Announcements

- Reading Chapter 19
- Last day to request midterm #2 re-grades is Th
Booting the OS

- How does the OS get loaded and started?
  - Process is called booting
    - want to use the OS to load itself
    - but what loads the OS?
- ROM monitor
  - knows how to read from a fixed location on disk and jump into it
- Bootstrap program
  - knows how to load a program from the filesystem and jump into it
- Alternative:
  - put more info into ROM about booting
    - MAC OS has most of the info in ROM
    - hard to change OS without changing ROMs
Booting the OS (cont.)

- put info into ROM about booting
  - MAC OS has most of the info in ROM
  - hard to change OS without changing ROMs

- Network Booting
  - ROM knows how to request a boot packet from the network
    - once the packet is received, execute it
  - useful for systems without local disks
  - used by OS developers to ease edit/compile/boot cycles
Booting in GeekOS

- **PC Architecture**
  - Reads first sector on drive and then executes it
  - Hardware thinks it is a 16 bit 8088 processor at boot
    - Provides backwards compatibility

- **Boot Sector**
  - contains code to read
    - kernel.bin into memory
    - setup.bin into memory
    - uses bios to access drives
  - Includes a boot record to find kernel

- **Setup code**
  - Detects amount of memory
  - Moves processor to protected mode
  - Jumps to 32 bit code (and 32 bit mode)
  - Sets up initial kernel stack
GeekOS Booting Notes

- **Kernel and setup files**
  - Are normal files in whatever filesystem we have
  - Bootinfo record in boot sector tells how to find them
  - Must be in contiguous blocks on disk
    - A restriction in the boot sector code

- **Once booted**
  - Boot sector is ignored by main filesystem
  - Rest of disk is available to be used as desired

- **Have special utility to write boot sector**
  - Gosfs has a call GOFS_BootInfo
Swap Space

- **Where is swap space located?**
  - Is it a “normal” file in the filesystem?
  - Is it in a special location on disk?

- **“normal” file**
  - simple, just looks like a file
  - easy to change size
    - use normal tools
  - slow since it requires all of the filesystem overhead

- **separate disk partition**
  - faster
  - harder to change size (need a new partition)
Backups

- Disks can fail, so need to provide a way to copy them
- Need to plan for disasters too
  - What if the building burns down?

- Two types of backups
  - full backup (all of the data on disks)
  - incremental (data that has changed since last backup)
    - can mark changed files with a field
    - can use the date of the file compared to the last backup
      - permits several levels of backup
    - may want multiple levels of incremental (day, week changes)
Backups

- Does the system need to be shutdown for backups?
  - what if a file is moved during a backup?
    - it could get copied 0, 1, or 2 times.
  - easy answer is to shutdown the machine for backup
  - more typical setup:
    - Compute backup set
    - Backup files
    - Compute new backup set
      - Add any files that were missed
Security

- **security vs. protection**
  - protection provides a mechanism to control access to resources
  - security also includes external features such as users

- **security requires precluding unauthorized**
  - access to data
  - modification of data
  - destruction of data

- **several major types of security**
  - physical: must protect access to resource itself
    - if you have physical access to a machine, you can break security.
  - users: if a user gives away access (or info) computer security if useless
  - software: OS and system software must provide protection
Who do you trust?

- It’s easy to get paranoid
- Do I trust a login prompt?
- Do I trust the OS that I got from the vendor?
- Do I trust the system staff?
  - should I encrypt all my files?
- Networking
  - do you trust the network provider?
  - do you trust the phone company?
- How do you bootstrap security?
  - always need one “out of band” transfer to get going