Operating systems

Ch 1-2
What is an OS?

• For what computing environments?
An x64 processor is screaming along at billions of cycles per second to run the XNU kernel, which is frantically working through all the POSIX-specified abstraction to create the Darwin system underlying OS X, which in turn is straining itself to run Firefox and its Gecko renderer, which creates a Flash object which renders dozens of video frames every second because I wanted to see a cat jump into a box and fall over.

I am a god.
user 1

user 2

user 3

... user n

system and application programs

compiler

assembler

text editor

... database system

operating system

computer hardware
How does the main role of the OS differ between

• A standalone computer at home
• A terminal system (i.e., SunRay)
• An embedded system
• A multicore system
• A clustered system
Hardware protection

• What is there?
Problem 1.13

• Which of the following instructions should be privileged?
  – Set value of timer
  – Read the clock
  – Clear memory
  – Issue a trap instruction
  – Turn off Interupts
  – Modify entries in device-status table
  – Switch from user to kernel mode
  – Access I/O device
What takes time?

• How can the OS help to speed up things?
What components are there in an OS?
How to diagnose the OS?

• Tools available for different components?
OS-level programming

• How is it done?
  – Language?
  – Special libraries?
  – Special compilers?
  – Special development tools?
How is the OS structured?

- Kernel types?
- Where is it stored?
- Boot-strapping?
What are VM:s good for?