

####一、无名师与万行码

无名师曾来访的程序员说：“Unix传统上认为，一行shell脚本胜过万行C程序。”
这个程序员自以为对C极其精通，说：“这不可能，UNIX内核正是用C实现的。”
无名师回道：“确是如此。不过，UNIX传统上认为，一行shell脚本胜过万行C程序。”
程序员颇为沮丧：“但是在C中我们可领会到尊者Ritchie的智慧。我们与操作系统和机器合而为一，可以获取无与伦比的性能。”
无名师回道：“诚如你言。不过，Unix传统上认为，一行shell脚本胜过万行C程序。”
程序员冷笑着想愤然离去。无名师向学生Nubi颌首示意，Nubi在黑板上写下一行shell脚本，问道：“尊敬的程序员，看看这行管道，用纯C实现，是不是要一万行C代码？”
程序员沉吟念诵。最终他承认如此。
“你需要多长时间来实现和调试那个C程序？”Nubi问道。
“很长”，来访的程序员承认。“但傻子才会干这个而不去完成更有价值的任务。”
“那么谁更了解Unix传统？”无名师问道。“是写一万行代码的，还是看到任务的无谓而不去编码的？”
听到此，程序员眼中一亮。

####二、无名师与脚本狂

无名师和学生吃早饭时，从黑客大陆来了个陌生访客。
“Ihear y00 are very l33t,”他说。“P133z teach m3 all y00 know”。（我听说你很牛，请把你会的都教给我。）
无名师的学生面面相觑，都没听懂这类粗鄙言语。无名师微笑道：“你想弄懂Unix？”
“I want to b3 a wizard hax0r”，陌生人回答，“and 0wn ever3one’s b0xen。”（我想当个顶尖黑客，能掌握所有人的机器。）
“我不教这个”，无名师答道。
陌生人很激动。“D00d, y00 r nothing but a p0ser。”，他说。“If y00 n00 anything, y00 wud t33ch m3。”（哥们儿，敢情你没真本事啊，你要知道点东西就教给我了。）
“有条路”，无名师说，“可以将你带入真知。”他在纸上写了个IP地址。“黑掉这台机器，这对你来说应该不费什么力气，它的管理员不称职。回来后告诉我你发现了什么。”
陌生人鞠了一躬就离开了。无名师把他的早饭吃完。
几天过去了，几个月过去了。没人再想起陌生人。
数年过去了，黑客大陆来的陌生人回来了。
“你混蛋！”他说，“我黑掉了那台机器，你说的没错，太容易了。但是我被FBI抓起来扔进监狱了。”
“好”，无名师说，“你可以继续下一课了。”他在另一张纸上写了个IP地址交给陌生人。
“你疯了？”陌生人喊道。“经过这事，我再也不黑别人的机器了。”
无名师脸现微笑。“这里就是”，他说，“真知的开始。”
听到此，陌生人眼中一亮。

####三、无名师的双路论

无名师如是教导学生：
“达摩教义有条准线，这在尊者McIlroy的符咒“做一件事并做好”中得到体现。它强调软件应当具有简单一致的行为，这符合Unix惯例，人和其它程序便很容易想象其心理模型。
“但达摩教义还有另一条准线，体现在尊者Thompson的符咒“有怀疑，用穷举”中，很多经文都教导我们现在得到的90%，比等不来的100%更有价值。它强调实现的健壮性和简单性。
“现在告诉我：什么程序符合Unix传统？”
想了一会儿后，Nubi沉思道：
“老师，这些教义有冲突。”
“简单的实现往往对边缘情况有欠考虑，比如资源耗竭、无法关闭竞争窗口以及在未完成事务中超时等等。”
“发生边缘情况时，软件行为往往不规律、难以猜测。这当然不是Unix传统。”
无名师颌首同意。
“另一方面，大家都知道精巧的程序很脆弱。更进一步说，每个对边缘情况的修正往往牵扯到程序的核心算法，还牵扯处理其它边缘情况的代码。”
“于是，对边缘情况防患于未然、确保描述的简单性，反而会使得代码过分复杂、bug成堆、根本无法发售。这当然不是Unix传统。”
无名师颌首同意。
“那么，什么是正确的达摩道？”Nubi问道。
无名师说：
“当鹰飞翔时，它忘记爪子与地面相触？当虎捕食时，它忘记腾空的一刻？VAX只重三斤！”
听到此，Nubi眼中一亮。

####四、无名师与方法论

无名师和学生Nubi在圣地行走，无名师习惯在晚间为城市和乡村的Unix新门徒布道。
一次，聆听者中混入了一名方法论者。
“优化程序时不对热点进行反复衡量，就像渔夫把网撒入空洞中。”无名师说。
“那么，”方法论者说，“管理资源时不持续地衡量你的产能，不也像渔夫将网撒入空洞中么？”
“我一次碰到一个渔夫时，他正将网撒入船下的湖中，”无名师说，“他摸了好一会儿船底，像在寻找他的船。”
“但是，”方法论者说，“如果他把网撒入湖中，为什么还要找船呢？”
“因为他不会游泳。”无名师答道。
听到此，方法论者眼中一亮。

####五、无名师的GUI论

一晚，无名师和Nubi参加一个程序员的探讨会。有个程序员问Nubi和他的老师来自哪看看学校。当得知他们是Unix大道的追随者时，程序员颇为不屑。
“Unix命令行工具太粗糙太落后”，他讥讽道。“现代的、设计得当的操作系统可以在图形用户界面中做任何事情。”
无名师一言不发，只是指着月亮。旁边的一条狗对着他的手狂吠。
“我不明白。”程序员说。
无名师依然缄默，指着一幅佛祖相像，然后又指着一扇窗。
“你想说什么？”程序员问。
无名师指着程序员的头，接着指着一块大石。
“请把话说清楚！”程序员要求道。
无名师深深蹙眉，轻拍程序员的鼻子两下，把他扔到旁边的垃圾箱中。
程序员试图从垃圾堆挣扎出来之时，那条狗跑过来在他身上便溺。
此时，程序员眼中一亮。

####六、无名师与Unix班

一个Unix狂热者听说无名师掌握Unix大道真知，便跑来求教。无名师对他道：
“当尊者Thompson发明Unix时，他并不理解它。随后他理解了，受益了，不再发明了。”
“当尊者McIlroy发明管道时，他只知道它将传递软件，并不知道它能传递思想。”
“当尊者Ritchie发明C时，他将程序员放到缓冲溢出、堆损坏和烂指针bug的地狱中惩罚。”
“说实话，这些尊者又瞎又蠢！”
狂热者对无名师的用词极为愤怒。
“这些智者”，他抗议道，“给了我们Unix的大道。我们嘲笑他们，就是混淆是非，比转世为牲畜和MCSE还不如。”
“你的代码全无污点和缺陷？”无名师问。
“不，”狂热者承认，“没人不犯错。”
“这些尊者之智，”无名师说，“就是了解自身之愚。”
听到此，狂热者眼中一亮。

####七、无名师的Unix传统论

一学生对无名师说：“我们听说SCO公司把握着纯正的Unix。”
无名师颌首。
学生继续说，“我们还听说OpenGroup公司也把握着纯正的Unix。”
无名师颌首。
“这怎么可能？”学生问。
无名师答道：
“SCO确实把握着Unix源码，但是Unix的源码不是Unix。OpenGroup确实把握着Unix的名称，但Unix的名称不是Unix。”
“那么，什么是Unix传统？”学生问。
无名师答道：
“非源码。非名称。非思想。非实物。恒变。不变。”
“Unix传统是简单和空。正是简单，正是空，才使它更胜强飓风。”
“以自然法则前行，在程序员手中，吸纳各种优良设计。与之竞争的软件最终必与之想像：空，空，真空，虚无，万岁！”

听到此，学生眼中一亮。

####八、无名师与最终用户

无名师又一次布道时，一个最终用户听了他的智慧，跑来求教。
他对无名师三鞠躬。“我欲学习Unix大道，”他说，“但是弄不懂命令行。”
一个旁观的新门徒开始嘲讽最终用户，说他脑子一锅粥，说只有经训练者、有智慧者才配使用Unix。
无名师抚手不语，命这个嘲笑最终用户的新门徒前坐，坐到最终用户身边。
“告诉我，”他对新门徒说，“你写过什么代码，有过什么突出设计。”
新门徒吸嘴了两句，然后沉默了。
无名师转向最终用户。“告诉我”，他问，“为何你要寻求大道？”
“我用的软件并不能令我满意”，最终用户答，“既不稳定，也不美观。听说Unix之道尽管艰难，但超越一切，我愿抛去一切诱饵和虚像。”
“那么，”无名师问，“你为何想尽办法让软件帮你做事？”
“我是个建筑工”，最终用户答道，“这座城市里的很多房屋都出自我手。”
无名师转向新门徒。“家猫也能欺负老虎”，无名师说，“但是猫叫永远比不过虎吼。”

听到此，新门徒眼中一亮。

####解读

1. 无名师与万行码描述了程序设计的经济原则：宁可花机器一分，不花程序员一秒。也就是说，尽量使用机器完成更多地层次的工作。这条原则直接引出了生成原则：避免手工hack，尽量编写程序去生成程序。
2. 无名师与脚本狂则描述了一条黑客准则，就像W. Richard. Stevens一样，作为世界上最优秀的黑客之一，生前不曾著文攻死后却让全世界的顶尖黑客击们专门著文怀念，技术应该仅仅是一种技术，而不应成为一种手段。
3. 无名师的双路论则讲述了程序设计的透明性原则和健壮原则。透明行原则要求设计要可见，以便审查和调试。而健壮原则则力求程序的健壮，而做到这一点的方法就是理解程序的逻辑，而做到让程序的内部逻辑易于理解，方法就是坚持透明性原则。
4. 无名师与 方法论描述的是程序设计的优化原则：雕琢之前要现有原型，跑之前先学会走。用Domald Knuth的话说就是过早优化是万恶之源。这也是Unix世界的一条非常明确的传统：先制作原型，再精细雕琢，优化之前先确保能用。“极限编程”5
5. 无名师的GUI这一则确实比较难以理解，不过我认为这一则要求我们要分清事物的表象和本质。在这则故事中，“月光”、“佛”、“窗”、“大石”指的是图形化的东西，但是无名师缄默的时候程序员根本不可得知无名师的意思，也就是说，图形
6. 无名师与Unix狂，在原文其实已经有答案了——“这些尊者之智，”无名师说，“就是了解自身之愚。”
7. 无名师的Unix传统论则是说，真正的掌握不是对工具的掌握，也不是对思想的掌握。因为工具是变化的，思想也不是永恒不变的，只有摆脱工具和各种思想的束缚，做到真正的 “空”，不断吸纳优良的设计思想才是至上的设计。
8. 无名师与最终用户则表述了和第五则大致的思想：事物的本质才是决定事物的最终要因素，无论表象上怎样掩饰，都无法遮掩事物的本质对周围环境的影响。