Announcements

- Reading Chapter 13
- Midterm #2 is on Tuesday
  - Covers through Tu lecture
  - Can repeat info from first midterm
Access Times

- **Seek:** Move disk arm over appropriate track
  - Seek times vary depending on locality
  - Times are order of milliseconds

- **Rotational delay:** Wait until desired information is under disk arm
  - A disk that rotates at 10,000 RPM will take 6.0 ms to complete a full rotation
  - Improving only a few percent per year

- **Transfer time:** time taken to transfer a block of bits
  - Minimum transfer is one sector
  - Depends on recording density of track, rotation speed, block size
  - Achieved transfer rate for many blocks can also be influenced by other system bottlenecks (software, hardware)
  - Rates range from 2 to 40 MB per second
Disk Scheduling

- First come, first served
  - ordering may lead to lots of disk head movement
  - i.e. 1, 190, 3, 170, 4, 160 etc.
  - total number of tracks traversed: 863

- Shortest seek time first: select request with the minimum seek time from current head position
  - move head to closest track
  - i.e. 1,3,4,160,190
  - total number of tracks traversed: 189
  - potential problem with distant tracks not getting service for an indefinite period
Disk Scheduling

- **Scan scheduling** - read-write head starts at one end of the disk, moves to the other, servicing requests as it reaches each track
  - Consider example: 1, 190, 3, 170, 4, 160
  - If head starts at track 64 and moves towards 0, the ordering would be 4,3,1,160,170,190
  - Total distance 265

- **C-Scan (circular scan)**
  - disk head sweeps in only one direction
  - when the disk head reaches one end, it returns to the other
  - Consider example: 1, 190, 3, 170, 4, 160
  - If head starts at track 64 and moves towards 0, the ordering would be 4,3,1,190,170,160
  - Total distance 282
Disk Cache

- Buffer in main memory for disk sectors
- Cache contains copy of some of the sectors on a disk. When I/O request is made for a sector, a check is made to find out if sector is in the disk cache
- Replacement strategy:
  - Least recently used: block that has been in the cache longest with no reference gets replaced
  - Least frequently used: block that experiences fewest references gets replaced
Virtual Memory and File Cache

- Both need to contend for memory
- Possible solutions:
  - Fixed size allocation of buffer cache (i.e. 20% of memory)
  - Unified buffer cache and virtual memory system
    - All pages (memory and file buffer) compete for all of memory
    - Allows large processes or lots of file access as needed
Memory Mapped Files

- **Can treat files like memory**
  - Allows fast random access to files
  - Uses file cache to make operations fast

- **Interface**
  - Use mmap call to map file into memory (similar to open)
  - Use normal memory operations to access file (instead of read/write)
  - Use munmap to “close” file
Bad Blocks

- **Some blocks on a disk may not work**
  - could be bad from the start (when disk is installed)
  - could go bad during use

- **Two options to manage bad blocks**
  - disk drive maps the blocks to “replacement” blocks
    - special blocks that are held in reserve for this purpose
  - OS keeps track of where the bad blocks are located and avoids them

- **Replacement blocks**
  - can be located in tracks at one location, or around the disk
  - provide correct behavior, but change disk performance

- **Even if the disk re-maps bad blocks**
  - OS could lose data stored on disk
  - needs to be able to recover filesystem from partial update
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