

---

## Lecture 2

### Hints for computer system design

---

## Background

- The content of this lecture is derived from Lampson's "Hints for Computer System Design" in 9<sup>th</sup> SOSP, 1983
- Key principle is separating interface (how clients interact with the system) from implementation
  - Functionality
  - Speed
  - Fault tolerance

---

## Functionality

- Keep it simple
  - Do one thing at a time, and do it well
    - And get it right
    - Make it fast, rather than general or powerful
    - Don't hide power
    - Leave it to the client
- Maintain stability/continuity
  - Keep basic interfaces stable
  - Keep a place to stand
- Get the implementation to work
  - Plan to throw one away, you will anyhow (Brooks)
  - Keep secrets
  - Divide and conquer
  - Use a good idea again
- Handle normal & worst case separately

---

## Speed

- Split resources in a fixed way if in doubt
- Use static analysis if you can
- Use dynamic translation when its reasonable
- Cache results
- Use hints
- Use brute force
- Compute in the background
- Batch if possible
- Safety first
- Shed load to control demand

## Fault Tolerance

---

- End to end (saltzer)
- Log updates
- Make actions atomic or restartable

## Resources

---

- Papers for next week's discussion
  - The impact of operating system structure on memory system performance
    - Chen & Bershad
  - The performance of  $\mu$ -kernel-based systems
    - Hartig et al.
- Fun papers
  - Hints for computer system design
    - Lampson
    - See also Lampson's Turing Award lecture
  - The rise of worse is better
    - Richard Gabriel (Section 2.1 of Lisp: Good news, bad news, how to win big)
    - See <http://www.dreamsongs.com/WorselsBetter.html> for the whole story