

## Lecture 3 Outline

**Reading:** White, §3, 4

---

1. Memory hierarchy
  - a. RAM
  - b. Floppy disks
  - c. Hard drives (including RAID)
  - d. Optical disks (CDs, DVDs, BluRay)
2. Other hardware
  - a. Input devices: keyboard, mouse, scanner, camera, haptic devices (touchpad, touchscreen, etc.)
  - b. Output devices: monitor, printer, speakers, haptic devices
  - c. Interfaces: USB, Firewire, SCSI, IDE, SATA
    - i. Interfaces to I/O devices: video and sound cards, network cards (Ethernet, FDDI, etc.)
3. How it starts: the boot process
  - a. Idea: execute small segment of code in memory (BIOS) that loads bigger program, that loads system (“boot-strap”)
  - b. When powered up, PC loaded with particular memory address
  - c. Address is in BIOS, which has program to run self-test, load information needed to read from secondary storage (more on this later)
  - d. Program then determines which secondary storage device to read from (on PCs, order can be set in BIOS; usually floppy, CD, hard drive)
  - e. Program reads in boot block (boot sector), which tells computer where to look for the operating system
  - f. Computer loads, runs operating system
4. What is an operating system?
  - a. Manages resources, runs applications, handles input and output
  - b. Provides interfaces to programs so they can access resources, advise OS on management
  - c. 3 biggies now: Windows, UNIX-like systems, MacOS X
    - i. Windows: many versions, oldest is DOS; current ones are Windows XP, Windows Vista, Windows 7
    - ii. UNIX-like: Linux (a UNIX clone) and its variants, the “BSDs” (FreeBSD, OpenBSD, NetBSD), Solaris
    - iii. MacOS X: based on FreeBSD (called “Darwin”), but with many extra layers and enhancements
5. Example 1: reading what you type
  - a. Interrupts and all that (IRQ, controller, vectors)
  - b. How the computer remembers where it was when the interrupt arrives
  - c. How the computer processes the interrupt
  - d. How the computer picks up where it left off
  - e. Nested interrupts
6. Example 2: writing a file to disk
  - a. Device drivers and direct memory access
  - b. Operating system checks request to write (file name legal, file does not exist or is writable, and so forth)
  - c. Operating system tells device driver where to put the file
  - d. Device driver interacts with disk hardware to copy file out; done in chunks (blocks)
7. Example 3: turning on a Plug-and-Play Windows based system
  - a. BIOS identifies devices by unique numbers burned into devices’ read-only memory (ROM)
  - b. Windows creates enumerators, one per type of device (ISA for hard drives and cards, port for keyboard and mouse, and so forth)
  - c. Windows allocates interrupts for each device based on enumerators
  - d. Windows then loads appropriate device drivers to use those interrupts