

# Linux/Unix设计、编程、矛盾

你总是要解决二元对立的状态，这是不可调和的

# Unix的影响

- Unix的K.I.S.S原则：Keep It Simple, Stupid。
  - 自从Unix诞生后，其简单的设计，追求文本配置和通信是那个时代特殊环境造就的。
  - Unix的设计思想指导了大批程序员。我们总是说没有什么是一成不变的，而大量程序员却仍然盲目的坚持几十年前的思想。
- 
- 时至今日， Unix以及很多软件的设计都是极具参考价值的。但是规则怎么使用却不是一成不变的。
  - 也许之前我们用Simple和Stupid可以囊括一个软件，而现在我们抛弃了这个原则，这会让软件混乱不可维护，但是我们可以用很多个Simple和Stupid的模块组合起来。

# 设计你的软件

- 事实上，如何设计一个系统是一个除了多年的工程实践和思考的结果之外，还需要一些灵感和顿悟的过程。
- 你不要指望用什么方法让你的软件彻底的模块化，要知道一旦当你完成了灵活的组合式编程的体系，那么在内部，一定会有一些设计精密，相互耦合的部件在完成核心的工作。我们所在的这个世界，两种状态一直是相互作用的，走向极端的情况会让你痛苦不堪！

# 从无到有的过程

- 过度的构思会让你犹豫不决，而盲目的行动会让你误入歧途。
- 季文子三思而后行，子闻之，曰：“再，斯可矣。”（孔子从未说过三思而后行，事实上，孔子的回答是，想两遍差不多就行了，行动最重要。）
- 你应该在形成初步的方案以后，通过有形的实在的代码来验证，之后你就进入了想法和代码相互作用的过程，软件运转由此而生。