Lecture 17: FS Organization

- File access API
- Naming & directories
- VFS
- Storage: Contiguous, LL, FAT, Inodes
- Disk optimized for sequential I/O
- File I/O optimized for ""

  - `open (filename)` → find file, prepare file for R/W
  - return file descriptor (`fd`)
  - (number identifying opened file)

  associated with each `fd`, there is a
  "current position", starting @ 0

  - `read (fd, buf, length)`:
    - read data from file into buffer
    - starting from "current position"

  - `write (fd, buf, length)`:
    - write to current position, advance curr. position

  - `seek`:
    - move "current position" pointer
    - (non-sequential: inefficient)

  - Close

memory mapped files (`mmap (fd, address range)`)

- file cached in memory
- mapped into the process' address space, read/write from file with loads:
- stores.
- because working on a cached copy
- need to explicitly sync data to disk.

Sync(fd): block until data written back to disk.
Some filesystems don't support directories at all.
- distributed filesystems (e.g. Amoeba 53)
Filesystem features:

- Metadata stored with files:
  - Size
  - Ownership/permissions
  - Type of file
    - Linux: part of filename (.exe, .exe64, .text)
    - Unix: stored under file (readable files: ELF, etc.)
  - Name of file
    - Usually stored as part of directory
      - Supports "hard links"
Storing files

- F1
- F2
- F3

Continuous alloc:
- internal
- external
- need to be able to expand files.
2nd strand: Linked list of blocks per file
- sequentially read by following pointers.

- simple
- sequential reads/append easy

- seeking is expensive
- hard to write into middle
  (except inserting blocks at block boundary)
- storage is not divided into nice powers-of-two sized chunks.
File allocation table (FAT)

- Like linked list allocation, but "short" pointers all stored on the side.
- FAT (cached in memory) if small enough.
- Works for small disks, not great for large disks.
- Seeking: \( O(n) \) memory accesses, \( \frac{\text{disk accesses}}{\text{number of files}} \) but small # of free.

MS-DOS FS (designed by Microsoft).
- Used in Windows.
- Used everywhere.
UFS Directories
- Directories are files
- Contents of that file (not notation)
  have mapping from filenames to contents
  to address of contents.