如何成为黑客

2025-09-10 · Articles · Manual, 转载

> How To Become A Hacker, Eric Steven Raymond, Thyrsus Enterprises, < esr@thyrsus.com >

How To Become A Hacker

Eric Steven Raymond, Thyrsus Enterprises, < esr@thyrsus.com >

Copyright © 2001 Eric S. Raymond

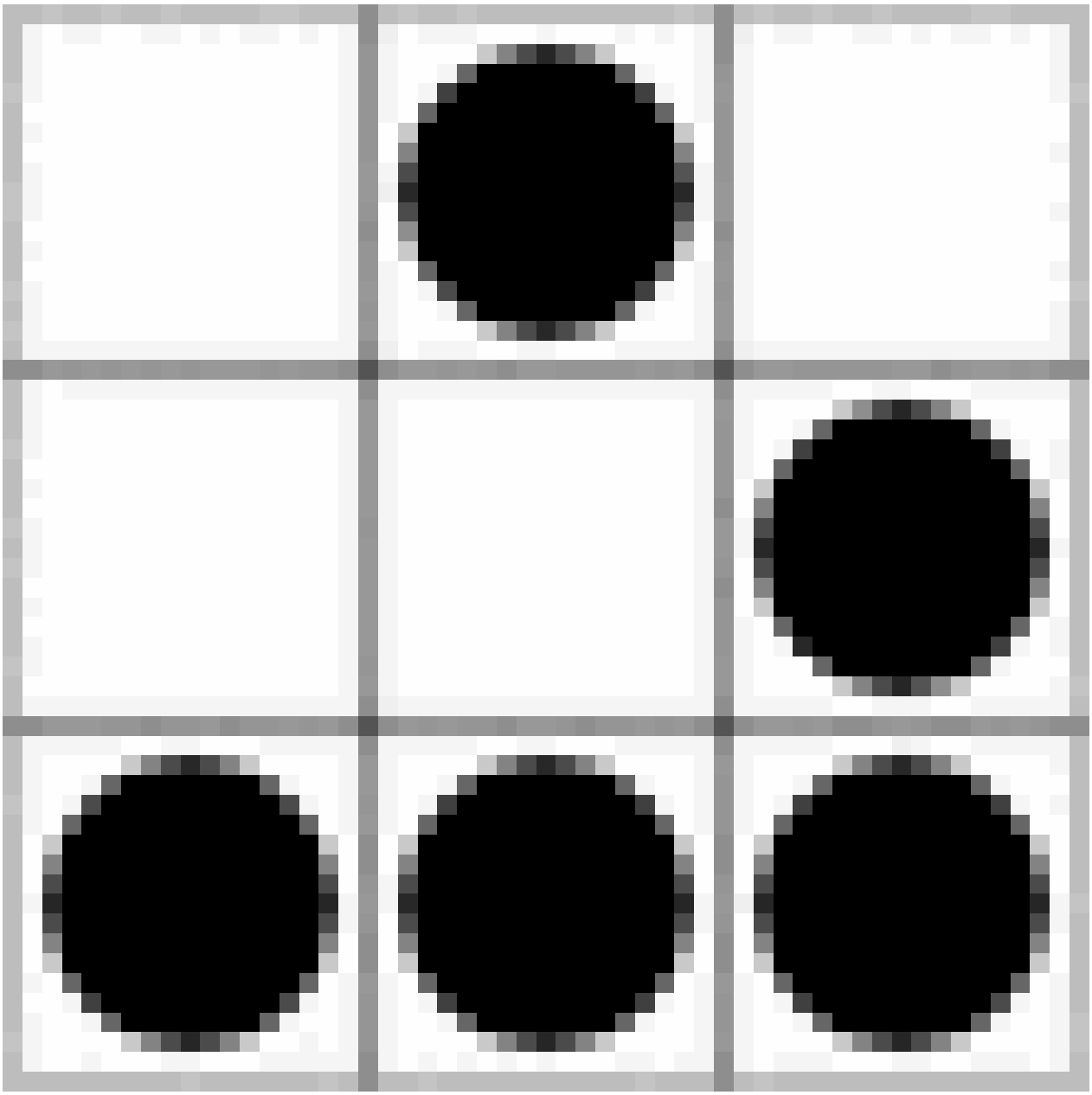
翻译：柯非, < zer4tul@gmail.com >

这篇译文基于 2020.01.03 更新的原文修订版 1.52。

如果对译文有任何意见或者建议，**请发 Issue，或直接发 Pull Request 给原仓库维护者。**

原仓库：

zer4tul/hacker-howto



为何会有这篇文档

身为新黑客词典（The Jargon File）和许多其他广为人知的同类文章的作者，我常收到热心的网络新人的电子邮件，问及（大意上是）“如何成为一名魔法师似的黑客?”。1996 年的时候我注意到这个重要的问题并没有相关的 FAQ 或文档页面，所以我写了一份。许多黑客认为这篇文章是权威的，我觉得它应该是吧。此外，我不会寻求在这个话题上的独立著作权，如果你不喜欢在这里读到的内容，自己写一篇吧。

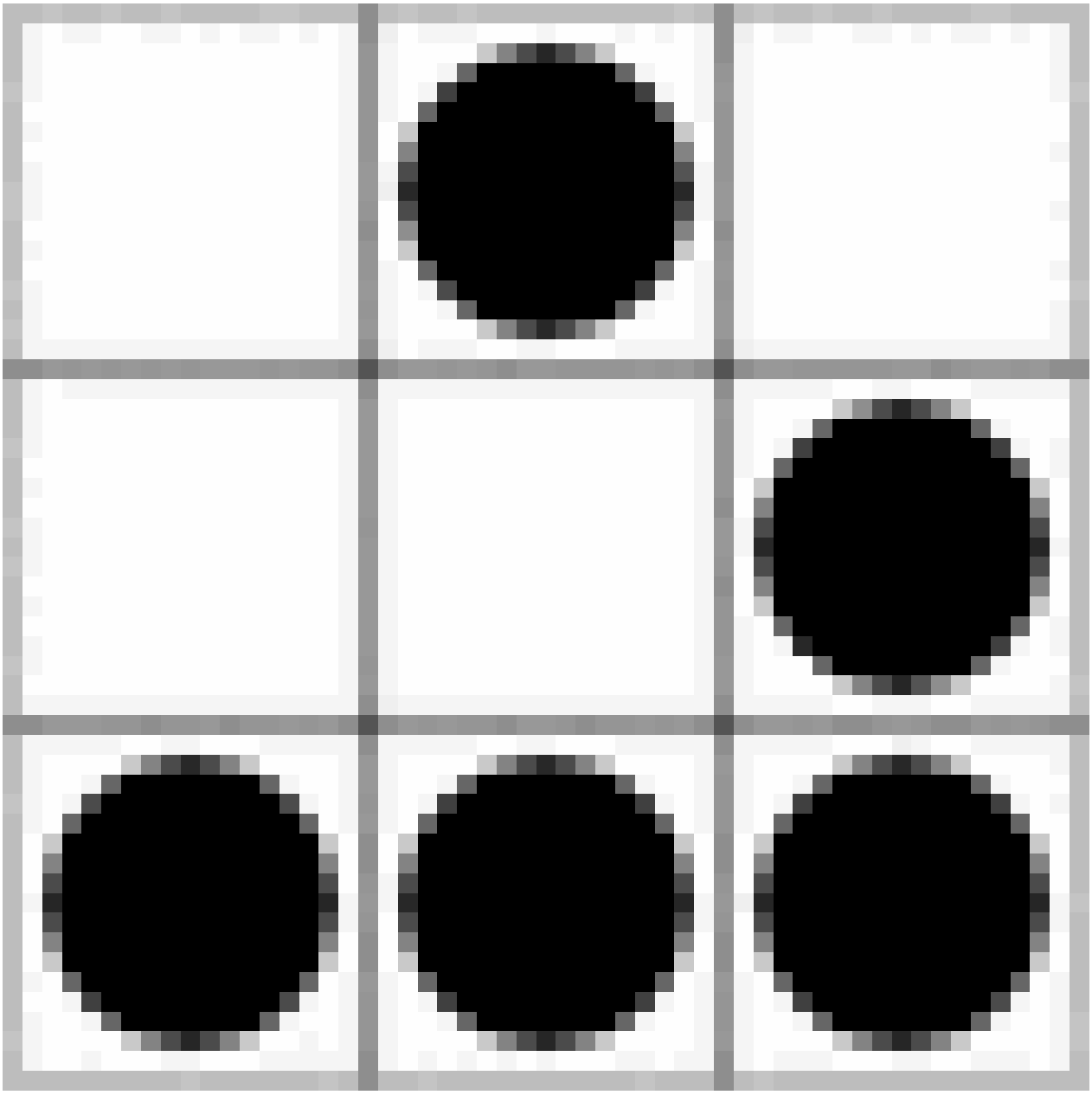
如果你是在离线阅读本文，可以在 http://catb.org/~esr/faqs/hacker-howto.html 找到本文的最新版本。（译注：本文的最新中文版可以在这里找到）

注意：本文的末尾有一系列常见问题。请在向我发邮件询问关于本文的任何问题前 **再三阅读**。

目前本文有许多语言的翻译版：阿拉伯语，白俄罗斯语，中文，捷克语，丹麦语，荷兰语，爱沙尼亚语，德语，希腊语，意大利语，希伯来语，挪威语，波斯语，巴西葡萄牙语，罗马尼亚语，西班牙语，土耳其语，瑞典语。请注意，由于本文不定期更新，这些翻译版可能存在不同程度的过时。

本文里九宫格中的 5 个黑点的装饰图被称作 glider。这是一个使很多黑客多年痴迷的被称作康威生命游戏（LIFE）中，具有令人惊奇特性的简单图案。我认为它是很好的黑客精神徽章 —— 抽象，初见的时候感觉有点神秘，通过它复杂的逻辑可以通向整个世界。如果你想了解更多关于 glider 的信息，请看这里。

如果你觉得这篇文章有价值，请在 Patreon 或者 SubscribeStar 上给我一点赞助。也请考虑通过 Loadsharers 赞助其他为你提供了有价值代码的黑客。小额的赞助也能够聚小流成江海，使为你提供帮助的人从繁重的劳动中解放出来，创造更多的价值。



什么是黑客

新黑客词典（Jargon File）中有数个“黑客”的定义，主要形容“技术专才”或“有志解决问题及超越极限之人”。要成为黑客，有两个要点。

这可以追溯到几十年前第一台分时小型电脑诞生，ARPAnet 实验也刚展开的年代，那时有一个由程序设计专家和网络名人所组成的，具有分享特点的文化社群。这种文化的成员创造了“hacker”这个词。他们建立了互联网，他们发明了现在使用的 Unix 操作系统。他们管理 Usenet 讨论组。他们令 WWW 运

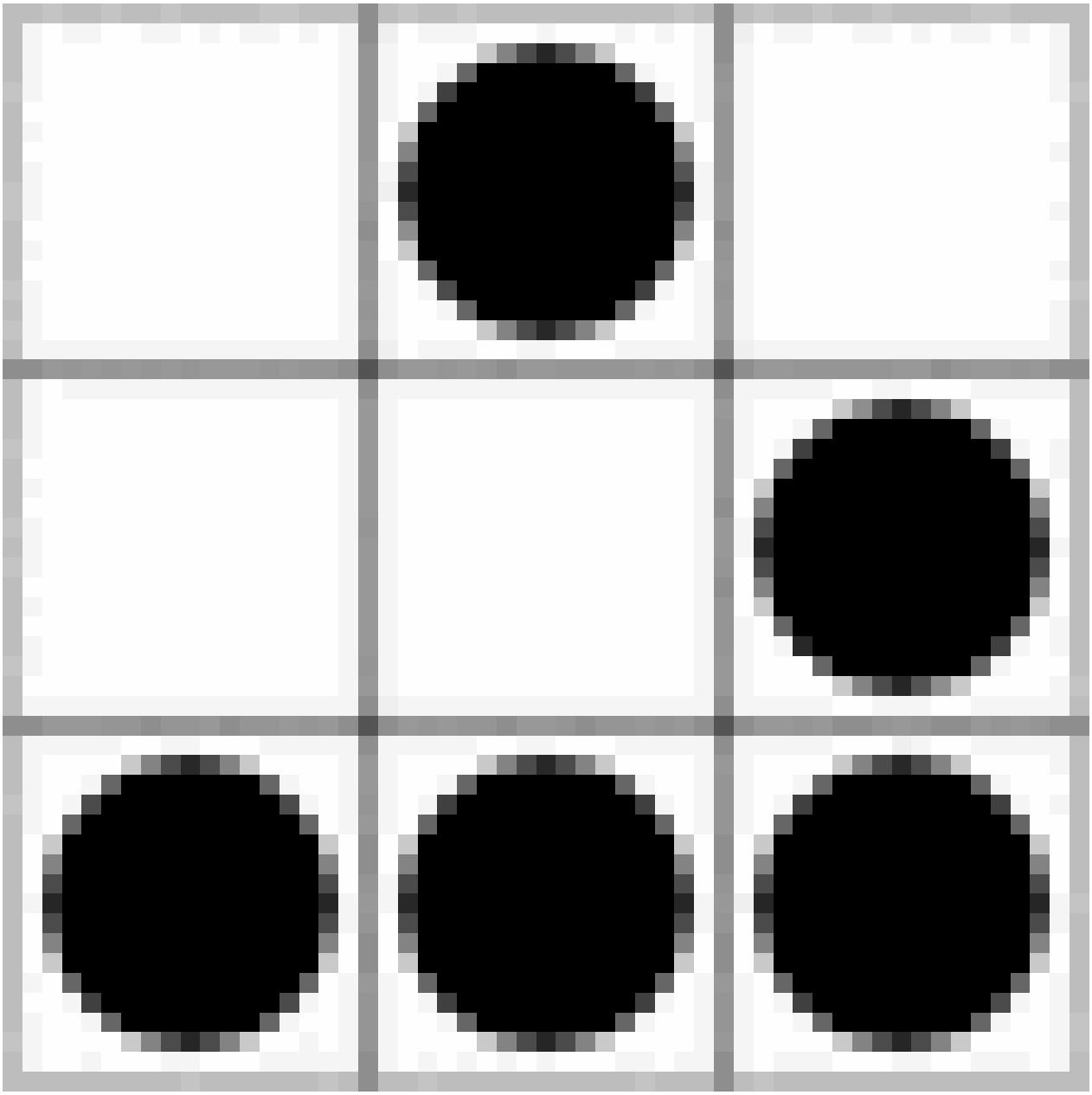
作。因此，若你有上述的特性及参与同类的社区，亦有对以上种种作出贡献，同时社区的人知你是谁又称你为“hacker”，你便是黑客。

然而，黑客的理念并非只局限于软件社区。有很多人将黑客的态度应用于其他事物，如电子或音乐上——实际上，黑客的理念存在于任何科学及文学。由于了解黑客的理念及精神，软件社区的黑客亦会称后者为黑客。有些人亦认为黑客的理念是独立于黑客所从事的媒体。然而，我们将在这篇文章专注讨论软件黑客的技巧，态度及传统。

另外，有一群人亦称自已为“黑客”，他们（多数是年青人）用电脑侵入其他电脑的系统作出破坏。黑客们称这群人为“Cracker（破坏者）”，亦不认同他们为黑客。多数黑客会认为 Cracker 是懒惰，不负责任，不杰出的人。有能力侵入安全系统并不能使你成为黑客，正如可以用铁丝来偷车并不能使你成为汽车工程师一样。不幸的是很多作家及报道均称这群人为“黑客”。这一直使黑客们非常恼火。

黑客与 Cracker 的主要区别在于，前者搞建设，后者搞破坏。

如果你想成为一个黑客，请继续读下去。如果你只想做一个 Cracker，请到 alt.2600 讨论组，并做好当你发现自己不如想象中聪颖的时候进 5 到 10 次监狱的准备。关于 Cracker 我就说这么多。



黑客的精神

1. 世上仍有大量迷人的事情等待解决
2. 同样的问题不应被重复处理两次
3. 拒绝重复和沉闷的事情
4. 自由万岁
5. 精神不能代替能力

黑客们解决问题，建设事物，他们崇尚自由和无私的双向帮助。要被他人承认是一名黑客，你必须表现得你具备了这样的态度。而要表现得你具备了这种态度，你必须彻彻底底的坚持它。

如果你认为培养黑客的态度只是一条在这个文化圈中得到认同的路子，那就错了。成为具备这种素质的人对 **你** 非常重要 —— 使你保持学习和成为黑客的自发性。正如所有创造性艺术一样，成为大师的最有效途径就是效仿大师的精神——不仅从理念上，还要从态度上效仿。

或许下面的这首现代禅诗很好的阐述了这个意思：

To follow the path:
沿着这样一条道路：

look to the master,
关注大师，

follow the master,
跟随大师，

walk with the master,
与大师同行，

see through the master,
洞察大师，

become the master.
成为大师。

如果你想成为一名黑客，反复阅读以下内容直到你相信它们：

世上仍有大量迷人的事情等待解决

作为一名黑客可以享受很多乐趣，同时需要付出相当多的努力。努力需要动力。成功的运动员从锻炼身体、超越身体极限中获得精神愉悦。类似的，作为一名黑客，你可以从解决问题、磨练技术和锻炼智力中获得乐趣。

如果你天生不是这样的人，那你需要设法变成这样的人以使你能够成为一名黑客。否则你将会发现你的精力会被诸如性、金钱、社会上的虚名之类让人分心的东西所消磨掉。

（你还需要对自己的学习能力树立信心——相信尽管你对某一问题了解得不多，只要你能解决其中一部分，并从中学习，你可以解决其他的部分——直到解决它。）

同样的问题不应被重复处理两次

创造性的智慧是非常有价值且稀缺的资源。它们不应当被浪费在重复发明轮子上，世上仍有大量迷人的新问题等着解决。

作为一名黑客，你应该坚信其他黑客的时间非常宝贵——所以你有义务共享信息，解决问题之后公布方案，这样其他人可以去解决新的问题，而不是忙于应付旧问题。

注意，“同一个问题不应该被重复处理两次”并不是说你必须认为所有已有方案都是最优的，或每个问题只有唯一的解决方案。通常我们从一个问题的最初解决方案中能够学习到很多东西。这很好，并且对于我们思考如何能做得更好来说，这通常是必要的。我们反对的是人为的技术、法律上的，或者机构性的设置障碍（例如闭源软件），使得一个好的方案不能被重复使用，逼得人们重造轮子。

（你不必认为你必须将 **所有** 你的创造发明都公布出去，虽然这样做的黑客将会赢得大家极度尊重。适当卖一些钱来换取足够的食物、租金和电脑并不违反黑客的价值观。用你的技能来养家糊口甚至致富都可以，只要你在做这些的时候别忘记你是一名黑客。）

拒绝重复和沉闷的事情

黑客（以及富有创造力的所有人）不应当被愚蠢的重复性劳动所困扰，因为这意味着他们并没有在做只有他们才能做的事情——解决新问题。这样的浪费会伤害所有人。因此，无聊和乏味的工作不仅仅是令人不爽，而是罪恶。

作为一个黑客，你应该坚信这一点并尽可能的将枯燥的工作自动化，这不仅仅是为了你自己，也为了其他人（尤其是其他黑客）。

（这里有一个例外。黑客有时会做一些看起来重复或枯燥的事情以进行脑力休息，或以此来锻炼一种技能，或以此获得某种除此以外无法获取的经验。但这是有选择的——有脑子的人不该被强迫做枯燥的事。）

自由万岁

黑客是天生的反独裁主义者。任何能向你发号施令的人能够迫使你停止解决令你着迷的问题。同时，按照独裁者的一般思路，他通常会给出一些极端愚昧的理由。因此，不论何处，任何独裁主义的作法，只要它压迫你和其他黑客，你就要和它斗到底。

（这并非向所有权威挑战。儿童需要监护，罪犯要被看管起来。如果服从命令得到某种东西比起用其他方式得到它更节约时间，黑客可以同意 接受某种形式的权威。但这是一个有限度的，有意的交易；那种权威想要的个人服从不是你应该同意给予的。）

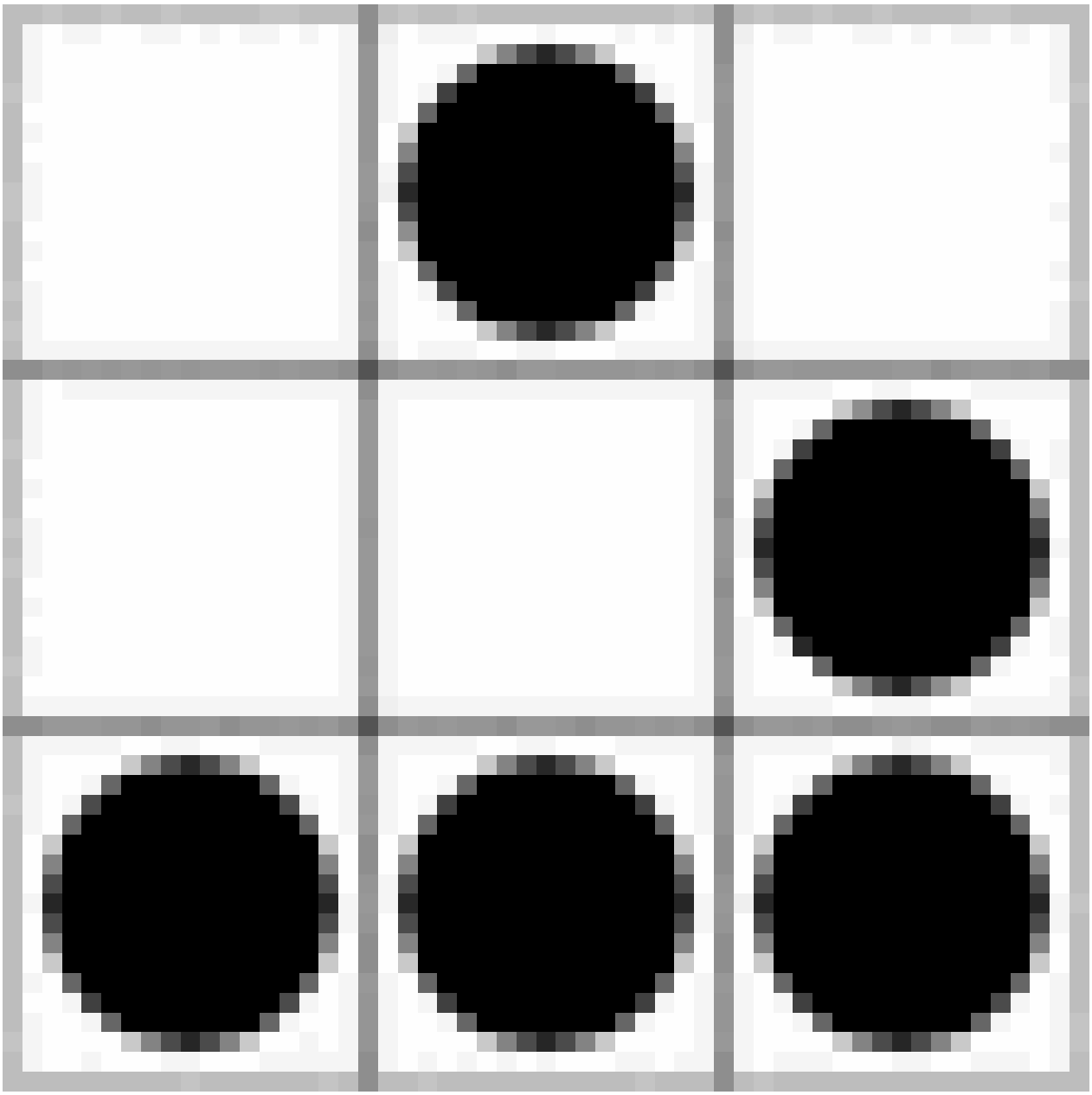
权威喜欢审查和保密。他们不信任自愿的合作和信息共享——他们只喜欢由他们控制的所谓“合作”。因此，作为一个黑客，你得对审查、保密，以及使用武力或欺骗去压迫有行为能力的人们的做法有一种本能的敌意。 同时你要有为此信念斗争的意愿。

精神不能代替能力

作为一个黑客，你必须培养起这些精神。但是仅仅有精神并不能使你成为黑客，也不能使你成为运动健将或摇滚明星。成为一名黑客还需要智力，实践，奉献精神和辛勤工作。

因此，你需要学会有怀疑态度和尊重任何能力。黑客不会为装模作样的人浪费时间，但他们尊重能力——尤其是从事黑客工作的能力，不过任何能力都是好的。很少人能具备的高要求能力尤其好，其中涉及脑力，技巧和专注方面的能力最好。

尊重能力，你就会享受到提高自己的能力所带来的乐趣——辛苦的工作和奉献将不再是苦差而是一种高度娱乐。想要成为一名黑客，这一点尤其重要。



基本黑客技能

1. 学习编程
2. 获取一个开源的 Unix 并学习运行和使用它
3. 学习使用万维网（World Wide Web，WWW）和 HTML（超文本标记语言）
4. 学习实用英语

黑客的态度很重要，但技能更重要。态度不能替代能力，在被别的黑客称你为黑客之前，你有一些基本技能需要掌握。

这些基本技能随着时间的推移和技术的革新也缓慢的变化着。例如以前的内容中包括了使用机器语言编程，最近包含进了 HTML。总的来说当前包括以下内容：

学习编程

理所当然，这是最基本的黑客技能。如果你一门计算机语言都不懂，我建议你从 Python 学起。它设计良好，文档详尽，并且对新人十分友好。尽管它是一门很好的入门语言，但它不只是玩具水平。它非常强大灵活，并且适用于大型项目。我写过一篇详细的对 Python 的评价。在 Python 的网站可以找到很好的教程。在 Computer Science Circles 也有一篇不错的第三方教程。

我早前曾经建议使用 Java 作为入门语言，但这篇评价改变了我的看法（请在文档中搜索“The Pitfalls of Java as a First Programming Language”）。如同文中尖锐指出的一样，一个黑客不能“像五金店中的管道工一样处理问题”。你需要知道所有的组件事实上都 **干了什么**。现在，我认为最好先学 C 和 Lisp，然后再学 Java。

比 Java 更好的选择是学习 Go。这种相对较新的语言很容易从 Python 上手，而且学习它可以使你在可能的下一步（学习 C 语言）中得到很大的帮助。此外，未来几年的一个未知数是 Go 将在多大程度上取代 C 语言成为真正的系统编程语言。在未来，这种情况可能会在 C 语言的大部分传统领域内发生。

另外有一点需要注意。如果一门语言帮你做了太多工作，它会同时是一个好的生产工具和一个不好的初学对象。不仅语言有这个问题，Web 框架比如 RubyOnRails，CakePHP，Django 也很容易让你流于表面，面对困难问题的时候束手无策，甚至无法对一个简单问题进行追查并给出解决方案。

如果你需要做一些重要的编程工作，你需要学习 C 语言，它是 Unix 的核心语言。C++跟 C 关系密切。如果你了解其中一种，学习另外一种应该不难。但是这两种语言都不适合作为入门学习。此外，如果你越避免用 C 编程，你的工作效率会越高。

C 的执行效率非常高，并且非常节省机器资源。不幸的是 C 的高效是通过让你手动进行许多底层资源（例如内存）管理来获得的。底层代码复杂并且容易出 bug，你需要花费很多时间来进行调试。鉴于当今的机器性能如此之高，这样的做法通常很不划算——通常更好的做法是使用一种稍微慢一些，不那么高效，但是能够 **大幅** 节省你的时间的语言。那便是 Python。

其他对黑客而言比较重要的语言包括 Perl 和 LISP。Perl 很实用，它广泛应用于动态网页和系统管理方面，因此即使你从不写 Perl 代码，至少也得能看懂。许多人使用 Perl 的理由和我建议你使用 Python 的理由一样，都是为了避免用 C 完成那些执行效率需求不那么高的工作。你需要能够看懂他们的代码。

LISP 之所以值得一学是基于另外的理由——当你最终掌握了它的时候，你将会获得巨大的启迪。它将使你在今后成为一个更好的程序员，即使你实际上很少使用 LISP 本身。（你可以通过为 Emacs 文本编辑器或者 GIMP 的 Script-Fu 编写插件或修改现有插件来很容易的学习 LISP。）

当然，你最好五种语言都会（Python，C/C++，Java，Perl 和 LISP）。除了是重要的黑客语言之外，它们也代表了截然不同的编程思路和方法，每一种都能让你受益匪浅。

但是单纯的堆砌语言是不可能成为一个黑客，甚至程序员的。你需要学会如何独立于任何具体的语言之外来思考编程问题。作为一名真正的黑客，你需要通过手册和你已有的知识掌握到在几天之内学会一门语言的要点。这意味着你需要学习许许多多不同的语言。

这里我无法给你完完全全的指导教会你如何编程——这是个复杂的技能。但我可以告诉你，书本和课程也不能做到（最好的黑客中，有许多，也许 **几乎** 都是自学成才的）。你可以从书本上学到语言的特点——这只是皮毛，但要使书面知识成为自身技能只能通过实践和虚心向他人学习。因此要做到（1）**读** 代码及（2）**写** 代码。

Peter Norvig，Google 最顶级的黑客之一，也是世界上最受欢迎的 AI 教材（译注：指“人工智能：一种现代方法”和“人工智能程序设计范例：通用 Lisp 语言的案例研究”等）的共同作者。他写了一篇名为 Teach Yourself Programming in Ten Years 的短文。他在文中提到的“编程成功的诀窍（recipe for programming success）”特别值得留意。

学习编程就像学习用优美的自然语言书写一样。最好的办法就是阅读大师的名著，试着自己写点东西，再读一些，再写一点，再读一些，再写一点……如此往复，直到你的作品达到如你在范文中所见的简洁和健壮。

我必须再提一下 How To Learn Hacking，这是一些简单的说明，但是学起来并不容易。

以前很难找到适合阅读的好代码，因为几乎没有大型程序的代码能够供新人阅读和练手。这种情况已经发生戏剧性的变化。开源软件，编程工具和操作系统（都是由黑客创造的）现在随处可见。这刚好带我们到下一个话题……

获取一个开源的 Unix 并学习运行和使用它

我假定你拥有或者能使用一台个人电脑（现在的孩子真幸福。黑客文化建立之初电脑贵得要死，没人买得起）。新手们向黑客技能迈出的最重要一步就是获取一份 Linux 或 BSD-Unix 的拷贝，将其安装在个人电脑上，并运行它。

没错，世上除了 Unix 还有其他操作系统。但它们都是以二进制形式发布的——你读不到源码，你也不能修改代码。在类似 Microsoft Windows 那样的闭源操作系统上学习黑客技术就像戴着脚镣学跳舞。

在 Mac OS X 上倒是可以，不过它只有一部分是开源的——你可能会撞墙，也必须很小心的避免养成依赖 Apple 专有代码的坏习惯。如果你专注于底层的 Unix，你可以学到一些有用的东西。

Unix 是互联网上的操作系统。虽然你不懂 Unix 仍然可以学会使用互联网，但若你不懂 Unix，你将不能在互联网上从事黑客活动。因此，现今的黑客文化是严重以 Unix 为中心的。（曾经不是这样，并且有一些老派的黑客对此仍然感到不太高兴。但是现今 Unix 和互联网的羁绊如此之强，连 Microsoft 也无法撼动分毫。）

所以，请安装一套 Unix - 我个人喜爱 Linux 但还有其他种类的（并且，你 **可以** 在同一台电脑上运行 Linux 和 Windows）。学习它，使用它，调教它。用它在互联网上冲浪。阅读它的代码，修改它的代码。你将获得比 Windows 操作系统上更好的编程工具（包括 C，LISP，Python 和 Perl）。你会觉得其乐无穷，学到比你想像更多更好的知识。

想要获取更多和学习 Unix 相关的信息，请参考 Loginataka。你或许还想看看 Unix 编程艺术（译注：这里给出的是原文链接。国内有翻译版出售）。

我认为博客 Let’s Go Larval!对于处在学习 Linux 阶段中的用户是非常易懂和有用的。这篇文章 How I Learned Linux 就是一个很好的起点。

想开始 Linux 之旅，请参考 Linux Online!。你可以从那里下载 Linux 或者（更好的主意是）找到一个当地的 Linux 用户群为你的安装过程提供帮助。

在本文最初的 10 年间，我认为从一个初学者的角度来说，所有 Linux 发行版都差不多。不过在 2006~2007 年间，一个事实上最好的选择出现了：Ubuntu。其他发行版各有所长，而 Ubuntu 对初学者最友好。注意，相比 Ubuntu 默认那个丑陋的几乎不可用的“Unity”桌面，Xubuntu 和 Kubuntu 更好用一点。

你可以在 www.bsd.org 找到 BSD 相关的帮助和资源。

一个试水的好办法是试试被 Linux 爱好者称为“Live CD”的东西，那是一个完全在光盘或者 U 盘上运行，而不修改你硬盘的发行版。它运行起来比较慢，因为光盘很慢，但是这是一个在做出任何不可挽救的改变前看看可行性的办法。

我写过一篇关于 Unix 和互联网基础的入门文章。

我曾经不建议新手独自安装 Linux 或者 BSD。现在它们的安装程序已经做得足够好，你作为新人也完全搞得定。尽管如此，我仍然建议和你当地的 Linux 用户群取得联系并寻求帮助。这没坏处，并且可以让整个过程更顺利。

学习使用万维网（World Wide Web，WWW）和 HTML（超文本标记语言）

大多数的黑客造物在你所不知的地方发挥着作用，帮助工厂、办公室和学校运转，这看上去跟普通人没太大关系。Web 是一个大大的例外，即便 **政客** 也承认这个巨大耀眼的黑客玩具正在改变着世界。单就这一个原因（当然还有其他理由）你就需要学习掌握 Web。

这并不仅仅意味着如何使用浏览器（谁都会），而是要学会如何写 HTML，Web 的标记语言。如果你不会编程，写 HTML 会教你一些有助于学习的思考习惯。因此，先完成一个主页。尝试坚持使用 XHTML，一种比标准 HTML 更清晰的语言。（Web 上有很多很好的初学者指南，例如这个）。

但仅仅拥有一个主页不能使你成为一名黑客。Web 里充满了各种网页。大多数是毫无意义的，零信息量的垃圾——界面时髦，能夺人眼球的垃圾还是垃圾（更多信息访问 The HTML Hell Page）。

所以，你的页面必须有内容——得是有趣并且/或者对其他黑客来说有用的内容。这是我们下一个议题要说的……

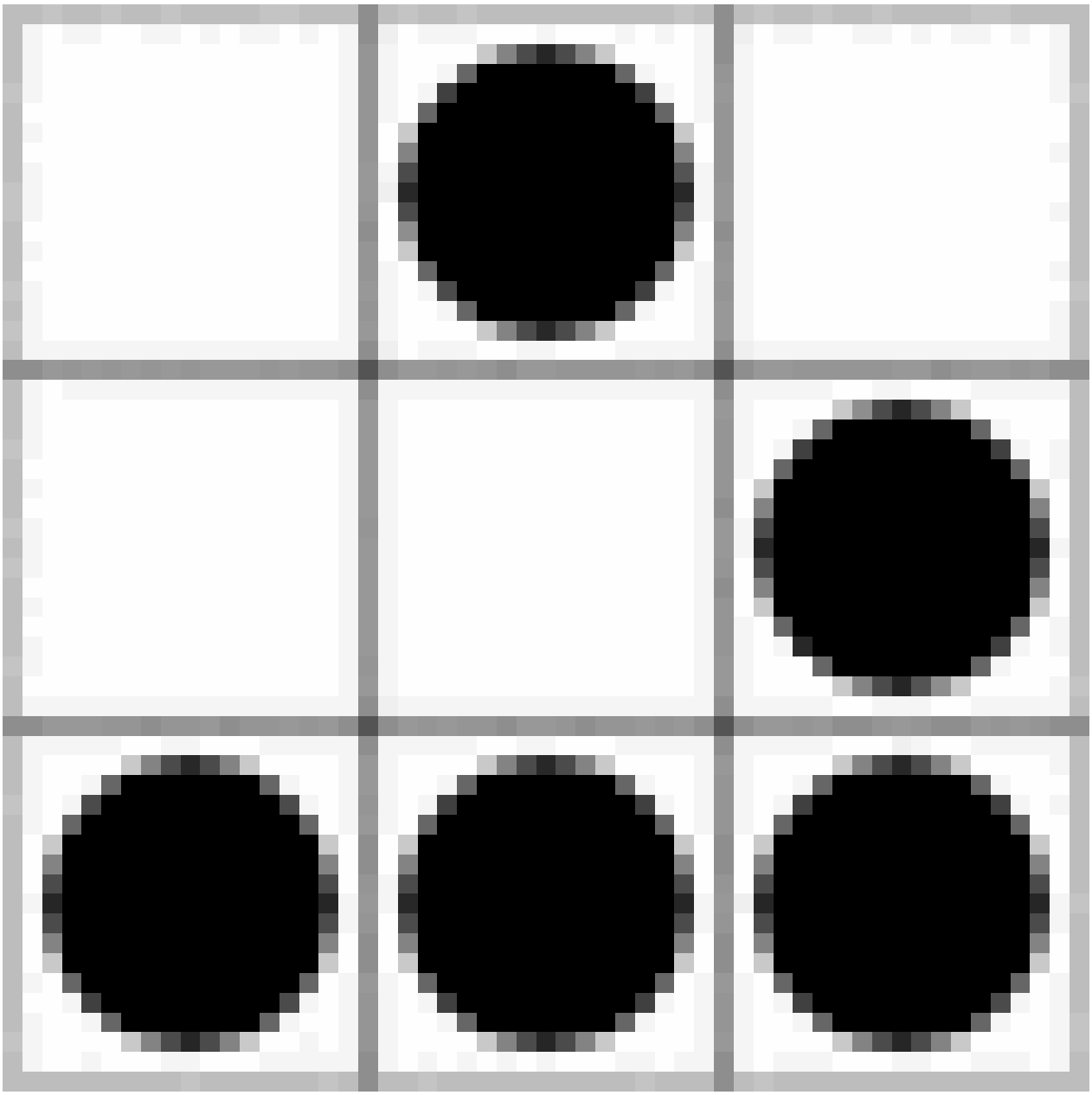
学习实用英语

作为一个美国人和一个以英语为母语的人，我以前很不情愿提到这点，免得成为一种文化上的帝国主义。但相当多以其他语言为母语的人一直劝我指出这一点，那就是英语是黑客文化和 Internet 的工作语言，你需要懂得以便在黑客社区顺利工作。

大概 1991 年的时候我了解到许多黑客在技术讨论中使用英语，即使在他们的母语都相同，英语对他们而言只是第二语言的时候也常如此。据我所知，当前英语有着比其他语言丰富得多的技术词汇，因此是一个对于工作来说相当好的工具。基于同样的理由，英文技术书籍的翻译（如果有的话）通常都不能令读者满意。

芬兰人 Linus Torvalds 用英语注释他的代码（很明显这不是凑巧）。他流利的英语成为他能够管理全球范围的 Linux 开发人员社区的重要因素。这是一个值得学习的例子。

即使作为一个以英语为母语的人也不代表你就具备了成为黑客所需的语言技能。一般而言，如果你写得像个半文盲似的，文中充斥着各种语法、拼写错误，多半得不到理睬。虽然不严谨的文笔并不总是意味着不严谨的思维，但我们发现这两者之间的关联还是挺紧密的。而我们不需要这种思维不严谨的人。如果你现在还没有具备这样的书写能力，赶紧培养。



在黑客社区中立足

1. 编写开源软件
2. 帮助测试和调试开源软件
3. 发布有用的信息
4. 帮助维护基础设施运转
5. 为黑客社区服务

像大部分非盈利社区一样，黑客社区靠声誉运转。你设法解决有趣的问题，但问题是否有趣及解决方法是否有效，需要由那些和你具有同样甚至更高技术水平的人去评判。

因此，要玩黑客这个游戏，你需要以其他黑客对你技能的评判作为对自己的评价（所以我说，在其他黑客称你为黑客之前，你不是一个真正的黑客）。这个事实常被人误解（从 1990 年代后有所好转，但还是很严重），人们认为黑客都是不在乎别人的评价，孤僻的人。这实际上是一个黑客文化的禁忌。

特别地，黑客被人类学家们称为 **奉献社区**。在这里你不是凭借你对别人的统治来建立地位和名望，也不是靠美貌，或拥有其他人想要的东西，而是靠你的奉献。尤其是奉献你的时间，你的创造力和你的技术成果。

要想获得黑客的尊重，你基本上有 5 件事情可干：

编写开源软件

首先（也是最传统和最重要的）是写一些其他黑客觉得有趣或有用的程序，并且开放源代码。

（我们曾经把这些程序称为“自由软件（free software）”，但是太多人不能确定这里的“free”是什么意思。现在我们通常使用“开源”软件这个词。

黑客间最受尊敬的圣人是那些编写了大型的，功能强劲且满足了广泛需求的开源软件供他人使用的人。

但是这里有段有趣的历史。虽然黑客一直敬重开源软件开发者，并且他们是我们社区的核心，但是直到 1990 年代中期，绝大部分黑客绝大多数时间是在闭源软件上工作的。在我 1996 年写本文的第一版的时候仍然如此。到 1997 年之后开源软件逐渐成为主流并改变了这一点。现在，“黑客社区”和“开源软件开发者”本质上是对同一文化和同一人群的两种表述——但值得记住的是，曾经不是如此。（想了解更多，请看“历史记录：黑客活动，开源，和自由软件”。）

帮助测试和调试开源软件

黑客也尊敬那些为开源软件进行测试和除错的人。在这个并非完美的世界上，我们不可避免地要花大多数的开发时间在调试阶段。 这就是为什么许多开源软件作者都会高度评价那些好的 beta 测试员（知道如何清楚描述出错症状，很好地定位错误，能忍受快速发布中的 bug，并且愿意使用一些简单的诊断工具），认为他们像红宝石一样珍贵。一个好的测试员可以使如恶梦的测试及除错工作变为一件值得经历的小烦恼。

如果你是个新手，试着找一个你感兴趣的正在开发的程序，尝试做一个好的 beta 测试员。 你会自然地从帮着测试，进步到帮着抓 bug，到最后帮着改程序。你会从中学到很多，并且与未来会帮到你的人结下友谊。

发布有用的信息

另一个好事是收集整理有用有趣的信息做成网页或文档如 FAQ（常见问题）列表，且让他们容易获得。

技术性 FAQ 的维护者往往如同开源软件作者一样很受人尊重。

帮助维护基础设施运转

黑客社区（也包括互联网发展）是靠自愿者组成的。有大量重要但平淡的事情需要处理——管理邮件列表，新闻组，维护大型软件归档库，开发 RFC 和其他技术标准等。

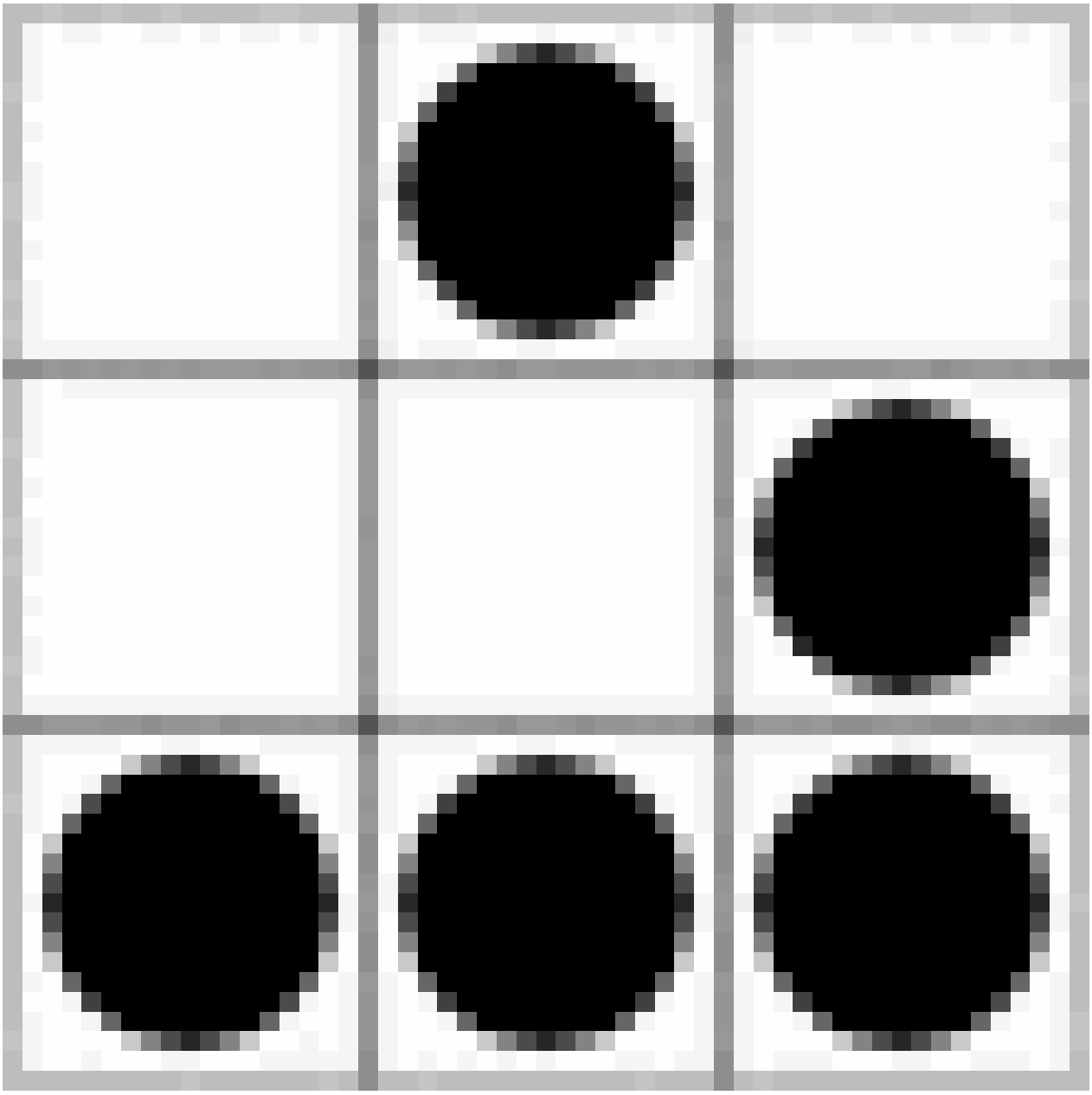
做以上事情的人会得到很多人的尊敬，因为大家都知道这些事情需要大量的时间并且不如编写软件那么有趣。这类工作需要使命感。

为黑客社区服务

最后，你还可以为黑客社区做服务和宣扬（比如写一篇“如何成为黑客”的文章 :-)）。通常你不会做这些工作，直到你已经做了以上四种中的一样，并且取得了相当的知名度。

黑客社区没有既定的领导者，但是有被人们尊重的英雄，长老级人物，史学家和发言人。当你在这个圈里足够久，你可能会成为他们中的一员。但请谨记，黑客对于自我夸耀的长老并不认同，因此不要尝试

大举追求这种名誉。与其奋力追求，不如先摆正自己的位置，等它自己到你手中，那时需要做到谦虚和优雅。



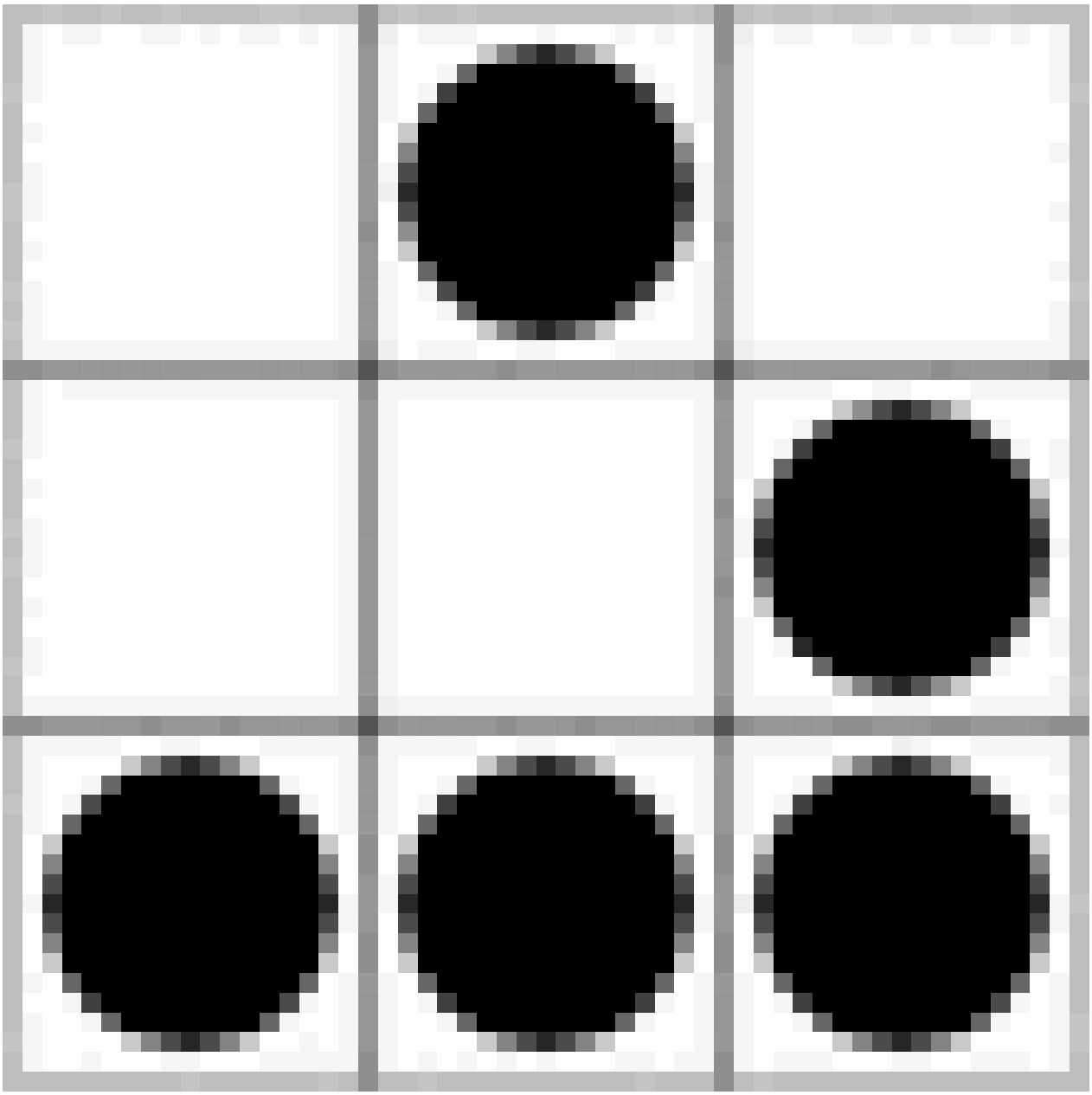
黑客与书呆子（Nerd）的关系

与流行的传说不同，黑客并不是书呆子。但这确实对你成为黑客有帮助，并且很多黑客确实是书呆子。做一个深居简出的人有助于使你更能集中精力做一些重要的事，例如思考和从事黑客活动。

因此，许多黑客甚至以“极客（geek）”（译注：这个词原本在美国俚语中指“反常的人”）为名——这是一种宣布他们独立于普通社会的方式（此外，黑客也通常沉迷于其他一些事情例如科幻和战略游戏）。“书呆子”这个词通常在1990年代也被如此使用，那时候“书呆子”这个词略含贬义，而“极客”贬义更重。2000年以后这两个词的关系发生了转变，至少在美国流行文化上是如此，现在甚至在非技术专家中也出现了以标榜为极客为豪的情况。

如果你能集中足够的精力做好黑客工作同时还能有正常的生活，这很好。现在做到这一点比我在1970年代还是新手的时候要容易的多；如今主流文化对技术怪人要友善的多。甚至有越来越多的人意识到黑客通常是很好的恋人和配偶侯选。

如果你因为生活上的不如意而成为黑客，那也不错——至少你不用分神了。或许今后你能有一个不错的生活。



风格

重申一下，作为一名黑客，你必须进入黑客式思维模式。当你不在电脑边的时候你仍然有很多有益的事情可做。它们不能替代真正的黑客活动（没有什么可以），但是很多黑客都这么干，并且感到它们与黑客精神存在某些根本的联系。

- 学会流畅的使用母语写作。虽然认为程序员写不好文章的误解仍然很普遍，但是有数量令人惊讶的黑客（包括我所知造诣最高的那些）都是不错的写手。
- 阅读科幻小说。参加科幻聚会（一个接触黑客和可能成为黑客的人的好方法）。

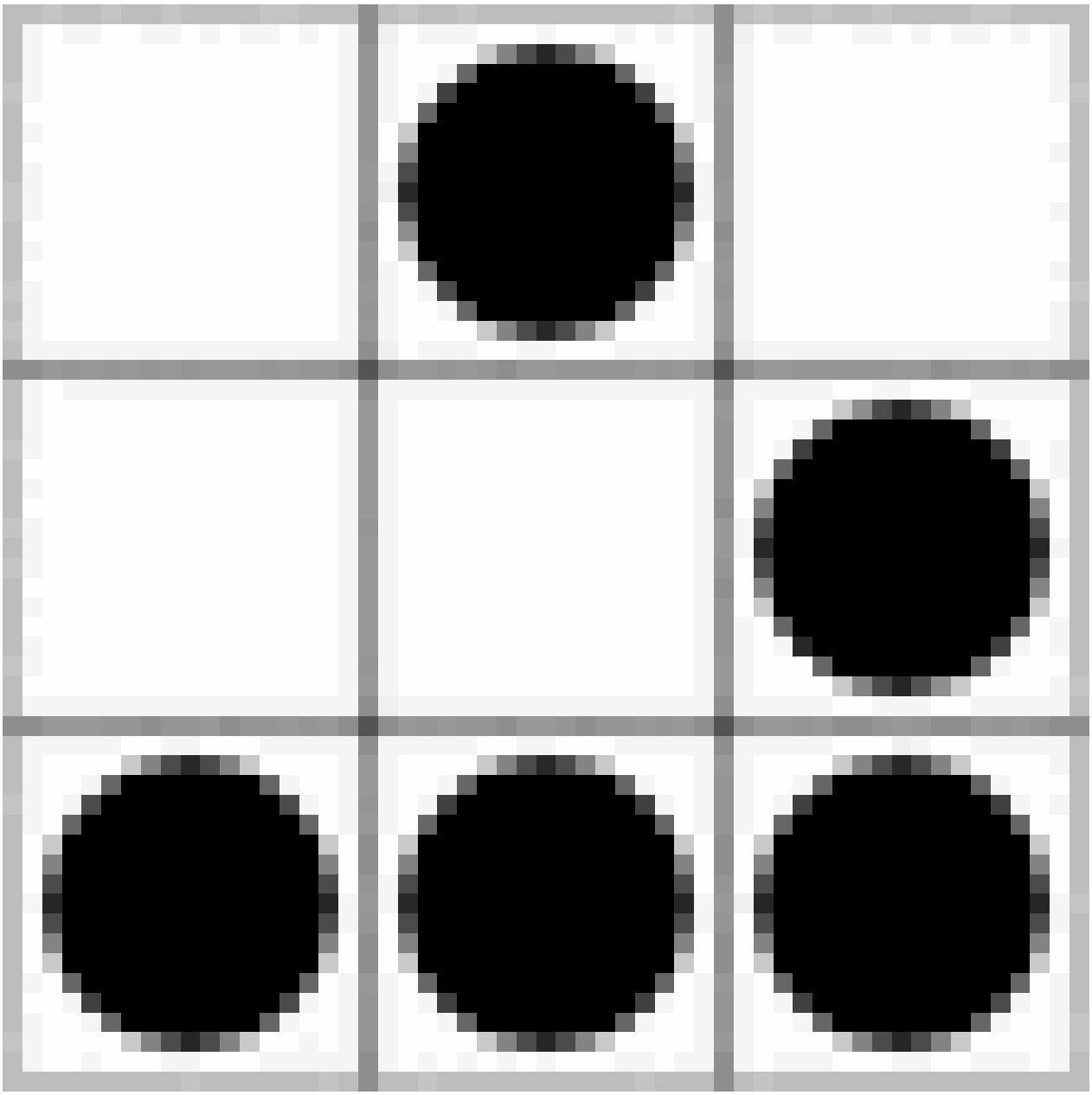
- 加入黑客空间（hackerspace）并做一些东西出来（另外一个接触黑客和可能成为黑客的人的好方法）。
- 习武。武术的精神修炼与黑客之道惊人的相似。黑客中比较常见的当然是亚洲的空手格斗技巧，例如跆拳道、空手道及其变种、中国功夫、合气道、柔术（译注：这里指的是日本传统武术，而不是柔身术或软功）。西方击剑和亚洲剑术也有相当的追随者。在持枪合法的地区，射击从 1990 年代起也越来越受欢迎。与黑客之道最契合的武术是那些强调精神修炼、放松意识，强调控制而不是单纯的蛮力的类型。
- 学习一种冥想修炼。黑客中一直以来最受欢迎的是禅（很重要的是学禅并不要求你有特定的宗教信仰）（译注：这里指的是日本禅宗，而不是汉地佛教禅宗）。其他方式也可以，但是请注意一定选择那些不会要求你相信很疯狂东西的方式。
- 修习音乐。学会鉴赏特别的音乐。学会玩某种乐器，或唱歌。
- 提高对双关语、文字游戏的鉴赏能力。

这些事情，你已经在做的越多，你就越是天生做黑客的料。至于为什么偏偏是这些事情，原因并不完全清楚，但它们都涉及用到左右脑混合使用，这似乎是关键所在。黑客们既需要清晰的逻辑思维，有时又需要偏离逻辑跳出问题的表象。

工作即娱乐，娱乐即工作。对于真正的黑客来说，“玩”，“工作”，“科学”和“艺术”之间没有界线，或者说，它们在一个高层面的创造性趣味里融合在一起。另外，不要对一点点技能就感到满足。虽然大多数黑客自称是程序员，他们实际上在其他相关的方面也很可能相当强悍——常见的是系统管理、页面设计和 PC 硬件故障处理。一个黑客，如果他是一名系统管理员，他很可能对脚本编程和页面设计也相当在行。黑客不会半途而废，如果他们要学习一门技能，他们会将其学好。

最后，一些你 **不应** 做的事。

- 不要使用愚蠢，哗众取宠的 ID 或昵称。
- 不要卷入 Usenet（或其他任何地方）的骂战。
- 不要自称为“赛博朋克（cyberpunk）”，也不要浪费时间跟他们打交道。
- 不要发送含有大量拼写和语法错误的 email 和帖子。
- 做出以上事情只会招来嘲笑。黑客的记性都很好——你犯下的错误会令你将要经过多年才可以被其他黑客接受。
- 网名的问题值得深思。将身份隐藏在虚假的名字后是骇客、warez d00dz 及其他低等生物幼稚愚蠢的行为特点。黑客不会做这些事；他们对他们所作的感到骄傲，而且乐于人们将作品与他们的 **真名** 相联系。 因此，若你现在用假名，放弃它。黑客社区里只会将用假名的人视为失败者。



历史记录：黑客活动，开源，和自由软件

当我在 1996 年末刚开始写这篇文档的时候，很多情况跟现在是不同的。简单的介绍一下这个变化对于对开放源代码、自由软件和 Linux 跟黑客社区的关系感到困惑的人们可能会有所帮助。如果你对这些不感兴趣，可以直接跳过这里，前往 FAQ 和参考资料部分。

我描述的黑客精神和黑客社区远早于 1990 年出现的 Linux。我最初进入这个圈子大概是在 1976 年，其原因可以追溯到 1960 年代早期。但是在 Linux 出现前，多数黑客行为是在专有操作系统，或一些自主研发的实验性系统上，例如 MIT 的 ITS，这个系统从未在实验室以外的地方使用过。虽然在早期（Linux 出现之前）有过一些试图改变这种状况的努力，但是它们的影响都非常轻微，仅限于真正怀抱这样理想的人群，即使在当时的黑客社区这也是绝对少数，更不论对于世界范围内的通用软件群体的影响了。

现在被称为“开放源代码”的行为，其历史与黑客社区一样久远，但是直到 1985 年这都只是一个无名的民间行为，没有相关的理论和宣言。这段史前时代在 1985 年结束，大黑客 Richard Stallman（“RMS”）尝试给了它一个名字——“自由软件（Free Software）”。但是这个命名行为也是一个强制要求行为，他

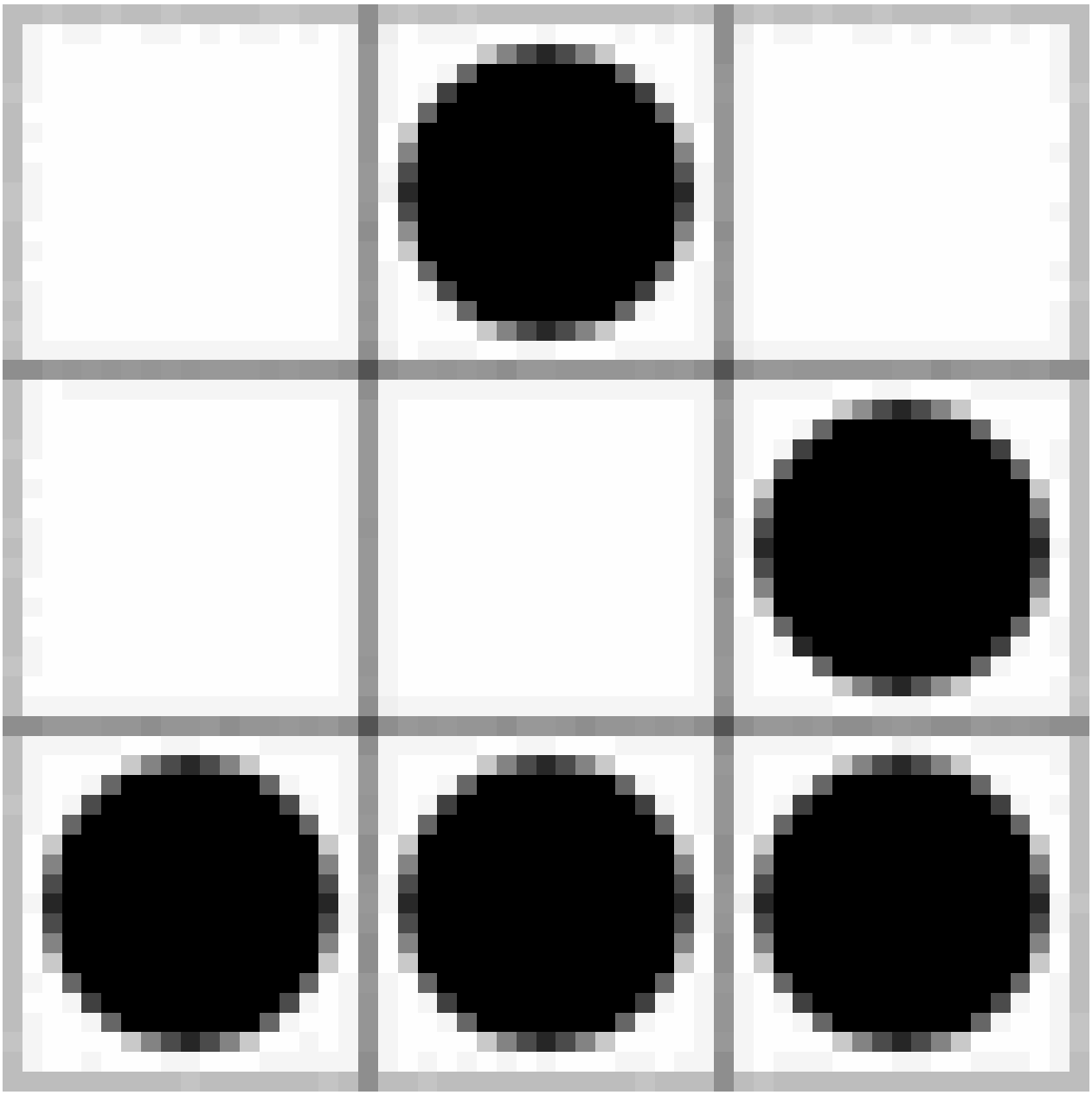
为“自由软件”标签加上了大多数已有的黑客社区从不接受的意识形态的包袱。“自由软件”的标签被黑客社区中的一部分重要人物（尤其是与 BSD Unix 有关联的社区）明确拒绝，并且其余的大部分人也在严肃并且持保留意见的情况下使用它（包括我本人）。

除此之外，大约在 1990 年代中期以前，RMS 想要在“自由软件”口号下定义和引领黑客社区。在 Linux 崛起之后，这受到了极大挑战。Linux 为开放源代码开发活动提供了一个天然的环境。许多在现今被称为“开放源代码”条款下发布的项目纷纷从专有 Unix 向 Linux 迁移。围绕 Linux 的社区呈现爆炸式的增长，比在 Linux 出现前的黑客文化规模更大且更多样化。RMS 想要将这些活动与他的“自由软件”运动关联起来，但是由于 Linux 社区爆炸式的多样性和该社区的创始人，Linus Torvalds 的公开怀疑所阻碍。Torvalds 仍然使用“自由软件”这一词汇，因为找不到更好的替代品，但他公开拒绝了 RMS 的意识形态观念。许多年轻黑客纷纷效仿。

在 1996 年，当我第一次发布本文的时候，黑客社区正在围绕 Linux 和一些其他开放源代码的操作系统（尤其是 BSD Unix 的继承者）进行重组。我们中的许多人曾经在封闭源代码的操作系统上花费大量时间开发封闭源代码软件的集体记忆并没有因此褪色，但是这看起来已经是过去。黑客们将自己作为黑客的定义与开发源代码项目例如 Linux 和 Apache 越来越紧密的结合在一起。

然而“开放源代码”这个词直到 1998 年初才出现。当它出现之后，多数黑客社区在 6 个月之内采用了它，除了与“自由软件”在意识形态层面绑定的极少数例外。自 1998 年起，尤其是 2003 年之后，“黑客”和“开放源代码（和自由软件）开发”越来越紧密相连。今天，几乎已经无法也没必要将它们区分开，并且这一点看起来在将来也不会改变。

然而，曾经并不是这样，这一点值得我们记住。



其他资源

Paul Graham 写了一篇名为“Great Hackers”和一篇名为“Undergraduation”的文章，有很多精妙的见解。

年轻的黑客可能会发现 Things Every Hacker Once Knew 有趣并且有用。

有一篇名为 How To Be A Programmer 的文章，对如何成为程序员做了详尽的说明。它的价值不仅限于代码及其相关能力上，对于如何在一个程序员团队中工作也有建设性价值。

我写过一篇 A Brief History Of Hackerdom。

我写过一篇名为“大教堂与市集（The Cathedral and the Bazaar）”的文章，在文中解释了 Linux 和开源社区是如何运作的。在它的续集“开拓智域（Homesteading the Noosphere）”中，我进一步深入探讨了这个问题。

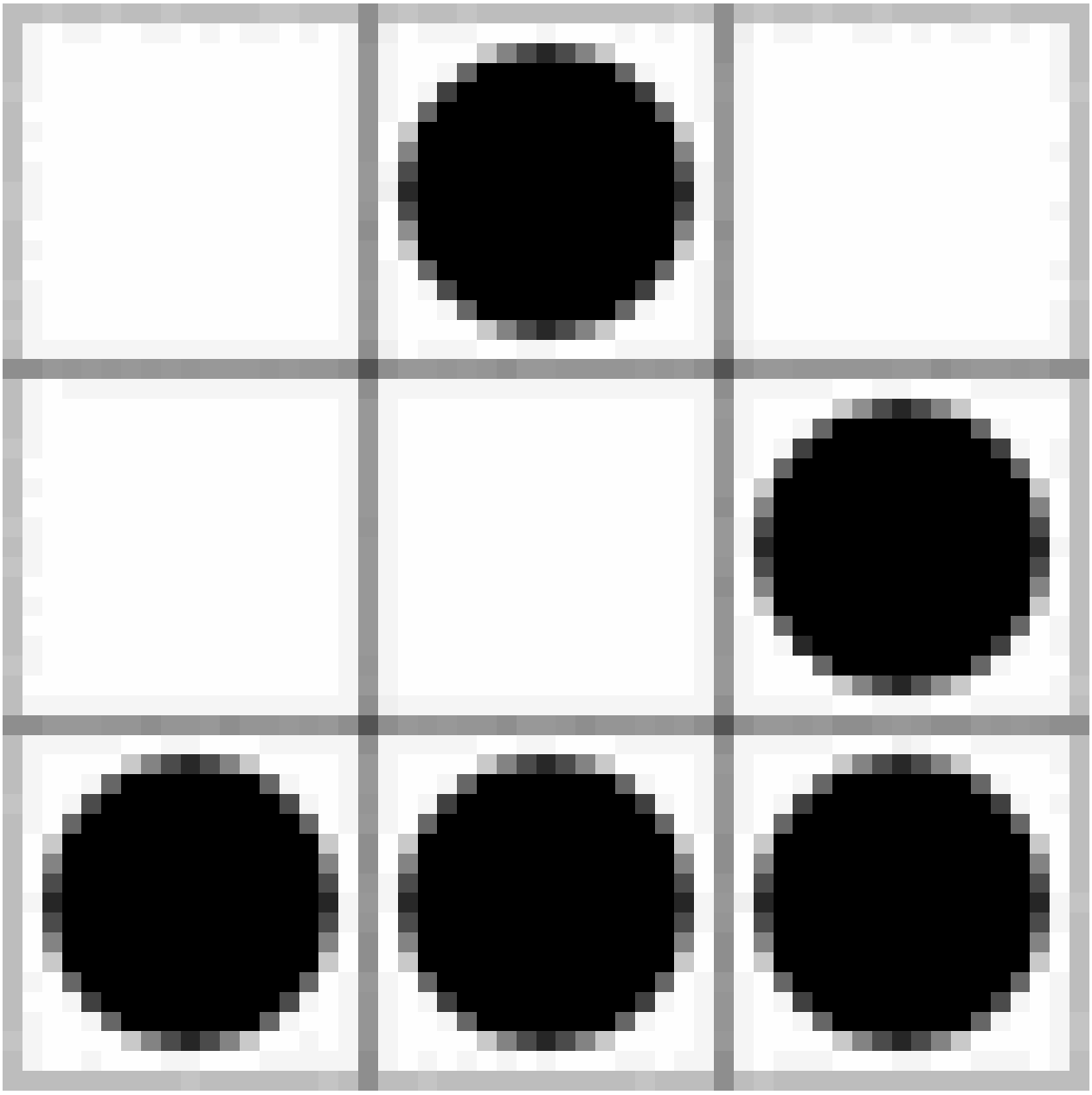
Rick Moen 写了一篇很棒的关于如何运营一个 Linux 用户组的文章。

Rick Moen 和我共著了一篇关于提问的智慧的文章。它将使你更容易获取到帮助。

如果需要个人电脑、Unix 和互联网如何工作的基础知识，参阅 Unix 和互联网工作的基本原理。

当你发布软件或补丁时，试着按软件发布实践操作。

如果你对禅诗感兴趣，你可能会喜欢 Rootless Root: The Unix Koans of Master Foo。



常见问题

Q: 如何证明我已经是一名黑客了?

Q: 你能教我做黑客吗?

Q: 我该如何开始?

Q: 我该什么时候开始学? 现在会不会太迟了?

Q: 学会黑客之道要多长时间?

Q: Visual Basic 是一门好的入门语言吗?

Q: 你能教我“黑”掉一个网站，或者教我怎么黑它吗?

Q: 我怎么样才能得到别人帐号的密码?
Q: 我如何入侵/查看/监视别人的 email?
Q: 我如何才能盗取 IRC 的频道管理员权限?
Q: 我被人入侵了。你能帮我避免以后再被攻击吗?
Q: 我的 Windows 软件出现问题了。你能帮我吗?
Q: 我在哪里可以找到能与之交流的真正的黑客?
Q: 你能推荐一些有关黑客的好书吗?
Q: 成为一名黑客我需要擅长数学吗?
Q: 我该从哪种语言开始学?
Q: 我需要什么样的机器配置?
Q: 我想做贡献。你能帮我挑选一个问题来处理吗?
Q: 我得因此憎恨和反对 Microsoft 吗?
Q: 但开放源代码软件不会使程序员丢饭碗吗?
Q: 哪里有免费的 Unix?

Q: 如何证明我已经是一名黑客了?

A: 问自己以下三个问题:

* 你能够流畅的编写代码吗?
* 你是否与黑客社区的目标和价值观产生共鸣?
* 是否有知名黑客称你为黑客?

如果你对三个问题的回答都是肯定的，你就已经是一名黑客。哪怕只有一个回答是否定的也不行。

第一个问题是关于技能的。如果你具备了前文提到的基本技能，就应该没问题。如果你已经有相当数量的代码被开源项目所接受，可以跳过这个问题。

第二个问题是关于精神的。如果前面的五条黑客的精神明显比其他地方的描述更贴近你的真实生活，你就已经通过了一半。这是内在的一半，外在的一半是你对黑客社区长期项目的贡献程度。

这里有一个不完整但具有指示性的项目列表：Linux 的改进和发展是否有你的贡献？你是否对软件自由充满激情？你是否反对垄断？你是否为了让计算机成为这个世界更丰富多彩、更人性化的工具而努力？

请注意。黑客社区有一些特定的，主要是防御性的政治倾向——其中两条是维护言论自由和抵御可能使开放源代码非法的“知识产权”。有一些长期项目是公民自由组织，例如电子前哨基金会（EFF），外在的态度包括支持他们。除此之外，大多数企图将黑客精神系统化为一个具体的政治程序的黑客都值得怀疑。我们曾经为此付出过代价，并了解到这些想法通常会导致分裂并且令人心烦意乱。如果有人想要以黑客精神的名义来招揽你，那是他们搞错了重点。最好的回应恐怕是“闭嘴，给他们看代码（Shut up and show them the code）。”

第三个问题是递归的。我在“什么是黑客”中提到过，成为一名黑客，就是成为一个具有分享特点的文化社群的一员。很久之前，黑客相比现在是一个松散的，不自知的群体。但是近 30 年来，由于互联网使得黑客文化的核心更易发展和维护，人际网络方面取得了长足进步。这种改变最简单的代表就是，在这个世纪，我们有了自己的 T 恤。

社会学家在研究类似黑客社区这样的被统称为“无形学院（invisible colleges）”的人际网络时注意到，这类网络通常都有门卫——具有社区授权的核心成员会审核新人的进入申请。由于黑客社区是“无形学院”中比较松散和非正式的一种，门卫的身份也是非正式的。但是所有黑客天生就知道并非每个黑客都是门卫。在取得门卫的头衔前需要具备特定的资历和成就。这很难度量，但是每个黑客在看到它的时候就能知道。

Q: 你能教我做黑客吗?

A: 自从第一次发布这份文档，我每周都会收到一些请求，（通常一天几封）要我“教会他们做黑客”。遗憾的是，我没有时间和精力来做这个；我自己的黑客项目，及我作为一个开放源代码倡导者 的四处奔波已经占用了我 110% 的时间。

即便我想教你，黑客也依然基本上是一项自行修炼的的态度和技术。当真正的黑客想帮助你的时候，如果你乞求他们一汤匙一汤匙“喂”你的话，你会发现他们不会尊重你。

先去学一些东西。显示你在尝试，你能靠自己去学习。然后再去向你遇到的黑客请教特殊的问题。

如果你发 E-mail 给一位黑客寻求他的帮助，有两件首要记住的事情。第一，写出来的文字显得懒且粗心的人通常非常懒于思考且非常马大哈，不能成为好黑客——因此注意拼写正确，使用正确的语法及发音，否则你可能会无人理睬。第二，不要试图要求回复到与你的发信地址不同的另一个帐号。这样做的人一般是使用盗用帐号，不会有人有兴趣为虎作伥帮助窃贼的。

Q: 我该如何开始

A: 对你而言最佳的入门方式也许是去参加 LUG（Linux 用户组）的聚会。你可以在 LDP 的综合 Linux 信息页面上找到类似的组织；也许有一个在你附近的，而且非常有可能与一所大学或学校挂钩。如果你提出要求，LUG 成员兴许会给你一套 Linux，当然此后会帮你安装并带你入门。

下一步（如果你在附近找不到 LUG 的话，这是第一步）找一个你感兴趣的开放源代码项目。读它的代码，并且检查它的 bug。学着做贡献，并从此入门。

入门的唯一方式是提升你的技能。如果你还想问我关于如何开始的私人建议，我还是会给你相同的答案，因为没有捷径。我还会在心里认为你可能是个失败者——因为你没有耐性读完这个 FAQ，并且也没有足够的智商从文中理解到入门的唯一途径就是提升你的技能。你没救了。

Q: 我该什么时候开始学？现在会不会太迟了？

A: 你有动力学的时候就可以。多数人是在 15 到 20 岁之间开始感兴趣的，但据我所知也有在这个年龄区间之外的例外。

Q: 学会黑客之道要多长时间？

A: 这取决于你的聪明程度和努力程度。多数人如果足够专注的话，能在 18 个月到 2 年之间学会一套令人尊敬的技能。但是，不要以为这就结束了。在黑客领域（在其他很多领域也一样），需要 10 年时间精湛技艺。如果你是一个真正的黑客，你要用你的余生来学习和完善你的技术。

Q: Visual Basic 是一门好的入门语言吗？

A: 你问这个问题，那通常意味着你想在 Microsoft Windows 下从事黑客活动。这本身就不是个好主意。我将在 Windows 下学习黑客技巧比喻为戴着脚镣学跳舞，这不是开玩笑。别这么做。Windows 很糟糕，而且它从来没有变好一点。

Visual Basic 有一个很重要的问题，主要是源于它不可移植。虽然已经有 Visual Basic 的开源实现，但 ECMA 的可执行标准只覆盖了其编程接口的很小一部分。在 Windows 中，大多数函数库是由单一供应商（Microsoft）专有的。如果你不能非常小心的选择你所使用的特性（比任何新手所能做到的都更小心），你最终很可能被束缚在 Microsoft 决定支持的那些平台上。如果你从 Unix 系统开始，有更好的编程语言和更好的函数库可用。例如 Python。

此外，如同其他 Basic 语言一样，Visual Basic 是一种设计糟糕的语言，它会教给你坏的编程习惯。别让我详细列举和解释它们，这够写一本书了。找一门设计优良的编程语言来学。

坏习惯之一就是依赖单一厂商提供的函数库、控件和开发工具。通常，一门不能支持至少 Linux 或一种 BSD，或其他第三方操作系统的语言，都不适合应付黑客工作。

Q: 你能帮我“黑”掉一个网站，或者教我怎么黑它吗？

A: No。任何读完这份 FAQ 后还问这个问题的人都是无药可救的蠢材，即使有时间我也不会理睬。任何发给我的此类 e-mail 都会被忽略掉或被痛骂一顿。

Q: 我怎么样才能得到别人帐号的密码?

A: 这是破坏行为。滚开，白痴。

Q: 我如何入侵/查看/监视别人的 email?

A: 这是破坏行为。从我面前消失，混蛋。

Q: 我如何才能盗取 IRC 的频道管理员权限?

A: 这是破坏行为。去死，蠢货。

Q: 我被人入侵了。你能帮我避免以后再被攻击吗?

A: 不能。每次问我这个问题的都是些运行 Microsoft Windows 的菜鸟。不可能有效保护 Windows 系统免受破坏行为攻击。代码和架构上的大量缺陷使保护 Windows 的努力犹如隔靴搔痒。唯一可靠的预防是换到 Linux 或者其他设计得至少足够安全的系统上。

Q: 我的 Windows 软件出现问题了。你能帮我吗?

A: 是的。进入 DOS 模式，然后输入“format c:”。你遇到的问题将在几分钟内消失。

Q: 我在哪里可以找到能与之交流的真正的黑客?

A: 最佳办法是在你附近找一个 Unix 或 Linux 的用户组，参加他们的聚会（你可以在 ibiblio 的 LDP 站点找到一些指向用户组的链接）。

我过去曾说过在 IRC 上找不到真正的黑客，但我发觉现在情况有所改变。显然一些真正的黑客的社区像 GIMP 及 Perl，也有 IRC 频道了。）

Q: 你能推荐一些有关黑客的好书吗?

A: 我维护着一份 Linux Reading List HOWTO，也许会对你有用。Loginataka 也很有意思。

关于 Python 的介绍，请访问 Python 官方站点上的入门资料

Q: 成为一名黑客我需要擅长数学吗?

A: 不。黑客倒很少使用常规的数学或算术。尤其是你不会用到三角学、微积分或数学分析（在特定领域，这些学科很有用，例如 3D 电脑图像）。一些有限数学（包括布尔代数，集合论，组合数学，图论）的背景知识会有帮助。了解一些系统的逻辑和布尔代数是有好处的。一些基础的离散数学（包括有限集合论、组合数学和图论）会有帮助。

更重要的是：你需要能够像数学家一样进行逻辑性地思考和进行缜密的推理。在这一点上绝大部分数学理论帮不了你，你需要有能够应付数学的修养和智力。如果你不够聪明，你成为黑客的希望很渺茫。如果你的修养不够，最好培养起来。

我认为一个好的了解你当前状况的办法是，找一本 Rymond Smullyan 的书 What Is The Name Of This Book?。Smullyan 那些有趣的逻辑题很符合黑客精神。如果你能解答它们，这是个很好的信号。如果你能享受解题的过程那就更好了。

Q: 我该从哪种语言开始学?

A: XHTML（最新的 HTML 方言）——如果你还不懂的话。市面上有一大堆的封面精美，宣传得天花乱坠的 糟糕的 HTML 书籍，不幸的是很少有好的。我最喜欢的是 HTML: The Definitive Guide。

但 HTML 不完全是一种编程语言。当你准备开始编程时，我推荐从 Python 起步。 你会听到一大群人推荐 Perl，并且 Perl 依然比 Python 流行得多，但是难学得多且（以我之见）设计得不是很好。

C 确实重要，但它要比 Python 或 Perl 难多了。不要尝试先学 C。

Windows 用户不要满足于 Visual Basic。它会教给你坏习惯，而且它不可以移植，只能在 Windows 下运行。避免它。

Q: 我需要什么样的机器配置?

A: 过去个人电脑计算能力相当不够且内存小，给黑客的学习过程设置了人为的障碍。不过从 1990 年代以后就不是这样了，任何配置比一台 Intel 486DX50 好的机器都有足够的能力进行开发工作，跑 Xorg，及进行 Internet 通讯，同时现在能买到的最小的磁盘都已经绰绰有余。

选择用来学习的机器时重要的一点是注意配件是否是 Linux 兼容的（或 BSD 兼容，如果你选择学 BSD）。同刚才提到的一样，大多数现在的机器都是符合的；唯一的值得注意的地方在于 调制解调器和打印机；有些具备为 Windows 设计的配件的机器不会在 Linux 下工作。

关于硬件兼容性有一个 FAQ；最新版本在这里。

Q: 我想做贡献。你能帮我挑选一个问题来处理吗?

A: 不行。因为我不知道你擅长什么，也不知道你对什么感兴趣。你需要做到自我驱动，否则无法进步。这也是为什么让别人帮你挑选方向几乎都不会有用。

Q: 我得因此憎恨和反对 Microsoft 吗?

A: 不，你不必如此。不是因为 Microsoft 不令人讨厌，而是因为黑客文化早在 Microsoft 出现之前就存在了，且将在 Microsoft 成为历史后依然存在。你耗费在憎恨 Microsoft 的任何力气不如花在热爱你的技术上。写好的代码——那会相当有效地打击 Microsoft 又不会让你得到恶报。

Q: 但开放源代码软件不会使程序员丢饭碗吗?

A: 看起来不太可能——目前为止，开放源代码软件产业似乎创造了更多的就业机会而不是减少就业机会。如果写一个程序比起不写来是纯经济收益的话，那么在写完后，程序员应该得到报酬不管程序是否是开放源代码。并且，无论写出多么“免费自由”的软件，都存在更多对新的，定制的软件的需求。我有这方面更多的论述，放在 Open Source 网站资料中。

Q: 哪里有免费的 Unix?

A: 如果你的机器上还没有安装 Unix，我在本文的其他地方已经指出了从哪里可以获取到常用的免费 Unix。作为一名黑客，你需要自立自强，以及自学能力。现在开始吧……

作者 · 许可证 CC BY-NC-SA 4.0