ECE 598 – Advanced Operating Systems
Lecture 16

Vince Weaver

http://www.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

29 March 2016

Announcements

- Homework #7 is delayed
- Project topics due.

Brief File Review

File Operations

- Create
- Delete (Unlink)
- Open
- Close
- Read
- Write

- Append
- Seek
- Get attributes
- Set attributes
- Rename/move
- Memory Map

File Descriptors

- On Linux/Unix ”everything is a file”
- When open a file or object, get a number that indexes into a table, each referring to a file.
- Low-level syscalls mostly operate on file descriptors

Directories/Folders

- Root directory
- Hiearchical
- Path names. /, ., ..
- Operations
  - create
  - delete
  - opendir

- closedir
- readdir
- rename
- link
- unlink

Filesystems

- Often a MBR (master boot record) and partition table
- Disks divided into paritions
  - Why partitions?
  - Split up system (/, /boot, /usr, /home)
  - Why is boot separate? Smaller so boot loader can access, maybe a different fs type.
  - Dual-booting operating systems
  - Swap partitions

- Then individual filesystems

Filesystems – High Level

- Often first is called superblock, conaining all master info.
- Systemcalls: “mount” to put it in the proper place, “statfs” gives info on filesystem (including disk space, df)
- Some sort of free list, saying what areas are free (bitmap or pointers)
- inodes, an array of data structures containing master info for each file (and if file is small, contents of file)

- root directory entry
- directory layout
- file data

File Layout

- Contiguous. Files consecutive blocks. Simple. Fast to read (just read X blocks) Has fragmentation problems like with memory alloc.
  Ever used? read-only, CD-ROMs

- Linked list. Inode points to first part, each block points to next. No fragmentation, seeking through file involves lots of reads.
  Can also instead have the pointers in one single block, each pointing to next block. File allocation table. Whole

thing has to be in mem at once.

- inode table. List of blocks in file, last block reserved to point to next inode table.

Disk performance

- Traditionally a lot of this came down to hardware.

- Spinning rust disks; head movement, cylinders/sectors. Reading consecutive faster, random access bad (milisecond bad)
  More complicated, fancy disk interfaces and embedded processors. Large caches (why can that be bad), shingled disks?

- Much of this goes away with flash disks, but still emulate

old disk interface

- Name lookup can also be slow.

Common Filesystems

- Windows: NTFS, FAT
- Linux: EXT4, BTRFS
- OSX: HFS+
- Media: ISO9660, UDF