UNIVERSITY OF CALIFORNIA AT BERKELEY
College of Engineering
Department of Electrical Engineering and Computer Sciences
Computer Science Division

CS 162
Fall 1988

John Ousterhout

Mini-Sermon #1: Simplicity

- There are few absolutes in Computer Science. When building computer systems, most things involve tradeoffs. One exception is simplicity.

- Forces against simplicity:
  - If you can stump the TA you must be smart? Intelligence = ability to do things others can't understand? Better algorithm must be more complicated?
  - Advertising: make your product have more features than anybody else's product?

- Reasons for simplicity:
  - Sales: can't sell to a confused customer.
  - Easier to build, maintain. Ability to make systems more powerful is limited by complexity, both for designers and users. By finding simple ways of doing things, can add more power before system becomes unmanageable. UNIX started out incredibly simple, which made it easier for the system to grow later.
  - Faster: e.g. RISCs.
  - Cheaper.

- How to get simplicity:

    - Don’t accept complexity!
    - Ask ‘‘what is the simplest it can possibly be?’’ and ‘‘suppose we throw out this feature: what do we lose?’’ Don’t keep anything around unless it’s truly important. When in doubt, leave it out.
    - Study different systems to see how they are alike and how they are different. Try to find a single, simple way of doing all the common things.

- Simplicity and abstraction are really two words for the same thing.

- Instead of using creativity to confuse people, use it to find easy, ‘‘obvious’’ ways of doing things. Ultimate compliment is when somebody tells you ‘‘Heck, I could have thought that up!’’