

Free as in Freedom (2.0): 理查德·斯托曼 与自由软件运动

萨姆·威廉姆斯
第二版：理查德·斯托曼修订版

本书是《*Free as in Freedom 2.0: 理查德·斯托曼与自由软件运动*》，这是《*Free as in Freedom: 理查德·斯托曼的自由软件的运动*》一书的修订本。

Copyright © 2002, 2010 Sam Williams

Copyright © 2010 Richard M. Stallman

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License.”

封面上理查德·斯托曼的照片由 Peter Hinely 提供。第 7 章中 PDP-10 照片由 Rodney Brooks 提供。第 8 章中圣·IGNUcius 的照片由 Stian Eikeland 提供。

目录

Foreword by Richard M. Stallman	v
Preface by Sam Williams 萨姆·威廉姆斯的前言	ix
1 For Want of a Printer 从一台打印机说起	1
2 2001: A Hacker's Odyssey 黑客路漫漫	25
3 A Portrait of the Hacker as a Young Man 黑客正年少	49
4 Impeach God 逆天行道，选举上帝	75
5 Puddle of Freedom 自由一隅	123
6 The Emacs Commune Emacs 公社	163
7 A Stark Moral Choice 道分左右义无旁支	191
8 St. Ignucius 圣·Ignucius	237

9	The GNU General Public License GNU 通用公共许可证	267
10	GNU/Linux GNU/Linux	311
11	Open Source 开源	339
12	A Brief Journey through Hacker Hell 开往黑客地狱的短暂旅途	371
13	Continuing the Fight 斗争仍在继续	383
	Epilogue from Sam Williams: Crushing Loneliness 萨姆·威廉姆斯跋语：粉碎孤独	407
	Appendix A – Hack, Hackers, and Hacking 附录 A – 黑客的三层含义	445
	Appendix B – GNU Free Documentation License	461

理查德·斯托曼的前言

萨姆·威廉姆斯的前言

萨姆·威廉姆斯
于美国史泰登岛

Chapter 1

从一台打印机说起

我畏惧希腊人，哪怕他们带着礼物来。
——维吉尔，《埃涅阿斯纪》

这可是台全新的打印机！怎么又卡纸了？

理查德·M·斯托曼（Richard M. Stallman）刚刚发现了这令他头痛不已的卡纸问题，那年他 27 岁，是麻省理工学院人工智能实验室一名资深的程序员。一个小时前，他给那台激光打印机提交了一个 50 页的打印任务，之后，他就忙于其它事务去了。回来时，本想拿到一叠散发着油香的文件，可这会儿，他只能站在那台毫无生机打印机前，直楞楞地看着它，上面，总共才四页并非他想要的文档。换句话说，斯托曼那份 50 页的文件，外加别人的半份文档，全被卡住了，成了实验室网络上的两个孤魂野鬼。

干程序员这行也是有风险的：你总得等着机器干活。不过斯托曼可是个能苦中做乐的人。再怎么讲，等着机器干活与盯着机器干活，毕竟是两码事。就说现在吧，斯托曼正盯着打印机，看它慢吞吞地一页一页往外吐纸。这情景对他来说可一点也不陌生。作为一个程序员，他

成天都在改进机器和控制这些机器的程序，好让它们更有效率的工作。当下，他恨不得掀开这台打印机，仔细查究究竟是哪出的毛病。

可惜，凭编程的能耐，没法解决机械工程的问题。斯托曼只能在重新打印文件的这会功夫，想点别的：能不能用别的办法，绕道解决这个卡纸的问题呢？

这台打印机来人工智能实验室有多久了？斯托曼努力的回忆着。这机器是施乐公司（Xerox Corporation）捐赠的。它可是新一代产品的原型，是从上一代的复印机改进而来。和上一代比，它不再是简单地复印，而是从网络上接收数据，打印成专业品质的文档。它诞生于著名的施乐帕罗奥多研究中心（Xerox Palo Alto Research Facility）。再过大约十年，将会发生一场打印机革命，那时，许多计算机厂商都会投身其中。而这款打印机则是这场革命的先行者。

人工智能实验室的程序员们，本能般地喜欢折腾这些新酷产品。当初他们麻利地把这打印机请回来，把它和实验室中的各种复杂的计算设备连在了一起，结果甚是喜人。比起以前的打印机，这台新打印机打印速度飞快。平均大约一秒钟打印一页纸。以前要花二十来分钟打印的东西，如今两分钟就解决了。新机器的打印精度也更高了。让它给画个圆，它决不会给你打出个鸡蛋来；让它划条直线，它也绝不会给你拐着弯瞎划拉。

怎么看，这打印机都是份难以拒绝的珍贵礼物。

可没过几周，这机器的缺点就逐渐显现了。其中最要命的，就是卡纸问题。凭着工程师的直觉，程序员们很快就洞察出了这问题背后的原因。倘若是一台复印机，它旁边总会站个人直接操作它。遇到卡纸问题，这个人总能立即发现，着手解决，不至于有什么大影响。而施乐公司的那些工程师们则可以把时间和精力，用在其他更棘手的问题上了。用工程师的行话讲：这系统需要用户参与。

如今，施乐公司在复印机的基础之上，设计了这个打印机。这个小小的变化，却使得人机关系发生了微妙而深刻的变化。现在，在同一时刻，这机器不再只服务于一个用户。它要同时给整个网络的用户

提供打印服务。这些用户也不再老老实实在机器前面，而是坐在网络的另一端，可能还远离打印机，向这台机器发布打印命令，并希望它能按要求完成任务。等这些用户忙过其它事情回过神来，想起打印的文件时，他们才发现，大部分的内容已被一张纸卡住了。

斯托曼是人工智能实验室中第一个发现问题的人，也是他第一个提出解决方案。几年前，实验室还用着一台旧打印机，也有类似的问题。这台打印机的控制程序，运行在实验室里的一台 PDP-11 计算机上。那会儿，斯托曼修改了这个控制程序，解决了这个问题。当然，他没法直接修理打印机来解决卡纸的问题。他只是修改控制程序，让程序定期检查打印机是否工作正常，再把检查结果传给实验室的核心计算机——一台 PDP-10。同时，他也修改了运行在 PDP-10 主机上的“不兼容分时系统 (ITS, Incompatible Timesharing System)”，处理这些检查结果。生怕哪个粗心用户忘了去查看打印机，斯托曼还让控制程序在卡纸的时候，向所有等待打印任务的用户发送一条提醒消息。消息简短明确：“打印机被卡住了，请前去维修。”收到这消息的人都是等着要用打印机的，所以问题就迎“人”而解。

斯托曼这解决方案虽然并不能说很讨巧，但却也漂亮。它没解决打印机机械部件的问题，却提出了一个次优方案：完善人机交互，及早报告问题。多亏了斯托曼加的几行代码，人工智能实验室的员工，不用再跑来跑去地查看打印机状态了。这样，每周起码节省了 10 到 15 分钟时间。用编程的术语讲，斯托曼的方案提高了网络的整体智能。

“你要是收到这条消息，你可就坐不住了。你多半等不及别人去解决卡纸问题。”斯托曼回忆着其中的逻辑，“你得亲力亲为，跑到打印机跟前。一两分钟之后，要是问题还没解决，就会聚来两三个人。这其中，总会有个人知道怎么修理。”

类似的巧法子的人工智能实验室的一大特色。尤其在其中的资深程序员之间，屡见不鲜。实际上，那些天才的程序员，不屑于用“程序员”这词。他们更喜欢用圈子里的行话，用“黑客”这个词。这词儿涵盖了不少内容：从抖机灵、甩包袱，到改进现有软件，优化计算机系

统。而更深之处，则蕴含着旧时美国移民的智慧。对于黑客来说，有这么一条铁律：从头开发个软件只是小儿科；要想在别人面前展现自己的聪明才智（并让别人对自己留下印象），就得做点更有意思的事：改进一个程序让它体积更小、跑得更快、功能更强，代码更优雅，这才是真本事。¹

这条铁律，恰恰影响着施乐这样的公司，让它们乐意把自己的产品（和配套的软件）捐赠到黑客聚集的地方。要是这些黑客们对这些产品产生了兴趣，他们以后就可能有兴趣去这个公司工作。在 20 世纪 60 年代到 70 年代初，黑客们也常常会开发一些实用的软件，提供给公司发布给其他顾客使用。

凭着这份礼尚往来的精神，斯托曼在这台新打印机面前并不慌张。他只要找个法子，把以前那套修改方案拿过来，修理修理这机器，或者用圈子里的话说，“黑”它一下，就万事大吉了。可是，他刚找了找这台施乐激光打印机的软件，就碰了一鼻子灰。这打印机根本不提供软件，至少没提供任何给人读的程序代码。要知道，那个时代，大多数公司都会发布软件的源代码（一系列可供人阅读的文本文件，用于详细定义程序的行为）。可当下，施乐公司提供的程序，却只有编译好的可执行文件。程序员倒是可以打开这些文件，可打开之后看见的，全是零和一。除非他们乐意去翻译这些二进制信息，否则这些东西看起来只是一堆胡言乱语。

有一种叫作“反汇编器”的程序，可以把这些看不懂的零和一转换成机器底层指令，但是要进一步了解这些指令实际在“做”的事情则是一件费时费力的工作，这样的工作被称作“反向工程”。要想把打印机的控制程序进行反向工程，也许会花去超过五年的时间。卡纸的问题还没有严重到让斯托曼愿意花这么多时间去解决，所以他把这件事放到了一边。

施乐公司看上去不太友好的软件条款与黑客社区所习惯的氛围格格不入。打个比方，如果想为一台老式的打印机开发一个在 PDP-11 上运行的控制程序，或为另一台 PDP-11 开始一个处理显示终端的程序，

人工智能实验室的黑客们需要在 PDP-10 主机上运行一个交叉汇编的程序，才能创建出他们需要的程序。他们可以自己从头开发一个这样的交叉汇编程序，但是上哈佛大学上学的斯托曼，在哈佛的计算机实验室中找到了这样一个程序。这个程序都可以在 PDP-10 这种型号的计算机上运行，即使两台机器上运行的操作系统不同也没有关系。斯托曼并不需要知道这份代码是谁写的，因为源代码中并没有标明。但是他可以把这份代码带到人工智能实验室，通过修改其中的一部分代码，这个程序就可以在人工智能实验室的“不兼容分时系统 (ITS)”上运行了。没花多大劲，人工智能实验室就得到了一个可以用于充实他们的软件架构的程序。斯托曼甚至还在程序中加入了一些原来没有的新功能，使程序变得更加强大了。“这个程序我们一直用了很多年。”斯托曼说。

对于七十年代的程序员来说，把软件拷贝来拷贝去，就好比从邻居家借碗酱油那么稀松平常。所不同的是，你从邻居那拿了多少酱油，邻居就少了多少。而从别人那里拷贝一个程序出来，并不影响别人继续使用那个程序。要说有影响，也是正面的。因为斯托曼对这程序还加了些功能，别的黑客可以自由地拷来斯托曼改进的程序，这也算是计算机世界的礼尚往来了。后来，有个从 Bolt, Beranek & Newman 这家公司来的程序员，拷走了斯托曼的程序，他把这个程序移植到了 Twenex 系统上，并加上了一些新的功能，斯托曼则又把改进后的程序拷了回来，用回到人工智能实验室里。他们两人打算共同来维护一个通用的版本，这个版本既可以在 ITS 上运行，也可以在 Twenex 上运行。

“开发一个程序就好比开发一座城市。”斯托曼回忆着人工智能实验室的软件设施：“有些部分会被换掉，有些得重建翻修。新的东西会逐步加进来。可如果你仔细看某个部分，也许会感慨道：‘这块从风格看应该是六十年代早期的；之后的部分估计是七十年代中期完成的。’”

这套简洁的系统得以让知识逐渐积累。这种积累为发明和创造提供了稳固的基石。人工智能实验室以及世界各地的黑客们，都得益于

这一系统。参与到这个文化小圈子的程序员，不一定都把自己称为黑客。不过绝大多数人都有着理查德·M·斯托曼般的情结：要是是一个程序或者一个补丁可以解决你自己的问题，它没准也能帮到别人。独乐乐不如众乐乐，何不分享给大家，起码也算积德行善了。

这种合作的精神不断的被商业秘密限制和贪婪之心所侵蚀，变成了一种商业秘密和合作精神的古怪搭配。加州大学伯克利分校的计算机科学家们基于他们从 AT&T 得到的 Unix 系统开发了一个名为 BSD 的操作系统。伯克利向那些已经向 AT&T 支付了 50000 美元软件许可证费用的学院提供 BSD 系统，仅仅只收取复制磁带的成本费。伯克利的黑客们还是尽可能的发扬分享合作的精神，但只能在 AT&T 所允许的范围内，并且他们还没有意识这两种不同的模式之间所存在的巨大冲突。

施乐公司不分享源代码的行为虽然让斯托曼感到烦恼，不过倒也没激怒到他。斯托曼在找源代码的过程中，甚至都没想过联系施乐公司：“人家都送咱打印机了，何必还非得兴师动众地直接管他们要代码呢？再说了，我敢肯定他们不发布源代码是有一种有意的行为，非要找他们要代码也是徒劳的。”

好消息最终还是来了：一名卡耐基梅隆大学计算机科学系的教授手头有一份这台激光打印机的源代码。

斯托曼对卡耐基梅隆大学并没有什么好印象。1979 年，一位在卡耐基梅隆大学的计算机博士研究生，名叫布莱恩·瑞德 (Brian Reid)。他开发了一个很不错的文本排版软件，名叫 Scribe。但是他却拒绝公开这个软件的代码，震惊了整个开发社区。这个文本排版软件是世界上第一个可以通过标记来指定特定语意（比如“强调这个单词”或“这是一段引文”）的排版软件，而在它之前，一般的排版软件都是通过低级的排版命令（比如“把这个词改为斜体”或“缩小这一段的页边距”）来表达类似的需求。瑞德决定把 Scribe 出售给一家匹兹堡名叫 Unilogic 的公司。他当时刚好博士毕业，正打算把开发 Scribe 的任务转手给别人，免得它流落到公共领域 (Public Domain) 里。（为什么

他会做出这样的决定，迄今为止还令人费解。)为了让这笔交易更具吸引力，瑞德在 Scribe 中动了点小手脚：他在里面放了个“定时炸弹”，让用户有 90 天的免费试用期，90 天一过，如果用户不交费，则不能再使用这个软件。

在斯托曼看来，这样的做法背叛了程序员应有单纯、简单的精神气质。瑞德的做法不但没有弘扬分享再传播的精神，而且竟然还助纣为虐，开辟了一条让软件公司强迫程序员为信息付费的新途径。不过斯托曼也没有太在意这个问题，因为他本人并不太使用 Scribe 这个软件。

Unilogic 提供了一个免费的版本给人工智能实验室使用，但是这个版本中仍然保留了那个“定时炸弹”。这个软件在实验室中正常使用了一段时间，但是有一天，有用户发现它不能正常工作了。一位名叫霍华德·坎农 (Howard Cannon) 的系统工程师花了几个小时调试了这个软件，最终发现了隐藏在软件内部的“定时炸弹”，他改动了一些代码，把它绕了过去。坎农对此非常愤慨，遇到人就抱怨说 Unilogic 浪费了他好几个小时的时间去解决一个故意隐藏在软件中的问题。

几个月后，斯托曼因为工作原因，需要去卡耐基梅隆大学出差访问。访问过程中，他在计算机系找到了那位据说拥有激光打印机软件代码的教授。运气不错，那天他还刚好就在办公室。

工程师之间的讨论总是开门见山的。几句寒暄之后，斯托曼就直入主题，说明自己是为了激光打印机的控制程序代码而来，有了这份代码，他就可以对此进行一些改进。可令斯托曼感到吃惊的是，这位教授竟然拒绝了他的要求。

“他跟我说他答应公司不能泄露代码。”斯托曼说。

记忆总是这么有趣。二十年过去了，斯托曼关于这段经历的回忆大部分都是空白。他都不记得当初为了什么事去的卡耐基梅隆大学，连具体哪年都给忘了。他也不记得那位教授是谁。根据 Scribe 的作者瑞德的回忆，惹到斯托曼的那位很可能是罗伯特·斯布鲁 (Robert Sproull)。他曾是施乐公司 PARC 研究所的研究员，如今在 Sun 研究所任部门

主管。二十世纪七十年代，斯布鲁在施乐公司 PARC 研究所负责激光打印机程序的主要开发者。八十年代他拿到了卡耐基梅隆大学的教职，并在那里继续他的激光打印机相关的工作。

可是，直接给斯布鲁发邮件，询问当初的事情，他却不记得这事了：“实在是无可奉告。”斯布鲁在邮件中写道：“我确实没记得有这么一件事。”

“斯托曼想要的代码包含的可都是前沿技术，那可是斯布鲁去卡耐基梅隆之前，花了几年的时间研发的成果。”瑞德回忆道。如果确实是这样，那可是一个重大的误解，因为斯托曼想要的代码是麻省理工学院已经用了很长一段时间的打印机的程序代码，并不是要最新版本。不过在他们那次简短的交谈中，也许根本都还没来得及谈及是要的哪个版本。

在后来的各种演讲中，斯托曼一再提及这件事。强调斯布鲁由于签署了保密协议，从而不能泄露源代码。如今在各个 IT 公司，签署保密协议 (NDA) 已经非常普遍，而在当时为软件代码签署保密协议还算先例。这也间接地反映出施乐公司对激光打印机项目的重视程度。“施乐公司打算把激光打印机做成商业产品。”瑞德说，“他们除非疯了，才会把代码泄露出去。”

但在斯托曼看来，签署这种保密协议完全是一种自私的行径。那会儿大家都把软件认为是一种共有的资源，卡耐基梅隆大学一些研究员的做法，就意味着否定这个公理。这对斯托曼来说，就好比是一个农夫眼看着用来灌溉庄稼的河流干枯了。循着河道往上巡查，竟然看到一道大坝从天而降，拦住河水。大坝上赫然印着几个大字：施乐公司。

现在来看，那时施乐公司用各种花言巧语，把一位程序员同胞带入了这片新天地，与世隔绝。不过一开始，斯托曼也没把矛头指向施乐公司，而是怪罪到了个人性格上。“我当时非常生气，都不知道怎么表达。我二话没说，扭头就走了。”斯托曼说，“没准我还使劲摔了门，谁知道呢。我能记得的，就是当时想赶快离开。我特地拜访他的原因

就是希望能与他合作，所以我完全没想到过会被拒绝。当他拒绝我的时候，我觉得无话可说，非常的失望与生气。”

二十多年过去了，可斯托曼当初的怨气还在。他甚至把这个事件描述为人生转折点。然而，那之后几个月中，在人工智能实验室以及斯托曼身上发生的各种事，却比这次打印机事件还令人难以接受。斯托曼，本是一名孤独的黑客，本能地对绝对权威有戒心。在经历了这一系列事件后，变成了一位斗士，把传统的自由、平等、博爱的精神引入软件开发领域。而这次的打印机事件，在其一生的无数事件中则最值得一书。

“这是我第一次遭遇所谓的保密协议，但这马上让我明白，在保密协议面前，总会有一些无辜受害者。”斯托曼坚定的说，“在这次事件中，我就是那个受害者。人工智能实验室和我本人都扮演了受害者的角色。”

斯托曼后来解释说：“如果他只是因为个人原因拒绝与我合作，也许事情也就这样不了了之了。也许我会觉得他是一个蠢蛋，但也就仅此而已。但是事实上，他并不是因为个人原因而拒绝我，是因为他先前已经向公司保证了不会泄露源代码，不但不能把源代码给我，也不能给其它任何人，那这个事情性质就变得严重了。”

尽管之前也有过类似的不快经历，可都不能与这次的打印机事件相提并论。这次事件让斯托曼彻底意识到，这些一系列的事件，正悄悄地侵蚀自己所珍视的文化——那个神圣不可侵犯的“黑客”小圈子的核心文化。他说：“我一直有个信念，那就是软件应该可以可以被分享的，但是我不太确定应该怎么去把它变成现实。我的想法还不太成熟，也不太系统，所以没有办法把它精确的表达出来，让其它人理解。经历了这些事件，我开始渐渐认识到问题的本质和它的严重性。”

作为一个世界顶级研究机构的顶级程序员，斯托曼之前一直都无视那些程序员同行所作的各种妥协和让步，因为他们还不至于影响到斯托曼的工作。而如今施乐激光打印机的到来，让斯托曼开始注意到其他计算机用户一直在忍受的程序和机器。

如今，这台激光打印机已经强势入驻到人工智能实验室了，而外面的世界也悄悄发生着变化。除了偶尔卡纸以外，打印机工作还算正常。可是按照个人喜好修改软件则不可能了。从软件业的角度看，这打印机的出现是个信号，预示着软件是公司的重要财产。谁也不再愿意发布软件源代码，因为这可能会给潜在竞争对手机会，让他们可以轻松山寨自己的产品。而在斯托曼看来，这打印机简直就是个卧底。十来年的尝试，私有软件总算在人工智能实验室中占得一角。私有软件，也就是如今所说的专有软件，这次被打扮成礼物，潜入得悄无声息。

施乐公司后来还发出邀请，让一些程序员再使用它们的礼品。斯托曼说，要是再早几年，他没准也无法拒绝这种免费午餐。是那次打印机事件，让斯托曼建立起了道德防线。它不仅燃起了斯托曼久抑的怒火，还让他对以后的各种礼品心存戒备；它更让斯托曼开始思考一个让他自己也坐立不安的问题：要是以后哪个黑客同行进到自己的办公室，向他索要代码，他究竟会不会拒绝呢？

“要是谁跟我建议，让我以这种方式背叛我的同事，我就会回忆起在打印机事件中，我和我的同事，被别人背叛的感觉，那份怒火终生难忘。”斯托曼说，“我会回敬给分发专有软件的人：‘谢谢你的软件，非常棒！可是我不能接受你的那些保密协议，很遗憾，我不会用它。’”

斯托曼带着这种态度，经历了动荡的八十年代。在这期间，麻省理工学院的同事们纷纷离开人工智能实验室，走进公司，签署了保密协议。大多数保密协议都有解密时间，而这则成了很多黑客们的借口。他们会辩解说：软件迟早会成为公共资源。只要保证软件在早期的开发阶段不被泄露，就可以保证让各位黑客朋友们可以进入到顶尖项目中工作。这些借口，在斯托曼看来，是迈向深渊的第一步。把一个出色的项目放到社区以外去开去，能有什么好处可言呢？

斯托曼很快就会明白，要拒绝这些要求和邀请，不仅需要一些个人牺牲，更会被其他一些黑客们疏远。这些黑客虽然也对各种保密协议嗤之以鼻，但却会更圆滑地对它加以评判。拒绝提供源代码，在斯

托曼看来，不仅违背了二战以来深植入软件开发中的科学精神，更违背了“己所不欲，勿施于人”的道德准则。

打印机事件的重要意义恰在于此。正如斯托曼所言，倘若没有这次事件，他的人生也许就会落入平常，纠结着，一边开发专有软件，一边痛苦地编写没人会看到的代码。当然也不会有着如今清晰的思路，更不会去解决别人沿未想过的各种问题。最重要的，他心中也不会有那份不平，推动着他去追求他的政治理想和道德信仰。

“从那日起，我决定绝不参与其中。”谈起软件保密协议和类似的事情，斯托曼如是说。在他看来，这是一场以个人自由换取便利的交易。“我决定绝不让第二个人为此成为像我一样的受害者。”

Endnotes

¹参考[附录 A – 黑客的三层含义](#)，深入了解“黑客”这个词的内在含义。

Chapter 2

黑客路漫漫

纽约大学计算机学院坐落在沃伦·韦弗大楼 (Warren Weaver Hall) 之中，和著名的华盛顿广场公园相隔两个街区。大楼周围充斥着从空调压缩机里传出来的股股热浪。这阵热浪好似屏障一般，驱散着各路闲杂人等。倘若冲破这道屏障，面前还会出现另一道防线：整座大楼的唯一一道入口，你被一个安检岗拦下。

走过安检口，气氛似乎缓和了些。可依旧还能看到各处林立的警示牌，指向消防出口。一路下来，你会有这样的感受：哪怕是在 2001 年 9 月初的纽约，这样一座现代化的都市之中，也要记得，小心驶得万年船。

可当你继续向前，走到中央大厅的时候，这份谨慎的感受却被一群人冲破了。他们之中，有些看起来像是纽约大学的学生。大部分则蓬着头，好似等待着一场盛大的音乐会。在这短暂的上午，这群人占据了沃伦·韦弗大楼。这下子，安检人员反倒闲下来了。他们不规矩的斜躺坐在椅子上，看着电视剧。要是哪位访客，问起“演讲”的事，这位保安就会连话也懒得说一句，干脆冲着旁边的大礼堂一耸肩，然后继续看他的电视剧。

有如此神通，能让保安放半天假的这位演讲者，此时正坐在大礼堂中。他就是理查德·M·斯托曼：GNU 工程的创始人，自由软件基金会的第一任主席，1990 年麦克阿瑟奖 (MacArthur Fellowship) 获得者，同年美国计算机协会 (Association of Computing Machinery) 格雷·莫瑞·霍普奖 (Grace Murray Hopper Award) 获得者，2001 年日本武田基金会 (Takeda Foundation) 的武田奖 (Takeda Award) 共同获得者。当然，他也是我们熟悉的那位麻省理工学院人工智能实验室的黑客。前阵子黑客相关的网站上，包括 GNU 工程的官方网站上 (<http://www.gnu.org>)，都刊载了一条新闻：理查德·M·斯托曼将在他的家乡，纽约曼哈顿，发表演讲，回应微软在前些时候关于 GNU 通用公共许可证 (GNU General Public License) 的抨击。

斯托曼演讲的主题，是关于自由软件运动的历史和未来。这次演讲的地点尤为重要。不到一个月前，微软的副总裁克雷格·蒙迪就是出现在纽约大学斯特恩商学院 (Stern School of Business)，抨击 GNU 通用公共许可证 (GNU General Public License) ——简称 GPL。GPL 是斯托曼 16 年前想出来的法律武器，用来对抗工业界中越来越盛行的专有软件。1980 年，斯托曼经历了那次施乐打印机事件之后，预感到了软件专有化的潮流逐渐到来。为了抗衡这潮流，斯托曼提出了自由软件的概念：即用户可以自由地使用，学习，修改和再发布的软件。如今，GPL 已经俨然成为了自由软件社区的核心工具。简单说来，GPL 是一个软件使用协议，它利用版权法，将自由软件锁定在公众可以自由使用和修改的领域。一旦锁定，这个软件就不会再被专有化。不仅仅是这个软件本身被锁定为自由软件，这个软件的任何衍生品也会成为自由软件。也就是说，倘若某个软件以 GPL 形式授权发布，这个软件以及任何它的衍生品，都可以被用户自由使用和修改。所谓一个软件的衍生品，也就是任何使用了该软件的代码的作品。哪怕一个软件仅仅使用了某个 GPL 授权软件的一小部分代码，这个软件也将被要求以自由软件形式发布。恰恰是这个原因，软件业的很多人把 GPL 称为病毒式许可证，因为它像病毒一般，“感染”所有它触及到的程序。¹

随着信息产业的发展，全球越来越依赖软件和软件标准。在这样的环境下，谁都无法忽视 GPL。哪怕曾经嘲笑过 GPL 的公司，也不能再把它视为空中楼阁。因为越来越多的软件都是以 GPL 形式授权：Linux，一个最初由芬兰大学生林纳斯·托瓦兹（Linus Torvalds）于 1991 年开发出的类 UNIX 操作系统内核；GNU Emacs；GNU 调试器；GNU 编译器等等都是以 GPL 授权的。这些工具在一起，形成了一个完整的自由操作系统。世界各地的黑客，为这套操作系统贡献着代码。每个黑客也可以自由地拥有这样一套操作系统。如今，很多计算机公司都不再把这样一套自由操作系统视为威胁。相反，IBM，惠普，Sun 等公司都依赖这个操作系统，并在这套系统之上开发和出售自己的软件产品。²

在黑客这个圈子里，这个操作系统也被当作战略武器，对抗微软公司。这个总部坐落于华盛顿州雷德蒙德市的软件公司，从二十世纪八十年代开始，已经俨然成了软件业的垄断巨头。作为 Windows 操作系统的拥有者，微软公司面对业界同行转向 GPL 许可证的事实，已经再也坐不住了。Windows 中，几乎每行代码都是私有的。至少按照法律条款看，它是归微软所有。倘若有谁不慎在 Windows 中放入一点 GPL 授权的代码，整个 Windows 产品都会被“感染”为自由软件。在微软看来，这就好比把哪吒请进了龙宫，整个公司都得翻江倒海，它必须把整个操作系统的代码都以自由软件的形式发布。竞争对手就可以自由的复制、修改这个系统，并销售修改后的版本，这可能瞬间颠覆微软公司龙头老大的地位。

因此，微软极度关注 GPL 的普及情况。也正因如此，蒙迪才来纽约大学，抨击 GPL 以及“开源”软件的开发销售模式。（微软并不承认“自由软件”这个概念，而是习惯把矛头指向 chapter 11 章节中即将谈到的“开源”阵营，回避自由软件运动的存在。）这才引得斯托曼也决定来纽约大学，在同一所校园内，回应蒙迪的抨击。

二十年的时间，对软件业来说，可是个不短的年头。遥想二十多年前的 1980 年，斯托曼还在咒骂着人工智能实验室的施乐打印机；微

软，这个被当今黑客视为全球软件业巨头的公司，还是个私人的创业小公司；而 IBM，这个被当年的黑客视为全球计算机业巨头的公司，还没推出个人计算机；至于 IBM 的个人计算机推动了整个计算机业的发展，则也是之后的事情了。而当今我们习以为常的很多技术，在那时都还没出现。包括 WWW 网络，卫星电视，32 位终端游戏机等等。如今的很多计算机巨头，则也一样在当时还没建立。包括 AOL，Sun，亚马逊，康柏，戴尔等等。

高科技产业在这短暂时间内迅速成长的过程中，则充斥着关于 GPL 的争论。GPL 的支持者声称：由于计算机硬件平台的短暂寿命，为了避免买到过时产品，用户会倾向于购买大品牌的产品。由此带来的结果，造成了软件市场赢者通吃的局面。³而如今，由于软件被认为是专有的，具有垄断地位的公司就会滥用它的权利，使得整个产业停滞不前。垄断企业会堵住所有去路，让竞争者无法生存，也让后起的创业公司无处立足。

GPL 的反对者，则恰恰相反。他们声称销售软件和购买软件一样，具有风险。倘若没有法律保证软件可以被私有化，人们将再也看不到杀手级应用的繁荣（所谓杀手级应用，即某种技术足以打开一片全新市场）⁴，公司也将失去继续创新的动力。正如蒙迪于五月三日在纽约大学所说的，GPL 的“病毒”特性为各大公司带来了威胁，威胁着它们赖以生存的软件产品。蒙迪道：

它严重地威胁着独立的商业软件。出售软件不仅仅是要收回发行成本，还有更多的人力附加价值。而 GPL 则让这种买卖变成天方夜谭。⁵

在过去的 10 年中，无论是 GNU/Linux 还是 Windows，都有了长足的进步，赢得了各自的成功。两者双赢的局面，似乎使得无论支持或反对 GPL 都有了足够的理由。但是，作为像斯托曼这样的自由软件活动家来说，成功并不是事情的关键。真正的问题并不是自由软件和私有软件哪个更为成功，而是哪个更道德。

不管怎么说，这场自由软件与私有软件的战役在软件工业发展史上是一次重要的事件。哪怕像微软这么强大的企业，也需要以来第三方软件开发者。恰恰是他们开发的工具，程序，游戏才使得 Windows 系统更加吸引用户。回顾近二十年软件业的发展，哪怕不提自家公司的成长，蒙迪依旧提醒他的听众，不要被自由软件运动的风潮吹昏头脑：

二十年的经验表明，保护知识产权的经济社会，配合降低研发成本的商业模式，可以极大地促进经济发展和财富分配。⁶

这段来自蒙迪的警告，成了斯托曼今日演讲的背景。现在，距离蒙迪演讲后不到一个月，斯托曼就站在演讲礼堂前方，背对着黑板，摩拳擦掌，迫不及待地准备发表他的演说了。

这二十年来，相比软件业翻天覆地的变化，斯托曼的改变则更加明显。当年他曾是那位精瘦，不留胡须的黑客，曾在那台 PDP-10 前，没日没夜地和它畅谈。而如今，在讲坛上的那位，已是人近中年，发福，留发，蓄须。他会花大把时间回复邮件，在公众或程序员面前，发表演说，组织演讲——正如今天这次一样。他今天穿着海蓝色的 T 恤，一条棕色的涤纶裤子，看起来好似刚刚踏出沙漠的隐士。

人群之中，则充斥着和斯托曼有着类似打扮的听众。很多人带着笔记本电脑和蜂窝调制解调器，以及各种录音录像设备，准备把斯托曼的演讲传上互联网，那里还有更多的听众等着他的言论。听众中，男女比例大约十五比一。大约每七八个女听众里，就有一位拿着企鹅的毛绒玩具。这是 Linux 的吉祥物。还有些女听众则抱着泰迪熊。

人潮涌动，斯托曼则离开了演讲台，坐在第一排的听众席上，把一直开着机的笔记本电脑拿出来，开始敲键盘。接下来的十分钟里，进来的学生，教授和粉丝逐渐走到他旁边，围观他工作。而斯托曼对此则完全不在意。

既然在大学演讲，学院派的规矩是少不了的。演讲嘉宾介绍则是重要一环。对斯托曼的介绍可谓阵容强大。纽约大学的两位教授分别为他做两段开场白。第一位，是来自纽约大学斯特恩商学院高新技术研究中心的主任，麦克·乌列茨基。

“大学之地，争辩之所；争辩之处，兴趣所在。”乌列茨基道，“本次讲座，恰是秉承此道。个人愚见，所谓开源，甚是有趣。”

还没等乌列茨基把话说完，斯托曼就站起来挥着手喊道：

“我是搞自由软件的，开源是另外一码事！”

这一嗓子引来了一阵哄堂大笑。笑声褪去，掌声渐起。当下，听众中绝大多数都对斯托曼的这份咬文嚼字有所耳闻，更知道在 1998 年，他和开源阵营支持者的那次争论。斯托曼的这一行为，就如同整点新闻一般，早在大家意料之中。

乌列茨基草草结束介绍，走下讲堂。接下来对斯托曼做介绍的，是纽约大学计算机系的教授埃德蒙·舍恩伯格。作为一个计算机程序员，又是 GNU 工程的贡献者，他很清楚该如何用词。他站在当代程序员的角度，扼要回顾了斯托曼的事业。

“放眼全球，脚踏实地，斯托曼是典范。他严肃地审视了软件源代码不对公众公开的事实，发展出了一套严密的逻辑体系。这套体系敦促着我们，让我们不得不重新思考如何开发软件，何谓知识产权，以及软件社区究竟是何物。”⁷

舍恩伯格的致辞迎来了更多的掌声。斯托曼在掌声中，合上笔记本电脑，挪出身子，走上讲台。

演讲开始，斯托曼表现得更像是个老练的喜剧演员，让人没法察觉这是一场严肃的政论讲座。“我首先要感谢微软公司给了我这样一次机会，来到贵校，畅抒己见。”斯托曼玩笑道，“过去几周里，我都觉得自己像个没落作家，自己的作品都无人问津。”

为了照顾新人，斯托曼用一套类比，简要介绍了自由软件。他把软件代码比作烹饪菜谱。二者都提供清晰的步骤，说明如何一步一步完成某个特定任务。同时，人们都可以很容易地按照自己的需求，对它们

进行修改。“你不用按照菜谱的每一步严格执行操作。”斯托曼说，“你可以少放点调料。喜欢蘑菇，放些蘑菇；口味淡就少放盐；加点胡椒粉什么的。”

最重要的，斯托曼强调，软件和菜谱都很容易被分享。倘若有位客人来家里吃晚餐，那么把菜谱给他无非是花些时间，费点笔墨。拷贝软件则要求更少，只要轻点鼠标，费点电。而分享之后，你则起码有两份收获：增进了友谊；同时，下次你需要帮忙的时候，对方也会有所回报。

“想象一下，要是菜谱全被封锁在一个黑匣子里，那会如何？”斯托曼话峰一转，“你不知道他们用了什么调料，只有他们才能更改配方。你如果把菜谱抄出一份，送给朋友，他们就把你叫贼，把你关进牢房，长达数年。如果你早就习惯了把菜谱传来传去，这样的世界一定会让你觉得不可理喻。可这一切恰恰发生在专有软件的世界之中。在这个世界里，很平常的社交行为被严格禁止，或者被想方设法避免。”

类比过后，斯托曼又提起了施乐打印机事件。和烹饪菜谱的类比一样，打印机事件也是一个称手的工具。两者介绍完，听众就可以了解到如今的软件业究竟发生了多大改变。斯托曼的介绍，把听众拉回了曾经那个年代，那时还没有亚马逊和它的一键支付；没有微软和它的 Windows 系统；没有甲骨文数据库。这样的背景之下，听众有了很大的想象空间，可以不受当下这些所谓的大公司影响，重新审视所谓的软件所有权。

讲起施乐打印机事件，斯托曼是轻车熟路。他好似律师做法庭最后陈词一般，字斟句酌。当说到卡耐基梅隆大学的那位计算机教授拒绝给他源代码的时候，斯托曼道：

“他背叛了我们。”斯托曼稍有停顿，接着说，“但他不止背叛了我们，更有可能背叛你！”

“你”字一出，斯托曼就伸出食指，指向在座听众。听众之中，有人稍有皱眉。而斯托曼的目光，则移到了前排，一位听众正在低头偷

笑。“而且我觉得，他更有可能背叛你。”斯托曼指着刚才偷笑的那位听众。

这个临场的包袱，把一个人的窃笑变成了全场大笑。各种行为，好似舞台剧一般。笑声中，他总结道：“要想不被他背叛，你只能盼着晚点投胎。”笑声又起，“因为这位教授承诺，拒绝和地球上大多数人合作。”

斯托曼一字一顿：“他签署了保密协议。”

理查德·马修·斯托曼，从久经历练的学术精英，到一呼百应的运动领袖。这一蜕变本身就饱含寓意。它诉说着斯托曼的顽强与固执，诉说着他的决心和毅力。更清晰地诠释了自由软件运动的价值和远见。这之中，也当然包含了斯托曼编写的高质量代码，字里行间，凝结了斯托曼的心血，将他的经历，铸成了传奇。传奇之上，更能看到 GPL 不可遏制的蓬勃生命力。而 GPL 本身，作为法律界的一大创新，已经被公认为斯托曼的重要成就。

当下，计算机和相关软件技术已成为整个世界的一大支柱。理查德的经历，也更彰显了在如此背景之下，政权民意的风云变化。

这一切的一切，也许恰恰能解释，为什么斯托曼发起的运动能如此长久不衰，而很多当年的名牌大厂，却早已风光不再的原因。遥想当年，1984 年，斯托曼刚刚发起了 GNU 工程⁸。面对这一工程，自由软件运动内外，都充斥着对这个工程的无视，嘲讽，甚至攻击。一路走来，GNU 工程虽然有过几次跳票延期，却还能在大多数时候按时交付，完成一个又一个的发布计划。踏过了十八个寒暑，GNU 工程也日渐成熟，在软件市场中，得以赢得一席之地。而这近二十年来，自由软件的理想被理查德精心呵护，渐渐传遍大江南北。

要想了解这潮流背后的缘由，须得兼听八方言论。这不仅包括斯托曼自己的评价，更要倾听和他在同一战壕里作战的战友们的叙述。其实，斯托曼的个性并不复杂，倘若你坚信“所见即所得”，那么斯托曼这个人就不难被理解。

“想要了解斯托曼这个人，你必须要把各处细节联系起来，看成一个有机的整体。”伊本·莫格林（Eben Moglen）说道。他是自由软件基金会法律顾问，同时也是哥伦比亚大学法学院教授。“在斯托曼身上有着各种古怪脾气，这也许会把人拒之千里。而这份不同寻常，恰恰就构成了斯托曼这个活生生的人。他对挫败异常敏感，他对道德准则恪守不渝。他不肯妥协的个性，在关键问题上不肯让步的固执，这一切的总和，最终让我们看到了当今的斯托曼。”

一个简单的打印机事件，变成了燎原之火，燃遍全球，足以和全球最富有的软件公司抗衡。要想回顾和解释这一切，并非易事。首先要了解软件所有权，以及它是如何走到当今的重要位置的；更要和人类健忘的本性做斗争；还要能从关于斯托曼的各种神话和攻击之中，看出本质。最后，还要能理解斯托曼在软件领域的过人天赋；以及他如何把这份天赋用到其他领域；还有在这一过程中的各种成败得失。

当我请斯托曼做一番自我总结的时候，他也强调了莫格林提到的个性和原则：“坚强固执是我的本性。很多人在尝试做一些事，遇到了困难，就退缩放弃了。可我从不言弃。”

他也同样感激自己的运气。倘若当初没有那次打印机事件，没有其中的各种人事冲突，没有当年的各种机缘巧合，他也许不会放弃麻省理工学院的研究职位，不会重新抉择，选择一条与众不同的道路，并为之奋斗终身。是身边的各种因素，最终让斯托曼得以做出不同凡响的成绩。

“我正好有合适的技能。”斯托曼回顾着当初发起 GNU 工程的决定，总结道，“除了我以外，没人在做这事。我就觉得：‘责任在身，我若不做，舍我其谁。’”

Endnotes

¹事实上，GPL 的没有这么夸张的“威力”。把你的程序与另一个 GPL 许可证的程序放在同一台计算机上运行并不会把你的程序也变成自由软件。

斯托曼说：“把一个东西说成是病毒是一种很粗暴的类比，也许把 GPL 比成是吊兰还稍微贴切一些，如果你把它切下来放到别处，它会继续生长。”

要了解 GNU 通用公共许可证的更多信息，请访问：<http://www.gnu.org/copyleft/gpl.html>

² 尽管这些软件都在 GNU/Linux 上运行，但它们本身不一定是自由软件，相反的，他们中的大部分都是私有软件。跟 Windows 一样，这些私有软件同样不尊重你的自由。虽然它们对于 GNU/Linux 的成功起到了一定的推动作用，但它们对于自由软件运动中实现自由的目标并没有什么贡献。

³ 参考 Shubha Ghosh 于 2000 年 1 月在 *Gigalaw.com* 上发表的文章：《Revealing the Microsoft Windows Source Code》。<http://www.gigalaw.com/>

⁴ 其实杀手级应用并不一定需要是私有化的。我觉得读者应该这么来理解：软件市场就像买彩票一样，预斯的收益越高，越多的人愿意参与其中。有关杀手级应用的现象，Philip Ben-David 2007 年 12 月 7 日发表的文章《What Happened to the “Killer App”?》对此有清晰的阐述。<http://www.ecommercetimes.com/story/5893.html>

⁵ 参考微软资深副总裁克雷格·蒙迪的《The Commercial Software Model》一文，节选于他 2001 年 5 月 3 日在纽约大学斯特恩商学院所做的演讲。<http://www.microsoft.com/presspass/exec/craig/05-03sharedsource.asp>

⁶ 同上。

⁷ 如果他今天再来发表这样的演讲，斯托曼一定会指出“知识产权”这个词所存在的认识上的偏差和可能造成的混淆。请参考：<http://www.gnu.org/philosophy/not-ipr.html>

⁸ GNU 是 “GNU’s not Unix” 的缩写。斯托曼在 2001 年 5 月 29 日纽约大学的演讲中，讲述了这个缩写的起源：

作为黑客，我们总是想给程序起一些有趣或古怪的名字，因为给程序起名字也是编写程序的乐趣中的一部分。我们也有使用递归缩写的传统喜好，用这样的方式表明我们所新写的程序与某个现有的程序具有一定的相似性……在为 GNU 工程起名字时，我试图寻找一个递归缩写来表达 “Something Is Not UNIX” 含义，我用 26 个字母去取代 “Something”，但得到的结果都差强人意，看起来并不像是一个单词。于是我决定把 Is 这个单词采用缩写的形式，这样我就可以得到一个三个字母缩写词，也就是说是 “Something’s Not UNIX”。我尝试了各个字母，最终决定选用 “GNU” 这个词。

GNU 作为一个双关词，斯托曼建议软件用户在发音时，对第一个字母 “g” 进行发音（也就是说，把它读作 “gah-new”）。这一方面是为了避免与表示白尾角马这种非州羚羊的单词 “gnu” 发生混淆，另一方面也是避

免与“新 (new)”这个形容词发生混淆。“我们已经为 GNU 工程奋斗了 17 年，它也一点也不‘新’了。”

来源：作者的笔记，以及斯托曼 2001 年 5 月 29 日在纽约大学演讲的在线讲稿 “Free Software: Freedom and Cooperation”。<http://www.gnu.org/events/rms-nyu-2001-transcript.txt>

Chapter 3

黑客正年少

理查德·斯托曼的母亲，爱丽丝·李普曼（Alice Lippman）始终还记得她发现理查德过人天赋的那一刻。

“我记得他那会才八岁。”李普曼回忆着。

那是 1961 年，李普曼刚刚离异，做了单亲妈妈。她家位于曼哈顿上西城，是一个一居室的小公寓。一个周末下午，她正在自己家里翻着一期《科学美国人》杂志，打发时光。她翻开自己最喜欢的一版，马丁·加德纳（Martin Gardner）主编的专栏：《数学游戏》。作为一个教艺术的代课老师，李普曼常常被这一专栏的各种头脑游戏吸引。她看了一眼在旁边沙发上老老实实呆着的理查德，决定开始挑战本周的思考题。

“其实我做这些题目做得不是很好。”李普曼坦言，“可作为艺术家，我发现它们可以帮我缓解压力，激发灵感。”

可她刚开始做，就遇到了障碍。眼看她就打算扔开杂志，放弃解答了，可这时候，她发现有人正在轻拽她的衣袖。

“那是理查德，”她回忆道，“他想问我要不要帮忙。”

她看看自己儿子，又回过头看看杂志上的题目，她还是有点半信半疑。“我问理查德，你读过这杂志了？”李普曼说，“理查德说读过了。而且他竟然已经解决这个题目了。之后，我就记得他开始给我讲解答题思路。”

听着自己儿子清晰的逻辑，李普曼开始的半信半疑瞬间转变，一切让她觉得难以置信。“我知道理查德很聪明，”她说，“可这次却是实实在在让我见识了他究竟有多聪明。”

三十几年过去了，李普曼回忆起这事，最后笑道，“老实说，我到现在都没明白那道题是怎么做的。我能记得的，只是当时非常震惊，感叹理查德居然能做出那个题目。”

李普曼坐在餐桌前，这里是她在曼哈顿住过的第二个公寓，在一座公寓楼里，是一个三居室的宽敞房子。1967年，她再婚嫁给了莫里斯·李普曼（Maurice Lippman），就带着斯托曼搬到了这所公寓里。时过境迁，莫里斯已经去世，斯托曼也长大成人。谈起当年抚养斯托曼，这位犹太妈妈流露出的尽是自豪和欣慰。在餐厅里的橱柜里，摆放着一张八乘十英寸的照片。照片里是满脸络腮胡须的斯托曼，身穿博士服。这张照片摆在那里，让其他几张李普曼侄子侄女的照片显得不再引人注目。不过在客人问起这张照片之前，李普曼总会先调侃几句，解释一下为什么把它放到这么显眼的地方。

“理查德坚持要把这张相片给我。这是他当年获得英国格拉斯哥大学（University of Glasgow）荣誉博士学位的时候照的。他跟我说：‘你看，这是我有生以来参加的第一个毕业典礼。’”¹

抚养理查德这么一个孩子，李普曼总是有很多话可说。每当别人说起理查德最新的各种或古怪或固执的行为时，她总会翻出更多类似陈年旧事作为回应。

“他以前是十足的保守派。”她说，把双臂铺开，以示强调，“曾经就在这张桌子前，我们两人展开过一次很激烈的争论。我曾经是城市公立学校教师，其中，我是第一批坚持要成立工会的人。可理查德却觉得工会是腐败之源。他甚至非常反对社会保险。他认为，人们用

买保险的钱，完全可以投资其他东西，赚更多的钱。当初谁会想到他十年之后会成为一个如此的理想主义者。我就记得当初我女儿曾问过我：‘你说以后理查德长大了会是什么样？法西斯极右翼分子？’”²

李普曼 1948 年和理查德的父亲丹尼尔·斯托曼结婚，于 1958 年离异。李普曼成了单亲妈妈，抚养和监护理查德的重担几乎都压在了她的身上。一路走来，她见证了理查德蔑视权威的个性，更见证了他对知识的如饥似渴。这两股力量在理查德身上几经纠缠，也让李普曼和理查德有过不少摩擦冲突。

“他好像一直都不想吃饭一样。”李普曼回忆着理查德八岁到 1970 年中学毕业的那段时间，“他就跟个聋子似的，我每次都得叫他九到十次，他才肯过来吃饭。”

同样是这件事，到了斯托曼嘴里，就掺进了政治的意味。

“我特别喜欢读书。”斯托曼说，“赶在我想读书的时候，我妈妈恰巧叫我去吃饭，我肯定不会去。我就觉得，凭什么我不能读书？凭什么我妈妈可以指点我该做什么，该读什么书。我坚持民主和个人自由。这些思想不应该仅仅局限在成人身上，未成年人也同样适用。”

斯托曼坚信个人自由高于任何权威。这样的思想也被他带到了校园之中。斯托曼比其他的孩子早上了两年学。和很多天才儿童一样，斯托曼也有着自己的烦恼。那次他帮妈妈解决完杂志上的数学题之后没多久，老师就妈妈约去学校，进行了一次长谈。

“他拒绝写作文写文章。”李普曼回忆着当时的情景，“除了高中最后一年，他最后一次写作文是四年级。那是一篇关于西方计数系统的文章。”

凭借着分析性思维的天赋，斯托曼沉浸在数理化的世界，而代价则是其他方面严重欠缺。老师们看来是头脑简单的行为，在李普曼看来，倒觉得是斯托曼好强所致。比起其他斯托曼天生就不太在行的领域来说，数学和自然科学给斯托曼提供了太多的学习机会。大概在 10 到 11 岁那会，斯托曼的同学们都在玩触式橄榄球。她记得有一次斯托

曼回家特别生气。“他当时非常想和他们一起玩，可他身体的协调性实在太差了。”李普曼回忆道，“这实在让他非常恼火。”

这份感觉让斯托曼更埋头在理科世界里。可哪怕在数理化的世界，他的好强有时也会成问题。在八岁的时候，斯托曼就已经把微积分教材翻了个透。他觉得自己完全没必要在大人面前装成傻傻的天真宝宝。他中学的时候，妈妈曾从附近的哥伦比亚大学里雇来了一位大学生，以期在斯托曼面前扮演大哥哥的角色。结果这位学生来了一次就没再来了。“我觉得斯托曼当时讨论的问题可能超出他的极限了。”李普曼推测。

另一个李普曼喜欢提起的小事，发生在六十年代早期。那时李普曼已经离异两年，也从皇后区搬了出来。斯托曼那会七岁，迷上了火箭模型。他常常去河滨公园发射火箭模型。一开始他只是随便瞎玩，可后来他开始记录每次的发射数据。起初，这些行为并没引起李普曼注意，她觉得这兴趣就和玩数学游戏一样。可有次 NASA（美国国家航空航天局）发射火箭前，李普曼问斯托曼要不要看电视上的现场直播。

“他可逗了，上来就跟我说：‘可我的研究成果还没发表呢！’这话说的就好像真有点资料要给 NASA 看似的。”斯托曼自己不记得这件事了，但他觉得他当时应该是有点苦闷，因为他根本没有任何东西可以给 NASA 看。

这个小事倒是很好地表现了斯托曼性格中独具一格的特点——紧迫感。这种紧迫感一直伴随在他左右。其他孩子还在桌前玩耍的时候，他在房间里读书；别的孩子在痴迷橄榄球运动员约翰尼·尤尼塔斯的时候，斯托曼则崇拜着火箭专家。“我当时就是这么古怪。”斯托曼在 1999 年接受一次采访的时候，如此总结着自己的童年，“没过几年，我的朋友就只剩下各位老师了。”³ 对于他自己古怪的性格，斯托曼并不觉得丢人，但是他对于自己不擅长社交这一点总是深感遗憾。不过，正是他特有性格和社交能力的缺乏造成了他与现实社会之间的隔阂。

虽然斯托曼总比同龄人显得古怪很多，李普曼还是决定让他追求自己的兴趣和热情。12 岁那年，斯托曼参加了科学夏令营，之后上了

私立中学。学校的老师建议让斯托曼参加“哥伦比亚科学之星计划”。该计划旨在为具有天赋的纽约中学生提供更多的学习机会。斯托曼参加了这个计划，于是每周六，他都会赶去哥伦比亚大学。

丹·柴斯（Dan Chess）也参加了科学之星计划，他回忆中的斯托曼，哪怕在众多同龄的科学爱好者之中，也算是古怪的。“我们都是书呆子，技术宅，可斯托曼在这圈子里仍是特立独行的。”如今的柴斯已经是亨特学院数学系的教授，他回忆道：“他更是聪明绝顶。我见过很多聪明人，可斯托曼是我见过最棒的。”

另外一个科学之星计划的成员，赛思·布莱德巴特（Seth Breidbart）也提供了类似的证据。布莱德巴特如今也是个程序员，依然和斯托曼保持联系。他回忆当年只有 15 岁、剃了短发的斯托曼甚至有些“令人生畏”。

“这可难解释了。不是说斯托曼会拒人千里，只是说在他身上，总能看到一种紧迫感。他知识丰富，可有些地方又非常爱钻牛角尖，非常固执。”布莱德巴特回忆说。

“紧迫感”，“爱钻牛角尖”，“固执”，这样几个形容词放在一起，放在今天哪个中学生身上，多半会让人觉得是青春期综合症的症状。其实，在 2001 年 12 月的《连线》杂志中，就有一篇名为《天才综合症》的报道，采访了几个自闭症儿童患者，他们都在数学和科学方面具有天赋。报道中，患者家长对患者的描述，在很多方面都类似李普曼对斯托曼的形容。哪怕是斯托曼，提到自己精神方面，也曾犹豫再三。在一份 2000 年《多伦多之星》报纸上的资料中，斯托曼曾描述自己为“濒临自闭”，以此解释自己一直以来，在社交方面的孤僻，并介绍了自己如何不断克服这种心理。⁴

这种判断和当下流行的“异常行为”研究是分不开的。《天才综合症》的作者史蒂夫·西尔贝曼（Steve Silberman）介绍，美国的心理学家在近年来才把自闭症作为一系列症状行为特征的统称。这些行为包括：不擅运动，不擅社交，但又对数学，计算机和类似的具有规律的系统有着极度的依恋，并擅长于此⁵。

“我当初也许还真有这种毛病吧。”斯托曼说，“可自闭症患者的另一个症状是对音乐和节奏迟钝。我还能跳舞呢，而且我对节奏特别敏感，喜欢复杂的节奏。这么看倒也未必是自闭症了。”⁶

柴斯则非常反对这种推断：“我从不觉得斯托曼有自闭症。他当初就是社交方面有点障碍，可我们这群人都这样。”

而李普曼则认为确有其事。她说了斯托曼婴儿时期的一些事情，佐证这个结论。自闭症患者的一个显著症状，就是对噪声和颜色异常敏感。李普曼记起了两个事情：“当时斯托曼还是个婴儿，我们带他去海边，每次去，在离海有两三个街区的时候，他就开始大哭。去了三次，我们才发现，原来是他觉得海浪的声音太吵了。”接着，她又说了斯托曼当时对颜色的反映：“斯托曼姥姥的头发是亮红色的，每次他姥姥抱起他的时候，他也会大哭大叫。”

这几天，李普曼读了一些关于自闭症的书。她觉得斯托曼的这些行为绝非巧合：“我确实觉得理查德当时有很多自闭症儿童的症状。我真后悔当初对这种病没什么了解。”

李普曼说，时间一久，斯托曼也逐渐开始适应了环境。七岁的时候，他开始喜欢坐在地铁的第一节车厢里，透过司机窗口看着地铁隧道，记下地铁的路线。这必须得习惯了地铁的噪音才行。“只有启动时候的噪音会让他不安。”李普曼说，“一启动，他就跟被谁打了一下似的。不过他后来也逐渐习惯了。”

李普曼回忆，斯托曼以前也和其他孩子一样，活泼快乐有生气。可家中一系列的变故，让他越来越内向，逐渐与人疏远。

第一件事就是斯托曼的父母离异。李普曼和斯托曼的父亲，丹尼尔·斯托曼都有所准备，打算慢慢把离异的事实告诉斯托曼，希望不对他产生太大影响。可结果依旧让斯托曼改变很多。李普曼回忆：“他一开始好像还没把这件事放在心上，可当我和他一起搬到新公寓的时候，他彻底意识到发生了什么。他的第一反应是：‘爸爸的那些家具哪去了？’”

在接下来的十年，每周工作日，斯托曼都住在母亲的公寓；休息日则要跑去皇后区，在爸爸家里住。辗转之间，他也见识了两种完全不同的抚养方式。这一切，让他从此决定不会自己抚养小孩。理查德的父亲曾是一名退伍的二战老兵，于 2001 年去世。每当说起他，理查德总是对他又尊敬又恼火。一方面，这是一位有担当的士兵，他曾学习法语，只为了更好地帮助盟友；另一方面，他又是个总会把事情变得残酷不堪的父亲。⁷

“我爸爸脾气很坏。”斯托曼说，“他虽然从不大吼大叫，但总能用一种冷嘲热讽的方式，去批评指责你。”

至于说起和母亲住在一起的日子，斯托曼的评价就不再那么模棱两可了：“那就跟天天在打仗似的。我最大的痛苦就是，别人好歹可以说句‘我想家了’，可对我来说，所谓的‘家’，根本就是个并不存在的空中楼阁。”

父母离异后的最初几年，斯托曼还可以把祖父母和外祖父母家当作一个避风港。可他 10 岁左右的时候，几位老人相继去世，斯托曼因此受到了巨大的打击。“以前还能去爷爷奶奶，或者姥姥姥爷家，能觉得有人疼我，有人爱我。上大学之前，那是唯一一个让我觉得如此温暖的地方。”

李普曼把祖父母和外祖父母去世列为第二个沉重打击理查德的家庭变故。“他当时悲痛欲绝”。他和几位老人关系特别好。他们去世之前，斯托曼特别开朗外向，都属于远近一带的‘孩子王’的角色。他们一走，斯托曼一下子变得极度消沉。”

对于斯托曼来讲，这种消沉则是一种躲避悲痛的行为。他把自己的青少年时期形容为“不寒而栗”。他说，就好比一个聋子，被扔到了一群音乐爱好者中间，眼睁睁看着他们聚在一起，谈音论乐。

“我总有一种感觉，觉得无法理解周围人在说什么。”斯托曼回忆当初，说着自己如何被悲痛感包围以至无法与外界沟通，“我明白他们说的每个词，可总觉得在谈话背后，隐藏着我无法理解的东西。我不能理解别人为什么对旁人的话题如此感兴趣。”

悲伤往往会唤起青少年极度的自我意识，会强烈展示自己的个性与与众不同。那会儿，别的孩子都喜欢把头发留长，斯托曼则一头短发；全部青少年都开始迷恋摇滚乐，斯托曼则钟情与古典乐。他热衷科幻小说，喜欢读《Mad》杂志，爱看深夜电视节目。他发展出了这种特立独行的个性，以此回避与这个世界的格格不入。

李普曼回忆起斯托曼少年时的个性：“开玩笑，俏皮话，双关语。只要你在桌上抛出一句话，他肯定能借此抖个包袱耍个宝。”

在外面，斯托曼则省下了这些笑话，只留着和那些鼓励他追求兴趣的成年人沟通。其中一位，是 12 岁那年科学夏令营的顾问。他曾送给了斯托曼一份打印版的 IBM 7094 计算机的手册。作为一个迷恋数字和科学的少年，这份手册仿佛是来自上帝的礼物。⁸那年暑假快结束的时候，斯托曼已经根据这份 7094 的手册，自己在纸上写了一份程序。他迫不及待地期待着有一天能在真正的机器上跑跑自己的程序。

可惜那会还是二十世纪六十年代，十几年之后才会出现第一台个人计算机。至于斯托曼第一次用上计算机，则是在这次夏令营之后几年。11 年级结束的那个暑假⁹，在那个暑假，斯托曼加入了 IBM 纽约科学研究中心，写下了他有生以来第一个跑在计算机上的程序——用 PL/I 语言编写的预处理器。“我开始是用 PL/I 写的，后来程序越写越大，大到计算机里放不下了，于是就用汇编语言重写了一遍。”斯托曼回忆说。

高中毕业后的暑假，IBM 纽约科学研究中心给他提供了一个工作机会，任务是用 Fortran 语言编写一个数值分析程序。他只用了短短几周时间就完成了这个程序，但是他发现 Fortran 语言很不合他的胃口，他发誓再也不用这种语言开发任何东西了。于是，他在暑假剩余的时间中用 APL 开发一个文本编辑器。

与此同时，斯托曼又从洛克菲勒大学生物系拿到了一份实验员的工作。尽管斯托曼已经确定未来将向数学或物理方向发展，可他严密的思维还是打动了这个生物实验室的主任。斯托曼进入大学没几年，李普曼还收到了一个电话。“那是洛克菲勒大学的教授，他想知道斯托曼

近况如何。当知道如今斯托曼在搞计算机的时候，他很吃惊。这位教授始终觉得斯托曼非常适合成为一个优秀的生物学家。”

斯托曼的理性分析能力，也惹上了哥伦比亚大学的教授。参加哥伦比亚大学科学之星计划的时候，布莱德巴特回忆说：“一般情况下，每隔一两个小时，斯托曼都能挑出讲课老师的一个小疏忽。而且他总会当面把这个问题指出来。就因为这个，我们当时都挺崇拜他的；可也因此让我们都疏远了他。”

听了布莱德巴特的回忆，斯托曼笑道：“我当时可能确实有点不太会讨好别人。我总觉得在这些老师教授身上，能看到一种好学求进的精神。可在周围的学生身上，很难看到这一点。至少在学生身上看到的不太一样。”

每周六都和这群优秀学生处在一起，斯托曼也难免会跟他们一样，考虑以后的计划。哥伦比亚科学之星计划种的学生，大多数都会把未来的大学瞄准在两所学校上：哈佛大学和麻省理工学院。李普曼知道自己的孩子希望能去所常春藤学校，她也必须要严肃对待这件事情。可看看当下的斯托曼，正值 15 岁，还天天和学校老师对着干。虽然上一年，斯托曼的美国历史、化学、法语、代数全都是 A，可他还在继续拒绝写作文。在全 A 的成绩中，只有英语一门拿了个 F。这种特立独行如果放在麻省理工学院，也许最多是笑了笑了事，可倘若要去哈佛，则是个很严重的问题。

在斯托曼 11 年级的时候，李普曼曾咨询过心理医生。医生一下子注意到斯托曼拒绝写作文的行为，以及他和老师之间的矛盾。斯托曼显然是聪明过人，智商上完全胜任哈佛的任何课程。可他到底能不能静下心来，踏踏实实坐在课堂里，完成每学期要求的期末论文呢？心理医生建议先让他在纽约市的公立学校读一年 12 年级，那里的英语课程也会要求期末论文。看看他究竟能否坚持下来。于是，11 年级结束后，斯托曼进入了路易斯·D·布兰德斯高中的暑期班。这是一所位于 88 街的公立学校。在这所学校里，很多文科和艺术类课程都是必修课。斯托曼再也没法像以前那样躲过这些课程了。

一个夏天过去，斯托曼终于融入了纽约市的主流公立学校。与每周六的科学之星计划的学习相比，让他踏踏实实坐在公立学校的课堂里确实不容易。可他最终还是做到了，李普曼很是骄傲。

“他被迫学会了妥协，但他最终成功了。”李普曼说，“我居然只被老师们请去了一次，这简直算是奇迹了。当时微积分老师把我请过去了，因为理查德经常在上課期间打断他讲课。我问老师他因为什么打断讲课。他说理查德总是会说老师的证明不够严谨。我就问老师：‘那他说对了吗？’老师说：‘对是对了，可我不可能给全班讲这些证明，那样的话其他学生都会听不懂的。’”

第一个学期过去了，一切皆见了分晓。英语课 96 分，一雪前耻。其他课程，包括美国历史，大学先修课程的微积分，微生物学，斯托曼依旧遥遥领先。物理课斯托曼则满分过关。尽管斯托曼依旧不合群，但他终究是在布兰德斯高中完成了 11 个月的学习，并且在 789 班取得了全班第四的成绩。

学校以外的活动里，斯托曼表现得更加勤奋。每周他都会去洛克菲勒大学，完成实验员的工作；周六则一路避开反越战抗议者的队伍，去哥伦比亚大学参加科学之星计划的学习班。终于又有机会，科学之星计划的学生凑在一起，聊起未来将去哪所大学。

布莱德巴特回忆：“绝大部分学生都会去哈佛或者麻省理工学院。当然也有人去其他几所常春藤学校。大家聊着说着，斯托曼始终没说话。这时候有个人跳出来，问斯托曼究竟去哪所大学。”

三十多年过去了，布莱德巴特依旧清晰地记得那一刻。“哈佛大学”几个字从斯托曼口中一出，人群一时没了声音。一切都好像事先排练好的一样，坐在角落中的斯托曼露出了一丝微笑。

布莱德巴特说：“那微笑好像在说：‘没错，你们还没把我甩开呢。’”

Endnotes

¹本章内容的主要背景资料来源于 Michael Gross 的一篇采访《Richard Stallman: High School Misfit, Symbol of Free Software, MacArthur-Certified Genius》，Gross 是 1999 年出版的《*Talking About My Generation*》一书的作者，这本书中记录了很多对著名人士的访谈。虽然对于斯托曼的访谈最终没有收录到这本书中，但是 Gross 把它作为补充材料在发布在了这本书的网站上。这篇访谈的网址迁移过很多次，目前根据读者的反馈，你可以在这个地址读到这篇文章：<http://www.mgross.com/MoreThgsChng/interviews/stallman1.html>

²RMS：我不记得我跟她说过这些。我只能说我现在很不赞同这里描述的观点。在我十几岁时，我遇到的问题跟其他同龄人并不相同。

³同上。

⁴朱迪·丝蒂特，《多伦多之星》，2000 年 10 月 9 日，C03 版。

他对自由软件和社会合作的愿景明显与他孤立的个人生活格格不入。他的古怪与加拿大钢琴家格连·古尔德非常相似，他们都是非常聪明、善于表达的人，但是同时也都很孤独。斯托曼觉得自己在某种程度上被自闭症所折磨，他总是觉得自己与其它人打交道非常困难。

⁵史蒂夫·西尔贝曼，《天才综合症》，《连线》2001 年 12 月刊。http://www.wired.com/wired/archive/9.12/aspergers_pr.html

⁶参考 John Ratey 与 Catherine Johnson，“Shadow Syndromes”。

⁷很遗憾，我最终没能采访到丹尼尔·斯托曼。本书写作之初，理查德·斯托曼曾告诉我他父亲患上了老年痴呆症。2001 年，我继续写本书的时候，得知丹尼尔·斯托曼已经不在人间。

⁸作为一个无神论者，斯托曼恐怕不喜欢这种描述。不过无论如何，这份礼物肯定是斯托曼喜欢的。参见 Gross 1999 年的那篇文章中的描述：“我一听说有计算机这种东西，就迫不及待地想要看看玩玩。”

⁹译者注：美国小学 5 年，初中 3 年，高中 4 年。11 年级相当于中国的高二。但和中国不同的是，美国学生一般都会在 11 年级把 SAT 等大学入学考试考完，在 12 年级提前学习一些大学课程，即所谓的大学先修课程（Advanced Placement）。

Chapter 4

逆天行道，选举上帝

尽管理查德·斯托曼和母亲的关系比较紧张，但他还是继承了母亲的一个特质：对激进政治极度热情。

可这个特质要在十几年之后才在斯托曼身上显现出来。他早年的生活，用他现在的话说，是与“政治绝缘”的¹。正如艾森豪威尔时期的大多数美国家庭一样，五十年代，斯托曼一家试图重拾二战以前的那份宁静、安逸的正常生活。

“理查德的父亲和我虽然都是民主党人，可我们觉得那段时间的生活还算过得去。”李普曼回忆着当年和理查德父亲住在纽约市皇后区的日子，“我们当时并没有参与太多的当地或者全国的政治活动。”

然而，五十年代末期，李普曼的离异打破了这一切。她要重新搬回纽约市曼哈顿区，这不仅仅是改个住址，它更意味着一份全新的独立生活的开始，更意味着重回喧嚣的大都市生活。

“关于政治活动，我最初的体验是在离婚期间。当时我走进皇后区公共图书馆，却发现里面只有一本关于离婚的书。那里当时被天主教严格控制着，至少在我们住的阿母赫斯特区是这样的。这是我第一次

意识到，我们的日常生活，被周围的各种权威力量悄悄地控制着。”李普曼说。

重回童年居住过的曼哈顿上西城，李普曼被周围的变化震惊了。十五年前，她离家去亨特学院读书。如今，由于战后住房资源紧张，当地居民对高楼有大量需求，把几个街区变成了一个政治决斗场，把人们分成了对立的两组。一组人，主要是政客和商人，他们希望把这一片拆迁扩建，以应付越来越多的白领移居到这里；另一组人，主要是来自爱尔兰或波多黎各的房客，他们都不富裕，所幸已经在周围找到了便宜舒适的住房。

一开始，李普曼不知道站到哪一边。作为一个新来的住户，她觉得的确有必要扩建。可作为一个拿着微薄收入的单亲母亲，她和那些房客有着一样的忧虑，觉得周围越来越多的项目，都是为有钱人开发的。心中忿忿不平，李普曼开始寻找机会，来和庞大的政治机器做斗争，避免让政客把上西城变成上东城那样，只供有钱人吃喝玩乐。

李普曼第一次去民主党总部是在 1958 年。当时，她希望为理查德找个日托所，以便在自己工作的时候，有人照顾理查德。在本市的低收入人员帮助中心里，她一下子被那里的环境吓到了。“我就记得当时那股酸臭的坏牛奶味儿，黑洞洞的走廊，还有那么一丁点的救助物资。我以前是护士学校的老师，那里的环境和这里反差太大了。我们看了一眼那个日托所的房间，就立马走人了。那环境太让我恶心。”

这次造访民主党总部，让李普曼很失望。“大家说的一点也没错，乌烟瘴气，不是个干净地方。”李普曼说，这是她第一次意识到，党内腐败也许才是一切症结所在，大家也许才因此鄙视甚至仇视穷人。李普曼没有再踏入民主党总部一步。她看到当地众多俱乐部，都志在促进民主党的党内改革。李普曼加入了其中的一个俱乐部：伍德罗·威尔逊/FDR 民主党改革俱乐部，从此与坦慕尼协会余党²作斗争。李普曼开始出现在俱乐部内部的例行会议中，和市政委员会会议会上，以此她取得了更多话语权。

李普曼说：“我们的主要目标，是对抗以卡米思·德·萨皮奥 (Carmino De Sapio) 为首的坦慕尼协会和它的追随者³。我是市政委员会的一名代表委员。主要提议修建更多廉价住房，反对单纯修建舒适豪宅。”

这种参与在六十年代变成了更大的政治活动。1965 年，李普曼已经开始公开支持一些民主党的议会候选人，比如威廉·菲茨·赖安，他在各个民主党改革俱乐部的帮助下，进入议会，并成为当时第一批反对越战的议员。

不久之后，李普曼也开始公开反对美国涉足印度支那问题。“肯尼迪政府把军队送去越南开始，我就一直反对越战。从一开始，我就读各种来自前线的报道文章。很多文章都预测美国政府会因此深陷泥潭，我也非常赞同这个观点。”

这种反战之声充斥在理查德的家中。1967 年，李普曼再婚。她的新任丈夫莫里斯·李普曼是一名空军少校，越战开始他辞职以示反战。莫里斯的儿子，李普曼的继子，安德鲁·李普曼当时正在麻省理工学院读书。他暂时还可以申请延期入伍。可随着战事升级，战争时间一久，入伍时间到期，他还是要去服役。当战事延续到七十年代，理查德虽然年纪还小，可也不得不考虑入伍的问题。他要么选择去越南参战，要么选择到加拿大避开兵役。

李普曼说：“越南问题成了家里的重点，我们总是不停地讨论它：如果战争持续下去我们怎么办？要是理查德或者安德鲁收到征兵令怎么办。我们全家都反对这征兵令，更反对这场战争。我们从心里觉得这场战争是不道德的。”

对于理查德来说，越战给他带来的情感是复杂的：恐惧，不解，以及最终给他带来的一种政治上无能为力的感受。理查德如此地厌恶权威，他甚至无法忍受私立学校的威权制度。别说参军，哪怕就只让他想想军队里的训练，都会令他不寒而栗。

“我当时被吓坏了。可我也不知道能做什么，该怎么做。更没胆量去上街参加游行。”理查德如是说。1971 年，政府最终取消了学生延期

入伍的政策。抽签征兵⁴的结果让理查德很是担心：他3月18日的生日，征兵顺序中比较靠前。“我很难想象要移民到加大拿或者瑞典。要一个人离开过日子，我当时可不行。我根本不知道怎么照顾自己，我对自己在生活自理方面的能力非常不自信。”

理查德说，他至今依然感慨家人在公开场合的反战行为。他记得父亲车上有一个车贴，把美莱村屠杀⁵比作纳粹的大屠杀。这个车贴是父亲亲自制作的，他还做了很多这样的车贴，分给大家。理查德说，他很受父亲情绪的鼓舞。“我很崇拜他这么做。”理查德说，“可我没想过自己能做什么。当时我很害怕自己的一辈子就这样被征兵令毁了。”

可理查德也逐渐开始不喜欢反战运动的方向和调调。正如其他科学之星计划的学生一样，理查德每周末都会看到聚集在哥伦比亚大学的抗议群众⁶。最终，理查德形容说，各种非理智力量把反战运动变得与各种其他青年人的非理智活动一样可怕。一瞬间，理查德那个年级的姑娘们都不再崇拜披头士，改去崇拜各路反战领袖，如阿比·霍夫曼（Abbie Hoffman）和杰里·鲁宾（Jerry Rubin）。理查德，这个中学生里的异类，面对各种流行趋势已然有些应接不暇了，如今又来了“要做爱不作战”这种花哨口号，实在让他觉得甚是沮丧。他显然不喜欢战争，至少不喜欢这次在东南亚的战争。当然这也并不意味着就会有姑娘愿意与他一起共度春宵。

“我并不喜欢这种对抗主流的文化。”理查德坦言，“我不喜欢他们的音乐，不喜欢毒品。我可害怕毒品。我更厌恶他们那套反理性反知识的论调，厌恶他们对各种技术的偏见。因为无论怎么说，我还是喜欢计算机的。我还经常遇到各种没头脑的反美意见，这也让我很反感。有些人的头脑真的太过简单了，他们觉得如果要反对美国参加越战，就意味着支持北越政权。他们就天真到想不出另外一种可能。”

这样的评论标志着理查德自己政治思想的逐步成熟。对于理查德来说，政治上的信心和自信心成正比。到了1970，理查德已经在数学和科学以外的一些领域里树立起了自信。当然，在数学方面的信心和能力，作为基础，让理查德可以从纯逻辑的角度分析越战。这种分析

最终带来的结论，让理查德觉得，尽管反对越战，但不可否认，战争在维护自由和捍卫正义方面起到的积极作用。

八十年代，理查德已经不再沉默。他带着这份自信，在华盛顿市参加了维护堕胎权利的聚会。“我开始不满从前的自己，觉得不该像当初那样逃避游行抗议，这是我的权利，更是我的责任。”

1970 年，理查德离开了家，离开了每晚家中餐桌上的越战和政治讨论，去了哈佛大学。理查德回忆，从曼哈顿母亲的公寓，转到马萨诸塞州剑桥市的哈佛大学宿舍，对自己来说是一种“逃离”。可同龄人来看，他的这次大逃离并没有给他带来多少解放。

“他在哈佛的第一年很痛苦。”丹·柴斯，另外一位去了哈佛大学的科学之星的学生，回忆道，“简单的人际交往对于理查德来说都是非常困难的。可这种交往在哈佛又是无法避免的。哈佛校园就是个大社交场。”

为了让这种过渡更顺畅，理查德重新埋头在自己的老伙伴中间：数学和自然科学。与很多科学之星计划的成员一道，理查德轻松通过数学 55 的课前考试，获得了选修数学 55 的资格。所谓数学 55，是哈佛大学一门数学类课程的官方编号。它专门为未来数学家设计，以难度大而闻名全校。在这门课上，科学之星的成员凑成了一个小团体。“我们数学党的。”柴斯笑道：“跟科学之星比，哈佛不在话下。”

可要修成正果，理查德，柴斯和其他科学之星成员，必须要通过数学 55。这个课程把本该四年学完的数学知识，全部放在两个学期里。只有真正肯下功夫的人，才能啃下这块硬骨头。“那真是门让人神往的课。”戴维·哈伯特（David Harbater）回忆道。他曾是这门课上“数学党”的成员，如今已是宾夕法尼亚大学数学系教授。“很保守地说，几乎没有几个给大学新生设计的课程，能这么难，这么深，课业量也很少有这么大的。我经常给人这么形容这门课的难度：第二个学期，我们已经开始讲巴拿赫空间的微分几何了。一般人听了肯定瞪大眼睛，因为绝大多数人到了研究生第二年才开始说点巴拿赫空间的东西。”

那年这门课刚开始有 75 个学生，到第二学期结束，就只有 20 个学生还在坚持⁷。戴维·哈伯特说，这 20 个学生当中，“只有 10 个人真正听懂了课上讲的是什么。这 10 个人里，8 个人后来做了数学教授，另外有一个做了物理教授。”

“最后那一个，”戴维·哈伯特强调，“就是理查德·斯托曼。”

赛思·布莱德巴特也是数学 55 课上的学生，他记得当时理查德就显出了与众不同的特质。

“遇到一些事，他总是特别能坚持。当时，我们经常用一种错误的方法来解题。每次我们要定义一个函数的时候，我们都会先定义一个函数，然后证明这个函数是良构的。这么做本身是用错了术语。理查德第一次也犯了这个错。可后来，他都是先定义一个关系，然后证明这个关系是一个函数。其实证明的过程都一样，但他用了正确的符号和术语。而其他没有一个人这么做。他就是这么个人。”

数学 55 的课堂上，理查德又一次展现了他的聪明才智。布莱德巴特早早就承认了这一点。而柴斯好胜的个性则很难让他服输，但到了第二年，他还是不得不承认，理查德也许是整个课堂中最优秀的数学家。如今，柴斯已是亨特学院数学系的教授，他回忆：“在一堂实验分析的课上，理查德想出了一个关于复测度的证明，那个证明实际上是借鉴了变分法里的一些技巧。我是第一次看到有人能自己想出来，用这么漂亮的方法解决这个问题。”

对于柴斯来说，这的确是个恼人的时刻。他就好像一只小鸟飞到了透明玻璃前，他要花些时间，才会不甘心地意识到，有些地方，虽然看得到，却未必是自己能力所及的。

“这恐怕就是数学之美吧。”柴斯说：“你不必需要成为一流的数学家，就可以欣赏数学天才的作品。我就处在这么一个尴尬的地方。我算不上一流数学家。如果理查德当初选择做数学家的话，他肯定是一流的。”⁸

有得就必然有失，理查德在课堂上的得意，换不来社交上的成功。其他数学党的成员一般都会凑在一起讨论数学 55 的作业，而理查德

则自己一个人应付作业题。住宿问题上理查德也坚持如此。在哈佛大学住宿申请的表格上，理查德清晰地说出了自己对室友的要求：“我说我希望能有个几乎看不见，听不见，摸不着的室友。”宿舍管理部门这次倒是难得如此识相，竟然接受了理查德的要求，在他入学的第一年，给他安排了一个单人间。

布莱德巴特是数学党成员中，唯一一个在大一期间和理查德同住一个宿舍楼的。他回忆，理查德当时的确慢慢地学着如何和别人沟通交往。他记得，当时其他几个宿舍的学生，都被理查德严密的逻辑分析能力感染，都喜欢跟他凑在宿舍楼的大厅或者餐厅里，谈天说地。

布莱德巴特说：“我们凑一起经常会扯上一些异想天开的事情，比如如何解决各种世界问题，或者某种东西如果成真了会怎么样。好比说，有人发明了一种长生不老药会如何。你该怎么办？这东西的政治影响会是什么？如果说每人都吃了这个药，最后谁都不会老，最后人越来越多，到头来还是会因为资源困乏，大家都得死掉。老是不会老了，但终究也会死。可如果限制发布这个药，好比你可以说，只有现在活着的人能拿到这个药，新生儿不能吃长生不老药，那最后世界上就会被分出一批下等人。理查德总能比大家更早看出各种决策的优劣。”

理查德至今还清晰地记得当时的讨论：“我总是对长生不老的话题感兴趣。我们要是能看到 200 年之后，世界是什么样子，那会如何？”出于好奇，理查德找了几个朋友，问他们愿意不愿意长生不老：“我很惊讶，不少人觉得长生不老并非什么好事情。”很多人说，死亡从某种意义上讲是个好事，因为一直活下去的话生命会显得越来越没有意义。而苍老的好处则是让人们能逐渐意识到死亡的到来。

尽管众人已经公认，理查德会是个一流数学家，也是个非正式的辩论能手，可他自己却一直避免参加各种有明确排名或分界线的竞赛。布莱德巴特清晰地记得，理查德大一的时候，大家眼看着他回避参加帕特南竞赛（Putnam Competition）。帕特南竞赛是专门针对美国和加拿大数学系的本科生的竞赛。参赛者借此可以知道自己的数学水平，更重要的，比赛的结果经常被各大院校当作选拔研究生和博士生的依

据。按照学生中间的流言来看，如果拿到好成绩，就可以去任何一所大学读研究生或博士，而且是免学费，全额奖学金。当然，这也包括哈佛大学。

和数学 55 一样，帕特南竞赛也是各残酷的竞技场。两场考试，一共六个小时。这比赛是铁了心要把学生分出三六九等。布莱德巴特，这位参加了科学之星计划，也上了数学 55 的学生，依然觉得这个竞赛是他参加过的最难的考试。“就跟你这么说吧，满分 120，我第一年的时候分数是 30 几分。就这分数，我还能排上第 101 名。”

可让人吃惊的是，理查德，这位全班数学最好的学生，竟然不参加这个比赛。布莱德巴特记得当时他们听到这个消息之后，几个人把理查德围在墙角，让他解释为什么不参加。“他说他怕做得不好。”布莱德巴特回忆。

一听这话，布莱德巴特和其他几个学生迅速地写下了几个帕特南竞赛得题目，扔给理查德，让他做。“他全都做对了。我当时就觉得，所谓‘做得不好’，对他来说也许就是拿个亚军，或者做错一道题。”

理查德对这个事情的记忆则略有出入。“我确实记得他们给我题目了，可我好像记得我就做出来一道，反正肯定没全做对。”无论如何，理查德承认，正如布莱德巴特说的，他当时的确是出于害怕才不参加这竞赛的。理查德只是在班上指出同学或者老师的错误。但他却不喜欢，甚至害怕参加任何正面竞争。结果，他就总是避开类似的比赛。

理查德说：“同样的原因，我也不喜欢下棋。每次我下棋的时候，我都害怕那种一步走错，满盘皆输的情况。而这种恐惧，最后总是成真。”于是，理查德也就只能靠避免下棋，来解决这个问题。

究竟是不是因为这种恐惧，才没让理查德成为数学家，我们不得而知。但无论如何，大一结束的时候，理查德已经有了新的兴趣：编程。这兴趣从理查德中学的时候就逐渐形成，但潜伏了很久，最终在大一后期成为了他的一个主要兴趣，让他投入了自己的全部热情。其他数学系的学生都靠上艺术和历史课来放松，理查德则跑去机房缓解压力。

当年第一次在 IBM 纽约科学中心编程的经历，诱使着理查德去了解更多。他说：“在哈佛学习快一年的时候，我开始跑去哈佛的几个计算机实验室，看看他们那里有什么新东西。刚到那里，我就问他们能不能给我一份使用手册。”把这些使用手册带回家，理查德开始自习阅读，比较各个机器这间的差别和联系。

第一年快结束的时候，理查德有天听说在麻省理工学院，有个特殊的实验室。那个实验室就在校区旁边的技术广场大厦九层。传说，这个实验室是专门为顶尖的人工智能研究而设立，里面各种高级计算机和软件。

理查德被这一切吸引住了，决定亲自前往一探究竟。

从哈佛大学到那里并不远，走路大概三公里，坐地铁只要十分钟。可等到了那里，理查德才发现，麻省理工学院和哈佛大学，简直就像两个世界。麻省理工学院里真可说是“五步一楼，十步一阁”。各个建筑纠缠在一起；相比之下，哈佛大学则显得明快宽敞。两所大学，一张一弛，一阴一阳。这不仅体现在建筑风格上，在校风学风上也是如此。麻省理工学院的学生，多是中学里的怪才，他们好开玩笑，喜欢恶作剧；而哈佛大学的学生，则更多是家有深厚背景的孩子，或是从小就有政治抱负的青年。

这种学校风格同样延伸到计算机实验室中。与哈佛的各个计算机实验室不同，麻省理工大学的这个人工智能实验室，没有门卫；没有拿着小本子，记下等候使用终端的人，给他们排队的人；也没有贴着“严禁触摸”标识的那种氛围。到了那里，理查德就看到了几个空着没人用的终端，和几个机械臂，提醒着人们这里正在做着人工智能方面的试验。理查德找到一位在这里上班的雇员，问他能不能送自己一份使用手册。理查德回忆：“那人说他手头确实有几份闲置的手册，可好多东西都没写在手册里。怎么说，他们也是黑客。”理查德笑道，暗指黑客们往往会直奔下一个项目，而不去把之前的项目记录在文档里。

在对方的帮助下，理查德最终得到了一份比手册更好的东西：一个在实验室的工作职位⁹。他的第一个工作任务，是在 PDP-10 上写一

个 PDP-11 的模拟器。于是，他第二周又来到人工智能实验室，找了个终端，开始写代码。

回想起来，理查德觉得，人工智能实验室把一份工作“随便”扔给一个素不相识的外校学生这件事，并没什么不妥。“当年都是这样，如今也是如此。我见一个人，他要是很优秀，我就雇他。等什么呢？很多刻板的人喜欢在每件事上都安插各种官僚流程。这实在太愚蠢。要是一个人真的很优秀，他就不会花时间准备各种冗长拖沓的面试。他应该踏踏实实坐电脑前头写代码。”

要想了解什么是“刻板”和“官僚流程”，理查德说只要去哈佛的计算机实验室看看就行了。在那里，使用终端的先后顺序，是按照年级排序的。作为一个本科生，理查德经常要等好几个小时才能用上计算机。等待本身并不是件难事，但却是个很无聊的事情。尤其是你明明知道，在各个教授的办公室里，都有一个能用的计算机终端，可每个办公室的房门都是紧锁的，你就只能在那里按部就班地等着用那几台公共终端。在理查德看来，这实在是暴殄天物。尽管他后来也会偶尔去哈佛大学的计算机实验室，但他显然更喜欢人工智能实验室的平等环境。他说：“在那里总算能呼吸到新鲜空气了。人工智能实验室里，人们更关心手头的工作，而不是各自的地位状态。”

理查德很快了解到，人工智能实验室这种先到先用的原则，很大一部分要归功于当年一批警觉的小团体。其中很多人，参与了当年国防部资助的 MAC 项目（Project MAC）。在这个项目里，他们开发了第一个分时操作系统。而这个小组中的有些人，如今已经在世界计算机史册上留了名。包括实验室中的 Lisp 专家理查德·格林布拉特（Richard Greenblatt），他设计了国际象棋程序 MacHack，让休伯特·德莱弗斯（Hubert Dreyfus）关于人工智能的观点受到重创¹⁰；还有著名的杰拉尔德·萨斯曼（Gerald Sussman），他曾设计 HACKER 这个程序，利用机器人解决堆垛问题；也有著名的数学怪才比尔·高斯伯（Bill Gosper），他当年发现了生命游戏（LIFE game）中的模式并因此获奖¹¹。

这批小团体里的人把自己称作“黑客”。当然将来，他们也会把理查德归类为“黑客”。在成为“黑客”的过程中，理查德被逐渐地引进黑客文化的小圈子里。这群人，会痴迷于不断挖掘计算机的极限。他们遇到挑战，可以一连 36 个小时坐在终端显示器前面工作。最重要的，他们对计算机和各种相关信息都有无尽的需求。他们会公开大谈如何用计算机和软件改变世界。理查德也逐渐学到，黑客们会本能地蔑视任何阻止他们实现这一伟大梦想的障碍和困难。这些障碍之中，最为重要的有以下几个方面：一是低劣的软件，二是各种学术官僚主义，三是自私自利的垄断的行为。

理查德在这里学到的东西里，还包括黑客们解决问题的各种新技巧和相关的轶事。这其中，就包括黑客们如何打开教授们紧锁的办公室门，“解放”被囚禁的计算机终端。哈佛大学的计算机终端每天都被娇生惯养；相比之下，麻省理工学院的教授们更了解如何应付有限的终端资源。要是哪位教授，把某个终端锁在自己的办公室里过夜，黑客们就会想法把终端重新搞到手。之后，还会大摇大摆走到教授面前，劝诫他不要如此对待实验室的同僚。黑客们的做法多种多样，有些人会直接撬锁，他们自称这是“黑”锁术；有些人则会把天花板打开，爬到天花板上，从天花板和房顶中间的通风空隙中爬入教授的办公室；在大厦的第九层，地板和楼层地面中间有空隙，本来是用于铺设机房的电线，而黑客们则会撬开地板，从下面溜进办公室。理查德说：“我当初还给他们展示过一架手推车，上面放着一个实心的圆柱形金属，这东西后来就曾用来撞开办公室的门¹²。”

黑客们如此坚持，使得教授们很难起私心而阻碍整个实验室的工作。当然黑客们也没有无视教授们的个人需求，只是教授的需求不能影响到别人。比如，有些教授有时候会辩称称，自己办公室里有重要的私人物品，所以必须要锁门。黑客们则会回应：“你锁上办公室的门没问题，但是别把实验室的终端也锁在里面。”

虽然在人数上，黑客们处于劣势，但黑客的文化在不断追求、不断进步的计算机领域还是占了上风。这些黑客，很多是实验室的成员，

或者是学生。实验室的计算机上，有些硬件就是他们设计的；而几乎所有软件，都是这些黑客写的。他们不仅创造了整个系统，还维护着这个系统的运行。他们的工作至关重要，因此他们拒绝来自任何方面的压抑。他们会根据其他用户的需求写程序，也会同时维护着自己的个人项目。有时候，这些个人项目会越来越庞大，甚至影响深远。黑客们每天为计算机系统增砖添瓦，并以此为乐。

这种工作的结果之一，就是运行在实验室核心的 PDP-10 计算机上的操作系统。它被称做 ITS，全称“非兼容型分时系统”(Incompatible Time Sharing system)。这个系统从头到脚都体现着黑客的文化。它的诞生，本来是用来抗议 MAC 项目的操作系统：CTSS，即“兼容型分时系统”(Compatible Time Sharing System)。这种抗议从名字中的“非兼容型”几个字上就能看出来。当时，黑客们觉得 CTSS 的很多设计太过严苛，因为它限制了程序员改进现有程序的行为。沿袭着黑客们一贯以来的作风，设计 ITS 的决定本身就是政治上正确的。CTSS 本身是为 IBM 7094 设计的，ITS 则是为 PDP-6 设计。为了让黑客们更好地写出自己的操作系统，人工智能实验室的管理员保证可以让黑客们使用 PDP-6 计算机。在分封制度盛行的学术圈，这策略确实奏效。虽然 PDP-6 以前是和其他几个系共用的，可后来最终还是归了人工智能的研究人员。理查德来之前的那段时间，黑客们终于利用 ITS 和这台 PDP-6 做基础，正式宣告自己独立于 MAC 项目¹³。

1971 年，ITS 被安装在更新一代的 PDP-10 上。那台老的 PDP-6 则被单独拿出来以做他用。人工智能实验室这台 PDP-10 有大约 1MB 的内存，在当时算是非常大的了。到七十年代末，又给它的内存加了一倍。MAC 项目也为实验室带来了两台 PDP-10，它们都被放在大厦九楼，运行着 ITS 系统。PDP-10 本身没有内存分页机制。为此，对硬件感兴趣的黑客们为这三台 PDP-10 设计了内存分页模块¹⁴。

作为一名黑客学徒，理查德一下子被 ITS 迷住了。尽管 ITS 会吓跑一些门外汉，可它却比其他商业操作系统提早几年实现了很多特性，

甚至有些特性是今天很多商业操作系统都没有提供的。这包括多任务系统，在线调试任何运行中的程序，全屏编辑等等。

理查德回忆：“ITS 内部有一套非常优雅的机制，可以让一个程序实时监测另一个运行中的程序。你可以获得另外某个程序的各种信息，内容清晰，接口明确。”这个功能不仅对调试程序有很大帮助，而且可以轻松地开关和控制别的程序。

另外一个功能，就是允许一个程序暂停另一个程序，让那个程序停在指定的两个指令之间。在别的操作系统里，类似的功能只能让程序停留在某个系统调用中，把程序停在这个状态，会有很多内部隐藏状态用户都看不到。而在 ITS 里，这一功能则可以保证一条指令一条指令地监视另一个程序运行，并且能做到停顿之后，一样能继续运行。

理查德说：“如果你说：‘把这个任务暂停’，那它肯定会停在用户态。而且会停在两条用户态的指令中间。任何运行过程中的变量状态，都会保留。你再说：‘继续跑它’，它就真的会继续跑下去，不会因为暂停而搞乱程序，导致前后不一致。还不止如此，你还能把某个变量或者哪个状态给修改了。让它继续跑，然后再把它改回去，这程序就能和当初一样。所有状态都是用户可见的，没有什么隐藏状态。”

从 1971 年开始，理查德每周都会定期跑去人工智能实验室。周日到周五，理查德会呆在哈佛大学。一到了周五下午，他就坐上地铁，直奔麻省理工学院，在那里度过整个周末。他一般会在晚饭时间赶到，然后约上五六个黑客，一起去吃中餐。他们跳进一辆旧车上，穿过哈佛大桥，直奔波士顿。接下来的四个小时里，他们凑在一起就开始谈天说地。话题从 ITS 系统，一直到汉语和象形文字的内在逻辑。晚饭结束，这群人再回到麻省理工，写代码一直到天亮。期间也许会在半夜三点多的时候，再跑去唐人街一趟。

第二天上午，理查德也许会继续折腾代码，也有可能找个沙发睡上一觉。醒来之后，他会继续写代码，找点中餐吃吃，然后回到哈佛。有时候，他会再多呆一天，等到周日再回去。这些中餐不仅美味，而且比哈佛大学食堂的菜内容丰富，营养全面。吃上一顿能管一天。所

以理查德他们一般一天也就吃一顿饭。（对于他来说，早餐一般不在考虑范围之内。一方面，早餐中的食物他不喜欢；另一方面，早餐时间他一般都会睡大觉。）

作为一个中学时期很少有社交活动的宅男，如今和这么多跟自己类似的人混在一起，确实让理查德甚是陶醉。他们这些人，都对计算机感兴趣，都喜欢科幻小说，喜欢吃中餐。“我记得很多次，我们从唐人街回来，在车上看日出。”十五年后，在瑞典皇家技术研究所的一次演讲上，理查德回想当年，一切依旧历历在目：“欣赏日出真的是个很享受的事情。那是一天之中最平静的时刻。也是一天之中上床睡觉的好时候。倘若这时候徒步走回家，伴着几缕阳光，耳畔回响着鸟鸣，再想想你整整一夜完成的工作，你会有种由内而外地舒适和满足感。”¹⁵

理查德和这群黑客呆得越来越久，他也逐渐开始了解和接纳黑客们的世界观。曾经他坚持个人自由，如今，在这基础之上，又注入了公共责任的概念。当其他人违反了这种公共准则，理查德会毫不留情地当面指出。在麻省理工学院的第一年，理查德也曾想法打开教授办公室的门，拿回属于实验室的计算机终端。作为一个黑客，理查德也贡献了自己的开门技巧。以前大家一直用着一个开门的方法，据说是格林布拉特（Greenblatt）最先使用的。这个方法要求用到一条硬的电线，把这条电线折几个弯，从一盒磁带里抽出磁条，拴在弯过的电线一头，再把磁条做成一个绳套。开门的时候，先从门下的缝隙里把带磁条一头的电线躺着穿过门缝，等电线和磁条都进了门，再把电线立起来。来回晃几下电线，直到磁条做的绳套能套在门把手上。一旦套紧，黑客就可以把电线往后拉，这样，门就从里面被打开了。

当理查德自己亲自尝试这个方法的时候，他发现这法子操作起来难度不小。要让磁条套上门把手，可是个体力活。而之前要怎么把电线折出几个弯，也是有学问的。为此，理查德开始另寻他法：他开始想法把天花板上的板子移开，然后爬进天花板和房顶中间的通风空隙中。爬到办公室上方，跳到桌子上，一切就大功告成了。这法子一直都行得通，可这一路，总会沾上一身让人痒痒的玻璃纤维。理查德开

始考虑怎么解决这个问题。他想到，如果不把电线从门下缝隙滑过去，而是从天花板上悬下去开门，会不会可以呢？

理查德自己开始尝试。他并没有用电线，而是把一盒磁带里的磁条拿出来，拿着两端，向下悬成一个 U 形，把磁条有粘性的部分冲上。移开天花板上的两块板子，站到高处，把 U 形的磁条从通风道甩到房间里。磁条会从天花板上垂下去，晃悠几下，把 U 形磁条套在门把手上。接着把磁条往上一拉，门就顺势打开了。就这样，理查德又为黑客开门技巧做了贡献。

“有时候，门锁打开后你还得踮两脚才能把门弄开。”理查德回忆起这个方法的一点瑕疵：“拉磁条的时候你得小心点，一般是站在一个椅子上，下面还要再垫个桌子。这时候保持平衡就很重要。”

这些活动从一个方面反映出理查德期望表达自己的政治诉求。人工智能实验室的实干精神，让理查德打破了中学以来的沉默。打开办公室的门和上街游行虽然不可同日而语，但从效率的角度分析，与游行比起来，它快速有效，能直接解决问题。

在哈佛大学的最后一年，理查德把人工智能实验室的那套古灵精怪的行事理念用到了哈佛的校园里。

“他跟你说那条蛇了吗？”他母亲在一次采访中突然问起，“他还有他宿舍那边的几个人一起，把一条蛇拿去参加了学生会选举。而且最后那条蛇还拿到不少选票。”

根据理查德的回忆，那条蛇实际是拿去参加了卡瑞尔楼的学生干部竞选。那是理查德住的宿舍楼，并非全校范围内的学生竞选。理查德的确记得当时那条蛇拿到了不少选票，但很大一个原因，是因为这条蛇的姓和它主人一样。“人们也许以为是给这条蛇的主人在投票。”理查德说，“我们的选举海报上写那条蛇是个很有‘潜力’的候选人。因为竞选开始前几周，它刚刚‘潜’到通风口里，我们好几个人极尽全力”也没能找到它。”

理查德和几个朋友还提名了楼管的三岁儿子。“给他选举时候提出的政治纲领，就是七岁强制退休。”不过，理查德的这些恶作剧与麻省

理工学院里的选举恶作剧比起来，实在是小巫见大巫了。在麻省理工学院最成功的一次选举恶作剧，是提名一只名叫伍德斯托克的猫参加全校学生委员会的选举。最后这只猫得到的选票比许多候选人的选票都多。“他们最后也没有宣布伍德斯托克到底得了多少票。他们把这些选票当作了废票。可选举结果一出，大家看到了一大堆的废票，从废票的数量上看，伍德斯托克应该是可以赢得这次选举了。几年之后，伍德斯托克据说是在马路上被一辆车压死了。”理查德如此回忆道。他说他当年和伍德斯托克的选举没有一点关系，“可我很崇拜这事¹⁶。”

在人工智能实验室，理查德的政治活动更加积极，产生的矛盾也更尖锐。在七十年代，学校的一些教授和管理员一度发出通知，说即将停止在实验室的计算机上运行 ITS，也将停止支持很多黑客们喜欢的设计。ITS 允许任何人坐在终端前，做任何事情。包括把整个计算机关机。这在当年，可是个大事。因为很多人都会通过终端使用同一台计算机，关机意味着其他人也必须暂停手上的工作。不过，如果别人关机，你会收到即将关机的通知。要是你手头的工作还没完成，你也可以取消这次关机操作。到了七十年代中期，一些教授开始对文件的安全保护提出要求，要求他们能够控制哪些用户才能访问自己的文件。可人工智能实验室的这些黑客们，还是坚持要维护一个没有控制的系统。

于是，理查德对这些安全控制特性从道德和实践上给予了否定。从道德上说，人工智能实验室的这个圈子里，大家都有着开放和信任的传统。从实践上看，ITS 系统本身也是为了开发和协作而设计的，并非要控制哪些用户。任何与这初衷相违背的特性都需要额外的精力来开发维护，完全没有必要。为了让安全特性更难实现，理查德把文件描述符的最后一点富裕空间，用来存储用户的最后修改时间。最后修改时间是个很重要的信息，谁也不想把它去掉。可这么一来，整个文件描述符也就没有任何地方来存储安全信息了。

“设计 ITS 的黑客们认为，文件访问控制这个特性，是为自私的系统管理员设计的，这样他们这些管理员就可以凌驾在任何用户之上。”

理查德事后解释：“黑客们可不希望有谁被任何人控制约束，所以他们不会实现这样的功能。这么做的结果是，一旦系统出了毛病，任何人都能赶在第一时间修复它。”（因为没有访问控制功能去阻止你去访问任何文件。）¹⁷

这种警觉，让黑客们得以在人工智能实验室营造出一个没有管制的世界。可在旁边的麻省理工学院计算机科学实验室里，对安全敏感的教授们则取得了胜利。动态建模组最先在 1977 年在系统中使用了密码登陆。又一次，斯托曼挺身而出，纠正了这种自己看来“不道德”的行为。他先获得了密码登录系统代码的访问权限，然后写了一个程序，解密已经加密存储的密码。然后他开始给系统上的每个用户发邮件，劝说他们放弃使用密码。这封邮件大概是这样的：

我看到你用“starfish”做密码了，我建议你把密码直接改成“回车”，我用的就是这个密码。这密码简单好记，而且可以让你忘了密码和安全这些繁杂琐事。

所谓用“回车”做密码，就等同于没有密码。这个密码让其他用户也可以访问自己的帐号。这和以前没有密码的日子没什么分别。这就是理查德和黑客们要表达的：他们拒绝任何光鲜亮丽的枷锁，他们会嘲弄任何企图使用枷锁的想法。他们知道，这种安全保护能力很弱的系统，不可能防范任何入侵者。可这无所谓，因为完全没必要考虑什么入侵者，从来就没有入侵者，只有访客。

理查德在一次为 1984 年出版的《黑客》一书的采访中，很骄傲地提到，当年计算机科学实验室里，有五分之一的人换了“回车”做密码¹⁸。

理查德的这种反密码抗议，以及各种拒绝安全保护的行为，最终还是要失败的。八十年代起，哪怕是人工智能实验室，也开始支持密码登陆系统。可即便如此，理查德的行为标志着他个人政治思维成熟的里程碑。理查德迈出了一大步。当年他还是怯懦的中学生时，哪怕

遇到决定生死的大事，他也不敢上街公开反对。如今，他则成了一个活跃人士，把各种嘻笑怒骂当作家常便饭。

反对计算机安全这事，体现了理查德早年性格中的几个关键特质：对知识如饥似渴，对权威厌恶嘲弄；可又烦恼于别人对自己的各种偏见不解，被一些人看作异类。这些也正反映出了未来主导他行动的道德基础：对社区的责任感和信任；还有遇到问题，直接行动的黑客精神。用软件界的行话说，反密码抗议行动体现的正是理查德 1.0 版。虽然有些地方还很不完善，但大部分已经成熟。

理查德自己，则不太赞成把他这样的早年的黑客经历强调得太多。他说：“早年间，很多人都和我有同样的想法。当时那么多人最后选择了不用密码，这事实本身就说明他们很多人认为这么做是对的。我只是在这方面比较活跃。”

不过理查德依旧很感激人工智能实验室氛围唤醒了他作为政治活动家潜质。曾经还是少年的时候，理查德目睹了很多政治事件，他很难想象自己能做出什么有影响的事情。如今虽只过弱冠之年，他却已经可以在自己的领域里指手划脚，自信地为整个社区负责，也为个人自由出头。他说：“我加入了这个圈子，这个圈子的文化非常强调尊重每个人的自由。我很快就意识到这是个好东西，可花了很长的时间后，我才觉得这是一种道德责任。”

理查德不仅在人工智能实验室获得了黑客们的尊重。在哈佛的第一年，理查德参加了一个世界民间舞蹈小组。这个小组刚刚在卡瑞尔宿舍楼成立。理查德一开始也没打算加入，因为他觉得自己根本不适合跳舞。可他朋友却说：“你不试试怎么知道呢？”后来他真的去试了，出乎意料的是，他不仅很擅长跳舞，而且很享受其中的乐趣。这个无心之举最后竟然发展成了他的另一大爱好。这也给了他不少机会和女孩子搭讪，可惜在大学期间，每次搭讪都没能换来单独的约会。跳舞让理查德明白自己不再是那个手脚不协调，不会玩橄榄球的 10 岁少年了。他由此变得更加自信，灵活，有活力。在八十年代早期，理查德走得更远了：他加入了麻省理工学院民间舞蹈表演队。他穿着巴尔干半

岛的传统服饰，在观众面前表演舞蹈。这让他甚是陶醉。舞台上面对观众的经历，也为他日后在各种公共场合中演讲打下了基础。

尽管跳舞和写代码没能提高多少他的社交能力，可刚到哈佛时那种疏远不自然的感觉却从此烟消云散。1977年，在一次科幻小说聚会上，他看到了南希徽章制作机，你可以随便写点格言座右铭什么的，把它制作成徽章。理查德很是兴奋，他当场制作了一个徽章，上面写着“选举上帝”。

在理查德看来，所谓“选举上帝”可以从很多层面上来解释。作为一个从小就坚持无神论的人，理查德把“选举上帝”看作是信仰争论的另一种角度。“那时候，很多人都在讨论上帝究竟是不是存在。‘选举上帝’四个字给了人们一个全新的角度。如果说，咱们这位上帝能耐这么大，大到可以创造整个世界，可却把这个世界中的问题扔在那里，不闻不问，那我们何必要膜拜这么一个上帝呢？这样的上帝难道不该放到法庭上审讯一番吗？”

另一方面，“选举上帝”也影射了七十年代的水门事件。它把尼克松比作暴虐的天神。水门事件对理查德的影响很深。理查德从小就厌恶权威。如今长大成人，这份厌恶在人工智能实验室的黑客精神影响下，更加牢固。黑客们看来，水门事件好似莎士比亚剧一般，诠释着上层权力斗争如何戏虐无权的人。它恰似一幅缩影，展现着人们如何出于安全和方便的考虑，放弃自由和开放。

戴上徽章，理查德自信满满。好奇的路人倘若问上一句，必然换来一篇准备已久的长篇大论：“我乃耶和华，从天而降，替天行道。除忧治病，铲逆为公。我眼中所见之事，并非肉眼凡胎所能看；我心中所思之物，也非莽夫俗子所能解。神谕无价，安全第一，行道之法，乃是天机，万万不可泄露。你等世人，尽管加信于我，毋疑毋问，顺天听命。旦有不从，即为妖孽，从此名姓入簿，打入非常之所，不得入三界，列五行。妖孽之地，行严法，加重税，世代如此，不得翻案。”

如果只是觉得这段话是对水门事件的恶搞，那只听懂了一半。另外一半，恐怕只有那群黑客的同类能听懂。百年之前，阿克頓男爵（Lord

Acton) 曾经发出警告：绝对权力导致绝对的腐败。而很多美国人，恐怕忘了这个警告的前半部分：权利本身导致腐败。理查德并没有直接列举各种腐败案例，他直指过分信任权利，从而酿成腐败的整个社会。

理查德回忆起徽章的事情，说：“我觉得，为什么把讨论就停留在这里了？如果仅仅看到水门这一桩事件，你也许只是不再信任尼克松和他的幕僚。可明天你也许就又开始信任尼克柏，尼克树，你依旧信任权力。我对待这些掌权者都是一个态度：倘若他滥用权力，就活该在哪天被剥夺权力。”

Endnotes

¹ 参见迈克尔·格劳斯的采访：《斯托曼·斯托曼：中学校园里的怪才，自由软件界的旗手，麦克阿瑟奖获得者》（1999 年）。

² 译注：坦慕尼协会曾是纽约市民主党的政治机器。曾一度控制了曼哈顿一带民主党候选人的提名。二十世纪三十年代因操纵选票丑闻而一蹶不振。但在五十年代又有起色，六十年代左右逐渐退出政治舞台。

³ 卡米思·德·萨皮奥 (Carmin De Sapio) 曾是坦慕尼协会的第一个意大利裔主席。关于他和纽约二战后的政治格局，参见约翰·达文波特的文章《擒虎：卡米思·德·萨皮奥，和坦慕尼协会的最后时光》，《纽约事务》(1975):3:1. John Davenport, "Skinning the Tiger: Carmine DeSapio and the End of the Tammany Era," New York Affairs (1975): 3:1.

⁴ 译注：抽签征兵，1969 年至 1975 年期间，美国实行的一种征兵制度。根据生日，在适龄青年男性中，随机分配征兵顺序。顺序在前的人将最先收到征兵令。

⁵ 译注：美莱村屠杀，1968 年 3 月 16 日，美军因怀疑村民掩护越共逃亡，在越南广义省美莱村展开了屠杀。

⁶ 哥伦比亚科学之星计划的另一个学生，柴斯回忆当时的抗议游行“背景噪声”。“我们都关心政治。可科学之星计划对我们来说更重要。我们从没有翘课去游行。”

⁷ 译注：美国大学允许学生在学期开始之前选课，上过几周，如果觉得不合适，可以放弃这门课。所以一个课程在开始的时候会有很多人上，但到了学期结束只有一部分人坚持选下了这门课，拿到成绩。

⁸ 理查德对此持怀疑态度。“我离开数学和物理界，转去搞计算机，其中一个重要原因，是我觉得在数学和物理领域很难有什么新的突破。我只能每天学习别人以前做了什么。在计算机领域，我每天都可以有些新东西。”

⁹译注：美国学生一般会在校内或校外找些工作，一方面可以积累经验，但很重要的一方面是可以挣钱交学费。

¹⁰译注：休伯特·德莱弗斯 (Hubert Dreyfus)，美国哲学家。曾在 1965 年断言，计算机下国际象棋不可能战胜一个十岁孩子。参见他的论文《Alchemy and Artificial Intelligence》

¹¹参见史蒂芬·李维 (Steven Levy)，《黑客》(1984 年，美国企鹅出版社)，第 144 页。史蒂芬用了五页来描述高斯伯如何迷恋生命游戏。生命游戏是英国数学家约翰·康威建立的数学模型。强烈建议读一下《黑客》一书，可以作为本书的补充读物。

¹²杰拉尔德·萨曼 (Gerald Sussman)，他既是麻省理工学院的教授，也同时是人工智能实验室里的一名黑客。他在人工智能实验室工作的时候，理查德还没来。根据他的回忆，当时黑客们根本没有破门进入教授的办公室。

¹³*Ibid.*

¹⁴原谅我草草地带过了 ITS 的历史。这个操作系统至今依然被很多黑客视为黑客文化的缩影。关于这个操作系统的更多历史，参见西姆森·加芬克尔 (Simson Garfinkel) 的书《信息社会的建筑师：麻省理工学院计算机实验室的三十五年》(麻省理工学院出版社，1999 年)。Architects of the Information Society: Thirty-Five Years of the Laboratory for Computer Science at MIT

¹⁵参见理查德在瑞典皇家技术研究所的演讲 (1986 年 10 月 30 日)：<http://www.gnu.org/philosophy/stallman-kth.html>

¹⁶本书最后一审期间，理查德发来邮件说，他从哈佛大学也得到了很多政治上的鼓舞。理查德说：“我大一的时候，修了中国历史这门课。读了很多关于反对秦朝暴政的故事。虽然很多故事也许没有史料依据，但依旧振奋人心。”

¹⁷参见上述的理查德在瑞典皇家技术研究所的演讲。

¹⁸参见史蒂芬·李维 (Steven Levy)，《黑客》(1984 年，美国企鹅出版社)，第 417 页。

Chapter 5

自由一隅

[RMS：本章中，我纠正了第一版中的一些错误陈述。包括对我想法和感受的描写。除非标明，我都保留了威廉原文中关于他个人感受的描写。]

如果谁和理查德在一起，呆上超过一分钟，他的感受一般都是这样：不管他那一头的长发，不管他怪异的举止，给人印象最深刻的，是理查德凝视你的眼神。透过他绿色的眼睛，你能看到一种真实的信仰。

要说理查德眼神犀利，都要算是轻描淡写了。他不只是看着你，他简直就是要看透看穿你的心。哪怕你出于礼貌，暂时把自己的视线从他身上移开，你依旧能感受到他的眼神像灯塔一样，锁定在你的身上。

也许这就是为什么，很多描写理查德的文章，都会从信仰的角度开始写起。在 1998 年 salon.com 上的一篇文章，标题定为《自由软件圣徒》。作者安德鲁·伦纳德描述理查德那双绿色眼睛的时候，写道：“[它们] 散发出一种旧约中先知般的力量¹。” 1999 年《连线》杂志的一篇文章中，形容理查德的络腮长胡须好似拉斯普京²的胡须一般³。《伦敦卫报》则把理查德的微笑说成好似“门徒看到耶稣”一般⁴。.....

这些比喻有一定道理，但似乎也有不足。它们都忽视了理查德人格中脆弱的一面。盯着理查德的眼睛再久些，你会发现其中潜在的改变。一开始，是一种强势带着几分催眠般的眼神；可再看几次，就会发现那眼神其实是在寻求沟通，建立联系。如果他个性中真的有着一些自闭症的成分，那么他的这种眼神倒确实符合这种个性。哪怕当他目光如炬地盯着你的时候，你依旧能够感受到他眼神中透出的几分迷茫和疏远。恰似受伤的猛兽一般，眼中传出的有几分绝望。

我第一次见识传说中理查德的目光，是在 1999 年 3 月的第一届 LinuxWorld 大会上。大会在加利福尼亚州的圣何塞市举行，是一场“Linux”社区的大聚会。其中一个环节，就是在媒体面前介绍理查德。理查德决心在这次见面上简要回顾 GNU 工程的历史以及这个工程政治上的目标，以此说明自己在社区中的工作和价值。

作为一个报道此次大会的记者，我在 GNOME 1.0 的媒体发布会上，被理查德亲自上了一课。当时，我的第一个问题在无意之中触碰到了敏感话题：“请问您觉得随着 GNOME 逐渐成熟，是否会推进 Linux 的商业应用？”

“请你不要再把整个操作系统都叫做 Linux。”理查德回应，眼光迅速聚焦到我的身上：“Linux 是一个操作系统的内核，它仅仅是操作系统的一小部分。你所说的 Linux，实际还包含了很多软件。这些软件并不是莱纳斯·托瓦尔兹（Linus Torvalds）开发的。它们是由 GNU 工程的志愿者开发的，这些志愿者用自己的业余时间创造了这些软件，让用户最终可以自由使用的一个操作系统。如果不提及这些开发人员的工作，那是既不礼貌，也不尊重历史的。所以我坚持把这个操作系统称作 GNU/Linux，也希望请你这么称呼它。”

把这段话记在采访本上时，我感到整个会场异常地安静。等到我抬起头来，才看到理查德的眼睛眨都不眨地盯着我，让我心生一丝胆怯。这时候，另外一名记者提问，正确地使用了 GNU/Linux 的名字，才打破僵局。GNOME 项目的组长，米格尔·德·伊卡萨回答了这名记者的问题。米格尔回答到一半，理查德才把目光从我身上移开。这

时我才觉得些许放松。等到理查德继续纠正了另外一个记者的措辞问题后，我才有种罪孽被洗清的感觉。他起码不再盯着我了——我对自己说。

从理查德的角度看，这种面对面的时刻是有意义的。这次大会结束的时候，各路记者在理查德面前已经改用 GNU/Linux 这名字。而《连线》的记者则开始忙着写一篇通讯，其中把理查德比作斯大林之前的革命者，被黑客和各大公司从历史上抹去了姓名，他们还闭口不谈 GNU 工程，因为不喜欢 GNU 工程中的政治目标⁵。其他的报道也相继出现，尽管没有在报道中使用 GNU/Linux 的名字，但它们大多都提及了理查德的贡献，说到他十五年前发起的开发自由操作系统的项目。

再次见到理查德，是 17 个月后。在这 17 个月中间，理查德又一次前往位于硅谷的圣何塞，参加 1999 年 8 月的第二届 LinuxWorld 大会。尽管没有演讲，但理查德此次还是有目的的。他是代表自由软件基金会，领取莱纳斯·托瓦尔兹社区贡献奖。理查德接受奖杯的时候，打趣道：“这次代表自由软件基金会领取莱纳斯·托瓦尔兹社区贡献奖，感觉就像唐僧代表师徒四人去领取悟空奖。”

可惜，这次的玩笑并没有吸引多少媒体的关注。倒是这周的红帽公司（Red Hat, Inc.）上市，引起了各路记者的关注。这是一家以发布 GNU/Linux 发行版为主的软件公司⁶。这次上市，印证了我们大多数记者的猜测：“Linux”一词，会和“电子商务”，“点 com”等一样，成为华尔街上的流行语。股票市场的热闹会把大家搞得像中了千年虫病毒一般，忘掉过去，丢掉历史，把自由软件或开源的政治方面抛诸脑后。

也许这就是为什么，在 2000 年第三届 LinuxWorld 大会上，大家再也没看到理查德的身影。

我第二次再和理查德见面，是在第三届 LinuxWorld 大会之后不久。当时我听说理查德又要来硅谷，我就和他联系好，定在加利福尼亚州帕罗奥多市（Palo Alto）进行采访。这个地点选的很有意思。不仅仅

是因为他上次缺席 LinuxWorld 大会，还因为这里作为硅谷重镇，是除了微软所在的华盛顿州雷蒙德市以外，最支持专有软件经济模型的几个城市之一。而理查德则花费了自己整个青春和大半生的时间，去与我们文化之中的自私与贪婪作斗争。我很好奇，他来到这座城市，看着车库大小的平房里，都在做着几十万美元的买卖的现实，理查德会作何反应。带着好奇心，我离开奥克兰市，驱车前往帕罗奥多。

按照理查德给我的地址，我来到了 Art.net 的总部。这是一家汇集各路“网上艺术家”的非盈利组织。位于城市北部，坐落在一所被篱笆围起来的房子里。房子有些旧，但又透着些小清新。这氛围顿时又让我觉得，理查德混在硅谷，似乎也不是个很怪的想法。

我看到理查德坐在一个背阴的房间里，正在他那台灰色笔记本电脑上敲着键盘。我一进门，他就看着我，投来了那著名的目光。互相寒暄，他又把目光聚焦回笔记本屏幕上。

“我刚刚写了一篇关于黑客精神的文章。”理查德一边打字，一边说：“过来看看？”

我走过去开始读文章。房间阴暗，屏幕上的编辑软件主题，又被理查德设置成了黑底绿字。我读了几个字，眼睛才逐渐适应。文章开始回忆了理查德有次去一家韩国餐馆的经历。当时服务员摆桌的时候，在理查德面前放了三双筷子。一般人恐怕都会把多余的两双筷子拿开。可理查德却又借此开始玩起把戏，他企图找个法子，用上所有的筷子。和很多黑客技巧一样，最终的解决方案既聪明灵巧，可又傻里傻气。理查德以这个事情来作为整个文章的引子。

我读的时候，感觉理查德一直在盯着我。起身看了他一眼，见他一脸骄傲的表情，好似孩子般笑着站在那。我夸奖了一下他的文章，勉强让他眉毛抬了抬。

“稍等一下，我们马上就出发。”他说。

理查德坐回座位，开始继续敲键盘。他用的笔记本是个灰色的四方盒子，和 LinuxWorld 上看到的大多笔记本不同，理查德的这台没那么光鲜亮丽，也不是什么新款式。笔记本的键盘之上，还放着一个

更小，更轻的键盘。理查德的一双大手，则在这个小键盘上飞舞。九十年代中期，理查德双手的剧痛让他无法容忍，以至于他曾一度雇用了一位打字员。今天，他用的键盘，是一种特殊的，比常规键盘按键力度更小的键盘。

理查德在工作时，会全然不顾外界刺激。看着他眼睛盯着屏幕，手指飞舞，让人觉得理查德和电脑，恰似老友重逢，当下正促膝长谈。

理查德用力敲了几下键盘，合上笔记本电脑，拔掉电源，结束了写作。

“吃午饭去吧？”理查德问。

理查德抱怨着自己脚踝的伤，和我一起徐步走到我的车前。他左脚跟腱三年前受了伤，让他这么一个民间舞爱好者从此告别任何舞蹈活动。理查德叹道：“我可喜欢跳舞了，不能跳舞对我来说简直是个悲剧。”

这悲剧从理查德的身材上就体现出来了。由于缺乏锻炼，理查德开始发福。脸颊越来越圆，啤酒肚也凸出来了。你能明显看出，他发胖简直就是一瞬间完成的。他自己都还来不及适应，以至于在走路的时候，都要撑着腰，好似一个孕妇一般。

走到中途，理查德停下脚步，在一簇玫瑰花前伏下身子。剥开花瓣，理查德把鼻子凑到花蕊前，深吸一口气，然后起身长叹，甚是满足。

“嗯，*rhinophytophilia*。”他揉揉腰说。⁷

开车到餐馆不到三分钟。根据前自由软件基金会执行总监蒂姆·内伊（Tim Ney）的建议，我让理查德选择去哪家餐馆。尽管很多报道，把理查德描述成苦行僧一般，可事实上，理查德在饮食方面，可谓是个美食家。作为一个布道全球的自由软件使者，一个额外收获就是可以尝遍各地美食。内伊介绍：“他去了几乎全世界所有的主要城市，这让他总能知道全城最好的餐馆在哪。理查德知道菜单上每个菜都是什么，点上一桌美食，他经常以此为荣。”

今天，理查德选择了一家广式点心店。这家店和帕罗奥多市的主干道学院街（University Ave.）相隔两个街区。选这里，一部分是因为理查德刚刚去过中国，期间在香港做了短暂停留。理查德本人并不喜欢太辣的东西，所以他并不喜欢川菜和湘菜。“我对辣的东西不感冒。”理查德坦言。

快十一点的时候，我们赶到餐馆。结果发现居然在餐馆的门前已经排了长队，我们要等二十分钟才能入席。我深知黑客们不喜欢浪费时间，于是屏住呼吸，生怕理查德大动肝火。可出乎意料，理查德竟然坦然接受了这个事实。

理查德对我说：“就只有我们俩来了，真可惜。人多了吃起来才热闹。”

等待期间，理查德开始走起舞步。他步步小心，可还是能看出有功底。我们开始聊起时事。理查德说，这次缺席 LinuxWorld 大会，最遗憾的部分是没能出席 GNOME 基金会的成立发布会。这个基金会是 Sun 公司和 IBM 一手操办，它很好地诠释了理查德的观点：所谓自由软件，并不与自由市场和自由经济相冲突。然而，理查德依旧不满意大会的论调。

“发布的时候，各个公司都在说着 Linux，闭口不提 GNU 工程。”理查德说。

这种冷漠恰恰和地球另一边的热络形成对比。尤其是亚洲各国，甚是热情。看看理查德 2000 年的行程安排，就能看出自由软件的迅猛增长。在 115 天之中，理查德只在美国停留了 12 天，其他时间，则在印度，中国或巴西境内。他的旅行，让他见识了自由软件如何被翻译成不同语言，进入各国文化。

“在印度，很多人对自由软件感兴趣。因为他们觉得这样可以用很低的成本，建设起自己的计算机基础设施。”理查德说：“在中国，自由软件的概念就传播得相对慢了一些。我们经常把软件自由和言论自由并提，表示这是软件用户的一种基本自由，可在中国，这种类比显

然不太奏效。不过无论如何，我这次的中国之行，还是在一定程度上提高了自由软件的关注度。”

话题很快转到 Napster，这是一家加利福尼亚州圣马特奥市的软件公司。这几个月，这家公司成了各大媒体的宠儿。这家公司开发了一款倍受争议的软件，使得音乐爱好者之间可以互相拷贝音乐文件。借着互联网的东风，这类称作 p2p（即端到端）的软件逐渐流行。如今，Napster 俨然成了大型音乐盒，让各路音乐爱好者可以免费欣赏音乐。这一下子就惹恼了各大唱片公司。

尽管 Napster 是个专有软件，但它的流行却印证了理查德的想法：一旦某个作品进入数字世界，或者说，一旦复制这个作品变得便宜简单，那么人类本能地分享这个作品的行为将会难以阻止。Napster 给了用户一片分享音乐的空间，这家公司接着就可以借助用户量来寻求更多商业机会。

Napster 的迅猛成功让各大唱片公司开始担心。我来帕罗奥多之前的几天，玛莉莲·帕特尔法官刚刚在美国地方法院通过了一项由美国唱片协会提交的禁令。该禁令禁止了文件共享服务。而 Napster 之后上诉到上诉法院，使得这个禁令被美国第九巡回上诉法院暂停。但在 2001 年年初，上诉法院判定 Napster 的确违反了版权法⁸。美国唱片协会发言人希拉里·罗森对这个判决的评价是：“保护了创作者，完善了在线市场方面的法律⁹。”

对于理查德这样的黑客来说，Napster 的商业模型内包涵着复杂的信息。一方面，这家公司大力倡导黑客社区所欣赏的信息共享精神；而另一方面，又出售着基于专有软件的服务。这种复杂性，让理查德对这一事件少言寡语，因为他知道，自己已经花了很大功夫去传播自己的思想，而对这种部分和自己思想相同；但很大程度又有矛盾的事件发表评论，很容易被媒体断章取义，平添混乱。可理查德依旧承认，从 Napster 事件上，他也认识到了一些有价值的东西。

“以前，我觉得人们私下分享娱乐信息就 [足够] 了。”理查德说：“可如此庞大数量的用户，在 Napster 上分享音乐，这让我觉得，不仅仅

是私下朋友和朋友之间的共享很重要。在公共场合，公开给大家分享信息的自由，也同样重要，不可剥夺。”

话到此，餐厅服务员赶来，告诉我们有了空位。于是我们被领班带到桌前，坐在餐厅一角，身旁的墙壁，挂着一整面镜子。

服务员拿来菜单，和笔，让我们在菜单上勾选要点的菜。还没等服务员把茶水拿上桌，理查德就已经开始在自己的菜单上点菜了。“香煎鲜蝦腐皮卷”理查德读着菜单：“腐皮，我喜欢这口感。点一个。”

这一下子又引来了理查德关于中餐和这次访问中国的即兴评论：“中国的饮食真是博大精深。”理查德大声说，这要算是他整个上午嗓门最大的一次了：“各种我在美国都没听说过的食物。还有很多当地特产，当地美食。当时突然我又想起来，我要办份杂志，专门记录每顿大餐的菜谱。”

这讨论接着又转向韩国菜。同样也是在 2000 年 6 月，理查德的亚洲之行中，也去了韩国。他的到来，在当地媒体引起了一个不小的轰动，这部分原因，或许还要部分归功于当时也在韩国的微软创始人比尔·盖茨。理查德的照片这次上了首尔的热销报纸的头版，把比尔·盖茨的照片竟然挤到了下面。除了这件事情让理查德非常满意之外，当地的美食也让他欢欣鼓舞。他回忆：“我吃了一碗冷面。那面条口感非常独特，我们这边都不用那种面条做冷面。我敢肯定，那冷面是我有生以来吃到的最正点的冷面。”

“正点”一词在理查德嘴里应该是很高评价了。因为正当理查德介绍冷面，我突然觉察到他的眼神飘到了我的右后方。

“你后面坐了个很正点的妹子。”理查德说。

我赶紧回头，瞟到了一个女人的背影。这女人很年轻，二十来岁的样子。穿了一件白色裙子，裙子上贴着闪闪发亮的饰品。她和同桌的一位男性已经用完餐，正在结帐。当两人起身离开的时候，我发现了理查德眼中瞬间失落的神情。

“啊！别啊！他们要走了。我估计以后也见不到她了。”理查德抱怨道。

一声叹息，理查德心神回位。这倒给了我一个机会，问问理查德的感情生活和审美偏好。因为坊间传言，他可是个轻佻的公子。有些时候，很多传言还有些自相矛盾。很多黑客都声称，理查德和女性见面，都会以吻手背的方式来打招呼¹⁰。

在 Salon.com 上，有一份 2000 年 5 月 26 日的采访，把理查德描写成了一位黑客界的花花公子。文中把自由软件和自由恋爱做类比，作者安娜利·奈唯姿把理查德说成是拒绝传统家庭观念的叛逆者，他还说：“我相信爱情，但不相信一夫一妻¹¹。”

当我问起这方面问题时，理查德把菜单放下后说：“这个 很多男人似乎都会不自觉地蔑视女性，看到女性就想到上床一类的事情。就连很多女人也自觉不自觉地融入到男性的这种倾向的思维中，可我无法理解这种态度。”

我提到，在 1999 年，一本名为《开源软件之声》的书中，理查德在一篇文章里提到，当初他想以女朋友的名字来命名 GNU 操作系统的内核。这位女孩子名叫 Alix，因为是以 x 结尾，所以和 Unix 界对操作系统内核的命名传统非常契合，比如 Linux 就遵照了这个命名传统。Alix 当初是个 Unix 系统管理员，而且曾经和她朋友开玩笑地提到过：“应该有个内核以我的名字命名。”理查德于是决定把 GNU 的内核命名为 Alix，以此给她一份惊喜。可最后内核的主要开发人员把内核名字改成了 Hurd，把 Alix 作为其中一个模块的名字。Alix 的一个朋友看到源码中出现了 Alix 的名字，于是把这件事情告诉了她。Alix 听后非常感动。之后 Hurd 又经历了几次重新设计，最终还是去除了 Alix 这个模块¹²。

整个一个上午，理查德脸上都挂着笑容。我问起关于吻手背的事情，理查德说：“哦，是的。我的确这么做。我觉得这种方式可以给很多女性带来亲切感，她们也都会喜欢。这是个很好的机会，你可以借此拉近彼此距离。”

亲切感是理查德生命中永恒的主题。问起这个问题，他非常坦诚：“在我的生命里，确实很少体验到亲切感，只是在我头脑之中还

存在着这种感受。”可聊着聊着，话题就变得尴尬。理查德几次回应都只是从嘴里蹦出几个单字，再问几个问题，他就举起菜单，插话道：

“你想吃点烧麦吗？”

上菜间隙，谈话越显尴尬。我开始和他闲聊起来，问他关于中餐的问题，问他当初他在人工智能实验室做程序员的时候，每周去唐人街吃饭的问题，还问了他关于中文口语和汉字书写之间的逻辑联系。本想借此绕回主题，可每次尝试，都被理查德一套太极功夫挡了回去。

“我上次去中国的时候听到人们说上海话。”理查德说，“很有趣，和普通话非常不同。我问他们把同一个词分别用上海话和普通话说一遍。有些时候你能大概听出点联系。我问他们声调是不是也类似，他们说不一样。这让我觉得很有趣。因为曾经有个假说认为，不同声调曾经是不同音素，之后语言不断演化，几个音素被不同声调取代。如果这个假说成立，那么声调不同就意味着这个方言是在音素演变成声调之前就独立发展的。”

第一个上来的菜，是萝卜糕。我和理查德都花了些力气，才把这块大的萝卜糕切开。它闻起来像是白灼生菜，吃起来又像煎土豆饼配培根。

我决定继续讨论刚才说到一半的话题，我问他，是否觉得自己的童年经历使自己养成了如此特立独行的性格。其中很出名的一个事，就是他从 1994 年开始，就致力于纠正广大计算机用户和各地媒体的误读，让他们使用 GNU/Linux 一词来替代 Linux。

“我觉得，[特立独行的性格]的确让我[避免了对流行观点的盲从]。”理查德嚼着一个饺子，继续说道，“很多人都说会感受到旁人观点的压力，而我却从不在意。我觉得我一直在拒绝盲从别人的观点。随波逐流不会有什么好处，不会带来任何改变。因此，我也没太在意别人对我的看法。”

理查德以音乐喜好为例，向我讲述他叛逆的倾向。在他少年时期，大多数同龄的中学生都迷恋魔城音乐和摇滚乐，而理查德则喜欢听古典乐。他回忆 1964 年，自己中学生时候，披头士在《埃德·沙利文秀

场》栏目出现¹³，他的同学们都冲出去买披头士的最新专辑或单曲。而就在那个时候，理查德决心抵制披头士音乐。

理查德说：“我喜欢披头士之前的一些流行乐。可我不喜欢披头士。我更不喜欢人们追求披头士的那份疯狂。当时就好像说：谁能攒齐披头士的所有专辑，谁就是最崇拜披头士的。”

理查德的抵制行为显然没造成多大影响，他还试图找点其他法子戏弄一下盲目的同龄人。他说，他曾经考虑过要自己组织个乐队，专门调侃披头士。

“我想把这乐队叫做‘东京花’，或者‘日本披头士’。”

我知道理查德喜欢世界各地的民族乐，我就问他对民谣歌手鲍勃·迪伦什么看法。理查德摇摇头：“我倒是更喜欢彼得，保罗和玛丽。他们让我想起了伟大的俳客（folk）音乐。”

我问他什么是俳客音乐，理查德解释说是一个在科幻圈子里的术语，专门指代为的现成歌曲重新填词后创作的歌曲。（在最近几年，一些俳客乐作者也会自己谱曲。）传统的俳客乐有《意大利面条之上》（On Top of Spaghetti），这是借用《旧思墨客山之上》（On Top of Old Smokey）的曲调¹⁴。还有由俳客乐大师，“奇士”埃尔·杨可维克（“Weird” Al Yankovic，“奇士”是此人绰号，也被人称“奇士埃尔”）填词，改变自《罗拉》（Lola）的歌曲《尤达》（Yoda），显然，《尤达》是描写《星球大战》里的绝地武士尤达的。“

理查德问我是不是想听他唱俳客歌。我说想听。他很快开始唱起来，声音出奇地清晰。用的是《随风飘散》（Blowin’ in the Wind）的曲调¹⁵：

一只土拨鼠要拨多少土，
才能称得上是土拨鼠？
波兰人要播多少栏目，
才配得上波兰这名属？
黑奴兄弟啊，你要黑掉多少弓弩，
才敢被人称作黑奴。

这一切的答案，亲爱的，
都掉进了你耳朵，粘到了听小骨，
答案啊，都粘到了听小骨……

曲罢音散，理查德唇角又露出了孩子般的笑容。我环视周围，旁边的几个亚洲人正享受着自己周日的饕餮，没把心思放在这位大胡子叔叔身上¹⁶。犹豫了几秒，我也冲理查德笑了笑。

“这个粟米球你还吃吗？”理查德眨着眼睛问到。我还没回答，理查德就拿筷子夹起粟米球往嘴里送，“那我就不客气了！”他说。

饭菜吃完，我们准备进入采访的正题。理查德端起茶杯，靠在椅子上。我们继续开始讨论 Naspter 和自由软件运动的话题。我问，自由软件的原则是否应该扩展到其他领域，比如音乐领域上？

“把一个问题的答案无条件地嫁接到另外一个问题上，这可就不对了。”理查德说。说起歌曲音乐和程序，他说：“正确的思路是，根据每类不同的作品做分析，由此再得出相应的结论。”

当说起版权作品的时候，理查德把整个领域分为三个类型。第一个类型，是所谓的“功能类”。软件，辞典，教科书都属于此类。第二个类型，可以称为“证据类”。科技论文，古籍文献都属于此类。这些作品，如果可以被任意修改，则没有了价值。这一类也同样包括对个人感受的记录，比如日记，游记，自传。修改这些作品，则会影响到对某人的回忆或观点。因此理查德觉得对此类作品，读者则不能有权修改。第三类，则是“艺术类”，包含了艺术和娱乐作品。

这三类作品中，应该无条件赋予用户权力去修改的是“功能类”的作品；而对于“证据类”和“艺术类”作品，则要根据作者意愿，才能赋予用户一定的修改权力。理查德坚信，无论哪类作品，用户出于非商业目的的复制与分享行为都应该被允许。如果说，一旦允许如此，就会有互联网用户把各种图片，音乐或者书籍复制成百上千份，然后把拷贝发送给几百个陌生人，那么也要允许这种行为。“很显然，私下偶尔分享这些作品的行为，必须被允许。只有警察国家才会禁止这些行为。”理查德说。“把一个人和他的朋友隔离，这是反社会的行为。而

Napster 事件则让我觉得，我们不仅要允许私下朋友之间的分享，更应该，也必须允许用户，出于非商业目的，把这些作品的拷贝分享给公众。因为太多的人希望如此，并且觉得这么做非常有价值。”

我问他，法庭是不是会接受这种放任的观点。理查德打断了我：

“这是个错误的问题。你把这个话题从道德伦理的领域，转移成了法律领域里的释法问题。这是一个话题的两个完全不同的方面。从一个反面跳到另一个是没有任何意义的。法庭如何解释现有法律是有严格规定的，而这些法律也接着这些规定被各路出版商搬上台面。”

这段话反映出了理查德的政治哲学：现有的法律系统保护各大公司，让他们能利用现有版权法保护自己的软件，但这并不意味着计算机用户必须得遵守所谓的规则。自由本身是个伦理问题，不是法律问题。他说：“我们需要超越现有法律，暂时忘记现在实际是什么样，去思考我们究竟应该要什么样。我不是要立法，而是在考虑法律应该做些什么。我觉得，禁止朋友之间分享拷贝的法律，和种族歧视法律一样，不该得到尊重。”

说起种族歧视法，我又问了另外一个问题：他是否被曾经的那些政治领袖影响或鼓舞？比如五六十年代的民权运动，理查德的政治思想也是基于类似的普世价值：自由，正义，公平竞争。

.....

“我们不是一个团体，但我们做着同样的事情。”理查德一边嚼着东西一边说。

我接着提起另外一位民权活动家，马尔科姆·艾克斯（Malcolm X），把他的观点和理查德的做类比。马尔科姆曾是美国非洲裔人的新宗教组织“伊斯兰国度”的领袖。和他一样，理查德也喜欢谈论有争议的话题，疏远潜在的盟友，并且传播着一套自恰的信息，无视与现有文化体系的整合。

理查德嚼完嘴里的东西，表示他拒绝这种类比：“我要传达的内容更类似马丁·路德·金的思想。这是一个普世价值观。它谴责一些伤

及他人的行为。我的思想并不是要人去憎恨谁，也不是针对一小撮人。我希望每个人都能珍视自由，也能拥有自由。”

很多人都指责理查德，说他经常拒绝触手可得的政治盟友。有人从心理学的角度，认为这是理查德的一个性格特征。他拒绝使用已经被大众接受的“开源”一词，拒绝与这个圈子的人合作项目。他为自由软件奋斗了二十多年，他的名字已经深深地和自由软件绑在了一起，他的政治资本也是要不断投资在这四个字上。在 LinuxWorld 上领奖时，他的那翻类比，更加深了人们对他的这一印象。茫茫众生之间，很多人早已把跟风赶潮当作一种美德。而理查德在这些人之中，则更显得头脑守旧，不肯在政治大趋势和市场面前有半点让步。

红帽公司总裁罗伯特·杨（Robert Young）总结理查德看似矛盾的政治行为时，说道：“我崇拜也尊敬理查德和他所做的一切。我对他唯一的批评就是，有些时候，他对待朋友甚至比对待敌人还要无情。”

[RMS: “朋友”一词只能部分地适用与罗伯特·杨和红帽公司的身上。他们做的一些事情，确实称得上是朋友。比如，红帽公司为自由软件开发贡献了很多，这其中包括很多 GNU 工程的程序。但红包公司也同样做了一些违背自由软件运动主旨的事情。比如，他们的 GNU/Linux 发行版中，就包含一些非自由软件。从措词上看，他们把整个系统称为 Linux，是对 GNU 工程的无视。他们宣扬“开源”的同时，不考虑自由软件，也是违背我们的价值观。如果红帽公司和罗伯特·杨能与我们同道，我乐意和他们一起共同协作。可很多时候，并非如此。]

理查德这种独行侠的作风，并非是他不关注其他政治人物。在他 MIT 的办公室里，你能一下子看到很多左派文章，记录着全球各地公民权被侵犯的事件。登陆他的个人网站，stallman.org，你可以看到很多文章，反对千禧年版权法案，抵制反毒品运动，抵制世贸组织等等。理查德解释说：“我们必须谨慎地为自由软件运动挑选政治盟友。因为有些自由软件支持者也许并不支持有些运动。比如，我们会避免把自由软件运动和其他政党建立联系，因为我们不希望这样的行为去驱散那些自由软件的支持者。”

鉴于理查德这种活动家的倾向，我问他为什么不尝试发出更大的声音？为什么不借助他在黑客圈的影响力，去推进他的政治主张？[RMS: 可我其实已经在这么做，但只有在时机成熟的时候才如此。这也是我为什么建立了 stallman.org 这个个人网站]

理查德整理了一下头发，思考了一阵。[RMS: 我下面的回答有些所问非所答。对于我下面的话，更合适的问题也许是：“你为什么专注于自由软件，而不是其他你关注的问题？”我记得当时被问的问题似乎是这个。]

答道：“软件自由，仅仅是自由一隅，冰山一角。我不希望过分强调这一角。因为还有更多的领域需要自由，这一点十分重要。和这些自由相比，自由软件则显得微不足道了。自由软件是我的一份责任，因为这项自由恰好落在我熟悉的领域之中。可除此以外，还有很多极其重要的问题需要关注：警察暴力，刑讯逼供；人们对某些毒品的偏见；种族歧视；如何让人们过上安逸的生活；如何让有堕胎经历的人免遭歧视；如何避免极权政治等等。这些都和我们的生活息息相关，它们都比我的工作重要。我只是希望能了解如何在这些问题上做点什么。”

理查德之后的一席话，让我觉得他利用了各种政治活动增进了个人信心。他在自由软件上花了大量时间和精力，很多活动，他虽然支持，但若真投入其中，则要犹豫再三，要权衡自己的时间和精力。

“我真心希望我能为解决这些问题，贡献出自己的一份力量。倘若我的贡献能让这些问题有所好转，我将引以为豪。可这些问题困难至极，多少比我更优秀的人，前赴后继，才只能做到今天这地步。当初我审视这一切，我发现依然有一处自由无人守护，于是我才投身至此。软件自由也许并非举足轻重，但守卫之人，仅我一个。”

吃完最后一口，理查德要结帐。服务员还没来收钱，他又掏出一张白色纸币。一眼看去，这张纸币显然是张假币，我忍不住拾起来仔细端详了一下。它自然不是美国铸币局制造的。上面画的不是华盛顿，不是林肯，而是一头卡通猪。上面写的也不是“美利坚合众国”（United

States of America)，而是“放纵贪婪中”（Untied Status of Avarice）。这张纸币一钱不值¹⁷。服务员来的时候，理查德拽着他的袖子：

“我又给你加了零美元的小费。”理查德说着，又露出了他标志的微笑。

这位服务员可能没有领会什么意思，或者真的被这假币迷惑住了。他也回敬笑了一下，快步走开了。

“看来我们可以离开了。”理查德说。

Endnotes

¹参见安德鲁·伦纳德的文章《自由软件圣徒》，1998年8月：http://www.salon.com/21st/feature/1998/08/cov_31feature.html

²译注：拉希普京，全名葛列格里·叶菲莫维奇·拉斯普京。沙俄末期的一名颇受争议的东正教教徒。当时的沙皇尼古拉二世和皇后亚历山德拉笃信神秘主义，拉希普京便因此得宠。后被尤苏波夫亲王、狄密翠大公、普利希克维奇议员等人合谋刺杀。

³参见利安得·克尼的文章《被 Linux 遗忘的人》，《连线》，1999年3月5日：<http://www.wired.com/science/discoveries/news/1999/03/18291>

⁴参见《有操守的程序员：坚持自由，坚持自由软件的理查德·斯托曼》，《伦敦卫报》，1999年11月6日；<http://www.guardian.co.uk/uk/1999/nov/06/andrewbrown>。把理查德比作宗教信徒的文章还有很多。Linux 的作者 Linus Torvalds 曾用更极端的语言把理查德比作宗教信徒。参见他的自传《好玩而已：一场无意间发起的革命》Just For Fun: The Story of an Accidental Revolutionary, HarperCollins Publishers, Inc., 2001, 第 58 页：理查德是自由软件之神。

⁵参见利安得·克尼的文章《被 Linux 遗忘的人》，《连线》，1999年3月5日：<http://www.wired.com/science/discoveries/news/1999/03/18291>

⁶译注：如今红帽公司的业务已经逐渐扩大，并不仅仅以发布发行版为主。但依然以 GNU/Linux 的各种解决方案为核心。

⁷开始，我还以为理查德说的是这花的学名。几个月后，我才发现理查德所谓的 rhinophytophilia，原来也是他的一个玩笑。他实际是把嗅花的行为比作鼻子和花做爱。更多理查德关于花的玩笑，参见：<http://www.stallman.org/articles/texas.html>。译注：rhinophytophilia 是理查德自创的一个希腊词汇。rhino 意为“鼻子”；phyto 意为“花”；philia 意为“爱”。

⁸

⁹

10

11

12

¹³译注：1964 年，披头士乐队第一次在美国电视上表演，当时参加的是《埃德·沙利文秀场》这一节目（Ed Sullivan Show）。从此以后披头士在美国也开始火爆。

¹⁴译注：《旧思墨客山之上》（On Top of Old Smokey）是美国一首著名的民谣乐。原歌曲讲述主人公在旧思墨客山上失去真爱。《意大利面条之上》（On Top of Spaghetti）是由汤姆·格雷泽（Tom Glazer）填词，讲述主人公吃意大利面条的时候，旁人打喷嚏把顶上的肉丸子吹到了地上。

¹⁵译注：《随风飘散》（Blowin’ in the Wind）是鲍勃·迪伦编写的著名民谣歌曲。在电影《阿甘正传》中，珍妮曾在舞台上演唱过该曲。

16

17

Chapter 6

Emacs 公社

二十世纪七十年代，从哪个角度看，麻省理工学院的人工智能实验室都是个特别的地方。里面有尖端的项目，顶级的研究人员，他们为整个实验室赢来了美名。而实验室内部的黑客文化，无政府主义的基调又为实验室平添了一层反抗权威的个性。几年之后，实验室中很多科学家和明星开发人员纷纷离开，黑客们这才意识到自己曾经的环境是多么独特。

在一篇 1998 年《福布斯》杂志上的文章中，斯托曼回忆当时实验室的氛围，说：“那就好像是个伊甸园。在那样的环境中，我们不可能拒绝互相合作。”¹

这样的描述也许略显夸张，但却反映了一个很重要的事实：技术广场 545 号楼 9 层，那里曾经不仅仅是一个工作场所。对于像理查德一样的黑客来说，那里更像是家。

“家”这个字，在理查德的心中，有着特别的份量。他少年时，家中的变故和经历让他直到上了大学，才对家这个概念有所感悟，并心存感激。他曾把哈佛的宿舍当作自己第一个真正的家。描述起当年离开宿舍的时候，他甚至依然心存悲伤。他有次提起自己的大学生涯，说

道大学期间，最悔恨的一件事情，就是被哈佛踢出校门。我问他究竟是触怒了何方神圣，才被赶出学校的。这才发现理查德早有准备：

“哈佛有个规矩，你修了太多的课程，就必须得毕业了。”理查德道。

离开了宿舍，又不想回纽约。理查德跟随着和格林布拉特，高斯伯，萨斯曼等人的足迹，来到了麻省理工学院，继续读博士。他在剑桥市一带租了一所公寓，但没有多久，他就把人工智能实验室当作了自己真正的家。在 1986 年的一次演讲中，理查德回忆起那时的人工智能实验室：

“比起其他人，我睡在实验室里的日子可能更久些。因为每隔一两年，我总会因为各种原因，会有那么几个月的时间没地方住。这期间，我就住在人工智能实验室里。我一直觉得那里很舒服，冬暖夏凉。那个时候，睡在实验室一点也不稀奇。他们都是热情满满，一直写代码调程序，因为你实在不想停下来。等到实在太累了，就在旁边找片平坦舒服的地方，倒头睡下。大家都是这么不拘一格。²。”

实验室里的这种氛围，有时候也会引来麻烦。有些人把它看作是个宿舍，而外人看来就像是电子大烟馆。1976 年，麻省理工学院的一位研究人员约瑟夫·魏岑鲍姆（Joseph Weizenbaum）曾在一本名为《计算机的能力与人类的理性》（Computer Power and Human Reason）中，用“电脑流浪汉”来形容类似人工智能实验室里的这批人，并对此多有指责：“衣服打着褶，头发不洗，胡子不刮，蓬头垢面，这群人不管自己的形象，更不会关注外面的世界。这种电脑流浪汉就存在于我们的现实世界之中，他们沉迷于计算机，一切生活只能依赖电脑，而他们的日子里只有计算机³。”

如今这篇文章已经发表了二十多年，而理查德听到“电脑流浪汉”这词依然心存不快。一说起这事，理查德就用一般现代式来形容，就好像魏岑鲍姆在他身边一般：“他就希望大家把这当成一份职业。你干

这活就该只为挣钱，干完活你就走人，回到家就什么都不记得。他所谓的这种正常生活，在我看来根本就是个悲剧。”

然而，黑客的生活中，也依然会有悲剧。理查德回忆，在他从人工智能实验室的兼职黑客，到转成全职黑客的这些年间，经历了一系列不平静的遭遇。他只能靠玩弄计算机来度过这段难熬的日子。正如他之前说的，第一个不幸遭遇就是从哈佛毕业。之后他凭着对物理学的兴趣，在麻省理工学院开始攻读物理系的博士学位。选择麻省理工学院是个非常顺理成章的事情。一来，有众多著名校友做榜样：有 1936 年毕业的校友，晶体管发明者之一威廉·肖克利；1939 年毕业的校友，著名的理论物理学家理查德·费曼；1951 年毕业的校友，夸克之父默里·盖尔曼。除了这些名人以外，吸引理查德的，自然还有人工智能实验室和里面那台全新的 PDP-10 计算机。理查德说：“我那会越来越喜欢编程，但当时觉得，我没准能两者兼顾，编程、物理两不误。”

白天读着物理学的博士课程，晚上跑去人工智能实验室鼓捣计算机，理查德试着在这种生活里保持平衡。两头的忙碌，让他只能在每周末偷出几分闲暇，跑去民间舞俱乐部，放松一下身心。而这个活动，也成了他唯一能认识异性朋友的机会。然而，在麻省理工学院的第一年结束时，悲剧再次降临。他膝盖受伤，从此不能再跳舞。一开始，他还以为只是个小伤，养一阵子就会好。他还是照常参加俱乐部活动，聊聊天，听听音乐。可暑假结束，他膝盖依旧疼痛，加上新学期课程开始，理查德开始担心。他回忆：“我膝盖一直也没见好，这就意味着我很可能终生都无法跳舞了。我当时伤心透了。”

离开了大学时的宿舍，又没法继续跳舞，理查德丧失了任何社交的机会。跳舞是唯一能让他在女性面前获得成就感的活动，他甚至有时候还能借此和女生单独出来约会。不能跳舞本身已经让他够痛苦了，而这也同样意味着，他从此恐怕就没机会和女生约会了。

“我当时就觉得自己像个泄了气的气球。我失去了做各种事的动力，我彻底地绝望了。”

这之后，理查德再次远离了这个世界，专心投入人工智能实验室的工作。1975 年十月，他放弃了麻省理工学院的课程，从此没再回到课堂。软件开发，曾经只是他的兴趣，如今则成了他的责任。

回头看看这段往事，理查德觉得，从一个学生转变成一个全职黑客几乎是命中注定的。他觉得，自己早晚都会放弃其他任何追求，专心成为一名黑客。他说：“物理和数学领域里，我很难做出一些自己原创的贡献。倘若我能在那里有所成就，我自然乐见其成。可我始终没能在这些领域里，找出自己的套路。我甚至都不知道从何开始。而在软件领域，我立即就能知道怎么写软件，怎么让它跑起来，怎么做出有用的东西。软件领域里的知识，让我有了继续深究下去的动力，也让我从中感受到了无尽的乐趣”

这份乐趣，并非只有理查德一人能感受到。很多黑客早已乐此不疲。人工智能实验室里，充斥着很多这样的人，他们和理查德一样，都在学业中途，转职成了全职黑客。其中大多数人都是数学或电子工程专业出身，他们当初选专业，一来为拿个文凭，二来可以体验解决旷世难题后的那份欣喜。当年圣·托马斯·阿奎那（St. Thomas Aquinas）编写《神学大全》时，曾号称感受过上帝的异象。而这份体验，黑客们也常有感受。他们通过集中精神，身心齐动，也达到了内心上的超脱，开始了忘我地工作。虽然理查德和其他黑客都拒绝毒品，但在电脑前写上二十来小时的代码，也同样能给他们带来一份飘飘然的感觉。

这其中，最能让他们陶醉的，恐怕就是那份无法言状自我满足感。理查德简直天生就是个黑客。他少年时熬夜学习的经历，让他可以轻易胜任长时间的工作，并且只睡很少的觉。而他的数学背景，又让他有着过人的严密逻辑和数感。很多让其他黑客望而生畏的设计难题，放到理查德面前却变得轻而易举。

杰拉尔德·萨斯曼（Gerald Sussman）是人工智能实验室的教授，1985 年之后，他还担任了自由软件基金会董事会成员。回忆起理查德，他赞叹：“他太特别了。思维清晰，设计流畅。”萨斯曼曾邀请了理查德加入人工智能实验室，参与了 1973 年和 1975 年的项目。两个人工

智能的项目目标，都是通过编写程序，让计算机能够像人类的电子工程师一样，分析电路。这个项目需要一位 Lisp 语言的专家，同时，还能了解人类是如何解决类似的电路分析的问题。Lisp 语言曾是专门设计用来编写人工智能程序。1975 年的项目，则开辟了相关性制导回溯技术 (dependency-directed backtracking) 的先河，这个技术又名真值维护 (Truth maintenance)，基本思路是首先产生几个假设，然后检测在假设前提下是否存在矛盾，如果存在矛盾，则重新设计假设。

这些正式项目之外，理查德也会花时间维护着自己的个人项目。黑客们都喜欢改进实验室的各种基础软件。理查德当时手头最大的个人项目，就是实验室的一个编辑器软件，名为 TECO。

七十年代，理查德在 TECO 上的工作和之后的自由软件运动是一脉相承的。而这段历史在计算机史上，也是值得一书的。五六十年代那会，计算机刚刚进入大学校园，所谓编程还是个很抽象的概念。那会的程序员，如果想要和计算机沟通一下，就必须得拿着一大摞卡片，上面打着孔，记录着软件中的指令。他们得把这一叠卡片交给系统管理员，让管理员把它们一张一张插到计算机里。计算机执行完卡片上的指令，然后把结果用打孔的方式，输出在另外几张卡片上。程序员则拿着这些计算机输出的卡片，回去分析解读。这个流程，通常被成为“批处理”。批处理是个非常费时费力的活，而且也给了计算机管理员太大的权力。黑客们痛恨权威的传统，恐怕很大一部分原因，要归咎于当时的计算机管理员权力过大，他们有权决定哪个程序优先运行。

1962 年，很多计算机科学家和黑客都参与了麻省理工学院的 MAC 计划 (Project MAC)。这一计划是人工智能实验室的前身，它曾试图解决批处理带来的问题。MAC 计划引入了“分时” (Time sharing) 的概念。这个概念之前曾叫做“偷时” (Time stealing)，它利用一个程序运行中间的空隙，来执行另外一个程序，由此多个程序可以有效利用计算机。同时，电子打字机也被引入进计算机系统，作为人际交互的设备。从此人们再也不用靠打孔来和计算机交互，人们可以利用打字机，把命令敲进去，然后等着计算机把结果一行一行打印到纸上。

二十世纪六十年代，交互界面的设计也有了长足进步。在 1968 年，一次讲座从此出名。讲座上，斯坦福研究中心的科学家，道格·英格巴特（Doug Engelbart）展示了第一款现代图形用户界面的原型。他们把计算机和电视机连起来，并且还加入了一个定点设备——英格巴特把这个定点设备昵称为“老鼠”，也就是我们今天的鼠标。这套系统的交互性比麻省理工学院的分时系统更好。他们把电视机当作一个高速打印机，来显示各种输出。这套系统还允许用户使用鼠标来移动屏幕上的光标，并且实时地显示出光标的位置。用户可以使用鼠标把光标移动到屏幕上的任何字符上。

不过，这些发明要等二十年才能进入市场。到了七十年代，显示器逐渐开始作为显示设备，取代了电子打字机。这一改变，使得计算机可以使用全屏幕来显示。这就不必像以前一样，每次只打印几行内容在纸上。

人工智能实验室的 TECO 程序，是早期的几个使用全屏幕显示的程序之一。TECO 是“文本编辑及修正程序”（Text Editor and COrrector）的缩写。它的前身，诞生于 PDP-6 和电子打字机的年代，是黑客们把它一步一步升级至今⁴。

比起以前的那些编辑器，TECO 是一大进步，但仍旧存在不足。如果要创建并编辑某个文件，程序员必须要输入各种命令，才能完成不同的操作。这个过程非常抽象。今天的编辑器，你每敲进一个字母，都会屏幕上显示出来。而使用 TECO 来编辑文件，则需要输入一些命令，然后告诉它“命令结束”，然后才能把文件修改好。经过一段时间的练习，一名黑客可以使用一套漂亮的命令来完成很大的修改。不过这种技能，正如理查德所说，是需要“类似下盲棋一样的脑力消耗⁵”。

为了辅助这种编辑流程，人工智能实验室的黑客们开发了一套系统，可以把屏幕分为两部分，分别显示正在编辑的文件内容，和输入的命令。这个小改变的确有用，可是想要使用 TECO 依旧需要很多技巧，和事先规划。

在当时，除了 TECO，还有几个其他的全屏编辑器。1976 年，理查德去了一趟斯坦福的人工智能实验室。在那里，他见到了一个叫做 E 的编辑器。这个程序有个功能，可以在根据用户的输入，实时更新显示器上的内容，让用户看到最新的修改结果。用七十年代的话讲，E 是早期的几个“所见即所得”的编辑器之一。所谓“所见即所得”，常常写作 WYSIWYG，是 What you see is what you get 的缩写。它意味着用户可以直接在显示出来的文本上进行编辑，而不用再另外使用一个后台的编辑器程序⁶。

理查德一下子对这技术感兴趣了，他决定回到麻省理工学院之后，把这个功能加到 TECO 上。他发现 TECO 上有个功能叫做“Control-R”。这个功能是卡尔·米克尔松（Carl Mikkelsen）开发的。使用这个功能的快捷键和它的名字一样：Control-R。利用这个快捷键，用户可以让 TECO 实时地显示出编辑的内容。不过有个小瑕疵：用户只能使用五行来显示编辑的内容，这显然是不够的。理查德重新实现了这个功能，让程序可以使用整个屏幕来显示编辑内容。他还做了个不起眼，但影响深远的修改：他还允许用户把一连串 TECO 命令绑定在快捷键组合上。这一连串的命令称为“宏”。TECO 的资深用户早就把各种常用的命令组合记下来了。理查德的这个改变让他们可以通过宏更快捷地使用这些组合。最后的结果，是一个允许用户自己扩展的“所见即所得”的编辑器。盖·斯蒂尔（Guy Steele）曾是当年人工智能实验室的一名黑客，他回忆起理查德的这个改动，说：“这在当时是个突破⁷。”

理查德回忆，加入“宏”之后，各种创造接踵而来。他说：“大家都开始把各种常用命令的组合写成宏，然后把各自的宏互相分享，大家一起不断改进，再分享。这样，这些宏越来越强大，涵盖了很多常用操作。这些宏俨然成为了一套单独的系统软件⁸。”

越来越多的人开始使用宏，把各种宏加入到自己的 TECO 编辑器上。随着大家对宏的狂热，TECO 作为编辑器的功能反而倒在其次了。理查德说：“我们开始意识到，TECO 已经不仅仅是个编辑器，更是个

编程语言。”用户不断改进自己的想法，不断交流分享，大家身在其中，自得其乐⁹。

又过了两年，各种创新最后带来了一些麻烦。大家互相交流分享，使用的宏越来越多。每个人手头都有自己的一套宏，由此就引来了各种兼容性问题。盖·斯蒂尔说：“我们遇到了沟通障碍。”

这种障碍阻碍了大家的交流，盖·斯蒂尔说。黑客们当年设计了 ITS 系统，其中一个很重要的设计原则，就是帮助程序员们分享知识，互相改进各自的工作。这就意味着，程序员可以坐到另外一个同事的电脑前，打开这个同事的程序代码，直接修改代码，或者加上几行注释。斯蒂尔解释说：“有些时候，要想教人怎么写程序，或者怎么调试某个程序，最好的办法就是坐在电脑前，实际演示给他们看。”

引入宏之后的第二年，这种做法变得越来越难。黑客们开始在自己的 TECO 编辑器里加入各种宏，不断扩展 TECO 的功能。这就导致你坐到一台电脑前，第一件事情是需要了解每个宏都是做什么的。

一番受挫之后，盖·斯蒂尔决定着手解决这个问题。他收集到四套常用的宏，打算根据每套宏实现的功能，列出常用的功能，写成一份文档，并根据这四套宏，实现出一套符合这份文档的宏。在实现的过程中，盖·斯蒂尔的工作吸引了理查德的注意。

盖·斯蒂尔回忆说：“他站在我身后，看着我的屏幕，问我在做什么。”

盖·斯蒂尔是个说话温和的黑客，并没怎么和理查德打过交道。回忆起当时的情景，他依旧记忆犹新。在当时的人工智能实验室里，站在别人身后看看别人的工作，是个非常平常的举动。理查德，在当年是 TECO 的维护者。看到盖·斯蒂尔的工作，他觉得非常有趣，于是决定帮助他完善这个工作。

说起这事，盖·斯蒂尔笑道：“我常跟人们这么说，我做了最初的百分之零点零零一的工作，理查德把剩下的都做了。”

这个名字被称为 Emacs，是理查德建议的名字。Emacs 是 Editing macros 的缩写，意思是“宏编辑器”。它标志这宏出现之后的

又一个进步。它也参考了当年各种软件的名字，理查德注意到，在 ITS 系统上，还没有哪个软件名是以 E 开头的。把它叫做 Emacs，用户就可以自己在设置中把它叫做 E，一个字母就能运行这个程序。这又得归咎到黑客简约的风格上了¹⁰。

当然，一开始，并不是所有人都转头使用 Emacs。用户依旧可以继续使用他们以前的 TECO 编辑器。不过大多数人都觉得转用 Emacs 更加方便。Emacs 有着更强的扩展性，用户可以按照自己的需求，轻松地替换或者增加某些功能。

盖·斯蒂尔说：“一方面，我们试图创建一个统一的命令集。另一方面，我们还希望程序能有很强的扩展性，因为这一点至关重要。”

这个时候，理查德又遇到了一个难题：如果用户做出了修改，但是并不把这些修改分享出去，沟通障碍早晚会再次出现。本着黑客的分享精神，理查德在源代码里写上了使用条款。用户可以自由地修改和分发这个软件的代码，但必须把所有的改动都发回来。理查德把这称作“加入 Emacs 公社”。当年的 TECO 最后变得不仅仅是个编辑器，如今这个 Emacs 则变得不仅仅是个软件。对于理查德来说，这更是个社会契约。在 1981 年的一份备忘录上，理查德在 Emacs 这个项目下写道：“Emacs 是以分享的形式发布的。对 Emacs 的各种修改，必须要发给我，以便加入到以后的版本中。”¹¹

Emacs 原本只能运行在 PDP-10 上。不过很快，其他平台上的用户也希望能运行 Emacs 编辑器。之后十年，各种创造依旧继续。带来了各种类似 Emacs 的编辑器，运行在不同平台上。Emacs 公社的规矩则不适用于这些编辑器上，因为它们完全是另外一套代码。这些编辑器中，有些在名字上就提及了理查德的 Emacs 编辑器。比如：Sine，全称 Sine is not Emacs；Eine，全称 Eine is not Emacs 以及 Zwei，全称 Zwei was Eine initially。真正的类 Emacs 软件必须要提供用户可编程的扩展。一些 Emacs 的克隆软件只是使用了 Emacs 的快捷键，并没有提供扩展机制，这些一般被成为“伪 Emacs”。其中一个例子就是

Mine, 全称 Mine is Not Complete Emacs, 意为“Mine 不是一个完成的 Emacs”。

理查德在人工智能实验室开发 Emacs 的时候, 黑客圈子里还有很多其他的开发, 也干得热火朝天。正如第一章提及的那样, 1979 年, 布莱恩·瑞德 (Brian Reid) 决定在 Scribe 里放入“定时炸弹”, 这样就可以让没有付费的用户无法使用这个软件, 以此让 Unilogic 公司获利。布莱恩的这个决定在理查德看来不是个好兆头。布莱恩回忆起理查德对此的看法, 说: “他觉得, 这简直就是法西斯行为。”布莱恩之后创立了 Usenet 新闻组上的 alt 分支, 可每次提起理查德的指责, 他依旧愤愤不平: “他说所有的软件都得免费, 他觉得所有软件的收费行为都是违背人性的¹²¹³。”

虽然理查德无法阻止布莱恩给用户装上“定时炸弹”, 可他还是有一套法子, 来遏制其他类似的行为。作为 Emacs 代码的主要维护者, 理查德开始利用他的力量, 来发起一轮政治影响。当时, 为了去掉系统的登录密码, 他正跟计算机系的机房管理员闹得不可开交。在冲突的关键时刻, 他发起了一轮“软件抗议”: 如果实验室的成员不反对系统登录密码, 理查德就不给他们 Emacs 用¹⁴。理查德的这种做法, 只是做出一个姿态。因为如果哪个人想用 Emacs, 他们还是可以自己动手安装的。理查德是想借此传达一个信息: 如果你在 ITS 系统上使用密码, 那么将会被人嫌弃, 受人指责。

在接受《黑客》一书作者史蒂芬·李维 (Steven Levy) 采访时, 理查德回忆说: “很多人觉得我绑架了实验室的各位成员, 我在勒索大家。从某种意义上说, 我的确如此。因为我觉得那些管理员在用暴力威胁着每一个人, 我只能以暴制暴¹⁵。”

渐渐地, Emacs 成了黑客文化的宣传品。Emacs 有着极强的扩展性。这种扩展性增进了用户之间的交流合作。不仅如此, Emacs 甚至要求这样的合作。如果用户不把自己的修改贡献出来, 就很可能用不到最新的版本, 用不上最新的功能。而每次更新, 各种新功能可是不少。1984 年, 理查德又重写了 Emacs, 命名为 GNU Emacs。如今, GNU

Emacs 的用户遍布全球，大家把它已经扩展得异常强大。用户可以把它用作电子表格，当作计算器，用作数据库，用作网络浏览器等等。以至于之后的 Emacs 开发者，都找不出什么词来概括 Emacs 的功能了。理查德说：“这就是我们想要传达的。Emacs 里包含的东西既有用，又有趣。”

和理查德同时代的人工智能实验室成员更是感谢理查德的贡献。哈尔·埃布尔森（Hal Abelson）曾是麻省理工学院的一名博士。七十年代，他曾在萨斯曼教授手下做过研究。之后，帮助理查德成立自由软件基金会，并成为董事会成员。形容起 Emacs，他说：“那绝对是个精品佳作！”他描述，Emacs 既可以让程序员不断添加新功能，还不会影响整个系统。理查德的经验，为未来大规模合作开发的软件工程铺平了道路。“它结构稳定，可以接受世界各地的人贡献代码。这种松散的大规模协作开发，恐怕要算前无古人了¹⁶。”

盖·斯蒂尔（Guy Steele）对此也表示了钦佩。如今，盖·斯蒂尔已经是 Sun 公司的一名科研人员。回忆起理查德，他脑海中浮现的是“一位才华横溢的程序员，他可以瞬间写出大量几乎没有 bug 的代码。”尽管他和理查德性格不太合得来，可他们俩还是合作了一段时间。这段时间里，盖·斯蒂尔对理查德的那种暴风闪电般的编程风格印象深刻。他记得在七十年代末，他俩曾一起为 Emacs 添加“重排版”的功能。所谓“重排版”，最初是斯蒂尔的点子。它可以让用户使用快捷键，来为 Emacs 中编辑的代码重新排版，使得代码更加易读。这个功能更增强了软件“所见即所得”的质量。这个功能一下子吸引了理查德的注意，接着斯蒂尔和理查德决定一起开发这个功能的改进版。

斯蒂尔回忆：“我们早晨开始坐到电脑前。我敲键盘，他就坐我身边，告诉我怎么写。”

俩人就这样，持续开发了 10 个小时。斯蒂尔说，这 10 个小时，他们俩人谁也没休息，甚至都没聊多余的话。到最后，他们把整个功能精简到一百多行代码。斯蒂尔回忆：“我当时手指就没离开键盘。我就

觉得我俩的想法直接就流到了屏幕上。他告诉我写什么，我就按他说的写。”

整整 10 个小时，光是这时间长度，就足够显示出理查德的编程风格了。斯蒂尔离开实验室，走出技术广场 545 号大楼。外面已是夜幕降临，他虽然早就习惯了这种马拉松式的编程，但这次却别有一番滋味。跟理查德一起工作，他必须要集中精力，心无旁骛。如今回忆起来，斯蒂尔说，理查德的这份才智和精力，既让人振奋，又令人生畏。“我回想起来，第一感觉是觉得那是个很好的经历，流程紧凑，快速高效。再细想想，妈呀，我可不想再来一次。”

Endnotes

¹ 参见乔希·麦克休 (Josh McHugh) 的文章《黑客之爱》，《福布斯》1998 年 8 月 10 日。http://www.forbes.com/forbes/1998/0810/6203094a.html

² 参见理查德在瑞典皇家技术研究所的演讲 (1986 年 10 月 30 日): http://www.gnu.org/philosophy/kth.html

³ 参见约瑟夫·魏岑鲍姆 (Joseph Weizenbaum) 的书，《计算机的能力与人类的理性》(Computer Power and Human Reason) (W.H. Freeman 出版社，1976 年)，第 116 页。

⁴ 根据《黑话词典》(Jargon File) 中的记载，TECO 的名字原意为“磁带编辑及修改器”(Tape Editor and Corrector)，参见 http://www.catb.org/jargon/html/T/TECO.html

⁵ 参见理查德·斯托曼，人工智能实验室备忘录 (1979 年)，《EMACS：可扩展，可定制的全屏编辑器》。本书的引文来自：http://www.gnu.org/software/emacs/emacs-paper.html

⁶ 参见理查德·斯托曼，《Emacs 全屏编辑器》(1987 年)，http://www.lysator.liu.se/history/garb/txt-1-emacs.txt

⁷ *Ibid.*

⁸ *Ibid.*

⁹ *Ibid.*

¹⁰ *Ibid.*

¹¹ 参见理查德·斯托曼，人工智能实验室备忘录 (1979 年)，《EMACS：可扩展，可定制的全屏编辑器》。本书的引文来自：http://www.gnu.org/software/emacs/emacs-paper.html

¹²理查德在 1996 年接受《MEME》杂志采访时，曾说 Scribe 的出售方式令人生厌。但他没有提及作者布莱恩·瑞德的名字。他说：“问题是没有人指责这个学生的行为，大家反而纷纷效仿。”参见《MEME 杂志》2.04，<http://memex.org/meme2-04.html>

¹³译注：布莱恩此处对理查德的理念理解有误。理查德所说的是，所有软件必须是自由（free）的，而非免费的。然而英文中，自由和免费一次都是 free。因此很多人都把理查德的思想误解为“软件不能收费”。实际上，理查德自己也曾出售过 Emacs 的拷贝。关于自由软件可以出售的观点，参见 <http://www.gnu.org/philosophy/selling.html>

¹⁴参见史蒂芬·李维（Steven Levy），《黑客》（1984 年，美国企鹅出版社），第 419 页。

¹⁵*Ibid.*

¹⁶写作本章时，我刻意把重点放到 Emacs 在社会上的影响里，而不是它的技术层面。如果想了解更多技术层面内容，参见理查德 1979 年在人工智能实验室的备忘录。尤其推荐《关于软件部署后的开发流程研究》一节（Research Through Development of Installed Tools）。非技术人员也可以读懂这一节的内容，而且它很好地同时反映出了理查德的技术观点和政治观点。以下为节选：“EMACS 的开发并非借助于事前的精心设计。因为这样的流程最多只能达到一开始计划的结果，也只能满足一开始设定的需求。我在开发出这款可扩展的编辑器之前，并不能预见这种编辑器的存在，也并不能预估它的价值。EMACS 之所以有今天，因为我在不断地改进它，并以此到达未曾想过的结果。”

Chapter 7

道分左右义无旁支

1983 年 9 月 27 日，众多计算机用户像往常一样，登录 Usenet 的 net.unix-wizards 新闻组。一条不太寻常的消息映入眼帘。这条消息在当天凌晨过后发出来，准确时间是零点三十分。消息来自署名 rms@mit-oz 的用户。标题甚是简单，却异常乍眼：《重写 UNIX 系统》（New UNIX implementation）。这个消息并非是介绍新的 UNIX 发行版，而是更像一篇邀人入伍的檄文，文章开头写道：

这个感恩节假期，我将开始写一个完全的 UNIX 兼容系统，叫做 GNU，意为“GNU's Not Unix (GNU 不是 UNIX)”。这个系统任何人都可以自由使用。你如果愿意贡献时间，金钱，程序或者设备，我随时欢迎¹。

对于 UNIX 的资深用户来说，这个消息显得太过理想主义，甚至有些自大傲慢。面对已经非常成熟的 UNIX 系统，这则消息不仅号称要从头克隆一套类似的操作系统，甚至还要在开发过程中改进现有 UNIX 的设计。这则消息的作者声称，新的 GNU 系统，会提供各种常用软件。包括文本编辑器，用来执行各种命令和程序的 shell，编译器，

还有“一些其他的东西²”。除此以外，它还提供很多原本 UNIX 并不提供的功能，也非常吸引眼球。包括一套基于 Lisp 语言的图形用户界面；一个防崩溃的文件系统；以及一套基于麻省理工学院内部网络系统的网络协议栈。

“GNU 可以运行现有的 UNIX 程序，但和现有的 UNIX 并不完全相同，”作者写道：“我们会在开发过程中，融合我们的经验和对其他操作系统的了解，不断改进现有的 UNIX 系统设计。”

读者的各种质疑显然早在意料之中，整个消息最后，作者加入了一段《我是谁?》的自我介绍：

我是理查德·斯托曼，我创造了众人效仿的 EMACS 编辑器。现在在麻省理工学院人工智能实验室。我的工作涉及编译器，编辑器，调试器，解释器，还有非兼容分时系统 (ITS) 以及 Lisp 机的操作系统。我创造了 ITS 上早期的独立于各终端硬件的显示技术。我还在 Lisp 机上，实现了一个防崩溃的文件系统，和两个窗口系统³。“

似乎是命中注定一般，理查德的 GNU 系统没能赶在感恩节假期结束前发布。不过理查德依旧赶在 1984 年 1 月的时候做出了不小的成果，他自己也融入到了 UNIX 软件开发的世界之中。对于这位从 ITS 世界来的架构师来说，在 UNIX 中设计软件似乎更像是设计城郊购物中心，而非设计紫禁城皇宫。即便如此，设计成类 UNIX 系统依旧有它的优势。ITS 的确强大，但仍有一处致命缺陷：麻省理工学院的黑客们创造的 ITS 系统，是专门针对当时强大的 DEC PDP-10 计算机设计。可到了八十年代，人工智能实验室的管理员决定淘汰掉实验室的 PDP-10 计算机。当年辉煌一时的 ITS 系统也因此成了一座无人的鬼城。而 UNIX 的设计则与此不同。它的设计非常强调移植性，它并不依赖于某个特定硬件环境。因此，面对硬件更替，UNIX 则毫无压力。UNIX 最初是由当时 AT&T 贝尔实验室的几个年轻科学家创造。之后，这套系统被大家纷纷传阅，不断改进。在资金并不富裕的学术

界，UNIX 系统甚是流行。和麻省理工学院的同僚们不同，UNIX 的作者可用的硬件资源非常有限。他们必须把软件设计得可以在各种良莠不齐的硬件上运行无阻。最初是针对 PDP-11 系统设计，PDP-11 是 DEC 推出的另一种 16 位计算机。在麻省理工人工智能实验室的黑客们看来，它顶多只能用来跑些小程序。之后，UNIX 又陆续开始支持 32 位计算机，比如 VAX11/780。到了 1983 年，一些公司开始推出更强大的一代被称作“工作站”的桌面计算机。其中以 Sun 公司产品最为引人注目。这些计算机运行 UNIX，有着更小的体积，却和当年的 PDP-10 性能相当。

为了增强可移植能力，UNIX 的开发者在软件和计算机之间，加入了一个抽象层。和人工智能实验室的 ITS 系统不同，UNIX 并非使用特定于某个硬件平台的指令集开发。它的开发者们创造了一种被称为“C 语言”的更抽象的编程语言，借此屏蔽了底层的硬件差异。开发者们于是可以放眼全局，集中设计各个组件之间的协调机制，设计组件的接口，而不必把精力分散到每个组件的开发和移植上。他们由此创造了一套系统，可以轻易地移植到几乎各种计算机上。如果某个用户对哪个组件并不满意，他们可以按照定义好的接口，修改现有组件，甚至重新开发一个同样功能的组件，再把改进的组件放回原位，一切依旧会运行正常。简而言之，UNIX 的设计具备极大的移植性，也促进了计算机这个市场。这一切，也带来了 UNIX 的蓬勃发展⁴。

理查德之所以决定开发一套名为 GNU 的类 UNIX 系统，是因为人工智能实验室已经停止使用 ITS 了。而整个实验室的黑客文化，也伴随这 ITS 的消亡，渐渐分崩离析。这个改变对理查德来说，可谓打击重大。施乐公司打印机事件，让他意识到了专有软件的不义。而实验室的黑客社区几乎解散，则让理查德必须面对抉择：要么在专有软件面前投降，要么起身反抗它。

ITS 的消亡过程历经了很长时间。1980 年的时候，实验室里的大部分黑客都在开发 Lisp 机（Lisp Machine）和运行其上的操作系统。

Lisp 是一种非常优雅的编程语言。它最初由人工智能领域的先驱, 约翰·麦卡锡 (John McCarthy) 发明。二十世纪五十年代, 他曾是麻省理工学院人工智能领域的科学家。他发明的 Lisp 语言, 非常适合编写复杂程序, 来处理不具备很好结构的数据。Lisp 这个名字来自 LIsT Processing, 即链表处理。之后, 约翰·麦卡锡离开了麻省理工学院, 去了斯坦福大学的人工智能实验室。麻省理工学院的黑客们则改进了 Lisp 语言, 并创造了一个 Lisp 方言, 名为 MACLISP。其中的“MAC”, 指的是“MAC 项目”。MAC 项目是一个由美国国防部高等研究计划局 (DARPA) 资助的项目。借助这个项目, 诞生了如今的人工智能实验室。整个实验室由黑客理查德·格林布拉特 (Richard Greenblatt) 领导。在七十年代末期, 他们设计出了专门用来高校地执行 Lisp 程序的计算机, 命名为 Lisp 机。接着, 开发了一整套基于 Lisp 的操作系统。

到了 1980 年, 两组互相竞争的黑客各自成立了两家公司, 分别制造和销售 Lips 机。理查德·格林布拉特成立了“Lisp 机公司” (Lisp Machines Incorporated)。他试图避免引入外界的投资, 创造一个真正的“黑客公司”。另外一大部分黑客们, 则加入了名为 Symbolics 的传统创业公司。到了 1982 年, 这些黑客则完全放下了在麻省理工学院的工作。

随着黑客们的离去, 实验室的程序和机器要么没人修, 要么也要等上好久才有人打理。理查德·斯托曼说, 更糟糕的是, 实验室正经历了前所未有的“人事变动”。以前, 黑客们虽说人少, 却是实验室中很重要的一伙人。如今, 黑客几乎绝迹, 而“不喜欢 PDP-10 的教授和学生则依旧那么多⁵。”

1982 年, 人工智能实验室接到通知, 要求替换那台已经服役了 12 年的 PDP-11 计算机。PDP-11 曾是迪吉多公司 (Digital Equipment Corporation, 简称 DEC) 出品的 16 位计算机。如今迪吉多公司的主打产品是 Decsystem 20。对于应用程序来说, Decsystem 20 和 PDP-11 兼容。但是如果想要在上面运行 ITS 这样的操作系统, 则需要投入大量劳力, 把系统从 PDP-11 移植到 Decsystem 20 上。如今的人工智能

实验室早已物是人非，实验室中的编程能手几乎都已走光。而实验室中的一些教授则大力鼓吹 Twenex 系统，它是一款由迪吉多开发的商业操作系统。如今的黑客们在人数上占了劣势，只好将就着用 Twenex 系统。

.....

几年之后，理查德回忆起那时的情景：“那些教授们说：‘没有足够的黑客们来维护 ITS 系统，我们将会面临各种灾难。要避免这些发生，我们只能投靠商业软件。我们可以让商业公司来提供维护。’后来的事实证明，他们的这番论调大错特错。可当时他们的确这么做了⁶。”

一开始，黑客们觉得 Twenex 只不过又是一个专制的标志。和之前那些专制标志一样，只要把它推翻，就万事大吉了。这个系统从名字看，就像是对黑客们的挑衅：迪吉多公司对这个系统的官方名字是 TOPS-20，意为“顶尖 20”。它是 TOPS-10 的后继产品，而 TOPS-10 则是迪吉多当年给 PDP-10 配备的标准系统。不过 TOPS-20 并非是在 TOPS-10 基础上做的开发。它的主要代码都继承自 BBN 公司（Bolt, Beranek and Newman）为 PDP-10 开发的 Tenex 系统⁷。理查德则由此把 TOPS-20 称作 Twenex，因为他实在不想用 TOPS-20 这个名字。他说：“这个系统可根本算不上‘顶尖’，我说什么也不会这么夸它。所以我在 Tenex 里添了个字母 w，把它叫做 Twenex。”

在黑客界，人们把跑着 TOPS 系统的 Decsystem 计算机戏称作 Oz，意为“奥兹国”。奥兹国这名字源自《绿野仙踪》这部小说。因为在黑客界，人们都说这样一台计算机需要有个 PDP-11 作为终端机才能运行。于是，人们看到这台计算机连着 PDP-11 作为终端，就不禁想到了《绿野仙踪》中那位吹牛皮的奥兹国大法师。黑客们常戏谑道：“我是伟大全能的奥兹国大法师！我可没看到身后那台 PDP-11。”⁸

黑客们当年第一次看到 Decsystem 的时候也许还会借机嘲笑两句，可很快他们就在 Twenex 面前笑不起来了。Twenex 系统增进了系统的安全强度，而且 Twenex 上的软件也都考虑了各种安全措施。当年，黑客们试图避免设置任何密码和安全设施，那些行为就好像在和

管理员玩猫捉老鼠的游戏。而如今，他们面对的，则是一场彻头彻尾的战斗了。系统管理员辩解说，如果没有这些安全设施，Oz 系统就随时都会面临崩溃。而黑客们则辩解说，避免崩溃的最好办法，就是公开程序的源代码。遗憾的是，黑客人数上不占优，而且大多数黑客也逐渐失去了当年的那份果断。最终，系统管理员还是胜利了。

一开始的政策是，任何一个人工智能实验室的成员，都拥有一个“特权车服务”（wheel privilege），可以用来绕过系统的安全限制。可是，任何有“特权车”的人，都有权吊销别人的“特权车”。这样一来，就促使一小撮黑客总是尝试吊销其他人的特权，借此掌控全局。

见得此状，理查德破解了密码，利用开机时的调试器，成功破坏了几个这样企图夺权的阴谋。在第二轮“政变”之后，理查德向全体人工智能实验室的人发出了警告信⁹。

信中写道：“还有另外一波强大的力量，企图剥夺我们的权利。不过现在，这些官僚们已经暂时被我们打败了。“这是封匿名信，信底的签名是“自由奥兹国电台”。

不过，所谓的匿名信并没把理查德挡在幕后。到了 1982 年，理查德关于密码和安全设施的抗议早就内外闻名。以至于有不少人工智能实验室以外的人，通过 ARPAnet 网络，使用理查德的登录帐号，访问人工智能实验室的计算机。他们当初使用的 ARPAnet，是如今互联网的雏形。它曾是一个研究项目，旨在构造一个大型计算机网络。这个网络之后不断发展，最终成了今天大家见到的互联网。唐·霍普金斯（Don Hopkins）在八十年代是加利福尼亚州的一名程序员，他当年就曾利用理查德的帐号访问人工智能实验室的计算机。他当年从黑客圈的小道消息里听说，想要使用麻省理工学院大名鼎鼎的 ITS 系统，只需要使用一个简单的用户名和密码登录就可以：用户名，RMS；密码，RMS。

霍普金斯说：“麻省理工学院让我和很多其他的人免费使用他们的计算机，这让我受益终生。这在当时，对大家来说可是份厚礼。”

这种允许“游客”的政策，在 ITS 时代还能被麻省理工学院的管理人员容忍¹⁰。可等到 Oz 成为实验室连接 ARPAnet 的主节点后，这行为就被逐渐禁止了。一开始，理查德依旧使用原来简单的用户名和密码，这样外部人员还是可以访问实验室的计算机。但是脆弱的 Oz 禁不起这么折腾，很快管理员就禁止了外部人员的访问。因为有些访客会有意无意地把整个系统搞垮。再后来，管理员们严重警告了理查德公开用户名和密码的行为。对此，理查德并没有放弃使用 Oz 系统，而是在回应中强调了自己的底线¹¹。

理查德之后说：“当年人工智能实验室第一次要求设置密码，我依旧坚持自己的原则：计算机上不该有密码。我也因此坚决不会帮助他们维护一个充满安全设施的计算机系统。”¹²

八十年代早期，理查德的这次抗议之后，系统管理员掌控下的伟大全能的 Oz 系统和实验室的黑客之间，关系更加紧张。不过随着时间的流逝，这种紧张关系反而被黑客圈子内部的矛盾逐渐取代。在 Decsystem 20 来到实验室的时候，实验室的黑客圈子已经俨然分成了两派：LMI 派和 Symbolics 派。

LMI 和 Symbolics 分别是两家做 Lisp 机的计算机公司。Symbolics 得到了一些外部的投资，它招募了很多人工智能实验室的黑客，把其中的一部分黑客安排去改进 Symbolics 的 Lisp 机的操作系统。八十年代末期，这家公司已经吸纳了人工智能实验室的 14 位黑客作为兼职顾问，负责开发他们的 Lisp 机。剩下的黑客们，除了理查德以外，则都在 LMI 公司就职¹³。理查德则更享受人工智能实验室里没有压力的工作环境，更不想把自己划进哪个阵营。所以他哪个公司都没加入。

一开始，这些在公司干活的黑客们还会抽出些时间，在麻省理工学院继续做些工作，也会给麻省理工学院的 Lisp 机的操作系统贡献些代码。LMI 和 Symbolics 的 Lisp 机操作系统都是从麻省理工学院的 Lisp 机操作系统上衍生而来。在使用许可证上，麻省理工学院要求两家公司必须允许麻省理工学院使用它们的操作系统。但是，这两家公司可以禁止麻省理工学院再发布他们的产品。不过，1981 年间，两家

公司之间倒是有个不成文的君子协议，他们都允许麻省理工学院使用并发布两家公司的代码。这样，麻省理工学院的 Lisp 机操作系统就包含了来自外部公司的各种改进，而基于许可证，改进后的系统代码有可以继续流入各家公司。这相当于公司之间可以共享代码。如此，那些同时在麻省理工学院和公司工作的黑客们，就可以继续保持中立。

1982 年 3 月 16 日，理查德清晰地记得这一天，那是他的生日。Symbolics 的主管们决定不再遵循当初那君子协议。他们禁止麻省理工学院发布包含 Symbolics 公司的代码。此举旨在打击 LMI。因为在 LMI 公司就职的黑客比较少，所以 Symbolics 的主管们觉得 LMI 从共享的代码中获益。切断代码流通的途径，Symbolics 企图借此把 LMI 赶出市场。他们决定执行许可证上的权利。为了不让 LMI 获得自己的代码，他们仅允许麻省理工学院的学生运行他们的操作系统。借此，他们可以间接地让学院中的用户为自己做测试。同时，如果哪个用户对系统做了任何改进，则也会为 Symbolics 公司所用。

理查德在那时候负责维护实验室中的 Lisp 机，最开始的几个月还多亏了 LMI 公司的创始人，格林布拉特 (Greenblatt) 的指点帮助。当下，Symbolics 的做法可是惹恼了理查德。那些在 Symbolics 公司工作的黑客们都曾给实验室的 Lisp 机系统贡献过代码，如今，还有很多错误和 bug 遗留在这些代码中，甚至有些特性还都是半成品。如今，Symbolics 高层的做法无疑相当于下了“最后通牒”，令身在 Symbolics 的黑客们无法改进麻省理工学院的 Lisp 机系统。理查德也发起了反攻，他切断了连接实验室和 Symbolics 公司的微波通信信道，发誓再也不会 Symbolics 生产的机器上工作。并且决心要继续完善麻省理工学院的 Lisp 机系统上，以此帮助 LMI，打击 Symbolics。理查德回忆说：“当初我觉得，人工智能实验室是个中立国，就好像二战时的比利时一样。如果德国入侵比利时，比利时就要对德宣战，就得和英法一个阵营。”

Symbolics 公司的主管们逐渐意识到，自己公司操作系统的最新特性，总会被理查德克隆到麻省理工学院的 Lisp 机系统上。这让他们

甚为不满。理查德知道版权法这么个东西，所以他为学院的系统添加的特性，都是自己从头开始写的。他身在人工智能实验室，可以读到 Symbolics 操作系统提供给实验室的源代码。他会先读读 Symbolics 的代码，理解要解决的问题和解决方案，最后再自己重新写一遍，确保和 Symbolics 的实现完全不同。可 Symbolics 公司的主管们却不管这个。他们在理查德使用的计算机终端上安装了间谍软件，企图抓住理查德剽窃的证据。到了 1983 年年初，他们把这件事情告到麻省理工学院的管理层。可依旧拿不出多少像样的证据，只有一些看似类似的代码片段。

人工智能实验室的主管找来理查德，给他看了 Symbolics 公司指责他的证据。理查德一一否认，他说这些相似的代码都是 Symbolics 诞生之前，在学院 Lisp 系统里就有的。之后，理查德话锋一转：他自己已经克隆了 Symbolics 公司的众多特性，在那几千行的代码中，如今 Symbolics 公司却只能提供这么点证据来证明理查德剽窃，这恰恰说明理查德的确没有抄袭 Symbolics 的代码。人工智能实验室最终承认了理查德并没有剽窃，他也就继续开发到 1983 年¹⁴。

不过这件事倒是确实给理查德提醒了。“以防万一，对于新的特性和重大改变，我不再参考 Symbolics 的代码，而只是参考他们提供的用户文档来做开发。”对于一些重大的新特性，理查德则往往在文档发布之前，就着手设计开发。等到 Symbolics 公司发布文档，他再对代码修改，以便兼容 Symbolics 的接口。之后，如果 Symbolics 提供了补丁，他再阅读补丁的代码，以便确认自己的实现中是否存在类似 bug。如果存在，则尝试用不同的实现来解决。

整个过程也坚定了理查德的决心。理查德的开发，逐渐把人工智能实验室中的成员拉回到麻省理工学院的 Lisp 机系统上。这些用户也为理查德持续地提供错误报告。麻省理工学院继续允许 LMI 直接访问学院 Lisp 机系统的代码。理查德说：“我当初就觉得，我要是这辈子只能再做一件事，那我也要惩罚 Symbolics。”这话说得甚是坦白。显

然，理查德不是个和平主义者。随着矛盾的升级，给双方带来的情感变化也逐渐加大。

由此带来的绝望感，让理查德又有了家园被毁的感受。人心散了，人工智能实验室的黑客文化也逐渐褪去。之后在与史蒂芬·李维的采访邮件中，理查德把自己形容为 Ishi——加利福尼亚州最后一位 Yahí 族印第安人。这样的说法倒是为理查德的经历凭添了几分史诗色彩¹⁵。不过在 Symbolics 工作的各位黑客们则另有看法。他们并非觉得 Symbolics 好像要上灭下绝。很多理查德当年的同事，都觉得 Symbolics 倒是迟到的正义。在把 Lisp 机商业化的过程中，Symbolics 把黑客文化中的“工程师做主”的原则从象牙塔中带入了商业公司里，让公司里不再有“外行领导内行”的现象。其他黑客们也没觉得理查德是在捍卫什么，而是觉得他只是代表着一种过时的想法而已。

私下的过结也让事态趋于严重。理查德说，Symbolics 公司雇走大量黑客之前，理查德就觉得很多黑客都会有意躲避自己。这些黑客之后大部分都加入了 Symbolics 公司。理查德回忆：“他们很少问我去不去唐人街吃饭了。当年从格林布拉特（Greenblatt）开始，实验室就有个传统：谁要是去吃饭，谁就发个消息或者亲自走一圈，问问实验室里还有谁想一起去。可大概在 1980 年到 1981 年间，逐渐就没什么人问我了。他们不邀请我倒也罢了。之后，他们之中有人告诉我，当年他被人告之，不许他跟我说大家一起出去吃饭故意不带我这个事。”

虽然理查德对这种排斥行为非常反感，但并没对此有什么行动。可 Symbolics 的最后通牒则改变了一切，这事从此也不再是私人恩怨了。当 Symbolics 公司不再给用户提供源代码的时候，理查德决心要对此进行反击。他日夜坐在他的办公室里，在麻省理工学院的 Lisp 机系统上，实现了 Symbolics 提供的各种新功能和 bug 修正。他把修改后的版本代码分发给各个用户，包括 LMI 公司的客户。这样，LMI 公司的客户也可以拥有 Symbolics 系统类似的功能。

这也为理查德在黑客圈子里凭添了几分名气。理查德早就因为 Emacs 而名声大噪。如今，他一个人单枪匹马，对抗整个 Symbolics 公

司的开发团队，而且这团队之中还尽是各色传奇黑客。理查德的这一行为本身，就足以成为信息时代的一段传奇。史蒂芬·李维在《黑客》一书中称这一行为是“黑客杰作”；并把理查德比作现代的约翰·亨利（John Henry）¹⁶。史蒂芬·李维在书中说，很多 Symbolics 公司工作的黑客们都钦佩理查德的能力。他引述了比尔·高斯伯（Bill Gosper），他曾在 Symbolics 的帕罗奥多（Palo Alto）分部工作，对于理查德那段时期的产出，比尔·高斯伯甚是惊奇：

我读过理查德那段时期的一些代码，有些代码写的并不怎么出色（至少在我看来），但我还是要说：“且慢，他就只有一个人，他没法和别人整晚探讨，他是单枪匹马啊！一个人能完成如此大量的工作，简直是逆天了¹⁷！”

对于理查德来说，这段时期和 Symbolics 的竞争，既让他既感骄傲，又让他感到了深深的难过。理查德是个彻头彻尾的自由主义者，自己的父亲也曾在二战期间为自由而战。他不会期盼勉强得来的和平。在加入人工智能实验室之前，他本就心向自由。如今与 Symbolics 之间燃起的硝烟，则又把他推向了一个极端。无巧不成书，这次的矛盾，恰恰发生在人工智能实验室黑客文化消退的时候。理查德曾被这文化滋润，如今眼见它消失殆尽，其中心酸，难以言表。曾有一日，在他编程的间歇，偶尔路过实验室的设备间。眼见那台曾在实验室服役的 PDP-10 摆放在房间之中，无人问津。当年那几个忙碌闪烁的状态灯，如今也黯淡无光。往事片段，涌上心头。眼看着这台几十年前的计算机，仿佛看着家中亲人，静静的躺放在那里，魂归西天。

“我当时泪如泉涌。”理查德说，“它就在那儿，可却没人关心，无人维护。眼前的一切在告诉我，我们当年的那个黑客大家庭，早就不复存在了。”

然而，留给理查德感伤怀古的时间却并不多了。无论曾投入过多少人力物力，整个 Lisp 机产业却仅仅是昙花一现。计算机小型化的脚

步一步不停。带来了更新，更强大的微处理器。这一波趋势，如同风卷残云般，将其他竞争者一举赶出主流市场。

伴随这波风潮而来的，是成千上万的专有软件。每个专有软件，都带着自己的使用许可证和保密协议扑向用户。令其他黑客无法触碰其中的代码。很多软件使用许可证对用户粗暴无礼。但是在 1983 年，这些专有软件依旧成了主流，填补了市场。也让潜在的竞争对手望而却步。软件，曾经一度只是各个硬件厂商的随机赠品，如今却成了业界新宠。当下，用户们开始极度索要新软件，新功能；而至于是否可以知道软件内部究竟做了什么，则甚少提及。

个人计算机的到来把这股潮流推向了巅峰。苹果，Commodore 等公司跨入了百万富翁的行列。它们出售个人计算机，并在上面安装了自己公司的操作系统。这些计算机的用户们可不再像当年的黑客，他们并不关心软件的源代码，要是买来的软件不附带源代码，他们也不会大呼小叫。一些还恪守当年黑客信条的人，曾试图把黑客的这种传统带入个人计算机这个新兴市场。但无论如何，这个唯利是图的新兴市场推动着程序员写出更多的软件，也同时带来了更多的使用许可证。

这其中，最为著名的程序员恐怕要算比尔·盖茨了。这位哈佛的辍学生比理查德小两年入学哈佛，算是理查德的学弟。不过理查德当年并不认识他。1983 年 9 月 27 日，理查德在 thenet.unix-wizards 新闻组上发布了 GNU 工程计划。而七年之前，比尔·盖茨在软件开发者的社区中发表了那封著名的公开信。当年，微软还是坐落在新墨西哥州，阿尔伯克基市的一家小公司。那时，曾有很多 PC 用户私下拷贝微软的软件。由此，才让比尔·盖茨发表了那封《致爱好者的公开信》。

“有谁会在没有任何报酬的情况下来做这些专业工作？”比尔·盖茨在信中发问：“什么样的爱好者可以为他的产品投入三人年¹⁸的精力，开发完软件还得去发现错误，编写文档，最后还要免费发布他的产品¹⁹？”

比尔·盖茨 1976 年 2 月发表了这封公开信，那会还没有几个人工智能实验室的黑客读过这信。可显然，这封信代表了商业软件公司

和软件开发者对于软件态度的转变。既然市场都这么说了，那还为什么把软件当作免费赠品呢？从七十年代走到八十年代，出售软件已经不再仅仅为了弥补开发的成本，它变成了一种政治宣言。当年，里根政府正在极力清除大萧条时期政府制定的各种避免竞争的限制，越来越多的程序员也把黑客文化视为一种反对竞争，乃至违背美国精神的东西。最起码，他们觉得所谓黑客精神，最多是股复古风，不过是六七十年代反对大型企业态度的延续，一种企盼回到理想年代的情结。

理查德曾生活在六十年代，却被当成是活在五十年代的人。所以所谓的复古，不赶潮流对他来说倒也没什么。可作为一个一度使用顶级计算机，顶级软件的程序员，面临着那些带有各种使用许可证，禁止用户随意拷贝或修改的专有软件，理查德却面临了一场艰难的道义选择：摆在眼前的有两条道，要么默许专有软件，然后忘掉自己当初对它的各种反对；要么倾其一生，创造一套独立于各种专有软件的系统。在和 Symbolics 公司对抗了两年之后，理查德自信有能力去走第二条路。他说：“我倒是可以从此不再使用计算机。可这是我看家本事了。别的工作怕也做不来。没准可以做个餐厅服务员，不过也去不了什么大餐馆。”

要让理查德放弃他所热爱的计算机和编程，放弃从剑桥以来这个最大的乐趣来源，转头去做个餐厅服务员或是别的工作，理查德可绝对不能答应。他没有退却，决定主动出击。

作为一个无神论者，理查德不愿把这一系列事件归咎于命运，因果或是缘分。他决定避免使用专有软件，并且创造完全自由的一套操作系统及其外围软件，来帮助其他用户获得自由。作出这个选择，对于他来说是再自然不过了。毕竟，凭借理查德内心的那份反抗精神，和他的智慧能力，他选择了一条少有人走的路。这条路，甚至还没被很多人发现。在他的一篇名为《GNU 工程》的文章中，理查德曾引用了犹太先贤希肋耳（Hillel）的话来表明他的决心：

我不为我，谁人为我？我只为我，我为何物？此时不为，更待何时²⁰？

在外演讲时，理查德会避免借用任何宗教语言，而采用一些世俗的说法来描述。他说：“我问自己，我，作为一个操作系统开发者，究竟可以做什么，来改变现状？这个问题稍加思考，很快就可以看出来，能解决这个问题的恰恰需要是一个操作系统开发者。”

理查德说，一旦想通了这点，其他的就“顺理成章了”。在 1983 年，麻省理工学院从 Symbolics 采购了他们公司的第二代的 Lisp 机。在这批 Lisp 机上，根本无法运行麻省理工学院自己的 Lisp 机操作系统。旧机器被这些新采购的机器替代，几乎没有什么人再使用麻省理工学院的 Lisp 机系统，也就没人来报告 bug。理查德也就无法继续开发学院的 Lisp 机系统。他必须停下这个工作，不过这也恰恰是他想要做的。因为麻省理工学院的 Lisp 机系统并不是理查德心目中的自由软件：用户虽然可以获得系统的源代码，但是却不允许用户独立发行它。另一方面，麻省理工学院的系统也依旧继续开发着：LMI 没有被 Symbolics 打倒，他们依旧在开发自己的软件。

理查德可不希望花费自己的一生，来惩罚那些摧毁黑客社区的人。他要自己创建一个新的社区。他决定远离那些与他理想背道而驰的软件。他打算要投入自己的全部精力，开发出心中理想的软件，帮助自己 and 他人远离那些不讲道义的程序。他发誓要创造一个自由的操作系统，“或者为此而奋斗致死”。为此，他在 1984 年 1 月辞去了麻省理工学院的工作，专职去开发 GNU 系统。

辞去工作可以让他在法律上与麻省理工学院断绝联系。不过，依然有很多人工智能实验室的朋友支持理查德的工作，让他得以继续使用实验室的设备。之后，甚至给他准备了一个独立的办公室。凭借理查德的能力，他在开发 GNU 系统之余，还兼职一些咨询师的职位，借此收入来支持 GNU 系统的继续开发。在从麻省理工学院辞职的过程中，他拒绝了任何机构拥有 GNU 系统。这位曾经一度畏惧社交活动的人，如今则把这种心理发挥到极致，让自己的社交障碍变成了一堵防火墙，隔离了各种可能的法律纠纷。

在项目开发的最初几个月，理查德甚至也把自己隔离在 UNIX 社区之外。尽管他在 net.unix-wizards 新闻组发布的消息获得了很多同道的支持，但在最初阶段，只有很少人加入了这个工程。

当年一个 UNIX 用户组的一位领导李奇·莫林(Rich Morin)说：“整个社区（对于理查德的项目）的反响比较一致。大家都会说，‘嗯，不错，好啊。是个挺棒的想法。期待你的代码，希望能看到它成功。’”

可说到底，这可确实是个超大的工程。理查德决定尽量采用已有的自由软件。于是他开始调研已有的自由或免费的程序，试着修改它们并把它们纳入 GNU 工程之中。第一个相中的，是一款名为 VUCK 的编译器。它可以把 C 语言编写的程序转换成可执行的机器码。VUCK 是荷兰人开发的，荷兰语中，是“自由大学编译器”（Free University Compiler Kit）的缩写。带着几分期待，理查德联系 VUCK 的作者，询问它们这个软件是不是自由的。作者告诉理查德，所谓“自由大学”，指的是阿姆斯特丹自由大学（Vrije Universiteit Amsterdam）。虽然大学的名字里有“自由”二字，但并不意味着软件也是自由的。这番反复令理查德非常沮丧。

理查德回忆：“他的回复带着几分嘲讽。他说，这所大学是自由的，可编译器不是。” VUCK 的作者非但没能给理查德提供什么帮助，反而建议理查德放弃开发 GNU 工程，并希望他转而为 VUCK 开发些插件，没准可以提高 VUCK 的销量，而且答应会按利润给理查德分成。理查德说：“由此，我决定，为 GNU 工程开发的第一个软件，将是一个多语言，跨平台的编译器²¹。”

放弃了 VUCK，理查德转向另一款名为 Pastel 的 Pascal 编译器。它由劳伦斯·利弗莫尔国家实验室(Lawrence Livermore National Lab)开发。Pastel 的作者告诉理查德，这个编 Lawrence Livermore National Lab 译器可以自由传播和修改。遗憾的是，这款编译器占用内存太多，以至于无法在很多平台上运行。它会对整个文件做语法分析，然后把所有的状态都存入内存，等到编译结束，才释放所占用的空间。在当年，这种做法只能在大型机上行得通。而运行 UNIX 的机器，普遍没

有那么大的内存。哪怕是运行 32 位 UNIX 系统的机器，也很少能提供如此大的内存给一个程序。一开始，理查德还取得了一些小进展。他给这个编译器做了一个 C 语言的前端，然后跑在内存较大些的 Vax 机器上。可是，当他把这个编译器移植到 68010 上的时候，编译器则总是崩溃。仔细查清原因，理查德发现是内存太小造成的。于是，他决定只能从头开发一个编译器。最终，他实现了这个编译器，并把它命名为 GNU C 编译器，缩写 GCC。不过，在 1984 年，大家也还不知道这个编译器后续开发将会如何。于是，理查德一方面慢慢等着编译器相关的计划逐渐成型；另一方面着手开发 GNU 的其他组建。

在 1984 年 9 月，理查德开始开发 GNU 版本的 Emacs 编辑器。这个编辑器试图克隆并改进他主导了十几年的 Emacs 项目。在当年，UNIX 社区里，有两大流行的编辑器：一款是 Sun 公司的创始人比尔·乔伊开发的 vi 编辑器；另一款是贝尔实验室的科学家，UNIX 创始人肯·汤普森开发的 ed 编辑器。这两款编辑器都很不错，也很流行。但是它们都没有提供类似 Emacs 的扩展功能。

如今回忆起来，理查德说，他决定为 UNIX 开发 Emacs，并没从什么战略角度考虑。“我就是想在 GNU 系统里用 Emacs。而且由我来开发它再合适不过了。”

又一次，他找到了一些现成代码，希望能借此节省些时间。在开发 UNIX 版的 Emacs 的过程中，理查德很快找到了卡耐基·梅隆大学毕业的一位博士生詹姆斯·高斯林 (James Gosling)²²。他用 C 语言开发了一款名为 Gosling Emacs 的编辑器，简称 Gosmacs。它克隆了部分 Emacs 的功能，内建了一个简单的 Lisp 语言解释器，实现了简单的 Lisp 语法，并把这种 Lisp 方言成为 Mocklisp。尽管高斯林已经把 Gosmacs 卖给了一家名为 UniPress 的私人软件公司，但是高斯林在读博士期间，还曾通过邮件，授权给了一位参与过 Gosmacs 早期开发的人，让这位开发者可以独立发布 Gosmacs。本着互助的精神，这位开发者将代码给了理查德，并担保可以自由使用。

一开始，理查德以为他只要修改一些用户命令，就可以实现当年 PDP-10 上的 Emacs 的功能。然而，他发现 Gosmacs 使用的 Mocklisp 解释器实在太山寨，很难称得上是个真正的 Lisp 解释器。于是理查德不得不先去实现一个真正的 Lisp 系统。理查德要大刀阔斧地修改 Gosmacs 的代码了。他要让这个编辑器用上 Lisp 的灵活又强大的数据结构。1985 年中期，GNU Emacs 正式在因特网上发布。发布的代码中，只有很少的几个文件里，还保留着 Gosmacs 原来的代码。

接下来，UniPress 得到了理查德利用 Gosmacs 开发 GNU Emacs 的消息。他们拒绝承认高斯林曾把发行权授予了别人。理查德找不到当初的通信邮件，来证明自己的清白。于是，他把剩下的那些从 Gosmacs 拿来的代码清理干净，解决了这场纠纷。

不过，这次的纠纷又让理查德对专有软件有了更多的思考。在 1986 年，理查德在瑞典皇家技术研究所的演讲中曾引用了这次与 UniPress 的纠纷，并把它作为典型案例，来说明专有软件的危害。

理查德曾说：“有时候，我真觉得我这辈子应该找来一大堆专有软件，给它们一个个破解刻称光盘，然后拿到街角去免费赠送给路人。没准那样倒是可以更快地给用户带来些自由。要比自己亲自操刀写软件方便得多。不过要是哪样，可能很多人都不敢要我送出的光盘²³。”

抛开各种压力不说，抛弃高斯林写的代码倒未必是件坏事。长远来看，对于理查德和整个自由软件运动，这个做法不失为明智之举。它让理查德必须去直面 Emacs 社区里的各种弱点。让他重新思考这种非正式的信任关系带来的额外麻烦。它也催促着理查德更加明确自由软件运动的政治目标。在 1985 年，理查德照常又发布了新一版的 Emacs。同时，他也发布了《GNU 宣言》。这个宣言是他 1983 年 9 月那篇文章的扩展。在宣言中，理查德用了很长的一节，引述了各种在商业或学术领域里，为专有软件辩护的观点。其中一个经典观点就是：“难道程序员们的这些创造就不值得什么物质奖励吗？”理查德对此的回应是：

“如果要让大家对什么行为做物质上的奖励，那么这个行为必须是对社会有一定贡献的。创造本身的确是一种有益于社会的活动。但前

提是这个社会可以自由地使用那些创造出来的东西。程序员创造了某些有用的软件，如果这值得奖励；那么按照这个逻辑，要是他们借着作者的身份限制软件的使用，那他们就得接受惩罚了²⁴。”

随着 GNU Emacs 的发布，GNU 工程也慢慢地积累起了给大家读的代码。而 GNU 工程也逐渐要面对越来越多的琐事。越来越多的 UNIX 开发人员开始使用 GNU 的软件，各种捐款，礼品，以及索取磁带的请求也接踵而至。为了解决这些非技术类问题，理查德拉来了一些挚友，组建了自由软件基金会（Free Software Foundation，简称 FSF）。这是一个非盈利性组织，旨在促进 GNU 工程完成目标。理查德担任主席，理查德的很多朋友和黑客同道都成为了董事会成员。自由软件基金会为 GNU 提供了一个对外的窗口。

罗伯特·查瑟尔（Robert Chassell）当年曾是 LMI 公司的一名程序员，之后成为了自由软件基金会的五个注册董事会成员之一。他曾在自由软件基金会担任会计。这个职务一开始并不起眼，后来却越来越重要了。

查瑟尔回忆：“我们在 1985 年那会，花费和收入大概在两万三千美元这个级别。理查德有一间自己的办公室，我们还租了点地方。我把所有东西，尤其是很多磁带，放在我桌子底下。后来，LMI 借给我们一点地方，我们才把磁带和类似的东西放在了那里。”

除了提供一个门面意外，自由软件基金会还吸引了很多抱有类似理想的程序员。UNIX 市场融入了越来越多的竞争者，理查德的 GNU 工程也逐渐占有了一席之地。UNIX 厂商们为了能栓住用户，它们开始拒绝为用户提供 UNIX 的源代码。不过这个行为最终倒让 GNU 获得了更多的用户。UNIX 界的人士曾一度把理查德视为不切实际的疯子。如今则有不少人把他看作软件界的预言家。因为有越来越多的人，遇到了理查德描述的那些专有软件的问题。

查瑟尔曾描述过这种感觉：“很多人一直对此没有意识。等到事情真的发生在自己身上，才有所察觉。他们花了几年的时间，去开发完

善一个软件，可到头来却被别人占位已有，不许他们再碰。等这种事情发生几次，你就会对自己说，‘等会，这真的不对头。’”

查瑟尔加入自由软件基金会的一个重要原因，源自曾经的一次个人经历。在加入 LMI 之前，查瑟尔曾为 Cadmus 公司写了一本 UNIX 的入门书。Cadmus 是美国剑桥市的一家软件公司，之后关门倒闭了。而查瑟尔写的这本书，也随之灰飞烟灭。查瑟尔曾几次试图买回这本书的版权，但都无果而终。

“我知道，那本书如今就在某个书架上，别人用不了，读不上，也没法复印。它从此就与世长辞，不见天日。”查瑟尔说，“容我自夸一下，它真的是本不错的入门书。要是能再给我三四个月的时间，可没准可以把它改成一本很好的 GNU/Linux 的入门书。整个经历，只能留在我的记忆中，别的都没了。”

眼睁睁地看着自己的作品就此消失，曾经的雇主破产关门，查瑟尔说，他好像感到了当年理查德心中的那份怒火。“对我来说，有件事情越来越清楚：你要想过个像样的日子，你就不想有一丝遗憾。”查瑟尔说：“有深入了解的自由，有修改的自由，无论这是什么，这想法本身确实改变了这个世界。他让人觉得，在自己短短的生命之中，所做之事是值得付出的。因为它还没被谁夺走，也许还没被人弃置遗忘。至少，你还和它有所联系。不会让你觉得自己丢失了一段美妙的人生。”

Endnotes

¹ 参见理查德·斯托曼，《GNU 计划初始宣言》，1983 年 9 月

² 同上。

³ 同上。

⁴ 参见马歇尔·柯克·麦库西克 (Marshall Kirk McKusick) 的文章《伯克利 UNIX 二十年》，该文收录于《开源软件文集》中。该书中文版已由中国电力出版社出版。

⁵ 参见理查德在瑞典皇家技术研究所的演讲(1986 年 10 月 30 日):<http://www.gnu.org/philosophy/stallmkth.html>

⁶ 同上。

⁷ 参见《黑话词典 3.0.0》<http://catb.org/jargon/html/T/TWENEX.html>

⁸ 参见http://www.as.cmu.edu/~geek/humor/See_Figure_1.txt。

⁹ 参见理查德在瑞典皇家技术研究所的演讲(1986年10月30日):<http://www.gnu.org/philosophy/kth.html>

¹⁰ 参见《麻省理工学院人工智能实验室游客守则》<http://www.art.net/~hopkins/Don/text/tourist-policy.html>

¹¹ 参见理查德在瑞典皇家技术研究所的演讲(1986年10月30日):<http://www.gnu.org/philosophy/kth.html>

¹² 同上。

¹³ 参见史蒂芬·李维 (Steven Levy), 《黑客》(1984年, 美国企鹅出版社), 第423页。

¹⁴ N. P. 纽奎斯特所 (N. P. Newquist) 所著的《大脑创作家》(The Brain Maker) 一书中说, 人工智能实验室实验室警告理查德, 让他从此远离 Lisp 机项目。此说法后被证伪。

¹⁵ 史蒂芬·李维 (Steven Levy) 在《黑客》一书中, 称理查德为“最后一位真正黑客”。不过李维的这个称呼经常被人误解。李维在书中使用“真正黑客”一词来专指麻省理工学院的黑客们。以此来和其他黑客社区区分。当麻省理工学院的黑客文化逐渐消退, 只剩下理查德一个人的时候, 他显然就被称为“最后一位真正黑客”了。李维并非想说, 除了理查德以外, 别人都算不上黑客。但是后人常常以此解读, 特别那些没有读过《黑客》一书的人。理查德自己也从不用这一称呼来形容自己。

¹⁶ 译注: 约翰·亨利 (John Henry), 非裔美国人, 钢钻工人。传说中他身高力大, 曾在19世纪末期参与修筑贯穿美国东西海岸的铁路。期间为保住自己和工友的工作职位, 一人单挑一台蒸汽机钻。最终获胜, 但因体力透支而离世。约翰·亨利从此成为美国工人勤奋, 坚毅的象征。他的画像甚至还出现在二战期间, 美国的宣传画册上。

¹⁷ 参见史蒂芬·李维 (Steven Levy), 《黑客》(1984年, 美国企鹅出版社), 第426页。

¹⁸ 译注: 人年 (man-year) 软件开发用的单位, 用户衡量软件开发的投入精力。假定每个开发人员的能力相同, 那么3人年就指三人一年, 或一人三年的工作成果。

¹⁹ 参见比尔·盖茨1976年2月3日发表的《致爱好者的公开信》。http://en.wikipedia.org/wiki/Open_Letter_to_Hobbyists

²⁰ 参见理查德的文章《GNU 工程》<http://www.gnu.org/gnu/the-gnu-project.html> 理查德在引用这句话的时候特意说明: “作为一个无神论者, 我并不服从任何宗教领袖。但我依旧也会赞同他们的某些言论。”译注: 理查德引用的此句来源于犹太教的经典《犹太圣传》的民刑卷, 先贤篇第一章第14节。上下文为: “名声, 欲扬反失; 学问, 不进则退; 不读经, 毋宁死; 而盗用冠冕者, 终将灭亡……我不为我, 谁人为我? 我只为我, 我为何物? 此时不为, 更待何时?”“此句的意思是强调命运掌握在自己的手中, 要靠自己的奋斗, 而不能指望别人。

²¹参见理查德的文章《GNU 操作系统和自由软件运动》，刊于《开源软件文集》一书。中国电力出版社曾出版过该书中文版，洪峰等人翻译。

²²译注：詹姆斯·高斯林就是后来的 Java 之父。

²³参见理查德在瑞典皇家技术研究所的演讲(1986 年 10 月 30 日):<http://www.gnu.org/philosophy/stallmktth.html>

²⁴参见理查德所著的《GNU 宣言》<http://www.gnu.org/gnu/manifesto.html>

Chapter 8

圣 · Ignucius

茂宜高性能计算中心 (Maui High Performance Computing Center) 坐落在夏威夷群岛中的茂宜岛上。这是一座单层建筑，建在满是红色火山灰的山岗上，俯瞰茂宜岛的商业中心纪平镇。这个计算中心身在夏威夷，坐拥迷人的山景海风，更有身价千万的房产。这地方好像和高新技术占不上一点边。它跟麻省理工学院技术广场的见楞见角，死气沉沉的气氛完全不同。哪怕跟阿尔贡国家实验室，洛斯阿拉莫斯或者新墨西哥的那些实验室比，茂宜高性能计算中心依旧显得特别。总会难免让人觉得，茂宜计算中心的科学家们每天可能并不怎么关心自己的研究课题，没准他们天天都会把时间花在晒太阳，吹海风上。

不过这个第一感觉可不怎么准确。确实，茂宜计算中心的研究人员的确占尽了地利，也自然会借此休养。不过他们对待工作还是非常认真的。Top500.org 是一个记录全世界超级计算机排名的网站。根据这个网站的资料，茂宜高性能计算中心的 IBM SP Power3 超级计算机可以每秒完成八千三百七十亿次浮点运算，世界排名前 25。这台超级计算机归夏威夷大学和美国空军共同拥有和维护。它主要负责完成军队补给调度计算和高温物理研究。

简而言之，茂宜计算中心是个非常独特的地方。这地方既有学术研究的文化氛围，又融入了夏威夷上那份慵懒宜人的情调。二者一阴一阳，相得益彰。2000 年，他们在自己的官方网站放上了这样一条标语：在天堂之上计算。

这样一个地方，一般你是见不到理查德·斯托曼身影的。他这次来到这里，站在一位员工办公室的窗口，盯着外面如画的风光，口中却蹦出了一句简单的批评：“阳光太强。”自然，他从计算机界的圣地麻省理工学院来到这里，是有话要讲的。哪怕再强的阳光，刺入这位黑客的双眼，他倒也在所不辞。

我赶到现场的时候，会议室里早已人满为患，大家都在等待着理查德的演讲。听众中，男女比例倒是比纽约那次的演讲平衡些，不过也没平衡到哪儿去：大约 85% 是男性，15% 的女性。有一半的听众，都身穿土黄色的裤子，配上印着商标的高尔夫衬衫。另外一半，则显得更加融入了当地的风俗。身穿宽大的大花 T 恤，皮肤则已被晒成了褐色。这打扮在夏威夷可是正常装束。唯一能显示他们技术本色的东西，也就只剩下了些小玩意儿了：诺基亚手机，掌上电脑，还有索尼 VAIO 笔记本。

理查德身穿纯蓝色 T 恤，棕色涤纶休闲裤，白色袜子。不用说，这身行头在当下确实太显眼了。房间中淡淡的花香，倒是让人没太注意理查德缺少阳光滋润的皮肤¹。不过他一头长发和络腮胡须却依旧引人注目。外人一看，就知道他不是本地人。他就差在脑门上贴个标签，写着：“美国本土人士。”[RMS：把自己打扮得和别人不一样有什么不对吗？]

演讲开始前，理查德自己在会议室前面闲逛。旁边几个人鼓捣着录像录音器材。这些人都身穿“茂宜 FreeBSD 用户组”（Maui FreeBSD User Group, MFUG）的 T 恤。所谓 FreeBSD，是一款从伯克利软件发行版的 UNIX 系统分支而来的操作系统。而伯克利版 UNIX 则是七十年代由加州大学伯克利分校开发的一个 UNIX 分支。说起来，FreeBSD 倒还是 GNU/Linux 的一个竞争对手。在黑客的世界里，理

查德的演讲总会被记录拍摄下来，留做档案。作为当地的著名自由软件组织，茂宜 FreeBSD 用户组可不能让同行失望。他们也要录音录像，让远在德国汉堡，印度孟买，俄罗斯新西伯利亚等世界各地的黑客们，都能听到 RMS 的箴言。

理查德的这些演讲记录，就像是感恩而死乐队 (Grateful Dead) 的现场演出一样，都是被迫捧者现场录下，私下传递。把理查德和感恩而死乐队做类比，可是有说头的。每当理查德描述自由软件的商业盈利模式的时候，他都会提起感恩而死乐队。这个乐队当年允许粉丝们在现场录音录像，这就令它不仅是一个简单的摇滚乐队。它俨然成了感恩而死音乐部落的中心。逐渐地，这个部落越来越大，令感恩而死乐队拒绝了各种唱片公司的合约，完全靠巡回演出，就可以撑起整个乐队的开支。在 1994 年，这支乐队的最后一次演出仅门票收入就高达五千两百万美元²。

当然，还没有哪个软件公司的成就可以媲美感恩而死乐队。不过，自由软件社区的这个部落越来越大，在九十年代后期，大家也逐渐开始认为分享发布软件的源代码没准是个好事。各大公司纷纷出马，企图建立起各自的忠实粉丝群。这其中就包括 IBM, Sun, 惠普及等。它们也许并没有打心眼里接受自由软件的看法，但他们的行为却是在响应理查德的号召。ZDNet 的软件专栏作家埃文·列伊博维奇 (Evan Leibovitch) 把 GPL (GNU 通用公共许可证，理查德创立的一种自由软件许可证，用于规定软件作者授予软件用户的各种权利) 描述为信息时代的大宪章。他认为，人们对于 GNU 的拥护持续高涨，并非单纯地跟风。他曾写道：“这个趋势让用户可以重新掌控自己的命运。就好像当年《大宪章》给了英国人民权利。GPL 让消费者作为计算机软件用户有了自己的权利和自由³。”

自由软件社区的这种部落文化也解释了为什么今天这四十来位古怪的程序员们，可以暂时不管各自的工作，也不去上网冲浪，而宁愿能挤在这个会议室里，听理查德的讲座。

和那次在纽约的讲座不一样，这次没有人为他做开场白，理查德自己也没做自我介绍。FreeBSD 用户组的人把设备一架好，理查德就自己踏上前来，开始演讲，声音一下盖过了台下的私语。

“一个社会该给软件的使用设立何种规则？每当想到这个问题，大多数人总是站在软件公司的立场思考。他们的想法往往像是在自我圆场。”理查德开门见山：“他们想：我们如何设立规则，才能让大家多给我们钱呢？我很幸运，在七十年代，曾加入了一个大家庭，在那里，大家都会互相分享软件。正因如此，我总会从另外一个角度去思考这个问题：我们应该如何设立规则，才能帮助我们构建一个更好的社会，一个可以让每个人都受益的社会？也正因如此，我得到了完全不一样的答案。”

理查德又一次提起了施乐打印机事件，形式和以前很像，同样是讲到最后，手指指向听众，口中说着：“也许就会背叛你！”之后，他又花了一两分钟，来解释 GNU/Linux 的名字问题。

“有些人会和我说：‘你为什么要花这么多精力去在乎人们叫它什么？为什么非要在乎这种名利上的事？如今任务已经做完了，你何必非要在乎人们有没有注意到 GNU 呢？’这个……如果这话描述的是事实，那我确实同意这话的观点。可现实是：我们的任务并没有完成。我们的任务不是要创造一套操作系统；我们的任务是要给计算机用户自由。要完成这个任务，我们必须尽一切努力，让用户可以自由地使用计算机和软件⁴。”

理查德最后说：“我们要走的路还很长。”

对于一些听众来说，理查德的这些话是老生常谈了。对于其他人来说，这些话则有点令人不可思议了。旁边那个身穿高尔夫 T 恤的听众已经开始打盹了，理查德停下来，让旁边的人把这位叫醒。

“曾经有个人，说我的声音特别像安慰剂，他问我是不是做过催眠师之类的工作。”理查德说着，下面一阵笑声。在笑声中他接着说：“我想这意味着我可以帮你们快速进入梦乡。可能你们之中有些人正想要这样。我觉得我确实也不该打搅你们，你要是真想睡觉，尽管睡吧。”

演讲最后，斯托曼简要地探讨了软件专利的问题。这个问题逐渐在软件业和自由软件社区中受到关注。专利的概念和想法本是为现实的物理世界准备。可把它用在信息世界中，就越来越显得古怪了。

版权和专利是两个不同的概念，也有着完全不同的目的。对于软件来说，版权法用来约束拷贝或修改软件代码的行为。一个程序的版权是归开发者所有。但是版权不能保护作者的想法。换句话说，开发者可以克隆一个已经存在的软件，只要他不去复制别人的代码，就不会涉及到版权问题。因为这是在重现别人的想法，所以这在版权法的作用范围之外。

如果开发者直接阅读软件的机器码，尽管会比较费事费力，但这依旧是合法的。同理，开发者也可以通过阅读软件的机器码，再自己独立实现一个类似的软件。这个做法通常被称作“逆向工程”。

可软件专利就不同了。按照美国专利局的说法，个人或公司的计算方面的创新想法，可以获得专利（准确说，所谓创新想法，也就是专利局没听说过的想法）。理论上讲，专利拥有者需要公开自己的创新想法，由此可以垄断这个创造至少二十年，计时从专利提交起开始。但是实际上，公开专利的创新想法并不能怎么让公众受益。因为程序的想法往往是不言自明的。或者可以通过逆向工程获得。而专利法和版权法不同，它禁止任何人实现专利上描述的想法，哪怕开发者是独立开发的。

在软件世界里，二十年的时间可能覆盖了一个市场的整个生命周期。这样，专利就成了一种策略。当年，微软和苹果这些大公司为版权打得不可开交。而如今那些互联网企业，则使用专利来为自己开路，赶走竞争对手和新人。这里有个著名的案例。2000年，亚马逊购物曾试图为它的“一键购买”功能注册专利。简单说，这个功能就是允许用户事先存储好个人信息——比如信用卡信息，收件人地址等——然后在购买商品的时候，只需要按一个按钮，就可以下单，不必再跳转到其他页面填写地址，付款等信息。对于大多数公司来说，软件专利更像是一种防卫措施。几个公司可能会合作共有一系列专利，用来阻

止另外一个联盟的专利。尽管如此，还是有不少领域被软件专利占满，阻碍了各种可能的竞争对手。比如图像处理领域，加密领域等等。比较实际的例子是，一些字体渲染算法，不会出现在自由软件中，因为苹果公司持有这些算法的专利。

对于理查德来说，软件专利更督促着黑客要时刻警觉，而且还凭添了几分戏剧色彩。也恰恰是软件专利，让自由软件一直强调的政治策略再次得以重申。即，用户的自由比其他更重要。理查德说，高性能，低价格等特性的确是 GNU/Linux, FreeBSD 等这些自由软件的优势。但是除了性能，价格这些东西以外，还有更重大的问题，那就是用户和开发者的自由。

这个观点在圈子里也饱受争议：开源的支持者们喜欢强调自由软件的实用性，会避免探讨用户自由之类的问题。他们可能不会强调这些，转而强调黑客开发的工程模型。强调同行互相审校的重要性。他们认为，GNU/Linux 或 FreeBSD 的开发模式可以创造出更好，更强大，更值得信赖的软件。

但是，“开源”一词也并非意味着没有任何政治诉求。对于开源支持者来说，开源有着两重含义。第一，它避免了在英文中使用 Free 一词。自由软件的英文是 Free Software，因为 Free 在英文中既有自由的意思，也有免费的意项，所以难免会产生歧义。很多商业公司会觉得自由软件等于免费软件。使用“开源”一词的动机之一，就是希望消除这种歧义。第二，它让商业公司可以从技术角度去审视自由软件，而不必在道义过多解读。艾瑞克·雷蒙德（Eric Raymond）是开放源代码促进会（Open Source Initiative，缩写：OSI）的联合创始人之一，也是支持使用“开源”一词的著名黑客。他曾在 1999 年发表了一篇文章，用以表明他拒绝接受理查德的政治路线。文章标题是《少说话，秀代码》（Shut Up and Show Them the Code）：

RMS[即，理查德·M·斯托曼的缩写]的用词对于我们这样的人来说很有吸引力。我们黑客都是思考者，都是理想主义者。我们都时刻准备着响应支持“自由”，“原则”和“权利”的号召。哪怕我们有时

候会与理查德的一些意见不同，但我们依旧希望他的做法能有效。我们觉得他的言辞应该有效。可是，我们看到世界上另外 95% 的人和我们不一样，理查德的言论对他们无效。面对这些，我们开始迷惑，甚至怀疑。

雷蒙德写道，那另外 95% 的人中，包括很多公司经理，投资人，还有普通的计算机用户，他们不是黑客，他们会倾向于跟从专有商业软件的大市场。雷蒙德认为，如果不能赢得这些人的支持，黑客们将只能游离于主流之外，去追随自己的理想。

理查德向我们谈论起“计算机用户权利”的时候，他向我们发出了一份吸引人却危险的邀请函，引导着我们重犯旧错。我们应该拒绝它——不是因为它的原则是错的，而是因为那样的词汇用在软件上，只能说服我们自己，而不能说服别人。事实是，它把大多数圈外人都拒之千里⁵。

但是，理查德却反对雷蒙德的說法：

雷蒙德对我们失败的解釋非常有误导性。因为我们压根都没有失败呢。我们为自己设定的目标很远，我们还有多长的路要走。可是要知道，我们已经走过了很多路，克服了很多困难。

雷蒙德关于大众的悲观判断是夸大其辞。很多人不是黑客，但是却同样关注我们强调的政治问题，他们也认为自由的问题比开源所强调的技术优势要重要。这些人之中，甚至会有一些政治领导人。

厄瓜多尔和巴西的政府恰恰就是因为考虑到道义问题，而不是技术问题，才让政府计算机使用自由软件。他们可不是黑客，但他们也理解其中的自由。

对于理查德来说，开源的重大问题，在于得出了过于宽松的结论。它可以说服用户去使用自由软件，但却无法提供更多的理由，让用户

完全放弃专有软件。这只能给用户部分自由，却不能教会用户去珍视自由。最终，用户还是很可能继续掉入专有软件的陷阱，本来的那些自由也从此丢失。举个例子，如果因为专利问题，让某个自由软件无法添加某项功能，用户该做何选择？

大多数开源支持者们也会和理查德一样，极力反对软件专利。甚至很多专有软件的开发者也一样反对软件专利，因为软件专利也同样威胁着他们的项目。然而，理查德指出，软件专利会阻止人们实现某些功能。这里，开源和自由软件在理念上的分歧就显现出来了：

“这个时候，我们如果做不出更好的软件，并不是能力不足。而是因为有人利用专利，故意阻止我们为公众服务。倘若一位用户使用的自由软件中，由于某项专利，而缺少一个功能，用户会作何反应呢？如果当初这些用户是被开源软件的理念说服而使用这个自由软件，那么他们当初学到的是：给予用户自由是为了创造更强大可靠的软件。这时候，遇到软件缺少功能，他们会说，‘你们没能给我一个强大可靠的软件，它少了这个功能，你们骗了我。’相反，如果这个用户是因为认同自由软件的价值观，而使用自由软件的话，那么他们会说：‘那些专利竟然阻止了这个功能的实现，剥夺了我的自由。’只有用户认同自由软件的价值观，那么面对大量的软件专利，我们才可能幸存。”

看着理查德阐述自己的政治诉求，并不会让人迷惑或厌恶。他衣着朴素，说起话来有理有据，逻辑清晰。听众中一位提问：如果完全避免使用专有软件，可能无法用到最新的技术，该怎么办？理查德引述自己的信念，回答了这个问题：“我觉得自由本身比任何新近技术都重要。如果面前有一个先进的专有软件，和一个技术落后的自由软件，那么我宁愿选择后者。因为无论如何，我都不会靠出卖自由，而换取更新的技术。我的原则是，如果我不能和你分享这个软件，我就不会使用它。”

有些人可能会把道义问题和宗教信仰联系起来。理查德的这个回答则更可能给人一种宗教信仰的感觉。不过，理查德的行为不同于犹太人只吃洁食的行为，也不同于摩门教徒不喝酒。理查德并不是要守

着清规戒律，他只是不想牺牲自己的自由。他的演讲解释了他的思路：专有软件会剥夺你的自由，如果你想要自由，你就要拒绝使用专有软件。

理查德仅仅使用自由软件，拒绝任何专有软件。他把这描述为个人信仰，并且希望把这份信仰与大家一同分享。但他不会强迫别人认同自己。大多数听众，在听过理查德的演讲之后，都会明白理查德所说的正确的软件之路。

演讲最后，像个总结一样，理查德开始了一个不太寻常的仪式。他从一个百货商店的塑料购物袋里，掏出了一条黑色长袍。他把长袍打开，套在了身上。接着，又从袋子里拿出来了一个圆形的老式计算机硬盘碟片，把它戴在头上。这碟片一戴上，就像极了基督教圣徒头顶的光环。看到这身扮相，人群中传来了一阵笑声。

“我是圣·IGNUcius⁶，来自 Emacs 教堂。”理查德举起右手，好似在为众人祈福：“我保佑你的计算机，孩子们。”

大家的笑声在几秒钟之内变成了一阵掌声。大家在鼓掌的时候，理查德头上的那个硬盘碟片反射来一道强光，一下看起来，好像理查德头上真的顶了个光环。他眨了眨眼睛，又让我觉得他好似宗教领袖。

理查德接着解释道：“Emacs 最开始是一个编辑器，之后，很多人把它当作一种生活方式，还有一部分人把它作为一种信仰。我们称他们称作 Emacs 信徒。”

很多人以为理查德对于自由软件一直都是不苟言笑，好像一个禁欲主义者一样。眼下这个玩笑倒是很好地否认了这种认识。在黑袍和光环之下，理查德好似在说：“你们尽情笑吧，我知道我这样很搞怪。”[理查德注：嘲笑别人行为怪异是不礼貌的，我并不鼓励这种做法。不过我是希望大家看到我圣·IGNUcius 的扮相，能够笑起来]

之后和理查德聊起圣·IGNUcius 这个舞台形象，他说他最早是在 1996 年开始在演讲中有这么个安排的。那会他早就开发了 Emacs 编辑器，那时也还没有“开源”这个词。理查德说，他当时希望能有个方式，能够“自娱自乐”一下；同时，也能告诉听众，理查德虽然固执，

但还不至于疯狂到不苟言笑。不过这之后，理查德说，一些人借着这个舞台形象，把理查德形容为一个软件界的空想家。艾瑞克·雷蒙德就在 1999 年的一次与 Linux.com 的采访中，说道：

我说过，RMS（理查德·M·斯托曼名字的缩写）很会算计。我并不是说他不诚实，也不是说他虚情假意。我的意思是，他和很多布道者一样，非常有舞台感。有时候，他的这种行为是有意识的——你看过他扮成圣·IGNUcius 的样子吗？他身穿黑袍，把一个碟片放在头顶。还有很多时候，那是无意识的行为。他很会掌握分寸，既能让你想去反驳，又不至于把人们吓跑⁷。

理查德显然不同意艾瑞克·雷蒙德的說法。他说：“这不过是我自娱自乐罢了。谁要是从中还看到了别的东西，那只能说他们想多了。那都是他们脑子里的想法，不是我的。”

理查德也承认，他的这个表演有做作的成分。“就是开个玩笑。我很享受被万众瞩目。”他如是辩解。为了让自己的演讲技能可以更加娴熟，他甚至还加入过 Toastermasters。这个社团旨在帮助人们提高演讲技能。理查德也曾向别人建议过 Toastermasters。他加入了这个社团，逐渐掌握了一些舞台技巧。在这次茂宜高性能计算中心的演讲之后几天，我和理查德聊天的时候，问起他是否有种所谓的“格鲁桥·马克思情结”——就是像著名喜剧演员格鲁桥·马克思（Groucho Marx）所说的那样，拒绝加入任何希望收留他的俱乐部⁸。理查德立即回复道：“不，但是我在很多方面都很敬仰格鲁桥·马克思，我说过很多话都曾受到过他的启发。不过我也受到过汉普·马克思（Harpo Marx）的启发。”

理查德一直以来对俏皮话，双关语的喜爱，恐怕的确是受了格鲁桥·马克思的影响。不过话说回来，双关语和文字游戏也从来都是黑客们的最爱。要说理查德真是有哪点会像格鲁桥，恐怕是他开玩笑之后，那副依旧淡定的表情。他的那些玩笑总是在不经意间蹦出来，他自

己则一脸严肃，眉毛都不抬。在你笑过之后，甚至反而会觉得理查德正在内心中笑话你。

不过你别担心，理查德可没笑话你。看到他在茂宜高性能计算中心演讲时候的圣·IGNUcius 的扮相，你的顾虑就打消了。他这表演虽然不比单口相声，但理查德总还是有能耐把一屋子的工程师聚在一起。“Emacs 教会之中，不必戒色禁欲。但也需你心存底线，道清德厚，有所顾忌。”理查德继续对听众说道：“你要在自己的电脑之上，趋邪扶正。专有软件，卸载删除。辟得净土，以敬用户。系统上下，软件自由。茫茫众生，守此信条，即入本会，得道称圣。对了，到时候你没准也能拿到这样一个光环。”

圣·IGNUcius 的演出以一个圈内冷笑话做结尾。在大多数 UNIX 或者类 UNIX 系统上，和 Emacs 齐名的另外一款编辑器名为 vi。vi 编辑器最早是由比尔·乔伊（Bill Joy）开发。他当年是加州大学伯克利分校的学生，如今是 Sun 微系统公司的首席科学家。理查德在摘掉自己头顶的“光环”之前，给 vi 这个竞争对手开了个小玩笑。“人们会问，在 Emacs 教会中使用 vi 是否算作原罪。倘若使用的是自由版的 vi，那不能称之为罪。可以算是自罚赎罪。那么，Happy Hacking!”⁹

一个简短的提问环节过后，听众们围在理查德周围，有些人向他索要签名。理查德拿着一位女士打印出来的 GNU 通用许可证，说：“我可以答应你在上面签名，不过你要答应我，在指代那整个操作系统的时候，使用 GNU/Linux 这个词。并且要告诉你的朋友也这么做。”

他的这个举动倒是证实了我的想法。和演员，政客不同，理查德的行为会始终如一，不会切换到“台下模式”。在圣·IGNUcius 这个角色中看到的那份理想主义，会一直体现在理查德的每个言行之上。在演讲之后的晚餐上，一位程序员向理查德提起了自己参与“开源”软件的经历，理查德还没顾得上咽下嘴里的食物，说道：“你是说自由软件吧。用自由软件这个词恐怕更恰当。”

在演讲结束后的提问环节，理查德承认自己很多时候有些咬文嚼字。“很多人会劝我：‘咱们先把人邀请进我们的圈子，然后再教会他

们什么是自由。’这也许是个好策略，但是我们当下的情况是，基本所有人都在邀请朋友们进到这个圈子里，可却没有几个人会告诉朋友自由是什么。”

最后的结果，用理查德的话说，就是好比很多第三世界国家建立的城市。“你把百万千万的人扔到城市中，结果是在城市里建起了成千上万的城中村，贫民窟。但是没人开始着手做下一步的工作：把城中村，贫民窟里的人解救出来。如果你觉得讨论软件用户的自由是有意义的，那么请加入我们，开始做第二步工作。很多人都在做第一步，但这第二步却无人问津。”

这“第二步”工作直指自由软件运动的核心：自由软件的重点并非装机量用户量，而是赋予用户自由。有些人希望能从内部瓦解整个专有软件工业，这实在是痴人梦话。理查德说：“从内部瓦解实在不现实，除非你能像当年戈尔巴乔夫那样攀到高位，否则你多半会被同化。”

又有很多人举手提问，理查德随机选择了一位身穿高尔夫衬衫的听众，这位听众问道：“如果没有专利，你会建议如何防止商业间谍？”

理查德答：“嗯……这其实是两个互不相关的问题。”

提问者补充道：“我是说，如果有人想要窃取另一个公司的软件。”

理查德立即反应道：“等等，你说‘窃取’？不好意思，你的这番陈述里包含太多偏见了。我只能说，我很难认同这些偏见。”接着，理查德转向问题本身：“各个公司，它们可能会开发专有软件，或者开发任何别的东西。无论开发什么，多少都会有些商业机密，它们无论如何都会保守这些机密。这个事实恐怕很难改变。在早年间——哪怕到了八十年代——绝大部分程序员都没想过把软件去申请专利。人们会发表文章记录自己有趣的想法，如果他们不认同自由软件的理念，他们可能会在文章中甚少提及各种细节。如今，他们同样写篇文章，笼统地记录下那些想法，依然保守着各种细节。如今不同的是，他们把这文章拿去专利局，就可以申请下专利。软件专利并没能把这个事实改变多少。”

“但如果不会影响他们公开发表的文章……”另一位听众起立发言，但还没说完，就被打断了。

“但它的确影响了。”理查德说：“在版权局能查到的公开文章就是告诉你，从现在开始往后 20 年，这个想法任何人都不能随使用。这能算是什么好处吗？还请注意，他们在文章中，会把想法描述得像天书一般难懂，而且会尽量把话说得模棱两可，这样就可以让这版权覆盖尽可能多的地方。没有几个人会从那些文章里去学习哪个版权所保护的技术。你能从那些文章中了解到的信息，只能告诉你什么技术你不能。”

听众一时安静下来。这个演讲从下午 3:15 开始，现在已经将近五点。下面很多听众已经有些坐不住了。理查德也察觉到了听众的倦意，他目光扫视了全场，说道：“那么，今天就到这儿吧。”他又停顿了一下，等到确认没有人有问题之后，理查德又重复了一遍那句口头语：

“Happy Hacking。”理查德说。

Endnotes

¹理查德注：所谓“缺少阳光滋润”，或者“越晒太阳越健康”的概念实在是非常误导人。在太阳底下不被晒伤的前提是你需要有足够的维生素 D。而过度在阳光下暴晒，不仅可能损害皮肤，甚至可能要人命。

²参见《1985 至 1995 北美演出收入排行榜》<http://www.dead101.com/1197.htm>

³参见埃文·列伊博维奇 (Evan Leibovitch) 在文章《谁会害怕大灰狼》，2000 年 12 月 15 日，ZDNet Tech Update. <http://www.zdnet.com/news/whos-afraid-of-the-big-bad-wolves/298394>

⁴.....

⁵同上。

⁶圣·IGNUcius (St. IGNUcius)，名字中包含了 GNU 三个字母。这个名字可能来自基督教圣人伊格那丢 (St. Ignatius)。伊格那丢被罗马皇帝投入野兽笼中而死。理查德的圣·IGNUcius 本身没有任何宗教意义 (理查德本人是无神论者)，仅仅出于玩笑心理，起了这么个名字：既能在字面上和基督教圣徒有些类似，又能把 GNU 三个字母放进名字里。

⁷.....

8
9

Chapter 9

GNU 通用公共许可证

1985 年新春伊始，理查德·斯托曼完成了 GNU 工程的第一个完整的软件：适用于 Unix 系统的 Emacs 编辑器。这个编辑器用 Lisp 语言实现。为了能让别的用户可以自由使用这个软件，他需要基于早期的“Emacs 公社”探索一种全新的发布方式。

在 Gosmacs 之前，自由修改软件与软件作者的特权之间就已经形成了一种对立的关系。在 1976 的版权运动中，修订后的美国版权法案把适用范围扩展到了软件作品上。根据该法案的第 102(b) 节的规定，个人或公司可以对软件程序的“实现方式”声明版权，但不能对“程序中的实际处理过程或方法”声明版权。¹

换句话说，如果以几何教科书为例：作者可以对书籍的文本声明版权，但不能对几何中的数学原理或用于解释这些原理的教学描述声明版权。所以，不管理查德对 Emacs 的使用做如何的规定，其它的程序员都可以合法的为 Emacs 开发一些新的功能或命令，而且，他们确实也这样做了。Gosmacs 就是为不同系统开发的 30 多种基于原始 Emacs 的衍生版本之一。

“Emacs 公社”只适用于理查德本人所写的原始 Emacs 版本。即使通过法律的手段来实施这种协议，它也不适用于类似于 Gosmacs 这种独立开发的衍生版本。依据自由软件运动的哲学，把 Gosmacs 变成非自由的软件是一件不道德的事情，因为作为一个私有软件，它将无法让它的用户享有自由的权力。TODO 但这个问题与 Gosmacs 内在的思想的最初来源没有任何的关系。

在版权的保护下，如果有人想从一个程序中直接复用代码（包括修改后复用），都需要得到程序原始作者的许可。新的版权法案适用于所有软件作品，即使这个软件没有明确的声明版权。早期的黑客们并不清楚这一点，但为软件作品添加版权声明的做法却从那个时候开始慢慢得以普及。

理查德得把这些版权声明的出现看成是一个不好的兆头。那时候，大部分程序都会从以前的程序中借鉴一些代码，然而，总统的大笔一挥，美国政府就剥夺了程序员和软件公司重用其它软件代码的权力。版权法案向原本不太正式的软件版权领域注入了一剂强心针。简单的说，这项法案使得原先黑客之间的较量变成了律师之间的较量。在这样的法律框架下，软件公司相对于黑客来说占据了天然的有利地位。有些人把在版权声明加入自己的名字看成是对代码质量的一种担保，不过大多数情况下，版权声明中都只会写公司的名字，对于个人来说，只能使用其它的方式来标记他们写过的代码。

然而，在领导 GNU 工程的这些年中，斯托曼注意到，版权法案使得软件的作者可以在其保护的范围内允许他人进行一些相关的活动，也可以对这些活动规定一些条件。他回忆说：“我见过一些附有版权声明的电子邮件，并且标明‘允许原样复制’。这给我了一些灵感。”这些许可证中通常带有一个条款，那就是在任何情况下需要保留这个许可证文本。斯托曼决定对此发扬光大。比如，可以允许用户任何分发软件，甚至分发修改后的版本，但是这些版本都必须具备相同的许可证，允许再次分发。

所以，斯托曼认为，用版权来保护软件作品并不是不道德的。只不过是人们使用版权的方式不太好：以版权保护的名义剥夺用户的自由。版权的出现似乎就是为了这个目的，大部分的软件作者也想不到别的使用版权的方式。但是，其实版权完全可以用其它的方式来使用：用版权来保证软件的自由，并保证它一直是自由的。

1985 年，GNU Emacs 16 发布，斯托曼起草了一份基于版权的许可证，允许用户随意复制并分发该软件。他还允许软件的用户分发修改后软件，但是要求分发后的软件与原来的软件使用相同的许可证。其它人不能夺取修改后的软件的版本，也就不能像 Gosmacs 一样把软件变为私有软件。并且，许可证还要求软件的源代码必须是可获取的。这些限制条件就杜绝了非自由的 GNU Emacs 版本的出现。

斯托曼认为，Emacs 16 的版权声明虽然为 Emacs 公社树立了一个很好的样板，但它的内容还“不够正式”。不久以后，自由软件基金会开始起草一份严密的许可证，基金会的一行管理人员和帮助创建基金会的律师也都参与其中。

马克·费雪是波士顿一名专注于知识产权法的律师，他回忆起当年与斯托曼讨论许可证问题的情景：“对于许可证的用处，斯托曼有非常清醒的认识。他有两条基本原则，第一条是要让软件变得尽可能的开放（费雪说这番话时，也许是受到了开源运动中一些表述的影响，斯托曼本人从来没有打算让软件变得“开放”），第二条是尽可能鼓励其它人接受和使用这种对软件不多加限制的软件许可证。”许可证中很多的条款都是为了符合第二条原则而设立的。

这最后一个条款是一个革命性的创造，人们需要一段时间才能真正理解。费雪回忆说，在那个时候，他只是把这个条款看成是 GNU Emacs 与用户之间达成的一成交易，也是相当于是给 GNU Emacs 打上了价格标签。斯托曼并不向软件的最终用户收取费用，但用户需要把自己对软件修改贡献出来作为他们所付出的代价。费雪认为，这条合同条款非常特别——

“在那个时候，让软件用户接受这样的一种软件‘价格’即便不是独一无二，也是数一数二的。”费雪说。

.....

斯托曼说，作出这样的改动是因为他本人对于原来 Emacs 公社条款中自己的“老大哥”角色感到不满。尽管以前有很多用户会向他发来很多有用的改动，他仍然觉得强调要求用户这么做是不公平的。

“要求用户公开所有的改动是不对的，更何况还要求他们把这么多的改动都发给某一个拥有特权的开发者。这种中央集权的方式，与整个社区人人平等的氛围格格不入。”

1985 年，斯托曼发布 GNU Emacs 的新版本时，GNU Emacs 通用公共许可证第一次与公众见面。随后，斯托曼接受了很多来自黑客社区的对改进许可证用语的修改意见。约翰·吉尔摩是最早接受这种软件许可证的黑客之一，他后来成为了一名软件活动家并为 Sun 公司提供咨询服务。作为他工作的一部分，吉尔摩把 Emacs 移植到了 SunOS 上，SunOS 是 Sun 公司内部的一种 Unix 发行版。在移植的过程中，吉尔摩按照 GNU Emacs 许可证的要求把他所作的修改都进行了公开。吉尔摩并不认为遵守 GNU Emacs 许可证是对他工作的一种限制，相反的，他认为这种许可证正好体现了黑客的精神气质。“不过，在那个时候，很多的许可证的条款都还不是很正式。”吉尔摩回忆道。

吉尔摩引用了一个 Unix 工具软件 `trn` 的版权声明，从中也可以看出那时的许可证有多么的不正式。`trn` 是拉里·华尔的一个作品，他后来还开发了 Perl 语言。有了 Patch 工具，Unix 程序可以很容易地把自己的代码以补丁的型式插入到一个大型软件中。华尔很清楚这一点，所以他把下面的版权声明写入了软件的 README 文档：

版权所有 (C) 1985，拉里·华尔

你可以任意的复制 `trn` 软件包或它的一部分，只要你保证不以它来牟利，并且不得假装你是这个软件的作者。²

如此的声明，映射着黑客的精神和气质，也映射出要把这种松散、非正式的精神转换为严谨的法律版权语言的困难之处。在撰写 GNU Emacs 许可证时，斯托曼做了很多的工作，保证许可证的表述中不存在任何可能被别人利用，把这个软件变成私有软件的漏洞。他以一种律师和黑客都可以理解的方式，体现出了黑客的精神。

没过多久，就有一些黑客们开始讨论如何把 GNU Emacs 许可证移植到他们自己的其它软件作品上。1986 年 11 月的一天，吉尔摩受到 Usenet 上一个贴子的启发，写了一封电子邮件给斯托曼，建议对 GNU Emacs 许可证做一些修改：

或许你可以考虑把许可证中的“EMACS”字眼替换为“软件”或其它类似的名词。我们希望在不久的将来，Emacs 就不再是 GNU 系统中最大的一部分，但这种许可证可以适用于整个系统。³

吉尔摩并不是唯一一个建议用这种更通用做法的人。1986 年年底，斯托曼正着手开发 GNU 工程中的下一个里程碑——源代码调试器，与此同时，他也同样在考虑如何修改 Emacs 许可证使它适用于 Emacs 和这个新的调试器。斯托曼的解决方案是：移除许可证中所有的 Emcas 字眼并把它转变为一个保护所有 GNU 软件版权的许可证。GNU 通用公共许可证，缩写为 GPL，就这样诞生了。

在打造 GPL 时，斯托曼使用了软件版本号的表示方式，用小数表示原型版本的许可证，用整数表示成熟的版本。1989 年，斯托曼打入 Unix 阵营的两个重要作品，也就是 GNU 调试器发布的一年后，他发布了 GPL 的 1.0 版本（斯托曼从 1985 年就开始了 GPL 这个项目）。这个许可证包含了一个表明其政治态度的序言：

通用公共许可证被设计成确保你拥有分发或出售自由软件的权力，确保你可以获取软件的源代码，确保你可以修改软件或在别的自由软件中使用这个软件，并确保你了解你拥有以上的权力。

为了保证你的权力，我们禁止任何人剥夺你的以上权力或要求你放弃这些权力。这些限制条件也作为你分发或修改这个软件时所必须承担的责任。⁴

在打造 GPL 时，斯托曼也不得不对原来 Emacs Commune 的非正式协议进行修改。在原来 Emacs Commune 的协议中，社区的成员需要公开任何对 Emacs 所进行的改动，而现在，程序员只需跟斯托曼一样公开那些与相同协议公开的软件的衍生作品的代码即可。斯托曼粉碎了自由软件价格标签，免得它们破坏了自由软件的精神教义。用户可以不受斯托曼的限制对软件进行创新工作，并保证了整个社区都可以获得相同的软件版本。.....

“GPL 的设计过程与其它自由软件开发的过程非常相似，由一个大的社区共同来讨论它的结构、它的条款。社成员的意见可能会相左，所以还需要对它的条款进行调整甚至进行妥协，这样才能更容易为大众所接受。”另一位曾帮助斯托曼一起设计 GPL 的律师杰里·科恩说，“这样设计方式非常有效，GPL 的各个版本从被人们怀疑逐渐变为广为接受。”

1986 年，斯托曼接受 Byte 杂志采访时，把 GPL 总结为一些富有趣味的条款。除了宣传黑客价值观，斯托曼说，读者还应该“把它看成是智力的较量。使用软件贮藏者所建立的法律系统来对抗他们。”五年后，斯托曼开始使用一些不那么敌对的条款来描述 GPL。“我在考虑那些有关感觉伦理、感觉政治和感觉法律相关的问题”，他说，“我需要设法让它可以融入现有的法律系统。从内心来说，这项工作是为一个新社会立法的过程，但是因为我不是政府，所以事实上我并不能改变法律。我需要设法在现在的法律系统的基础上去设计它，这在以前的设计中从来没有这样做过。”

.....

.....

.....

.....

正当斯托曼在思考自由软件有关伦理、政治和法律上的问题时，一位名叫 Don Hopkins 的加州黑客给他寄了一份 68000 微处理器的手册。Hopkins 是一名 Unix 黑客，同时也是一名科幻小说爱好者，他早些时候向斯托曼借阅了这本书。为了表达对斯托曼的感激之情，他在信封上贴了一些从当地科幻小说爱好者社团获得的一些贴纸。其中，有一张贴纸吸引了斯托曼的眼球。“Copyleft (L)，保留所有权利。”在 GPL 的第一个版本发布以后，斯托曼称赞这张贴纸给了他灵感，并把 Copyleft 作为自由软件许可证的昵称。以后，Copyleft 这个名字以及它的标识——一个倒着写的 C，成为了自由软件基金官方对 GPL 的同义词。

德国著名社会学家 Max Weber 曾经说过，所有伟大的宗教都是建立在对神的感召力的常规化或制度化的基础上的。他认为，每一种成功的宗教，都是把神的感召力或这个宗教最初创始人的想法转换成一种社会化、政治化和伦理化的东西，这样才可能长时间在不同的文化之间进行传递。

如果不考虑其宗教本质，GNU GPL 可以看成是现代的去中心化软件开发模式中一种典型的常规化的例子。程序员和软件公司即使不信奉斯托曼，也会很乐意去接受 GPL 表面上所呈现出来的公开的价值。一部分人把 GPL 看成是对自己软件作品的先发制人的保护机制而接受它，另一部分人虽然由于 GPL 的条款过于苛刻而反对它，但依然把它看成是一种很有影响力的担保方式。

Keith Bostic 就是属于后一个群体中的一名黑客，GPL 1.0 发布的时候，他是 University of California 的一名雇员。Bostic 所就职的 Computer System Research Group(SRG) 系，从二十世纪七十年代开始就从事 Unix 系统的开发，并且负责 Unix 中很多重要组件的开发。比如，作为现代 Internet 通信基石的 TCP/IP 网络协议。到了八十年代，Unix 商标所有者 AT&T 开始专注于 Unix 的商业化，他们选择了由 Bostic 和他在伯克利的同事们一起开发的 Berkeley Software

Distribution(BSD) 这个 Unix 的学术分枝作为商用 Unix 的主要技术来源。

虽然伯克利 BSD 的源代码在研究人员和商业软件程序员群体中广为分享, 把它商业化仍然是一个大问题。因为伯克利的代码中也混入了很多 AT&T 的私有代码。所以伯克利版本只能在那些已经向 AT&T 购买了 Unix 源代码许可的机构使用。当 AT&T 提高了它的软件许可费用后, 这种看上去无辜的授权协议, 越来越成为一种负担。

1986 年, Bostic 受到雇佣, 完成一个把 BSD 移植到 DEC 的 PDP-11 计算机上的个人项目。Bostic 回忆说, 在这段时间里, 斯托曼有时会突然来到西海岸, 所以他有机会与斯托曼进行一些密切的接触。“我直到现在还生动的记得当时斯托曼坐在 CSRG 一台借来的工作站上与我争论有关软件版权的问题。争论完了, 我们就一起去吃晚餐, 然后在晚餐过程中继续争论有关版权的问题。”博斯蒂克不无得意地说。

这样的争论最终停止了, 但并不是以一种斯托曼所喜欢的方式。1989 年 6 月, 伯克利把 Unix 中网络相关的代码从 AT&T 拥有版权的部分代码剥离, 并以加州大学许可证的形式发布。许可证的条款非常的开明, 代码的使用方只需在衍生代码中注明原始代码出自于加州大学即可。⁵与 GPL 不同, 对代码的进行商业化的利用是被允许的。唯一制约这种许可证广为应用的原因是: BSD 的网络部分并不是一个完整的操作系统。人家可以学习这份代码, 但它只能与其它私有代码放到一起才能真正运行起来。

在接下来的几年中, Bostic 和其它加州大学的同事们一起, 重新编写 BSD 中所缺少的部分, 把 BSD 转换一个完整的可以自由分发的操作系统。虽然这项工作由于 AT&T 对于 Unix 这个品牌的所有权而陷入了一些法律上的困境, 但是到了九十年代初, 这个项目几乎已经完成了。甚至在这之前, 很多伯克利开发的工具软件已经开始进入到斯托曼的 GNU 工程中。

Bostic 回忆说：“如果没有 GNU 的影响，我想我们不可能如此坚定的走得那么远。GNU 中有一种力量在推动我们的前进，那就是它让我们很欣赏的理念。”

到了八十年代末，GPL 开始在自由软件社区中产生了如地心引力般的效果。一个程序要成为自由软件并不一定要以 GPL 发布，比如那些 BSD 的工具软件，但是如果以 GPL 来发布，就会更直接的表明自己作为自由软件身份。Bruce Perens 说：“我觉得 GPL 的存在，激发了人们去思考自己的软件作品是否可以成为自由软件，以及应该以什么样的许可证来发布它。” Bruce 是一个知名的 Unix 工具软件 Electric Fence 的开发者，也是 Debian GNU/Linux 开发团队的领导。在 GPL 发布后的几年，Perens 说他决定废弃 Electric Fence 所使用的自己土法炮制的软件许可证而改用斯托曼所创造的由律师审查过的许可证。他回忆说：“事实上这是非常容易做到的。”

Rich Morin 是一名程序员，他对斯托曼最初的 GNU 宣言在一程度上是抱有怀疑的，但是他很惊讶的发现各种软件开始慢慢的聚集到了 GPL 的大伞下。作为一名 SunOS 用户组的领头人，Morin 在 80 年代的一项重要工作就是分发一些载有最好的免费软件和自由软件工具的磁带。这项工作要去联系软件的原始作者，询问他们的作品是否有版权保护并且他们是否愿意把这个软件放入公有领域。Morin 说，到了 1989 年左右，很多最好的软件程序都开始使用 GPL 许可证了。”作为一名软件分发人员，我一看到 GPL 这个词，我就知道我不担心版权的问题了。” Morin 回忆说。

为了补偿制作磁带在 Sun 用户组中分发的成本，Morin 需要向订阅者收取一部分的费用。现在，这些程序都改用 GPL 了，Morin 收集软件制作磁带只需要花以前一半的时间，所以从此可以开始有了一点小小的赢利。Morin 有着灵敏的商业嗅觉，自此，他把自己的爱好变成了一项事业，创办了 Prime Time Freeware 公司。

这样的商业探索完全是在自由软件推广计划的日程表中的。斯托曼在 GPL 的序言中说“当我们说自由软件这个词的时候，我们在说的

是自由，而不是免费”。到了 80 年代后期，斯托曼用一句更容易记忆的话来表述这个观点“不要把 free 想象成是免费啤酒中的免费，它是自由演讲中的自由”。

但是从总体上来看，商业界还是无视了斯托曼的诉求。对于少数企业家来说，自由软件所关注的自由依然与自由市场中的自由是一样的。在商业市场的方程式中拿出软件的所有者，你就算是一个最小的软件公司，也可以自由的与 IBM 和 DEC 这样的大公司自由竞争。

Michale Tiemann 是最初几个抓住了这个概念的企业家之一，他是一名程序员，同时也是斯坦福大学的一名研究生。在 80 年代，Tiemann 曾经一度追随 GNU 工程，就像一名爵士乐的发烧友追随一名他所喜爱的艺术家一样。然后，到了 1987 年，GNU C 编译器的发布，让他开始认识到自由软件的巨大潜力。GCC 可以看成是一个“炸弹”，Tiemann 说软件就存在于斯托曼作为程序员的身后。

“就像所有的作家都梦想要写出一部伟大的美国小说一样，80 年代时，每个程序员都在想写一个伟大的美国编译器。”Tiemann 回忆说，“突然间斯托曼完成了这个工作，这实在是很让人感到羞辱的。”

“你在说的其实是一个单点故障，GCC 就是一个”，Bostic 回应说，“那时候没人有编译器，而 GCC 就是唯一的一个。”

与其与斯托曼竞争，Tiemann 决定在他的作品基础来进行创造。GCC 最早的版本大概有 110000 行代码，但是在 Tiemann 看来，这些代码都很容易理解。他只用了不到五天的时间就掌握了这份代码，并且再花了一周时间就把它移植到了另一个新的硬件平台上——国家半导体的 32032 微处理器上。到了下一年，Tiemann 就开始修改这些源代码，并开发了一个原生的 C++ 语言的编译器。有一天，他在贝尔实验室教授一门编程课程时，Tiemann 发现有些 AT&T 的开发者还在努力做相同的事情。

“教室里大概有四五十人，我问他们有多少人在参与开发原生的编译器”Tiemann 回忆说，“主持人说具体的开发人员的人数是机密，但是如果看看这个教室里的人，就可以得出一个大概的结论。”

Tiemann 继续回忆说,不久以后,他就灵机一动有了个好想法。“我已经在个项目上投入了 6 个月的时间。我开始思考,倒底是我自己还是这份代码带来了这样的高效率,谁更应该在自由市场中获得奖励?”

Tiemann 从 GNU 宣言中获得了更多的灵感,这些内容虽然会伤害到一些软件开发商,但是它鼓励其它软件开发商可以从消费者的角度出发,更好的利用自由软件。虽然 GPL 让商业软件无法垄断市场,但是它让一些更有眼光的开发商可以从提供服务和咨询的角度去开拓业务、开展竞争,而这些领域其实是软件市场中利润最高的两个地方。

在 1999 年的一篇文章中, Tiemann 回忆起斯托曼的宣言带来的影响。“它听上去像是一个社会主义者的争辩,但是它们还是大不相同的。我在里面发现了一个隐藏着的商业计划。”⁶

.....

..... John Gilmore 是另一位 GNU 工程的粉丝, Tiemann 和他一起成立了一个软件咨询服务,专注于提供对 GNU 程序的定制化服务。这就是后来的 Cygnus 公司,它在 1990 年 2 月时与客户签订了第一份开发合同。到 1990 年年底,他已经签订了价值超过 72.5 万美元的软件支持和开发合同。

.....

GNU Emacs, GDB 和 GCC 是三个最为重要的开发工具,不过它们并不是斯托曼在 GNU 工程最初五年中开发的唯一软件。截止 1990 年,斯托曼还开发了 GNU 版本的 Bourne Shell (改名为 Bourne Again Shell, BASH), YACC (改名为 Bison) 和 awk (改名为 gawk)。与 GCC 一样,这些 GNU 程序都被设计为可以在各种系统上运行,而不是局限于某一个开发商的平台。斯托曼和他的同事们不但把程序做得更具弹性,也把它们做得越来越有用。

Prime Time Freeware 公司的 Morin 指出了一个很简单但很重要的自由软件包“hello”,这个软件包体现了 GNU 软件大一统的设计理念。Morin 说,“hello 软件包其实里面就是一个五行代码的 hello worldC 语言程序,不过按 GNU 发行包的标准进行了打包。它还包含

了 Texinfo 的文档和 configuration 脚本。这个软件包为其它的软件开发者提供了一个参考，展示了很多 GNU 工程是如何让软件包可以平滑的移植到不同的环境中。这件事非常重要，这不但影响到所有斯托曼的软件作品，也影响到其它所有的 GNU 工程的软件。”

在斯托曼看来，开发新的软件是第一要务，而改进软件则是位于第二位的。斯托曼在接受 Byte 杂志采访时说：“对于每一个软件作品，我也许可以去改进它。但有时重新实现整个软件也许是更好的做法，因为这样会让整个系统变得更好。在一定程度上，我在这个领域已经工作了很多年并且接触过很多不同的系统，因此，我有很多想法可以对其造成影响。”⁷

不管怎么说，GNU 的工具在 80 年代后期发挥出了它的优势，斯托曼的人工智能实验也以能设计出严密的程序而在整个软件开发社区中出名。

Jeremy Allison, 20 世纪 80 年代一名 Sun 的用户，同时也是一名程序员，是 Samba 这个自由软件项目的维护者。他在 90 年代时曾经嘲笑过 GNU 的工具。在 80 年代后期，Allison 开始使用 Emacs，并对它的社区开发模式产生了兴趣。Allison 故意给斯托曼发去了一段很可能会被他拒绝的代码。

Allison 说：“这就像是 the Onion 的头条新闻，孩子们在祈祷，但是上帝说‘不’”。

.....

斯托曼作为一名程序员的名气越来越大，但他也在努力尝试转型为一名项目经理。尽管 GNU 工程在开发各种系统工具上捷报频传，它还是没有能够开发出一个真正可用的内核。所谓“内核”，是指在所有的 Unix 系统中起到交通警察作用的核心程序，它决定了各种设备和程序应该如何去使用处理器和各种资源。到了 80 年代末，这个情况引发了越来越多的抱怨。跟其它 GNU 工程的软件一样，斯托曼先是试图寻找一个现有的程序，通过修改它来开展内核的开发。根据 1987 年 1

月的“GNUsletter”所描述的，斯托曼已经开始尝试修改 TRIX，TRIX 是一个麻省理工学院开发的 Unix 内核。

这个时期，内核开发的延迟只是困扰斯托曼的众多问题中的一个。1989 年 Lotus Development Corporation 起诉了一家竞争对手，Paperback Software International，控告他抄袭了 Lotus 的 1-2-3 电子表格程序中的菜单设计。Lotus 的官司，加上苹果和微软的“视觉体验”的斗争，给 GNU 工程带来了颇多的麻烦。虽然这两场官司跟 GNU 工程都没有直接的关系，因为他们都主要是有关个人电脑上的操作系统和应用软件而不是类 Unix 的硬件系统，但是他们对整个软件开发的文​​化产生了深远的影响。斯托曼觉得自己不能袖手旁观，他召集了几个程序员朋友在一本杂志上刊登了一则广告。然后他基于这则广告，成立了一个抗议小组，反对那些参与诉讼的公司。他把自己的这个组织命名为“自由编程联盟”，他们在 Lotus 公司和波士顿审理 Lotus 这起案子的法庭外发起了抗议。

这次抗议非常有效果。⁸

他们在软件工业发展史上留下了一笔。应用软件正悄悄的取代操作系统成为软件公司主要的战场。然而，GNU 工程想要建立一个自由操作系统的愿景都尚未实现，这使它看起来有点跟不上时代了。不过也正是这个原因，让斯托曼觉得有必要组织一股新的力量来对付这次对“视觉体验”诉讼，这在一些观察家眼中则更显得 GNU 工程正在变得过时。

.....

1990 年，John D. 和 Catherine T. MacArthur 基金会授予了斯托曼 MacArthur 院士的称号，嘉奖他的天才能力，并授予他“天才奖金”。这份 24 万美元的奖金用于资助 GNU 工程并声援自由软件哲学，这帮助 GNU 工程缓解了些短期的困难。最重要的一点就是，它让斯托曼这个在 FSF 不拿薪水而靠做咨询来养活自己的员工，可以有更多的时间专注于编写 GNU 的代码。

略带讽刺意味的是，这个奖励也赋予了斯托曼选举权。在获奖前不久，斯托曼的公寓的一场大火几乎让他损失了一切财产。在获奖的时候，斯托曼还只是在 545 科技广场存身的一个流浪汉⁹。“选举中心不同意我把那里作为我的住址，”斯托曼回忆说，“不过报纸上那篇 MacArthur 奖新闻也么形容我，于是他们就让我注册了。”¹⁰

最关键的是，MacArthur 捐助的资金让斯托曼有了更多的自由，他可以全身心的投入软件自由的事业了。有了这些资助，他决定更多的去四处宣讲 GNU 工程的使命。

很有意思的是 GNU 工程和自由软件运动最后的胜利差不多就是源于这些旅行。1990 年，斯托曼去芬兰赫尔新基的 Polytechnic 大学访问。在听众中，有一名叫 Linux Torvalds 的学生，那时他只有 21 岁，后来他就是 Linux 内核的发明者。Linux 内核完美的填补了 GNU 工程中所空缺的那一部分。

托瓦兹那时在附近的赫尔新基大学上学，斯托曼给他留下的第一印象是颇为奇怪的。“我这辈子头一次看到这样一个长头发、长胡子的古怪黑客。”托瓦兹在 2001 年自传《Just for Fun》中回忆道，“在赫尔新基很少能见到这样的人。”¹¹

托瓦兹并不了解斯托曼的宏大计划中的政治性的一面，但他仍然很欣赏这个计划深层次的逻辑：没有人可以写出没有错误的代码。通过共享软件，黑客们把不断改进程序的目标放到了个人野心之上。

与他同辈的很多程序员一样，托瓦兹并不关注类似于 IBM 7094 这样的大型机，而是那些五花八门配置的家用电脑。作为一个大学生，托瓦兹是通过在学校的 MicroVAX 上学习 C 语言，一步步进入 Unix 的世界的。这种阶梯式的学习曲线使托瓦兹对于访问主机的障碍有着一种不同的视角。对于斯托曼来说，主要的障碍是官僚作风和特权控制。对于托瓦兹来说，则是地理位置和赫尔辛基寒冷的冬天。为了避免穿过整个赫尔辛基大学的校园去登陆他的 Unix 系统，托瓦兹很快开始设法找到一种可以在他温暖的校外公寓中就访问 Unix 的方法。

这项研究把托瓦兹引向了 Minix 操作系统，Minix 操作系统是 Dutch 大学教授 Andrew Tanenbaum 所开发的一个用于教学用途的轻量级 Unix 系统。这个系统可以在 386 电脑上正常运行，而这已经是托瓦兹能够负担的最强大的机器，但是，这个系统仍然缺乏一些必要的功能。最明显的一点是他没有终端模拟的功能，有了这项功能能让托瓦兹可以在家里模拟出一个学校电脑的终端并登陆到学校的 MicroVAX 计算机上。

.....

1991 年的夏天，托瓦兹把 Minix 重头重写了一遍，并加上了上了一些他所开发的特性。到夏天结束的时候，他把这个作品称为“GNU/Emacs of terminal emulation programs”。¹²

..... 他对这个作品很有信心，并在一个 Minix 新闻组上找到了一份 POSIX 标准，POSIX 标准定义了一个程序是否是 Unix 兼容的。几周后，托瓦兹发布了一则让人可以联想到斯托曼在 1983 年启动 GNU 工程时的所发的贴子的消息：

使用 Minix 操作系统的各位大家好。

我正在开发一个（自由的）操作系统（只是出于兴趣，这个系统不会像 386 (486) AT 上的 gnu 系统那么庞大和专业）。这个系统从 4 月份开始开发，现在已经逐渐可用了。我期待听到各位喜欢或不喜欢 Minix 的人们的反馈，因为这个系统参考了它的不少设计（比如出于实用的需要，它们的文件系统采用了相同的物理布局，除此之外还有一些别的相同之处）¹³

在不到一个月的时间内，这个贴子收到了大量的回复。托瓦兹在 Internet 上的一个 FTP 服务器上发布了这个操作系统的 0.01 版，这是最早向公众开放征求意见的版本。为了实现这个目标，托瓦兹需要给这个新的系统起一个名字。在他自己的电脑硬盘上，这个程序被命名为 Linux，这个名字体现了它是一个 Unix 的变形版本，所以名字以

字母 X 结尾。托瓦兹认为这个名字太过于自恋，所以在 FTP 上把它的名字改成了 Freax，但 FTP 管理员又把名字改了回来。

尽管托瓦兹准备开发一个完整的操作系统，但他和其它开发人员都意识到大部分重要的工作软件都已经有了现成的版本，这些软件是由 GNU、BSD 和无数自由软件开发者共同完成的。对于 Linux 的开发团队来说，GNU C 编译器是这些工具软件中最重要的一个，有了它才使用 C 语言进行开发成为可能。

.....
.....

你花了你生命中宝贵的半年时间去做一个东西，并且希望它能广为流传并从中获利，但你不希望别人用它来牟利。我希望人们都能了解 Linux，并且能对它进行修改，让它更符合每个人的期望。但我同样希望能知道别人都拿它来做了一些什么，我希望永远可以看到那些别人所改动的源代码，这样的话，我也能有机会对它继续进行改进。¹⁴

Linux 的 0.12 版本中第一次完成的集成了 GCC，那时托瓦兹决定在这次发布时宣布他与自由软件运动的联盟。他废除了 Linux 内核原来的许可证，并替换为 GPL。这个决定带来了一场移植软件的狂欢，托瓦兹和其它 Linux 爱好者寻找了各种 GNU 程序并把它们移植到了 Linux 中。在不到 3 年的时间后，Linux 开发者发布了第一个可以用于生产环境的 Linux 版本，1.0。里面包含了经过大面积修改的 GCC、GDB 和一系列 BSD 工具。

到了 1994，这个混杂了各种程序的操作系统在黑客圈子里赢得了极大的关注，很多观察者因此认为托瓦兹在切换到 GPL 后的最初几个月里，并没有把程序真正的交给社区。在第一期《Linux 通讯》杂志中，出版商 Robert Young 对托瓦兹进行了一次面对面的采访。当 Young 向这名芬兰程序员提问是否后悔放弃了对 Linux 源代码的私有

权时，托瓦兹的回答是否定的。托瓦兹说：“即使是做一次事后诸葛亮，我仍觉得选择 GPL 是 Linux 项目初期一个最佳的设计决策。”¹⁵

这个决策完全与斯托曼和自由软件基金会所宣称的 GPL 的普适性完全没有冲突。虽然斯托曼一开始并没有注意到这点，但是 Linux 的爆发式的成长反过来让 Emacs 受到了更多的关注。这一次的创新，决非是一个像 Control-R 这样的小小的软件技巧，而是让一个类 Unix 系统可以在 PC 上运行的新奇事物。虽然动机不同，但是结果是一致的：创造一个只由自由软件构成的完整的操作系统。

就像托瓦兹在最初发表到 comp.os.minix 新闻组的电子邮件所写的样，他最初确实只是把 Linux 作为是 GNU 开发者发布 HURD 内核前的一个替代品。这个初始的想法对于自由软件基金会而言是一个打击。以托瓦兹所设想的，他只是喜欢把玩具拆开再组装起来的广大儿童中的一员。更不要说只要在一块被废弃的硬盘上用闲暇时光就可以创造出来的一个成功的项目。托瓦兹用放弃对 Linux 的控制并接受 GPL 的施舍表现出了他年轻而杰出的智慧。

托瓦兹回忆起 1991 年斯托曼的演讲以及其后他决定对 Linux 使用 GPL 授权的往事，他写道：“我也许没有看到指路的明灯，但我从他的演讲中领悟到了一些东西。”¹⁶

Endnotes

¹参见哈尔·艾贝尔森、麦克·费雪和乔安妮·科斯特洛的《软件与版权法》，1997 年修订版。<http://groups.csail.mit.edu/mac/classes/6.805/articles/int-prop/software-copyright.html>

²参见 Trn Kit README。<http://stuff.mit.edu/afs/sipb/project/trn/src/trn-3.6/README>。

³参见约翰·吉尔摩寄给作者的电子邮件。

⁴参见理查得·斯托曼等“GNU 通用公共许可证：第一版”1989 年 2 月。<http://www.gnu.org/licenses/old-licenses/gpl-1.0.html>

⁵.....

⁶.....

⁷ 参考理查德·斯托曼, BYTE (1986).
8
9
10
11
12
13
14
15
16

Chapter 10

GNU/Linux

1993 年，自由软件运动走到了一个十字路口。从乐观的方面讲，黑客文化正在变得越来越成功。《连线》杂志作为一本新兴的时尚杂志，用有关数据加密、新闻组、自由软件等热门话题树立了自己的地位。而 Internet 这个原本之在黑客和科学家群体中流行的词汇，也渐渐的融入了主流社会，连时任美国总统的克林顿也不忘提到它。昔日作为数码爱好者玩具的个人电脑，也得到了广泛的普及，黑客们开发的软件也进入了新一代的计算机用户的视野。然而，GNU 工程却还没完成它创建一个完整的自由操作系统的目标，用户们通常都使用 Linux 系统。

不管你怎么想，但这听起来是个好消息，或者至少看起来是。经过了十年的挣扎，黑客和黑客的价值观终于开始被主流社会所接受。大众开始认识它们。

事实确实如此吗？从悲观的角度来说，任何被接受的信号也同时意味着麻烦正在滋长。当然，成为一个黑客突然变成了一件很酷的事情，但是对于一个社区而言被人疏远的繁荣真得算是一件好事吗？当然，白宫表明态度说 Internet 是一个很好的趋势，甚至已经在上面注册了自己的域名，whitehouse.gov，但是它也在与各个公司、支持审查

机关和执法机构一同开会，设法管理西部荒原般的 Internet 文化。是的，个人电脑正在变得越来越强大，但是 Intel 做为 PC 市场上的芯片供应商，已经形成了一种由私有软件开发商垄断的局面。虽然有很多人开始使用通过 Linux 接触到自由软件，但是与此同时有成百上千的人成为了微软 Windows 的用户。.....

.....

到头来，这是 Linux 奇妙的天性。没有像 GNU 那样的设计错误，也有 BSD 那样的法律争论，Linux 的高速发展完全是出乎意料的，它的成功也是偶然的，甚至于编写了它的代码的程序员们也没法想象事情为什么会是这样的。Linux 比一个完整的操作系统更为复杂，它完全就是一个黑客们把各种顶级工具混杂在一起的产物：从 GCC、GDB 和 glibc（GNU 工程新开发的 C 库）到 X（MIT 的计算机科学实验室开发的基于 Unix 架构的图形化用户界面），还包括 BSD 开发的那些类似于 BIND（the Berkeley Internet Naming Daemon，用于提供把 IP 地址与域名进行映射的网络服务）和 TCP/IP 协议。整个体系的顶石，当然就是 Linux 内核本身，它是一个在 Minix 基础上重新开发的全新的内核。托瓦兹和他高速成长的 Linux 开发团队并没有把所有东西都从头造一遍，而是遵循了古老的毕加索的格言：“能工摹其形，巧匠摄其魂。”后来托瓦兹本人也以类似的说法评价他自己成功的秘诀：“我是一个非常懒惰的人，并且喜欢把别人的成果占为己有。”¹

这种惰性，从提高效率的角度来看，还是非常值得景仰的，但是从政治的角度来看就变得很棘手。一方面，它表现了从托瓦兹这一方面来说所缺乏的意识形态议程。与 GNU 开发者不同，托瓦兹从一开始就没有想过要让他的团队一起来开发一个完整的操作系统，他只是想做一个自己可以玩的东西。就像汤姆·索亚用石灰水刷白篱笆一样，托瓦兹的天才能力并不表现在他的大局观上，而是在于他可以用最快的速度召集到一批黑客来完成开发任务。托瓦兹和他找来一同工作的黑客们成功了，而其他则产生了一些疑问：那么，准确来说，Linux 究竟是个什么东西呢？它是不是对斯托曼在《GNU 宣言》中所表术的

自由软件哲学的一种表现形式呢？还是只是一个人们可以在家用电脑上使用的简单的工具集合呢？

到了 1993 年下半年，越来越多的 Linux 用户开始倾向于后一种定义并开始基于 Linux 开发各种变种。他们甚至开始把这些衍生版本（或者说“发行版”）包装起来并向 Unix 爱好者出售，结果充其量就是些小打小闹。

”这就是 Red Hat 或其它商业发行版出现前的情况。“伊恩·默多克回忆道，那时他还是普渡大学计算机系的一名学生。”如果你翻阅 Unix 的杂志，你会看到很多名片大小的小广告，推广‘Linux’。很多这样的公司都是采用短视的方式来运作，他们并不认为把自己的代码加到 Linux 中有什么不对的地方。”

默多克，作为一名 Unix 程序员，还记得他第一次在家里的 PC 电脑上下载安装 Linux 时被这个系统“扫地出门”的情景。“这是一件充满乐趣的事情”，他说，”这让我感觉到自己参与到了这个项目中。“然而，各种低质发行版的泛滥渐渐消耗了他的早期的热情。默多克认为，最好的参与到这个项目中的方式是开发一个不添加任何额外组件的 Linux 发行版，由于他开始收集各种最好的自由软件工具，打算把它们整合到他自己的发行版中。默多克说，我希望能创造出一个对得起 Linux 这个名字的系统。

为了“激起兴趣”，默多克在网上发布了他的想法，包括在 comp.os.linux 这个 Usenet 的新闻组中。最早回复他的电子邮件中有一封来自 rms@ai.mit.edu。作为一个黑客，默多克立刻就认出了这个邮件地址。这就是理查德·斯托曼，GNU 工程的创始人，一个在默多克心目中被认为是”黑客中的黑客“的人物。在邮件地址列表中看到他的邮件地址，默多克感到有点疑惑。为什么斯托曼这个正在引领自己的操作系统工程的人物，会关心默多克对于 Linux 的所发的牢骚？

默多克打开了这封邮件。

”他说自由软件基金会正在开始进一步关注 Linux，并很有兴趣基于 Linux [sic] 来完成一个操作系统。总体上来看，好像斯托曼对我们的目标很有兴趣，是他们的哲学中的一部分。”

这封信展现了斯托曼立场的 360 度大转弯。在 1993 年之前，斯托曼一直都不干涉 Linux 社区的事务。事实上，当 Linux 在 1991 年开始出现在 Unix 编程的范畴中开始，他都总是回避这个操作系统。斯托曼说，当他得知有一个可以在 PC 平台上运行的类 Unix 系统时，他找了一个朋友帮忙去了解这个系统的相关情况。“他汇报说这个系统是以 System V 的作为原型的，这是一个比较低劣的 Unix 版本。并且，他告诉我，这个系统是不可移植的。”

那个朋友的报告并不正确。Linux 虽然是为基于 386 的机器设计，Linux 同样是生根于低成本的平台。他的朋友没有发现 Linux 一个重大的优势，那就是它是当时市面上唯一一个可以自由修改的操作系统。换句话说，在接下来的三年时间中，当斯托曼还在听取 Hurd 团队的 Bug 报告时，托瓦兹正在赢得广大程序员的支持，他们后来成为把 Linux 移植到各种新的硬件平台的中坚力量。

直到 1993 年，GNU 工程还是没能发布一个可以使用的操作系统内核，对于 GNU 工程甚至于自由软件运动本身都产生了不小的影响。1993 年 3 月，辛姆森·加芬克尔在《连线》杂志上发表了一篇文章，把 GNU 工程描述为一个“深陷泥沼”的项目，全然无视这个项目中所创造出来的那些成功的工具。² 这个项目和作为这个项目非营性的赞助商的自由软件基金会的成员回忆说，当时的心情比加芬克尔的文章本身造成的伤害还要不好，“情况很明显，至少对于那时的我来说是这样，那就是引入一个全新的操作系统需要一个时间窗口”，查瑟尔说，“一旦错过了个时间窗口，人们对新的系统就不会有太大的兴趣，这也正是现在正在发生的情况。”³

1990-1993 年工作期间，GNU 工程一直处在挣扎中。有些人为此批评斯托曼，但 GNU Emacs 早期团队中的一员并后来成为斯托曼批评家的 Eric Raymond 认为，问题的主要原因是体制上的。“自由软件

基金会太骄傲了”，雷蒙德说。“他们把自己的目标从一个成熟的操作系统上转移到了进行操作系统的研究工具”，更糟糕的事，“他们认为除了自由软件基金会以外没人可以影响他们。”

默多克，作为一名并不参与 GNU 工程内部运作的人，更具有包容的观点。”我认为问题的一方面是他们有点过于好高骛远，把钱投入了不正确的地方。”他说。“微内核在 80 年代和 90 年代初期是一个热门的话题。很不幸的是那正是 GNU 工程开始设计他们的内核的时间，结果就是带来了很多的包袱，这些包袱很难被轻易甩掉。”

.....

斯托曼引用了一些客观因素来解释延期的原因。莲花公司和苹果公司的官司带来了政治上的分心，这使得斯托曼不能全心投入到 Hurd 的团队中去。斯托曼还说，GNU 工程中各个部分之间缺乏有效的沟通。“我们做了很多的工作才使调试环境可以正常工作起来。”他回忆道，“而那时维护 GDB 的团队又并不是很合作。”

但是，斯托曼也说，他和其它 GNU 工程的成员低估了把 Mach 微内核扩展成为一个完整的 Unix 内核的难度。“我以为 Marh 内核中与硬件通信的部分已经调试无误了。”斯托曼在 2000 年一次演讲中回忆 Hurd 团遇到的问题时说。“如果有这个为基础，我们可以进展的更为顺利一些”。但是，事实上，调试这些异步的多线程程序非常困难。一些时序处理上的问题会导致文件损坏，这一点也不好玩。最终的结果就是，我们花了很多很多年的时间，还是只能发布出一个测试的版本。

”4

.....

.....

虽说是赶了个晚集，斯托曼还是有着巨大的影响力。当自由软件基金会宣布它会向默多克的软件项目提供资金和精神支持时，其它各方面的支持也开始源源不断的涌入。默多克启动了一个名为 Debian 的新项目，这个名字是用他和他夫人 Deborah 的名字构成的。在短短的几个星期后，这个项目就发布了第一个版本。“[理查德的帮助]

使得这个出于个人兴趣开发的小小的项目在一夜间变成了社区中每一个人都很关注的明星项目。”默多克说。

1994 年 1 月，默多克发布了“*Debian* 宣言”，这个宣言与斯托曼十年前发布的“*GNU* 宣言”在精神上非常一致，它解释了与自由软件基金会紧密合作的重要性。默多克写道：

自由软件基金会对于 *Debian* 的未来具有非常重要的意义。一方面他们可以帮助分发 *Debian*，向全世界发出信号，*Linux* 不是也不应该是一个商业的产品，但这并不意味着 *Linux* 就不如商业产品。如果你们有人不同意这个观点，我可以以 *GNU Emacs* 和 *GCC* 的例子来证明，它们都不是商业软件，但它们都对商业市场产生了重大的影响。

已经到了集中精力关注 *Linux* [*sic*] 的未来的时候了，不应该以 *Linux* 社区毁灭和它的未来的消亡作为代价来实现个人富裕的目标。虽然开发和发布 *Debian* 也许并不能回答我在宣言中提出的问题，但我希望它至少可以帮心大家更多关注这些问题并帮助解决它们。⁵

宣言发布后不久，自由软件基金会提出一个要求。斯托曼希望默多克把他的发行版叫作“*GNU/Linux*”。默多克说，刚开始的时候，斯托曼曾希望他们使用“*Linux*”这个名字，表示 *GNU* 是 *Linux* 系统的核心。但是这个名字在 *Usenet* 和一些即席的黑客团体中调查中得到的嘘声迫使斯托曼退而求其次，改用 *GNU/Linux* 这个相对自然一点的名字。

.....
.....
.....

在黑客的世界中，分支是一个很有意思的现象。虽然黑客伦理中允许一个程序员对程序的代码做任何的改进，大部分黑客都希望自己的改动能进入程序源代码的主干中，这样才能保证与其它的人程序有最大程度上的兼容。在 *Linux* 开发的初期就对 *glibc* 创建分支就意味着可能会失去成百上千的 *Linux* 开发者。这也会造成 *Linux* 与斯托曼和 *GNU* 团队期望开发出的 *GNU* 系统之间的不兼容性。

作为 GNU 工程的领导者，斯托曼早在 1991 年时就体会到了创建软件分支的不良后果了。那时候，有一批就职于 Lucid 公司的 Emacs 开发人员与斯托曼闹翻了，因为斯托曼不原意把他们对 Emacs 代码修改合入 Emacs 代码主干，于是他们不得不创建了一个名为 Lucid Emacs 的分支版本。⁶

.....
.....
.....
.....

虽然有不少人觉得把 Linux 看成是 GNU 工程的一个变种在政治上未免显得有点贪心，默多克却对自由软件情有独钟，他觉得斯托曼要求把 Debian 称为 GNU/Linux 是合理的。“团结开发者比争抢名号更为重要”，他说。

很快，默多克又收到了一些更为技术化的要求。尽管默多克已经习惯于这些政治化的问题，在实际的软件设计和开发模式上，他仍然坚持他自己的模式。这起先被看成是 Debian 开发社区一种团结的象征，但是慢慢的这成为 Debian 社区与 GNU 工程之间的一些矛盾。

默多克笑着说：“我可以告诉你，在很多方面我与斯托曼的看法都不一致。实事求是的说，理查德是一个很与之合作的人。”.....

1996 年，默多克从普渡大学毕业后，打算找人接手正在蓬勃发展的 Debian 项目。那时候，他已经把主要的管理职责都转交给了布鲁斯·佩伦斯 B，布鲁斯·佩伦斯是以 GPL 发布的 Unix 下一个知名的软件 Electric Fence 的作者。佩伦斯与默多克一样，当 GNU/Linux 的与 Unix 的巨大相似性一经展现就喜欢上了它。他也和默多克一样，对斯托曼和自由软件基金会的政治目标抱有极大的兴趣，即使还有没有真正与他们有过实际的接触。

“我记得斯托曼设计出了《GNU 宣言》，GNU Emacs 和 GCC 后，有篇文章说他正在为 Intel 提供咨询服务。”佩伦斯说，回忆起他在 80 年代末期第一次与斯托曼接触的情景。“我写信给他，问他是怎么做到

一方面宣扬自由软件又一方面为 Intel 工作的。他回信说：我是作为 Intel 开发自由软件的咨询师身份为他工作的。他的回答很有礼貌，而且也很有道理。”

然而，作为一个杰出的 Debian 开发者，佩伦斯对于默多克和斯托曼之间的设计斗争感到失望。作为开发团队的领导者，佩伦斯说他决定让 Debian 与自由软件基金会保持距离。他说，“我决定不采用理查德风格的微管理模式。”

佩伦斯表示，斯托曼对这样的决定大吃一惊，但仍然有能力去驾驭它。“他给了我们一些时间冷静下来，然后发了一个邮件说我们需要建立一种合作关系。他要求我们把这个系统叫作 GNU/Linux，并一直这样称呼它。我觉得这样是可以的。我单方面做出了这样的决定，每个人都舒了一口气。

在很长的一段时间里，Debian 都被看成是最适合黑客们的 Linux 发行版之一，在 1993 至 1994 年这段时间，Slackware 也是一个非常流行的发行版。.....

虽然 Linux 不是一个真正意义上面向黑客的系统，但是 Linux 在商业 Unix 的市场中也找到了自己的位置。在北卡罗莱纳州，一个名叫 Red Hat 的 Unix 公司，把他们的注意力慢慢转向了 Linux。Red Hat 的首席执行官是罗伯特·杨，他是前 *Linux* 杂志的编辑，并在 1994 年时采访过林纳斯·托瓦兹，问他对于把内核以 GPL 来发布是否后悔。对于杨来说，托瓦兹的回答让他对 Linux 有了进一步深层次的认识，他不再寻求通过传统软件销售策略的方式把 GNU/Linux 边缘化，而是开始考虑如果一家公司像 Debian 那样思考问题，会出现什么情况。比如，发布一个完全由自由软件构成的操作系统。1990 年，迈克尔·蒂曼和约翰·吉尔摩成立的 Cygnus Solutions 的公司，早已经向世人证明通过销售自由软件相关的定制化服务是可以盈利的。如果 Red Hat 对于 GNU/Linux 采用类似的策略会如何？

“在西方的科学传统上，我们需要站在巨人的肩膀上。”杨说，指的是托瓦兹和艾萨克·牛顿爵士。“在商业上，这就告诫我们，应该在

前进的途中避免重新造轮子。GPL 模式的美在于把你的代码放入公有领域。⁷如果你是一个独立软件提供商并且想开发一些软件，你需要一个调制解调器拨号程序，但是你没有必须重新开发一个拨号程序。你可以直接从 Red Hat Linux 中“盗取”PPP 的实现，并把它作为你的拨号程序的核心。如果你需要一个图形工具集，你不需要重新去实现一个你自己的图形库。你只需要下载 GTK，就可以立刻应用前人的最佳成果。于是，你就可以专心做你的软件提供商，少花一点时间在软件管理上，而多花一些时间在你客户所想要的功能上。”

杨并不是唯一一个从自由软件的高效性中受到启发的 CEO。1996 年末，大部分 Unix 公司开始清醒过来，并嗅到了源代码的香味。那时候，Linux 距离成为一个完整的商用操作系统还有一两年的路要走，但是黑客社区还是感觉到这一天已经离得很近了：一场巨大的变革即将到来。Intel 386 芯片，Internet 和万维网像一波怪物到来一样深入的影响了整个市场，而 Linux 这个可以自由使用其源代码并以宽松的许可证发布的程序包，则是这一波怪物中最大的一个。

对于试图讨好斯托曼但后来又厌恶斯托曼的徽管理风格的伊恩·默多克来说，他在自由软件运动中花费了很多的精力，这波变革看上去既是对他的赞美又是对他的惩罚。与很多 Linux 发烧友一样，默多克看到过托瓦兹最初的那个帖子，他看到过托瓦兹最初把 Linux 看作是“兴趣爱好”的想法。他也看到过托瓦兹对 Minix 创造者安德鲁·塔嫩鲍姆的敬意：“如果 GNU 内核在去年春天时就已经完成，我就不会再启动我自己的这个项目”⁸ 很多人一样，默多克知道，这种好机会已经错过了。他也看到了 Internet 即将带来的巨大的机遇。

“在早期参与到 Linux 项目中充满着乐趣”，默多克回忆到。“在同一段时间，有些东西在向前进步，有些东西则成为过眼烟云。如果你回头去看看 [comp.os.minix] 上那些老贴子，你会看到这样的观点：这是在 Hurd 开发完成前我们可以先玩着的东西。人们总是性急的。它很好玩，但是如果 HURD 来得更早一些的话，我想 Linux 可能就根本不会出现。”

1996 年底，这些“万一”问题已经盖棺定论。称呼它为 Linux 或是 GNU/Linux，用户们已经用行动证明了。36 个月的时间窗口已经关闭，意味着即使 GNU 工程可以开发出 HURD 内核，除了 GNU 的核心黑客们也不会有更多人会注意它。第一个类 Unix 的自由软件操作系统就在这里，并且充满活力。黑客们所要做的就是安静的坐下来，等待下一波新的想法出现在他们的头脑中。即使是长着蓬乱的头发的理查德·斯托曼本人也不例外。.....

Endnotes

¹托瓦兹在很多场合都引用过这个说法。到目前为止，最确切的一次是在埃里克·雷蒙德的文章“教堂与集市”中出现的。(1997 年 5 月)

²参考辛姆森·加芬克尔的文章，“斯托曼深陷泥沼？”，连线杂志 (1993 年 3 月)。

³

⁴参考他在茂宜高性能计算中心的演讲。.....

⁵

⁶

⁷杨在这里使用了“公有领域”一词，是不严谨的。严格来说，这个词的含义是“不受版权保护”。以 GNU GNPL 许可证发布的代码并没有进入公有领域，因为它必须要保留版权，这样才能对它应用 GNU GPL。

⁸

Chapter 11

开源

[RMS: 在这一章中，我删除了一些引文。因这些内容都只是与开源有关的，与我的生活和工作都没有什么关系。]

1995 年 11 月，自由软件基金会成员彼得·萨卢斯（他于 1994 年出版了《*Unix 的四分之一世纪*》一书），在 GNU 工程的 system-discuss 邮件列表中发布了一个通知，为即将在马萨诸塞州召开的“在剑桥自由发布软件”大会征集黑客的论文。这场由自由软件基金会赞助的大会将于 1996 年 2 月召开，这是第一个纯由自由软件开发工程师的参加的大会，为了表现出自由软件程序员的团结一致，大会欢迎各种有关 GNU、Linux、NetBS、386BSD、FreeBSD、Perl、Tcl/tk 和任何可以自由获取和分发的工具软件相关的文章来大会交流。萨卢斯在通知中写道：

在过去的 15 年中，自由和低成本软件变得越来越重要。本次大会将召集各种可以自由分发的软件的开发者和通过各种渠道发布这类软件的发布者。大会上会有各种教程和

参考文献，也会有林纳斯·托瓦兹和理查德·斯托曼的主题演讲。¹

大会组委会成员埃里克·斯蒂芬·雷蒙德是最早收到萨卢斯的电子邮件的人之一，虽然他不像邮件列表中其它人那样是某个项目的开发负责人或是代表某个公司，但他在 GNU Emacs 的中的贡献以及作为《新黑客词典》（一本收录黑客社会各种黑话的书）的编辑的身份，使他在黑客社区中享有一定的声誉。

对于雷蒙德来说，1996 年的这次大会是一个重要的契机。他在二十世纪 80 年代在 GNU 工程中非常活跃，但到了 1992 以后他就已经不再直接参与这个工程。与许多与他有一样情况的人那样，他对斯托曼的“微管理”风格颇有微词。“理查德对于我在清理 Emacs LISP 库中的代码时进行的修改非常不满。”雷蒙德回忆到，“这使得我感到很沮丧，我再也不想与他一起合作了。”

除了不再直接参与 GNU 工程，雷蒙德在自由软件社区中仍然非常活跃。当萨卢斯建议在大会上由斯托曼和托瓦兹做主题演讲时，雷蒙德非常强烈的建议选用别的方案。斯托曼是老一辈 ITS/Unix 黑客的传承人而托瓦兹是新一代的更有活力的 Linux 黑客的代表，这样的搭配是一种联盟的象征，但只会对一部分像雷蒙德这样的年轻有抱负的黑客有好处。”我像是那个脚踏两只船的人。”雷蒙德说。

会议召开的日子一天天临近，两个团体之间的紧张关系也变得更加明显。两个团体在一个问题上的观点是一致的：这次会议是他们第一次有机会当面见到芬兰神童托瓦兹。托瓦兹出人意料的证明了自己是一个出色和平易近人的演讲者。除了一点点轻微的瑞典口音，托瓦兹敏捷的思路和谦虚的才智让到场的听众感到惊讶。²

更让人吃惊的是，雷蒙德说，托瓦兹对其它杰出黑客们的无情抨击，包括所有黑客中最杰出的那位理查德·斯托曼。在大会的最后，托瓦兹的半黑客半懒鬼的作风赢得了新老两代与会人员的青睐。

“这是一个关键的转折点,”雷蒙德回忆道,”在 1996 年之前,理查德是整个自由软件文化的唯一一个权威的精神领袖。持有不同意见的人都并不在公众面前直接表达。托瓦兹是第一个打破这个禁忌的人。”

打破禁忌的最后一击出现在大会的最后。在一个有关微软 Windows 在市场中占有垄断地位的讨论中,托瓦兹承认他自己是微软的 PowerPoint 这个幻灯片制做软件的发烧友。从老一辈自由软件纯化论者的眼光中,这就像是一个摩门教徒在教堂中自夸自己对威士忌酒的热爱。从托瓦兹和他越来越多的追随者的言论中,这样的观点并不奇怪。为什么为了表明自己观点就得刻意去回避私有软件呢?做为黑客,不是为了去忍受,而是为了把工作完成。

“这是一个振聋发聩的观点,”雷蒙德回忆说,“然而,他可以做到这一点,因为 1995 年和 1996 年,他正不断的接受打击。”

但斯托曼表示他完全不记得 1996 年的大会上有什么紧张的气氛,只不过,他记得是那次被托瓦兹的厚脸皮刺激到了而已。“在 Linux 的文档中,有一段描述,让人们把 GNU 编码标准打印出来,然后再把它们撕烂。”斯托曼举例道。.....

“好吧,所以他对我们的一些做法持有反对意见。有不同意见是可以的,但是他选择了一种格外令人生厌的方式去表达这样的观点。他完全可以换个说法:“我觉得你可以使用另一种方式来缩进你的代码。这样就不会有攻击性了。”

对于雷蒙德来说,其它黑客给予托瓦兹的热烈评价和真诚的欢迎从另一个角度上证实了他的怀疑。Linux 开发者与 GNU/Linux 开发者之间几乎存在着一个代沟。很多像托瓦兹那样的 Linux 黑客是在私有软件的世界中成长起来的。除非软件的质量确实很低下,否则没什么人会对软件的许可证问题发出抱怨。在自由软件的世界中,也许隐藏着一个 PowerPoint 的自由软件替代品。但是,在黑客们真正开始转向使用这个软件前,为什么要嫉妒微软开发了这样一个好用的软件并对此保留版权呢?

.....

在 GNU 工程开始的前十年中,斯托曼已经在程序员中建立起了威信。他也在软件设计和人员管理中建立了一种不妥协声望。.....

在 1996 年大会开始前不久,自由软件基金会遭遇了一次大面积的离职事件,事件的起因很大程度是因为斯托曼。Brain Youmans, Salus 招聘的现任的 FSF 职员回忆到:“在那时候, Peter [Salus] 是留在办公室中的唯一职员”。.....

.....
.....
.....
.....
.....

另一方面,雷蒙德觉得,托瓦兹找到了一种突破 Brooks 法则的方式。在 IBM OS/360 项目经理 Fred P. Brooks 于 1975 年出版的《人月神话》一书中, Brooks 提出了 Brooks 法则: 向一个软件项目中增加开发者的作法只会造成项目的进一步延迟。对于黑客们来说, 写软件就像做汤, 增加厨师的人数对于改善汤的口味并没有太多作用。雷蒙德从中感觉到了一些革命性的变化: 托瓦兹确实在聘请更多的厨师进入厨房的同时, 做出了更好的软件。³

雷蒙德把他的发现写成了论文, 并把它变成若干次演讲, 在宾夕法尼亚州 Chester County 的一些朋友和邻居们面前宣传。在一个名为《教堂与集市》的演讲中, 他比较了 GNU 工程与托瓦兹和内核黑客位使用的管理风格进行了比较。.....

雷蒙德说这些演讲的反响很热烈, 不过都不如次年春天他在德国召开的 1997 Linux Kongress 上的那次演讲。

”在 Kongress 上, 演讲结束后听众都起身鼓掌“, 雷蒙德回忆道。”我把这次成功归纳成两个因素。首先, 这说明听众们为所听到的内容感到激动。其次, 这意味着即使存在一些语言交流上的障碍, 听众们依然感到兴奋。“

最后，雷蒙德把这些演讲的内容写成了论文，同样起名为《教堂与集市》。这篇论文的题目来自雷蒙德的核心思想。GNU 程序就是“教堂”，它们都是有计划的修建的宏伟的黑客精神的纪念碑，经得起时间的考验。而另一方面，Linux 则更像是一个嘈杂的大集市，它是在 Internet 去中心化的松散组织结构中开发出来的。

相似的类比同样也适用于斯托曼和托瓦兹。斯托曼代表着经典的教堂架构，他像一个程序魔法师一样，在消失了 18 月以后，带来了类似于 GNU C 编译器这样的神作。而托瓦兹则像是一个亲切的宴会主人。他让别人来领导 Linux 的设计讨论，并只在关键时候提出一些参考建议。托瓦兹所创建的这种开发模式正好反映了他的懒散个性。从托瓦兹的观点来看，最好的管理工作不是要加强对事情的控制，而是要保持思维的活跃度。

雷蒙德总结说：“我觉得 Linux 最聪明的地方和最重要的 hack 就是创造出来 Linux 内核，而不是他所创造的 Linux 开发模式。”⁴

在总结托瓦兹的成功管理经验方面，雷蒙德做自己也完成得非常出色。…………跟预期的情况一样，听众并不仅仅被黑客所震惊，也有很多人对自由软件运动的快速成长表达了强烈的兴趣。Netscape 就是其中的一个，。这家成立于加州山景城的创业公司，刚刚经历了一场持续了三年的与微软争夺 Web 浏览器的市场的争夺。出于对雷蒙德演讲的好奇和希望赢回失去的市场份额的期望目的，Netscape 的执行官们居间把这次演讲的信息带回了公司总部。几个月以后，1998 年 1 月，Netscape 宣布计划把它的旗舰级产品 Navigator Web 浏览器的代码公开，希望能得到黑客们的支持，完成以后的开发。…………

Netscape CEO Jim Barksdale 说雷蒙德的“教堂与集市”文章对于公司做出这样的决策有着深远的影响，公司很快把雷蒙德地位抬升到了黑客界的知名人士。为了紧紧抓住这次机会，雷蒙德出差去西海岸去进行采访，向 Netscape 的执行官们提供建议，并且参与了 Netscape Navigator 源代码开放庆祝的聚会。Navigator 的源代码的 code name 是“Mozilla”，这个名字一方面象征这个程序的源代码有 30 万行的

代码，同时也象征着它的血统。作为一个 Mosaic 浏览器的非商业化版本，Marc Adressen 在 Illionis 大学开发这个浏览器时也参考了 Mosaic 的设计，这就再次证明了大部分程序员在打算开发一个新的程序时，总是会借鉴一些现有的、可以修改的程序。.....Tim O'Reilly 当时也在 Linux Kongress 的会场聆听演讲，他是 O'Reilly & Associates 出版公司的创始人，这家公司专注于出版软件使用手册和软件相关的书籍（同样也是本书的出版商）。听完了雷蒙德在 Kongress 上的演讲后，O'Reilly 迅速地邀请雷蒙德参加下半年在加州 Monterey 举办的首届 Perl 大会，在会议上再次分享这个主题。.....

雷蒙德不记得这次讨论的内容，但是他记得这次会议中的提到的另一个问题。不管斯托曼和其他黑客如何努力的去提醒人们自由软件中的“Free”一词是指“自由”而不是“免费”，但这一点仍然没有得到大众的真正认知。第一次听到这个名词时，许多商业管理者都把它当作是“零成本”的同义词，而并不注意它蕴含着的其它相关信息。除非黑客们能找到一种方法解决这个认识上的偏差，否则自由软件运动就会面临一个高耸难攀的山峰，即使有 Netscape 的案例供参考，也很难一举登顶。

于是，彼德森成立了一个组织，专门致力于推广自由软件，他们提供了另一个词语：开源。

彼德森回忆道，她是在与一个朋友讨论 Netscape 的决定的时候，想到了“开源”这个词语。她不记得是她自己首创了这个词语，还是从别的什么领域借用了这个词语，但是她确实记得她的朋友并不喜欢这个词语。⁵

彼德森说，但是在这次会议上，这个词语得到的反馈却是截然不同的。彼德森回忆说：“我很犹豫要不要提出这个建议，我对这个组织并不了解，所以我在不经意中去使用这个词语，而不是把它作为一个新词语去强调。”然而，这个词语出乎意料的受到了人们的关注。在这次会议结束时，大部分的与会者，包括雷蒙德，对于这个词语都非常满意。

雷蒙德说，在 Mozilla 发布会前几天 O'Reilly 组织的一次讨论自由软件的会议之前，他从来不公开使用“开源”这个词语来替代“自由软件”。O'Reilly 把这次会议叫作“免费软件峰会”，期望把媒体和社区的注意力吸引到其它一些对于 Netscape 发布 Mozilla 具有有价值的项目上去。“这些家伙有如此多的相似之处，但我很惊讶他们互相之间谁都不了解谁。”O'Reilly 说，“我同时也想让世界知道自由软件文化已经产生了如此深远的影响。人们如果再不对自由软件引起重视，就是错过了一次很好的机会。”

邀请这些人聚集在一起，是 O'Reilly 一个深思熟虑的决定，并将产生深远的政治影响。他只打算邀请西海岸的开发者们，比如 sendmail 的作者 Wall, Eric allman, BIND 的作者 Pal Vixie。不过也有一个人是例外，那就是长居在宾夕法尼亚的雷蒙德，因为他正好来参加 Mozilla 的发布会，所以也马上邀了他。同样还有弗吉尼亚的 Guido van Rossum，他是 Python 语言的作者。“Frank Willison 是我的主编，他是 Python 的斗士，他决定邀请 Guido 参加，甚至都没有跟我商量。”，O'Reilly 回忆说，“我非常高兴能请到他出席，不过刚开始的时候，我确实只是想组织一个本地聚会。”

对于一些观察者来说，没有邀请斯托曼参加这次会议可以算是有点怠慢。“正是由于这个原因，我决定不去参加这次会议。”佩伦斯回忆这次峰会时回忆道。雷蒙德参加了这次会议，并认为即使邀请斯托曼也是徒劳的。有关这次会议的组织者 O'Reilly 怠慢斯托曼的传言，在 O'Reilly 与斯托曼公开表明在软件手册的权限问题上的不同意见后，变得更让人信服。在会议前，斯托曼曾表示，自由软件手册应该可以像自由软件本身一样自由的复制和修改。而 O'Reilly 那时认为非自由的图书是对自由软件的增值，它们能让自由软件被更多的社区所熟悉。O'Reilly 和斯托曼还为这次会议的名称争论，斯托曼坚持要使用“自由软件”来代替不那么政治化的“免费软件”一词。

回头去看，O'Reilly 仍然认为不邀请斯托曼的决定是怠慢了他。O'Reilly 说：“在那个时候，我没有见过理查德·斯托曼本人，不过在我

们的电子邮件沟通中，斯托曼看上去是个很强势的人，并且不太愿意加入对话。我希望能够保证在会议中展示出 GNU 的传统，所以我邀请了约翰·吉尔摩和 Michael 蒂曼，这两个人我都比较熟悉，我知道他们对于 GPL 的价值都非常有热情，不过看上去他们更愿意参与有关自由软件项目和传统的优势和弱点的讨论。从后面发生的事情来看，我有点后悔没有邀请理查德，但是我没有这么做并不能看成是对 GNU 工程或理查德·斯托曼本人的不敬。”

无论是不是怠慢了斯托曼，O'Reilly 和雷蒙德说的“开源”一词成功的赢得了很多与会者的支持。与会者分享了各种想法和经验，并且一起进行了一场头脑风暴，讨论如何提升自由软件的形象。讨论的焦点是如何才能展示自由软件的成功之处，尤其是在实现 Internet 的基础架构方面，而不是讨论 GNU/Linux 是如何与微软 Windows 进行竞争的问题。不过跟很多早期在 VA 召开的会议一样，这些讨论的内容在不经意间就转向了讨论“自由软件”这个词自身所存在的问题。作为会议的主办者，O'Reilly 记得峰会的与会者托瓦兹的一段很有见地评论。

“那时候，Linus 刚刚搬到硅谷来，他解释说其实他本人也是最近才知道‘Free’这个词在英文中有两个意思，一个意思是自由，另一个意思是免费”。⁶

Gygnus 的创始人迈克尔·蒂曼建议用 *sourceware* 一词来代替容易引起误解的“自由软件”一词。”不过大部分人都对这个词没什么兴趣。“O'Reilly 回忆到，”直到 Eric 抛出了‘开源’一词。”

虽然“开源”这个词对一些人很有吸引力，但是要让它取代“自由软件”变为一个官方的用词还有很长的路要走。结束一天的会议前，与会者开始对三个词语进行投票，“自由软件”、“开源”和“*sourceware*”。O'Reilly 说，15 名与会者中有 9 人把票投给了“开源”。虽然有一部分人对这个词还有些意见，不过所有的与会者都同意在以后与媒体的讨论中可以使用这个词汇。O'Reilly 说，“我们希望发出一些团结的信号”。

没过多久，“开源”这个词就进入了词典。这次峰会后不久，O'Reilly 带着与会者们参加了一个纽约时报、华尔街日报和其它著名出版社记者参与的发布会。在短短几个月中，托瓦兹就登上了福布斯杂志的封面，正文中还出现了斯托曼、Perl 的作者 Larry Wall 和 Apache 团队领导者 Brian Behlendorf。开源开始向商业开放。

对于像蒂曼这样的参会者来说，发出团结的信息才是最重要的事情。尽管他的公司通过销售自由软件和相关服务已经捞到了第一桶金，他还是可以感觉到很多其它的程序员和企业在面对自由软件时还是困难重重。

蒂曼说：“毫无疑问，使用 free 这个词在很多场合都会造成误解。开源这个词则对于商业来说更加友好，也更容易引起商业的注意力。自由软件则更是关注于一种道德上的正义。不管好不好，我们都觉得与开源团队合作是有益无害的。”

斯托曼没有马上对“开源”这个新词语作出回应。雷蒙德说斯托曼简单了解了一个这个词语，其实只是为了抛弃它。“我知道这一点是因为我跟斯托曼就这个问题有过直接的对话。”

.....

到了 1998 年年底，斯托曼正式表明了立场：开源这个词在讨论自由软件技术优点时，是一个在沟通时便于使用的词语，但是它也诱使演讲者弱化软件自由的重要性.....。由于存在这样的不足，斯托曼坚持使用“自由软件”这个词汇。

TODO..... 1999 年 LinuxWorld 对话与博览会，托瓦兹把它看成是 Linux 社区产出的宣传会，在这次会议上，斯托曼总结了他的立场，他恳求他的追随者抵抗住诱惑。

斯托曼在一次小组讨论上说：“我们已经展示了很多我们已经取得的成就，所以我们不需要去讨好公司而放弃我们的目标。如果公司能向我们提供帮助，我们可以接受。如果公司不愿意，我们也没有必要去改变自己来争取这些帮助。只要你能不断向着目标慢慢前进，最终

一定能实现目标。如果你们什么事情都采用折中的方式，不真正往前前进，就永远也不能实现目标。”

事实上，在 LinuWorld 之前，斯托曼已经表现出他正在慢慢疏远他的同盟。Freeware 峰会后几个月，O'Reilly 的主办了他的第二年度 Perl 大会。这一次，斯托曼参加了会议。在一次称赞 IBM 决定在他的商用方案中整合 Apache Web 服务器这个自由软件的小组讨论中，斯托曼抢过了听众的话筒，把矛头指向了讨论人 John Ousterhout，他是 Tcl 脚本语言的作者。斯托曼把 Ousterhout 称为是自由软件社区的寄生虫，因为他通过他自己的名为 Scriptics 的创业公司稍售一个商业版本的 Tcl。“我不觉得 Scriptics 是 Tcl 继续存活下去所必须的。”斯托曼在听众们的窃窃私语中坚定地表明道。

“这是一种很尴尬的场面。”Prime Time Freeware 的李奇·莫林回忆说，“John 做出了一些很让人值得尊敬的东西——Tcl, Tk 和 Sprite。他是一个真正的贡献者”。莫林很理解斯托曼以及他的立场，但他也同样理解那些被斯托曼“无理”的行为扫了兴的人们。

TODO.....

TODO.....

斯托曼在 Perl 大会上的爆发，影响了另一位潜在的追随者——布鲁斯·佩伦斯。1998 年，埃里克·雷蒙德打算发行开源倡议组织，OSI，一个监管“开源”这个词语使用并向感兴趣的公司提供标准的组织。雷蒙德请佩伦斯来企企划这个标准。⁷

佩伦斯不久就从 OSI 的高管位子上辞职，他对这个组织与斯托曼和 FSF 对立的立场表示遗憾。同样的，回头去看，在 FSF 以外重新定义自由软件的需求，佩伦斯理解了为什么其它的黑客们仍然觉得需要与之保持距离。“我非常敬仰理查德”，佩伦斯说，“我觉得如果他能更为平衡的处理各方关系的话，他会把事情完成得更好。这包括花几个月时间远离自由软件。”

斯托曼的偏执不太会抵消与开源斗士动力之间的公共关系。1998 年 8 月，芯片厂商 Intel 购买了 GNU/Linux 供应商 Red Hat 的股份，

纽约时报在随后的文章中评价 Red Hat 为“自由软件和开源”运动的产物。⁸半年后, John Markoff 在 Apple Computer 杂志上发表文章, 标题中称该公司使用“开源”的 Apache 服务器。⁹.....

这样的势头与很多公司拥抱“开源”一词的动力非常一致。到了 1999 年 8 月, Red Hat 这家乐意把自己标榜为“开源”的公司纳斯达克上市。同年 12 月, VA Linux 公司(曾经的 VA Research)正式 IPO。这家公司的股票从开盘的 30 美元一股一路彪升到 300 美元, 并最后回稳在 239 美元左右。股东们很幸运的抄了底, 并且享受到了 698% 的纸面收益, 打破了斯达克的历史记录。.....

.....

不管这些评论是否减轻了雷蒙德和其它开源拥护者是为金钱为目的的嫌疑, 但是他们向开源社区传达了一个最基本的信息: 想要宣传自由软件的概念, 只需要一张友善的面孔和合乎情理的信号。不用像斯托曼那样正面与商业市场斗争, 雷蒙德、托瓦兹和其它黑客社区的新领导者们需要一种更宽松的方式: 在某些领域忽略市场, 通过别的方式去影响它。不能扮演被学校开除的那个角色, 而应该成为社会的名流、在这个过程中不断扩大自己的影响力。

雷蒙德说: “在理查德最不如意的日子里, 他认为我和 Linus 托瓦兹搞阴谋绑架他的革命。理查德拒绝开源一词的使用和他故意对我的观点做出的意识形态上的分裂是由于他自己对理想主义和地域性奇怪混合的结果。从外人的眼光看来, 可能会以为这是理查德的个人野心膨胀的结果。我不这样认为。我觉得这不仅仅是野心, 这更是因为理查德把自己和自由软件的理念紧紧的绑在了一起, 在他看来, 任何对自由软件的威胁就是对他本人的威胁。”

.....

让人觉得讽刺的是, 开源和像雷蒙德这样的开源拥护者的胜利并不能撼动斯托曼作为领导者的角色。斯托曼的一举一动都有机会给他带来新的追随者。而雷蒙德的领地的主管则没有这样的能力。斯托曼时常手握武器, 不是因为出于原则, 而更多是出于习惯: 比如, 他一

开始反对 Linux 内核，现在则不愿意成为一个为了软件问题理想冒险的政治形象。”

TODO.....

除此之外，近期关于开源的争论也表明：当斯托曼手握武器时，通常是他找到了一种可以让他的事业有所进展的方式的时候。伊恩·默多克说：“斯托曼最重的一个性格特征就是他不会动摇自己的立场。如果需要，为了让别人接受他的观点，他可以等上十年。”

默多克所发现的斯托曼这个不动摇的特质既新鲜又宝贵。就算斯托曼不再是自由软件运动的唯一领袖，他仍然是自由软件社区的北极星。默多克说：“你可以确定他永远会坚持他自己的观点。大部分人都不是这样的。不管你是否同意他的观点，你都必须尊重他的观点。”因为事实常常证明：他站得更高，看得更远。

Endnotes

¹参考彼得·萨卢斯，“FYI-Conference on Freely Redistributable Software, 2/2, Cambridge” (1995) (由特里·威诺格拉德提供)。

²尽管 Linus 托瓦兹是芬兰人，但他的母语是瑞典语。“The Rampantly Unofficial Linus FAQ” (<http://catb.org/~esr/faqs/linus/>) 对此作出以下解释：

芬兰有很多（大约 6%）说瑞典语的少数人口。他们把自己叫作 “*finlandssvensk*” 或 “*finladsvenskar*”，并且把他们自己认定是芬兰人。他们很多的家庭都在芬兰居住了很多年。瑞典语是芬兰两种官方语言中的一种。

³Brooks 法则是对他的书中如下的描述的简称：

由于设计软件本质上是一个系统性的工作，沟通的成本很高，把任务分割成多个子任务会使沟通成本的大大增加。所以，向软件项目中增加人手只会增加软件开发时间，而不是缩短。

参考 Fred P. Brooks, 《人月神话》(Addison Wesley Publishing, 1995)

⁴参考埃里克·雷蒙德的文章“教堂与集市”，1997 年。

⁵参考 Malcolm MacLachlan, “Profit Motive Splits Open Source Movement,” *TechWeb News* (August 26, 1998), <http://www.techweb.com/article/showArticle?articleID=29102344>.

⁶译注：原文在这里用了两个西班牙语词汇，libre 和 gratis。

⁷See Bruce Perens et al., “The Open Source Definition,” The Open Source Initiative (1998), <http://www.opensource.org/docs/definition.html>.

⁸.....

⁹.....

Chapter 12

开往黑客地狱的短暂旅途

[RMS: 在这一章中，我只添加了少量的注释，以这样的形式出现。
]

理查德·斯托曼双目凝视着车外，眼睛一眨不眨，我们正坐在这辆租来汽车里途经津汇城区，等待着信号灯变绿。

我们正在前往附近一个名为芭雅的小镇，去会见一些软件开发者和他们的妻子，然后一同去参加一小时以后另一地点开场的晚宴。

斯托曼刚刚在茂宜高性能计算中心完成一次演讲，晚宴的开始时间距离演讲结束只有两小时时间。演讲开始前，津汇对我们的到来显得如此的热情，而现在，则让人感觉到它处处在为难我们。跟很多海边城市一样，津汇只有一条主干道贯穿全城。行驶在津汇的主干道上，看着路边的汉堡店、房产中介和比基尼商店，让人感觉到像是一条巨大的绦虫正在吞下一片穿着钢盔铁甲的食物。除了一直向前开，没有别的选择，车流就像一条蜿蜒的溪流。200 码开外，信号灯变绿了。当我们开始往前挪动车子时，灯又变黄了。

对于斯托曼这样长期在东部生活的人来说，把夏威夷午后明媚的阳光浪费在拥堵的马路上足以诱发他的脑血栓。[RMS: 因为如果我在

开车，我就无法回复电子邮件，这对于我来说实在是太痛苦了。] 更糟糕的是，其实在四分之一英里前，有一些可以右转的道路，如果车辆趁早转弯，就可以避免这可怕的交通拥堵。不幸的是，我们需要跟着前面车辆，那辆车的司机是一名实验室的程序员，他认识去往目的地的路，并且是他决定带我们从芭雅的观光道路通行，而不是绕行附近的彼拉尼高速公路。

斯托曼叹着气失望的说：“这实在是太可怕了，我们为什么没走另一条路呢？”

终于，我们前方 1/4 英里处的信号灯变绿了。但我们仍然只能前进很短的一段距。10 分钟内这样的过程重复了好几次，直到我们最终慢慢的挪到了相临一条公路的十字路口后。

我们前面的车没有理会这个十字路口，继续向前行驶。

斯托曼沮丧的挥动着双手，抱怨道：“他为什么不拐弯呢？你们不觉得奇怪吗？”

我决定不必要回答这两个问题。我知道我正与斯托曼在毛伊岛同乘一辆车，他在开车，这让人很难以至信。事实上，两小时前我还不知道斯托曼会不会开车。而现在，我们在车上欣赏着马友友的“阿巴拉契旅行”专辑中大提琴所发出的让人感到悲伤的低音音符，看着太阳渐渐从我们的左边落下，我尽可能的让自己沉浸在这样的氛围中。

当下一个转弯的机会出现的时候，斯托曼打开了右转向灯，试图提醒前车的司机。不过仍然没有好运气。我们又一次慢慢的驶过了十字路口，在距离下一个信号灯至少 200 码的地方停了下来。斯托曼终于忍不住大发雷霆。

“他简直就是故意无视我们的存在。”他愤愤地说。一面又像机场的信号官指挥飞机降落一样打着手势，尝试着吸引向导的注意力。向导像是完全没有注意到这一点，在接下来的 5 分钟里，我们只能在他的后视镜里看到他一小部分的脑袋。

我从斯托曼的车窗向外看，附近的卡胡拉威岛和拉拿夷岛与落日一起构成了一副美丽的画面。这是一副美得让人窒息的画面，我想，如

果你一个夏威夷本地人，一定会因为这样的美景而忘记了堵车的烦恼。我试图把斯托曼的注意力引向这里，但是他的注意力仍然一心集中在前面那个无视我们的司机身上，完全不搭理我。

司机开过另一个亮着绿灯的路口，完全无视边上“下一个路口向右转驶入彼拉尼高速”的标识。我咂了咂嘴。我记起以前一位 BSD 程序员基思·博斯蒂克警告过我：“斯托曼无法容忍傻瓜，如果有人说了或者做了一些什么蠢事，他会看着他的眼睛说：‘这样做太愚蠢了。’”

看着前方心不在焉的司机，我觉得他做的就是所谓的蠢事，而不只是一些不太聪明的事，这些蠢事正在让斯托曼备受煎熬。

”他好像完全没有想过应该如何有效的到达目的地，所以才选择了这么一条路。”斯托曼说。

“有效”这个词像是一股坏气味停留在了空中。很少有事情能比“低效”更刺激到一个黑客的神经了。当年，正是因为施乐的打印机一天要检修两三次所来带的低效才激发了斯托曼想要获取打印机源代码的想法。也正是因为要重头编写被商业公司所绑架的商用软件的低效，才引发了斯托曼与 Symbolics 之间的斗争，从而诞生了 GNU 工程。让·保罗·萨特曾经说，如果别人是地狱，那么黑客的地狱就是重复其它人愚蠢的错误，毫不夸张的说，斯托曼的一生，就是在尝试把人类从地狱的火焰中拯救出来。

这种有关地狱的隐喻在我们缓慢通过这美景时变得更为明显。四处都是商场、停车场和缺乏设计的信号灯，看上去不像是个城市，倒更像是一个设计得很糟糕的软件。城市的规划者把城市设计成所有的车辆都要通过主干道，而不是把车流分散到支路和高速公路上。从黑客的角度来说，坐在车里并被围困在这一团糟的交通中，就像是用最大的音量听一张录有在木板上钉钉子声音的 CD。

“不完美的系统会激怒黑客。”史蒂芬·李维说过这样的话，这是我决定与斯托曼同坐一辆车前应该听取的另一个忠告，“这是黑客们通常不喜欢开车的原因之一：这是一个充满不确定性的程序，交通信号灯总是随机的变化，还有横七竖八的单行道，导致交通经常堵塞。这实

在是太不必要了（李维强调说），只要让黑客们重新安排一下信号灯，打开交通灯控制盒，重新设计整个系统。”¹

更让人感到沮丧的事，是我们的向导的愚笨，他没有像一个黑客那样本能的做出最聪明的选择，选择一条更为聪明的捷径，而是坚持陪着城市设计者玩他们愚蠢的游戏。像但丁《神曲》中的维吉尔一样，不管我们是否希望，他都打定主意要让我们完整的体验这个黑客的地狱。

在我还没有来得及告诉斯托曼我的发现以前，前方的司机终于在路口右转了。斯托曼耸起的肩膀总于放松了一些，车里紧张的气氛稍稍消散了一些。然而，当前面的司机把车慢慢停下来时，紧张的气氛又回来了。道路两侧的放着“前方施工”的标识，虽然彼拉尼高速公路就在前方不到四分之一英里的距离，我们的车与高速公路间的一条两车道的公路被一辆停着的推土车和两大堆土方完全堵住了。

我们的向导在我们的面前忽然把车笨拙的调了一个头，斯托曼一时都没有反应过来。当他看到了面前的推土机和“禁止通行”标识后，他终于再一次爆发了。

“为什么，为什么，为什么？”他抱怨道，把头一仰，“你早该知道前面的路被封住了，你早该知道不能走这条路。你是故意这么做的。” [RMS：我原来的意思是说他故意选择了那条拥堵的路，跟下面所解释的一样，我觉得这里所引用的我说的话并不准确。]

向导的车已经调好了头，从我们身边开过，开回那条拥塞的主干道。在他经过我们身边时，他摇了摇头，给了我们一个抱歉的耸肩。他露出牙齿，司机的姿势显示出了一个外地人的失望，但是这已经被岛国人的宿命所中和了。从我们租来车关闭的窗户望出去，我们似乎看到了这么一句话：“嘿，这就是毛依岛，你想做什么？”

斯托曼再也无法忍受了。

“你可不可以不要再笑了！”他咆哮道，雾气蒙上了他的眼镜片，“这全他妈是你的错。如果听我的走另一条路就不会这样了。” [RMS：这些话记录的似乎并不准确，因为我从来不把“他妈的”当成副词来使

用。由于我说这些话里并不是在接受采访，所以威廉姆斯并没有录下我说的话。我很确定这里的文字确实还原了当时的情景，但是这些话也许是反应了他对我所说的话的个人理解，并不是我的原话。]

斯托曼加重了“听我的”一词，紧紧的抓住方向盘并两次把它拉向自己。斯托曼的样子就像是一个在汽车座位里发脾气的小孩子，他的声音进一步加强了这样的形象。斯托曼又生气又郁闷，眼泪几乎就要掉下来。

幸运的是，眼泪最终没有到来。就像夏天的暴雨，斯托曼的愤怒转瞬即逝。他轻轻叹了口气，把车挂到倒档，并开始调头。当我们回到城市主干道上时，他的表情让人过目不忘，就跟我们提前半小时离开酒店时的表情一样。

我们花了不到五分钟时间，到达了下一个十字路口。这里可以容易的驶入高速公路，很快，我们就加大马力向着芭雅驶去。刚才斯托曼左肩上若隐若现的黄色太阳现在变成了橙红色出现在我们的后视镜中。金色的阳光撒向高速公路两边的树木。

接下来的 20 分钟，只剩下了汽车的声音，包含着引擎和轮胎的轰鸣声。像是大提琴与小提琴的三重奏，演绎着阿巴拉契的民歌旋律。

Endnotes

¹参考史蒂芬·李维的《黑客》一书（企鹅出版社，美国，平装，1984 年）第 40 页。

Chapter 13

斗争仍在继续

对于理查德·斯托曼来说，时间并不能治愈所有的伤口，但却也算是一个不错的医生。

“教堂与集市”一书出版后四年，雷蒙德给斯托曼留下伤口虽已收疤，但仍然在隐隐作痛。斯托曼一直在抱怨林纳斯·托瓦兹被人们提升到了世界上最知名黑客的行列的这件事。他回忆起一件 1999 在 Linux 商展台前一度很流行的 T 恤衫，这件 T 恤衫上的图案模仿了星球大战当年的宣报海报，上面的托瓦兹像卢克·天行者一样挥舞着激光剑，而斯托曼的脸出现在 R2D2 的上方。这件 T 恤衫之所以惹怒了斯托曼是因为它把斯托曼描绘为托瓦兹的死党，同时还把托瓦兹提升到了自由软件和开源社区的领袖位置，即便这个位置托瓦兹本人也不愿意担当。“这实在是太讽刺了！”斯托曼悲哀的说，“拿起宝箭正是林纳斯所不愿意去做的。他希望自己成为一次运动的焦点，所以它并不想发起战争。战争有什么好的呢！”

与此同时，对于托瓦兹来说，“拿起宝箭”也确实不是他想做的，他敞开大门让斯托曼去提升他作为黑客社区的伦理仲裁人的影响力。不考虑他所受的委屈的话，斯托曼必须承认，在过去的几年中，无论是对

他还是对他的组织，这样的结果都是很好的。由于 GNU/Linux 未预计到的成功而被边缘化，但是斯托曼还是成功的重新捕获了倡议。在 2000 年 1 月到 2001 年 12 月之间，他的演讲覆盖了六大洲并且访问了一些对于软件自由这个话题可能存在更敏感的理解的国家，比如中国和印度。

除了像一个出色的布道者那样，斯托曼也学会了如何去利用他作为 GNU GPL 的海岸警卫队员的权力。在 2000 年夏天，当 1999 年 Linux IPO 的泡沫散去的时候，斯托曼和自由软件基金会获得两个重要的胜利。2000 年 7 月，挪威软件公司 Troll Tech 宣布对他们所开发的 Qt 库使用 GPL 许可证，Qt 库是 GNU/Linux 操作系统中一个重要的图形开发库。几周后，一个对开源潮流跃跃欲试却又不想放弃对他们的私有软件控制权的 Sun 公司宣布对他们的 OpenOffice¹ 应用套件采用 LGPL 和 Sun Industry Standards Source License(SISL) 进行双重许可。

对于这些胜利，其实斯托曼并没有做什么有针对性的努力。对于 Trolltech 的案例来说，斯托曼只是扮演了自由软件教主的角色。1999 年，Trolltech 想寻求一种与自由软件基金会要求相符的软件许可证，但在仔细审阅许可证后，斯托曼发现了一些条款在法律上存在冲突，使得 Qt 无法与 GPL 许可证的自由软件绑定在一起。Trolltech 的管理团队疲于与斯托曼在许可证条款上斤斤计较，决定把 Qt 拆分成两个版本，一个使用 GPL 协议，另一个使用 QPL 协议，让开发者可以有办法绕开斯托曼所发现的协定冲突。.....

.....

In 1999, these two efforts were making good progress, and the management of Trolltech were starting to feel the pressure. So Trolltech released Qt under its own free software license, the QPL. The QPL qualified as a free license, but Stallman pointed out the drawback of incompatibility with the GPL: in general, combining GPL-covered code with Qt in one program was impossible without violating one

license or the other. Eventually the Trolltech management recognized that the GPL would serve their purposes equally well, and released Qt with dual licensing: the same Qt code, in parallel, was available under the GNU GPL and under the QPL. After three years, this was victory.

Once Qt was free, the motive for developing Harmony (which wasn't complete enough for actual use) had disappeared, and the developers abandoned it. GNOME had acquired substantial momentum, so its development continued, and it remains the main GNU graphical desktop.

在 Sun 公司的例子里，他们希望能按照自由软件基金会的游戏规则来进行。在 1999 年的 O'Reilly 开源大会上，Sun Microsystem 的共同创始人和首席科学家比尔·乔伊对该公司的“community source”许可证辩护，这个许可证本质是放松了限制，允许用户不与 Sun 签署协议免费复制或修改 Sun 的软件。乔伊发表演讲一年后，Sun 公司的的副总经理马科·博瑞斯再次登上这个演讲台，发布该公司的对于 OpenOffice 使用的新许可证。OpenOffice 是为 GNU/Linux 系统设计的办公套件。

博瑞斯说：“我可以用三个字每来拼写它，GPL。”

在那时候，博瑞斯说 Sun 的决定与斯托曼没有什么直接的关系，而是更多的因为 GPL 软件的蓬勃发展的势头。“现在所发生的是，很多人认识到不同的软件产品与不同的社区，选用什么样的许可证就取决于你希望赢得哪个社区的注意力”，博瑞斯说，“对于 OpenOffice，我们希望能与 GPL 的社区密切合作。”²然而这个斗争的成果也并不完全，因为 OpenOffice 仍然会推荐用户去使用一些非自由的插件。

这样的评论一方面承认了 GPL 在业界的地位，另一方面也承认了在幕后创建这种许可证人的政治天才。伊本·莫格林说：“他一定不是一个律师，不然怎么可能把 GPL 起草成这个样子。”莫格林是哥伦

比亚大学的法学教授，也是自由软件基金会的法律顾问，他接着说：“但是它很有效，它之所以有效是因为理查德·斯托曼的设计哲学。”

作为一个曾经的专业程序员，莫格林追溯他自己 1990 年时为斯托曼提供的无偿专业服务，那时斯托曼在一件私人事件上请求莫格林的法律援助。莫格林后来与加密学专家菲利普·齐默曼合作与联邦政府进行相关的法律斗争，他说他对于这次请求觉得很荣幸。³

“我告诉他说我每天都在使用 Emacs，打这个官司可以赚到不少的律师费来付帐单。”

从那以后，莫格林也许比任何一个人都能更有机会去近距离研究斯托曼的黑客哲学与现实领域的交叉。莫格林说斯托曼对待法律文书和软件源代码的方式基本上是一致的。“作为一个律师，我必须得说，对待一份法律文书的正确方式是找出他里面存在的问题。”莫格林说，“在每一个法律的立法过程中都有很多的不确定性，每一个律师都希望能为他的代理人抓住这些能带来好处的不确定性。理查德的目标是从另一个方面达到相同的目的。他的目标是去除那些不确定性，但这自古以来就被证明是不可能的。要起草一个在任何国家任何情况下在任何法律系统中都适用的许可证是不可能的。但如果你朝着这个目标去努力，你就会走到跟他相同的一条路上。最终结果的简洁设计几乎到达了期望的效果。一个真正的律师很难把你引导到这么好的结果上。”

作为一个负责推动斯托曼运动进程的人，莫格林非常理解这种“也许是”的同盟所带来的挫败感。“理查德是一个从不妥协的人，在他认为是一些核心的根本问题上”，莫格林说，“他从不尝试通过语言上技巧或换用一种更委婉的说法来取悦别人，但这种技能又是在人类社会中所必不可少的。”

.....

.....

.....

“你知道么？”莫格林补充说，“总体上来说，他是对的。”

莫格林这样的评价否定了斯托曼对自己的评价：“我不善于玩游戏，”斯托曼针对一些从未谋面的人认为他是一个精明的战略家的评价，曾这样说：“我不善于看到未来的形势并且参与一般人所会做的那些事情。我的方法是永远关注事情的根基，也就是说‘让我们一起尽可能的打好基础吧’。”

GPL 变得越来越有名，并且继续保持它的吸引力，这是对斯托曼和他的 GNU 同事所构建的基础架构的最好礼物。斯托曼已经不能再标榜自己为“最后一个真正的黑客”，但他可以通过建立自由软件运动的伦理框架来树立他的新的形象。其它现代的程序员在这个框架中工作是否舒服并不是最重要的事情，最重要的是他们已经选择了使用斯托曼最伟大的遗产这一事实。

在这个时候来讨论斯托曼的遗产似乎有点太早。在写作本书时，斯托曼只有 48 岁，还有不少的时间可以从这些“遗产”中增加或减少一些东西。并且，自由软件自身运行的自动驾驶的本质，也让超脱于天天都在进行的软件工业的斗争分析斯托曼的一生变得更有吸引力，我们应该在一种更为严肃的历史的背景中来分析他。

为了维持他的形象，斯托曼拒绝一切投机的机会。“对于那些无法预知未来会怎样的东西，我没有办法做出相应的计划，”斯托曼说，用他的那早来的墓志铭来总结便是：“我说过，‘我要去斗争，没有人知道我会到达哪里’。”

在斗争中，斯托曼协调睦和了那些原本可能会成为他最大竞争对手的人们。这也是他直率、讲究伦理性格的一个体现。在必要的时候，他以前的一些政治上的对手仍然愿意帮他说一些好话。斯托曼的理论与斯托的黑客天才很容易让一名传记作者感到疑惑：如果有一天斯托曼由于某些原因丢失了他的强烈的个性，人们会如何去看待他？

在本书的早期草稿中，我把这个问题称为“100 年”问题。期望用这个词来刺激斯托曼的目标和他的工作，我找了几位软件工业的杰出人物，让他们把自己置身当前的时间外，把他们自己放到一个历史学家的角度，从 100 年以后往回看自由软件运动。从当前有利的观点来

看，不难发现斯托曼与以前的美国人之间的相似之处，他们在有生之年所做的一小点东西，在历史上就能影响整整的一代人。比如，与亨利·戴维·梭罗相比，他是一个先验论的哲学家，曾经写出《公民不服从》一文；还有约翰·缪尔，塞拉俱乐部的创始人，也是现代环保运动的先驱；类似的人还有威廉·詹宁斯·布莱恩，他有着“伟大的平民”的称号，他是平民主义运动的领导人，垄断部门的敌人，他虽然很强，但是他的光辉已经在历史的长河中渐渐的褪去。

斯托曼并不是第一个软件看成是公有财产的人，但是因为他创造了 GPL，让他必定会在历史书籍中留下一笔。基于这样事实，看上去很值得跨出当年的时间去审视理查德·斯托曼的遗产。假设到了 2102 年，软件开发者们还会继续在使用 GPL 吗？还是它早已经被抛弃到了路边？“自由软件”这个词会像“自由白银”一词在今天一样的古朴还是会体现出对未来政治事件的神奇预测？

..... 试图预测未来是一件很冒险的事情，但是大部分人在遇到问题时，总是会迫切的想去解决它。

莫格林说：“一百年以后，理查德和其他一些人也许不仅仅是能在历史中留下一笔，他们会成为一个时代的主角。”

莫格林所说的应该出现在未来教科书中的“其他一些人”包括约翰·吉尔摩，他是斯托曼 GPL 的顾问，也是后来电子前哨基金会的创始人。还有西奥多·霍尔姆·尼尔森，也叫泰德·尼尔森，是 1982 年出版的《文学机器》一书的作者。莫格林说，斯托曼、尼尔森和吉尔摩都在历史上以他们各自特有方式做出了巨大的贡献。尼尔森创造了“超文本”的概念，解决了在数字时代标定信息所有权的方式。吉尔摩和斯托曼则是指出了信息控制所带来了负面的政治效果，并且他们还创建了一些组织，包括吉尔摩所创建的电子前哨基金会和斯托曼的自由软件基金会，这些组织都致力于解决这些不好的效果。在他们两个之间，莫格林觉得斯托曼的活动更为个人化，从本质上来说没有那么政治化。

“理查德是一个独一无二的人，在很早的时候，他就认清了非自由软件在伦理上的一些不好之处。”莫格林说，“这与斯托曼本人的个性非常相关，这种个性引起很多人研究的兴趣，尝试把它描写成是一种附带现象或者甚至是理查德·斯托曼个人生活中的一个败笔。”

吉尔摩对自己能够同时与古怪的尼尔森和暴躁的斯托曼打交道这件事认为是一种“混杂的荣幸”，他很赞成莫格林的观点，写道：

我的猜测是，斯托曼的作品会像托马斯·杰斐逊的一样伟大：他是一个思路清晰的作家，并且对他的原则非常清楚……至于理查德能否像杰斐逊一样具有影响力，主要取决于我们今天所说的“公民权利”这个抽象的概念在一百年以后是不是比今天的“软件”或“技术限制”更为重要。

另一项被人们所忽视的斯托曼的遗产，吉尔摩写道，是由 GNU 工程所领导的合作式的软件开发模式。这种模式虽然有时也存在一些缺陷，但还是成为了软件开发工业中的一项标准。吉尔摩说，这种合作式的软件开发模式事实上比 GNU 工程、GPL 许可证或任何其它一个由斯托曼开发的软件本身更具影响力：

在 Internet 出现以前，很难实现异地的软件合作开发，即使是一个互相了解和信任的团队也是如此。理查德引领了软件合作开发的潮流，而且还是未曾谋面的无组织的开发志愿者们。为了实现这个目标，理查德没有创建什么基础工具（比如 TCP 协议、电子邮件列表、diff 和 patch 工具，tar 文件格式，RCS 或 CVS 或 remote-CVS），但他选择了那些现有的可以帮助程序员群体开展有效合作的工具。

.....

2001 出版的《思想的未来》一书的作者，斯坦福大学法学教授劳伦斯·莱斯格同样表达了乐观的观点。与其它法律界的学者一样，莱斯格把 GPL 看成是现在所谓“数字大众”的主要保障。通过把很多社

区开发的软件、网络和电信标准聚集起来，在过去的 30 年中，带来了 Internet 的指数级的增长势头。与其它的 Internet 先驱相比，比如万尼瓦尔·布什、文顿·瑟夫和约瑟夫·利克莱德这些让大众看到计算机技术的发展人，莱斯格认为斯托曼的影响力更为个人化和内省式的，并且是独一无二的：

[斯托曼] 把争论变为了现实。他让人们看到危机，并且他创造了一种方式让这些理想得以延续……也就是说，我并不能确定应该怎么把他与瑟夫或利克莱德去比较。他们的创新之处截然不同。这并不是某一段代码或者是让 Internet 活跃起来。这更多的是让人们去认识 Internet 的某种特殊的价值。我觉得这样的人前无古人，后无来者。

当然，并不是所有人对斯托曼的看法都是一成不变的。埃里克·雷蒙德作为开源的拥护者，认为斯托曼的领导角色从 1996 年开始就弱化下去了，如果从 2102 年的水晶球去看有很多混合的信号：

我认为斯托曼的作品（GPL、Emacs 和 GCC）会被认为是革命性的作品，是信息世界的奠基石。我觉得历史不会像斯托曼理论所设想的那样，也不会像他个人所倡导的领地化和邪教领袖行为那样的。

作为斯托曼本人，他也看到各种混合的信号：

20 年后，历史对 GNU 工程的评价，会取决于谁赢得了自由的斗争，正确的去使用公共的知识。如果我们失败了，我们就会成为沧海一粟。如果我们成功了，人们也许也不能正确的认识到 GNU 操作系统的角色，如果大家误以为这个系统叫“Linux”的话，那他们其实是搞错了情况，并且没有理解形成这一切的原因。

但是，即使我们赢了，一百年以后人们所能了解到的历史，也更多的依据于政治上占有主导地位的一方的观点。

斯托曼找到了与他相似的历史人物：19 世纪的约翰·布朗。他是一名激进的废奴主义者，他在梅森-迪克森线的一边被看成是一个英雄；而在另一边，则被看成是一个疯子。

约翰·布朗的奴隶起义最终没有真正开始，但是，他的行为唤醒了人们对于废奴的要求。在南北战争中，约翰·布朗是一个英雄：100 年后，尤其在 20 世纪，历史教科书都把他描绘成一个疯子。在法律隔离的时期，当顽固成为一种无耻的行为，美国开始部分接受南部希望摆脱奴役的现实，历史书上对于南北战争和相关的事件的绘述都显得颇为偏袒一方。

这样的比较，记录了斯托曼工作的自我认识的理快性格特点，也记录了他当前声誉的两面性。尽管我们没有看到斯托曼的声誉降到过去重建时声名狼藉的布朗那么低，但是除了他难得的像打仗一样的类比，他所做的一切很难激发暴力，不难想象，即使在未来斯托曼的观点变得烟消云散时也一样。让自由软件这个词汇变得流行，不像是一个群体运动，倒像是一系列与私有化诱惑的个人战斗。斯托曼看上去处在一个不利的地位，尤其是在面对那些顽固的对手们时。⁴

再一次的，总有一种美好的愿望，那就是有一天历史会证明斯托曼留下的是一笔伟大的遗产。莫格林在过去的十年中密切关注着事态的变化，他提醒那些误解了斯托曼个性的人，以为他是生活的假象的副现象的人。莫格林说，如果没有这样的个性，就没有这么多可以讨论的作品。作为一名曾经的最高法院职员，莫格林说：

我曾经服务过的最伟大的人是瑟古德·马歇尔。我知道是什么让他成为了一个伟大的人。我明白为什么他可以用他的方式去改变世界。我不是在把他们两个作比较，因为他们两个并没有什么共同点。瑟古德·马歇尔是一个社会中的人，他表着一个被驱逐的社会和那个驱逐他的社会，但他仍然是一个社会中的人。他的能力是社交能力。但他也只是一个人。.....Different as they were in every other respect, that the person I most now compare

him to in that sense, all of a piece, compact, made of the substance that makes stars, all the way through, is 斯托曼.

为了把斯托曼的这种形象描述清楚,莫格林回忆起 2000 年春天的一幕。VA Linux IPO 成功的故事仍在商业媒体中产生共鸣,差不多有半打的自由软件相关的内容在这些新闻中被提及。被暴风雨般的文章和评论所包围,莫格林回忆起与斯托曼一起共进午餐的时候就像是被台风眼中抛弃的遇难者。在接下来的几个小时中,他说,他们的对话渐渐的转向一个主题:加强 GPL 的位置。

“我们坐在哪里讨论对于东欧出现的一些问题我们应如何去解决,以及当内容的所有权问题成为威胁自由软件的一个问题时我们应怎么做。”莫格林回忆说,“在我们讨论的时候,我简单的想了想对于旁边路过行人来说,我们是什么样的一个形象。这就是我们,两个长着小胡子的无政府主义者,设计着下一步的行动。并且,理查德正在清理他头发里的小碎屑,并把它们若无其事的掉进了汤里。任何听到我们谈话内容的人都会觉得我们是疯子,但是我知道:我知道革命就将从我们这张桌子开始。这就是让革命开始的地方。这个人就是发起这场革命的领导者。”

莫格林说在那一刻,比其它任何时候,都准确的描述了斯托曼风格的简洁性。

“这是件很有趣的事情,”莫格林回忆道,“我对他说,‘理查德,你得明白,你和我两个人在这次革命中一分钱也没赚到。’然后我还得给你付午餐钱,因为他没有钱来付。”⁵

Endnotes

¹Sun 公司为了遵守商标相关的规定,使用了一个很拙劣的名字,叫“OpenOffice.org”

²参考作者对马科·博瑞斯的采访(2000 年 7 月)。

³有关齐默曼的官司,可以参考史蒂芬·李维的《加密》一书,第 287 至 288 页。在本书印刷版的第一版中,把这次官司写成是莫格林帮助齐默曼与国家安全局(NSA)打

官司。跟据李维提供的信息，齐默曼是被美国检察长办公室和美国海关调查，而不是NSA。

⁴

⁵RMS:

萨姆·威廉姆斯跋语：粉碎孤独

[RMS：由于这一章节的内容主要是萨姆·威廉姆斯的一些个人观点，我对正文的修改都将以方括号或省略号进行标记。我所做的改动主要是为了澄清一些技术或法律上的问题，同时，我删除了一部分具有攻击性或是缺少信息量的段落。我也加了一些以“RMS:”打头标记的注释，用于回应文中的一些观点。在这一版中，威廉姆斯本人也进行了一些修改，他的改动没有明显的标识出来。]

为一个还没有去世的人撰写传记有点像是在写一个话剧。台前的话剧与幕后的故事相比，总是显得苍白无力。

在《马尔科姆·艾克斯传记》一书中，亚历克斯·哈里为广大读者展示了一部幕后的话剧。在这本书的后记中，哈里不再以代笔人的身份出现，而是表达了他自己的心声。在后记中，哈里讲述了一名被伊斯兰民族组织发言人当作“工具”和“间谍”的自由记者是如何突破个人和政治壁垒最终出版马尔科姆·艾克斯传记的故事。

我并不想把这本书与《马尔科姆·艾克斯自传》做什么比较，不过我还欠哈里一个人情债，没有他的后记作榜样，我不会写出这么一部坦诚的作品。在过去的一年中，这个后记被我当成解决一个从来不愿意与他人合作的传记人物的问题指导手册。[RMS：我在自己的职业

生涯中确实习惯于对一些别人不假思索就接受的事物说“不”，所以我有时看上去也确实是一个不太愿意与他人合作，但这并不是我做事的动机。] 我从一开始就想为这本传记写一篇类似的后记，一方面是向哈里致敬，另一方面也是想让读者了解这本书是如何写成的。

这个故事背后的故事开始于奥克兰的一个公寓，就像书中所记载的一样，故事在硅谷、茂宜岛、波士顿和剑桥展开。最后，成了一部现实版的双城记：纽约——全球出版界的中心和加州的塞瓦斯托波尔——索诺马县的出版中心。

故事开始于 2000 年 4 月，那时候我正在为命运多舛的 BeOpen 网站 (<http://www.beopen.com>) 写一些文章。其中的第一项任务就是与理查德·马修·斯托曼进行一次电话采访。这次采访进行得很顺利，Slashdot (<http://www.slashdot.org>) 网站在他的每日精彩文章中给了这篇文章一个链接。Slashdot 是 VA 软件有限公司（在此之前名为 VA Linux 系统公司，更早的时候叫 VA 研究所）的一个热门的 Herd 新闻聚合网站。短短几小时内，BeOpen 网站的访问量激增，读者们纷拥而来。

如果只是按预期的设想，故事到这里就该结束了。采访结束后 3 个月，当我在加州的蒙特雷参加 O'Reilly 的开源软件大会时，我收到一封来自特蕾西·帕蒂森的电子邮件，她是纽约一家大出版社的涉外版权经理。

发送至：sam@BeOpen.com

主题：RMS 采访

日期：2000 年 7 月 10 日，星期一，15:56:37 -0400

亲爱的威廉姆斯先生，

我在 BeOpen 上读到了您对理查德·斯托曼先生所进行的访谈内容，它非常有趣。很久以来，我对 RMS 和他的作品都很有兴趣，看到您的作品感觉到很高兴，因为我觉得你

的作品非常出色，它准确的表述了斯托曼从事 GNU-Linux 和自由软件基金会相关工作时所表达出来的一些精神。

我来信的目的希望能读到更多这方面的文章，我相信我不是唯一一个提出这样要求的人。你那里还有更多的信息或消息源可以把你的访谈作进一步的展开，并把它写成是斯托曼的一个传记吗？也许可以在里面添加更多有关他个性的花絮和背景资料，这样的作品也许能让非程序员群体产生更大的兴趣。

特蕾西让我打电话给她进一步讨论这个主意，我照做了。特蕾西告诉我，她的公司在准备出版一系列在线的电子书，这个系列中希望能有一些有趣的内容，以便吸引第一批读者。这个系列的电子书需要 30000 词左右的内容，大约 100 页，并且她已经说服她的老板出一本黑客群体中的主要人物的传记。她的老板很赞同这个建议并且已经开始着手物色有故事的人物做为选题，她正好看到了我在 BeOpen 上发表的斯托曼访谈，所以就发邮件给我了。

特蕾西是这样问我的：你愿意把这篇访谈扩展成为一本完整的人物传记吗？

我立即就做出了答复：没问题。在签署合同之前，特蕾西建议我先准备一个故事的大纲，让她先给她的上级看一下。两天后，我把润色后的大纲发给了她。一周后，特蕾西回邮件通知我，她的老板开了绿灯。

我必须承认，刚开始的时候我并没想到要让斯托曼直接参与到这样一个电子书的项目中来。作为一个记录开源斗争的记者，我知道斯托曼是一个固执的人。我至少已经收到了半打他给我的邮件，指责我不应该使用“Linux”一词，而应该用“GNU/Linux”。

除此之外，我也很清楚斯托曼一直是期望能通过一切的渠道向公众表达他的观点。如果我从这个角度来向他介绍我们的出版计划，也许他会更容易接受一些。如果不这样做的话，我就只能依赖网上的各种文档、访谈和对话录音，最终写出一本并不权威的传记。

在我的研究过程中，我曾经看到一篇斯托曼在 2000 年 6 月时所写的名为“自由还是版权？”的短文，发表在麻省理工学院的《技术观察》杂志上，这篇短文把电子书描述成是各种软件罪恶的混合物。斯托曼认为，电子书的罪孽不但在于读者们需要使用一些私有的软件才能去阅读，而且电子书中用来保护作品内容不被非法复制的技术也非常的粗暴。读者不能下载一个可以自由传播的 HTML 或 PDF 文件，而是得下载一个被加密过的文件。本质上来说，购买电子书意味着购买了一把无法转让的钥匙，这把钥匙用于开启加密内容。任何试图不用授权的密钥解密电子书内容的行为，都构成违反《数字千年版权法》的行为，这部 1998 年制定的法律用于加强 Internet 上的版权控制。读者把加密的图书转换成开放的文件格式也同样构成犯罪，即使他们这么做的目的只是为了在自己家里的另一台电脑上阅读这本电子书。与拥有一本传统的纸质书不同，读者不可以出借、复制或转卖电子书，读者所拥有的所有权利，就只是在一台授权过的设备上阅读这本电子书，斯托曼警告说：

如果我们阅读纸质书籍，则我们还能享有一些最基本的自由。但是，如果电子书取代了印刷的纸书，则没有什么好处。通过使用“电子墨水”，人们可以在已经印有内容的“纸”上重新“印”上其它内容，即使是报纸，也会变得没有实体，转瞬即逝。想象一下：不会再有旧书店；不能把书借给你的朋友；不能从公共图书馆借阅图书。不再让人有机会不付钱就阅读的“漏洞”。（如果从微软公司阅读器的广告的内容来看，还不再允许匿名购买书籍。）这就是那些出版社想让我们面临的问题。¹

不容质疑，这篇文章引起了一些担忧。特蕾西和我都没有讨论过有关她的公司所使用的软件或是电子书的版权 [许可证] 协议会是什么样。我向特蕾西谈起过这篇《技术观察》上的文章，并请她提供一些

有关她所在公司的电子书政策的信息给我。特蕾西答应一有消息就通知我。

我很希望工作可以得到尽快的开展，所以我决定先打电话给斯托曼并跟他先大致聊一下有关这本书的计划。在电话中，斯托曼很明确的表达了他对这本书的兴趣和担心。“你读过我那篇有关电子书的文章吗？”他问。

我告诉他我的确读过那篇文章，并且我正在等候出版社对于此事的回应。斯托曼给出两个条件：他不愿意对他反对的电子书许可证条款做出任何妥协，也不愿意被别人看成是他做出了妥协。“我不想参与任何让我看起来像个伪君子的事情。”他说。

对斯托曼来说，相比软件的问题来而言，版权是个更大的问题。他说他可以不考虑出版社或相关的第三方公司使用什么软件来发布电子书，只要公司可以在版权声明中说明，允许读者自由的发布电子书的内容。斯托曼说，可以参考斯蒂芬·金的《植物》所使用的出版模式。2000年6月，金在他自己的网站上发布声明：他自己将以连载的形式出版《植物》一书，这本书定价为13美元，每一个章节是1美元。如果有75%的读者为前一章付费，他就会发布一个新的章节。截止8月份，这个计划看上去实施得很不错，金已经发布了前两个章节，第三章也即将发布。

“如果这本书允许被自由的原样复制的话，我就可以接受。”斯托曼说。[RMS：我记得我当时还提出了有关加密的问题，后面的段落中会谈到这个问题。如果这本书必须要用一个非自由软件才能阅读的话，我是不会答应的。]

我把这个信息传达给了特蕾西，并对她和我可以有办法商讨出一个公平的协议，抱有很大的信心。我打电话给斯托曼预约这本书的第一次采访。斯托曼同意接受采访，而且没有再次询问协商的进展。在第一次采访后不久，我又趁热打铁的安排了第二次采访（这次是在津汇），时间是赶在他出发去塔希提岛进行为期14天的度假之前。[RMS：这不是一次纯粹度假，我在那里也发表了一次演讲。]

正当斯托曼在度假时，从特蕾西那里传来了坏消息。她们公司的法务部门不希望改变这本电子书的 [许可证] 声明。如果读者想要把这本电子书转移给别人，就必须 [先对电子书进行破解，然后才能把它转换成其它的开放格式，比如 HTML。这样的做法是违法的，并且会面临诉讼。]

手里拿着两次新鲜出炉的采访内容，我不希望最终出版的书籍中无法包含这些内容。我很快决定动身去纽约，与我的代理人和特蕾西见面，商量一个折中的方案。

飞抵纽约后，我见到了我的代理人，亨宁·格特曼。这是我们的第一次面对面的会面，看起来亨宁对于找到一个妥协的方案并不乐观，至少从出版社的角度来看是这样。这些历史悠久的大出版社本来就对电子书的出现充满疑惑，所以他们并不情愿对于可能让读者更容易逃避付费的版权条款进行改变。虽然亨宁是一个关注于技术图书的代理人，他仍对我所处的戏剧般的窘境感到好奇。我告诉他有关我已经进行的两次采访的内容，并且告诉他我已经承诺斯托曼我们不会以让他看起来是个“伪君子”的方式去出版本书。亨宁理解我被伦理所绑架的困境，建议我把这一点作为谈判的关键点。

亨宁说，除此之外，我们还可以采用胡萝卜加大棒的策略。所谓的胡萝卜就是出版一本向黑客社区内在的伦理致敬的电子书，就会带来不小的公众影响力。而所谓大棒就是出版一本并不能真正弘扬黑客文化的电子书。在德米特里·斯柯里亚诺夫的官司在 Internet 的引起轰动 (*cause célèbre*) 前九个月，我们就已经预料到程序员找到破解电子书的方法只是个时间问题。我们能想象，如果出版社出版一本内容是有关理查德·M·斯托曼的加密的电子书，那无异在封面上用大字标明“请破解这本电子书”。

见过亨宁后，我打了个电话给斯托曼。为了能让我们的“胡萝卜”显得更具诱惑力，我向他介绍了一些我们可能会使用的妥协方案。比如，以 [双重] 许可证的方式来发布书籍的内容的出版方案，就像太阳计算机系统公司 (Sun Microsystems) 对 OpenOffice 这个自由桌面办

公套件所做的那样。这样的话，出版社就可以以正常的电子书出版方式出版一个 DRM 加密的²电子书，这样就可以应用到那些电子书软件的好处，同时再以 HTML 格式发布一个可以自由复制的版本。

斯托曼告诉我，他不反对双重 [许可证] 的方式，但他不同意自由版本比受限制的版本地位要更低一等。另外，他说，[他仔细想了想，OpenOffice 的例子不适用现在的情况，因为但对于这本书，] 他完全有能力控制最终的产出，他有权绝拒合作。

[RMS: 问题在于，如果我同意出版一个受限的版本，这会不会是一个错误的决定。我可以支持自由版本的 Sun OpenOffice，因为它是一个自由软件，并且有比没有要好，同时我也可以拒绝使用非自由的版本。在这个问题上，我并没有自相矛盾，因为 Sun 公司发布非自由的版本并不需要征求我的意见，我无须为这个版本的存在负责。但在这个问题上，如果我同意出版一个不允许自由复制的版本，那这个责任就都在我身上。]

我的其它一些建议也没有什么实质性的效果，唯一可以得到斯托曼认可的是他可以做出一些让步 [RMS: 也就是说，又妥协了一些]，允许电子书的 [许可证] 限制改为“允许除了商业使用外的各种形式的分享”。

在我挂断电话前，斯托曼建议我告诉出版社说我已经答应了斯托曼这部作品必须是 [可以自由分享的]。我告诉斯托曼我不能答应这个说法 [RMS: 确实是这样，因为他从一开始就已经接受了我的条件]，但我很清楚如果没有他的合作我是无法完成这本书的。斯托曼看上去很满意，以他惯有的结尾方式说了声“开心黑客”，然后挂断了电话。

次日，我和亨宁见到了特蕾西。特蕾西说她的公司同意以不加密的方式发布本书的一些摘要，但摘要的长度需要限制在 500 个词以内。亨宁告诉她，这样的让步不足以把我对斯托曼的伦理义务中解救出来。特蕾西说，她的公司对于一些像 Amazon.com 这样的在线商家也有一些合同上的限制。即使出版社同意把这本书的电子版开放出去，这也还面临着可能被其它合作伙伴的诉讼。是想要出版社管理层的妥协

还是让斯托曼妥协，这个决定权在我。我可以违反先前与斯托曼的约定，继续使用这些采访素材，也可以恪守新闻工作者的职业操守，对出版这本书的口头承诺食言。

会议后，我和代理人转移到了第三大街上的一个酒吧。我用他的手机给斯托曼打了个电话，不过没人接电话，所以只能给他留了个言。亨宁离开了一会儿，给我一些时间来思考。当他回来时，他拿着电话，在我面前晃了晃：

“是斯托曼打来的。”亨宁说。

这次对话从一开始就很不愉快。我把特蕾西所说的有关出版社的合同限制问题转告给斯托曼。

斯托曼很粗鲁的回复道：“那么，为什么我需要因为他们的合同限制而作出妥协？”

我说，因为让一个大出版公司为了一本 30000 多字的电子书去冒一个卷入法律诉讼的风险，是一个非常离谱的要求。[RMS：他的假设是我不可能因为一点小小的原则去拒绝这个机会。]

“难道你还没有明白吗？”斯托曼说，“这正是我要坚持我的立场的原因，因为我需要发出一个胜利的信号，我要他们像以前一样，在自由和商业之间作出选择。”

“胜利的信号”这个词回响在我的脑海中，我感觉到自己的思维一瞬间游走到了路边的人行道上，人行道上满是匆匆而过的行人。来到约定的酒吧，我很高兴的发现，这个地方距离那首 1976 年拉莫内斯的歌曲“第 53 和第 3”所纪念的街角只有半个街区的距离，当我还是一个音乐人时，我非常喜欢演奏这个曲子。就像歌曲中所描绘的那个一直受挫折的家伙，当这些事情放到一起，我感觉到一点崩溃的意味。这种反讽是非常明显的。经历了几周高兴的记录别人的不幸的日子，我发现自己正处于一个罕见的胜利边缘：让理查德·斯托曼做出妥协。当我还在犹豫不决，是不是要恳求出版社改变立场，并透露出我的对他们的同情时，斯托曼就像一头闻到血腥味的野兽，对我发起了攻击。

“这是怎么回事？你是在要我吗？你是想按他们的意志去办吗？”斯托曼咄咄逼人的问道。

[RMS：从这句话中可以看出威廉姆斯没有正确的理解这次对话的内容。他把我比作是一只食肉动物，但事实上我只是拒绝了他试图让我接受的条件。正如前文所述，我已经作出了不小的让步，我只不过是拒绝对于原则性的问题作出让步。我经常这样做。一些对此不满的人就说我“完全不接受让步”，这完全就是不理解所形成的夸张了；有关这个问题，可以参考我的文章：<http://www.gnu.org/philosophy/compromise.html>。那时候，我所担心的是他完全不顾我们先前达成的二人协议，不管我的反对出版一本通过 DRM 保护的图书。我所嗅到的不是“血腥味”，而是存在背叛的可能性。]

我再次提出有关双重版权的方案。

“你所说的版权应该是指许可证。”斯托曼冷冷的说。

“哦，是的，是许可证，或者版权，差不多吧。”我说。我像是水里一条受了伤的金枪鱼。

“你他妈的就不能按我所交待你的去做吗？”他咆哮道。[RMS：我觉得这句话可能会引起误解，首先把“他妈的”当成副词来使用绝不是我的说话风格，其次，这句话也不太符合当时的语境。这句话听起来像是在责备一名不听话的下属。我觉得他有一种伦理上的义务，但是他不是我的下属，所以我也不会用这样的语气说话。通过笔记而不是录音来记录对话，他没法保证记录下准确的内容。]

我应该是一直站在出版社的立场上跟他争辩到底了，因为我的记事本上记着这次对话中斯托曼的最后那段老生长谈的话：“我不关心这些。他们在做的就是一些邪恶的事情。我不能支持作恶！再见。”[RMS：这样说听起来像是我总结说他不能接受一个否定的答复，但是想不接受他的建议而结束这次对话的唯一办法就是挂断电话。]

当我放下电话，我的代理人递给我一杯刚倒的健力士啤酒。“我想你现在需要这个。”他笑着说，“我看到你打电话时混身在发抖。”

我确实在发抖，直到半杯健力士啤酒下了肚，才慢慢平静下来。我有一种奇怪的感觉，似乎自己变成了“邪恶”的代言人。三个月前，当然坐在奥克兰的公寓中构思我的第一部作品时，也有过类似但更强烈的感觉。现在，我正坐在一个我只从摇滚歌曲中接触到过的世界里，与出版社的管理层开会，与我一天前才见到的代理人喝酒。这一切确实太超现实了，就像是自己的生活出现在了电影里一样。

正当这个时候，我内心的荒谬因子突然爆发了。最初的震动让我流露出一丝笑意。我的代理人一定认为我像是一个脆弱的作家，正在经历一次前所未有的情感崩溃。而对于我自己来说，我只是开始认识到自己的境遇是如此的讥诮。无论最终能不能成交，我已经完成了一个漂亮的故事。只不过是我要找一个地方来讲述这个故事。我渐渐收起了笑容，干了杯中的酒。

“欢迎来到斗争的前线，我的朋友。”我与代理人一边碰杯一边说，“并且享受斗争吧。”

如果这个故事真的是一个戏剧的话，这里就是应该加入一个短暂、浪漫的插曲的地方了。我们的会议很紧张，结果也令人感到非常沮丧，于是特蕾西邀请我和亨宁与她一起去找她的一些同事们出去喝酒。我们离开了第三大街上的酒吧，前往东村，见到了特蕾西和她的朋友们。

一到那里，我就跟特蕾西展开了交谈，并且小心的避免谈论工作。我们的交谈非常愉快，也很放松。在告别时，我们约定次日晚上再见。这次交谈确实非常的愉快，以至于斯托曼的电子书的问题都快消失在记忆中了。

回到奥克兰，我召集了一些新闻出版界的朋友和熟人，向他们讲述了我所遇到的困境。他们中的很多人都责备我在采访斯托曼前的谈判中放弃了太多的原则。一个新闻学院的老教授建议我忘掉斯托曼所谓“两面派”的说法，把故事完整的写下来。很多了解斯托曼与对媒体打交道的精明能力的记者表达了对我的同情，但他们一致说：这是你自己的事。

我决定把这本书出版的事先放到一边。因为就拿最重要的采访一事而言，我其实也并没有太多的进展。除此以外，这让我有了不先通过亨宁而直接与特蕾西沟通的机会。到了圣诞节，我们扯平了相互访问的次数，她飞到西海岸来了一次，我也第二次飞去了纽约。新年前夜，我向她求婚了。我决定选择纽约作为我们定居的地方。二月分，我带着我的笔记本电脑和所有有关斯托曼传记的研究笔记，和未婚妻一起飞去了肯尼迪机场。我和特蕾西在五月十一日结婚了。这就是有关这本书的不成功的交易的最美妙结果。

整个夏天，我开始考虑把我的采访笔记改编成杂志的文章。从伦理上来说，我觉得这样做是合理的，因为最初的采访条款上没有规定采访内容的适用范围。实话说，隔了八个月的时间，我觉得为斯托曼写传记变得舒服了一些。9月份我们通过电话以后，我只收到过斯托曼的两封电子邮件。两封邮件的内容都是谴责我在 *Upside Today* 这个网络杂志的文章中使用“Linux”一词而不是“GNU/Linux”。除此之外，我很享受这份安静。六月，在纽约大学的演讲后一周，我抽空写了一篇有关斯托曼的 5000 字的杂志稿件。这一次，文字变得非常的流畅。我想，是我跟斯托曼之间的距离感帮助了我，它帮助我恢复了曾经失去的情感观点。

七月，那是在收到特蕾西的第一封电子邮件后的整整一年，我收到了来自亨宁的一个电话。他告诉我，位于加州塞瓦斯托波尔的 O'Reilly & Associates 出版集团对于出版斯托曼的传记很有兴趣。这个消息让我很开心。在全世界的所有出版社中，O'Reilly 看上去是对于造成先前胎死腹中的图书的原因最为知晓的公司，他同时也是为埃里克·雷蒙德出版《教堂与集市》一书的公司。作为一个记者，我以前也重度依赖 O'Reilly 出版的开源一书来作为历史的参考。我也知道有不少书的章节，包括斯托曼曾经写过的一章，都是以允许随意分布的 [许可证] 条款来出版的。如果电子出版的问题再次出现，那这些信息就会很有用处。

果不其然,电子出版还是遇到了一些困难。亨宁告诉我, O'Reilly 打开把这本传记同时以纸质书和它的新的 Safari Tech Books Online 订阅服务的形式出版。亨宁提醒我, Safari 的用户许可证中有一项特殊的限制³, 但是 O'Reilly 同意允许让用户在任意媒界上复制并分享这本书的文字内容。所以, 作为作者, 我需要在两个许可证之间做出选择: Open Publication License 或 GNU 自由文档许可证。

我研究了这两种许可证的内容和背景。Open Publication License (OPL)⁴ 允许读者以整体或部分的方式复制和重新分发相关作品, 无论是在“物理的媒介还是电子的”, 只需要保证继续保持 OPL 许可证即可。在满足这两个条件的前提下, 它也允许对作品进行修改。同时, OPL 也包含一些选项, 可以由作者来选用, 可以限制读者在没有得到作者允许前不得对作品进行“实质性的修改”或以书籍的形式出版。

GNU 自由文档许可证 (GFDL) 则允许在任何媒介上复制和分发文档, 只要求保留相同的许可证。⁵ 它也允许在一些条件下对文档进行修改。与 OPL 不同的是, 它没有给作者禁止他人修改的权力。它也不给作者拒绝修改内容以避免他人出版一本类似的竞争作品的权力。然而, 除了版权所有人以外的他人如果要出版 100 份以上的内容, 则需要在封面和封底上提供一些必要的信息。

在研究这些许可证的时候, 我也在 GNU 工程的官方网站上的”各种许可证与对此的说明“⁶ 文章页面上找到了一些斯托曼对于 Open Publication License 的批评。斯托曼对它的批评主要关注在有关创建修改后的版本以及作者有权选择拒绝修改这两点上。斯托曼认为, 如果作者并不打算行使这两项限制权力, 那他完全可以改用 GFDL, 这样可以避免在修改后版本中突然被加上相关限制条件的风险。

修改行为在两种许可证中的重要性都反映了它们的设计初衷, 那就是允许软件手册的读者有机会去改进这些手册并把它们贡献到社区中去。由于我的书并不是一本手册, 所以我对于两种许可证中对于修改的条款都有一些顾虑。我只是希望能够让我的读者有机会去交换图书的副本或是复制图书的内容, 如果他们买了纸质的图书, 也可以享

有相同的自由。我感觉这两种许可证都可以满足我的需求，所以我就跟 O'Reilly 签了相关的合同。

但是，有关可以随意修改的条款还是引发了我的好奇心。在我与特蕾西的早期谈判中，我曾经谈过对电子书使用 GPL 风格的许可证好处。我说，再不济的情况下，这样的许可证也可以对这本电子书的宣传起到正面的作用。在最好的情况下，这可以鼓励读者参与到图书的写作过程中来。作为作者，我非常希望其他读者能帮我一起改进我的作品，而我也可以坐享其成。另外，看着一本书不断的演化也是一件很有趣的事情。我把未来的版本描绘成类似于《塔木德》的在线版本，我原始的文字在中间，两边的空白由第三方的评论来装饰。

我的灵感来自于 Xanadu 项目 (<http://www.xanadu.com/>)，这是泰德·尼尔森在 1960 年构想的传奇般的软件概念。在 1999 年的 O'Reilly 开源大会上，我看到了这个项目的开源版本 Udanax，它的演示效果很让人震惊。在一个演示中，Udanax 把一篇父文档和它的衍生作品分成两栏纯文本进行展示。只要点击一下按钮，程序就可以把原文档和衍生作品中的每一个相关的句子联系起来。理查德·M·斯托曼传记的电子版不需要真正的用 Udanax 实现，但是既然我们在技术上有这样的可能性，那为什么不让读者能有一个参与的机会呢？⁷

当 O'Reilly 的编辑劳里·帕瑞其给我在 OPL 和 GFDL 中选择一个的机会时，我又一次萌发了这样的奇思妙想。到了 2001 年 9 月，也就是我签合同的时候，电子书几乎已经成了一个无人关心的话题。包括特蕾西所在的出版社在内的大部分出版社，都已经因为读者缺乏兴趣而停止了标名电子书出版印记。我开始怀疑，如果这些出版社都不把电子书看成是一种出版形式而只是看成是一种社区建设，那为什么这些出版印记还需要存在？

签完合同后，我通知斯托曼我们的图书项目可以继续下去了。我提到了 O'Reilly 在许可证方面给我们的选择，可以选择 OPL 或者 GFDL。我告诉他我倾向于使用 OPL，其原因只是因为我觉得没有理由要给 O'Reilly 的竞争者机会去只是换个封面就出版一本内容一样的

新书。斯托曼回信说建议使用 GFDL，因为 O'Reilly 以前也多次使用这个许可证。不管去年所发生过的那些事情，我建议他跟我做个交易。如果斯托曼能给我更多采访的机会，并且促成本书在 O'Reilly 出版，我就选用 GFDL 许可证。斯托曼同意接受更多的采访，但是对于出版相关的事宜的支持程度会取决于这本书的内容。这看上去很公平，于是我约定在 2001 年 12 月 7 日在剑桥对斯托曼再进行一次采访。

这次采访的时间与我的妻子特蕾西去波士顿出差的行程正好一致。在出发前两天，特蕾西建议我邀请斯托曼一起吃个饭。

她说：“不管怎么说，他是促成我们联姻的那个人。”

我给斯托曼写了一封电子邮件，他很快回复接受这次邀请。次日，我开车去波士顿，在酒店接上特蕾西，就转去了麻省理工学院。当我们到达科技广场，敲门进去时，斯托曼正在跟别人交谈。

“我希望你们能不介意”，他一边说，一边把门打开，门开得很大，我和特蕾西勉强可以听清斯托曼的说话的对象的声音。她是莎拉，一个 20 多岁的年轻女子。

“我私自邀请了一个其它人一起参加我们的饭局”，斯托曼诚恳的说，他给了一个跟在帕罗奥多的饭店里见面时一样的狡猾的微笑。

实话说，我并没感到太多的意外。我几周前就听斯托曼的母亲说斯托曼有了个新的异性朋友。“事实上，他们上个月一起去了日本，理查德去接受 Takeda 奖”，李普曼当时是这么跟我说的。⁸

在去饭店的路上，我了解到了 Sarah 和理查德头一次见面的情景。有意思的事，这个情景让我觉得很熟悉。莎拉在构思一部小说时，听说了斯托曼，并对于他是什么样的性格觉得很好奇。她想在自己的作品中以斯托曼为原型设定一个人物形象，为了能了解这个形象的个性，她对斯托曼进行了一次采访。事情就是这样开始的，他们两个人从 2001 年年初开始约会了，她说。

“我非常景仰理查德建立起这个完整的政治运动的方式，而这个运动只是为了满足他个人的一个想法。”莎拉向我解释着她关注斯托曼的原因。

我的妻子立刻抛回一个问题：“所以，真正的问题是什么？”

她马上说：“粉碎孤独。”

晚餐的时候，我安排女士们与斯托曼多聊聊天，尝试发现他在过去一年中为什么会变成不那么强硬。不过我没有找到任何线索。斯托曼的眼光数次停留在我妻子的胸部，表现得比我印象中的样子更为卖俏，但他还是跟平时一样保持着敏感的状态。一度我的妻子加重语气说“但愿不会”，也只是得到了一个典型的斯托曼式的指责。

“我不想打断你，但是上帝并不存在！”斯托曼说。[RMS：我当时一定是个冷面笑匠。他可以把当成了一个聪明的家伙，但不能责备我。]

过了一会儿，晚餐结束后，莎拉离开了，斯托曼开始看上去放松了一些警惕。当我们路过附近一家书店时，他承认过去的一年让他改变了对他生活的愿景。“我曾经以为我会一辈子孤独，”他说，“不过很幸运的我错了。”

告别前，斯托曼给了我一张他的“好运卡”，一张写有斯托曼的地址、电话号码和爱好（“分享好书、好吃的和国外的音乐和舞蹈”）的名片，以便于我安排最后一次采访。

次日，我们几个一起去吃点心，斯托曼看上去比前一天晚上更沉浸于热恋中。回忆起他于克利尔舍堂的宿舍管理员有关“永生血清”好坏的争执，斯托曼表示希望有一天科学家们能研究出不朽的关键因子。“现在，我终于开始在生活中找到快乐，我希望我能一直快乐下去。”他说。

当我提到莎拉所说的“粉碎孤独”的评价，斯托曼没有看到在身体和精神层面的孤独和黑客层面的孤独之前的联系。“分享代码的推动力来自朋友情谊，而友情在更低的一个层面。”他说。不过，后来又提到这个话题时，斯托曼承认了孤独，或者说是持续的孤独的恐惧[RMS：在黑客与黑客之间，社区的层面上]在 GNU 工程最初的岁月里，是他做出的各种决定起到了一个非常主要作用。

他说：“我对计算机的迷恋不是任何别的东西的结果。如果我人缘更好，有更多追随我的女性，我对计算机的迷恋也不会有什么改变。当然，我感觉到自己没家，找到一个人又失去她，找到另一个又摧毁她，这对我一生都有很大的影响。我失去的是我们的宿舍，我摧毁的是人工智能实验室。没有家和社区的归属感所带来的不安全感非常巨大，它们迫使我去战斗，把它们抢回来。”

这次采访过后，我禁不住感受到我们这类人在情感上的相似。听到莎拉说斯托曼所吸引她的地方，听到斯托曼本人描述激发他投身自由软件运动的情感因素，我体会到了我自己撰写这本书的本质原因。从2000年7月开始，我就学会去赞扬理查德·斯托曼让人赞扬又让人排斥的双重性格。就像我的前辈伊本·莫格林，如果把这样的人格当成是一种附带现象，或是把它与整个自由软件运动隔离开来去分析，那是一种巨大的错误。在某种程度上，这两者是相辅相成，完全互补，不可分割的。

[RMS：威廉姆斯把他的感受进行了客观的描述，包括正面的和负面的，作为对我的分析，但是这些同时也是他自己对于外观、服从和商业成功的态度。]

我肯定不会每一位读者对于斯托曼都有相似程度的认同，在读完这本书后，有些人仍然会完全不认同斯托曼，但我敢断定有些人会认同他。很少人能像完全像理查德·M·斯托曼这样特立独行，但我真诚的希望有这样的一个完整的榜样，并且有GFDL的帮忙，其它人能够有类似的动力补充更多他们的观点，让这个榜样的形象更加地丰满起来。

Endnotes

¹参考《自由还是版权?》(2000年5月), <http://www.technologyreview.com/articles/stallman0500.asp>。

²RMS: 威廉姆斯在这里用的词是“商业”，属于用词不当，因为这个词的意思是“与交易有关”。其实不管以哪种方式出版电子书，只要是由一个公司出版的，就都是“商业”的。

³参考“Safari 在线科技书籍;订阅者协议:服务条款”,<http://my.safaribooksonline.com/termsofservice>。截止 2009 年 12 月,这些电子书需要非自由的阅读器软件来阅读,所以建议大家抵制这种电子书

⁴参考“The Open Publication License: Draft v1.0” (1999 年 7 月 8 日), <http://opencontent.org/openpub/>。

⁵参考“The GNU Free Documentation License: Version 1.1” (2000 年 3 月),<http://www.gnu.org/copyleft/fdl.html>。

⁶参考: <http://www.gnu.org/philosophy/license-list.html>。

⁷我会很乐意向任何愿意把本书“移植”到 Udanax 平台的人提供帮助, Udanax 是 Xanadu 的自由软件版本。请访问<http://www.udanax.com/>了解这项引人入胜的技术。

⁸另外,我没有找到有关 Takeda 基金会要嘉奖斯托曼、林纳斯·托瓦兹和坂村健的相关信息,似乎斯托曼前往日本去领取的是第一次也是最后一次所谓的“Techno-Entrepreneurial Achievement for Social/Economic Well-Being”。有关这个奖项和它 100 万美元的奖金的更多信息,可以参考 Takeda 的网站。<http://www.takea-foundation.jp>。

附录 A – 黑客的三层含义

要理解“黑客”一词内在的含义，需要追溯这个词的词源和这么多多年来的演化过程。

在新黑客词典这本在线的软件程序员专业术语词典中，对“Hack”一词列出了 9 种不同的含义，“Hacker”一词也有相对应的多种含义。同时，上面也引用了一段麻省理工学院的黑客费尔·阿格雷所撰写的短文，这篇文篇提醒读者不要被这个词给人的第一印象所迷惑。“Hack 这个词只有一种意思，”阿格雷争辩道，“这个意思非常微妙和有深度，难以用语言来描述。”理查德·斯托曼把这个词表述为一个短语：“顽皮的小聪明”。

不管是讨论这个词的狭义定义还是广义定义，大部分现代的黑客都会把这个词追溯到麻省理工学院，从 20 世纪 50 年代起，这个词就成为学生们互相交流时的使用的一句黑话。1990 年，麻省理工学院博物馆还发表了一篇有关黑客现象的论文。根据这篇论文的描述，50 年代时，这个学校的学生们使用“Hack”这个词的方式与现在的学生使用“Goof（混混）”一词的含义类似。在宿舍的窗户上挂上一个破飞机的行为就可以算是 Hack。但是，任何残忍的或恶意的行为就不能算是 Hack，比如怂恿另人去打破宿舍窗户或损坏学院的雕像。“Hack”一词内在含义是在不损害他人的前提下寻找创新点和乐趣的精神。

这种精神赋予了 Hack 这个词的动名词形式“Hacking”更多活力。在 20 世纪 50 年代，如果一个学生花上半天的时间打电话或拆装收

音机，这样行为就可以被称作是“Hacking”。而在现代，人们通常会用“Goof”一词的动词形式“Goofing”或“Goofing Off”来描述这类行为。

经过整个 50 年代的演化，“Hack”这个词的含义变得更加尖锐和反叛。50 年代的麻省理工学院校园里充满了竞争，而且黑客的出现，正是这种竞争文化的产物和延续。Goof 和 Pranks（喜欢恶作剧的人）突然成为学生们发泄多余精力的方式，他们甚至会跑去学校的教务处去做鬼脸，沉溺于各种稀奇古怪的想法和被繁重的研究生课程所遏制的行为。学院众多走廊和地下蒸气管道提供了无畏于紧闭的大门和“禁止进入”标识的学生们足够多探索的机会。学生们开始把这种不被规章制度允许的探索称为“地道黑客”。在地面上，校园的电话系统也提供了类似的探索机会。通过业余的实验和一些专业的研究，学生找到一些恶作剧的办法。学生们从相对古老的“地道黑客”活动中获得了灵感，很快就参与到了这种新型的“电话黑客”活动中。

强调创新游戏和无限制的探索活动，成为后来黑客活动的文化基础。20 世纪 60 年代出现了一些最早自称为是计算机黑客的人，他们来自麻省理工学院里一个起源于 50 年代末期、名为铁路科技模型俱乐部（Tech Model Railroad Club）的学生组织。这个俱乐部中有一个紧密的小团队叫做“信号与电源（S&P）委员会”，是这个俱乐部的电子电路系统的主要负责团队。这个系统是一个由继电器和开关组成的复杂系统，有点像当时学校电话的控制系统。俱乐部的成员只须通过电话拨打一些特定的号码就可以控制火车的运行。

负责建造和维护这套系统的电子工程爱好者们把这样的活动认为是与电话黑客活动一脉相成。他们不但接受了黑客这个概念，而且还把它变得更为完善。从 S&P 黑客的观点来看，如果某一段火车铁轨的控制电路中可以减少一个继电器，就意味着可以节省一个继电器供以后使用。黑客这个词的含义渐渐从“闲玩”的同义词演化成了在闲玩的同时去提高俱乐部的铁路系统的整体性能和效率。不久以后，S&P

委员会的成员就自豪的宣布，把改进和重构铁路的底层电路的活动称为“黑客活动”，而参与这样活动的人就是“黑客”。

复杂的电子电路对黑客们有着天然的亲和力，抛开那些无视紧锁的大门和“禁止进入”标识的传统的麻省理工学院的学生，新一代的黑客们很快就在校园中找到了一种新的玩具。这种被称为 TX-0 的机器，是最早进入商用市场的计算机之一。到了 20 世纪 50 年代末，整个 S&P 的团队都已经把兴趣转移到 TX-0 的控制室里，用他们的创新精神来折腾这个机器。从那时起，计算机程序设计领域开始进入“黑客”一词的词源。“黑客行为”不再是指焊接一些看起来很奇怪的电路板，而是更多的是与软件程序开发联系起来，还包括少许“正式”的软件开发方法或过程。它同样意味着提高效率和改进现有程序让它们运行的更快，因为只有这样才能节省更多的系统资源。从词源上来讲，这个词也指仅仅为了好玩或娱乐而编写一些没有实质性用途的程序。

“空间大战 (Spacewar)” 游戏是体现这种黑客定义的经典例子，这是世界上第一个在电脑上运行的交互式的视频游戏。早在 20 世纪 60 年代，麻省理工学院的黑客们就开发出了这个游戏，“空间大战”游戏诠释了所有传统黑客的定义：它既无聊又随机，除了可以让一群黑客们通宵达旦的娱乐外，没有什么实质性的用途。从软件的观点来说，它倒也算是程序开发技能上的一次伟大的创新。同时，它也是一个完全自由的软件。由于黑客们只是为了好玩而设计了它，所以他们觉得没有什么理由要去刻意保护这样的创作成果。因此，这个游戏可以在程序员之间广为分享。到了 20 世纪 60 年代末，“空间大战”已经成为全世界大型机程序员们最爱的一种消遣方式。

60 年代的计算机黑客与 50 年代的地道黑客和电话黑客的最大区别，就在于这种合作创新与公有的软件所有权。传统的黑客更崇尚个人或小团体的活动。地道黑客和电话黑客活动与校园里的情况紧密相关，他们这种不被规章制度所允许的活动天然的不鼓励通过公开的交流来分享新的发现。计算机黑客则不同，他们在科学领域中工作，这是一个完全基于协作和鼓励创新的氛围。黑客们与“官方”的计算机

科学家并不一定是最好的搭档，但是由于计算机技术的高速发展，这两种计算机程序员之间很快就形成了一种或多或少的合作共生关系。

黑客们大多都很讨厌官僚的制度。他们把阻碍自由访问计算机的安全系统看作是系统中的一个 Bug，只要有机会，就要想办法绕过它或把它修复。因此，在 1970 年时，绕过安全系统（但不是出于恶意的目的）被看成是一种典型的黑客行为，这样做不但获取了计算机的访问权限，还增加了饭后的谈资（受害者可能会说：“我觉得我被黑了。”）。但这并不是黑客活动的核心理念。黑客们在遇到一些安全系统阻碍时，会发挥自己的才智绕过它们。但是，如果可以选择的话，麻省理工学院人工智能实验室的黑客们更倾向于从一开始就不要构建这样阻碍，从而可以把自己的才智用到更有意义的地方。如果没有所谓的安全系统，也就没有人想着要去破坏它。

这是早期计算机黑客们惊人的技能的一种证明，很多后来的程序员，包括理查德·马修·斯托曼，都渴望能够做到这一点。到了 70 年代中后期，“黑客”这个词又有了更精确的内涵。在一般人看来，计算机黑客是指那些可以为了写代码而写代码的人。他们觉得这是对于编程技能的一种证明。“黑客”一词跟“艺术家”这个词很像，它的涵义超越了它字面上的意思。把一名程序员称为是黑客，是对他的一种尊称，而把自己称为是黑客则表明了极强的自信。不管是哪一种情况，随着计算机的普及，“计算机黑客”这种早年很少见的称谓也变得越来越常见。

对“黑客”的定义越来越精确，“计算机”黑客则还有一层超越了字面的涵义。麻省理工学院人工智能实验室的黑客们有一些共同的特质，包括，喜欢吃中国菜、讨厌抽烟、不沾烟、酒和其它一些可能会上瘾的药物。这些特质也渐渐成为一些人心目中对黑客群体的印象，这种印象对于新加入这个群体的人也产生了潜移默化的影响。然而，这些黑客文化在人工智能实验室黑客社区中渐渐的消失了。如今，大部分黑客在这些点上表现通常都与他周围的环境保持一致。

麻省理工学院、斯坦福和卡耐基梅隆大学这样的精英大学中的黑客谈论他们所敬仰的黑客特质时，会谈及有关黑客活动的伦理，并开始公开宣传“黑客伦理”：这些不成文的规定影响着黑客们的日常行为。在 1984 年出版的《黑客》一书中，作者史蒂文·利维在进行了很多研究和咨询以后，把黑客伦理定义为五种核心的黑客信条。

20 世纪 80 年代，计算机开始普及，破坏安全系统的行为也变多了。大部分破坏安全系统的行为都具有某些犯罪动机，实施这些破坏的内鬼其实根本就不能被称作是黑客。然而，警察和计算机管理员都会把破坏规则的行为看成是一种恶意的行为，他们把这种“入侵”计算机的行为追溯到了那些持有“不能伤害别人”理念的黑客身上。一些杂志的报道中，常常会从计算机管理员的视角出发，把“黑客行为”当成是破坏安全系统的同义词。尽管像《黑客》这样书用很多的篇幅去介绍黑客精神中的探索本质，并且弘扬黑客文化，但是对于大部分报纸记者和读者仍然把“计算机黑客”当成是“电子盗贼”的同义词。

到了 80 年代末，很多美国的青少年都开始有机会接触计算机。他们中的一部分有些叛逆的性格，受到杂志上歪曲了的“黑客”形象的误导，开始通过破坏计算机安全系统来发泄他们的对社会的的不满，就像一些人通过砸玻璃来发泄一样。他们自称为“黑客”，但是他们从来没了解过麻省理工学院的黑客们反对恶意行为的信条。一些年轻的程序员开始在一些错误的方向利用他们的计算机才能：制作和传播计算机病毒、恶作剧入侵计算机系统、故意让计算机宕机。“黑客”这个词开始渐渐成为一个反叛、无政府主义的象征，这样的形象吸引了更多不了解黑客本质的人开始走向错误的方向。

黑客们为了改变这种错误的认识已经自发的进行了近二十年的斗争。斯托曼为了解决这个问题，创造了一个新词来表达“破坏计算机安全系统的人”的意思：“骇客”，避免人们错误的使用“黑客”一词。而且，黑客和骇客这两个词的还是常常引起人们的误解。这两个词表达的含义还是有一些重叠的地方，并不是“黑客在这里，骇客在那里，

我们水火不容”。黑客行为和骇客行为只是同一种行为的不同层面上的属性，就像“年轻”和“高”是一个人的两个不同层面的属性。

其实，大部的黑客活动都与安全无关，所以不能称之为骇客活动。大部分骇客活动都是为了牟取利益或出于恶意，不会是出于“玩”的心态，所以不能称之为黑客活动。在少数情况下，可能会有个别的行为可以同时看成是黑客行为与骇客行为，但大部分情况下都不会是这样。黑客精神中包含对规则的不敬，但大部分黑客并不会去破坏规则。骇客活动从定义上来说就是不守规则，但是也并不一定就是恶意或有害的。在计算机安全领域中还区分“黑帽”骇客和“白帽”骇客，如果一个骇客的目的是搞破坏、恶意的，他就是一个“黑帽”；那些在安全系统中寻找漏洞是为了修复它的骇客，被称为“白帽”。

这种忌讳恶意行为的核心原则成为联系 21 世纪初的黑客与 20 世纪 50 年代的黑客的文化纽带。值得注意的是，在过去的 40 年中，计算机黑客概念的演化了很多。最初这个词的意思是指有点恶作剧的行为，比如发现一些隐蔽的地道，这层含义现在依然存在。2000 年秋天，麻省理工学院博物馆举行了一个专门向古老的黑客传统致敬的展览，名为黑客殿堂。这个展览上展出了很多 20 世纪 20 年代的老照片，其中有一张是有人愚弄警察的巡逻车的照片。1993 年，有学生向这种行为致敬，把一模一样的巡逻车、警灯放到了学院主楼的圆顶上。巡逻车的仿制车牌上写着 IHTFP，这是一个在麻省理工学院很流行的缩写词，包含着很多的含义。其中最引人注意的意思是始创于 20 世纪 50 年代麻省理工学院压力最大的时候的流行语，“我憎恨这个鬼地方 (I hate this fucing place)”。然而，在 1990 年，博物馆基于这个缩写词创作了一本期刊：《黑客、蠢举与恶作剧学报 (*The Journal of the Institute for Hacks, Tomfoolery, and Pranks*)》，这本期刊的主要内容有是关黑客的历史，它精妙描述了黑客活动的本质。

“在黑客的文化中，优雅、简洁的创新是被高度评价的，就像在纯科学的领域，”《波士顿环球报》的记者伦道夫·赖安在 1993 年的一篇警车展览相关的文章中写道，“黑客与一般校园中的那些喜欢恶作剧的

人在一些需要细致计划、工程化和使用手段的事件中的表现不太一样，他们本质上更为聪明和具有创造力。” Ryan 写道，“黑客的本质应该是好的，不能去搞破坏，应该是安全的，这是一条不成文的规则。事实上，黑客们有时也参与破坏他们自己的作品”。

主张把计算机黑客的文化限定在特定的伦理边界中是一个不错的想法，但却是不太可能的。尽管大部分软件黑客们渴望优雅和简单的精神气质，但是软件媒体很少提供回头的机会。摆脱警察的巡逻车要比破碎一个想法更为容易，尤其是一个出现的正是时候的想法。

“黑客”这个词曾经只是学生们一个含糊其辞的黑话，现在却成为了一个语言学上的一个台球，服从于政治的束缚和伦理上的差异。这也许就是为什么很多黑客和记者们喜欢使用这个词语的原因。没有人可以预料这个词以后会如何被使用，但是，我们可以决定我们自己如何去使用它。当你在谈论“破坏安全系统”时，请使用“骇客”这个词，而不要使用“黑客”，这样不但可以表现出你对斯托曼和本书中提到的其他黑客们的尊敬，也可以让黑客精神可以不断传承下去，造福更多计算机用户。

Appendix B – GNU Free Documentation License