File Systems

44-550: Operating Systems

What is a Disk? What is a Bit?

- Hardware that stores data (sequence of bits)
- Can be organized various ways
- HDD (or SSD) frequently contain a boot sector/block to perform initial system boot
- Disks may have one or more *partitions*
  - Conclusions: multi-booting systems

File Systems

- Partitions organize, store, and retrieve their data in ways specified by the *file system*
  - FAT/FAT32
  - NTFS
  - ext2, ext3, ext4
  - Lustre
  - HDFS
  - Virtual File Systems
  - Many others (check out: https://www.kernel.org/doc/Documentation/filesystems/)

FAT/FAT32

- File Allocation Table
- MS-DOS/Windows
- Allocation table contains pointers to files
- FAT entries point to other FAT entries to create linked clusters
  - Cluster size: 1K to 32K
- Maximum partition size: 2TB
- Maximum file size: 4GB
- No file permissions/transaction log
- Maintains free clusters in linked list fashion
- 3 pointers must be changed when creating/adding cluster to a file
- Can move whole FAT chain to free list when deleting a file
- "fsck" difficult because there is no transaction log

NTFS

- New Technology File System
  - Microsoft's more reliable file system
- Cluster size: 512B to 64KB
- Maximum partition size: 256TB
- Maximum file size: 16TB
- File names: unicode
- Permissions: yes
- Transaction log: yes
- Uses a Master File Table
- Each directory entry points to an entry in the MFT
- MFT entries are of fixed size
- Once one is full another is allocated for a file and linked to the previous MFT entry
- Bit map to keep track of free clusters
- Capable of encrypting all data on disk

EXT4

- Fourth version of the extended filesystem
- Journaling file system for Linux: tracks unwritten changes in a *journal*
  - Circular log in the file system
  - Writes to the journal are atomic
- Older versions: ext, ext2, ext3
- Maximum volume size: 1 Exabyte (2^60 bytes)
- Maximum file size: 16TB (when using 48-bit block addressing, they're working on 64 bit...)
- 64000 subdirectories allowed per directory
- Uses extents: contiguous physical blocks
  - Improves file performance and reduces fragmentation
- Single extents can map onto 128MB of contiguous space with a 4KB block size
- Can preallocate space on disk
- Uses delayed allocation to allocate multiple blocks at once allowing for more contiguous files
- Uses nanosecond timestamps for files

Lustre

- Lustre (Linux cluster): parallel distributed file system used on half of the top 30 super computers
- Three parts:
  - Metadata server has single metadat target per Lustre filesystem with filenames, directories, access permissions, and file layout
    - Stored on a single local disk file system; used for pathname and permission checks
  - One or more Object Storage Servers (OSSes) that store file data on one or more Object Storage Targets (OSTs)
  - Clients that access and use the data which are presented with a unified namespace for all files and data in the FS
- Size only limited by sum of capacities of OSTs
- Most use enhanced ext4 called ldiskfs

And Many Many More

- Tons of other file systems exist
  - CD-ROM
  - DVD
  - Networked file systems (NFS, AFS, DFS)
- Virtual file systems allow access to multiple file systems by a single application

Mounting and Unmounting

- All media must be *mounted*
  - File management system must determine the type of media and start appropriate file system manager
  - Inform the virtual file system of its existence
- Correctly undoing these steps is called *unmounting*
  - Unmounting ensures all writes are correctly made to a disk
  - Disconnecting removable media without properly unmounting could cause data loss

IO Systems

- Takes an application's I/O request and sends it to the physical device, then sends response back to the application
  - Ideally, this performance should be optimized
  - One CPU (@2.0 GHz, 2e9 cycles per second) clock cycle takes around 2ns
  - HDD seek time: 10 ms (so 1e7 ns)
  - SSD seek time: .08ms (so 8e4 ns)
  - DDR3-1333 9-9-9 has a latency of 13.5 ns
  - Add in communication time...
- Recall:
  - Device interface controls the device while the CPU is doing work
  - Think SATA controller, Northbridge/Southbridge, etc
    - Heck, even the GPU isn't anything but a controller for pretty fast memory and some special purpose computation hardware

Direct Memory Access (DMA)

- Interrupts for slow devices and faster devices cause lots of CPU overhead
- DMA works with the OS (Example, reading from a file)
  - App performs system call to read data from a file
  - Device converts request to list of commands
  - Commands sent to the device interface hardware
  - The device interface initiates and controls disk operations (CPU does other things)
  - Device interface uses DMA hardware to transfer data directly to and from the memory buffer
  - Once the data has been transferred, the device interface sends an interrupt to the CPU to indicate completion of the request

Disk IO Scheduling (HDD)

- First come first served(FCFS)
  - Service IO request in order they arrive
- Shortest seek time first (SSTF)
  - Handle request that requires moving the arm the least amount
  - requests far away can starve
- Elevator (SCAN)
  - Once arm starts moving in one direction, it keeps moving until the last track is reached
  - Once it reaches the last track, it moves back
- Circular SCAN
  - Like SCAN, but arm does not stop to handle requests while moving in the reverse direction