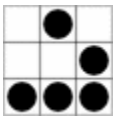
如何成为黑客

作者：Eric Steven Raymond　译者:刘海粟

（此文部分内容参考Ken Lee前辈的译文）



为何有此文章？

身为《黑客辞典》[1]和其他一些知名文档的作者，我常收到热心的网络新人的电子邮件，问及“如何可以成为一名神奇的黑客？”。 早在1996年我便注意到，好像还没有任何一个“常见问题”或者网络文档论述过这个重要的问题，于是我决定撰写此文。我相信很多黑客都在考虑我现在考虑的这些问题。然而，我并不认为我在这个问题上是唯一的权威，如果你不认同下面读到的这些内容，那请你写下你的看法。

如果你正在阅读离线文档，那么你可以在这里找到本文的最新版。

本文的装饰物——这个五点九宫图被称为“滑翔机”。在一款令无数黑客痴迷的名为“生命游戏”[2]的数学仿真游戏中，这个简单的图案蕴含着令人吃惊的特性。我认为它是一个能够彰显黑客本质的图形徽章——抽象，起初在表面上有些神秘，但自身拥有一个具有错综复的逻辑性的入口通向整个世界。在这里可以了解更多关于滑翔机徽章的信息。

什么是黑客

在《黑客辞典》里有不少关于“黑客”的定义，大多和“精于技术”或“乐于解决问题并超越极限”之类的形容相关。然而，若你想知道如何成为一名黑客，只要牢记两点即可。

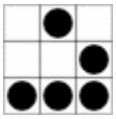
这是一个社区和一种共享文化，可追溯到那群数十年前使用最初的共享小型机以及最早的“阿帕网”[3]的专业程序员和网络专家们。而这样一群人便被称之为“黑客”。黑客们建立了互联网，黑客们让Unix操作系统成为了今天这个样子，黑客们管理着讨论组[4]，黑客们维持着万维网的运转。如果你是这种文化的一部分，你为这些做出过你的贡献并让其他的人知道你且以“黑客”之名称呼你，那你就是一名黑客！

然而，黑客精神却并未被禁锢在“软件黑客”这一范畴之中。有很多人将黑客精神应用到了其他的领域，例如电子或是音乐——事实上，你可以在所有的科学与艺术的最高境界中发现黑客精神的影子。由于这种共同的精神，软件黑客亦称后者为“黑客”——甚至有人认为黑客精神是脱离于黑客这项工作而独立存在的。但在以下的文章中，我们将重点讨论软件黑客的技能与态度，以及能令他们被称之为黑客的“共享文化”这一传统。

而另外一群人总是声称自己是黑客，但他们不是。这些人(大多是年轻人)入侵其他人的计算机或是破坏通信系统。真正的黑客称这群人为“骇客”[5]并与他们划清界线。真正的黑客大多认为骇客是懒惰、不负责任以及不明智的一群人。能够破坏保安系统并不能让你成为一名黑客，正像能够偷车并不能让你成为一名汽车工程师一样。不幸的是，很多记者与作家都被误导而使用“黑客”一词去形容“骇客”，而这将永远激怒那些真正的黑客。

最根本的区别在于：黑客是创造事物，而骇客是破坏事物。

若你想成为黑客，请看下去。若你只想成为骇客，请到 alt.2600 讨论小组并做好准备当你发现你远没有自己想象的那样明智之后，去坐五到十年的牢。这就是我要对骇客说的。



黑客的态度

黑客解决问题并创造事物，他们信仰自由与互助。想要以黑客的身份为大家所接受，你需要表现出你拥有这种态度，而若要表现出你拥有这种态度，则需要你真的信仰这种态度。

而如果你认为培养黑客态度仅仅是融入黑客文化的一种方式，那你就错了。真正的去信仰这些对你来说非常重要——它可以帮助你学习并保持你的动力。像所有的艺术创作一样，成为大师最有效的方法是遵循大师的思维——不仅是智慧，还有情绪！

以下这首现代禅谒便说明了问题：

遵循此法：
仰慕大师，
跟随大师，
比肩大师，
超越大师，
成为大师。

所以，要成为黑客，请反复阅读以下几点直到你牢记并坚信它们为止。

1．这世上有无数奇妙的问题等待我们去解决。

成为黑客是一件非常有趣的事情，但乐趣的背后需要付出大量的努力。努力需要动力。一名出色的运动员的动力往往来自于自身肢体运动与超越自身极限的一种天然成就感。正如一名黑客应该从解决问题、磨练技能以及挥洒智慧之中获取喜悦一样。

如果你不是这种人，那为了成为一名黑客，你必须让你自己成为这种人。否则你会发现你的黑客热情将被诸如性、金钱以及社会价值观等事物慢慢的消磨干净。

(你亦需要培养一种对于自学能力的信仰——坚信即便你现在不能解决全部的问题，但如果你可以解决哪怕一小部分的问题并从中学习，那么你都可以学习到足够的东西去解决下一小部分的问题——直到将整个问题全部解决为止。)

2.没有任何一个问题应该被解决两次

创造力是一个宝贵而又有限的资源。世上有如此多的新奇问题等待我们去解决，创造力不该被浪费在“再发明”这样的怪圈中。

想要成为一名黑客，就要去相信其他黑客的思考时间都是很宝贵的——以致于分享信息对于你来说是一个道义层面上的责任。解决问题并给出解决方案，好让其他的黑客可以去解决新的问题而不是永远在旧问题上止步不前。

然而，请注意：“没有任何一个问题应该被解决两次”并非是要你认为现有的解决方案

是神圣不可侵犯的或者它是当前问题的唯一解决方案。通常，我们遇到一个前所未见的问题时，总是从该问题最初的那个解决方案中学会很多东西。这很好，而且通常很必要，这意味着我们可以做的更好。而不好的则是人工技术上、法律上或是体制上的阻碍(譬如闭源代码)，它阻止人们使用一个现有的解决方案，而迫使人们陷入“再发明”的怪圈之中。

(你不必认为你有责任公布出你全部的作品——虽然这样做的黑客会从其他黑客那里获得最多的尊重。用它们赚取足够的钱来维持你的食物、租金和计算机——这并不违背黑客的价值观。利用你的黑客技术来维持一个家庭甚至致富是一件好事——只要你在做这些的时候没有忘记你对于黑客艺术以及黑客朋友的忠诚。)

3．无聊与繁冗的工作等同于邪恶

黑客——或是其他一些创作者——不应在那些无聊的或是愚蠢的重复工作中充当苦力，因为如果这样那就意味着他们没有去做一项只有他们才可以做的事情——解决新问题。这种蠢事对谁都没有好处。因此，无聊与繁冗的事情对黑客而言不仅只是不良的，甚至是是邪恶的。

要成为一名黑客，你必须信仰此观念并将所有无聊与繁冗的事情尽可能的自动化，这不止为自己，也是为其他人(尤其是其他的黑客)。

(这里有一个显而易见的例外。黑客们有时会做一些看起来重复或无聊的事情用以令思维清晰，或是为了训练某些特殊的技能与经验并且别无他法。但这完全取决于个人的选择——没人能强迫一个善于思考的人去完成这些事情。)

4．自由很好

黑客是天生反独裁的。若有人可以对你下达命令，那他也可以阻止你去解决那些令你着迷的问题——并且基于这种独裁的思维模式，他总能找到一些愚蠢到令人吃惊的理由去这么做。所以无论何时何地都要与独裁主义战斗，这样才不至于让你或是其他黑客被他扼杀。

(这并不等同于向所有权威挑战。儿童们需要被教导而罪犯需要被压制。同样，黑客需要接受某些权威统治，前提是从中的获益应该多于去执行统治者的命令。但这一切都该是有限度的，并且是一个平等的契约关系。而不该是一个强加于个人意愿之上的独裁思想。)

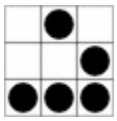
独裁依靠的是“审查”与“保密”。他们总是怀疑“自愿合作”和“信息共享”——他们只喜欢在他们控制下的“合作”。所以要成为一名黑客，你必须练就一种对审查、保密或使用暴力和欺骗手段去压迫人民的行为的本能敌对情绪。并且你该因这种信仰而采取行动。

5．态度不能代替能力

作为一名黑客，你当拥有以上的态度。但仅有一个正确的态度不能让你成为黑客，最多能让你成为一名优秀的运动员或是摇滚明星。想成为一名黑客还需要智力、实践、奉献以及勤奋。

因此，你该时刻持有怀疑的态度并尊重每一种能力。黑客们不会让一个过难的问题去浪费他们的时间，但他们崇拜能力——特别是黑客技能，但其实任何的能力都是有价值的。很少有人能掌握的能力尤为可贵，而这其中若含有高度的智慧、技巧与集中力就再好不过了。

若你尊重能力，你便会非常乐于锻炼自身的能力——那么勤劳工作与无私奉献就会成为一种紧张的竞赛而非一件苦差事。这种态度对于成为一名黑客也是至关重要的。



基本的黑客技能

黑客的态度固然重要，但技能却更为重要。态度无法替代能力，有一些必备技能和基本工具是任何一个梦想成为黑客一员的人都应该掌握的。

随着时间的流逝，这些工具也在慢慢的吐故纳新。例如，过去需要用机器码编写的一些程序，现在却更多的用 HTML 去编写。但很显然，你需要掌握以下这些：

1.学习如何编程

这当然是最根本的黑客技能。如果你不熟悉任何计算机语言，我建议从 Python 开始。它设计清晰、文档详实，对于初学者来说非常友好。尽管它是一种非常好的初学语言，但它绝非玩具。他非常强大与灵活，并且适于开发大型项目。我已经写了一份更加详细的“Python 评价”，更多的教程可以在 Python 网站上获取到。

我过去也常推荐 Java 作为早期学习的一种语言。但一些批评[6]改变了我的想法。黑客不能像他们说的那样“像一个五金店里的管子工一样的去解决问题”，而是应该知道每一个部件是如何运作的。现在我认为最好是先学习 C 和 Lisp，而后再是 Java。

此处可能有一个很普遍的问题。如果一种编程语言令你吃不消，那它可能是优秀的编程工具却不易学习。并非只有编程语言才有这种问题，像 RubyOnRails、CakePHP、Django 这样的页面应用平台会很容易有一种肤浅的认识，但当你遇到一个难题时却发现无力应对，即便是去纠正一个很简单的错误。

如果你对程序的要求很严格，那你必须要去学习 C——Unix 的核心语言。而 C++和 C 非常的相近，如果你学会了其中的一种，那学习另外一种将不会有什么困难。然而，这两者都不是初学者的好选择。并且实际上，越是能够避免用 C 编程，你的效率就越高。

C 语言非常高效，也很节约计算机资源。不幸的是，C 是通过要求你进行大量低级的手动资源(如内存)管理来实现这种“高效”的。所有这些低级代码都是复杂和易错的，并会占用你大量的时间去调试。在机器越来越强大的今天，这可不是个明智的权衡——一个明智的选择应该是用编程语言让机器变得不那么高效，而让你的时间变得越来越高效。因此，选择 Python 吧。

而其他对于黑客具有重要意义的语言包括 Perl 和 LISP。Perl 值得学习的主要原因是：它被广泛的应用于动态网页和系统管理中，所以即便你不会写 Perl 程序你也应该学会如何阅读 Perl 代码。很多人选择 Perl 的原因与我建议 Python 的原因相同，既避免使用 C 编程而使不需要 C 的机器更高效。你至少应该能理解它们的代码。

LISP 值得学习的原因则不同——在你掌握它之后你会收到一些意想不到的经验和启示。这些经验会让你在今后成为一个更加出色的编程者，即便你并不经常使用 LISP。(你可以很容易的使用 Emacs 文本编辑器或是 GIMP 的 Script-Fu 插件来编写 LISP)

最好能学习全部这五种语言——Python、C/C++、Java、Perl、LISP.此外，作为最重要的黑客语言，它们代表了完全不同的编程方式，并且每一种都有它们自身的价值。

但你该知道若只是简单的堆砌这些语言，那你根本无法达到黑客的水平而仅仅是一个程序员而已——你应该去学习如何去用一种独立于任何一种编程语言之外的思维模式去规划你

的程序。作为一名真正的黑客，你该达到可以在数日之内学会一种新的编程语言的能力，且凡是手册中做出过说明的就应该是你知道的。这意味着你得学习一些非常不同的语言。

我不可能在这里完整的介绍如何学习编程——这是一个很复杂的技能。但我可以告诉你书籍与课程并不有效。许多，或者是大部分的优秀黑客都是自学的。你仅能从书中学到语言的基本特征——就这点东西。而将知识融为自然的思维模式只能从实践与模仿中学到。而能做到这样的只有(1)阅读代码和(2)编写代码。

Peter Norvig——Google 公司的顶尖黑客之一以及被广泛应用的关于人工智能的教科书作者之一——撰写了一篇非常优秀的文章《教自己在十年内学会编程》，他“编程成功的秘诀”是非常值得关注的。

学习编程就像学习去写好一门自然语言一样。最好的方法就是去阅读一些大师写好的东西为规范，然后自己写一些东西，再多读些，再多写点，再多读些，再多写点……反反复复直到你的作品看起来像模像样了。

寻找优秀的代码来阅读曾经是件很不容易的事情，因为很少有大型程序的代码供新手黑客阅读和思考。而现在则发生了巨大的改变：开源的软件、编程工具和操作系统（全部由黑客创造）现在已经被广泛的使用。而这恰巧让我引入了我们的下一个主题……

2.获取一个开源的 Unix 系统[7]并学着去使用和运行它

我假设你拥有一台个人计算机或至少是可以使用一台(花些时间来欣赏一下这是多么的弥足珍贵。黑客文化可以追溯到那个计算机昂贵到个人无力承受的时代)。对于每个想收获黑客技能的新人来说，第一步也是最重要的一步就是获取一份 Linux 或是 BSD-Unix 亦或是 OpenSolaris 的拷贝，之后在你的机器上安装并运行它。

的确，除了 Unix 系统之外这世上还有很多的操作系统。但他们都是以二进制形式发布的——你无法阅读其中的代码，也无法修改他。想要在装有 Windows 系统或是其他什么闭源系统的机器上学习黑客技能就像是穿着盔甲学跳舞一样。

在 Mac OS X 上学习还是可行的，但仅有一部分系统是开源的——你会四处碰壁，并要时刻注意不要养成依赖苹果私有代码的坏习惯。如果你将精力集中到 Unix 的引擎中，那你将学会不少有用的东西。

Unix 是互联网的操作系统。虽然你不了解 Unix 也可以学习如何使用互联网，但你终究无法成为一名互联网黑客。正因如此，当今的黑客文化拥有一个强大的 Unix 核心。(这并不总是真的，有些早期的黑客为此不悦。但 Unix 与互联网的联系已足够强大以致微软的魔爪都不足以削弱这种联系。)

所以，选择一款称心的 Unix 吧——我本人很中意 Linux，但也有别的选择(当然，你也可以同时在一台机器上同时运行 Linux 和 Windows)。学习它，运行它，用它来思考，用它与互联网对话，阅读代码，修改代码。你会获得那些在微软的操作系统上做梦也无法得到的优秀编程工具(包括 C、LISP、Python 和 Perl)，并将从中获得很多的快乐。当你成为一名黑客高手的时候再回头来看，你会发现你获取到了比你想象的更多的知识。

关于 Unix 的更多知识，可以参见“Loginataka”。你或许也想看看《Unix 编程艺术》。

若你想着手于 Linux，那可以看看“Linux 在线”网站。你可以从那里下载或者(这样可能更好)寻找一个本地的 Linux 用户社群来指导你如何安装。

在这份指导诞生的头十年里，我始终认为站在一个初学者的角度来看，所有的 Linux 发行版几乎都是等价的。而在 2006-2007 年之间，一个最好的选择诞生了：Ubuntu.任何一款 Linux 发行版都有自己的优势，而 Ubuntu 却是最多新手青睐的选择。

你可以在 BSD 的官网中找到 BSD Unix 的指导和源代码。

一个很好的尝试方法就是使用被 Linux 迷们称为 LiveCD 的东西，它将系统完全的运行在 CD 中而不会改变你硬盘中的任何数据。这样可能有些慢，因为 CD 运行速度比较慢，但这是一个很好的方法来测试你能否适应这一切。

我还写了一篇入门教程来阐述“互联网与 Unix 基础”。

我曾经反对新手独自安装 Linux 或是 BSD。但时至今日，安装程序已经变得足够好以致新手也可以自己完成所有的安装了。尽管如此，我依然建议你和当地的 Linux 用户社群保持联系并寻求帮助。这没什么坏处，而且还可以让一切变得更顺利。

3.学习如何使用万维网以及编写 HTML

大部分由黑客文化创造的事物是远离人们的视线的，建立工厂、办公室、学校这些事情显然和黑客没什么直接关系。只有网页是一个例外，即便是政治家也要承认这个伟大而光辉的黑客玩具改变了世界。仅此一条理由(当然还有很多其他理由)，你就应该学习如何在互联网上工作。

这并不仅仅是打开一个浏览器这么简单的事情(这任何人都会作)，而是要学习如何编写 HTML——网页的标记语言。如果你不懂如何编程，那 HTML 会教给你一些习惯来帮助你学习编程。因此建立一个自己的主页，并坚持用 XHTML 编写——一种比过去的 HTML 更简洁的语言。(网上有很多的相关教程)

但仅是拥有一个主页并不能让你更接近一名黑客。网上到处都是各种各样的主页，大多都是些毫无意义、没有技术含量的垃圾——看上去很绚丽的垃圾。记住，垃圾都是一样的。(更多的信息可以参考“HTML 地狱”)

要想让你的页面变得有价值，那它必须要有内容——那些对其他黑客来说有趣且有益的内容。而这将是下一个主题中要说的……

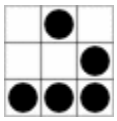
4.如果你不懂英语，请学习它。

我本人作为一个以英语为母语的美国人，过去并不愿提及这一点，因为这看起来有点像文化帝国的暴政。但一些其他语种的朋友多次敦促我指出英语是黑客文化和互联网的工作语言。因此掌握英语对于你在黑客社区的交流非常有帮助。

早在 1991 年我就发现很多黑客以英语作为第二语言在技术论坛中交流。在那时，我意识到英语较之其他语言拥有最多的技术词汇，因此也是完成黑客工作的最佳语言。基于同样的原因翻译用英语写成的技术类图书也往往不能令人满意(虽然都已经做完了)。

Linus Torvalds[8]，一个芬兰人，用英文注释自己的代码(他显然是想也没想就这样做了)。他流利的英语也是他能够在世界范围内招募到 Linux 社区的志愿开发者的一个重要因素。这是一个值得学习的榜样。

作为一个以英语为母语的人并不能保证你拥有很实用的黑客语言。如果你的书写不够流畅或是有着千疮百孔的拼写错误，很多黑客(包括我本人)将会无视你。虽然马虎的拼写并不总意味着草率的思想，但它们之间通常还是有关联的——我们不需要一个草率的思想者。如果你还不能写的很好，那就去学吧。



黑客文化现状

正像许多文化与金钱无关一样，黑客帝国是建立在名誉之上的。你试图去解决一些有趣的问题，而到底这个问题是否有趣以及你的解决方案是否出色，这些往往只有你的技术同行或是高手们才有能力去评价。

因此，当你玩这场黑客游戏的时候，你要知道你的得分主要取决于其他黑客对你的评价(这也是为什么在其他黑客皆称你为黑客之前你并不算是个真正的黑客)。这一事实常被掩盖于黑客独自工作的表面之下，同时黑客文化也避讳(虽然上世纪 90 年代后期这种避讳在逐渐减弱，但依然存在)承认个人的动力会受到外部观念的左右。

具体的来说，黑客帝国所秉承的就是人类学家所称之为的“奉献文化”。你从中获得地位与荣誉并不因为你领导着其他人，亦不因你的美丽，不因你有他人所求之物，而仅因为你的给予。或者说，你献出了你的时间和创造力，收获了技能。

若要获得黑客们的尊重，以下五件事是你应该做的：

1.编写开源软件

首先(最核心也是最传统的)，就是写一些其他黑客认为有趣或是有用的程序，并将源代码开放出来供整个黑客团体使用。

(我们也常将这种东西称为“自由软件”，但这会引起很多人的疑惑——到底这个“自由”该如何定义。所以我们现在更多的称其为“开源”软件。)

在黑客帝国中那些受到最多尊重的大神，是编写并放出那些有广泛需求的强大程序供他人使用的人。

此处有一个历史的转折点。虽然黑客们总是视身边的开源开发者为社区的强大核心，但在上世纪 90 年代中期之前，大部分黑客大多数时间还是在做闭源开发工作。在 1996 年我撰写这份指南的第一版时，这种现象依然普遍。而开源软件成为主流则是在 1997 年之后的事情。今天，“黑客社区”与“开源开发者”基本成了描述同一种文化和群体的两种说法——但应该记住这并非是始终如此。

2.帮助测试和调试开源软件

应服务于那些代表并调试开源软件的人。在这个并不完美的世界中，我们不可避免的花费大量的编程时间在调试错误上。这就是为什么开源软件作者会告诉你一个优秀的测试者(那些知道如何清晰的描述问题，能够准确的定位错位，能容忍预先版的缺陷，并愿意制做一些简单的诊断程序的人)对他们来说是比宝石还要珍贵的。甚至他们中的某些人可以让整个调试阶段从一场旷日持久而精疲力竭的噩梦变成一段不无益处的小困扰。

如果你是一个新人，那么找到一款你感兴趣且正在开发阶段的程序，并试着成为一名优秀的测试者。从帮助测试程序到帮助调试再到帮助修改，这是一个很自然的过程。你会从中学会很多，并且会在日后得到你所帮助的人的回报。

3.公布有用的信息

另一件有益的事情就是在诸如“常见问题”这样的网页或是文档中收集和过滤有趣的或

是有用的信息。并将这些公诸于众。

重要的技术问答的维护者会得到同开源作者同样的尊重。

4.帮助维护基础设施的运作

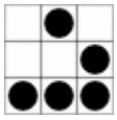
黑客文化(就此而言，还有互联网工程的开发工作)的运作离不开志愿者的参与。为了维护它的运作，有大量必须却枯燥的工作需要去完成——管理邮件列表，监管新闻组群，维护大型软件的存档站点，发展 RFC 以及其他技术标准。

而做这些事的人也会受到相当的尊重，因为谁都知道这些工作会消耗大量的时间而且显然没有编写代码有趣。做这些来奉献吧。

5.服务于黑客文化本身

最后，你可以服务并传播这文化本身(比如写一篇准确描述如何成为黑客的文章:-))。通常你不必如此，除非你已经在以上四条中取得了一定的地位与尊重。

准确的讲，黑客文化中没有领导，但却有诸如文化英雄、长老、历史学家或是发言人一类的角色。当你在这个战线中呆的足够久了，你也会成为以上几种人之一。但请注意：黑客们很排斥那些太过于个人主义的长老，所以赤裸裸的去追求这种名誉是危险的。与其费力的去追求这些，不如摆正自己的位置，并虚心和谦恭的面对自己现有的地位。



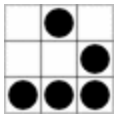
黑客与书呆子的关系

并不像传言中的那样，成为一名黑客并不一定要成为一个书呆子。但很多的黑客确实是书呆子。某些时候做一些远离社会的工作有助于你将精力集中于眞正重要的事情上，比如思考和钻研。

正因如此，许多黑客因被称为“极客”[9]而引以为傲——这标志着他们独立于通常社会观念之外(这也正是为何科幻小说或是策略游戏一类的东西对黑客如此钟爱)。“书呆子”这种称呼在上世纪 90 年代颇为盛行，因为那时这个词仅含有轻度贬义而“极客”在当时则贬义更甚。而到了 2000 年之后，他们两者调换了角色，至少在美国的流行文化中是如此的。时至今日，即便不是技术人才也会因以“极客”自居为傲。

如果你能集中精力去研究黑客技术而又能养家糊口，那就太棒了。在我还是新手的上世纪 70 年代，想做到这一点远没有现在这么简单——当今的主流文化对技术怪才越来越友善了。甚至有很多人发现黑客其实可以作为高素质的恋人或配偶。

如果你因为没有家庭而研究黑客技术，那这也不错——起码没有那么多麻烦让你分神。而且你可能很快就会组建一个自己的家庭了。



风格要点

再次重申，想要成为黑客，首先要进入黑客的思维方式。以下这些即便你不在电脑前也可以做的事情可能会对你有所帮助。这些并不能替代对黑客技术的钻研(什么都不能替代)但很多黑客都这样做，并认为这些和黑客的本质有着某种关联。

- 学着去写好你的母语。虽然很多人想当然的认为程序员基本都不写作，但其实相当多的黑客(包括我认识的所有那些有所成就的人)都是优秀的作家。
- 阅读科幻小说，并参加科技会议(这是结识黑客或原黑客的好方法)。
- 以武术的形式训练。武术所要求的坚定的意志与黑客所要求的非常相似。最受黑客们喜爱的还是亚洲的徒手格斗术，诸如跆拳道、空手道、中国功夫、合气道或是柔术。西方的击剑与亚洲的剑道也有非常多的追随者。上个世纪90年代末期，射击运动也开始逐渐兴起。最符合黑客精神的武术是那些强调坚定意志、放松精神、增加控制力的，而不是强调原始的力量、运动力和柔韧性的。
- 学习一种冥想之法。黑客之间最流行的便是禅法(最重要的是任何人都可以从禅法中受益而不必改变自己的宗教信仰)。其他的方式也行得通，但需注意不要选择一条让你做出什么疯狂举动的道路。
- 拥有一对善于倾听音乐的耳朵。学会去欣赏各种奇妙的声音，并学着去演奏乐器或是歌唱。
- 提高你对双关语和文字游戏的认识。

以上这些你做的越多，你就越接近天然的黑客本质。为何会如此我也不是很清楚，但它们都能锻炼左右脑之间的协作这一点看来很重要。作为一个黑客必须能够做到在进行逻辑推理的同时还能够跳出一个问题的表面逻辑。

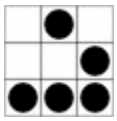
工作既娱乐，娱乐既工作。对于一个真正的黑客而言，“娱乐”、“工作”、“科学”与“艺术”之间的界线趋于消失，或者是合成为一种高级的创造性娱乐。但不要被狭隘的技术范畴所局限。虽然很多黑客都自称为程序员，但他们几乎都拥有众多其他的相关技能——常见的有系统管理、网页设计和PC硬件故障排除。每一个黑客都是一个系统管理员，同时也是熟练的脚本程序员和网页设计者。黑客们从不半途而废，如果他们决定学习某项技能，他们最终都要精通此道。

最后，以下这些不要做：

- 不要使用一个愚蠢而浮夸的ID或假名。
- 不要在讨论组中打口水仗(其他地方也不要)。
- 不要以“叛客”[10]自居，也不要在这种人身上浪费时间。
- 不要发送或邮寄满是拼写和语法错误的邮件。

为了你的荣誉你该远离这些事情。黑客们的记性都很好——你可能因为早期做过的一些蠢事而多年不被认可。

使用假名的问题值得尤其注意。将自己的身份隐藏起来是愚蠢而幼稚的行为，是骇客、破解者[11]或其他什么低级生活方式的象征。而黑客们不这样做，他们为自己所做的事情为荣，并很乐于见到这些事情与自己的真名联系在一起。所以如果你有一个假名，请放弃它，在黑客文化中这只能标志着你是一个失败者。



历史标注：黑客、开源和自由软件

当初在1996年末我撰写此文的时候，许多与此文相关的事物并不是他们现在的样子。在这里用一些文字来澄清这些变化，以便让人们理解开源、自由软件和Linux对于黑客社区是怎样的一种关系。如果你对此并不关心，你可以直接跳到下一个章节中去。

我所描绘的黑客风尚与社区远早于1990年Linux的出现。我最初加入这一团体是在1976年，而这个团体则最早可追溯到20世纪60年代初期。但在Linux出现之前，大部分的黑客研究都是在私有系统或是像麻省理工学院的ITS这种从不在实验室外部署的少数本土实验性系统中完成的。虽然他们尝试去改变这一现状，但收效甚微。即便是在黑客社区中也只有极少数的信仰者在使用，更别说世界范围内流行的软件了。

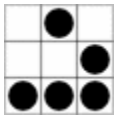
现在所谓的“开源”其实可以追溯到黑客社区创建之处，但直到1985年之前它都是一种不成文的民间行为而不是一种由理论和宣言所倡导的自发性运动。而这一切都在1985年划上了句号，头号黑客Richard Stallman(“RMS”)试图给他起个名字——“自由软件”。但他的命名行为同时也成了一场主张运动，他为“自由软件”加入了一个思想整合，致使很多的黑客社区都不愿接受。因此，“自由软件”受到了了黑客社区一些坚定的少数派的强烈反对(特别是与BSD Unix相关的人士)，但依旧得到了其余大多数人(包括我)低调而有所保留的使用。

尽管有所保留，在RMS倡导的定义和领导下，上世纪90年代中期还是在黑客社区中树立起了“自由软件”的大旗。而它受到了Linux的崛起所带来的一个严峻挑战。Linux给了开源开发一个天然的归宿。许多现在被称之为开源的项目都是在那时从私有的Unix系统迁移到了Linux平台上。围绕着Linux建立的社区瞬间爆发，比“前Linux时期”的黑客文化更庞大且趋于异化。RMS坚持要将这一系列的行动加入到他的“自由软件”运动中去，但却被迸发出的Linux社区以及他的创始人Linus Torvalds所阻止。Torvalds虽然沿用了“自由软件”的称呼，但却拒绝了RMS的思想整合。因此众多的年轻黑客纷纷效仿。

在1996年我首次公布这份黑客指南的时候，许多黑客社区迅速的围绕着Linux和少数其他的开源系统(尤其是那些BSD Unix的后裔)展开了重组。实际上在社区看来，那些我们用了几十年的时间在闭源系统上开发的闭源软件虽然还未过时，但看上去已如死去一般。越来越多的黑客们加入到Linux或Apache开源项目中去。

然而“开源”这个术语在1998年之前仍未出现。当它出现后，大多的黑客社区都在半年内通过了这一定义，而仅有少数黑客仍沿用“自由软件”的说法。自1998年以来——尤其是2003年以后——“黑客技术”与“开源(或自由软件)开发”这两个词变得越来越接近。时至今日，这两个概念仅有微小的区别，而且看起来未来也不会有什么变化了。

这些值得记住，然而，事情不总是这样的。



其他资源

Paul Graham 写了一篇名为《伟大的黑客》的文章，另一篇《在校生》则彰显了他的智慧。

有一篇名为《如何做个程序员》的文档是本文很好的补充。它不止针对代码和技能提供了有价值的建议，还提出了在程序团队中如何策划项目。

我还写了一篇《黑客帝国简史》。

我写过一篇论文——《大教堂与集市》，来解释 Linux 和开源文化是如何运作的。我在他的续篇《家园与智域》中更加直观的解释了这一问题。

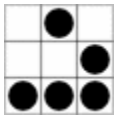
Rich Moen 写了一篇优秀的文档，叫做《如何建设一个 Linux 用户组》。

Rich Moen 还和我合作了一篇文档《如何提出精明的问题》。这将帮助你找到一条最有效的寻求帮助的途径。

如果你需要个人计算机、Unix 或是互联网的工作原理上的指导，可以参考《Unix 与互联网基础指南》

若你想发布软件或是编写软件补丁，你可以参考《软件发布实践指南》的指引。

如果你喜欢禅谒，你或许也会喜欢《无根之根:名家大师的 Unix 公案》。



常见问题

问：我如何知道我已经是一名黑客了？

答：问自己以下这三个问题：

- 是否能流利的说出代码？
- 是否能被黑客社区的目标与价值观所认同？
- 是否曾有过颇具规模的黑客社区成员称你为黑客？

如果以上这三个问题的答案都是肯定的，那么你已经是一名黑客了。缺一不可。

第一项测试是关于能力。只要你有本文前面所讲的最基本能力便可通过这一项。若你已经从一个开源项目中获取了大量的代码，那这并非难事。

第二项测试是关于态度。如果你像是与生俱来的拥有前文所述的黑客五大心态而不像是后天学来的，那你已经成功了一半了。这一半出于内在。而外在的另一半则要看你能在多大程度上加入到黑客社区的长期项目中去。

这有一份不完成却具有代表性的项目名单：Linux 的改进与传播是否与你有关？你是否热衷于软件的自由化？反对垄断？你是否信仰计算机该被赋予让全世界变得更加富有与人道的重任？

但这里有一个值得注意的地方。黑客社区需要争取一些利益——主要是捍卫言论自由以及抵制那些会让开源非法化的“知识产权”。有些长期项目的外在表象就是像“电子前沿基金会”这样的民间自由组织那样对这些运动提供支持。但在此之上，很多黑客希望将黑客态度规范为一种怀疑论的政治态度，我们知道这只是枉费心机而已。如果有人以黑客态度为名聘你进入国会任职，那他们就错了。正确的回应应该是“不说话而只给他们看你的代码”。

第三项测试含有一个有趣的递归。我在“什么是黑客”那一节中提到过，作为一名黑客就该是一个特别的亚文化和一个与之相关的社会网中的一员——一个是内在而一个是外在。很久以前，黑客是一个非常缺乏凝聚力且个人主义的群体。但在过去的三十年，社会网的增加与互联网越来越容易的连接黑客亚文化的核心的重要性越来越显著。一个很显著的变化就是——在本世纪，我们有了自己的T恤。

那些研究网络并喜欢把黑客文化放到“无形的学院”专栏下的社会学家已经注意到，这些网络的一个重要特点就是它们有守门人——需要有社会权威性的核心成员去批准新成员加入到网络中。因为黑客文化这一“无形的学院”是松散与随性的，所以它的守门人也是很随性的。但每个黑客都从心底里清楚的一点就是——不是任何黑客都是守门人。守门人需要拥有某些资历地位或成就才能有此头衔。这很难量化，但每个黑客见到之后便知道他是。

问：你能教我黑客技术么？

答：自从发表此文开始，我每周(甚至是每天)都会收到很多类似于“请教我黑客技术”的要求。很不幸，我没有时间和精力去做这些。而且作为一个开源主义者，这已经耗去了我110%的时间了。

即便我有时间也不会教，黑客的基本能力和态度就是自学。你会发现当一个真正的黑客愿意教你些什么的时候，他会很反感用一种填鸭式的方式把自己会的东西灌输给你。

先学些东西，起码说明你尝试过了，并且有一定的自学能力。然后再向黑客们提出一些具体的问题。

如果你发邮件向黑客们寻求一些建议，有两件事你最好知道。首先，你会发现对于黑客而言，一个在书写上懒惰而粗心的人往往在思考上也是懒惰而粗心的——所以请注重拼写的准确性，并且不要忽视语法和标点的精准。第二，不要要求回复到一个非你发件账户的另一个账户上去。我们发现这样做的人往往都是使用盗取的邮箱来发件的窃贼，而我们无意鼓励或协助一个窃贼。

问：那么我该如何开始？

答：你最好的起点应该是参加一个Linux用户群的讨论会。你可以在LDP通用Linux信息站找到很多这样的用户群。这其中可能有些离你不远，还有些与大学或学院相关。如果你有需求，Linux用户群的成员可能会给你提供一款Linux并帮助你安装和学习它。

问：你是从什么时候开始起步的？我现在学这些是不是有些晚了？

答：无论什么时候想要学习都是好的。大多数人都是在15-20岁之间对此产生兴趣，但其实有不少的例外。

问：想成为一名黑客我需要学习多久？

答：那要取决于你的天才程度和勤奋程度。只要足够的努力，大部分人都可以在十八个月到两年之内收到显著的成效。但不要以为这就是终点了。在黑客领域(其实也包括其他领域)要用大概十年左右的时间才能达到精通。而如果你是个真正的黑客，你会用毕生的时间去学习和磨练自己的技艺。

问：Visual Basic 是一种不错的入门语言么？

答：如果你问这个问题，那基本可以确定你想在 Windwos 系统下学习黑客技术。这本身就是个糟糕的想法。想要在 Windows 系统下学习黑客技术就相当于想要穿着盔甲学习跳舞，我没开玩笑。不要迈出这一步，它太丑陋了，而且永远会丑陋下去。

关于 VB 还有一个很重要的问题——它不可移植。虽然确实有一个用 VB 实现的开源原型，但 ECMA 的适用范围并不比一个小型设备的编程接口更广泛。在 Windows 中，大部分的库仅供一个供应商(微软)专有。如果你不是非常小心的去甄别该用哪个库——小心到一个新手不足以达到的地步——那你就会很不幸的被限制在微软公司所支持的这几个仅有的平台中了。如果你从 Unix 起步，那将有更多优秀的语言和库可以选用。比如 Python。

并且就像其他 Basic 语言一样，VB 是一个先天设计不良的语言，它同样会带给你不良的编程习惯。不，别让我详细的说明，这足够出一本书了。还是学一种设计优良的语言吧。其中一个最坏的习惯就是逐渐依赖一个供应商的专有库、组件和开发工具。一般来讲，任何一种语言若不支持 Linux 或一种 BSD，并且/或是不能支持三个以上的供应商提供的操作系统，那这种语言就不适于黑客学习。

问：你能帮我破解系统，或是教我如何破解么？

答：不！读完这篇文章之后还会有此一问的人简直是不可救药。通常类似内容的邮件都会被我忽略掉或者是得到一份粗暴的回复。

问：我如何能得到别人的账户和密码？

答：这是骇客所为，走开，蠢货！

问：我如何才能进入/阅读/监视别人的邮箱？

答：这是骇客所为，滚吧，白痴！

问：我怎样才能窃取到 IRC 通道的运算特权？

答：这是骇客所为，滚远点，笨蛋！

问：我被攻击了，你能帮助我免受攻击么？

答：不行。我经常被问及类似的问题，而提问的大都一群是使用 Windows 的可怜虫。在 Windows 系统中不存在有效的安全方案来阻止骇客的攻击。代码与构架上太多的缺陷致使保护 Windows 系统的安全就像是用筛子划船一样困难。唯一可行的安全方案只有换成 Linux 或其他一些类似的操作系统，起码它们有专门的安全设计。

问：我的 Windows 软件出问题了，你能帮我么？

答：好的。进入 DOS 命令行并输入“format c:”。你所遇到的所有问题都将在几分钟之内解决。

问：在哪里我可以找到真正的黑客进行交流。

答：最好的方法是找一个你附近的 Unix 或 Linux 用户群并参加他们的会议(你可以在 ibiblio 的 LDP 站中找到许多用户群列表的链接)。

(我经常说你不会在任何 IRC 上找到真正的黑客，但我意识到这一点已经改变了。像是 GIMP 或 Perl 这样的黑客社区已经开始有自己的 IRC 了。)

问：你能提供一些与黑客技能相关的书籍么？

答：我维护的一个“Linux 阅读指南”可能对你有帮助。而“Loginataka”可能也会令你感兴趣。

至于 Python 的介绍，可以直接到 Python 的网站上去找教程。

问：要成为黑客是否需要精通数学？

答：不，黑客并不需要用到多少正式的数学或是算术。特别是不需要三角、微积分或是解析几何(像 3D 电脑绘图这种少数的特殊程序领域是个例外)。了解一些典型的逻辑运算和布尔数学即可。一些有限数学(包括有限集理论、排列组合或是象限理论)也会对你有所帮助。

更重要的是，你需要有一个像数学家那样良好的逻辑思维和精准的推理思路。当数学无法帮助你的时候，你需要用毅力与智力去改造数学。如果你缺乏这种智力，那你很难成为黑客；如果你缺乏这种毅力，那你最好现在就培养。

我认为确定逻辑能给你带来什么的一条不错的方式是去看 Raymond Smullyan 的《这本书叫什么？》。Smullyan 那些有趣的逻辑难题把黑客精神体现的淋漓尽致。试着去解决这些问题是一个不错的开始，享受解决问题的过程则更为可贵。

问：我应该首先学习何种语言？

答：如果你还不懂 XHTML(HTML 最新的分支)那就先从它开始。有太多的 HTML 书籍流于表面而粗陋不堪，真正优秀的却凤毛麟角。我比较钟爱的是《XHTML 与 HTML:权威指南》。

但 HTML 并不能完全算作是编程语言。当你准备开始编程的时候，我建议从 Python 开始。你可能会听到许多人建议学 Perl，而且 Perl 也比 Python 更流行。但它更难于学习(至少我这么看)并且设计并不如 Python 那样优秀。

C 的确非常重要，但也比 Python 和 Perl 都要难。不要从 C 学起。

Windows 用户，不要安装 Visual Basic。它会带给你不良的习惯，并且离不开 Windows 系统。尽量避免这种事情发生。

问：我需要怎样的硬件？

答：在过去，个人计算机的运算速度不足且存储量小，这给黑客学习带来了人为的阻碍。但从上世纪 90 年代中期开始，这一切都划上了句号。只要是 Intle 486DX50 以上配置的机器都足以胜任开发工作、界面以及互联网交流，并且你能买得起的哪怕是最小的硬盘都已经足够存放所需的数据了。

在选择机器时最重要的一点就是看硬件是否能被 Linux 兼容(或 BSD 兼容，如果你选择走这条线路)。同样，这一点现在也是几乎所有机器都能做到了。唯一需要重点关注的在于调制解调器和网卡——有些设备属于 Windows 专有而不能在 Linux 下工作。

有一个关于硬件兼容性的说明，在这里可以找到最新版本。

问：我想要做些贡献，你能帮我挑选一个项目去做么？

答：不行，因为我不知道你的天赋和兴趣。你必须学会自我激励，否则无法成功。这也就是

为什么让别人选择你的方向永远都不会成功。
试试这个：在鲜肉网的项目公告里看几天。当你发现了一个项目让你觉得“酷！我想做这个！”的时候，就加入好了。

问：我是否应该痛恨或攻击微软呢？

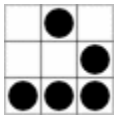
答：不，不用。这并不是说微软不令人讨厌。黑客文化先于微软出现，而且在微软消失后也不会消失。有时间去痛恨微软还不如去磨练自己的技术。写出一流的代码——这是最有力的痛击微软的方式。

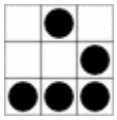
问：但开源软件不会让程序员无法生活么？

答：看起来不会——目前看来，开源软件行业是在创造就业机会而不是减少。如果编写一个程序相较之该程序出现之前能让网络经济有所提升，那无论这个程序完成后是否开源，程序作者都会从中得利。而且无论有多少“自由软件”出炉，依然会有更多的需求与订单。我在开源页面中对此问题做了更多的阐述。

问：我从哪能获得免费的 Unix？

答：如果你还没有在机器中安装 Unix，在这个页面的其他地方我给出了大多数人通常使用的免费 Unix 链接。作为黑客，你应该有主动性、自发性和自学能力。那么就从现在开始吧……





译者注释：

[1]原文为 Jargon File，既著名的《黑客辞典》。 返回>>

[2]原文为 Life，是由 John Horton Conway 在 1970 年遵循冯·诺依曼的自我复制理论编写的一款逻辑游戏。 返回>>

[3]原文为 ARPAnet，是当今国际互联网的前身。 返回>>

[4]原文为 Usenet. 返回>>

[5]原文为 Cracker，意为“破坏者”，本文译为“骇客”。与 Hacker 所代表的“黑客”作为区分。 返回>>

[6]原文中给出的链接已失效，此处链接为我根据原文说明自行搜索的，不确定和作者原本引用的内容是否相同。 返回>>

[7]原文为 Unixes，此处可理解为广义的 Unix 系统，既包括 Unix 的所有衍生版以及拥有独立内核代码的其他 Unix-Like 系统(如 Linux)。 返回>>

[8]林纳斯·托瓦兹，Linux 系统之父，现在主要从事 Linux 系统内核的维护工作。 返回>>

[9]原文为 Geek，又译为“技客”或“奇客”。原为俚语，指反常的人、畸形的人或野人。现多指对计算机或网络技术有狂热兴趣并投入大量时间钻研的人。 返回>>

[10]原文为 Cyberpunk，此处与前文的“Cracker/骇客”含义相近。 返回>>

[11]原文为 warez d00dz，形容从事破解和 0-day 漏洞攻击工作的人的专有名词。 返回>>