EECS 482
Introduction to Operating Systems

Winter 2018

Baris Kasikci

Slides by: Harsha V. Madhyastha

Naming and directories

- How to specify file to be accessed?
  - File name, click on icon, or describe contents
- File name is usually hierarchical
  - E.g., /home/barisk/482/notes
  - Allows users to group related files into one folder
  - Allows easy searching, e.g., “ls /home/barisk/482”
- Must translate file name to disk block # of header
  - What data structure to use to store mapping?
  - A hash table?
    - » Doesn’t naturally represent directories
  - Tree of directories

Directories

- Directory: mapping information for a set of files
  - Name of file → file header's disk block # for that file
  - Often a simple array of (name, file header's disk block #) entries
- Directories are stored on disk
- Directories and files are largely equivalent
  - Same storage structure
  - Directory entry can point to file or directory
- Any differences between files and directories?
  - User files are relevant to the user, OS files to the OS

Example: /home/barisk/482/notes

1. Read file header for / (root directory)
   - Contains pointers to data blocks of / directory
2. Read data blocks of /
   - Contains list of the files and directories in /. For each entry, contains mapping from name -> header's disk block #
   - One of those entries is "home"
3. Read file header for /home
4. Read data blocks for /home
5. Read file header for /home/barisk
6. Read data blocks for /home/ barisk
7. Read file header for /home/ barisk /482
8. Read data blocks for /home/ barisk /482
9. Read file header for /home/ barisk /482/notes
10. Read first data block for /home/ barisk /482/notes

Eliminated by caching file header for current working directory

Unified view of multiple storage devices

- Combine multiple storage devices into a file system
  - Each device contains own file system (starting with its root)
  - A directory entry can point to the root of a different device
- Example: loginlinux.engin.umich.edu

  / (root)

    bin (same device as /)

    etc (same device as /)

    tmp (separate storage device)

    afs (network storage “device”)
- Directory now can map name to 1) file, 2) directory, or 3) device

File caching

- File systems store lots of data structures on disk
  - Data blocks
  - Directories
  - File headers (inodes) and indirect blocks
  - Free lists
- How to improve performance?
  - Data layout to minimize seeking overhead
  - Cache data in memory
- Should the file cache be in kernel's virtual address space or in physical memory?
  - Either is fine, but if virtual, pin it

File cache vs. Virtual memory

- Both use physical memory as a cache for disk
  - Virtual memory: Use disk for increased capacity
  - File systems: Use memory for faster performance

- File cache and virtual memory compete for physical memory
  - Local vs. global replacement
  - Why have two mechanisms that both cache disk data in memory?

Memory-mapped files

- Use the paging system to cache both virtual address space **and** disk
  - Map file into a virtual address space
  - Point the backing store for that part of the address space at the file's data blocks

- Example: How to load a program executable from disk to memory?

File cache design

- Normal design issues for caches
  - e.g., cache size, block size, replacement policy
- How to keep copies on disk and in memory consistent with each other?
- Two options:
  - Write through
  - Write back
- Pros and cons?
- What guarantees does your Project 3 give?

Project 3 due in a week

- When we said “defer work”…
- Assert! Assert! Assert!
  - E.g. that page table is consistent with software page table
- Test multi-process cases
  - E.g. ./myrandomtest.sh
    - » Find a random test_* file and run it

Project 4

- Secure, multi-threaded network file server
  - Network programming, file systems, client-server systems, and security protocols
  - Experience writing significant concurrent program

- Start soon (3x LoC as projects 2 and 3)
  - Due Apr. 14th (final on April 23rd)

Multiple updates and reliability

- File system must ensure reliability/durability
  - Okay to lose data in address space
  - Data in file system must survive system crashes and power outages
- Challenge: Crashes in midst of multi-step updates

- Example: Transfer $100 from Baris's account to Tia's account
  1. Deduct $100 from Baris
  2. Add $100 to Tia
- What happens on crash between steps 1 and 2?