Course Review

ECE595
Dec 6

Y. Charlie Hu

Topic for Last Lecture

A: Distributed File System
Votes: 2

B: 2nd semester review; Practice final exam questions
Votes: 7

C: Introduction to Distributed Systems
Votes: 2

D: Other?
Vote: 3

Plan

- Reminder: course evaluation due Dec 9
- Final exam: EE 226, Dec 11, 10:30am-12:30pm

- 2nd half semester review (handout)
- Course summary (annotated slides from week1)

- Break: Movie quiz

- Topic review and practice final exam questions
- Quiz 3

Course Summary
(annotated slides from Lecture 1)

[lec1] What is an OS?

"Code" that *sits between*:

- programs & hardware
- different programs
- different users

But what (and how) does it do/achieve?

[lec1] What is an OS?

- Resource manager
- Extended (abstract) machine
- A giant interrupt handler!

Makes computers efficient and simple to use

[lec1] What is an OS?

Resource manager (answer1)

- Allocation
- Reclamation
- Protection

[lec1] What is an OS?

Resource manager

- Allocation
- Reclamation
- Protection

Finite resources
Competing demands

Examples:

- CPU
- Memory
- Disk
- (Network)

[lec1] What is an OS?

Resource manager
- Allocation
- Reclamation
- Protection

"The OS giveth
The OS taketh away"

Implied at termination
Involuntary at run time
(cpu, memory)
Cooperative (yield cpu)

[lce1] What is an OS?

Resource manager
- Allocation
- Reclamation
- Protection

"You can't hurt me
I can't hurt you"

Implies some degree of safety & security
- CPU
- Memory
- Disk
  - What is the essence of all mechanisms?

[lec1] What is an OS?

Extended (abstract) machine (answer 2)

- Much more ideal environment than the hardware
  - Ease to use
  - Fair (well-behaved)
  - Portable (back-compatible)
  - Reliable
  - Safe
- Illusion of infinite, private resources
  - Single processor → many separate processors
  - Single memory → many separate, larger memories

[lec1] Separating Policy from Mechanism

Policy – decisions on how to use tool
Examples:
- CPU scheduling policies
- Page replacement policies
- Buffer cache replacement policies
- Disk allocation policies

Mechanism – tool to achieve some effect
Examples:
- Priority scheduling vs. lottery scheduling
- FIFO w/ 2nd chance vs. Clock: a simple FIFO w/ 2nd chance

Separation leads to flexibility

[lce1] Is there a perfect OS?

Portability
Security
Fairness
Robustness
Efficiency
Interfaces

- Conflicting goals
  - Fairness vs efficiency
    - SJF vs. RR
    - FIFO vs. SCAN
  - Efficiency vs robustness
    - Buffer caching
- Don't know future
  - CPU scheduling
  - Page replacement
  - Disk scheduling

[lec1] There is no magic in OS design

This is Engineering

- Imperfection
  - Don't know future
- Tradeoffs
  - Segmentation vs. paging
  - Read/write API vs. mmap
- Constraints
  - hardware, cost, time
    - FIFO w/ 2nd chance
    - Enhanced version
    - Approx. LRU
- Optimizations
  - After functionality
  - 1-level paging -> 2-level
  - Basic FS → Buffer caching

Nothing's Permanent

- High rate of change
  - Killer-app: web servers→ Big data
  - Arch: uniprocessor → Multi-core
- Cost / benefit analyses
  - motivation for mmap (data sci app)
  - Sema impl uni → multiprocessor
- One good news:
  - Lots of inertia
    - Principle of locality
      - TLB
      - Demand paging
      - Buffer caching
    - Extra level of indirection
      - Dynamic memory relocation
      - 1-level paging → 2-level paging
      - UNIX multi-level indexed files

[lec1] About this course…

Principles of OS design

- Some theory
  - SJF optimal
  - Working set modeling
- Some rational
  - Optimize the common case
    - Sequential file access → prefetching
  - Locality → caching
  - Why mmap()?
- Lots of practice
  - Locality exists/how much?
  - Dist. of file size (UFS inode)
  - Buffer cache size vs. VM size?

Goals

- Understand OS design decisions
- Basis for future learning

To achieve the goals:

- Learn concepts in class
- Get hands dirty in labs

Great ideas in Computer System Design (1)

- "*All computer science problems can be solved with an extra level of indirection*"

-- David Wheeler

1. Dynamic memory relocation
   - Base&bound, segmentation, paging
2. One-level paging → Two-level paging
3. UFS multi-level indexed files
4. Boot block → stores bulk of the bootstrap program

(5. NFS: transparency via VFS)

Great ideas in Computer System Design (2)

- Principle of locality → Caching

1. TLB
2. Demand paging (VM)
3. Buffer cache in FS
4. On-disk cache
5. Client caching in NFS

(6. Hardware cache, L1, L2, etc.)

Plan

- Reminder: course evaluation due Dec 9
- Final exam: EE 226, Dec 11, 10:30am-12:30pm

- 2nd half semester review
- Course summary (annotated slides from week1)

- Break: Movie quiz

- Topic review & Practice final exam questions
- Quiz 3

Movie "Paycheck"

- Which ECE595 concept does the movie storyline remind you of?

  A: Priority CPU scheduling

  B. Virtual memory

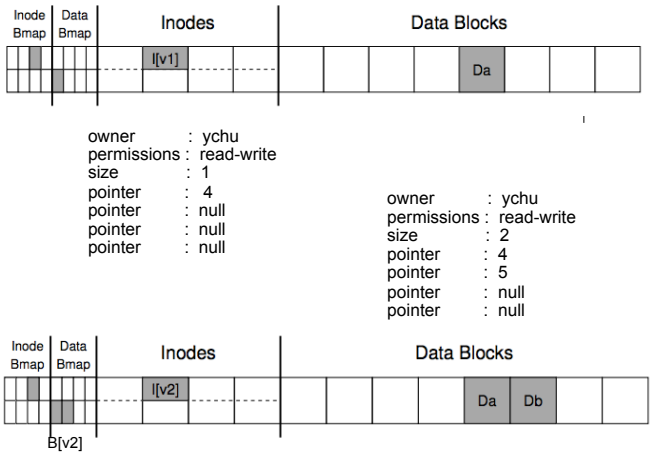
  C. Journaling file system

Topic review

- Journaling file system

- mmap()

- RAID



21

Appending a disk block to a file

[lec23] 3. Crash Recovery

- After a system crash in the middle of a file system operation, file system metadata may be in an *inconsistent state*
  - Independent of buffer caching
- Solution 1: fsck() after crashing
- Solution 2: journaling file system – making file operation "atomic"

22

Journaling file system
(without non-volatile mem)

- Keep a small log (on disk), write each set of changes to it first
  - Always append to end of log
  - All synchronous writes go to this log
  - Each set of operations for a task is a transaction
  - Log entries replayed on the file system in the background
  - Upon reboot, if log not empty, know what to do or undo
- The changes are thus made to be atomic, in that they either
  - succeed (succeeded originally or are replayed completely during recovery), or
  - are not replayed at all (are skipped because they had not yet been completely written to the journal before the crash occurred).

23

Topic review

- Journaling file system
- mmap()
- RAID

24

Quiz 3

- What is an inode? List at least 3 pieces of information typically stored in it.
- UFS organizes files in a tree structure. We discussed how to find "/usr1/ece595/file" in class. But how do you find "/"?
- How do you implement renaming a file from "file1" to "file2"?
- If "/hardlink" is a hard link that points to "/home/user1/file.txt", list the all the disk reads that are required for UFS to find the first data block of the file. Assume no disk blocks are cached, each inodes is stored in a different disk block, and that each directory contains no other files and fits in one data block.

True-or-False

- When a file is being accessed by a process, i.e., read or written, the current position within the file where it is being accessed is stored in the in-memory copy of its inode.

Practice (Metadata overhead)

- Consider a UNIX File System implementation with a logical block size of 512 bytes, and a traditional 128-byte inode containing, among other things, 10 direct block pointers, single-, double- and triple-indirect pointers. Assume 32-bit pointers.
  - What is the maximum number of bytes of metadata overhead that a large file can have?

Practice question (FS Interface)

Consider the following code for processing 8MB of data from a file on disk:

```
int *p = (int *) malloc (8*1024*1024);
fd = open ("file", "r");

read (fd, p, 8*1024*1024);
for (i=0;i<2*1024*1024;i++) p[i] = i;
```

- Assuming a page size of 4KB. At the end of each code segment, how many page faults have happened?

Practice question (RAID)

- Consider a RAID Level 5 organization comprising five disks, with the parity for sets of four blocks on four disks stored on the fifth disk. How many blocks are accessed in order to perform the following? How many parallel reads/writes are needed?
  - A write of 2 contiguous blocks of data

Practice question (RAID)

- True-or-False:
  It is faster to write 1 block in a RAID Level 5 organization (with 4+1 disks) than in a RAID Level 1 organization (with 2 disks)

Practice question

- Assume that
  - you have a mixed configuration comprising disks organized as RAID Level 1 and as RAID Level 5 disks;
  - the system has flexibility in deciding which disk organization to use for storing a particular file.
  - you have a mixed workload of *frequently-read* and *frequently-written* files
- Which files should be stored in the RAID Level 1 disks and which in the RAID Level 5 disks in order to optimize performance?

My best wishes

- Be well
- Do good work
- Keep in touch

from Garrison Keillor, *A prairie home companion radio show*