
黑客道与教育传统的复兴

- 由知识构筑的壁垒
- 黑客与艺术
- 黑客的基本特征
- 泛系成才七故
- 黑客道的教学思想
- 黑客道的教学方法
- 黑客道的教学内容
- 黑客道的商业模式
- 古代伟大教育传统的复兴

题解

本章讨论自由软件在教育领域内的商业模式。自由软件的源代码是自由的，它为计算机科学的数学提供了一个极好的思想载体或者媒介，两者的结合是非常自然的。另一方面，自由软件的源代码是自由的，但是其大量的(有些部分晦涩难懂)的源代码又在软件作者和计算机用户之间构筑了高大而坚厚的壁垒，从而形成了关于知识的严重不对称性。

黑客是具有能力攻克或有效打破这一不对称性壁垒的程序员，他们是计算机科技界最宝贵的资源，他们具有极强的创新能力。然而，以往的教学实践表明具有一些特殊性格和气质的黑客是现行的教育体制下难以批量培训出来的。

在泛系方法论的指导下，笔者创造性将数学中的不动点定理扩变后运用到了计算机科学的教学中，并提出了一整套称为“黑客道”的新教学思想、教学方法和教学内容，并以此培训活动作为“一、百、万”工程的组成部分。

特别值得指出的是，黑客道从实践中认识到中国古代传统教学思想中合理的、积极的因素，并大胆地利用现代科学理论的教学内容赋予了它以新的内涵。黑客道培训计划中内在的教学思想与其说是对现行(源自西方的)教育体制的改革(或叛逆)，不如说是试图复兴中国古代优秀的教育传统。笔者相信复兴这一伟大的教育传统将是中国(乃至世界)教育变革的一个重要方向。

由知识构筑的壁垒

什么是知识构筑的壁垒呢？让我们先看这一组统计数据：

- 支持各种体系结构的 Linux 内核(2.2版本)的源代码包含约 4500 个 C 语言程序和汇编程序，这些程序存放在大约 270 个子目录中，源代码约有

200 万行，占据的磁盘空间达 58MB。这本《自由软件：新的游戏规则》的文本文件只有五十多万字，约 1MB 多一点，因此 Linux 内核的源代码全部打印出来的话，将有这本书的约 50 倍厚！

- Linux 内核只是 GNU/Linux 系统的一个内核，占整个操作系统源代码数据量的 5% 左右！！
- 普通人的大脑在某一时刻能记忆的源代码内容打印在 A4 大小纸张上只有五页左右，这种情况下天才黑客的记忆量难超过二十页!!!

知识构筑的壁垒是森严无形的，但是无疑是客观存在的。没有受到专业训练的人根本看不懂一段黑客用 C 语言编写的代码，即使是受到了专业训练的程序员在理解软件的原作者的源代码，对隐藏在代码中的原作者的思路理解也往往颇费功夫。当然与普通人相比，黑客是已经越过了由这些知识的不对称性构筑的壁垒的人，或者更加准确地说，是面对某个知识壁垒而有信心在较短的时间内越过壁垒、打破这种不对称性的人。

理查德·斯托曼(RMS) 当年启动 GNU 工程时，决定模仿 UNIX 系统而开发 GNU 操作系统时就表现出了黑客的这种能力和自信，他后来回忆道：

“答案很清楚：最急需的首先是一个操作系统。……作为操作系统的开发人员，我拥有的技能正适合这项工作。因此尽管我不认为成功垂手可得，但是我觉得这是我的天赋。我决定将系统设计得与 UNIX 兼容以便移植，同时也便于 UNIX 用户移民到这一新的操作系统上来。”¹

RMS 眼中的 UNIX 系统的结构和代码就如同我们常人眼中的桌子和板凳一般普通，他在 1984–1994 年这十年中创作力极其旺盛，而且不可思议地高产，GNU 工程的许多优秀软件都是他这段时间的作品。

注意，这里必须要澄清一下“黑客”(Hacker)的含义，因为这个词已经被大众媒体彻底搞错。黑客的准确定义是：“热爱编写程序，并且希望通过编写程序使自己变得聪明的人。”² 因此，这里所说的黑客与公众媒体上所说的破坏他人的计算机系统、盗取他人的计算机数据和秘密的 Cracker 一词具有本质的区别。Cracker 一词应该翻译成“骇客”。

黑客与艺术

黑客行为的艺术性是一个特别突出的特征，具有艺术性的东西是具有趣味的，趣味与艺术又是黑客文化中的一对双胞胎。与这对双胞胎打交道必须要有美

¹ 摘自“The GNU Operating System & the Free Software Movement”，Richard Stallman, 1998。

² The use of “hacker” to mean “security breaker” is a confusion on the part of the mass media. We hackers refuse to recognize that meaning, and continue using the word to mean, “Someone who loves to program and enjoys being clever about it.”

学方面的理论指导才能驾驭。遗憾的是，在现行的教育体制下，德智体的发展得到了重视，而美学的教育却没有得到赢得应有的地位，许多学生毕业后没有个性。社会对人才的需求是各种各样的，从事计算机科学技术的工作者无疑必须具有创造性，具有创造性的人才能为社会作出更大的贡献。

创造性源自想象、天才、灵感。伟大的德国哲学家黑格尔对艺术家的特点作了深入精彩的分析，在《美学》一书中，他提出了艺术家的要素：一位艺术家，无论他从事什么样的艺术，都需要具有**想象力、天才和灵感**，艺术表现必须具备**客观性**，艺术家必须具有**主观的作风、自己的风格**，他还特别提到了**艺术家必须具备独创性**。

黑格尔的美学观点对于分析黑客行为所具有的艺术性也是适用的，在下面几个小节中，笔者按照他提出的三个方面特征来讨论黑客应该具备的艺术性质。

黑客的想象、天才与灵感

想象

“想象”一词在黑格尔的《美学》中称为“**die Phantasie**”，准确的译法是“**形象思维**”。黑格尔认为形象思维能力与漫无边际的幻想是不同的。形象思维能力从哪里来？心理学家的回答是它与发散思维能力有关，而发散思维能力需要人打破条条框框的约束，用心理学术语说是思维必须不断地发生“**迁移**”。

仅有发散思维还不够构成形象思维，它还必须以某种连贯系统的形式体现在解决现实问题的方案中。伟大的数学教育家 G·波利亚在《数学与猜想》中写道：**一个好的数学家，必须是一个好的猜想家。好的猜想源自发散思维，或者说具有“启发性”(heuristic)的思维活动。**波利亚对此作了大量的研究，他特别重视类比与归纳在发散思维中的运用，他认为类比和归纳是产生好的猜想的有力思想武器³。有了一个好的猜想，再对猜想给予严谨的数学证明，才能构成数学上的创新，因此它的确不同于漫无边际的幻想。⁴

形象思维与直觉是密不可分的，艺术家都很重视“**直觉**”的在艺术创作中的作用。中国古代文人强调在琴、棋、书、画四个方面应有一定修养或者造诣，这些艺术方面的修养对于今天的程序员提高审美能力仍然有很大帮助。

弹琴者须懂音乐，古代许多文人都痴迷于音乐。孔子“**听韶乐，三月而不知肉味**”，白居易听到琵琶女的演奏，“**如听仙乐耳暂明**”，随即得到灵感，写出了

³ 按照泛系理论的话说，就是善于“**辨异同、排次序**”。

⁴ 波利亚还著有一本小册子《怎样解题——数学教学法的新面貌》，其中他列举了一个非常有趣的例子来说明他的观点，证明 $1^3 + 2^3 + 3^3 + \dots + n^3 = (1 + 2 + 3 + \dots + n)^2$ 在这个等式的证明过程中，他充分展示了如何将归纳、类比、启发等技巧结合起来解题。建议每一位想提高自己解决问题能力的读者都去读一读。

《琵琶行》这样的千古传世诗篇。欣赏和演奏音乐者应该明乐理，乐理由乐谱来表达。乐谱也是一种代码，音乐作品的美感体现在什么地方呢？

美国著名音乐理论家柏西·该丘斯(Percy Götschius)对此作过深入的分析，他在《音乐的构成》(The Structure of Music)一书中提出音乐创作具有四种极其重要的特性，其中两种属于精神活动的范围，另外两种属于技术方面。

关于技术方面的特性，主要为音乐的“连贯性”和“主题发展”，本书不讨论音乐技巧，因此这里不去深入说明这两种特性⁵。音乐中属于精神活动的特性在于“诗意”和“音美”这种美学性质。柏西·该丘斯说：

“诗意或者说音乐作品的戏剧性将生命的气息吹入蛰伏未动的音体后，予音乐以灵魂，眼光远大、遇事认真的真正作曲家，不管他怎样关注于急切的技术工作，不会轻视这种精神方面的影响。”

而“音美”，主要是说音乐的“动人性”、“可爱性”。该丘斯写道：

“神志清爽的人不会欢喜丑陋而厌恶美丽。一个人面对音乐的珠宝——如西贝柳斯(J. Sibelius)的第一交响乐徐缓乐章，柴科夫斯基的音诗(tone-poem)《罗密欧与朱丽叶》(Romeo and Juliet)，或者勃拉姆斯第二交响曲第三乐章，心中如何因喜悦而扩张呀！更不必说舒曼作品中无数精美的段落、舒伯特许多作品中难以描述的美。现代音乐常通篇是不协和音的堆积，这就放弃了音美这特性，使音乐变成丑陋了。”

该丘斯对现代音乐的批评显得有些厚古薄今，但是这些见解的确清楚地说明了音乐艺术的本质特征。音乐对于提高人的艺术修养、陶冶人的情操所能起的作用是非同一般的。这些作用对于黑客往往也有潜在的影响，自由软件社团内许多有名的黑客具有很高的音乐素养，有些擅长演奏乐器，甚至多种乐器，有些还善于作曲。

笔者本人也从音乐中领悟过美学的力量，学生时代的笔者因家境贫寒，无法购买键盘乐器学习音乐，于是从同学那里借来一把西班牙吉他练习弹奏，成了一位吉他音乐的爱好者，练习的音乐曲子中有泰雷加所作的“阿尔汉布拉宫的回忆”，这是一首旋律优美的曲子，前后有两段，通篇采用轮指的技巧演奏。除了左右手配合之外，右手的大拇指单独演奏一条旋律发展，而食指、中指和无名指则配合演奏另一条旋律发展，这三个手指弹奏的音时长必须相同而且力度必须均匀，曲子的演奏难度是较大的。后来在学习 C 编程时遇到了多线程的概念。多线程是一个可以充分发挥处理器计算能力的编程技巧，它是多个执行动作可以共享同一地址空间。多线程的程序必须处理好线程之间的同步问题。由于原来理解过两条旋律如何同步，因此笔者很快就理解了多线程的同步处理方法。

⁵ 实际上，计算机的运行系统具有与音乐连贯性和主题发展相通之处。

音乐是时间艺术的一种，音乐中的节奏是音乐形象构成的极其重要的要素之一，可以说没有节奏就没有音乐。计算机中也是这样，数字计算机内部必须提供时钟，而时钟的产生为计算机程序的运行提供了原动力。没有时钟，程序的定时、算术和逻辑运算、中断、进程等概念都建立不起来，而学习音乐可以很好地帮助学员掌握这些概念。难怪毕达哥拉斯学派认为“万物皆数”，音乐也是数。

音乐的作用不仅体现在对这些计算本质和编程技巧理解可以起潜移默化的辅助作用，更重要的是，音乐可以帮助人将心境平静下来⁶，佛家说“定能生慧”，道家说“静能入禅”。实际上他们两家说的基本上是同一个意思：人心原本像是一面光亮的镜子，但是平时被诸多杂念的灰尘遮盖了本来的面貌。当杂念的灰尘被拂去，人的心境重归平静之后，心智深处原来被杂念掩盖的许多潜意识资源就可以充分发挥出来，转化为明意识。

中国象棋、国际象棋或者围棋中讲究博弈技巧。博弈技巧其实就是算法的一种特定的形式，为了达到某个目的，棋手制订出一步一步的行动计划并实施。许多计算机编程高手也是下棋的好手，人工智能领域的许多研究人员都对下棋特别感兴趣，而且编写了许多下棋的软件，这些软件的水平非常高，已经可以在计算机与国际象棋的世界冠军的对决中获胜了。

书法修养在制作字库的程序中是显而易见的，许多计算机字库的作者很难说他们究竟是计算机科学家，还是艺术家，其间的界线是相当模糊的。

绘画能力与设计软件的图形用户界面之间的关系是非常直接的，GNU/Linux 上图形用户界面是基于 X Window 系统的，最初的 X 系统上实现了图像显示的策略，但是对于图形显示的外观则没有规定，留给了应用软件开发人员自行处理。X 系统的推出比 MS-Windows 的时间要早，但是就是因为没有一个设计良好的艺术风格(加上 PC 上可以运行的 UNIX 系统少见)，被 MS-Windows 这样的系统占了先机。后来在 GNU/Linux 之所以越来越受欢迎，很多程度上是因为 GNOME 和 KDE 越来越成熟。GNOME 和 KDE 这两个社团中有不少具有艺术修养的黑客，他们贡献了许多具有艺术性的作品，从而大大地美化了原本单调的图形用户界面。使用过 Red Hat 9.0 的用户惊呼：这个系统的图形用户界面比 MS-Windows 还要好！

其他的传统艺术，例如雕塑等都对黑客的成长具有很大作用。这里就不一一列举说明了。

许多科学家也具有良好的直觉。例如，美国数学家斯梅尔(Stephen Smale) 是拓扑学方面的大师，他对现代混沌学研究有重大的影响，⁷ 其文风往往开门见

⁶ 黑客道在以往教学实践中已经发现：以约翰·塞巴斯迪安·韩德尔等人为代表的巴洛克音乐具有神奇的功效，这些曲子可以非常有效地平静人的心境！

⁷ 他曾经证明将一个完整的球面不撕开的情况下由内向外翻开是完全可能的，在高维庞加莱猜想证明方面工作极为出色。

山、直入主题，文章用词浅显易懂，重视向读者提供直觉，有些结论甚至不给出证明，因此被一些传统的数学工作者批评为“不够严谨”。但是事实上，他的确是一位非常重视数学直觉的数学家，在很多数学领域都有着杰出的见解和开拓性的成就。

爱因斯坦的狭义相对论是他通过在头脑中做“追光”实验得到灵感而产生的，广义相对论则更是通过关于“对称性”的直觉将万有引力的传递速度与光速等价，将力学还原为一种直观的几何学而建立起来的。

许多计算科学家也是艺术家，高德纳教授就是这么一个典型人物。他多才多艺，除了编写经典著作和设计了英文的 **Computer Modern Romans** 字库之外，他还是一位作家，他在上中学就创作了一篇小说，名为“波泽比度量衡体系”(The Portzebie System of Weights and Measures)，就被美国著名的《疯狂》(Mad)杂志发表，他在这篇作品中提出了一种可以替代公制的新的度量衡制度，可以使用任何一本杂志的厚度为长度单位，既风趣，又富于想象力，虽然滑稽可笑，但是度量衡却设计得严谨、无懈可击，深受读者喜爱。⁸

天才

关于天才的话题很多，笔者喜欢套用计算机的术语来说明天才与常人的区别。天才头脑是性能优越的硬件，软件开发中的许多问题可以使用“硬件”直接完成。一般人的头脑没有天才那么好的天份，需要通过后天的努力(相当于通过“软件”)才能完成程序创作中的许多问题。

灵感

灵感是创造学、思维学、心理学的研究对象。什么是灵感呢？古希腊哲学家亚里斯多德对“灵感”的定义是“在一段微不足道的时间里，通过猜想而获得事物本质的联系”。波利亚认为灵感是“好念头”。他把“好念头”作为“灵感”的同义词。

灵感与直觉有密切联系，灵感可以描述为当对心智空间作拓扑变换之后形成的新的心智图象。这种心智图象如果可以容易地当作某个常识形象画出来或者用文字形容出来，则就是一种直觉。

黑客编程的客观性

黑客的活动突出地体现在程序的源代码上，而源代码一旦创作出来，就独立于黑客而客观存在了。正是这种客观性使得计算科学有了一个载体，使得其他黑客可以在不同的时间、不同的地点来对程序进行验证或者重复。

⁸ 关于高德纳教授的介绍，可以参见本书“排版、印刷和网络出版”一章。

黑客的功力体现在源代码中。它是评价黑客创造性的最重要指标之一。在自由软件社团内，某程序员发表了多少篇学术论文往往不是最吸引别人关注和兴趣的关键，而某个设计精良的软件往往会给该作者带来最大的满足感、荣誉感和成就感。

自由软件社团已经创作了一大批工具来保障这种客观性的再现。由于有了这些工具，无论是芬兰的 **Linus Torvalds**，还是墨西哥的 **Miguel de Icaza** 都可以成为影响全球自由软件社团的人物，他们事先并不一定需要什么博士学位，或者教授头衔。无论身份或者社会地位的高低，在追求真理的黑客(hacking)之路上，每一个人的权力是平等的。

关于原创性

与编程的艺术性和客观性相比，编程的原创性是黑客艺术性中的最为核心的特性。黑客的原创性之于软件创作，如同象棋比赛中置对方于死地的将军、足球比赛中的射门得分那么激动人心！

所谓的原创性工作一定是前人没有尝试的、或者没有发现的、或者没有完成的活动。抄袭被自由软件社团的黑客们认为是耻辱的行为；而在别人自由的源代码的基础上加入新的思想和代码则认为是创新，创新者可以得到社团成员的尊重。

自由软件社团常常以种种方式给予这种原创性以极大的尊重，一种常见方式是向自由软件的原作者提供捐赠，以便让软件的作者可以获得更多的支持完善软件的功能或者开发新的特性，不少优秀的软件就是在这种商业模式的基础上发展起来的。

黑客的基本特征

世有伯乐，然后有千里马；千里马常有，而伯乐不常有。这句话道破了能鉴赏千里马能力的伯乐的重要性。识别哪些人是可塑之才首先是教员的责任。在黑客道教学中，教员必须识别哪些学员是真正的千里马，是可以成为黑客的人。

无疑地，黑客是熟练掌握了计算机科学技术的人，他们有高超的编程技巧。教员应该在黑客道教学中向学员传授这些知识和本领。除此之外，黑客道认为黑客与常人相比还有些一些很特别的东西。笔者观察到的黑客有别于其他人的明显之处在于他们具有“三心二意”的心理特征：好奇心、细心、耐心；愿意与他人交流(高情商)、愿意向自由软件社团贡献自己的创作(高社商)。

好奇心

亚里斯多德说：“求知是人的本能”。好奇心是创造力的重要来源之一。大量的自由软件项目的起源是作者发现没有某个现存的软件可以解决他们手中遇到的问题。因为兴趣的驱使，软件作者往往对了解他人的作品具有极大的兴趣，并作出不懈的努力以了解和掌握他人的想法。

许多软件尽管代码量庞大，但是仍然吸引了众多的黑客学习和使用它。许多黑客在学习的过程中还得到灵感，贡献了新的代码改进原来的软件。一些功力很高的黑客往往重现编写一个类似的软件来满足自己的好奇心。例如一批黑客编写了基于 X Window 系统的 XEmacs 来模仿 RMS 的 GNU Emacs 的功能，并增添了很多图形用户界面的支持。

细心

编写软件本质上是一项科学探索活动。与其他任何科学研究活动一样，它需要创作人员具备细心的素质，粗心大意、马马虎虎的人是不适合从事软件创作的。如果某个学生不关注细节，那么其性格是不利于他成长为一名黑客的。当然，细心的素养既可以是天生的。也可以通过后天的培养而得到。以笔者亲身经历为例，笔者在学习软件开发以前，总是心不在焉(mind absent)，但是在编写一些程序后，开始喜欢关注细节，在这方面有了进步。

如果程序员具有细心的心理品质，那么他一定可以从中受益非浅。例如编写 C 程序时，C 语言的语法具有很多诡异之处和语义陷阱，细心的程序员会在编写程序时聪明地避开这些陷阱，高级的黑客甚至可以在编写程序的过程中利用这些陷阱增强程序的功能。⁹

与男性相比，女性往往具有更加细心的心理特点，历史上第一位程序员就是女性。我们不用奇怪，将来的黑客中一定会有很多女性程序员，如果更多的女性能有平等的机会参加编程实践的话。

耐心

耐心，或者说恒心是黑客的另一大心理特征。由于软件创作的涉及面比较宽广，对某些领域的知识和技能要求较高，没有持之以恒的毅力的人是难以进步到较高境界的。许多想法不错的自由软件项目之所以没有完成，并非程序员的能力或者水平不够，而是由于种种原因，程序员不能坚持所致。

⁹ 参见本书“迈向新大陆”一章。

高情商

与他人交流的意愿，与人交流需要当然有交流的本领。例如在互联网上，你与国外的某位黑客交谈，恐怕你就需要掌握专业知识，以及英语。即使与国内的程序员交流，也要学会如何清楚、准确地表达自己的思想。理解别人的思想需要有一定的基本功。而具备交流本领的人首先还必须要有与人交流的意愿。交流的本领可以通过后天的培养而得到，但是与人交流的意愿则属于情商的一种，是与先天的性格因素密切相关的，后天的培养不一定能达到目的。

高社商

黑客普遍具有改造社会的志向。在此，笔者杜撰了“高社商”这一名词，社商是反映个人与社会沟通能力、驾驭和运筹社会资源的能力、改造社会的意志一个指标。中国古代儒家提出“达则兼及天下，逆则独善其身”，古希腊的柏拉图提出的观念更加通俗：政治家应该是哲学家，哲学家应该从政。这些观点某种程度上都是说人才必须具备高社商。高社商是衡量一个人素质和修养的一个尺度。

具有“三心二意”的人是黑客道中的千里马，教员须倾注心血培养，策之以其道，食之尽其才，鸣之通其意。一方面，同时具备了“三心二意”的人在现实生活中少之又少，而且也不是所有的人都能经过训练成为黑客¹⁰。另一方面，中国是人口大国，人口基数庞大，每年毕业的几万名计算机专业的学生中恐怕只有1%的黑客成才率，总数仍相当可观，几年积累起来会是一个相当大的数字。从这个角度看，通过长期不懈的努力，我们从黑客的质量与数量上成为世界头号强国与大国是完全有可能的，关键是我们培养人才的教育体制、教学思想、教学方法、教学内容等要得当。

泛系成才七故

泛系方法论对于人才的培养方法提出了一整套见解，其中《泛系百经》中有“成才法”总结了成才七故：心方才学体境勤。成才(包括黑客的培养)与这七个方面都密切相关。

这里的“心”是指道德与修养。为人正大者容易成才，为人心术不正者难以成才，而且心术不正者具有某项本领之后，还可能会对社会造成损害。对于黑客道培训对象而言，如果一个学员在掌握了自由软件的开发方法之后，转而抛弃对自由的信仰而去开发专有软件，那么他不但对社会没有贡献，反而成了分割社会的破坏分子。因此黑客道培训始终把追求自由理念的道德训练放在

¹⁰ 如果经过一段时间培训，学员自己还是无法培养出“三心二意”的素质，那么我们建议他最好是去干别的事情。

第一位。“方”是指学习的方法。方法得当者，成才时可以事半功倍；而方法不当者，则事倍功半，甚至终生不能成才。“才”是指基本的才干、基本的能力。学习编写程序需要很多方面的基本能力。例如分析能力、综合能力、理解能力、表达能力，等等。“学”是指原来积累的知识。编写程序需要黑客掌握很多基础知识，基础知识牢固扎实的人容易成才。“体”是指是否具有健康的身体，这一点下面还将详细谈到。“境”是指周围的环境，对于黑客而言，这种环境目前最基本要求是具有计算机，具备上互联网的条件。“勤”是指能否集中时间勤奋地专心学习和实践。

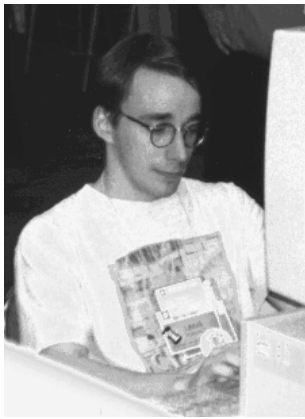


图1 黑客 Linus

为了说明泛系成才七故在黑客成长中的作用，本文以黑客 Linus Torvalds 为例展开分析和说明。Linus Torvalds 是 Linux 内核的原创者，这个内核是目前日趋流行的 GNU/Linux 操作系统采用的内核，2000 年世界上安装了 GNU/Linux 操作系统的 PC 数量已经超过了二千万台，而且目前装机数量还在不断上升，它开始成为市场的主流操作系统之一。许多曾经对自由软件不屑一顾的公司也开始在业务中采用这一操作系统。GNU/Linux 的成功堪称自由软件运动发展的一个里程碑。在 Linux 内核推出之前，也有一些开发自由内核的尝试，最终还是 Linus 率先取得了成功。

他的原创性体现在 Linux 内核的开发过程中采取了几个重要的决策，这些重要而正确的决策结合在一起最终成就了他。Linux 内核的原创性现在看来可以总结出以下三个主要特点，其中一个是道义上的，另两个是技术上的。

首先，Linux 内核在软件的编写方法上有很大突破。Linux 内核是按照自由软件基金会制订的 GPL 许可证发布的，这是一个至关重要的决定。因为正是 Linux 内核的许可证是 GPL 的，因此它的源代码是自由的。这保证了其他人加入进来，可以检查其源代码，完善其性能。

在 Linux 内核诞生以前，软件工业界流行着布鲁克斯定律，而 Linux 内核的开发模式彻底推翻了布鲁克斯定律。

布鲁克斯曾是 IBM 360 型计算机设计小组的一位骨干。他在 IBM 公司长期从事开发和开发的管理工作，布鲁克斯定律的隐含了一个假设，含义是开发任何一个软件都需要投入一定的时间，而且这个时间不能低于某个最小量，若投入的开发时间低于这个最小量，软件就无法完成。这个定律是建立在大量软件工程实践基础上的。在此基础上，他认为增加人力并不能缩短这个时间最小量。换句话说，“人多不等于力量大”，而且恰恰相反，人一多可能还会增加开发时间。因为人员增多之后，开发人员之间的沟通、理解、同步的时间会增加，管理人员与开发人员之间的工作量也会随之加大。具体到程序的错误

(bug)这一指标上,布鲁克斯定理的结论是一个软件项目的 bug 数量与开发人员人数的平方成正比。¹¹

前后参加过 Linux 内核开发并贡献过源代码的人数超过了 1000 人。按照布鲁克斯定律,这个软件包的错误应该很多,问题应该是非常大的。但是事实恰好相反,这种松散形式结构组织起来的 Linux 内核开发团队设计出来的代码质量非常高,比许多商业公司的同类软件都好。在 Linux 内核这个项目上,布鲁克斯定律彻底失效了。

《教堂与集市》的作者恩里克·雷蒙(Eric S. Raymond)在见到并使用了 GNU/Linux 操作系统后赞叹道:“[这些黑客]把健康和卓越融进了过去的标准……看到这些精采的代码作为一个工作系统展现在我面前,我获得了一种强有力的体验,而不仅仅从知识和智力上了解到所有的代码都摆在那儿,就好像是自己多年来在一大堆的汽车零部件中搜寻,有一天却突然发现那些同样的零部件被装配进一辆红色的法拉利轿车中,车门开着,钥匙在锁上摆动,发动机温和地鸣响……”

Linus 的聪明之处在于他知道如何“滚雪球”,并将“雪球”越滚越大。荀况在《劝学》中说:“吾尝终日而思矣,不如须臾之学也。吾尝企而望矣,不如登高之博见也。登高而招,臂非加长也,而见者远;顺风而呼,声非加疾也,而闻者彰;假舆马者,非利足也,而至千里;假舟楫者,非能水也,而绝江河。君子生非异也,善假于物也。”

正是这种“善假于物”的策略,使得 Linus 设计的 Linux 内核得以快速发展,加入了 Linux 内核开发的人都是世界顶级编程高手,这些黑客以往已经在 UNIX 上工作了很长的时间,具有丰富的开发经验,而 Linux 模仿 UNIX 内核的设计,以及 GNU 工程大量健壮的开发工具更是使这些黑客如鱼得水,这些因素聚集在一起,使得越来越多高质量的代码像滚雪球一样组织起来,形成了一个健壮的 Linux 内核。前后参加过 Linux 内核开发并且贡献过代码的人在 Linux 内核的源代码软件包内的自述文件中都列了出来,这个自述文件就是 Linux 内核开发小组的光荣榜。

第二, Linux 内核的设计以实用为目的。Linux 内核的结构设计在学术上没有什么值得可圈可点的大创新,它沿用了 UNIX 内核的基本设计思路,计算机学术界对这种设计评价是相当低的,荷兰的 Andrew Tanenbaum 教授从一开始就对 Linux 的结构嗤之以鼻,因为 Linux 内核设计没有更好地采用模块化的设计思想。但是 Linus 还是在许多地方出色地展现了他的高超编程技巧。

Linux 内核采用了模块动态链接技术。这在当时的 UNIX 社团内算是一个比较新的技巧和进步,传统的 UNIX 内核是一个巨大的二进制代码块,尽管 UNIX 内核的设计在源代码层次上是相当模块化的,但是一旦编译成二进制代码后,

¹¹ 详细介绍可以参见布鲁克斯所著的《人月神话》。

UNIX 内核就是“铁板”一块，这种“铁板”一块般的结构是无法在运行时按照需要剪裁和扩充的。不再使用的设备驱动程序将继续保留在内核中，从而浪费内核存储空间，而 Linux 内核自从第 1.2 版之后就提供了对模块的动态加载和卸载的支持。

模块在运行时可以动态加载和卸载类似于 MS-Windows 系统编程概念中的“动态链接库”(DLL)的概念。隐藏在模块动态加载和卸载之后的思想简单而高效，当内核试图访问一个不能使用的资源时，它会通知用户程序，而不仅仅是像传统的 UNIX 内核那样返回一个错误。这是一个相当精巧的设计，某种程度上它使 Linux 内核具有了一些微内核的技术特点。

第三，Linux 内核突出了可移植性的重要性。Linux 内核的设计是从 Intel 80386 机器上起步的，Linus 设计它的初始动机就是希望在 PC 上能运行一个 UNIX 系统。PC 一般采用 Intel x86 及其兼容的中央处理器，在内存寻址方式上，Intel x86 具有分段和分页两种机制，分段机制是 Intel x86 上具有的特征，其他同档次的 CPU 体系结构上没有类似的分段机制，而只有分页机制。的确从内存寻址的角度看，Intel x86 的分段机制是多余的，因为从 Intel 80386 开始，硬件已经支持从逻辑地址到线性地址的转换，而物理地址可以直接等于线性地址，Intel 公司主要是为了考虑让以前开发的程序能在新的硬件上运行才保留了分段寻址机制¹²。Linus 从 Alpha 芯片的内存寻址机制中得到了启发，在 Linux 内核的设计中他偏向了支持分页寻址，这使得 Linux 内核后来可以移植到 Intel x86 以外的许多硬件平台上。当然，可移植性的取得有 GCC 很大的功劳，因为 Linux 内核主要是利用 C 语言编写的，而此前 C 代码的编译器 GCC 已经移植到几乎所有的硬件平台上了。

在创作 Linux 内核的过程中，我们可以看出“体”的重要性，由于软件创作是一项繁重的脑力劳动，因此黑客们必须注意自己的身体健康，软件创作是青年人从事的职业，没有好的身体是很难搞软件创作的。健康的身体确实是干任何事业的物质保证。

历来成功的人都非常重视身体，司马迁说：凡人所生者神也，所托者形也。曾国藩说得更加直接：功名看器宇，事业看精神。建立功名必须要有好的身体。毛泽东在年轻时就提出：野蛮其身体，文明其精神。他也正是这样做的，他常年坚持游泳，冬天里也没有停止，“自信人生两百年，会当水击三千里”是他那时的写照。

对于程序员而言，为什么身体健康这么重要呢？近年出现了“极限编程”这个新名词。极限编程有几个版本，依笔者的理解，就是在人的心境进入一种极度专一的状态之后，可以迸发出创新的点子，得到非常高的编程效率，得到的代码质量也非常高。当人的心境进入这种状态之后，大脑的新陈代谢非常

¹² 64 位的 Intel x86 已经开始取消了内存寻址中的分段机制，IA-64 转而只支持分页机制。

快，如果没有好的身体作为保障，“登山则气迫、涉水则筋挛”的人(除非有顽强的毅力)很难坚持较长时间。



图2 疲惫的 Linus

Linus Torvalds 创作 Linux 内核的时间是从 1990 年开始至 1991 年期间，当时的 Linus 是芬兰的一名大学生，由于兴趣的驱动，他自己花钱购买了一块 Intel 80386 的主板开始在上面编写程序。他非常投入，读者可以从照片看出那一时间他比较消瘦，这与他那一段时间专注编程有关系，有时他疲惫不堪，就倒在教室的课桌上打瞌睡，他段时间除了编写 Linux 内核，其他什么事情都抛开了。这就是一个极限编程的实际例子。

Linus 的成功相当大的程度上还归功于他所处的环境。芬兰这个国家出现像 Linus Torvalds 这样的黑客以及像“诺基亚”(Nokia)这样的 IT 企业不是偶然的，芬兰是北欧的小国，在英语中，芬兰的拼写为“Finland”，意思是“最后的一块土地”。的确，在地图上，从芬兰再往北走就是北极而没有土地和路了。芬兰政府非常注重国民教育，它是北欧全面实行高福利、高税收的国家之一。在教育领域，高福利政策体现在公民从幼儿园到大学接受教育全部免费。

这一体制下，芬兰的教育搞得非常好，从幼儿园到大学，教学设施和条件都很完善。在 1990 年时，全球的 Internet 的规模还相当小，即使在发源地美国，Internet 的用户数量也不算大，但是芬兰赫尔辛基大学就已经为大学生提供了 Internet 接入服务，正式这种便利的上网条件使得 Linus Torvalds 有机会接触到网络上的多名黑客，并能与他们交流，得到了他们的很多帮助。也许不是巧合，赫尔辛基大学自上个世纪七十年代开始就非常注意收集关于操作系统方面的研究和有关资料，这些前人积累的条件都被 Linus Torvalds 善加利用了。

以笔者之管见，包括芬兰在内的北欧社会运作模式在目前全球的范围内是一种相当先进的制度，因为在这种社会体制下，社会整体的精神文明和物质文明程度都到达了一个相当高的水平，这种模式比美国的模式更好，非常值得其他国家认真学习和借鉴。¹³

¹³ Linus Torvalds 后来移民到了美国，但是成了美国公民的他的创造力显然已经没有当年大学时代旺盛，实际上，Linux 内核发展到现在，Alan Cox 等人在项目的日常维护上的贡献实际上已经超过了他。

黑客道的教学思想

黑客道之道

“道”一词作为哲学概念最早是老子提出的，他以此为基础发展并提升了一套完整哲学。《道德经》开篇就提出：

道，可道，非常道。名，可名，非常名。

这一表述使得“道”成了一个不容易说清楚的东西，可以说清楚的东西就不是自然界原本存在的东西了。这给后人出了一道难题，围绕什么是“道”，历史上一些大学问家又作了多种解释。下面是比较有名的一段，摘自《淮南子》：

夫道者，覆天载地，廓四方，析八极，高不可际，深不可测。包裹天地，禀授无形。原流泉勃，冲而徐盈，混混滑滑，浊而徐清。故植之而塞于天地，横之而弥于四海，施之无穷，而无所朝夕。舒之幂于六合，卷之不盈于一握。

这一段对什么是道的描述更加让人摸不着头脑¹⁴。中国古代哲学的学说的理论体系起点很高，在春秋战国时代就达到了非常高的水平，在系统普适性方面做得比较好，但是进入封建社会后，特别是在“独尊儒术、废黜百家”之后，这些学说的确切性和形式的具体性方面因历史条件的限制一直存在不足和缺陷，“这些学说往往是思辨性的、潜科学的、前科学的研究，若不计它们可能包含的伪科学的或者诡辩的成分外，它们之所以不是一种显科学，不能够充分可观控建模，原因之一在于它们缺乏属于可以操作的研究，因此往往可以牵强附会、仁者见仁、智者见智，可以作随心所欲的或者顺社会势力风向的解释。”

显然，这些带有神秘色彩和宗教倾向的学说不具备现代科学技术具有的“普适性、确切性和具体性”特征的，因此很难利用这些学说建立数学模型。我们能否站在现代科学技术的基础上，对中国古代哲学补充以西方现代科学的手段和研究方法，以此作为一种现代拿来主义的发展方向呢？

理性思维是现代科学技术的一大标志，而“数学与哲学是人类理性思维、理论思维最为美丽的花朵，它们花色不同、运筹各异，共同生成了人类远离其他生物的一对翅膀。数学是智慧神明之府，是模拟与分析事物、运筹显生的奇学妙术，数学是悟性的高速公路，是高科技的理论基础之一。

数学、哲学[和逻辑]，它们是理性之最，但是它们在自我剪裁时，又自我否定、自我超越。它们本身又证明了理性把握世界的相对局限性，它们引入了形形色色的悖论，它们进入形形色色的悖论。但是，非理性永远不可能取代理性，理性在逐步溶析化解非理性的无限进程中也永远不可能取代全部非理性，理

¹⁴ 令人吃惊的是，不少现代理论物理学家很推崇道家的这些哲学思想。

性与非理性复合成文化与潜化的冰山，浮在上面的理性虽然偏少，它仍是人类所能够看到的、欣赏到的、明晰的美。¹⁵”

计算科学曾经是、将来仍然是数学的一个分支，建立在计算科学上的计算机技术要与历史上的科学技术和未来的科学技术联系而显生，就要不断地开拓数学化的工作，泛系可以是计算机科学技术与数学之间的相互显生的中介网络。

站在泛系方法论和自由软件两者的肩膀上创立的“黑客道”是刻意追求三兼顾的，即兼顾教学理论的相对普适性、内容的相对确切性、形式的相对具体性。黑客道中的“道”具有理性思维的美，它可以由不同的人反复操作，可以在不同的时间和不同的地点实践并重复，因此它是一种科学的“道”，而不是传统宗教意义或者朴素哲学意义上的那种神秘兮兮的、难以言状的“道”。

黑客道培训计划的建立在一整套新的教学思想上的，而这个新的教学思想有两个主要来源，一个是吴学谋教授在历经约半个世纪的探索中创建的泛系方法论(PM)，另一个是理查德·斯托曼等人创作的 GNU 工程。

泛系方法论是融哲理、事理、数理、技理于一体的新型跨领域研究。在长达半个多世纪的研究中，它发扬了中国传统哲学与方法论中侧重从事物的关系、关系的转化与广义对称中考察事物的特点，提出了一整套相对普适又相对精确的概念、原理、方法、模式和技术，统一地处理了数理科学、系统科学、思维科学、自然科学、社会科学与哲学的许多专题，得到了一系列形式具体的新结果。

GNU 工程是 1984 年启动的，在过去近二十年的发展中，开发了一大批高质量的软件作品，成果集中体现在 GNU/Linux 这个 GNU 系统的变形系统中。正如前面提到的，全球 GNU/Linux 操作系统的用户在 2000 年已经超过了两千万之众，而且是目前商业市场上份额增长最快的操作系统。它取得的成功表明，计算机用户不需要某个公司凌驾于其上，也可以拥有充足的软件资源可用，来完成这样或者那样特定的任务。

由于历史条件的限制，泛系理论在数学等学科上的工作较多，在计算机工程技术方面的探索工作做得较少。自由软件资源虽然很丰富，而且很多软件作品都是高质量的，但是，这些软件的开发人员的开发方法几乎从来没有人进行过认真而系统的整理，以上升到一个更加自觉的、理性的高度。

在泛系方法论和 GNU 工程的基础上，应用不动泛系原理，笔者创造性地开发了旨在培养黑客的一系列课程，它不完全同于 IEEE 与 ACM 推出的教学内容，与国内绝大多数高等院校计算机专业的教学思想则是完全不同。黑客道培训计划的“一、百、万”工程中的重要组成部分¹⁶，该计划实施的目的是有两

¹⁵ 以上摘自吴学谋教授所著《泛系：不合上帝模子的哲学》。

¹⁶ 关于“一、百、万”工程的详细说明，可参读本书“泛系、自由与一、百、万工程”一章。

个：在社会中培养大量的理解并支持自由软件的哲学思想和参与自由软件运动的黑客，使黑客具有娴熟的自由软件开发能力。

黑客道的教学思想

黑客道的教学思想的创意源自笔者学习数学的经验。将从数学中学到的模型经过适当泛系改造后应用到教育领域中来。

从不动点定理到不动泛系

“不动点”原本是一个数学概念，具体研究发端于拓扑学的研究。不动点定理与费马猜想、连续统假设等问题一样都是世界数学名题。荷兰数学家布劳威尔(Brouwer)证明了不动点定理(Fixed Point Theorem),¹⁷ 这个定理的数学证明需要一些高深数学技巧，我们在这里不作展开说明，有兴趣的读者可以在许多高级拓扑学教科书上找到其证明过程。¹⁸ 尽管这一定理的表述相当数学化，但是其中不动点的概念是很容易理解的，我们可以列举几个具体的例子加以说明。

例一：设有两份一样的中国地图，如果我们把其中的一张地图搓揉成一团(当作一个整体地搓揉，不撕裂分解)，然后随机地将纸团抛到另一张地图上，纸团落在地图上的那一点有一个城市，那么必定在纸团中可以找到一个点与落点城市同名。这个城市我们称之为两份地图在“搓揉”变换下的不动点。¹⁹

例二：设有两把具有同样刻度(但是长度单位不同)、长短不一样的直尺(例如一把为 10 英寸、另一把为 10 厘米)，将一把直尺放在另一把直尺上，沿着它任意滑动，滑动停止后，我们可以在较长的直尺上找到一个点，其刻度标志与较短直尺上某个点的刻度标志重合。这个重合的刻度标志所在的点称为两把直尺在“滑动”变换下的不动点。

例三：如果你有一个计算器，可以作如下实验：从某个数开始(例如 5)，然后反复按 \cos 函数键，也就是： $5 \rightarrow \cos(5) \rightarrow \cos(\cos(5)) \rightarrow \cos(\cos(\cos(5))) \rightarrow \dots$ 这时计算器的显示屏上依次显示的结果是： $5 \rightarrow 0.996194698 \rightarrow 0.999848852 \rightarrow 0.999847141 \rightarrow 0.999847142$ ，最后的数字停留在 0.999847142 上，以后无论按多少次 \cos 键，这个结果都不会变化，这个过程就是相当于求方程 $\cos x = x$

¹⁷ 这个定理的表述为：设 $f: B^n \rightarrow B^n$ 是 n 维球体 $B^n = \{x \in R^n : \|x\| \leq 1\}$ 到自身的一个连续映射，则有 $x \in B^n$ 使得 $f(x) = x$ ，即 f 必有不动点。

¹⁸ 具有讽刺意味的是，布劳威尔是直觉主义学派的创始人，这一数学流派否认实无穷的存在。但是在这一定理的证明中，他却使用了选择公理，而承认选择公理就等于承认了实无穷的存在。

¹⁹ 或者设想一幅无孔洞的地图，将它完整复印到一张尺寸可不相同的橡皮膜上。让橡皮膜任意连续变形甚至可以有折皱，并将橡皮膜紧密贴合在原地图上。只要它被原地图完全覆盖，则在橡皮膜上至少存在一个点，与原地图上有相同经纬度值的点重合。

的一个根 $x = 0.999847142$ ，它是函数 $f(x) = \cos x$ (函数可以视为一种变换) 的一个不动点。

迄今为止，数学上的不动点定理在两个领域获得了广泛应用并且取得了巨大成功：一是经济学理论研究领域，一是混沌学研究领域。在经济学领域，曾经有牛津大学的 John R. Hicks 教授和哈佛大学的 Kenneth J. Arrow 教授提出了一般均衡理论，对一般经济均衡理论和福利理论作出了先驱性的贡献，两位学者因为这一方面的成就获得 1972 年度的诺贝尔经济学奖。后有 Arrow 教授的学生 G. Debreu 在他们的基础上，把新的解析方法引入经济理论以及对一般均衡理论严格重新叙述而获得 1983 年度的诺贝尔经济学奖，他采用的解析方法就是建立在不动点定理基础上的。

现在公认庞加莱是混沌学研究的始祖，但混沌学领域研究和应用不动点是上个世纪末期才在学术界成为一大潮流，科学家们似乎忽然发现大自然中广泛存在的混沌现象在相当长的时期内居然被忽略掉了。其实与混沌有关的非线性力学现象早就被人观察到，例如亚里斯多德认为空气中的羽毛的下降速度就比铁球慢(从而认为轻的物体比重的物体下降慢)，这一结论存在了近两千年，直到被伽利略著名的比萨斜塔实验所推翻，而羽毛的确在空气中下降得比铁球慢的现象在伽利略之后就没有人再去研究，科学研究的重点转向了以线性为主的对象上，非线性的问题最终还是要转化为线性模型来解决。气象学家洛伦兹观察到的混沌迫使人重新思考非线性的问题，由此推出的一连串结论令人惊愕不已。例如，长期(例如五年或者十年之后)的气象预报注定是不能成功的，没有人今天能够告诉我们十年之后的这一天北京市的天气是晴还是阴。混沌学中研究的许多模型中都有不动点，洛伦兹混沌中的两个吸引子就是庞加莱截面上的不动点，台风中心的风眼也是不动点。由混沌学开创的研究还只是刚刚拉开序幕。

抽象的点、抽象的不动，在现实世界中，我们面对的林林总总实际问题中，往往又是不现实的，实际上是相对的、近似的。实际上，不动点定理的“点”可以由我们推广成为一般的泛系，包括集合、网络、泛权场网、林林总总的关系、转化、泛对称、泛极与泛通等等。不动点定理的“不动”就现实显生为所谓“泛系不动”，也就是林林总总的泛系异同关系、同一关系、相容关系、模拟关系、广义的拓扑近邻关系、逼近关系、动静关系、泛对称关系等等。

不动点定理中的函数映射发展为一般的泛系转化或者泛系运转(对某泛系空间或者泛系族的泛系运转)。不动泛系定理就是在一些泛系转化下，某些泛系是“泛系不动”的。笔者在阅读大量泛系著作的基础上²⁰，对在不动点定理基础上专门发展起来的不动泛系原理作了深入研究，并把它应用到了教育学中，创立了“黑客道”(Hackerdom Training Program)。

²⁰ 参见吴学谋教授所著的《从泛系观看世界》、《泛系理论与数学方法》、《泛系导论》中的有关内容。

教育学中的不动泛系

在教育领域应用不动泛系的前提是明确其具体含义，即信息与知识如何表示为不动泛系(fixed pansystems)。知识是采用信息来传递的，而信息论的基础是仙农在 1948 年所著的《通信的数学理论》中奠定的，信息论对于消息是否构成或者含有信息作了研究，结论是它取决于消息本身的“不定度”，不定度是信息论中的一个术语。试比较下面两个句子：

“他有时迟到。”

“他昨天迟到了一刻钟。”

显然第二个句子所提供的信息量比第一个句子的要大。信息论发明了“不定度”这个术语来量化这种属性，可以通过计算而得到不定度的值。这个值在一定范围内变化即表现出异同。故有无相生，难易相成，长短相形，高下相盈，音声相和，前后相随。

信息在特定系统下的不定度达到一定的量时就可以称“奇”，由对比(即异同关系的运算)可得到奇。这种因比较异同关系引起的奇可以引起我们感观上的某种精神错乱，而这种精神错乱如果能让人愉快，那它就是心理上可以感觉到的美。好的相声或者小品节目之所以让人捧腹大笑、喜闻乐见，往往是因为演员表演的内容与我们的常识的比较有出奇的地方，而且这种出奇又被控制在一定的限度内，让观众感到有趣，于是就情不自禁地笑起来了。

从不定度出发，信息论定义了通信理论中“熵”的概念，从泛系理论的观点来看，这些研究属于一些特定背景下对于异同关系和泛序的特定的分析。

有了对异同关系和泛序的定量分析，就有可能得到概念(往往要使用语言文字来描述)的抽象，而观察概念之间的关系可以得到一些判断，知识就是一些判断组成的系统，判断中往往蕴涵着逻辑关系。罗素对逻辑与知识之间的关系作了深入的研究，其认识和研究成果比较集中地体现在他的《逻辑与知识》一书的表述中。按照罗素的观点，知识和知识体系是可以有一套逻辑规则来表达的。我们认为逻辑规则可以被视为一套关于关系和关系运算的系统。

技能是运用知识解决问题的能力，知识与技能的表达方式是我们探索教育学原理中必须搞清楚的问题，只有明白了知识的表达方式，那么教学实践中的可操作性才有可能实现。而有了对知识的结构的概念之后，我们便可以利用数学工具开始建立对知识的系统化的描述了。

人工智能的成就之一是研究了使用图论作为工具来表达知识的可能性。同一套知识系统可以使用文字来描述，可以使用图来描述，也可以使用代数中的矩阵来描述。

反演变换与再现变换

师者，传道授业解惑也。学员能否在学习期间就以实践入手来掌握计算机科学的要领很大程度上取决于教师的传道授业解惑的水平，而这又涉及到教育体制、教学思想、教学方法、教学内容等诸多方面的问题。

这里有几个简单而基本的事实是我们不能忽视的：

一、教学活动在师生之间从时序上看总体是串行的，但是在某些时间段内教学内容必须有某种平行、交叉以及重复。

二、计算科学是一个有机的整体，在计算科学基础上发展起来的计算机技术是一门综合性的技术，虽然计算科学和计算机技术中的某个专题具有相对独立性，但是它们都不能被机械地肢解剖裂开来。

三、计算科学在实践中具有非常强的可操作性。离开了软件的源代码，就无法真正理解计算的本质，也难以学会并精通编程功夫。

从教师的角度看，教学的过程是将自己的知识或者能力结构在学生的客体上进行反演的一种变换。通过教学活动，教师将自己掌握的知识传授到学生的头脑中去，学生则利用从教师提供的输入信息在自己的头脑中“从无到有”建立知识的结构。

伟大的数学家和数学教育家 G·波利亚说：教师最重要的任务之一是帮助他的学生。这个任务并不很容易，它需要时间、实践、奉献和正确的原则。

学生应该获得尽可能多的独立工作经验。但是，如果把问题留给他一人而不提供任何帮助，或者帮助不足，那么他就根本得不到提高。而如果教师的帮助太多，就没有什么工作留给学生了。教师应当帮助学生，但是不能太多，也不能太少，这样才能使学生有一个合理的工作量。

如果学生没有能力做很多，那么教师至少应该给他一些独立工作的感觉。要做到这一点，教师应当谨慎地、不露痕迹地帮助学生。

然而，最好是顺其自然地帮助学生。教师应该把自己放在学生的位置上，他应当看到学生的情况，应当努力地去理解学生正在想什么，然后提出一个问题或者指出一个步骤，而这正是学生自己原本应该想到的。

显然，教学只有在学生理解的基础上才能往前实施，教学内容应该顾及学生原有的知识结构和理解能力，如何判定学生的理解状态就成了教学中的一个关键。如果把学员的知识和技能当作一张复杂的图，教员在教学中通过某种手段在头脑中也应该具有这样一张图(不然教员就无法理解学员在那一时刻在想什么问题)。

当教学实施时，学员头脑中的图在不断地变化着，引起学生理解能力的变化。教员必须不断地判断学生理解力的状态起伏，找出学员理解力的不动泛系，因此这两张不断变化的“地图”必须动态地找准不动泛系(不动点)。

教员将学员的图读出并与自己头脑中的图比较(以找出学生理解状态在某一时刻的不动泛系)的过程称为的“反演”，而学员的图从不懂的状态变化到懂的状态的过程是知识和技能“再现”。反演和再现是教与学不可分割的两个方面。如何实施反演的变换是黑客道教育方法的研究对象，而如何实施再现则是黑客道学习方法的研究重点。

教师可以通过提问、观察学员动手实践的结果等多种方法了解、判断学员的理解能力达到了一个什么状态。孔子是这方面的专家，孔子的教学方法是纯启发式的，他不搞填压教学。孔子最得意的学生颜回说：“夫子循循然善诱人，博我以文，约我以礼，欲罢不能。”孔子的诱导就是建立在不断判断学生的不动泛系基础上的，他只有理解了学员头脑中思考的状态，才能为他提出问题创造条件。

孔子本人作为学生又是实施再现变换的高手，《史记》的“孔子世家第十七”中就生动地记载了孔子向师襄学习弹琴时进行再现的事迹：

孔子学鼓琴与师襄子，十日不进，师襄子曰：“可以益矣。”孔子曰：“丘已习其曲矣，未得其数也。”有间，曰：“已习其数，可以益矣。”孔子曰：“丘未得其志也。”有间，曰：“已习其志，可以益矣。”孔子曰：“丘未得其为人也。”有间，有所穆然深思焉，有所怡然高望而远志焉，曰：“丘得其为人，黯然而黑，几然而长，眼如望羊，如王四国，非文王其能谁能为此也！”师襄子辞席再拜，曰：“师盖云《文王操》也。”

这段记载表明，孔子在事先不知道答案的情况下，通过自己的反复演习和再现音乐，居然还原出了乐曲的原作者及其形象，可见作为学生的孔子，再现能力是多么高超！

知识与技能在学生的头脑中从无到有、从陌生到熟练、从熟练发展到高度技巧化是一个相当复杂的过程。古代的教学讲究“切磋琢磨”之说。²¹后人将这种方法比作学习过程中精益求精的过程，我们也可以用这些字词来描述黑客不断逼近高超技能的过程。

横看成岭侧成峰，
远近高低各不同。
不知庐山真面目，
只缘身在此山中。

²¹ “切、磋、琢、磨”的本意是古代制作玉石的方法：切是将含有玉的石头切割成较小的石块，磋是将石块中的石头部分去掉，只留下需要的部分，琢是将留下的部分进行加工，得到造型，磨则是将得到的造型打磨抛光得到最后的成品。

这是苏东坡所作的《题西林壁》，这首七绝是对泛系相对论极好的注解，也是对黑客道教学思想中反演和再现相结合的最好解说。教学双方必须多次完成“岭”和“峰”的互换，“山中”和“山外”多次表里互换才能得到对黑客道的体验。每一次这样的转换都是在视为对心智图像的某个拓扑空间的对带泛权维数的折叠、卷曲、重合等操作。

另一方面，反演和再现变换下的不动泛系原理还可以解释清楚为什么一流的教师和一流的学生之间实施教学时学生可能不会成才、或者不能顺利成才、不能快速成才的根源：

现在的计算机专业的大学生在三到四年的时间内要学二十到三十多门专业课程。目前高等院校中，基本上是一位先生讲授一门课程，因此每位学生在几年的大学生涯中要接受许多位老师的指导，目前的教学计划是分门别类的，因此每位教师都在学生的头脑中打入了点，而点与点之间的连线(联系)则没有有意识地建立起来。尽管学生在毕业之前要完成毕业设计，但是时间太短，来不及将所有学到的东西融会贯通。因此结果是大多数学生在大学毕业后知道的名词术语一大堆，但是这些名词术语之间的联系是什么却没有建立，所学到的知识是处在零散的状态，没有融合为一个有机的整体。在大学生毕业之后，学生往往还需要花费相当长的时间在实践中积累经验，才能成为具有开发能力的程序设计人员。

不动泛系原理与认知距离度量

布劳威尔证明的不动点定理和相关的不动点理论虽然告诉了我们不动点的存在性，但是却没有说明某个变换下的不动点具体在什么地方，这个问题在数学家研究了长达半个世纪后，由美国耶鲁大学的斯卡福教授(Herbert E. Scarf)于1967年取得了重大突破，他提出了一套算法，由这套算法出发数学家们得到了一系列的寻找不动点的卓越成果。

斯卡福教授的算法细节涉及许多数学技巧，这里不作展开介绍，其大意是利用由有限个点的系列来逼近不动点，从而寻找到不动点。受到这一算法的启发，结合自由软件的教学实践，黑客道创造了一种判断学生的不动泛系的算法，用以在教学中度量学生心理认知距离，作为黑客道教学追求可操作性的理论基础。

在黑客道中，我们可以将学员从不懂到理解这一过程抽象为一条多维空间中的一条曲线(即学习曲线)，多维空间的坐标系统由成才七故(当然还有其他建模方式)而组成，曲线的起点为学员不懂状态的泛系点，曲线的终点为学生理解后的状态所在的泛系点。

教员的任务是在教学过程中经常判断学员位于这一曲线中的什么位置(每一次判断都是寻找不动泛系的过程)，学员当前的状态离理解的状态还有多远，

问题的实质是“距离”的度量。只有距离摸准了，学员下一步应该朝哪个方向走最好才会有较好的答案，只有能够准确地度量心理认知距离，开展教学活动才能有科学量化的依据。

有趣的是，在教学实践中，教员往往可以反演出学生头脑中的多个不动泛系，而不只是一个不动泛系。教员可以从多个不动泛系中寻找一个最适合学员往前深入的不动泛系。由于每一次反演都有可能得到多个不动泛系，某种意义上说，对认知距离的度量没有一成不变的模式，具体选择哪个不动泛系往前发展取决于教员的方法和经验，从理论上讲，有多少位教育家，就有多少种不动泛系的度量方法和运用方法。

黑客道的教学方法

建立在求解不动泛系和度量心理认知距离上的黑客道教学方法的显著特点有三：

- 目标牵引，内化成瘾。
- 实践：理论七三开。
- 强化简化抓关键，才学基础少而精。

下面几节详细说明这几个特点。

目标牵引，内化成瘾

子曰：“知之者，不如好之者，好之者，不如乐之者。”

爱因斯坦说：“兴趣是最好的老师。”

美国心理学家布鲁纳说：“最好的学习动因是学员对学习材料有内在的兴趣”。

为了让学员有兴趣学习“黑客道”课程，教员必须将计算机科学的美感全面地展示给学员，并且依靠美感使学员爱上编写程序这一工作，使学员上瘾，达到爱不释手的程度，使编写程序成为学员的终身爱好，并让这种爱好成为学员自己最好的老师。黑客道的教学方法是启发式的，而不是填压式的。

高明的教员善于调动学员自身的动因，他们除了教授学员基本的知识之外，更加注重教授方法，使学员可以做到举一反三，乃至举半反万。

“子罕言利，与命与仁。不愤不启，不悱不发，举一隅不以三隅反，则弗复也。”孔子用这种教学方式培养了不少出类拔萃的人才，例如端木赐，字子贡，“好废举，与时转货资，喜杨人之美，不能匿人之过。常相鲁、卫，家累千金”。按

照现在的话说，子贡就是当时工商界著名的人物，他善于言辞，曾经出使很多诸侯国，其外交活动改变了当时的国际形式：“存鲁、乱齐、破吴、强晋而霸越。子贡一使，使势相破，十年之中，五国有变。”

那么，趣味究竟是什么？或者问，教员应该如何激发学员的好奇心呢？

孟德斯鸠对于什么是趣味，以及趣味的构成要素作了深入的探索，他说：“当我们看到一事物对我们有用时感到快乐的时候，我们就说它是好的，当我们看到一事物时感到快乐，却没有发现它在当前有什么用处时，我们就说它是美的”。学员常有这样的体验：以往单调的政治学习之所以让人厌倦，很大程度上是因为它既不能给他们带来新的信息或者新的思想，又不能解决他们的实际问题，因此显得没有用处。于是乎，政治学习便容易流于形式了。

这些关于趣味的见解如何体现或应用在计算科学中呢？或者问，计算科学的美究竟体现在什么地方呢？

这是一个可以具有多种答案的问题。黑客道是从泛系理论来看待这一问题的。从泛系观看来，引起趣味的必要条件是事物必须拥有某种内在“序”，具有某种关系的相互转化(泛系五互)，以及泛对称性。我们认为计算科学首先是起源于数学的，因此对于数学美的认识原则也同样适用于分析计算科学的美。

泛系数学认为数学是研究事物或者泛系的量、度、异同、泛序、运转、广义量、形式与形式化、结构等的跨域性学科，其结构大多表现为泛对称、因果、泛导(广义的微积或者变变关系与运转)、泛转(由局整关系和形影关系生成的转化)、模拟的形式，各门数学可相对统一于形式广义系统或者泛导或者泛转模拟的研究。

对于数学理论群落中的计算科学，泛系从哲理和数理的层次对单值化作了深入研究，并从各个相关的数学分支抽象得到了统一的理解框架，成为指导计算机科学的学习、研究和创新的有力武器。计算科学中单值化很好地体现着泛关系、泛转化和泛对称性的美学原则。我们认为，**单值化是计算科学美的集中体现**，突出计算科学中的单值化的美学原则是黑客道培训计划与教学的最鲜明、最本质的特征之一。

教员不断地向学生展示计算科学的美可以使学生在黑客道的学习过程中发生正迁移这种心理现象。这种美学方面的系统训练可以对学生产生两个方面的影响：一是为学员提供一批思维定势，学生在掌握了方法并形成了这些思维定势之后，可以将方法适当地运用到实际问题的求解上。二是当学生原有已掌握的思维定势与现实问题不适应时，学生可以迅速地打破原有的思维定势，去尝试新的方法来解决实际问题。

实践：理论七三开

对计算机科学的美的感知必须通过学员自己的亲身动手实践来获得。

遗憾的是，笔者通过调查发现，国内条件很好的一批大学的计算机专业学生的上机编程实践的时间还不到全部学习时间的 30%，大部分的时间是由教员在向学员填鸭式地灌输知识，条件较一般的学校的学生上机编程的时间更少，有些学校因为师资力量、实验室设施的限制，以及不顾自身办学条件盲目扩大招生，导致学生实践的时间更是少得屈指可数。学生在毕业后找不到编写程序的感觉并不完全是学生不聪明，而是学生动手实践的时间实在太少了。一个从不下水或者很少下水练习的人是学不会游泳的。

黑客道鼓励学员上机动手实践，上机动手编写程序的时间应该占整个学习时间的 70% 以上。针对每一个学员的具体情况，教员应该事先给学员定出一个实践方向，并讲授一个一般的原则，然后布置学生练习的问题、上机实践，并观察学员的掌握情况，随时调整下一步的学习方向。教学时鼓励学员在编程实践中“犯错误”，鼓励学员在学习实践的过程中“犯遍所有的错误”，教员的职责之一就是当学生犯了错误时，帮助学生找出错误，告诉学员为什么错了，错在什么地方，纠错的方法是什么，将来如何避免犯同样的错误，下一步的学习方向是什么等等。

重视动手实践，动手实践与理论学习有机结合，从实践与理论学习的时间比例从三七开颠倒为七三开，是黑客道教学方法与现行高校计算机专业教学的本质区别之一。

强化简化抓关键，才学基砖少而精

庄子曰：“生也有涯，而知也无涯，以有涯随无涯，殆矣！”他的这种见解在当今社会上更有现实意义，我们生活的时代是一个信息大爆炸的时代，每天冒出的新生事物层出不穷，技术的发展真是一日千里，想学完所有的知识是不可能的。

然而，在方法论原则指导下的方法是有限的，是可以在有限的时间内学习并掌握的。这一思想在许多教育家的教学中得到了很好的贯彻。例如，塞戈维亚是上个世纪西班牙古典吉他演奏大师，也是著名的教育家，他培养了一大批优秀的现代青年吉他演奏家，据他的许多学生回忆，在他的吉他进修班上，绝大部分时间是在练习分解和弦，通过练习各种各样的分解和弦来练习指法。真正练习的大型曲目很少，但是他要求每个学生把一两首曲子真正练习到完美的境界。可见，塞戈维亚的教学方法是非常重视基本功训练的。只有这些基本功练好了，演奏高难度的曲目才会具有艺术的美，这也暗和了泛系成才法中的“才学基砖少而精、组合爆炸易适应”两句。

国际象棋、中国象棋和围棋中棋手讲究研究棋谱和定式，这些东西就是棋类艺术中的基本功，他们一有时间就反复推演。武术家每天都要拳不离手，他们无一不重视基本功的训练。

由少量的要素经过少量的方法组合，形成组合爆炸的道理是很容易理解的。音乐中的音符只有七个，音乐的组成要素也只有和声、节奏、音阶(旋律的演进)、曲式、对位、赋格等少量几种，但是就是这么一些简单的要素可以谱写出无穷的音乐篇章。魔方是一个正六面体，玩法只有转动这一种，但是这一种玩法可以拼出的排列组合几乎是无穷多的，UNIX 系统中的基本的实用工具约有三百个左右，但是程序员利用这批实用工具配合管道(pipeline)和一些控制结构就可以创造无穷的脚本程序(scripts)来完成五花八门的计算任务。

同样的道理，计算机黑客需要掌握的基本功并不多。掌握了这些基本功，黑客就可以在实际工作中，根据问题的特点，迅速学习新的事物和知识，从而获得驾驭方法、运筹事物的能力。

黑客道的教学内容

后世许多学究推崇孔子的强志博文，并将他给神化了。但是具有讽刺意味的是，孔子本人对此却是予以否认的，“予一以贯之”——他用一种基本原则贯穿于所有的知识之中，以现代的话说，孔子的才学基砖和技能结构是系统性的。

在泛系方法论和 GNU 工程的基础上，应用不动泛系原理，笔者创造性地开发了旨在培养黑客的一系列课程。与国际上流行的 IEEE 与 ACM 推出的教学计划和教学内容相比，具有独特的原创性，与国内绝大多数高等院校计算机专业的现行的教学方法则是完全不同。

黑客道的培训课程教学内容是在黑客道的教学思想指导下发展和完善起来的，它试图将所有计算机科学分支当作一个有机的整体，以一种普适的原则来统帅各门课程的教学，这个普适的原则就是泛系方法论。方法论是高度概括而高度抽象的，方法论统摄、指导实践，源代码则是“器”，当然其中有“神”。这个精神与自由软件的源代码想结合起来体现在黑客道的教学实践中。

已经开发的黑客道培训课程共有十二门：

1. 泛系导论与数学基础
2. 计算机经典论文选读
3. 计算机体系结构与组成
4. 计算机程序的结构与解释
5. 基本编程工具
6. 软件文档创作

7. 数据结构与算法
8. 编译原理与编译器构造
9. 信号的分析与综合
10. 操作系统原理与构造
11. 数据通信与网络协议
12. 网络数据处理与数据库

下面依次解说各门课程的内容与相互关系。

泛系导论与数学基础

对于从事软件开发的黑客而言，无论怎样强调数学的重要性都不过分。但遗憾的是，数学课程，无论它的哪个分支，总是背上了难学、难懂的坏名声，这在相当大的程度上是不应该的。数学家 H. Weyl 就说过：“数学因它总是以抽象的方式来讨论问题而弄得名声狼籍，其实这个坏名声只有一半是该当的。”

除了抽象思维的特性之外，数学大量采用公式和符号也是一个明显的特点，在许多人心目中，数学充满了古怪、离奇的数学符号，数学书籍和论文就像“天书”一般难懂。数学家 H. Töplitz 说：“数学由于其语言、标记方法和奇怪的符号，像是被一座高墙与外世隔绝。对于墙外的人，他觉得墙里面大部分都是秘密的、沉闷的、平凡的数字，是那些按照不可逃避的必然规律起作用的、没有生气的数学家。”那些墙外的人对于数学符号当然不会有喜悦的积极情绪。

教师的错误引导也是使事情变得更糟糕的一个因素，美国新泽西州普林斯顿“教育测试服务中心”(ETS)在 1956 年发表了一篇论文，里面提到的情况至今没有根本的改观：“.....数学在各个课程中是最不得人心的一门功课，其名声不佳..... 未来的教师通过初级学校学会讨厌数学..... 他们返回初级学校又教育新一代讨厌数学。”

更加糟糕的是，墙内的许多数学家和他们编写的相当数量的数学教材还没有意识到问题的严重性，他们的数学论文和教材鲜有背景介绍，几乎没有顾及到教育心理学方面的要求。

苏联数学家阿诺尔德曾经写道：“对于我来说，要读当代数学家们的著述，几乎是不可能的，因为他们不说‘彼嘉洗了手’，而是写道：‘存在一个 $t_1 < 0$ ，使得 t_1 在自然映射 $t_1 \mapsto (t)$ 之下的像属于脏手的集合，并且存在一个 t_2 ， $t_1 < t_2 \leq 0$ ，使得 t_2 的像属于脏手的集合的补集。’不过，有几位数学家——比方说米尔诺和斯梅尔——所写的文章，是仅有的不这样故弄玄虚的例子。”

连阿诺尔德这样的数学大师阅读数学论文都感到困难重重，更何况我们普通人呢?!

根据笔者的经验和调查，这是目前数学教学中普遍性地不重视直觉开发和训练的后果。学生如果没有理解数学思维的本质，并具备相当的数学直觉时，看到的只有一堆堆枯燥乏味的公式和符号，这是绝大多数人不喜欢甚至厌烦数学的根本原因。

正是因为传统的数学教学不充分重视数学直觉的培养和在思维中的作用，因此绝大多数学生觉得入门难，很多初学者在攻打数学的城池时就这样被挡在了城外；或者牺牲在数学的城墙上；能够入门者，往往又被诸多细节问题纠缠住，一辈子无法从数学的城池中跳出来，淹死在数学的汪洋大海里；而极少能够出来的人出来时已经是两鬓苍苍、头发全白。

泛系理论提供的方法论是极具特色的，我们之所以开设《泛系导论与数学基础》这门课程，就是希望学员能用一种统一的观点来看待计算科学的各个方面。泛系方法论中有许多关于计算本质的深刻认识，其中之一就是“泛系显微原理”，泛系显微原理的核心就是单值化，而单值化恰好正是是计算科学的一个核心问题。没有一定形式的单值化就无法谈及计算、可计算性、控制论中或者抽象自动机中的可观测性。一旦明白了单值化的意义，就可以利用泛系理论重新认识全部计算科学理论，就可以对计算科学中的数学教学内容和讲授方法进行重新的安排，并从泛系理论的观点出发，站在新的高度来创作计算科学的教材。

计算科学的重点在于解决计算中的各种问题，它侧重于计算的问题的能行性，计算科学侧重于在有限的步骤中求出问题的解，它特别倚赖于数据结构和算法的研究和实现，这是计算科学有别于与其他传统的数学理论的一个特征。

数论

数字计算机从根本上讲只做两件事情：算术运算和逻辑运算。因此数论和数理逻辑对于计算机专业的学生就是基础课中的基础了。数论的内容大致可以分成两个部分讲解：研究整数的初等数论，和以解析方法作为研究工具的解析数论(选修)²²。

初等数论主要研究整数的性质和方程(组)的整数解，多项式则是代数学中的一个非常基本的研究对象，整数与多项式两者初看起来似乎彼此并不相同，但是整数和多项式中的许多概念、结果和方法是近世代数中抽象概念的基本模型和源泉，在当今计算机科学与工程中发挥的作用越来越大。

²² 在学完初等数论后，有兴趣的学员还可以学习一些解析数论的内容。解析数论在近一百多年来，通过不断引进解析方法来研究素数分布，孪生素数，Goldbach 猜想，Waring 问题，整点问题，整数拆分等著名数学问题而发展起来的。在“信号的分析与综合”这门课程中，我们将运用解析数论的一些工具对计算机系统信号进行观察与控制。

数论的基本内容包括：算术基本定理，数的整除性，同余，原根和指数，二次剩余定理，不定方程，域上的一元多项式，代数基本定理，整系数多项式，多元多项式等。

算术基本定理的表述就是单值化的一个具体体现：一个整数可以分解为一个或者多个素数的乘积。这是一种典型的泛对称关系。

离散数学

传统的离散数学分为集合论、数理逻辑、图论、代数结构、组合数学等五个分支。目前在高等学校计算机专业的数学教学实践中，普遍存在着分析数学的教学比重过大、离散数学的比重偏小的问题。当然，我们绝对不能否定分析数学的作用，但是对于计算机专业的学生而言，掌握计算机专业知识时，离散数学的功底如何，则直接左右其对计算机科学的认识水平。

离散数学的几大分支学科与计算机科学密切相关，它们都可以在泛系数学的观点下来组织教学。

- 集合论

集合论是离散数学的一个重要组成部分，实际上，集合论是几乎所有现代数学的基础，没有集合论而描述一个现代数学概念几乎是不可能的了。

选择公理是集合论的一个基本公理，它就很好地体现了单值化的美学原则。它告诉了人们为什么可以以偏概全，它是“强化简化抓关键”的数学依据。当然，理解选择公理需要讲解与此相关的许多教学内容，包括：集合，关系与映射，基数，序数，极大原理，格，代数系(后面接上“代数结构”)。在掌握了集合论的基础上，学员可以选择性地学习“点集拓扑学”，内容包括：拓扑空间与连续映射，连通性，有关可数性的公理，分离性公理，紧致性，积空间，映射空间与商空间，完备的度量空间。

在学习了集合论之后，结合泛系导论的讲授，就可以引导学生学会如何“强化简化抓关键”了。

- 数理逻辑

数理逻辑是数学和传统逻辑相结合的产物，它是研究推理、特别是研究数学中的推理的科学。数理逻辑使用数学符号来研究数学各个领域共同使用的逻辑推理。数理逻辑的一个核心就是如何归约一个逻辑系统。归约的实质就是一种单值化。

数理逻辑是计算机基础课程中非常重要的一个环节，在“计算机程序的结构和解释”中，我们选择 Scheme 语言作为编程语言的目的之一就是深化学员对于数理逻辑的认识和理解。

数理逻辑也是计算机基础课程中的一个难点，为了得到好的教学效果，应该从历史发展的角度来讲授这门课程，向学员提供这门学科的背景。数理逻辑的发展迄今经历了三个阶段，每一个阶段对计算机科学都有影响：第一阶段是以莱布尼兹的开创性工作开始的，布尔代数和关系逻辑是这一阶段的成就标志，在计算机编程实践中，布尔代数和关系逻辑运算比比皆是。第二阶段是与集合论成为数学基础密切相关的，这一部分的内容在集合论中介绍。第三阶段是以哥德尔定理为标志开始的，研究重点与计算的能行性有关。

数理逻辑的教学内容是很丰富的，与命题逻辑相关的内容：真值联结词，真值函数，重言式；命题演算，命题逻辑的公理化和形式化；范式，完全性，一致性，公理的独立性；不同的命题逻辑，古典命题逻辑的不同的公理化等。

范式的讲解与语言规范的讲解相关，例如在“计算机程序的结构与解释”这门课中，我们将学习 Scheme 语言的规范，称为 R^5RS ，它是用“扩展巴克斯-劳尔范式”(EBNF) 严格定义的。

与狭谓词逻辑相关的内容：狭谓词逻辑里的形式结构，狭谓词演算，演绎定理，范式，判定问题，一致性和完备性，狭谓词逻辑的不同系统，有等词的狭谓词演算，摹状词。

谓词演算对于学习 Scheme 语言是相当重要的，在“编译原理和编译器构造”中对于理解 GCC 编译器中的代码也很重要。

- 图论

我们经常有这样的感觉，在描述一个概念或者事物之间关系时，千言万语的文字也比不上一幅良好的插图来得直接了当。图论是数学中处理离散对象的一个重要工具，其教学内容有：图的基本概念，通路问题，图的矩阵表示，树，穿程问题，二分图的匹配问题，平面图及色数。带权图及其应用，嘉量原理等。

应用图论的一个基本思想是利用图来描述离散对象基于关系，然后通过图的矩阵表示，将非线性的问题转化为线性的问题来研究和处理，而这也是一个单值化的方法。

- 代数结构(选修内容)

教学内容：代数系统，半群与独异点，群，环与域，格与布尔代数。

- 组合数学(选修内容)

组合数学是围绕三大存在定理展开的：广义鸽洞原理(Ramsey 定理)，把半序集分解为不相交链的 Dilworth 定理，集合的相异代表组的存在定理

(Hall 定理)。这三大组合数学定理都是单值化的特定形式。教学内容包括：组合存在定理，基本的计算公式，组合计数方法，组合计数定理，组合设计与编码，组合最优化问题。

递归论

教学内容包括：初等函数集，原始递归函数，递归函数集，算子，递归枚举性，判定问题(其中 **Church–Turing Thesis** 是计算机理论的重要基础之一)，谱系及计算复杂性，规约与不可解度等。

与其他黑客道培训课程的关系：递归论是“计算机程序的结构与解释”的数学基础，递归论的一个重要结论就是，可以使用前继函数、后继函数、测零函数和不动点算子四个要素来构造任何可计算函数，它是函数型编程语言的数学基础之一。

线性代数

教学内容：行列式；线性方程组；矩阵；线性空间与线性变换；欧氏空间与二次型。

数学分析

黑客道有意识地降低了分析数学的教学时间的比重。一是可以腾出更多的时间让学员学习离散数学，二是学员可以在以后的实践中需要相关的分析数学知识时加以补充。与组合数学类似，分析数学可以视为是围绕三大基本定理展开的：隐函数定理、Taylor 余项定理、Gauss–Stokes 公式(定理)。这三大基本定理可以围绕着泛系单值化原理统一起来学习。

概率论

教学内容：随机性与概率；独立实验概型；一维随机变量及其分布；多维随机变量及其分布；随机变量的数字特征；中心极限定理与大数定律；数理统计方法介绍(可选)；平稳随机过程简介；时间序列分析；随机模拟。

除了泛系导论与集合论，其他的教学内容都可以随着黑客道的其他课程讲授时再展开或者补充。

计算机经典论文选读

子曰：“温故而知新”。

伟大的法国数学家庞加莱曾说：“如果我们想要预见数学的将来，适当的途径是研究这门科学的历史和现状。”

其实，学习任何一门科学，我们都必须通晓这一科学的历史背景，理解这一科学的来龙去脉。我们从浩如烟海的计算机专业文献库中，遴选了自 1936 至 2001 年期间的各个历史时期计算机科学家发表的对计算科学和计算机技术具有重大影响的计算机经典论文近 30 篇²³：

1. “On Computable Numbers, with an Application to the Entscheidungsproblem”, Alan Turing, 1936。《论可计算的数》一文是阿兰·图灵于 1936 年发表的，这篇论文被公认为是现代计算机科学诞生和独立的标志，黑客道课程将结合对图灵机计算模型的介绍，讨论计算、逻辑与数学的关系。
2. “First Draft of a Report to EDVAC”, John von Neumann, 1945。冯·诺依曼是著名的“冯·诺依曼体系”的提出者，其核心思想后来演变为程序与数据等价的观点。
3. “Three Models for the Description of Language”, Noam Chomsky, 1956。乔姆斯基提出了语言处理的有限自动机模型，在语言学、数学、逻辑学、计算科学之间建立了一座桥梁，它是当今编译理论的基础。
4. “Recursive Functions of Symbolic Expressions and Their Computation by Machine”, John McCarthy, 1960。麦卡锡是 Lisp 语言的发明人，他提出了基于符号表达式的 λ 演算的计算机实现方案(Lisp 语言)，开创了符号计算的新领域。
5. “Revised Report on the Algorithmic Language ALGOL 60”, Peter Naur (ed.), 1963。这篇文章在巴克斯工作的基础上提出了巴克斯-劳尔范式，成为描述编程语言规范的形式化工具。
6. “PL/I List Processing”, Harold W. Lawson, 1967。这篇文章首次明确提出了指针变量的概念与实现方式，对以后的计算机软件设计产生了巨大影响。
7. “Multiple Computer Networks and Intercomputer Communication”, Lawrence G. Roberts, 1967。这篇文章提出了互联网的模型和原理。

阅读时可以参考“The Past and Future History of the Internet”, Venton Verf, Robert Kahn, et al, 1997。这篇文章回顾了互联网的发展历史和未来的发展趋势。

²³ 这些论文主要是以软件人员的角度而选择的。

8. “Forth — a Language for Interactive Computing”, Charles H. Moore and Geoffrey C. Leach, 1970。这篇文章详细地说明了堆栈是如何被引入到编程语言中, 并加以利用的, Forth 语言是在软件中倚赖堆栈进行计算的语言, 对后期的许多编程语言有很大影响。
9. “System R4 Relational”, Dr. Edgar F. Codd, 1970。这篇文章提出了关系性数据库的思想, 关系性数据库与网络性数据库和层次性数据库是三大数据库流派, 目前实际应用中(特别是在企业级别的数据库产品应用中) 关系性数据库产品为主流产品。
10. “The UNIX Time-sharing System”, Dennis Ritchie and ken Thompson, 1974。UNIX 操作系统是计算机发展历史上一个划时代的操作系统, 而且至今仍然被广泛使用。

阅读时可以参考“*The Development of the C Language*”, Dennis M. Ritchie, 1996。C 语言是为开发 UNIX 而发明的编程语言, 兼有高级语言和低级汇编语言的特征, 是目前开发系统软件的首选编程语言。

11. “Pansystems Methodology: A Transfield Investigation of Generalized System-Transformation-Symmetry”, Wu XueMou, 1976。吴学谋教授的这篇“泛系方法论: 广义系统-转化-对称性的跨领域研究”是网络型科学诞生的标志。
12. “The WEB System for Structured Documentation”, Donald E. Knuth, 1983。高德纳教授是世界著名的计算机科学家, 他对算法理论有非常深入的研究, 他发明的 $\text{T}_{\text{E}}\text{X}$ 排版系统是将计算机科学理论与印刷行业完美结合的产物, 这一软件导致是现代出版技术革命的发端, 它对科技信息的自由传播产生了重大影响。

阅读时可以参考“*Literate Programming*”, Donald E. Knuth, 1984。“作文式编程法”是受元数学的方法论影响和启发的产生的, $\text{T}_{\text{E}}\text{X}$ 系统中底层的 WEB 模型是采用作文式编程法开发的第一个计算模型。

13. “The GNU Manifesto”, Richard M. Stallman, 1984。

《GNU 宣言》是计算机科学家在软件领域道德自省后理性思考的结果, 由这一宣言引发的自由软件运动极大地改变了计算机工业的格局。

阅读时可以参考“*The GNU Operating System and the Free Software Movement*”, Richard M. Stallman, 1999。开发 GNU 操作系统是自由软件运动核心工程, 是为保护和培养自由软件运动的开发而创建的, 其目标是开发一个类 UNIX 的操作系统, 以及一个符合 GPL 的完整工具集。在 GNU 工程发轫后, 它得到了众多黑客的支持, 成为目前自由软件运动发展的平台。本文回顾了自由软件思想诞生的经过, 阐明了伟大的 GNU 工

程的实质与自由软件思想的精髓，剖析了自由软件运动的发展现状和面临的挑战。

14. “The Quantum Theory, the Church–Turing Principle and the Universal Quantum Computer”, David Deutsch, 1985。“量子理论，丘奇–图灵原则与通用量子计算机”。大卫·多依奇的这篇论文正式提出了量子计算的模型，实用的量子计算机尽管还没有制造出来，但是计算模型已经提出。
15. “The X Window System”, Robert W. Scheifler et al, 1986。X 视窗系统是将分布式计算与图形用户界面首次成功结合的产物。
16. “Information Management: A Proposal”, Tim Berners–Lee, 1989。这篇论文提出了万维网的计算模型，引发了网络出版的革命。
17. “The Programming Language Oberron”, Niklaus Wirth, 1990。Wirth 教授是 Pascal 语言的发明人。在总结了 Pascal 语言设计的得失的基础上，他提出了编程语言可以扩展为完整的操作系统的思想，并开发了 Oberron 来予以验证。

阅读时可以参考“Revenge of the Son of the Lisp Machine”, Matthew Flatt, et al, 1999。Lisp Machine 是二十世纪八十年代流行的一种计算机体系结构，后因种种原因退出了市场，但是“编程语言就是操作系统”的思想最近因为硬件技术的飞速发展又开始复苏。²⁴

阅读这篇文章时还可以参读“Scripting: Higher Level Programming for the 21’st Century”, John K. Ousterhout, 1998。随着计算机硬件的飞速发展，计算机系统硬件成本大幅度下降，但是软件系统中程序员的成本不降反升，因此如何提高程序员的开发效率是目前的一个主要课题。脚本语言由于其高度的模块化和对象化，比系统编程语言更加贴近人的思维特点，因此越来越受到重视。本文说明了为什么脚本语言将是二十一世纪程序员编程的主要工具。

18. “Molecular Computation of Solutions to Combinatorial Problems”, L. M. Adleman, 1994。目前的计算机都是电子计算，而分子计算机则由于电子计算机无法比拟的超大规模平行计算能力最近受到计算机科学界的高度重视。
19. “Network Computing in Beowulf Clusters”, Thomas L. Sterling, 1995。群集计算是利用多台计算机协同工作得到高计算性能的一条有效途径。
20. “Main Principle of E2K Architecture”, Boris Babayan, 2000。天才的俄罗斯学者将软件的资源可分割的原理创造性地应用到了计算机通用中央

²⁴ 笔者的 MING/OS 也是受到了这一思潮的影响。

处理器芯片的设计中，使得系统硬件资源可分割和按需调度，得到了低功耗、抗病毒、高性能的新型 CPU 芯片设计。

21. “The Anatomy of the Grid”, Ian Foster, et al. 2001。网格计算是下一代计算机网络发展的主要方向之一，它是以高性能计算机、高性能数据库和宽带通信网络为基础发展起来的。

在教员的指导下，学员通过阅读这些经典论文的原文可以建立对计算机科学的轮廓或者整体观念，培养对计算机英语的理解能力。²⁵

另外，尽管汉语在国际上的重要性日益上升，但是目前计算机科学界和工程界的绝大多数论文和专业文献都是用英文发表的。因此，作为一个黑客，英语是必须掌握的语言。掌握了这一工具，就可以为从事黑客事业(hacking)带来无穷的便利。

计算机体系结构与组成

这门课的内容看起来比较空泛，但实际上有着非常具体的教学内容。黑客道教学的重点是结合“泛系导论和数学基础”中的内容，说明图灵计算机的原理，计算机是如何从这个模型演变到个人计算机的体系结构的？组成计算机的零部件是什么？这门课程还应与“操作系统原理与构成”、“信号的分析与综合”等结合起来讲授。

计算机系统的层次结构；组合逻辑与微程序控制器；运算方式和处理器(Intel x86 的 32位的微处理器)；存储器，高速缓存技术，存储系统与寻址方式；指令系统；局部总线，系统总线，通信总线；时钟与定时器，计数器，DMA 控制器；中断与中断控制器；串口与并口；键盘和显示器；其他计算机外部设备。

这门课程的延续时间较长，实施教学的时间可以当学员研读了“计算机经典论文选读”中的有关内容后开始，学员对教学内容会有几次反复的由表到里、再由里到表的学习过程。黑客，尤其是对底层的系统软件感兴趣的黑客必须掌握这门课程的内容。从这门课程开始，学员开始接触到计算机的硬件，为在“操作系统原理与构成”中学会从裸机零部件开始组装操作系统打好基本功。

计算机程序的结构与解释

作为一名黑客，编写程序当然是最基本工作。目前绝大多数高等院校的计算机系都开设了 C 或者 C++ 编程语言作为学生的必修课。无疑地，C 语言应该是黑客必须掌握的编程语言，但是对于初学者，学习 C 语言的难度是相当大

²⁵ “一、百、万”工程有一个出版计划，拟针对这 20 篇经典论文，翻译中文译文，并利用泛系方法论进行深入分析和评论。

的，这是因为 C 语言最初是为了开发 UNIX 操作系统而发明的，因此它是系统编程语言，由于 C 语言具有存储管理等低级(靠近硬件特性)的功能，因此它自从诞生后一直是系统编程的首选。

初学者难于掌握 C 语言的原因在于学习道路上有五个拦路虎：神秘的硬件、古怪的操作系统、困难的数据结构与算法理论、巨大的库函数、魔术般的 C 编译器。初学者往往很难在较短的时间内消灭这几个拦路虎。

以往大量的编程实践还表明，黑客只会一门编程语言是不够的，至少应该学会以下三种语言：Lisp (Scheme)、C、汇编语言。黑客道培训计划的开发者在长期教学实践的基础上，结合对计算机编程的理解，为学员提出了一种与现在高校计算机系完全不同的学习编程语言的顺序：先学习 Scheme 语言，然后学习 C 语言，最后学习汇编语言。

Lisp 是函数型编程语言。为什么在黑客道的培训中，我们如此强调函数型编程语言的重要性呢？这是因为函数型编程语言让学生将精力集中于解决问题的算法实现上，而不是在问题求解在计算机内部是怎样具体实现的细节上。函数型编程语言具有数学基础清晰明了、程序简洁、易于进行程序推理和正确性证明等特点，它与人工智能、并行处理（特别是大规模平行处理能力的新型计算机系统）、软件工程等领域密切相关。

函数型编程语言有很多种，例如 Lisp、Haskell、SML 等。黑客道培训计划选择了 Scheme 语言作为培训的入门语言。Scheme 语言是 Lisp 语言的一个现代变种，诞生于 1975 年，由 MIT 大学的 Guy Lewis Steele Jr. 和 Gerald J. Sussman 两位教授发明，Scheme 有一组精心挑选的特性组合，语言的规模很小，但是功能却非常强大，被称为是编程语言王国的“皇后”。

Scheme 与 Lisp 一样，里面有各种各样的“原子”，因此它是从底向上构造程序的理想编程语言。原子论的思想源自古希腊的留基波和德谟克里特，后来被化学家道尔顿证实，其思想在现代物理学中的夸克学说中仍然得到了保留和发展，基于原子论的模块化思想是每一位黑客都应该掌握的。

Scheme 语言又是针对符号计算的编程语言，符号与计算的关系，早在古代就被斯多葛学派注意到了，这一学派哲学家提出的符号问题，就是为了建立严密的三段论逻辑的需要，后来由莱布尼兹发展，到现代的符号主义流派，都反应了符号与计算的密切关系。吴文俊研究员提出的数学定理机械化证明也可以视为符合这一哲学思想的一种探索。

具体地讲，Scheme 具有以下精心挑选的编程特色：

- 词法定界(Lexical Scoping)
- 动态类型(Dynamic Typing)
- 良好的可扩展性

- 尾递归(Tail Recursive)
- 函数可以作为值返回
- 支持一流的计算连续
- 传值调用(passing-by-value)
- 任意的算术运算精度

在笔者写作本书时，世界上已经有 275 所大学和学院采用 Scheme 语言作为计算机专业学生的入门课程，仅美国就有包括 MIT 在内的 104 所大学采用 Scheme 作为大学计算机专业的入门课程。除大学外，美国还有 51 中学采用 Scheme 作为学习计算机程序设计的语言。可见计算机科学界和教育界对于这一语言的评价之高。

目前这门课程采用了 Harold Abelson, Gerald J. Sussman, Julie Sussman 等合著的 *Structure and Interpretation of Computer Programs*。这是一本经典著作，被 MIT 计算机系选为入门教材。

教学内容包括：对计算过程的抽象，对计算数据的抽象，对计算状态的处理，元语言特性，在基于寄存器的计算机系统上实现 Scheme 语言(学完这门课程与其他相关课程后，学生在毕业设计阶段应该可以在教员的指导下动手自己设计一个简单的 Scheme 解释器。)

读者也许会问：为什么黑客道培训计划中没有讲授 C++、Java 等编程语言？

答案很简单：一旦掌握了 C 语言，C++ 和 Java 等语言都很容易学会。

基本编程工具

“工欲善其事，必先利其器”。

“磨刀不误砍柴工”。

这些古训都说明了熟练使用工具的重要性。GNU 工程的一个伟大之处在于它提供了一组软件开发工具，从源代码的编辑器 GNU Emacs，到程序的编译器 GCC、调试器 GDB、项目管理工具 Make、版本控制系统 CVS，等等。GNU 编程工具就像是木匠手中的斧头、锯子、木线等基本工具。木匠没有熟练地掌握这些工具是难以制作出一个像样的家俱的。同理，黑客如果不掌握这些 GNU 编程工具是无法动手进行真正的软件开发工作的。

我们从 GNU 工程大量的开发工具中精选了六种最为基本的编程工具：GNU Emacs、GCC、GDB、GNU Building System、CVS、Bash。²⁶

²⁶ 另一个重要的基本工具 Texinfo 放在“软件文档创作”那门课程中与 TeX 一起讲授。

GNU Emacs—文本编辑器

黑客们常用的编辑器有两种：GNU Emacs 和 vim。vim 是从 UNIX 平台上的 vi 演变过来的，程序的二进制代码的体积较小，运行速度较快，但是扩展性一般；黑客道培训计划采用了 GNU Emacs 作为首选的编辑器，因为 GNU Emacs 具有良好的可扩展性，而且它为黑客提供了一个完整的工作环境。

GNU Emacs 是“the Extensible, Customizable Self-Documenting Display Editor”。GNU Emacs 内嵌入了功能强大的 Emacs Lisp 解释器，因此是一个黑客的完整的工作环境，这门课程中，我们不讲授 Emacs Lisp 编程的内容(Emacs Lisp 编程的内容放在“计算机程序的结构与解释”中讲授)，重点放在 GNU Emacs 的文本编辑功能的使用上。

根据我们的教学体会，尽管 GNU Emacs 的命令很多，但是常用的命令却只有约 20 个左右，掌握了这 20 个左右的编辑命令，黑客马上就可以动手开发程序了。

GCC—编译器

编译器是一个非常复杂的软件，在这门课中，我们只介绍如何使用 GCC 编译器，关于编译的原理与 GCC 内部的构造，将留在另一门课程“编译原理与编译器构造”中讲授。

GNU Debugger—调试器

GDB 是 GNU 工程的符号调试工具，总体上看，GDB 可以完成四种任务：

- 启动一个程序，在程序中标明可以影响程序行为的任何东西；
- 让程序在某个特定的条件下停止运行；
- 当程序停止后，检查程序背后所发生的问题；
- 修改程序，以便可以尝试修改错误之后的运行情况，并接下去进行调试。

学会了 GDB 之后，学员就可以对程序进行调试并改进自己的代码了。

GNU Building System—软件包构建系统

上面介绍的编辑器、编译器和调试器是黑客开发程序时每天都要使用的基本工具。而当开发的程序模块越来越多时，就需要将所有的程序模块打包起来，GNU Building System 就可以完成这个工作。初看起来，GNU 构建系统是一个庞大的软件，但是里面的模块是可以分解开来学习的。

第一个需要学习的工具是 GNU Make，第二个和第三个需要学习的工具是 Autoconf 和 Automake，第四个需要学习的工具是 M4。

在 GNU 构建系统中，M4 是 GNU 工程中用于宏处理的一个实用工具程序，可以由 M4 生成 `configure.in` 文件。`configure.in` 这个文件经过 `autoconf` 的处理就可以得到 `configure` 可执行文件；`acconfig.h` 文件经过 `autoheader` 处理即可得到 `config.h.in` 文件，这个文件与 `configure` 文件一起被 `configure` 运行处理后即得到 `config.h` 文件。`Makefile.am` 文件经过 `automake` 程序的处理可以得到 `Makefile.in` 文件，在运行 `configure` 命令时，这个文件生成 `Makefile` 文件。上面生成的 `config.h` 和 `Makefile` 文件经过 `make` 的处理，就可以得到最终的应用程序的可执行代码了。

这门课程最终以讲解如何生成 Debian GNU/Linux 形式的 `deb` 软件包和 Red Hat 形式的 `rpm` 软件包而结束。除了 GNU 工程提倡的 `.tar.gz` 发行形式之外，Debian GNU/Linux 发行版本的 `.deb` 和 Red Hat 的 `.rpm` 是目前自由软件社团使用最为流行和广泛使用的两种软件包发行形式。²⁷

CVS—共时版本管理系统

有关 CVS 的内容可以编写另一本与本书同样厚(甚至更厚)的著作来说明。的确，CVS 对于自由软件社团的重要性实在太大了。CVS 工作机制的内幕是“拷贝 → 修改 → 合并”，为了掌握 CVS 的用法，学员需要学习 `revision`(修订版本)，`repository`(源代码库)，`working copy`(工作拷贝)，`check out`(检验)，`commit`(提交)，`update`(更新)，`conflict`(冲突) 等概念和命令的使用方法。

CVS 背后的设计模型是针对网络的，因此如何配置基于网络的 CVS 是这一课程的另一个难点。一些高级的主题包括如何使用 Scheme 和 JavaScript 编写 CVS 的前端界面和后端的处理程序。

Bash—命令解释器

GNU 系统的用户对计算机输入的命令是通过 `bash` 这个命令解释器传递给操作系统内核的。`bash` 同时也是一门程序设计语言。可以利用它来创建各种脚本、设置环境变量、完成对输入和输出进行重定向等任务。当学生学完计算机程序的结构与解释之后，可以将 `bash` 提供的各种工具通过系统接口纳入到 Scheme 的解释环境中，这是创建强大的脚本的一种方法。熟悉 `bash` 后，学员在学习“网络数据处理与数据库”时编写 CGI 程序时会感到很轻松。

²⁷ 软件包发行的其他内容可以参见本书“万方乐奏有于阕”一章。

软件文档创作

计算机软件包括程序和文档两个部分，没有文档的软件是不完整的。遗憾的是，长期以来，计算科学的教学中存在着轻视乃至忽略软件文档重要性的问题，自由软件社团以往对于文档的重要性也不太重视，黑客道的教学思想则认为，对于一种长命的自由软件项目，开发软件文档的重要性要大于开发软件源代码。黑客必须重视软件文档的创作和维护。

本课程分为两个方面：软件文档的写作，软件文档的生成。

根据软件完成的任务，需要开发的文档类型可能是多种多样的²⁸。无论什么类型的文档，必须满足以下要求：

- 针对性强
- 文档内容准确、清晰、完整。
- 连续性好，具有可追溯性。

由于软件需求说明书和详细设计说明书是最重要的两个文档，因此黑客道培训课程中将编写这两种类型的文档作为重点讲授内容。

软件需求说明书是软件开发人员与用户的约定，好的软件需求说明书可以大大简化软件项目的实施。而对于开发人员，编写“开发日志”(ChangLog)是极为重要的，有了完整的开发日志，其他人就有可能从中得知原作者在开发程序时的思路，从而为反演提供前提。

文档最终是供用户、或者开发人员、或者管理人员阅读的，因此一旦编写好了软件文档的内容之后，就应该将文档按照某种格式打印或者显示出来。黑客道采用了 TEI 这种基于 SGML/XML 的规范来标记文档，并将这种文档转化为其他格式的文档。TeX 是一种重要的格式化工具，也是一种特殊用途的编译器，黑客道培训计划采用 TeX 作为软件文档创作的一个重要教学内容。Texinfo 可以与 TeX 结合起来讲解。²⁹

数据结构与算法

前面在介绍“计算机程序的结构与解释”时我们已经提到 C 语言编程是与数据结构和算法理论密切相关的，常见的数据结构：链表，堆栈与队列，集合，哈希表，树，堆与优先队列，图等。本课程的教学重点是让学员通过 C 语言理

²⁸ 国家标准(GB8567-1988)中列出了用户文档类，用户手册，管理人员文档，维护人员文档四大类，十四种类型的文档：用户手册，操作手册，维护修改建议；软件需求(规格)说明书，数据要求说明书，概要设计说明书，模块开发卷宗，详细设计说明书，可行性研究报告，项目开发计划。测试计划，测试报告，开发进度月报，开发总结报告等。

²⁹ 关于 TeX 的详细介绍，可参阅本书“排版、印刷和网络出版”一章。

解并掌握各种数据结构的具体实现方式，以及具有数据结构在内存中的存储模型的细节。

C 指针是教学的重点和难点。但是只有真正理解了指针变量的实质，学员才能真正理解数据结构的核心理念。³⁰

算法是解决一个问题所要采取的由一系列步骤构成的计算方法。黑客道培训计划将 Scheme 语言作为首选的编程语言的一个主要原因在于 Scheme 是一种优秀的算法语言，美国计算机科学先驱 Alan J. Perlis 说过，能有 100 个函数操作同一个数据结构要比只有 10 个函数却操作 10 种数据的数据结构好得多。Scheme 就是这样一种算法语言，Scheme 的数据结构从根本上讲只有列表(list) 这一种，程序和数据都采用相同的符号表达式表达，但是程序员可以定义各种函数对这一数据结构进行各种各样的操作处理。从这个意义上说，Scheme (或者说 Lisp) 是功能最强大的编程语言。

教学内容：查找与排序，数字方法，数据压缩，数据加密，图的算法，集合的算法。

在掌握数论的基础上，这门课程的第二部分要讨论算法的问题，在计算机科学中，一旦问题提出，解决问题的应该是建立问题的数学模型，所有的模型(问题)可以分成两大类：第一大类包括那些已经证明不存在任何算法的模型，例如，对于两个 Scheme 程序，它们是否等价？计算机科学家已经证明了不存在一个统一的算法来判定这样的等价性；第二大类包括所有那些已经证明确有算法的模型。在第二大类中，模型又可以分成三个小类：

- 已经证明只有指数型算法的那些模型，即困难问题或者困难模型。
- 已经找到了多项式算法的那些模型，即 P 类问题。
- 其余的模型的现有算法是指数型的，但是我们猜一个解时，却可以在多项式的时间内加以肯定，即 NP 类问题。

当今计算机科学界，包括国内不少有识之士现在都异常强调算法研究的重要性，有人建议国家成立算法库，来推动对算法的研究。的确，某个软件能否高效率地解决问题，其关键不仅仅在于编程技巧，而根本在于解决问题的思路与方法。

随着计算机硬件性能的大幅度提高，算法研究目前呈现出越来越兴旺的局面。无论在底层的操作系统和上层的应用软件领域，在上面提到的 P 类和 NP 类算法研究中，需要解决的实际问题层出不穷。

软件工业界常常出现几个人的小公司因发明出一种高效率的算法和相应具有竞争力的技术产品而淘汰掉市场上的大的竞争公司的先例(也有很多大公司收购这类小公司的案例)。信息技术发达的国家，从政府、公司，到大学，研究

³⁰ 笔者所著《Programming with C Pointers》可以作为这门课程的教材。

人员都高度重视计算机算法的研究。从某种意义上说，算法是计算机软件业的灵魂。一个创新的好算法甚至可以创造一个新的产业。

CISCO 公司生产销售的路由器上的操作系统是目前互联网上事实上的标准操作系统，具有很大的市场份额。它最初是由桑迪·勒纳和伦纳德·波萨克开发的。起初 CISCO 公司规模非常小，但后来被唐·瓦伦丁领导的红杉联合公司看中，于 1988 年以两亿美元的价格买断了他们二人在 CISCO 公司的原始股。

PGP 数字签名的软件就是 Phil Zimmermann 这位数学家开发的。它给后来的 GPG 设计小组带来了设计的启发。GPG 是自由软件社团最流行的加密解密和数字签名软件包。

据有关部门统计，2002年中国大陆有软件从业人员约 40 万人，但是分散在 5000 多家中小型公司中，大多数的软件公司的规模一般都不大，只有十几人，属于作坊式的开发规模。这类小公司在资金方面是难与大的公司竞争的，一条好的出路就是注重算法研究，因为算法研究主要还是依靠人来完成。在这一点上，大公司和小公司在资金和公司规模上的差别体现不明显，或者根本体现不出来，许多有名的算法和基于算法的程序都是一些个人的创造。只要认真分析市场需求，准确定位，集中精力踏踏实实搞开发，小型的软件公司和个人在算法创新上的空间和成功机会都是相当大的。³¹

编译原理与编译器构造

编译原理是目前计算机科学理论中发展得相对比较成熟和成型的分支学科，它是计算机科学教学中相当核心的内容，也是黑客必修的专业基础课程之一。尽管绝大多数学习计算机科学的学生可能自己不会去编写一个编译器，但是掌握了编译器的思想和方法对于程序员是有巨大好处的。掌握编译原理是可以让学员终生收益的，因为很多计算任务都会或多或少地用到编译原理。

遗憾的是，据笔者调查，绝大多数大学的计算机专业在讲授这门重要的课程时，没有结合某个编译器的源代码来讲解，因此学员对于编译原理的认识和理解难以达到实用的水平。

黑客道彻底摒弃了传统的教学方法，而是采用结合 GCC 这个伟大的自由软件来讲解编译原理，通过分析 GCC 的源代码，讲授编译器的构造。教学内容：扫描器(scanner)，解析器(parser)；语义分析程序(semantic analyzer)；源代码优化程序(source code optimizer)；代码生成器(code generator)；目标代码优化程序(target code optimizer)。

³¹ 关于未来软件公司与软件产业的形态，可以参考本书“软件工厂的奥秘”一章。

Flex 和 Bison 是自由软件基金会的两个杰出的工具软件，Flex 可以帮助程序员快速开发出词法扫描器，Bison 则可以帮助程序员快速编写出句法分析程序。黑客道讲授的 Scheme 语言符合是 R⁵RS 规范的，教学的实习项目就是按照 R⁵RS 规范，利用 Flex 和 Bison 这两个工具编写出 Scheme 的词法扫描器和句法分析器。

GCC 的代码生成器是别具一格的，它受到 Lisp 的启发，采用了 RTL 作为代码生成器的前端，因此理解了 Scheme 语言思想的学员，可以较快掌握 GCC 的代码生成模块。从 RTL 到最终的代码生成需要结合目标机器的指令集完成，这里面还涉及代码优化技术，黑客道结合“计算机体系结构和组成”来讲解这一部分的内容。

信号的分析与综合

总体上看，整套“黑客道”培训课程是针对黑客的培养制订的，黑客主要是指软件的开发人员，而不是计算机硬件开发人员，但是作为计算机软件开发人员，也应该对计算机硬件有相当深入的了解。

传统的教学内容包括“电子学与电路”、“数字逻辑电路”、“数字逻辑电路与硬件接口抽象”等内容。从今后的发展趋势来看，现有的教学内容和教学方法都必须进行彻底地调整才能适应技术的飞速发展。

模拟信号到数字信号的变换(A/D)，以及数字信号到模拟信号的变换(D/A)是这一门课程的教学重点，当各种物理信号被转化为数字后，就可以用数字计算机进行处理。反之，如果计算机系统中的数字可以转化为物理量，那么就可以将计算机广泛应用到工业控制，家用电器等众多领域的自动化控制中。

在实用的层面上，目前已经开发出来的 A/D 和 D/A 转换的微处理器和芯片种类非常多，因此程序员的主要任务将是针对这些微处理器的接口编写程序，完成对硬件电路的控制。除了使用 C 语言开发程序之外，学员应该努力掌握汇编语言。这是黑客道培训计划中关于编程语言学习的最后一个阶段，这一阶段直接涉及到硬件，因此学员在完成这门课程的学习之后可以真正理解计算机的工作原理。“信号的分析与综合”这门课就是围绕这一目的而开发的。

一个非常值得关注的发展趋势是，由于超大规模集成电路技术的发展，目前出现了一类设计硬件的软件人员，VHDL 和 Verilog HDL 技术就是这一趋势的突出代表³²。

由于我们的集成电路硬件生产工艺与世界先进水平有很大差距，短期内除非有非常大的投入，很难超越世界先进水平，但是在硬件的设计领域，我们则有

³² 参见本书的“二月春风似剪刀”一章，那里我们将专门深入讨论这个主题。

很大的后发优势，而且这一领域我们与发达国家的差距相差还不算太大，通过努力是可以在较短的时间内迎头赶上的。黑客道培训计划中增加了对于这两种先进的硬件描述语言的介绍内容。

这门课程这一课程含有三个方面的教学内容：

- 信号变换的数学原理
- 汇编语言
- 硬件描述语言入门

信号变换的数学原理课程需要学员现掌握数论，数学分析等基础，教学内容包括：从元件到方程；拉普拉斯变换，模拟系统，微分方程； z 变换，数字系统，递归方程；卷积；系统函数，频率响应，数字模拟；模拟与数字综合；傅立叶级数；傅立叶积分。学员可以结合黑客道课程中的“泛系导论和数学基础”来学习，以补充提高数学技能。

与二十世纪八十年代以前的程序员相比，今天的程序员的生产率已经大大提高了，即使在原来许多必须采用汇编编程的场合下，**C** 也完全取代了汇编代码。今天采用汇编语言编写程序的理由似乎只剩下以下三条：

一、为了兴趣。很多人喜欢与硬件直接打交道，而且控制处理器这个层次上的工作的确非常容易使人上瘾，而使用高级语言则难以体会到这样的乐趣，

二、为了性能。汇编语言的指令与机器指令是一对一的，与此相比，一条高级语言的语句则往往对应着一堆的机器指令，在一些情况下，程序员为了提高程序的运行速度，往往借助于汇编代码来改写某些高级语言代码运行起来效率不高的部分模块，以优化程序性能。但是这样的情形因为硬件性能的大幅度提高而日趋少见了。

三、为了交流。即使你不开发汇编语言，但是与别人交流时，也要能理解别人开发的汇编程序，许多场合下，使用汇编代码是显示处理器如何工作的唯一办法。而且这些场合下，往往没有高级语言的支持(例如，因为没有可用的 **C** 编译器)才使用汇编代码。

这一课程的教学难点有两个：一是如何将各类物理量的变化转化为数字信号(以及其逆过程)，二是在于汇编代码与高级语言(例如 **C**)之间的参数传递，以及如何理解和按照程序控制的转移。³³ 除了编程方面的技巧和电子学等方面的一些基础知识之外，本课程特别注重动手实践，学员在学习中应该学会使用电烙铁、万用电表、示波器等基本工具。

³³ 笔者曾经编写过《GAS 汇编程序设计》一书(*Programming with GNU Assembler*)，可供学员选择作为教材。

操作系统原理与构造

按照传统操作系统理论，这一课程需要讲授进程管理、文件系统、内存管理、设备管理、任务管理、网络管理等内容，GNU/Linux 提供了系统的所有源代码，因此我们现在有可能结合操作系统的源代码来深入讲解这一课程的。

学习这门语言之前，应该先讲授“数据结构与算法”，学员应该熟练掌握了 C 语言和 GNU 基本编程工具。

C 语言的库函数在操作系统的内核实现中占有中重要的地位，内核是利用各种 C 库函数结合算法编写的一整套策略，这些策略处理以下的问题：进程管理、文件系统、存储管理、设备管理、任务管理、网络管理。这些策略最终都可以通过算法和数据结构来理解。

GNU 操作系统里具有三千多个软件包，黑客道培训课程从中提取了约 300 个软件包，要求学生掌握从这 300 个软件包的源代码开始，一步一步地在“裸”的个人计算机硬件平台上拼装出一个最小、完整并实用的 GNU 操作系统出来。

这就如同军队里士兵们技术比武时，先将机枪拆散成零部件，然后让士兵在规定的时间内还原回来一样。如果士兵可以将这些拆散的零部件重新拼装成原样，则表明士兵对于机枪的组成原理有了清楚的认识。

同样的道理，一旦学生熟练掌握了 GNU 操作系统的拼装方法，那么就为学生将来定制系统打下了良好的基础。因为这时学生对整个系统的理解可以达到“庖丁解牛、游刃有余”的程度。

从头拼装一个完整的操作系统的难点在于解耦软件包之间，程序与程序之间，模块与模块之间的环形依赖关系，例如在编译 Linux 内核之前需要有 GCC 编译器，而安装 GCC 编译器又需要实现安装 GNU Make，安装 GNU Make 又需要先安装 GCC 编译器，因此需要妥善解决“鸡与蛋孰先？”的问题。而这正好需要学员对于操作系统中的软件包具有相当深入的了解才行。

数据通信与网络协议

通信协议有三大类：电路交换类的协议、存储转发类的协议，以及基于分组交换类的协议。互联网时代的通信协议将越来越依赖于基于分组交换协议。下一代的通信网络的技术基础是倾向于以 IP 网络技术为核心的。

“数据通信与网络协议”讲授分组交换技术的基本原理，IP 网络通信技术的内幕³⁴，并选择了两个非常重要的网络应用作为重点讲授内容：万维网和电子邮件。

³⁴ 教员可视学员的基础讲解互联网的域名机制。

HTTP 协议是 WWW 的支撑基础，黑客道将讲解 HTTP 协议是如何在 GNU/Linux 操作系统上使用 socket 机制来实现的，重点讲解 Linux 内核和 GNU C Library 中的相关模块，以及 Apache 和 Mozilla 是如何与这些模块交互工作的。

Apache 和 Mozilla 是目前功能最全面的 Web 服务器和浏览器，黑客道培训计划中选择它们通过其源代码来讲解 Web 服务器和浏览器的工作原理(学员应复习“数据结构与算法”中的基本功)。

Internet Email 是另外一种流行的互联网应用，尽管它倍受垃圾邮件的困扰，但是却仍然是目前最常用的互联网应用之一。电子邮件系统是相当复杂的，它包括 MTA (邮件传输代理)、MUA (邮件用户代理)、MDA (邮件投递代理)和 MRA (邮件收取代理)四个部分。黑客道通过几个具体的应用程序来说这四个模块的内幕，MTA (邮件传输代理)是通过 sendmail (或者 exim) 来讲授的，MUA (邮件用户代理)选择了 Evolution (或者 xfmil)，MDA (邮件投递代理)则讲授 sendmail 如何与 UUCP 相配合，MRA (邮件收取代理)则讲授 POP3 和 IMAP 这两个程序。

网络数据处理与数据库

按照传统的分类方法，数据的组织形式除了无结构的纯文本之外，可以分为结构化的关系性数据，以及面向对象的不可解析二进制数据(数据结构只能被给定的应用程序解析)三大类。结构化的关系性数据广泛地运用在关系性数据库管理系统(RDBMS)的数据存储中，并且在此基础上发展起来了具有工业力度的数据库查询语言(SQL 规范)和相关工具，互联网在某种程度上是世界上最大的数据库，万维网是互联网的窗口系统，而万维网上的数据是半结构化的，意味着万维网上的数据是传统意义上的三种数据的组成的某种复合体，并加上了对数据进行说明的元数据，因此万维网上的数据是半结构化的数据。

通过引入元数据，传统的无数据结构纯文本数据获得了数据结构，而文本是几乎所有计算机上都支持的数据。

开设这门课程是基于这样的考虑：计算机工业发展的重点已经经历了从硬件(Hardware)到软件(Software)，又从软件到“讯件”(Infoware)的转移。在软件行业占据计算机工业重心的时代，管理信息和数据库需要专业的数据库管理员，他们以前在使用这些数据库系统时需要掌握许多晦涩的计算机搜索工具及其语法和命令，而互联网时代的网络应用程序只需要一个 Web 浏览器就可以方便直观的“指一点”式操作就可以替代原来那些复杂的操作，好的网络应用程序开始向人与人之间的交流方式靠拢。讯件与软件的关键差别就在于讯件是以用户为中心来考虑程序的设计的，用户的输入就是程序。例如，在线购买图书时，用户现选择自己所需要的图书，作出一个或者多个选择，输

入少量的数据(例如购买数量,支付方式等),然后按发送按钮,服务器得到用户的输入之后,就动态地生成一个定单报告反馈给用户。

讯件的用户界面常常是动态的,而非像传统的软件应用程序那样是一成不变的,例如某个图书经销商制订的畅销图书及时排名榜就是根据网上读者的反馈数据动态综合后直接完成的,而不是经销商根据自己的每个月的库存报告发布的。

为了完成这些计算任务,在网络的客户端和服务端都需要有讯件来支持,客户端的讯件目前是 JavaScript 这样的 Web 浏览器中嵌入的解释器来支持的,而服务器端的讯件则是通过 CGI 程序来支持的。黑客道中讲授的 Scheme 语言就非常适合编写 CGI 程序, Scheme 语言中可以执行的程序和数据是按照相同的形式表达的,因此它非常适合编写针对网络应用的讯件程序。

黑客道培训课程将讲授如何在 Apache 这样的 Web 服务器上安装由 Scheme 编写的 CGI 程序,这些 CGI 程序又如何生成动态的讯件应用。Scheme 编写的 CGI 脚本又如何与 PostgreSQL 这样成熟的具有工业力度的数据库引擎交互等。

XML 是目前对万维网上数据进行定义的规则,相关的教学重点是介绍 XML 规则的内容(以 XHTML 和 TEI P5为教学内容),XML 数据的转换方法。转换 XML 数据将采用与一些与编译原理中类似的技术,因此在学习这门课程之前,应该先讲授“编译原理和编译器构造”。

黑客道培训计划讲授如何使用 Scheme 语言来解析 XML 文档。XML 可以视为是 Lisp 的符号表达式的一种变形,黑客道学员在掌握了 Scheme 语言之后,可以立即掌握 XML 的核心思想——这也是黑客道培训课程为什么从 Scheme 语言开始讲授编程语言的原因之一。

使用 Scheme 语言编写 CGI 程序,将 Scheme 的 CGI 程序代码存放在 Apache 服务器上,因此这门课程与“数据通信与网络协议”相互联系,可以结合起来讲授。Web 浏览器前端的代码可以使用 JavaScript 和 XHTML 编写³⁵。

在自由软件社团中,PostgreSQL 是业已成熟的具有工业力度的关系型数据库管理系统,一些商业大型数据库管理系统上的许多概念发端于这一系统。PostgreSQL 数据库是一种典型的“面向对象”的关系型数据库管理系统,黑客道培训计划选择它作为教学工具是基于下面的考虑:

³⁵ JavaScript 与 ECMA-262 规范有直接关系,未来在 MING/OS 系统中,将由 Scheme 来模拟基于 ECMA-262 的 JavaScript 编写 Web 应用程序的前端。

- 它是自由软件
- 它在传统的 RDBMS 上引入了对象的概念³⁶
- 它提供了对事务交易的完整支持³⁷
- 它支持客户机 / 服务器计算模型
- 它支持 SQL 标准，并且具有多种编程语言的接口。
- 它是 32 位的数据库管理系统，支持大对象的数据库管理。
- 它支持中文，而且系统文档齐全。

因为 PostgreSQL 的技术特点这么丰富，因此它是网络应用，特别是基于 WWW 的网络程序中的数据库引擎的良好选择。

学习完这门课程之后，学生应该会设计 Web 网页、管理网络数据、设计符合电子商务要求的企业级的应用程序等技能。IT 技术与传统产业相结合是下一轮计算机科学技术发展的重要推动力之一，这类人才是大有可为的。黑客道的教学实践表明，学员如果熟练掌握了这些自由软件的使用技巧和编程技术，就可以在企业市场上谋求得在一份好的工作。

黑客道培训的商业模式

正如前面提到的，“一、百、万”工程中研究项目多，工作量巨大，因此它必须设法解决资金来源的问题，不能让资金等方面的问题构成阻止我们向社会传播自由理念的障碍，而实施黑客道培训是“一、百、万”工程一个重要的资金来源。黑客道培训是“一、百、万”工程的一个重要组成部分。某种程度上讲，当黑客道培训计划完成之时，也就是“一、百、万”工程开始获得成功之日。

一个问题是，黑客道的教学思想是公开的(读者已经从这篇文章中看到了其大致轮廓)，为黑客道培训计划开发的教材也是自由出版物，自由出版物也是可以自由再拷贝、自由再发行的，如何通过培训项目得到资金的来源呢？

在自由软件基金会和几位专业律师的帮助下，黑客道建立了具有自己特色的商业模式。我们先确定了收入产生的基础。回到伟大的德国哲学家康德的见解：权利要么是强制相结合的，要么是资历相结合的。由于黑客道培训计划的开发者在这一特定领域的有独特的资历，因此可以获得相应的权利。

这种权利体现在可以通过“传道、授业、解惑”来收取一定的学费上。古时的学生要向先生交“束修”，就是现代所说的学费。另一个明显的收入来源是出版并销售“一、百、万”工程中为“百”字工程创作的教材、辅导材料、文档等教学材料的拷贝。有了黑客道教学思想的指导原则，我们就可以像当年法国

³⁶ 提供了对数据类型和操作的良好扩展性。

³⁷ 可以开发具有工业力度的企业级应用程序(例如 ERP 之类的应用)。

的布尔巴基学派利用结构主义观点改写数学理论教材一样³⁸，我们可以站在新的高度来重新认识计算科学理论，并改写和创作成完整而且成系列的计算科学的教材。

更重要的权利体现在（如前面提到的），在探索泛系黑客道教学法的过程中，笔者发现了教育学中的不动泛系原理，结合不动点算法，创立了度量不动泛系中的心理认知距离的一套方法，这套度量方法构成了一种“技术诀窍”（Know-how）。虽然黑客道的教学思想、教学方法和教学内容都是自由公开的，这套技术诀窍不是一种秘方，但是他人若想理解和掌握这种教学方法却需要我们付出技术支持的时间和其他付出，我们可以围绕推行这套教学方法建立服务业务。从这个角度看，**黑客道是黑客生产线的软件系统！**

这种黑客生产线可以在许多条件成熟的地方建立起来，而围绕黑客道教学思想，教学方法和教学内容，我们可以按照现代商业社会的商业运营模式（特许、连锁、直销、专卖等）向教育市场提供系列化的技术支持。

商标

黑客道培训商业模式的核心是：为了保持黑客道培训计划的技术特色和独特个性，推动和促进“一、百、万”工程的发展，必须制订和彻底地实施品牌战略。黑客道品牌的核心价值构建在其独特的教学思想，个性鲜明的教学方法和完整的教学内容之上，它是制造黑客生产线软件系统的依据。我们认为，只有坚持不懈地实施品牌战略，才能吸引相当多的学员来参加自由软件运动，带来相当多的资源完成“一、百、万”工程，达到多个方面多赢的结果。

为此，我们注册了黑客道培训的服务商标，黑客道服务商标的注册为实施品牌战略提供了法律保障。

黑客道培训计划从教学思想到教学内容都是自由而开放的，谁学谁用谁再创。从理论上说，任何其他人都是可能自行地摸索出与黑客道类似的或者更好的教学方法的，而且我们也鼓励这种努力。但是这一探索的时间成本和难度是相当高的，如果没有相当的时间和智力上的保证，则购买黑客道技术服务的“投入 / 产出比”会低得多，来得更加直接和高效。因此，我们既期待这一领域的竞争者，又欢迎这一领域的合作者。

³⁸ 第二次世界大战之后在法国崛起的布尔巴基学派就是从结构主义观点出发，以代数结构、序结构和拓扑结构三种基本结构为基础来重新认识数学，重构整个数学理论群落的。布尔巴基学派的笔杆子，也是其代表作家，法国科学院院士刁东雷（Diodonné）就从结构主义观点出发编写了大量的数学著作。

古代伟大教育优秀传统的复兴

细心的读者可能已经注意到本文多次引用了关于孔子的言论。的确，黑客道是有意在某些方面遵循着孔子的教学思想。

祖国的教育传统是圣人孔子开创的，作为我国古代伟大的思想家、教育家，他整理、编纂、总结了《诗》、《书》、《易》、《礼》、《乐》、《春秋》等前人的文化遗产，并传授给学生，“有教无类”，历史上首创了私人讲学的风气，打破了贵族垄断教育的局面，培养造就了大批人才，受业身通者七十有七人³⁹，皆异能之士也，得以弟子三千，成为儒家学派的创始人。在汉朝时，儒家思想对社会的影响就已经遍布到各个方面，孔子布衣，传十余世，学者宗之，自天子王侯，中国言六艺者折中于夫子，可谓至圣矣。

孔子的教学内容绝大多数是学以致用用的，其六艺中除了“礼”以外的“乐、射、御、数、术”都是那个时代的实用技术。他开办了中国的第一所民办音乐学校、射击学校、驾驶学校、计算学校、谋略学校，他“述而不作”，其教学内容既是分科性的，又是综合性的，其教学与现行的应试教育形成了鲜明的对比。

孔子的教育思想中存在着非常丰富的内涵，这是祖先给我们留下的一笔弥足珍贵的文化遗产。他重视学生的道德训练，提倡学生服务社会。他的教育思想以人性出发，因材施教，充分照顾学生的性格和能力特点，不搞一刀切。他鼓励学习时举一反三，注重激发人的潜能。这些丰富的教学思想与方法在那么早的时代就如此完备成熟，不能不说是中华文明的一大奇迹。假令孔子而在，余虽为之执鞭，所忻慕焉。

孔子死后中国历史上出现了许多假孔子，他们打着孔子的旗号，但是言行则偏离了孔子原来的思想。这一点在教育领域尤其显著，他的学说逐渐被封建统治者利用改造成了禁锢人思想的工具。因此去伪存真、还孔子学说之真实原貌就成了后世思想与文化变革的一大内容。

近代教育制度是从清朝末年和民国初期从西方引进并逐步建立起来的，这一变革于晚清伴随着西方列强入侵的枪炮声开始，“五四运动”则是这一西学东渐的重要成功标志。它倡导“科学”与“民主”一大成就即是引入了当时属先进的西方教育体制，并把真孔子和假孔子一并全部打倒了。

又经历了约半个世纪的时间跨度后，中国从原苏联照搬了一套教育体制，并沿用下来，至今仍然没有完全摆脱其窠臼。实践表明，就计算科学的教学而言，固守这一“重书本、轻实践”的教育体制是一条危险的死胡同，因为它扼杀了人的个性和创造才能，这是与时代发展潮流对人的要求相悖的。

³⁹ 一说为七十二人。

现代教育制度是着眼于像制造工业产品一样来培养人才，这种类似工业品生产线培养出来的人几乎都是没有个性的。⁴⁰ 爱因斯坦对现代教育制度曾经提出过尖锐的批评：“现代教育制度如果没有将人的好奇心扼杀殆尽，堪称奇迹。”

实际上，目前教育界已经很清醒地感到了社会变革对教学带来的压力，很多人也在一直呼吁教育革命，然而他们之中一些人大谈特谈“教育革命”多年后，却于实际不见大的成效，这恐怕是与忽视或者缺乏现代科技发展的“三兼顾”内在规律有一定关系。依笔者看来，复兴以孔子开创的中国古代优秀教育传统、充实之以现代科技成就的内涵，是一条看得见、摸得着的可实践的路线。废除现行教育体制的弊端、代之以真孔子的优秀教学传统则应该是当代教育界肩上的重大责任。

应该清醒地看到，实施这种复古路线或许比实施所谓的“革命路线”的阻力要小些，但是在实际运作中的难度仍然是相当大的，现代复古者仍然应以孔子为榜样。孔子是那个时代的一位复古者，从司马迁所著《史记·孔子世家第十七》和其他历史记载来看，孔子一生有着极高的政治理想，即使在遭到打击、排挤、嘲讽、围困时仍热情不减。为了实现自己的理想，他的一生中有相当多的时间是与众弟子一道周游列国，到处游说而度过的，尽管到处碰壁，但是仍执着追求，明知不可为而为之。这种精神正是我们这一代复古者学习的楷模。

人类进入了二十一世纪网络信息时代，现代社会的物质条件较两千多年前孔子所处的时代更不知要进步了多少倍，孔子在那个时代在那样的条件下能够培养出三千多名弟子，今天我们有互联网、无线网络、宽带、电视、飞机、火车、多媒体、投影仪等等物质基础的支持，我们这些复古者应该完全有信心比他做得更好。

教育需要变革、教育亟待变革是一个世界性的潮流和挑战，世界各国、特别是西方发达国家都在积极努力地探索新时代下的新教育模式。过去我们虽然在前几轮计算机技术的发展中落后了许多，但是我们应该清醒地看到我们具有巨大的后发优势。在教育领域，这些后发优势如果能和我们的古代优秀的教育传统结合起来，那么我们一定可以为世界教育变革提供一种先进的模式，从而为人类在新千禧年的进步作出我们的贡献。

版权所有 © 2003 洪峰

2003年1月20日，初稿。

2003年6月30日，第一次修改；

⁴⁰ 说得严重一些，应试教育体制下培养的学生特点就是会应付考试。

2003年7月09日，第二次修改；
2003年7月14日，第三次修改；
2003年7月16日，第四次修改；
2003年7月20日，第五次修改；
2003年8月14日，第一版定稿。
2003年8月22日，第1.1版。
2003年11月5日，第1.2版。

依照 GNU Copyleft 许可证的条款，本文可原封不动地拷贝和自由地再发行，
但在拷贝和再发行的拷贝中必须原封不动地含有这段版权声明。
