel mode access for:
  - Accessing kernel data structures, e.g. list open files
  - Mapping device mem to main mem, e.g. graphics card
  - Kernel registers, e.g. page table offset
  - Privileged instructions, e.g. switch to kernel mode

Switching between User/Kernel mode



Interrupts

- Hardware interrupts
- Software-generated interrupts (called Traps)
  - System calls: user has OS do something on its behalf, `trap` or `syscall` instruction.
  - Exceptions: if privileged instruction called when in user-level, handled similar to a system call

Trends

- OS used to handle concurrency for us (time sharing), now applications are making smarter use of concurrency (threading)
  - ParLab
- Cloud computing
  - RAD Lab

Summary

- OS multiplexes hardware resources & provides clean abstraction for applications.
- Dual mode operation, interrupts, exceptions
- Parallel and cloud computing

- Take CS162 to pick up where we're leaving off and actually build an OS (and see more of me)!