12/02 CS240 - Hints

Announcements

- Final Exam: Tuesday, December 9, 3:30 PM - 6:30 PM in Gates B3
  - Format like Midterm: open papers, no electronic devices
  - Cumulative with strong emphasis on material not covered by the Midterm Exam
  - Less time pressure than the Midterm Exam

- Course Feedback Now Open until end of day 12-15
  - http://course-evaluations.stanford.edu

For last class (Thursday 12-04)

Final Exam Review Discussion

CS 244C: Advanced Networking and Distributed Systems

- "New" course next quarter from Keith Winstein & David Mazières
- CS 244 (Advanced Topics in Networking) + CS244B (Distributed Systems)
  - Read research papers
  - Quarter-long research project with mentoring
- Placeholder website: https://cs244c.scs.stanford.edu

Paper

- Hints for Computer System Design
  - SOSP 1983 - The 9th ACM Symposium on Operating Systems Principles
  - Butler Lampson
    - Highly revered systems researcher
    - Key contributed to Xerox PARC: Personal Computer, Ethernet, Laser printers, …

,

Keep it simple

- KISS
- Do one thing at a time, and do it well - minimum essentials
  - Don't generalize
    - Why is a large, complex interface bad?
  - Predictable cost, don't offer something you can't delivery
    - PL/1 example
  - Altos -> Pilot
    - VM uses FS, FS uses VM

Keep it simple

- Get it right

  `FindIthField` counter example

Everything should be made as simple as possible, but no simpler

Lampson's examples

- CONNECT function

```
for i := 0 to Length(directoryPassword) do
        if directoryPassword[i] ≠ passwordArgument[i] then
            Wait three seconds; return BadPassword
        end if
    end loop
    Connect to directory; return Success
```

  - RDTSC
- *BitBlt*
- Dorado Memory System

Simplicity and generalization advice

- Make it fast
  - Paper:
    - RISC
    - Need tools

Simplicity and generalization advice

- Don't hide power
  - File system example
  - Which CS240 papers?

Simplicity and generalization advice

- Use procedure arguments
  - libC qsort, bsearch
  - Unix fork/exec vs Windows Create
- Special languages
  - Spy?
- Which CS240 papers?
- Event callback criticism

Simplicity and generalization advice

- Leave it to the client
  - Monitor example?
  - Unix small program

Continuity

- Keep basic interfaces stable
  - Mesa example
  - Linux example

Continuity

- Keep a place to stand
  - Backward compatibility
  - World swap

Making implementations work

- Plan to throw one away
- Keep secrets
  - Dangers?
- Divide and conquer
  - Quantizing?
- Use a good idea again

Handling all the cases

- Handle normal and worst cases separately
  - Normal case
  - Worst case
- Reference counting overflow
  - CS240 Paper?

Speed

- Split resources
  - Compared with sharing
- Use static analysis
  - Workload prediction
- Dynamic translation
  - Translation cache of Smalltalk

Speed

- Cache answers
  - CS240 papers?
- Use hints
  - What is difference from caches?
  - Alto & Pilot disk labels?
  - Ethernet?
  - In ESX?
  - In VMware software virtualization?
  - HTTP Redirect

Speed

- When in doubt, use brute force
  - Big-O notation
  - Rich Sutton's Bitter Lesson
- Compute in background
  - From CS240 papers
- Use batch processing
  - Pretraining
  - Inference

Speed

- Safety first
  - Avoid disaster
  - Hardware advances made life easier
    - Memory management
    - Compute sharing - multimedia

Speed

- Shed load
  - Admission control
  - ARPANET example

Fault-tolerance

- End-to-end
  - Relationship to hints
  - ARPAnet routing
  - Two problems? Checking, load
- Log updates
  - CS240 Papers?
  - Approach: Update procedure name and arguments
- Make actions atomic or restartable
  - What are atomic and restartable? Idempotent?
  - Shadow pages?