Principles, rules, hints and art of computer system design

Lin Zhong
Keep it simple, stupid!
Keep It Simple

• *When in doubt, use brute force*
  • time favors brute force

• *When in doubt, leave it out*
  • adding a feature increases development effort disproportionately
When in doubt, leave it out

• Second-system effect
  • The second system is the most dangerous one to a designer
  • Features left out in the first make into the second

*The Mythical Man-Month* by Frederick Brooks
When in doubt, leave it out

- Escalating complexity principle
  - Adding a feature increases complexity out of proportion

*Principles of Computer System Design* by Salzter & Kaashoek
Escalating complexity principle

• Incommensurate scaling rule
  • Changing a parameter by a factor of ten requires a new design
Incommensurate scaling rule
Incommensurate scaling rule
Can we build a skyscraper by scaling up a small building?
Can we build an arbitrary large ....

- parking lot?
- processor?
- engineering team?
Brooks’s Law

- Adding manpower to a late software project makes it later
Incommensurate scaling rule

• Changing a parameter by a factor of ten requires a new design

• Don’t adopt a new design unless it is better by a factor of ten
Keep It Simple

• Avoid excessive generality

  • If it’s good for everything, it’s good for nothing.

*Principles of Computer System Design* by Salzter & Kaashoek
Make it available

Open design principle
Let anyone comment on the design;
You need all the help you can get

*Principles of Computer System Design* by Salzter & Kaashoek
Linus’s Law

• “Given enough eyeballs, all bugs are shallow” —Linus Torvalds
No guarantee

Introduced by Robin Seggelmann in 2011, code reviewed by Stephen Henson, into OpenSSL source code, 12/31/2011
Bug reported 04/01/2014
OpenSSL

- Open-source implementation of SSL and TLS
- Written in C
HOW THE HEARTBLEED BUG WORKS:

Server: Are you still there? If so, reply "moose" (5 letters).

User: moose

Server: Heartbleed enabled. User: moose

Server: Are you still there? If so, reply "bird" (4 letters).

User: Bird

Server: Heartbleed enabled. User: Bird

Server: Are you still there? If So, reply "Hah" (500 letters).

User: Hah

Server: Heartbleed enabled. User: Hah

Server: Are you still there? If so, reply "Hah" (500 letters).

User: Hah

Server: Heartbleed enabled. User: Hah

Note: Under certain circumstances, such as when the server is using a weak encryption protocol, Heartbleed can be used to leak sensitive information from the server. It is recommended to update the server's software to a more secure version to prevent such attacks.
Debate

• 15 minutes presentation by each side (30 minutes)
• 5 minutes rebuttal by each side (10 minutes)
• 5 minutes discussion among team (5 minutes)
• Open debate (25 minutes)
• Audience Q&A (20 minutes)
Debate topic 1
End to end principle

• “functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level”

*End-to End Arguments in System Design* by Saltzer, Reed and Clark, 1984
Ends & things in-between

Examples: TCP
Ends & things in-between

Examples: RISC vs. CISC
Functions

- Security
- Reliability: error control
- Safety
Keep It Simple

• Avoid excessive generality
  • If it’s good for everything, it’s good for nothing.

*Principles of Computer System Design* by Salzter & Kaashoek
Debate topic 2
Separation of mechanism and policy

- Mechanism: access control & allocation of resources
- Policy: who has access and get what resources
Separation of mechanism and policy

- Examples
  - Scheduling
    - When to schedule a process is policy
    - Stop, resume a process is mechanism
  - Power management
    - When to turn off a component is policy
    - Suspend/resume a component is mechanism
Why separation?

• Policy changes faster
• Policy and mechanism require different privilege
• Policy and mechanism may come from different sources.
Separation of concerns